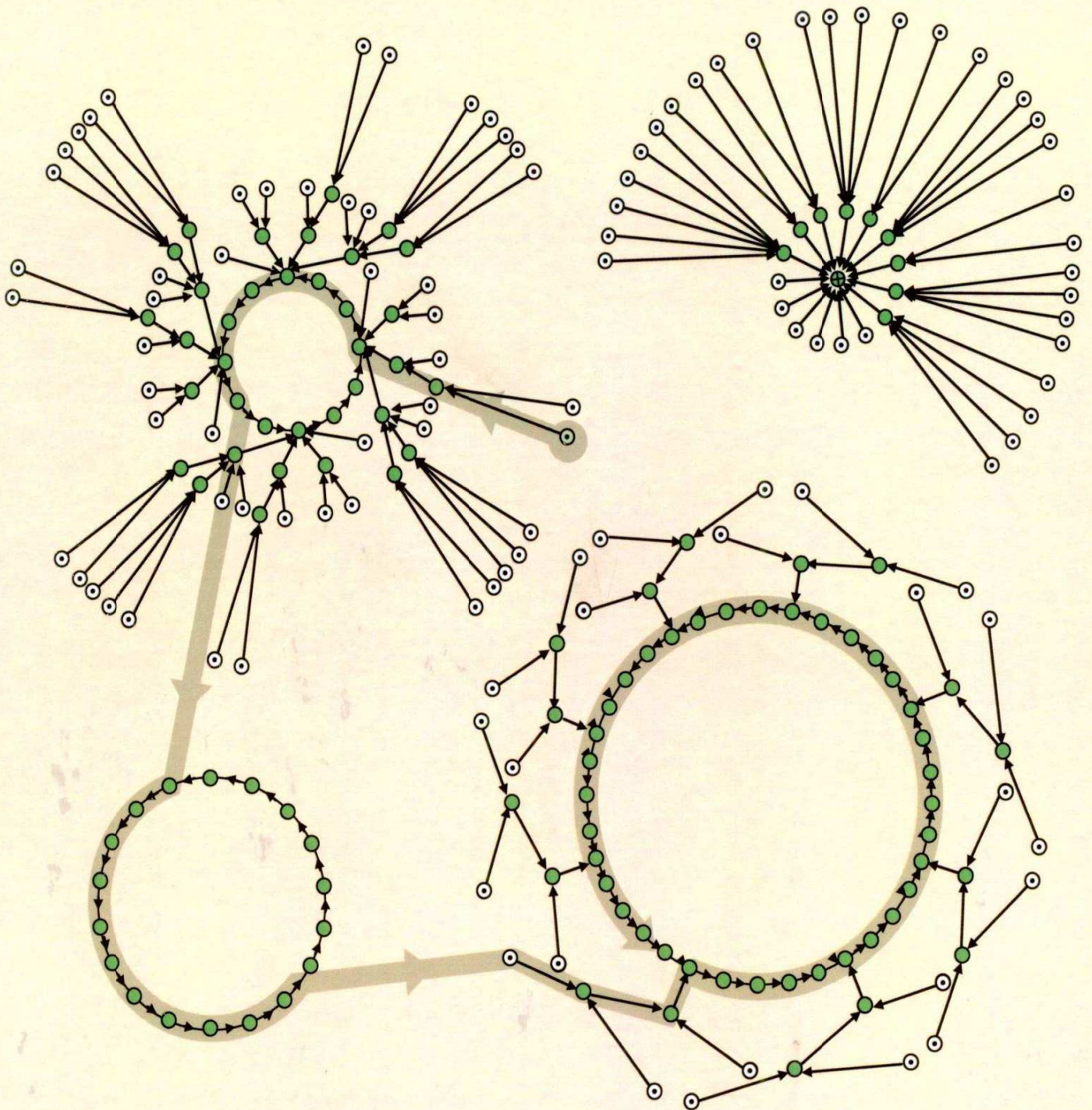


GRZEGORZ BOCEWICZ  
MODELE MULTIMODALNYCH  
PROCESÓW CYKLICZNYCH



POLITECHNIKA KOSZALIŃSKA

Grzegorz Bocewicz

*Modele multimodalnych  
procesów cyklicznych*

KOSZALIN 2013

MONOGRAFIA NR 254  
WYDZIAŁ ELEKTRONIKI I INFORMATYKI

ISSN 0239-7129  
ISBN 978-83-7365-316-0

Przewodniczący Uczelnianej Rady Wydawniczej  
*Miroslaw Maliński*

Recenzja  
*Jerzy Brzeziński*  
*Franciszek Seredyński*

Redakcja  
*Agnieszka Czajkowska*

Projekt okładki  
*Tadeusz Walczak*

© Copyright by Wydawnictwo Uczelniane Politechniki Koszalińskiej  
Koszalin, grudzień 2013

WYDAWNICTWO UCZELNIANE POLITECHNIKI KOSZALIŃSKIEJ  
75-620 Koszalin, ul. Raławicka 15-17

---

Koszalin 2013, wyd. I, ark. wyd. 18,3, format B5, nakład 120 egz.  
Druk: INTRO-DRUK, Koszalin

# Spis treści

Spis ważniejszych oznaczeń.....	5
Spis ważniejszych skrótów.....	11
Wprowadzenie.....	13
1. Modele problemów harmonogramowania cyklicznego .....	23
1.1. Problem harmonogramowania cyklicznego.....	24
1.1.1. Klasyfikacja.....	31
1.1.2. Decyzyjny problem harmonogramowania cyklicznego .....	40
1.2. Problemy harmonogramowania cyklicznego w systemach współbieżnych procesów .....	46
1.3. Problemy harmonogramowania cyklicznego w systemach współbieżnych procesów multimodalnych .....	64
1.4. Podsumowanie.....	71
2. System współbieżnych multimodalnych procesów cyklicznych SWMPC .....	73
2.1. Struktura SWMPC .....	74
2.1.1. Poziomy zachowań procesów współbieżnych.....	74
2.1.2. Wielopoziomowy model struktury SWMPC.....	82
2.2. Zachowanie SWMPC .....	93
2.2.1. Przestrzeń stanów .....	93
2.2.1.1. Stan .....	103
2.2.1.2. Funkcja przejścia.....	111
2.2.2. Digrafy zachowań .....	120
2.2.3. Wielopoziomowy model przestrzeni stanów.....	130
2.3. Modele deklaratywne problemów: analizy, syntezy i składania.....	132
2.4. Rozstrzygalność problemów analizy, syntezy i składania .....	135
2.5. Podsumowanie .....	145
3. Harmonogramowanie SWMPC.....	147
3.1. Metoda redukcji stanów początkowych.....	148
3.1.1. Wyznaczanie przebiegów cyklicznych o dowolnym okresie $\alpha_H$ .....	148
3.1.2. Wyznaczanie przebiegów cyklicznych o ustalonym okresie $\alpha_H$ .....	158
3.1.3. Złożoność obliczeniowa .....	161
3.2. Metoda generowania harmonogramów cyklicznych .....	163
3.2.1. Digraf pierwszeństwa operacji .....	165
3.2.2. Harmonogramowanie cykliczne procesów lokalnych .....	174
3.2.3. Harmonogramowanie cykliczne procesów multimodalnych.....	178
3.2.4. Złożoność obliczeniowa .....	188

3.3. Metoda wyznaczania parametrów struktury systemów cyklicznych.....	191
3.4. Podsumowanie.....	207
4. Synteza SWMPC.....	209
4.1. Problem składania struktur .....	210
4.1.1. Warunki składania struktur SWMPC.....	215
4.1.2. Reguła „zamka błyskawicznego” .....	222
4.2. Synteza struktur regularnych .....	235
4.3. Podsumowanie.....	243
5. Re-harmonogramowanie SWMPC.....	247
5.1. Problem wzajemnej osiągalności cyklicznych przebiegów ustalonych.....	248
5.2. Procedura wyznaczania stanów o wspólnej alokacji .....	253
5.3. Sieć wzajemnej osiągalności cyklicznych przebiegów ustalonych.....	260
5.4. Podsumowanie.....	261
6. Zakończenie .....	263
Literatura.....	269
Spis rysunków .....	279
Spis tabel.....	285
Skorowidz .....	287

## Spis ważniejszych oznaczeń

$\mathbb{N}$	- zbiór liczb naturalnych $\mathbb{N} = \{0, 1, 2, \dots\}$
$\mathbb{N}^+$	- zbiór liczb naturalnych dodatnich $\mathbb{N}^+ = \{1, 2, 3, \dots\}$
$\mathbb{C}$	- zbiór liczb całkowitych $\mathbb{C} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
$\mathbb{W}$	- zbiór liczb wymiernych $\mathbb{W} = \left\{\frac{a}{b} \mid a \in \mathbb{C}, b \in \mathbb{N}^+\right\}$
$\mathbb{NW}$	- zbiór liczb niewymiernych
$\mathbb{R}$	- zbiór liczb rzeczywistych $\mathbb{R} = \mathbb{W} \cup \mathbb{NW}$
$\mathbb{R}^+$	- zbiór liczb rzeczywistych dodatnich $\mathbb{R}^+ = \{a \mid a \in \mathbb{R}, a > 0\}$
$A \times B$	- iloczyn kartezjański zbiorów $A$ i $B$
$ A $	- moc (liczność) zbioru $A$
$x \wedge y$	- konkatencja sekwencji $x$ i $y$ , gdy: $x = (x_1, \dots, x_n)$ a $y = (y_1, \dots, y_m)$ , to $x \wedge y = (x_1, \dots, x_n, y_1, \dots, y_m)$
$A \cup B$	- suma mnogościowa zbiorów $A, B$
$A \cap B$	- iloczyn mnogościowy zbiorów $A, B$
$A \setminus B$	- różnica mnogościowa zbiorów $A, B$
$G_1 \dot{\cup} G_2$	- suma digrafów: $G_1 \dot{\cup} G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ gdy $G_1 = (V_1, E_1)$ i $G_2 = (V_2, E_2)$
$\text{mod}(a, b)$	- dzielenie modulo $a$ i $b$ : $r = \text{mod}(a, b)$ gdy $a = k \cdot b + r$ i $0 \leq r <  b $ , $a, b, c, k \in \mathbb{C}$ ,
$\text{modl}(a, b)$	- funkcja zwracająca wartość $r = \text{modl}(a, b)$ gdy $a = k \cdot b + r$ i $0 < r \leq  b $ , $a, b, c, k \in \mathbb{C}$ , (zmodyfikowane dzielenie modulo),
$\lceil a \rceil_B$	- górne całkowitoliczbowe ograniczenie liczby $a$ względem zbioru $B$ : $\lceil a \rceil_B = \min\{k \in B : k \geq a\}$
$O$	- zbiór operacji w przypadku problemu BCSP; zbiór sekwencji operacji w przypadku SWMPC
$o_i$	- $i$ -ta operacja w problemie BCSP
$t_i$	- czas wykonania operacji $o_i$
$x_i(k)$	- moment rozpoczęcia $k$ -tego wykonania operacji $o_i$
$x_i$	- moment rozpoczęcia 0-wego wykonania operacji $o_i$ : $x_i = x_i(0)$
$T$	- sekwencja czasów wykonania operacji w przypadku problemu BCSP; zbiór sekwencji czasów wykonania operacji w przypadku SWMPC
$X$	- harmonogram cykliczny dla problemów BCSP, CJP, CFP, COP, CPSP, CRJP, CRPSP, CRFP,
$X'$	- sekwencja momentów rozpoczęcia $x_i$ dla BCSP; $x_{i,j}$ dla CJP, CFP, COP, CPSP, CRJP, CRPSP, CRFP,
$M$	- zbiór maszyn (dla problemu CJP)
$Z$	- zbiór zadań (dla problemu CJP): $Z = \{Z_1, \dots, Z_i, \dots, Z_n\}$
$O_i$	- sekwencja operacji zadania $Z_i$ : $O_i = (o_{i,1}, \dots, o_{i,j}, \dots, o_{i,l(i)})$ dla CJP; sekwencja operacji procesu $P_i$ w SWPC
$o_{i,j}$	- $j$ -ta operacja zadania $Z_i$
$t_{i,j}$	- czas wykonania operacji $o_{i,j}$ dla CJP; czas wykonania operacji $o_{i,j}^h$ , $h =$ $1 \dots ls(i)$ dla SWMPC

$x_{i,j}(k)$	- moment rozpoczęcia $k$ -tego wykonania operacji $o_{i,j}$
$x_{i,j}$	- moment rozpoczęcia 0-wego wykonania operacji $o_{i,j}$
$T_i$	- sekwencja czasów wykonania operacji $t_{i,j}$ dla CJP; sekwencja czasów wykonania operacji $i$ -tego procesu lokalnego $P_i$ dla SWMPC
$SC^{lp}$	- struktura SWMPC
$lp$	- liczba poziomów zachowań procesów multimodalnych SWMPC
$SL$	- struktura poziomu zachowań procesów lokalnych
$SM^l$	- struktura $l$ -tego poziomu zachowań procesów multimodalnych
$R$	- zbiór zasobów SWMPC (SWPC): $R = \{R_k \mid k = 1 \dots lk\}$
$lk$	- liczba zasobów SWMPC (SWPC)
$C(R_k)$	- liczba dostępnych jednostek zasobu $R_k$ (tzw. pojemność zasobu $C(R_k) \in \mathbb{N}^+$ )
$P$	- zbiór procesów lokalnych SWMPC (SWPC): $P = \{P_i \mid i = 1 \dots ln\}$
$ln$	- liczba procesów lokalnych SWMPC (SWPC)
$p_i$	- marszruta $i$ -tego procesu lokalnego $P_i$
$U$	- zbiór marszrut procesów lokalnych
$P_i^h$	- $h$ -ty potok $i$ -tego procesu lokalnego $P_i$
$ls(i)$	- liczba potoków $P_i^h$ $i$ -tego procesu lokalnego $P_i$
$O_i^h$	- sekwencja operacji realizowanych w ramach potoku $P_i^h$ : $O_i^h = (o_{i,1}^h, \dots, o_{i,j}^h, \dots, o_{i,lr(i)}^h)$
$o_{i,j}^h$	- $j$ -ta operacja potoku $P_i^h$
$lr(i)$	- liczba operacji (długość marszruty $p_i$ ) potoku $P_i^h$ $i$ -tego procesu lokalnego $P_i$
$x_{i,j}^h(k)$	- moment rozpoczęcia operacji $o_{i,j}^h$ w $k$ -tym cyklu systemu
$x_{i,j}^h$	- moment rozpoczęcia operacji $o_{i,j}^h$ dla 0-wego cyklu systemu: $x_{i,j}^h = x_{i,j}^h(0)$
$\omega_{i,j}^h$	- liczba jednostek zasobu niezbędnych do wykonania operacji $o_{i,j}^h$
$m^l P$	- zbiór procesów multimodalnych w SWMPC $l$ -tego poziomu zachowań: $m^l P = \{m^l P_i \mid i = 1 \dots lw(l)\}$
$lw(l)$	- liczba procesów multimodalnych $m^l P_i$ należących do $l$ -tego poziomu zachowań,
$m^l p_i$	- marszruta $i$ -tego procesu multimodalnego $m^l P_i$
$m^l U$	- zbiór marszrut procesów multimodalnych
$m^l P_i^h$	- $h$ -ty potok $i$ -tego procesu multimodalnego $m^l P_i$
$lsm(i, l)$	- liczba potoków $m^l P_i^h$ procesu multimodalnego $m^l P_i$
$m^l O_i^h$	- sekwencja operacji realizowanych w ramach potoku: $m^l P_i^h$ : $m^l O_i^h = (m^l o_{i,1}^h, \dots, m^l o_{i,j}^h, \dots, m^l o_{i,lm(i,l)}^h)$
$m^l o_{i,j}^h$	- $j$ -ta operacja potoku $m^l P_i^h$
$lm(i, l)$	- liczba operacji (długość marszruty $m^l p_i$ ) potoku $m^l P_i^h$ procesu multimodalnego $m^l P_i$
$lm(i)$	- liczba operacji (długość marszruty $mp_i$ ) procesu multimodalnego dla SWMPC z jednym poziomem zachowań ( $l = 1$ )
$m^l x_{i,j}^h(k)$	- moment rozpoczęcia operacji $m^l o_{i,j}^h$ w $k$ -tym cyklu systemu

- $m^l x_{i,j}^h$  - moment rozpoczęcia operacji  $m^l o_{i,j}^h$  w 0-wym cyklu pracy systemu  
 $m^l \omega_{i,j}^h$  - liczba jednostek zasobu niezbędnych do wykonania operacji  $m^l o_{i,j}^h$   
 $m^l \mu_{i,j}$  - potok poziomemu  $(l - 1)$  wymagany do wykonania operacji  $m^l o_{i,j}^h$   
 $m^l T_i$  - sekwencja czasów wykonania operacji  $i$ -tego procesu multimodalnego  $m^l P_i$ :  

$$m^l T_i = (m^l t_{i,1}, \dots, m^l t_{i,j}, \dots, m^l t_{i,lm(i,l)})$$
 $m^l t_{i,j}$  - czas wykonania operacji  $m^l o_{i,j}^h$ , dla  $h = 1 \dots lsm(i, l)$   
 $H^l$  - zbiór wszystkich potoków  $l$ -tego poziomu zachowań:  

$$H^l = \{m^l P_i^h \mid i = 1 \dots lw(l), h = 1 \dots lsm(i, l)\}$$
 $LP$  - liczba wszystkich potoków realizowanych w strukturze  $SC^{lp}$   
 $g$  - gęstość struktury  
 $\Delta t$  - zwłoka czasowa związana ze zmianą potoków na zasobach  
 $\sigma_k^0$  - reguła priorytetowania dla zasobu  $R_k$ , określająca kolejność dostępu procesów lokalnych do tego zasobu:  

$$\sigma_k^0 = (s_{k,1}^0, \dots, s_{k,d}^0, \dots, s_{k,lh(k,0)}^0)$$
 $\sigma_k^l$  - reguła priorytetowania dla zasobu  $R_k$ , określająca kolejność dostępu do tego zasobu procesów multimodalnych  $l$ -tego poziomu zachowań:  

$$\sigma_k^l = (s_{k,1}^l, \dots, s_{k,d}^l, \dots, s_{k,lh(k,0)}^l), \quad l = 1 \dots lp$$
 $lh(k, l)$  - długość reguły  $\sigma_k^l$ ,  $l = 0 \dots lp$   
 $\theta^0$  - zbiór reguł priorytetowania procesów lokalnych  
 $\theta^l$  - zbiór reguł priorytetowania procesów multimodalnych  $l$ -tego poziomu zachowań  
 $\psi_i^0$  - liczba wykonań każdego potoku procesu  $P_i$  w ramach jednego cyklu systemu,  
 $\psi_i^l$  - liczba wykonań każdego potoku procesu  $m^l P_i$  w ramach jednego cyklu systemu,  $l = 1 \dots lp$   
 $\Psi^0$  - sekwencja liczby wzajemnych wykonań procesów lokalnych:  

$$\Psi^0 = (\psi_1^0, \dots, \psi_i^0, \dots, \psi_{ln}^0),$$
 $\Psi^l$  - sekwencja liczby wzajemnych wykonań procesów multimodalnych  $l$ -tego poziomu zachowań:  

$$\Psi^l = (\psi_1^l, \dots, \psi_i^l, \dots, \psi_{lw(l)}^l), \quad l = 1 \dots lp$$
 $R(k)$  - zbiór wszystkich zasobów, będących poprzednikiem zasobu  $R_k$  w marszrutach procesów  $P_i$   
 $\alpha$  - okres harmonogramu  $X$  dla problemu BSCP, CFP, COP, CPSP, CRJP, CRPSP, CRFP; okres realizacji procesów lokalnych SWMPC (SWPC)  
 $m^l \alpha$  - okres realizacji procesów multimodalnych  $l$ -tego poziomu zachowań  
 $\mathcal{P}$  - przestrzeń stanów, digraf:  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$   
 $\mathbb{S}^{lp}$  - zbiór stanów dopuszczalnych SWMPC  
 $\mathbb{E}^{lp}$  - zbiór łuków określających dopuszczalne przejścia między stanami zbioru  $\mathbb{S}^{lp}$  (zgodnie z funkcją przejścia  $\delta^{lp}$ )  
 $S^r$  - stan SWMPC:  

$$S^r = \left( \left( (S^{lr}, m^1 S^r), \dots, m^l S^r, \dots \right), m^{lp} S^r \right)$$
 $Sl^r$  - stan procesów lokalnych:



$$Sl^r = (A^r, Z^r, Q^r)$$

- $A^r$  - sekwencja alokacji potoków procesów lokalnych w stanie  $Sl^r$   
 $a_{i,k}^{r,h}$  - alokacja potoku  $P_i^h$  na zasobie  $R_k$  w stanie w stanie  $Sl^r$   
 $Z^r$  - sekwencja semaforów dla procesów lokalnych w stanie  $Sl^r$   
 $z_k^r$  - semafor zasobu  $R_k$  w stanie  $Sl^r$   
 $Y^0$  - sekwencja praw alokacji dla procesów lokalnych w stanie  $Sl^r$   
 $Q^r$  - sekwencja indeksów semaforów dla procesów lokalnych w stanie  $Sl^r$   
 $q_k^r$  - indeks semafora  $z_k^r$  w regule priorytetowania  $\sigma_k^0$  w stanie  $Sl^r$   
 $m^l S^r$  - stan procesów multimodalnych  $l$ -tego poziomu zachowań:  

$$m^l S^r = (m^l A^r, m^l Z^r, m^l Q^r)$$
  
 $m^l A^r$  - sekwencja alokacji procesów multimodalnych  $l$ -tego poziomu zachowań w stanie  $m^l S^r$   
 $m^l a_{i,k}^{r,h}$  - alokacja potoku  $m^l P_i^h$  na zasobie  $R_k$  w stanie w stanie w stanie  $m^l S^r$   
 $m^l Z^r$  - sekwencja semaforów dla procesów multimodalnych  $l$ -tego poziomu zachowań w stanie  $m^l S^r$   
 $m^l z_k^r$  - semafor zasobu  $R_k$  w stanie  $m^l S^r$   
 $m^l Q^r$  - sekwencja indeksów semaforów dla procesów multimodalnych  $l$ -tego poziomu zachowań w stanie  $m^l S^r$   
 $m^l q_k^r$  - indeks semafora  $m^l z_k^r$  w regule priorytetowania  $\sigma_k^0$  w stanie  $m^l S^r$   
 $S^e \rightarrow S^f$  - stan  $S^f$  osiągalny bezpośrednio ze stanu  $S^e$   
 $\delta^{lp}$  - funkcja przejścia między dwoma stanami dopuszczalnymi:  

$$S^f = \delta^{lp}(S^e) \text{ gdy } S^e \rightarrow S^f$$
  
 $PA^r$  - zbiór rodziców stanu  $S^r$   
 $CH^r$  - zbiór potomków stanu  $S^r$   
 $D_C$  - cykliczny przebieg ustalony:  $D_C = (S^{d_1}, \dots, S^{d_i}, S^{d_{i+1}}, \dots, S^{d_{ld}})$ ,  
 $S^{d_1} = \delta^{lp}(S^{d_{ld}})$ ,  $S^{d_{i+1}} = \delta^{lp}(S^{d_i})$  dla  $i = 1 \dots (ld - 1)$   
 $ld$  - liczba stanów przebiegu SWMPC  
 $V(D_C)$  - zbiór stanów cyklicznego przebiegu ustalonego  $D_C$   
 $DC$  - zbiór wszystkich cyklicznych przebiegów ustalonych przestrzeni  $\mathcal{P}$   
 $D_T$  - przebieg przejściowy  
 $V(D_T)$  - zbiór stanów przebiegu przejściowego ustalonego  $D_T$   
 $DT$  - zbiór wszystkich przebiegów przejściowych  $D_T$  przestrzeni  $\mathcal{P}$   
 $S^*$  - stan blokady  
 $W(D_C)$  - digraf zachowań typu wir, prowadzący do cyklicznego przebiegu ustalonego  $D_C$   
 $Tr(S^*)$  - digraf zachowań typu drzewo, prowadzący do stanu blokady  $S^*$   
 $\mathcal{P}^l$  - przestrzeń stanów  $l$ -tego poziomu zachowań:  
 $G$  - digraf pierwszeństwa operacji  $G = (V_G, E_G)$  procesów lokalnych  
 $V_G$  - zbiór wierzchołków digrafu  $G$   
 $v_{i,j}^h(k)$  - wierzchołek digrafu pierwszeństwa operacji  $G$   
 $E_G$  - zbiór łuków digrafu  $G$   
 $X_{DC}$  - harmonogram cykliczny realizacji procesów lokalnych:  

$$X_{DC} = (X'_{DC}, \alpha)$$
  
 $X'_{DC}$  - zbiór momentów rozpoczęcia  $x_{i,j}^h$  operacji  $\sigma_{i,j}^h$

- $G^l$  - digraf pierwszeństwa operacji  $G^l = (\mathbb{V}_G^l, \mathbb{E}_G^l)$  procesów multimodalnych  $l$ -tego poziomu zachowań
- $\mathbb{V}_G^l$  - zbiór wierzchołków digrafu  $G^l$
- $m^l v_{i,j}^h(k)$  - wierzchołek digrafu pierwszeństwa operacji  $G^l$
- $\mathbb{E}_G^l$  - zbiór łuków digrafu  $G^l$
- $m^l X_{DC}$  - harmonogram cykliczny realizacji procesów multimodalnych  $l$ -tego poziomu zachowań:
- $$m^l X_{DC} = (m^l X'_{DC}, m^l \alpha)$$
- $m^l X'_{DC}$  - zbiór momentów rozpoczęcia  $m^l x_{i,j}^h$  operacji  $m^l o_{i,j}^h$
- $X_{i,j}^l$  - zbiór wartości zmiennej  $m^l x_{i,j}^h$
- $X^{lp}$  - harmonogram cykliczny SWMPC:
- $$X^{lp} = \left( \left( (X'_{DC}, \alpha), (m^1 X'_{DC}, m^1 \alpha), \dots \right) \dots (m^{lp} X'_{DC}, m^{lp} \alpha) \right)$$
- $SC^*$  - zbiór podstruktur  $SC^{lp}$ :
- $$SC^* = \{SC_1^{lp}, \dots, SC_i^{lp}, \dots, SC_{lc}^{lp}\}$$
- $SC_i^{lp}$  -  $i$ -ta podstruktura struktury  $SC^{lp}$
- $X_i^{lp}$  - harmonogram cykliczny osiągalny w SWMPC o strukturze  $SC_i^{lp}$
- $Rp_i$  - zbiór zasobów podstruktury  $SC_i^{lp}$
- $Pp_i$  - zbiór procesów lokalnych podstruktury  $SC_i^{lp}$
- $Up_i$  - zbiór marszrut procesów  $Pp_i$
- $Op_i$  - zbiór sekwencji operacji procesów  $Pp_i$
- $Tp_i$  - zbiór sekwencji czasów wykonania operacji procesów  $Pp_i$
- $m^l Pp_i$  - zbiór fragmentów procesów multimodalnych  $l$ -tego poziomu zachowań podstruktury  $SC_i^{lp}$
- $m^l P_i(a, b)$  - fragment procesu  $m^l P_i$  związany z wykonaniem operacji:  $m^l o_{i,a}^h, m^l o_{i,a+1}^h, \dots, m^l o_{i,b}^h$  wszystkich należących do tego procesu potoków  $m^l P_i^h$  ( $h = 1 \dots lsm(i, l)$ )
- $m^l Up_i$  - zbiór marszrut fragmentów procesów  $m^l Pp_i$
- $m^l Op_i$  - zbiór sekwencji operacji fragmentów procesów  $m^l Pp_i$
- $m^l Tp_i$  - zbiór sekwencji czasów wykonania operacji procesów  $m^l Pp_i$
- $\oplus$  - operator składania podstruktur  $SC_i^{lp}$
- $\oplus$  - operator składania podstruktur  $SC_i^{lp}$  wzdłuż łańcucha zasobów wykorzystanych w procesach o tym samym zwrocie
- $\ddot{\oplus}$  - operator składania podstruktur  $SC_i^{lp}$  zgodnie z regułą „zamka błyskawicznego”
- $\cup$  - operator łączenia harmonogramów cyklicznych,
- $PS$  - problem spełniania ograniczeń:  $PS = (V, D, C)$
- $V$  - zbiór zmiennych decyzyjnych
- $D$  - rodzina dziedzin zmiennych decyzyjnych
- $C$  - zbiór ograniczeń
- $\cup$  - operator łączenia problemów spełniania ograniczeń względem ograniczeń dodatkowych  $C_D$ :
- $$PS_c = PS_a \cup PS_b = ((V_a \cup V_b, D_a \cup D_b), C_a \cup C_b \cup C_D)$$
- gdy

$$PS_a = ((V_a, D_a), C_a), PS_b = ((V_b, D_b), C_b)$$

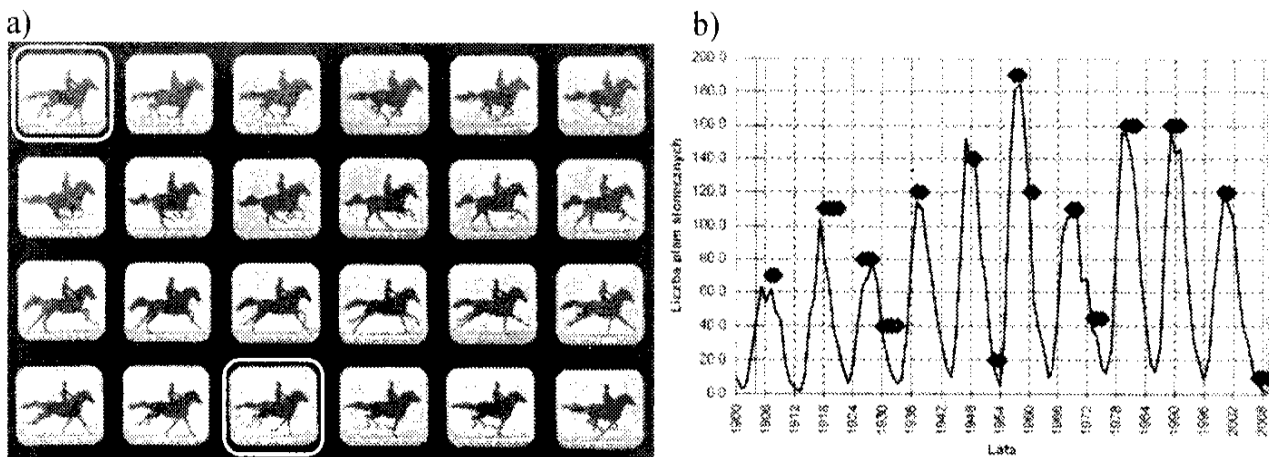
- $(i)SC^{lp}$  - podstruktura izomorficzna struktury  $SC^{lp}$
- $\overset{\cdot\cdot}{\oplus}$  - operator składania podstruktur izomorficznych  $(i)SC^{lp}$  zgodnie z regułą „zamka błyskawicznego”
- $\overset{\cdot\cdot}{\oplus}$  - operator składania podstruktur izomorficznych  $(i)SC^{lp}$  dla  $i = 1 \dots lc$ :
- $$\overset{\cdot\cdot}{\oplus}_{i=1}^{lc} ((i)SC^1) = (1)SC^1 \overset{\cdot\cdot}{\oplus} \dots \overset{\cdot\cdot}{\oplus} (i)SC^1 \overset{\cdot\cdot}{\oplus} \dots \overset{\cdot\cdot}{\oplus} (lc)SC^1$$
- $(i)X^{lp}$  - harmonogram cykliczny generowany przez podstrukturę  $(i)SC^{lp}$
- $\overset{\cdot\cdot}{\cup}$  - operator łączenia harmonogramów  $(i)X^{lp}$  dla  $i = 1 \dots lc$ :
- $$\overset{\cdot\cdot}{\cup}_{i=1}^{lc} ((i)X^1) = (1)X^1 \overset{\cdot\cdot}{\cup} (2)X^1 \overset{\cdot\cdot}{\cup} \dots \overset{\cdot\cdot}{\cup} (lc)X^1$$
- $\mathcal{A}$  - przestrzeń alokacji, digraf:  $\mathcal{A} = (\mathbb{A}, \mathbb{E}^{\mathcal{A}})$
- $\mathbb{A}$  - zbiór alokacji stanów  $Sl^r$
- $\mathbb{E}^{\mathcal{A}}$  - zbiór łuków digrafu  $\mathcal{A}$

## Spis ważniejszych skrótów

BCSP	<i>Basic Cyclic Scheduling Problem</i> , elementarny problem harmonogramowania cyklicznego
AOA	<i>Activity-On-Arrow</i> , czynność na łuku
CPM	<i>Critical Path Method</i> , metoda ścieżki krytycznej
PERT	<i>Program Evaluation and Review Technique</i>
CJP	<i>Cyclic Jobshop Problem</i> , cykliczny problem gniazdowy, tzn. problem, w którym zadaniom przypisane mogą być różne sekwencje operacji wykorzystujące różne zestawy maszyn
CRJP	<i>Cyclic Robotic Jobshop Problem</i> , problem gniazdowy cyklicznego zrobotyzowanego gniazda produkcyjnego, tzn. cykliczny problem gniazdowy, w którym uwzględniono transport międzyoperacyjny
CFP	<i>Cyclic Flowshop Problem</i> , cykliczny problem przepływowy, tzn. problem, w którym wszystkim zadaniom przypisane są takie same sekwencje operacji wykorzystujące wszystkie dostępne maszyny
CRFP	<i>Cyclic Robotic Flowshop Problem</i> , problem przepływowy cyklicznego zrobotyzowanego gniazda produkcyjnego, tzn. cykliczny problem przepływowy, w którym uwzględniono transport międzyoperacyjny
COP	<i>Cyclic Open Shop</i> , cykliczny problem otwarty, tzn. problem, w którym kolejność realizacji operacji w poszczególnych zadaniach jest dowolna
CROP	<i>Cyclic Robotic Open Shop</i> , problem otwarty cyklicznego zrobotyzowanego gniazda produkcyjnego, tzn. cykliczny problem otwarty, w którym uwzględniono transport międzyoperacyjny
CPSP	<i>Cyclic PERT-Shop Problem</i> , problem cyklicznego harmonogramowania przedsięwzięć
CRPSP	<i>Cyclic Robotic PERT-Shop Problem</i> , problem cyklicznego harmonogramowania przedsięwzięć produkcyjnych w zrobotyzowanym gnieździe produkcyjnym
DSZ	Dyskretny System Zdarzeniowy
ESP	Elastyczny System Produkcyjny
FIFO	<i>First In First Out</i> , pierwszy na wejściu, pierwszy na wyjściu
SWP	System Współbieżnych Procesów
SWPC	System Współbieżnych Procesów Cyklicznych
SWMPC	System Współbieżnych Multimodalnych Procesów Cyklicznych
MIP	<i>Mixed Integer Programming</i>
MILP	<i>Mixed Integer Linear Programming</i>
PSO	Problem Spełniania Ograniczeń
u.j.c.	umowna jednostka czasu

## Wprowadzenie

Procesy występujące w przyrodzie (intuicyjnie rozumiane jako ciąg występujących po sobie zdarzeń) uznawane są za cykliczne, w przypadku gdy tworzące je zdarzenia powtarzane są wielokrotnie w stałych odstępach czasu. Życie codzienne dostarcza wielu przykładów takich procesów. Należą do nich cykle dnia i nocy, zmiany faz Księżyca, cykliczne zmiany pór roku, przyływy i odpływy morza, itp. Wśród wielu innych przykładów warto wymienić sposoby przemieszczania się ssaków, np. konia – rys. W.1a (obejmujące trucht, kłus, galop), występowanie plam na Słońcu rys. W.1b, podział komórek bakterii czy chociażby przebieg zajęć w roku akademickim.

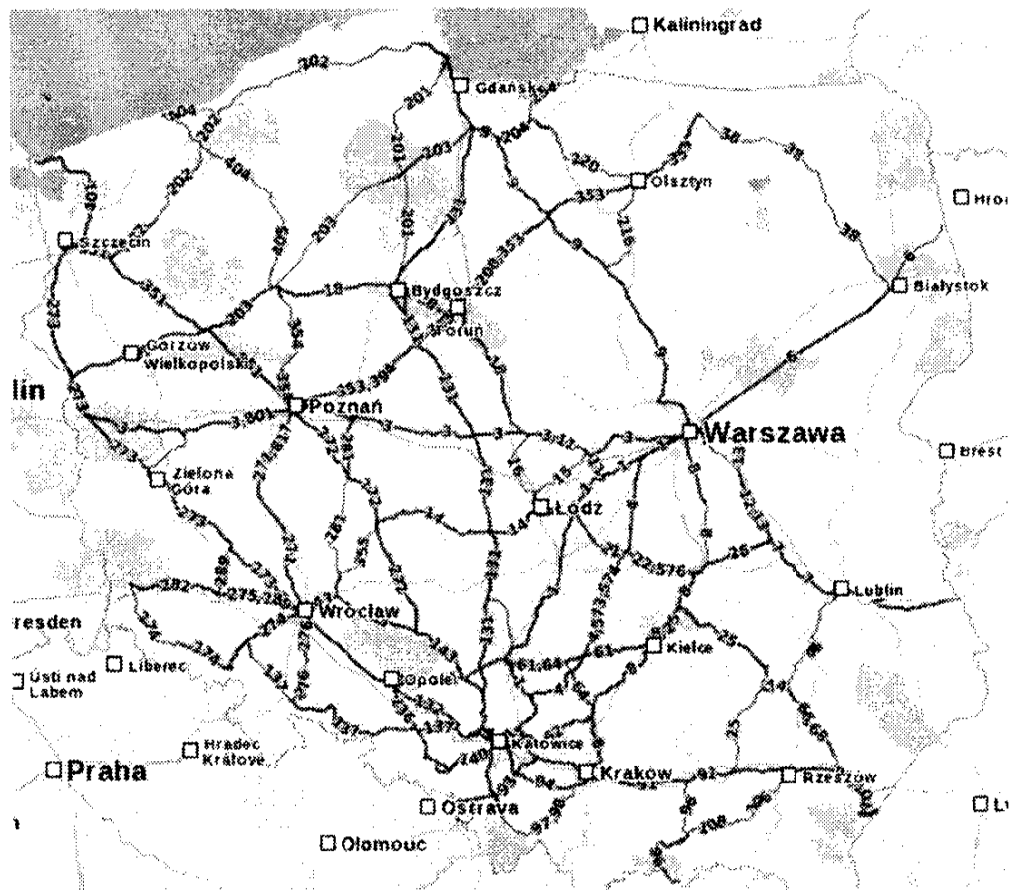


Rys. W.1. Galop konny (<http://whitmanhansonphoto.wordpress.com>) a), liczba plam słonecznych oraz występowanie recesji w USA (od roku 1900) (<ftp://ftp.ngdc.noaa.gov>) b)

Cechą charakterystyczną tego typu procesów jest ich przewidywalność. Przykładowo znając cykl pylenia roślin można przewidzieć termin odpowiednich działań profilaktycznych (prewencyjnych) związanych z zakupem stosownych środków przeciwalergicznymi. Innym tego typu przykładem jest metoda poszukiwania planet pozasłonecznych oparta na tzw. tranzycie polegającym na przejściu ciała niebieskiego przez tarczę gwiazdy. Cykliczne osłabianie jasności gwiazdy może świadczyć o krążącej wokół niej planecie.

Procesy cykliczne mogą być postrzegane jako wyodrębnione z otoczenia rytmicznie występujące realizacje powtarzających się zdarzeń (stanów) lub też jako złożenia współbieżnie realizowanych (tzn. wykorzystujących wspólne zasoby) pojedynczych procesów cyklicznych. Procesy cykliczne przebiegające w ramach współbieżnie realizowanych procesów mogą być realizowane bez zakłóceń (np. cotygodniowa, bezkolejowa realizacja zajęć akademickich dwóch różnych wydziałów we wspólnej auli wykładowej) lub też prowadzić do ich wzajemnej blokady.

Typowy przykład rozważanej dalej klasy procesów współbieżnych, tzw. Systemów Współbieżnych Procesów Cyklicznych (SWPC) (Banaszak et al. 2008) stanowi system kolei pasażerskich PKP. Pojedyncze procesy cykliczne występujące w tym systemie stanowią pociągi zatrzymujące się, według zadanego rozkładu jazdy, na stacjach rozmieszczonych wzdłuż marszruty wytyczonej w dostępnej sieci kolejowej PKP (rys. W.2).



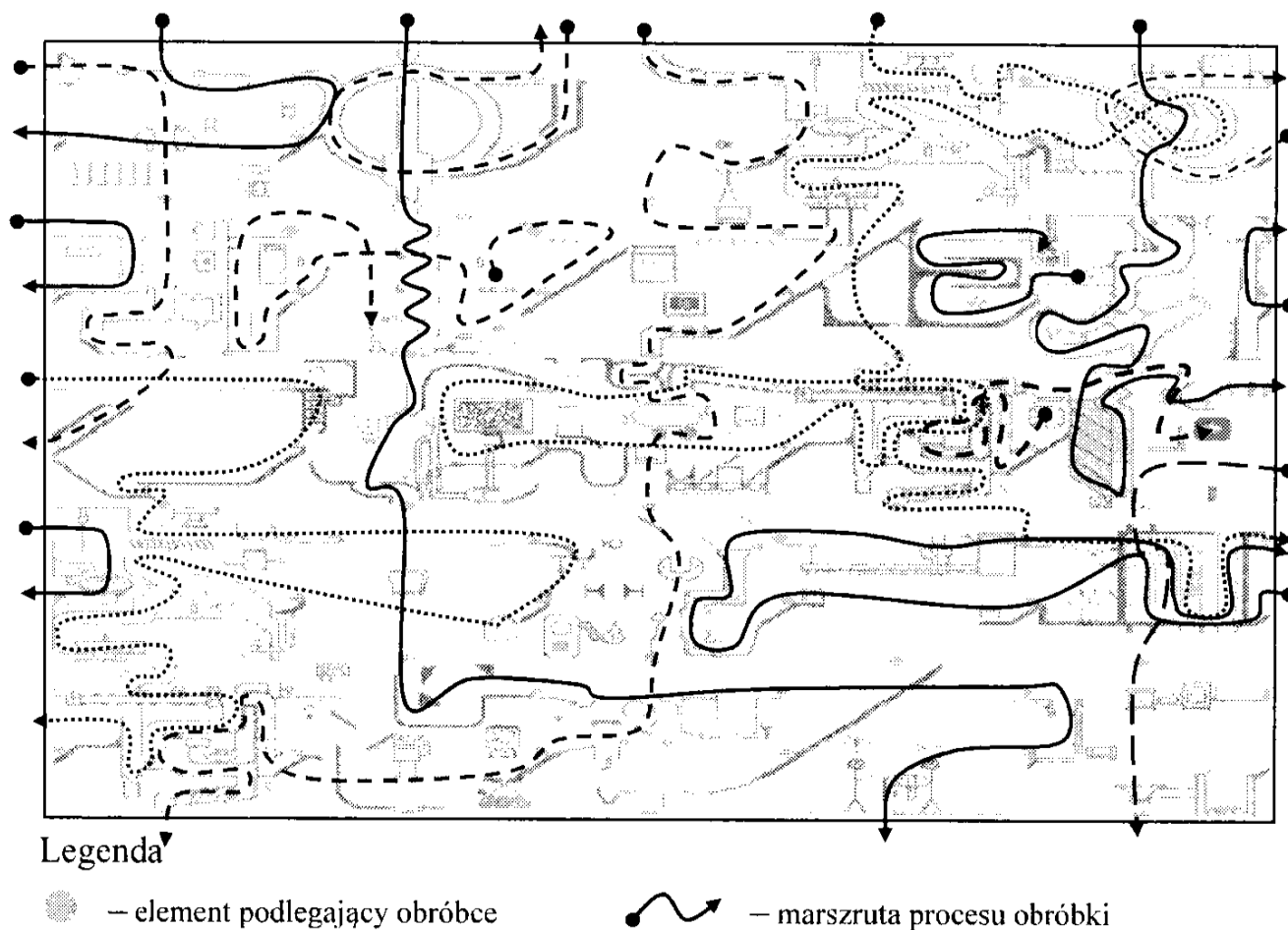
Rys. W.2. Sieć kolejowa PKP (<http://rozklad.sitkol.pl>)

Przyjmując, że stacje i niektóre fragmenty torowisk są współdzielone przez tego typu procesy (pociągi) łatwo widać, że planowanie bezkolizyjnego rozkładu jazdy PKP nie jest zadaniem trywialnym. Widać też, że różne alternatywne rozkłady jazdy pociągów są w różnym stopniu wrażliwe na losowo pojawiające się zakłócenia związane z występującymi na różnych trasach opóźnieniami pociągów. Wśród innych związanych z tego typu planowaniem problemów warto wymienić sezonowe zmiany rozkładu jazdy, rozbudowę infrastruktury sieci kolejowej, synchronizację rozkładu jazdy linii pasażerskich z liniami towarowymi, zagwarantowanie (przyjmując sieć kolejową PKP jako rodzaj infrastruktury krytycznej (Dz. U. z 2007 r. Nr 89, poz. 590)) wymaganego marginesu bezpieczeństwa (np. przepustowości na pewnych odcinkach i/lub kierunkach sieci).

Przedstawiony przykład ilustruje wpływ struktury systemu (w rozważanym przypadku sieci kolejowej i współdzielących jej zasoby pociągów) na jego zachowanie skutkujące określonymi rozkładami jazdy (harmonogramami) linii obsługujących np. połączenia lokalne lub dalekobieżne, czy też pasażerskie lub towarowe. Wskazuje również na systemowy charakter problemu planowania rozkładów jazdy, w szczególności akcentując trudny do przewidzenia (nieliniowy) wpływ zmian struktury linii kolejowej i/lub rozstrzygnięcia konfliktów zasobowych związanych z priorytetowaniem dostępu pociągów do współdzielonych przez nie zasobów infrastruktury linii. Oznacza to, że szacowanie i/lub wariantowanie alternatywnych zachowań SWPC wpisuje się w tematykę problemów harmonogramowania cyklicznego, problemów występujących np. w zadaniach wyznaczania rozkładów jazdy (Bocewicz i Banaszak 2013c), transmisji telekomunikacyjnych (Polak 2004), planowania produkcji (Banaszak 1991,

Fanti et al. 1997; Banaszak et al. 2008), itp. Różne aspekty, intensywnie rozwijanego w pracach (Brucker i Kampmeyer 2005; Kampmeyer 2006; Levner et al. 1997, 2007, 2010; Smutnicki 2012), nurtu związanego z harmonogramowaniem cyklicznym wchodzącego w zakres Badań Operacyjnych są sprowadzane do problemów formułowanych w ujęciu teorio-systemowym (Bertalanffy 1984) i rozwiązywane metodami z obszaru Sztucznej Inteligencji (Levner et al. 2010).

Warto zwrócić uwagę na fakt, że problemy harmonogramowania cyklicznego formułowane w dyskretnych dziedzinach zmiennych decyzyjnych (np. przyjmując, że momenty rozpoczęcia i czasy wykonania operacji procesów są całkowitoliczbowe) mają diofantyczny charakter (Banaszak i Bocewicz 2011). Oznacza to, że znaczna część zadań związanych z wyznaczaniem harmonogramów cyklicznych (a zatem i oceną istnienia zachowania cyklicznego) sprowadza się do odpowiedniego problemu rozstrzygalności, tzn. do poszukiwania warunków (i związanych za nimi algorytmów) decydujących o istnieniu jakiegokolwiek rozwiązania dopuszczalnego.



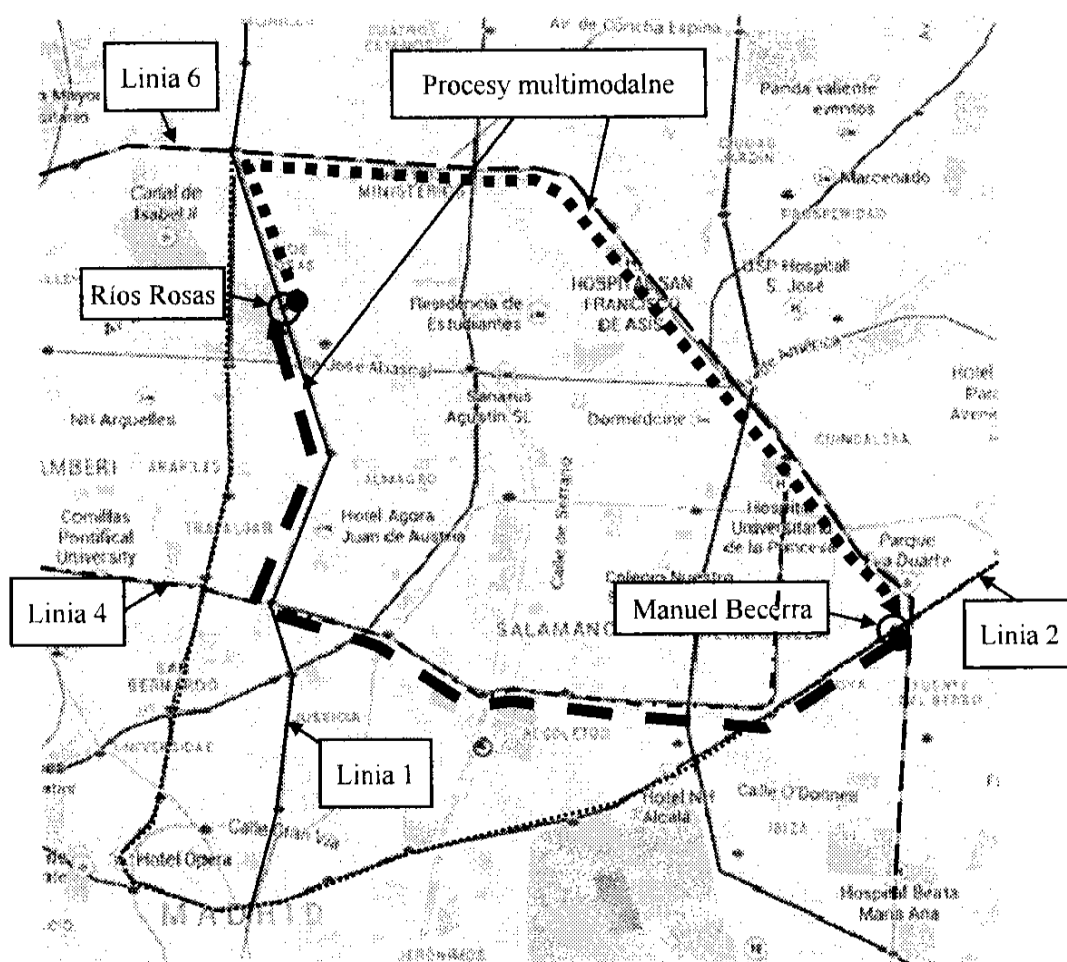
**Rys. W.3.** System składający się z współbieżnie realizowanych procesów cyklicznych

Znaczenie problemu rozstrzygnięcia o istnieniu rozwiązań dopuszczalnych ilustruje system produkcyjny z rys. W.3 realizujący jednoczesną wielowersyjną produkcję 17 różnych wyrobów. Kolejne elementy danej partii wyrobów wprowadzane są, obrabiane i opuszczają system zgodnie z wyznaczonym taktem. Każdemu z procesowi obróbki przyporządkowana jest inna marszruta. Marszruty różnych wyrobów współdzielą te same maszyny. Łatwo zauważyć, że rozważany wariant przepływu produkcji jest szczególnie podatny na występowanie konfliktów zasobowych prowadzących do

blokad lub zagłodeń. Możliwość występowania blokad oznacza, że nie w każdym systemie osiągalne są zachowania cykliczne (rytmiczna produkcja wielowersyjna) i odwrotnie, nie każde zachowanie cykliczne jest osiągalne w zadanym systemie – tzn. nie zawsze istnieje rozwiązanie równań diofantycznych formułujących rozważany problem. Oznacza to, że przestrzeń potencjalnych rozwiązań może zawierać bardzo niewiele rozwiązań dopuszczalnych lub też może nie zawierać ich wcale. Przeszukiwanie tego typu przestrzeni wiąże się zwykle z ogromnymi nakładami obliczeniowymi (NP-trudny charakter rozważanych problemów).

W systemach klasy SWPC często dochodzić może do sytuacji, w których realizacja pewnej grupy procesów warunkowana jest realizacją innych procesów (np. realizacja operacji procesu produkcyjnego warunkowana jest realizacją odpowiednich operacji procesów transportowych). Procesy takie, nazywane procesami multimodalnymi (Bocewicz i Banaszak 2011, 2013a, 2013b, 2013c; Bocewicz et al. 2011a, 2012b) – realizowane na „odcinkach” lokalnych procesów cyklicznych stanowią naturalne rozszerzenie systemów SWPC.

Przykładów procesów multimodalnych dostarczają sytuacje związane z codziennym dojazdem do pracy (autobus – kolej podmiejska – metro), serwisem poczty kurierskiej (np. DHL) itp. Z kolei, przykładami współbieżnie realizowanych procesów multimodalnych są podróźni kursujący wzdłuż marszrut, odcinki których obsługiwane są przez różne linie metra, przewoźników linii autobusowych, kolejowych, itp.

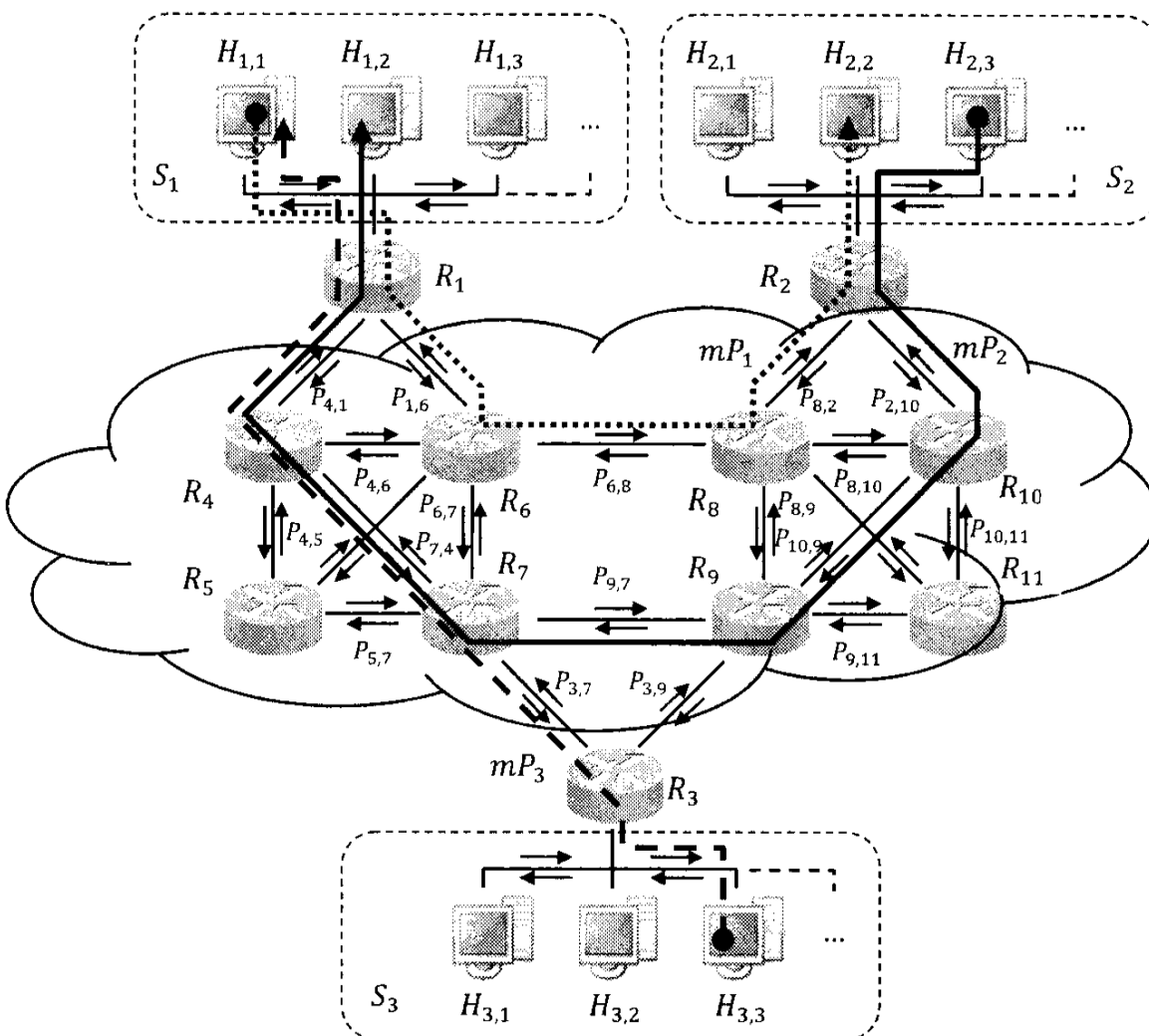


Rys. W.4. Madryt – połączenia linii metra (maps.google.pl)





Rys. W.4 przedstawia dwa procesy multimodalne reprezentujące transport pasażerów między stacjami Ríos Rosas i Manuel Becerra linii metra w Madrycie (Hiszpania). Realizacja tych procesów (transport pasażerów) jest warunkowana realizacją procesów lokalnych reprezentujących przemieszczające się pociągi. W przypadku strumienia pasażerów z Ríos Rosas do Manuel Becerra wymagany jest dostęp do Linii 1 i 6 (pasażerowie w celu przemieszczenia się między stacjami muszą korzystać z pociągów), przypadku strumienia pasażerów z Manuel Becerra do Ríos Rosas wymagany jest dostęp do Linii 2, 4 i 1.


Analogicznego przykładu dostarcza ilustracja strumieniowej transmisji danych między hostami trzech sieci LAN:  $S_1$ ,  $S_2$ ,  $S_3$  (patrz rys. W.5). Dane transmitowane są pomiędzy routerami wzdłuż odcinków, reprezentujących media transmisyjne za pośrednictwem cyklicznie przebiegających procesów lokalnych. Przebiegi tych procesów warunkowane są tak przez fizyczne parametry danego medium (pasmo przeniesienia, prędkość transmisji, itp.), jak i wykorzystane do ich synchronizacji protokoły komunikacyjne (IPv4, IPv6, ICMP, CLNS/DECN, itp.).




### Legenda

$R_i$   – router  $R_i$

 – medium transmisyjne

$P_{i,j}$   – proces odpowiedzialny za transmisję danych między routerami  $R_i, R_j$

$mP_i$   – strumień danych (proces multimodalny  $mP_3$ )

Rys. W.5. Sieć komputerowa

W rozważanym przypadku, procesom multimodalnym, które do swojej realizacji wykorzystują procesy odpowiedzialne za transmisje wzdłuż wykorzystywanych odcinków sieci, tzn. strumieniom danych odpowiadają procesy:  $mP_1$  – oznaczony linią kropkowaną,  $mP_2$  – oznaczony linią ciągłą oraz  $mP_3$  – oznaczony linią przerywaną. Oznacza to, że transmisja danych wzdłuż strumienia odpowiadającego procesowi  $mP_1$  warunkowana jest dostępem do kolejnych procesów lokalnych:  $P_{1,6}$ ,  $P_{6,8}$ ,  $P_{8,2}$ . Z kolei, strumienie  $mP_2$ ,  $mP_3$ , wymagają dostępu odpowiednio do procesów:  $P_{2,10}$ ,  $P_{10,9}$ ,  $P_{9,7}$ ,  $P_{7,4}$ ,  $P_{4,1}$  oraz  $P_{3,7}$ ,  $P_{7,4}$ ,  $P_{4,1}$ . Jak widać strumienie te współdzielą między sobą procesy:  $P_{7,4}$ ,  $P_{4,1}$ .

Systemy uwzględniające możliwość występowanie procesów multimodalnych są nazywane dalej Systemami Współbieżnych Multimodalnych Procesów Cyklicznych (SWMPC). W systemach tego typu rozważane problemy sprowadzają się najczęściej do odpowiedzi na następujące pytania:

- Czy w zadanym SWMPC osiągalne jest zachowanie cykliczne?
- Czy istnieją warunki (np. mechanizmy synchronizacji procesów, marszruty procesów, czasy wykonania operacji, itp.), spełnienie których gwarantuje osiągalność w SWMPC zachowania cyklicznego?
- Czy istnieje możliwość zmiany zachowania cyklicznego bez konieczności wstrzymania realizacji procesów?

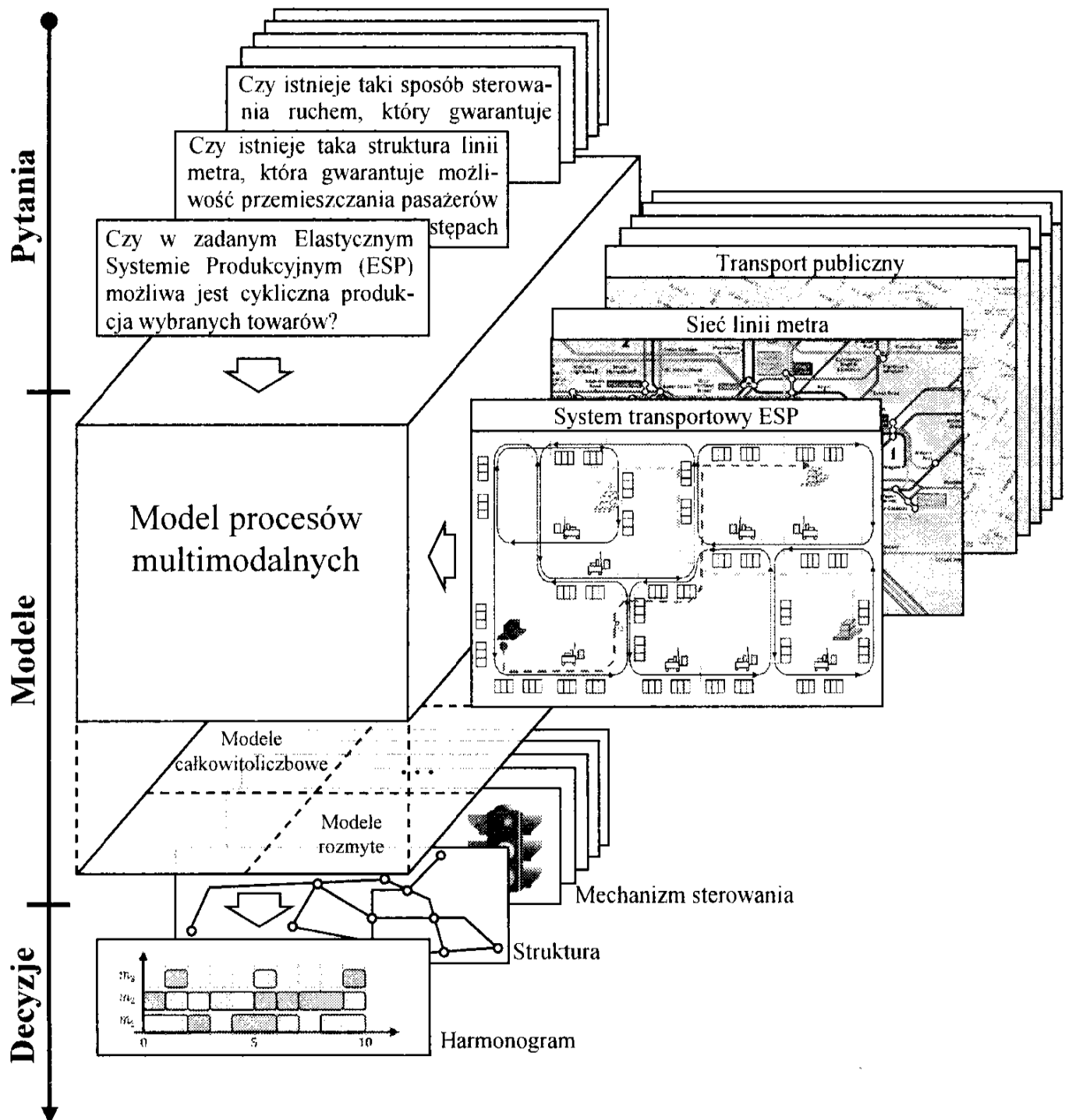
Powyższe pytania odnoszą się do trzech klas problemów:

- **problemów analizy**, polegających na wyznaczeniu harmonogramów cyklicznych osiągalnych w danym systemie, obejmują one większość problemów omawianych w literaturze (np. ocenę możliwości transmisji strumieniowej w sieci z rys. W.5),
- **problemów syntezy**, polegających na poszukiwaniu parametrów systemu gwarantujących zadane cechy osiągalnych w systemie zachowań cyklicznych (np. poszukiwanie rozkładu jazdy pociągów Linii 2, 1 i 4 – rys. W.4 – gwarantującego dzienny transport pasażerów w liczbie co najmniej 8000 lub np. poszukiwanie marszrut strumieni, w sieci z rys. W.5, gwarantujących prędkość transmisji odbioru telewizji HDTV – 32 Mb/s),
- **problemów wzajemnej osiągalności zachowań cyklicznych** (tzw. problem reharmonogramowania), polegających na ocenie możliwości zmiany zachowania systemu pomiędzy kilkoma osiągalnymi zachowaniami cyklicznymi (np. zmiany rozkładu jazdy „faworyzującego” kierunek Ríos Rosas → Manuel Becerra na rozkład „faworyzujących” kierunek Manuel Becerra → Ríos Rosas – rys. W.4).

Spośród podejść najczęściej wykorzystywanych do rozwiązywania tego typu problemów należy wymienić: metody programowania matematycznego (Kampmeyer 2006), formalizmy (max,+) algebry (Polak et al. 2004), teorię sieci Petriego (Yalcin i Boucher 2000; Fanti i Zhou 2004), itp. Podejścia te nie były stosowane do rozwiązywania problemów harmonogramowania cyklicznego procesów multimodalnych. W tym kontekście celem niniejszej pracy są oryginalne deklaratywne modele systemów multimodalnych procesów cyklicznych, modele umożliwiające opracowanie obliczeniowo efektywnych, dedykowanych metod wspomaganie decyzji z zakresu

wyżej przedstawionych problemów planowania SWMPC. Ilustrację przyjętej metodyki wspomagania decyzji w problemach planowania i sterowania SWMPC przedstawia rys. W.6.

W prezentowanym podejściu zakłada się, że rozważany SWMPC (np. system transmisji danych, system komunikacji miejskiej, system produkcyjny, itp.) oraz związany z nim problem (typu analizy lub syntezy) formułowane są w postaci trójek zbiorów: zmiennych decyzyjnych, ich dziedzin oraz ograniczeń wiążących zmienne. Rozwiązania problemów, w odpowiednich reprezentacjach bądź to diagramów Gantta, bądź też parametrów struktury systemu, czy wreszcie sposobu sterowania procesami wymuszającego pożądaną zmianę zachowania, są uzyskiwane na drodze poszukiwania wartości zmiennych decyzyjnych spełniających zadany zbiór ograniczeń.



Rys. W.6. Przyjęty proces podejmowania decyzji

Inaczej mówiąc, przyjęty deklaracyjny charakter modelu SWMPC implikuje deklaracyjny charakter formułowanych w nim problemów decyzyjnych, w szczególności formułowanych jako Problemy Spełniania Ograniczeń (Tsang 1993; Russell i Norvig 2009). Przyjęte założenia w sposób naturalny implikują zastosowanie technik programowania z ograniczeniami implementowanych w takich środowiskach jak: ILOG, Oz Mozart, ECL<sup>i</sup>PS<sup>e</sup>.

Warto zauważyć, że przyjęta metodyka dopuszcza algorytmy i modele zawierające zmienne o charakterze rozmytym i/lub losowym, liniowe modele programowania całkowitoliczbowego, heurystyki populacyjne (algorytmy ewolucyjne, algorytmy mrówkowe), itp.

Praca składa się z sześciu rozdziałów uzupełnionych stosownym wykazem literatury przedmiotu oraz skorowidzem.

W rozdziale pierwszym przedstawiono przegląd podstawowych wersji problemu harmonogramowania cyklicznego oraz autorską, decyzyjną (uwzględniającą diofantyczny charakter) wersję tego problemu. Omówiono również możliwości harmonogramowania procesów w systemach klasy SWPC i SWMPC i w ich kontekście sformułowano problemy typu: analizy zachowań, syntezy oraz składania SWMPC.

W rozdziale drugim przedstawiono deklaracyjny model SWMPC, na który składają się dwa elementy: struktura i zachowanie. Pierwsza część rozdziału obejmuje opis struktury i przyjętej koncepcji modelowania wielopoziomowego, zakładającej istnienie wielu poziomów zachowań procesów multimodalnych. Druga część rozdziału dotyczy wielopoziomowego modelu zachowania determinowanego przez strukturę SWMPC. Przyjęto, że zachowanie SWMPC jest reprezentowane przez przestrzeń stanów, w której instancje zachowania cyklicznego reprezentowane są przez cykliczne przebiegi ustalone – składowe digrafów zachowań (tzw. wirów). Ostatnia część rozdziału formułuje zagadnienia osiągalności struktury i zachowania SWMPC w kontekście dedykowanych analizie, syntezie i składaniu tych systemów problemów spełniania ograniczeń. Dla problemów tych dokonano oceny ich rozstrzygalności oraz oszacowano związaną z nimi złożoność obliczeniową.

W rozdziale trzecim przedstawiono dwie autorskie metody rozwiązywania problemów analizy SWMPC, metody opartej na redukcji stanów początkowych oraz metody generowania harmonogramów cyklicznych w wyniku poszukiwania acyklicznych digrafów pierwszeństwa operacji. Dla obu metod przedstawiono algorytmy implementowane w środowiskach programowania z ograniczeniami oraz dokonano oceny ich złożoności obliczeniowej. Ponadto, przedstawiono zastosowanie drugiej z metod do rozwiązywania problemów syntezy SWMPC. Obie metody pozwalają rozwiązywać problemy o skali spotykanej w praktyce, w trybie on-line.

W rozdziale czwartym przedstawiono koncepcję składania struktur SWMPC. Scharakteryzowano dwa rodzaje warunków gwarantujących zachowanie cykliczne systemu o strukturze złożonej z wielu podstruktur systemów cyklicznych. Warunki te wykorzystane zostały w autorskiej metodzie składania struktur regularnych. Metoda ta stanowi rozszerzenie metod przedstawionych w rozdziale trzecim, umożliwia ocenę zachowania systemu o skali większej niż dopuszczały to metody wcześniejsze, wykorzystując w tym celu informacje o zachowaniu tylko niewielkiego fragmentu jego struktury.

W rozdziale piątym omówiono problem wzajemnej osiągalności zachowań cyklicznych SWMPC. W oparciu o wprowadzony wielopoziomowy model zachowania SWMPC zaproponowane zostały warunki umożliwiające przechodzenie pomiędzy wybranymi instancjami zachowania cyklicznego systemu bez konieczności relokacji realizowanych w nim procesów. Możliwość płynnej zmiany zachowań wykorzystana została w metodzie wyznaczania zachowań cyklicznych stanowiących kombinację cyklicznych przebiegów ustalonych generowanych w strukturze rozważanego systemu.

Rozdział szósty przedstawia podsumowanie uzyskanych wyników oraz dyskusje niektórych problemów otwartych stanowiących kontynuację omówionych wcześniej problemów harmonogramowania procesów multimodalnych.

W celu zwiększenia czytelności pracy zastosowano różne sposoby formatowania. Ważniejsze wnioski i pojęcia pojawiające się po raz pierwszy są wyróżnione poprzez **wytluszczenie**. Sformułowania problemów, związane z nimi pytania, własności i twierdzenia są wyróżniane *kursywą*. Koniec przykładu oznaczony jest symbolem ■, koniec dowodu twierdzenia i własności symbolem □.

W pracy wykorzystano doświadczenia Autora zebrane podczas badań prowadzonych na Politechnice Koszalińskiej na Wydziale Elektroniki i Informatyki, przy współpracy z Politechniką Wrocławską (Instytut Informatyki, Automatyki i Robotyki), Politechniką Poznańską (Wydział Inżynierii Zarządzania), Politechniką Warszawską (Wydział Zarządzania) oraz Uniwersytetem w Aalborg (Dania) – projekt TAPAS no 260026, Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future.

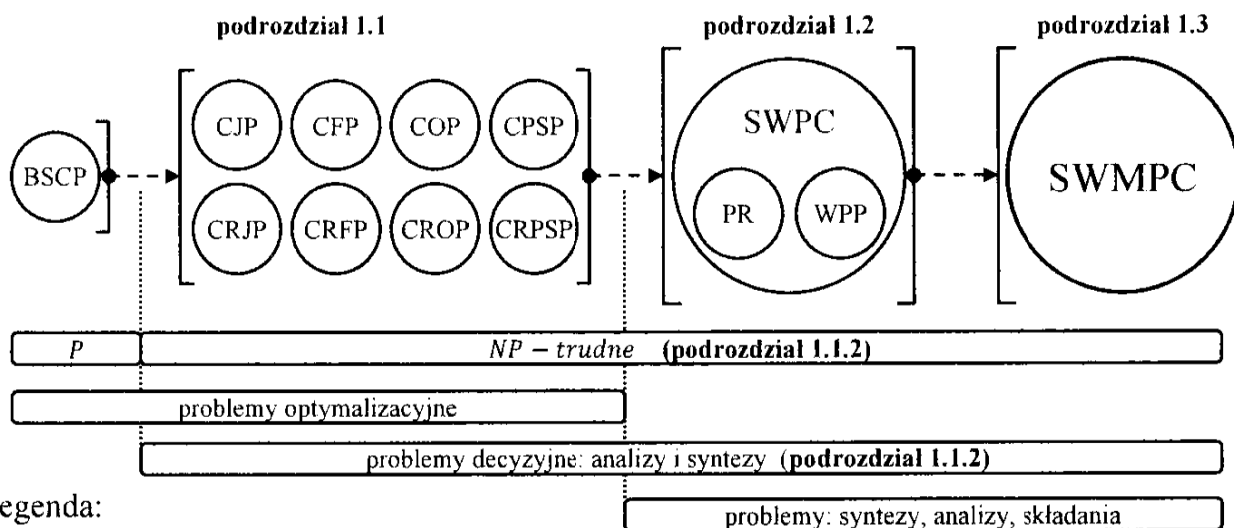
Autor będzie wdzięczny za wszelkie uwagi krytyczne dotyczące niniejszego opracowania, które przesyłać należy na adres: Katedra Podstaw Informatyki i Zarządzania, Politechnika Koszalińska, ul. Śniadeckich 2, 75 – 453 Koszalin, lub [bocewicz@ie.tu.koszalin.pl](mailto:bocewicz@ie.tu.koszalin.pl)

# 1. Modele problemów harmonogramowania cyklicznego

Powszechnie przyjmuje się, że problemy harmonogramowania cyklicznego są szczególnym przypadkiem problemów harmonogramowania (Błażewicz et al. 2007; Lerner et al. 2010, Pinedo 1995), w których zadania (operacje) powtarzane są nieskończenie wiele razy w stałych odstępach czasu nazywanych okresem. Istnieje bardzo wiele odmian tego typu problemów różniących się od siebie zbiorem uwzględnianych parametrów/zmiennych (takich jak zasoby, procesy, operacje, itp.), przyjętymi założeniami/ograniczeniami (np. niewyłączalność, skończona pojemność zasobów, wzajemne wykluczanie) jak i formą zadanego pytania (problemy optymalizacyjne, decyzyjne).

W niniejszym rozdziale opisano modele problemów harmonogramowania cyklicznego najczęściej spotykane w literaturze przedmiotu, modele specyfikujące rozważane problemy w kontekście charakteru występujących w nich zmiennych oraz ograniczeń łączących je relacji. W pierwszej kolejności przedstawiono klasyczną definicję elementarnego problemu harmonogramowania cyklicznego BCSP (ang. Basic Cyclic Scheduling Problem). BCSP stanowi podstawę do wielu rozszerzeń przedstawionych w dalszej kolejności. Spośród nich wyszczególniono problemy harmonogramowania w gniazdach produkcyjnych (tzw. problemy gniazdowe), takie jak problem ogólny CJP (ang. Cyclic Jobshop Problem), przepływowy CFP (ang. Cyclic Flowshop Problem), otwarty COP (ang. Cyclic Open Shop) oraz problemy planowania przedsięwzięć CPSP (ang. Cyclic PERT-Shop Problem). W odróżnieniu od BSCP problemy gniazdowe charakteryzują się przynależnością do klasy problemów NP-trudnych.

## Harmonogramowanie cykliczne



Legenda:

BSCP	– elementarny problem harmonogramowania cyklicznego	CRJP	– problem gniazdowy zrobotyzowanego cyklicznego gniazda produkcyjnego
CJP	– cykliczny problem gniazdowy	CRFP	– problem przepływowy cyklicznego zrobotyzowanego gniazda produkcyjnego
CFP	– cykliczny problem przepływowy	CROP	– problem otwarty cykliczny zrobotyzowanego gniazda produkcyjnego
COP	– cykliczny problem otwarty	CRPSP	– problem cyklicznego harmonogramowania przedsięwzięć produkcyjnych w zrobotyzowanym gnieździe produkcyjnym
CPSP	– problem cyklicznego harmonogramowania przedsięwzięć		
SWPC	– system współbieżnych procesów cyklicznych	SWMPC	– system współbieżnych multimodalnych procesów cyklicznych

Rys. 1.1. Struktura rozdziału 1

Problemy gniazdowe, rozszerzane dalej do problemów harmonogramowania cyklicznego zrobotyzowanego gniazda produkcyjnego, zwykle rozważane są jako problemy optymalizacyjne.

Liczne założenia dotyczące realizacji operacji (brak buforów, brak oczekiwania, niewywłaszczalność zasobów, wzajemne wykluczanie, itp.), sprawiają, że możliwe są sytuacje, w których zachowania cykliczne (ilustrowane przy pomocy cyklicznych diagramów Gantta) nie są osiągalne. W takich przypadkach rozważane są problemy decyzyjne, których celem jest określenie czy istnieją cykliczne harmonogramy (problemy analizy) lub warunki konieczne do ich istnienia (problemy syntezy).

Harmonogramowanie współbieżnych procesów cyklicznych jak i współbieżnych procesów multimodalnych, należy do najbardziej złożonych problemów harmonogramowania cyklicznego. Problemom tym (analizy, syntezy i składania SWPC) poświęcono ostatnią część rozdziału.

Struktura zagadnień przedstawianych w tym rozdziale została zilustrowana na rys. 1.1.

### 1.1. Problem harmonogramowania cyklicznego

Problem harmonogramowania cyklicznego w swojej podstawowej formie zdefiniowany został jako problem optymalizacyjny, tzw. elementarny problem harmonogramowania cyklicznego (BCSP) (Romanovskii 1967; Chrétienne 1991; Munier 1996; Kampmeyer 2006).

Dany jest skończony zbiór operacji:  $O = \{o_1, \dots, o_i, \dots, o_n\}$ . Każdej operacji  $o_i$  przyporządkowany jest jej czas wykonania  $t_i \in \mathbb{N}^+$  (czasy  $t_i$  są elementami sekwencji  $T = (t_1, \dots, t_i, \dots, t_n)$ ). Operacje  $o_i$  powtarzane są nieskończenie wiele razy, a ich każdemu ( $k$ -temu) wykonaniu odpowiadają stosowne momenty rozpoczęcia  $x_i(k) \in \mathbb{C}$ .

W pracy przyjęto, że zarówno czasy wykonania jak i momenty rozpoczęcia operacji przyjmują wartości naturalne. W ogólności w problemach harmonogramowania cyklicznego wartości te należą do  $\mathbb{R}^+ \cup \{0\}$  (Kampmeyer 2006).

Zbiór  $X = \{x_i(k) \mid i = 1 \dots n; k \in \mathbb{C}\}$  nazywany jest **harmonogramem cyklicznym** gdy spełniona jest następująca zależność:

$$x_i(k) = x_i(0) + \alpha \cdot k, i = 1 \dots n; k \in \mathbb{C}, \quad (1)$$

gdzie zmienna  $\alpha \in \mathbb{N}^+$  określa **okres** harmonogramu.

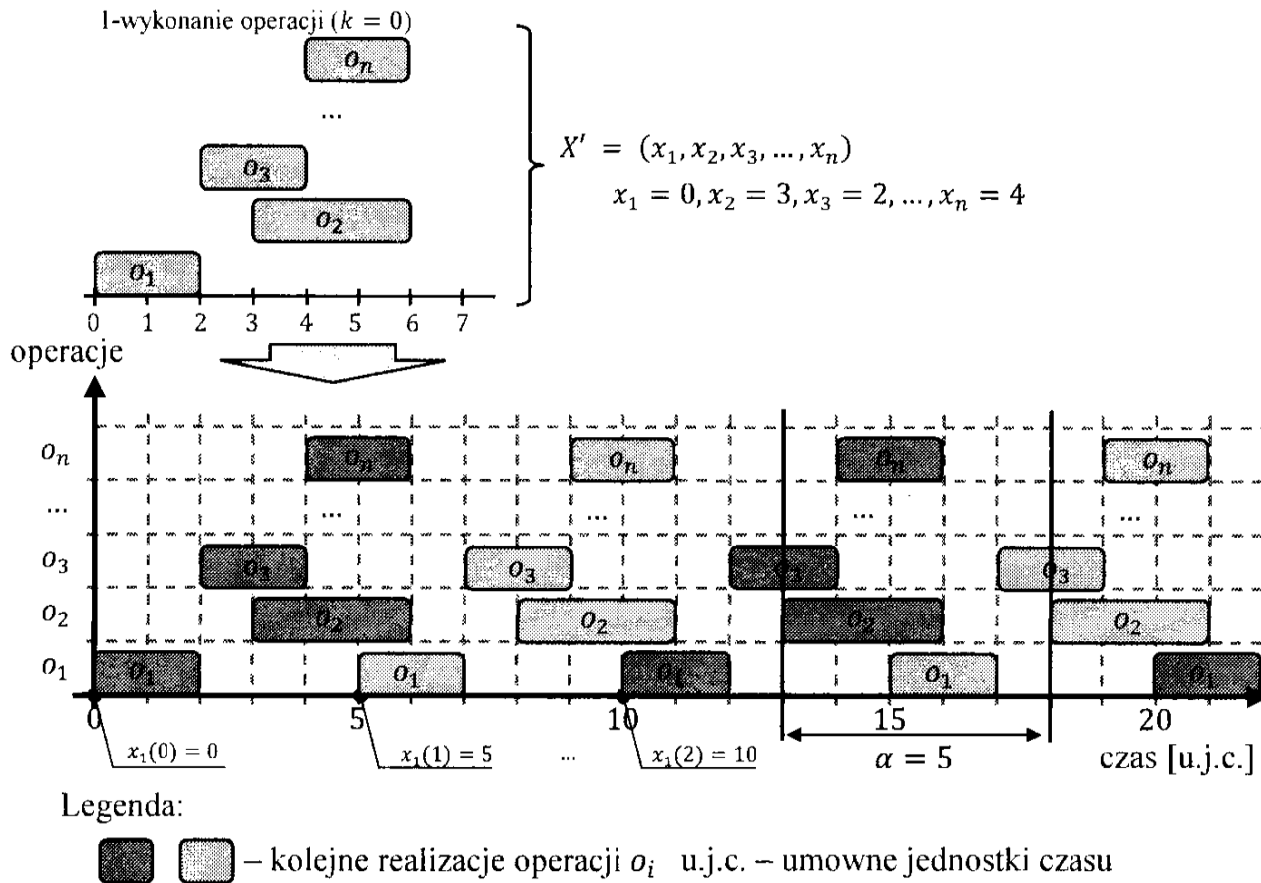
Każdy harmonogram cykliczny można zilustrować graficznie w postaci **cyklicznego diagramu Gantta**, przykład takiego diagramu został przedstawiony na rys. 1.2.

Harmonogram  $X$  może być również przedstawiany jako para sekwencji  $X'$  i okresu  $\alpha$  (Kampmeyer 2006):

$$X = (X', \alpha), \quad (2)$$

gdzie  $X' = (x_1, \dots, x_i, \dots, x_n)$  jest sekwencją, której elementy określają momenty rozpoczęcia pierwszego wykonania operacji  $o_i$  ( $k = 0$ ):  $x_i = x_i(0)$ . W interpretacji graficznej, sekwencji  $X'$  odpowiada taki fragment harmonogramu, który jest powtarzany

cyklicznie z okresem  $\alpha$  (rys. 1.2). Oznacza to, że dysponując wartościami elementów sekwencji  $X'$  oraz wartością okresu  $\alpha$  możliwe jest odtworzenie (zgodnie z (1)) wartości momentu rozpoczęcia  $x_i(k)$  dla dowolnego  $k$ -tego wykonania operacji  $o_i$ .



Rys. 1.2. Diagram Gantta reprezentujący harmonogram cykliczny  $X$

Oprócz ograniczeń (1) związanych z rytmicznym wykonywaniem operacji przyjmuje się również ograniczenia wynikające z ich czasów wykonania:

$$x_i(k + 1) \geq x_i(k) + t_i, t_i \in \mathbb{N}^+, \tag{3}$$

oraz ograniczenia kolejnościowe określające porządek ich wykonania.

Ograniczenia te opisane mogą być w postaci grafu skierowanego (digrafu)  $G = (O, E)$ , gdzie wierzchołki odpowiadają elementom zbioru operacji  $O$  natomiast łuki elementom zbioru  $E \subseteq O \times O$ . Każdemu łukowi  $(o_i, o_j) \in E$  przypisane są dwie wartości  $L_{i,j}$  oraz  $H_{i,j}$ .  $L_{i,j} \geq 0$  jest wielkością określającą wzajemne opóźnienie między operacjami  $o_i, o_j$  (takim opóźnieniem może być na przykład czas  $t_i$ ). Z kolei wielkość  $H_{i,j} \in \mathbb{N}$  określa dystans między kolejnym wykonaniem operacji  $o_i, o_j$ , między którymi ma wystąpić opóźnienie  $L_{i,j}$ . Ograniczenie związane z łukiem  $(o_i, o_j)$  ma postać:

$$x_j(k + H_{i,j}) \geq x_i(k) + L_{i,j}. \tag{4}$$

Przykładowo ograniczenie (3) jest reprezentowane przez pętlę  $(o_i, o_i)$ , gdzie  $L_{i,i} = t_i$  i  $H_{i,i} = 1$ . Z kolei ograniczenie kolejnościowe typu  $o_i < o_j$  (operacja  $o_i$  poprzedza operację  $o_j$ ) reprezentowane jest przez łuk  $(o_i, o_j)$ , gdzie  $L_{i,j} = t_i$ , a  $H_{i,j} = 0$ .



Korzystając z przyjętej notacji problem harmonogramowania BCSP cyklicznego definiowany jest następująco:

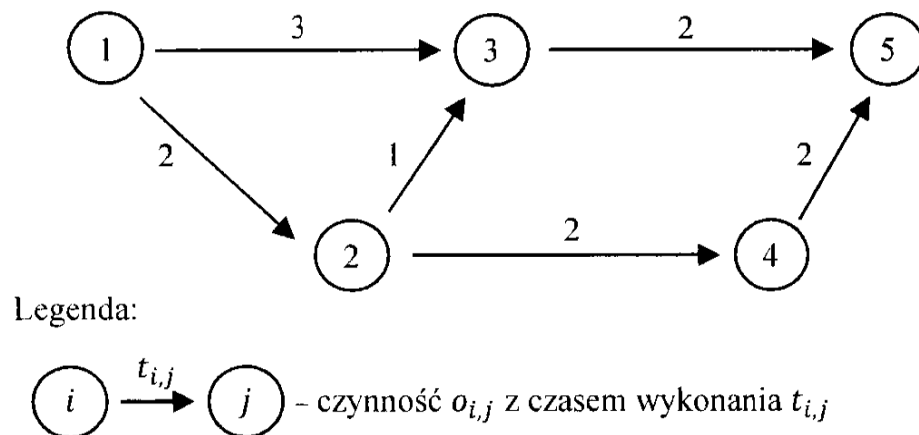
Dany jest zbiór operacji  $O = \{o_1, \dots, o_i, \dots, o_n\}$ , znane są ich czasy wykonania  $T = (t_1, \dots, t_i, \dots, t_n)$  oraz znany jest digraf relacji kolejnościowych  $G = (O, E)$  wraz z wartościami  $L_{i,j}$ ,  $H_{i,j}$ . Poszukiwany jest cykliczny harmonogram  $X$  (2) minimalizujący okres  $\alpha$ .

Rozważany problem sprowadza się do wyznaczenia wartości momentów  $x_i(k)$  rozpoczęcia operacji spełniających ograniczenia (1), (3), (4) i minimalizujących okres  $\alpha$ . Podstawiając zależność (1) do (3) i (4) ograniczenia te można uprościć do postaci:

$$\alpha \geq t_i, i = 1 \dots n, \quad (5)$$

$$x_j \geq x_i + L_{i,j} - \alpha \cdot H_{i,j}, i \neq j, i, j = 1 \dots n. \quad (6)$$

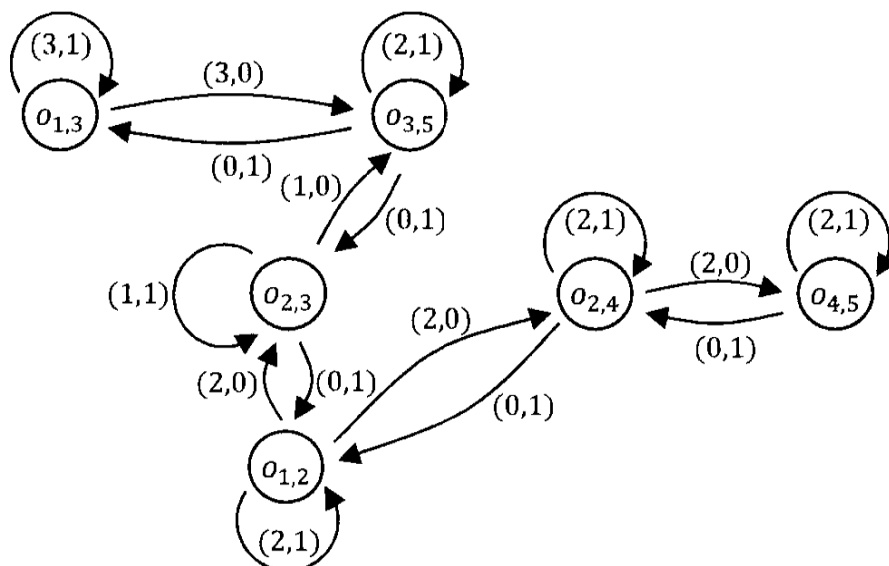
**Przykład 1.** Ilustracja elementarnego problemu harmonogramowania cyklicznego.



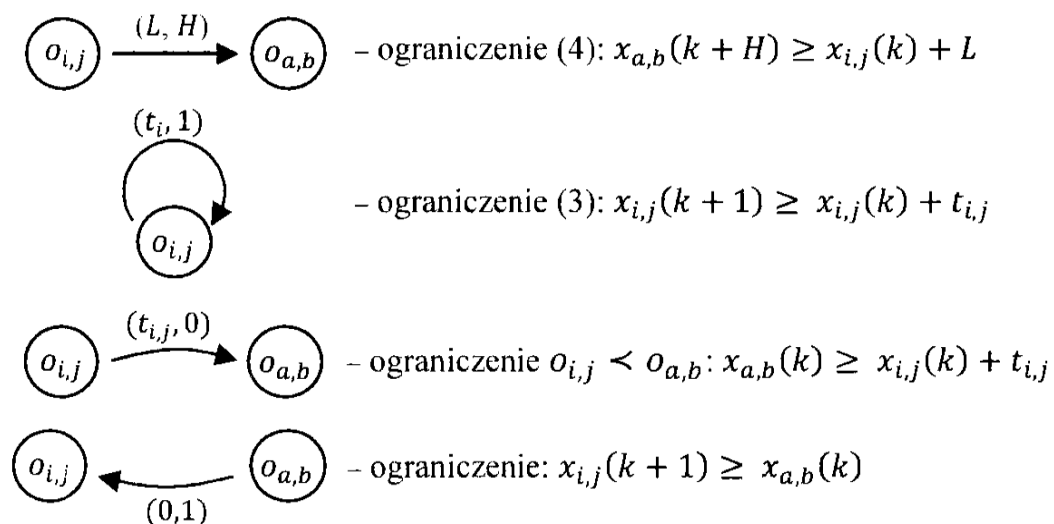
**Rys. 1.3.** Sieć czynności AOA (activity-on-arrow) przykładowego przedsięwzięcia

Na rys. 1.3 przedstawiono sieć czynności (operacji) przedsięwzięcia, w ramach którego realizowanych jest 6 operacji:  $O = \{o_{1,2}, o_{1,3}, o_{2,3}, o_{2,4}, o_{3,5}, o_{4,5}\}$ . Czasy realizacji operacji zadane są wartościami sekwencji  $T = (2,3,1,2,2,2)$ . Operacje (a tym samym przedsięwzięcie) powtarzane są cyklicznie. Zakłada się, że  $k$ -te wykonanie operacji  $o_{i,j}$  może odbyć się pod warunkiem rozpoczęcia  $k-1$  wykonania operacji będącej jej następnikiem. W tym ujęciu poszukiwana jest odpowiedź na pytanie: Jaki harmonogram  $X$  realizacji operacji minimalizuje okres  $\alpha$ ?

Ograniczenia opisujące relacje zachodzące między wykonaniem operacji zbioru  $O$  zostały przedstawione w postaci digrafu  $G$  z rys. 1.4. Digraf  $G$  opisuje zarówno ograniczenia kolejnościowe występujące między operacjami (łuki między różnymi wierzchołkami) jak i ograniczenia związane z ich cyklicznym wykonywaniem (pętle). Ograniczenia (zgodne z (4)) determinowane przez ten digraf zostały przedstawione w tab. 1.1, z kolei ich uproszczenie związane z (5) i (6) w tab. 1.2.



Legenda:



Rys. 1.4. Digraf G dla sieci z rys. 1.3

Tabela 1.1. Ograniczenia determinowane przez digraf G z rys. 1.4

Ograniczenia odpowiadające pętlom, łuki oznaczone przez $(t_{i,j}, 1)$	Ograniczenia kolejnościowe dla łuków oznaczonych przez $(t_{i,j}, 0)$	Ograniczenia kolejnościowe dla łuków oznaczonych przez $(0,1)$
$x_{1,2}(k + 1) \geq x_{1,2}(k) + 2$	$x_{2,3}(k) \geq x_{1,2}(k) + 2$	$x_{1,2}(k + 1) \geq x_{2,3}(k)$
$x_{1,3}(k + 1) \geq x_{1,3}(k) + 3$	$x_{3,5}(k) \geq x_{1,3}(k) + 3$	$x_{1,3}(k + 1) \geq x_{3,5}(k)$
$x_{2,3}(k + 1) \geq x_{2,3}(k) + 1$	$x_{3,5}(k) \geq x_{2,3}(k) + 1$	$x_{2,3}(k + 1) \geq x_{3,5}(k)$
$x_{2,4}(k + 1) \geq x_{2,4}(k) + 2$	$x_{2,4}(k) \geq x_{1,2}(k) + 2$	$x_{1,2}(k + 1) \geq x_{2,4}(k)$
$x_{3,5}(k + 1) \geq x_{3,5}(k) + 2$	$x_{4,5}(k) \geq x_{2,4}(k) + 2$	$x_{4,2}(k + 1) \geq x_{4,5}(k)$
$x_{4,5}(k + 1) \geq x_{4,5}(k) + 2$	-	-

**Tabela 1.2.** Uproszczenie ograniczeń z tab. 1.1. zgodnie z (5) i (6)

Ograniczenia odpowiadające pętlom, łuki oznaczone przez $(t_{i,j}, 1)$	Ograniczenia kolejnościowe dla łuków oznaczonych przez $(t_{i,j}, 0)$	Ograniczenia kolejnościowe dla łuków oznaczonych przez $(0,1)$
$\alpha \geq 2$	$x_{2,3} \geq x_{1,2} + 2$	$x_{1,2} \geq x_{2,3} - \alpha$
$\alpha \geq 3$	$x_{3,5} \geq x_{1,3} + 3$	$x_{1,3} \geq x_{3,5} - \alpha$
$\alpha \geq 1$	$x_{3,5} \geq x_{2,3} + 1$	$x_{2,3} \geq x_{3,5} - \alpha$
$\alpha \geq 2$	$x_{2,4} \geq x_{1,2} + 2$	$x_{1,2} \geq x_{2,4} - \alpha$
$\alpha \geq 2$	$x_{4,5} \geq x_{2,4} + 2$	$x_{2,4} \geq x_{4,5} - \alpha$
$\alpha \geq 2$	-	-

Spośród wielu, występujących w praktyce, przykładów zagadnień harmonogramowania cyklicznego (Brucker i Kampmeyer 2005; Kampmeyer 2006) warto zwrócić uwagę na problemy planowania przedsięwzięć CPM/PERT (Levner et al. 2010) jak również niektóre (zakładające nieograniczony dostęp do zasobów) problemy planowania cyklicznej produkcji wielopotokowej (Castro et al. 2008; Mayer et al. 2008; Pinto et al. 2005).

Należy zaznaczyć, że problem harmonogramowania cyklicznego postaci BCSP ma charakter liniowy (tzn. ograniczenia (1)–(4), a także (5)–(6) są nierównościami liniowymi), a do jego rozwiązania mogą być użyte klasyczne metody programowania liniowego (np. metoda Simpleks). Problem należy więc do klasy  $P$  (ang. deterministic polynomial), znane są również (Dasdan et al. 1998) algorytmy jego rozwiązywania o złożoności obliczeniowej rzędu  $O(n^4)$ .

Wyznaczony przy użyciu metody Simpleks, cykliczny harmonogram  $X$  (2) przedsięwzięcia z przykładu 1, ma postać:

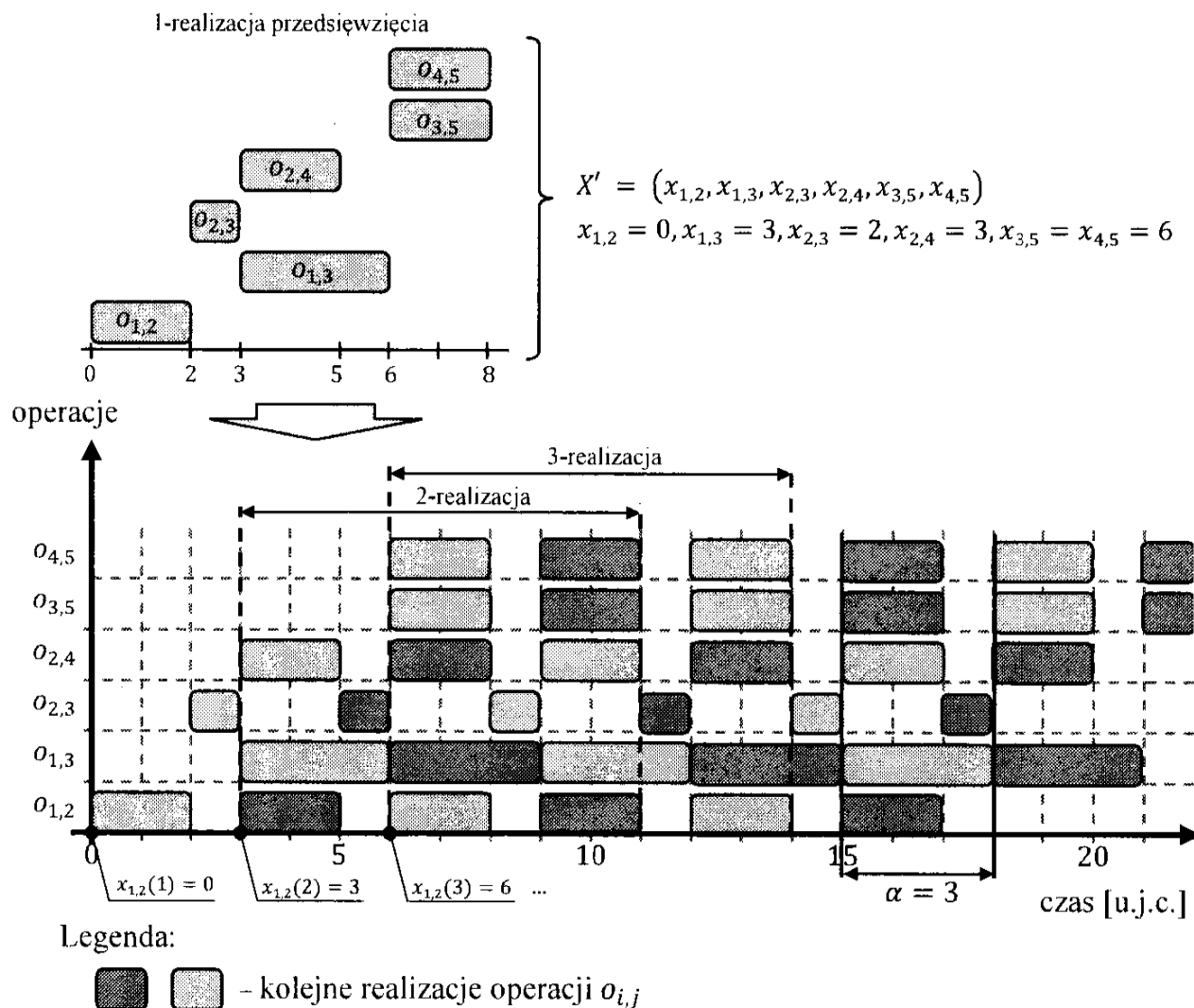
$$X = (X', \alpha)$$

$$X' = (0,3,2,3,6,6,0), \alpha = 3.$$

Graficzna ilustracja otrzymanego rozwiązania została przedstawiona w postaci diagramu Gantta na rys. 1.5. Jak widać każda realizacja przedsięwzięcia trwa 8 u.j.c. (u.j.c. – umowna jednostka czasu) natomiast okres powtarzania operacji  $\alpha$  wynosi 3 u.j.c.

Przedstawiony diagram ilustruje ideę harmonogramowania cyklicznego zgodną z rys. 1.2. Zgodnie z nią, w celu określenia harmonogramu  $X$  (2) wyznacza się taki jego fragment (wyrażany przez sekwencję  $X'$ ), który powtarzany cyklicznie z okresem  $\alpha$  nie spowoduje kolizji między realizacją kolejnych operacji oraz zagwarantuje spełnienie zadanych ograniczeń kolejnościowych (wynikających z digrafu  $G$ ). Jak pokazano na rys. 1.5 otrzymany fragment harmonogramu spełnia te założenia.

Przy poszukiwaniu harmonogramu założono domyślnie jednokrotną realizację każdej operacji. W ogólności nie musi tak być, w ramach jednego cyklu (obejmującego operacje charakteryzowane przez sekwencje  $X'$ ) operacje mogą być wykonywane dowolną (ale z góry zadaną) ilością  $k$  razy. Problemy harmonogramowania cyklicznego zakładają tzw.  $k$ -periodyczność operacji (Bocewicz et al. 2012b).



Rys. 1.5. Harmonogram  $X$  cyklicznej realizacji przedsięwzięcia z rys. 1.3

Otrzymane wyżej rozwiązanie jest rozwiązaniem optymalnym. Rodzące się w tym momencie pytanie wiąże się z gwarancją istnienia zbioru rozwiązań dopuszczalnych zawierających rozwiązanie optymalne (Papadimitrou 1994). Inaczej mówiąc powstaje pytanie: jaka właściwość digrafu  $G$  zagwarantuje istnienie niepustego zbioru rozwiązań dopuszczalnych?

W rozważanym przypadku oznacza to, że rozwiązanie optymalne wymaga gwarancji istnienia rozwiązań spełniających ograniczenia (5) i (6) (np. dla przykładu 1 ograniczenia z tab. 1.2). Biorąc pod uwagę założenie, że zmienne  $x_i$  oraz  $\alpha$  występujące w tych nierównościach przyjmują wartości z dziedziny liczb naturalnych, kwestia istnienia rozwiązań dopuszczalnych (a tym samym dopuszczalnego harmonogramu  $X'$ ) sprowadza się do oceny istnienia rozwiązań odpowiednich równań (w tym przypadku nierówności) diofantycznych.

Równanie diofantyczne (od Diofantosa (Sprindzuk 1993)) to równanie wielu zmiennych, którego niewiadome przyjmują wartości całkowite lub naturalne (najczęściej z wyłączeniem zera). Odpowiedni problem diofantyczny sprowadza się do rozwiązania równania postaci  $F(x_1, \dots, x_n) = 0$ , gdzie  $F(x_1, \dots, x_n)$  jest wielomianem (w ogólnym przypadku może to być dowolna funkcja) o współczynnikach całkowitych lub wymiernych. Jego rozwiązanie sprowadza się do rozstrzygnięcia czy ma ono rozwiązanie wymierne, czy też nie ma żadnego rozwiązania.

Matijasiewicz pokazał (Matijasiewicz 1970), że nie istnieje algorytm rozstrzygający czy dane równanie diofantyczne  $F(x_1, \dots, x_n) = 0$  ma rozwiązanie całkowite (w ten sposób rozwiązał jednocześnie X problem Hilberta (Hilbert 1900)).

Oznacza to, w ogólnym przypadku, brak metody gwarantującej istnienie rozwiązań dopuszczalnych. Brak jednej, ogólnej metody wymusza poszukiwanie metod (warunków) dedykowanych dla określonych klas równań. Przykładami równań diofantycznych, dla których znane są warunki ich rozstrzygalności są między innymi:

- równanie  $x_1^n + x_2^n = x_3^n$ , gdzie  $n \geq 2$ ,  $x_1, x_2, x_3 \in \mathbb{C} \setminus \{0\}$  związane z wielkim twierdzeniem Fermata,
- równanie liniowe  $n \cdot x_1 + m \cdot x_2 = k$ , gdzie  $n, m, k \in \mathbb{N}$ ,  $x_1, x_2 \in \mathbb{C}$ .
- równanie Pella  $x_1^2 + n \cdot x_2^2 = 1$ , gdzie  $x_1, x_2 \in \mathbb{C}$ ,  $n > 0$ ,
- równanie  $2^{x_1} + 1 = x_2^2$ , gdzie  $x_1, x_2 \in \mathbb{C}$ .

Układ nierówności (6) można również łatwo sprowadzić do układu równań diofantycznych wprowadzając zmienną pomocniczą  $x'_{i,j}$ :

$$x_i - x_j - \alpha \cdot H_{i,j} + x'_{i,j} + L_{i,j} = 0, \quad i \neq j, \quad i, j = 1 \dots n, \quad (7)$$

gdzie:  $x_i, x_j, \alpha, x'_{i,j} \in \mathbb{N}$  – niewiadome równań diofantycznych.

W pracy (Hanen i Munier 1995) przedstawiono warunki istnienia rozwiązań dla równań tego typu. Innymi słowy, scharakteryzowano właściwości jakie powinien spełniać digraf  $G$  aby dla BCSP istniał dopuszczalny harmonogram realizacji operacji zbioru  $O$ .

Niech  $\mu$  oznacza cykl w digrafie  $G$ . Opóźnieniem cyklu  $\mu$  nazywana jest wielkość:

$$L(\mu) = \sum_{(o_i, o_j) \in \mu} L_{i,j}, \quad (8)$$

z kolei dystans cyklu  $\mu$  opisany jest następująco:

$$H(\mu) = \sum_{(o_i, o_j) \in \mu} H_{i,j}. \quad (9)$$

BCSP posiada cykliczny harmonogram dopuszczalny  $X$  (2) (nierówności (5), (6) są rozstrzygalne) jeżeli każdy cykl  $\mu$  występujący w  $G$  spełnia jeden z trzech poniższych warunków:

1.  $H(\mu) > 0$ ,
2.  $H(\mu) < 0$  i  $L(\mu) < 0$ ,
3.  $H(\mu) = 0$  i  $L(\mu) \leq 0$ .

W przykładzie 1, digraf  $G$  z rys. 1.4 reprezentujący sieć czynności z rys. 1.3 zawiera 5 cykli. Każdy z nich spełnia pierwszy warunek ( $H(\mu) = 1$ ), a zatem istnieje niepusty zbiór rozwiązań (harmonogramów) dopuszczalnych.

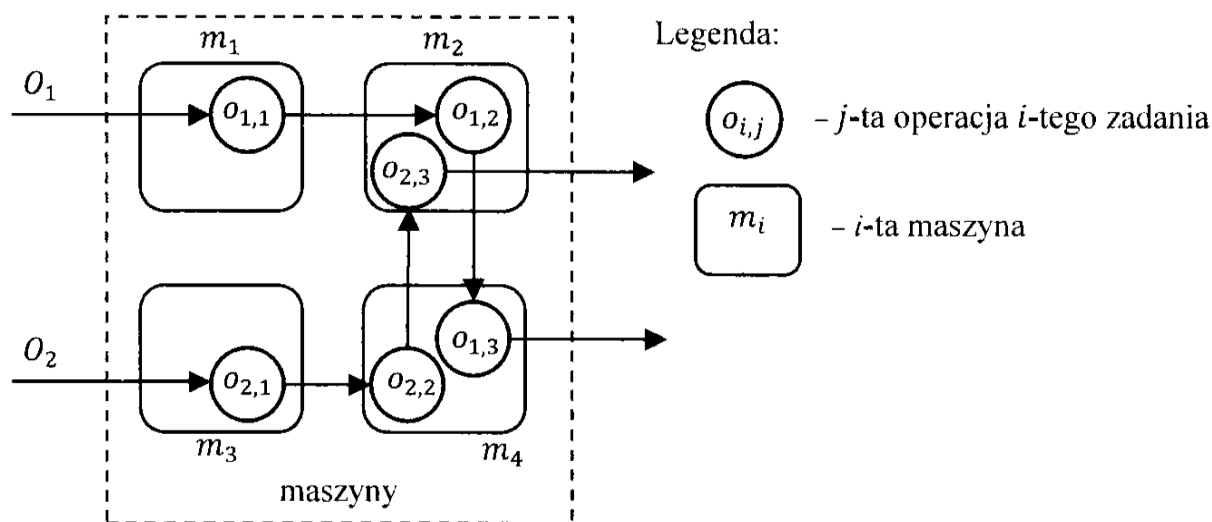
Należy podkreślić, że przedstawione warunki odnoszą się do poprawności digrafu  $G$  (rozumianej jako rozstrzygalność ograniczeń determinowanych przez ten digraf).

Istnienie rozwiązań dopuszczalnych może również wynikać z natury opisywanych problemów. Wystarczy ponownie odnieść się do przykładu 1, dla przedsięwzięć, w których nie występują konflikty zasobowe (np. prowadzące do blokady) zawsze istnieje „lepszy” lub „gorszy” harmonogram, który w skrajnym przypadku może być cyklicznie powtarzany z okresem równym czasowi realizacji przedsięwzięcia. W takim przypadku z góry wiadomo, że digraf  $G$  jest poprawny (spełniony jest jeden z powyższych warunków) a ograniczenia (5), (6) rozstrzygalne.

Podsumowując, BCSP jest rozstrzygalny co charakteryzuje się wielomianową złożonością obliczeniową. Znane są również warunki gwarantujące istnienie jego rozwiązań dopuszczalnych. W kolejnym podrozdziale przedstawiono bardziej złożone problemy oraz dokonano ich szczegółowej klasyfikacji.

### 1.1.1. Klasyfikacja

Przyjmuje się, że operacje realizowane w ramach BCSP nie wymagają zasobów (maszyn, środków transportu, narzędzi, itp.) lub też dostępne zasoby są w takiej ilości, że nie dochodzi do rywalizacji o ich dostęp. W praktyce sytuacja taka zachodzi stosunkowo rzadko, zwykle zasoby występują w ograniczonej ilości co prowadzi do konieczności rozstrzygania wielu jednoczesnych żądań o dostęp do nich – np. w sytuacjach, w których pasażerowie żądają dostępu do określonych środków transportu, gdy operacja montażu elementów wymaga dostępu do wielu narzędzi i/lub przeszkolonego pracownika, itp.



Rys. 1.6. Cykliczny problem gniazdowy (cyclic jobshop problem)

W takich przypadkach harmonogramowanie cykliczne wiąże się z koniecznością przydzielania dostępnych zasobów do poszczególnych operacji. Przykład takiego problemu przedstawiony został na rys. 1.6. Wyróżnione zostały dwie grupy operacji:  $O_1 = (o_{1,1}, o_{1,2}, o_{1,3})$  i  $O_2 = (o_{2,1}, o_{2,2}, o_{2,3})$ . Operacje powtarzane są cyklicznie zgodnie z kolejnością przedstawioną na rysunku. W odróżnieniu do BCSP (patrz digraf kolejnościowy rys. 1.3) każda operacja do swojego wykonania potrzebuje odpowiedniego środka/zasobu, np. maszyny. Operacja  $o_{1,1}$  wymaga maszyny  $m_1$ , operacja  $o_{1,2}$  maszyny  $m_2$ , operacja  $o_{1,3}$  maszyny  $m_4$  i analogicznie operacje  $o_{2,1}$  maszyny  $m_3$ , operacja  $o_{2,2}$  maszyny  $m_4$  i operacja  $o_{2,3}$  maszyny  $m_2$ .

$o_{2,1}, o_{2,2}, o_{2,3}$  wymagają kolejno  $m_3, m_4, m_2$ . Maszyny  $m_4, m_2$  wykorzystywane są przy realizacji operacji obu grup:  $o_{1,2}, o_{2,3}$  oraz  $o_{2,2}, o_{1,3}$ . Ze względu na to, że każda maszyna w danym momencie może wykonywać tylko jedną operację, operacje  $o_{1,2}, o_{2,3}$  oraz  $o_{2,2}, o_{1,3}$  wzajemnie się wykluczają. Oznacza to, że poszukiwany harmonogram cykliczny nie powinien dopuszczać by operacje te mogły być realizowane jednocześnie.

Przedstawiony problem należy do klasy cyklicznych problemów gniazdowych CJP (ang. cyclic jobshop problem). W odróżnieniu od BCSP w problemach tego typu operacje podzielone są na grupy realizowanych cyklicznie zadań  $Z = \{Z_1, \dots, Z_i, \dots, Z_n\}$  (Błażewicz et al. 2007). Każdemu z zadań  $Z_i \in Z$  przypisana jest sekwencja (również powtarzanych cyklicznie) operacji  $O_i = (o_{i,1}, \dots, o_{i,j}, \dots, o_{i,l(i)})$ , gdzie  $o_{i,j}$  – oznacza  $j$ -tą operację  $i$ -tego zadania natomiast  $l(i)$  – liczbę operacji  $i$ -tego zadania  $Z_i$ . Każdej operacji  $o_{i,j}$  przypisany jest czas wykonania  $t_{i,j} \in \mathbb{N}^+$  oraz moment jej rozpoczęcia w  $k$ -tym wykonaniu  $x_{i,j}(k) \in \mathbb{N}$ . Czasy wykonania  $t_{i,j}$  operacji sekwencji  $O_i$  są elementami sekwencji  $T_i = (t_{i,1}, \dots, t_{i,j}, \dots, t_{i,l(i)})$ . Operacje  $o_{i,j}$  wykonywane są w kolejności wskazanej przez sekwencje  $O_i$ . Każdej operacji  $o_{i,j}$  przyporządkowana jest maszyna  $\mu(o_{i,j}) = \mu_{i,j} \in M$  ( $M = \{m_1, \dots, m_i, \dots, m_{l(M)}\}$  – skończony zbiór maszyn), na której operacja ta jest wykonywana.

Przyjmuje się, że w danym momencie na danej maszynie może być wykonywana tylko jedna operacja, maszyny są niewyłączalne (tzn. zasób nie jest dostępny do czasu ukończenia operacji), a aktualnie wykorzystywana maszyna jest zwalniana w momencie ukończenia realizowanej na niej operacji.

Zbiór  $X = \{x_{i,j}(k) | i = 1 \dots n; j = 1 \dots l(i); k \in \mathbb{C}\}$  nazywany jest harmonogramem cyklicznym (Levner et al. 2010) gdy dla każdego  $x_{i,j}(k) \in X$  spełniona jest zależność:

$$x_{i,j}(k+1) = x_{i,j}(k) + \alpha, \quad (10)$$

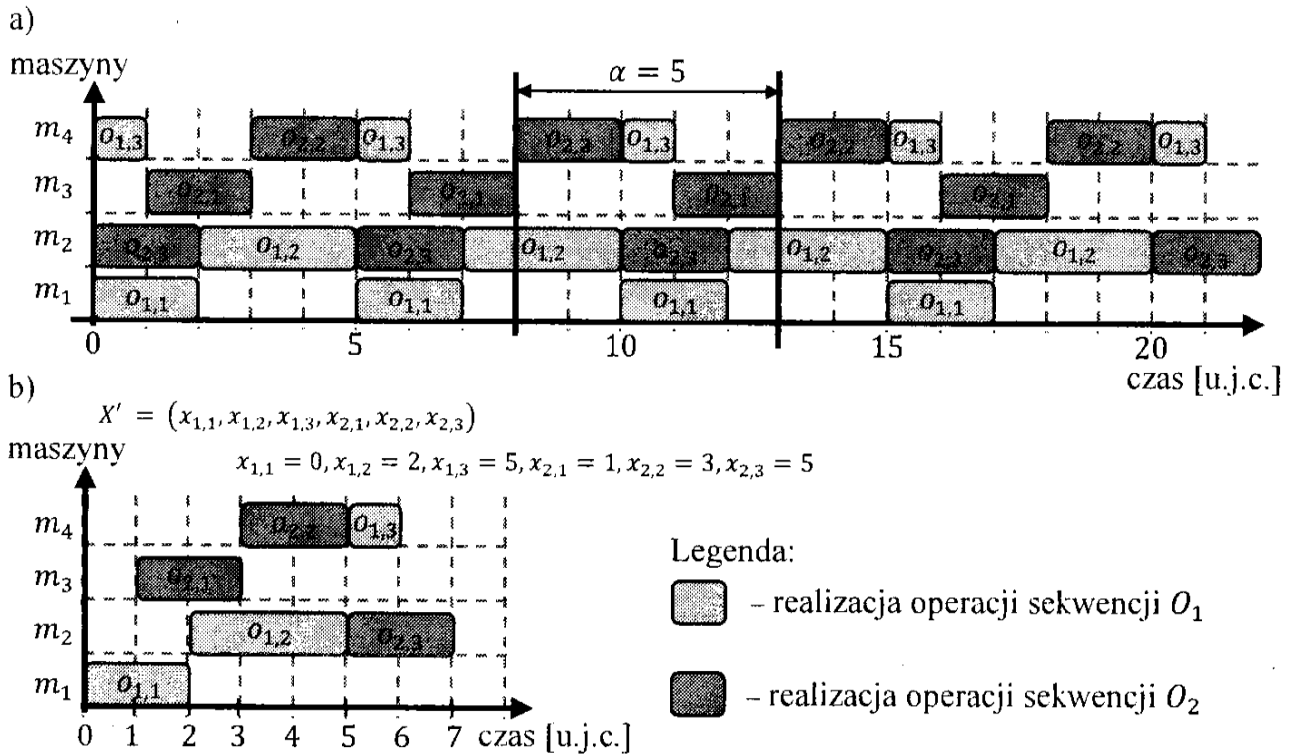
gdzie:  $\alpha$  – okres harmonogramu cyklicznego.

Przedstawiona definicja, podobnie jak w przypadku BCSP, zakłada jednokrotną realizację operacji w ramach pojedynczego cyklu (1-periodyczność). Założenie to wynika z faktu, że harmonogramowanie 1-periodyczne daje lepsze rezultaty (harmonogramy o krótszym okresie  $\alpha$  wykonywania zadań  $Z_i$ ) niż wielo-periodyczne (Kampmeyer 2006).

Problem CJP definiowany jest jako problem optymalizacyjny:

*Dany jest zbiór dostępnych maszyn  $M$  oraz zbiór zadań  $Z$  wraz z sekwencjami operacji  $O_i$ . Znane są sekwencje czasów wykonania operacji  $T_i$  i maszyny  $\mu_{i,k} \in M$ , na których są one wykonywane. Poszukiwany jest cykliczny harmonogram  $X$  minimalizujący okres  $\alpha$ .*

Przykład optymalnego harmonogramu cyklicznego  $X$  dla problemu z rys. 1.6, w którym przyjęto sekwencje czasów wykonania operacji  $T_1 = (2,3,1)$ ,  $T_2 = (2,2,2)$ , przedstawia rys. 1.7. Otrzymany harmonogram spełnia założenie niewyłączalności maszyn.



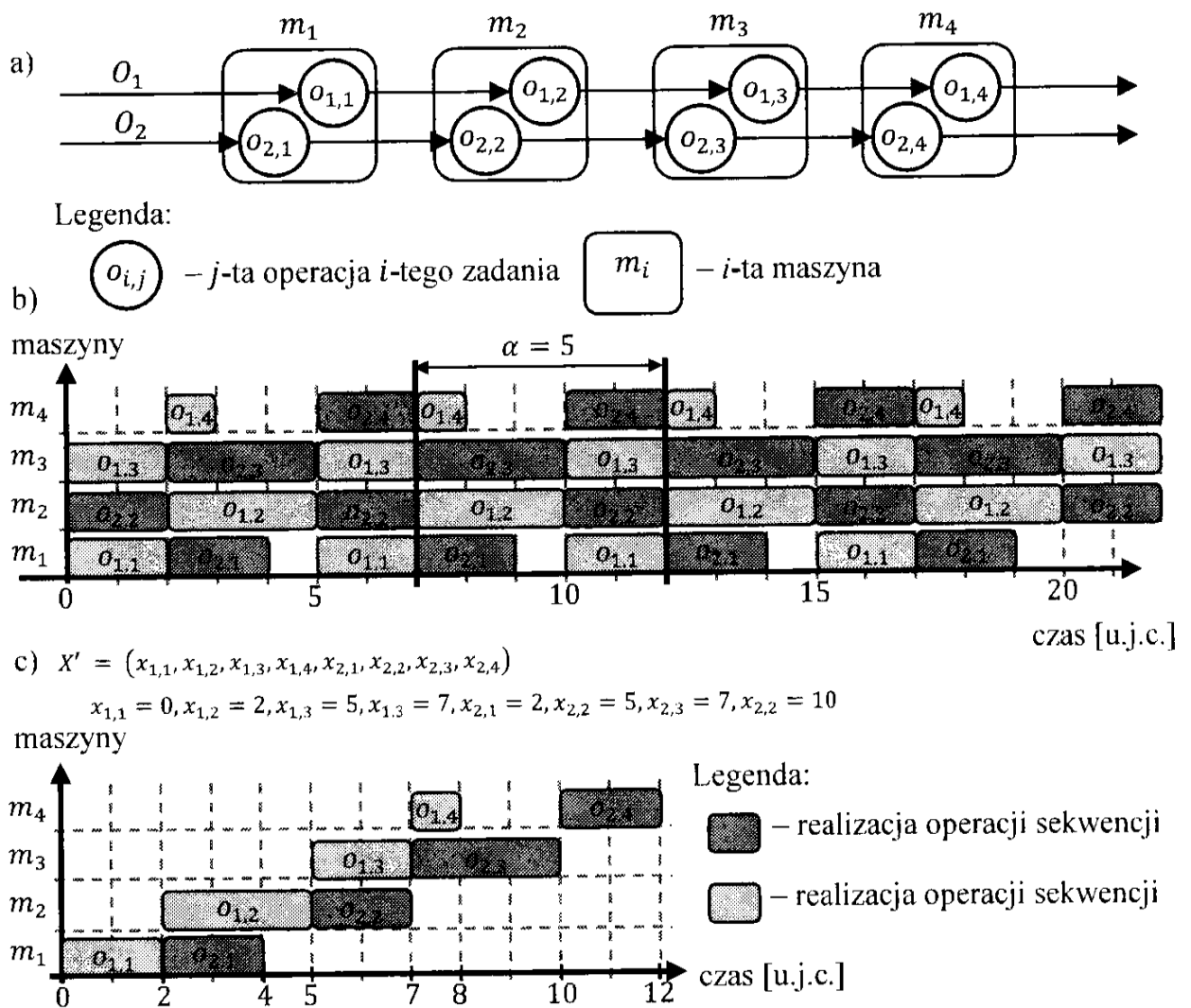
Rys. 1.7. Cykliczny harmonogram dla problemu z rys. 1.6 a), fragment harmonogramu odpowiadający sekwencji  $X'$  b)

Wyznaczony okres wynosi  $\alpha = 5$  i jest determinowany przez operacje realizowane na zasobie  $m_2$ . Zasób ten stanowi wąskie gardło, tzn. operacje na tym zasobie są wykonywane bez żadnych przestoju. Z kolei, czasy ukończenia sekwencji operacji  $O_1$ ,  $O_2$  są równe i wynoszą 6 jednostek czasu.

W ogólnym przypadku, wyznaczanie harmonogramu cyklicznego dla problemu CJP wymaga znacznie więcej nakładów obliczeniowych niż dla problemu BCSP. Problemy tego typu należą do klasy problemów NP-trudnych (Hall et al. 2002; Levner et al. 2010).

W literaturze przedmiotu wielokrotnie rozważane były zarówno uproszczenia, jak i rozszerzenia CJP. Przykładowo w pracy (Roundy 1992) rozważano problem z pojedynczym zadaniem składającym się z identycznych operacji. Autorzy wykazali, że tego typu problem jest problemem NP-trudnym i zaproponowali algorytm oparty na metodzie podziału i ograniczeń. W pracy (Matsuo et al. 1991) rozważano przypadek CJP związany z problemem harmonogramowania pracy dźwigów (ang. multi-product crane scheduling). W tym szczególnym przypadku zaproponowany został algorytm o złożoności wielomianowej. W pracach (Hanan 1994; Kampmeyer 2006) analizowany był CJP z ograniczeniami kolejnościowymi reprezentowanymi jak w przypadku BCSP, tzn. w postaci digrafu  $G$  (patrz rys. 1.4). Wykazano, że tego typu rozszerzenie należy do klasy problemów silnie NP-zupełnych. W pracy (Hall et al. 1998) rozważano wersję CJP tylko z dwoma maszynami, dla przypadku gdy każde zadanie ma tylko dwie operacje przedstawiono wielomianowy algorytm jego rozwiązania. Z kolei, w pracy (Lee i Posner 1997) analizowano przypadek CJP, w którym kolejność operacji na każdej maszynie jest znana a priori.



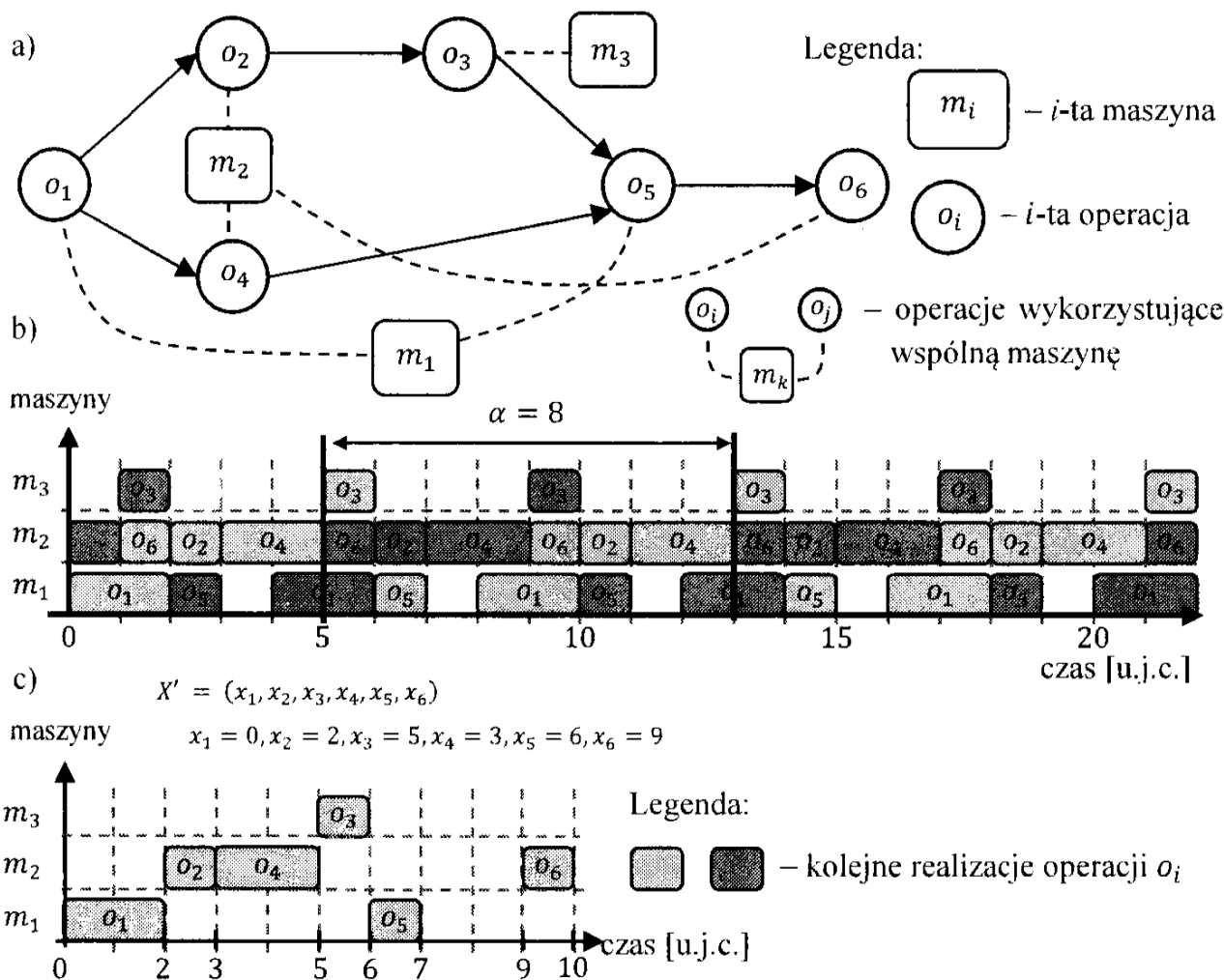


Rys. 1.8. Przykład problemu przepływowego a), optymalny harmonogram cyklicznej realizacji operacji b), fragment harmonogramu odpowiadający sekwencji  $X'$  c)

Szczególnym przypadkiem problemu CJP jest cykliczny problem przepływowy CFP (ang. cyclic flowshop problem), w którym wszystkie realizowane zadania ze zbioru  $Z$  opisane są taką samą sekwencją operacji  $O$  przechodzącą przez wszystkie maszyny ze zbioru  $M$ . Rys. 1.8a przedstawia przykład takiego problemu oraz optymalny harmonogram cyklicznej realizacji operacji (przyjęto, że  $T_1 = (2,3,2,1)$ ,  $T_2 = (2,2,3,2)$ ).

Cykliczny problem przepływowy został opisany już w latach 60. XX wieku (Levner et al. 2007). W pracach (Cuninghame-Greene 1960, 1962) rozważano cykliczny proces produkcyjny, który zgodnie z współczesną notacją może być uznany jako CFP. Wśród metod rozwiązywania tego typu problemów należy wyróżnić metody wykorzystujące aparat  $(\max,+)$  algebry (Cuninghame-Greene 1979; Cohen et al. 1985; Baccelli et al. 1992; Hanen 1994; Hanenand i Munier 1995; Lee 2000). Modele o podobnych zastosowaniach (Levner et al. 2007) zostały opisane również w pracach, (Degtyarev i Timkovsky 1976; Timkovsky 1977). Oczywiście CFP doczekał się różnych rozszerzeń, przykładem może być CFP uwzględniający możliwość wielokrotnego wykorzystania tych samych maszyn w ramach jednego zadania (powrotów do maszyn już wykorzystanych, ang. the re-entrant flowshop problem). (Graves et al. 1983).

W przypadku gdy w CJP kolejność operacji w poszczególnych zadaniach nie jest narzucona, tzn. gdy każdemu z zadań  $Z_i$  przypisany jest zbiór operacji  $O_i$  a nie ich sekwencja, problem nazywany jest problemem otwartym COP (ang. cyclic open shop). Brak z góry określonego porządku realizacji operacji sprawia, że problemy te znalazły zastosowanie w sytuacjach planowania zajęć (przykładowo planowanie cotygodniowego harmonogramu zajęć dla studentów). Problemy harmonogramowania należące do tej klasy zostały opisane m.in. w pracach (Mahadev et al. 1993; Kubale i Nadolski 2005; Modarres i Ghandehari 2008). Autorzy ci pokazali, że analogicznie jak CJP problem COP jest problemem NP-trudnym.



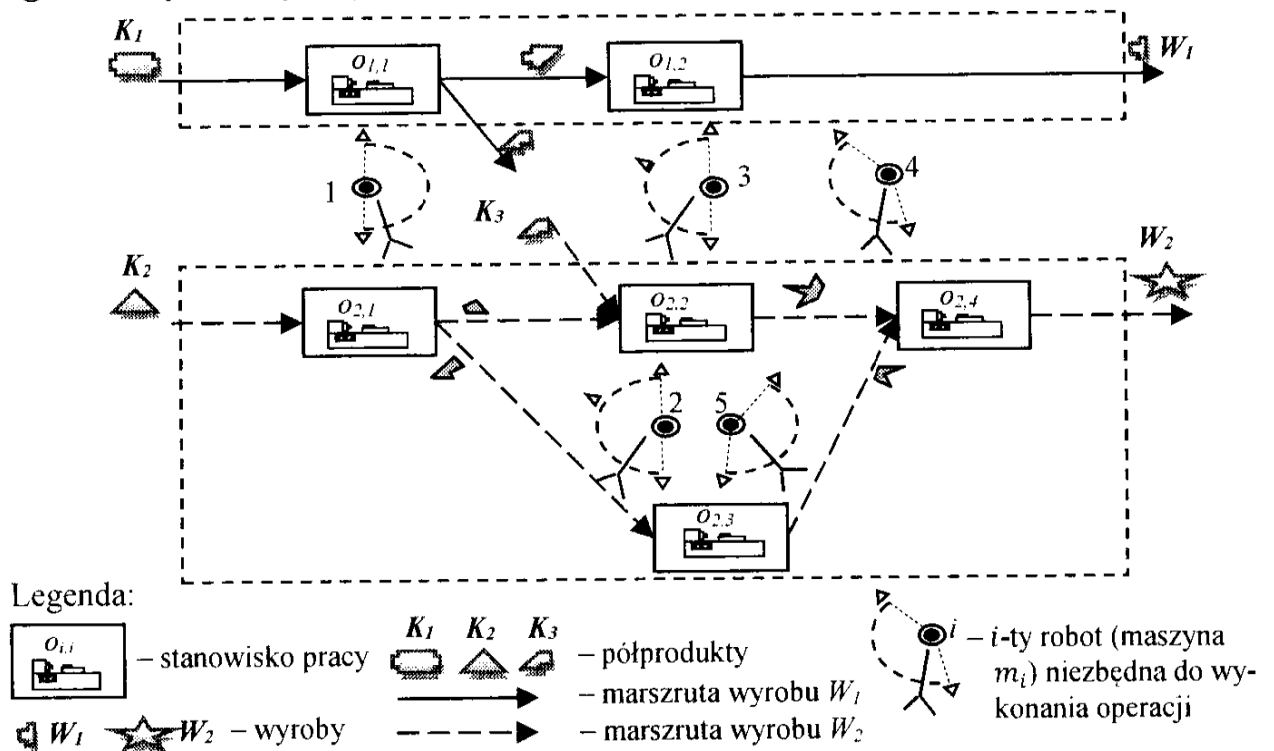
**Rys. 1.9.** Przykład harmonogramowania przedsięwzięcia z uwzględnieniem maszyn, (El Amraoui et al. 2008), digraf ograniczeń kolejnościowych a), optymalny harmonogram b), fragment harmonogramu odpowiadający sekwencji  $X'$  c)

W literaturze przedmiotu (Levner et al. 2010), w większości przypadków analizowane są problemy z określonym porządkiem operacji. Na szczególną uwagę zasługuje rozszerzenie CJP o możliwość uwzględniania częściowego porządku wykonywanych operacji (w przypadku CJP i CFP kolejność operacji jest liniowa). Ograniczenia kolejnościowe mogą być prezentowane za pomocą digrafu jak z rys. 1.9a a także (zgodnie z notacją BCSP) za pomocą digrafu  $G$  z rys. 1.4 (Kampmeyer 2006). Przykład harmonogramu minimalizującego okres  $\alpha$  oraz gwarantującego wzajemne wykluczenie operacji na maszynach, przedstawia rys. 1.9b. Problemy tego typu należą do klasy cyklicznego harmonogramowania przedsięwzięć CPSP (ang. cyclic PERT-Shop pro-

blem). Podobnie jak w przypadku CJP przyjęto, że do każdej operacji  $o_{i,j}$  przypisana jest co najmniej jedna niewyłączalna maszyna  $\mu_{i,j}$ . Problemy tego typu występują między innymi w planowaniu wielowariantowej produkcji potokowej (rys. 1.10).

Podobnie jak problemy CJP, CFP i COP również problem CPSP należy do klasy problemów NP-trudnych. W klasie tej istnieją również warianty (zwierające maksymalnie 2 maszyny) o złożoności wielomianowej (Munier 1996; Hall et al. 2000).

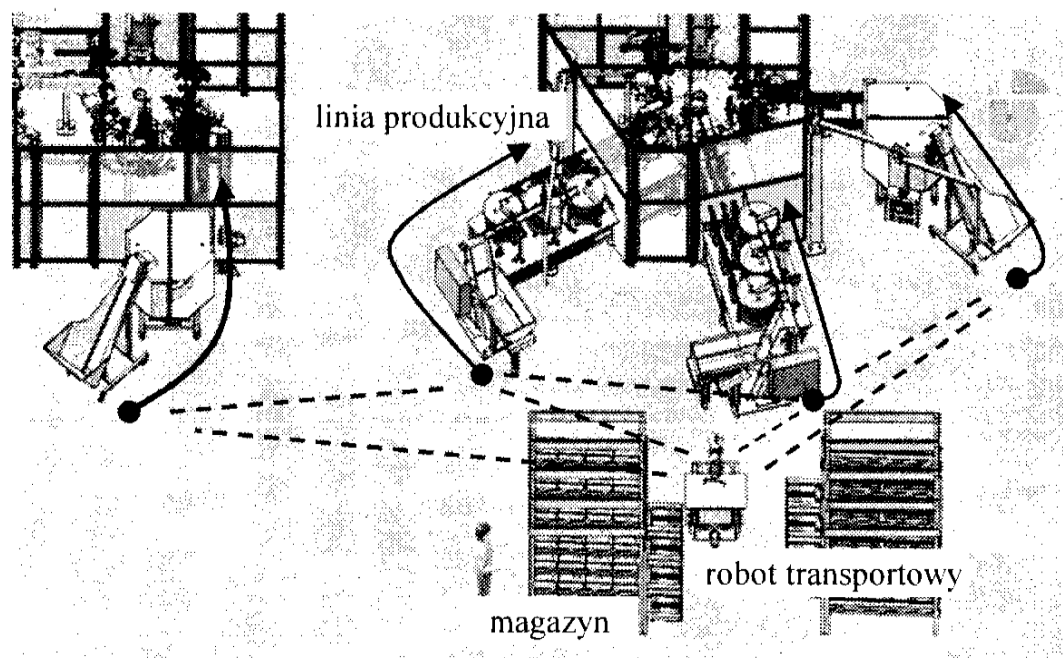
Ciągły rozwój przemysłu jak i metod związanych z planowaniem przepływu produkcji spowodował, że oprócz operacji technologicznych zaczęto zwracać szczególną uwagę na operacje związane z transportem międzystanowiskowym. Integracja procesów technologicznych i transportowych doprowadziła do rozwinięcia problemów harmonogramowania cyklicznego do tzw. problemów harmonogramowania cyklicznego zrobotyzowanych gniazd produkcyjnych (ang. cyclic robotic scheduling).



Rys. 1.10. Przykład systemu wielowariantowej produkcji potokowej (Bocewicz et al. 2009b)

Problemy te stanowią rozszerzenie problemów CFP, CJP, COP oraz CPSP o możliwość transportu międzyoperacyjnego, realizowanego przez urządzenia typu: robot, transporter, suwnica itp. (rys. 1.11).

W tym kontekście problem harmonogramowania cyklicznego zrobotyzowanego gniazda produkcyjnego polega na znalezieniu momentów rozpoczęcia operacji (momenty te determinują jednocześnie kolejność realizacji operacji na maszynach) oraz marszrut poruszania się robotów (sekwencji określających w jakiej kolejności roboty przemieszczają się pomiędzy maszynami), tak aby cykl produkcyjny był jak najkrótszy (okres  $\alpha$  harmonogramu  $X$  był jak najmniejszy). Roboty należy traktować jako specjalny rodzaj maszyn  $M$ . Można je definiować na wiele sposobów, to jakie parametry zostaną uwzględnione (prędkość poruszania, pojemność robota, mobilność, struktura, itp.) silnie wpływa na postać uzyskanego rozwiązania oraz na czas jego wyznaczenia (złożoność obliczeniową problemu).



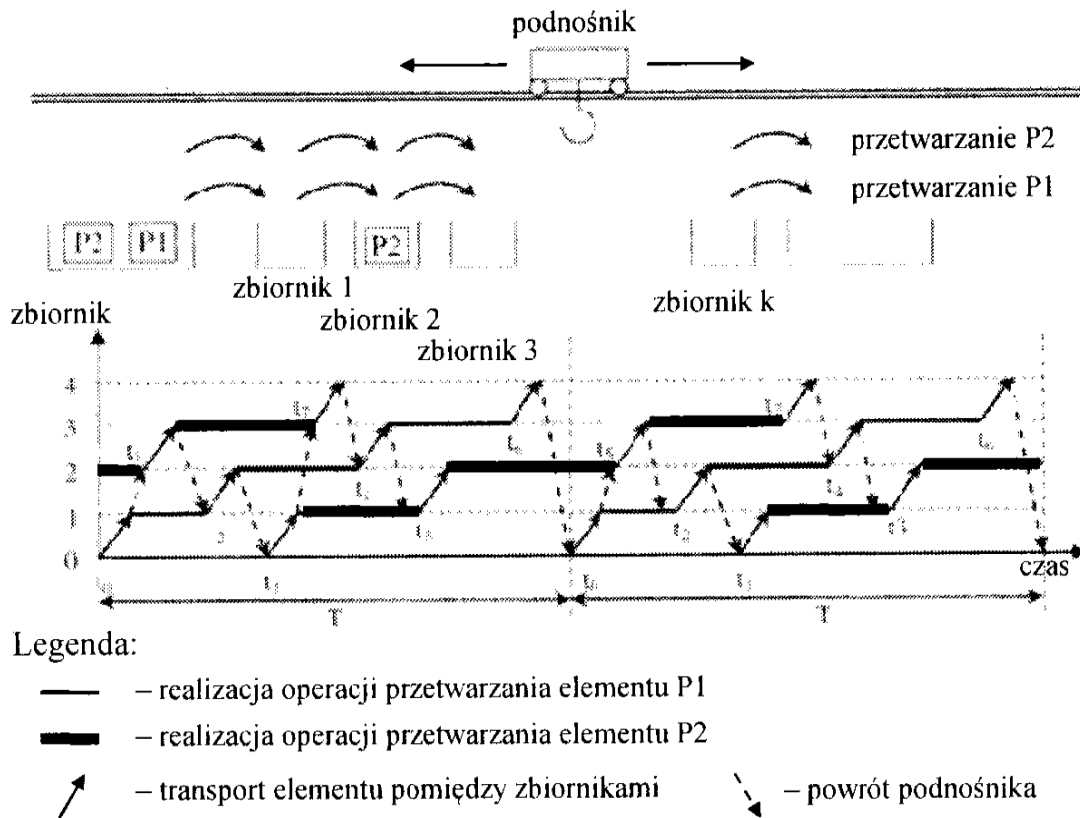
Rys. 1.11. Przykład gniazda produkcyjnego z robotem mobilnym (Dang et al. 2012)

Problemy tego typu (w szczególności cykliczny problem przepływowy zrobotyzowanego gniazda produkcyjnego) zostały wprowadzone w latach 60. XX wieku (Levner et al. 2007). W pracach (Aizenshtat 1963; Suprunenko et al. 1962) rozważana była linia produkcyjna obsługiwana przez maszyny transportowe.

Jednym z pierwszych był cykliczny problem przepływowy zrobotyzowanego gniazda produkcyjnego CRFP (ang. cyclic robotic flowshop problem). Do jego rozwiązania opracowana została metoda zabronionych przedziałów (ang. the method of prohibited intervals), wykorzystywana również w innych zastosowaniach: (Che et al. 2003; Kats i Levner 1997; Kats et al. 1999; Levner et al. 1997; Livshits et al. 1974). Przykład zrobotyzowanego systemu przepływowego został przedstawiony na rys. 1.12.

Problemy harmonogramowania cyklicznego zrobotyzowanego gniazda produkcyjnego typu: gniazdowy CRJP (ang. cyclic robotic jobshop problem) oraz harmonogramowania przedsięwzięć CRPSP (ang. cyclic robotic PERT-shop problem) stanowią swoiste uogólnienie problemów klasy CRFP. Ich opis można znaleźć odpowiednio w (Kampmeyer 2006; Kats et al. 2008).

Problemy CRFP, CRJP, CROP, CRPSP należą do klasy NP-trudnych. W pracy (Livshits et al. 1974) przedstawiono transformację problemu przepływowego do problemu plecakowego. Podobnie w (Lei i Wang 1989) przedstawiono redukcję problemu harmonogramowania cyklicznego do problemu komiwojażera. W szczególnych przypadkach opracowane zostały metody o złożoności wielomianowej. Przykładem tego może być problem cyklicznego harmonogramowania pracy podnośników (ang. cyclic hoist scheduling problem), dla pojedynczego zadania i stałej sekwencji operacji podnośnika, przedstawiony w pracach (Matsuo et al. 1991, Lei 1993, Joachim i Soumism 1995; Chen, et al. 1998; Kats et al. 2008). Ze względu na dużą (co najmniej wykładniczą) złożoność obliczeniową większość prac koncentruje się na problemach z jednym robotem (Phillips i Unger 1976; Kise 1991; Kats i Levner 1998; Hall et al. 1998; Lei i Liu 2001; Brucker et al. 2002; Hindi i Fleszar 2004; Che i Chu 2005).



Rys. 1.12. Przykład zrobotyzowanego systemu przepływowego (El Amraoui et al. 2008)

Bogaty przegląd problemów harmonogramowania cyklicznego (BSCP, CFP, CJP, COP, CPSP, CRFP, CRJP, CROP, CRPSP), w którym zagadnienia złożoności obliczeniowej zostały szczegółowo opisane w pracy (Levner et al. 2010). Autorzy wyodrębnili tylko 24 instancje problemów o złożoności wielomianowej (typu P). Są to głównie problemy o niewielkiej skali zawierające maksymalnie dwa roboty (2-cyclic robotic scheduling) i/lub dwie maszyny (multiple part-type production in 2-machine cyclic robotic no-wait cells, with additive regular cells). Pozostałą część stanowią problemy o charakterze NP-trudnym. Zestawienie ich przedstawione zostało w tab. 1.3. Do ich klasyfikacji zastosowano notację  $\alpha|\beta|\gamma$  (Dawande et al. 2007). Pierwsza litera określa rodzaj problemu: F- problem przepływowy, J-problem gniazdowy, O-problem otwarty, opcjonalnie występujący indeks dolny określa liczbę maszyn. Druga litera określa ograniczenia problemu (patrz poniżej). Trzecia litera określa funkcje celu. Przykładowe oznaczenia mają postać:

- $\beta = \text{no-wait}$  oznacza, że każde zadanie musi być wykonane bez jakiegokolwiek oczekiwania na dostęp do maszyn,  $\text{no-wait}(a, b)$  oznacza że, ograniczenie odnosi się tylko do maszyn a, b, między tymi maszynami nie może dochodzić do przestojów,
- $\beta = \text{no-buffer}$  oznacza, że między maszynami nie ma bufora, zadanie nie opuszcza maszyny do momentu uzyskania dostępu do kolejnej maszyny,  $\text{no-buffer}(a, b)$  oznacza brak bufora tylko między maszynami a, b,
- $\gamma = \text{Wt}$  oznacza, że poszukiwany jest harmonogram z minimalnym czasem wykonania zadań zbioru Z,
- $\gamma = T$  oznacza, że poszukiwany jest harmonogram z minimalnym okresem  $\alpha$ .

**Tabela 1.3.** NP-trudne problemy harmonogramowania cyklicznego (Levner et al. 2010)

<i>Cyclic (no-robot) flowshop problems</i>	<i>Cyclic robotic problems</i>
$F_2 \text{min } T Wt$ $F_3 \text{no-wait} T$ $F_3 \text{no-buffer} T;$ $F_3 \text{no-wait} T$ $F_3 \text{no-wait}(1,2), \text{wait}(2,3) T$ $F_3 \text{no-buffer}(1,2), \text{wait}(2,3) T$ $F_3 \text{wait}(1,2), \text{no-wait}(2,3) T$ $F_3 \text{wait}(1,2), \text{no-buffer}(2,3) T$ $F_3 \text{no-wait}(1,2), \text{no-buffer}(2,3) T$ $F_3 \text{no-buffer}(1,2), \text{no-wait}(2,3) T$ Permutation non-identical-jobs 3-machine no-buffer flowshop $F \text{no-buffer}, t_{i,j} = 0 \text{ or } 1 T$ $F_m (m \text{ fixed } m > 2) \text{no-buffer} T$	Cyclic 1-robot flowshop (non-Euclidean case) Min-makespan 1-robot flowshop Cyclic 1-robot flowshop 1-robot flowshop (Euclidian case)) 3-machine m-product flowshop 3-machine no-wait flowshop 1-robot 2-parallel-machine makespan 1-robot 2-parallel-machine maximum lateness 2-machine gripper cyclic robotic Single product, identical parts, 1-cycle robotic cells with general travel times 2-machine, 1-robot, single product, 1-unit cycles, two criteria: min cycle time and min total manufacturing cost Single-gripper, 1-cycle and 2-cycle robotic cells with travel times
<i>Cyclic jobshop problems</i>	<i>Cyclic PERT-shop problems</i>
$J_2 \text{min } T Wt$ $J_2 \text{no-wait} T$ $J_2 \text{no-buffer} T$ The (no-robot) jobshop with disjunctive resources The cyclic jobshop with limited number of operations Cyclic 2-machine jobshop with fixed number of operations Cyclic 3-machine jobshop with fixed number of operations 3-machine re-entrant jobshop	Cyclic jobshop/PERT-shop with m machines and disjunctive resources Cyclic PERT-shop when the underlying graph is a circuit Cyclic PERT with unit durations A single machine cycling scheduling with fixed durations and Precedence
<i>Other related cyclic scheduling problems</i>	
Traveling salesman problem Cyclic lot size 1-machine scheduling Cyclic staffing problem Cyclic knapsack A single machine cycling scheduling with fixed durations Cyclic open shop Cyclic resource allocation problem	Cycling graph chromatic scheduling Cyclic vehicle routing problem $O_2 \text{no-wait} T$ $O_2 \text{no-buffer} T$ $O_2 \text{min } T Wt$ FMS scheduling with two criteria: throughput rate and WIP

Jak widać analizowane problemy są problemami optymalizacyjnymi. W problemach tych główny akcent kładziony jest na efektywny sposób (algorytm) wyznaczania rozwiązania, natomiast kwestią drugorzędną są warunki istnienia zbioru rozwiązań dopuszczalnych. W większości przypadków przyjęte powyżej założenia (np. możliwość występowania buforów) sprawiają, że istnienie niepustego zbioru rozwiązań dopuszczalnych jest oczywiste. W praktyce istnieje jednak wiele czynników (niektóre z nich opisane zostaną w dalszych rozdziałach) sprawiających, że ocena tego typu jest równie kosztowana obliczeniowo jak poszukiwanie rozwiązań optymalnych.

### 1.1.2. Decyzyjny problem harmonogramowania cyklicznego

Opisywane do tej pory problemy harmonogramowania cyklicznego można podzielić na dziewięć grup rys. 1.13. Wszystkie są problemami optymalizacyjnymi i oprócz BCSP mają charakter NP-trudny (pomijając szczególne instancje problemów opisanych w (Levner et al. 2010)).

W podrozdziale 1.1 wskazano, że poszukiwanie rozwiązań optymalnych może odbywać się przy założeniu, że zbiór cyklicznych harmonogramów dopuszczalnych jest niepusty (patrz BCSP). Sytuacja, w której nie istnieje dopuszczalny harmonogram cykliczny może wystąpić w problemach typu no-buffer, no-wait.

Problemy harmonogramowania cyklicznego								
problemy harmonogramowania przedsiębiorzeń			problemy gniazdowe					
nieuwzględniające transportu międzyoperacyjnego		uwzględniające transport międzyoperacyjny				nieuwzględniające transportu międzyoperacyjnego		
BCSP	CPSP	CRPSP	CRJP	CRFP	CORP	COP	CJP	CFP
P	NP-trudne							

Rys. 1.13. Klasyfikacja problemów harmonogramowania cyklicznego

W większości problemów harmonogramowania cyklicznego zbiór rozwiązań dopuszczalnych jest niepusty. Wynika to głównie z natury rozważanych problemów a dokładniej mówiąc z charakteru narzuconych w nich ograniczeń. W problemie gniazdowym CJP, przepływowym CFP i otwartym COP zawsze da się zaproponować harmonogram, gdzie na każdej maszynie, w każdej chwili czasu jest wykonywana tylko jedna operacja. Podobnie jest z problemami harmonogramowania przedsiębiorzeń CPSP w skrajnym przypadku operacje każdego wykonania przedsięwzięcia można zrealizować szeregowo (tak jakby każde z nich wymagało tej samej maszyny). Stąd też bardzo atrakcyjne w przypadku tego typu problemów są metody związane z algorytmami populacyjnymi (algorytmy genetyczne, mrówkowe, itp.), algorytmami dobrze sprawdzającymi się w przypadku gdy przestrzeń, w której poszukuje się rozwiązań zawiera wiele rozwiązań dopuszczalnych (Dang et al. 2012).

Sytuacja komplikuje się w przypadku ograniczeń związanych z brakiem oczekiwania (no-wait) i/lub braku buforów (no-bufer) (patrz tab. 1.3). W pracy (Banaszak i Bocewicz 2011) opisany został przykład, wyznaczania planu zajęć, w którym zbiór rozwiązań dopuszczalnych jest zbiorem pustym.

**Przykład 2.** Ilustracja harmonogramowania planu zajęć

Dany jest zbiór przedmiotów  $P = \{P_1, P_2\}$  realizowanych w dwóch grupach studenckich  $E_1, E_2$ ,  $E = \{E_1, E_2\}$ . Zakłada się, że zajęcia z przedmiotów  $P_1, P_2$  mogą być realizowane tylko w oddzielnych grupach przez różnych wykładowców. Znane są czasy trwania prowadzonych zajęć:  $T_1 = (t_{1,1}, t_{1,2}) = (1, 2)$  (czas trwania zajęć  $P_1$ , dla grupy  $E_1$  wynosi 1 u.j.c., a czas trwania zajęć  $P_2$  wynosi 2 u.j.c.),  $T_2 = (t_{2,1}, t_{2,2}) = (3, 2)$ . Przyjmuje się, że między zajęciami nie występują przerwy (tzw. „okienka”), a zajęcia dla obu grup rozpoczynają się o tej samej godzinie. Poszukiwana jest odpowiedź na pytanie: Czy istnieje plan zajęć spełniający narzucone wyżej ograniczenia?

Tak sformułowany problem jest szczególnym przypadkiem cyklicznego problemu otwartego COP (wersja decyzyjna). Grupy wykładowe  $E = \{E_1, E_2\}$  odpowiadają zadaniom  $Z = \{Z_1, Z_2\}$ , zajęcia operacjom  $O_1 = \{o_{1,1}, o_{1,2}\}$ ,  $O_2 = \{o_{2,1}, o_{2,2}\}$ , przedmioty  $P = \{P_1, P_2\}$  maszynom  $M = \{m_1, m_2\}$ . Problem formalnie definiowany jest następująco:

*Dane są:*  $x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$  – terminy rozpoczęcia zajęć (operacji:  $o_{1,1}, o_{1,2}, o_{2,1}, o_{2,2}$ ) kolejno dla: grupy  $E_1$  przedmiot  $P_1$ ; grupy  $E_1$  przedmiot  $P_2$ ;  $E_2 - P_1$ ;  $E_2 - P_2$ ; oraz  $\alpha$  – okres realizacji zajęć.

*Zakłada się że:*

$$\alpha, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \in \mathbb{N}$$

$$(x_{1,1} = x_{2,2}) \vee (x_{1,2} = x_{2,1}) \quad \left. \begin{array}{l} \text{zajęcia dla obu grup rozpoczynają się o tej samej} \\ \text{godzinie} \end{array} \right\}$$

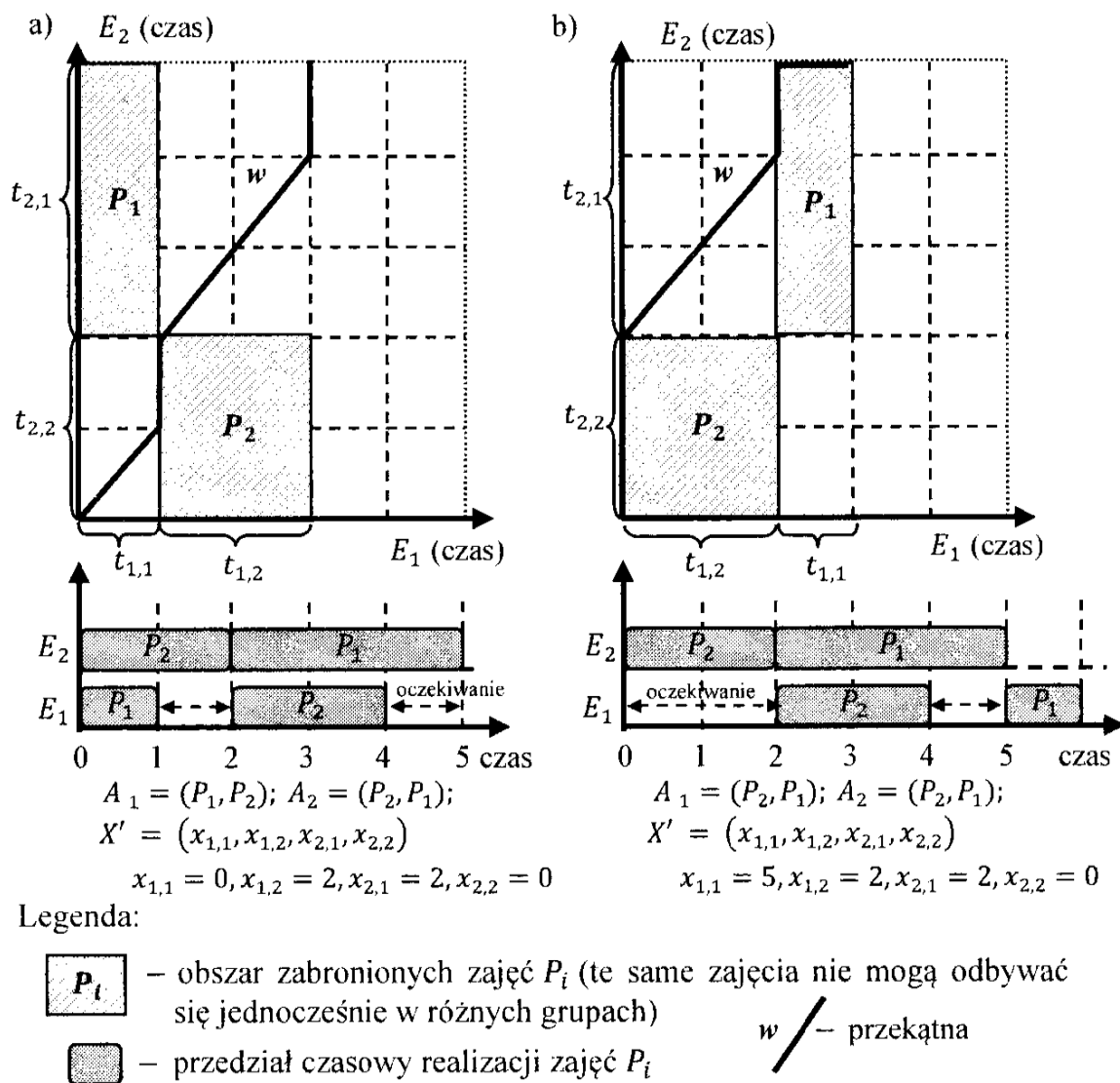
$$\left. \begin{array}{l} (x_{1,1} = x_{1,2} + t_{1,2}) \vee (x_{1,2} = x_{1,1} + t_{1,1}) \\ (x_{2,1} = x_{2,2} + t_{2,2}) \vee (x_{2,2} = x_{2,1} + t_{2,1}) \\ (x_{2,1} \geq x_{1,1} + t_{1,1}) \vee (x_{1,1} \geq x_{2,1} + t_{2,1}) \\ (x_{2,2} \geq x_{1,2} + t_{1,2}) \vee (x_{1,2} \geq x_{2,2} + t_{2,2}) \end{array} \right\} \begin{array}{l} \text{zajęcia realizowane są bez tzw.} \\ \text{„okienek”} \\ \text{te same przedmioty nie mogą być} \\ \text{jednocześnie realizowane w różnych} \\ \text{grupach} \end{array}$$

$$\left. \begin{array}{l} (x_{1,1} \geq x_{1,2} + t_{1,2} - \alpha) \wedge (x_{2,2} \geq x_{2,1} + t_{2,1} - \alpha) \vee \\ (x_{1,2} \geq x_{1,1} + t_{1,1} - \alpha) \wedge (x_{2,1} \geq x_{2,2} + t_{2,2} - \alpha) \end{array} \right\} \begin{array}{l} \text{zajęcia realizowane} \\ \text{są cyklicznie} \end{array}$$

*Poszukiwane są takie wartości  $\alpha, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$  spełniające powyższe ograniczenia. Innymi słowy poszukiwana jest odpowiedź na pytanie: czy istnieją takie terminy rozpoczęcia zajęć, które gwarantują ich bezokienkową realizację?*

Powyższy problem nie jest wprost problemem diofantycznym można go jednak zdekomponować na 32 alternatywne układy równań diofantycznych (Banaszak i Bociewicz 2011). Pozytywne rozwiązanie choć jednego z nich pozwala określić cykliczny harmonogram bezokienkowy i tym samym określić przestrzeń rozwiązań dopuszczalnych. Rozwiązanie tak określonego problemu (ocena rozstrzygalności układów równań diofantycznych) niesie jednak za sobą konieczność przeglądu pełnego wykładniczej liczby wariantów (wartości  $\alpha, x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$ ). W tym celu wykorzystano następującą jego reprezentację.



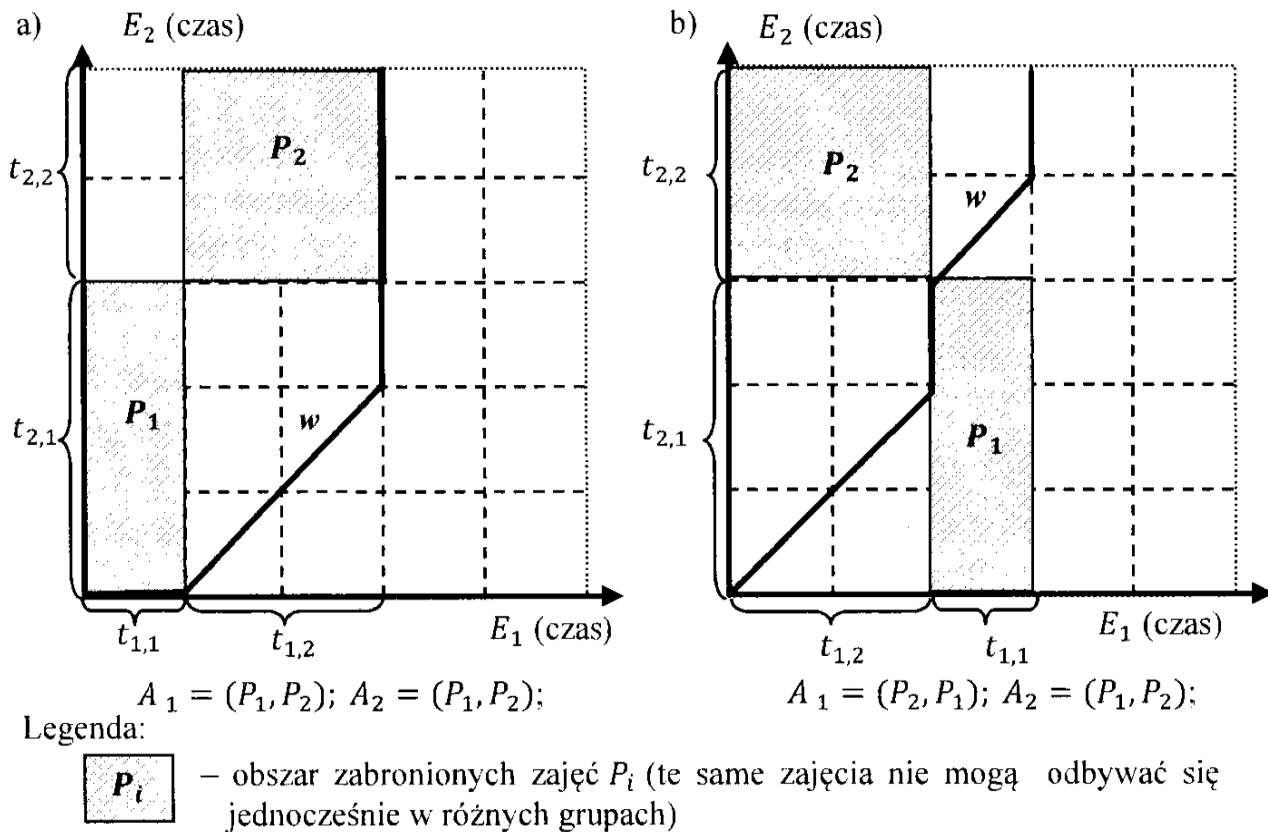


Rys. 1.14. Możliwe rozkłady zajęć preferujące grupę  $E_2$  wraz z odpowiadającymi im diagramami Gantta

Niech sekwencje  $A_1$  i  $A_2$  określają porządek prowadzenia przedmiotów odpowiednio dla grup  $E_1$  i  $E_2$ . Porządki te determinują możliwe wartości terminów rozpoczęcia operacji  $x_{i,j}$ . Harmonogram cykliczny można uzyskać w wyniku przeglądu kolejnych wariantów sekwencji  $A_1$  i  $A_2$ . W tym celu rozważyć należy cztery przypadki:  $A_1 = (P_1, P_2)$ ,  $A_2 = (P_2, P_1)$ ;  $A_1 = (P_2, P_1)$ ,  $A_2 = (P_2, P_1)$ ;  $A_1 = (P_1, P_2)$ ,  $A_2 = (P_1, P_2)$  oraz  $A_1 = (P_2, P_1)$ ,  $A_2 = (P_1, P_2)$ .

Do analizy rozważanych wariantów wykorzystano metodę graficzną (Bocewicz i Bzdyra 2005). Metoda ta jest zwykle wykorzystywana przy planowaniu zajęć dla dwóch grup (choć można rozszerzyć ją na wariant  $> 2$  (Bocewicz i Bzdyra 2005)). W układzie współrzędnych na osiach czasu, odpowiadających kolejnym grupom  $E_1$  i  $E_2$ , rozmieszczane są przedziały czasowe realizacji zajęć. Przedziały rozmieszczane są wzdłuż osi jeden za drugim, zgodnie z przyjętymi porządkami  $A_1$  i  $A_2$ . Wewnątrz wykresu umieszczane są obszary zabronione (prostokąty), odpowiadające kolejnym przedmiotom  $P_1$  i  $P_2$ , w taki sposób by rzuty obszarów na każdą z osi pokrywały się z przedziałem realizacji danego przedmiotu. Przyjęty porządek realizacji zajęć jest „bezokienkowy” jeżeli w tak powstałym wykresie można poprowadzić przekątną, prostą pod kątem  $45^\circ$  przez środek układu, tak by nie miała ona części wspólnej

z obszarami zabronionymi. Jeżeli nie jest to możliwe to każde załamanie prostej oznacza konieczność uwzględnienia przerwy w prowadzeniu zajęć (a zatem istnienie okienek).



Rys. 1.15. Przestrzenie obszarów zabronionych preferujące grupę  $E_1$

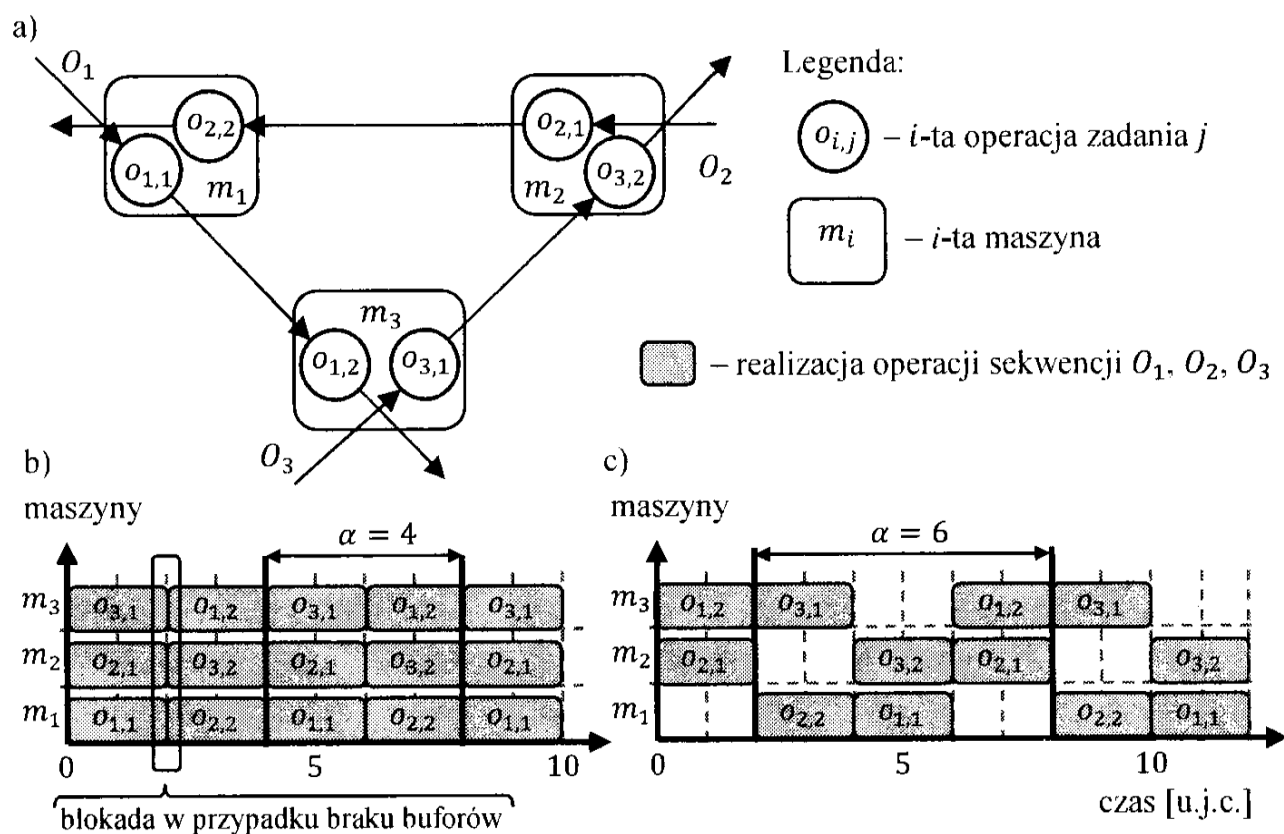
Rys. 1.14 ilustruje przestrzenie obszarów zabronionych (układ współrzędnych  $E_1, E_2$ ) oraz diagramy Gantta harmonogramów odpowiadających możliwym rozkładom zajęć preferującym zajęcia grupy  $E_2$  (plan zajęć tej grupy nie zawiera okienek). Przyjmując, że każde z zajęć prowadzone jest przez jednego wykładowcę, pierwszy z planów przedstawiony na rys. 1.14a gwarantuje ciągły przebieg tylko dla wykładowcy prowadzącego zajęcia  $P_2$ , z kolei drugi z planów przedstawionych na rys. 1.14b gwarantuje ciągły przebieg zajęć prowadzonych już przez obu wykładowców (jednak prowadzi do oczekiwania grupy  $E_1$ ). Z kolei rys. 1.15a ilustruje możliwy plan zajęć preferujący grupę  $E_1$  kosztem grupy  $E_2$ , patrz rys. 1.15b. W żadnym z rozważanych przypadków nie jest możliwe przeprowadzenie prostej pod kątem  $45^\circ$  – występują załamania zatem nie istnieje harmonogram spełniający zadane ograniczenia.

Przedstawiony przykład pokazuje sytuację, w której nie istnieje harmonogram gwarantujący brak oczekiwania zarówno dla prowadzących zajęcia jak i dla grup wykładowczych. Oznacza to, że nie w każdym przypadku istnieją rozwiązania dopuszczalne.

Czy w kontekście opisywanej klasy problemów istnieją zatem warunki gwarantujące niepustą przestrzeń rozwiązań dopuszczalnych? Lub inaczej, czy istnieje taka postać zbioru parametrów charakteryzujących problem, dla której istnieć będą rozwiązania dopuszczalne?

W przypadku BCSP (patrz podrozdział 1.1) takim warunkami są warunki poprawności digrafu  $G$  (związane z parametrami określonymi w (7), (8)) a w przypadku przedstawionego przykładu jest to warunek równości czasów trwania prowadzonych zajęć:  $t_{1,1} = t_{1,2} = t_{2,1} = t_{2,3}$  (wyklucza się w ten sposób okienka między zajęciami – patrz harmonogram rys. 1.14a). Rozwiązanie to sprawdza się jednak tylko w tym jednym prostym przypadku.

Przykładowo na rys. 1.16 przedstawiono problem gniazdowy CJP wraz z harmonogramem rys. 1.16b gwarantującym brak oczekiwania tak między operacjami każdego zadania, jak i pomiędzy samymi zadaniami. Innymi słowy na żadnej maszynie nie dochodzi do przestojów. Harmonogram taki nie jest jednak dopuszczalny w sytuacji stosowania ograniczenia zakładającego brak buforów (no-bufer). Ograniczenia braku buforów i wzajemnego wykluczania powodują, że zadanie nie zwalnia zajmowanej maszyny aż do momentu uzyskania dostępu do kolejnej (żądaney) maszyny.



**Rys. 1.16.** Przykład problemu CJP z trzema zadaniami a), harmonogram bez oczekiwania b), harmonogram bez blokad c)

W szczególnych sytuacjach, tego typu oczekiwanie wielu zadań, może prowadzić do ich wzajemnego zablokowania. Na rys. 1.16b przedstawiono sytuację, w której operacje  $o_{3,1}$ ,  $o_{2,1}$ ,  $o_{1,1}$  oczekują wzajemnie na zwolnienie zajmowanych przez siebie zasobów. W przypadku braku buforów realizacja kolejnych operacji jest niemożliwa – żadne z zadań nie otrzymuje dostępu do żądanych zasobów przez co nie zwalnia zasobów aktualnie zajmowanych.

Harmonogram pozwalający na uniknięcie blokad przedstawia z kolei rys. 1.16c, nie spełnia on jednak warunku braku oczekiwania między kolejną realizacją zadań (w ramach zadań operacje są nadal wykonywane bez oczekiwania). Oznacza to,

że spełnienie jednego ograniczenia wyklucza drugie co w dalszej konsekwencji oznacza, że nie jest możliwe znalezienie bezblokadowego harmonogramu z jednoczesnym brakiem przestojów na maszynach. Świadomość występowania tego typu sytuacji jest niezbędna w przypadku rozwiązywania problemów o charakterze optymalizacyjnym. Poszukiwanie rozwiązania optymalnego w przestrzeni niezawierającej rozwiązań dopuszczalnych sprowadza się do jej całkowitego przeglądu. Rozwiązywanie takiego problemu odpowiada problemowi decyzyjnemu z odpowiedzią NIE (Hopcroft et al. 2006; Papadimitrou 1994). Opisane przykłady są bardzo proste, bardziej złożone problemy odwzorowujące rzeczywiste systemy produkcyjne charakteryzują się zwykle zbiorem wielu ograniczeń. Ocena ich niesprzeczności, czy też ocena parametrów systemu, dla których opisujące go ograniczenia będą niesprzeczne, jest równoznaczna z oceną rozstrzygalności równań diofantycznych.

Brak gwarancji, a tym samym brak ogólnej metody rozstrzygalności istnienia rozwiązań dopuszczalnych stawia pod znakiem zapytania zasadność formułowania tego typu problemów jako problemów optymalizacyjnych. Rozważane problemy powinny mieć charakter decyzyjny to znaczy powinny odnosić się do istnienia rozwiązań dopuszczalnych, lub do istnienia takich parametrów problemów, które gwarantują, że rozwiązania dopuszczalne będą istniały.

Niech  $\pi(V, C, Q)$  oznacza problem harmonogramowania cyklicznego opisany przez zbiór parametrów  $V$  (zawierający np. zbiór zadań  $Z$ , zbiór operacji  $O$ , digraf ograniczeń kolejnościowych  $G$ , czasy wykonania operacji, zbiór maszyn  $M$ , itp.), ograniczenia  $C$  (brak oczekiwania, brak buforów, brak przestojów itp.), pytanie  $Q$  (np. jaki harmonogram  $X$ , minimalizuje okres  $\alpha$  realizacji zadań?).

Ze względu na rodzaj stawianego pytania  $Q$ , wyróżnia się dwa podejścia konstruowania problemów decyzyjnych (Bubnicki 1999; Bocewicz i Banaszak 2009) związanych z harmonogramowaniem cyklicznym:

1. podejście typu „w przód”, w którym problem  $\pi$ , nazywany dalej **problemem analizy**, sprowadza się do pytania:

Czy istnieje cykliczny harmonogram  $X$  spełniający ograniczenia  $C$  problemu opisanego zdany zbiorem parametrów  $V$ ?

2. podejście typu „wstecz”, w którym problem  $\pi$  nazywany dalej **problemem syntezy**, sprowadza się do pytania:

Czy istnieje taka postać parametrów  $V$  charakteryzujących problem, która gwarantuje istnienie cyklicznego harmonogramu  $X$  spełniającego ograniczenia  $C$ ?

Wyznaczenie zbioru rozwiązań dopuszczalnych (podejście w przód) lub wartości parametrów  $V$  (podejście wstecz), dla których taki zbiór istnieje, pozwala w kolejnym etapie poszukiwać rozwiązań, które optymalizują zadane kryteria. W tym ujęciu udzielenie odpowiedzi na jedno z powyższych pytań stanowi podstawę do rozwiązywania problemu optymalizacyjnego. Oczywiście dla wielu problemów (choćby dla podstawowych sformułowań CJP, CFP, CPSP) udzielenie odpowiedzi na tego typu pytania jest z góry pozytywne. Istnieje jednak cały szereg problemów, w których

ocena istnienia harmonogramu cyklicznego  $X$  nie jest tak oczywista. Problemy takie rozważane były w pracach (Bocewicz et al. 2007, 2009a, 2009d, 2011a, 2011b, 2011c, 2011d, 2012a, 2012b; Wójcik 2007). Opisowi i analizie tego typu problemów poświęcona jest znaczna część tej pracy.

## 1.2. Problemy harmonogramowania cyklicznego w systemach współbieżnych procesów

Głównym czynnikiem ograniczającym liczbę rozwiązań dopuszczalnych jest zazwyczaj możliwość występowania blokad (w przykładzie z rys. 1.16. przyczyną powstania blokad był brak buforów składowania międzystanowiskowego). Zjawisko to odgrywa kluczową rolę w problemach harmonogramowania, występujących w systemach współbieżnych procesów cyklicznych (SWPC). Systemy te stanowią podklasę Dyskretnych Systemów Zdarzeniowych (Banaszak et al. 2008).

Podstawowym pojęciem SWPC jest proces. Przez **proces** rozumiany jest uporządkowany w czasie ciąg zmian/zdarzeń/stanów (w dalszej części używane będzie pojęcie stan) zachodzących po sobie, zmian właściwości lub wartości zmiennych opisujących dany obiekt. Każdy kolejny stan systemu spowodowany jest bądź to przez poprzedni stan bądź też przez oddziaływanie zewnętrzne (Fokkink 1999; Silberschatz et al. 2004; Ben-Ari 2005; Banaszak et al. 2008). **Proces cykliczny** jest to proces, którego każdy stan powtarzany jest w stałych odcinkach czasu, nieskończoną ilość razy.

Oprócz wyżej przedstawionego podziału, stosuje się też podział procesów na: procesy ciągłe i dyskretnie (Banaszak 1991; Fokkink 1999; Reveliotis 2005). **Procesy ciągłe** to takie, gdzie na dowolnym skończonym odcinku czasu wyróżnić można nieskończoną liczbę stanów, przy czym różnice między nimi mogą być nieskończenie małe. **Procesy dyskretne**, z kolei to takie, w których wyróżnić można tylko skończoną liczbę stanów, przy czym różnice między nimi mają skończoną wartość.

Dalsze rozważania ograniczają się do klasy **cyklicznych sekwencyjnych procesów** dyskretnych, w których relacją determinującą kolejność występowania stanów jest relacja porządku liniowego. Przykładem cyklicznego procesu sekwencyjnego jest na przykład pojedyncza linia metra, w której pociąg sekwencyjnie przemieszcza się między kolejnymi stacjami, tzn. w każdej chwili wykonywana jest tylko jedna operacja (Silberschatz et al. 2004).

Procesy w systemach procesów cyklicznych mogą być realizowane równolegle lub współbieżnie. Dwa procesy są **równoległe** (Ułasiewicz 2007) jeżeli jeden z nich rozpoczyna się (każda kolejna operacja/akcja tego procesu) przed zakończeniem drugiego i wykonywane są jednocześnie z wykorzystaniem oddzielnych zasobów. Współbieżne wykonanie procesów oznacza z kolei, że jeden proces rozpoczyna się przed zakończeniem innego i związane jest z ograniczonym dostępem procesów do współdzielonych zasobów (Banaszak et al. 2008). Inaczej mówiąc, dwa procesy traktowane są jako **współbieżne**, jeżeli ich wykonaniu towarzyszy konieczność wzajemnej interakcji. Przykładem może być tutaj współdzielenie przez pociągi (procesy) peronów na stacjach. Równoczesna realizacja wielu procesów (zadań) w oparciu o wspólne zasoby ma na celu zwiększenie efektywności działania systemów, w szczególności skró-

cenie czasu realizacji zadań (Iszkowski i Maniecki 1982; Krishnamurthy 1989; Ben-Ari 2005).

W przedstawionym kontekście przez **system współbieżnych procesów cyklicznych** SWPC rozumiany jest zbiór procesów  $P = \{P_i \mid i = 1 \dots ln\}$  realizujących cyklicznie operacje w oparciu o zbiór wspólnie wykorzystywanych zasobów  $R = \{R_k \mid k = 1 \dots lk\}$ . Przyjmuje się, że system procesów cyklicznych dodatkowo charakteryzują:

- $C(R_k)$  – liczba dostępnych jednostek zasobu  $R_k$ , (tak zwana **pojemność zasobu**  $C(R_k) \in \mathbb{N}^+$ ),
- $O_i = (o_{i,1}, o_{i,2}, \dots, o_{i,j}, \dots, o_{i,lr(i)})$  – sekwencja **operacji** realizowanych przez proces  $P_i$ , gdzie  $lr(i)$  – liczba operacji procesu  $P_i$ .

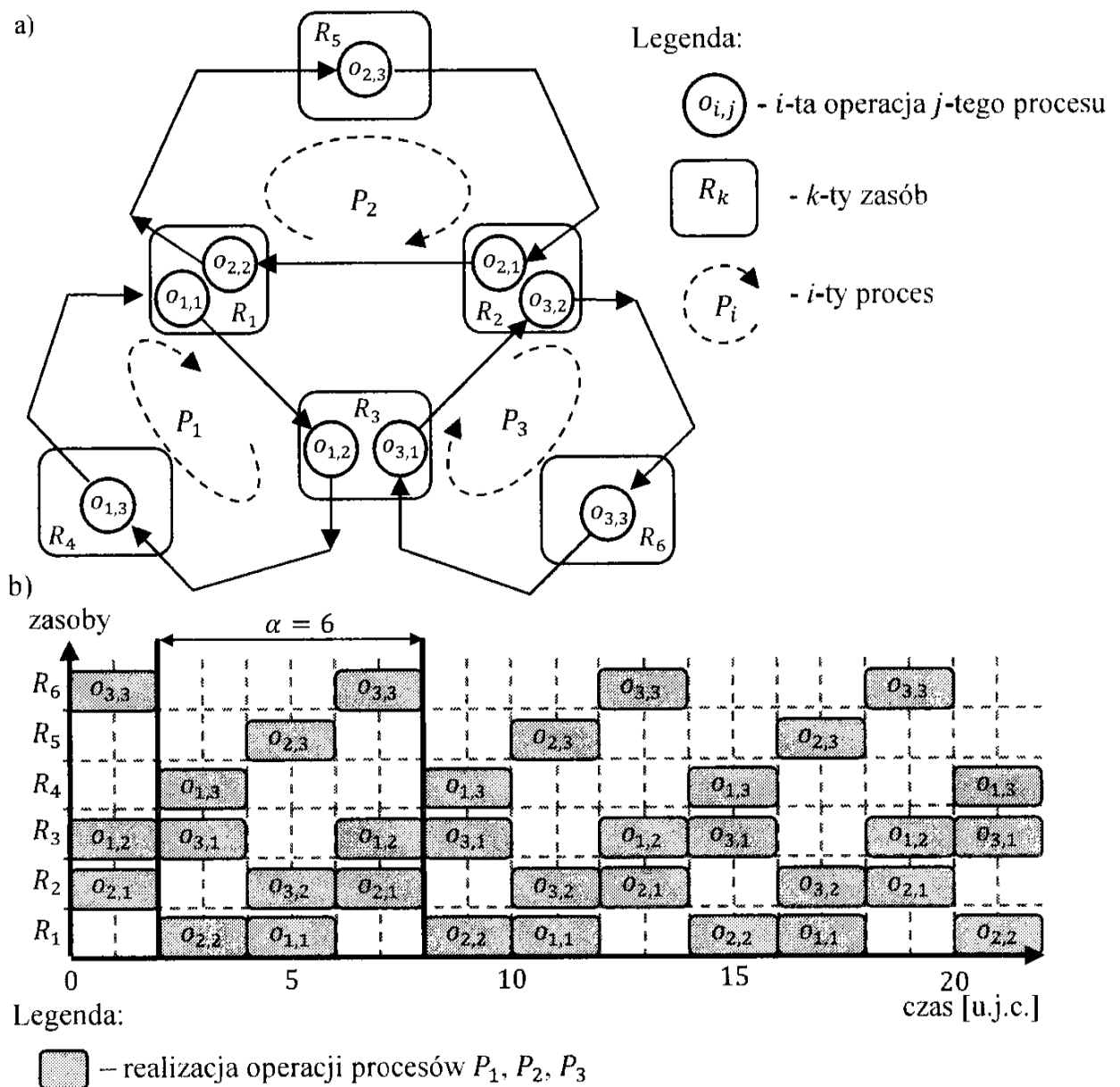
Każda operacja  $o_{i,j}$  procesu  $P_i$  realizowana jest w oparciu o zasoby określone przez **marszrutę procesu**  $p_i = (p_{i,1}, \dots, p_{i,j}, \dots, p_{i,lr(i)})$ , gdzie  $p_{i,j} \in R$  oznacza zasób wykorzystywany (wykorzystanie zasobu wiąże się z zajęciem tego zasobu przez proces) do realizacji operacji  $o_{i,j}$ . W ogólności przy realizacji operacji  $o_{i,j}$  dopuszcza się możliwość wykorzystania wielu zasobów, jednak takie przypadki nie będą dalej rozważane. Zakłada się, że do wykonania operacji  $o_{i,j}$  niezbędnych jest  $w(R_k, o_{i,j})$  jednostek zasobu  $R_k$  (gdzie  $R_k$  jest elementem  $p_i$ ).

Procesy są realizowane cyklicznie – to znaczy, że po wykonaniu wszystkich operacji  $O_i$  sekwencja rozpoczyna się na nowo. Innymi słowy następnikiem ostatniej operacji  $o_{i,lr(i)}$  sekwencji  $O_i$  jest operacja pierwsza  $o_{i,1}$ . Każdej operacji  $o_{i,j}$  przypisany jest czas wykonania  $t_{i,j} \in \mathbb{N}^+$  oraz moment jej rozpoczęcia w  $k$ -tym wykonaniu  $x_{i,j}(x) \in \mathbb{N}$ .

W tym miejscu należy zwrócić uwagę na podobieństwo tak definiowanego SWPC z opisem problemu gniazdowego CJP (prezentowanego na przykład na rys. 1.6). Procesy  $P_i$  można utożsamiać z zdaniami  $Z_i$ , którym podobnie jak  $P_i$  przypisywane są sekwencje operacji  $O_i$ , z kolei zasoby  $R_k$  można traktować jako maszyny  $m_k$  niezbędne do realizacji tych operacji. Różnica polega na tym, że w ogólnym przypadku porządek  $P_i$  nie musi być liniowy a operacje mogą korzystać z wielu zasobów mających różną od 1 pojemność  $C(R_k)$ . Ponadto, przyjmuje się, że procesy wykonywane są w trybie współbieżnym, tzn.:

- dostęp do współdzielonych zasobów odbywa się zgodnie z zasadą wzajemnego wykluczania (ang. mutual exclusion),
- zasoby są niewywłaszczalne i nie mogą być odebrane przed zakończeniem operacji,
- proces zajmuje wykorzystywany zasób do momentu rezerwacji kolejnego zasobu niezbędnego do wykonania następnej operacji.

W ujęciu procesowym cykliczna realizacja zadań  $Z_i$  (problemów CJP, CFP, CPSP) może być interpretowana jako potokowa realizacja odpowiednich procesów cyklicznych  $P_i$ . Oznacza to, że w ogólnym przypadku problemy CJP, CFP, CPSP, mogą być modelowane przez systemy współbieżnych procesów cyklicznych.



Rys. 1.17. System współbieżnych procesów cyklicznych z trzema procesami a) dopuszczalny harmonogram cyklicznej realizacji operacji b)

Przykładem takiego systemu jest SWPC z rys. 1.17a (przyjęto, że graficzną reprezentacją procesów są cykle). Przedstawiony system jest jednym z możliwych rozszerzeń systemu przedstawionego na rys. 1.16. Zgodnie z wprowadzoną notacją jest on opisany przez:

- zbiór zasobów:  $R = \{R_1, R_2, R_3, R_4, R_5, R_6\}$ , gdzie:

pojemności zasobów mają postać:

$$C(R_1) = C(R_2) = \dots = C(R_6) = 1,$$

- zbiór procesów:  $P = \{P_1, P_2, P_3\}$ , gdzie:

sekwencje operacji procesów mają postać:

$$O_1 = (o_{1,1}, o_{1,2}, o_{1,3}), O_2 = (o_{2,1}, o_{2,2}, o_{2,3}), O_3 = (o_{3,1}, o_{3,2}, o_{3,3})$$

marszruty procesów mają postać:

$$p_1 = (R_1, R_3, R_4), p_2 = (R_1, R_5, R_2), p_3 = (R_2, R_6, R_3)$$

zapotrzebowania operacji na zasoby:

$$\begin{aligned} w(R_1, o_{1,1}) &= w(R_3, o_{1,2}) = w(R_4, o_{1,3}) = 1, \\ w(R_1, o_{2,1}) &= w(R_5, o_{2,2}) = w(R_2, o_{2,3}) = 1, \\ w(R_2, o_{3,1}) &= w(R_6, o_{3,2}) = w(R_3, o_{3,3}) = 1. \end{aligned}$$

Powyższy zbiór wielkości jest wykorzystywany do określania tzw. struktury SWPC. Przez **strukturę SWPC** rozumie się zazwyczaj taki zbiór parametrów, który charakteryzuje zasoby (ich liczbę, pojemność, itp.) oraz realizowane procesy (związane z nimi operacje, marszruty, zapotrzebowanie na zasoby, itp.).

Rys. 1.17b przedstawia przykład cyklicznego harmonogramu  $X$  realizacji operacji procesów zbioru  $P$ , harmonogramu będącego reprezentacją cyklicznego zachowania SWPC. Przez cykliczne **zachowanie SWPC** rozumiany jest dalej zbiór wielkości określających sposób realizacji operacji (momenty rozpoczęcia operacji, stany alokacji itp). Analogicznie jak w BCSP (zgodnie z (1) i (2)), zachowanie SWPC reprezentowane w postaci cyklicznego harmonogramu definiowane jest następująco:

$$X = ((X_1, \dots, X_{ln}), \alpha), \quad (11)$$

gdzie:

$X_i = (x_{i,1}, \dots, x_{i,j}, \dots, x_{i,lr(i)})$  – sekwencja terminów rozpoczęcia operacji procesu  $P_i$ ,

$x_{i,j} = x_{i,j}(0)$  – termin rozpoczęcia operacji dla  $k = 0$  wykonania procesu  $P_i$ ,  
 $x_{i,j}(k) = x_{i,j}(0) + \alpha \cdot k$ ,  $k \in \mathbb{C}$  (warunek jest słuszny tylko w przypadku 1-periodyczności),

$ln$  – liczba procesów,

$\alpha$  – okres harmonogramu  $\alpha \in \mathbb{N}^+$ .

W zależności od przyjętych założeń, przedstawione intuicje struktury i zachowania cyklicznego mogą być różnie definiowane; ich szczegółowe definicje przedstawiono w rozdziale 2.

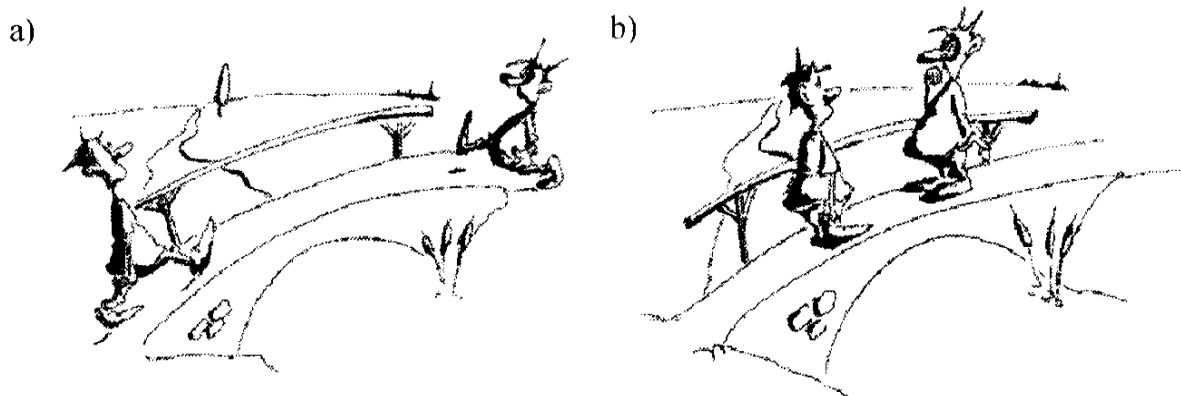
Jak już pokazano w przykładzie 2 oraz na rys. 1.16 harmonogram cykliczny spełniający stawiane oczekiwania nie zawsze jest osiągalny. Analogicznie jest w przypadku SWPC. Ograniczoność dostępnych, współdzielonych zasobów systemu, wymusza konieczność synchronizacji (koordynacji) współbieżnie przebiegających procesów w taki sposób, aby uniemożliwić ich wzajemnie niekorzystne oddziaływania (np. dostęp do współdzielonych zasobów w tym samym czasie).

Brak synchronizacji gwarantującej cykliczną realizację procesów (o zadanym cyklicznym harmonogramie  $X$ ), skutkuje sytuacjami prowadzącymi do zawieszenia pracy systemu, tzn. **blokad** (impasów), i/lub **zagłódzeń** procesów (tzn. brakiem dostępu procesów do żądanych zasobów).

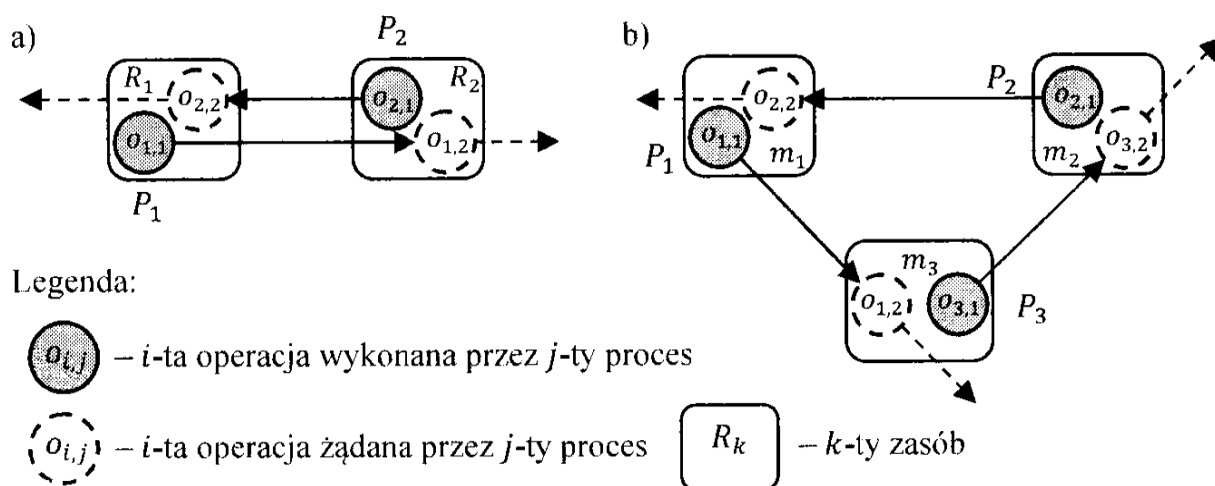
Stan blokady (impasu) występuje wówczas, gdy kilka współbieżnych procesów wzajemnie oczekuje na zwolnienie zajmowanych przez siebie zasobów. Źródłem blokad w systemach ze współdzielonymi zasobami jest między innymi konieczność spełnienia następującego wymogu: **proces może zwolnić zajmowany (już wykorzystany) zasób dopiero, gdy zdoła zająć zasób następny** (Dijkstra 1965; Coffman et al 1971; Banaszak i Krogh 1990; Ben-Ari 2005).



Przykład ilustrujący intuicję blokady przedstawiono na rys. 1.18. Przechodnie nadchodzący z przeciwnych stron (rys. 1.18a) do kładki mieszczącej tylko jedną osobę mogą ilustrować dwa niezależne (o tych samych priorytetach) wykonujące się procesy. Sytuacja, w której każdy przechodzień nastąpił na swój koniec kładki (rys. 1.18b) oznacza ich zastój – nie mogą kontynuować swoich marszrut.



Rys. 1.18. Ilustracja blokady procesów (Banaszak et al. 2008)



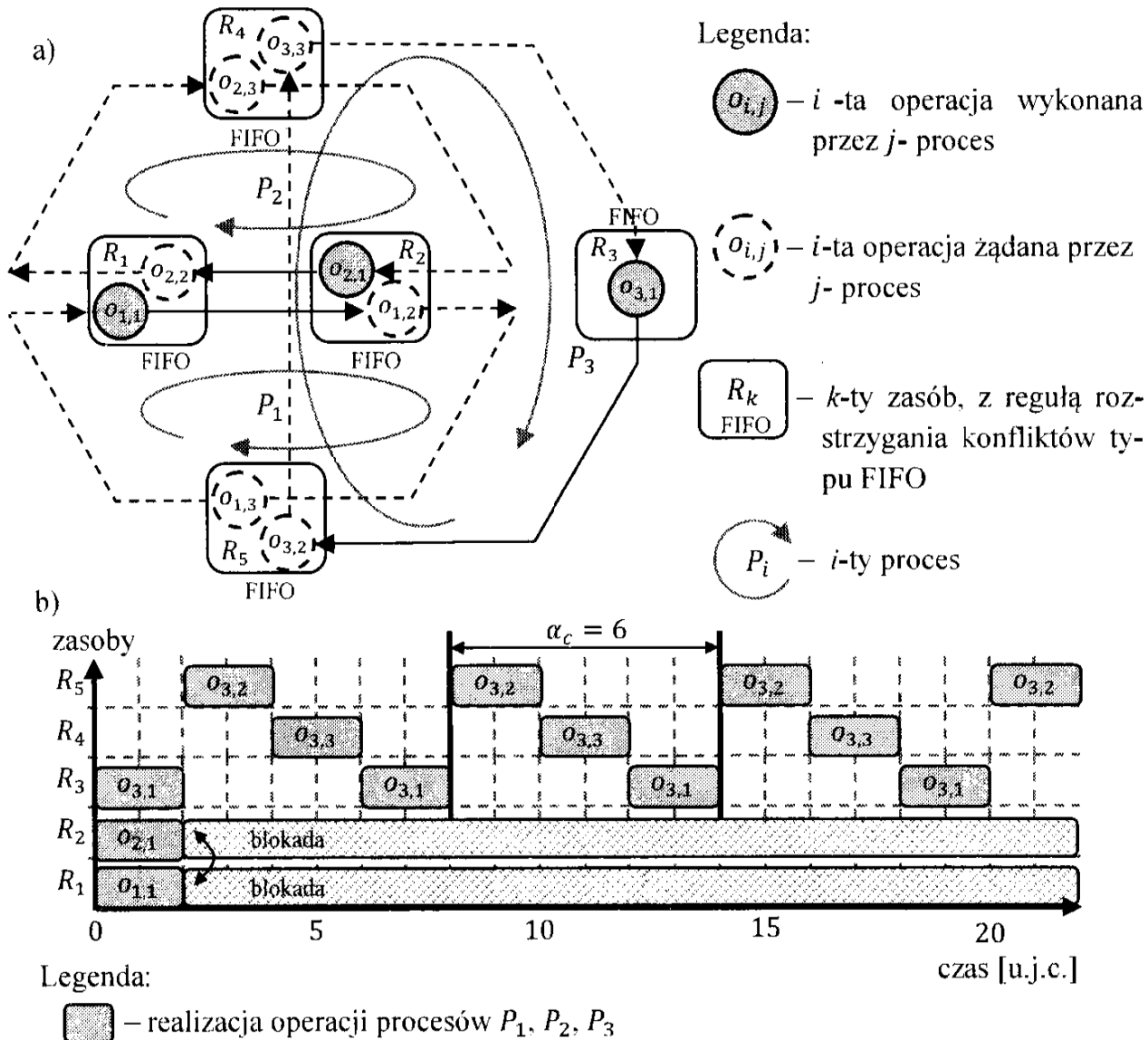
Rys. 1.19. Przykłady blokad: w której uczestniczą dwa procesy a) w której uczestniczą trzy procesy b)

W sposób bardziej formalny, stan blokady ilustruje następujący przykład. Dany jest proces  $P_1$ , zajmujący zasób  $R_1$ , żądający dostępu do zasobu  $R_2$ . Zasób ten jest zajęty przez proces  $P_2$ , który z kolei żąda dostępu do zasobu  $R_1$ , zajmowanego właśnie przez proces  $P_1$ . Powstaje zamknięty łańcuch żądań zasobowych – stan blokady, który zatrzymuje poprawne funkcjonowanie systemu (rys. 1.19a). Przykład blokady z udziałem trzech procesów przedstawia rys. 1.19b, odpowiada on blokadzie z harmonogramu przedstawionego na rys. 1.16b – zamknięty łańcuch żądań zasobowych składa się w tym przypadku z trzech operacji.

**Zagłodzenie** (ang. starvation) ma miejsce w przypadku, gdy pewien proces, na przykład z powodu posiadania niższego niż inne procesy priorytetu wykonania, nie może uzyskać dostępu do współdzielonego zasobu. Korzystając raz jeszcze z analogii do ruchu ulicznego, sytuację zagłodzenia dobrze oddaje próba wjazdu samochodu z ulicy podporządkowanej na ruchliwą ulicę główną. Wystąpienie zagłodzenia impli-

kuje częściową cykliczność systemu, w której niektóre operacje, np. występujące w zagłódzonych procesach, nie są realizowane cyklicznie.

Zjawisko częściowej cykliczności może być również spowodowane wystąpieniem blokady, w której bierze udział część procesów SWPC. Przykład takiej sytuacji przedstawia rys. 1.20. Procesy  $P_1$  i  $P_2$  blokują się wzajemnie, natomiast proces  $P_3$  może być bez przeszkód realizowany cyklicznie (z okresem  $\alpha_c = 6$ ).



**Rys. 1.20.** SWPC charakteryzujący się częściową cyklicznością: struktura a), harmonogram realizacji operacji b)

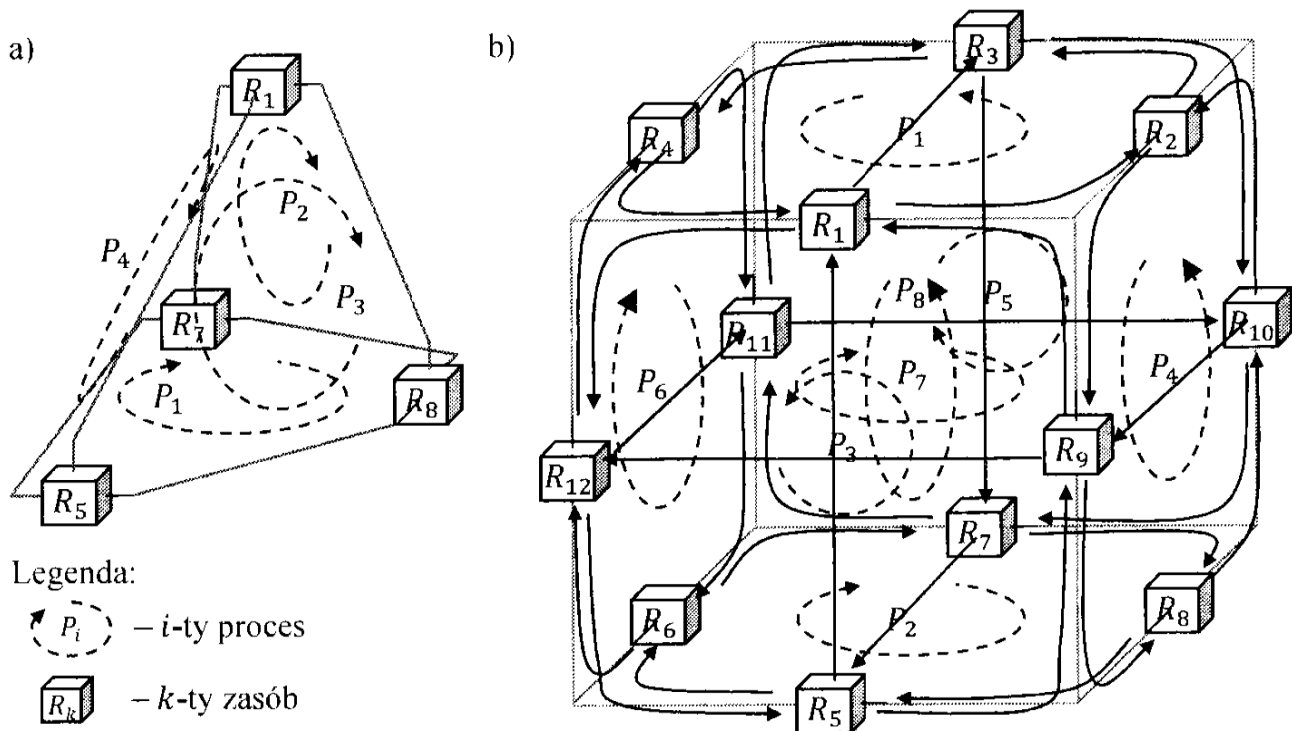
W przedstawionym przykładzie zastosowano regułę synchronizacji dostępu do zasobów typu FIFO (first in first out). Wykorzystywane w praktyce mechanizmy synchronizacji wyrażają się w różnych koncepcjach implementacji zasady wzajemnego wykluczenia, tzn. koncepcjach sterowania dostępem do współdzielonych zasobów. Korzystając z takiego zasobu proces wchodzi do tzw. sekcji krytycznej (ang. critical region). Zgodnie z zasadą wzajemnego wykluczenia, w tym samym czasie w sekcji krytycznej może działać tylko jeden proces. W ogólnym przypadku w sekcji krytycznej (determinującej ograniczenia zasobowe systemu) może działać więcej procesów, nie więcej jednak niż umożliwia to zakładana, dopuszczalna pojemność współdzielonego zasobu (Raynal 1986; Milner 1995; Silberschatz et al. 2004). Przykładem takie-

go mechanizmu synchronizacji są statyczne reguły priorytetowania  $\sigma_k$  określające stały porządek dostępu procesów do zasobów  $R_k$  (np.  $\sigma_k = (P_A, P_B, P_C, P_B)$  jest sekwencją determinującą ustalony porządek obsługi procesów na zasobie  $R_k$  w kolejności:  $\dots, P_A, P_B, P_C, P_B, P_A, P_B, \dots$ ).

Jak widać na prezentowanych przykładach zjawisko występowania blokad i zagłodeń jest determinowane przez implementowany mechanizm synchronizacji procesów. Oznacza to, że istnienie cyklicznego harmonogramu SWPC jest zależne, od przyjętego mechanizmu synchronizacji (zmieniając mechanizm synchronizacji w SWPC z rys. 1.20 można uzyskać harmonogram cykliczny).

W sposób naturalny nasuwa się pytanie czy istnieją takie SWPC, dla których żaden mechanizm synchronizacji (rozstrzygnięcia konfliktów) nie gwarantuje cyklicznego zachowania? Powracając do analogii równań diofantycznych istnienie tego typu struktur jest równoznaczne z istnieniem równań sprzecznych.

Odpowiedź na takie pytanie jest pozytywna, istnieją takie systemy a ich dwa przykłady przedstawiono na rys. 1.21. System z rys. 1.21a zawiera 4 cykliczne procesy realizowane na 4 zasobach (przyjęta pojemność zasobów wynosi 1).



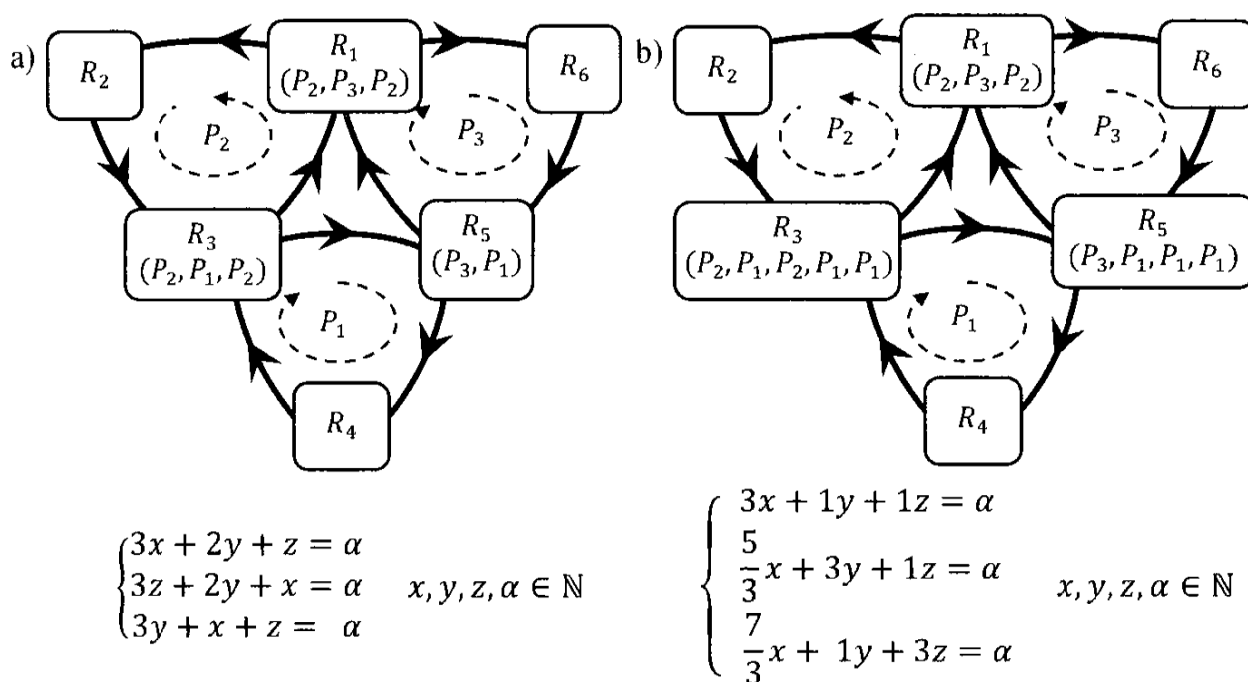
**Rys. 1.21.** Przykłady SWPC prowadzących do blokad dla każdego stanu początkowego i dowolnego mechanizmu synchronizacji: system z czterema procesami i czterema zasobami a), system z ośmioma procesami i dwunastoma zasobami (Bocewicz et al. 2012a) b)

W systemie występuje blokada, gdyż w każdym momencie każdy z zasobów jest zajmowany przez jeden z procesów. Brak wolnych zasobów uniemożliwia „przemieszczenie” się (zmianę stanu) procesów. Drugi przykład (rys. 1.21b) przedstawia system zawierający 8 procesów i 12 zasobów (z czego w każdym momencie osiem z zasobów jest zajętych przez procesy), układ marszrut jest jednak taki, że w którymś „narożniku” zawsze dochodzi do blokady typu rys. 1.19b bez względu na przyjęty stan początkowy oraz mechanizm synchronizacji procesów.

Brak harmonogramu cyklicznego jest konsekwencją braku rozwiązań odpowiednich równań diofantycznych. W ogólności już sama konstrukcja tego typu równań dla SWPC, z określonym mechanizmem synchronizacji, bywa dość trudna i silnie zależna od parametrów charakteryzujących problem. Przykład dwóch systemów (wykorzystujących statyczne reguły wyboru priorytetu  $\sigma_k$  obsługi żądań zasobowych) opisanych przez układy równań diofantycznych ilustruje rys. 1.22. Układy równań opisują związek występujący między okresem  $\alpha$ , a liczbą wykonań procesów w ramach jednego cyklu:  $x$  (liczba wykonań procesu  $P_1$ ),  $y$  (liczba wykonań procesu  $P_2$ ),  $z$  (liczba wykonań procesu  $P_3$ ).

Oznacza to, że problemy harmonogramowania w systemach współbieżnych procesów cyklicznych można rozważać w dwóch kategoriach:

- zapobiegania blokadom,
- unikania blokad.



Legenda:

- $R_k$  – zasób lokalny  $R_k$  (wykorzystywany tylko przez jeden proces)
- $R_k (P_A, P_B)$  – zasób współdzielony  $R_k$  z regułami priorytetowania  $\sigma_k = (P_A, P_B)$  (jako pierwszy obsługiwany jest proces  $P_A$  jako drugi  $P_B$ )

Rys. 1.22. Przykłady SWPC wraz z opisującymi je układami równań diofantycznych (Bocewicz i Banaszak 2010)

**Zapobieganie powstawaniu blokad** opiera się na wykorzystaniu takich zasad projektowania **struktury systemu** (struktura definiowana jest przez m.in.: zbiór zasobów  $R$ , marszrut procesów  $p_i$ , zapotrzebowań operacji na zasoby, itp.) lub zasad obsługi żądań zasobowych procesów, które uniemożliwiają spełnienie, co najmniej jednego z **warunków koniecznych dla powstania blokady** (w stanie blokady warunki te muszą być spełnione jednocześnie) podanych w pracach Coffman i inni (Coffman et al. 1971, Commoner 1972) oraz (Silberschatz et al 2004):

- wyłączności (wzajemnego wykluczania) – w danej chwili jednostka zasobu może być wykorzystywana przez tylko jeden proces;
- niewywłaszczalności (szeregowania bez wywłaszczania) – zasoby przydzielone procesowi nie mogą mu być odebrane przed zakończeniem wykonywania określonej operacji, tj. zasoby przydzielane procesom nie podlegają wywłaszczaniu;
- częściowego przydziału (przetrzymywania i czekania) – istnieje proces, który przetrzymuje przydzielone mu i potrzebne w dalszym ciągu zasoby, a jednocześnie oczekuje na zwolnienie zasobów przetrzymywanych przez inne procesy;
- czekania cyklicznego – istnieje zamknięty łańcuch procesów, z których każdy oczekuje na zasoby przetrzymywane przez poprzednika w łańcuchu.

Przykładem takich systemów są systemy przedstawione na rys. 1.22, w których warunek czekania cyklicznego nie jest spełniony. Z kolei w systemach z rys. 1.21 wszystkie z powyższych warunków są spełnione.

Znane są różne podejścia związane z tego typu klasą problemów np. algorytm liniowy Havendera (Havender 1968, Silberschatz et al 2004), oparty o zasadę zachowania liniowego porządku numerów zasobów przydzielanych do procesu, algorytmy implementujące podejście oparte na regule „wszystko albo nic” (Ben-Ari 2005, Raynal 1986), lub algorytmy wzajemnego wykluczania procesów w dostępie do strefy krytycznej Raynala (Raynal 1986, 1988). W pracach (Deitel et al. 2003; Silberschatz et al 2004), przedstawione zostały również metody oparte na zaprzeczeniu warunku *czekania cyklicznego*.

**Unikanie blokad** sprowadza się do analizy stanów danego SWPC i zarządzaniu przydziałami zasobowymi w taki sposób, aby eliminowane były stany prowadzące do blokady. Jest to podejście dynamiczne, w którym wykorzystuje się informację o aktualnym stanie systemu oraz przyszłym zachowaniu procesów w celu eliminacji stanów rozdziału zasobów, dla których spełnione są równocześnie warunki konieczne dla istnienia blokady (Coffman et al. 1971, 1973; Silberschatz et al 2004, Cho et al. 1995). W celu unikania blokad wykorzystywana jest procedura analizy bezpieczeństwa stanów systemu, która korzystając z informacji o przyszłych żądaniach zasobowych procesów sprawdza, czy istnieje sekwencja przydziałów zasobowych prowadząca do zakończenia wszystkich zadań.

Jednym ze sposobów unikania blokad jest przegląd zupełny dostępnej przestrzeni stanów. W ogólnym przypadku jest to jednak podejście obliczeniowo nieefektywne. Sprawdzenie czy dany rozdział zasobów nie prowadzi do blokady może wymagać analizy wykładniczej liczby stanów. Jednym z najbardziej znanych testów jest test oparty na metodzie bankiera (Dijkstra 1965; Habermann 1969; Silberschatz et al. 2004).

Problem unikania blokad można interpretować jako problem analizy (podejście „w przód”), w których dla znanych parametrów  $V$  struktury systemu (zbiór zasobów  $R$ , procesów  $P$ , mechanizm synchronizacji  $\theta$  itp.) i ograniczeń  $C$  (wzajemne wykluczanie procesów, niewywłaszczalność zasobów, itp.) badane jest jego zachowanie (istnienie cyklicznego harmonogramu  $X$  realizacji procesów). Formalnie **problem analizy systemów współbieżnych procesów cyklicznych** jest definiowany następująco:

*Dane są zbiory zasobów  $R$  o znanych pojemnościach, procesów  $P$  o znanych marszrutach  $p_i$  i sekwencjach operacji  $O_i$  (opisanych przez czasy wykonania  $T_i$  oraz zapotrzebowania na zasoby  $w(R_k, o_{i,j})$ ), a także mechanizm synchronizacji  $\Theta$ . Poszukiwany jest cykliczny harmonogram  $X$  realizacji procesów.*

Z kolei problemy zapobiegania blokadom można interpretować jako problem syntezy (podejście typu „wstecz”), w którym poszukiwane są parametry  $V$  struktury systemu gwarantujące istnienie zachowań cyklicznych spełniających zadane ograniczenia  $C$ .

Warto zauważyć, że problemy typu wstecz dotyczą poszukiwania mechanizmów synchronizacji gwarantujących bezblokadową pracę systemu dla wszystkich możliwych lub pewnych podzbiorów stanów początkowych (Bocewicz et al. 2007, 2009d, 2011c, 2011d; Bocewicz i Banaszak 2010, 2013a, 2013b). W tym kontekście **problem syntezy SWPC** przyjmuje postać:

*Dane są zbiory zasobów  $R$  o znanych pojemnościach procesów  $P$ , o znanych marszrutach  $p_i$  i sekwencjach operacji  $O_i$  (opisanych przez czasy wykonania  $T_i$  oraz zapotrzebowaniu na zasoby  $w(R_k, o_{i,j})$ ). Poszukiwany jest mechanizm synchronizacji  $\Theta$ , gwarantujący istnienie cyklicznego harmonogram  $X$  realizacji procesów  $P$ .*

W kontekście rozstrzygania konfliktów zasobowych odrębną grupą problemów, o której należałoby wspomnieć, są problemy wykrywania i likwidacji blokad. Metody wykrywania i likwidacji blokad opierają się na okresowym sprawdzaniu, czy w systemie procesów nie pojawił się stan blokady. W przypadku wykrycia stanu blokady jest realizowany algorytm jej likwidacji polegający na odłączaniu określonych zasobów od procesów wstrzymanych, aż do momentu, gdy kolejne uwolnienie zasobu wyprowadza system ze stanu blokady. Stosownym przykładem jest algorytm Menasce i Muntza (Raynal 1988, Silberschatz et al. 2004).

Omawiane dotychczas problemy odnoszą się do systemów klasy WPP. W problemach występujących w tego typu systemach, dostępna jest informacja o strukturze częściowych żądań zasobowych procesów, tj. zasobów żądanych i uwalnianych przez procesy podczas wykonywania określonych operacji. Dodatkowo zakłada się, że dostępna jest informacja o globalnym stanie rozdziału zasobów (znana jest aktualna informacja o zasobach przydzielonych wszystkim procesom) – metody oparte na tym założeniu nazywane są metodami scentralizowanymi (Cellary 1982; Raynal 1986, 1988; Silberschatz et al. 2004). Klasa WPP stanowi podklasę problemów PR ( $WPP \subset PR$ ) związanych z systemami rozproszonymi, w których zwykle nie jest dostępna informacja o strukturze procesów (ich marszrut) ponadto nie zawsze jest znany globalny stan systemu (Brzeziński 2001).

W klasie WPP można wyróżnić różne podklasy związane z problemami rozstrzygania konfliktów zasobowych. Odpowiednie problemy uwzględniają informację o strukturze procesów (liniowy lub częściowy porządek operacji) oraz o strukturze zasobów wykorzystywanych przez operacje (jeden rodzaj zasobu lub wiele rodzajów zasobów na operację; jedna jednostka zasobu lub wiele jednostek danego zasobu na

operację). Na tej podstawie można wyróżnić trzy parametry, umożliwiające definiowanie podproblemów w rozważanej klasie WPP:

- parametr  $or$ , opisujący porządek operacji w procesach; może on przyjmować dwie wartości  $or \in \{o_l, o_c\}$ , gdzie:  $o_l$  – liniowy porządek operacji w procesach (analogicznie jak w przypadku problemów CJP, CFP);  $o_c$  – częściowy porządek operacji w procesach (np. rozdzielanie i złączanie procesów jak w przypadku przedsięwzięć cyklicznych CPSP);
- parametr  $k$ , określający strukturę zbioru zasobów niezbędnych do wykonania operacji; może on przyjmować dwie wartości  $k \in \{k_s, k_m\}$ , gdzie:  $k_s$  – każda operacja korzysta z jednego rodzaju zasobu,  $k_m$  – każda operacja może korzystać z kilku rodzajów zasobów, należących do określonego zbioru zasobów;
- parametr  $j$ , opisujący sposób pobierania jednostek zasobów danego rodzaju wykorzystywanych w ramach operacji; może on przyjmować dwie wartości  $j \in \{j_u, j_g\}$ , gdzie:  $j_u$  – każda operacja korzysta z jednej jednostki każdego zasobu wymaganego przy realizacji tej operacji,  $j_g$  – każda operacja może korzystać z dowolnej liczby jednostek zasobu wymaganego przy realizacji tej operacji;

Wymienione założenia pozwalają zdefiniować w klasie WPP osiem klas, które w sposób symboliczny można opisać za pomocą notacji  $WPP|or, k, j$ , gdzie:  $or \in \{o_l, o_c\}$ ,  $k \in \{k_s, k_m\}$ ,  $j \in \{j_u, j_g\}$ . Klasy te zostały przedstawione w tab. 1.4.

Jeżeli przyjąć, że liniowy porządek stanowi szczególny przypadek częściowego porządku a wykorzystanie pojedynczych zasobów jest szczególnym przypadkiem wykorzystania wielu zasobów to można zauważyć, że problemy  $WPP|o_c, k_m, j_g$  stanowią najogólniejszą klasę problemów występujących w systemach klasy WPP, natomiast problemy  $WPP|o_l, k_s, j_u$  najbardziej szczegółową. Wynika stąd, że algorytmy opracowane dla klasy  $WPP|o_c, k_m, j_g$  mogą być wykorzystywane do rozwiązywania wszystkich pozostałych klas WPP.

W literaturze znanych jest wiele algorytmów umożliwiających rozwiązanie problemów klasy WPP, występujących w:

- systemach operacyjnych, komputerach wielozadaniowych, transmisji danych – są to głównie algorytmy bezblokadowej synchronizacji gwarantujące wzajemne wykluczania procesów (Dijkstra 1965; Raynal 1986, 1988; Ben-Ari 2005) oraz unikania blokad (Deitel et al. 2003; Silberschatz et al. 2004), wykorzystujące testy dopuszczalności stanów (Coffman i Denning 1973; Cellary 1981, 1982; Reisig 1985; Baccelli et al. 1992; Lautenbach i Thiagarajan 1979; Suraj 1984; Tazza 1987).
- Elastycznych Systemach Produkcyjnych (ESP) – są to algorytmy dedykowane dla problemów o liniowej strukturze procesów (Banaszak i Krogh 1990; Banaszak 1991; Fanti nad Zhou 2004; Zhou i Fanti 2004; Reveliotis 2005) lub częściowo uporządkowanej (Roszkowska 1990), problemów z operacjami wykorzystującymi jeden zasób (Banaszak i Krogh 1990; Roszkowska 1990; Fanti et al. 1997, 2000; Reveliotis et al. 1997; Lawley et al. 1998), lub wiele różnych zasobów jednocześnie (Abdallah i Elmaraghy 1998; Ezpeleta i Recalde 2004; Fanti 2004; Huang et al. 2001; Park i Reveliotis 2000, 2001; Reveliotis 1997; Wójcik et al. 1992).

Tabela 1.4. Klasa WPP (Banaszak et al. 2008)

Klasa	Opis	Źródła
$WPP o_l, k_s, j_u$	każdy proces jest liniowo uporządkowanym ciągiem kolejnych operacji, a każda operacja korzysta z jednej jednostki określonego zasobu	sieci komputerowe typu „store-and-forward” (SFN) (Gerla i Kleinrock 1980; Merlin i Schweitzer 1980) systemy ESP (Banaszak 1991; Narahari i Viswanadham 1985; Roszkowska E., Banaszak 1991)
$WPP o_l, k_s, j_g$	każdy proces jest liniowo uporządkowanym ciągiem kolejnych operacji, a każda operacja może korzystać z kilku jednostek określonego zasobu; spotykane	Elastyczne Systemy Produkcyjne ESP (Banaszak 1991, Fanti et al. 1997)
$WPP o_l, k_m, j_u$	każdy proces jest liniowo uporządkowanym ciągiem kolejnych operacji, a każda operacja może korzystać z wielu różnych zasobów, ale tylko z jednej jednostki każdego zasobu	ESP w których operacje wymagają kilku różnych zasobów (np. dwóch różnych robotów) (Wójcik et al. 1992; Xing et al. 2005)
$WPP o_l, k_m, j_g$	każdy proces jest liniowo uporządkowanym ciągiem kolejnych operacji, a każda operacja może korzystać z wielu różnych zasobów i dodatkowo z kilku jednostek określonych zasobów	ESP w których operacje mogą korzystać z kilku jednostek należących do różnych typów zasobowych (np. dwóch różnych robotów oraz kilku buforów maszyny) (Fanti 2004; Hsieh 2009; Park i Reveliotis 2001)
$WPP o_c, k_s, j_u$	każdy proces jest częściowo uporządkowanym ciągiem operacji, a każda operacja korzysta z jednej jednostki określonego zasobu	operacje montażu i demontażu elementów występujące w ESP (Roszkowska 1990, 2004)
$WPP o_c, k_s, j_g$	każdy proces jest częściowo uporządkowanym ciągiem operacji, a każda operacja może korzystać z kilku jednostek określonego zasobu	ESP, w których występują operacje montażu i demontażu elementów, a do wykonania danej operacji mogą być potrzebne, co najmniej dwa bufory maszyny lub dwa roboty (Roszkowska 1990)
$WPP o_c, k_m, j_u$	każdy proces jest częściowo uporządkowanym ciągiem operacji, a każda operacja może korzystać z wielu różnych zasobów, ale tylko z jednej jednostki każdego zasobu	ESP, w których występują operacje montażu i demontażu elementów z udziałem kilku różnych zasobów (np. dwa roboty oraz bufor maszyny) (Abdallah i Elmaraghy 1998; Ezpeleta i Recalde 2004; Park i Reveliotis 2001, 2002; Reveliotis i Lawley 1997)
$WPP o_c, k_m, j_g$	każdy proces jest częściowo uporządkowanym ciągiem operacji, a każda operacja może korzystać z wielu różnych zasobów i dodatkowo z kilku jednostek określonych zasobów	ESP, w których występują operacje montażu i demontażu elementów z udziałem kilku jednostek, należących do różnych typów zasobowych (np. dwóch różnych robotów oraz kilku buforów maszyny) (Ezpeleta i Recalde 2004; Park i Reveliotis 2001; Reveliotis i Lawley 1997)



Opracowane rozwiązania wykorzystują różne modele systemów procesów, m.in. implementujące modele formalne:

- algebry liniowej; wykorzystujące zasadę liniowej numeracji zasobów (Lawley et al. 1998); konstrukcję równań i nierówności algebraicznych z wykorzystaniem zmiennych stanu, (Reveliotis i Ferreira 1996; Lawley et al. 1997, 1998); algebrę  $(\max, +)$  (Polak et al. 2003; Polak et al. 2004; Majdzik i Banaszak 2006);
- grafów skierowanych; (Coffman i Elphick 1971; Merlin i Schweitzer 1980; Wysk et al. 1994; Błażewicz et al. 1994; Fanti et al. 1996a, 1997, 2000; Roszkowska 2000; Gebraeel i Lawley 2001; Silberschatz et al. 2004);
- automatów i sieci Petriego; wykorzystujące, konstrukcję i analizę grafu stanów automatu (Yalcin i Boucher 2000; Fanti i Zhou 2004) i sieci Petriego (Narahari i Viswanadham 1985; Banaszak 1991; Seiche et al. 1991; Viswanadham i Narahari 1992; Jeng i DiCesare 1993; Xing et al. 1996; Fanti 2004; Wang et al. 2004; Zhou i Fanti 2004); analiza nieznakowanych syfonów (Ezpeleta et al. 1995; Chu i Xie 1997; Abdallah i Elmaraghy 1998; Xie i Jeng 1999; Gürel et al. 2000; Park i Reveliotis 2000, 2001; Huang et al. 2001; Lawley i Sulistyono 2002; Tricas i Ezpeleta 2003);
- modelowania deklaratywnego; wykorzystujące techniki programowania z ograniczeniami (ang. constraint programming CP) (Wójcik et al. 2005; Bocewicz et al. 2007, 2008, 2012c; ).

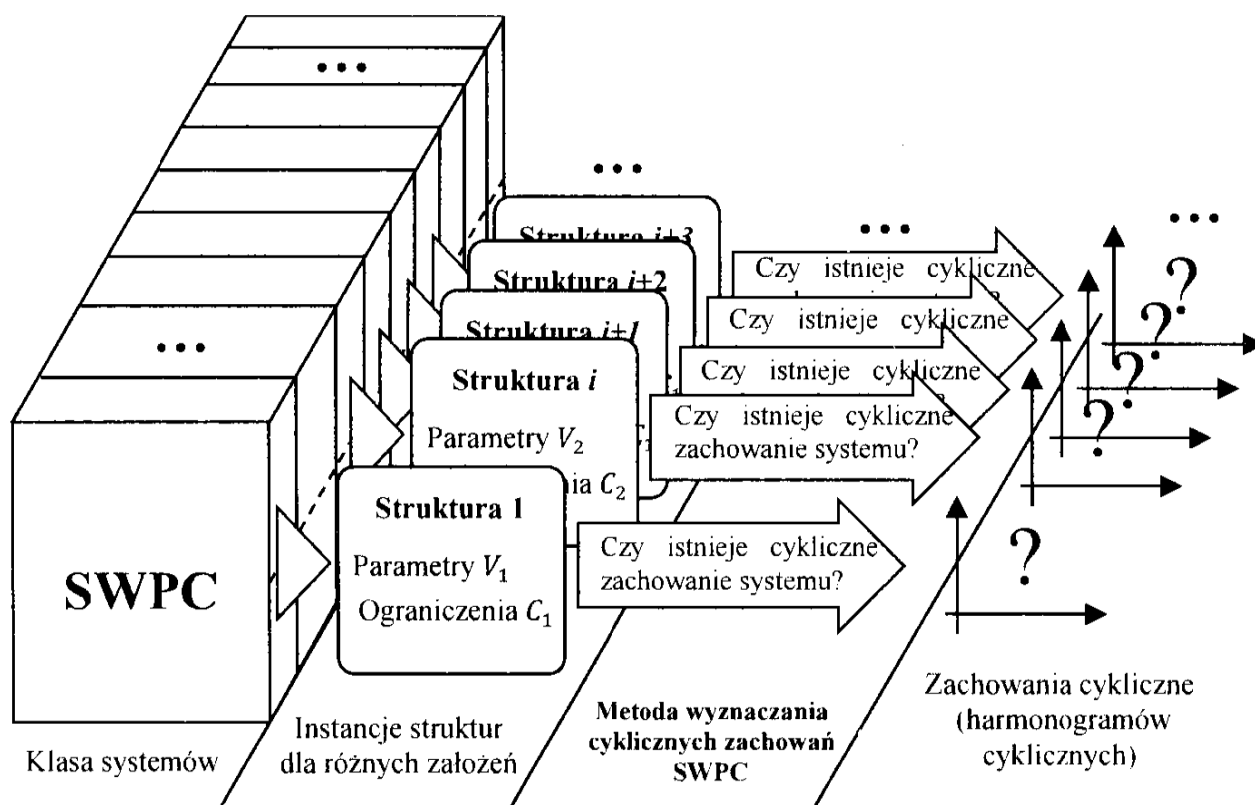
Przedstawiona analiza ograniczona została do SWPC scharakteryzowanych przez parametry  $or, k, j$ , określające wyłącznie podstawowe podklasy WPP. Bardzo często brane są pod uwagę dodatkowe czynniki takie jak: możliwość wykorzystywania alternatywnych zasobów jak i/lub alternatywnych marszrut procesów, zmienna pojemność zasobów w czasie, czy też nieodnawialność zasobów (raz wykorzystana jednostka zasobu nie może być użyta ponownie), itp.

Należy podkreślić, że przedstawiony spis źródeł odnosi się tylko do problemów występujących w scentralizowanych systemach typu WPP. W ogólności liczba potencjalnych wariantów problemów jak i dedykowanych dla nich modeli i metod jest jeszcze większa (klasa PR).

Tak duża licznosc, spotykanych w literaturze, rozwiązań prowadzi do wniosku, że podobnie jak w przypadku równań diofantycznych nie należy się spodziewać istnienia tak jednego, ogólnego modelu, jak i jednej uniwersalnej metody pozwalającej na ocenę cyklicznego zachowania SWPC. Kontynuując analogię równań diofantycznych, w których istnienie rozwiązań determinowane jest przez rodzaj/strukturę równania, istnienie zachowania cyklicznego jest zależne od dziedziny problemu w tym przypadku struktury SWPC (spełniającej określony zbiór założeń/ograniczeń). Oznacza to, że zastosowanie metod i związanych z nimi warunków gwarantujących istnienie zachowań cyklicznych, ogranicza się wyłącznie do klas systemów, dla których zostały one opracowane. Wynika stąd, że wszelkie próby opracowania podejść do wyznaczania harmonogramów cyklicznych lub warunków je gwarantujących, mogą być podejmowane wyłącznie w kontekście określonego zbioru założeń struktury SWPC.

Wspomniany wcześniej podział problemów na problemy analizy (rozumiane jako wyznaczanie cyklicznego zachowania systemu) i syntezy (rozumiane jako wyznaczanie warunków gwarantujących istnienie cyklicznego zachowania) leży u podstaw metodyki modelowania deklaratywnego rys. 1.23 i rys. 1.24. W metodyce tej wyróżnia się dwa podejścia.

W pierwszym podejściu (problem analizy – rys. 1.23), główny akcent kładziony jest na poszukiwaniu metod wyznaczania zachowań cyklicznych SWPC. Innymi słowy dla zadanej struktury systemu wybranej klasy SWPC, charakteryzowanej przez zbiór parametrów  $V_i$  (zbiór zasobów  $R$ , procesów  $P$ , itp.) i ograniczeń  $C_i$  (wzajemne wykluczanie, procesy liniowo uporządkowane, itp.) poszukiwane jest zachowanie cykliczne systemu (np. reprezentowane przez cykliczny harmonogram  $X(1)$ ).



Rys. 1.23. Analiza SWPC

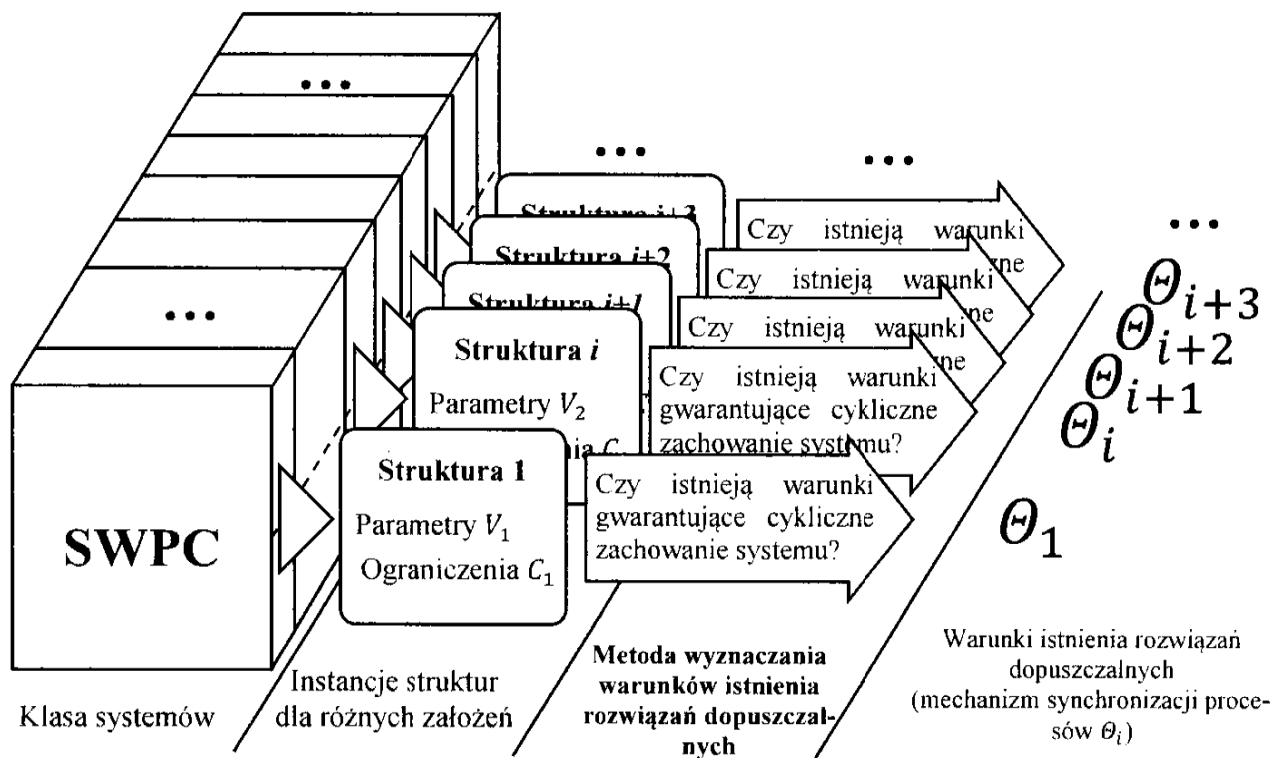
W drugim podejściu (problem syntezy – rys. 1.24) poszukiwana jest metoda wyznaczania warunków, spełnienie których gwarantuje istnienie harmonogramów cyklicznych dla danego SWPC. Warunkami tego typu mogą być właściwości parametrów  $V$  charakteryzujących system (np. ograniczenia określające postać marszrut procesów) jak i mechanizm synchronizacji  $\theta$ , określający sposób dostępu procesów do zasobów współdzielonych.

Wracając do analogii równań diofantycznych podejście stosowane w problemach analizy należy utożsamiać z próbą wyznaczenia metody rozwiązania równań diofantycznych charakteryzujących SWPC (tzn. próbą wyznaczenia harmonogramu cyklicznego), z kolei podejście stosowane w problemach syntezy można utożsamiać z próbą poszukiwania metody wyznaczania warunków gwarantujących istnienie rozwiązań dopuszczalnych (warunków, spełnienie których gwarantuje istnienie harmonogramu cyklicznego). Należy zaznaczyć, że zarówno w przypadku problemów ana-

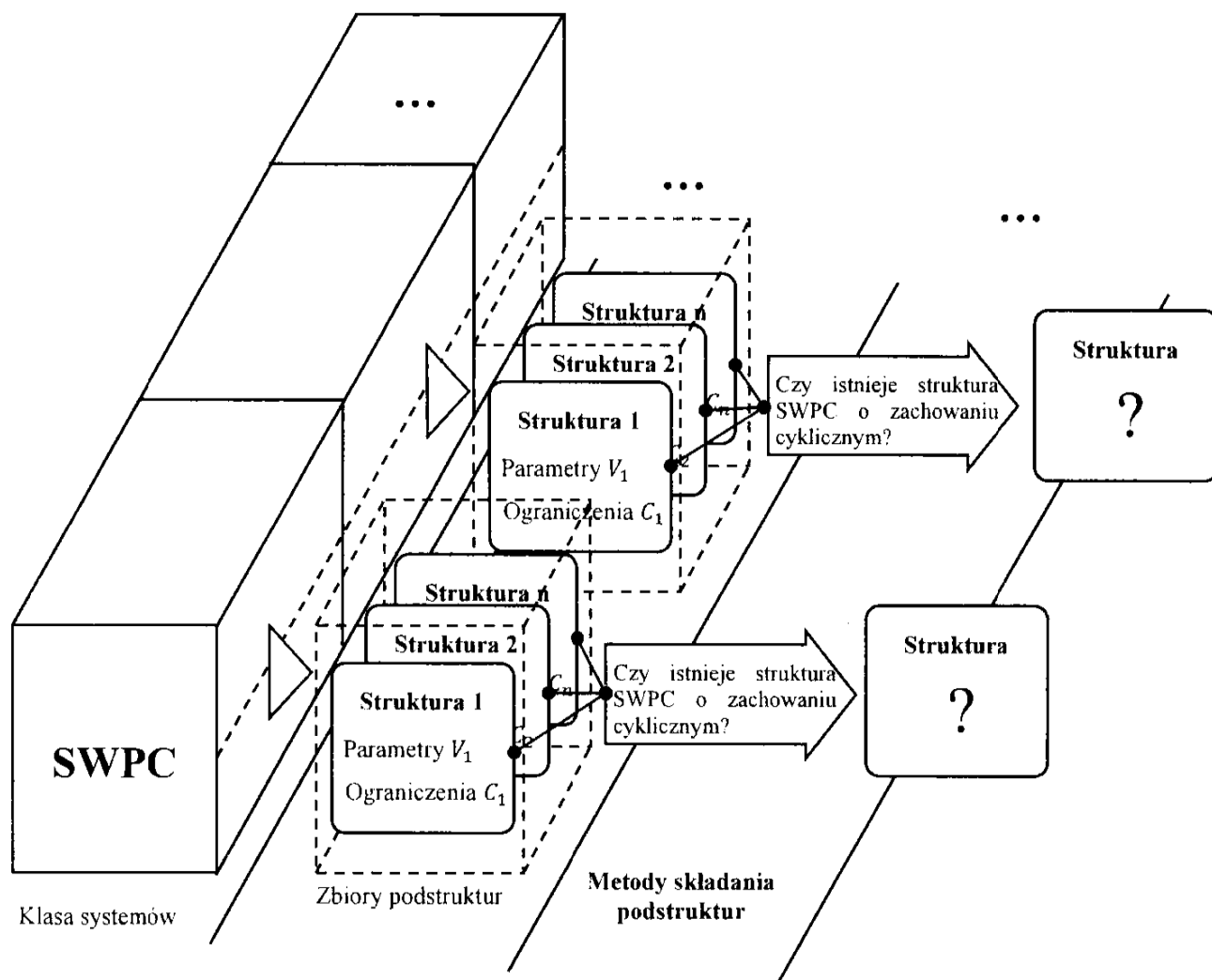
liza jak i syntezy literatura przedmiotu nie dostarcza zbyt wielu efektywnych obliczeniowo metod (większość z nich należy do klasy problemów NP-trudnych).

W praktyce ogranicza to badania SWPC do struktur o stosunkowo niewielkich rozmiarach (zawierających nie więcej niż kilkanaście procesów (Wójcik 2007; Levner et al. 2010, Polak et. al 2004; Bocewicz i Banaszak 2013a, 2013b). Celem zwiększenia skali rozwiązywalnych przypadków zaproponowana została koncepcja składania (kompozycji) SWPC z innych, np. systemów o znanych zachowaniach cyklicznych (Bocewicz et al. 2012a).

Koncepcja składania (konstrukcji) SWPC dla oceny cyklicznego zachowania SWPC sprowadza się do łączenia wcześniej wyodrębnionych (np. w procesie dekompozycji) podstruktur (rys. 1.25). W podejściu tym poszukiwana jest metoda składania podstruktur, gwarantująca istnienie zachowania cyklicznego struktury wynikowej (struktury powstałej w wyniku składania podstruktur). W przedstawionym ujęciu problem składania struktury SWPC stanowi szczególny przypadek problemu syntezy, w którym zasada składania pełni rolę warunków gwarantujących cykliczne zachowanie SWPC.



Rys. 1.24. Synteza SWPC



Rys. 1.25. Składanie SWPC

**Przykład 3.** Ilustracja wykorzystania koncepcji składania podstruktur, do oceny istnienia zachowań cyklicznych SWPC.

Dany jest system jak na rys. 1.22a, o strukturze opisanej przez:

- zbiór zasobów:  $R = \{R_1, R_2, R_3, R_4, R_5, R_6\}$ , gdzie

pojemności zasobów mają postać:

$$C(R_1) = C(R_2) = \dots = C(R_6) = 1,$$

- zbiór procesów:  $P = \{P_1, P_2, P_3\}$ , gdzie

sekwencje operacji procesów mają postać:

$$O_1 = (o_{1,1}, o_{1,2}, o_{1,3}), O_2 = (o_{2,1}, o_{2,2}, o_{2,3}), O_3 = (o_{3,1}, o_{3,2}, o_{3,3})$$

marszruty procesów mają postać:

$$p_1 = (R_4, R_3, R_5), p_2 = (R_3, R_1, R_2), p_3 = (R_6, R_5, R_1)$$

zapotrzebowania operacji na zasoby:

$$w(R_4, o_{1,1}) = w(R_3, o_{1,2}) = w(R_5, o_{1,3}) = 1,$$

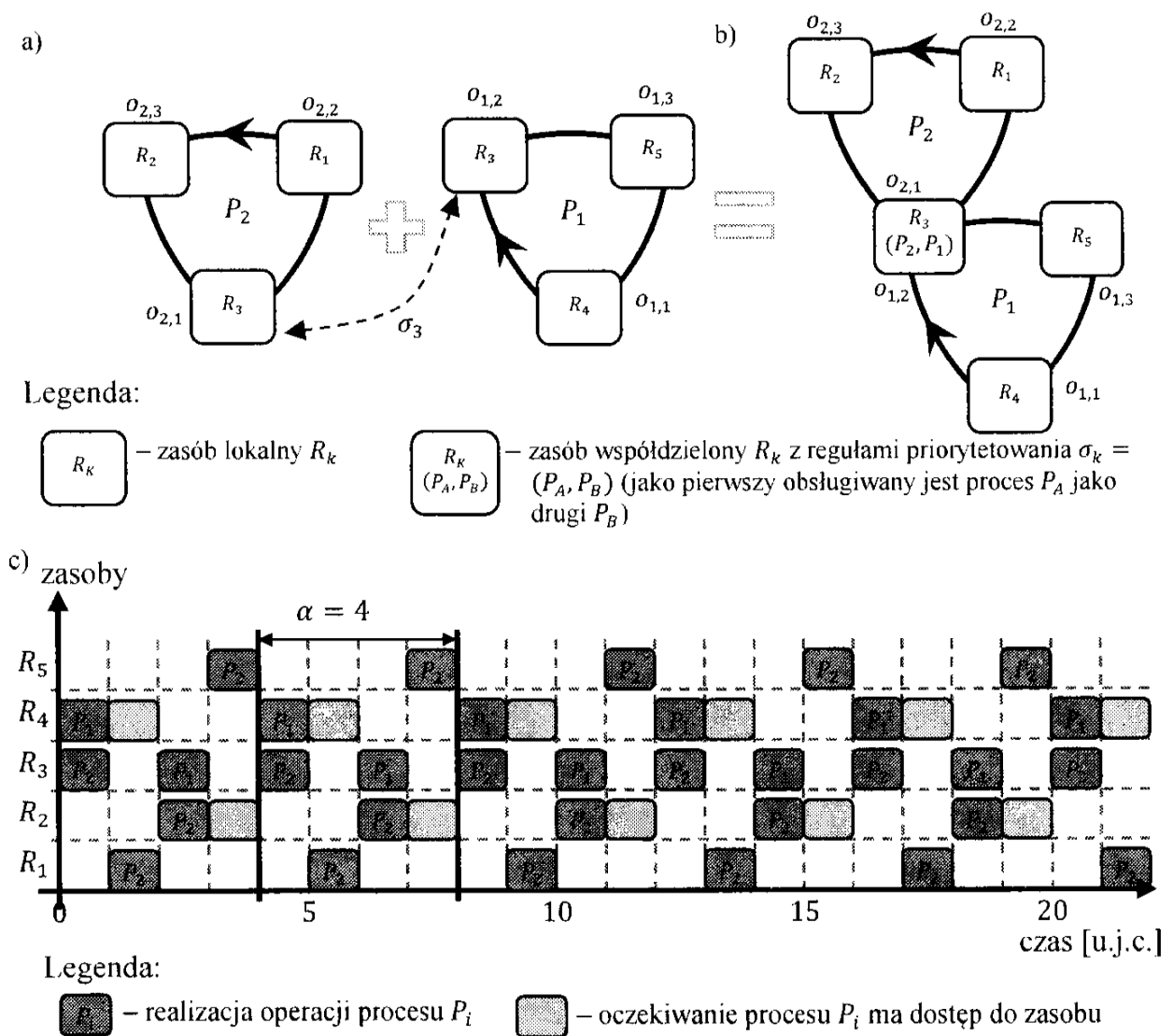
$$w(R_3, o_{2,1}) = w(R_1, o_{2,2}) = w(R_2, o_{2,3}) = 1,$$

$$w(R_6, o_{3,1}) = w(R_5, o_{3,2}) = w(R_1, o_{3,3}) = 1.$$

Ocena zachowania cyklicznego tego typu systemu może odbywać się na wiele sposobów (np. poprzez rozwiązanie układu równań z rys. 1.22). Podejście oparte na koncepcji składania SWPC zakłada dekompozycję jego struktury.

W tym przypadku struktura podzielona została na 3 podstruktury, każda zawierająca jeden proces ( $P_1, P_2, P_3$ ). W rozważanym systemie osiągalne będzie zachowanie cykliczne jeżeli uda się złożyć otrzymane podstruktury w taki sposób by żaden z etapów składania nie doprowadził do blokady.

Struktura SWPC generowana jest w podejściu przyrostowym. W pierwszym etapie (rys. 1.26) składane są podstruktury związane z procesami  $P_1, P_2$ . Miejscem łączenia jest zasób  $R_3$ . W ramach pierwszego etapu składania wyznaczany jest mechanizm synchronizacji procesów w dostępie do zadanego, łączącego składane podstruktury, zasobu.

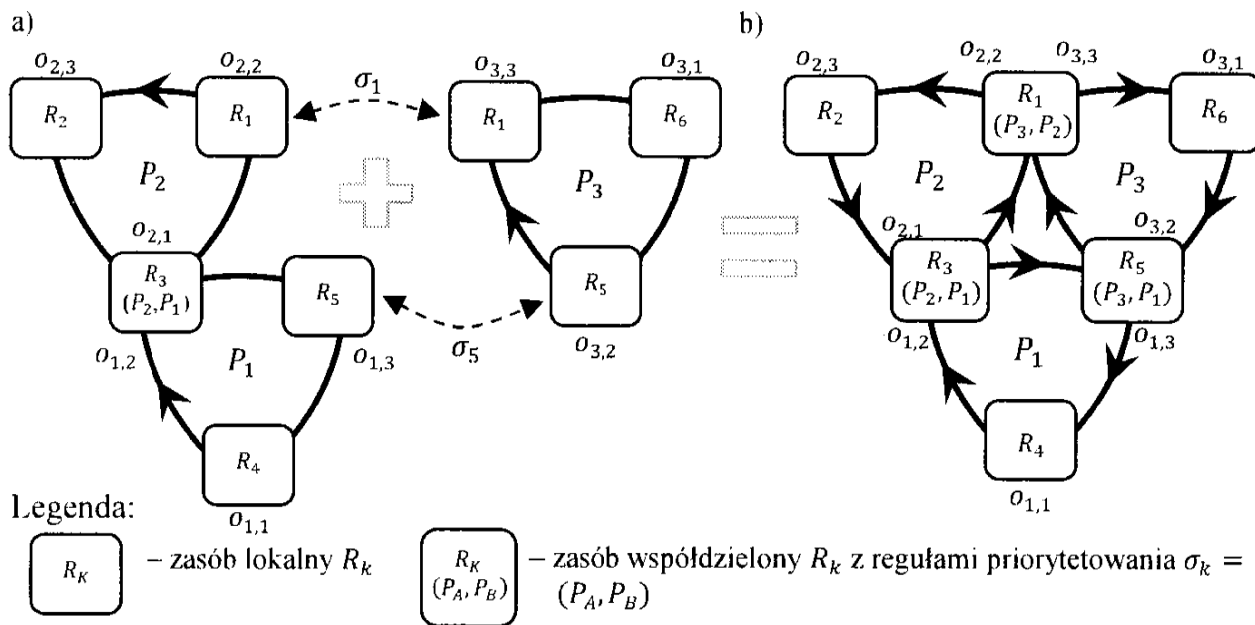


**Rys. 1.26.** Składanie SWPC, elementy składowe a), system wynikowy b) harmonogram realizacji operacji systemu wynikowego c)

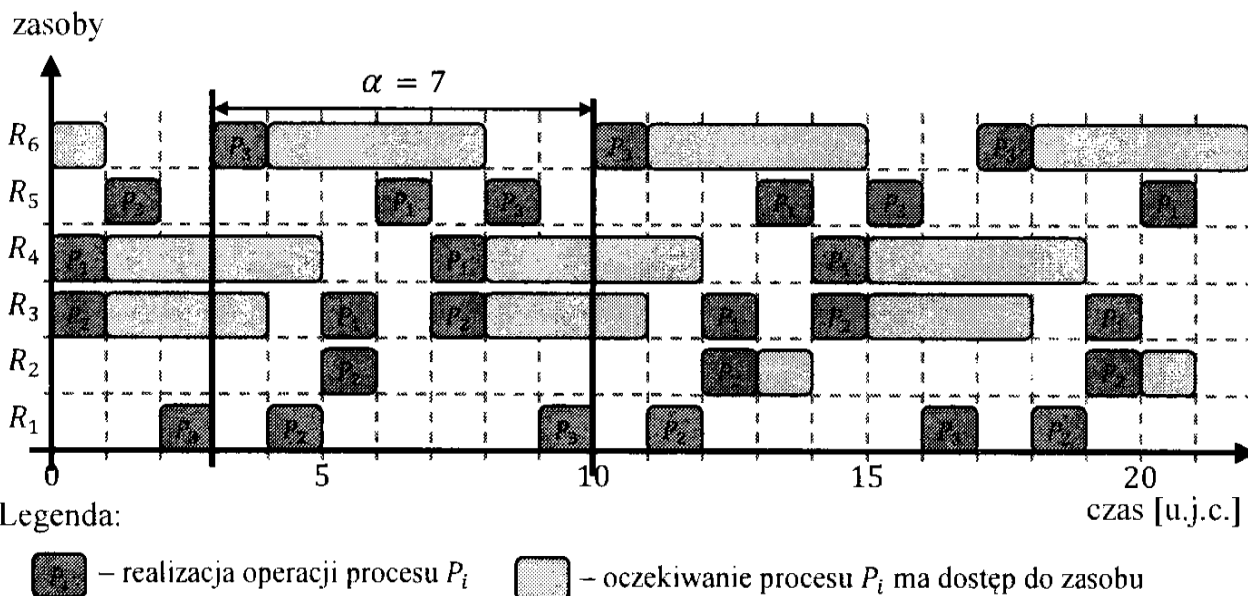
Przyjęto, że stosowane są statyczne reguły priorytetowania  $\sigma_i$  reprezentowane w postaci sekwencji określającej porządek dostępu do zasobów. W rozważanym przypadku reguła dla zasobu  $R_3$  ma postać  $\sigma_3 = (P_1, P_2)$  co oznacza, że w pierwszej kolejności dostęp do zasobu otrzymuje proces  $P_1$  a następnie  $P_2$ . Otrzymany w ten

sposób system charakteryzuje się zachowaniem cyklicznym (harmonogram przedstawiony jest na rys. 1.26c).

W drugim etapie dołączana jest kolejna podstruktura związana z procesem  $P_3$  (rys. 1.27). Tym razem miejscami łączenia są zasoby  $R_1$  i  $R_5$ . Otrzymany system jest cykliczny dla reguł  $\sigma_1 = (P_3, P_2)$ ,  $\sigma_5 = (P_3, P_1)$  (odpowiednio dla zasobów  $R_1$  i  $R_5$ ). Harmonogram cykliczny dla takiego złożenia został przedstawiony na rys. 1.28. W SWPC osiągalne jest zachowanie cykliczne z okresem  $\alpha = 7$ .



Rys. 1.27. Składanie SWPC, elementy składowe a), system wynikowy b)



Rys. 1.28. Harmonogram realizacji operacji systemu wynikowego przedstawionego na rys. 1.27b

Przedstawiony przykład pokazuje, że ocenę istnienia zachowania cyklicznego SWPC można w szczególnym przypadku sprowadzić do wieloetapowego poszukiwania reguł priorytetowania/synchronizacji procesów (w miejscach występowania zasob-

bów łączących składane systemy). W ogólności problemy tego typu sprowadzają się do określania odpowiedniego sposobu składania podstruktur SWPC. Związane z tym pytanie: w oparciu o jakie zasady dobierać mechanizm synchronizacji? i tym samym w oparciu o jakie zasady można łączyć struktury elementarne? prowadzi do następujących problemów:

*Dane są, zbiór struktur SE (o znanych parametrach  $V$ , ograniczeniach  $C$  i zachowaniach cyklicznych reprezentowanych systemów – harmonogramach  $X$ ) oraz zasady ich składania. Poszukiwana jest taka kompozycja struktur zbioru SE, w której osiągalne jest zachowanie cykliczne (cykliczny harmonogram  $X$ ).*

lub

*Dany jest, zbiór struktur SE (o znanych parametrach  $V$ , ograniczeniach  $C$  i zachowaniach cyklicznych reprezentowanych systemów – harmonogramach  $X$ ). Poszukiwane są zasady składania (zasady doboru reguły priorytetowania), gwarantujące złożenie struktur zbioru SE w strukturę, w której osiągalne jest zachowanie cykliczne (cykliczny harmonogram  $X$ ).*

Przez zasady składania rozumie się w tym przypadku zbiór reguł, który gwarantuje cykliczne zachowanie SWPC powstałego w wyniku złożenia kilku struktur zbioru SE. Przykłady tego typu zasad przedstawione zostaną w rozdziale 4.

### 1.3. Problemy harmonogramowania cyklicznego w systemach współbieżnych procesów multimodalnych

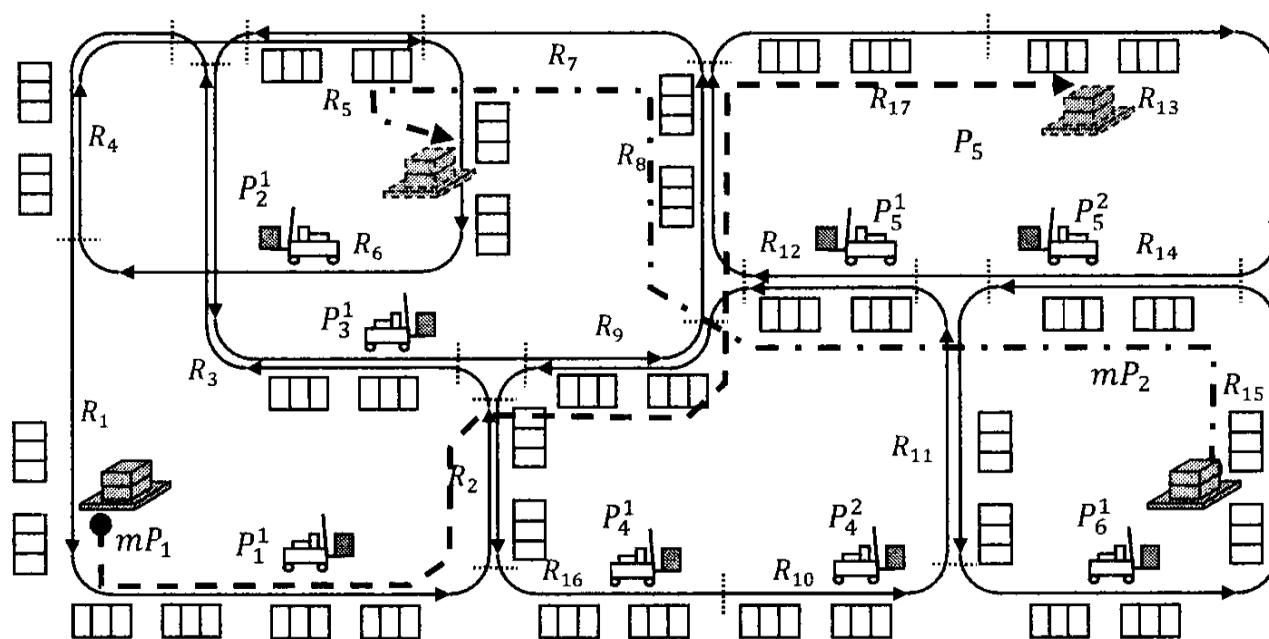
Sytuacja, w której procesy są traktowane jako procesy multimodalne występuje wówczas gdy operacje wymagają do swojej realizacji, oprócz zasobów  $R$ , również innych procesów. Przykładem może być tutaj podróż pasażera różnymi liniami metra. Każda z linii może być reprezentowana przez procesy cykliczne realizujące operacje w postaci przemieszczania się i postoju pociągów na stacjach. Zasobami w tym przypadku są stacje oraz odcinki linii metra. Pasażer może być z kolei reprezentowany przez proces, którego operacje związane są z podróżą oraz operacjami wsiadania/wysiadania. Aby jednak podróż pasażera była możliwa niezbędny jest pociąg, który traktowany jest jako przemieszczający się zasób. „Proces pasażera” wymaga więc do realizacji operacji obecności „procesu pociągu”. Multimodalność w tym przypadku oznacza, że pasażer w trakcie swojej podróży może wielokrotnie przesiadać się między pociągami tzn. wielokrotnie zmieniać proces lokalny wymagany do realizacji swojego planu podróży.

**Procesy multimodalne** to takie, które wymagają innych procesów do swojej realizacji. W celu rozróżnienia, **procesy lokalne** są określane jako procesy niewymagające innych procesów do swojej realizacji (czyli procesy omawiane do tej pory).

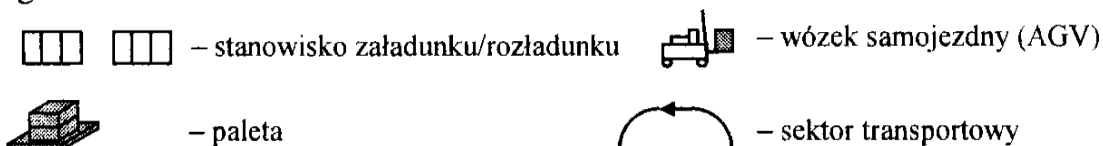
Multimodalność jest pojęciem występującym w logistyce związanym z tzw. transportem multimodalnym obejmującym zarówno transport towarów i/lub ludzi przy użyciu różnych środków transportu (Foo et al. 1999; Friedrich et al. 1999; Horn 2002; Bielli et al. 2006; Guo 2008).

Oprócz logistycznego (transportowego) obszaru występowania procesów multimodalnych można wyróżnić obszary związane z systemami produkcyjnymi, gdzie procesy multimodalne realizowane są wzdłuż marszrut produkcyjnych produkowanych wyrobów (Bocewicz et al. 2011a, 2011b, 2014; Bocewicz i Banaszak 2011, 2013a, 2013b).

W proponowanym, multimodalnym podejściu, wyróżnienie pewnych elementów SWPC (takich jak marszrutu produkcyjne, marszrutu transportowe, strumienie danych itp.) jako odrębnej grupy procesów multimodalnych pozwala na prowadzenie bardziej szczegółowej analizy niż ma to miejsce w przypadku podejść uwzględniających wyłącznie procesy lokalne. Przykładowo traktowanie marszrut produkcyjnych jako procesów multimodalnych zależnych od środków transportowych (reprezentowanych przez procesy lokalne) pozwala m.in. na ocenę i analizę związków między cyklicznym zachowaniem tych procesów (np. związek między harmonogramem cyklicznym środków transportowych a cykliczną realizacją produkcji). W tym ujęciu obszar zastosowań obejmuje analizę i syntezę systemów zawierających procesy wzajemnie się wykorzystujące (realizacja jednych jest warunkowana obecnością innych procesów), tj. systemy transportowe, zrobotyzowane systemy produkcyjne, systemy transmisji danych, itp.



Legenda:

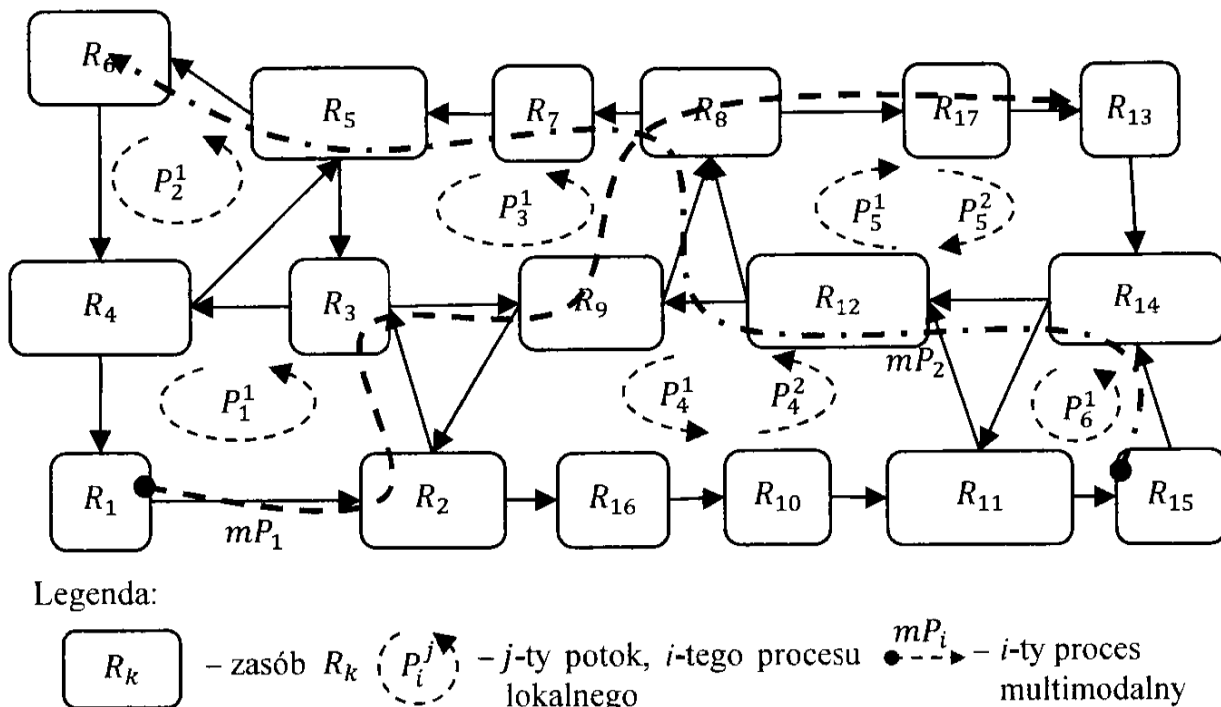


Rys. 1.29. Przykład systemu transportowego ESP (Bocewicz i Banaszak 2013b)

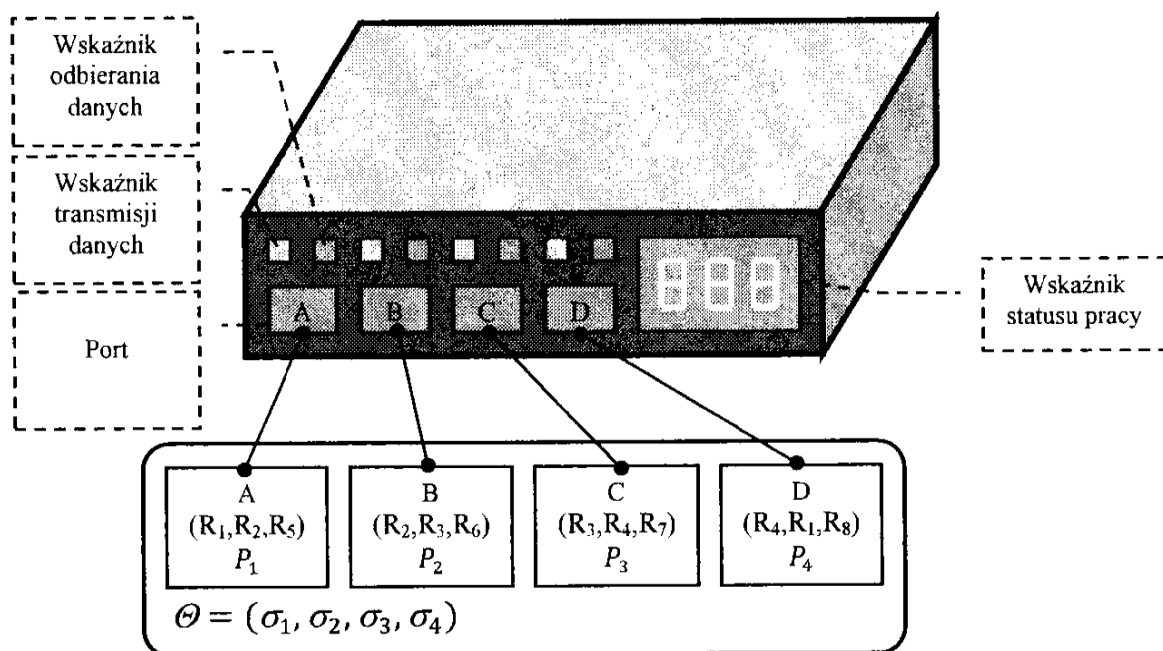
Przykładem systemu z procesami multimodalnymi może być cykliczny system wózków samojezdnych AGV (ang. Automated Guided Vehicle) przedstawiony na rys. 1.29. W systemie tym pomiędzy stanowiskami przemieszcza się (wzdłuż cyklicznych trajektorii) 8 wózków. Wózki są wykorzystywane do transportu elementów (palet) zgodnie z cyklicznie powtarzanimi trajektoriami. Rys. 1.30 przedstawia reprezen-



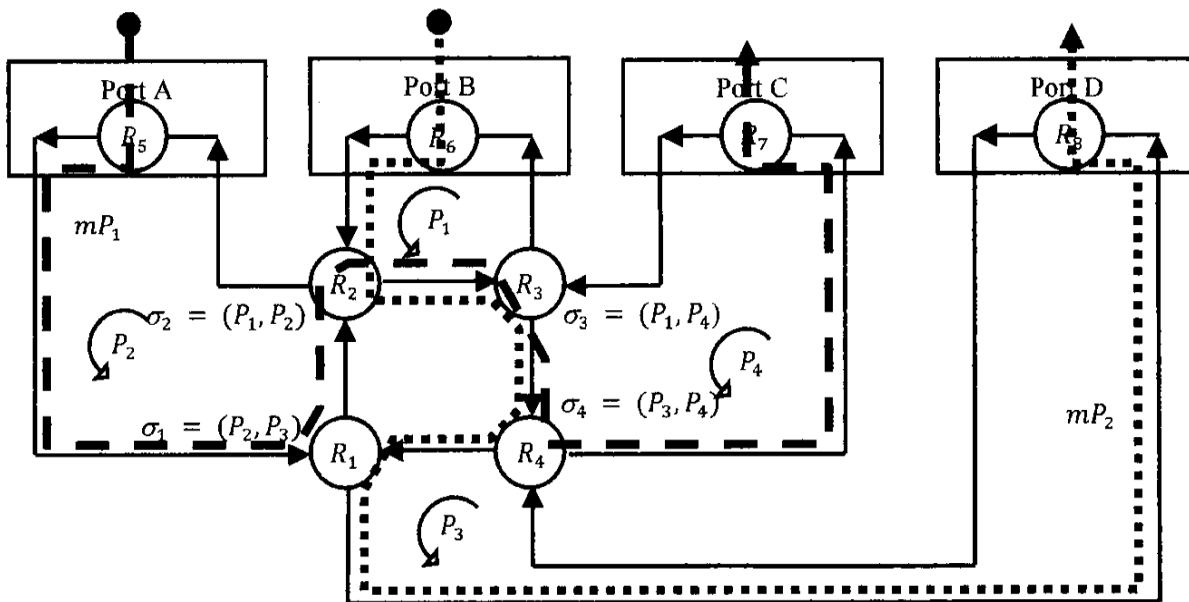
tacje systemu transportowego jako **System Współbieżnych Multimodalnych Procesów Cyklicznych (SWMPC)**. W systemie występuje 8 procesów lokalnych (wózki):  $P_1^1, P_2^1, P_3^1, P_4^1, P_4^2, P_5^1, P_5^2, P_6^1$ , przy czym procesy z tą samą marszrutą są traktowane jako wielokrotna wersja (potok) jednego procesu (indeks górny określa numer potoku), oraz dwa procesy multimodalne  $mP_1, mP_2$ .



Rys. 1.30. SWMPC dla systemu z rys. 1.29 (Bocewicz i Banaszak 2013b)



Rys. 1.31. Przykład czteroportowego routera transmisji danych dla usługi VOD (ang. Video On Demand) wspomagający stosowanie QoS (ang. Quality of Service) (Polak 2004)



Legenda:

$A, B, C, D$  – nazwa portu

$P_i$  – proces lokalny reprezentujący transport pakietów pomiędzy zasobami

$\Theta$  – zbiór reguł alokowany na zasobach dzielonych portów

$mP_i$  – proces multimodalny reprezentujący strumień pakietów pomiędzy portami

**Rys. 1.32.** Reprezentacja routera z rys. 1.31 jako systemu współbieżnych multimodalnych procesów cyklicznych

Przykładem wykorzystania SWMPC w systemach transmisji danych jest z kolei system z rys. 1.32 reprezentujący router usługi VOD (ang. Video On Demand) przedstawiony na rys. 1.31 (Polak 2004). W tego typu systemie występują cztery procesy lokalne:  $P_1, P_2, P_3, P_4$  odpowiedzialne za transport pakietów danych pomiędzy zasobami wejścia/wyjścia portów oraz dwa procesy multimodalne  $mP_1, mP_2$  reprezentujące strumienie pakietów. W rozważanym przypadku pakiety transmitowane są pomiędzy portami  $A - C$  oraz  $B - D$ .

Współbieżne multimodalne procesy cykliczne  $mP_i \in mP$  definiowane są przez (Bocewicz et al. 2013a, 2013b):

- marszrutę procesu  $mP_i$ :

$$mp_i = (mp_{i,1}, \dots, mp_{i,j}, \dots, mp_{i,lm(i)}), mp_{i,j} \in R, \quad (12)$$

wyznaczaną jako połączenie wybranych fragmentów marszrut procesów lokalnych ( $P = \{P_i \mid i = 1, \dots, ln\}$  procesów nie wymagających do swojej realizacji innych procesów):

$$mp_i = \left( mpr_{i_1}(a_{i_1}, b_{i_1}) \wedge mpr_{i_2}(a_{i_2}, b_{i_2}) \wedge \dots \wedge mpr_{i_y}(a_{i_y}, b_{i_y}) \right), \quad (13)$$

gdzie:

$$mpr_i(a, b) = \begin{cases} (p_{i,a}, p_{i,a+1}, \dots, p_{i,b}) & \text{dla } a \leq b \\ (p_{i,a}, p_{i,a+1}, \dots, p_{i,lr(i)}, p_{i,1}, \dots, p_{i,b}) & \text{dla } b > a \end{cases}$$

$$a, b \in \{1, \dots, lr(i)\},$$

jest fragmentem marszruty  $p_i$  zawierającym elementy od  $p_{i,a}$  do  $p_{i,b}$ .

$x \wedge y$  – oznacza konkatencję sekwencji  $x$  i  $y$ , w przypadku gdy  $x = (x_1, \dots, x_n)$  a  $y = (y_1, \dots, y_m)$  to  $x \wedge y = (x_1, \dots, x_n, y_1, \dots, y_m)$ .

Marszruta  $mp_i$  jest sekwencją zawierającą fragmenty marszrut  $mpr_i(a, b)$  procesów lokalnych  $P_i$ , które są wykorzystywane przy realizacji procesu  $mP_i$ . Przykładowo marszruty procesów  $mP_1, mP_2$  z rys. 1.30 mają postać:

$$\begin{aligned} mp_1 &= ((R_1, R_2, R_3) \wedge (R_9, R_8) \wedge (R_{17}, R_{13})) = (R_1, R_2, R_3, R_9, R_8, R_{17}, R_{13}), \\ mp_2 &= ((R_{15}, R_{14}) \wedge (R_{12}, R_8) \wedge (R_7, R_5) \wedge (R_6)) = (R_{15}, R_{14}, R_{12}, R_8, R_7, R_5, R_6). \end{aligned}$$

Zakłada się że dwa występujące po sobie fragmenty marszrut  $mpr_{i_1}(a_{i_1}, b_{i_1}) \wedge mpr_{i_2}(a_{i_2}, b_{i_2})$  są połączone ze sobą przez wspólny zasób  $p_{j, b_{i_1}}$ . Przykładowo fragmenty marszrut  $(R_1, R_2, R_3)$  procesu lokalnego  $P_1$  ( $p_1 = (\overline{R_1, R_2, R_3}, R_4)$ ) oraz  $(R_9, R_8)$  procesu lokalnego  $P_3$  ( $p_3 = (R_3, \overline{R_9, R_8}, R_7, R_5)$ ) wykorzystane w  $mp_1$ , łączą się przez zasób  $R_3$ .

- sekwencję operacji  $mP_i$ :

$$mO_i = (mo_{i,1}, mo_{i,2}, \dots, mo_{i,j}, \dots, mo_{i,lm(i)}), \quad (14)$$

gdzie:  $mo_{i,j}$  –  $j$ -ta operacja procesu  $mP_i$ ,  $lm(i)$  – liczba operacji  $mP_i$ .

Każdej operacji  $mo_{i,j}$  przyporządkowany jest proces lokalny niezbędny do jej wykonania  $\mu(mo_{i,j}) = m\mu_{i,j} \in P$  ( $P$  – zbiór wszystkich procesów lokalnych), przykładowo w ramach procesu  $mP_1$  realizowanych jest 7 operacji wykorzystujących 3 procesy lokalne:  $m\mu_{1,1} = m\mu_{1,2} = m\mu_{1,3} = P_1$ ;  $m\mu_{1,4} = m\mu_{1,5} = P_3$ ;  $m\mu_{1,6} = m\mu_{1,7} = P_5$ . Ponadto operacji  $mo_{i,j}$  przyporządkowuje się czas wykonania  $mt_{i,j} \in \mathbb{N}^+$ . Należy zwrócić uwagę, że zwykle wymaga się by czas ten nie był większy od czasu wykonania (czas wykonywania operacji) procesu lokalnego.

W ogólnym przypadku można wprowadzić kolejny poziom procesów multimodalnych wykorzystujących do swojej realizacji procesy multimodalne niższego poziomu (wielopoziomowość procesów multimodalnych będzie opisana szerzej w kolejnym rozdziale). Dla przykładu, wspomniany już wcześniej pasażer linii metra może przewozić paczkę, którą gdzieś po drodze przekaze innemu pasażerowi. Paczki reprezentowane są w tym przypadku przez procesy multimodalne wykorzystujące inne procesy (procesy pasażerów) jako środki transportu.

Oznacza to, że systemy współbieżnych multimodalnych procesów cyklicznych (SWMPC) stanowią naturalne rozszerzenie systemów SWPC omawianych do tej pory. Mogą one również zastępować niektóre problemy harmonogramowania cyklicznego prezentowane w podrozdziale 1.1.1. Przykładowo problemy harmonogramowania cyklicznego zrobotyzowanego gniazda produkcyjnego: CRFP, CRJP, CROP, CRPSP mogą być traktowane w kategorii SWMPC. W takim przypadku (np. rys. 1.11 i rys. 1.12) roboty mobilne odpowiedzialne za operacje transportowe reprezentowane są przez procesy lokalne natomiast zadania zbioru  $Z$  i związane z nimi operacje  $O_i$ , przez procesy multimodalne.

Analogicznie jak w przypadku procesów lokalnych zachowanie cykliczne uwzględniające realizację operacji procesów multimodalnych  $mP_i$  reprezentowane jest przez cykliczny harmonogram:

$$mX = ((mX_1, \dots, mX_q), m\alpha), \tag{15}$$

gdzie:

$mX_i = (mx_{i,1}, \dots, mx_{i,lm(i)})$  – sekwencja terminów rozpoczęcia operacji procesu  $mP_i$ ,

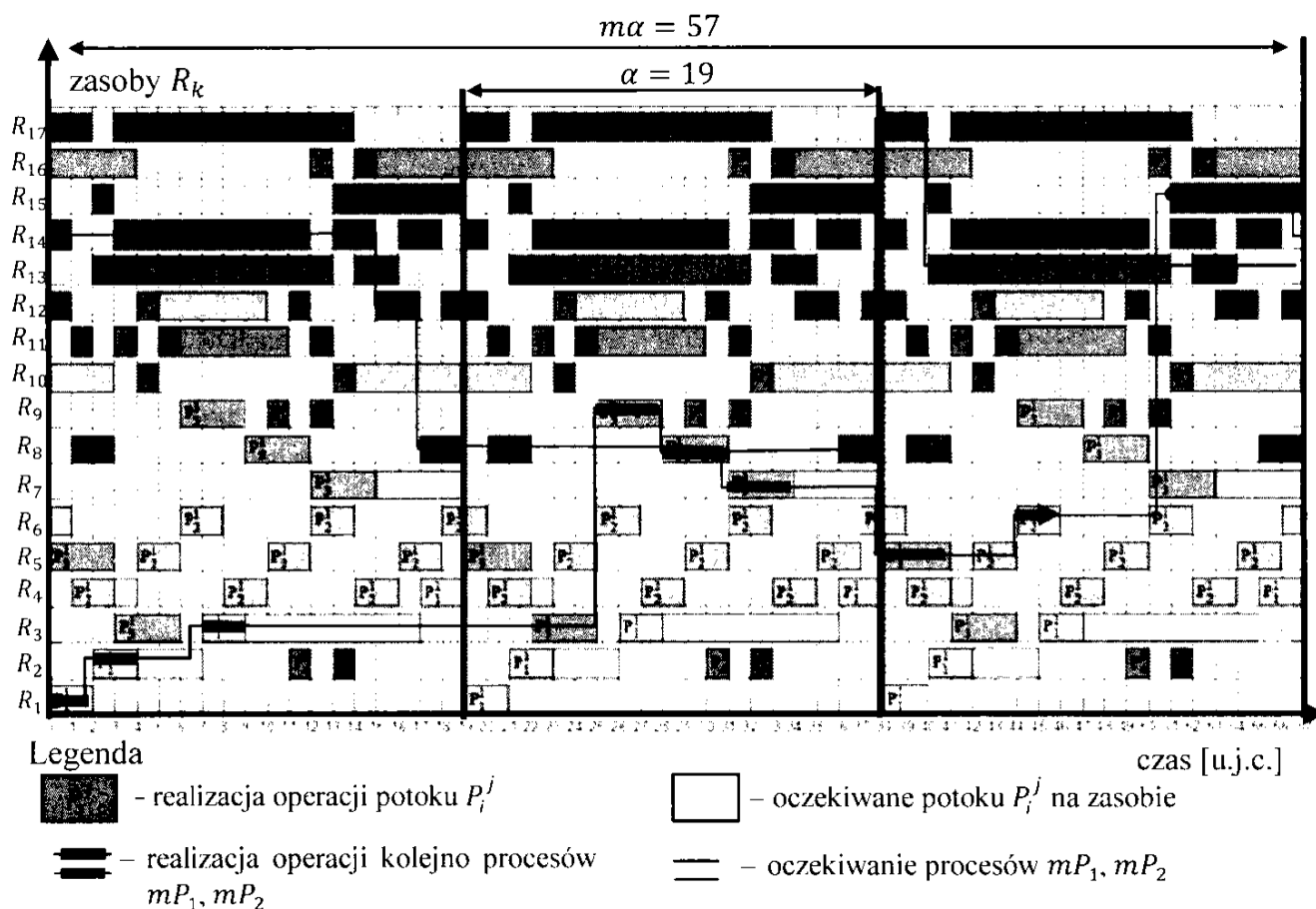
$mx_{i,j} = mx_{i,j}(0)$  – termin rozpoczęcia operacji dla  $k = 0$  wykonania procesu  $mP_i$ ,

$$mx_{i,j}(k) = mx_{i,j}(0) + m\alpha \cdot k, k \in \mathbb{C},$$

$q$  – liczba procesów,

$m\alpha$  – okres harmonogramu  $m\alpha \in \mathbb{N}^+$ .

W SWMPC para  $(X, mX)$  (patrz (11) i (14)) opisuje harmonogram cyklicznej realizacji operacji. Okres procesów lokalnych  $\alpha$  nie musi być równy okresowi procesów multimodalnych  $m\alpha$ . Na rys. 1.33 przedstawiony został przykład takiego harmonogramu dla systemu z rys. 1.30 ( $\alpha = 19$ ,  $m\alpha = 57$ ).



Rys. 1.33. Cykliczny harmonogram realizacji procesów lokalnych i multimodalnych systemu z rys. 1.30

Współbieżna realizacja procesów multimodalnych może odbywać się przy różnych założeniach. Przyjmuje się, że procesy multimodalne wykorzystują procesy lokalne w trybie niewyłączalnym, jednak założenie wzajemnego wykluczania oraz prze-

trzymywanie procesu lokalnego do momentu rezerwacji nowego procesu jest już opcjonalne. Możliwość wyboru różnych założeń sprawia, że systemy z procesami multimodalnymi mogą być wykorzystywane do reprezentacji wielu rzeczywistych systemów (transportowych, produkcyjnych, transmisji danych itp.)

Przykładowo przedstawiony na rys. 1.33 harmonogram dopuszcza jednocześnie przebywanie procesów multimodalnych na wspólnym zasobie  $R_8$  (różne palety mogą przebywać jednocześnie na jednym stanowisku) jak i jednocześnie wykorzystanie wspólnego procesu lokalnego (palety przewożone są przez jeden wózek).

Oprócz wyżej przedstawionych można rozważać również cały szereg założeń dodatkowych dopuszczających alternatywne marszruty procesów multimodalnych, nieprecyzyjne (niepewne) czasy wykonania operacji, opóźnienia związane z przesiadkami lub przebrojeniem maszyn, itp.

Analogicznie jak w przypadku SWPC, w klasie SWMPC wyróżnia się problemy analizy, syntezy oraz składania. Problemy te definiowane są następująco:

### Problem analizy

*W SWMPC zawierającym:*

- *zbiór zasobów  $R$  o znanych pojemnościach,*
- *procesy lokalne  $P$  opisane przez zbiór parametrów  $V_l$ :  
marszruty  $p_i$  i sekwencje operacji  $O_i$ ,*
- *procesy multimodalne  $mP$  opisane przez zbiór parametrów  $V_m$ :  
marszruty  $mp_i$  i sekwencje operacji  $mO_i$ ,*

*oraz zbiór założeń  $C$  (ograniczeń typu: niewywłaszczalność, wzajemne wykluczanie, zbiór reguł priorytetowania  $\Theta$  procesów  $P$  i  $mP$ , itp.),*

*poszukiwany jest cykliczny harmonogram  $(X, mX)$  realizacji procesów lokalnych i multimodalnych.*

### Problem syntezy

*W SWMPC zawierającym:*

- *zbiór zasobów  $R$  o znanych pojemnościach,*
- *procesy lokalne  $P$  opisane przez zbiór parametrów  $V_l$ :  
marszruty  $p_i$  i sekwencje operacji  $O_i$ ,*
- *procesy multimodalne  $mP$  opisane przez zbiór parametrów  $V_m$ :  
marszruty  $mp_i$  i sekwencje operacji  $mO_i$ ,*

*oraz zbiór założeń  $C$  (ograniczeń typu: niewywłaszczalność, wzajemne wykluczanie, itp.),*

*poszukiwane są warunki (np. reguły priorytetowania  $\Theta$  procesów  $P$  i  $mP$ ) gwarantujące istnienie cyklicznego harmonogram  $(X, mX)$  realizacji lokalnych i multimodalnych?*

### Problem składania

*Dany jest zbiór struktur  $SE$  SWMPC charakteryzowanego przez:*

- *zbiór zasobów  $R$  o znanych pojemnościach,*
- *procesy lokalne  $P$  opisane przez zbiór parametrów  $V_l$ :  
marszruty  $p_i$  i sekwencje operacji  $O_i$ ,*
- *procesy multimodalne  $mP$  opisane przez zbiór parametrów  $V_m$ :  
marszruty  $mp_i$  i sekwencje operacji  $mO_i$ ,*

*poszukiwane są zasady składania (zasady doboru reguły priorytetowania  $\Theta$  procesów  $P$  i  $mP$ ), gwarantujące złożenie struktur zbioru  $SE$  w strukturę systemu, w którym osiągalne jest zachowanie cykliczne procesów lokalnych i multimodalnych (cykliczny harmonogram  $(X, mX)$ ).*

Tak rozumiane problemy analizy, syntezy oraz składania będą rozwiązywane w kolejnych rozdziałach na przykładzie przyjętej klasy SWMPC.

#### 1.4. Podsumowanie

Rozważanie problemów harmonogramowania cyklicznego w dziedzinie zmiennych dyskretnych (całkowitoliczbowych) sprawia, że modele opisujące te problemy mogą być postrzegane w kategorii układów równań diofantycznych. Oznacza to, że problem harmonogramowania cyklicznego sprowadza się do problemu rozwiązywalności odpowiednich równań diofantycznych tzn. do oceny istnienia ich rozwiązań.

Brak ogólnych metod pozwalających na rozwiązywanie tego typu problemów zmusza do stosowania zindywidualizowanego, dedykowanego dla określonych zbiorów założeń podejścia. Przykładowo założenia systemów współbieżnych procesów cyklicznych, typu: wzajemne wykluczanie, wymagają stosowania podejść uwzględniających możliwość występowania zjawiska blokad i zagłódzeń.

Zjawisko blokad może prowadzić do sytuacji, w których zbiór rozwiązań dopuszczalnych jest zbiorem pustym. Brak gwarancji istnienia rozwiązań dopuszczalnych sprawia z kolei, że optymalizacyjne problemy harmonogramowania cyklicznego zastępowane są problemami decyzyjnymi, których celem są metody wyznaczania rozwiązań dopuszczalnych lub też warunków gwarantujących ich istnienie.

Przykładem tego typu problemów są problemy harmonogramowania cyklicznego w Systemach Współbieżnych Multimodalnych Procesów Cyklicznych SWMPC, w których wyróżnić można, poza procesami lokalnymi, wiele poziomów procesów multimodalnych (transport pasażerów przez pociągi, transport przesyłek przez pasażerów itp.). Dla każdego z tak określonych poziomów można przyjmować odrębny zbiór założeń i ograniczeń.

Mnogość wariantów SWMPC, oraz diofantyczny charakter rozważanych problemów uzasadnia przyjętą we wstępie (rys. W.6) koncepcję wspomaganie decyzji. Pytania odnoszące się do istnienia w danym systemie zachowania cyklicznego, jak i warunków gwarantujących takie zachowanie (rys. W.6), odpowiadają prezentowanemu w rozdziale problemom analizy, syntezy i składania.

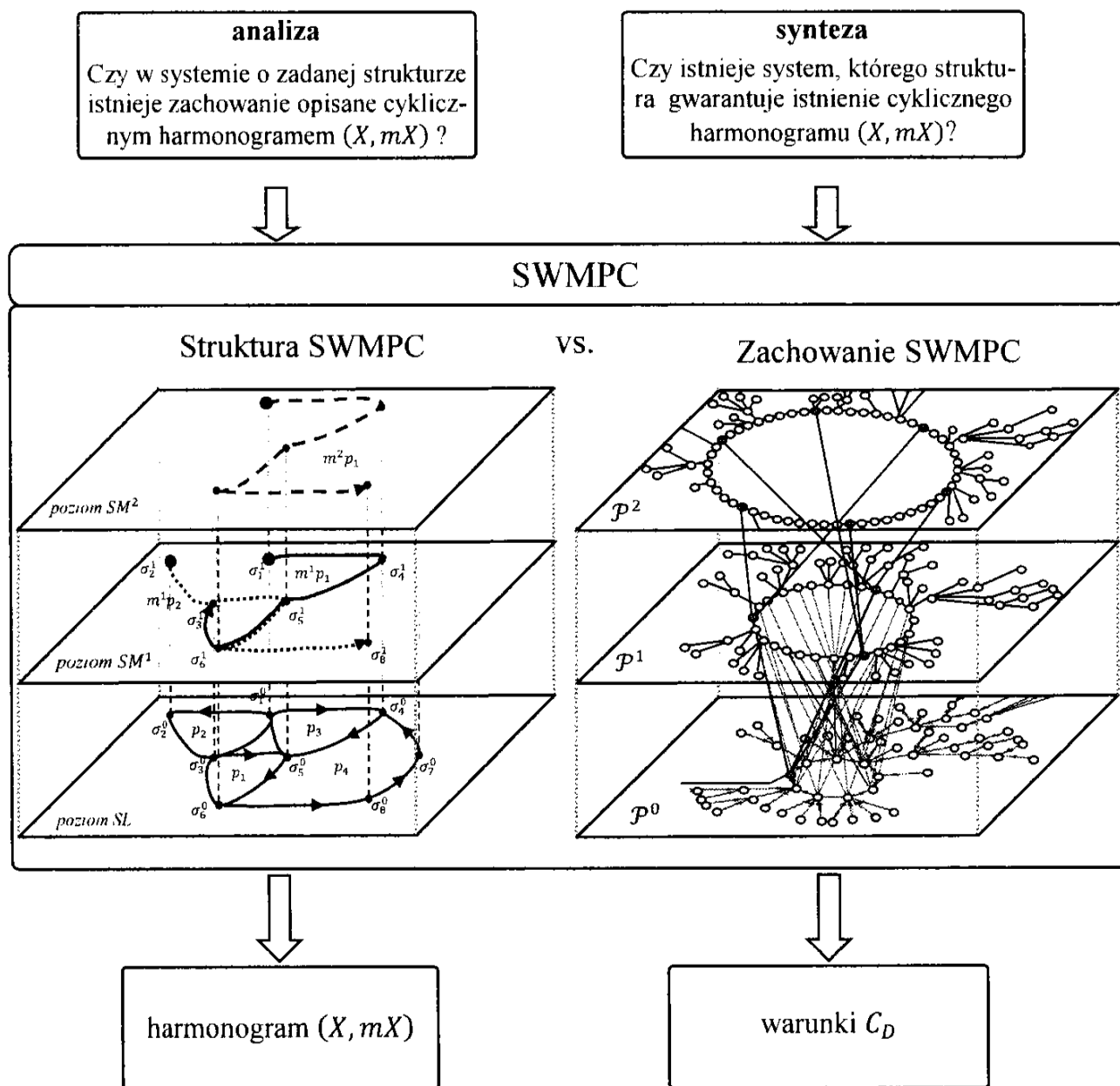
Na problemy harmonogramowania cyklicznego można również spojrzeć jako na Problemy Spełniania Ograniczeń (PSO) (znane z obszarów programowania deklaratywnego). Problemy analizy, syntezy i składania formułowane są w tym przypadku

jako trójka: zbioru zmiennych decyzyjnych  $V$ , dziedzin  $D$  określających ich (całkowitoliczbowe) wartości oraz ograniczeń  $C$  opisujących związki występujące między nimi. Rozwiązanie PSO sprowadza się do wyznaczania (oceny istnienia) takich wartości zmiennych decyzyjnych  $V$ , dla których spełnione będą wszystkie ograniczenia ze zbioru  $C$ .

Podjęcie deklaratywne będzie wykorzystywane w kolejnych rozdziałach przy reprezentacji modeli odnoszących się do problemów harmonogramowania cyklicznego.

## 2. System współbieżnych multimodalnych procesów cyklicznych SWMPC

Przyjęta w poprzednim rozdziale koncepcja problemów harmonogramowania procesów realizowanych w ramach SWMPC (problemów analizy, syntezy oraz składania) pozwala spojrzeć na model tego typu systemu w kontekście jego struktury i zachowania. Pojęcie struktury intuicyjnie rozumiane jest jako zbiór zmiennych/parametrów oraz ograniczeń charakteryzujących marszruty, zasoby, mechanizm synchronizacji, itp. Z kolei zachowanie utożsamiane jest z dopuszczalnym harmonogramem  $(X, mX)$  realizacji operacji procesów. W tym kontekście struktura i zachowanie systemu są wzajemnie od siebie zależne. Określona struktura determinuje potencjalne zachowanie systemu, z kolei określone zachowanie może być osiągalne w systemach o różnych strukturach.



Rys. 2.1. Związek problemów analizy i syntezy SWMPC z jego zachowaniem i strukturą



Związek ten widoczny jest także w sposobie formułowania problemów harmonogramowania cyklicznego. Problem analizy związany z poszukiwaniem cyklicznego harmonogramu o zadanych parametrach (np. określających ilość wykonań procesów w cyklu) jest przykładem, w którym dla zadanej struktury systemu poszukiwane jest oczekiwane dopuszczalne zachowanie. Z kolei, poszukiwanie struktury systemu (np. marszrut procesów, pojemności zasobów, mechanizmów synchronizacji, itp.), w którym osiągalne jest zadane zachowanie cykliczne związane jest z problemem syntezy odpowiedniej struktury.

Celem rozdziału jest opis modelu SWMPC w kontekście jego struktury i zachowania. Wyróżnienie elementów struktury i zachowania (rys. 2.1) w sposób naturalny wpisuje się w podejście deklaratywne, pozwalające rozważać problemy harmonogramowania cyklicznego (analizy, syntezy oraz składania) w kategoriach problemów spełniania ograniczeń.

## 2.1. Struktura SWMPC

### 2.1.1. Poziomy zachowań procesów współbieżnych

W celu przybliżenia przyjętej koncepcji struktury SWMPC warto ponownie odwołać się do przykładu linii metra. Procesy multimodalne interpretowane jako pasażerowie wykorzystujący różne linie metra, można zilustrować tak jak na rys. 2.2. Na przedstawionym planie wyróżnione zostały dwie trasy  $m^1P_1$  oraz  $m^1P_2$ . Pasażerowie podróżujący trasą  $m^1P_1$  wykorzystują dwie linie metra  $P_1$  i  $P_3$  pasażerowie trasy  $m^1P_2$  wykorzystują linie  $P_3$ ,  $P_2$ . Sami pasażerowie są z kolei wykorzystywani do przewozu przesyłek. Trasa przesyłek  $m^2P_1$  składa się z fragmentu trasy  $m^1P_1$  oraz  $m^1P_2$ . Miejscem przekazywania przesyłek między pasażerami jest stacja  $R_5$ .

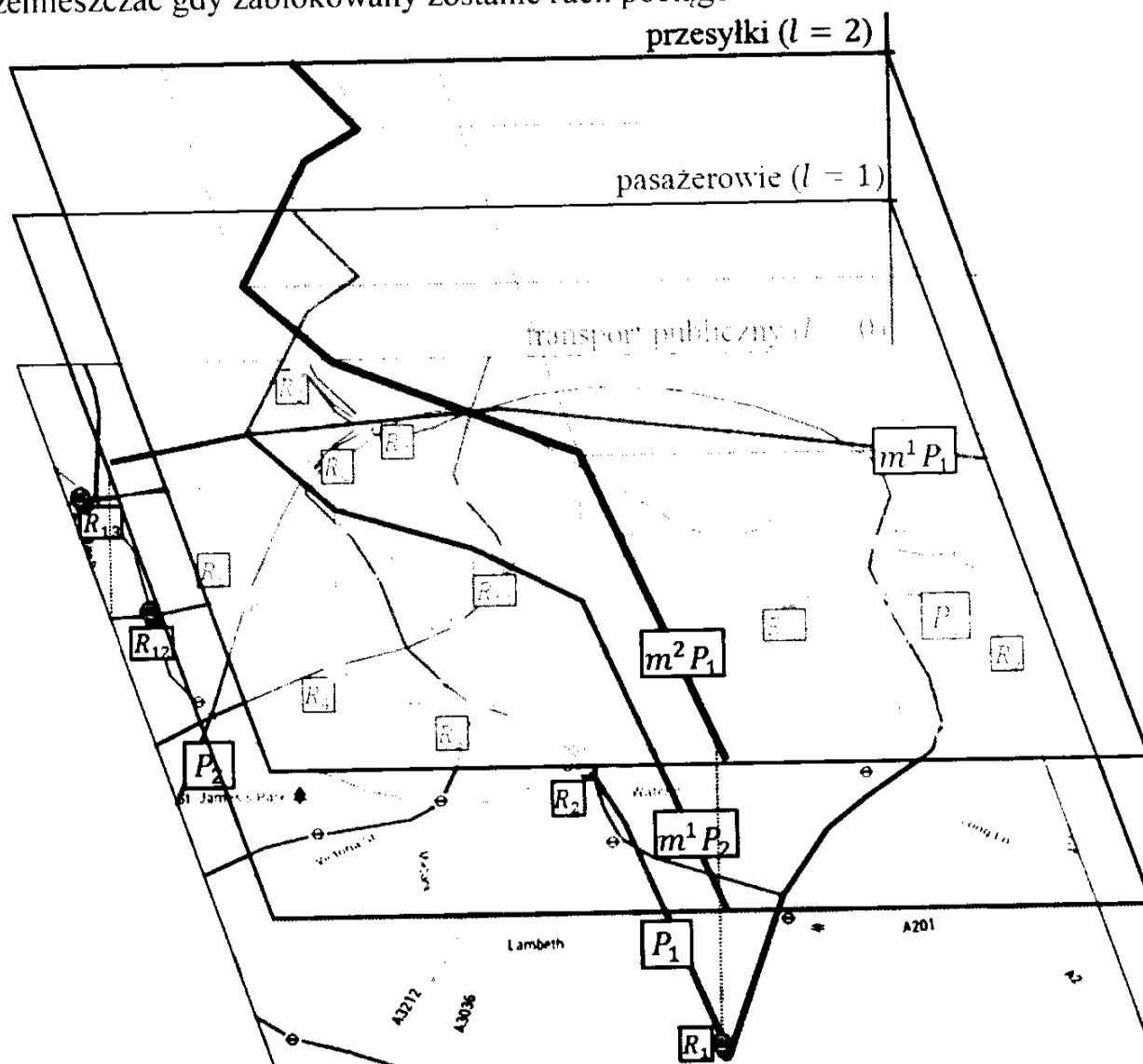
Trasa przesyłek została umieszczona na poziomie wyższym niż trasy pasażerów, natomiast trasy pasażerów na poziomie wyższym niż linie metra. Przynależność trasy (procesu) do poziomu  $l$  została oznaczona przez indeks górny przy literze  $m$ :  $m^lP_i$ .

Procesy umieszczone na najniższym poziomie (linie metra) – procesy lokalne – są niezależne od innych procesów. Realizacja tego typu procesów zależy jedynie od przyjętych, na określonych zasobach (stacjach), mechanizmów synchronizacji.

Realizacja procesów z kolejnych poziomów (pasażerowie) – procesy multimodalne pierwszego poziomu – podobnie jak linie metra są zależne od mechanizmów synchronizacji jak i również od zachowania procesów lokalnych niższego poziomu. Relację tę można scharakteryzować następująco: możliwość przemieszczenia pasażera między stacjami jest uwarunkowana tym czy pasażer posiada bilet upoważniający go do przejazdu (reguły synchronizacji) oraz obecnością pociągu na stacji (możliwością skorzystania z procesu lokalnego). Inaczej mówiąc zachowanie procesów multimodalnych pierwszego poziomu jest uzależnione od zachowania procesów lokalnych.

Podobna relacja łączy wyższe poziomy. Transport przesyłek – procesów multimodalnych poziomu 2 – jest również zależny od obecności środków transportu, czyli pasażerów – procesów multimodalnych poziomu 1.

Przesyłki nie będą mogły być przenoszone jeżeli pasażerowie utkną w korku (blokadzie) i nie będą mogli się przemieszczać. Sami pasażerowie też nie będą mogli się przemieszczać gdy zablokowany zostanie ruch pociągów.



Rys. 2.2. Model wielopoziomowy multimodalnego transportu publicznego (Londyn, maps.google.pl)

Uogólniając, zachowanie cykliczne procesów  $l$ -tego poziomu jest warunkowane zachowaniem cyklicznym procesów z poziomu  $l - 1$ .

Związek między poziomami zachowań ma istotne znaczenie w przypadku oceny ogólnego (obejmującego wszystkie poziomy) zachowania cyklicznego SWMPC. Aby system był cykliczny wymagana jest cykliczność na każdym poziomie zachowań. Blokada procesów na którymkolwiek z poziomów zachowań (nawet jeżeli na niższych poziomach zachowanie cykliczne jest osiągalne) jest blokadą całego systemu. Przykładowo wstrzymanie (zablokowanie) pociągów linii metra uniemożliwia tak transport pasażerów jak i przenoszonych przez nich przesyłek.

Oznacza to, że cykliczne zachowanie jednego poziomu SWMPC implikuje cykliczne zachowanie kolejnego poziomu. Warunek cykliczności procesów niższego poziomu należy traktować wyłącznie jako warunek konieczny, ponieważ zachowanie procesów zależy również od przyjętych na danym poziomie mechanizmów synchronizacji.

Związek między mechanizmem synchronizacji a sposobem realizacji procesów stanowi przykład relacji struktura – zachowanie. Mechanizm synchronizacji wyrażany na przykład poprzez reguły priorytetowania stanowi element struktury determinującej jej potencjalne (cykliczne bądź nie) zachowania.

W ogólnym przypadku, oprócz wspomnianego wyżej mechanizmu synchronizacji procesów, struktura SWMPC charakteryzowana jest przez cały szereg innych parametrów. Formalna definicja SWMPC zakłada, że:

- W systemie istnieje skończony zbiór zasobów odnawialnych  $R = \{R_k \mid k = 1 \dots lk\}$ , które są wspólnie wykorzystywane przez skończony zbiór procesów lokalnych  $P = \{P_i \mid i = 1 \dots ln\}$  i multimodalnych  $mP$ .
- Procesy multimodalne dzielą się na podzbiory procesów należących do wspólnego poziomu zachowań:  $mP = \{m^l P \mid l = 1 \dots lp\}$ , gdzie  $m^l P$  oznacza zbiór procesów  $l$ -tego poziomu zachowań. **Poziom zachowań** ( $l$ -ty) należy rozumieć jako zbiór procesów wykorzystujących do swojej realizacji procesy należące do niższego poziomu ( $l - 1$ ). Zgodnie z tą zasadą procesy należące do  $m^l P$  wykorzystują procesy lokalne  $P$  (traktowane są one jako procesy zerowego poziomu  $l = 0$ ), procesy należące do  $m^2 P$  wykorzystują procesy z  $m^1 P$  i tak dalej, procesy należące do  $m^l P$  wykorzystują procesy z  $m^{(l-1)} P$ . W sytuacji przedstawionej na rys. 2.2 wyróżnia się dwa poziomy procesów multimodalnych:  $m^1 P$  i  $m^2 P$ . Do  $m^1 P$  należy proces  $m^1 P_1$  oraz  $m^1 P_2$  natomiast do  $m^2 P$  należy  $m^2 P_1$ . Poziom procesów określany jest indeksem górnym  $l$  przy literze  $m$ , w przypadku występowania jednego poziomu procesów multimodalnych (tak jak to jest w systemie z rys. 1.30) indeks ten będzie pomijany.
- Każdy zasób  $R_k \in R$  charakteryzowany jest przez pojemność  $C(R_k) \in \mathbb{N}^+$  określającą liczbę niepodzielnych jednostek, z których mogą korzystać procesy realizowane na tym zasobie. Przykładowo  $C(R_k) = 3$  oznacza, że na zasobie  $R_k$  mogą być jednocześnie realizowane 3 procesy wymagające 1 jednostkę zasobu, lub 1 proces wymagający 3 jednostek. Jeżeli to nie jest powiedziane inaczej przyjmuje się, że  $C(R_k) = 1$ .
- W przypadku procesów lokalnych dostęp do jednostek zasobu  $R_k \in R$  odbywa się zgodnie z regułą wzajemnego wykluczania, tj. jedna jednostka w danej chwili może być przydzielona tylko jednemu procesowi lokalnemu.
- W przypadku procesów multimodalnych należących do wspólnego  $l$ -tego poziomu dopuszcza się jednoczesne współdzielenie jednostek zasobu  $R_k \in R$ . Warunek wzajemnego wykluczania może być stosowany opcjonalnie w zależności od zastosowań SWMPC. Jeżeli to nie jest powiedziane inaczej przyjmuje się, że jednostki zasobu  $R_k$  mogą być współdzielone jednocześnie przez wiele procesów należących do  $l$ -tego poziomu.
- Jednostki zasobu  $R_k \in R$  (bez względu na ustalenia poprzedniego punktu) są współdzielone przez procesy multimodalne należące do różnych poziomów. Odwołując się ponownie do przykładu z rys. 2.2 oznacza to, że procesy pasażerów ( $m^1 P$ ) mogą przebywać jednocześnie na wspólnym zasobie (wykorzystywać jednostkę zasobu) z procesami przesyłek ( $m^2 P$ ).

- Każdemu  $i$ -temu procesowi lokalnemu  $P_i$  przypisana jest marszruta będąca sekwencją zasobów wykorzystywanych przez ten proces podczas jego realizacji:  $p_i = (p_{i,1}, \dots, p_{i,j}, \dots, p_{i,lr(i)}), p_{i,j} \in R$ .
- Każdemu  $i$ -temu procesowi multimodalnemu  $m^l P_i$  przypisana jest marszruta będąca sekwencją zasobów wykorzystywanych przez ten proces podczas jego realizacji (będąca rozszerzeniem (12) i (13)):

$$m^l p_i = (m^l p_{i,1}, \dots, m^l p_{i,j}, \dots, m^l p_{i,lm(i,l)}), m^l p_{i,j} \in R. \quad (16)$$

Marszruta (16) stanowi połączenie wybranych fragmentów marszrut procesów z poziomu  $l - 1$ :

$$m^l p_i = \left( m^{(l-1)} pr_{i_1}(a_{i_1}, b_{i_1}) \wedge m^{(l-1)} pr_{i_2}(a_{i_2}, b_{i_2}) \wedge \dots \wedge m^{(l-1)} pr_{i_y}(a_{i_y}, b_{i_y}) \right), \quad (17)$$

gdzie :

$$m^l pr_i(a, b) = \begin{cases} (m^l p_{i,a}, m^l p_{i,a+1}, \dots, m^l p_{i,b}) & \text{dla } a \leq b \\ (m^l p_{i,a}, m^l p_{i,a+1}, \dots, m^l p_{i,lr(i)}, m^l p_{i,1}, \dots, m^l p_{i,b}) & \text{dla } a > b \end{cases}$$

$$a, b \in \{1, \dots, lm(i, l)\},$$

jest fragmentem marszrut  $m^l p_i$  zawierający elementy od  $m^l p_{i,a}$  do  $m^l p_{i,b}$ .

$x \wedge y$  – oznacza konkatencję sekwencji  $x$  i  $y$ , w przypadku gdy:  $x = (x_1, \dots, x_n)$  a  $y = (y_1, \dots, y_m)$ , to  $x \wedge y = (x_1, \dots, x_n, y_1, \dots, y_m)$ .

Przykładowo, marszruta procesu  $m^1 P_1$  z rys. 2.2 (poziom  $l = 1$ ), jest utworzona z fragmentów  $(R_9, R_{10}, R_{11}, R_5)$  oraz  $(R_6, R_7, R_8)$  pochodzących z marszrut procesów lokalnych odpowiednio  $P_3$  i  $P_2$ :

$$m^1 p_1 = ((R_9, R_{10}, R_{11}, R_5) \wedge (R_6, R_7, R_8)) = (R_9, R_{10}, R_{11}, R_5, R_6, R_7, R_8).$$

Analogicznie marszruta procesu  $m^1 P_2$  składa się z fragmentów marszrut  $P_1$  i  $P_3$ :

$$m^1 p_2 = ((R_1, R_2, R_3, R_4, R_5) \wedge (R_{12})) = (R_1, R_2, R_3, R_4, R_5, R_{12}).$$

Marszruta procesu  $m^2 P_1$  (poziom  $l = 2$ ) jest natomiast utworzona z fragmentów marszrut  $m^1 P_2$  i  $m^1 P_1$  (poziom  $l = 1$ ):

$$m^2 p_1 = ((R_1, R_2, R_3, R_4, R_5) \wedge (R_6, R_7, R_8)) = (R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8)$$

Analogicznie jak (12) i (13) zakłada się, że występujące po sobie fragmenty marszrut:  $m^{(l-1)} pr_{i_1}(a_{i_1}, b_{i_1}) \wedge m^{(l-1)} pr_{i_2}(a_{i_2}, b_{i_2})$  łączy ze sobą wspólny zasób  $m^l p_{i,b_{i_1}}$ . Przykładowo, fragment marszrut  $(R_1, R_2, R_3, R_4, R_5)$  procesu  $m^1 p_2$  oraz fragment  $(R_6, R_7, R_8)$  procesu  $m^1 p_1$  wykorzystane w  $m^2 p_1$ , łączy zasób  $R_5$ .

- Realizowane w systemie procesy współbieżne są traktowane jako procesy wieloprotokowe. Potoki procesu lokalnego  $P_i$  rozumiane są jako procesy o wspólnej marszrucie  $p_i$ . Inaczej mówiąc można je traktować jako wielokrotną wersję procesu  $P_i$ . Potoki oznaczane są indeksem górnym  $h$  umieszczanym przy literze  $P$ . W tym ujęciu proces  $P_i$  jest traktowany jako zbiór potoków:  $P_i = \{P_i^1, P_i^2, \dots, P_i^h, \dots, P_i^{ls(i)}\}$ ,

gdzie:  $P_i^h$  oznacza  $h$ -ty potok  $i$ -tego procesu lokalnego. Przykład wielopotokowego systemu został już przedstawiony na rys. 1.30, procesy  $P_4$  i  $P_5$  zwierają po dwa potoki:  $P_4 = \{P_4^1, P_4^2\}$ ,  $P_5 = \{P_4^1, P_4^2\}$ . W przypadku systemów, w których wszystkie procesy są jednopotokowe indeks  $h$  będzie pomijany.

Analogicznie wyróżnia się potoki procesów multimodalnych  $m^l P_i$ . Są to procesy o wspólnej marszrucie  $m^l p_i$ :  $m^l P_i = \{m^l P_i^1, m^l P_i^2, \dots, m^l P_i^h, \dots, m^l P_i^{ism(i,l)}\}$ , gdzie:  $m^l P_i^h$  oznacza  $h$ -ty potok  $i$ -tego procesu multimodalnego należącego do  $l$ -tego poziomu zachowań.

- Realizacja potoku procesu lokalnego  $P_i^h$  (potok procesu multimodalnego  $m^l P_i^h$ ) związana jest z wykonaniem skończonej liczby operacji, których (liniowy) porządek jest określony sekwencją  $O_i^h = (o_{i,1}^h, o_{i,2}^h, \dots, o_{i,j}^h, \dots, o_{i,lr(i)}^h)$ ,  $o_{i,j}^h$  oznacza  $j$ -tą operację  $h$ -tego potoku  $P_i^h$ , ( $m^l O_i^h = (m^l o_{i,1}^h, m^l o_{i,2}^h, \dots, m^l o_{i,j}^h, \dots, m^l o_{i,lm(i,l)}^h)$ ), gdzie  $m^l o_{i,j}^h$  oznacza  $j$ -tą operację potoku  $m^l P_i^h$ ).
- Do wykonania operacji  $o_{i,j}^h$  potoku procesu lokalnego niezbędny jest zasób  $p_{i,j} \in R$  określony w marszrucie  $p_i$  procesu  $P_i$ . Liczba jednostek zasobu  $p_{i,j}$  niezbędnych do wykonania operacji  $o_{i,j}^h$ , określana jest przez  $\omega(o_{i,j}^h) = \omega_{i,j}^h \in \mathbb{N}^+$ . Jeżeli to nie jest powiedziane inaczej przyjmuje się, że  $\omega_{i,j}^h = 1$ .
- Do wykonania operacji  $m^l o_{i,j}^h$  potoku procesów multimodalnych  $l$ -tego poziomu niezbędny jest dokładnie jeden zasób  $m^l p_{i,j} \in R$  określony w marszrucie  $m^l p_i$  procesu  $m^l P_i$  oraz jeden potok  $m^l \mu_{i,j} \in m^{(l-1)} P$  z poziomu  $l-1$ . Wykorzystanie potoku  $m^l \mu_{i,j}$  możliwe jest tylko w trakcie wykonywania przez niego operacji na zasobie  $m^l p_{i,j}$ . W praktyce oznacza to, że operacja przemieszczenia pasażera może odbywać się tylko w trakcie przemieszczania się pociągu. Liczba jednostek zasobu  $m^l p_{i,j}$  niezbędnych do wykonania operacji  $m^l o_{i,j}^h$  określana jest przez  $\omega(m^l o_{i,j}^h) = m^l \omega_{i,j}^h \in \mathbb{N}^+$ . Jeżeli to nie jest powiedziane inaczej przyjmuje się, że  $m^l \omega_{i,j}^h = 1$ .
- Procesy multimodalne mogą współdzielić jednocześnie proces wymagany do realizacji ich operacji. W praktyce oznacza to, że kilku pasażerów może jednocześnie korzystać z tej samej linii metra.
- Dla każdej operacji  $o_{i,j}^h$  potoku procesu lokalnego określony jest czas jej wykonania  $t_{i,j} \in \mathbb{N}^+$  (potoki danego procesu charakteryzują się wspólnymi czasami wykonania operacji). Przypadki gdy zmienna  $t_{i,j}$  przyjmuje wartość losową lub rozmytą (Pawlewski 2013) nie są dalej rozważane.
- Dla każdej operacji  $m^l o_{i,j}^h$  określony jest czas jej wykonania  $m^l t_{i,j} \in \mathbb{N}^+$ .
- Dla wszystkich procesów, dla których obowiązuje warunek wzajemnego wykluczenia stosowany jest mechanizm synchronizacji w postaci **statycznych reguł priorytetowania**  $\Theta$ . Reguły są definiowane w postaci zbioru  $\Theta = \{\theta^l | l = 0 \dots lp\}$ , gdzie:  $\theta^l$  określa zbiór reguł na  $l$ -tym poziomie zachowania ( $l = 0$  – oznacza reguły dla procesów lokalnych,  $l > 0$  oznacza reguły dla kolejnych poziomów procesów multimodalnych). W przypadku gdy w systemie istnieje tylko jeden poziom ( $lp = 0$ ) indeks  $l$  jest dalej pomijany.

Przyjmuje się, że  $\theta^l = \{\sigma_1^l, \dots, \sigma_k^l, \dots, \sigma_{lk}^l\}$ , gdzie:  $\sigma_k^l$  – oznacza regułę określającą mechanizm synchronizacji przyjęty dla zasobu  $R_k \in R$  dla procesów z  $l$ -tego poziomu zachowania. Reguły  $\sigma_k^l$  przypisywane są dla każdego z zasobów bez względu na to czy jest to zasób współdzielony czy nie. **Raz określone reguły  $\theta^l$  nie ulegają zmianie w trakcie pracy SWMPC** – wyjątek stanowi zmiana reguł umożliwiająca zmianę cyklicznego przebiegu ustalonego, co szczegółowo opisane jest w rozdziale 5.

Reguła priorytetowania  $\sigma_k^l$  definiowana jest jako sekwencja, której elementy określają porządek, z jakim potoki procesów otrzymują dostęp do zasobu  $R_k$ :  $\sigma_k^l = (s_{k,1}^l, \dots, s_{k,d}^l, \dots, s_{k,lh(k,l)}^l)$ , gdzie:  $s_{k,d}^l$  określa potok  $s_{k,d}^l \in H_k^l$  wykonywany na zasobie  $R_k$  jako  $d$ -ty; gdzie:  $H_k^l$  oznacza zbiór potoków  $l$ -tego poziomu ( $l = 0$  – potoki procesów lokalnych,  $l > 0$  – potoki procesów multimodalnych) wykorzystujących w swoich operacjach zasób  $R_k$ . **Przyjmuje się, że każdy potok wykorzystujący zasób  $R_k$  otrzymuje do niego dostęp przynajmniej raz na jeden cykl systemu.** Inaczej mówiąc, każdy element zbioru  $H_k^l$  występuje co najmniej raz w regule  $\sigma_k^l$ .

Przykład tego typu reguł priorytetowania (choć tylko w ujęciu procesów lokalnych) został użyty w systemach z rys. 1.22 oraz w systemie z przykładu 3. Reguły stosowane w tym przykładzie mają postać:  $\theta^0 = \{\sigma_1^0, \sigma_2^0, \sigma_3^0, \sigma_4^0, \sigma_5^0, \sigma_6^0\}$ ,

$$\sigma_1^0 = (P_3, P_2), \sigma_2^0 = (P_2), \sigma_3^0 = (P_2, P_1), \sigma_4^0 = (P_1), \sigma_5^0 = (P_3, P_1), \sigma_6^0 = (P_3).$$

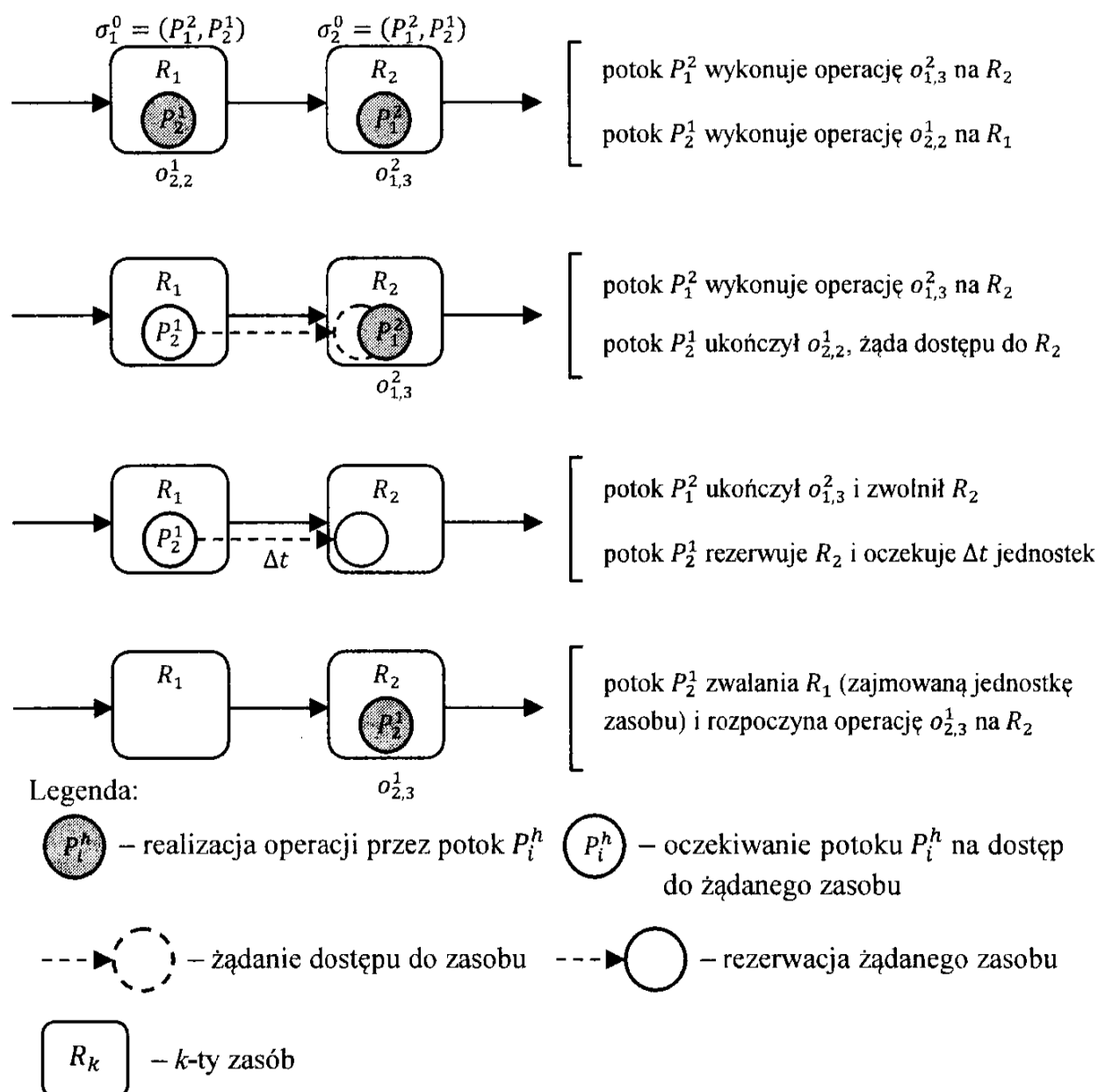
Reguła  $\sigma_1^0$ , (w której  $s_{1,1}^0 = P_3, s_{1,2}^0 = P_2$ ) oznacza, że dostęp potoków procesów do zasobu  $R_1$  odbywa się w kolejności  $\dots, P_3, P_2, P_3, P_2, \dots$ . W przypadku  $\sigma_2^0$  dostęp do zasobu ma tylko potok procesu  $P_2$ .

Kolejność potoków określona w regule  $\sigma_k^l$  oznacza kolejność w jakiej operacje tych potoków wykonują się na tym zasobie. Znaczy to, że potok określony przez  $s_{k,d}^l$  otrzyma dostęp do zasobu  $R_k$  tylko i wyłącznie pod warunkiem wykonania na tym zasobie operacji potoku go poprzedzającego, określonego przez  $s_{k,(d-1)}^l$ .

Dodatkowo, dla każdej reguły  $\sigma_k^l$  ze zbioru  $\theta^l$ , przyjmuje się stałą liczbę wystąpień każdego potoku. Znaczy to, że jeżeli potok procesu  $m^l P_i$  występuje w danej regule  $\sigma_k^l$  tylko raz to będzie on występował tylko raz również w każdej regule związanej z marszrutą tego procesu. Liczba wystąpień potoku procesu  $m^l P_i$  w regule  $\sigma_k^l$  oznaczana jako  $\psi_i^l \in \mathbb{N}^+$  określa jednocześnie liczbę wykonań potoku względem innych potoków obsługiwanych przez zasób  $R_k$ . Przykładowo w regule  $\sigma_1^2 = (m^2 P_1^1, m^2 P_2^1, m^2 P_1^1, m^2 P_2^1, m^2 P_1^1)$  dwukrotnie występuje potok  $m^2 P_2^1$ :  $\psi_2^2 = 2$  oraz trzykrotnie  $m^2 P_1^1$ :  $\psi_3^2 = 3$ . Tak więc, na trzy wykonania potoku  $m^2 P_1^1$  przypadają dwa wykonania potoku  $m^2 P_2^1$ . To, że liczba wykonań potoków należących do jednego procesu jest taka sama, wynika z tego że wielkość  $\psi_i^l$  odnosi się zawsze do procesu a nie do jego potoku. Dla każdego ( $l$ -tego) poziomu zachowań definiowana jest sekwencja liczby wzajemnych wykonań procesów:  $\psi^l = (\psi_1^l, \dots, \psi_i^l, \dots, \psi_{lw(l)}^l)$ .

- Rozpoczęcie kolejnej operacji  $o_{i,j}^h$  przez potok  $P_i^h$  procesu lokalnego składa się z następujących etapów:

- żądanie dostępu do  $\omega_{i,j}^h$  jednostek zasobu  $p_{i,j}$ , niezbędnego do wykonania operacji  $o_{i,j}^h$ ; żądanie dostępu do zasobu  $p_{i,j}$  odbywa się natychmiast po ukończeniu bieżącej operacji  $o_{i,(j-1)}^h$  (poprzedzającej  $o_{i,j}^h$ ) realizowanej na zasobie  $p_{i,(j-1)}$ ,
- oczekiwanie na dostęp do żądanego zasobu, oczekujący potok nie zwalnia wykorzystywanych przez niego jednostek zasobu  $p_{i,(j-1)}$ , uzyskanie dostępu do zasobu odbywa się zgodnie z przyjętą dla tego zasobu regułą priorytetowania,
- uzyskanie dostępu do żądanego zasobu (rezerwacja jednostek zasobu), potok otrzymuje dostęp do wszystkich żądanych jednostek (nie zakłada się częściowej rezerwacji jednostek zasobów),



**Rys. 2.3.** Etapy rozpoczęcia operacji  $o_{2,3}^1$  potoku  $P_2^1$  procesu lokalnego  $P_2$

- uwolnienie wykorzystywanych do tej pory jednostek zasobu  $p_{i,(j-1)}$  z opóźnieniem  $\Delta t \geq 0$  liczonym od momentu uzyskania dostępu do zasobu  $p_{i,j}$  (opóźnienie  $\Delta t$  jest związane z czasem reakcji na sygnał uzyskania dostępu do zasobu),
- zajęcie przez potok  $P_i^h$  żądanej ilości  $\omega_{i,j}^h$  jednostek zasobu  $p_{i,j}$  i rozpoczęcie operacji  $o_{i,j}^h$ .

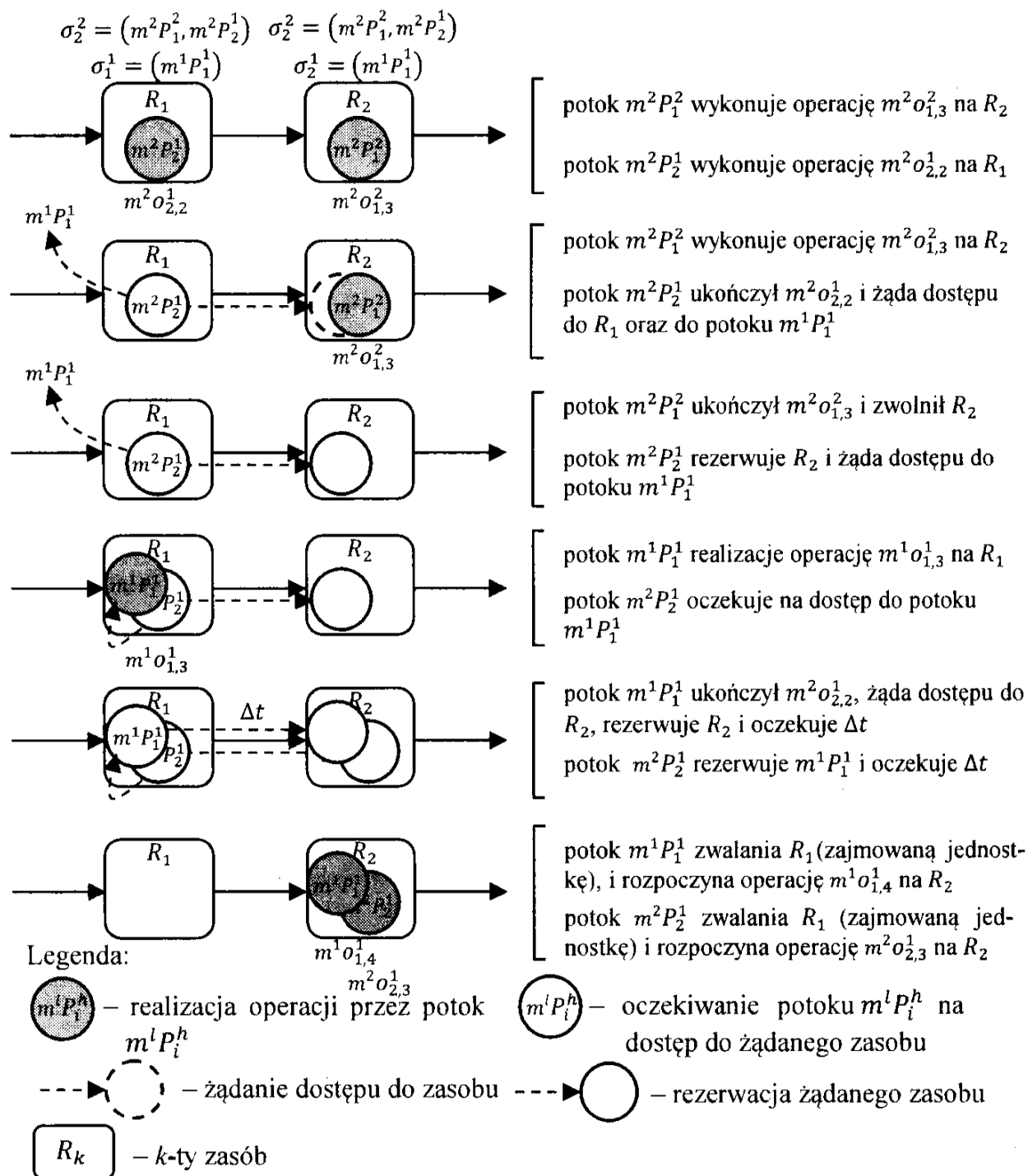
Przykład rozpoczęcia nowej operacji przez potok procesu lokalnego, zgodny z powyższymi etapami, został zilustrowany na rys. 2.3. W przykładzie przyjęto jednostkowe zapotrzebowanie procesów na zasoby:  $\omega_{i,j}^h = 1$ .

- Rozpoczęcie kolejnej operacji  $m^l o_{i,j}^h$  przez potok  $m^l P_i^h$  procesu multimodalnego składa się z następujących etapów:
  - żądanie dostępu do  $m^l \omega_{i,j}^h$  jednostek zasobu  $m^l p_{i,j}$  oraz do potoku  $m^l \mu_{i,j}$ , niezbędnych do wykonania operacji  $m^l o_{i,j}^h$ ; żądanie realizowane jest po ukończeniu bieżącej operacji  $m^l o_{i,(j-1)}^h$  (poprzedzającej  $m^l o_{i,j}^h$ ) realizowanej na zasobie  $m^l p_{i,(j-1)}$ ,
  - oczekiwanie na dostęp do żądanego zasobu, oczekujący potok nie zwalnia wykorzystywanych przez niego jednostek zasobu  $m^l p_{i,(j-1)}$ , uzyskanie dostępu do zasobu odbywa się zgodnie z przyjętą dla tego zasobu regułą priorytetowania,
  - uzyskanie dostępu do żądanego zasobu  $m^l p_{i,j}$  (rezerwacja jednostek zasobu), potok otrzymuje dostęp do wszystkich żądanych jednostek (nie zakłada się częściowej rezerwacji jednostek zasobów),
  - uzyskanie dostępu do potoku żądanego procesu  $m^l \mu_{i,j}$  (rezerwacja procesu), żądany potok musi przebywać na tym samym zasobie  $m^l p_{i,(j-1)}$ ,
  - uwolnienie wykorzystywanych jednostek zasobu ( $m^l p_{i,(j-1)}$ ) w tym samym momencie co potok procesu  $m^l \mu_{i,j}$  (potok  $m^l P_i^h$  oczekuje na potok  $m^l \mu_{i,j}$ ),
  - zajęcie przez potok  $m^l P_i^h$  żądanej ilości  $m^l \omega_{i,j}^h$  jednostek zasobu  $m^l p_{i,j}$  i natychmiastowe rozpoczęcie operacji  $m^l o_{i,j}^h$ ,
  - potok procesu  $m^l \mu_{i,j}$  jest zwalniany po zakończeniu realizowanej przez niego operacji.

Przykład rozpoczęcia nowej operacji przez potok procesu multimodalnego, zgodny z powyższymi etapami, został zilustrowany na rys. 2.4. W przykładzie przyjęto jednostkowe zapotrzebowanie procesów na zasoby:  $m^l \omega_{i,j}^h = 1$ .

- Przyjmuje się, że czas wykonania  $m^l t_{i,j}$  operacji  $m^l o_{i,j}^h$  potoku  $m^l P_i^h$  nie jest mniejszy niż czas wykonania operacji wykonywanej przez żądany potok  $m^l \mu_{i,j}$ . W kontekście przykładu z rys. 2.4 oznacza to, że czas wykonywania  $m^2 t_{2,3}$  operacji  $m^2 o_{2,3}^1$  nie może być mniejszy niż czas trwania  $m^1 t_{1,4}$  operacji  $m^1 o_{1,4}^1$  wykorzystywanego potoku.
  - Po zakończeniu realizacji potoku lokalnego/multimodalnego (ostatniej operacji) potok rozpoczyna się na nowo (ponowne rozpoczęcie pierwszej operacji).
- Reasumując, SWMPC obejmuje wielopotokowe procesy multimodalne różnych poziomów zachowań, tzn. procesy wykorzystujące do swojej realizacji inne procesy (niższych poziomów zachowań).

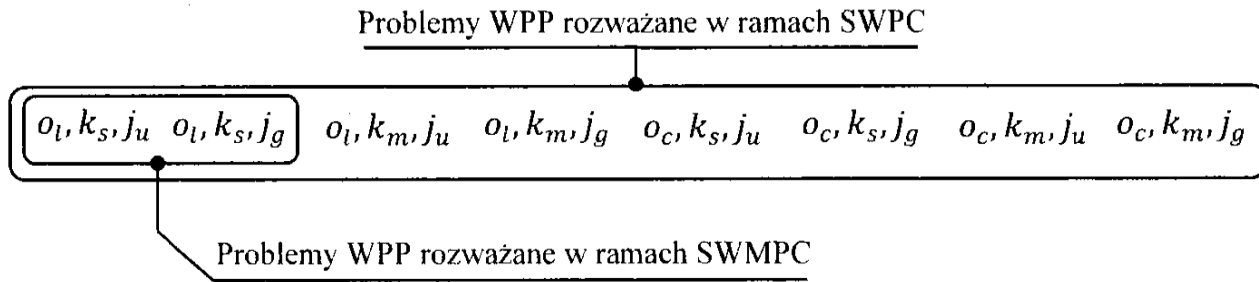




Rys. 2.4. Etapy rozpoczęcia operacji  $m^2 o_{2,3}^2$  potoku  $m^2 P_2^1$  procesu multimodalnego  $m^2 P_2$

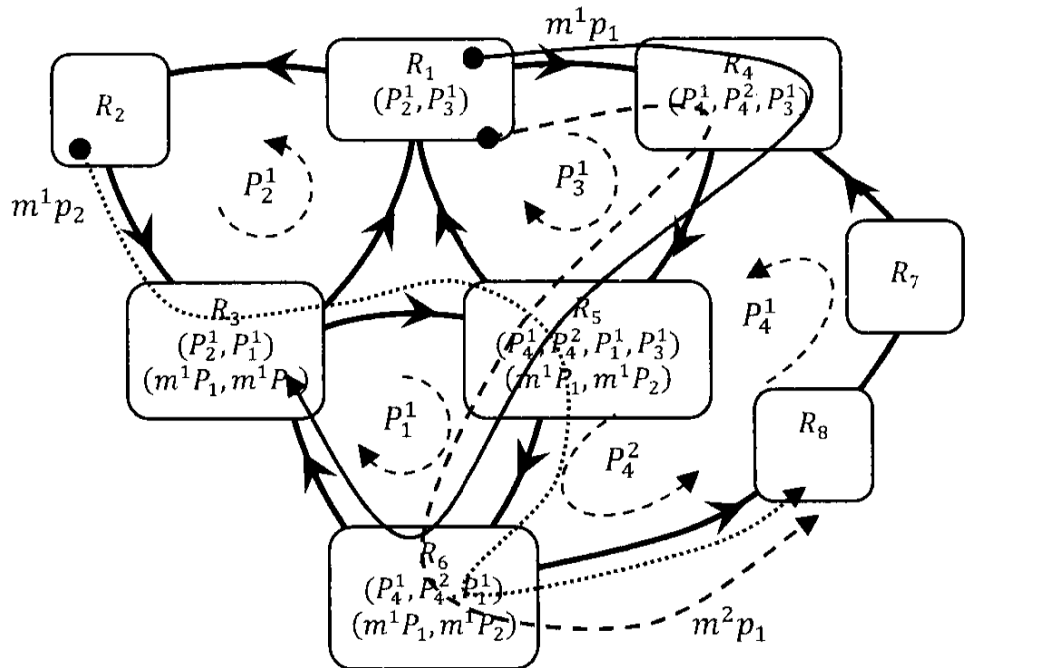
### 2.1.2. Wielopoziomowy model struktury SWMPC

W rozważanej klasie systemów przyjmuje się dwa następujące założenia: każdy proces charakteryzuje liniowy porządek, a każda wykonywana operacja wymaga tylko jednego zasobu. Przyjęcie tych założeń pozwala traktować problemy harmonogramowania SWMPC jako ogólny przypadek (uwzględniający występowanie procesów multimodalnych) problemów typu (rys. 2.5):  $WPP|o_i, k_s, j_u$  (gdy operacje korzystają tylko z jednej jednostki zasobu:  $\omega_{i,j}^h = 1, m^l \omega_{i,j}^h = 1$ ) oraz  $WPP|o_i, k_s, j_g$  (gdy operacje korzystają tylko z wielu jednostek zasobu:  $\omega_{i,j}^h \geq 1, m^l \omega_{i,j}^h \geq 1$ ).

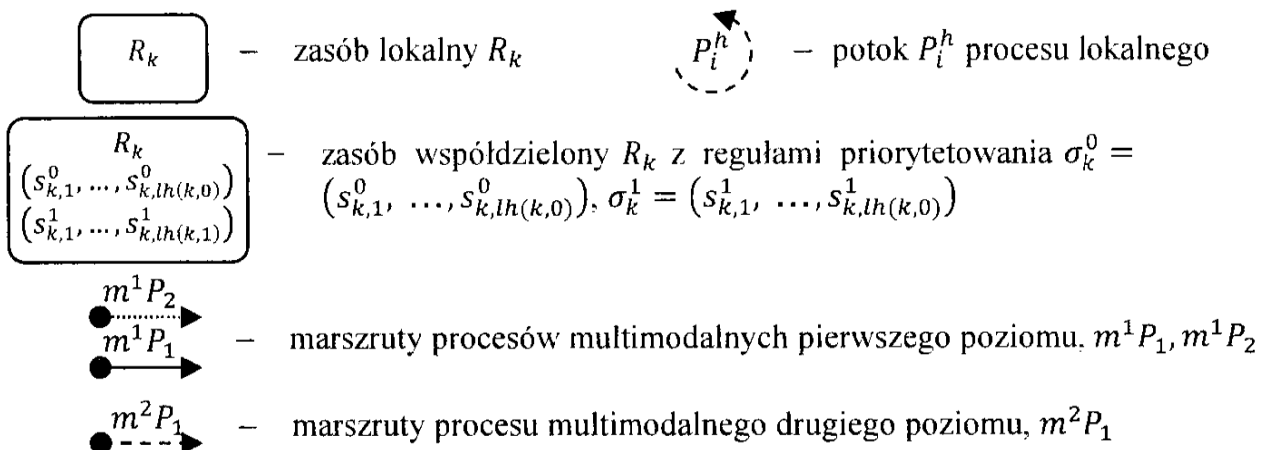


Rys. 2.5. Obszar zastosowań przyjętej klasy SWMPC w kontekście klasy WPP (tab. 1.4)

Przykładem ilustracji rozwiązywanej klasy może być SWMPC z rys. 2.6, w skład którego wchodzi 8 zasobów  $R = \{R_1, \dots, R_8\}$ , są to zasoby o pojemności równej jeden:  $C(R_k) = 1$ . Tego typu system należy do klasy  $WPP|o_l, k_s, j_u$ .



Legenda:



Rys. 2.6. Przykład SWMPC z trzema poziomami zachowań: jednym poziomem procesów lokalnych i dwoma poziomami procesów multimodalnych

W systemie wyróżnia się trzy poziomy zachowań: poziom procesów lokalnych  $P$  oraz dwa poziomy procesów multimodalnych  $mP$ .

**Procesy lokalne.** Zbiór procesów lokalnych zawiera 4 procesy  $P = \{P_1, P_2, P_3, P_4\}$ , którym odpowiadają następujące marszruty:

$$p_1 = (R_3, R_5, R_6), p_2 = (R_2, R_3, R_1), p_3 = (R_1, R_4, R_5), p_4 = (R_7, R_4, R_5, R_6, R_8).$$

Wśród procesów wszystkie poza  $P_4$  są procesami jednopotokowymi:

$$P_1 = \{P_1^1\}, P_2 = \{P_2^1\}, P_3 = \{P_3^1\}, P_4 = \{P_4^1, P_4^2\},$$

każdemu z nich przyporządkowane są sekwencje:

$$O_1^1 = (o_{1,1}^1, o_{1,2}^1, o_{1,3}^1), O_2^1 = (o_{2,1}^1, o_{2,2}^1, o_{2,3}^1), O_3^1 = (o_{3,1}^1, o_{3,2}^1, o_{3,3}^1), \\ O_4^1 = (o_{4,1}^1, o_{4,2}^1, o_{4,3}^1, o_{4,4}^1, o_{4,5}^1), O_4^2 = (o_{4,1}^2, o_{4,2}^2, o_{4,3}^2, o_{4,4}^2, o_{4,5}^2),$$

których operacje wykonywane są kolejno na zasobach określonych w sekwencjach  $p_1, p_2, p_3, p_4$ . Oczywiście oba potoki  $P_4^1, P_4^2$  procesu  $P_4$  realizują swoje operacje wzdłuż tej samej marszruty. Przyjmuje się jednostkowe czasy wykonywania operacji  $t_{i,j} = 1$ , dla  $i = 1 \dots 4, j = 1 \dots lr(i)$ .

Realizacja operacji odbywa się przy założeniu zasady wzajemnego wykluczania oraz niewywłaszczalności zasobów. W zawiązku z tym do każdego z zasobów zapisane zostały następujące reguły priorytetowania:

$$\theta^0 = \{\sigma_1^0, \sigma_2^0, \sigma_3^0, \sigma_4^0, \sigma_5^0, \sigma_6^0, \sigma_7^0, \sigma_8^0\},$$

gdzie:

$$\sigma_1^0 = (P_2^1, P_3^1), \sigma_2^0 = (P_2^1), \sigma_3^0 = (P_2^1, P_1^1), \sigma_4^0 = (P_4^1, P_4^2, P_3^1), \\ \sigma_5^0 = (P_4^1, P_4^2, P_1^1, P_3^1), \sigma_6^0 = (P_4^1, P_4^2, P_1^1), \sigma_7^0 = (P_4^1, P_4^2), \sigma_8^0 = (P_4^1, P_4^2).$$

Każda reguła określa kolejność dostępu potoków do zasobu (reguły jednoelementowe nie są przedstawione na rysunku). Reguły priorytetowania nie determinują pierwszego potoku obsługiwanego na zasobie, określają tylko ich wzajemną kolejność dostępu. Przykładowo dla  $R_6$  kolejność dostępu potoków określona przez  $\sigma_6^0$  jest następująca  $\dots, P_4^1, P_1^1, P_4^2, P_4^1, P_1^1, P_4^2, \dots$ .

Wszystkie procesy wykonują się w proporcjach 1:1, tzn. na jedno wykonanie jednego procesu przypada dokładnie jedno wykonanie wszystkich pozostałych. Odpowiadająca im sekwencja wzajemnych wykonań ma postać:  $\Psi^0 = (1,1,1,1)$ .

**Procesy multimodalne.** Zbiór procesów multimodalnych  $mP = \{m^1P, m^2P\}$  systemu z rys. 2.6 zawiera dwa zbiory procesów  $m^1P, m^2P$  odnoszące się do 1- i 2-poziomu zachowań.

Zbiór  $m^1P$  pierwszego poziomu zachowań zawiera dwa procesy multimodalne  $m^1P_1$  i  $m^1P_2$  wykorzystujące do swojej realizacji procesy lokalne. Proces  $m^1P_1$  (linia ciągła na rys. 2.6) wykorzystuje procesy  $P_3$  i  $P_1$ , proces  $m^1P_2$  (linia kropkowana) wykorzystuje procesy  $P_2, P_1, P_4$ . Marszruty (złożone z fragmentów marszrut procesów lokalnych) tych procesów są następujące:

$$m^1p_1 = ((R_1, R_4, R_5) \wedge (R_6, R_3)) = (R_1, R_4, R_5, R_6, R_3), \\ m^1p_2 = ((R_2, R_3) \wedge (R_5, R_6) \wedge (R_8)) = (R_2, R_3, R_5, R_6, R_8).$$

Oba procesy  $m^1P_1$  i  $m^1P_2$  są procesami jednopotokowymi (z tego względu przy ich oznaczaniu pomijany jest indeks górny).

Podobnie jak w przypadku procesów lokalnych przyjmuje się, że procesy multimodalne synchronizuje protokół wzajemnego wykluczania. Zasobom współdzielonym przez te procesy przydzielone zostały reguły prioryteowania:

$$\theta^1 = \{\sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1, \sigma_5^1, \sigma_6^1, \sigma_8^1\},$$

gdzie:

$$\begin{aligned} \sigma_1^1 &= (m^1P_1), \sigma_2^1 = (m^1P_2), \sigma_3^1 = (m^1P_1, m^1P_2), \sigma_4^1 = (m^1P_1), \\ \sigma_5^1 &= (m^1P_1, m^1P_2), \sigma_6^1 = (m^1P_1, m^1P_2), \sigma_8^1 = (m^1P_2). \end{aligned}$$

Reguły jednoelementowe nie zostały umieszczone na rysunku. Procesy wykonują się w proporcjach 1:1, to znaczy, że  $\Psi^1 = (1,1)$ .

Zbiór  $m^2P$  drugiego ( $l = 2$ ) poziomu zachowań zawiera tylko jeden proces multimodalny  $m^2P_1$  oznaczony na rys. 2.6 przerywaną linią. Proces wykorzystuje w swojej realizacji procesy multimodalne pierwszego ( $l = 1$ ) poziomu:  $m^1P_1, m^1P_2$ . Marszruta tego procesu jest następująca:

$$m^2p_1 = ((R_1, R_4, R_5) \wedge (R_6, R_8)) = (R_1, R_4, R_5, R_6, R_8).$$

Proces  $m^2P_1$ , jest procesem jednopotokowym. Z uwagi na fakt, że jest on jedynym procesem danego poziomu, zbiór reguł priorytetowania jest zbiorem pustym  $\theta^2 = \emptyset$ , natomiast  $\Psi^2 = (1)$ . Dla wszystkich procesów multimodalnych przyjęto jednostkowy czas wykonywania operacji:  $m^l t_{i,j} = 1$ .

Przedstawione parametry składają się na strukturę SWMPC determinującą jego zachowanie. Parametrami takimi są: zasoby (ich pojemność wpływa na potencjalne występowanie blokad), marszruty (określające zasoby współdzielone), reguły priorytetowania, ilość potoków, itp.

Formalnie, struktura SWMPC jest definiowana jest jako następująca  $n$ -tka:

$$SC^{lp} = \left( \left( \left( \left( (R, SL), SM^1 \right), SM^2 \right), \dots, SM^l \right), \dots \right), SM^{lp} \right), \quad (18)$$

gdzie:

$lp$  – liczba poziomów zachowań,

$R = \{R_k \mid k = 1, \dots, lk\}$  – zbiór zasobów,  $lk$  – liczba zasobów w systemie, o znanej pojemności  $C(R_k)$ ,

$SL = (ST_L, BE_L)$  – struktura poziomu zachowań procesów lokalnych, gdzie:

$ST_L = (P, U, O, T)$  – zmienne opisujące, procesy:

$P = \{P_i = \{P_i^h \mid h = 1 \dots ls(i)\} \mid i = 1 \dots ln\}$  – zbiór procesów lokalnych,  $P_i$  –  $i$ -ty proces,  $P_i^h$  –  $h$ -ty potok procesu  $P_i$ ,  $ln$  – liczba procesów lokalnych,  $ls(i)$  liczba potoków procesu  $P_i$ ,

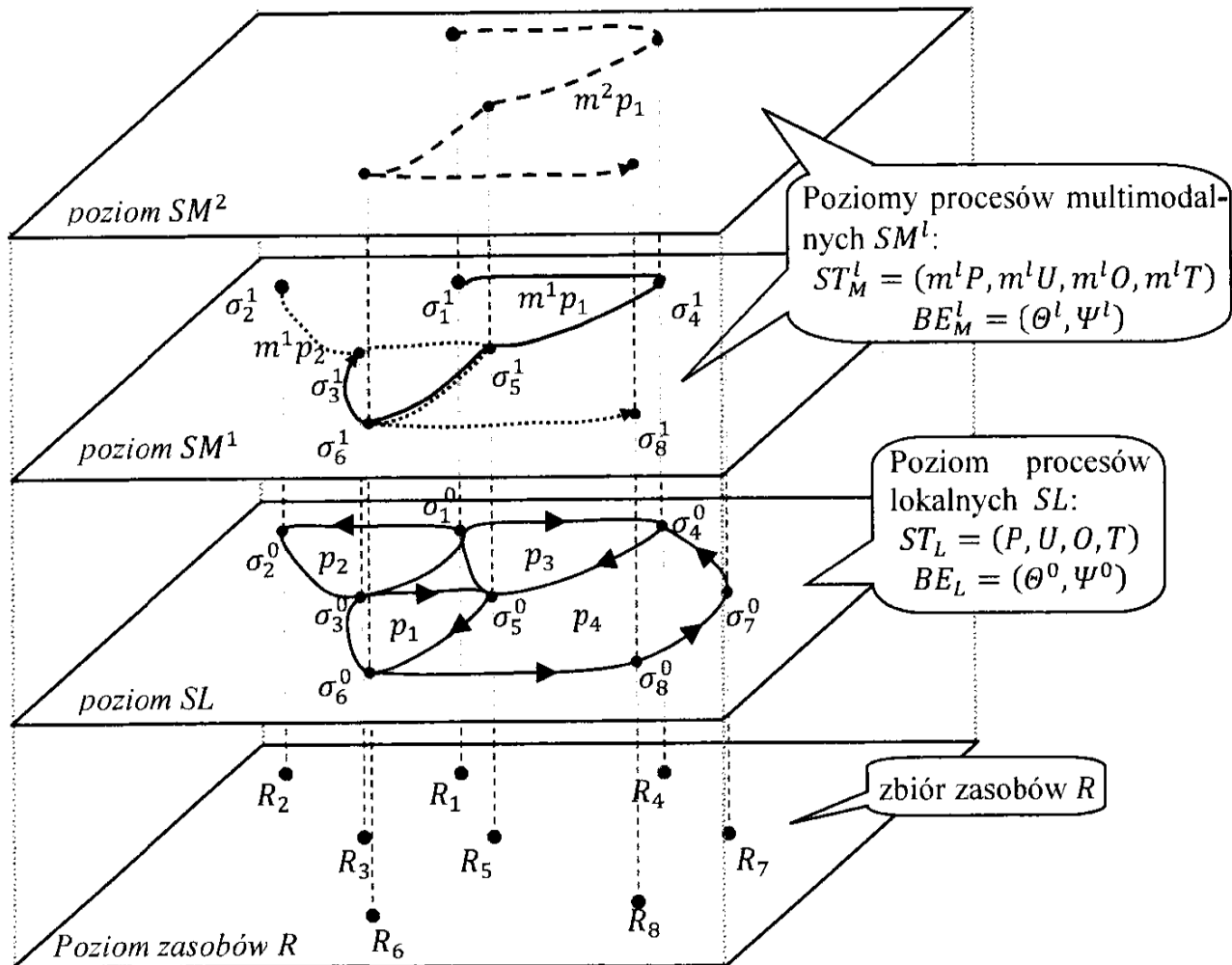
$U = \{p_i = (p_{i,1}, \dots, p_{i,j}, \dots, p_{i,lr(i)}) \mid i = 1 \dots ln\}$  – zbiór marszrut procesów lokalnych,  $p_i$  –  $i$ -ta marszruta,  $p_{i,j} \in R$  – zasób wymagany przy realizacji  $j$ -tej operacji każdego potoku procesu  $P_i$ ,  $lr(i)$  – długość marszruty  $p_i$ ,

$$\begin{aligned}
O &= \{O_i^h = (o_{i,1}^h, \dots, o_{i,j}^h, \dots, o_{i,lr(i)}^h) \mid i = 1 \dots ln; h = 1 \dots ls(i)\} - \text{zbiór} \\
&\text{sekwencji operacji, } o_{i,j}^h - j\text{-ta operacja potoku } P_i^h \text{ wymagająca do} \\
&\text{swojej realizacji zasobu } p_{i,j} \text{ w ilości } \omega_{i,j}^h, \\
T &= \{T_i = (t_{i,1}, \dots, t_{i,j}, \dots, t_{i,lr(i)}) \mid i = 1 \dots ln\} - \text{zbiór sekwencji cza-} \\
&\text{sów wykonania operacji, } t_{i,j} - \text{czas wykonania operacji } o_{i,j}^h, \text{ dla} \\
&h = 1 \dots ls(i), \\
BE_L &= (\Theta^0, \Psi^0) - \text{zmiennne opisujące zachowanie procesów,} \\
\Theta^0 &= \{\sigma_k^0 = (s_{k,1}^0, \dots, s_{k,d}^0, \dots, s_{k,lh(k,0)}^0) \mid k = 1 \dots lk\} - \text{zbiór reguł} \\
&\text{priorytetowania, } \sigma_k^0 - \text{reguła priorytetowania dla zasobu } R_k, s_{k,d}^0 - \\
&\text{potok procesu lokalnego, } lh(k, 0) - \text{długość reguły } \sigma_k^0 \\
\Psi^0 &= (\psi_1^0, \dots, \psi_i^0, \dots, \psi_{ln}^0) - \text{sekwencja wzajemnej liczby wykonań} \\
&\text{procesów lokalnych,} \\
SM^l &= (ST_M^l, BE_M^l) - \text{struktura } l\text{-tego poziomu zachowań procesów multimodal-} \\
&\text{nych, gdzie:} \\
ST_M^l &= (m^l P, m^l U, m^l O, m^l T) - \text{zmiennne opisujące, procesy:} \\
m^l P &= \{m^l P_i = \{m^l P_i^h \mid h = 1 \dots lsm(i, l)\} \mid i = 1 \dots lw(l)\} - \text{zbiór} \\
&\text{procesów multimodalnych } l\text{-tego poziomu, } m^l P_i^h - h\text{-ty potok} \\
&\text{procesu } m^l P_i, lw(i) - \text{liczba procesów } l\text{-tego poziomu, } lsm(i, l) - \\
&\text{liczba potoków procesu } m^l P_i, \\
m^l U &= \{m^l p_i = (m^l p_{i,1}, \dots, m^l p_{i,j}, \dots, m^l p_{i,lm(i,l)}) \mid i = 1 \dots lw(l)\} - \\
&\text{zbiór marszrut procesów lokalnych, } m^l p_i - i\text{-ta marszruta,} \\
&m^l p_{i,j} \in R - \text{zasób wymagany przy realizacji } j\text{-tej operacji każde-} \\
&\text{go potoku procesu } m^l P_i, lm(i, l) - \text{długość marszruty } m^l p_i, \\
m^l O &= \{m^l O_i^h = (m^l o_{i,1}^h, \dots, m^l o_{i,j}^h, \dots, m^l o_{i,lm(i,l)}^h) \mid i = 1 \dots lw(l); \\
&h = 1 \dots lsm(i, l)\} - \text{zbiór sekwencji operacji, } m^l o_{i,j}^h - j\text{-ta operacja} \\
&\text{potoku } m^l P_i^h \text{ wymagająca do swojej realizacji zasobu } m^l p_{i,j} \\
&\text{w ilości } m^l \omega_{i,j}^h \text{ oraz potoku } m^l \mu_{i,j}, \\
m^l T &= \{m^l T_i = (m^l t_{i,1}, \dots, m^l t_{i,j}, \dots, m^l t_{i,lm(i,l)}) \mid i = 1 \dots lw(l)\} - \\
&\text{zbiór sekwencji czasów wykonania operacji, } m^l t_{i,j} - \text{czas wyko-} \\
&\text{nania operacji } m^l o_{i,j}^h, \text{ dla } h = 1 \dots lsm(i, l), \\
BE_M^l &= (\Theta^l, \Psi^l) - \text{zmiennne opisujące zachowanie procesów,} \\
\Theta^l &= \{\sigma_k^l = (s_{k,1}^l, \dots, s_{k,d}^l, \dots, s_{k,lh(k,l)}^l) \mid k = 1 \dots lk\} - \text{zbiór reguł} \\
&\text{priorytetowania } l\text{-tego poziomu, } \sigma_k^l - \text{reguła priorytetowania } l\text{-} \\
&\text{tego poziomu dla zasobu } R_k, s_{k,d}^l - \text{potok procesu multimodalne-} \\
&\text{go, } lh(k, l) - \text{długość reguły } \sigma_k^l, \\
\Psi^l &= (\psi_1^l, \dots, \psi_i^l, \dots, \psi_{lw(l)}^l) - \text{sekwencja wzajemnej liczby wykonań} \\
&\text{procesów multimodalnych } l\text{-tego poziomu.}
\end{aligned}$$

Przyjmuje się, że rozważne są **struktury topologicznie spójne**, to znaczy, że marszrutom procesów każdego poziomu zachowań odpowiada reprezentacja graficzna w postaci spójnego digrafu. Przykładem systemu ze strukturą spójną jest system z rys. 2.6. Zgodnie z (18) struktura tego systemu ma postać:

$$SC^2 = \left( ((R, SL), SM^1), SM^2 \right).$$

Zbiór zasobów  $R = \{R_1, \dots, R_8\}$  systemu z rys. 2.6, poziomy procesów lokalnych  $SL$  (zawierający 4 procesy) oraz poziomy procesów multimodalnych  $SM^1$ ,  $SM^2$  (zawierające kolejno 2 i 1 procesy) ilustruje rys. 2.7, gdzie marszruty procesów tworzą spójne digrafy połączeń.



Rys. 2.7. Graficzna ilustracja struktury (18) na przykładzie SWMPC z rys. 2.6

Przedstawiona struktura systemu określa hierarchię, której najniższy poziom tworzy zbiór zasobów  $R$ . Poziom ten determinuje (m.in. poprzez liczbę zasobów, przyjęte pojemności, itp.) struktury kolejnych poziomów. W przypadku poziomu procesów lokalnych (poziom  $SL$ ), marszruty procesów (ich przebieg) są ograniczone przez liczbę i rodzaj przyjętych zasobów. Te z kolei determinują sposób prowadzenia marszrut procesów multimodalnych poziomu  $SM^1$ . Jako, że marszruty procesów poziomu  $SM^1$  są konstruowane z marszrut procesów lokalnych  $SL$  ich postać jest z kolei ograniczona przez zbiór potencjalnych kombinacji fragmentów marszrut procesów lokalnych. Analogicznie struktura poziomu  $SM^1$  ogranicza strukturę poziomu  $SM^2$  i dalej struktura  $l$ -tego poziomu zachowań  $SM^l$  ogranicza strukturę poziomu  $SM^{l+1}$ .

Przykładowo, w systemie ze strukturą z rys. 2.7 niedopuszczalny jest proces multimodalny z marszrutą  $m^1 p_3 = (R_2, R_1, R_5, R_8)$ , gdyż marszruta taka nie pokrywa się z żadną kombinacją marszrut procesów lokalnych. Analogicznie raz określona struk-

tura poziomu  $SM^1$  stanowi podstawę dla struktury poziomu  $SM^2$ . W omawianej strukturze istnieje tylko jedna marszruta procesu z  $SM^2$  łącząca zasób  $R_1$  z  $R_8$  (marszruta procesu  $m^2P_1$  – przerywana linia) – procesy poziomu  $SM^1$  dopuszczają tylko taką kombinację.

W przedstawionym opisie niższe poziomy hierarchii zachowań determinują struktury poziomów wyższych (podejście bottom-up). W ogólności jednak można rozważać odwrotny kierunek, w którym struktura poziomu procesów multimodalnych narzuca potencjalne postacie procesów niższych poziomów (podejście top-down). Przykładowo przyjęcie na poziomie  $SM^1$  procesu z marszrutą  $m^1p_3 = (R_2, R_1, R_5, R_8)$  wymusza istnienie na poziomie  $SL$  procesów lokalnych gwarantujących dopuszczalność tej marszruty.

W przedstawionej koncepcji wielopoziomowego modelu struktury SWMPC dopuszcza się definiowanie dowolnej liczby poziomów procesów multimodalnych, opcjonalność mechanizmów synchronizacji procesów multimodalnych różnych poziomów, a także wprowadzanie arbitralnych reguł priorytetowania.

Szczegółowy podział struktur SWMPC odnosi się do wielkości opisujących: pojemność zasobów  $C(R_k)$ , czasów wykonania operacji  $t_{i,j} / m^l t_{i,j}$ , zapotrzebowanie operacji procesów na jednostki zasobów  $\omega_{i,j}^h / m^l \omega_{i,j}^h$ , uwzględnienie protokołu wzajemnego wykluczania. Pozwala to wyróżnić 16 rodzajów systemów współbieżnie realizowanych, multimodalnych procesów cyklicznych, których zestawienie przedstawiono w tab. 2.1.


Zgodnie z przyjętym podziałem, system opisany strukturą z rys. 2.7 jest systemem typu  $MW_{1,1}$  charakteryzującym się: jednostkową pojemnością zasobów ( $C(R_k) = 1$ ), jednostkowym zapotrzebowaniem operacji na zasoby ( $\omega_{i,j}^h = 1, m^l \omega_{i,j}^h = 1$ ) oraz jednostkowym czasem wykonania operacji ( $t_{i,j} = 1, m^l t_{i,j} = 1$ ). Ponadto, w systemie tym, procesy multimodalne synchronizowane są protokołem wzajemnego wykluczania.


Problemy harmonogramowania SWMPC przedstawionych w tab. 2.1 są rozszerzeniem klas  $WPP|ol, k_s, j_u, WPP|ol, k_s, j_g$  z rys. 2.5 o elementy związane z procesami multimodalnymi. Problemy odnoszące się do systemów ograniczonych przez zieloną linię można traktować jako należące do  $WPP|ol, k_s, j_u$  a problemy ograniczone przez linię niebieską jako należące do  $WPP|ol, k_s, j_g$ . Problemy związane z systemami  $MW_{i,3}$  (dla  $i = 1 \dots 4$ ) mają z kolei charakter mieszany – procesy lokalne korzystają z dowolnej liczby jednostek zasobów z kolei procesy multimodalne tylko z jednej.


Podsumowując, struktura definiowana jako (18) w zależności od przyjętych założeń może reprezentować jeden z 16 prezentowanych w tab. 2.1 rodzajów SWMPC. Jako przykład ilustrujący może posłużyć podsystem transportu międzystanowiskowego AGV elastycznego systemu produkcyjnego.


**Tabela 2.1.** Podział SWMPC ( $WPP|_{O_l, k_s, j_u}$ ,  $WPP|_{O_l, k_s, j_g}$ ) uwzględniający kryteria: pojemności zasobów  $C(R_k)$ , zapotrzebowania operacji procesów na zasoby  $\omega_{i,j}^h / m^l \omega_{i,j}^h$  oraz czasów wykonania operacji  $t_{i,j} / m^l t_{i,j}$

		Rozstrzygnięcie dostępu p. multimodalnych do zasobów		pojemności zasobów $R$			
				$C(R_k) = 1, \forall R_k \in R$		$C(R_k) > 1, \forall R_k \in R$	
				zapotrzebowanie operacji procesów lokalnych na zasoby			
				$\omega_{i,j}^h = 1$		$\omega_{i,j}^h \geq 1$	
				zapotrzebowanie operacji procesów multimodalnych na zasoby			
				$m^l \omega_{i,j}^h = 1$			$m^l \omega_{i,j}^h \geq 1$
czasy trwania operacji	jednostkowy (=1)	wzajemne wykluczanie	$MW_{1,1}$	$MW_{1,2}$	$MW_{1,3}$	$MW_{1,4}$	
	niejednostkowy (>1)	brak wzajemnego wykluczania	$MW_{2,1}$	$MW_{2,2}$	$MW_{2,3}$	$MW_{2,4}$	
		wzajemne wykluczanie	$MW_{3,1}$	$MW_{3,2}$	$MW_{3,3}$	$MW_{3,4}$	
		brak wzajemnego wykluczania	$MW_{4,1}$	$MW_{4,2}$	$MW_{4,3}$	$MW_{4,4}$	

 - systemy systemu z rys. 2.6

 - systemy typu  $WPP|_{O_l, k_s, j_u}$

 - systemy o zachowaniu opisywanym stanem (19), patrz podrozdział 2.2.2

 - systemy typu  $WPP|_{O_l, k_s, j_g}$

$MW_{i,j}$ - typ SWMPC, w którym:

$i$ -określa czasy trwania operacji procesów i rodzaj dostępu procesów multimodalnych do zasobów,

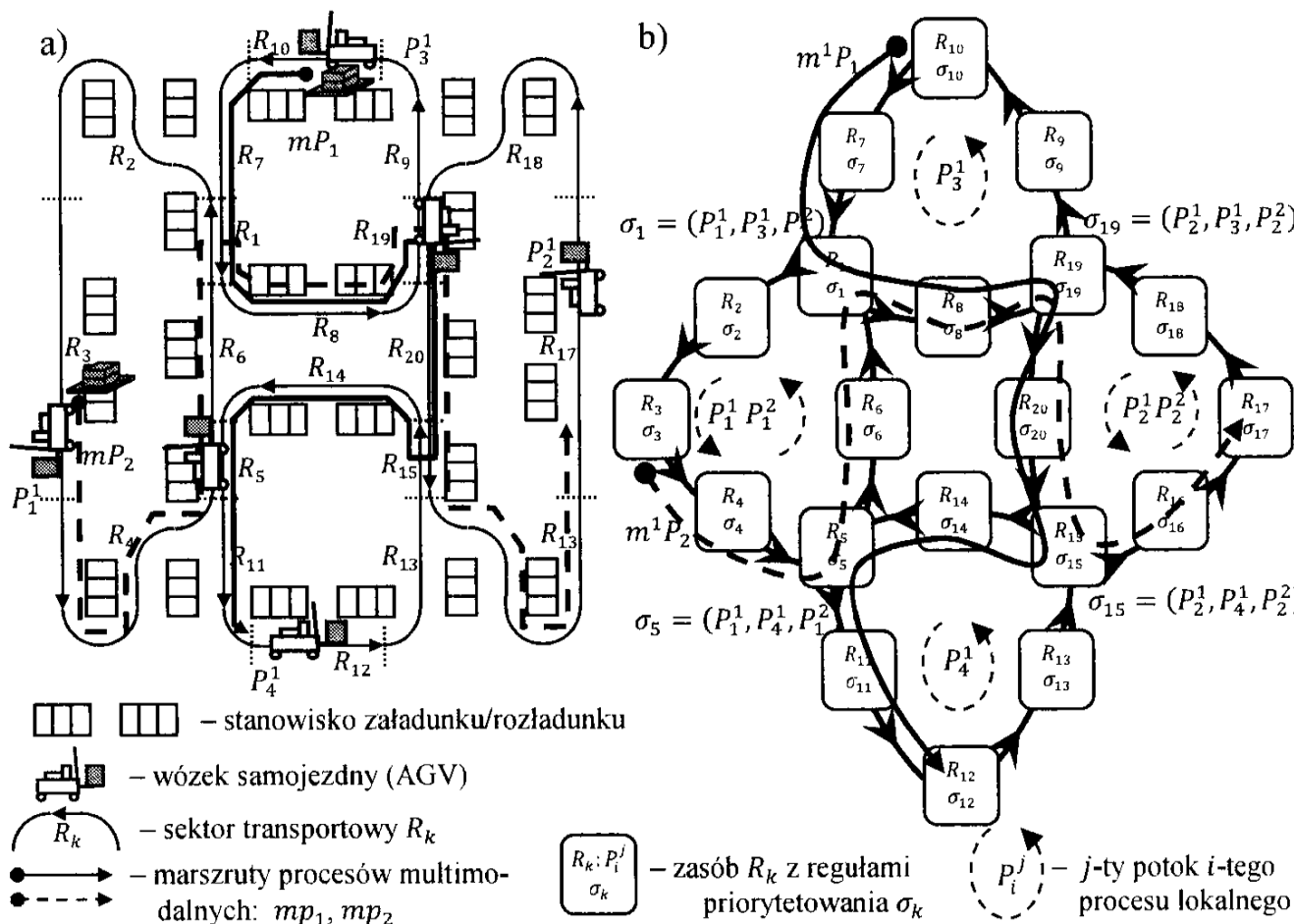
$j$ -określa pojemności zasobów oraz zapotrzebowanie operacji na zasoby



#### Przykład 4. Reprezentacja podsystemu transportowego AGV jako SWMPC.

Dany jest system AGV z rys. 2.8a, w którym wzdłuż zadanych tras porusza się 6 wózków samojezdnych, których zadaniem jest transport palet między stanowiskami. Na rysunku oznaczone zostały dwie marszruty transportu palet: pierwsza (linia czerwona) między stanowiskami  $R_{10}$  i  $R_{12}$  oraz druga między stanowiskami  $R_3$ ,  $R_{17}$  (linia niebieska). Przyjmuje się, że w danym momencie przy jednym stanowisku może przebywać tylko jeden wózek (wzajemne wykluczanie). Na stanowisku dopuszcza się jednoczesne rozmieszczenie kilku palet.

Strukturę rozważanego systemu klasy SWMPC z czterema procesami lokalnymi:  $P_1, P_2, P_3, P_4$  odpowiadającymi poruszającym się wózkom (procesy  $P_1$  i  $P_2$  są dwupotokowe:  $P_1 = \{P_1^1, P_1^2\}$ ,  $P_2 = \{P_2^1, P_2^2\}$ ) oraz dwoma procesami multimodalnymi  $m^1P_1, m^1P_2$  reprezentującymi trasy palet, ilustruje rys. 2.8a.



Rys. 2.8. Podsystem transportowy z 6 wózkami samojezdnymi a) jego model SWMPC b) (Bocewicz et. al 2014)

Struktura systemu (zgodnie z (18)) jest następująca:

$$SC^1 = ((R, SL), SM^1).$$

Zbiór  $R$  zawiera 20 zasobów  $R = \{R_1, \dots, R_{20}\}$  o pojemności  $C(R_k) = 1$ ,  $R_k \in R$ . Poziom procesów lokalnych  $SL$  charakteryzowany jest przez:

- zbiór procesów:

$$P = \{P_1, P_2, P_3, P_4\},$$

$$P_1 = \{P_1^1, P_1^2\}, P_2 = \{P_2^1, P_2^2\}, P_3 = \{P_3^1\}, P_4 = \{P_4^1\},$$

- zbiór marszrut:

$$U = \{p_1, p_2, p_3, p_4\},$$

$$p_1 = (R_1, R_2, R_3, R_4, R_5, R_6), p_2 = (R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}),$$

$$p_3 = (R_1, R_8, R_{19}, R_9, R_{10}, R_7), p_4 = (R_5, R_{11}, R_{12}, R_{13}, R_{15}, R_{14}),$$

- zbiór sekwencji operacji:

$$O = \{O_1^1, O_1^2, O_2^1, O_2^2, O_3^1, O_4^1\},$$

$$O_i^h = (o_{i,1}^h, \dots, o_{i,6}^h), i = 1 \dots 4, h = 1 \dots ls(i), \text{ gdzie: } ls(1) = ls(2) = 2,$$

$$ls(3) = ls(4) = 1,$$

Przyjęto, że zapotrzebowanie operacji  $o_{i,j}^h$  na zasoby jest w ilości  $\omega_{i,j}^h = 1$ ,

- zbiór sekwencji czasów wykonywania operacji:

$$T = \{T_1, T_2, T_3, T_4\},$$

przyjęto jednostkowe czasy wykonywania operacji  $T_i = (1,1,1,1,1,1)$ , dla  $i = 1 \dots 4$ ,

- zbiór reguł priorytetowania:

$$\theta^0 = \{\sigma_1^0, \dots, \sigma_{20}^0\},$$

reguły dla zasobów obsługujących tylko jeden proces mają postać:

$$\sigma_2^0 = \sigma_3^0 = \sigma_4^0 = \sigma_6^0 = (P_1^1, P_1^2), \sigma_{16}^0 = \sigma_{17}^0 = \sigma_{18}^0 = \sigma_{20}^0 = (P_2^1, P_2^2),$$

$$\sigma_8^0 = \sigma_9^0 = \sigma_{10}^0 = \sigma_7^0 = (P_3^1), \sigma_{11}^0 = \sigma_{12}^0 = \sigma_{13}^0 = \sigma_{14}^0 = (P_4^1),$$

reguły dla zasobów obsługujących dwa procesy:

$$\sigma_1^0 = (P_1^1, P_3^1, P_1^2), \sigma_5^0 = (P_1^1, P_4^1, P_1^2), \sigma_{15}^0 = (P_2^1, P_4^1, P_2^2), \sigma_{19}^0 = (P_2^1, P_3^1, P_2^2),$$

- sekwencja wzajemnej liczby wykonań:

$\Psi^0 = (1,1,1,1)$  – przyjęto, że każdy proces wykonuje się cyklicznie taką samą ilość razy względem innych procesów.

Dla poziomu procesów multimodalnych  $SM^1$  przyjęto, że możliwe jest jednoczesne korzystanie z tych samych zasobów. Oznacza to brak reguł priorytetowania  $\theta^1 = \emptyset$ . Poziom  $SM^1$  charakteryzowany jest przez:

- zbiór procesów:

$$m^1P = \{m^1P_1, m^1P_2\}, \text{ – procesy jednopotokowe,}$$

- zbiór marszrut:

$$m^1U = \{m^1p_1, m^1p_2\},$$

$$m^1p_1 = ((R_{10}, R_7, R_1, R_8, R_{19}) \wedge (R_{20}, R_{15}) \wedge (R_{14}, R_5, R_{11}, R_{12})) =$$

$$\begin{aligned}
 &= (R_{10}, R_7, R_1, R_8, R_{19}, R_{20}, R_{15}, R_{14}, R_5, R_{11}, R_{12}), \\
 m^1 p_2 &= ((R_3, R_4, R_5, R_6, R_1) \wedge (R_8, R_{19}) \wedge (R_{20}, R_{15}, R_{16}, R_{17})) = \\
 &= (R_3, R_4, R_5, R_6, R_1, R_8, R_{19}, R_{20}, R_{15}, R_{16}, R_{17}),
 \end{aligned}$$

- zbiór sekwencji operacji:

$$m^1 O = \{m^1 O_1, m^1 O_2\}$$

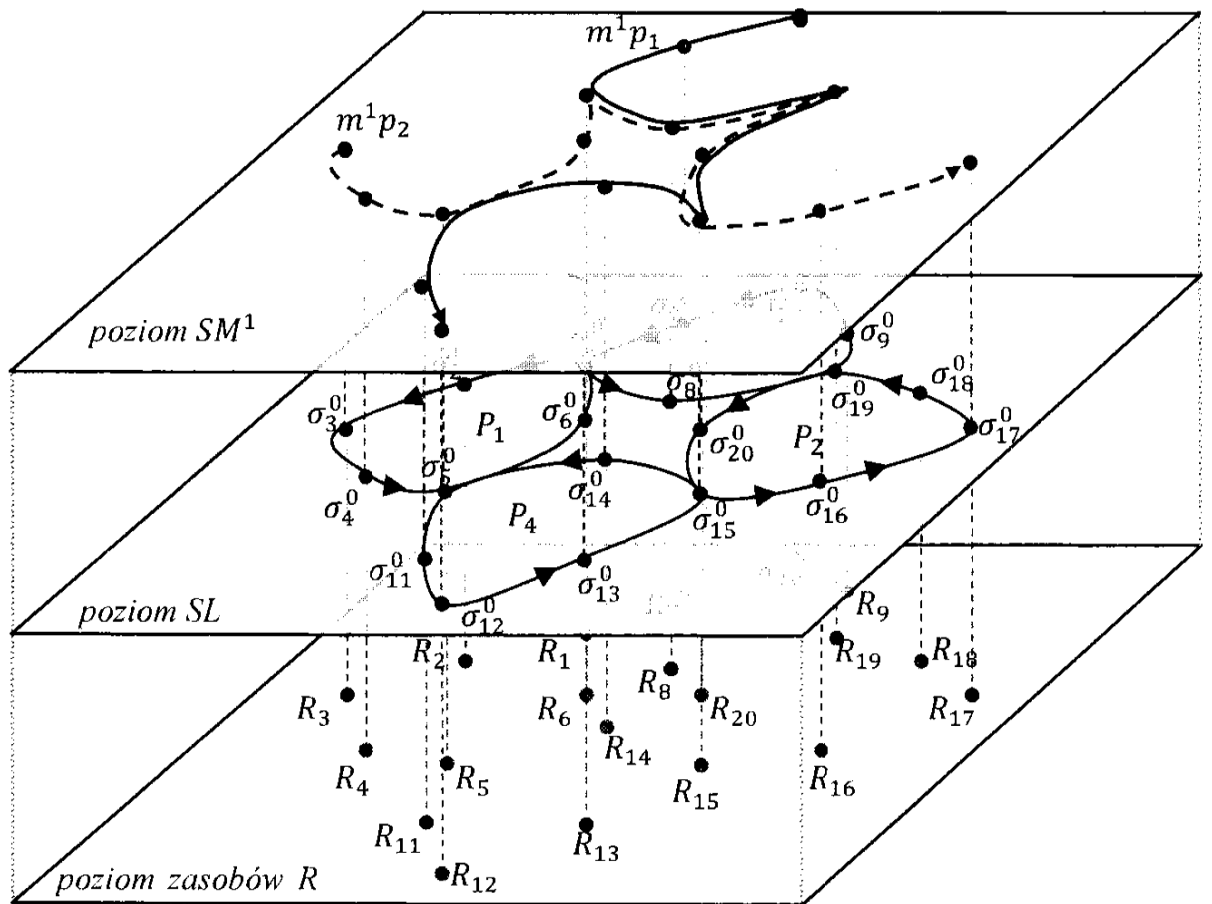
$$m^1 O_1 = (m^1 o_{1,1}, m^1 o_{1,2}, \dots, m^1 o_{1,11}),$$

$$m^1 O_2 = (m^1 o_{2,1}, m^1 o_{2,2}, \dots, m^1 o_{2,11}),$$

Przyjęto, że zapotrzebowanie operacji  $m^1 o_{i,j}$  na zasoby jest takie same dla wszystkich procesów i wynosi  $m^1 \omega_{i,j} = 1$ , a zapotrzebowanie na procesy lokalne jest następujące ( $m^1 \mu_i = (m^1 \mu_{i,1}, \dots, m^1 \mu_{i,j}, \dots, m^1 \mu_{i,lm(i,l)})$ ):

$$m^1 \mu_1 = (P_3^1, P_3^1, P_3^1, P_3^1, P_3^1, P_2^2, P_2^2, P_4^1, P_4^1, P_4^1, P_4^1),$$

$$m^1 \mu_2 = (P_1^1, P_1^1, P_1^1, P_1^1, P_1^1, P_3^1, P_3^1, P_2^2, P_2^2, P_2^2, P_2^2),$$



Rys. 2.9. Graficzna ilustracja struktury (18) dla SWMPC z rys. 2.8

- zbiór sekwencji czasów wykonywania operacji:

$$m^1 T = \{m^1 T_1, m^1 T_2\},$$

Analogicznie jak w procesach lokalnych przyjęto jednostkowe czasy wykonywania operacji  $m^1 T_1 = m^1 T_2 = (1, 1, 1, 1, 1, 1)$ ,

- sekwencja wzajemnej liczby wykonań:

$\Psi^1 = (1,1)$  – przyjęto, że jeden proces wykonuje się cyklicznie taką samą ilość razy względem drugiego procesu.

Graficzną ilustrację struktury przedstawiono na rys. 2.9. Jak widać marszruty procesów  $m^1P_1$ ,  $m^1P_2$  składają się odpowiednio z fragmentów marszrut  $P_3, P_2, P_4$  oraz  $P_1, P_3, P_2$ . System o takiej strukturze jest typu  $MW_{2,1}$  (patrz tab. 2.1), to znaczy, że:

- czasy wykonania operacji (zarówno procesów lokalnych jak i multimodalnych) są jednostkowe:  $t_{i,j} = 1, m^l t_{i,j} = 1$ ,
- w przypadku procesów multimodalnych protokół wzajemnego wykluczania nie obowiązuje,
- pojemności zasobów są jednostkowe:  $C(R_k) = 1$ ,
- zapotrzebowanie procesów lokalnych i multimodalnych na wykorzystywane przez siebie zasoby jest jednostkowe:  $\omega_{i,j}^h = 1, m^l \omega_{i,j}^h = 1$ .

■

Dalsze rozważania prowadzone są dla systemów klasy SWMPC spełniających wyżej wymienione założenia.

## 2.2. Zachowanie SWMPC

### 2.2.1. Przestrzeń stanów

Zachowanie systemu jest zwykle definiowane jako pewne celowe, skoordynowane działanie w określonym otoczeniu i w określonym czasie. Jest to potocznie przyjęta definicja wykorzystywana głównie w psychologii, w nurcie behawioryzmu (Nęcka et al. 2012). W naukach technicznych pojęcie „zachowanie systemu” definiowane bardziej precyzyjnie, zwykle w kontekście obszaru prowadzonych badań.

Przykładowo, w automatyce przez zachowanie rozumie się zmieniającą się w czasie wartość zmiennych opisujących obiekt. Bardzo często zachowanie utożsamiane jest z przebiegami czasowymi (funkcjami) reprezentującymi sygnały na wyjściu (zmiennie wyjściowe) obiektu, będące konsekwencją pewnego pobudzenia. Przykładowy przebieg sygnału wyjściowego jest w tym przypadku tylko przejawem zachowania, inaczej mówiąc pewnym działaniem dopuszczalnym w rozważanym obiekcie. Samo zachowanie należy rozumieć jako zbiór potencjalnych działań, które mogą zostać podjęte przez ten obiekt.

W systemach współbieżnych procesów cyklicznych, zachowanie rozumiane jest podobnie. Składa się na nie zbiór potencjalnych przebiegów, tj. ciągów występujących po sobie stanów systemu.

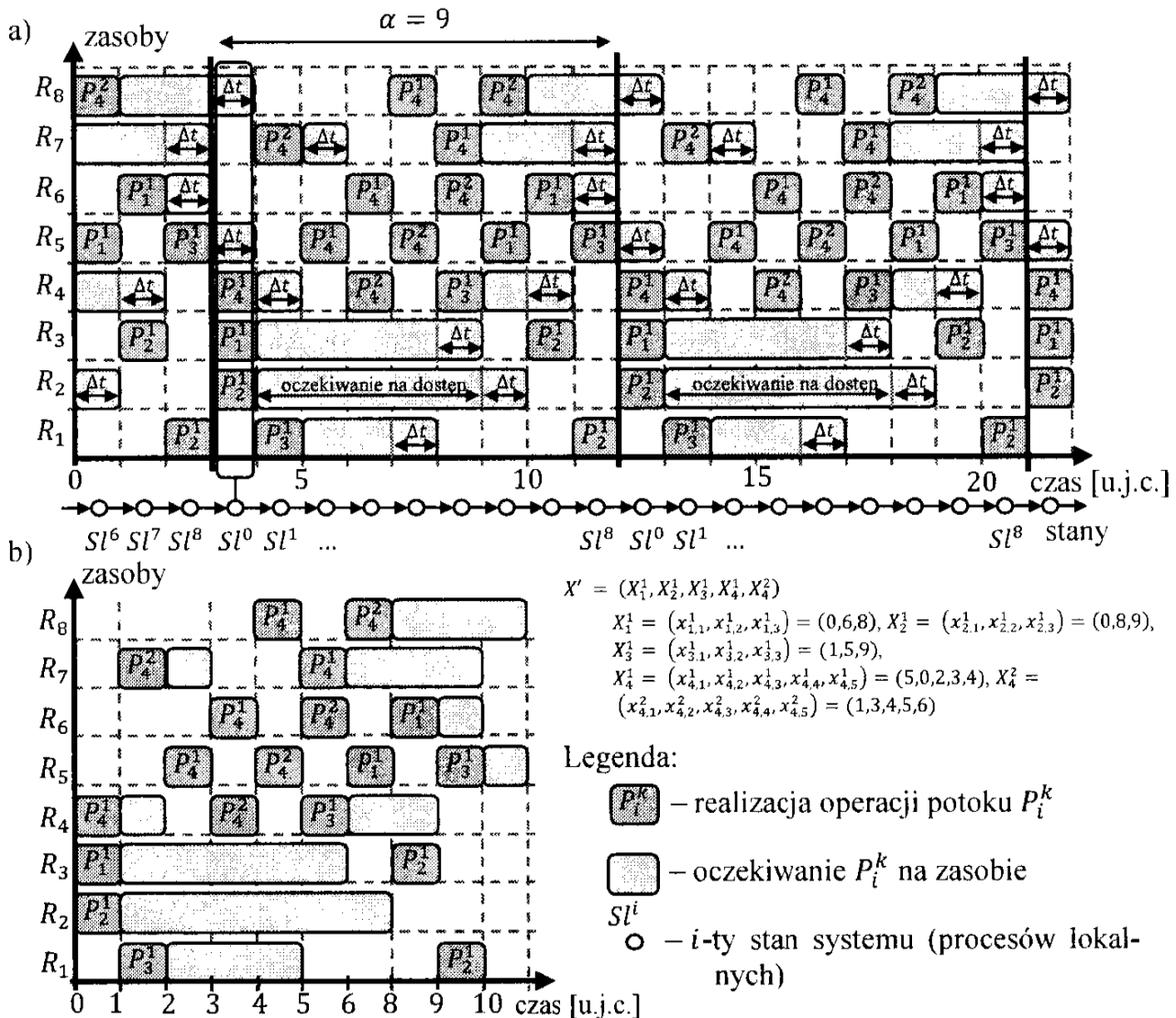
W tym miejscu należy zwrócić uwagę na zdefiniowane, w podrozdziale 1.2, pojęcie procesu rozumianego jako uporządkowany w czasie ciąg stanów. Tak rozumiany stan opisuje jednak to co się dzieje w ramach jednego procesu a nie całego systemu, ma więc tylko charakter lokalny (nie jest brana pod uwagę realizacja pozostałych procesów systemu). Ponadto w przyjętej interpretacji procesy określają co (jakie operacje), w jakiej kolejności oraz przy użyciu jakich zasobów i procesów (w przypadku multimodalności) należy wykonać. Nie określają one natomiast kiedy poszczególne

operacje zostaną wykonane. To jak przebiegać będzie realizacja procesów uzależnione jest od przyjętych reguł priorytetowania, pojemności zasobów, itp.

W tym ujęciu przez **przebieg SWMPC** rozumiany jest ciąg (sekwencja) występujących w czasie stanów, opisujących aktualną realizację operacji każdego z procesów oraz aktualne wykorzystanie wszystkich zasobów systemu (tzn. ich rezerwacje, a także przydział do operacji, prawa dostępu, itp.). Zbiór tego typu przebiegów stanowi o **zachowaniu SWMPC**.

We wszystkich omawianych do tej pory przykładach rozważane były wyłącznie zachowania reprezentowane przez cykliczny harmonogram  $X$ , definiowany jako zbiór momentów rozpoczęcia realizowanych w systemie operacji (w zależności od rodzaju problemu harmonogram  $X$  definiowano jako (2), (10), (11) lub (15)). Harmonogram cykliczny może być traktowany jako szczególna reprezentacja cyklicznego przebiegu ustalonego.

Przykładem takiej reprezentacji jest harmonogram z rys. 2.10 przedstawiający realizację potoków  $P_1^1, P_2^1, P_3^1, P_4^1, P_4^2$  procesów lokalnych systemu z rys. 2.6. W rozważanym przypadku przyjęto, jednostkowy czas wykonania każdej operacji tzn.:  $t_{i,j}^k = 1$ , oraz jednostkowy czas  $\Delta t = 1$  określający zwłokę związaną z reakcją potoku na sygnał zwolnienia żadanego zasobu.



Rys. 2.10. Cykliczny harmonogram realizacji potoków procesów lokalnych systemu z rys. 2.6 a), fragment harmonogramu odpowiadający sekwencji  $X'$  (11) b)

Przez **stan systemu**, wchodzącego w skład ustalonego przebiegu cyklicznego, rozumiana jest bieżąca alokacja potoków na zasobach. Oznacza to, że stan zawiera informację o tym gdzie (tzn. na jakich zasobach) w danym momencie przebywają poszczególne potoki. Przykładowo, na rys. 2.10, stany odpowiadają kolumnom harmonogramu reprezentującym przedziały czasowe o długości równej jednej jednostce czasu. Ze względu na przyjęte jednostkowe czasy wykonania operacji, w każdej kolejnej jednostce czasu dochodzi do zmiany alokacji potoków, tzn. zmiany stanu. Stany oznaczane są symbolicznie:  $Sl^i$  ( $i$  – jest numerem stanu) oraz graficznie:  $\circ$ . Opisany w ten sposób diagram harmonogramu ilustruje zatem zarówno realizacje operacji w czasie, jak i kolejno osiągalne stany systemu. Oznacza to, że na podstawie dostępnego harmonogramu  $X$  można również odtworzyć stany systemu. Własność ta jest wykorzystywana w rozdziale trzecim do wyznaczania cyklicznych przebiegów ustalonych systemu.

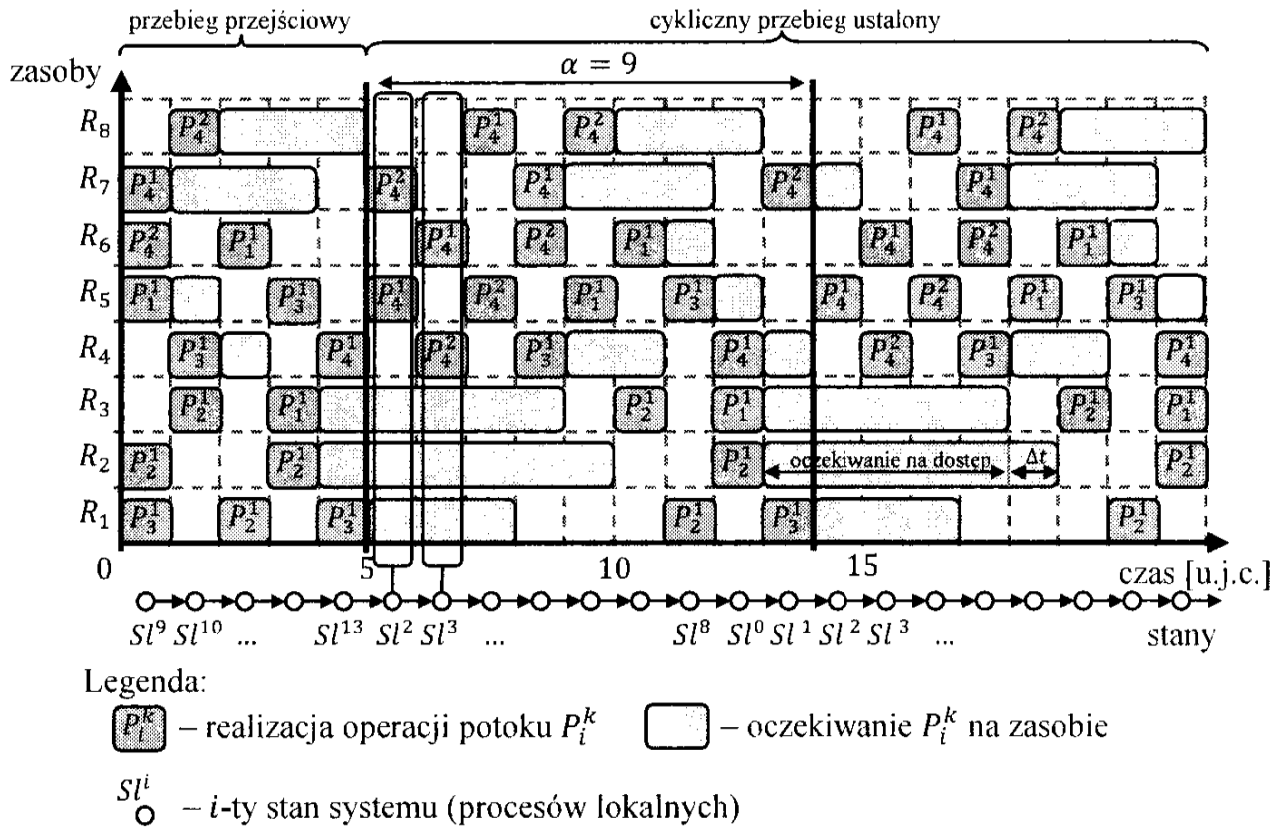
Utożsamienie alokacji potoków ze stanem jest najprostszą formą jego definicji, w bardziej złożonych sytuacjach, np. związanych z wielokrotnym występowaniem potoków w regułach priorytetowania (np.  $\sigma_1^0 = (P_3^1, P_2^1, P_3^1, P_3^1, P_3^1)$ ) prowadzi jednak do niejednoznaczności związanych z rozróżnianiem stanów (np. w sytuacjach zachodzenia tych samych alokacji ale w różnych momentach czasu). Z tego względu do opisu stanu wykorzystuje się również inne wielkości takie jak semaforey oraz indeksy, które szczegółowo opisane są w kolejnym podrozdziale.

Przyglądając się kolejnym stanom (kolumnom harmonogramu z rys. 2.10) można dostrzec, że brak jest stanu, w którym wszystkie potoki jednocześnie rozpoczynająby swoje operacje. Przykładowo w 4-tej jednostce czasu swoje operacje rozpoczynają potoki  $P_1^1, P_2^1, P_4^1$ , a pozostałe potoki  $P_3^1, P_4^2$  przebywają na zasobach w oczekiwaniu związanym ze zwłoką  $\Delta t$ . Podobnie sytuacja ma miejsce w pozostałych stanach.

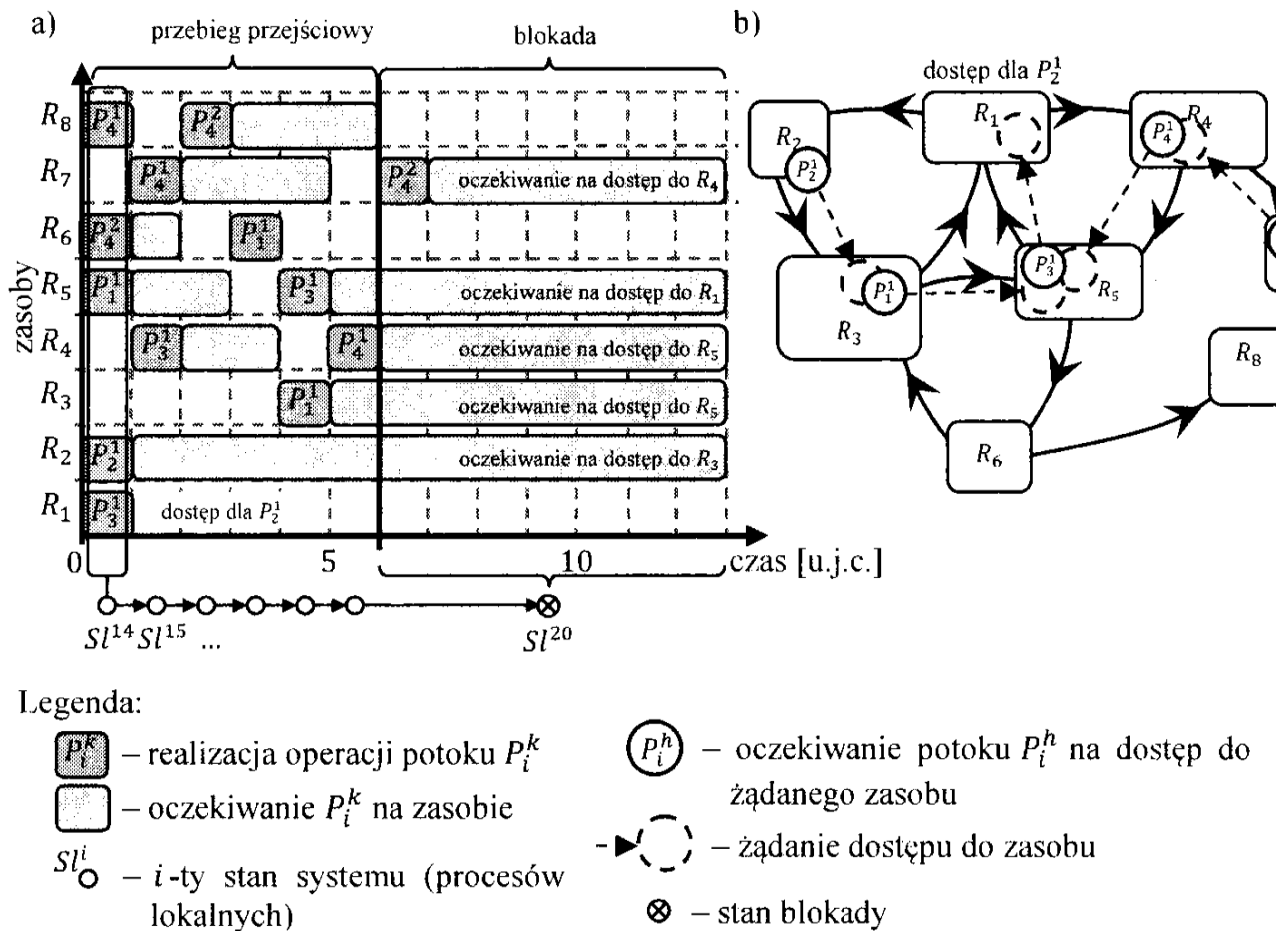
Powstaje więc pytanie dotyczące sytuacji, w których wymagane jest aby w stanie startowym inicjowana była jednoczesna realizacja wszystkich potoków? Tego typu założenia występują w systemach produkcyjnych, w których zakłada się jednoczesne uruchomienie wielu procesów/linii produkcyjnych lub, w których mechanizm sterowania nie dopuszcza do wprowadzenia zwłoki w uruchomieniu procesów produkcyjnych. W rozważanym przebiegu żaden ze stanów nie spełnia tego założenia. Czy istnieje zatem stan początkowy spełniający wyżej wymienione oczekiwania, tzn. stan z którego osiągalny jest ustalony przebieg cykliczny z rys. 2.10?

Rys. 2.11 przedstawia sytuację, w której ze stanu  $Sl^9$  charakteryzującego się tym, że wszystkie potoki rozpoczynają swoją operację (przyjęto, że każdy potok może rozpoczynać się z dowolnego zasobu związanej z nim marszruty) można osiągnąć cykliczny przebieg (przebieg z rys. 2.10). Zgodnie z harmonogramem z rys. 2.11 potoki rozmieszczone na zasobach  $R_1, R_2, R_5, R_6, R_7$  wchodzą w skład przebiegu przejściowego, który po 5 jednostkach czasu przechodzi w przebieg cykliczny.

Oczywiście przebieg z rys. 2.11 nie jest jedynym, który skutkuje cykliczną realizacją procesów. W ogólności przebiegów przejściowych może być bardzo wiele (w zależności od stanu początkowego systemu) choć nie zawsze muszą one prowadzić do cykliczności. Przykładowo, rys. 2.12 przedstawia przebieg realizacji procesów prowadzący do ich blokady.



Rys. 2.11. Przebieg przejściowy prowadzący do cyklu



Rys. 2.12. Przebieg prowadzący do blokady a), żądania zasobowe w stanie blokady b)

Obserwowana blokada jest przykładem źle rozstrzygniętego konfliktu zasobowego i może mieć miejsce w systemach wykorzystujących statyczne reguły priorytetowania  $\theta$  (reguły, w których raz ustalona kolejność dostępu potoków do zasobu nie ulega zmianie w trakcie działania systemu). Przyczyną zablokowania systemu jest w tym przypadku brak dostępu niektórych potoków do żądanych zasobów (rys. 2.12b). Realizacja potoków  $P_4^1, P_1^1$ , jest wstrzymana w wyniku braku dostępu do  $R_5$ , potok  $P_3^1$  wykorzystujący ten zasób nie może rozpocząć kolejnej operacji (na  $R_1$ ), gdyż prawo dostępu do zasobu  $R_1$  jest określone dla potoku  $P_2^1$ . Potok  $P_2^1$  nie może również przejść na zasób  $R_1$ , gdyż na drodze (na  $R_3$ ) stoi mu potok  $P_1^1$ . Tak więc wstrzymanie  $P_2^1$  uniemożliwia jego realizację na  $R_1$  i tym samym zmianę kolejności dostępu do tego zasobu (dostęp odbywa się zgodnie z regułą  $\sigma_1^0 = (P_3^1, P_2^1)$ ).

Warto zaznaczyć, że ten typ blokady nie występuje w systemach z regułami typu: FIFO, LIFO, itp., w których kolejność dostępu potoków do zasobów jest zależna od aktualnego stanu systemu. W systemach tych blokada powstaje wyłącznie jako skutek zamkniętego łańcucha żądań zasobowych (rys. 1.19). W blokadzie z rys. 2.12 łańcuch taki nie występuje. Oznacza to, że liczba potencjalnych blokad w systemach ze statycznymi regułami  $\theta$  jest większa niż w systemach z regułami zależnymi od stanu systemu.

Niech **procesy sąsiadujące** to takie, których potoki wykorzystują wspólne zasoby. Przykładowo w systemie z rys. 2.6 proces  $P_2$  sąsiaduje z  $P_1$  i  $P_3$ , ale już nie z  $P_4$ , (marszruty procesów  $P_2$  i  $P_4$  nie zawierają wspólnych zasobów). W systemach ze statycznymi regułami priorytetowania zachodzi następujący lemat:

#### Lemat 1

*Wstrzymanie realizacji procesu  $P_i$  w systemie o strukturze  $SC^{lp}$  (18), prowadzi do wstrzymania realizacji wszystkich procesów z nim sąsiadujących.*

Dowód lematu wynika z przyjętej postaci reguł prioryteowania  $\theta$ . Zakłada się (patrz założenie z podrozdziału 2.1.1), że potok procesu  $P_i$  występuje co najmniej raz w każdej regule związanej z zasobami marszruty tego procesu. Wstrzymanie realizacji operacji potoku procesu  $P_i$  oznacza więc wstrzymanie dostępu (realizacji) tych potoków, które występują w regułach prioryteowania razem z  $P_i$  – czyli potoków procesów sąsiadujących.

□

Powyższy lemat prowadzi do następującego twierdzenia:

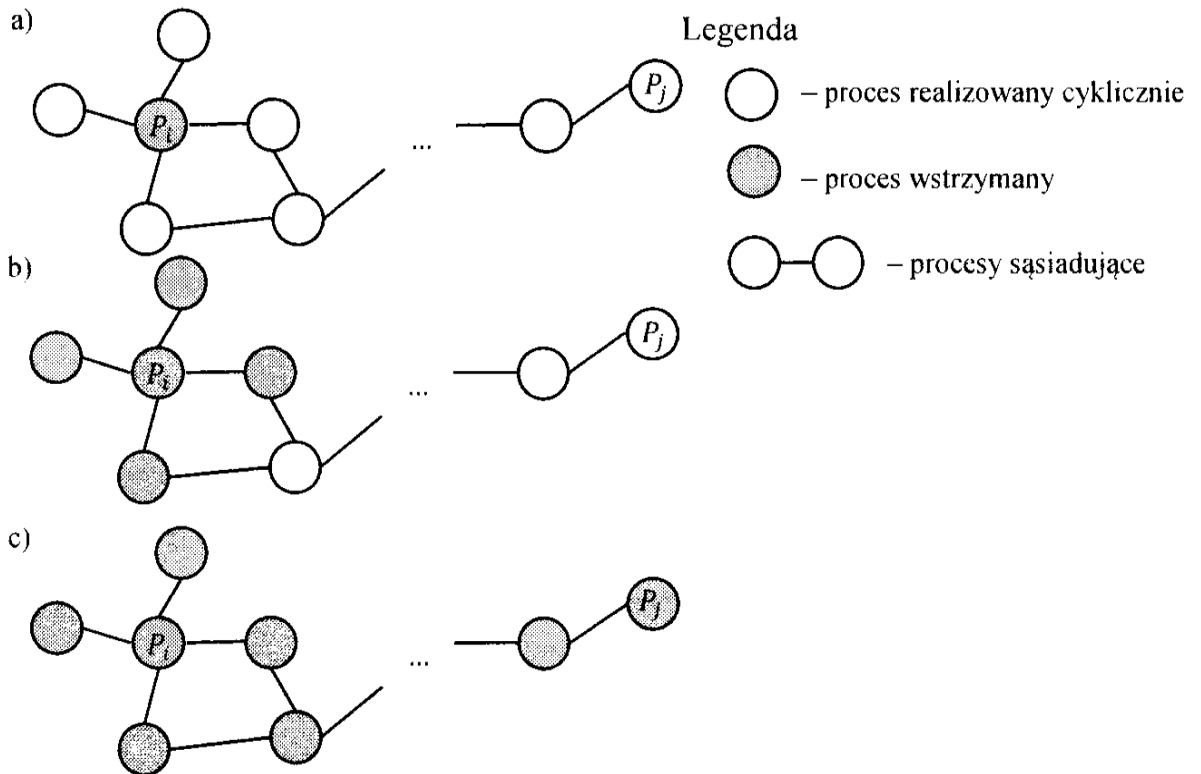
#### Twierdzenie 1

*Wstrzymanie realizacji co najmniej jednego procesu  $P_i$  w systemie o strukturze  $SC^{lp}$  (18) skutkuje zatrzymaniem realizacji operacji wszystkich procesów lokalnych systemu.*

Dowód: Do wykazania twierdzenia wykorzystany zostanie Lemat 1. Niech dopuszczalna będzie sytuacja sprzeczna z Twierdzeniem 1, w której pewien proces  $P_i$  systemu o strukturze  $SC^{lp}$  został wstrzymany (np. wyniku wystąpienia blokady) a pewien inny proces  $P_j$  tego systemu nadal realizuje swoje operacje cyklicznie. Sytuację tę przedstawia rys. 2.13a.



Zgodnie z Lematem 1 wstrzymanie realizacji  $P_i$  prowadzi do wstrzymania realizacji wszystkich procesów sąsiadujących z  $P_i$  (rys. 2.13b). Łatwo widać, że wstrzymanie tych procesów prowadzi do wstrzymania kolejnych sąsiadów aż w konsekwencji do wstrzymania samego procesu  $P_j$  (rys. 2.13c). Niemożliwa jest zatem cykliczna realizacja procesu  $P_j$  przy jednoczesnym wstrzymaniu procesu  $P_i$ . Ze względu na to, że struktura procesów lokalnych jest spójna (założenie struktury (18)), wstrzymanie dowolnego procesu prowadzi zawsze do zatrzymania wszystkich pozostałych rys. 2.13c – co kończy dowód Twierdzenia 1.



Rys. 2.13. Ilustracja wstrzymywania procesów sąsiadujących

□

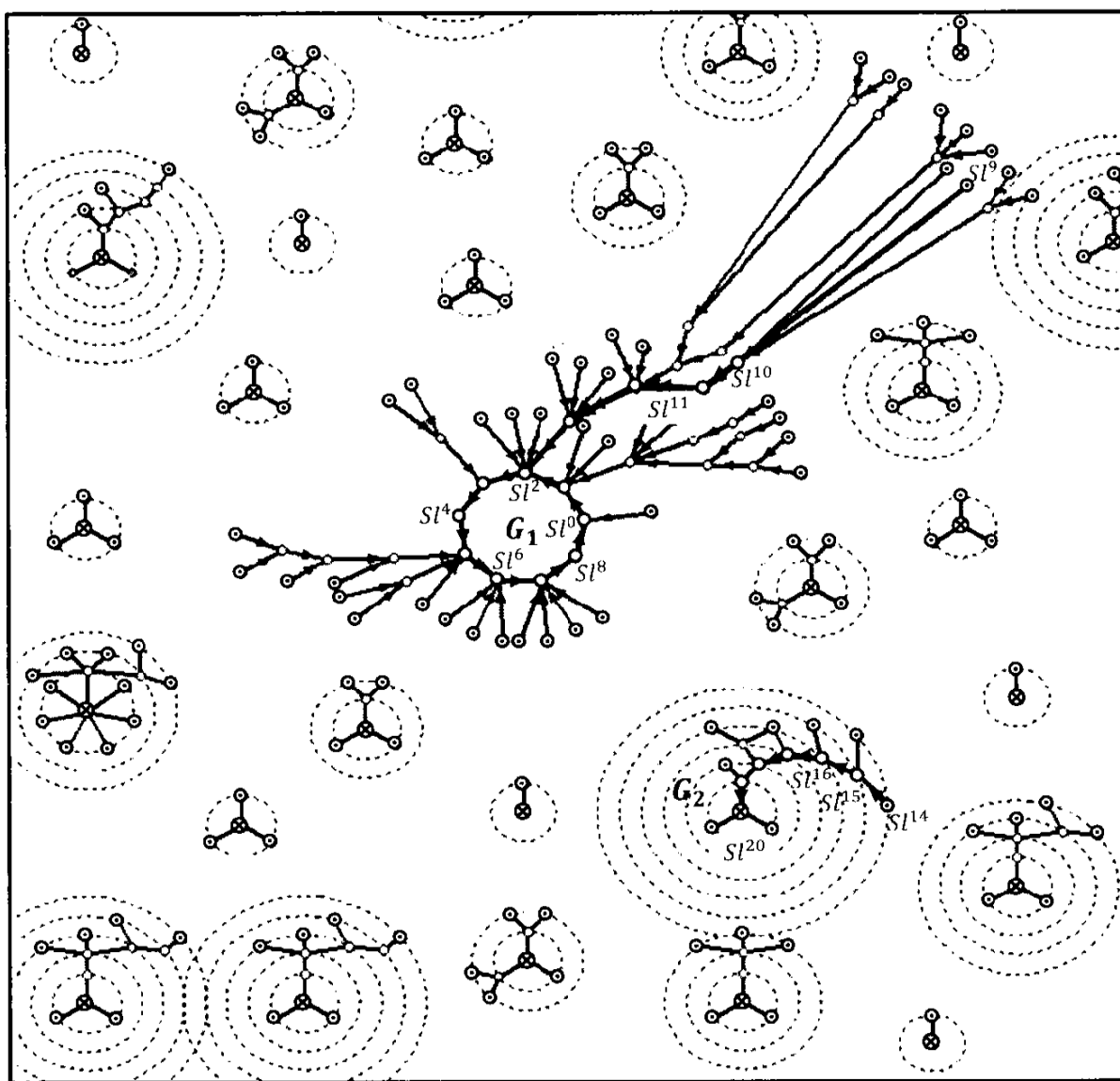
Wnioskiem praktycznym wynikającym z Twierdzenia 1 jest to, że jeżeli w systemie dojdzie do wstrzymania (w wyniku blokady) pewnych procesów, to wstrzymane zostaną wszystkie procesy realizowane w systemie. Oznacza to jednocześnie brak możliwości występowania w SWMPC częściowych cykliczności, tzn. sytuacji w której w blokadzie bierze udział część procesów (rys. 1.20). Innymi słowy można powiedzieć że, w SWMPC dopuszczalne są tylko dwa wykluczające się rodzaje **ustalonych** przejawów zachowania: **blokada systemu** (wszystkie procesy są wstrzymane) i **cykliczny przebieg ustalony** (wszystkie procesy są realizowane cyklicznie).

Odrębną grupę przejawów zachowań stanowią **przebiegi przejściowe** prowadzące do blokad lub cyklicznych przebiegów ustalonych. Przebiegi tego typu występujące na przykład przy rozruchu i wygaszaniu produkcji są zwykle pomijane w problemach harmonogramowania cyklicznego (Kampmeyer 2006; Levner et al. 2010).

Przyjęcie założenia, że w ramach oceny zachowania systemu interesujące są tylko przebiegi cykliczne może prowadzić do pewnych uproszczeń. Przykładowo zachowanie systemu z rys. 2.6 obejmuje 1374 osiągalne stany, które wchodzą w skład 908 przebiegów z czego tylko 1 jest przebiegiem cyklicznym (przebieg przedstawiony na

rys. 2.10). Na 907 przebiegów przejściowych 43 przebiegi prowadzą do przebiegu cyklicznego reszta z nich prowadzi do stanów blokadowych (jednym z tych przebiegów jest przebieg z rys. 2.12). Pominięcie w ocenie zachowania wszystkich przebiegów przejściowych powoduje usunięcie z rozważań ponad 99,8% potencjalnych instancji zachowania. O ile przebiegi prowadzące do blokady mogą rzeczywiście być nieistotne (95% możliwych przebiegów) o tyle znajomość 43 przebiegów przejściowych prowadzących do ustalonego przebiegu cyklicznego może mieć już praktyczne znaczenie.

Przykładowo świadomość istnienia przebiegów przejściowych jest szczególnie ważna w przypadku konieczności zmiany działania systemu. Rzadko zdarza się by przejście systemu z jednego przebiegu cyklicznego do drugiego (np. w wyniku zmiany planów produkcyjnych) odbywało się natychmiastowo bez przebiegu przejściowego.



Legenda:

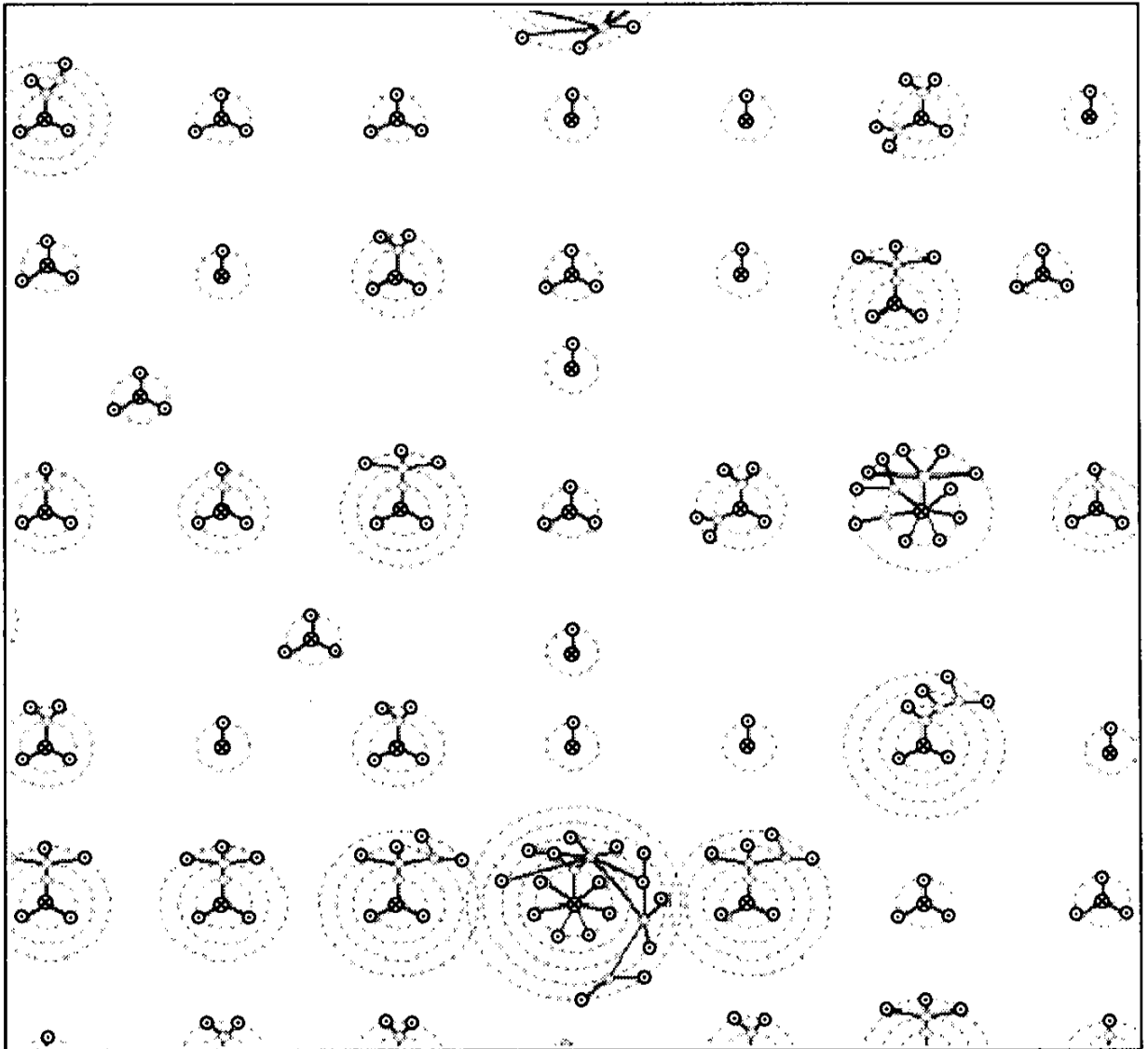
- |   |                     |         |  |
|---|---------------------|---------|--|
| ⊗ | – stan blokadowy    | ○→○→○→○ | – przebieg cykliczny z rys. 2.10                         |
| ⊙ | – stan początkowy   | ○→○→○→○ | – przebieg przejściowy prowadzący do blokady z rys. 2.12 |
| ○ | – stan dopuszczalny | ○→○→○→○ | – przebieg przejściowy prowadzący do cyklu z rys. 2.11   |

Rys. 2.14. Niespójny digraf jako przestrzeń stanów  $\mathcal{P}$  systemu z rys. 2.6

Podobnie jakakolwiek przypadkowa zmiana (związana z zakłóceniem, awarią, wypadkiem, zmianą parametrów struktury, itp.) powodująca „wybicie” systemu z jego przebiegu cyklicznego wymaga uruchomienia odpowiednich mechanizmów sterowania pozwalających na ponowny powrót do żądanego przebiegu.

Na uwagę zasługuje również fakt, że rozważany system charakteryzuje się tylko jednym przebiegiem cyklicznym (oczywiście cały czas mowa o zachowaniu na poziomie potoków procesów lokalnych). Niewielka liczba przebiegów cyklicznych jest charakterystyczna dla rozważanej klasy SWMPC, jest ona również przejawem diofantycznego charakteru rozważanych problemów harmonogramowania cyklicznego.

Zachowanie systemu traktowane jako zbiór przebiegów można zilustrować w postaci graficznej jako **przestrzeń stanów**  $\mathcal{P}$ . Przykład ilustrujący tę możliwość dla systemu z rys. 2.6, przedstawia rys. 2.14.



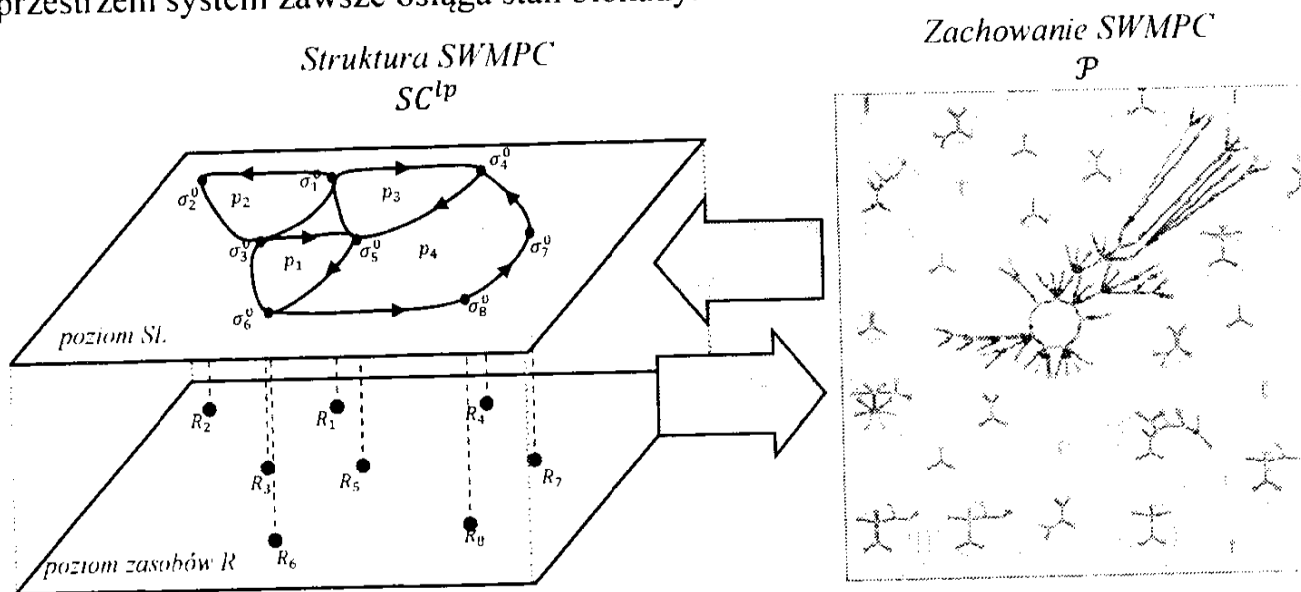
Legenda:

● – stan blokadowy      ○ – stan początkowy      ○ – stan systemu

**Rys. 2.15.** Przestrzeń stanów  $\mathcal{P}$  systemu z rys. 2.6, dla reguł:  $\sigma_1^0 = (P_2^1, P_3^1)$ ,  $\sigma_2^0 = (P_3^1)$ ,  $\sigma_3^0 = (P_2^1, P_1^1)$ ,  $\sigma_4^0 = (P_4^1, P_3^1, P_4^2)$ ,  $\sigma_5^0 = (P_4^1, P_3^1, P_1^1, P_4^2)$ ,  $\sigma_6^0 = (P_4^1, P_1^1, P_4^2)$ ,  $\sigma_7^0 = (P_4^1, P_4^2)$ ,  $\sigma_8^0 = (P_4^1, P_4^2)$

W kontekście proponowanej, graficznej reprezentacji przestrzeni stanów SWMPC jest niespójnym digrafem składającym się ze spójnych podgrafów modelujących przebiegi potencjalnych procesów zachodzących w systemie. Wierzchołkami digrafów są stany systemu, oznaczane w zależności od ich rodzaju symbolami:  $\circ\circ\circ$ . Łuki łączą ze sobą te pary stanów, w których następnik jest osiągnięty z poprzednika pary. Konsekwencją Twierdzenia 1 jest to, że wierzchołki odpowiadające stanom blokadowym ( $\otimes$ ) nie mają łuków wychodzących. Przebiegi SWMPC są reprezentowane przez podgrafy tworzące ścieżki. Na rys. 2.14. kolorami oznaczone zostały trzy przebiegi: przebieg niebieski  $D_1 = (Sl^0, Sl^1, \dots, Sl^8)$  odpowiadający cyklicznej realizacji z rys. 2.10 (zamknięta ścieżka), zielony  $D_2 = (Sl^9, Sl^{10}, \dots, Sl^{13})$  opowiadający przebiegowi przejściowemu prowadzącemu do przebiegu  $D_1$  oraz przebieg czerwony  $D_3 = (Sl^{14}, Sl^{15}, \dots, Sl^{20})$  prowadzący do blokady. Przebiegi te są ścieżkami w digrafach  $G_1, G_2$ , tzw. digrafach zachowań.

Przedstawiona przestrzeń stanów  $\mathcal{P}$  opisuje zachowanie SWMPC. Należy zaznaczyć, że otrzymane przebiegi reprezentują zachowanie determinowane przez strukturę  $SC^{lp}$  rozważanego systemu. Jakkolwiek zmiana w parametrach struktury (reguł priorytetowania, pojemności zasobów, itp.) niesie za sobą zmianę w przestrzeni  $\mathcal{P}$ . Przykładowo zmiana reguł priorytetowania, zasobów  $R_4, R_5, R_6$ , systemu z rys. 2.6 na:  $\sigma_4^0 = (P_4^1, P_3^1, P_4^2)$ ,  $\sigma_5^0 = (P_4^1, P_3^1, P_1^1, P_4^2)$ ,  $\sigma_6^0 = (P_4^1, P_1^1, P_4^2)$ , implikuje brak przebiegów cyklicznych – rys. 2.15. Innymi słowy oznacza to, że startując z dowolnego stanu przestrzeni system zawsze osiąga stan blokady.

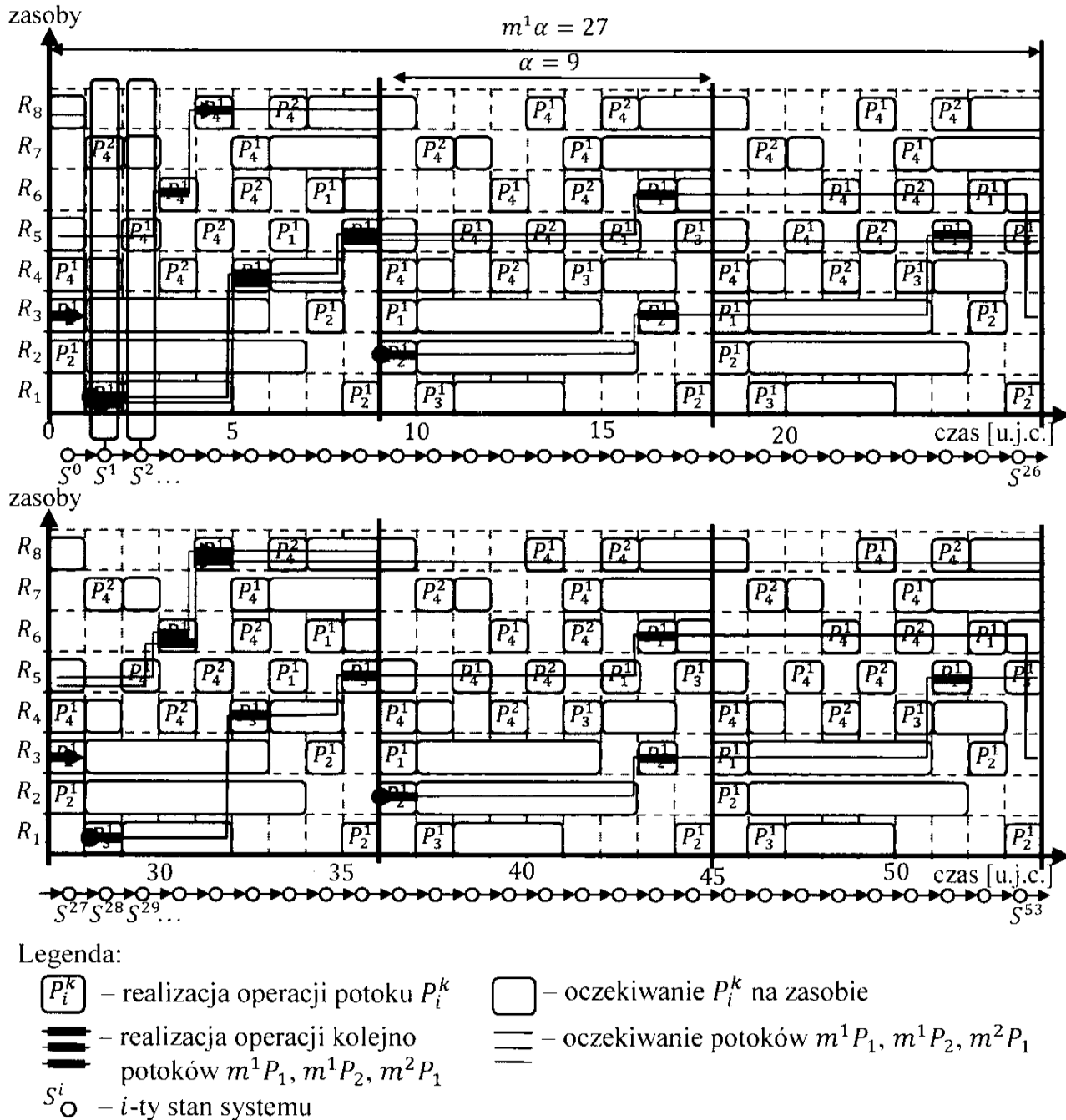


Rys. 2.16. Relacja struktura – zachowanie

Związek struktura – zachowanie ma charakter relacyjny (rys. 2.16), związek, w którym zachowanie jest konsekwencją przyjętej struktury i odwrotnie struktura determinuje potencjalne formy zachowania. W większości przypadków zakłada się nadrzędny charakter struktury, a więc przyjmuje się, że przestrzeń stanów SWMPC (zachowanie systemu) jest funkcją jego struktury  $\mathcal{P} = Y(SC^{lp})$ , struktury scharakteryzowanej odpowiednią postacią stanów systemu, warunkami określającymi ich dopuszczalność oraz funkcją przejścia między stanami.

Rozważane dotychczas przebiegi systemu z rys. 2.6 ograniczały się tylko do realizacji potoków procesów lokalnych.

Zachowanie uwzględniające występowanie procesów multimodalnych należących do  $l$ -tego poziomu jest definiowane analogicznie. Przykład przebiegu cyklicznego uwzględniającego realizację procesów pierwszego i drugiego poziomu zachowań został przedstawiony na rys. 2.17.



Rys. 2.17. Cykliczny harmonogram realizacji procesów systemu z rys. 2.6

Jak widać, w zależności od poziomu zachowań wyróżniane są różne okresy powtarzalności procesów:  $\alpha = 9$  dla procesów lokalnych,  $m^1\alpha = 27$  dla procesów multimodalnych pierwszego poziomu zachowań ( $m^1\alpha$  – okres realizacji procesów  $m^1P_1, m^1P_2$ ),  $m^2\alpha = 54$  dla procesu multimodalnego drugiego poziomu zachowań ( $m^2\alpha$  – okres realizacji procesu  $m^2P_1$ ). Ostatecznie, okres całego systemu wynosi 54 u.j.c., to znaczy jest równy okresowi najwyższego poziomu zachowań (tzn. maksymalnej wartości z:  $\alpha, m^1\alpha, m^2\alpha$ ).

Oznacza to, że Twierdzenie 1, sformułowane dla poziomu procesów lokalnych, może być rozszerzone na wszystkie pozostałe poziomy procesów multimodalnych, przebieg których warunkują przebiegi procesów lokalnych.

### Twierdzenie 2

*Wstrzymanie realizacji co najmniej jednego procesu  $m^l P_i$   $l$ -tego poziomu zachowań systemu o strukturze  $SC^{lp}$  (18) skutkuje zatrzymaniem realizacji wszystkich procesów  $l$ -tego poziomu.*

Dowód: ze względu na to, że procesy multimodalne, należące do  $l$ -tego poziom zachowań otrzymują dostęp do zasobów zgodnie z tymi samymi regułami prioryteowania  $\theta$  co procesy lokalne, dowód Twierdzenia 2 wygląda analogicznie jak w przypadku Twierdzenia 1. □

Twierdzenie 2 oznacza, że blokada (będąca konsekwencją wstrzymania procesów w wyniku niewłaściwego rozstrzygnięcia konfliktów zasobowych) może wystąpić na dowolnym poziomie zachowań. Przyjmuje się, że system osiąga stan blokady jeżeli dochodzi do blokady między procesami przynajmniej na jednym poziomie zachowań.

Wracając do przykładu transportu publicznego z rys. 2.2 system tego typu osiąga blokadę gdy dochodzi do zablokowania procesów na jednym z poziomów: pociągów, pasażerów lub paczek. Inaczej mówiąc system osiągnie ustalony przebieg cykliczny jeżeli na żadnym z poziomów zachowań nie dochodzi do wstrzymania procesów.

Zarówno przebiegi przejściowe jak i ustalone przebiegi cykliczne uwzględniające multimodalny charakter procesów, mogą zostać scharakteryzowane przez zachowanie opisane przestrzenią stanów  $\mathcal{P} = Y(SC^{lp})$ . Formalna definicja systemu klasy SWMPC o strukturze  $SC^{lp}$  typu (18) wymaga wprowadzenia kolejnych dwóch pojęć: **stanu systemu i funkcji przejścia.**

#### 2.2.1.1. Stan

Pojęcie stanu systemu można przybliżyć odwołując się do analogii skrzyżowań ruchu ulicznego. Skrzyżowanie jest traktowane jako zasób  $R_k$ , na którym w danym momencie może przebywać ograniczona (określona jego pojemnością  $C(R_k)$ ) liczba pojazdów (potoków). Dostęp do skrzyżowania limituje system sygnalizacji świetlnej, w którym sekwencja świateł: ... – czerwony – pomarańczowy – zielony – pomarańczowy..., modeluje schemat statycznej (tzn. niezależnej od tego co się dzieje na skrzyżowaniu) reguły priorytetowania. Stan takiego skrzyżowania związany z pewną chwilą czasu jest charakteryzowany wielkościami opisującymi alokację pojazdów (czyli to jakie pojazdy znajdują się na skrzyżowaniu) oraz aktualnym kolorem świateł.

Innymi słowy stan obejmuje, tak wielkości opisujące aktualne położenie potoków (na każdym poziomie zachowania), jak i obowiązujące w danym momencie prawa dostępu do zasobów (kolory świateł).

Powyższa uwaga wskazuje na potrzebę rozszerzenia stosowanego dotychczas pojęcia stanu intuicyjnie aczkolwiek arbitralnie odnoszącego się jedynie do sposobu

alokacji potoków na zasobach (rys. 2.10 – 2.12). W rzeczywistości alokacja potoków jest tylko jednym z elementów stanu. Poniżej formalna definicja stanu uwzględnia zarówno alokację potoków, jak i aktualne prawa dostępu do zasobów – w tym również wielopoziomowość struktury  $SC^{lp}$  (18). Uwzględnienie powyższych wielkości pozwala uniknąć niejednoznaczności związanych z rozróżnianiem stanów (stanów o wspólnych alokacjach występujących wielokrotnie w jednym cyklu) w systemach o jednostkowych czasach wykonywania operacji (Bocewicz i Banaszak 2010) – SWMPC klasy:  $MW_{1,1}$ ,  $MW_{1,2}$ ,  $MW_{2,1}$ ,  $MW_{2,2}$  (tab. 2.1). Dodatkowo należy podkreślić, że w odróżnieniu od poprzedniej poniższa definicja stanu **obejmuje zarówno procesy lokalne jak i procesy multimodalne**.

Stan SWMPC definiowany jest następująco:

$$S^r = \left( \left( (SL^r, m^1 S^r), \dots, m^l S^r \right), \dots \right), m^{lp} S^r, \quad (19)$$

gdzie:

- $S^r$  –  $r$ -ty stan SWMPC opisanego strukturą  $SC^{lp}$  (18),  $lp$  – liczba poziomów zachowań  $SC^{lp}$ ,
- $SL^r$  –  $r$ -ty stan procesów lokalnych, realizowanych na poziomie  $SL$ :

$$SL^r = (A^r, Z^r, Q^r), \quad (20)$$

gdzie:

$A^r = (A_1^{r,1}, \dots, A_i^{r,h}, \dots, A_{ln}^{r,ls(ln)})$  – **sekwencja alokacji potoków** procesów lokalnych ( $ln$  – liczba procesów lokalnych,  $ls(i)$  – liczba potoków  $P_i^h$  procesu  $P_i$ ):  
 $A_i^{r,h}$  – alokacja potoku  $P_i^h$ , w stanie  $SL^r$  tj.:

$$A_i^{r,h} = (a_{i,1}^{r,h}, a_{i,2}^{r,h}, \dots, a_{i,k}^{r,h}, \dots, a_{i,lk}^{r,h}),$$

gdzie:  $lk$  – liczba zasobów  $R$ ,  $a_{i,k}^{r,h} \in \{0,1\}$ ,  $a_{i,k}^{r,h} = 1$  oznacza zasób  $R_k$  zajęty przez  $P_i^h$ , **przez zajęcie zasobu  $R_k$  przez potok  $P_i^h$  rozumie się wykorzystanie przez  $P_i^h$  co najmniej jednej jednostki tego zasobu**;  $a_{i,k}^{r,h} = 0$  – oznacza zasób  $R_k$  niezajęty przez  $P_i^h$  (może być zajęty przez inny potok). Tego typu zapis alokacji dopuszcza zajmowanie (wykorzystanie) zasobów przez wiele potoków jednocześnie. W przypadku gdy zasoby są o pojemności równej 1 tzn.  $C(R_k) = 1$ , dla każdego  $R_k \in R$ , stosowana jest uproszczona forma:

$$A^r = (a_1^r, a_2^r, \dots, a_k^r, \dots, a_{lk}^r),$$

$A^r$  – **alokacja potoków** procesów lokalnych w  $r$ -tym stanie dla systemów z zasobami pojemności jednostkowej,  $a_k^r \in H^0 \cup \{\Delta\}$  ( $H^0$  – zbiór wszystkich potoków poziomu  $SL$ ,  $l = 0$ ),  $a_k^r = P_i^h$  – **alokacja potoku** oznaczająca, że  $k$ -ty zasób  $R_k$  jest zajmowany przez potok  $P_i^h$ , i  $a_k^r = \Delta$  – oznacza, że  $k$ -ty zasób  $R_k$  jest wolny.

$Z^r = (z_1^r, z_2^r, \dots, z_k^r, \dots, z_m^r)$  – **sekwencja semaforów**  $r$ -tego stanu,  $z_k^r \in H^0$  – **semafor** określający potok (jeden z potoków reguły  $\sigma_k^0$ , przypisanej do  $R_k$ )

który ma prawo dostępu do zasobu  $R_k$  w następnej kolejności; tj.  $z_k^r = P_i^h$  oznacza, że strumień  $P_i^h$  ma w  $r$ -tym stanie prawo dostępu w następnej kolejności do zasobu  $R_k$ .

$Q^r = (q_1^r, q_2^r, \dots, q_k^r, \dots, q_m^r)$  – sekwencja indeksów semaforów  $r$ -tego stanu,  $q_k^r$  – indeks determinujący pozycję wartości semafora  $z_k^r$  w regule priorytetowania  $\sigma_k^0$ ,  $z_k^r = s_{k,(q_k^r)}$ ,  $q_k^r \in \mathbb{N}$ . Na przykład,  $q_2^r = 2$  i  $z_2^r = P_1^2$  oznacza, że wartość  $P_1^2$  semafora  $z_2^r$  jest drugim elementem reguły priorytetowania  $\sigma_2^0$  przypisanej do zasobu  $R_k$ .

- $m^l S^r$  –  $r$ -ty stan procesów multimodalnych realizowanych na poziomie  $SM^l$ :

$$m^l S^r = (m^l A^r, m^l Z^r, m^l Q^r), \quad (21)$$

gdzie:

$m^l A^r$  – sekwencja alokacji potoków procesów multimodalnych poziomu  $SM^l$ :

$$m^l A^r = (m^l A_1^{r,1}, \dots, m^l A_i^{r,h}, \dots, m^l A_{lw(l)}^{r,lsm(lw(l),l)}),$$

$m^l A_i^{r,h}$  – alokacja potoku  $m^l P_i^h$ , w stanie  $m^l S^r$  tj.:

$$m^l A_i^{r,h} = (m^l a_{i,1}^{r,h}, m^l a_{i,2}^{r,h}, \dots, m^l a_{i,k}^{r,h}, \dots, m^l a_{i,lk}^{r,h}),$$

gdzie:  $lk$  – liczba zasobów  $R$ ,  $lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu,  $lsm(i,l)$  – liczba potoków procesu multimodalnego  $m^l P_i$ ,  $m^l a_{i,k}^{r,h} \in \{0,1\}$ ,  $m^l a_{i,k}^{r,h} = 1$  oznacza zasób  $R_k$  zajmowany przez  $m^l P_i^h$ ;  $m^l a_{i,k}^{r,h} = 0$  – oznacza zasób  $R_k$  niezajęty przez  $m^l P_i^h$  (ale może być zajęty przez innych potok).

$m^l Z^r = (m^l z_1^r, m^l z_2^r, \dots, m^l z_k^r, \dots, m^l z_m^r)$  – sekwencja semaforów  $r$ -tego stanu,  $m^l z_k^r \in H^l$  – określa potok (jeden z potoków reguły  $\sigma_k^l$ , przypisanej do  $R_k$ ), który ma prawo dostępu do zasobu  $R_k$ ; tj.  $m^l z_k^r = m^l P_i^h$  oznacza, że potok  $m^l P_i^h$  ma w  $r$ -tym stanie prawo dostępu do zasobu  $R_k$ . W przypadku gdy zasób  $R_k$  nie jest wykorzystywany przez żaden proces  $l$ -tego poziomu, wartość semafora jest następująca:  $m^l z_k^r = \Delta$  (tego typu zasoby nie posiadają reguły priorytetowania dla  $l$ -tego poziomu a zatem semafor nie ma na co wskazywać).

$m^l Q^r = (m^l q_1^r, m^l q_2^r, \dots, m^l q_k^r, \dots, m^l q_m^r)$  – sekwencja indeksów semaforów  $r$ -tego stanu,  $m^l q_k^r$  determinuje pozycję wartości semafora  $m^l z_k^r$  w regule priorytetowania  $\sigma_k^l$ ,  $m^l z_k^r = s_{k,(m^l q_k^r)}$ ,  $m^l q_k^r \in \mathbb{N}$ . Gdy  $m^l z_k^r = \Delta$  to  $m^l q_k^r = 0$ .

W przypadku gdy na danym ( $l$ -tym) poziomie zachowań zakłada się możliwość jednoczesnego współdzielenia zasobów przez procesy multimodalne (tzn. zbiór reguł jest pusty  $\theta^l = \emptyset$ ) semafony  $m^l Z^r$  oraz indeksy  $m^l Q^r$  są pomijane w opisie stanu  $m^l S^r$  (21). Stan  $m^l S^r$  definiowany jest wówczas z dokładnością do alokacji



procesów, co w konsekwencji znajduje swoje wykorzystanie przy modelowaniu zachowania systemów typu:  $MW_{2,1}$  i  $MW_{2,2}$  (tab. 2.1).

W definicji stanu (19) przyjęto ten sam sposób zapisu jak w strukturze  $SC^{lp}$  (18). Stan  $S^r$  jest traktowany jako kompozycja stanów  $Sl^r, m^1S^r, m^2S^r, \dots, m^{lp}S^r$  kolejnych poziomów zachowań. Podobnie jak w przypadku struktury systemu zapis ma charakter zagnieżdżony, każdy kolejny poziom zachowań zwiększa głębokość zagnieżdżenia. Stan  $Sl^r$  odpowiadający poziomowi procesów lokalnych oraz stan  $m^lS^r$  odpowiadający poziomom procesów multimodalnych składa się z trzech elementów: alokacji, semaforów oraz indeksów semaforów.

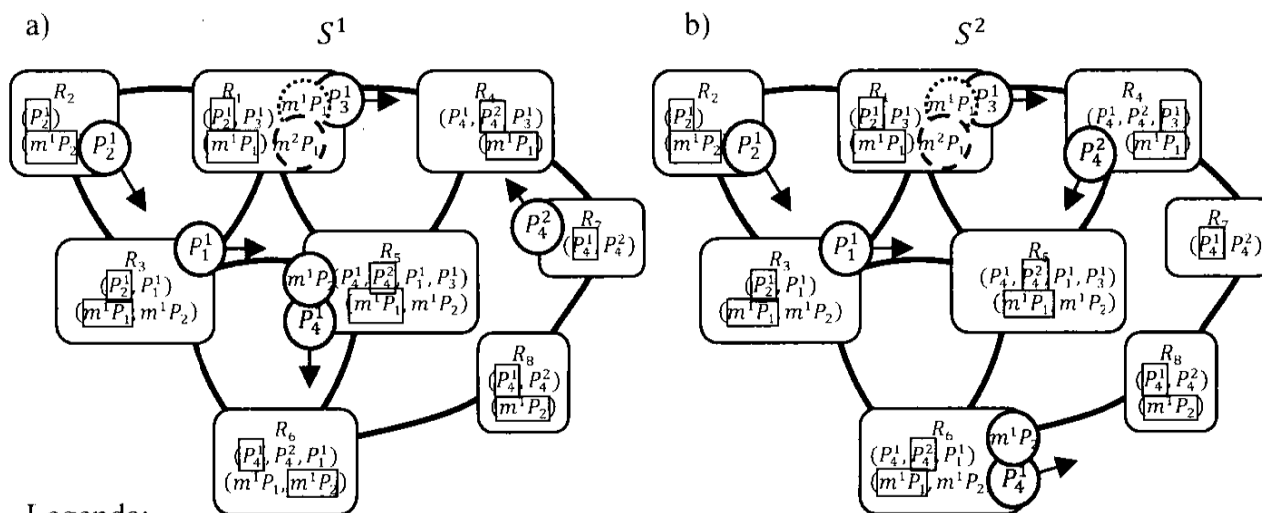
Nowymi, nie omawianymi wcześniej, elementami są sekwencje semaforów i indeksów. O ile alokacje określają aktualne położenie na zasobach potoków procesów o tyle semafony i indeksy określają aktualne wskazania reguł priorytetowania. Wartości semaforów określają potoki, które mają w danym momencie prawo dostępu do zasobów (pod warunkiem, że zasoby pozostaną zwolnione) z kolei indeksy określają pozycje potoku wskazanego przez semafor w regule priorytetowania (indeksy wprowadzone zostały ze względu na możliwość wielokrotnego występowania w regule tych samych potoków).

Wracając do przykładu ruchu ulicznego, stan skrzyżowania związany z pewną chwilą czasu można opisać dwoma wielkościami: alokacją pojazdów (umiejscowienie pojazdów na skrzyżowaniu) oraz aktualnym kolorem świateł (wartością semafora). Pojazd może wjechać na skrzyżowanie (tzn. zająć zasób) pod warunkiem, że jest ono wolne oraz ma do tego prawo (kolor zielony semafora).

Semafor „wyświetla kolory” (określając tym samym prawo dostępu) zgodnie z kolejnością określoną w regule priorytetowania. Z kolei indeksy semaforów określające pozycję semafora w regule, nabierają znaczenia w przypadku gdy w regule priorytetowania występują elementy powtarzające się. Takim elementem jest kolor pomarańczowy sygnalizacji świetlnej. Wyświetlenie przez semafor koloru pomarańczowego nie daje informacji, z którym wariantem koloru kierowca ma do czynienia. Nie wiadomo czy kierowca powinien szykować się do jazdy (zaraz zaświeci się zielone) czy też odwrotnie zatrzymać się (zaraz zaświeci się czerwone)? Numerowanie wskazań semaforów, zgodnie z kolejnością ich występowania, ma na celu uniknięcie tego typu niejednoznaczności.

W celu ilustracji, na rys. 2.18 przedstawiono dwa kolejne stany, osiągnane przez system z rys. 2.6. Są to stany, oznaczone jako  $S^1$  i  $S^2$ , występujące w przebiegu cyklicznym z rys. 2.17. Używając wprowadzonej terminologii stany te można przedstawić tak jak w tab. 2.2. W rozważanym systemie wyróżnia się trzy poziomy zachowań: procesów lokalnych  $SL$  i dwa poziomy procesów multimodalnych  $SM^1$  (procesy oznaczone kolorami niebieskim i czerwonym) oraz  $SM^2$  (proces oznaczony kolorem zielonym). Przy czym każdy stan z tab. 2.2 dzieli się na: stan poziomu procesów lokalnych (np.  $Sl^1$ ) oraz dwa stany procesów multimodalnych (np.  $m^2S^1$ ).

Stany z poziomu procesów lokalnych charakteryzują się pięcioelementową sekwencją alokacji. Każdy element, związany z jednym z potoków  $P_1^1, P_2^1, P_3^1, P_4^1$  lub  $P_4^2$ , jest sekwencją zero jedynkową zawierającą 8 elementów. Wartość każdego z nich określa zajętość zasobów  $R_1 - R_8$  przez potok.



Legenda:

$\odot P_i^h$  – oczekiwanie potoku  $P_i^h$  na dostęp do żądanego zasobu

$\odot m^1 P_1, \odot m^1 P_2, \odot m^2 P_1$  – oczekiwanie potoku  $m^1 P_1, m^1 P_2, m^2 P_1$  na dostęp do żądanego zasobu

$\odot m^1 P_2$  – realizacja operacji potoku  $m^1 P_2$ ,  $\odot P_i^h$  – realizacja operacji potoku  $P_i^h$

$\begin{matrix} R_k \\ (s_{k,1}^0, \dots, s_{k,d}^0, \dots, s_{k,th(k,0)}^0) \\ (s_{k,1}^1, \dots, s_{k,d}^1, \dots, s_{k,th(k,1)}^1) \end{matrix}$  – zasób współdzielony  $R_k$  z regułami priorytetowania  $\sigma_k^0 = (s_{k,1}^0, \dots, s_{k,d}^0, \dots, s_{k,th(k,0)}^0)$ ,  $\sigma_k^1 = (s_{k,1}^1, \dots, s_{k,d}^1, \dots, s_{k,th(k,1)}^1)$  oraz semaforami z wartościami  $z_k^0 = s_{k,d}^0$ ,  $z_k^1 = s_{k,d}^1$

Rys. 2.18. Dwa kolejne stany systemu z rys. 2.6: stan  $S^1$  a), stan  $S^2$  b)

Tabela 2.2. Stany:  $S^1$  i  $S^2$  w reprezentacji (19)

	$S^1 = ((Sl^1, m^1 S^1), m^2 S^1)$	$S^2 = ((Sl^2, m^1 S^2), m^2 S^2)$
SL	$Sl^1 = (A^1, Z^1, Q^1)$	$Sl^2 = (A^2, Z^2, Q^2)$
	$A^1 = (A_1^{1,1}, A_2^{1,1}, A_3^{1,1}, A_4^{1,1}, A_4^{1,2})$ $A_1^{1,1} = (0,0,1,0,0,0,0,0)$ $A_2^{1,1} = (0,1,0,0,0,0,0,0)$ $A_3^{1,1} = (1,0,0,0,0,0,0,0)$ $A_4^{1,1} = (0,0,0,0,1,0,0,0)$ $A_4^{1,2} = (0,0,0,0,0,0,1,0)$	$A^2 = (A_1^{2,1}, A_2^{2,1}, A_3^{2,1}, A_4^{2,1}, A_4^{2,2})$ $A_1^{2,1} = (0,0,1,0,0,0,0,0)$ $A_2^{2,1} = (0,1,0,0,0,0,0,0)$ $A_3^{2,1} = (1,0,0,0,0,0,0,0)$ $A_4^{2,1} = (0,0,0,0,0,1,0,0)$ $A_4^{2,2} = (0,0,0,1,0,0,0,0)$
	$Z^1 = (P_2^1, P_2^1, P_2^1, P_4^1, P_4^1, P_4^1, P_4^1, P_4^1)$	$Z^2 = (P_2^1, P_2^1, P_2^1, P_3^1, P_4^1, P_4^1, P_4^1, P_4^1)$
	$Q^1 = (2,1,2,2,1,1,2,1)$	$Q^2 = (2,1,2,2,2,1,1,1)$
SM <sup>1</sup>	$m^1 S^1 = (m^1 A^1, m^1 Z^1, m^1 Q^1)$	$m^1 S^2 = (m^1 A^2, m^1 Z^2, m^1 Q^2)$
	$m^1 A^1 = (m^1 A_1^1, m^1 A_2^1)$ $m^1 A_1^1 = (1,0,0,0,0,0,0,0)$ $m^1 A_2^1 = (0,0,0,0,1,0,0,0)$	$m^1 A^2 = (m^1 A_1^2, m^1 A_2^2)$ $m^1 A_1^2 = (1,0,0,0,0,0,0,0)$ $m^1 A_2^2 = (0,0,0,0,0,1,0,0)$
	$m^1 Z^1 = (m^1 P_1, m^1 P_2, m^1 P_1, m^1 P_1, m^1 P_1, m^1 P_2, \Delta, m^1 P_2)$	$m^1 Z^2 = (m^1 P_1, m^1 P_2, m^1 P_1, m^1 P_1, m^1 P_1, m^1 P_1, \Delta, m^1 P_2)$
	$m^1 Q^1 = (1,1,1,1,2,2,0,1)$	$m^1 Q^2 = (1,1,1,1,1,2,0,1)$
SM <sup>2</sup>	$m^2 S^1 = (m^2 A^1)$	$m^2 S^2 = (m^2 A^2)$
	$m^2 A^1 = (m^2 A_1^1)$ $m^2 A_1^1 = (1,0,0,0,0,0,0,0)$	$m^2 A^2 = (m^2 A_1^2)$ $m^2 A_1^2 = (1,0,0,0,0,0,0,0)$

Przykładowo, wartość jeden ( $\alpha_{4,5}^{1,1} = 1$ ) piątego elementu sekwencji  $A_4^{1,1}$  oznacza, że potok  $P_4^1$  zajmuje zasób  $R_5$  (tak jak to jest pokazane na rys. 2.18a). Warto zauważyć, że w ramach alokacji nie rozróżnia się tego czy potok wykonuje na danym zasobie operację (tak jak potok  $P_4^1$  w stanie  $S^1$  i potoki  $P_4^1, P_4^2$  w stanie  $S^2$ ), czy też jest w trakcie oczekiwania na dostęp do kolejnego zasobu (tak jak potoki  $P_1^1, P_2^1, P_3^1, P_4^2$  w stanie  $S^1$  oraz potoki  $P_1^1, P_2^1, P_3^1$  w stanie  $S^2$ ). Przyjęta postać alokacji nie dopuszcza także do sytuacji, w których potok wymaga więcej niż jednej jednostki zajmowanego zasobu  $\omega_{i,j}^h = 1$  ( $m^l \omega_{i,j}^h = 1$ ).

Podobnie jak elementy sekwencji alokacji, sekwencje semaforów i indeksów zawierają po 8 elementów (każdy związany z jednym zasobem systemu). Przykładowo, czwarty element sekwencji  $Z^1$  oznacza, że prawo dostępu do zasobu  $R_4$  ma potok  $P_4^2$ . Semafor wskazuje więc na drugi element reguły  $\sigma_4^0 = (P_4^1, P_4^2, P_3^1)$  ponieważ czwarty element indeksu  $Q^1$  jest równy 2. Zarówno semafony  $Z^1$  jak i indeksy  $Q^1$  zmieniają wartość czwartego elementu po tym jak w stanie  $S^2$  potok  $P_4^2$  zajmuje zasób  $R_4$ :  $z_4^2 = P_3^1, q_4^2 = 3$ .

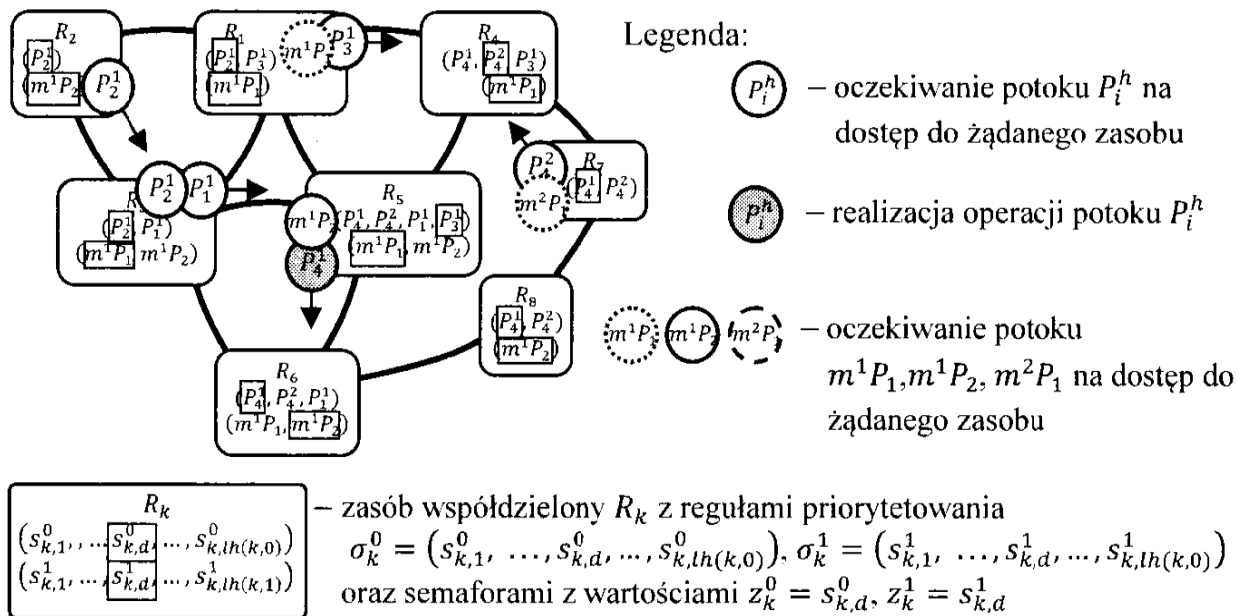
W przypadku procesów multimodalnych pierwszego poziomu, alokacja zawiera już tylko dwa elementy – w systemie występują tylko dwa procesy multimodalne 1 poziomu zachowań. Ze względu na to, że oba procesy są jednopotokowe pominięto w oznaczeniach  $m^1 A_1^1, m^1 A_2^1$  drugi element indeksu górnego. Semafony ( $m^1 Z^1, m^1 Z^2$ ) oraz indeksy ( $m^1 Q^1, m^1 Q^2$ ) są definiowane analogicznie jak w przypadku procesów lokalnych. Uwagę należy zwrócić na siódmy element semaforów o wartości  $\Delta$ . Oznacza on, że zasób  $R_7$  nie bierze udziału w realizacji procesów multimodalnych. Inaczej mówiąc, dla tego zasobu reguła priorytetowania nie została zdefiniowana. Z tego też powodu indeksy zasobu  $R_7$  przyjmują wartość 0.

Drugi poziom zachowań zawiera tylko jeden proces multimodalny (zielony kolor na rys. 2.17). Brak innych procesów powoduje, że nie dochodzi na tym poziomie do konfliktów zasobowych. W związku z tym, zasobom nie zostały przydzielone reguły priorytetowania i tym samym stan  $m^2 S^1$  ( $m^2 S^2$ ) nie zawiera semaforów i indeksów.

Proponowana koncepcja stanu (19) obejmuje informacje o realizacji procesów wszystkich poziomów zachowań oraz o wszystkich prawach dostępu do zasobów. W tym też sensie taki stan systemu ma charakter globalny (gdyż opisuje cały system).

Wykorzystanie do opisu SWMPC stanu (19) ogranicza się jednak dalej do systemów typu  $MW_{1,1}, MW_{1,2}, MW_{2,1}, MW_{2,2}$ , (patrz tab. 2.1 – systemy leżące w polu ograniczonym fioletową linią). Jest to wynikiem tego, że stan (19) nie uwzględnia momentów rozpoczęcia operacji oraz czasów ich wykonania. Ogranicza w ten sposób możliwość opisu systemów tylko do procesów, które charakteryzują się jednostkowymi czasami wykonania operacji:  $t_{i,j} = 1, m^l t_{i,j} = 1$ . Ponadto, w przyjętej definicji (19) zakłada się jednostkowe zapotrzebowanie procesów na zasoby:  $\omega_{i,j}^h = 1, m^l \omega_{i,j}^h = 1$ .

Warto zauważyć, że nie każdy potencjalnie możliwy stan jest stanem dopuszczalnym w systemie o zadanej strukturze  $SC^{lp}$ . Przykładowo, stan prezentowany na rys. 2.19 nie jest stanem dopuszczalnym w systemie z rys. 2.6. Potok  $m^2 P_1$  zajmuje zasób  $R_7$  nienależący do jego marszruty, potok  $P_2^1$  jednocześnie zajmuje zasoby  $R_2, R_3$ , natomiast semafor zasobu  $R_5$  wskazuje na niewłaściwy potok.



Rys. 2.19. Stan niedopuszczalny w systemie z rys. 2.6

Dla potrzeb dalszych rozważań przyjmuje się, że stan  $S^r$  (19) jest **stanem dopuszczalnym** w systemie  $MW_{1,1}, MW_{1,2}, MW_{2,1}, MW_{2,2}$  o strukturze  $SC^{lp}$  (18) wtedy i tylko wtedy gdy:

1. Każdy potok SWMPC przypisany jest tylko do jednego zasobu. Własność ta dla procesów lokalnych wyrażana jest następująco:

$$\sum_{k=1}^{lk} a_{i,k}^{r,h} = 1 \text{ dla } i = 1 \dots ln; h = 1 \dots ls(i), \quad (22)$$

gdzie:  $a_{i,k}^{r,h}$  – alokacja potoku  $P_i^h$  na zasobie  $R_k$ ,  $lk$  – liczba zasobów  $R$ ,  $ln$  – liczba procesów lokalnych  $P_i$ ,  $ls(i)$  – liczba potoków  $P_i^h$  procesu lokalnego  $P_i$ .

Dla procesów multimodalnych:

$$\sum_{k=1}^{lk} m^l a_{i,k}^{r,h} = 1 \text{ dla } i = 1 \dots lw(l); h = 1 \dots lsm(i, l); l = 1 \dots lp, \quad (23)$$

gdzie:  $m^l a_{i,k}^{r,h}$  – alokacja potoku  $m^l P_i^h$  na zasobie  $R_k$ ,  $lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu,  $lsm(i, l)$  – liczba potoków procesu multimodalnego  $m^l P_i$ ,  $lp$  – liczba poziomów zachowań procesów multimodalnych.

2. Alokacja potoków może odbywać się tylko na zasobach należących do marszruty procesu. Własność ta dla procesów lokalnych wyrażona jest następująco:

$$a_{i,k}^{r,h} = 0, \forall R_k \notin ps_i; i = 1 \dots ln; h = 1 \dots ls(i), \quad (24)$$

gdzie:  $a_{i,k}^{r,h}$  – alokacja potoku  $P_i^h$  na zasobie  $R_k$  ( $a_{i,k}^{r,h} = 0$  oznacza, że zasób nie jest zajmowany przez potok  $P_i^h$ ),  $ps_i$  – zbiór zasobów wchodzących w skład marszruty

$p_i$  procesu lokalnego  $P_i$ ,  $ps_i = \{p_{i,j} | j = 1 \dots lr(i)\}$ ,  $p_{i,j} \in R$  –  $j$ -ty element marszruty  $p_i$ ,  $lr(i)$  – długość marszruty  $p_i$ .

Analogicznie dla procesów multimodalnych:

$$m^l a_{i,k}^{r,h} = 0, \forall R_k \notin m^l ps_i; i = 1 \dots lw(l); h = 1 \dots lsm(i, l); l = 1 \dots lp, \quad (25)$$

gdzie:  $m^l a_{i,k}^{r,h}$  – alokacja potoku  $m^l P_i^h$  na zasobie  $R_k$  ( $m^l a_{i,k}^{r,h} = 0$  oznacza, że zasób nie jest zajmowany przez potok  $m^l P_i^h$ ),  $m^l ps_i$  – zbiór zasobów wchodzących w skład marszruty  $m^l p_i$  procesu multimodalnego  $m^l P_i$ ,  $m^l ps_i = \{m^l p_{i,j} | j = 1 \dots lm(i, l)\}$ ,  $m^l p_{i,j} \in R$  –  $j$ -ty element marszruty  $m^l p_i$ ,  $lm(i, l)$  – długość marszruty  $m^l p_i$ .

3. **Liczba potoków zajmujących zasób  $R_k$  nie może przekraczać pojemności zasobu  $C(R_k)$**  (założenie to jest słuszne w przypadku gdy zapotrzebowanie potoków na jednostki zasobów wynosi:  $\omega_{i,j}^h = 1$ ,  $m^l \omega_{i,j}^h = 1$ ). Własność ta dla procesów lokalnych wyrażona jest następująco:

$$\sum_{i=1}^{ln} \sum_{h=1}^{ls(i)} a_{i,k}^{r,h} \leq C(R_k) \text{ dla } k = 1 \dots lk, \quad (26)$$

gdzie:  $a_{i,k}^{r,h}$  – alokacja potoku  $P_i^h$  na zasobie  $R_k$ ,  $lk$  – liczba zasobów  $R$ ,  $ln$  – liczba procesów lokalnych  $P_i$ ,  $ls(i)$  – liczba potoków  $P_i^h$  procesu lokalnego  $P_i$ .

Analogicznie dla procesów multimodalnych (warunek ten obowiązuje tylko w systemach synchronizowanych protokołem wzajemnego wykluczania):

$$\sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i,l)} m^l a_{i,k}^{r,h} \leq C(R_k) \text{ dla } k = 1 \dots lk; l = 1 \dots lp, \quad (27)$$

gdzie:  $m^l a_{i,k}^{r,h}$  – alokacja potoku  $m^l P_i^h$  na zasobie  $R_k$ ,  $lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu,  $lsm(i, l)$  – liczba potoków procesu multimodalnego  $m^l P_i$ ,  $lp$  – liczba poziomów zachowań procesów multimodalnych,  $lk$  – liczba zasobów zbioru  $R$ .

4. **Wartości semaforów zajętych zasobów wskazują na potoki występujące w regule priorytetowania zaraz po zaalokowanych procesach.** Własność ta dla procesów lokalnych wyrażona jest następująco:

$$(a_{i,k}^{r,h} = 1) \Rightarrow (z_k^r = s_{k,d}^0) \text{ dla } i = 1 \dots ln; h = 1 \dots ls(i); k = 1 \dots lk, \quad (28)$$

gdzie:  $s_{k,d}^0$  – potok procesu lokalnego wykorzystującego do swojej realizacji zasób  $R_k$ ,  $s_{k,d}^0$  jest również  $d$ -tym elementem reguły priorytetowania  $\sigma_k^0 = (s_{k,1}^0, \dots, s_{k,d-2}^0, s_{k,d-1}^0, s_{k,d}^0, \dots, s_{k,lh(k,0)}^0)$ , będącym następnikiem potoków  $\dots, s_{k,d-2}^0, s_{k,d-1}^0$  aktualnie zajmujących zasób  $R_k$ ,  $lh(k, 0)$  – długość reguły  $\sigma_k^0$ .

Analogicznie dla procesów multimodalnych (warunek ten obowiązuje tylko w systemach synchronizowanych protokołem wzajemnego wykluczania):

$$(m^l a_{i,k}^{r,h} = 1) \Rightarrow (m^l z_k^r = s_{k,d}^l), \quad (29)$$

dla  $i = 1 \dots lw(l); h = 1 \dots lsm(i, l); l = 1 \dots lp; k = 1 \dots lk$ ,

gdzie:  $s_{k,d}^l$  – potok procesu multimodalnego  $l$ -tego poziomu, wykorzystującego do swojej realizacji zasób  $R_k$ ,  $s_{k,d}^l$  jest również  $d$ -tym elementem reguły priorytetowania  $\sigma_k^l = (s_{k,1}^l, \dots, s_{k,d-2}^l, s_{k,d-1}^l, s_{k,d}^l, \dots, s_{k,lh(k,l)}^l)$ , następnik potoku  $s_{k,d-1}^l$ , aktualnie zajmującego zasób  $R_k$ ,  $lh(k, l)$  – długość reguły  $\sigma_k^l$ ,  $lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu,  $lsm(i, l)$  – liczba potoków procesu multimodalnego  $m^l P_i$ ,  $lp$  – liczba poziomów zachowań procesów multimodalnych,  $lk$  – liczba zasobów zbioru  $R$ .

**5. Wartości indeksów semaforów są numerami występowania potoków w regule priorytetowania.** Własność ta dla procesów lokalnych wyrażona jest następująco:

$$q_k^r \in QE(z_k^r), k = 1 \dots lk, \quad (30)$$

gdzie:  $QE(z_k^r)$  – zbiór pozycji występowania potoku wskazanego przez semafor  $z_k^r$  w regule priorytetowania  $\sigma_k^0$ .

Przykładowo jeżeli reguła priorytetowania ma postać:  $\sigma_2^0 = (P_1^1, P_2^1, P_1^1)$  a semafor w stanie  $S^r$  przyjmuje wartość:  $z_k^r = P_1^1$  to zbiór pozycji występowania potoku ma postać:  $QE(z_k^r) = \{1, 3\}$  – co oznacza że potok  $P_1^1$  występuje na 1 i 3 pozycji reguły  $\sigma_2^0$ .

Analogicznie dla procesów multimodalnych (warunek ten obowiązuje tylko w systemach synchronizowanych protokołem wzajemnego wykluczania):

$$m^l q_k^r \in QE(m^l z_k^r), l = 1 \dots lp; k = 1 \dots lk, \quad (31)$$

gdzie:  $QE(m^l z_k^r)$  – zbiór pozycji występowania potoku wskazanego przez semafor  $m^l z_k^r$  w regule priorytetowania  $\sigma_k^l$ ,  $lp$  – liczba poziomów zachowań procesów multimodalnych,  $lk$  – liczba zasobów zbioru  $R$ .

Przedstawione powyżej własności (22) – (31) określają dopuszczalne stany systemu SWMPC o zadanej wielopoziomowej strukturze  $SC^{lp}$  (18). Zbiór wszystkich dopuszczalnych stanów  $S^{lp}$  obejmujący  $l$ -poziomów zachowania systemu dzieli się na podzbiory stanów występujących w cyklicznych przebiegach ustalonych i przebiegach przejściowych systemu.

### 2.2.1.2. Funkcja przejścia

Na rys. 2.14 i 2.15 przedstawione zostały przykładowe przestrzenie stanów  $\mathcal{P}$  systemu z rys. 2.6. Dopuszczalne stany tego typu systemu reprezentowane są jako wierzchołki digrafów wchodzących w skład tych przestrzeni. Łuki natomiast reprezentują możliwe przejścia między kolejnymi stanami dopuszczalnymi.

To, w jaki sposób stany połączone są ze sobą określa tzw. relacja przejścia. Relacja ta determinuje zbiór łuków  $\mathbb{E}^l$  łączących stany zbioru  $\mathbb{S}^{lp}$  ( $\mathbb{E}^{lp} \subseteq \mathbb{S}^{lp} \times \mathbb{S}^{lp}$ ). Przyjmując teorio-grafową interpretację przestrzeni  $\mathcal{P}$  odpowiadający jej digraf reprezentuje para:  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$ . Ponieważ rozważane dalej problemy są bądź to problemami analizy (tzn. zorientowanymi na analizę zachowań systemów o zadanej strukturze), bądź też problemami syntezy (tzn. zorientowanymi na syntezę struktury systemu gwarantującej jego określone zachowanie), tak więc w obu przypadkach poszukiwane są jednoznaczne rozwiązania bądź to zachowań bądź też struktur rozważanych systemów. W kontekście zachowania, przyjęte założenie oznacza jednoznaczność przejść pomiędzy stanami dopuszczalnymi składającymi się na dane zachowanie. Wymóg ten wymusza przyjęcie określonej funkcji przejścia determinującej osiągalność stanów w zbiorze  $\mathbb{S}^l$ .

Dane są dwa różne dopuszczalne stany SWMPC:  $S^e \neq S^f$ ,  $S^e, S^f \in \mathbb{S}^{lp}$ . Stan  $S^f$  jest osiągalny bezpośrednio ze stanu  $S^e$ , co oznaczane jest jako:  $S^e \rightarrow S^f$ , jeżeli spełnione są następujące warunki:

1. Warunki dotyczące poziomu  $SL$  zachowań procesów lokalnych:

(a) Jeżeli w stanie  $S^e$ :

- zasób  $R_k$  nie jest całkowicie zajęty, tzn. nie wszystkie jednostki zasobu (ich liczbę określa pojemność  $C(R_k)$ ) są wykorzystywane przez operacje potoków  $i$

- potok  $P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $z_k^e$  oczekuje na dostęp do tego zasobu,

to w stanie  $S^f$ :

- potok  $P_{\bar{i}}^{\bar{h}}$  zwalnia dotychczas zajmowany zasób  $R_{\bar{k}}$  i zajmuje zasób  $R_k$ ,
- semafor  $z_k^f$  oraz jego indeks  $q_k^f$  zmienia swoją wartość tak, aby wskazywać na kolejny potok określony zgodnie z regułą priorytetowania  $\sigma_k^0$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{ln} \sum_{h=1}^{ls(i)} a_{i,k}^{e,h} < C(R_k) \right) \wedge \left( a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \right] \Rightarrow \left[ \left( a_{\bar{i},\bar{k}}^{f,\bar{h}} = 0 \right) \wedge \left( a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \wedge \left( z_k^f = s_{k,\bar{d}}^0 \right) \wedge \left( q_k^f = \bar{d} \right) \right] \quad (32)$$

$$\text{dla } k = 1 \dots lk \text{ i } \forall R_{\bar{k}} \in R(k),$$

gdzie:

$lk$  – liczba zasobów  $R$ ,

$ln$  – liczba procesów lokalnych,

$ls(i)$  – liczba potoków procesu  $P_i$ ,

$C(R_k)$  – pojemność zasobu  $R_k$ ,

$\bar{i}, \bar{h}$  – indeksy potoku wskazanego przez semafor zasobu  $R_k$ :  $z_k^r = P_{\bar{i}}^{\bar{h}}$ ,

$\bar{k}$  – indeks zasobu  $R_{\bar{k}}$  zajmowanego przez potok  $P_{\bar{i}}^{\bar{h}}$ . Zasób  $R_{\bar{k}}$  jest zasobem, który poprzedza  $R_k$  w marszrucie procesu w marszrucie procesu  $P_i$ :  $p_i = (\dots, R_{\bar{k}}, R_k, \dots)$ . Zbiór  $R(k)$  zawiera wszystkie zasoby, które poprzedzają  $R_k$  w marszrutach procesów  $P_i$ ,

- $a_{i,k}^{e,h}$  – alokacja potoku  $P_i^h$  na zasobie  $R_k$  w stanie  $S^e$ ,  
 $a_{\bar{i},\bar{k}}^{e,\bar{h}}$  – alokacja potoku  $P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_{\bar{k}}$  w stanie  $S^e$ ,  
 $a_{\bar{i},\bar{k}}^{f,\bar{h}}$  – alokacja potoku  $P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_{\bar{k}}$  w stanie  $S^f$ ,  
 $a_{i,k}^{f,\bar{h}}$  – alokacja potoku  $P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_k$  w stanie  $S^f$ ,  
 $z_k^f$  – semafor zasobu  $R_k$  w stanie  $S^f$ ,  
 $q_k^f$  – indeks zasobu  $R_k$  w stanie  $S^f$ ,  
 $s_{k,\bar{d}}^0$  – element (potok) reguły priorytetowania  $\sigma_k^0$ , przypisanej do zasobu  $R_k$ ,  
występujący w regule zaraz po potoku  $P_{\bar{i}}^{\bar{h}}$ ,  $\bar{d} = \text{modl}(q_k^e + 1; lh(k, 0))$ ,  $\text{modl}(a, b)$  – funkcja zwracająca wartość  $r = \text{modl}(a, b)$   
gdy  $a = k \cdot b + r$  i  $0 < r \leq |b|$ ,  $a, b, c, k \in \mathbb{C}$ .

(b) Jeżeli w stanie  $S^e$ :

- zasób  $R_k$  jest całkowicie zajęty tzn. wszystkie jednostki zasobu (ich liczbę określa pojemność  $C(R_k)$ ) są wykorzystane przez realizowane na tym zasobie operacje i
  - potok  $P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $z_k^e$  oczekuje na dostęp do tego zasobu,
- to w stanie  $S^f$ :
- potok  $P_{\bar{i}}^{\bar{h}}$  pozostaje na dotychczas zajmowanym zasobie  $R_{\bar{k}}$ ,
  - semafor  $z_k^f$  oraz jego indeks  $q_k^f$  mają takie same wartości jak w stanie  $S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{ln} \sum_{h=1}^{ls(i)} a_{i,k}^{e,h} = C(R_k) \right) \wedge \left( a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \right] \Rightarrow$$

$$\Rightarrow \left[ \left( a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \wedge \left( a_{i,k}^{f,\bar{h}} = 0 \right) \wedge \left( z_k^f = z_k^e \right) \wedge \left( q_k^f = q_k^e \right) \right] \quad (33)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $lk$ ,  $ln$ ,  $ls(i)$ ,  $C(R_k)$ ,  $\bar{i}$ ,  $\bar{h}$ ,  $R_{\bar{k}}$ ,  $R_k$ ,  $R(k)$ ,  $a_{i,k}^{e,h}$ ,  $a_{\bar{i},\bar{k}}^{e,\bar{h}}$ ,  $a_{\bar{i},\bar{k}}^{f,\bar{h}}$ ,  $a_{i,k}^{f,\bar{h}}$ ,  $z_k^f$ ,  $q_k^f$  – określone jak w (32).

(c) Jeżeli w stanie  $S^e$ :

- potok  $P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $z_k^e$  nie oczekuje na dostęp do zasobu  $R_k$  i
  - potok  $P_{\bar{i}}^{\bar{h}}$  nie zajmuje zasobu  $R_k$ ,
- to w stanie  $S^f$ :
- potok  $P_{\bar{i}}^{\bar{h}}$  również nie zajmie tego zasobu,
  - semafor  $z_k^f$  oraz jego indeks  $q_k^f$  mają takie same wartości jak w stanie  $S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:



$$\left[ \left( a_{\bar{i},\bar{k}}^{e,\bar{h}} = 0 \right) \wedge \left( a_{\bar{i},k}^{e,\bar{h}} = 0 \right) \right] \Rightarrow \left[ \left( a_{\bar{i},k}^{f,\bar{h}} = 0 \right) \wedge (z_k^f = z_k^e) \wedge (q_k^f = q_k^e) \right] \quad (34)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

Z kolei jeżeli w stanie  $S^e$ :

- potok  $P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $z_k^e$  nie oczekuje na dostęp do zasobu  $R_k$  i
- potok  $P_{\bar{i}}^{\bar{h}}$  zajmuje zasób  $R_k$

to w stanie  $S^e$ :

- semafor  $z_k^f$  oraz jego indeks  $q_k^f$  nie ulegają zmianie.

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( a_{\bar{i},\bar{k}}^{e,\bar{h}} = 0 \right) \wedge \left( a_{\bar{i},k}^{e,\bar{h}} = 1 \right) \right] \Rightarrow \left[ (z_k^f = z_k^e) \wedge (q_k^f = q_k^e) \right] \quad (35)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $lk, ln, ls(i), C(R_k), \bar{i}, \bar{h}, R_{\bar{k}}, R_k, R(k), a_{i,k}^{e,h}, a_{\bar{i},\bar{k}}^{e,\bar{h}}, a_{\bar{i},\bar{k}}^{f,\bar{h}}, a_{\bar{i},k}^{f,\bar{h}}, z_k^f, q_k^f$  – określone jak w (32).

(d) Jeżeli w stanie  $S^e$ :

- na zasobie  $R_k$  nie jest zaalokowany potok  $P_i^h$  i
- semafor zasobu  $R_k$  nie wskazuje na ten proces  $z_k^r \neq P_i^h$ ,

to w stanie  $S^f$ :

- zasób  $R_k$  nadal nie jest zajmowany przez  $P_i^h$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left( a_{i,k}^{e,h} = 0 \right) \wedge (z_k^e \neq P_i^h) \Rightarrow (a_{i,k}^{f,h} = 0) \quad (36)$$

dla  $k = 1 \dots lk$ ;  $i = 1 \dots ln$ ;  $h = 1 \dots ls(i)$ ,

gdzie:  $lk, ln, ls(i), R_k, a_{i,k}^{e,h}$  – określone jak w (32).

(e) Jeżeli w stanie  $S^e$ :

- na zasobie  $R_{\bar{k}}$  jest zaalokowany potok  $P_{\bar{i}}^{\bar{h}}$  i
- semafor zasobu  $R_k$ , do którego potok  $P_{\bar{i}}^{\bar{h}}$  żąda dostępu, nie wskazuje na ten proces  $z_k^r \neq P_{\bar{i}}^{\bar{h}}$ ,

to w stanie  $S^f$ :

- zasób  $R_{\bar{k}}$  nadal jest zajmowany przez  $P_{\bar{i}}^{\bar{h}}$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left( a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \wedge (z_k^e \neq P_{\bar{i}}^{\bar{h}}) \Rightarrow \left( a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \quad (37)$$

dla  $k = 1 \dots lk$ ;  $\forall \bar{k} \in R(k)$ ;  $i = 1 \dots ln$ ;  $h = 1 \dots ls(i)$ ;

gdzie:  $lk, ln, ls(i), \bar{i}, \bar{h}, R_{\bar{k}}, R_k, R(k), a_{\bar{i},\bar{k}}^{e,\bar{h}}, a_{\bar{i},\bar{k}}^{f,\bar{h}}, z_k^f, q_k^f$  – określone jak w (32).

## 2. Warunki dotyczące $SM^l$ poziomu zachowań procesów multimodalnych:

(a) Jeżeli w stanie  $m^l S^e$ :

- zasób  $R_k$  nie jest całkowicie zajęty (zajętość dotyczy tylko  $l$ -tego poziomu zachowań) i
- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  oczekuje na zasobie  $R_{\bar{k}}$  na dostęp do  $R_k$  i
- zasób  $R_{\bar{k}}$  jest zajmowany przez potok  $m^{(l-1)} P_{\bar{i}}^{\bar{h}}$  niższego poziomu ( $l - i$ ) (żądany przez  $m^l P_{\bar{i}}^{\bar{h}}$  do jego wykonania), który w stanie  $m^l S^f$  zajmie zasób  $R_k$ ,

to w stanie  $m^l S^f$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  również zajmie zasób  $R_k$ ,
- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  zmieniają swoją wartość tak, aby wskazywać na kolejny potok określony zgodnie z regułą priorytetowania  $\sigma_k^l$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i,l)} m^l a_{i,k}^{e,h} < C(R_k) \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \right] \Rightarrow [m^l r_{\bar{i}}^{\bar{h}} = 1] \quad (38)$$

$$\begin{aligned} & \left[ \left( m^l r_{\bar{i}}^{\bar{h}} = 1 \right) \wedge \left( m^{(l-1)} a_{\bar{i},\bar{k}}^{e,\bar{i}} = 1 \right) \wedge \left( m^{(l-1)} a_{\bar{i},\bar{k}}^{f,\bar{i}} = 1 \right) \right] \Rightarrow \\ \Rightarrow & \left[ \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 0 \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \wedge \left( m^l z_k^f = s_{k,\bar{d}}^l \right) \wedge \left( m^l q_k^f = \bar{d} \right) \right] \quad (39) \end{aligned}$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:

$lk$  – liczba zasobów  $R$ ;  $lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu zachowań,  $lsm(i, l)$  – liczba potoków procesu  $P_i$ .

$C(R_k)$  – pojemność zasobu  $R_k$ ,

$\bar{i}, \bar{h}$  – indeksy potoku wskazanego przez semafor zasobu  $R_k$ :  $m^l z_k^r = m^l P_{\bar{i}}^{\bar{h}}$ ,

$\bar{i}, \bar{h}$  – indeksy potoku żądanego przez  $m^l P_{\bar{i}}^{\bar{h}}$  do wykonania operacji,

$\bar{k}$  – indeks zasobu  $R_{\bar{k}}$  zajmowanego przez potoki  $m^l P_{\bar{i}}^{\bar{h}}$  i  $m^{(l-1)} P_{\bar{i}}^{\bar{h}}$ .  $R_{\bar{k}}$  – zasób, który poprzedza  $R_k$  w marszrucie procesu  $m^l P_{\bar{i}}$  (i jednocześnie w marszrucie procesu  $m^{(l-1)} P_{\bar{i}}$ ).  $R(k)$  – zbiór zawierający wszystkie zasoby, bezpośrednio poprzedzające  $R_k$  w marszrutach procesów multimodalnych  $m^l P_i$  poziomu  $l$ ,

$m^l r_{\bar{i}}^{\bar{h}}$  – zmienna pomocnicza określająca uzyskanie dostępu (rezerwacji) potoku  $m^l P_{\bar{i}}^{\bar{h}}$  do zasobu  $R_k$ ,  $m^l r_{\bar{i}}^{\bar{h}} = 1$  – oznacza uzyskanie przez potok rezerwacji,  $m^l r_{\bar{i}}^{\bar{h}} = 0$  – oznacza brak możliwości rezerwacji zasobu,

$m^l a_{i,k}^{e,h}$  – alokacja potoku  $m^l P_i^h$  na zasobie  $R_k$  w stanie  $m^l S^e$ ,

- $m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}}$  – alokacja potoku  $m^l P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_{\bar{k}}$  w stanie  $m^l S^e$ ,
- $m^{(l-1)} a_{\bar{i}, \bar{k}}^{e, \bar{i}}$  – alokacja potoku  $m^{(l-1)} P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_{\bar{k}}$  w stanie  $m^l S^e$ ,
- $m^{(l-1)} a_{i, k}^{f, \bar{i}}$  – alokacja potoku  $m^{(l-1)} P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_k$  w stanie  $m^l S^f$ ,
- $m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}}$  – alokacja potoku  $m^l P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_{\bar{k}}$  w stanie  $m^l S^e$ ,
- $m^l a_{\bar{i}, k}^{f, \bar{h}}$  – alokacja potoku  $m^l P_{\bar{i}}^{\bar{h}}$  na zasobie  $R_k$  w stanie  $m^l S^f$ ,
- $m^l z_k^f$  – semafor zasobu  $R_k$  w stanie  $m^l S^f$ , dla  $l$ -tego poziomu zachowań,
- $m^l q_k^f$  – indeks zasobu  $R_k$  w stanie  $m^l S^f$ , dla  $l$ -tego poziomu zachowań,
- $s_{k, \bar{a}}^l$  – element (potok) reguły priorytetowania  $\sigma_k^l$ , przypisanej do zasobu  $R_k$ ,  
występujący w regule zaraz po potoku  $m^l P_{\bar{i}}^{\bar{h}}$ ,  $\bar{a} = \text{modl}(q_k^e + 1; lh(k, l))$ ,  $lh(k, l)$  – długość reguły priorytetowania  $\sigma_k^l$ .

(b) Jeżeli w stanie  $m^l S^e$ :

- zasób  $R_k$  nie jest całkowicie zajęty (zajętość dotyczy tylko  $l$ -tego poziomu zachowań) i
- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  oczekuje na zasobie  $R_{\bar{k}}$  na dostęp do  $R_k$  i
- zasób  $R_{\bar{k}}$  jest zajmowany przez potok  $m^{(l-1)} P_{\bar{i}}^{\bar{h}}$  niższego poziomu ( $l - i$ ) (żądany przez  $m^l P_{\bar{i}}^{\bar{h}}$  do jego wykonania), który w stanie  $m^l S^f$  nie zajmie zasobu  $R_k$ ,

to w stanie  $m^l S^f$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  pozostaje na  $R_{\bar{k}}$ ,
- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  w stanie  $m^l S^f$  mają takie same wartości jak w stanie  $m^l S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i, l)} m^l a_{i, k}^{e, h} < C(R_k) \right) \wedge \left( m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}} = 1 \right) \right] \Rightarrow [m^l r_{\bar{i}}^{\bar{h}} = 1] \quad (40)$$

$$\Rightarrow \left[ \left( m^l r_{\bar{i}}^{\bar{h}} = 1 \wedge \left( m^{(l-1)} a_{\bar{i}, \bar{k}}^{e, \bar{i}} = 1 \right) \wedge \left( m^{(l-1)} a_{i, k}^{f, \bar{i}} = 0 \right) \right) \Rightarrow \left[ \left( m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}} = 1 \right) \wedge \left( m^l a_{\bar{i}, k}^{f, \bar{h}} = 0 \right) \wedge \left( m^l z_k^f = m^l z_k^e \right) \wedge \left( m^l q_k^f = m^l q_k^e \right) \right] \quad (41)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $lk$ ,  $lw(l)$ ,  $lsm(i, l)$ ,  $C(R_k)$ ,  $\bar{i}$ ,  $\bar{h}$ ,  $\bar{i}$ ,  $\bar{h}$ ,  $\bar{k}$ ,  $R_{\bar{k}}$ ,  $R(k)$ ,  $m^l a_{i, k}^{e, h}$ ,  $m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}}$ ,  $m^{(l-1)} a_{\bar{i}, \bar{k}}^{e, \bar{i}}$ ,  $m^{(l-1)} a_{i, k}^{f, \bar{i}}$ ,  $m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}}$ ,  $m^l a_{\bar{i}, k}^{f, \bar{h}}$ ,  $m^l z_k^f$ ,  $m^l q_k^f$ ,  $m^l r_{\bar{i}}^{\bar{h}}$  – określone jak w (38) i (39).

(c) Jeżeli w stanie  $m^l S^e$ :

- zasób  $R_k$  nie jest całkowicie zajęty (zajętość dotyczy tylko  $l$ -tego poziomu zachowań) i

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  oczekuje na zasobie  $R_{\bar{k}}$  na dostęp do  $R_k$  i
- zasób  $R_{\bar{k}}$  nie jest zajmowany przez potok  $m^{(l-1)} P_i^{\bar{h}}$  niższego poziomu ( $l - i$ ) (żądany przez  $m^l P_{\bar{i}}^{\bar{h}}$  do jego wykonania),

to w stanie  $m^l S^f$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  pozostaje na  $R_{\bar{k}}$ ,
- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  mają takie same wartości jak w stanie  $m^l S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i,l)} m^l a_{i,k}^{e,h} < C(R_k) \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \right] \Rightarrow [m^l r_{\bar{i}}^{\bar{h}} = 1] \quad (42)$$

$$\Rightarrow \left[ \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 0 \right) \wedge \left( m^l z_k^f = m^l z_k^e \right) \wedge \left( m^l q_k^f = m^l q_k^e \right) \right] \quad (43)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $lk$ ,  $lw(l)$ ,  $lsm(i,l)$ ,  $C(R_k)$ ,  $\bar{i}$ ,  $\bar{h}$ ,  $i$ ,  $\bar{h}$ ,  $\bar{k}$ ,  $R_{\bar{k}}$ ,  $R(k)$ ,  $m^l a_{i,k}^{e,h}$ ,  $m^l a_{\bar{i},\bar{k}}^{e,\bar{h}}$ ,  $m^{(l-1)} a_{i,\bar{k}}^{e,i}$ ,  $m^l a_{\bar{i},\bar{k}}^{f,\bar{h}}$ ,  $m^l a_{\bar{i},\bar{k}}^{f,\bar{h}}$ ,  $m^l z_k^f$ ,  $m^l q_k^f$ ,  $m^l r_{\bar{i}}^{\bar{h}}$  – określone jak w (38) i (39).

(d) Jeżeli w stanie  $m^l S^e$ :

- zasób  $R_k$  jest całkowicie zajęty (zajętość dotyczy tylko  $l$ -tego poziomu zachowań) i
- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  oczekuje na zasobie  $R_{\bar{k}}$  na dostęp do  $R_k$ ,

to w stanie  $S^f$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  pozostaje na  $R_{\bar{k}}$ ,
- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  mają takie same wartości jak w stanie  $m^l S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( \sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i,l)} m^l a_{i,k}^{e,h} = C(R_k) \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{e,\bar{h}} = 1 \right) \right] \Rightarrow [m^l r_{\bar{i}}^{\bar{h}} = 0] \quad (44)$$

$$[m^l r_{\bar{i}}^{\bar{h}} = 0] \Rightarrow$$

$$\Rightarrow \left[ \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 1 \right) \wedge \left( m^l a_{\bar{i},\bar{k}}^{f,\bar{h}} = 0 \right) \wedge \left( m^l z_k^f = m^l z_k^e \right) \wedge \left( m^l q_k^f = m^l q_k^e \right) \right] \quad (45)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $lk, lw(l), lsm(i, l), C(R_k), \bar{i}, \bar{h}, \bar{k}, R_{\bar{k}}, R(k), m^l a_{i,k}^{e,h}, m^l a_{\bar{i},\bar{k}}^{e,\bar{h}}, m^l a_{\bar{i},\bar{k}}^{f,\bar{h}}, m^l a_{\bar{i},k}^{f,\bar{h}}, m^l z_k^f, m^l q_k^f, m^l r_{\bar{i}}^{\bar{h}}$  – określone jak w (38) i (39).

(e) Jeżeli w stanie  $m^l S^e$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  nie oczekuje na dostęp do  $R_k$  i
- potok  $m^l P_{\bar{i}}^{\bar{h}}$  nie zajmuje zasobu  $R_k$ ,

to w stanie  $m^l S^f$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  nie zajmuje również zasobu  $R_k$ ,
- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  mają takie same wartości jak w stanie  $m^l S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\begin{aligned} & \left[ \left( m^l a_{\bar{i},\bar{k}}^{e,\bar{h}} = 0 \right) \wedge \left( m^l a_{\bar{i},k}^{e,\bar{h}} = 0 \right) \right] \Rightarrow \\ \Rightarrow & \left[ \left( m^l a_{\bar{i},k}^{f,\bar{h}} = 0 \right) \wedge \left( m^l z_k^f = m^l z_k^e \right) \wedge \left( m^l q_k^f = m^l q_k^e \right) \right] \end{aligned} \quad (46)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ .

Z kolei, jeżeli w stanie  $S^e$ :

- potok  $m^l P_{\bar{i}}^{\bar{h}}$  wskazany przez semafor  $m^l z_k^e$  potok  $m^l P_{\bar{i}}^{\bar{h}}$  zajmuje zasób  $R_k$ ,

to w stanie  $m^l S^f$ :

- semafor  $m^l z_k^f$  oraz jego indeks  $m^l q_k^f$  mają takie same wartości jak w stanie  $m^l S^e$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left[ \left( m^l a_{\bar{i},\bar{k}}^{e,\bar{h}} = 0 \right) \wedge \left( m^l a_{\bar{i},k}^{e,\bar{h}} = 1 \right) \right] \Rightarrow \left[ \left( m^l z_k^f = m^l z_k^e \right) \wedge \left( m^l q_k^f = m^l q_k^e \right) \right] \quad (47)$$

dla  $k = 1 \dots lk$  i  $\forall R_{\bar{k}} \in R(k)$ ,

gdzie:  $\bar{i}, \bar{h}, \bar{k}, R_{\bar{k}}, R(k), m^l a_{i,k}^{e,h}, m^l a_{\bar{i},\bar{k}}^{e,\bar{h}}, m^l a_{\bar{i},k}^{f,\bar{h}}, m^l z_k^f, m^l q_k^f$  – określone jak w (38) i (39).

(f) Jeżeli w stanie  $m^l S^e$ :

- na zasobie  $R_k$  nie jest zaalokowany potok  $m^l P_i^h$  i
- semafor tego zasobu nie wskazuje na ten potok  $m^l z_k^r \neq m^l P_i^h$ ,

to w stanie  $m^l S^f$ :

- zasób  $R_k$  nadal nie jest zajmowany przez  $m^l P_i^h$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left( m^l a_{i,k}^{e,h} = 0 \right) \wedge \left( m^l z_k^e \neq m^l P_i^h \right) \Rightarrow \left( m^l a_{i,k}^{f,h} = 0 \right) \quad (48)$$

dla  $k = 1 \dots lk; i = 1 \dots lw(l); h = 1 \dots lsm(i, l)$ ,

gdzie:  $lk, lw(l), lsm(i, l), \bar{i}, \bar{h}, \bar{k}, R_{\bar{k}}, R(k), m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}}, m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}}$  – określone jak w (38) i (39).

(g) Jeżeli w stanie  $m^l S^e$ :

- na zasobie  $R_{\bar{k}}$  jest zaalokowany potok  $m^l P_{\bar{i}}^{\bar{h}}$  i
- semafor zasobu  $R_k$ , do którego potok żąda dostępu nie wskazuje na ten proces  $m^l z_k^r \neq m^l P_{\bar{i}}^{\bar{h}}$ ,

to w stanie  $m^l S^f$ :

- zasób  $R_{\bar{k}}$  nadal jest zajmowany przez  $m^l P_{\bar{i}}^{\bar{h}}$ .

W zapisie formalnym warunek ten przyjmuje postać:

$$\left( m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}} = 1 \right) \wedge \left( m^l z_k^e \neq m^l P_{\bar{i}}^{\bar{h}} \right) \Rightarrow \left[ \left( m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}} = 1 \right) \wedge \left( m^l r_{\bar{i}}^{\bar{h}} = 0 \right) \right] \quad (49)$$

dla  $k = 1 \dots lk; \forall \bar{k} \in R(k); i = 1 \dots lw(l); h = 1 \dots lsm(i, l)$ ,

gdzie:  $lk, lw(l), lsm(i, l), \bar{i}, \bar{h}, \bar{k}, R_{\bar{k}}, R(k), m^l a_{\bar{i}, \bar{k}}^{e, \bar{h}}, m^l a_{\bar{i}, \bar{k}}^{f, \bar{h}}, m^l r_{\bar{i}}^{\bar{h}}$  – określone jak w (38) i (39).

(h) Na danym ( $l$ -tym) poziomie zachowań co najmniej jeden potok  $m^l P_{\bar{i}}^{\bar{h}}$  powinien mieć uzyskany dostęp do żądanego zasobu ( $m^l r_{\bar{i}}^{\bar{h}} = 1$ ). Spełnienie tego warunku, gwarantując potokom rezerwację do żądanych przez nich zasobów, zgodnie z Twierdzeniem 2, chroni system przed blokadą. W zapisie formalnym warunek ten przyjmuje postać:

$$\sum_{i=1}^{lw(l)} \sum_{h=1}^{lsm(i, l)} m^l r_{\bar{i}}^{\bar{h}} > 0, \quad (50)$$

gdzie:  $m^l r_{\bar{i}}^{\bar{h}}$  – określone jak w (38) i (39).

Predykaty (32) – (49) mają charakter implikacji a ich konkluzje obejmują wszystkie potencjalne warianty wartości alokacji, semaforów i indeksów. Oznacza to, że w każdym stanie  $S^e$  systemu możliwa jest tylko jedna jego reakcja  $S^f$  określona przez funkcję przejścia  $S^f = \delta^{lp}(S^e)$ .

Łatwo zauważyć, że funkcję  $\delta^{lp}$  odwzorowuje zbiór par  $(S^e, S^f)$ ,  $S^e, S^f \in \mathbb{S}^{lp}$  spełniających zbiór predykatów (32) – (50) oznaczany dalej jako  $\Lambda^{lp}$ :  $\delta^{lp} = \{(S^e, S^f) \mid S^e, S^f \in \mathbb{S}^{lp}, S^e \neq S^f, \Lambda^{lp}\}$ . Funkcja ta nie jest funkcją różnowartościową, ani funkcją typu „na”, tzn. w zbiorze  $\mathbb{S}^{lp}$  mogą istnieć stany  $S^f$  nieosiągalne z żądanych innych stanów i odwrotnie, nie dla każdego argumentu  $S^e$  funkcja posiada wartość  $S^f$  (tzn. w zbiorze  $\mathbb{S}^{lp}$  mogą istnieć stany  $S^e$ , z których żaden inny stan nie jest osiągalny).

Podobnie jak w przypadku reprezentacji struktury (18) i stanu (19), zbiór predykatów  $\Lambda^{lp}$  jest złożony z podzbiorów odpowiadających kolejnym poziomom zachowań:  $SL$  i  $SM^l$ . Przykładowo funkcja opisująca przejścia między stanami systemu z dwoma poziomami zachowań  $SL$  i  $SM^1$  ma inną postać niż funkcja opisująca przejścia syste-

mu z trzema poziomami  $SL, SM^1, SM^2$ . W pierwszym przypadku zbiór  $\Lambda^{lp}$  zawiera predykaty (32) – (37) opisujące związek na poziomie lokalnym oraz predykaty (38) – (50) opisujące związek na poziomie multimodalnym (przyjmuje się w tym przypadku że:  $l = 1$ ). W drugim przypadku funkcja przejścia opisuje związek na dwóch poziomach multimodalnych, dlatego też zbiór  $\Lambda^{lp}$  zawiera dwa podzbiory predykatów typu (38) – (50) – jeden dla  $l = 1$  i drugi dla  $l = 2$ . Należy zwrócić uwagę że, w podzbiórach  $\Lambda^{lp}$  podobnie jak w przypadku struktury  $SC^{lp}$  dochodzi do wzajemnej zależności między poziomami zachowań. Predykaty opisujące związki poziomu  $l = 2$  wykorzystują w swym opisie procesy niższego poziomu (predykaty typu (39), (41) i (43)). Predykaty opisujące związki z poziomu  $l = 1$  wykorzystują z kolei procesy lokalne (w predykatkach (39), (41) i (43) procesy multimodalne z  $l = 0$  traktowane są jako procesy lokalne). Jak widać hierarchiczność struktury przenosi się kolejno na wszystkie elementy zachowania systemu: stan, funkcja przejścia i opisana dalej przestrzeń stanów.

Zbiór predykatów determinujący postać funkcji przejścia  $\delta^{lp}$  formalnie definiowany jest następująco:

$$\Lambda^{lp} = \{\lambda^0 \cup \dots \cup \lambda^l \cup \dots \cup \lambda^{lp}\}, \quad (51)$$

gdzie:

- $\lambda^0$  – zbiór predykatów (32) – (37) określających związki na poziomie lokalnym,
- $\lambda^l$  – zbiór predykatów (38) – (50) określających związki na poziomie multimodalnym kolejno dla  $l = 1 \dots lp$ ,  $A \cup B$  – suma mnogościowa zbiorów  $A, B$ .

Analizując postać predykatów (38) – (50) można dostrzec, że zakładają one wzajemne wykluczanie się procesów i odnoszą się do zachowania systemów typu  $MW_{1,1}, MW_{1,2}$ , (tab. 2.1). Z kolei, usunięcie z predykatów elementów opisujących semafor i indeksy umożliwia opis zachowania systemów typu  $MW_{2,1}, MW_{2,2}$ . Przedstawiony zbiór predykatów można również uprościć do postaci, w której każdy zasób ma pojemność równą jeden ( $C(R_k) = 1$ ) lub też rozszerzyć tak aby uwzględniały czasy wykonania operacji większe niż jednostkowe ( $t_{i,j} \geq 1, m^l t_{i,j} \geq 1$ ).

### 2.2.2. Digrafy zachowań

Dla elementów funkcji przejścia  $S^f = \delta^{lp}(S^e)$ , wprowadzono następujące pojęcia:

- stan  $S^e$  (argument funkcji  $\delta^{lp}$ ) nazywany jest **rodzicem**  $S^f$ ,
- stan  $S^f$  (wartość funkcji  $\delta^{lp}$ ) osiągnąony z  $S^e$  nazywany jest **potomkiem**  $S^e$ .

Niech  $PA^r$  oznacza **zbiór rodziców** stanu  $S^r \in \mathbb{S}^{lp}$ :  $PA^r = \{S^e \mid S^e \rightarrow S^r; S^e \in \mathbb{S}^{lp}\}$  a  $CH^r$  oznacza **zbiór potomków** stanu  $S^r$ :  $CH^r = \{S^e \mid S^r \rightarrow S^e; S^e \in \mathbb{S}^{lp}\}$ ,  $PA^r, CH^r \subset \mathbb{S}^{lp}$ ,  $PA^r \cap CH^r = \emptyset$  (zbiory te nie zawierają wspólnych elementów ze względu na warunek:  $S^r \neq \delta^{lp}(S^r)$  – co oznacza, że potomek stanu nie może być jednocześnie też jego rodzicem). Konstatacja ta prowadzi do pierwszej (dość oczywistej) własności przestrzeni  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$ :

*Własność 1*

Każdy stan dopuszczalny  $S^r \in \mathbb{S}^{lp}$  ma co najwyżej jednego potomka ( $|CH^r| \leq 1$ ).

Dowód: Niech możliwa będzie sytuacja, w której pewien stan  $S^r \in \mathbb{S}^{lp}$  posiada dwóch różnych potomków  $S^{f_1}$  i  $S^{f_2}$ :  $S^{f_1} \neq S^{f_2}$ ;  $S^{f_1}, S^{f_2} \in \mathbb{S}^{lp}$ . Oznacza to, że każdy z nich jest wartością funkcji przejścia dla argumentu  $S^r$ :  $S^{f_1} = \delta^{lp}(S^r)$  i  $S^{f_2} = \delta^{lp}(S^r)$ . Podstawiając do wyrażenia:  $S^{f_1} \neq S^{f_2}$  wartości funkcji przejścia  $\delta^{lp}(S^r)$  uzyskuje się następującą sprzeczność:  $\delta^{lp}(S^r) \neq \delta^{lp}(S^r)$ . Oznacza to, że stan  $S^r$  nie może mieć jednocześnie dwóch (lub więcej) potomków, co potwierdza słuszność Własności 1.

□

Nic nie stoi, z kolei, na przeszkodzie aby stan  $S^r$  posiadał wieloelementowy zbiór rodziców  $PA^r$ :  $|PA^r| \geq 0$ .

Własność 1 determinuje postać zbioru łuków  $\mathbb{E}^{lp}$  łączących wierzchołki przestrzeni  $\mathcal{P}$ . Jeden potomek i wielu rodziców stanu oznacza, że wierzchołek może być związany tylko z jednym łukiem wychodzącym i wieloma łukami dochodzącymi. Własność ta jest spełniona w prezentowanych na rys. 2.14, 2.15 i 2.20 przestrzeniach stanów  $\mathcal{P}$ . Jak widać każdy ze stanów przestrzeni charakteryzuje co najwyżej jeden łuk wychodzący.

Szczególną grupę stanów przestrzeni z rys. 2.14, 2.15 i 2.20 tworzą stany, które nie posiadają łuków dochodzących, inaczej mówiąc są to stany nie posiadające rodziców. Na rysunkach określane są one jako **stany początkowe**, głównie ze względu na to, że stanowią one początek każdego przebiegu z przestrzeni  $\mathcal{P}$ .

Formalnie, **stan początkowy**  $S^r$  to stan dopuszczalny  $S^r \in \mathbb{S}^{lp}$ , dla którego zbiór rodziców jest zbiorem pustym:  $PA^r = \emptyset$ . Zbiór wszystkich stanów początkowych należących do zbioru  $\mathbb{S}^{lp}$  oznaczany jest jako:  $SB = \{S^r | S^r \in \mathbb{S}^{lp}, PA^r = \emptyset\}$ .

Niech  $CH$  oznacza zbiór wszystkich stanów ze zbioru  $\mathbb{S}^{lp}$  będących potomkami:

$$CH = \bigcup_{\forall S^r \in \mathbb{S}^{lp}} CH^r, \quad (52)$$

gdzie:  $CH^r$  – zbiór potomków stanu  $S^r$ .

W rozważanym kontekście związek między zbiorem stanów początkowych  $SB$  a zbiorem  $\mathbb{S}^{lp}$  charakteryzuje następująca zależność:

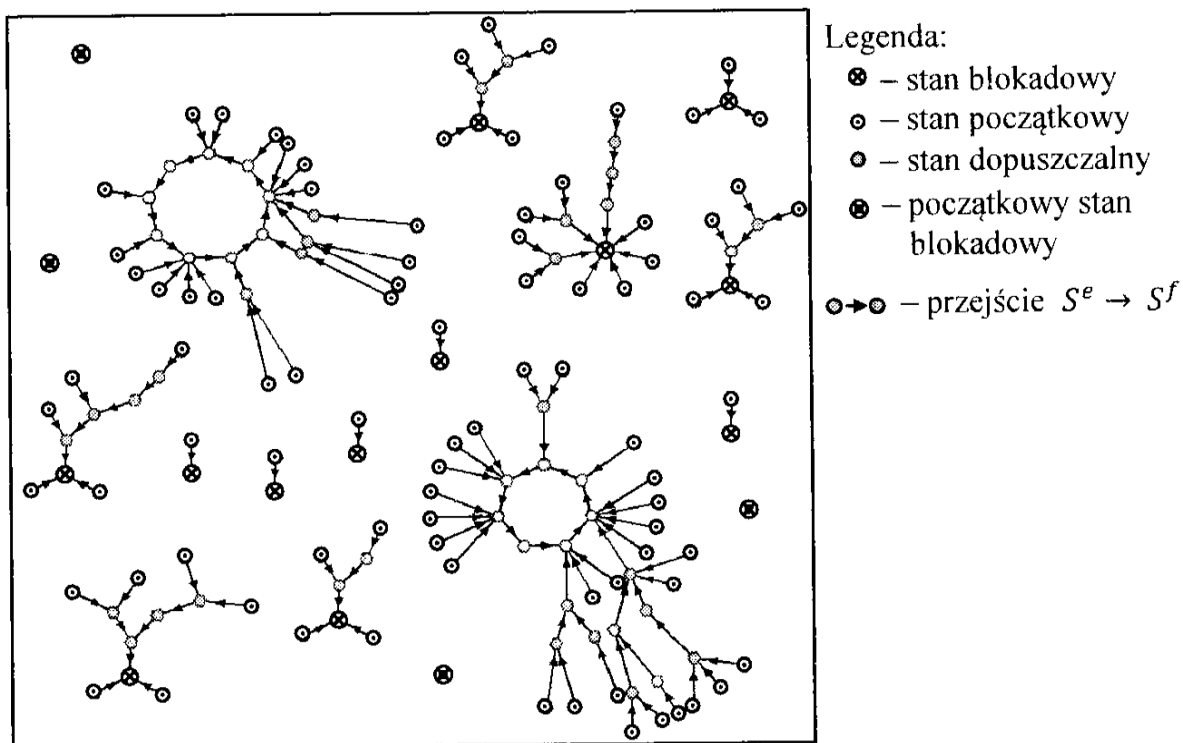
$$SB = \mathbb{S}^{lp} \setminus CH. \quad (53)$$

Dysponując tylko zbiorem  $SB$  oraz funkcją przejścia  $\delta^{lp}$  można wyznaczyć każdy przebieg wchodzący w skład tej przestrzeni. Innymi słowy przestrzeń ta może być scharakteryzowana tylko przez parę  $(SB, \delta^{lp})$ . Oznacza to, że w celu wyznaczenia zachowania SWMPC nie jest konieczna znajomość całego zbioru  $\mathbb{S}^{lp}$  wystarczy znać tylko stany wchodzące w skład zbioru  $SB$ . Tak więc umiejętność efektywnego wyznaczenia tego zbioru staje się kluczowa dla oceny zachowań SWMPC.

O ile stany początkowe rozpoczynają przebiegi przejściowe, o tyle stany blokadowe kończą je (oczywiście pod warunkiem, że system nie osiąga cyklicznego przebie-



gu ustalonego). Stany blokadowe charakteryzują się tym, że nie posiadają łuków wychodzących, czyli są to stany nie posiadające potomków.



Rys. 2.20. Przykładowa przestrzeń stanów  $\mathcal{P}$  SWMPC (Bocewicz et al. 2010)

Formalnie, **stan blokadowy**  $S^r$  to stan dopuszczalny  $S^r \in \mathbb{S}^{lp}$ , dla którego zbiór potomków jest zbiorem pustym:  $CH^r = \emptyset$ . Zbiór wszystkich stanów blokadowych ze zbioru  $\mathbb{S}^{lp}$  oznaczany jest jako:  $SD = \{S^r | S^r \in \mathbb{S}^{lp}, CH^r = \emptyset\}$ .

Uznanie stanu blokadowego jako stanu bez potomków jest konsekwencją Twierdzenia 2. Blokada procesów uniemożliwia osiągnięcie przez systemu nowego stanu (potomka), a zatem traktowana jest jako stan bez potomków. Przykład takiej sytuacji ilustruje harmonogram z rys. 2.12.

Niech  $PA$  oznacza zbiór wszystkich stanów ze zbioru  $\mathbb{S}^{lp}$  będących rodzicami:

$$PA = \bigcup_{\forall S^r \in \mathbb{S}^{lp}} PA^r, \quad (54)$$

gdzie:  $PA^r$  – zbiór rodziców stanu  $S^r$ .

Podobnie jak w przypadku stanów początkowych, zbiór  $SD$  definiuje poniższa zależność:

$$SD = \mathbb{S}^{lp} \setminus PA. \quad (55)$$

Tak zdefiniowane zbiory stanów blokadowych  $SD$  (55) i początkowych  $SB$  (52) prowadzą do kolejnej grupy, tak zwanych **początkowych stanów blokadowych**  $SBD$ :

$$SBD = SB \cap SD. \quad (56)$$

Są to stany będące jednocześnie stanami początkowymi i blokadowymi. Nie posiadają one potomków ani rodziców, czyli nie są związane z żadnym łukiem  $\mathbb{E}^{lp}$ . Stany tego typu występują w SWMPC bardzo rzadko, przykładowo brak jest tego typu stanów w przestrzeniach z rys. 2.14 i 2.15, ale występują już w przestrzeni z rys. 2.17. Stan systemu, który jest jednocześnie stanem blokadowym oraz początkowym nie ma właściwie żadnego znaczenia praktycznego, dlatego też stany  $SBD$  traktowane są jako pewna osobliwość w zachowaniu SWMPC i nie będą dalej szczegółowo opisywane.

Często ze względów praktycznych wyróżnia się kolejną grupę stanów dopuszczalnych, tak zwane stany startowe. **Stan startowy** to taki, w którym każdy potok procesu lokalnego rozpoczyna operacje na zajmowanym przez siebie zasobie. Pojęcie stanu początkowego odnosi się więc tylko do realizacji procesów lokalnych. Założenie to przyjęte zostało głównie dlatego, że procesy te zwykle reprezentują środki transportu, które w momencie rozpoczęcia pracy systemu powinny również rozpocząć realizację swoich operacji. W razie potrzeby założenie te łatwo rozszerzyć na wyższe poziomy zachowań. O stanach tego typu wspomniano już w przypadku opisu przebiegu z rys. 2.11, w przebiegu tym brakuje stanów startowych. Zbiór stanów startowych ze zbioru  $\mathbb{S}^{lp}$  jest oznaczany jako  $ST$ .

### Własność 2

*Każdy stan początkowy jest stanem startowym,  $SB \subseteq ST$ .*

Dowód: Niech istnieje stan początkowy, który wbrew Własności 2 nie jest stanem startowym. Oznacza to, że istnieją potoki procesów lokalnych, które są w trakcie kontynuowania rozpoczętej wcześniej (w poprzednich stanach) operacji lub są w trakcie oczekiwania po ukończeniu wcześniejszych operacji. W obu przypadkach wymagane jest istnienie stanów poprzedzających. Zgodnie z przyjętymi założeniami każdy stan początkowy  $S^r \in SB$  nie posiada rodzica czyli nie ma stanów poprzedzających, a zatem w stanie  $S^r$  nie mogą występować tego typu potoki. Oznacza to, że w stanie początkowym  $S^r$  wszystkie potoki procesów lokalnych rozpoczynają swoje operacje. Tak więc, stan początkowy  $S^r$  jest stanem startowym, co kończy dowód.  $\square$

Ciekawym następstwem Własności 2 oraz wyrażenia (56) jest to, że każdy stan ze zbioru  $SBD$  jest również traktowany jako stan startowy. W praktyce stan tego typu objawia się tym, że każdy z procesów rozpoczyna swoją operację w stanie blokadowym.

Definicja funkcji przejścia  $\delta^{lp}$  oraz podział stanów na stany początkowe  $SB$  i blokadowe  $SD$  pozwala na wprowadzenie formalnej definicji przebiegu SWMPC.

**Przebieg** SWMPC jest sekwencją  $D = (S^{d_1}, \dots, S^{d_i}, S^{d_{i+1}}, \dots, S^{d_{ld}})$ , różnych stanów dopuszczalnych:  $S^{d_i}, S^{d_{i+1}} \in \mathbb{S}^{lp}$ , w której każda kolejna para stanów spełnia wyrażenie:  $S^{d_{i+1}} = \delta^{lp}(S^{d_i})$ ,  $i = 1 \dots (ld - 1)$ . Odwzorowaniem tak definiowanego przebiegu jest digraf postaci:

$$G(D) = (V(D), E(D)), \quad (57)$$

gdzie:

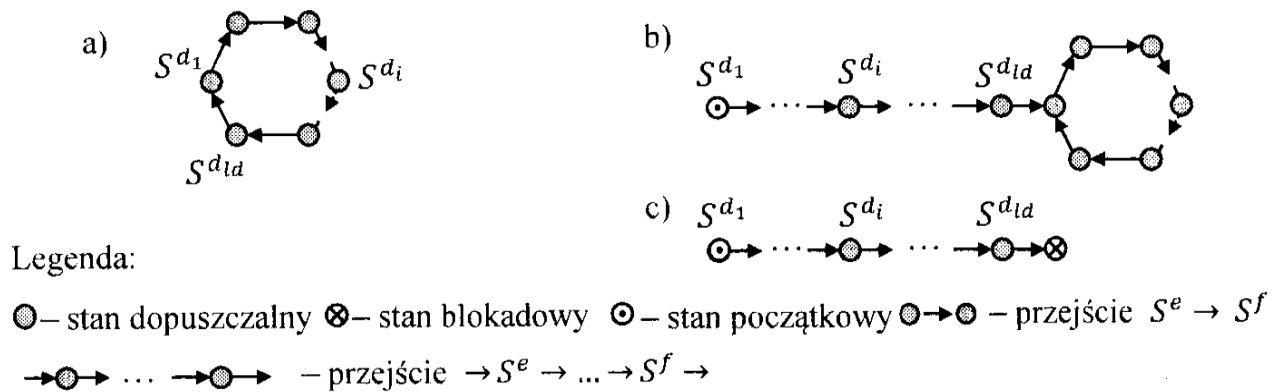
$$V(D) = \{S^{d_i} \mid i = 1 \dots ld\} - \text{zbiór stanów przebiegu } D,$$

$E(D) = \{(S^{d_i}, S^{d_{i+1}}) \mid i = 1 \dots (ld - 1); S^{d_{i+1}} = \delta^{lp}(S^{d_i})\}$  – zbiór łuków łączących stany przebiegu  $D$ .

Wśród przebiegów SWMPC można wyróżnić dwa podstawowe ich rodzaje:

- **cykliczne przebiegi ustalone**  $D_C$  – w których pierwszy stan  $S^{d_1}$  jest osiągalny z ostatniego  $S^{d_{ld}}$  t.j.:  $S^{d_1} = \delta^{lp}(S^{d_{ld}})$ .  
 $DC$  oznacza dalej zbiór wszystkich ustalonych przebiegów cyklicznych  $D_C$  utworzonych ze stanów zbioru  $\mathbb{S}^{lp}$ ,
- **przebiegi przejściowe**  $D_T$  – w których pierwszy stan jest stanem początkowym  $S^{d_1} \in SB$ , a ostatni stanem prowadzącym do blokady  $\delta^{lp}(S^{d_{ld}}) \in SD$  lub dowolnego stanu przebiegu cyklicznego:  $\delta^{lp}(S^{d_{ld}}) \in V(D_C)$ , gdzie:  $V(D_C)$  – zbiór stanów cyklicznego przebiegu ustalonego  $D_C$ .  
 $DT$  oznacza dalej zbiór wszystkich przebiegów przejściowych  $D_T$  utworzonych ze stanów zbioru  $\mathbb{S}^{lp}$ . Zakłada się, że stany  $V(D_T)$  tworzące przebieg przejściowy  $D_T$  nie tworzą przebiegów cyklicznych (patrz Własność 3).

Cykliczny przebieg ustalony ilustruje rys. 2.21a, kolejne dwa przebiegi przejściowe, z których pierwszy prowadzący do cyklu ilustruje rys. 2.21b, a drugi prowadzący do blokady ilustruje rys. 2.21c.



**Rys. 2.21.** Przebiegi SWMPC: ustalony przebieg cykliczny  $D_C$  a), przebieg przejściowy prowadzący do cyklu  $D_T$  b), przebieg przejściowy prowadzący do blokady  $D_T$  c)

Reasumując, każdy stan występujący w zbiorze  $\mathbb{S}^{lp}$  jest elementem jednego z następujących przejawów zachowań: **przebiegu przejściowego, ustalonego przebiegu cyklicznego oraz stanu blokadowego**. Stany przestrzeni  $\mathcal{P}$  spełniają zatem następujące własności:

### Własność 3

Przebiegi cykliczne  $D_C \in DC$  i przebiegi przejściowe  $D_T \in DT$  są rozłączne,  $\forall D_C \in DC$  i  $\forall D_T \in DT: V(D_C) \cap V(D_T) = \emptyset$  (gdzie:  $V(D_C)$ ,  $V(D_T)$  – odpowiednio zbiory stanów cyklicznego przebiegu ustalonego  $D_C$  i przebiegu przejściowego  $D_T$ ).

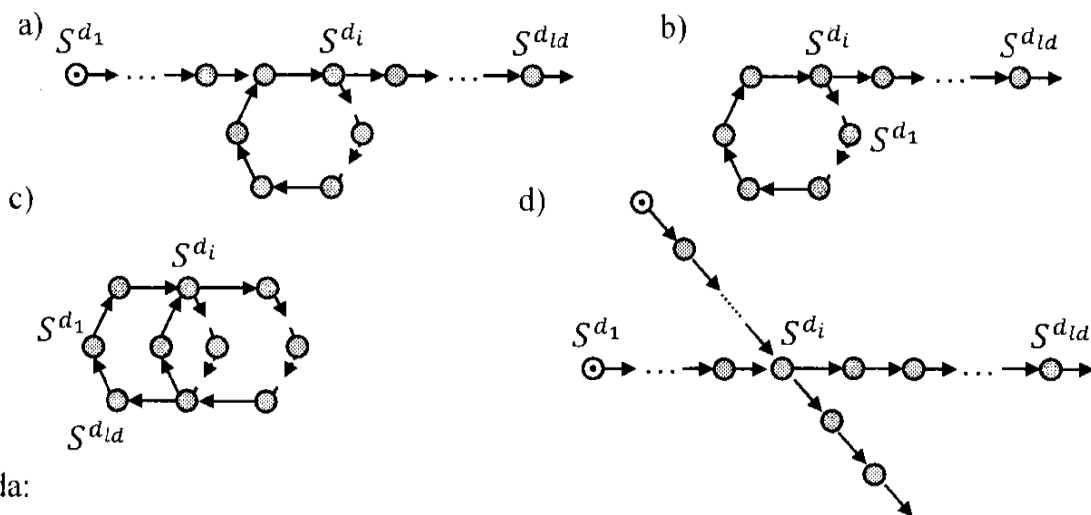
Dowód: Niech możliwa jest sytuacja, w której istnieją dwa przebiegi: przejściowy  $D_T$  i cykliczny  $D_C$ , posiadające wspólne stany tak jak na przykład na rys. 2.22a i 2.22b. Oznacza to, że musi istnieć wspólny stan  $S^{d_i} \in V(D_C) \cap V(D_T)$  posiadający co najmniej dwóch potomków. Jest to sprzeczne z Własnością 1, stwierdzającą, że: *Każdy*

stan dopuszczalny  $S^r \in S^{lp}$  ma co najwyżej jednego potomka ( $|CH^r| \leq 1$ ), zatem rozważane przebiegi są rozłączne. □

#### Własność 4

Przebiegi cykliczne  $D_C \in DC$  są rozłączne,  $\forall D_{C,1}, D_{C,2} \in DC, D_{C,1} \neq D_{C,2}: V(D_{C,1}) \cap V(D_{C,2}) = \emptyset$  (gdzie:  $V(D_{C,1}), V(D_{C,2})$  – odpowiednio zbiory stanów cyklicznych przebiegów ustalonych  $D_{C,1}, D_{C,2}$ ).

Dowód: Niech możliwa jest sytuacja, w której dwa przebiegi cykliczne posiadają wspólny stan  $S^{d_i}$ . W takim przypadku stan  $S^{d_i}$  musi posiadać co najmniej dwóch potomków (rys. 2.22c). Jest to sprzeczne z Własnością 1, zatem przebiegi cykliczne nie posiadają wspólnych stanów. □



Legenda:

● – stan dopuszczalny ⊗ – stan blokadowy ⊙ – stan początkowy ● → ● – przejście  $S^e \rightarrow S^f$   
 → ● →    → ● → – przejście  $\rightarrow S^e \rightarrow \dots \rightarrow S^f \rightarrow$

Rys. 2.22. Przykłady przebiegów niedopuszczalnych

#### Własność 5

Jeżeli dwa różne przebiegi przejściowe  $D_{T,1}, D_{T,2} \in DT$  posiadają wspólny stan  $S^{d_i}$ , to wspólne są również stany kolejne:  $S^{d_{i+1}}, \dots, S^{d_{ld}}$ .

Dowód: Zgodnie z Własnością 1, stan  $S^{d_i}$  może mieć co najwyżej jednego potomka  $S^{d_{i+1}}$ . Jeżeli więc stan  $S^{d_{i+1}}$  należy jednocześnie do  $D_{T,1}, D_{T,2}$  to każdy kolejny stan  $S^{d_{i+k}}$  jest wspólny dla tych przebiegów. Przypadek, w którym kolejne stany  $S^{d_{i+1}}, \dots, S^{d_{ld}}$  nie są wspólne dla obu przebiegów (rys. 2.22d) wymaga aby stan  $S^{d_i}$  miał co najmniej dwóch potomków, co jest sprzeczne z Własnością 1. □

Własności 3 – 5 prowadzą do następujących spostrzeżeń dotyczących zachowania SWMPC:

- Pomędzy przebiegami cyklicznymi nie mogą występować przebiegi przejściowe (Własność 3).

- Raz osiągnięty przebieg cykliczny nie może zostać zmieniony (Własność 4).
- Przebiegi przejściowe o wspólnych stanach prowadzą do tego samego przebiegu kończącego się albo blokadą albo cyklicznym przebiegiem ustalonym (Własność 5).

Dla zbioru stanów dopuszczalnych  $\mathbb{S}^{lp}$  słuszne jest również następujące twierdzenie:

### Twierdzenie 3

Jeżeli w niepustym zbiorze stanów dopuszczalnych  $\mathbb{S}^{lp}$  istnieje pusty zbiór stanów początkowych  $SB = \emptyset$ , to wszystkie stany tego zbioru tworzą ustalone przebiegi cykliczne  $D_C$ :

$$\mathbb{S}^{lp} = \bigcup_{D_C \in DC} V(D_C), \quad (58)$$

gdzie:  $V(D_C)$  – zbiór stanów cyklicznego przebiegu ustalonego  $D_C$ ,  $DC$  – zbiór wszystkich cyklicznych przebiegów ustalonych  $D_C$  utworzonych ze stanów zbioru  $\mathbb{S}^{lp}$ .

Dowód: Niech w zbiorze  $\mathbb{S}^{lp}$ , w którym  $SB = \emptyset$  istnieje stan dopuszczalny  $S^r$  nie będący stanem należącym do żadnego przebiegu cyklicznego. Stan  $S^r$  nie może należeć do zbioru  $SBD$  (początkowych stanów blokadowych), gdyż musiałby być stanem początkowym, a takie w zbiorze  $\mathbb{S}^{lp}$  nie występują. Tak więc stan  $S^r$  musi należeć do pewnego przebiegu przejściowego  $D_T = (S^{d_1}, \dots, S^r, \dots, S^{d_{id}})$ . Nie jest to jednak możliwe gdyż każdy dopuszczalny przebieg przejściowy rozpoczyna się od stanu początkowego (przebiegi przejściowe między przebiegami cyklicznymi nie są możliwe – Własność 3). Brak stanów początkowych oznacza zatem brak przebiegów przejściowych. Zatem jako jedyne przebiegi dopuszczalne pozostają przebiegi cykliczne. Stan  $S^r$  należy zatem do przebiegu cyklicznego, co kończy dowód.  $\square$

Twierdzenie 3 odgrywa istotną rolę przy wyznaczaniu przebiegów cyklicznych poprzez iteracyjne usuwanie stanów początkowych ze zbioru  $\mathbb{S}^{lp}$ . W wyniku usunięcia wszystkich stanów początkowych, zbiór  $\mathbb{S}^{lp}$  zawiera ostatecznie tylko stany należące do przebiegów cyklicznych. Metoda szczegółowo opisująca to postępowanie przedstawiona została w rozdziale 3.

Przykładem przestrzeni spełniającej Własności 3 – 5 jest przestrzeń z rys. 2.20. Warto zwrócić uwagę (zgodnie z Własnością 5), że przebiegi przejściowe o wspólnych stanach skutkują tym samym przejawem zachowania. Można je zatem łączyć tworząc spójne digrafy. Wśród tej klasy digrafów wyróżnia się digrafy przebiegów cyklicznych oraz digrafy stanów blokadowych.

Niech  $DT(D_C)$  oznacza zbiór wszystkich przebiegów przejściowych prowadzących do wspólnego przebiegu cyklicznego  $D_C \in DC$ . Digraf skierowany typu wir (59) zawiera wszystkie przebiegi przejściowe prowadzące do przebiegu cyklicznego  $D_C$ .

$$W(D_C) = G(D_C) \dot{\cup} \left( \bigcup_{\forall D_T \in DT(D_C)} G(D_T) \right), \quad (59)$$

gdzie:

$D_C$  – cykliczny przebieg ustalony,  $V(D_C) \subseteq \mathbb{S}^{lp}$  – zbiór stanów (wierzchołków digrafu) przebiegu  $D_C$ ,  $G(D_C)$  – digraf (57) przebiegu  $D_C$ ,

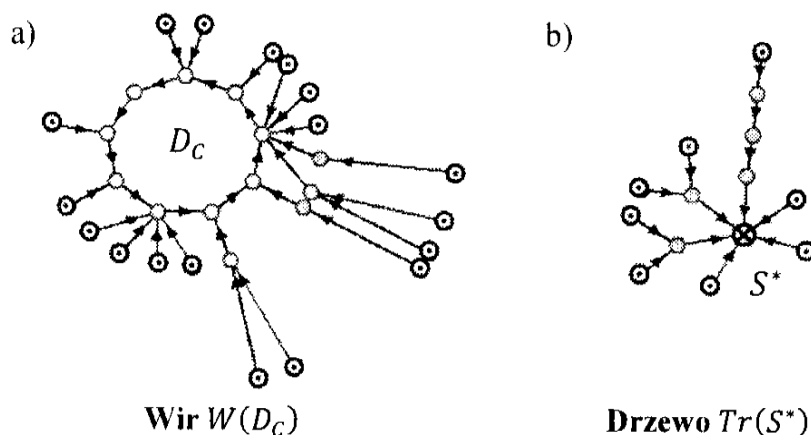
$D_T$  – przebieg przejściowy prowadzący do przebiegu  $D_C$ ,  $D_T \in DT(D_C)$ , gdzie:  $DT(D_C)$  – zbiór przebiegów przejściowych prowadzących do  $D_C$ ,  $G(D_T)$  – digraf (57) przebiegu  $D_T$ ,

$G_1 \dot{\cup} G_2$  – suma digrafów (Ross i Wrigh 2008)  $G_1 = (V_1, E_1)$  i  $G_2 = (V_2, E_2)$ :  
 $G_1 \dot{\cup} G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ ,

$$\bigcup_{G_i \in G^*} G_i = G_1 \dot{\cup} G_2 \dot{\cup} \dots \dot{\cup} G_a ,$$

dla  $G^* = \{G_1, G_2, \dots, G_a\}$ .

Przykład wiru został przedstawiony na rys. 2.23a. Jak widać rozpoczęcie realizacji procesów z jakiegokolwiek stanu początkowego należącego do tego wiru prowadzi w konsekwencji do przebiegu cyklicznego  $D_C$ .



Legenda:

⊙ – stan dopuszczalny ⊗ – stan blokadowy ⊙ – stan początkowy ⊙ → ⊙ – przejście  $S^e \rightarrow S^f$

Rys. 2.23. Digrafy zachowań: wir a), drzewo b)

W analogiczny sposób można definiować digrafy przebiegów prowadzących do wspólnego stanu blokadowego. Niech  $DT(S^*)$  oznacza zbiór wszystkich przebiegów przejściowych prowadzących do wspólnego stanu blokadowego  $S^* \in SD$ . Digraf typu drzewo (60) reprezentuje wszystkie przebiegi SWMPC prowadzące do **wspólnego stanu blokadowego**  $S^*$ .

$$Tr(S^*) = \bigcup_{\forall D_T \in DT(S^*)} G(D_T) , \quad (60)$$

gdzie:

$D_T$  – przebieg przejściowy prowadzący do stanu blokadowego  $S^*$ ,  $D_T \in DT(S^*)$ ,  
gdzie:  $DT(S^*)$  – zbiór wszystkich przebiegów przejściowych prowadzących do stanu blokadowego  $S^*$ ,

$G_1 \dot{\cup} G_2$  – suma digrafów  $G_1 = (V_1, E_1)$  i  $G_2 = (V_2, E_2)$ :  $G_1 \dot{\cup} G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ ,

$$\bigcup_{G_i \in G^*} G_i = G_1 \dot{\cup} G_2 \dot{\cup} \dots \dot{\cup} G_a,$$

dla  $G^* = \{G_1, G_2, \dots, G_a\}$ .

Przykład drzewa – digrafu przedstawia rys. 2.23b.

**Drzewa** stanowiące reprezentacje przebiegów prowadzących do blokad oraz **wiry** reprezentujące przebiegi prowadzące do ustalonych przebiegów cyklicznych nazywane są **digrafami zachowań** SWMPC. Zbiory wierzchołków rozważanych klas digrafów zachowań są zbiorami rozłącznymi.

#### Własność 6

*Digrafy zachowań (wiry i drzewa) oraz początkowe stany blokadowe stanowią komponenty spójności digrafu przestrzeni  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$ .*

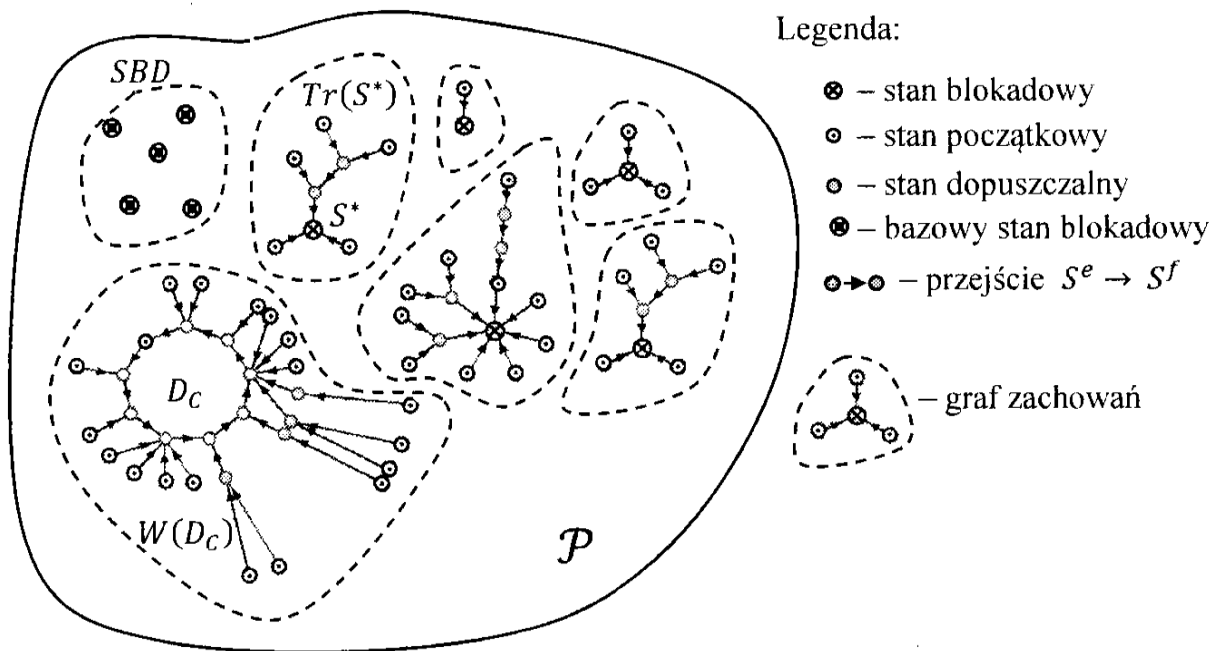
Dowód: Niech para wirów (59) posiada wspólny wierzchołek (stan). Jeżeli stan, reprezentowany przez ten wierzchołek, jest wspólny dla przebiegów przejściowych, to przebiegi te prowadzą do dwóch różnych ustalonych przebiegów cyklicznych – jest to sprzeczne z Własnością 5. Jeżeli natomiast istnieje stan wspólny dla przebiegów cyklicznych tworzących te wiry, to stoi to w sprzeczności z Własnością 4. Postępując analogicznie można wykazać tą własność również dla pary drzew i par mieszanych (drzewo – wir). □

Z Własności 6 wynika, że przestrzeń  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$  stanowi kompozycję różnego rodzaju komponentów spójności (rys. 2.24), digrafów zachowań (wirów i drzew oraz początkowych stanów blokadowych *SBD*) obejmujących wszystkie osiągalne w systemie przebiegi (instancje zachowań). Oznacza to również, że:

- jeżeli stan początkowy należy do cyklu to wszystkie stany z niego osiągalne również należą do tego cyklu, innymi słowy oznacza to, że ustalone przebiegi cykliczne są przebiegami rozłącznymi,
- inicjacja procesów z określonego stanu początkowego spoza cyklu (pomijając *SBD*) prowadzi do cyklu lub blokady, innymi słowy oznacza to, że przebiegi przejściowe prowadzą bądź to do cykli bądź też do blokad.

Rodzące się w tym kontekście pytania mogą dotyczyć zachowań osiągalnych w określonej strukturze systemu :

- Czy w systemie o zadanej strukturze  $SC^{lp}$  osiągalne są ustalone przebiegi cykliczne (wiry)?
- Jakiego rodzaju przebieg inicjowany w stanie  $S^r$  jest osiągalny w systemie o strukturze  $SC^{lp}$ ?
- Jaki stan  $S^r$  systemu o strukturze  $SC^{lp}$  gwarantuje osiągalność żadanego przebiegu cyklicznego ?
- ...



Rys. 2.24. Przestrzeń  $\mathcal{P}$  reprezentująca zachowanie SWMPC

Mogą to być również pytania związane ze strukturą determinującą zadane zachowanie:

- Czy przyjęte reguły priorytetowania gwarantują istnienie przestrzeni  $\mathcal{P}$  zawierającej przebiegi cykliczne (wiry)?
- Czy istnieje taka postać reguł priorytetowania, dla których systemem osiąga przebieg cykliczny bez względu na stan, od którego jest inicjowany?
- Jakie postacie reguł priorytetowania gwarantują istnienie przestrzeni  $\mathcal{P}$ , zawierającej przebiegi cykliczne (wiry)?
- Jakie wartości parametrów struktury  $SC^{lp}$  gwarantują istnienie przestrzeni  $\mathcal{P}$ , zawierającej przebiegi cykliczne (wiry)?
- ...

Jedną z najważniejszych cech zachowania SWMPC jest niezmienność raz osiągniętego przebiegu cyklicznego (własności 4 i 6). Oznacza to, że jeżeli system raz „wpadnie” w wir to nie będzie możliwa zamiana jego zachowania (w praktyce konieczne jest w takiej sytuacji wstrzymanie procesów i rozpoczęcie ich realizacji na nowo, od innego stanu początkowego). Tego typu izolacja zachowań niesie za sobą następujące pytania (wpisujące się w problematykę re-harmonogramowania zachowań cyklicznych):

- Jakie parametry struktury  $SC^{lp}$  należy zmienić by możliwa była zmiana aktualnego przebiegu cyklicznego na inny?
- W jakim stanie  $S^r$  osiągniętego przebiegu cyklicznego należy dokonać zmian parametrów struktury by możliwa była zmiana przebiegu cyklicznego na inny?
- Czy istnieje możliwość zmiany przebiegu cyklicznego bez konieczności osiągnięcia stanu przejściowego?
- Czy żądany przebieg jest osiągalny z zadanych przebiegów cyklicznych?
- ...



### 2.2.3. Wielopoziomowy model przestrzeni stanów

Przyjęty sposób reprezentacji struktury  $SC^{lp}$  (18) pozwala rozważać ją na różnych poziomach zachowania. Należy zauważyć, że zaproponowane definicje stanu  $S^r$  (19) i funkcji przejścia  $\delta^{lp}$  (determinowanej przez zbiór predykatów  $\Lambda^{lp}$  (51)) „przewidują” tego typu wielopoziomowość zachowań. Oznacza to, że przestrzeń stanów może być również formułowana z dokładnością do każdego poziomu z osobna:  $\mathcal{P}^0, \mathcal{P}^1, \dots, \mathcal{P}^l, \dots, \mathcal{P}^{lp}$ , gdzie  $\mathcal{P}^0$  oznacza przestrzeń stanów opisującą zachowanie procesów lokalnych (poziom zachowań  $SL$ ),  $\mathcal{P}^1, \dots, \mathcal{P}^l, \dots, \mathcal{P}^{lp}$  – kolejno przestrzenie stanów dla poziomów zachowań procesów multimodalnych (poziom  $SM^1, \dots, SM^l, \dots, SM^{lp}$ ). Przykłady przestrzeni  $\mathcal{P}^0, \mathcal{P}^1, \mathcal{P}^2$  opisujących zachowania procesów na poziomie  $SL, SM^1, SM^2$  systemu rys. 2.6 zostały przedstawione na rys. 2.25.

W ogólnym przypadku, przestrzeń stanów  $l$ -tego poziomu definiowana jest jako digraf:  $\mathcal{P}^l = (\mathbb{S}^l, \mathbb{E}^l)$  w którym zbiór stanów dopuszczalnych  $\mathbb{S}^l$  składa się ze stanów  $S^r = ((Sl^r, m^1 S^r), \dots, m^l S^r)$  (zgodnie z (19)), a zbiór łuków  $\mathbb{E}^l \subseteq \mathbb{S}^l \times \mathbb{S}^l$  jest determinowany przez funkcję przejścia  $\delta^l$  (określoną przez zbiór predykatów  $\Lambda^l$  (51)). Należy pamiętać, że stany zbioru  $\mathbb{S}^l$  zawierają informację o procesach poziomu  $l$  oraz wszystkich procesach z poziomów niższych ( $1 \dots l - 1$ ). Inaczej mówiąc, stany zbioru  $\mathbb{S}^l$  są determinowane przez stany poziomów niższych.

W prezentowanym przykładzie przestrzeń stanów  $\mathcal{P}^0$  opisująca zachowanie systemu na poziomie lokalnym jest tą samą przestrzenią, która została zilustrowana na rys. 2.14. Podobnie jak w przypadku struktury  $SC^{lp}$ , struktura wyższego poziomu zachowań jest determinowana przez strukturę poziomu niższego, w przypadku przestrzeni  $\mathcal{P}^1$  determinowana jest ona przez postać przestrzeni  $\mathcal{P}^0$ . Inaczej mówiąc przebieg będący przejawem zachowania procesów lokalnych (np. pociągów linii metra) kształtuje postać przebiegu opisującego zachowanie procesów multimodalnych (np. pasażerów). Odwołując się ponownie do przykładu linii metra, każdej podróży, w której bierze udział pasażer, odpowiada pewien plan poruszania się pociągów.

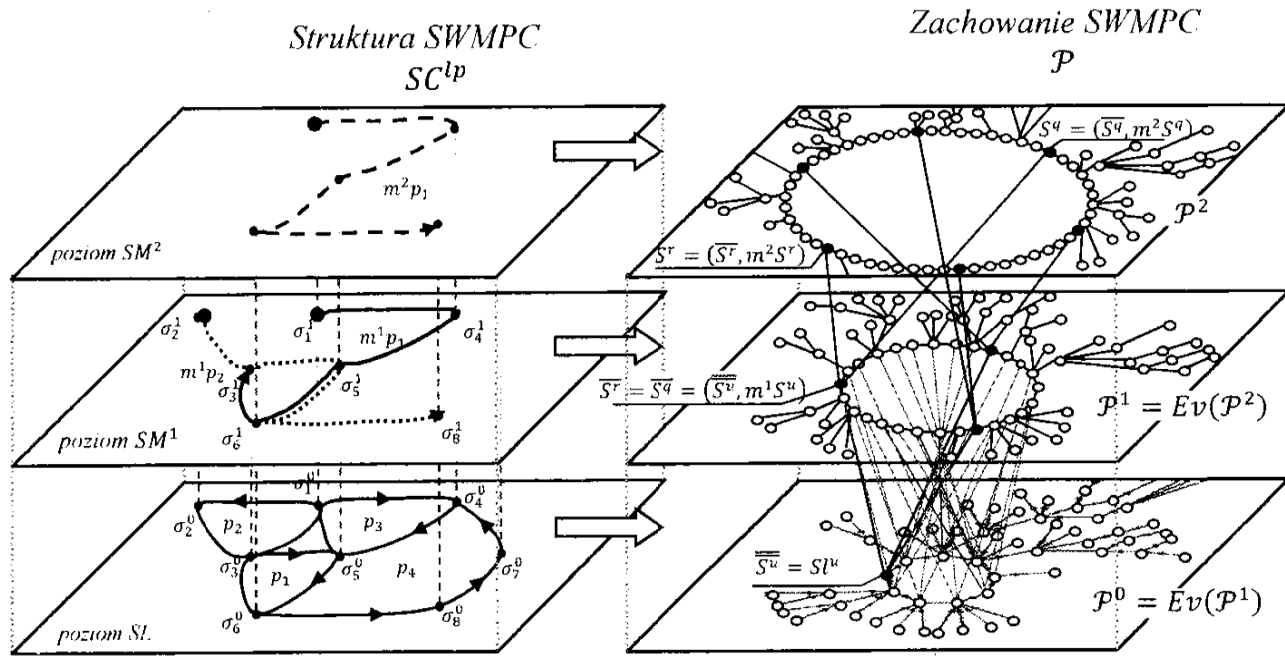
Wykorzystując podejście top-down do analizy zachowania SWMPC można powiedzieć, że przestrzeń  $\mathcal{P}^l$  reprezentująca zachowanie procesów z  $l$ -tego poziomu zachowań jest rzutem przestrzeni  $\mathcal{P}^{l+1}$  poziomu  $(l + 1)$ . Pojęcie **rzutu** przestrzeni  $\mathcal{P}^l$  jest definiowane następująco:

Dane jest zachowanie SWMPC scharakteryzowane przez przestrzeń stanów  $\mathcal{P}^{l+1} = (\mathbb{S}^{l+1}, \mathbb{E}^{l+1})$ , gdzie:

- $\mathbb{E}^{l+1} \subseteq \mathbb{S}^{l+1} \times \mathbb{S}^{l+1}$  – zbiór łuków digrafu  $\mathcal{P}^{l+1}$  determinowanych przez funkcję przejścia  $\delta^{l+1}$  (opisaną zbiorem predykatów  $\Lambda^{l+1}$  (51)),
- $\mathbb{S}^{l+1}$  – zbiór stanów dopuszczalnych na  $(l + 1)$ -wszym poziomie zachowań, gdzie stan dopuszczalny  $S^r \in \mathbb{S}^{l+1}$  określany jest następująco:

$$S^r = (((Sl^r, m^1 S^r), \dots, m^l S^r), m^{l+1} S^r) = (\overline{S^r}, m^{l+1} S^r). \quad (61)$$

Przyjmując, że zbiór stanów dopuszczalnych  $\mathbb{S}^l$  składa się ze stanów  $\overline{S^r}$ , a zbiór łuków  $\mathbb{E}^l \subseteq \mathbb{S}^l \times \mathbb{S}^l$  jest determinowany przez funkcję przejścia  $\delta^l$ , przestrzeń stanów  $\mathcal{P}^l = (\mathbb{S}^l, \mathbb{E}^l)$  nazywana jest rzutem  $\mathcal{P}^{l+1}$  i oznaczana przez  $\mathcal{P}^l = Ev(\mathcal{P}^{l+1})$ .



Rys. 2.25. Przestrzenie  $\mathcal{P}^0$ ,  $\mathcal{P}^1$ ,  $\mathcal{P}^2$  na przykładzie SWMPC z rys. 2.6

Powyższy opis określa w jaki sposób przestrzeń stanów  $\mathcal{P}$  rzutuje się na poszczególne poziomy zachowań ( $\mathcal{P}^0$ ,  $\mathcal{P}^1$ , ...,  $\mathcal{P}^l$ , ...,  $\mathcal{P}^{lp}$ ). W tym miejscu powstaje pytanie jak rzutują się jej poszczególne elementy takie jak wiry i drzewa? W przypadku wirów słuszne jest następująca własność:

#### Własność 7

Rzut wiru jest wirem.

Dowód: Ponieważ każdy wir utworzony jest z przebiegu cyklicznego oraz przebiegów przejściowych prowadzących do niego, dowód własności podzielony jest na dwie części. W pierwszej kolejności należy wykazać, że rzut ustalonego przebiegu cyklicznego jest również przebiegiem cyklicznym.

Przejścia między stanami dowolnego przebiegu cyklicznego  $D_C$  przestrzeni  $\mathcal{P}^{l+1}$  realizowane zgodnie z funkcją  $\delta^{l+1}$  (opisaną zbiorem predykatów  $\Lambda^{l+1}$ ) mogą być określone następująco:

$$\rightarrow S^{d_1} \rightarrow \dots \rightarrow S^{d_i} \rightarrow \dots \rightarrow S^{d_{td}} \rightarrow S^{d_1} \rightarrow \quad (62)$$

zgodnie z (61) wyrażenie to może być przedstawione jako:

$$(\overline{S^{d_1}}, m^{l+1} S^{d_1}) \rightarrow \dots (\overline{S^{d_i}}, m^{l+1} S^{d_i}) \rightarrow \dots (\overline{S^{d_{td}}}, m^{l+1} S^{d_{td}}) \rightarrow (\overline{S^{d_1}}, m^{l+1} S^{d_1}) \quad (63)$$

rzut takiego przebiegu (w przestrzeni  $\mathcal{P}^l$ ) jest utworzony ze stanów  $\overline{S^{d_i}} \in \mathbb{S}^l$ :

$$\rightarrow \overline{S^{d_1}} \rightarrow \dots \rightarrow \overline{S^{d_i}} \rightarrow \dots \rightarrow \overline{S^{d_{td}}} \rightarrow \overline{S^{d_1}} \rightarrow \quad (64)$$

Powstaje pytanie: Czy tak otrzymany przebieg jest dopuszczalny w przestrzeni  $\mathcal{P}^l$ ? Przejścia między stanami przebiegu (64):  $\overline{S^{d_i}} \rightarrow \overline{S^{d_{i+1}}}$  są realizowane zgodnie z funkcją  $\delta^l$  opisaną zbiorem predykatów  $\Lambda^l$  stanowiącym podzbiór  $\Lambda^{l+1}$  (51). Jeżeli dla stanów  $S^{d_i}$ ,  $S^{d_{i+1}}$  z przestrzeni  $\mathcal{P}^{l+1}$  zbiór predykatów  $\Lambda^{l+1}$  jest spełniony (przejście

$S^{d_i} \rightarrow S^{d_{i+1}}$  jest dopuszczalne (62)), to dla stanów:  $\overline{S^{d_i}}, \overline{S^{d_{i+1}}}$  jego podzbiór  $\Lambda^l$  będzie spełniony również, a zatem przejście typu  $\overline{S^{d_i}} \rightarrow \overline{S^{d_{i+1}}}$  jest również dopuszczalne. Oznacza to, że przebieg cykliczny (64) jest rzutem przebiegu cyklicznego (62). W analogiczny sposób można wykazać się, że każdy przebieg przejściowy prowadzący do przebiegu cyklicznego  $D_C$  ma rzut będący przebiegiem przejściowym prowadzącym do rzutu przebiegu  $D_C$ , co kończy dowód. □

W przypadku drzew tego typu własność nie zawsze jest spełniona. Ze względu na to, że blokada procesów (zgodnie z Twierdzeniem 2) może występować na dowolnym poziomie zachowań, procesy realizowane na poziomach niższych mogą być realizowane cyklicznie. Przykładowo blokada pasażerów linii metra oznacza blokadę wszystkich procesów warstw wyższych, ale pociągi (poruszające się na niższej warstwie) mogą nadal poruszać się cyklicznie. W modelu wielopoziomowym blokada procesów  $l$ -tego poziomu oznacza blokadę procesów poziomów wyższych ( $l + 1 \dots lp$ ), lecz niekoniecznie musi oznaczać blokadę procesów poziomów niższych ( $0 \dots l - 1$ ). Z tego powodu nie zawsze rzut drzewa (związanego ze stanem blokadowym) jest również drzewem.

Własność 7 może być wykorzystywana w praktyce przy wyznaczaniu cyklicznych przebiegów. Dzięki temu, że przebiegi cykliczne są odwzorowane na każdym poziomie, analizę zachowania całego systemu można przeprowadzać dla każdego poziomu oddzielnie. Przykładowo wyznaczenie w pierwszej kolejności przebiegu cyklicznego na poziomie  $SL$  (przebieg przestrzeni  $\mathcal{P}^0$ ) pozwala ograniczyć potencjalny zbiór stanów (rzutujących się na stany wyznaczonego przebiegu) poziomu  $SM^1$  (stanów przestrzeni  $\mathcal{P}^1$ ). Postępując analogicznie, wyznaczenie przebiegu cyklicznego dla poziomu  $SM^1$  pozwala ograniczyć poszukiwania na poziomie  $SM^2$ , itd. W ogólności wyznaczanie przebiegu cyklicznego SWMPC sprowadza się do wyznaczania kolejno jego rzutów z poziomów  $SL, SM^1, \dots, SM^{lp}$ . Własność ta będzie wykorzystywana w jednej z metod przedstawionych w rozdziale 3.

### 2.3. Modele deklaratywne problemów: analizy, syntezy i składania

W podrozdziale 1.1.2, przy opisie problemów harmonogramowania cyklicznego, zdefiniowane zostały problemy analizy i syntezy. Problemy te omawiane były również w podrozdziałach 1.2 i 1.3 gdzie rozszerzone zostały o trzeci typ problemu – problem składania. Bez względu na kontekst tych problemów związane z nimi pytania zawsze odnosiły się do cyklicznego harmonogramu bądź też do cech struktury systemu charakteryzowanych na przykład przez reguły priorytetowania.

Wprowadzone pojęcia struktury  $SC^{lp}$  i przestrzeni stanów  $\mathcal{P}$  (opisującej zachowanie SWMPC) pozwalają spojrzeć na te problemy szerzej. Analiza zachowania utożsamiana z problemami analizy nie sprowadza się jedynie do poszukiwania harmonogramów cyklicznych, ale również do poszukiwania przebiegów przejściowych, stanów blokadowych, itp. Z kolei wyznaczenie struktury  $SC^{lp}$  (problem syntezy) nie sprowadza się jedynie do określania reguł priorytetowania, ale również marszrut procesów, pojemności zasobów, itp. **Innymi słowy, problemy analizy można trakto-**

wać jako problemy osiągalności (w systemie o danej strukturze  $SC^{lp}$ ) określonego zachowania  $\mathcal{P}$ , natomiast problemy syntezy jako problemy istnienia struktury  $SC^{lp}$  gwarantującej zadane zachowanie  $\mathcal{P}$ .

Pytania związane z tymi problemami mogą odnosić się do: przebiegów cyklicznych zawierających stany startowe, przebiegów przejściowych żądanej długości, struktur charakteryzujących się narzuconą wzajemną liczbą wykonań procesów, itd.

Deklaratywny charakter stawianych przez użytkowników wymagań sprawia, że przedstawione typy problemów można rozważać w kategoriach problemów spełniania ograniczeń. Ogólnie **Problem Spełniania Ograniczeń PSO** związany z systemami współbieżnych procesów cyklicznych formułowany jest następująco:

$$PS = \left( (\{SC^{lp}, \mathcal{P}\}, \{D_{SC^{lp}}, D_{\mathcal{P}}\}), C_{PS} \right), \quad (65)$$

gdzie:

$\{SC^{lp}, \mathcal{P}\}$  – zbiór zmiennych decyzyjnych,  $SC^{lp}$  – struktura SWMPC (18),  $\mathcal{P}$  – przestrzeń stanów (zachowanie) SWMPC,

$\{D_{SC^{lp}}, D_{\mathcal{P}}\}$  – zbiór dziedzin zmiennych decyzyjnych,  $D_{SC^{lp}}$  – dziedzina określająca dopuszczalne struktury SWMPC,  $D_{\mathcal{P}}$  – dziedzina determinująca dopuszczalne przestrzenie stanów  $\mathcal{P}$ ,

$C_{PS} = C_{SC} \cup C_{\mathcal{P}} \cup C_D$  – zbiór ograniczeń określających związki między zmiennymi decyzyjnymi,  $C_{SC}$  – zbiór ograniczeń określających rodzaj struktury (założenia SWMPC patrz podrozdział 2.1.2 oraz tab. 2.1),  $C_{\mathcal{P}}$  – zbiór ograniczeń charakteryzujących zachowanie: założenia stanu SWMPC (19) (patrz podrozdział 2.2.1.1), warunki dopuszczalności stanów (22) – (31), warunki przejścia między stanami  $\Lambda^{lp}$  (51),  $C_D$  – zbiór ograniczeń dodatkowych, określających potrzeby użytkownika (np. istnienie cyklicznych przebiegów ustalonych).

W ogólnym przypadku przez rozwiązanie problemu spełniania ograniczeń rozumie się wyznaczenie takich wartości zmiennych decyzyjnych ze zbioru dziedzin, dla których spełnione będą wszystkie zadane ograniczenia (Russell i Norvig 2009). W przypadku problemu (65) rozwiązaniem jest taka postać struktury  $SC^{lp}$  i przestrzeni stanów  $\mathcal{P}$ , która spełnia założenia narzucane na strukturę, warunki dopuszczalności stanów, osiągalności przebiegów cyklicznych, itp.

W większości przypadków rozważanych problemów decyzyjnych przyjmuje się, że znana jest struktura  $SC^{lp}$  a poszukiwana jest przestrzeń stanów  $\mathcal{P}$ , lub na odwrót przyjmuje się, że znana jest przestrzeń stanów  $\mathcal{P}$  a poszukiwana jest struktura  $SC^{lp}$ . Konstatacja ta prowadzi do szczególnych postaci problemu (65) odpowiadających kolejno przykładowym uszczegółowieniom problemów analizy i syntezy:

- problem analizy:

$$PS_{ST} = \left( (\mathcal{P}, D_{\mathcal{P}}), C_{PS} \cup C_{ST} \right), \quad (66)$$

gdzie zmienną decyzyjną stanowi przestrzeń stanów  $\mathcal{P}$  a zbiór ograniczeń obejmuje m.in. ograniczenia  $C_{ST}$  charakteryzujące strukturę  $SC^{lp}$  (ograniczenia określające wartości parametrów (18) opisujących strukturę).

- problem syntezy:

$$PS_{RE} = \left( (SC^{lp}, D_{SC^{lp}}), C_{PS} \cup C_{RE} \right), \quad (67)$$

gdzie zmienną decyzyjną jest struktura  $SC^{lp}$  a zbiór ograniczeń obejmuje m.in. ograniczenia  $C_{RE}$  charakteryzujące zadaną przestrzeń stanów  $\mathcal{P}$  (ograniczenia określające postać przestrzeni stanów a w szczególnym przypadku określające założenia jakie powinna spełniać ta przestrzeń).

Rozwiązanie tak sformułowanych problemów pozwala na udzielenie odpowiedzi kolejno na następujące pytania:

- Czy dla zadanej struktury  $SC^{lp}$  systemu klasy SWMPC (którego działanie definiują  $C_{SC}, C_{\mathcal{P}}, C_{ST}$ ) istnieje przestrzeń stanów  $\mathcal{P}$  (zachowanie SWMPC) spełniająca zadane ograniczenia  $C_D$ ?
- Czy istnieje struktura  $SC^{lp}$  systemu klasy SWMPC spełniająca ograniczenia  $C_D$ , która gwarantuje zadaną przestrzeń stanów  $\mathcal{P}$ ?

Odrębny przypadek stanowi problem składania. Problem ten w ujęciu deklaratywnym może być zdefiniowany następująco:

Dany jest zbiór struktur systemów o znanych zachowaniach cyklicznych (przestrzeniach stanów):

$$SC^* = \{SC_1^{lp}, \dots, SC_i^{lp}, \dots, SC_{lc}^{lp}\}. \quad (68)$$

Poszukiwana jest odpowiedź na pytanie:

*Czy istnieje struktura  $\overline{SC}$  złożona ze struktur zbioru  $SC^*$ :*

$$\overline{SC} = SC_{j_1}^{lp} \oplus SC_{j_2}^{lp} \oplus \dots \oplus SC_{j_e}^{lp} \oplus \dots \oplus SC_{j_f}^{lp}, SC_{j_e}^{lp} \in SC^*, \quad (69)$$

gdzie:  $\oplus$  – operator składania struktur SWMPC,

*dla której istnieje rozwiązanie problemu (70)?*

$$PS^* = \left( (\mathcal{P}, D_{\mathcal{P}}), \{C_{SC} \cup C_{\mathcal{P}} \cup \mathcal{P} \text{ zawiera przebiegi cykliczne} \cup \overline{SC}\} \right). \quad (70)$$

Oznacza to, że w problemie składania poszukiwana jest taka kompozycja wybranych systemów, dla której istnieje zadany cykliczny przebieg ustalony.

Tego typu problem może być również formułowany w kontekście poszukiwania operatorów składania  $\oplus$ , operatorów gwarantujących zachowanie cykliczności systemów składanych z systemów o zachowaniach cyklicznych. Innymi słowy, składanie struktur SWMPC o znanych zachowaniach cyklicznych powinno gwarantować istnienie cyklicznych przebiegów SWMPC struktur wynikowych. Problemy składania są szerzej omawiane w rozdziale 4.

## 2.4. Rozstrzygalność problemów analizy, syntezy i składania

**Problemy analizy.** Problemy analizy są problemami rozstrzygalnymi tzn. problemami, dla których istnieje algorytm pozwalający w skończonym czasie uzyskać odpowiedź o braku bądź istnieniu jego rozwiązania (Papadimitrou 1994). W przypadku problemu (66) oznacza to, że ocena istnienia przestrzeni stanów  $\mathcal{P}$  spełniającej zadane ograniczenia, wymaga skończonej ilości czasu. W tym kontekście kluczowego znaczenia nabiera pytanie dotyczące złożoności obliczeniowej tego typu problemu?

W literaturze przedmiotu kwestia oceny złożoności obliczeniowej poruszana jest głównie w odniesieniu do systemów z procesami lokalnymi – SWPC. W pracach (Araki et al. 1977; Reveliotis et al. 1997) pokazano, że w systemach klasy  $WPP|o_l$ , dla których procesy składają się z ciągu uporządkowanych operacji zadanie unikania blokad (traktowane jako problem analizy – patrz podrozdział 1.2) jest problemem NP-zupełnym. Analogiczne wnioski uzyskano dla problemów rozważanych w dziedzinie sieci Petriego (Jones et al. 1977). Ponadto NP-zupełny charakter wykazano również dla problemów klasy  $WPP|o_c, k_s$ , dla których występuje tylko częściowy porządek operacji procesów (Roszkowska 1991, 2004; Roszkowska i Wójcik 1992).

Z przedstawionych rozważań wynika, że w ogólnym przypadku dla systemów z procesami lokalnymi, nie należy spodziewać się istnienia efektywnych metod rozwiązywania problemu analizy. Metody efektywne czasowo istnieją jedynie dla szczególnych przypadków, w których przyjmowane są założenia dodatkowe. Przykładowo w pracach (Banaszak 1991; Błażewicz et al. 1994; Fanti et al. 1996b; Reveliotis et al. 1997; Roszkowska i Jentink 1993; Roszkowska 2000) pokazano, że w sytuacjach unikania blokad związanych z systemami klasy  $WPP|o_l, k_s, j_u$ , w których każdy z dostępnych zasobów  $R_k \in R$  ma pojemność  $C(R_k) \geq 2$ , istnieje efektywny algorytm oparty o analizę digrafu reprezentującego aktualne żądania zasobowe procesów. Podobne wnioski otrzymano w pracy (Błażewicz 1988) przy założeniu, że pojemność zasobów  $C(R_k) \geq 1$ .

Zgodnie z przyjętymi założeniami rozważana klasa SWMPC stanowi rozszerzenie klasy  $WPP|o_l, k_s$  (rys. 2.5) o możliwość uwzględniania procesów multimodalnych. Rozszerzenie tego typu nie prowadzi do ograniczenia liczby dopuszczalnych stanów systemów a wręcz odwrotnie, z każdym kolejnym poziomem zachowań liczba stanów dopuszczalnych wzrasta. Innymi słowy, SWMPC charakteryzuje się większym niż SWPC obszarem poszukiwań (zwykle przez tego typu obszar rozumie się zbiór stanów dopuszczalnych). Można więc uznać, że skoro problemy analizy SWPC są NP-zupełne to ich rozszerzenie na obszar SWMPC również charakteryzuje ta sama złożoność. Za NP-zupełnością problemów analizy przemawia dodatkowo fakt, że klasa  $WPP|o_l, k_s$  jest traktowana jako szczególny przypadek, w którym nie występują procesy multimodalne.

**Problemy syntezy.** W przypadku problemów syntezy kwestia rozstrzygalności nie jest już tak oczywista. Podstawową cechą problemów nierozstrzygalnych jest nieograniczony zbiór potencjalnych rozwiązań. W przypadku problemu syntezy (67) zbiór tego typu obejmuje wszystkie potencjalne struktury rozważanego systemu. Zwykle w

tego typu systemach ustalonymi parametrami są zasoby  $R$  (o określonych pojemnościach  $C(R_k)$ ), z których budowana jest struktura.

Liczba struktur jakie można zbudować z zasobów zbioru  $R$  (zgodnie z (18)) jest determinowana przez pozostałe parametry: liczbę potoków, zbiory marszrut ( $U$  – dla procesów lokalnych i  $m^l U$  – dla procesów multimodalnych) oraz zbiory reguł priorytetowania  $\theta^l$ .

W celu wykazania rozstrzygalności problemu syntezy (67) wystarczy pokazać, że zbiór potencjalnych struktur jakie należy rozważyć w celu rozwiązania problemu jest skończony. Zbiór ten jest skończony jeżeli wszystkie parametry determinujące strukturę przyjmują wartości ze skończonych dziedzin.

Analizując kolejne parametry łatwo widać, że w przypadku implementacji protokołu wzajemnego wykluczania liczba potoków jest ograniczona przez liczbę zasobów oraz ich pojemności. Dodatkowo, przy definiowaniu przestrzeni stanów  $\mathcal{P}$  liczba ta wraz z liczbą poziomów zachowań jest zwykle z góry znana. Liczba potencjalnych marszrut jakie mogą zostać utworzone z zasobów zbioru  $R$  również jest liczbą skończoną. Wynika to z faktu, że każda marszruta może być reprezentowana przez ścieżkę w digrafie o wierzchołkach  $R$ . Jak wiadomo liczba ścieżek w digrafie ze skończoną liczbą wierzchołków jest też skończona (Ross i Wright 1999).

Oznacza to, że problemy syntezy zorientowane na poszukiwanie liczby potoków i/lub marszrut struktury SWMPC są problemami rozstrzygalnymi. Przykładowo, dla systemów z zasobami jednopojemnościowymi ( $C(R_k) = 1$ ) ograniczając się jedynie do procesów lokalnych (procesy multimodalne nie występują) można oszacować maksymalną liczbę potencjalnych struktur – liczba ta wynosi:  $\frac{lk(lk+1)}{2} \sum_{i=2}^{lk} \frac{lk!}{(lk-i)!}$  ( $lk$  – liczba zasobów). W dużym uproszczeniu można powiedzieć, że algorytm przeglądu przestrzeni potencjalnych struktur charakteryzuje się złożonością rzędu silnia  $f(n) = \mathcal{O}(n!)$ . Oznacza to, że podobnie jak w przypadku problemów analizy nie należy się spodziewać efektywnych (wielomianowych) algorytmów wyznaczania struktur  $SC^{lp}$ .

Kwestia rozstrzygalności problemu syntezy komplikuje się jednak w przypadku poszukiwania reguł priorytetowania  $\theta^l$ . Problem tego typu można sprowadzić do szczególnej postaci uwzględniającej tylko reguły procesów lokalnych:

$$PS_{\theta} = ((\theta^0, D_{\theta}), C_{SC} \cup C_{\mathcal{P}} \cup \mathcal{P} \text{ zawiera przebiegi cykliczne} \cup C_R \cup C_P \cup C_U), \quad (71)$$

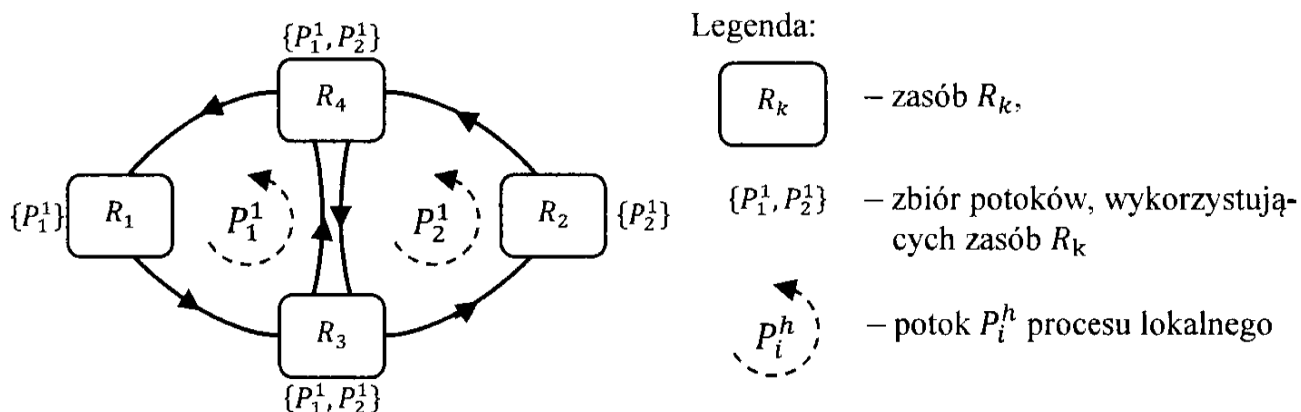
gdzie poszukiwanymi zmiennymi decyzyjnymi są reguły  $\theta^0$  synchronizujące procesy lokalne, gwarantujące istnienie cyklicznych przebiegów przestrzeni  $\mathcal{P}$  przy ustalonych parametrach struktury (18): zasobach  $R$  (ograniczenia  $C_R$ ), procesach  $P$  (ograniczenia  $C_P$ ), marszrutach  $U$  (ograniczenia  $C_U$ ).

Kłopot polega na tym, że w przyjętych założeniach struktury SWMPC nie ma żadnych ograniczeń limitujących postaci reguł priorytetowania. Przykładowa reguła  $\sigma_i$  może być sekwencją o dowolnej długości np.  $\sigma_5^0 = (P_4^1, P_4^2, P_1^1, P_3^1, P_1^1, P_1^1, P_1^1, P_1^1, P_4^1, P_4^2)$ . Brak ograniczeń na długość reguł priorytetowania oznacza, że proces weryfikacji kolejnych struktur określanych przez coraz to dłuższe reguły priorytetowania może trwać bez końca. Sytuacja tego typu ma miejsce na przykład w przypadku prezentowanych, w poprzednim rozdziale (podrozdział 1.2)

systemów prowadzących zawsze do blokad (rys. 1.21). Dodatkowe wątpliwości co do rozstrzygalności problemu (71) budzi skojarzenie z tak zwanym problemem Odpowiedniości Posta (Post 1946; Hopcroft et al. 2006). W problemie tym, który jest problemem nierozstrzygalnym, poszukuje się równoważnych ciągów słów, które podobnie jak w rozważanym przypadku mogą być ciągami dowolnej długości.

Pomimo podobieństw z problemem odpowiedniości Posta można wykazać, że problem syntezy (71) jest problemem rozstrzygalnym. W tym celu wykorzystany zostanie system z rys. 2.26 składający się z czterech zasobów:  $R_1$ – $R_4$  oraz dwóch jednopotokowych procesów lokalnych (potoki:  $P_1^1, P_2^1$ ). W systemie nie są znane reguły priorytetowania:  $\theta^0 = (\sigma_1^0, \sigma_2^0, \sigma_3^0, \sigma_4^0)$ . W związku z tym problem (71) sprowadza się do odpowiedzi na pytanie: Czy istnieją takie reguły priorytetowania  $\theta^0$ , dla których w systemie osiągalne będzie zachowanie cykliczne?

Poszukiwane są w zasadzie reguły  $\sigma_3^0, \sigma_4^0$  (zasoby  $R_3$  i  $R_4$  są zasobami współdzielonymi). Potencjalne postacie tych reguł mogą być dowolne np.:  $\sigma_3^0 = (P_1^1, P_2^1, P_2^1, P_2^1, P_2^1)$ ,  $\sigma_4^0 = (P_2^1, P_2^1, P_1^1, P_1^1, P_1^1)$ . Brak jest też ograniczeń na długość sekwencji oraz liczbę wzajemnych wykonań potoków. Brak ograniczeń na długość reguł priorytetowania prowadzi do pytania: czy przy poszukiwaniu reguł  $\theta^0$  można ograniczyć się zawsze do skończonej liczby kroków (skończonego czasu)?

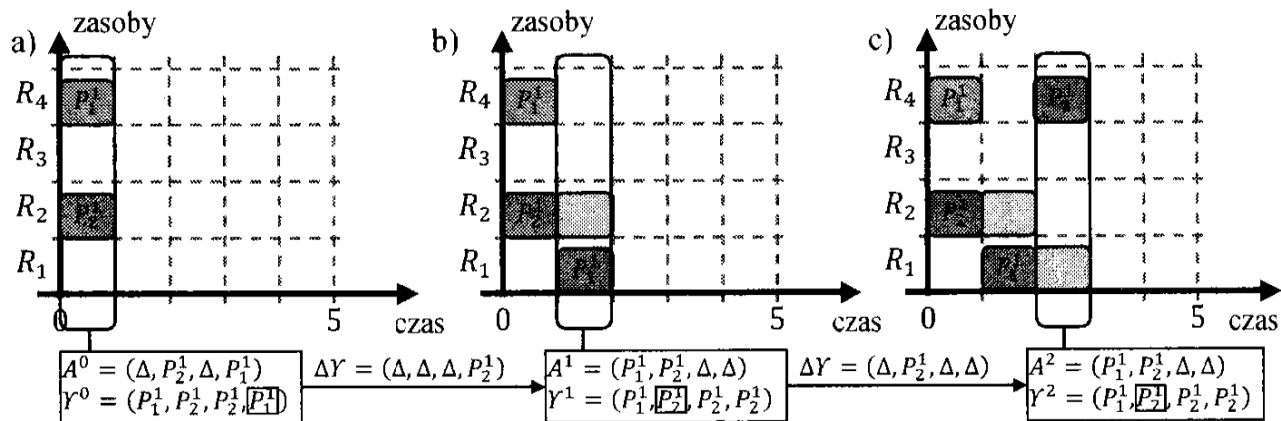


Rys. 2.26. SWMPC bez określonych reguł priorytetowania

Oczywistym jest, że przy pozytywnej odpowiedzi przegląd kolejnych (co raz dłuższych) wariantów sekwencji reguł priorytetowania prędzej czy później doprowadzi do rozwiązania. Pozostaje jednak przypadek gdy system jest blokadowy (jak jeden z systemów z rys. 1.21). W celu oceny rozstrzygalności tego typu przypadków, proponowane jest podejście polegające na konstrukcji harmonogramu realizacji procesów wraz z jednoczesnym konstruowaniem reguł priorytetowania. Podejście to wyjaśnia poniższy przykład.

Niech alokacja przedstawiona na rys. 2.27a reprezentuje stan początkowy systemu z rys. 2.26 (w ogólności może to być dowolny stan dopuszczalny). W stanie tym, potok  $P_1^1$  zaalokowany jest (i wykonuje operację) na zasobie  $R_4$  natomiast potok  $P_2^1$  na zasobie  $R_2$ . Zgodnie z (19) alokacja potoków na zasobach jest reprezentowana przez sekwencję:  $A^0 = (\Delta, P_2^1, \Delta, P_1^1)$ .





Legenda

 – realizacja operacji potoku  $P_i^j$      
  – oczekiwane potoku  $P_i^j$  na zasobie

Rys. 2.27. Harmonogram systemu z rys. 2.26

Umieszczenie potoków na zasobach  $R_2$  i  $R_4$  oznacza, że posiadają one prawo do przebywania na tych zasobach. **Prawo określające jakie potoki mogą przebywać na zasobach** zadane jest sekwencją:  $Y^0 = (P_1^1, P_2^1, P_2^1, P_1^1)$  definiowaną podobnie do semafora  $Z^0$  (19). Elementy sekwencji  $Y^0$  określają kolejno potoki mające prawo do alokacji (przebywania) na zasobach  $R_1 - R_4$ . Określają one zarówno potoki już zaalokowane (w przypadku zasobów zajętych) jak i te oczekujące na alokację (w przypadku gdy żądane zasoby są wolne). W odróżnieniu od semafora, wartości elementów sekwencji  $Y^0$  zmieniają się w momencie opuszczania zasobu przez potok. Należy przypomnieć, że semafor  $Z^0$ , w odróżnieniu od  $Y^0$  odnosi się do potoków, które jako następne otrzymają dostęp do zasobu – warunek 4 dopuszczalności stanów  $S^r$  (podrozdział 2.2.1).

Mechanizmy semaforów  $Z^0$  oraz praw alokacji  $Y^0$  mogą być stosowane zamienne – w zależności od potrzeb i/lub sytuacji. W rozważanym przypadku do reprezentacji praw dostępu użyta jest sekwencja  $Y^0 = (P_1^1, P_2^1, P_2^1, P_1^1)$  według której do zasobów  $R_1, R_4$  prawo alokacji (przebywania) ma prawo potok  $P_1^1$ , do zasobów  $R_2, R_3$  potok  $P_2^1$ . O ile prawo alokacji dla zasobów  $R_2$  i  $R_4$  wynika z zaalokowanych na nich potoków o tyle prawo do alokacji do zasobów  $R_1$  i  $R_3$  zostało wybrane arbitralnie (w ogólności mogą to być dowolne dopuszczalne wartości).

Stan początkowy z rys. 2.27a może więc być reprezentowany przez parę  $(A^0, Y^0)$ . W tak określonym stanie procesy po wykonaniu swoich operacji (przyjęte zostały czasy jednostkowe) żądają, zgodnie z procedurą przedstawioną na rys. 2.3, dostępu do kolejnych zasobów (określonych w ich marszrutach). Potok  $P_2^1$  żąda dostępu do zajętego zasobu  $R_4$ . Z kolei potok  $P_1^1$ , zgodnie z  $Y^0$ , żąda dostępu do zwolnionego zasobu  $R_1$ .

Zajęcie przez potok  $P_1^1$  zasobu  $R_1$  i rozpoczęcie wykonywania operacji na tym zasobie oznacza przejście systemu do kolejnego stanu  $(A^1, Y^1)$  (rys. 2.27b). Przejście między stanami, poza zmianą alokacji, wymaga również określenia nowego prawa alokacji na zasobie  $R_4$  (zwolnionym przez  $P_1^1$ ). Element sekwencji, dla którego należy określić nowe prawo alokacji jest oznaczany na rys. 2.26 ramką:  $\square$ . Ze względu na to, że na  $R_4$  wykonywane są operacje potoków  $P_1^1, P_2^1$  prawo alokacji może być określo-

ne tylko dla jednego z nich. Tak więc nic nie stoi na przeszkodzie by wybrać po raz kolejny potok  $P_1^1$ . W przypadku przejścia do stanu  $(A^1, Y^1)$  prawo na zasobie  $R_4$  zostało przyznane  $P_2^1$ . Zmiana praw alokacji na zasobach  $R_1 - R_4$  przy przejściu między stanami określana jest sekwencją  $\Delta Y = (\Delta, \Delta, \Delta, P_2^1)$ , gdzie:  $\Delta$  – oznacza brak zmiany wartości praw alokacji dla zasobów  $R_1 - R_3$ ,  $P_2^1$  – oznacza ustalenie dla zasobu  $R_4$  prawa alokacji dla potoku  $P_2^1$ .

W stanie  $(A^1, Y^1)$  (rys. 2.27b) potoki  $P_1^1, P_2^1$  pozostają zaalokowane na zasobach  $R_1, R_2$ :  $A^1 = (P_1^1, P_2^1, \Delta, \Delta)$  a prawo alokacji jest określone jako  $Y^1 = (P_1^1, P_2^1, P_2^1, P_2^1)$ .

Osiągnięcie przez system kolejnego stanu  $(A^2, Y^2)$  (rys. 2.27c) jest związane z przejściem potoku  $P_2^1$  na zasób  $R_4$ , z kolei  $P_1^1$  pozostaje na zasobie  $R_1$  (potok nie ma prawa alokacji do żądanego zasobu  $R_3$ ). Przejście do stanu  $(A^2, Y^2)$  związane jest z kolejnym wyborem prawa alokacji, tym razem na zasobie  $R_2$  (zasób ten został zwolniony przez potok  $P_2^1$ ). Ponieważ zasób ten jest wykorzystywany wyłącznie przez potok  $P_2^1$  tak więc prawo alokacji ponownie ustalane jest dla tego potoku:  $\Delta Y = (\Delta, P_2^1, \Delta, \Delta)$ .

Postępując analogicznie można wyznaczać następne alokacje procesów  $A^i$  tworząc w ten sposób harmonogram ich realizacji. Z każdą kolejną alokacją  $A^i$  należy dokonać wyboru nowych praw alokacji  $Y^i$ . Wybór ten dokonywany jest spośród skończonej liczby wariantów (liczba potoków, z których dokonywany jest wybór przy określaniu prawa alokacji jest skończony). Przejścia między kolejnymi alokacjami  $A^i$ , determinowane przyjętymi prawami alokacji  $Y^i$ , mogą być zilustrowane w postaci digrafu z rys. 2.28. Wierzchołki digrafu reprezentują pary  $(A^i, Y^i)$  natomiast łuki (wraz z przypisanymi do nich sekwencjami  $\Delta Y$ ) określają możliwe przejścia między stanami.

Pomimo tego, że digraf z rys. 2.28 jest digrafem zawierającym skończoną liczbę wierzchołków i łuków, pozwala odtworzyć dowolne zachowanie (przebieg) osiągalne w systemie z rys. 2.26.

Przykładowo kontynuacją przebiegu reprezentowanego przez harmonogram z rys. 2.27 jest przebieg cykliczny oznaczony na rys. 2.28 kolorem zielonym. Przebieg ten obejmuje kolejno stany reprezentowane przez pary:

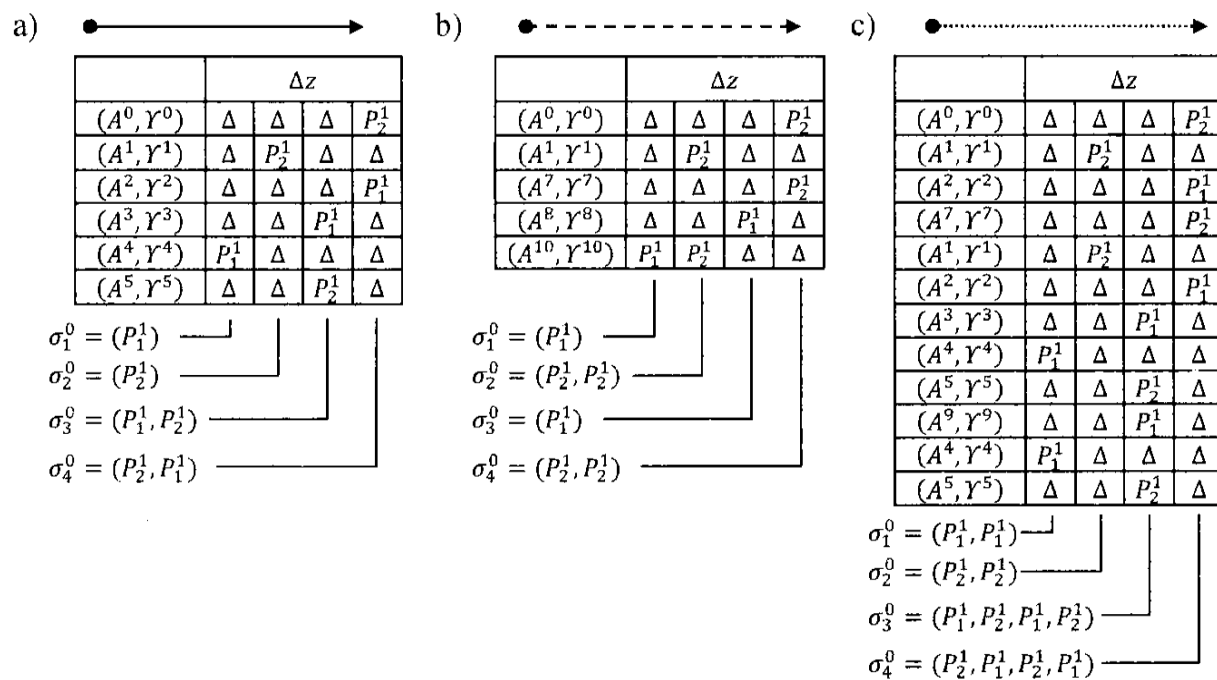
$$\dots \rightarrow (A^0, Y^0) \rightarrow (A^1, Y^1) \rightarrow (A^2, Y^2) \rightarrow (A^3, Y^3) \rightarrow (A^4, Y^4) \rightarrow (A^5, Y^5) \rightarrow \dots$$

Harmonogram realizacji operacji odpowiadający temu przebiegowi został przedstawiony na rys. 2.29. Prezentowany przykład pokazuje, że istnienie cyklicznego przebiegu ustalonego w danym systemie jest równoznaczne z istnieniem cyklu w digrafie przejść z rys. 2.28 (np. przebieg oznaczony linią ciągłą jest cyklem).

W digrafie z rys. 2.28 możliwe są też inne przebiegi. Przykładowo inny wybór prawa alokacji w stanie  $(A^2, Y^2)$  (tzn. ustalenie prawa alokacji na zasobie  $R_4$  dla  $P_2^1$ ) może prowadzić do przebiegu przejściowego (linia przerywana) skutkującego blokadą  $(A^{10}, Y^{10})$ . Jak widać, to jaki przebieg zostanie uzyskany jest zależne od początkowej alokacji potoków oraz praw alokacji określanych w trakcie osiągnięcia kolejnych stanów.



blokady rys. 2.30b nie może być osiągnięty gdyż potok  $P_2^1$  zgodnie z regułą  $\sigma_3^0$  ani razu nie otrzymuje dostępu do zasobu  $R_3$ . Zgodnie z przyjętymi warunkami reguły priorytetowania powinny być tak skonstruowane by każdy potok, wykorzystujący zasób  $R_k$  otrzymał do niego dostęp przynajmniej raz na jeden cykl systemu.



Rys. 2.30. Przykładowe reguły priorytetowania uzyskane z digrafu z rys. 2.28: dla przebiegu oznaczonego linią ciągłą a), dla przebiegu oznaczonego linią przerywaną b), dla przebiegu oznaczonego linią kropkowaną c)

W tym ujęciu kwestia istnienia ustalonego przebiegu cyklicznego sprowadza się do istnienia cyklu digrafu przejść (rys. 2.28), któremu odpowiadają dopuszczalne reguły priorytetowania (spełniające warunek przynajmniej jednokrotnego występowania potoku).

Pomijając przypadki przebiegów nierealizowalnych, wykorzystując digraf z rys. 2.28, można analizować przebiegi charakteryzujące się dowolnie długimi sekwencjami reguł priorytetowania. Przykładem może być przebieg oznaczony na rys. 2.28 linią kropkowaną. Przebiegowi temu odpowiadają reguły z dwukrotnym wystąpieniem każdego z potoków (rys. 2.30c). Wielokrotne występowanie potoków w regule wynika z możliwości tworzenia pętli np. poprzez powracanie do tych samych alokacji, pozwalających na wielokrotne powarzenie tych samych sekwencji alokacji, na przykład:  $\rightarrow (A^2, Y^2) \rightarrow (A^7, Y^7) \rightarrow (A^1, Y^1) \rightarrow$  (dla przebiegu oznaczonego linią kropkowaną).

Reasumując, digraf przejść z rys. 2.28 odwzorowuje wszystkie możliwe przejścia między alokacjami systemu bez znajomości reguł priorytetowania. Każdy przebieg utworzony z tego typu alokacji determinuje zbiór reguł priorytetowania (jak na rys. 2.30). Przebieg cykliczny jest osiągalny w systemie jeżeli w digrafie istnieje cykl, któremu odpowiadają reguły spełniające warunek występowania (przynajmniej jednokrotnie) każdego z potoków. Ponieważ w ramach jednego przebiegu wielokrotnie może być osiągnięta ta sama alokacja (patrz przebieg oznaczony na rys. 2.30 linią

kropkowaną) możliwe jest zatem rozważanie przebiegów zdeterminowanych regułami priorytetowania dowolnej długości.

Oznacza to, że digraf z rys. 2.28 odwzorowuje całą potencję zachowań osiągalnych w systemie z rys. 2.26.

**Poszukiwanie reguł priorytetowania, dla których system osiąga cykliczny przebieg ustalony sprowadza się więc do poszukiwania cyklu w odpowiadającym mu digrafie osiągalności.**

Jak wiadomo (Ross i Wright 1999), problem wyznaczania cyklu w digrafie jest problemem rozstrzygalnym. Oznacza to, że problem (71) poszukiwania reguł priorytetowania, dla których w systemie osiągalne jest zachowanie cykliczne jest również problemem rozstrzygalnym.

Należy zwrócić uwagę, że analizowany przykład odnosi się tylko do zachowania procesów lokalnych. Oczywiście nic nie stoi na przeszkodzie by podobne rozumowanie przeprowadzić w kontekście systemu uwzględniającego procesy multimodalne.

W ogólnym przypadku każdy SWMPC charakteryzowany strukturą  $SC^{lp}$  może być opisany przez digraf osiągalności jak z rys. 2.28, którego cykliczność oznacza istnienie reguł priorytetowania gwarantujących osiągalność cyklicznego przebiegu ustalonego.

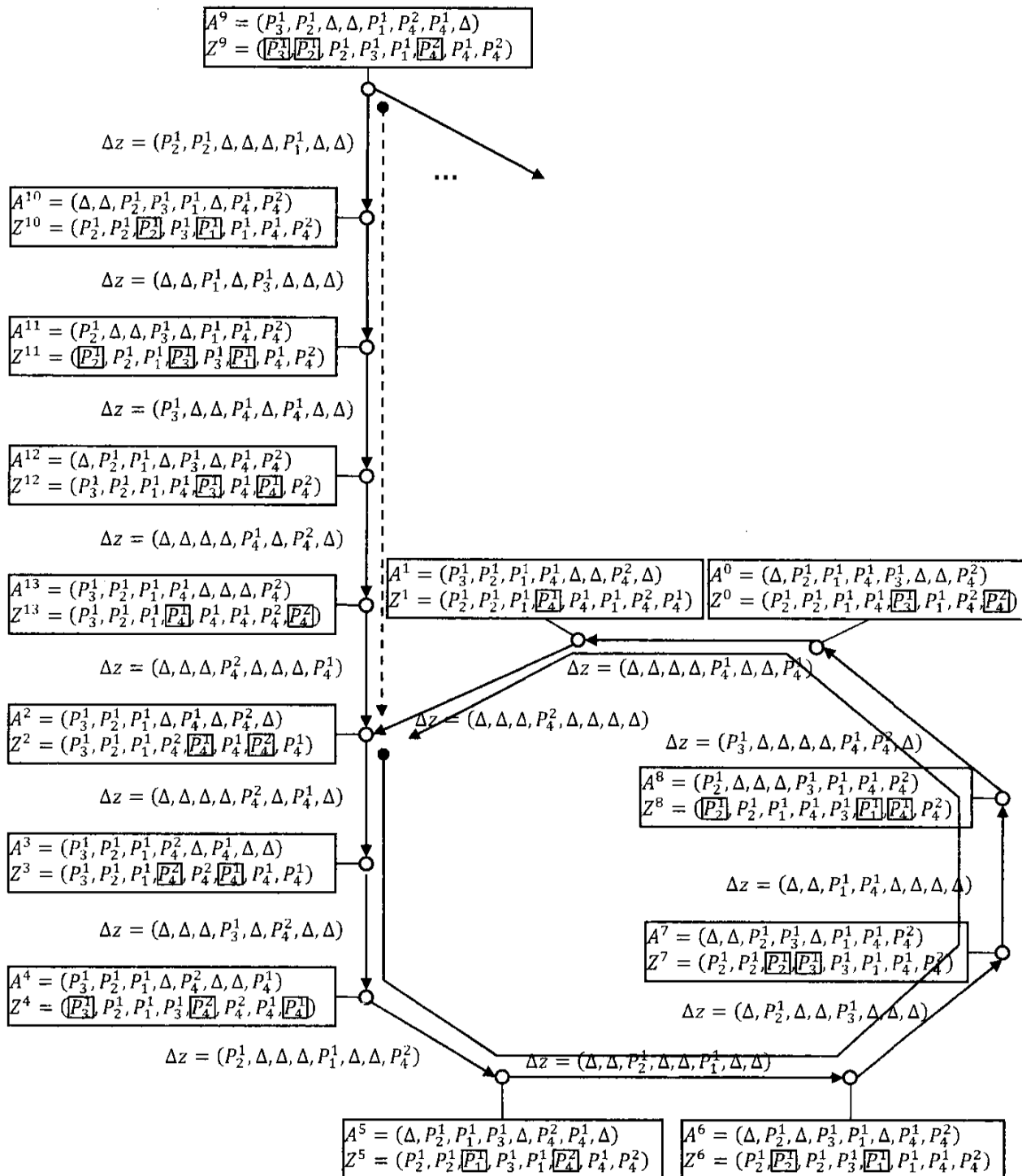
Oprócz poszukiwania cyklicznych przebiegów ustalonych, digrafy osiągalności (rys. 2.28) mogą być również użyteczne przy analizie własności przebiegów przejściowych. W podrozdziale 2.2.1 na rys. 2.11 przedstawiony został przebieg cykliczny wraz z prowadzącym do niego przebiegiem przejściowym. Przebiegi te reprezentowane w postaci digrafu przejść przedstawia rys. 2.31. Digraf ten jest podgrafem digrafu przejść charakteryzującego zachowanie procesów lokalnych systemu z rys. 2.6.

Warto zauważyć, że dla przebiegu przejściowego (obejmującego stany występujące w sekwencji  $(A^9, Y^9) \rightarrow (A^{10}, Y^{10}) \rightarrow \dots \rightarrow (A^{13}, Y^{13})$ ) oraz przebiegu cyklicznym (obejmującym stany występujące w sekwencji  $(A^0, Y^0) \rightarrow (A^1, Y^1) \rightarrow \dots \rightarrow (A^8, Y^8)$ ) obowiązuje ta sama kolejność alokacji potoków na zasobach współdzielonych. Wynika to z faktu, że raz ustalone reguły priorytetowania SWMPC nie ulegają zmianie (założenia SWMPC podrozdziału 2.1.1).

Własność ta potwierdza się w przypadku reguł priorytetowania uzyskanych z digrafu osiągalności z rys. 2.31. W rozważanym przypadku reguły uzyskane dla przebiegu przejściowego (oznaczonego na rys. 2.31 linią przerywaną) przedstawione na rys. 2.32a są fragmentami reguł (rys. 2.32b) odpowiadających przebiegowi cyklicznemu (oznaczonemu na rys. 2.31 linią ciągłą).

Przykładowo reguła dla zasobu  $R_5$  wynikająca z przebiegu przejściowego  $\sigma_5^{0'} = (P_3^1, P_4^1)$  jest fragmentem sekwencji reguły  $\sigma_5^0 = (P_4^2, P_1^1, P_3^1, P_4^1)$  przebiegu cyklicznego.

W ogólnym przypadku każdy przebieg przejściowy prowadzący do przebiegu cyklicznego charakteryzuje się regułami priorytetowania będącymi fragmentami reguł przebiegu cyklicznego, co znajduje swoje odbicie w Twierdzeniu 4.



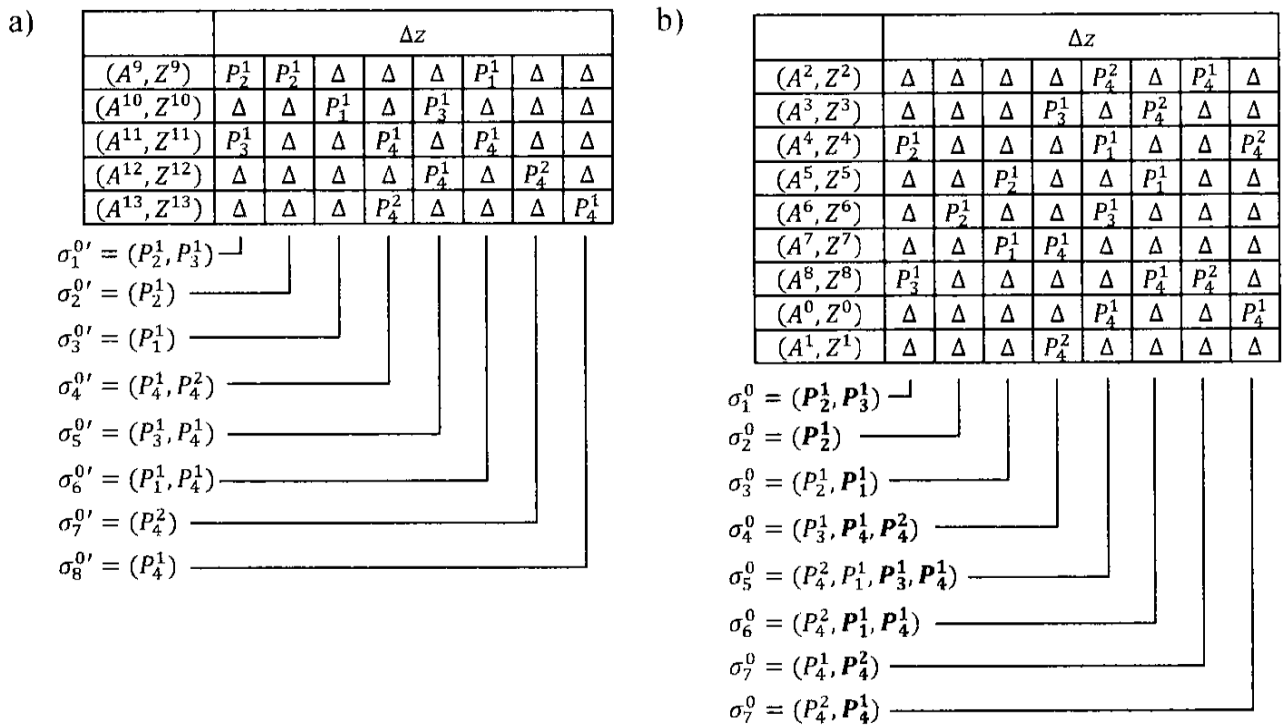
Rys. 2.31. Fragment digrafu osiągalności odpowiadający przebiegowi związanemu z harmonogramem przedstawionym na rys. 2.11

#### Twierdzenie 4

Jeżeli w przestrzeni stanów dopuszczalnych  $S^{lp}$  istnieje ustalony przebieg cykliczny  $D_C$ , do którego prowadzi przebieg przejściowy  $D_T$ , to liczba stanów przebiegu przejściowego nie jest większa od liczby stanów przebiegu cyklicznego:  $|V(D_T)| \leq |V(D_C)|$ .

Dowód oparty jest o przeprowadzoną wyżej dyskusję. Niech istnieje przebieg przejściowy  $D_T$ , prowadzący do przebiegu cyklicznego  $D_C$  zawierającego mniejszą liczbę stanów niż przebieg  $D_T$ :  $|V(D_T)| > |V(D_C)|$ . Przebiegi te mogą być przedstawione (podobnie jak na rys. 2.31) jako digraf osiągalności, w którym każdy stan jest repre-

zentowany jako para  $(A^i, Y^i)$  (w przypadku systemu z procesami multimodalnymi wielkości  $A^i, Y^i$  odnoszą się do alokacji i praw dostępu zarówno procesów lokalnych jak i wszystkich poziomów procesów multimodalnych).



Rys. 2.32. Reguły priorytetowania dla przebiegów z rys. 2.31: przebiegu przejściowego a), cyklicznego przebiegu ustalonego b)

W przyjętej reprezentacji każdemu z przebiegów odpowiada zbiór reguł priorytetowania wyznaczany z sekwencji  $\Delta Y$  (jak na rys. 2.32). Należy zwrócić uwagę że, zmiana praw alokacji  $\Delta Y$  zachodzi przy każdym ze stanów, a zatem liczba zmian jest równa liczbie stanów danego przebiegu.

Przebieg przejściowy  $D_T$  zawiera więcej stanów więc dochodzi do większej liczby zmian alokacji niż w przypadku przebiegu cyklicznego  $D_C$ . Większa liczba zmian praw alokacji powoduje, że co najmniej jedna reguła priorytetowania (uzyskiwana z sekwencji  $\Delta Y$ ) przebiegu przejściowego jest dłuższa od odpowiadającej jej reguły przebiegu cyklicznego. Jest to sprzeczne z własnością stwierdzającą, że reguły przebiegu przejściowego są fragmentami reguł przebiegów cyklicznych. Sprzeczność ta oznacza, że przebieg przejściowy nie może zwierać większej liczby stanów niż przebieg cykliczny do którego on prowadzi:  $|V(D_T)| \not\geq |V(D_C)|$ . □

Przedstawione twierdzenie jest wykorzystywane jako jedno z podstawowych elementów metody odtwarzania w przestrzeni stanów cyklicznych przebiegów ustalonych.

**Problem składania.** Oceniając rozstrzygalność problemu składania łatwo zauważyć, że zbiór potencjalnych rozwiązań obejmuje wszystkie możliwe kompozycje struktur otrzymanych ze zbioru  $SC^*$ . Liczbę tego typu struktur określa moc zbioru potęgowe-  
go  $\wp(SC^*)$ . Przyjmując, że  $lc = |SC^*|$  to  $|\wp(SC^*)| = 2^{lc}$ . Skończona liczba kompo-

zycji oznacza, że podobnie jak pozostałe problemy również rozstrzygalnym jest problem składania.

## 2.5. Podsumowanie

Deklaratywny model Systemu Współbieżnych Multimodalnych Procesów Cyklicznych wyróżnia strukturę  $SC^{lp}$  i przestrzeń stanów  $\mathcal{P}$  (zachowanie). Wielopoziomowy, hierarchiczny charakter struktury  $SC^{lp}$  umożliwia warstwową zhierarchizowaną analizę zachowania, analizę odnoszącą się do procesów multimodalnych wykorzystujących do swojej realizacji procesy poziomu niższego. Zachowanie procesów determinują zarówno parametry charakteryzujące marszruty, mechanizmy synchronizacji procesów multimodalnych itp., odpowiadających im poziomów struktury  $SC^{lp}$ , jak i zachowania poziomów niższych. Możliwość niezależnego wariantowania poszczególnych poziomów struktury pozwala modelować cały szereg wariantów SWMPC (tab. 2.1).

Przyjęta postać struktury systemu  $SC^{lp}$  determinuje potencjalne przejawy zachowania rozumianego jako przestrzeń  $\mathcal{P}$  dopuszczalnych stanów systemu. Modelem zachowania systemów klasy SWMPC jest niespójny digraf, komponentami spójności którego są digrafy zachowań (wir, drzewo) reprezentujące przebiegi prowadzące do blokady lub przebiegu cyklicznego. Wielopoziomowy charakter struktury  $SC^{lp}$  znajduje swoje odbicie również w definicjach stanu systemu, funkcji przejścia oraz przestrzeni stanów  $\mathcal{P}$ .

W rozważanym modelu formułowane i rozwiązywane są trzy podstawowe problemy (analizy, syntezy i składania) odpowiadające poniższym trzem pytaniom:

- Czy w SWMPC o danej strukturze istnieje zachowanie spełniające zadane żądania użytkownika?
- Czy istnieje struktura gwarantująca zachowanie SWMPC spełniające zadane żądania użytkownika?
- Czy istnieje struktura złożona z zadanych struktur SWMPC gwarantująca istnienie zachowania spełniającego zadane żądania użytkownika?

Wymienione problemy w sposób naturalny formułowane są w kategoriach problemów spełniania ograniczeń (66), (67), (70) oraz (71) stanowią uszczegółwienie rozważanego deklaratywnego modelu SWMPC tzn. dwójki (65), obejmującej strukturę i zachowanie  $\{\mathcal{P}, SC^{lp}\}$  oraz zbiór ograniczeń opisujących wzajemne relacje między nimi  $\{C_{SC}, C_{\mathcal{P}}, C_D\}$ . Deklaratywny charakter modelu umożliwia wykorzystanie komercyjnie dostępnych technik programowania z ograniczeniami. Formułowane w jego ramach problemy decyzyjne są rozstrzygalne i NP-zupełne.

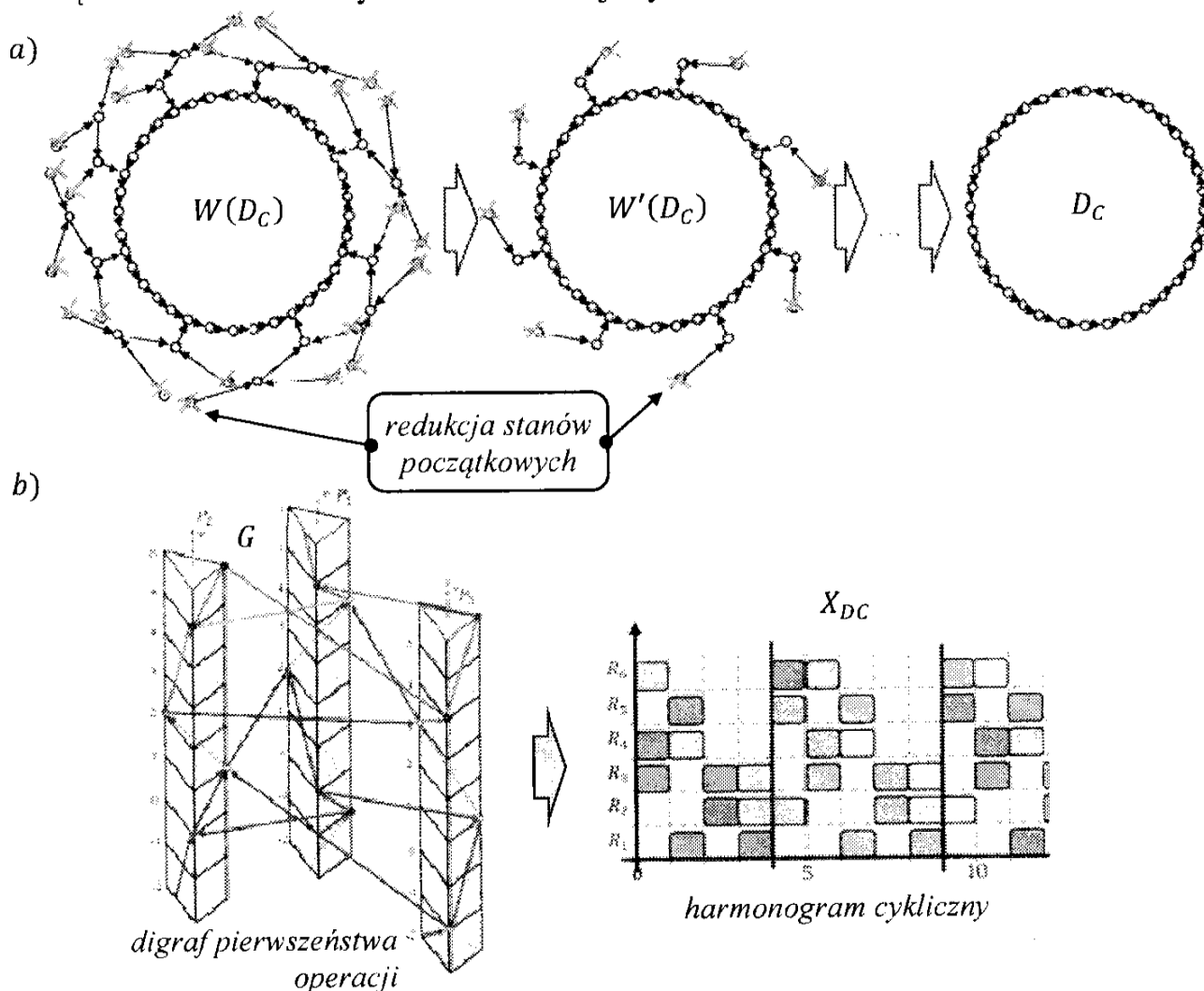


### 3. Harmonogramowanie SWMPC

Z faktu, że wspomniany wcześniej problem analizy (66) należy do problemów NP-trudnych, wynika potrzeba poszukiwania metod heurystycznego przeszukiwania przestrzeni  $\mathcal{P}$ , metod umożliwiających np. wyznaczenie przebiegów cyklicznych/wirów o parametrach spełniających oczekiwania użytkownika, w trybie on-line. W przyjętym podejściu problemy harmonogramowania SWMPC obejmują zarówno poszukiwanie przebiegów jak i odpowiadających im harmonogramów cyklicznych. Odpowiedzią na te potrzeby są dwie autorskie metody implementujące warunki odpowiednio Własności 1 i Twierdzenia 3. Pierwsza z nich umożliwia wyznaczenie w przestrzeni przebiegów  $\mathcal{P}$ , systemu o zadanej strukturze  $SC^{lp}$ , zbioru ustalonych przebiegów cyklicznych poprzez tzw. „redukcję” stanów początkowych. Druga natomiast pozwala na „bezpośrednie” generowanie zbioru harmonogramów cyklicznych.

Obie metody dedykowane są dla systemów typu:  $MW_{1,1}$ ,  $MW_{1,2}$ ,  $MW_{2,1}$ ,  $MW_{2,2}$  (patrz tab. 2.1) i pozwalają na efektywne (w czasie nieprzekraczającym 60 min) rozwiązywanie problemów o rozmiarach spotykanych w praktyce (np. związanych z analizą i/lub syntezą zachowań cyklicznych sieci routerów transmisji danych, systemów transportowych ESP, itp.).

Ideę działania omawianych metod ilustruje rys. 3.1.



**Rys. 3.1.** Metoda redukcji stanów początkowych a), metoda generowania harmonogramów cyklicznych w oparciu o digrafy pierwszeństwa operacji b)

### 3.1. Metoda redukcji stanów początkowych

Rozważana instancja problemu analizy (72) zawiera ograniczenia  $C_{STA}$ , spełnienie których gwarantuje występowanie w przestrzeni  $\mathcal{P}$  cyklicznych przebiegów ustalonych (lub bardziej ogólnie wirów):

$$PS_{STA} = ((\mathcal{P}, D_{\mathcal{P}}), C_{STA}), \quad (72)$$

gdzie:

- $\mathcal{P}$  – zmienna decyzyjna, przestrzeń stanów (zachowanie) SWMPC,
- $D_{\mathcal{P}}$  – dziedzina determinująca dopuszczalną postać przestrzeni stanów  $\mathcal{P}$ ,
- $C_{STA} = C_{SC} \cup C_{\mathcal{P}} \cup {}'\mathcal{P}$  zawiera przebiegi cykliczne'  $\cup C_{ST}$  – zbiór ograniczeń określających związki między zmiennymi decyzyjnymi:  $C_{SC}$  – ograniczenia określające rodzaj struktury (założenia SWMPC patrz podrozdział 2.1.1 oraz tab. 2.1);  $C_{\mathcal{P}}$  – ograniczenia narzucane na zachowanie: warunki dopuszczalności stanów (22) – (31), warunki przejścia między stanami  $\Lambda^{lp}$  (51);  $'\mathcal{P}$  zawiera przebiegi cykliczne' – ograniczenie dodatkowe zakładające istnienie niepustej przestrzeni cyklicznych przebiegów ustalonych,  $C_{ST}$  – ograniczenia określające postać struktury  $SC^{lp}$  (18) rozważanego systemu klasy SWMPC.

Rozwiązanie problemu (72), polega na wyznaczeniu przestrzeni  $\mathcal{P}$ , spełniającej wszystkie zadane ograniczenia tzn.:

- przejścia między stanami spełniają warunki przejścia  $\Lambda^{lp}$  (51),
- stany przestrzeni są stanami dopuszczalnymi w systemie o strukturze  $SC^{lp}$  (18),
- przestrzeń zawiera cykliczne przebiegi ustalone (wchodzące w skład wirów).

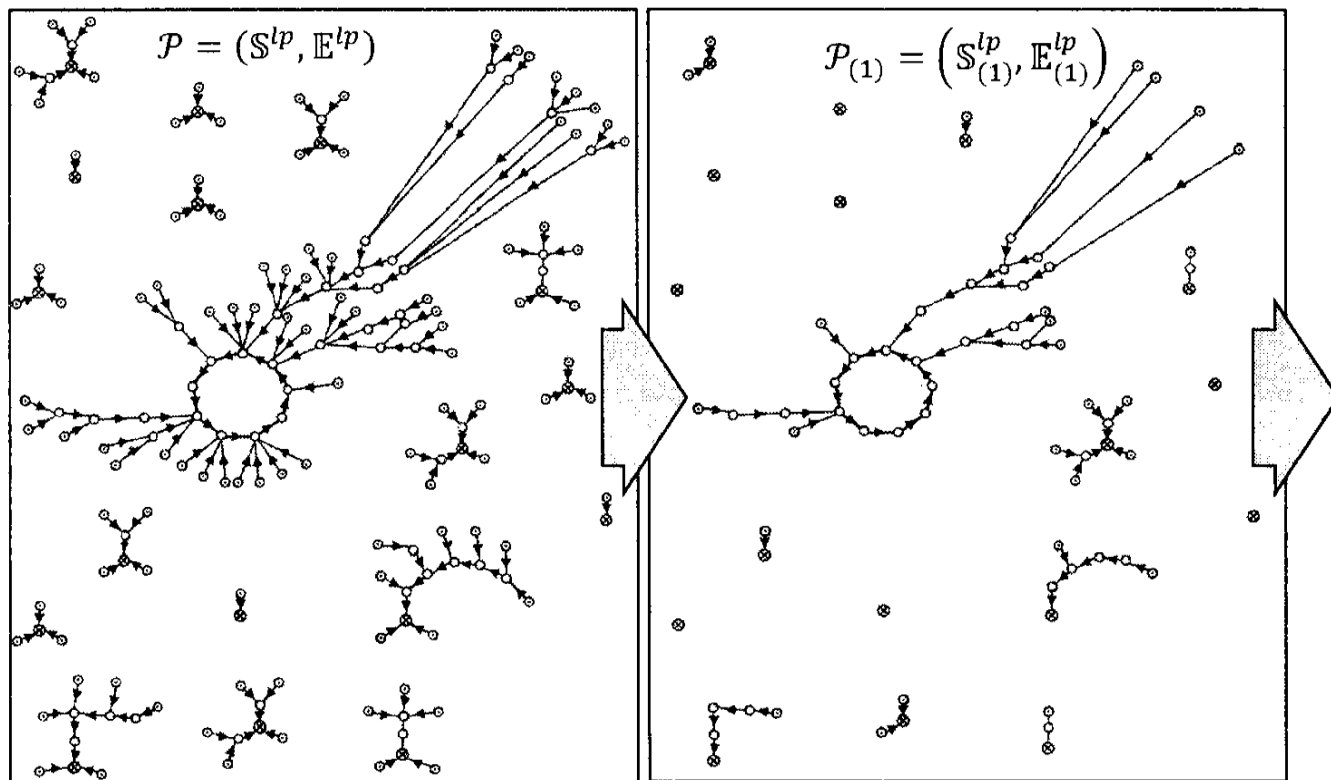
Jak już wspomniano, ze względu na NP-trudny charakter problemu, metody przeglądu zupełnego komponentów spójności (drzew, wirów) przestrzeni stanów  $\mathcal{P}$  okazują się być nieefektywnymi czasowo. Zaproponowane rozwiązanie umożliwia wyznaczanie wyłącznie takich obszarów przestrzeni  $\mathcal{P}$ , które prowadzą do przebiegów cyklicznych (wirów). Dwie wersje rozważanej metody redukcji stanów sprowadzają się odpowiednio do algorytmu generującego wszystkie wiry przestrzeni  $\mathcal{P}$  oraz do algorytmu generującego wiry o zadanej wartości okresu cyklicznego przebiegu ustalonego.

#### 3.1.1. Wyznaczanie przebiegów cyklicznych o dowolnym okresie $\alpha_H$

Proponowane podejście zakłada iteracyjne usuwanie stanów początkowych  $SB$  (tzn. stanów nieposiadających rodziców) z przestrzeni stanów  $\mathcal{P}$ . Postępowanie z tym związane ilustruje następujący przykład.

Niech w zbiorze stanów dopuszczalnych  $S^{lp}$ , systemu o strukturze  $SC^{lp}$ , istnieje niepusty podzbiór stanów początkowych:  $SB_{(1)} \neq \emptyset$  – gdzie indeks dolny odnosi się do numeru kolejno realizowanych kroków proponowanej procedury. W pierwszym kroku przyjmuje się, że wszystkie stany  $SB_{(1)}$  są niedopuszczalne. Oznacza to, że do ograniczeń określających dopuszczalność stanów (22) – (31) dodaje się ograniczenia

wykluczające stany  $SB_{(1)}$ . W ten sposób otrzymuje się nowy zbiór  $\mathbb{S}_{(1)}^{lp} \subset \mathbb{S}^{lp}$  pozabawiony stanów początkowych:  $\mathbb{S}_{(1)}^{lp} \cup SB_{(1)} = \mathbb{S}^{lp}$ . W reprezentacji grafowej oznacza to, że wszystkie digrafy zachowań przestrzeni  $\mathcal{P}$  zredukowane zostały o stany początkowe – co prowadzi do przestrzeni  $\mathcal{P}_{(1)}$ . Przykład redukcji stanów początkowych dla przestrzeni zachowań procesów lokalnych z rys. 2.14 został zilustrowany na rys. 3.2.



Legenda:

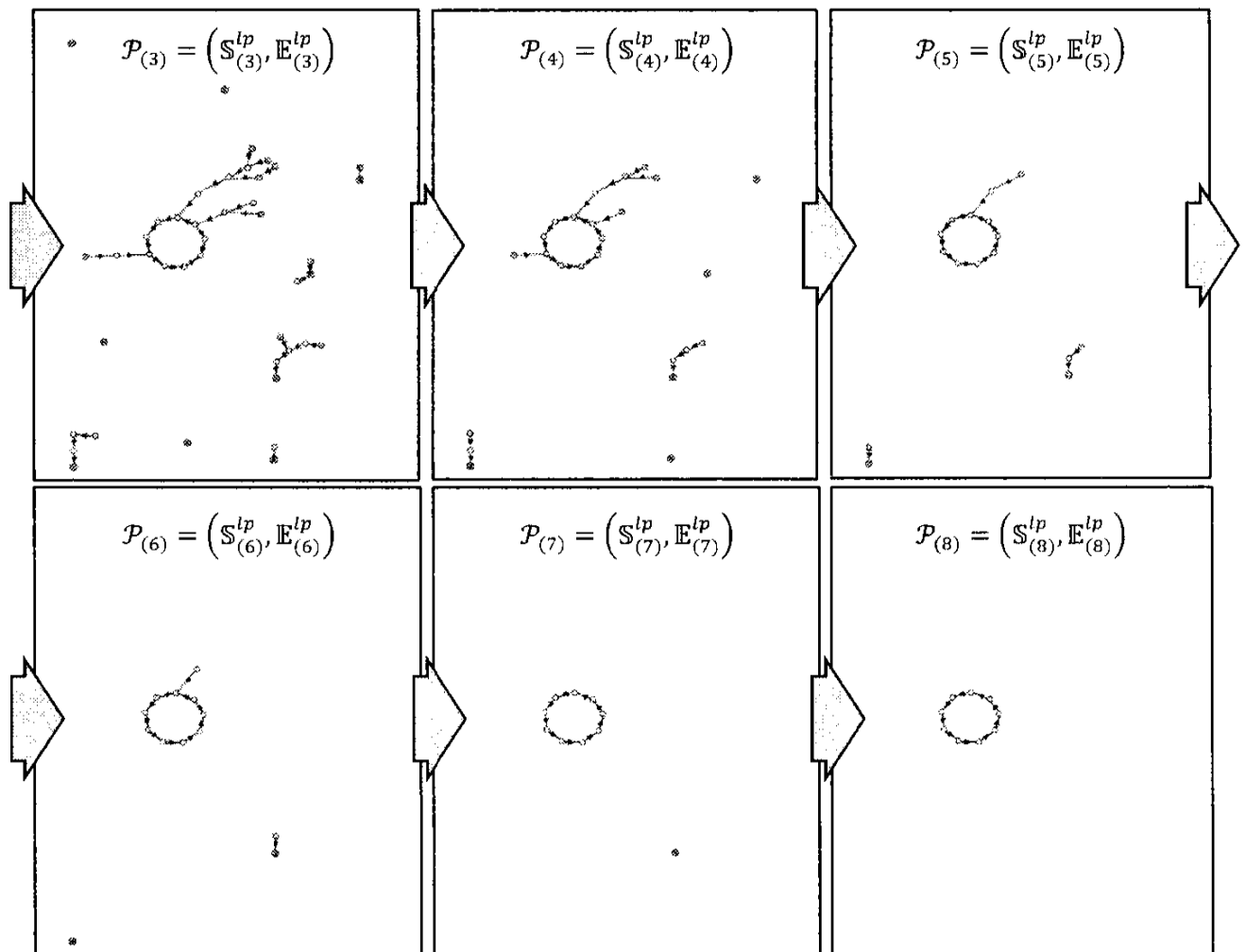
⊛ – stan blokadowy    ⊙ – redukowany stan początkowy    ○ – stan dopuszczalny

**Rys. 3.2.** Redukcja stanów początkowych  $SB_{(1)}$  zbioru  $\mathbb{S}^{lp}$  – pierwszy krok procedury wyznaczania cyklicznych przebiegów ustalonych (dla systemu z rys. 2.6)

Jak widać usunięcie stanów początkowych skutkuje skróceniem wszystkich przebiegów przejściowych, a każdy potomek stanu początkowego przestrzeni  $\mathcal{P}$  staje się stanem początkowym przestrzeni  $\mathcal{P}_{(1)}$ .

W kolejnym kroku ponownie wyznacza się zbiór stanów początkowych  $SB_{(2)}$ , jednak tym razem w zbiorze stanów  $\mathbb{S}_{(1)}^{lp}$  (zbiór stanów przestrzeni  $\mathcal{P}_{(1)}$ ). Stany początkowe (o ile istnieją) są w tym przypadku potomkami stanów zbioru  $SB_{(1)}$ . Jeżeli  $SB_{(2)}$  jest zbiorem niepustym  $SB_{(2)} \neq \emptyset$ , to podobnie jak w kroku pierwszym uznaje się je za stany niedopuszczalne, tzn. do ograniczeń (22) – (31) dodaje się kolejne ograniczenia wykluczające stany zbioru  $SB_{(2)}$ . W wyniku redukcji stanów zbioru  $SB_{(2)}$  otrzymuje się kolejny zbiór  $\mathbb{S}_{(2)}^{lp} \subset \mathbb{S}_{(1)}^{lp}$  (i tym samym przestrzeń  $\mathcal{P}_{(2)}$ ), dla którego cała procedura powtarzana jest ponownie. Warunek stopu sprawdza zawartość zbioru stanów początkowych. Jeżeli w  $i$ -tym kroku zbiór stanów początkowych jest pusty  $SB_{(i)} = \emptyset$ , to zgodnie z Twierdzeniem 3, każdy stan zbioru  $\mathbb{S}_{(i-1)}^{lp}$  (pod warunkiem, że zbiór ten nie jest pusty) jest stanem należącym do przebiegu cyklicznego.

Uzyskany w ten sposób zbiór  $\mathbb{S}_{(i-1)}^{lp} \subset \mathbb{S}^{lp}$  jest podzbiorem  $\mathbb{S}^{lp}$ , zawierającym tylko stany należące do cyklicznych przebiegów ustalonych. Kolejne kroki procedury wyznaczania przebiegów cyklicznych z rys. 3.2 zostały zilustrowane na rys. 3.3.



Legenda:

● – stan blokadowy    ○ – redukowany stan początkowy    ◦ – stan dopuszczalny

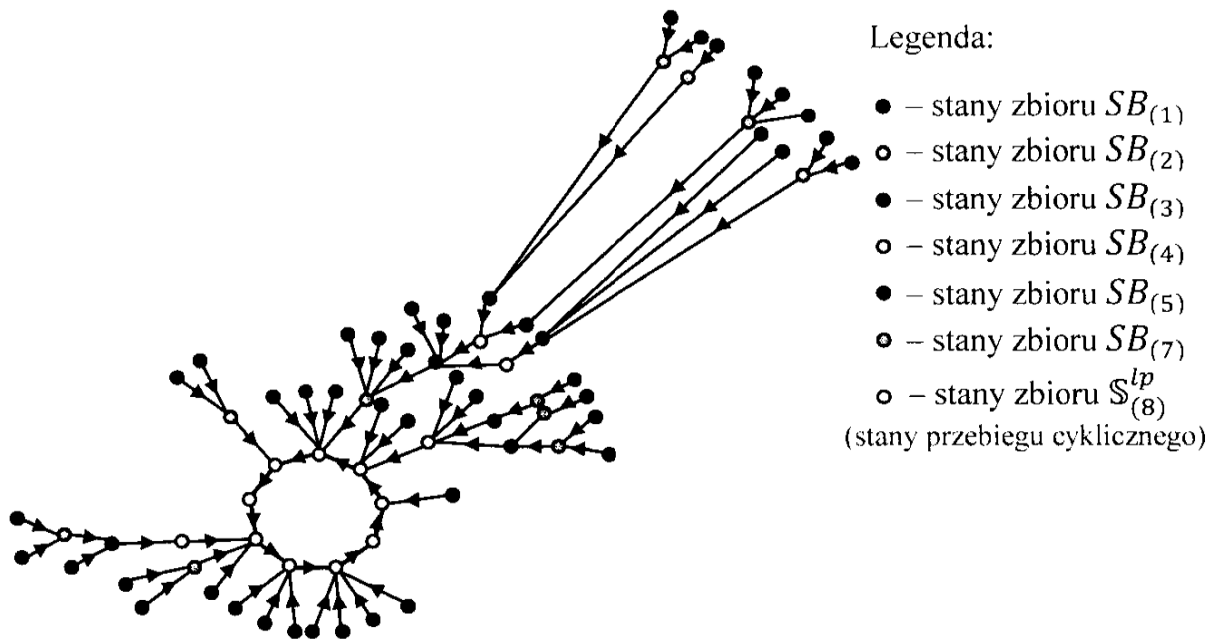
Rys. 3.3. Kolejne kroki redukcji stanów początkowych (kontynuacja rys. 3.2)

W ostatnim kroku otrzymano przestrzeń  $\mathcal{P}_{(8)}$ , w której nie występują już żadne stany początkowe. Oznacza to, że wszystkie stany zbioru  $\mathbb{S}_{(8)}^{lp}$  są częścią przebiegów cyklicznych (w tym przypadku jednego przebiegu). Znając stany należące do przebiegu cyklicznego oraz zbiory stanów początkowych  $SB_{(i)}$  kolejnych kroków, możliwe jest w dalszym etapie odtworzenie wszystkich wirów występujących w przestrzeni  $\mathcal{P}$ . Przykład odtworzenia wiru ze stanów zbioru  $\mathbb{S}_{(8)}^{lp}$ , oraz stanów początkowych uzyskiwanych w kolejnych krokach procedury  $SB_{(1)} - SB_{(7)}$ , został przedstawiony na rys. 3.4.

Najważniejszą rolę w przedstawionym postępowaniu odgrywa etap wyznaczania zbioru stanów początkowych. Kluczowym zatem staje się pytanie o sposób w jaki można wyznaczać zbiory  $SB_{(i)}$ ?

Warto zauważyć, że tak naprawdę w żadnym z etapów wyznaczania zbiorów  $SB_{(i)}$  nie jest konieczna znajomość zbioru stanów dopuszczalnych  $\mathbb{S}_{(i-1)}^{lp}$  (zbiory te są wy-

korzystane wyłącznie do ilustracji idei odcinania stanów początkowych). Do wyznaczania zbiorów stanów początkowych wystarcza znajomość funkcji przejścia  $\delta^{lp}$  (determinowanej przez  $\Lambda^{lp}$  (51)) oraz ograniczeń określających ich dopuszczalność (22) – (31).



Rys. 3.4. Odtworzenie wiru ze stanów przebiegu cyklicznego  $S_{(8)}^{lp}$  oraz stanów początkowych  $SB_{(1)} - SB_{(7)}$

Zgodnie z przyjętą w poprzednim rozdziale definicją, stan początkowy to stan, który nie posiada rodzica. Oznacza to, że na zbiór  $SB$  składają się stany dopuszczalne niebędące wartością funkcji przejścia  $\delta^{lp}$ . W tym ujęciu  $SB$  może być traktowany jako zbiór wszystkich rozwiązań problemu spełniania ograniczeń:

$$PS_{SB} = ((S^r, D_S), C_{SB}), \quad (73)$$

gdzie:

$S^r$  – zmienna decyzyjna, stan początkowy  $S^r \in S^{lp}$ ,

$D_S$  – dziedzina określająca (zgodnie z (19)) dopuszczalne wartości zmiennych (alokacja, semaforey, indeksy) charakteryzujących stan  $S^r$ ,

$C_{SB} = C_N \cup \{S^r \neq \delta^{lp}(S^e), \forall S^e \in S^{lp}\}$  – zbiór ograniczeń:

$C_N$  – ograniczenia dopuszczalności (22)–(31) stanów  $S^e$  i  $S^r$ , ograniczenia determinowane przez strukturę systemu  $SC^{lp}$ ;

$\{S^r \neq \delta^{lp}(S^e), \forall S^e \in S^{lp}\}$  – ograniczenie gwarantujące, że  $S^r$  nie jest potomkiem żadnego stanu dopuszczalnego  $S^e$  (spełniającego ograniczenia  $C_N$ ).

Rozwiązaniem problemu  $PS_{SB}$  (73) jest więc stan dopuszczalny  $S^r$ , dla którego nie istnieje żaden inny stan dopuszczalny  $S^e$ , będący jego rodzicem:  $S^r \neq \delta^{lp}(S^e), \forall S^e \in SB$ . Tak wyznaczony  $S^r$  spełnia założenia stanu początkowego, a zatem zbiór wszystkich, wyznaczonych w ten sposób, rozwiązań problemu  $PS_{SB}$  jest zbiorem  $SB$ .

Problem (73) można wykorzystać do wyznaczania zbiorów stanów początkowych  $SB_{(i)}$   $i$ -tego kroku omawianej metody. W tym celu problem (73) rozszerzany jest

o dodatkowy zbiór ograniczeń  $C_{SB}(i)$  wykluczających ze zbioru stanów dopuszczalnych wszystkie znane stany początkowe (wyznaczone w krokach poprzednich  $1 \dots (i - 1)$ ):

$$PS_{SB(i)} = \left( (S^r, D_S), C_{SB(i)} \right), \quad (74)$$

gdzie:

$S^r, D_S$  – definiowane jak w (73),

$C_{SB(i)} = C_N \cup \{S^r \neq \delta^{lp}(S^e), \forall S^e \in \mathbb{S}^{lp}\} \cup C_{SB}(i - 1)$  – zbiór ograniczeń:

$C_N, \{S^r \neq \delta^{lp}(S^e), \forall S^e \in \mathbb{S}^{lp}\}$  – definiowane jak w (73),

$C_{SB}(i - 1) = \{S^r \notin SB_{(1)}, S^e \notin SB_{(1)}, S^r \notin SB_{(2)}, S^e \notin SB_{(2)}, \dots, S^r \notin SB_{(i-1)}, S^e \notin SB_{(i-1)}\}$  – zbiór ograniczeń wykluczających ze zbioru rozwiązań stany początkowe wyznaczone w krokach  $1 \dots (i - 1)$ .

Rozwiązując tego typu problem można wyznaczyć wszystkie stany początkowe zbioru  $\mathbb{S}_{(i)}^{lp}$  (zbiór stanów dopuszczalnych dla  $i$ -tego kroku procedury). Oznacza to, że wykorzystując ten problem w iteracyjnej procedurze (rozwiązanie problemu (74) kolejno dla  $i = 1, 2, 3 \dots$ ) można, wyznaczyć zbiór stanów tworzących wszystkie przebiegi cykliczne systemu. Stosowany algorytm ma postać:

---

### Algorytm 1

**function** CYCLICSTEADYSTATEGENERATION( $D_S, C_N, \delta^{lp}$ )

$i \leftarrow 0$

$C_{SB}(0) \leftarrow \emptyset$

**do**

$i \leftarrow i + 1$

$PS_{SB(i)} \leftarrow \left( (S^r, D_S), C_N \cup \{S^r \neq \delta^{lp}(S^e), \forall S^e \in \mathbb{S}^{lp}\} \cup C_{SB}(i - 1) \right)$

$SB_{(i)} \leftarrow \text{SEARCHALL}(PS_{SB(i)})$

$C_{SB}(i) \leftarrow C_{SB}(i - 1) \cup \{S^r \notin SB_{(i)}, S^e \notin SB_{(i)}\}$

**while**  $SB_{(i)} \neq \emptyset$

$PS_C \leftarrow \left( (S^r, D_S), C_N \cup C_{SB}(i) \right)$

$V(DC) \leftarrow \text{SEARCHALL}(PS_C)$

**return**  $V(DC)$

**end**

---

gdzie:

$D_S, C_N, \delta^{lp}, C_{SB}(i - 1)$  – definiowane jak w (73), (74),

$PS_{SB(i)}$  – problem spełniania ograniczeń (74),

$SB_{(i)}$  – zbiór stanów początkowych  $i$ -tego kroku,

$V(DC)$  – zbiór stanów tworzących wszystkie przebiegi cykliczne  $DC$ , osiągalne w systemie o strukturze  $SC^{lp}$ ,

$\text{SEARCHALL}(PS)$  – funkcja zwracająca zbiór wszystkich rozwiązań problemu  $PS$ , (w przypadku braku rozwiązania, funkcja zwraca zbiór pusty).

Danymi wejściowymi Algorytmu 1 są ograniczenia dopuszczalności stanów (22) – (31), funkcja przejścia  $\delta^{lp}$  oraz dziedzina  $D_S$  określająca dopuszczalne wartości zmiennych opisujących stany systemu. Każda z tych wielkości jest determinowana przez przyjętą postać struktury  $SC^{lp}$  (18) oraz typ SWMPC (tab. 2.1).

Wynikiem zwracanym przez algorytm jest zbiór stanów  $V(DC)$ , który nie zawiera stanów początkowych. W takim przypadku możliwe są dwa warianty wyników:

- $V(DC) = \emptyset$  – oznacza, że w systemie o strukturze  $SC^{lp}$  cykliczne przebiegi ustalone są nieosiągalne,
- $V(DC) \neq \emptyset$  – oznacza, że każdy stan tego zbioru stanowi element cyklicznego przebiegu ustalonego.

Zbiór  $V(DC)$  jest uzyskiwany w wyniku rozwiązania problemu spełniania ograniczeń  $PS_C$ :

$$PS_C = ((S^r, D_S), C_N \cup C_{SB}(i)). \quad (75)$$

Do rozwiązania tego problemu wykorzystuje się funkcję  $\text{SEARCHALL}$ , która stanowi reprezentację procedur implementowanych w środowiskach programowania z ograniczeniami. Środowiska tego typu: ILOG, OzMozart, ECL<sup>i</sup>PS<sup>e</sup>, etc. są najczęściej wykorzystywanymi narzędziami rozwiązywania problemów spełniania ograniczeń. Choć procedury implementowane w tych środowiskach mogą się od siebie różnić, przykładem mogą tu być środowiska OzMozart i ECL<sup>i</sup>PS<sup>e</sup> wykorzystujące różne silniki obliczeniowe, to ich ogólna, obserwowana w praktyce wydajność jest bardzo podobna (są one tego samego rzędu  $\mathcal{O}$ ). Z tego też powodu, w ramach algorytmu nie są wyróżniane narzędzia wykorzystywane do rozwiązywania PSO, wskazywana jest jedynie funkcja ( $\text{SEARCHALL}$ ) odpowiedzialna za jego rozwiązanie.

Problem  $PS_C$  (75) podobnie jak  $PS_{SB(i)}$  (74) zawiera ograniczenia dopuszczalności stanów  $C_N$  oraz ograniczenia wykluczające stany początkowe  $C_{SB}(i)$ . W odróżnieniu jednak od  $PS_{SB(i)}$ , w problemie (75) nie występują ograniczenia narzucane na istnienie rodziców stanów:  $S^r \neq \delta^{lp}(S^e)$ ,  $\forall S^e \in \mathbb{S}^{lp}$ . Oznacza to, że rozwiązaniem  $PS_C$  jest każdy stan dopuszczalny, który nie został „zredukowany” (ograniczenia  $C_{SB}(i)$ ) w wyniku zakwalifikowania go do stanów początkowych – czyli zgodnie z Twierdzeniem 3 stan składający się na cykliczny przebieg ustalony.

W tym miejscu należy zwrócić uwagę na bardzo istotną cechę proponowanego algorytmu, skoro zbiór  $V(DC)$  zawiera stany tworzące wszystkie przebiegi cykliczne osiągalne w systemie o strukturze  $SC^{lp}$  i każdy z tych stanów jest rozwiązaniem problemu  $PS_C$ , **to ograniczenia  $C_{SB}(i)$  tego problemu należy traktować jako warunki wystarczające, gwarantujące występowanie cyklicznych przebiegów ustalonych.** Zbiór tego typu ograniczeń  $C_{SB}(i)$  jest determinowany przez zbiory stanów początkowych  $SB(i)$  uzyskane w efekcie wielokrotnego rozwiązania problemu (74).

**Przykład 5.** Ilustracja wykorzystania Algorytmu 1 do wyznaczania cyklicznych przebiegów ustalonych routera usługi VOD.

Dany jest system (rys. 1.32) reprezentujący router usługi VOD (rys. 1.31). W routerze odbywa się transmisja danych pomiędzy portami A–C oraz B–D. Strumienie pakietów (oznaczone liniami koloru niebieskiego i czerwonego) reprezentowane są przez dwa jednopotokowe procesy multimodalne:  $mP_1 = \{P_1^1\}$ ,  $mP_2 = \{P_2^1\}$ . Transmisja pakietów odbywa się za pośrednictwem czterech jednopotokowych procesów lokalnych (jak na rys. 1.31):  $P_1 = \{P_1^1\}$ ;  $P_2 = \{P_2^1\}$ ,  $P_3 = \{P_3^1\}$ ,  $P_4 = \{P_4^1\}$ . W systemie występuje osiem zasobów:  $R = \{R_1, R_2, \dots, R_8\}$ , gdzie zasoby  $R_1 - R_4$  reprezentują kolejno porty A, B, C, D, a  $R_5 - R_8$  klasyfikowane są jako zasoby wewnętrzne.

Rozważany system należy do klasy  $MW_{1,1}$  (tab. 2.1), co oznacza:

- czasy wykonania operacji (zarówno procesów lokalnych jak i multimodalnych) są jednostkowe:  $t_{i,j} = 1$ ,  $mt_{i,j} = 1$ ,
- procesy multimodalne synchronizowane są w oparciu o protokół wzajemnego wykluczania (tylko jeden pakiet może zajmować dany zasób),
- pojemności zasobów są jednostkowe:  $C(R_k) = 1$ ,  $k = 1 \dots 8$ ,
- zapotrzebowanie procesów lokalnych i multimodalnych na wykorzystywane przez siebie zasoby jest jednostkowe:  $\omega_{i,j}^h = 1$ ,  $m^l \omega_{i,j}^h = 1$ , (tzn. w trakcie wykonywania operacji wykorzystywany jest cały zasób).

Ponadto przyjęto następujące reguły priorytetowania:

$\theta^0 = \{\sigma_1^0, \dots, \sigma_8^0\}$  (dla procesów lokalnych):

$$\sigma_1^0 = (P_2^1, P_3^1), \sigma_2^0 = (P_1^1, P_2^1), \sigma_3^0 = (P_1^1, P_4^1), \sigma_4^0 = (P_3^1, P_4^1),$$

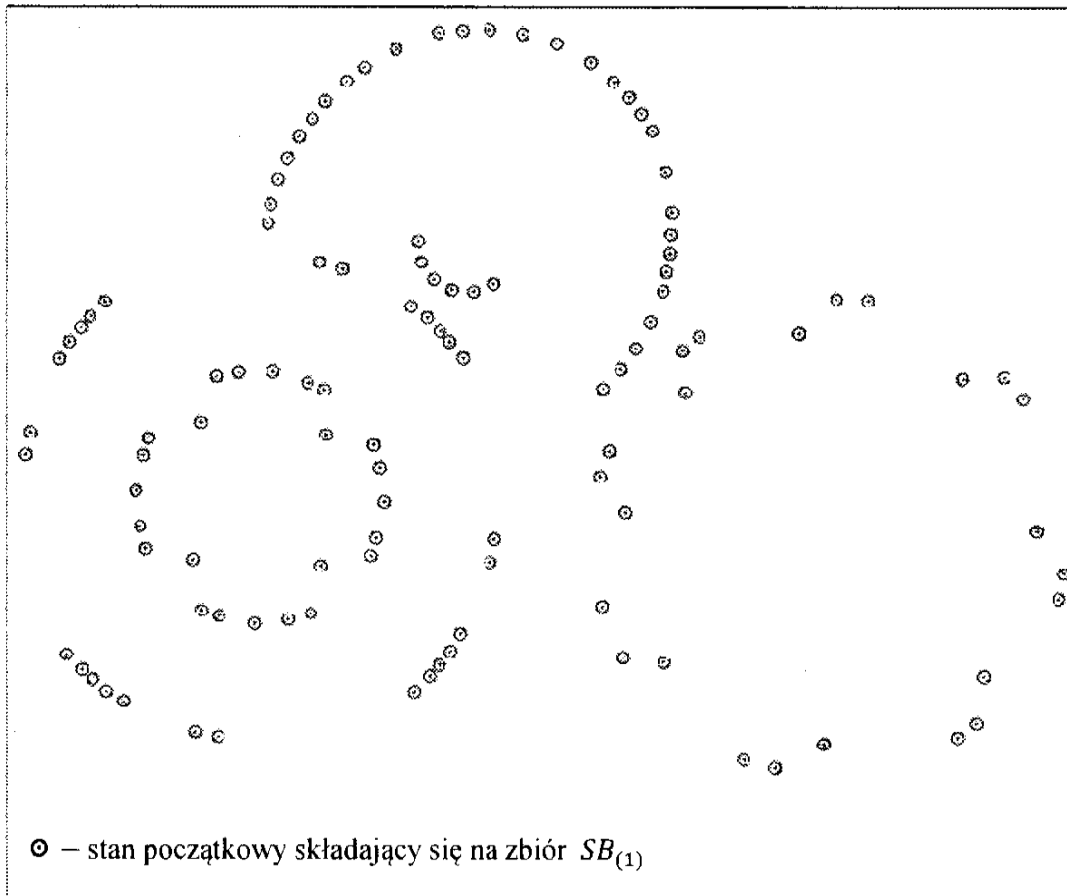
$\theta^1 = \{\sigma_1^1, \dots, \sigma_8^1\}$  (dla procesów multimodalnych):

$$\sigma_1^1 = (mP_2^1, mP_1^1), \sigma_2^1 = (mP_1^1, mP_2^1), \sigma_3^1 = (mP_1^1, mP_2^1), \sigma_4^1 = (mP_2^1, mP_1^1).$$

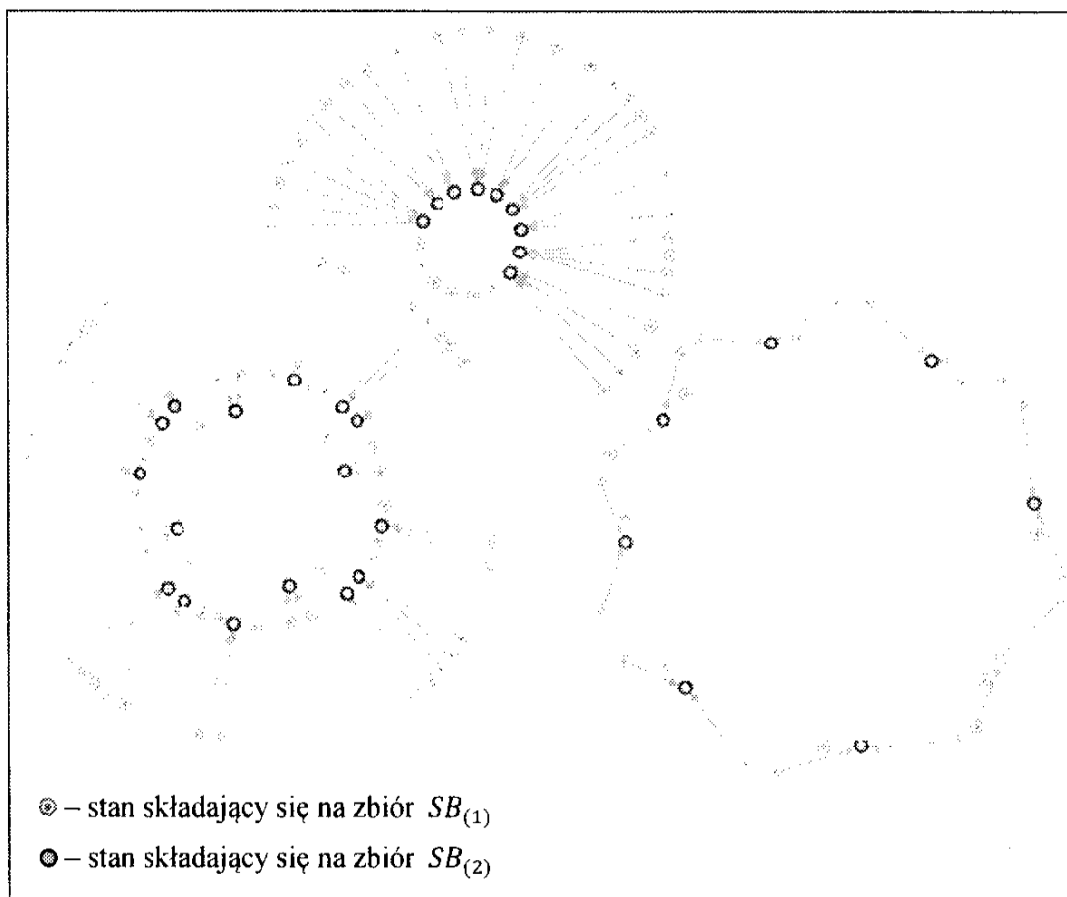
Dla tak zdefiniowanego systemu poszukiwana jest odpowiedź na pytanie: Czy w routerze z rys. 1.31 możliwa jest cykliczna transmisja pakietów? Do wyznaczenia wszystkich możliwych cyklicznych przebiegów ustalonych wykorzystany został Algorytm 1. W rozważanym przypadku wymagał on realizacji czterech kroków (rozwiązanie problemów:  $PS_{SB(1)}$ ,  $PS_{SB(2)}$ ,  $PS_{SB(3)}$ ,  $PS_C$ ), których wyniki przedstawiono na rys. 3.5 – 3.8. Rysunki 3.5 – 3.8 przedstawiają kolejne etapy odtwarzania przestrzeni  $\mathcal{P}$  w oparciu o wyznaczone zbiory stanów początkowych  $SB_{(i)}$ . W szczególności rys. 3.5 – 3.7 przedstawiają rozwiązania składające się na przebiegi przejściowe, natomiast rys. 3.8 przedstawia wyznaczone w ostatnim kroku cykliczne przebiegi ustalone.

Ocena istnienia przebiegów cyklicznych wymagała wyznaczenia 236 stanów. Całkowity czas obliczeń nie przekroczył 3 sekund (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM). Jak widać, w systemie osiągalnych jest pięć przebiegów cyklicznych:  $D_{C,1}, D_{C,2}, \dots, D_{C,5}$ .

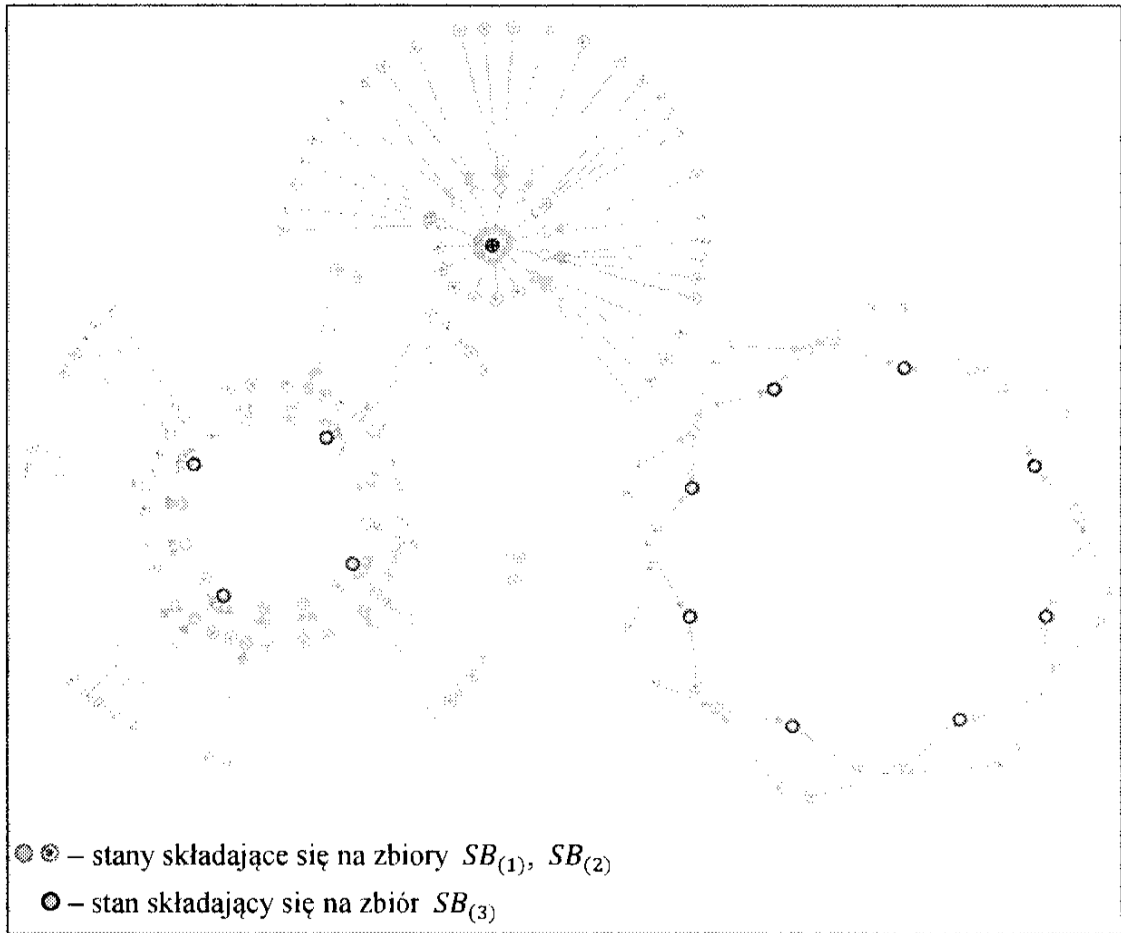




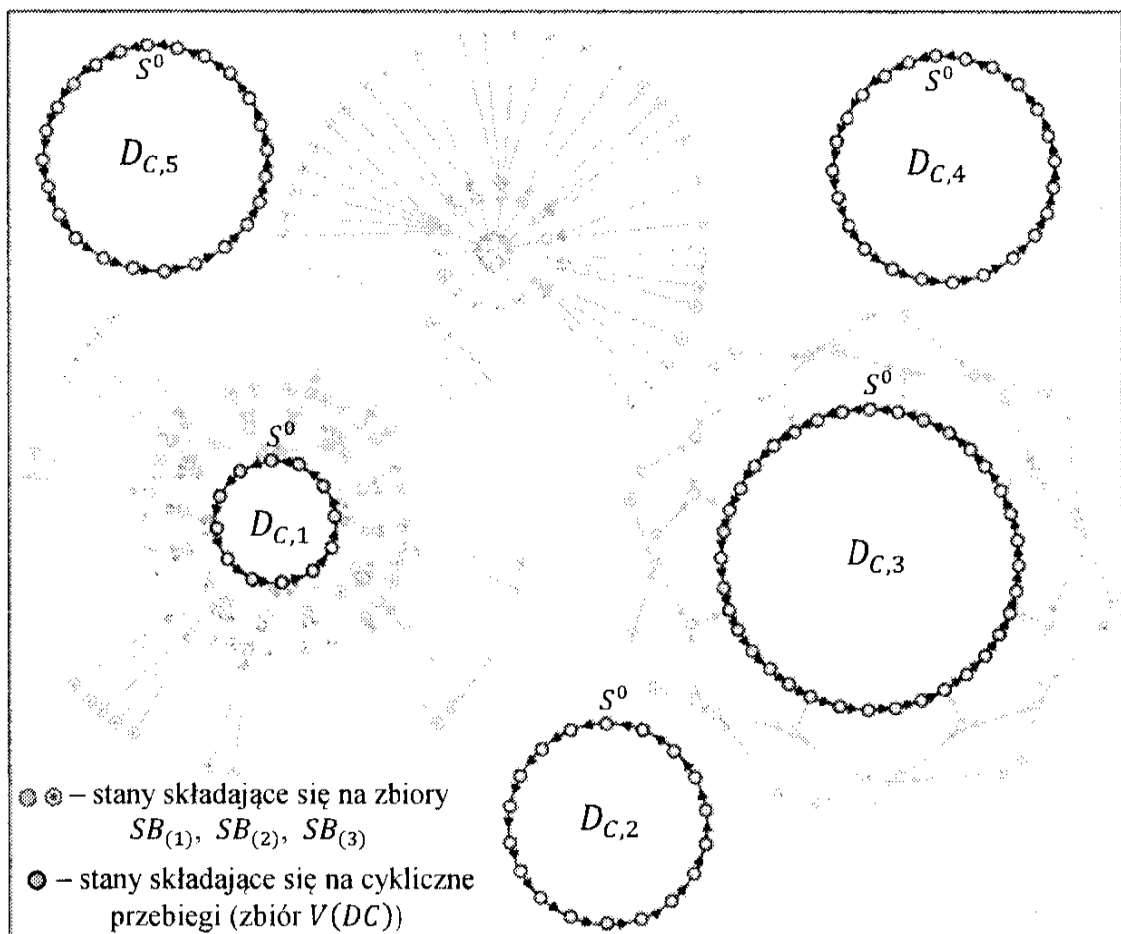
**Rys. 3.5.** Przestrzeń stanów początkowych  $SB_{(1)}$  (pierwszy krok Algorytmu 1:  $SB_{(1)} = \text{SEARCHALL}(PS_{SB_{(1)}})$ )



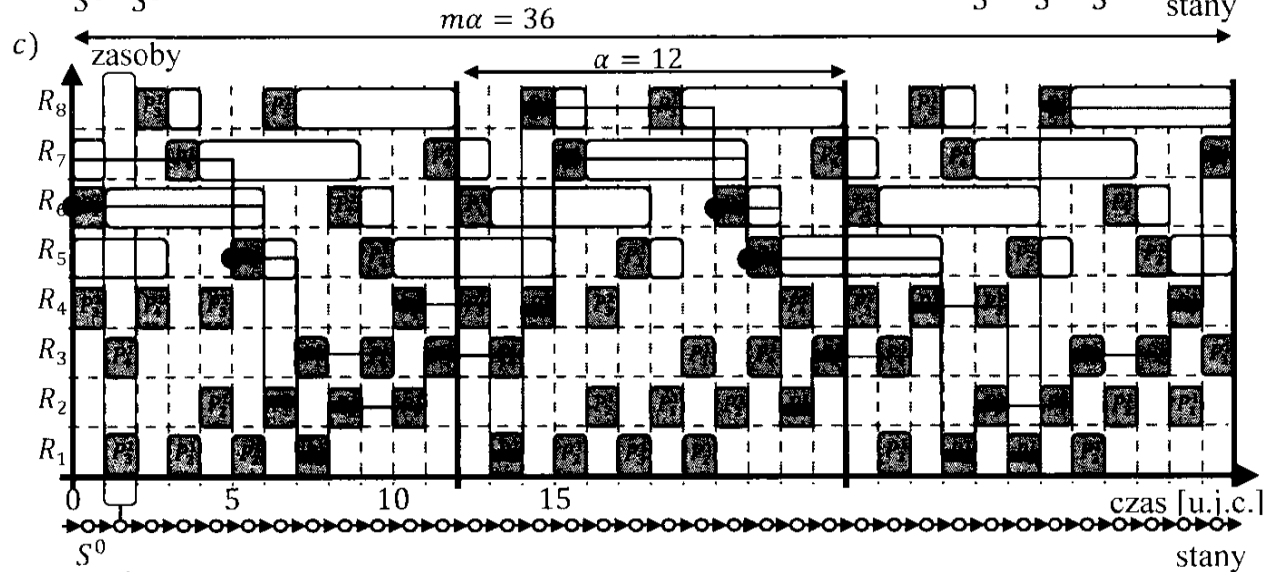
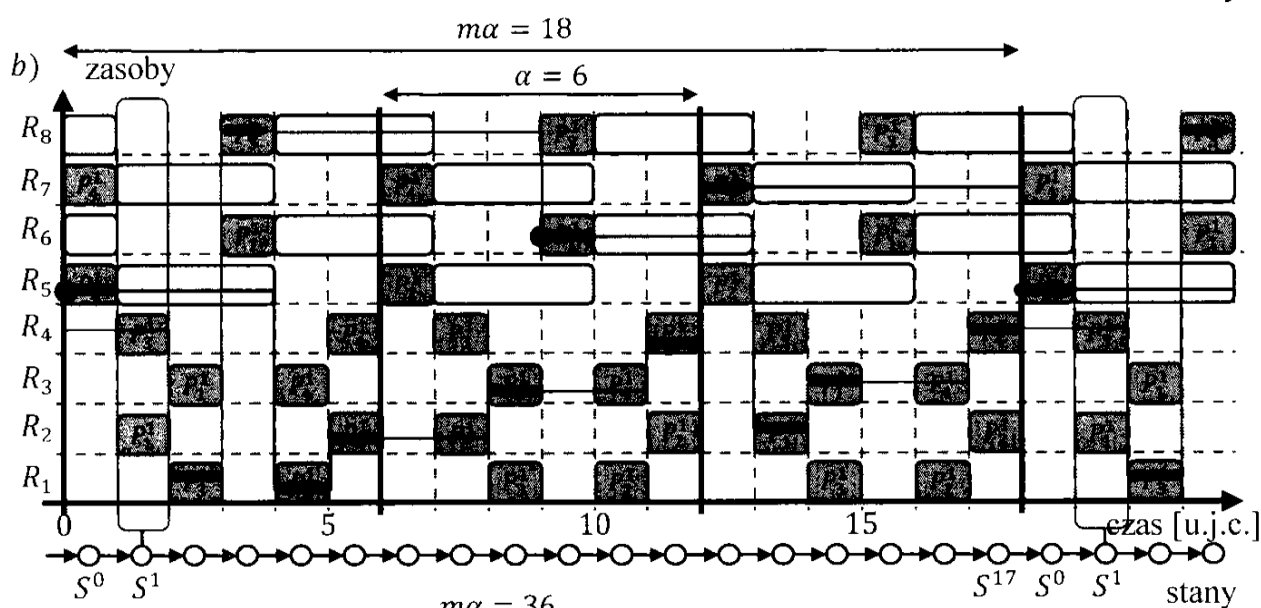
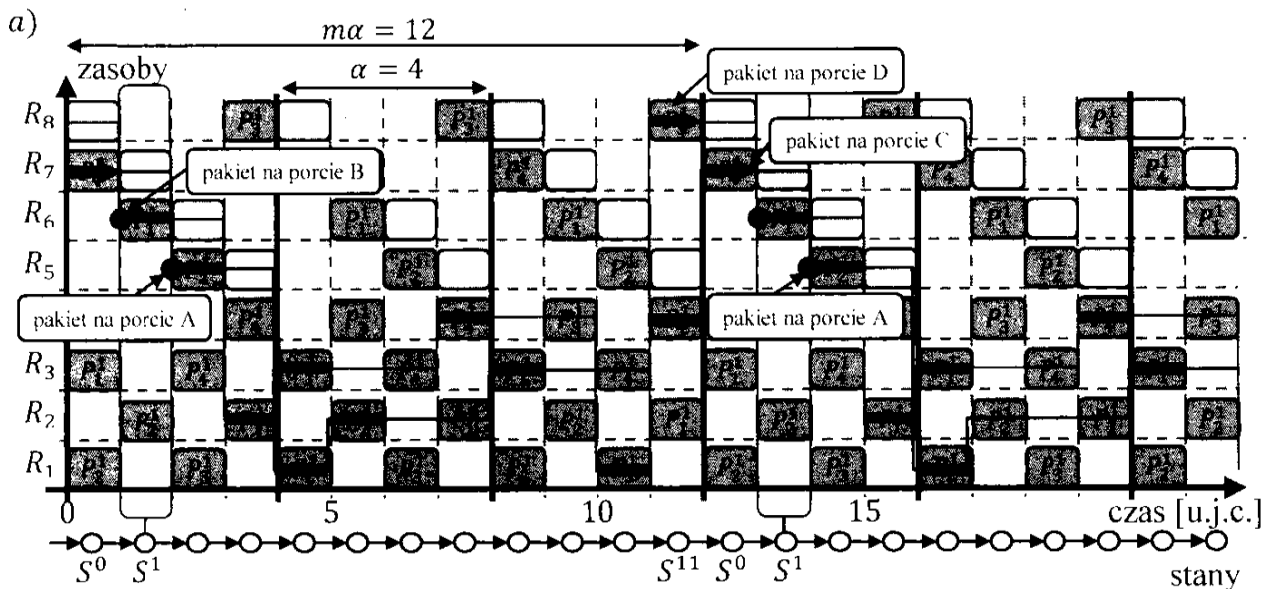
**Rys. 3.6.** Przestrzeń stanów  $SB_{(1)} \cup SB_{(2)}$  (drugi krok Algorytmu 1:  $SB_{(2)} = \text{SEARCHALL}(PS_{SB_{(2)}})$ )



Rys. 3.7. Przestrzeń stanów  $SB_{(1)} \cup SB_{(2)} \cup SB_{(3)}$  (krok 3:  $SB_{(3)} = \text{SEARCHALL}(PS_{SB(3)})$ )



Rys. 3.8. Osiągalne cykliczne przebiegi ustalone (krok 4:  $V(DC) = \text{SEARCHALL}(PS_C)$ )



Legenda

- realizacja operacji potoku  $P_i^j$
- oczekiwane potoku  $P_i^j$  na zasobie
- $s^i$  -  $i$ -ty stan systemu
- realizacja operacji kolejno procesów  $mP_1, mP_2$
- oczekiwanie procesów  $mP_1, mP_2$

Rys. 3.9. Harmonogramy cykliczne dla przebiegów:  $D_{C,1}$  a),  $D_{C,2}$  b),  $D_{C,3}$  c), z rys. 3.8

Oznacza to, że analizowany router może pracować w 5 trybach charakteryzujących się różnymi okresami realizacji procesów i co za tym idzie różnymi prędkościami transmisji danych. Innymi słowy, w SWMPC o zadanej strukturze (opisanej przez ustalone marszruty procesów, statyczne reguły priorytetowania, itp.) osiągalnych jest pięć różnych przejawów zachowania.

Harmonogramy odpowiadające przebiegom  $D_{C,1}$ ,  $D_{C,2}$ ,  $D_{C,3}$  przedstawione zostały na rys. 3.9a, 3.9b, 3.9c. Okresy tych przebiegów, wynoszące kolejno 12, 18 i 36 u.j.c., determinują potencjalne prędkości transmisji danych [pakiet/u.j.c.]. Dla przebiegów  $D_{C,1}$  i  $D_{C,2}$  jeden pakiet jest transmitowany w trakcie jednego okresu przebiegu, a zatem prędkość transmisji wynosi 1/12 oraz 1/18 [pakiet/u.j.c.]. W przypadku przebiegu  $D_{C,3}$ , ze względu na to, że w ramach jednego okresu transmitowane są dwa pakiety każdego strumienia, prędkość transmisji wynosi 1/18 [pakiet/u.j.c.]. Oznacza to, że w celu zagwarantowania największej prędkości transmisji danych pomiędzy portami A–C i B–D należy przyjąć realizację procesów zgodną z przebiegiem  $D_{C,1}$ . Przebieg  $D_{C,1}$  nie gwarantuje jednak analogicznej prędkości dla innych, niż właśnie rozważanych, kierunków transmisji danych (A–B, B–C, itp.).

Prezentowany przykład ilustruje sposób wykorzystania Algorytmu 1 w procesie wyznaczania cyklicznych przebiegów ustalonych (wirów), w systemie o znanej strukturze.

### 3.1.2. Wyznaczanie przebiegów cyklicznych o ustalonym okresie $\alpha_H$

Prezentowana postać problemu (72) ma charakter ogólny, tzn. zakłada samo istnienie cyklicznych przebiegów ustalonych. W ogólności istnieje możliwość wprowadzania dodatkowych ograniczeń, niosących za sobą konieczność modyfikacji Algorytmu 1, pozwalającą poszukiwać przebiegów cyklicznych o okresach nieprzekraczających zadanej wartości  $\alpha_H$ . Innymi słowy, pozwala rozważać problem analizy w kontekście pytania:

*Czy w systemie o strukturze  $SC^{lp}$  (charakteryzowanej przez  $C_{SC}, C_P$ ) osiągalna jest przestrzeń  $\mathcal{P}$  zawierająca przebiegi cykliczne o okresach nie większych niż  $\alpha_H$ ?*

W sformułowaniu odpowiedniego algorytmu pomocnym okazuje się być Twierdzenie 4, zgodnie z którym przebiegi przejściowe nie mogą być dłuższe (zawierać więcej stanów) niż przebiegi cykliczne do których one prowadzą. Oznacza to, że w trakcie poszukiwań przebiegów cyklicznych można odrzucać wszystkie zbyt „długie” przebiegi przejściowe (tzn. składające się z większej liczby stanów niż dwukrotna wartość  $\alpha_H$ ). Fakt ten pozwala na uproszczenie Algorytmu 1 – wyznaczanie kolejnych potomków stanów początkowych odbywa się bądź to do chwili ponownego napotkania jednego z nich (stany tworzą przebieg cykliczny), bądź też do chwili gdy liczba ich przekroczy dwukrotną wartość  $\alpha_H$  (stany prowadzą do blokady lub przebiegu cyklicznego z okresem większym niż  $\alpha_H$ ). Odpowiedni algorytm ma następującą postać:

**Algorytm 2**


---

```

function CYCLICSTEADYSTATEALPHAGENERATION( $D_S, C_N, \delta^{lp}, \alpha_H$ )
   $PS_{SB} \leftarrow ((S^r, D_S), C_N \cup \{S^r \neq \delta^{lp}(S^e), \forall S^e \in \mathbb{S}^{lp}\})$ 
   $SB \leftarrow \text{SEARCHALL}(PS_{SB})$ 
   $V(DC) \leftarrow \emptyset$ 

  for  $i = 1$  to  $|SB|$ 
     $S^{i,1} \leftarrow \text{FIRSTELEMENT}(SB)$ 
     $SB \leftarrow SB \setminus S^{i,1}$ 
     $V^i \leftarrow \{S^{i,1}\}$ 
    for  $j \leftarrow 2$  to  $2\alpha_H$ 
       $S^{i,j} \leftarrow \delta^{lp}(S^{i,j-1})$ 
      if  $S^{i,j} \notin V^i$  then  $V^i = V^i \cup \{S^{i,j}\}$ 
    end
     $V(DC^i) \leftarrow \text{CHECKCYCLIC}(V^i)$ 
     $V(DC) \leftarrow V(DC) \cup V(DC^i)$ 
  end

  return  $V(DC)$ 
end

```

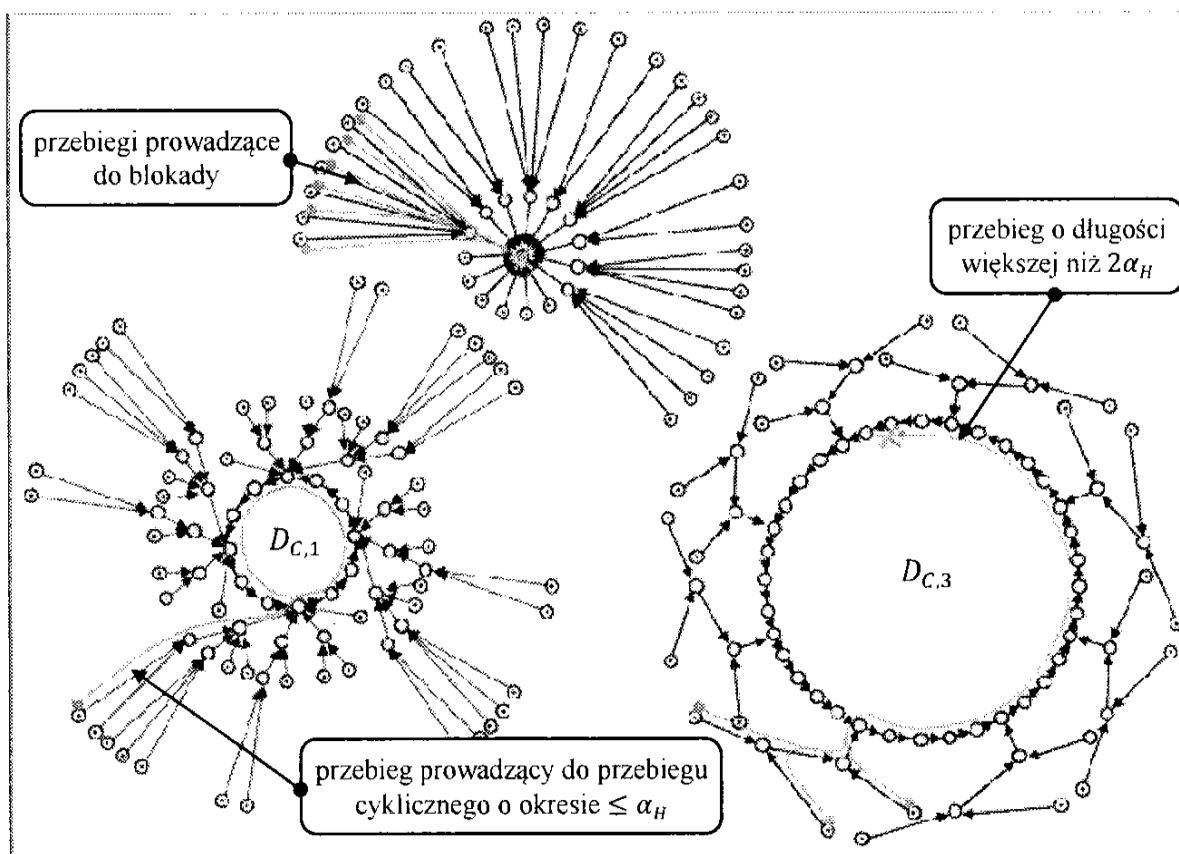
---

gdzie:

 $D_S, C_N, \delta^{lp}$  – definiowane jak w (73), $\alpha_H$  – dopuszczalna wartość okresu przebiegu cyklicznego, $PS_{SB}$  – problem spełniania ograniczeń (73), $SB$  – zbiór stanów początkowych osiągalnych w przestrzeni  $\mathbb{S}^{lp}$ , $V(DC^i)$  – zbiór stanów tworzących przebieg cykliczny  $D_C^i$  uzyskany ze stanu początkowego  $S^{i,1}$  (w przypadku braku przebiegu cyklicznego zbiór ten jest pusty), $V(DC)$  – zbiór stanów tworzących wszystkie przebiegi cykliczne z okresem nie większym niż  $\alpha_H$ , $|SB|$  – moc zbioru  $SB$ , $\text{SEARCHALL}(PS)$  – funkcja zwracająca zbiór wszystkich rozwiązań problemu  $PS$ , (w przypadku braku rozwiązania, funkcja zwraca zbiór pusty), $\text{FIRSTELEMENT}(SB)$  – funkcja zwracająca jeden element zbioru  $SB$  o najniższym indeksie, $\text{CHECKCYCLIC}(V^i)$  – jeżeli zbiór  $V^i$  zawiera stany składające się na przebieg cykliczny  $D_C^i$  to wynikiem funkcji jest zbiór tych stanów:  $V(D_C^i)$ , w przeciwnym wypadku zbiór pusty:  $V(D_C^i) = \emptyset$ .

Podobnie jak w poprzednim przypadku (Algorytm 1), wynikiem Algorytmu 2 jest zbiór stanów  $V(DC)$  składających się na wszystkie cyklicznie przebiegi ustalone, których okres nie przekracza wartości  $\alpha_H$ .

W odróżnieniu od Algorytmu 1, Algorytm 2 wymaga tylko jednokrotnego użycia funkcji SEARCHALL (w celu rozwiązania problemu  $PS_{SB}$ ). Wynikiem zwracanym przez tę funkcję jest zbiór stanów początkowych  $SB$ , dla których w dalszej kolejności wyznaczani są potomkowie  $S^{i,j} = \delta^{lp}(S^{i,j-1})$ . Przyjęto (zgodnie z Twierdzeniem 4), że liczba niepowtarzających się potomków każdego stanu początkowego nie może przekraczać wartości  $2\alpha_H$ . W sytuacji gdy warunek ten nie jest spełniony oznacza to, że przebieg przejściowy prowadzi do blokady lub do cyklicznego przebiegu ustalonego o okresie większym niż  $\alpha_H$ . Twierdzenie 4 traktowane jest więc jako warunek, spełnienie którego redukuje z przestrzeni  $\mathcal{P}$  wszystkie przebiegi nieprowadzące do żądanych przebiegów cyklicznych. Przebiegi otrzymane w wyniku wyznaczania potomków stanów początkowych zbioru  $SB$ , dla systemu z przykładu 5 przedstawia rys. 3.10. Przebiegi (oznaczone na rys. 3.10) są wyznaczone dopóty, dopóki nie osiągnięty zostanie albo stan blokady, albo cykliczny przebieg ustalony o okresie nie większym niż  $\alpha_H = 12$ . Oznacza to, że przebiegi zawierające więcej niż 24 stany (np.  $D_{C,3}$ ) są odrzucane jako takie, które nie prowadzą dożądanego przebiegu cyklicznego.



Rys. 3.10. Przykładowe przebiegi powstałe w przestrzeni stanów  $\mathcal{P}$  systemu z rys. 1.32 w wyniku wykorzystania Algorytmu 2

W rozważanym przypadku, w wyniku zastosowania Algorytmu 2 wyznaczony został przebieg cykliczny  $D_{C,1}$ , a odrzucony przebieg  $D_{C,3}$ . Otrzymane rozwiązanie wymagało wyznaczenia 117 stanów, a zatem było mniej kosztowne obliczeniowo niż w przypadku Algorytmu 1. Obyło się to jednak kosztem pominięcia cyklicznych przebiegów ustalonych  $D_{C,2}$ ,  $D_{C,4}$ ,  $D_{C,5}$  nie posiadających przebiegów przejściowych, od których Algorytm 2 rozpoczyna swoje poszukiwania. Pomijanie tego typu przebiegów stanowi istotną wadę tego algorytmu.

Heurystyczny charakter przedstawionych procedur wyznaczania cyklicznych przebiegów ustalonych przestrzeni  $\mathcal{P}$  implikuje potrzebę oceny ich efektywności (złożoności obliczeniowej), a zatem oszacowania rozmiaru problemów, dla których mogą być wykorzystywane.

### 3.1.3. Złożoność obliczeniowa

W obydwu prezentowanych algorytmach kluczową rolę odgrywa sposób wyznaczania zbioru  $SB$  tzn. sposób rozwiązywania problemów (73) i (74), (funkcja  $SearchAll(PS)$ ). Dla potrzeb dalszych rozważań przyjęto, że do rozwiązania problemów tych wykorzystywane są techniki programowania z ograniczeniami. Wyniki wstępnych eksperymentów (Bocewicz i Banaszak 2010, 2013b) wskazują na fakt, bardzo różnej, uzależnionej od wielu parametrów struktury systemu (m.in. od liczby zasobów, liczby potoków realizowanych w systemie potoków, liczby poziomów zachowań, oraz liczby realizowanych operacji) efektywności tego typu, komercyjnie dostępnych technik.

Celem oceny złożoności obliczeniowej wykorzystywanych technik wprowadzono parametr umożliwiający klasyfikację struktur, tak zwaną **gęstość struktury**  $g$ , definiowaną jako stosunek liczby potoków realizowanych w strukturze  $SC^{lp}$  do liczby wykorzystywanych w tym celu zasobów:

$$g = \frac{LP}{(lp + 1) \cdot lk}, g \in (0,1],$$

gdzie:

$lp$  – liczba poziomów zachowań procesów multimodalnych struktury  $SC^{lp}$ ,

$lk$  – liczba zasobów struktury  $SC^{lp}$ ,

$LP$  – liczba potoków realizowanych w strukturze  $SC^{lp}$ :

$$LP = \sum_{i=1}^{ln} ls(i) + \sum_{l=1}^{lp} \sum_{i=1}^{lw(l)} lsm(i, l),$$

$ln$  – liczba procesów lokalnych realizowanych w strukturze  $SC^{lp}$ ,

$ls(i)$  – liczba potoków procesu lokalnego  $P_i$ ,

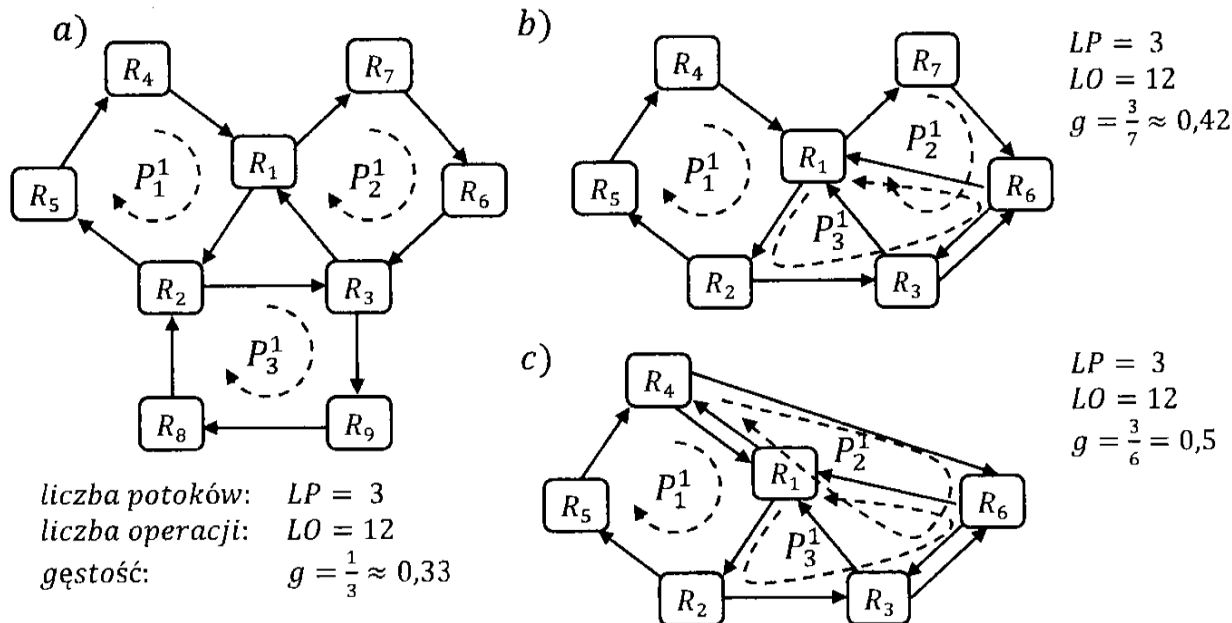
$lw(l)$  – liczba procesów multimodalnych  $l$ -tego poziomu zachowań,

$lsm(i, l)$  – liczba potoków procesu multimodalnego  $m^l P_i$ .

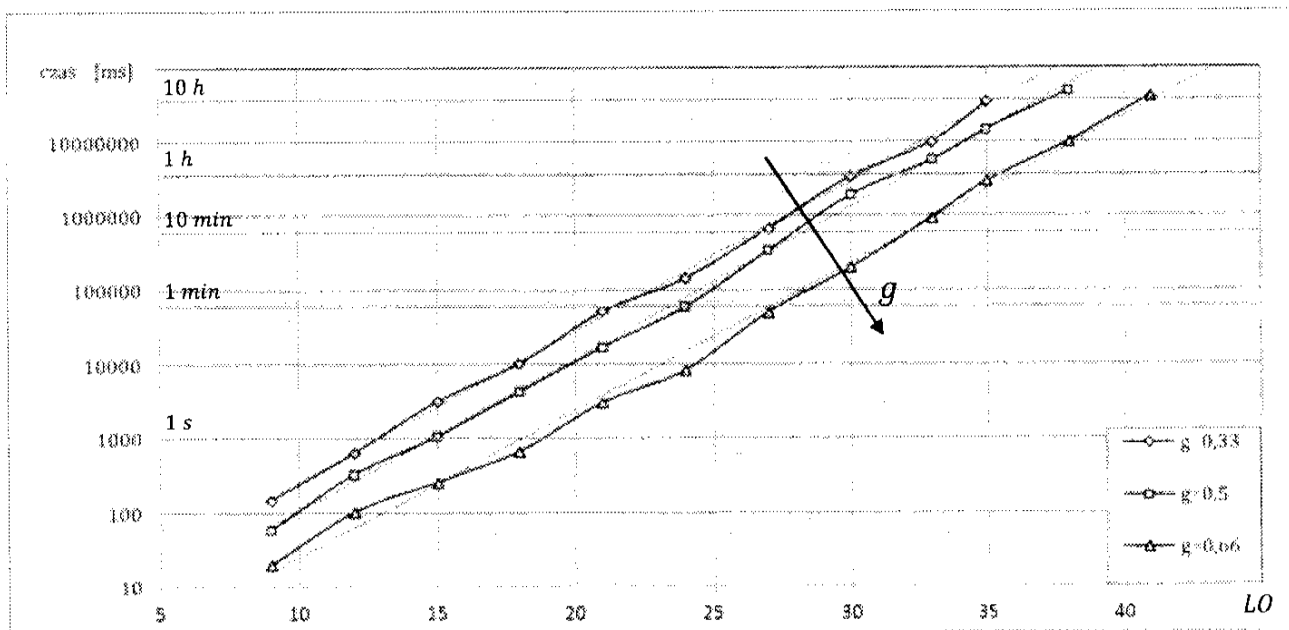
Celem ilustracji intuicji związanej z wprowadzonym pojęciem gęstości struktury na rys. 3.11 przedstawiono struktury charakteryzujące się stałą liczbą potoków ( $LP = 3$ ) oraz stałą liczbą realizowanych w ramach tych potoków operacji ( $LO = 12$ ), jednak różniące się wartością gęstości  $g$ . Jak widać, wraz ze wzrostem wartości gęstości te same procesy realizowane są na coraz mniejszym zbiorze zasobów. W strukturze z rys. 3.11a 3 potoki realizowane są na 9 zasobach, w strukturze z rys. 3.11b już na 7, a strukturze z rys. 3.11c na 6 zasobach.

W dalszych rozważaniach, przez struktury „gęste” rozumiane będą takie struktury, w których wiele procesów realizowanych jest na niewielu zasobach (wartość  $g$  dąży

do 1) i odwrotnie, przez struktury „rzadkie” rozumiane będą takie struktury, w który niewiele procesów jest realizowanych na wielu zasobach (wartość  $g$  dąży do 0).



Rys. 3.11. Przykłady SWMPC o różnych gęstościach:  $g \approx 0,33$  a),  $g \approx 0,42$  b),  $g = 0,5$  c)



Rys. 3.12. Funkcje czasów rozwiązania problemów typu (73) i (74) (czas odpowiedzi funkcji  $SEARCHALL(PS_{SB})$ ) dla struktur o gęstościach:  $g = 0,33$ ;  $g = 0,5$ ;  $g = 0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM)

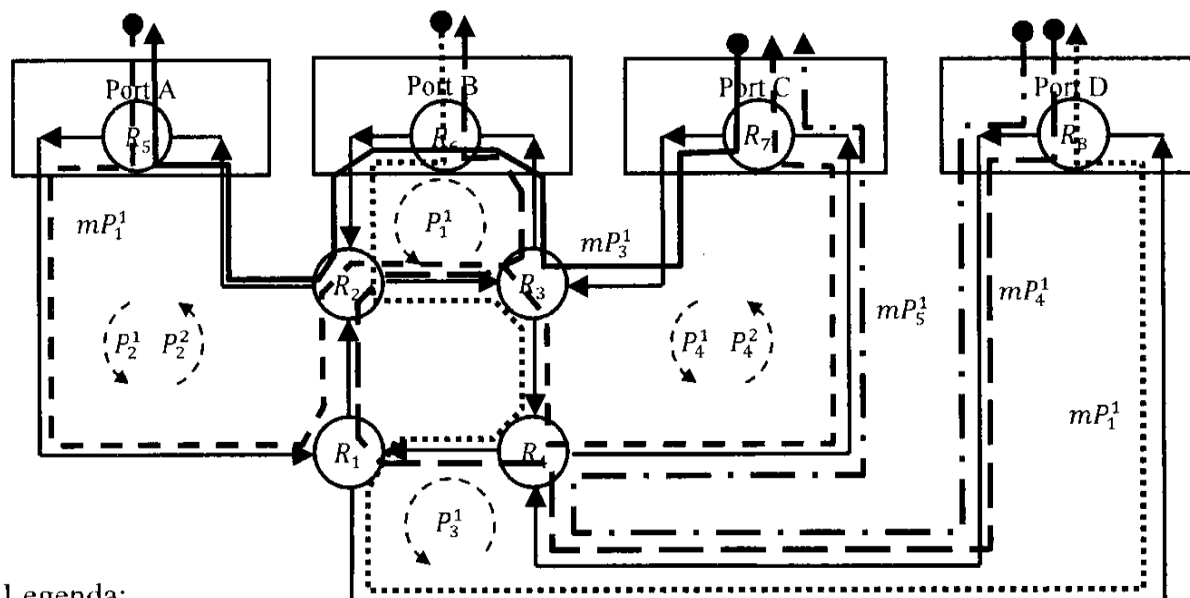
Na rys. 3.12 przedstawiono wykresy funkcji czasów (wykres w skali logarytmicznej) rozwiązania problemów typu (73) i (74) w zależności od sumarycznej liczby operacji (procesów lokalnych i multimodalnych) realizowanych w strukturze  $SC^{lp}$  ( $LO = 5 \dots 45$ ), a także od wartości jej gęstości ( $g = 0,33$ ;  $0,5$ ;  $0,66$ ). Otrzymane wykresy, przedstawiające funkcje wykładnicze, wyznaczono w eksperymentach przeprowadzonych w środowisku programowania z ograniczeniami Oz Mozart (Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM).



Czas rozwiązania nieprzekraczający 60 minut jest osiągalny w strukturach o gęstości  $g = 0,66$ , w których liczba  $LO$  nie przekracza 40 operacji. W strukturach gęstszych, dla tych samych wartości  $LO$  odpowiednie czasy były znacznie krótsze – wynika to z faktu, że w systemach o strukturach tego typu większość stanów jest stanami blokadowymi odrzucanymi już na wstępie w procesie propagacji ograniczeń.

Oznacza to, że proponowane algorytmy powinny być wykorzystywane w systemach o strukturach charakteryzujących się dużymi gęstościami. Praktycznym przykładem tego typu struktury jest struktura analizowanego już wcześniej routera VOD (patrz rys. 3.13), w którym uwzględniono możliwość wielopotkowej transmisji danych (6 potoków procesów lokalnych) między portami: A–C, C–A, B–D, D–B, D–C (5 potoków procesów multimodalnych). Gęstość tej struktury wynosi  $g = 0,6875$ .

Reasumując, wykorzystanie proponowanych algorytmów czasie krótszym od 60 minut ogranicza się do struktur gęstych  $g > 0,5$  systemów typu  $MW_{1,1}$ ,  $MW_{1,2}$ ,  $MW_{2,1}$ ,  $MW_{2,2}$ , (patrz tab. 2.1), w których sumaryczna liczba realizacji operacji nie przekracza 40.



Legenda:

$A, B, C, D$  – nazwa portu

$P_i^h$  –  $h$ -ty potok  $i$ -tego procesu lokalnego, reprezentujący transport pakietów między zasobami

$mP_i^h$  –  $h$ -ty potok  $i$ -tego procesu multimodalnego, reprezentujący strumień pakietów między portami

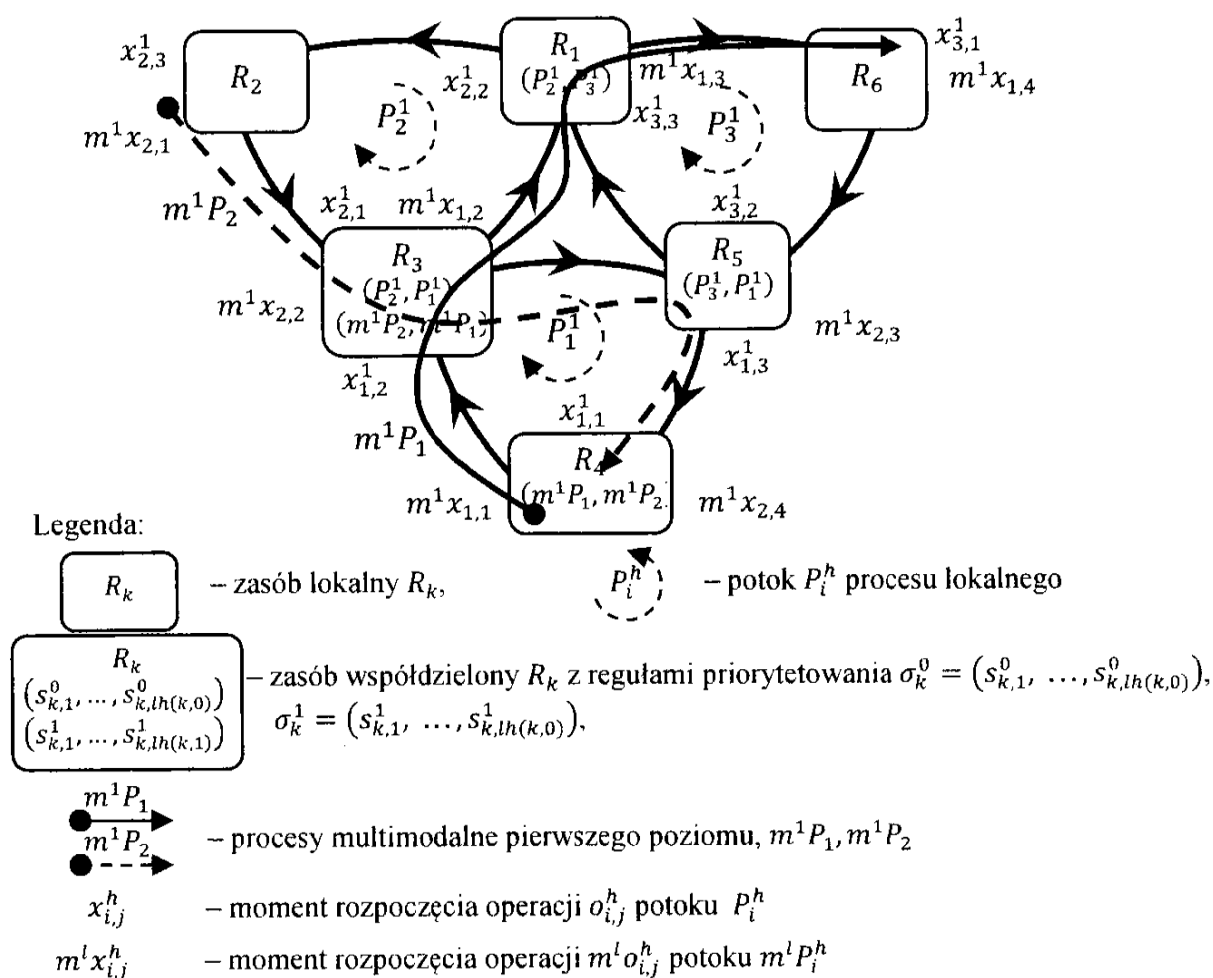
Rys. 3.13. SWMPC ( $g = 0,6875$ ) reprezentujący router VOD wielopotkowej transmisji danych między portami: A–C, C–A, B–D, D–B, D–C

### 3.2. Metoda generowania harmonogramów cyklicznych

Przestawione algorytmy rozwiązywania problemów analizy (72) wyznaczają (poprzez redukcję stanów początkowych) te fragmenty przestrzeni  $\mathcal{P}$  (wiry), które skutkują cyklicznymi przebiegami ustalonymi. Alternatywna metoda polega na wyznaczeniu cyklicznych przebiegów ustalonych w oparciu o momenty rozpoczęcia operacji w poszczególnych procesach. W celu jej zilustrowania na rys. 3.14 przedstawiono przykładowy system zawierający dwa poziomy zachowań (poziom procesów lokal-

nych  $SL$  i jeden poziom procesów multimodalnych  $SM^1$ ). Wśród procesów lokalnych wyróżnia się trzy procesy jednopotokowe:  $P_1, P_2$  i  $P_3$ , z kolei wśród procesów multimodalnych dwa procesy:  $m^1P_1$  i  $m^1P_2$  (również jednopotokowe). System jest typu  $MW_{3,1}$  (tab. 2.1) to znaczy:

- dopuszcza się nie jednostkowe czasy wykonania operacji (zarówno procesów lokalnych jak i multimodalnych),
- procesy multimodalne synchronizowane są w oparciu o protokół wzajemnego wykluczania,
- pojemności zasobów są jednostkowe:  $C(R_k) = 1, k = 1 \dots 6$ ,
- zapotrzebowanie procesów lokalnych i multimodalnych na wykorzystywane przez siebie zasoby jest jednostkowe:  $\omega_{i,j}^h = 1, m^l \omega_{i,j}^h = 1$ .



**Rys. 3.14.** Przykład SWMPC opisanego regułami priorytetowania:  $\sigma_1^0 = (P_2^1, P_3^1)$ ,  $\sigma_2^0 = (P_2^1)$ ,  $\sigma_3^0 = (P_2^1, P_1^1)$ ,  $\sigma_4^0 = (P_1^1)$ ,  $\sigma_5^0 = (P_3^1, P_1^1)$ ,  $\sigma_6^0 = (P_3^1)$  oraz  $\sigma_1^1 = (m^1P_1, m^1P_2)$ ,  $\sigma_3^1 = (m^1P_2, m^1P_1)$

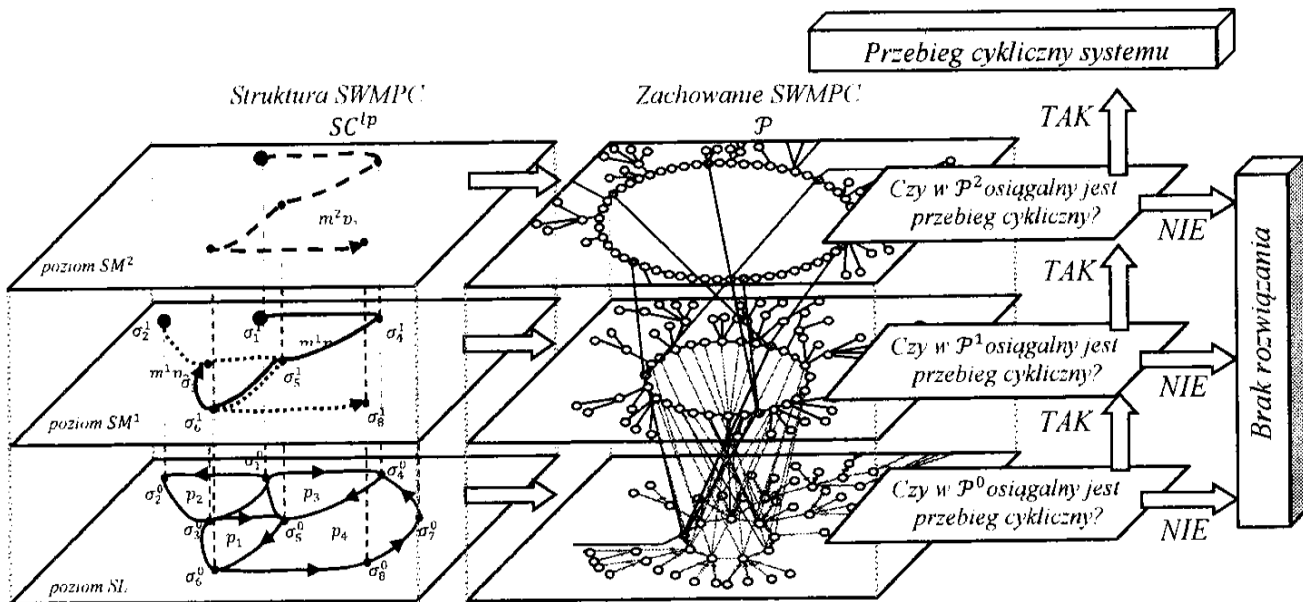
Dla tak zadanego systemu poszukuje się odpowiedzi na pytanie:

Czy w przestrzeni  $\mathcal{P}$  systemu z rys. 3.14 osiągalne są cykliczne przebiegi ustalone?

Do odpowiedzi na to pytanie (rozwiązanie problemu (72)) wykorzystane zostanie podejście bazujące na konstrukcji digrafów pierwszeństwa operacji. W podejściu tym wykorzystuje się sformułowaną w poprzednim rozdziale Własność 7 orzekającą, że

jeżeli w systemie osiągalny jest przebieg cykliczny to skojarzony z nim wir posiada swoją cykliczną reprezentację (rzut) na każdym poziomie zachowań. Własność ta pozwala wydzielić w procedurze poszukiwania etapy poszukiwania przebiegów cyklicznych, oddzielnie dla każdego z poziomów zachowań.

Procedurę rozpoczyna analiza zachowania najniższego poziomu ( $SL$  – procesów lokalnych). Jeżeli na tym poziomie cykliczne przebiegi ustalone (i tym samym reprezentujące je wiry) są osiągalne, to przechodzi się do następnego etapu związanego z poszukiwaniem przebiegów na kolejnym wyższym poziomie. Postępowanie jest powtarzane do momentu gdy na pewnym poziomie zachowań przebiegi cykliczne nie występują (system osiąga stan blokady) lub gdy na najwyższym poziomie zachowań również istnieje cykliczny przebieg ustalony (system jest cykliczny). Ideę przedstawionego podejścia, odpowiadającego analizie zachowań typu bottom-up, ilustruje rys. 3.15 przedstawiający strukturę systemu zawierającą trzy poziomy zachowań  $SL$ ,  $SM^1$ ,  $SM^2$ .



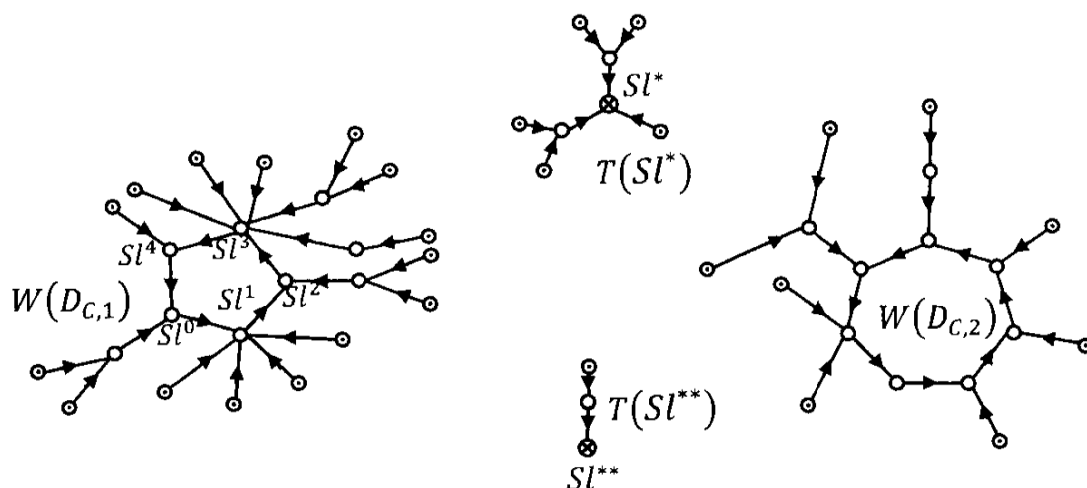
Rys. 3.15. Idea wyznaczania cyklicznych przebiegów ustalonych

Badanie osiągalności cyklicznych zachowań systemu determinowanych przez występowanie cyklicznych przebiegów ustalonych w kolejnych przestrzeniach  $\mathcal{P}^0$ ,  $\mathcal{P}^1$ ,  $\mathcal{P}^2$  umożliwia analiza digrafu pierwszeństwa operacji, digrafu odwzorowującego sposób realizacji operacji dowolnego przebiegu cyklicznego.

### 3.2.1. Digraf pierwszeństwa operacji

Rys. 3.16 ilustruje przestrzeń stanów  $\mathcal{P}^0$  (rzut przestrzeni  $\mathcal{P}$ ) dla pierwszego poziomu zachowań (poziom  $SL$ ) systemu z rys. 3.14 (dla wszystkich operacji przyjęte zostały czasy jednostkowe). W przestrzeni występują dwa wiry:  $W(D_{C,1})$ ,  $W(D_{C,2})$  (59) związane z przebiegami cyklicznymi  $D_{C,1}$  i  $D_{C,1}$  oraz dwa drzewa:  $T(Sl^*)$ ,  $T(Sl^{**})$  (60) prowadzące do dwóch stanów blokadowych  $Sl^*$ ,  $Sl^{**}$ . Zbiór stanów  $\mathcal{S}^0$  zawiera 50 stanów dopuszczalnych z czego, na rys. 3.16, wyróżnione zostały wyłącznie stany  $Sl^0 - Sl^4$  składające się na przebieg cykliczny  $D_{C,1}$ .

Każdy cykliczny przebieg ustalony przestrzeni  $\mathcal{P}^0$  jest reprezentowany przez **digraf pierwszeństwa operacji**  $G = (\mathbb{V}_G, \mathbb{E}_G)$ . Przykładowo digraf  $G_1 = (\mathbb{V}_{G,1}, \mathbb{E}_{G,1})$  z rys. 3.17b odpowiada realizacji operacji przebiegu cyklicznego  $D_{C,1}$ . Stany cyklicznego przebiegu ustalonego  $D_{C,1}$  (rys. 3.17a) związane są z przedziałami czasu, w których zachodzą. Przedziały te reprezentowane są przez kolejne poziomy prostopadłościanów oznaczonych symbolami realizowanych procesów:  $P_1, P_2, P_3$ . Każdy prostopadłościan  $P_i$  o podstawie wielokąta wypukłego z  $lr(i)$  wierzchołkami ( $lr(i)$  – liczba operacji potoku  $P_i^h$  procesu  $P_i$ ), reprezentuje oś czasu, względem której realizowane są operacje potoku  $P_i^h$  procesu  $P_i$ . Wierzchołki podstawy prostopadłościanu reprezentują zasoby wykorzystywane przy realizacji operacji potoku  $P_i^h$ .



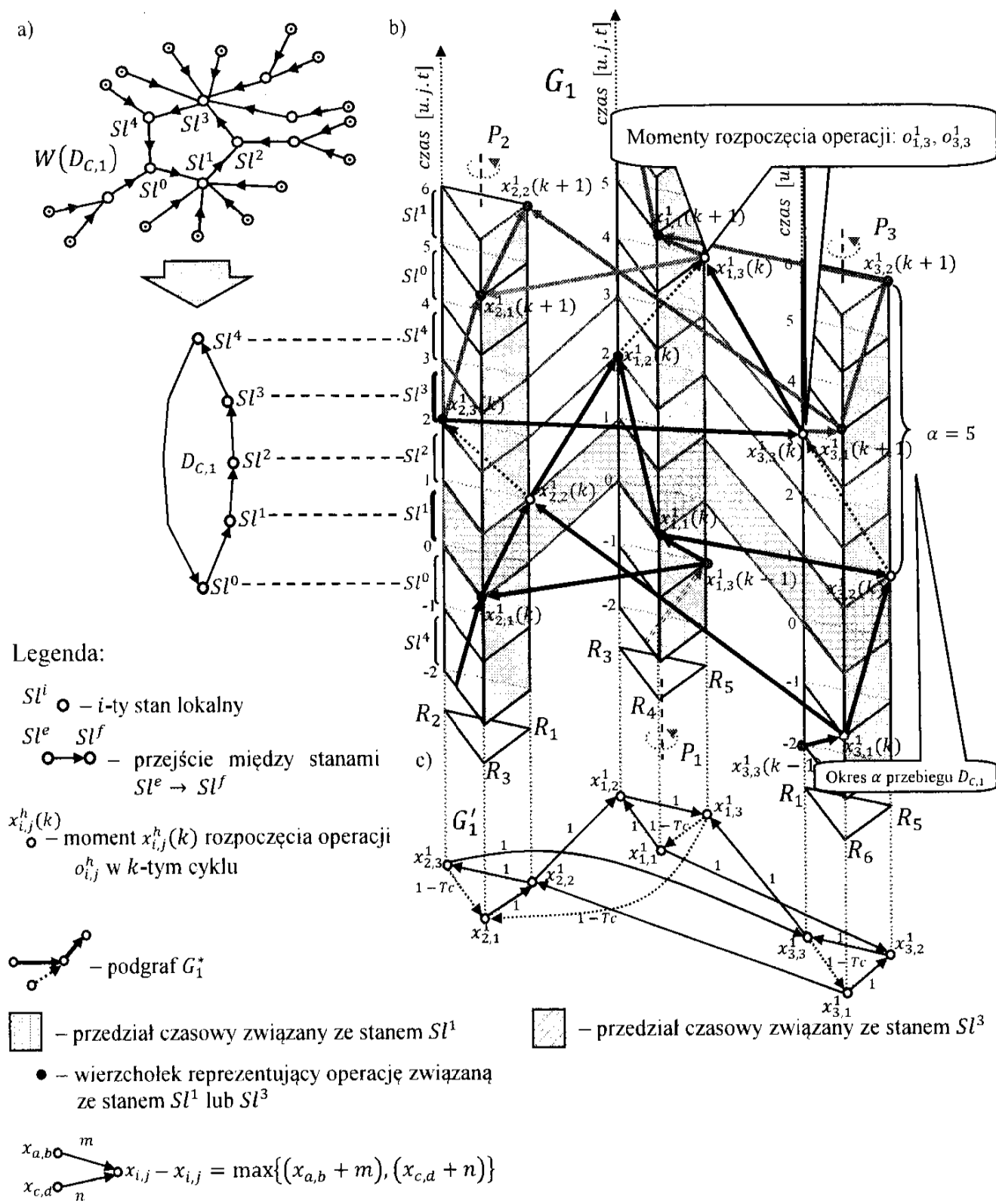
Legenda:

● – stan blokadowy      ● – stan początkowy      ○ – stan dopuszczalny

**Rys. 3.16.** Przestrzeń przebiegów  $\mathcal{P}^0$  dla systemu z rys. 3.14

Na zbiorze prostopadłościanów (gdzie jeden prostopadłościan odpowiada jednemu procesowi) rozpięty jest digraf pierwszeństwa operacji  $G_1$ . Wierzchołki  $v_{i,j}^h(k)$  tego digrafu, oznaczane symbolami: ●●● i związane z operacjami  $o_{i,j}^h$  potoków  $P_i^h$ , rozmieszczone są, na prostopadłościanach, na wysokości odpowiadającej momentom ich rozpoczęcia  $x_{i,j}^h(k)$ . Analogicznie do przyjętej na wstępie notacji (1), (10), (11), przyjmuje się, że każdej operacji  $o_{i,j}^h$  odpowiada moment jej rozpoczęcia  $x_{i,j}^h(k)$ , gdzie  $k$  określa numer realizowanego cyklu systemu. Wierzchołki występujących po sobie operacji połączone są łukami (zawsze skierowanymi zgodnie ze zwrotem osi czasu), których długości determinują czas zajęcia wykorzystywanych przez potoki zasobów (reprezentowanych przez wierzchołki podstawy). Łuki te odwzorowują również porządek realizacji operacji określony przez marszruty procesów  $P_i$ .

Oprócz łuków związanych z porządkiem operacji realizowanych w ramach jednego potoku (łuki składające się na ściany prostopadłościanów) występują również łuki łączące wierzchołki różnych prostopadłościanów. Ten rodzaj łuków opisuje pierwszeństwo operacji będące konsekwencją przyjętej postaci reguł priorytetowania  $\theta^0$ .



**Rys. 3.17.** Cykliczny przebieg ustalony  $D_{C,1}$  systemu z rys. 3.14 a), digraf pierwszeństwa operacji  $G_1$  reprezentujący  $D_{C,1}$  b),  $G'_1$  – rzut digrafu  $G_1$  na płaszczyznę c)

Łuk tego typu skierowany jest zawsze do wierzchołka związanego z operacją późniejszą (o większej wartości  $x_{i,j}^h(k)$ ). Opóźnienie spowodowane występowaniem tego łuku  $\Delta t$ , zdefiniowane w założeniach SWMPC (podrozdział 2.1), określa zwłokę związaną ze zmianą potoków na danym zasobie. W rozważanym przykładzie przyjęto, że  $\Delta t = 1$ .

Przykładowo, wartość zmiennej  $x_{1,2}^1(k) = 2$  (wierzchołek  $v_{1,2}^1(k)$  umieszczony jest na poziomie 2 – rys. 3.17b), jest większa od wartości  $x_{1,1}^1(k) = 0$  ponieważ operacja  $o_{1,2}^1$  występuje w marszrucie  $p_1$  po operacji  $x_{1,1}^1$ . Ponadto  $x_{2,2}^1(k) = 1 < x_{1,2}^1(k) = 2$  ponieważ zgodnie z regułą priorytetowania  $\sigma_3^0$  (kolejność operacji na  $R_3$ :  $\dots, o_{2,1}^1, o_{1,2}^1, \dots$ ) operacja  $o_{1,2}^1$  wykonywana przez potok  $P_1^1$  może rozpocząć swoją realizację tuż po tym jak zajmowany, przez potok  $P_2^1$ , zasób  $R_3$  (operacja  $o_{2,1}^1$ ) zostanie zwolniony, tj. po rozpoczęciu realizacji operacji  $o_{2,2}^1$ .

Tak rozumiany acykliczny digraf  $G_1$  składa się z nieskończonej liczby wierzchołków  $v_{i,j}^h(k)$  rozmieszczonych na prostopadłościanach zgodnie z porządkiem determinowanym przez zbiór marszrut procesów oraz reguły priorytetowania. Okresowy charakter digrafu związany jest z występowaniem wielokrotnie powtarzanego, skończonego podgrafu  $G_1^*$ . Podgraf  $G_1^*$ , z którego utworzony jest  $G_1$  został oznaczony na rys. 3.17b liniami pogrubionymi. W rozważnym systemie przebieg cykliczny  $D_{C,1}$  (rys. 3.17a) odpowiada następującej sekwencji przejść między stanami  $Sl^i$ :

$$\dots \rightarrow Sl^0 \rightarrow Sl^1 \rightarrow Sl^2 \rightarrow Sl^3 \rightarrow Sl^4 \rightarrow Sl^0 \rightarrow \dots,$$

w której z każdym stanem  $Sl^i$  związana jest realizacja pewnych operacji:

$$\rightarrow (o_{1,3}^1, o_{3,1}^1) \rightarrow (o_{1,1}^1, o_{2,1}^1) \rightarrow (o_{2,2}^1, o_{3,2}^1) \rightarrow (o_{2,3}^1, o_{1,2}^1) \rightarrow (o_{3,3}^1) \rightarrow (o_{1,3}^1, o_{3,1}^1) \rightarrow$$

Każdemu ze stanów  $Sl^i$  przypisana jest sekwencja  $(o_{a,b}^h, \dots, o_{c,d}^i)$  operacji rozpoczynających się w przedziale czasu związanym z tym stanem. Przykładowo pomarańczowym kolorem (rys. 3.17b) oznaczony został przedział czasu  $t = [0,1)$  związany ze stanem  $Sl^1$ . W przedziale tym, swoją realizację rozpoczynają operacje  $o_{1,1}^1, o_{2,1}^1$  (momenty rozpoczęcia tych operacji, oznaczone symbolem  $\bullet$ , wynoszą:  $x_{1,1}^1 = x_{2,1}^1 = 0$ ). Analogicznie, sekwencja  $(o_{2,3}^1, o_{1,2}^1)$  przypisana jest do stanu  $Sl^3$ , przedział czasu  $t = [2,3)$  odpowiadający temu stanowi oznaczony został zielonym kolorem.

Każdy ustalony przebieg cykliczny  $D_C$  może być więc opisany przez zbiór sekwencji operacji  $o_{a,b}^h, \dots, o_{c,d}^i$  oraz powiązany z nim zbiór momentów rozpoczęcia tych operacji. Operacje i momenty ich rozpoczęcia odpowiadające przebiegowi  $D_{C,1}$  zostały przedstawione w tab. 3.1. Każda kolumna tabeli zawiera stan, przypisaną mu sekwencję operacji oraz momenty rozpoczęcia tych operacji dla kolejnych cykli.

**Tabela 3.1.** Momenty rozpoczęcia operacji realizowanych w kolejnych stanach cyklicznego przebiegu ustalonego  $D_{C,1}$

Stany:	$Sl^0$	$Sl^1$	$Sl^2$	$Sl^3$	$Sl^4$
Operacje:	$(o_{1,3}^1, o_{3,1}^1)$	$(o_{1,1}^1, o_{2,1}^1)$	$(o_{2,2}^1, o_{3,2}^1)$	$(o_{2,3}^1, o_{1,2}^1)$	$(o_{3,3}^1)$
Momenty $x_{i,j}^h(k)$ dla					
$k$ :	-1	0	1	2	3
$k + 1$ :	4	5	6	7	8
$k + 2$ :	9	10	11	12	13
...	...				

Innymi słowy, tab. 3.1 zawiera wartości  $x_{i,j}^h(k)$  przypisane do wierzchołków  $v_{i,j}^h(k)$  digrafu  $G_1$ , wierzchołków odpowiadających realizacji operacji występujących w kolejnych ( $k$ -tych) cyklach systemu. Przykładowo wartości  $x_{2,2}^1(k)$  i  $x_{3,2}^1(k)$  określają momenty rozpoczęcia  $o_{2,2}^1, o_{3,2}^1$  kolejno dla cykli:  $k, k + 1, k + 2, \dots$ , itd., wartości tych momentów wynoszą: 1, 6, 11, ... . Operacje  $o_{2,2}^1, o_{3,2}^1$  (jak i stan  $Sl^2$ ) są zatem powtarzane z okresem równym 5 jednostek czasu ( $\alpha = 5$ ). Oznacza to, że w rozważanym systemie (rys. 3.14) procesy są realizowane cyklicznie, system osiąga kolejno stany określone w przebiegu  $D_{c,1} = (Sl^0, Sl^1, Sl^2, Sl^3, Sl^4)$ .

**Ostatecznie oznacza to, że digraf pierwszeństwa operacji może być wykorzystany do odtworzenia realizacji operacji każdego cyklicznego przebiegu ustalonego osiągalnego w przestrzeni  $\mathcal{P}^0$ .**

Porządek wykonywania operacji odwzorowywany przez digraf  $G_1$  scharakteryzuje ograniczenie (76):

$$x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}, \quad (76)$$

gdzie:

$x_{i,j}^h(k)$ ,  $x_{i,(j-1)}^h(k')$ ,  $x_{a,b}^c(k'')$  – wartości momentów rozpoczęcia operacji  $o_{i,j}^h, o_{i,(j-1)}^h, o_{a,b}^c$  związanych z wierzchołkami  $v_{i,j}^h(k), v_{i,(j-1)}^h(k'), v_{a,b}^c(k'')$  digrafu pierwszeństwa operacji, wierzchołki  $v_{i,(j-1)}^h(k'), v_{a,b}^c(k'')$  są poprzednikami wierzchołka  $v_{i,j}^h(k)$ , ponadto  $v_{i,(j-1)}^h(k')$  jest wierzchołkiem tego samego prostopadłościanu co  $v_{i,j}^h(k)$  (tzn. reprezentującym operację tego samego potoku) podczas gdy  $v_{a,b}^c(k'')$  jest wierzchołkiem innego prostopadłościanu (tzn. wierzchołkiem reprezentującym operację poprzedzającą zgodnie z przyjętą regułą priorytetowania). Wielkości  $k, k'$  oraz  $k''$  określają cykle w ramach, których wykonywane są operacje  $o_{i,j}^h, o_{i,(j-1)}^h, o_{a,b}^c$ . Wierzchołki  $v_{i,(j-1)}^h(k'), v_{a,b}^c(k'')$  mogą odnosić się do operacji realizowanych w ramach bieżącego cyklu:  $k', k'' = k$ , poprzedniego cyklu:  $k', k'' = k - 1$  a w przypadku wierzchołka  $v_{a,b}^c(k'')$  do następnego (względem operacji  $o_{i,j}^h$ ) cyklu:  $k'' = k + 1$ .

$t_{i,(j-1)}$  – czas wykonywania operacji  $o_{i,(j-1)}$ ,

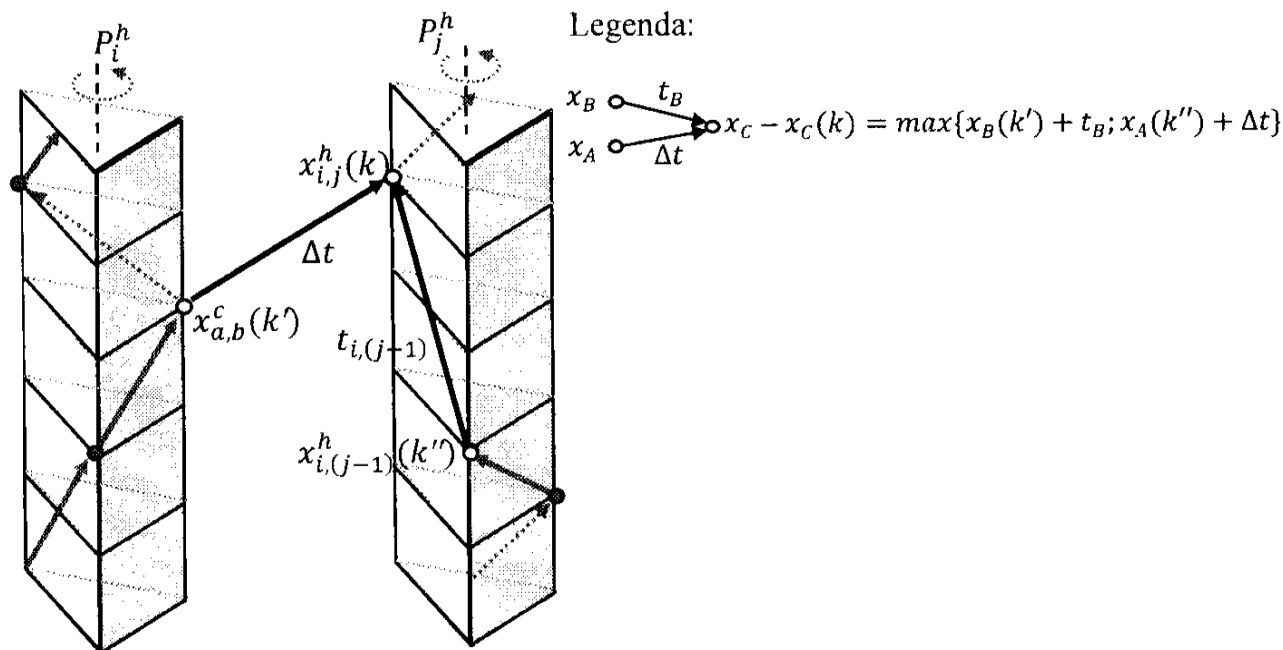
$\Delta t$  – opóźnienie związane ze zmianą procesów na zasobie (patrz założenia SWMPC – podrozdział 2.1.1).

W przypadku gdy do wierzchołka  $v_{i,j}^h(k)$  dochodzi tylko jeden łuk (z wierzchołka  $v_{i,(j-1)}^h(k')$ ) ograniczenie (76) upraszcza się do:  $x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{i,(j-1)}^h(k') + t_{i,(j-1)}\} = x_{i,(j-1)}^h(k') + t_{i,(j-1)}$ . Sytuacja taka ma miejsce w przypadku operacji realizowanych na zasobach lokalnych.

Graficzną ilustrację ograniczenia (76) przedstawia rys. 3.18. Przykładem takiego ograniczenia jest związek występujący pomiędzy momentami rozpoczęcia operacji  $o_{2,2}^1, o_{2,1}^1, o_{3,1}^1$  systemu z rys. 3.14 (wierzchołki  $v_{2,2}^1(k), v_{2,1}^1(k), v_{3,1}^1(k)$  digrafu  $G_1$ ):

$$x_{2,2}^1(k) = \max\{x_{2,1}^1(k) + t_{2,1}; x_{3,1}^1(k) + \Delta t\}, \quad (77)$$

gdzie momenty  $x_{2,1}^1(k)$ ,  $x_{3,1}^1(k)$  są przypisane do wierzchołków będących poprzednikami wierzchołka  $v_{2,2}^1(k)$ . Ograniczenie to wynika bezpośrednio z założeń przedstawionych w podrozdziale 2.1.1.



**Rys. 3.18.** Ograniczenie:  $x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}$

Zastosowanie operatora  $\max$  jest podyktowane brakiem możliwości przestoju (oczekiwania) procesów z innych przyczyn niż zajętość żadanego zasobu lub brakiem prawa dostępu (Bocewicz et al. 2013a). W literaturze (Smutnicki 2012; Smutnicki i Smutnicki 2007; Kampmeyer 2006; Brucker i Kampmeyer 2005) szeroko omawiane są przypadki, w których do opisu związku między wykonaniem występujących po sobie operacji wykorzystuje się ograniczenia w postaci nierówności (zastępującej operator  $\max$ ). W takich przypadkach dopuszcza się jednak możliwość dowolnie długiego czasu oczekiwania procesu na zajmowanym zasobie.

**Tabela 3.2.** Ograniczenia typu (76) dla digrafu  $G_1$  z rys. 3.17

nr	$x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}$ $t_{i,(j-1)} = 1, \Delta t = 1$
1	$x_{1,1}^1(k) = \max\{x_{1,3}^1(k-1) + 1; x_{1,3}^1(k-1) + 1\}$
2	$x_{1,2}^1(k) = \max\{x_{1,1}^1(k) + 1; x_{2,2}^1(k) + 1\}$
3	$x_{1,3}^1(k) = \max\{x_{1,2}^1(k) + 1; x_{3,3}^1(k) + 1\}$
4	$x_{2,1}^1(k) = \max\{x_{2,3}^1(k-1) + 1; x_{1,3}^1(k-1) + 1\}$
5	$x_{2,2}^1(k) = \max\{x_{2,1}^1(k) + 1; x_{3,1}^1(k) + 1\}$
6	$x_{2,3}^1(k) = \max\{x_{2,2}^1(k) + 1; x_{2,2}^1(k) + 1\}$
7	$x_{3,1}^1(k) = \max\{x_{3,3}^1(k-1) + 1; x_{3,3}^1(k-1) + 1\}$
8	$x_{3,2}^1(k) = \max\{x_{3,1}^1(k) + 1; x_{1,1}^1(k) + 1\}$
9	$x_{3,3}^1(k) = \max\{x_{2,3}^1(k-1) + 1; x_{3,2}^1(k) + 1\}$

Ograniczenie (77) oznacza, że operacja  $o_{2,2}^1$  (realizowana przez  $P_2^1$  na zasobie  $R_1$ ) może się rozpocząć tylko wtedy, gdy operacja  $o_{2,1}^1$  poprzedzająca (realizowana przez



$P_2^1$  na  $R_3$ ) zostanie zakończona ( $x_{2,1}^1(k) + t_{2,1}$ ) oraz zasób  $R_1$ , zajmowany przez potok  $P_3^1$ , zostanie zwolniony, tzn. jeżeli potok  $P_3^1$  zajmujący zasób  $R_1$ , rozpocznie swoją kolejną operację  $o_{3,1}^1$  (w momencie  $x_A(k) + \Delta t$ ). **Innymi słowy, ograniczenie to określa wzajemny porządek wykonywania operacji  $o_{2,2}^1, o_{2,1}^1, o_{3,1}^1$ .** Analogicznie do (77) można konstruować ograniczenia opisujące relacje występujące między momentami  $x_{i,j}^h(k)$  pozostałych wierzchołków digrafu  $G_1$ . Ograniczenia tego typu zebrane zostały w tab. 3.2.

Prezentowane rozważania prowadzą do następujących wniosków: każdy cykliczny przebieg ustalony  $D_C$  występujący w przestrzeni  $\mathcal{P}^0$  może być reprezentowany przez digraf pierwszeństwa operacji  $G$ , z kolei każdy digraf  $G$  determinuje zbiór ograniczeń typu (76) określających wzajemny porządek wykonywania operacji. Momenty  $x_{i,j}^h(k)$  spełniające tego typu ograniczenia (np. spełniające ograniczenia z tab. 3.2) składają się natomiast na **harmonogram cykliczny  $X_{DC}$** :

$$\begin{aligned} X_{DC} = \{ & x_{i,j}^h(k) \mid x_{i,j}^h(k+1) - x_{i,j}^h(k) = \alpha; \\ & x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}; \\ & (v_{i,(j-1)}^h(k'), v_{i,j}^h(k)) \in \mathbb{E}_G; (x_{a,b}^c(k''), v_{i,j}^h(k)) \in \mathbb{E}_G; \\ & i = 1 \dots ln; j = 1 \dots lr(i); h = 1 \dots ls(i); k, k', k'' \in \mathbb{C} \}, \quad (78) \end{aligned}$$

gdzie:

$x_{i,j}^h(k)$  – moment rozpoczęcia operacji  $o_{i,j}^h$ ,  $h$ -tego potoku procesu lokalnego  $P_i$  w  $k$ -tym cyklu,

$v_{i,j}^h(k)$  – wierzchołek digrafu pierwszeństwa operacji  $G$ :  $v_{i,j}^h(k) \in \mathbb{V}_G$ , gdzie:  $\mathbb{V}_G$  – zbiór wierzchołków digrafu  $G$ ,

$\mathbb{E}_G$  – zbiór łuków digrafu  $G$ :  $\mathbb{E}_G \subseteq \mathbb{V}_G \times \mathbb{V}_G$ ,

$ln$  – liczba procesów lokalnych,  $lr(i)$  – liczba operacji procesu  $P_i$ ,  $ls(i)$  – liczba potoków procesu  $P_i$ .

Harmonogram  $X_{DC}$  stanowi zatem zbiór momentów rozpoczęcia operacji, które powtarzają się ze stałym okresem  $\alpha$  i dla których spełniona jest relacja porządku (76).

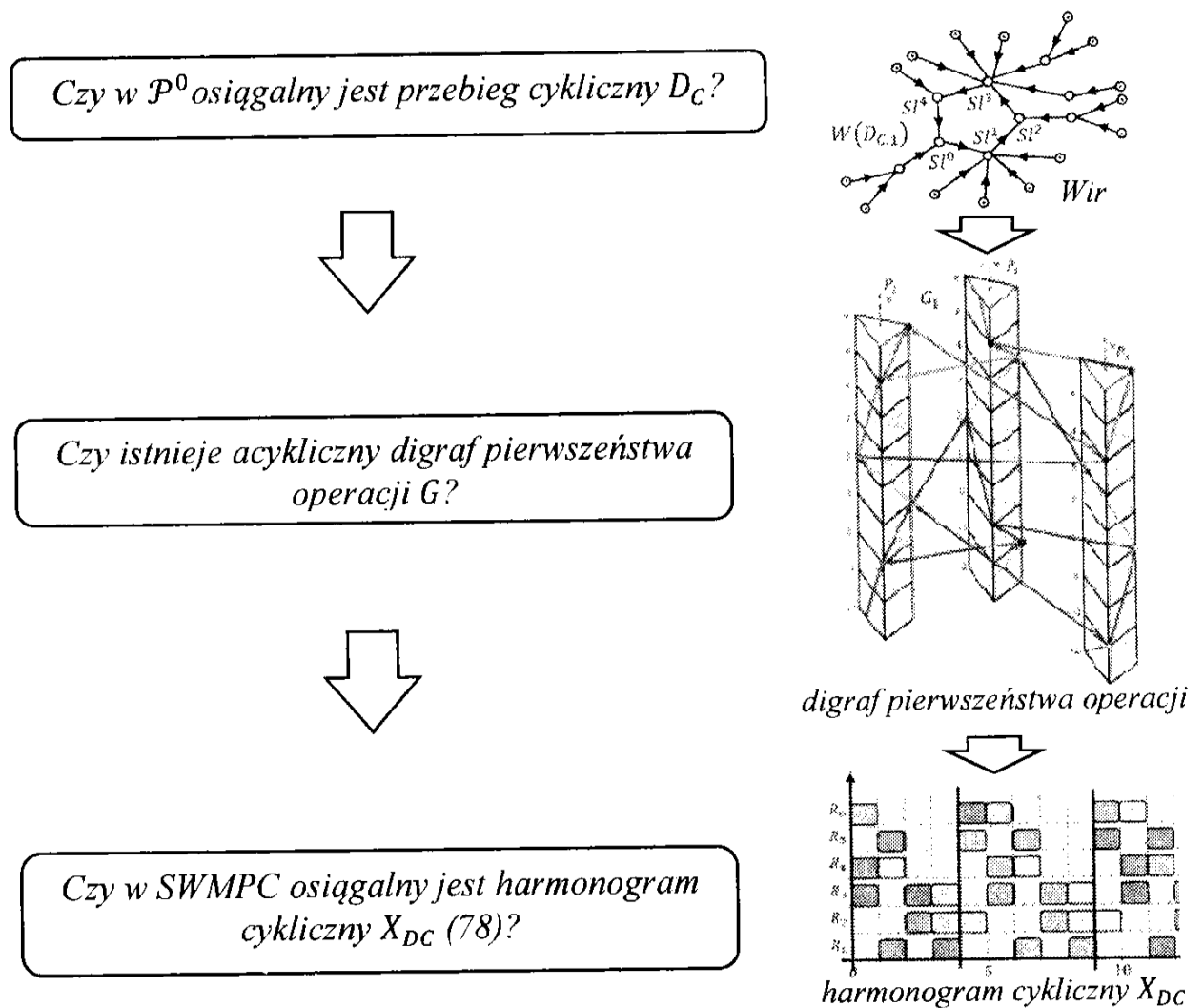
Oznacza to, że każdemu ustalonemu przebiegowi cyklicznemu  $D_C$  przestrzeni  $\mathcal{P}^0$  odpowiada dokładnie jeden harmonogram  $X_{DC}$  i odwrotnie, każdemu harmonogramowi  $X_{DC}$  odpowiada dokładnie jeden cykliczny przebieg ustalony  $D_C$ . Związek między harmonogramem  $X_{DC}$  a ustalonym przebiegiem cyklicznym został zilustrowany na rys. 3.19.

Digraf pierwszeństwa operacji  $G$  należy traktować jako graficzną reprezentację harmonogramu  $X_{DC}$ . Podobnie jak harmonogram  $X_{DC}$ , digraf  $G$  uwzględnia również czasy wykonania operacji, wzajemne relacje kolejnościowe, a także kolejność następstwa operacji w czasie. W przypadku digrafów pierwszeństwa operacji zachodzi twierdzenie:

#### Twierdzenie 5

*Jeżeli na  $l$ -tym poziomie zachowań (SL lub  $SM^l$ ) systemu o strukturze  $SC^{lp}$  istnieje acykliczny digraf pierwszeństwa operacji  $G$ , to zachowanie tego poziomu charakteryzuje harmonogram cykliczny  $X_{DC}$ .*

Dowód: Zgodnie z (76) i (78) dla wierzchołków dowolnego łuku  $(v_{i,j}^h(k), v_{a,b}^c(k)) \in \mathbb{E}_G$  digrafu  $G$  spełniona jest relacja:  $x_{i,j}^h(k) < x_{a,b}^c(k)$  (wierzchołek  $v_{a,b}^c(k)$  umieszczony jest wyżej niż  $v_{i,j}^h(k)$ ). Niech digraf  $G$  jest cykliczny, oznacza to istnienie zamkniętej ścieżki:  $(v_{i,j}^h(k), v_{a,b}^c(k), \dots, v_{i,j}^h(k))$ . Relacje związane z tą ścieżką przyjmują postać:  $x_{i,j}^h(k) < x_{a,b}^c(k) < \dots < x_{i,j}^h(k)$ , co prowadzi do sprzeczności:  $x_{i,j}^h(k) < x_{i,j}^h(k)$ . Sprzeczność ta oznacza, że nie może istnieć harmonogram cykliczny  $X_{DC}$  (ograniczenia (78) nie są spełnione) i tym samym nie można rozmieścić wierzchołków digrafu zgodnie z zadaniem porządkiem (wierzchołek  $v_{i,j}^h(k)$  powinien być umieszczony powyżej samego siebie). W przypadku gdy digraf  $G$  jest acykliczny zamknięte ścieżki nie występują, a tym samym relacje typu:  $x_{i,j}^h(k) < x_{i,j}^h(k)$  nie zachodzą. Tak więc wszystkie wierzchołki rozmieszczane są zgodnie z (76) co oznacza, istnienie harmonogramu cyklicznego  $X_{DC}$  i tym samym przebiegu  $D_C$ .  $\square$

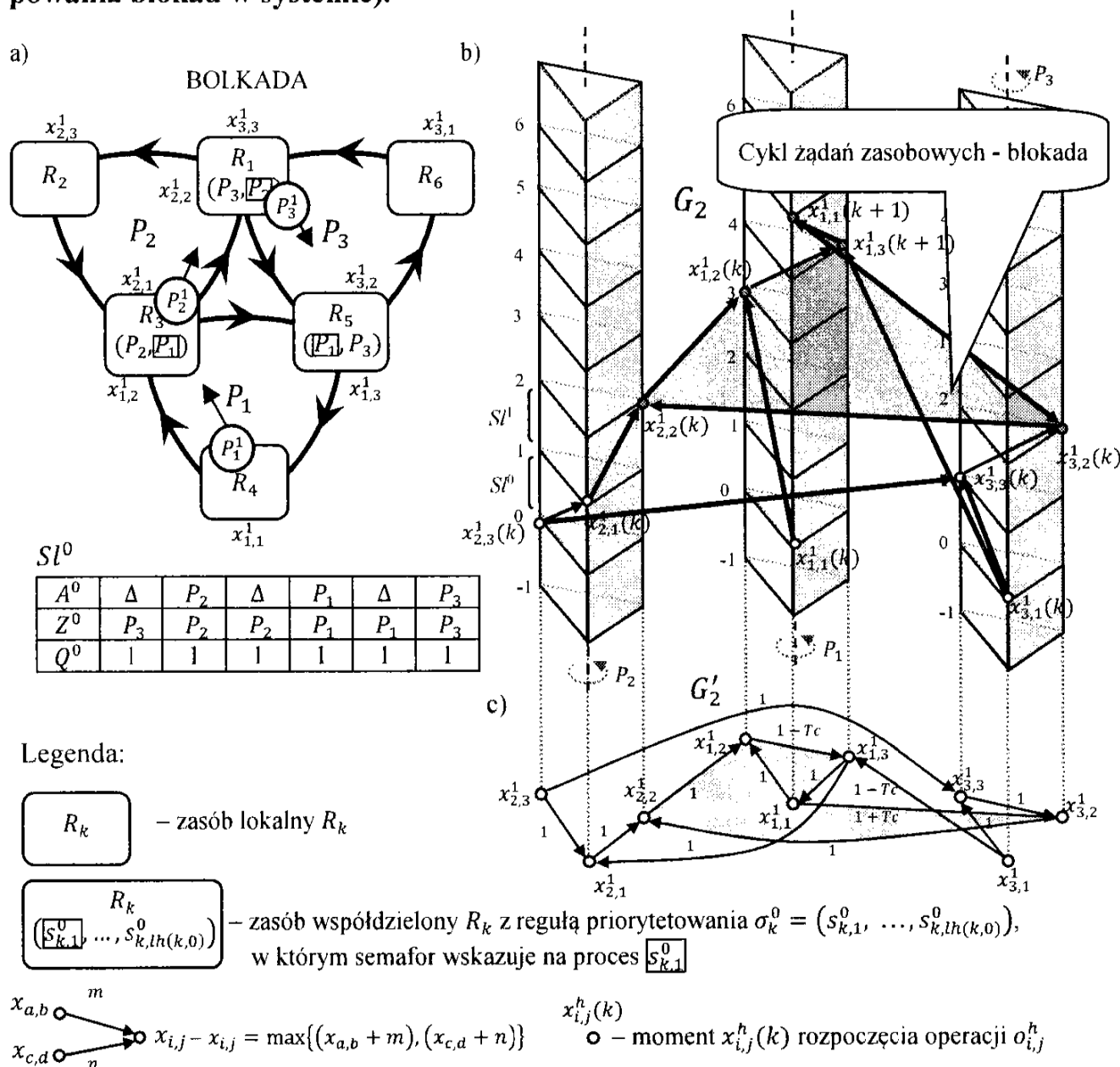


Rys. 3.19. Związki łączące cykliczny przebieg ustalony  $D_C$ , digraf pierwszeństwa operacji i harmonogram cykliczny  $X_{DC}$

Z powyższego twierdzenia wynika, że na to aby istniał harmonogram  $X_{DC}$  odpowiadający mu digraf pierwszeństwa operacji  $G$  musi być acykliczny. Przedstawiony na rys. 3.17b digraf  $G_2$  jest takim właśnie przykładem.

Celem ilustracji kontrprzykładu, wykorzystano ten sam system (rys. 3.14), w którym zmienione zostały reguły priorytetowania na zasobach  $R_1$  i  $R_5$ :  $\sigma_1^0 = (P_3^1, P_2^1)$ ,  $\sigma_5^0 = (P_1^1, P_3^1)$  oraz zwrot realizacji procesu  $P_3$  (rys. 3.20a). Dla stanu początkowego  $Sl^0$  – rys. 3.20a – system osiąga stan blokady. Zachowanie systemu reprezentowane jest przez digraf pierwszeństwa operacji  $G_2$  (rys. 3.20b) zawierający cykl oznaczony kolorem szarym. Istnienie cyklu oznacza, że digraf jest skończony (tzn. zawiera skończony zbiór wierzchołków). Digraf kończy się na poziomie wystąpienia cyklu, ze względu na sprzeczność:  $x_{i,j}^h(k) < x_{i,j}^h(k)$  wierzchołki nie mogą być dalej rozmieszczone.

Konsekwencją Twierdzenia 5 jest więc wniosek: **acykliczność digrafu pierwszeństwa operacji implikuje istnienie cyklicznego przebiegu ustalonego (brak występowania blokad w systemie).**



**Rys. 3.20.** Przykład SWMPC z trzema procesami lokalnymi a), odpowiadający mu cykliczny digraf pierwszeństwa operacji  $G_2$  b),  $G_2'$  – rzut digrafu  $G_2$  na płaszczyznę c)

### 3.2.2. Harmonogramowanie cykliczne procesów lokalnych

W kontekście Twierdzenia 5 naturalnym staje się pytanie odnoszące się do cykliczności digrafu pierwszeństwa operacji  $G$ . W dowodzie Twierdzenia 5 pokazano, że konsekwencją cykliczności digrafu pierwszeństwa operacji  $G$  jest sprzeczność ograniczeń typu (76). Inaczej mówiąc digraf jest acykliczny jeżeli relacje (76) (jak i wykorzystujące je relacje (78)) są niesprzeczne.

Przykład ograniczeń związanych z digrafem cyklicznym przedstawiono w tab. 3.3. Tabela zawiera ograniczenia (76) odpowiadające porządkowi realizacji operacji determinowanemu przez cykliczny digraf  $G_2$  (rys. 3.20b). Ograniczenia te, są wzajemnie sprzeczne (oznaczone zostały one symbolem x).

**Tabela 3.3.** Ograniczenia opisujące związki między momentami  $x_{i,j}^h(k)$  rozpoczęcia operacji determinowane przez digraf pierwszeństwa operacji  $G_2$  (rys. 3.20b)

nr	$x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k) + t_{i,(j-1)}; x_{a,b}^c(k) + \Delta t\}$ $t_{i,(j-1)} = 1, \Delta t = 1$	sprzeczność
1	$x_{1,1}^1(k) = \max\{x_{1,3}^1(k) + 1; x_{1,3}^1(k) + 1\}$	x
2	$x_{1,2}^1(k) = \max\{x_{1,1}^1(k) + 1; x_{2,2}^1(k) + 1\}$	x
3	$x_{1,3}^1(k) = \max\{x_{1,2}^1(k-1) + 1; x_{3,1}^1(k-1) + 1\}$	x
4	$x_{2,1}^1(k) = \max\{x_{2,3}^1(k) + 1; x_{1,3}^1(k) + 1\}$	
5	$x_{2,2}^1(k) = \max\{x_{2,1}^1(k) + 1; x_{3,2}^1(k) + 1\}$	x
6	$x_{3,2}^1(k) = \max\{x_{3,3}^1(k) + 1; x_{1,1}^1(k+1) + 1\}$	x
7	$x_{3,3}^1(k) = \max\{x_{2,3}^1(k) + 1; x_{3,1}^1(k) + 1\}$	

Podstawiając do wyrażenia nr 5 kolejno wyrażenia 6, 1, 3, 2 uzyskuje się wrazenie:

$$x_{2,2}^1(k) = \max\{x_{2,1}^1(k) + 1; \max\{x_{3,3}^1(k) + 1; \max\{\max\{x_{1,1}^1(k) + 1; x_{2,2}^1(k) + 1\} + 1; x_{3,1}^1(k) + 1\} + 1 + 1\} + 1\}, \quad (79)$$

które po uproszczeniu przyjmuje postać:

$$x_{2,2}^1(k) = \max\{(x_{2,1}^1(k) + 1); (x_{3,3}^1(k) + 2); (x_{1,1}^1(k) + 7); (x_{3,1}^1(k) + 5); (x_{2,2}^1(k) + 7)\}. \quad (80)$$

Wyrażenie (80) jest sprzeczne – jest ono szczególną postacią sprzecznego równania:  $x = \max\{a; (x + b)\}$ ,  $b > 0$  Można zatem przyjąć, że ocena cykliczności digrafu  $G$  sprowadza się do weryfikacji sprzeczności ograniczeń (76).

W ogólnym przypadku weryfikacja tego typu, sprowadza się do rozwiązania następującego problemu spełniania ograniczeń:

$$PS_X = ((X_{DC}, D_X), C_X), \quad (81)$$

gdzie:

$X_{DC}$  – harmonogram cykliczny (78),

$D_X$  – dziedzina określająca dopuszczalne wartości momentów rozpoczęcia operacji

$x_{i,j}^h(k) \in \mathbb{C}$ ,

$C_X$  – zbiór ograniczeń określających związki między momentami  $x_{i,j}^h(k)$  rozpoczęcia operacji jak w (78):

$$C_X = \{(x_{i,j}^h(k+1) - x_{i,j}^h(k) = \alpha; x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}) \mid (v_{i,(j-1)}^h(k'), v_{i,j}^h(k)) \in \mathbb{E}_G; (x_{a,b}^c(k''), v_{i,j}^h(k)) \in \mathbb{E}_G; i = 1 \dots ln; j = 1 \dots lr(i); h = 1 \dots ls(i); k, k', k'' \in \mathbb{C}\}.$$

Rozwiązaniem problemu (81) jest zbiór wartości momentów  $x_{i,j}^h(k)$  rozpoczęcia operacji gwarantujących spełnienie ograniczeń  $C_X$ , dotyczących cyklicznej realizacji procesów systemu o strukturze  $SC^{lp}$ . **Oznacza to, że jeżeli istnieje rozwiązanie tego problemu to relacje (76) są niesprzeczne, digraf  $G$  jest acykliczny a więc w systemie opisanym strukturą  $SC^{lp}$  osiągalny jest cykliczny przebieg ustalony.** Okazuje się, że rozwiązanie tak sformułowanego problemu jest dość kłopotliwe co wynika m.in. z faktu, że zbiór  $X_{DC}$  jest nieskończony, to znaczy zawierający okresowo powtarzające się (odpowiadające kolejnym cyklom  $k$ ) wartości zmiennych  $x_{i,j}^h(k)$  – jak na przykład w tab. 3.1. W celu uniknięcia konieczności wyznaczania wszystkich elementów zbioru  $X_{DC}$  wystarczy wyznaczyć tylko te, które powtarzane okresowo pozwolą odtworzyć wartość  $x_{i,j}^h(k)$  dla dowolnego cyklu  $k$ .

Zmienne  $x_{i,j}^h(k)$  można uniezależnić od zmiennej  $k$  poprzez wprowadzenie zmiennych zastępczych  $x_{i,j}^h$  odnoszących się wyłącznie do cyklu zerowego  $k = 0$ . W tym celu stosuje się wykorzystywaną już w ramach opisu harmonogramów cyklicznych (1), (10), (11) zależność:

$$x_{i,j}^h(k) = x_{i,j}^h + k \cdot \alpha, k \in \mathbb{C}, \quad (82)$$

gdzie:

$k$  – oznacza numer cyklu w ramach, którego wykonywana jest operacja  $o_{i,j}^h$ ,  
 $x_{i,j}^h = x_{i,j}^h(0)$  – moment rozpoczęcia operacji  $o_{i,j}^h$  dla 0-ego cyklu systemu.

Podstawiając (82) do wyrażenia (76) uzyskuje się:

$$x_{i,j}^h + k \cdot \alpha = \max\{(x_{i,(j-1)}^h + k' \cdot \alpha) + t_{i,(j-1)}; (x_{a,b}^c + k'' \cdot \alpha) + \Delta t\}. \quad (83)$$

Zakładając, że operacje  $o_{i,(j-1)}^h$ ,  $o_{a,b}^c$  (o momentach rozpoczęcia  $x_{i,(j-1)}^h(k')$ ,  $x_{a,b}^c(k'')$ ) są realizowane w ramach tego samego cyklu  $k$ , co operacja  $o_{i,j}^h$ :  $k' = k'' = k$ , oraz uwzględniając, że:  $\max\{A + B; C + B\} = \max\{A; C\} + B$  otrzymuje się następnie:

$$x_{i,j}^h + k \cdot \alpha = \max\{x_{i,(j-1)}^h + t_{i,(j-1)}; x_{a,b}^c + \Delta t\} + k \cdot \alpha, \quad (84)$$

$$x_{i,j}^h = \max\{x_{i,(j-1)}^h + t_{i,(j-1)}; x_{a,b}^c + \Delta t\}. \quad (85)$$

Z kolei w sytuacji, gdy operacje  $o_{i,(j-1)}^h$ ,  $o_{a,b}^c$  są realizowane w ramach cyklu poprzedzającego operację  $o_{i,j}^h$ :  $k' = k'' = k - 1$ , wyrażenie (83) upraszcza się do postaci:

$$x_{i,j}^h = \max\{x_{i,(j-1)}^h + t_{i,(j-1)} - \alpha; x_{a,b}^c + \Delta t - \alpha\}. \quad (86)$$

Wyrażenia (85), (86) stanowią wersję ograniczenia (76) pozbawioną zmiennej  $k$ . Postępując analogicznie proponowane uproszczenia można wprowadzić do pozostałych ograniczeń (patrz tab. 3.2), związanych z digrafem  $G_1$  (patrz rys. 3.17b), i ostatecznie zestawić w tab. 3.4.

Jak widać z tab. 3.4, dla każdego wariantu wartości  $k'$  i  $k''$  (patrz tab. 3.2) wyrażenie (83) upraszcza się do postaci pozbawionej zmiennej  $k$ . Brak zmiennej  $k$  pozwala ograniczyć poszukiwania momentów rozpoczęcia operacji harmonogramu  $X_{DC}$  tylko do wartości momentów  $x_{i,j}^h$  występujących w cyklu zerowym ( $k = 0$ ).

**Tabela 3.4.** Ograniczenia opisujące związki między momentami  $x_{i,j}$  rozpoczęcia operacji determinowane przez digraf pierwszeństwa operacji  $G'_1$  (rys. 3.17c)

nr	$x_{i,j}^h + k \cdot \alpha = \max\{(x_{i,(j-1)}^h + k' \cdot \alpha) + t_{i,(j-1)}; (x_{a,b}^c + k'' \cdot \alpha) + \Delta t\}$ $t_{i,(j-1)} = 1, \Delta t = 1$
1	$x_{1,1}^1 = \max\{x_{1,3}^1 + 1 - \alpha; x_{1,3}^1 + 1 - \alpha\}$
2	$x_{1,2}^1 = \max\{x_{1,1}^1 + 1; x_{2,2}^1 + 1\}$
3	$x_{1,3}^1 = \max\{x_{1,2}^1 + 1; x_{3,3}^1 + 1\}$
4	$x_{2,1}^1 = \max\{x_{2,3}^1 + 1 - \alpha; x_{1,3}^1 + 1 - \alpha\}$
5	$x_{2,2}^1 = \max\{x_{2,1}^1 + 1; x_{3,1}^1 + 1\}$
6	$x_{2,3}^1 = \max\{x_{2,2}^1 + 1; x_{2,2}^1 + 1\}$
7	$x_{3,1}^1 = \max\{x_{3,3}^1 + 1 - \alpha; x_{3,2}^1 + 1 - \alpha\}$
8	$x_{3,2}^1 = \max\{x_{3,1}^1 + 1; x_{1,1}^1 + 1\}$
9	$x_{3,3}^1 = \max\{x_{2,3}^1 + 1 - \alpha; x_{3,3}^1 + 1\}$

Ograniczenia tab. 3.4, podobnie jak ograniczenia z tab. 3.2 odwzorowujące porządek określony przez digraf  $G_1$ , (rys. 3.17b), reprezentują związki określone przez digraf  $G'_1$  (patrz rys. 3.17c) będący rzutem digrafu  $G_1$  na płaszczyznę prostopadłą do prostopadłościaków  $P_1^1, P_2^1, P_3^1$ . Digraf  $G'$  będący rzutem digrafu pierwszeństwa operacji  $G = (\mathbb{V}_G, \mathbb{E}_G)$ , formalnie definiuje się następująco:  $G' = (\mathbb{V}'_G, \mathbb{E}'_G)$ , gdzie:  $\mathbb{V}'_G = \{v_{i,j}^h | \exists k \in \mathbb{C}, v_{i,j}^h(k) \in \mathbb{V}_G\}$ ,  $\mathbb{E}'_G = \{(x_{a,b}^c, v_{i,j}^h) | \exists k, k' \in \mathbb{C}, (x_{a,b}^c(k'), v_{i,j}^h(k)) \in \mathbb{E}_G\}$ .

W przypadku digrafu  $G'_1$  wierzchołkom  $v_{i,j}^h$  (rys. 3.17c) odpowiadają momenty  $x_{i,j}^h$  rozpoczęcia operacji cyklu zerowego ( $k = 0$ ). Oznacza to, że znając digraf  $G'_1$  oraz okres przebiegu cyklicznego  $\alpha$ , można odtworzyć digraf pierwszeństwa operacji  $G_1$ , a więc podobnie, jak ma to miejsce w wyznaczaniu wartości  $x_{i,j}^h(k)$  dla dowolnego  $k$ , w oparciu o wartość  $x_{i,j}^h$ . Digraf  $G'_1$  stanowi więc skończoną reprezentacją digrafu  $G_1$  (Bocewicz i Banaszak 2013a). Dla przykładu, rys. 3.20c przedstawia rzut  $G'_2$  cyklicznego digrafu  $G_2$ .

Uproszczenie ograniczenia (76) do postaci (85), (86) nie zmniejsza jego ogólności, a pozwala uprościć sposób opisu harmonogramu  $X_{DC}$  (78). Podobnie jak w przypadku (2), harmonogram cykliczny może być reprezentowany przez parę:

$$X_{DC} = (X'_{DC}, \alpha), \quad (87)$$

gdzie:

$$X'_{DC} = \{x_{i,j}^h | x_{i,j}^h + k \cdot \alpha = \max\{(x_{i,(j-1)}^h + k' \cdot \alpha) + t_{i,(j-1)}; (x_{a,b}^c + k'' \cdot \alpha) + \Delta t\};$$

$$\begin{aligned} (v_{i,(j-1)}^h(k'), v_{i,j}^h(k)) \in \mathbb{E}_G; (x_{a,b}^c(k''), v_{i,j}^h(k)) \in \mathbb{E}_G; \\ i = 1 \dots ln; j = 1 \dots lr(i); h = 1 \dots ls(i); k, k', k'' \in \mathbb{C} \end{aligned} \quad (88)$$

W rozważanym przypadku poszukiwanie harmonogramu  $X_{DC}$  sprowadza się do wyznaczenia zbioru  $X'_{DC}$  oraz okresu  $\alpha$ . W odróżnieniu od (78) poszukiwany zbiór jest zbiorem skończonym (liczność  $X'_{DC}$  wynosi:  $|X'_{DC}| = \sum_{i=1}^{ln} (lr(i) + ls(i))$ ). Analogicznie jak w przypadku harmonogramu  $X_{DC}$  wyznaczenie zbioru  $X'_{DC}$  oraz wartości okresu  $\alpha$ , sprowadza się do rozwiązania następującego problemu spełniania ograniczeń:

$$PS'_X = ((V_{PS}, D_{PS}), C'_X), \quad (89)$$

gdzie:

$V_{PS} = X'_{DC} \cup \{\alpha\}$  – zbiór zmiennych decyzyjnych:  $X'_{DC}$  – zbiór momentów rozpoczęcia operacji (88),  $\alpha$  – okres przebiegu cyklicznego procesów lokalnych,

$D_{PS} = \{D_X, D_\alpha\}$  – zbiór dziedzin zmiennych decyzyjnych:  $D_X$  – dziedzina określająca dopuszczalne wartości momentów rozpoczęcia operacji, przyjmuje się, że dla każdego  $x_{i,j}^h \in \mathbb{C}$ ,  $D_\alpha$  – dziedzina okresu  $\alpha$ ,  $\alpha \in \mathbb{N}^+$ ,

$C'_X$  – zbiór ograniczeń określających związki między momentami  $x_{i,j}^h$  rozpoczęcia operacji (83):

$$\begin{aligned} C'_X = \{ x_{i,j}^h + k \cdot \alpha = \max\{ (x_{i,(j-1)}^h + k' \cdot \alpha) + t_{i,(j-1)}; (x_{a,b}^c + k'' \cdot \alpha) + \Delta t \} \\ (v_{i,(j-1)}^h(k'), v_{i,j}^h(k)) \in \mathbb{E}_G; (x_{a,b}^c(k''), v_{i,j}^h(k)) \in \mathbb{E}_G; \\ i = 1 \dots ln; j = 1 \dots lr(i); h = 1 \dots ls(i); k, k', k'' \in \mathbb{C} \end{aligned}$$

W rozważanym przykładzie (rys. 3.14) zbiór  $X'_{DC}$  ma postać:

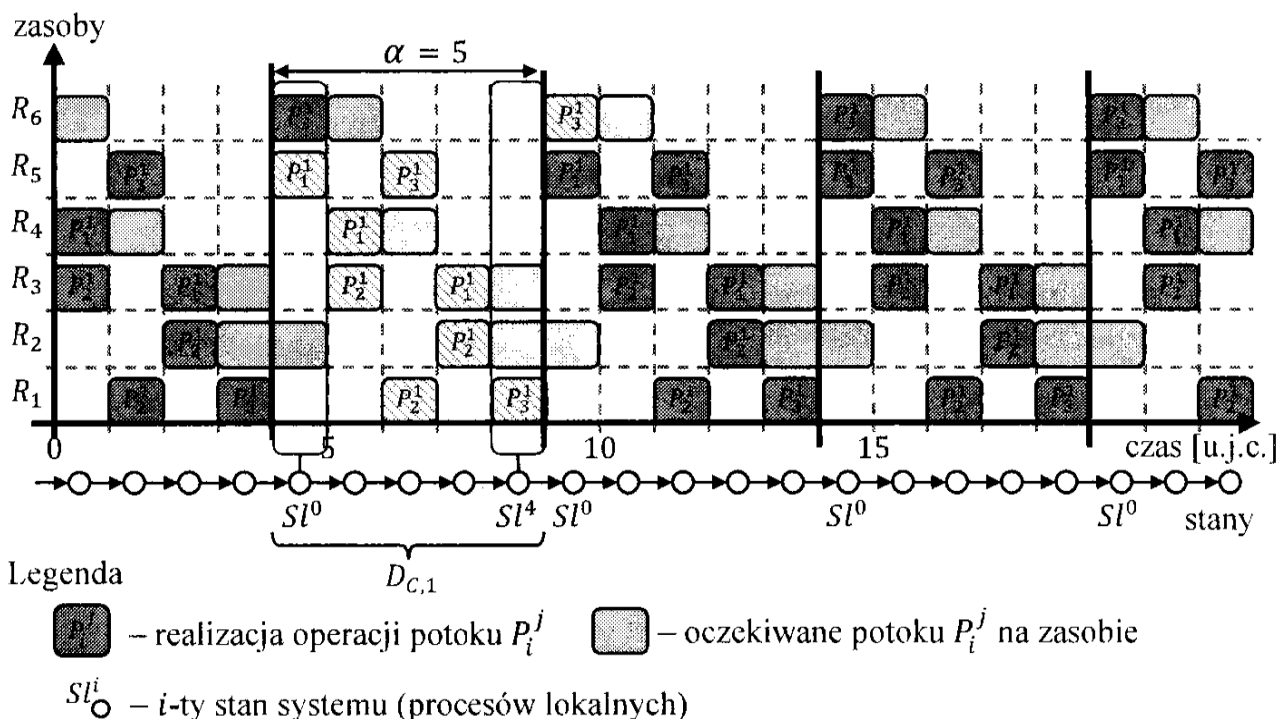
$$X'_{DC} = \{x_{1,1}^1, x_{1,2}^1, x_{1,3}^1, x_{2,1}^1, x_{2,2}^1, x_{2,3}^1, x_{3,1}^1, x_{3,2}^1, x_{3,3}^1\}, \quad (90)$$

natomiast ograniczenia  $C'_X$  przyjmują postać jak w tab. 3.4. Pierwsze rozwiązanie dopuszczalne problemu  $PS'_X$  (środowisko programowania z ograniczeniami OzMozart) dla tego typu systemu zostało przedstawione w tab. 3.5.

**Tabela 3.5.** Wartości elementów zbioru  $X'_{DC}$  i okresu  $\alpha$  dla SWMPC z rys. 3.14

	$x_{3,1}^1$	$x_{1,1}^1, x_{2,1}^1$	$x_{2,2}^1, x_{3,2}^1$	$x_{2,3}^1, x_{1,2}^1$	$x_{3,3}^1$	$x_{1,3}^1$	$\alpha$
Momenty $x_{i,j}^h$ :	-1	0	1	2	3	4	5

Uzyskane wartości pokrywają się z wartościami momentów rozpoczęcia operacji pierwszego (w ogólności  $k$ -tego) cyklu tab. 3.1. Istnienie rozwiązania problemu  $PS'_X$  oznacza niesprzeczność ograniczeń  $C'_X$  i w konsekwencji acykliczność digrafu  $G_1$ . System z rys. 3.14 osiąga więc przebieg cykliczny o okresie  $\alpha = 5$  u.j.c. – harmonogram realizacji operacji odpowiadający uzyskanemu przebiegowi  $D_{C1}$  przedstawia rys. 3.21.



**Rys. 3.21.** Harmonogram realizacji operacji ustalonego przebiegu cyklicznego  $D_{C,1}$  systemu z rys. 3.14

Dotychczasowe rozważania podsumować można następująco:

- każdemu cyklicznemu przebiegowi ustalonemu  $D_C$  przestrzeni  $\mathcal{P}^0$  odpowiada acykliczny digraf pierwszeństwa operacji  $G$ ,
- istnienie cyklicznego przebiegu  $D_C$  weryfikuje cykliczność odpowiadającego mu digrafu pierwszeństwa  $G$ ,
- jeżeli zbiór ograniczeń (reprezentujących porządek wykonywania operacji) determinowanych przez układ łuków i wierzchołków (83) rzutu  $G'$  digrafu  $G$  jest niesprzeczny, wówczas digraf pierwszeństwa operacji  $G$  jest acykliczny,
- niesprzeczność ograniczeń (83) weryfikuje niepustość zbioru rozwiązań problemu spełniania ograniczeń  $PS'_X$  (89),
- rozwiązanie  $PS'_X$  (89) implikuje istnienie cyklicznego harmonogramu  $X_{DC}$  (87).

**Reasumując, acykliczność digrafu pierwszeństwa operacji  $G$  jest warunkiem wystarczającym istnienia cyklicznego przebiegu ustalonego  $D_C$ . Warunek ten jest weryfikowany poprzez ocenę sprzeczności ograniczeń (83), sprowadzającą się do rozwiązania odpowiedniego problemu spełniania ograniczeń  $PS'_X$  (89). Oznacza to, że ocena istnienia cyklicznego przebiegu ustalonego  $D_C$  w przestrzeni  $\mathcal{P}^0$  sprowadza się do rozwiązania problemu  $PS'_X$  (89).**

### 3.2.3. Harmonogramowanie cykliczne procesów multimodalnych

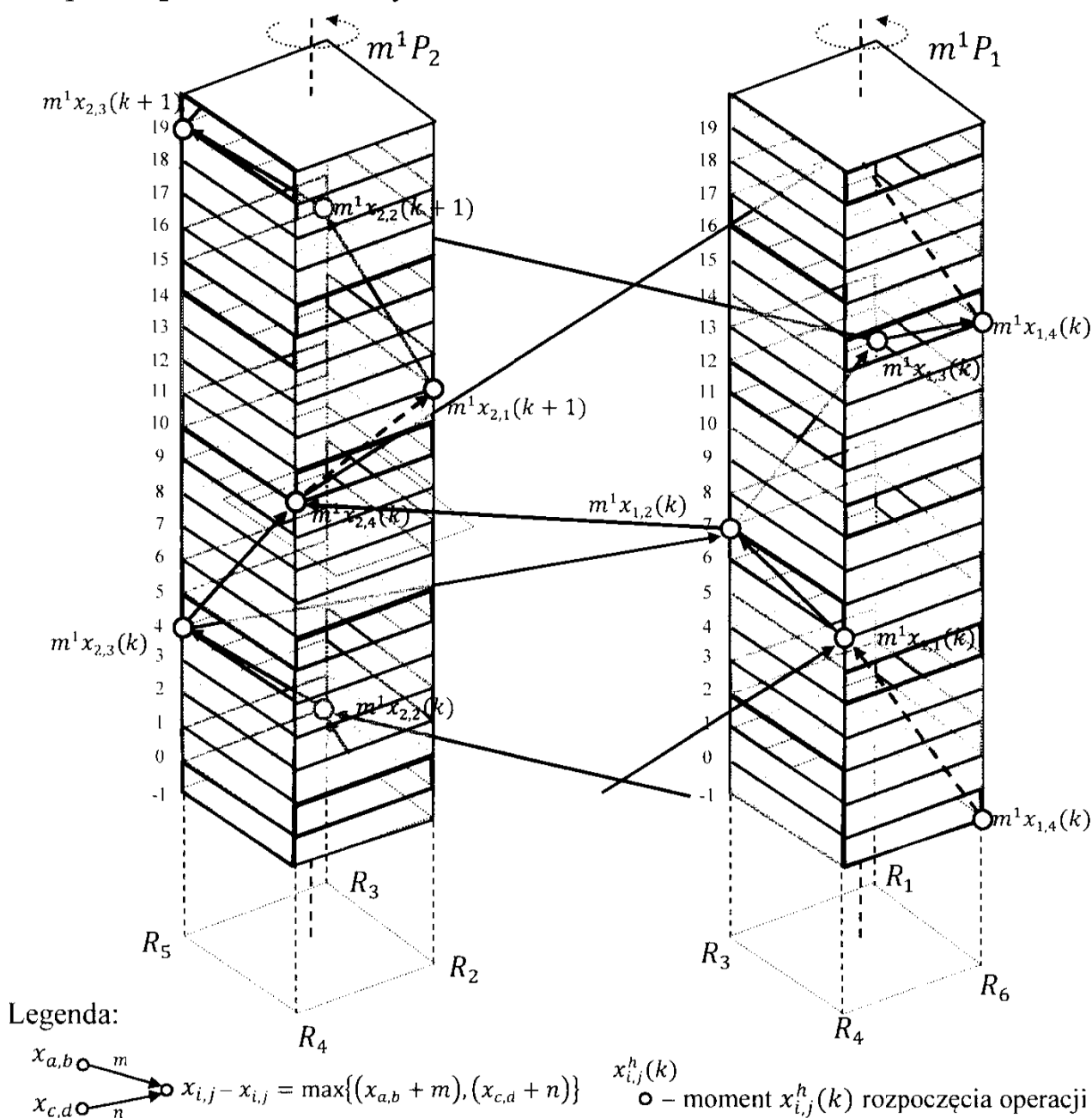
Analizowany wariant digrafu pierwszeństwa operacji odnosi się tylko do poziomu zachowań procesów lokalnych  $SL$  (charakteryzowany przestrzenią stanów  $\mathcal{P}^0$ ). Ze względu na to, że realizacja procesów multimodalnych odbywa się podobnie jak



procesów lokalnych, analogicznych własności należy się spodziewać również na wyższych poziomach zachowań  $SM^l$ .

Procesy multimodalne  $m^l P_i$  wykorzystują procesy niższego poziomu  $m^{(l-1)} P_i$  do realizacji swoich operacji. Znaczący to, że realizacja operacji procesu multimodalnego  $m^l o_{i,j}^h$  jest warunkowana realizacją (na tym samym zasobie), operacji  $m^{(l-1)} o_{a,b}^c$  potoku żądanego procesu  $m^l \mu_{i,j}$ . Każda operacja procesu multimodalnego jest wykonywana jednocześnie z operacją żądanego procesu niższego poziomu (patrz rys. 2.4). Oznacza to, że wartość momentu  $m^l x_{i,j}^h(k)$  rozpoczęcia operacji  $m^l o_{i,j}^h$  procesu multimodalnego jest limitowana przez wartość momentu  $m^{(l-1)} x_{a,b}^c(k)$  rozpoczęcia operacji żądanego procesu (w szczególności momentu rozpoczęcia operacji procesu lokalnego  $x_{i,j}^h(k)$ ).

Rys. 3.22 przedstawia digraf pierwszeństwa operacji  $G_1^1$  procesów multimodalnych  $m^1 P_1, m^1 P_2$ , dla SWMPC z rys. 3.14.



Rys. 3.22. Digraf pierwszeństwa operacji  $G_1^1$  dla procesów multimodalnych  $m^1 P_1, m^1 P_2$  systemu z rys. 3.14

W odróżnieniu od digrafu  $G_1$  wierzchołki  $m^1 v_{i,j}(k)$  digrafu  $G_1^1$  nie mogą być rozmieszczane na dowolnej wysokości. Ze względu na to, że procesy multimodalne wykorzystują do swojej realizacji procesy lokalne tak więc, operacje procesów multimodalnych oraz odpowiednie operacje żądanych procesów lokalnych, muszą rozpoczynać się jednocześnie. Oznacza to, że wyznaczony w ramach poziomu lokalnego harmonogram  $X_{DC}$  determinuje potencjalne wartości momentów rozpoczęcia operacji procesów  $m^1 P_1$ ,  $m^1 P_2$  (poziomy, na których można rozmieścić wierzchołki  $m^1 v_{i,j}(k)$ ).

Przykładowo, operacja  $m^1 x_{2,2}(k)$  procesu  $m^1 P_2$  realizowana na zasobie  $R_3$  (rys. 3.14) wykorzystuje proces  $P_2^1$  wykonywany na zasobie  $R_3$  w momencie określonym przez  $x_{2,1}^1(k)$ .

Zgodnie z wyznaczonym harmonogramem:  $x_{2,1}^1(k) = 0, 5, 10, 15, \dots$ , wartości te określają momenty, w których potok  $P_2^1$  pojawia się na zasobie  $R_3$  (tylko w tych momentach potok  $m^1 P_2$  może rozpocząć operacje na tym zasobie). Tak więc wierzchołek  $m^1 v_{2,2}(k)$  związany z operacją potoku  $m^1 P_2$  może być rozmieszczony wyłącznie na jednym z następujących poziomów: 0, 5, 10, 15, ... Poziomy, na których mogą być rozmieszczane wierzchołki  $m^1 v_{i,j}(k)$  digrafu  $G_1^1$  (determinowane przez harmonogram  $X_{DC}$ ) zostały oznaczone, na rys. 3.22, przez czerwone i niebieskie prostokąty.

Zgodnie z Twierdzeniem 5 (jest ono prawdziwe dla każdego  $l$ -tego poziomu zachowań), jeżeli digraf pierwszeństwa operacji jest acykliczny to w systemie osiągalny jest harmonogram  $m^l X_{DC}$  reprezentujący cykliczny przebieg ustalony  $l$ -tego poziomu zachowań.

Analogicznie jak w przypadku procesów lokalnych, **harmonogram cykliczny procesów  $l$ -tego poziomu  $m^l X_{DC}$**  może być wyrażony jako para:

$$m^l X_{DC} = (m^l X'_{DC}, m^l \alpha), \quad (91)$$

gdzie:

$$m^l X'_{DC} = \{m^l x_{i,j}^h \mid m^l x_{i,j}^h \in \mathcal{X}_{i,j}^l; m^l x_{i,j}^h = [m^l x_{i,j}^{h*}]_{\mathcal{X}_{i,j}^l};$$

$$m^l x_{i,j}^{h*} + k \cdot m^l \alpha = \max\{(m^l x_{i,(j-1)}^h + k' \cdot m^l \alpha) + t_{i,(j-1)}; (m^l x_{a,b}^c + k'' \cdot m^l \alpha) + \Delta t\}$$

$$(m^l v_{i,(j-1)}^h(k'), m^l v_{i,j}^h(k)) \in \mathbb{E}_G^l; (m^l x_{a,b}^c(k''), m^l v_{i,j}^h(k)) \in \mathbb{E}_G^l;$$

$$i = 1 \dots lw(l); j = 1 \dots lm(i, l); h = 1 \dots lsm(i, l); k, k', k'' \in \mathbb{C}, \quad (92)$$

gdzie:

$m^l x_{i,j}^h$  – moment rozpoczęcia operacji  $m^l o_{i,j}^h$ ,  $h$ -tego potoku procesu multimodalnego  $m^l P_i$  w 0-wym cyklu pracy systemu,

$\mathcal{X}_{i,j}^l$  – zbiór wartości  $m^l x_{i,j}^h$  determinowany przez harmonogram niższego  $(l-1)$  poziomu  $m^{(l-1)} X_{DC}$ ,

$[a]_B$  – cecha górna liczby  $a$  względem zbioru  $B$ :  $[a]_B = \min\{k \in B: k \geq a\}$

$m^l v_{i,j}^h(k)$  – wierzchołek digrafu pierwszeństwa operacji  $G^l$  dla  $l$ -tego poziomu zachowań:  $m^l v_{i,j}^h(k) \in \mathbb{V}_G^l$ , gdzie:  $\mathbb{V}_G^l$  – zbiór wierzchołków digrafu  $G^l$ ,

$\mathbb{E}_G^l$  – zbiór łuków digrafu  $G^l$ :  $\mathbb{E}_G^l \subseteq \mathbb{V}_G^l \times \mathbb{V}_G^l$ ,

$lw(l)$  – liczba procesów multimodalnych  $m^l P_i$  należących do  $l$ -tego poziomu zachowań,  $lm(i, l)$  – liczba operacji potoku  $m^l P_i^h$  procesu multimodalnego  $m^l P_i$ ;  $ls(i)$  – liczba potoków procesu  $m^l P_i$ .

$m^l \alpha$  – okres ustalonego przebiegu cyklicznego dla  $l$ -tego poziomu zachowań.

Podstawowym elementem harmonogramu  $m^l X_{DC}$  jest zbiór  $m^l X'_{DC}$  (92) zawierający momenty rozpoczęcia operacji w cyklu zerowym  $k = 0$ . Wartości momentów  $m^l x_{i,j}^h$  są limitowane przez harmonogram niższego poziomu (zbiór  $\mathcal{X}_{i,j}^l$ ) oraz porządek wykonywania operacji determinowany przez układ wierzchołków i łuków digrafu pierwszeństwa operacji  $G^l$ . Do odwzorowania kolejności realizacji operacji  $m^l o_{i,j}^h$  (wynikającej ze struktury  $SC^{lp}$  – marszrut procesów, reguł priorytowania  $\theta^l$ ) stosowana jest relacja typu *max*, odpowiadająca relacji (83) procesów lokalnych. Ze względu na to, że wartość  $m^l x_{i,j}^{h*}$  spełniająca relację *max* nie zawsze musi być elementem zbioru  $\mathcal{X}_{i,j}^l$  stosuje się „przybliżenie” (cecha górna  $m^l x_{i,j}^h = \lceil m^l x_{i,j}^{h*} \rceil_{\mathcal{X}_{i,j}^l}$ ) otrzymanej wartości do pierwszej większej należącej do  $\mathcal{X}_{i,j}^l$ . W praktyce oznacza to, że potok  $m^l P_i^h$  ukończył swoją bieżącą operację, otrzymał dostęp do żądanego zasobu ale musi jeszcze odczekać na dostęp do żądanego procesu  $m^l \mu_{i,j}$ . Sytuacje tego typu przedstawiono na rys. 2.4.

Weryfikacja cykliczności digrafu pierwszeństwa operacji podobnie jak w przypadku procesów lokalnych sprowadza się do wyznaczenia harmonogramu  $m^l X_{DC}$  spełniającego zbiór ograniczeń określonych w ramach (92), tzn. stanowiącego rozwiązanie poniższego problemu spełniania ograniczeń:

$$m^l PS'_X = \left( (m^l V_{PS}, D_{PS}^l), C_X^l \right), \quad (93)$$

gdzie:

$m^l V_{PS} = m^l X'_{DC} \cup \{m^l \alpha\}$  – zbiór zmiennych decyzyjnych:  $m^l X'_{DC}$  – zbiór momentów rozpoczęcia operacji (92),  $m^l \alpha$  – okres przebiegu cyklicznego procesów multimodalnych  $l$ -tego poziomu,

$D_{PS}^l = \{D_X^l, D_\alpha^l\}$  – zbiór dziedzin zmiennych decyzyjnych,  $D_X^l$  – dziedzina określająca dopuszczalne wartości momentów rozpoczęcia operacji, przyjmuje się, że  $m^l x_{i,j}^h \in \mathbb{C}$ ,  $D_\alpha^l$  – dziedzina okresu  $m^l \alpha$ ,  $m^l \alpha \in \mathbb{N}^+$ ,

$C_X^l$  – ograniczenia określające związki między momentami  $m^l x_{i,j}^h$  rozpoczęcia operacji  $l$ -tego poziomu zachowań (92) jak i ich odpuszczalne wartości  $\mathcal{X}_{i,j}^l$ .

W przypadku problemu  $m^1 PS'_X$ , charakteryzującego poziom procesów multimodalnych  $m^1 P_1, m^1 P_2$  systemu z rys. 3.14, zbiór  $m^1 X'_{DC}$  zawiera 8 elementów:

$$m^1 X'_{DC} = \{m^1 x_{1,1}, m^1 x_{1,2}, m^1 x_{1,3}, m^1 x_{1,4}, m^1 x_{2,1}, m^1 x_{2,2}, m^1 x_{2,3}, m^1 x_{2,4}\}, \quad (94)$$

natomiast zbiór ograniczeń  $C'_X$  obejmuje ograniczenia przedstawione w tab. 3.6. Tabela zawiera uproszczone (pozbawione zmiennej  $k$ ) postacie ograniczeń (92) (typu *max*) oraz zbiory  $\mathcal{X}_{i,j}^l$  dopuszczalnych wartości momentów rozpoczęcia operacji (określone na podstawie tab. 3.1 i 3.5).

Pierwsze rozwiązanie dopuszczalne problemu  $m^1PS'_X$  uzyskane w środowisku programowania z ograniczeniami OzMozart przedstawia tab. 3.7, a odpowiadający mu harmonogram rys. 3.23.

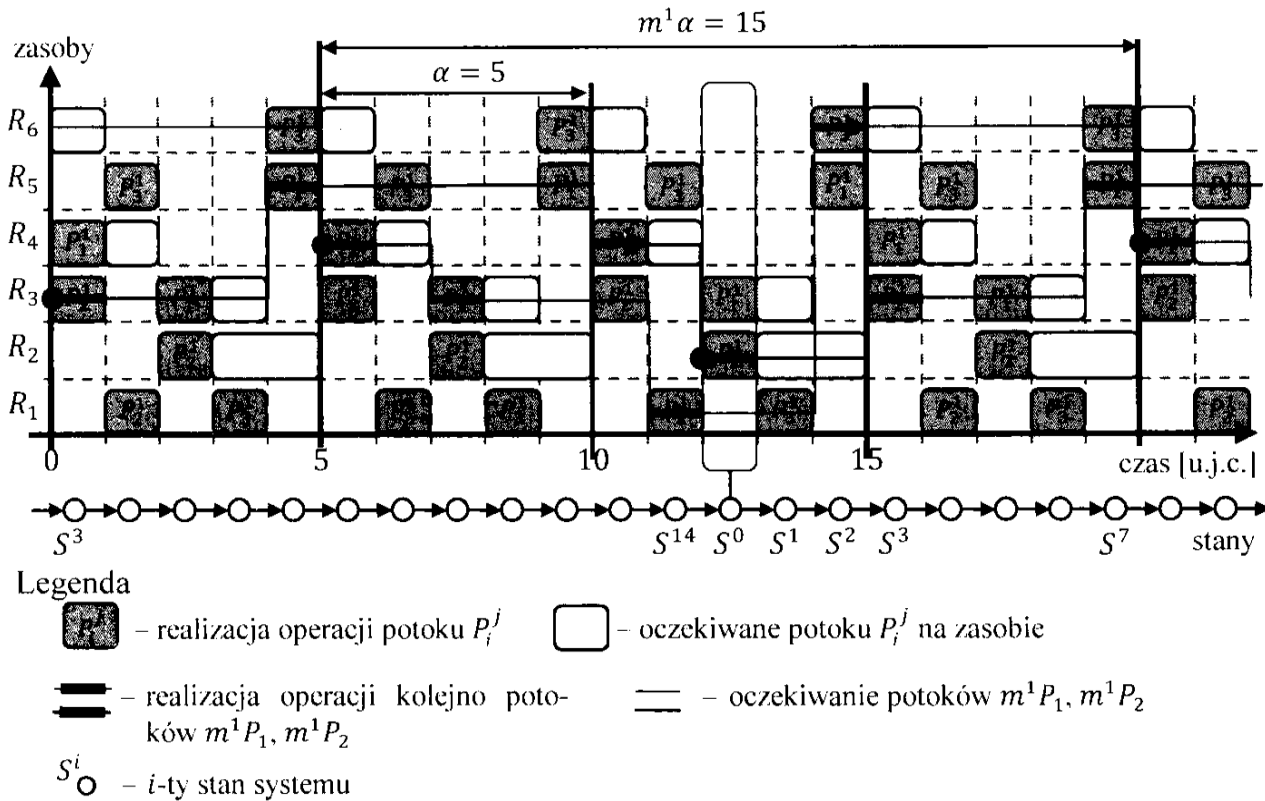
Procesy rozważanego systemu (rys. 3.14) mogą być realizowane cyklicznie. Wynika to z faktu, że dla każdego poziomu zachowań istnieje cykliczny przebieg ustalony. Dla procesów lokalnych okres przebiegu cyklicznego wynosi:  $\alpha = 5$  u.j.c., dla procesów multimodalnych  $m^1\alpha = 15$  u.j.c.

**Tabela 3.6.** Ograniczenia opisujące związki między momentami  $m^1x_{i,j}^h$  rozpoczęcia operacji procesów  $m^1P_1$ ,  $m^1P_2$  determinowane przez digraf pierwszeństwa operacji  $G_1^1$  (rys. 3.22)

nr	$m^1x_{i,j}^h = \lceil m^1x_{i,j}^{h*} \rceil_{x_{i,j}^1}$ $m^1x_{i,j}^{h*} + k \cdot \alpha = \max\{(m^1x_{i,(j-1)}^h + k' \cdot m^1\alpha) + t_{i,(j-1)}; (m^1x_{a,b}^c + k'' \cdot m^1\alpha) + \Delta t\}$ $m^1t_{i,(j-1)} = 1, \Delta t = 1$
1	$m^1x_{1,1} = \lceil \max\{m^1x_{1,4} + 1 - m^1\alpha; m^1x_{2,1} + 1 - m^1\alpha\} \rceil_{x_{1,1}^1}$
2	$m^1x_{1,2} = \lceil \max\{m^1x_{1,1} + 1; m^1x_{2,3} + 1\} \rceil_{x_{1,2}^1}$
3	$m^1x_{1,3} = \lceil \max\{m^1x_{1,2} + 1; m^1x_{1,4} + 1 - m^1\alpha\} \rceil_{x_{1,3}^1}$
4	$m^1x_{1,4} = \lceil \max\{m^1x_{1,3} + 1; m^1x_{1,1} + 1\} \rceil_{x_{1,4}^1}$
5	$m^1x_{2,1} = \lceil \max\{m^1x_{2,4} + 1 - m^1\alpha; m^1x_{2,2} + 1 - m^1\alpha\} \rceil_{x_{2,1}^1}$
6	$m^1x_{2,2} = \lceil \max\{m^1x_{2,1} + 1; m^1x_{1,2} - 1 - m^1\alpha\} \rceil_{x_{2,2}^1}$
7	$m^1x_{2,3} = \lceil \max\{m^1x_{2,2} + 1; m^1x_{2,2} + 1\} \rceil_{x_{2,3}^1}$
8	$m^1x_{2,4} = \lceil \max\{m^1x_{2,3} + 1; m^1x_{1,2} + 1\} \rceil_{x_{2,4}^1}$
Zbiory wartości dopuszczalnych $m^1x_{i,j}^h$ (na podstawie tab. 3.1 i 3.5)	
1	$m^1x_{1,1} \in X_{1,1}^1 = \{x_{1,1}^1(k) \mid x_{1,1}^1(k) = x_{1,1}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 0, 5, 10, 15, \dots\}$
2	$m^1x_{1,2} \in X_{1,2}^1 = \{x_{1,2}^1(k) \mid x_{1,2}^1(k) = x_{1,2}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 2, 7, 12, 17, \dots\}$
3	$m^1x_{1,3} \in X_{1,3}^1 = \{x_{1,3}^1(k) \mid x_{1,3}^1(k) = x_{1,3}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 1, 6, 11, 16, \dots\}$
4	$m^1x_{1,4} \in X_{1,4}^1 = \{x_{1,4}^1(k) \mid x_{1,4}^1(k) = x_{1,4}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 4, 9, 14, 19, \dots\}$
5	$m^1x_{2,1} \in X_{2,1}^1 = \{x_{2,1}^1(k) \mid x_{2,1}^1(k) = x_{2,1}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 2, 7, 12, 17, \dots\}$
6	$m^1x_{2,2} \in X_{2,2}^1 = \{x_{2,2}^1(k) \mid x_{2,2}^1(k) = x_{2,2}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 0, 5, 10, 15, \dots\}$
7	$m^1x_{2,3} \in X_{2,3}^1 = \{x_{2,3}^1(k) \mid x_{2,3}^1(k) = x_{2,3}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 4, 9, 14, 19, \dots\}$
8	$m^1x_{2,4} \in X_{2,4}^1 = \{x_{2,4}^1(k) \mid x_{2,4}^1(k) = x_{2,4}^1 + k \cdot \alpha, k \in \mathbb{C}\} = \{\dots, 0, 5, 10, 15, \dots\}$

**Tabela 3.7.** Wartości elementów zbioru  $m^1X'_{DC}$  i okresu  $m^1\alpha$  dla SWMPC z rys. 3.14

	$m^1x_{2,1}$	$m^1x_{2,2}$	$m^1x_{2,3}$	$m^1x_{1,1}$	$m^1x_{1,2}$	$m^1x_{2,4}$	$m^1x_{1,3}$	$m^1x_{1,4}$	$m^1\alpha$
Momenty $m^1x_{i,j}$ :	-3	0	4	5	7	10	11	14	15



Rys. 3.23. Harmonogram realizacji operacji procesów multimodalnych i lokalnych systemu z rys. 3.14

W ogólnym przypadku, przy większej liczbie poziomów zachowań:  $SL, SM^1, \dots, SM^{lp}$ , należy dokonać tego typu weryfikacji (rozwiązania  $m^l PS'_X$ ) kolejno dla każdego z poziomów. Pozytywne rozwiązanie, dla  $l$ -tego poziomu zachowań, determinuje dopuszczalne wartości momentów rozpoczęcia operacji (zbiory  $\mathcal{X}_{i,j}^{(l+1)}$ ) poziomów wyższych. Postępowanie to odpowiada schematowi z rys. 3.15 i można je sprowadzić do następującego algorytmu.

### Algorytm 3

**function** CYCLICSCHEDULEGENERATION( $D_X, D_\alpha, D_X^1, D_\alpha^1, \dots, D_X^{lp}, D_\alpha^{lp}, SC^{lp}$ )

$C'_X \leftarrow \text{CREATECONSTRAINTSL}(SL)$

$PS'_X \leftarrow ((X'_{DC} \cup \{\alpha\}, D_{PS}), C'_X)$

$(X'_{DC}, \alpha) \leftarrow \text{SEARCHONE}(PS'_X)$

$X^{lp} \leftarrow (X'_{DC}, \alpha)$

**if**  $X'_{DC} \neq \emptyset$  **and**  $lp > 1$  **then**

$C_X^1 \leftarrow \text{CREATECONSTRAINTSM}(0, X_{DC}, SM^1)$

**for**  $l \leftarrow 1$  **to**  $lp$

$m^l PS'_X \leftarrow ((m^l X'_{DC} \cup \{m^l \alpha\}, D_{PS}^l), C_X^l)$

$(m^l X'_{DC}, m^l \alpha) \leftarrow \text{SEARCHONE}(m^l PS'_X)$

**if**  $m^l X'_{DC} \neq \emptyset$  **then**

$X^{lp} \leftarrow (X^{lp}, (m^l X'_{DC}, m^l \alpha))$

```

       $C_X^{(l+1)} \leftarrow \text{CREATECONSTRAINTSM}(l, m^l X'_{DC}, SM^{(l+1)})$ 
    else
       $X^{lp} \leftarrow \emptyset$ 
    break
  end
end
end
return  $X^{lp}$ 
end

```

gdzie:

$D_X, D_\alpha, D_X^1, D_\alpha^1, \dots, D_X^{lp}, D_\alpha^{lp}, SC^{lp}$  – dane wejściowe definiowane jak w (89), (93) i (18),  $SC^{lp} = \left( (R, SL), SM^1, \dots, SM^{lp} \right)$ ,

$PS'_X$  – problem spełniania ograniczeń (89),

$m^l PS'_X$  – problem spełniania ograniczeń (93),

$X^{lp}$  – harmonogram realizacji operacji procesów systemu opisanego strukturą  $SC^{lp}$ ,

$\text{SEARCHONE}(PS)$  – funkcja zwracająca pierwsze dopuszczalne rozwiązanie problemu  $PS$ , dla  $PS'_X / m^l PS'_X$  jako rozwiązanie zwracana jest sekwencja  $(X'_{DC}, \alpha) / (m^l X'_{DC}, m^l \alpha)$  (w przypadku braku rozwiązania, każdy element sekwencji jest zbiorem pustym),

$\text{CREATECONSTRAINTSL}(SL)$  – funkcja zwracająca dla danej struktury  $SL$  (18), zbiór ograniczeń  $C'_X$  determinujący relacje między operacjami procesów lokalnych,

$\text{CREATECONSTRAINTSM}(l, m^l X'_{DC}, SM^{(l+1)})$  – funkcja zwracająca dla danej struktury  $SM^{(l+1)}$  oraz zbioru  $m^l X'_{DC}$ , zbiór ograniczeń  $C_X^{(l+1)}$  determinujący relacje między operacjami procesów poziomu zachowań  $l + 1$ .

Według algorytmu, dla danej struktury systemu  $SC^{lp}$  rozwiązywany jest problem  $PS'_X$ . Jego rozwiązaniem jest harmonogram cykliczny realizacji procesów lokalnych  $X_{DC}$ . Jeżeli harmonogram istnieje, to podejmowana jest próba wyznaczania harmonogramu cyklicznego (rozwiązanie  $m^l PS'_X$ ) dla procesów multimodalnych wyższych poziomów zachowań. Kolejno rozwiązywane problemy  $m^l PS'_X$  zawierają między innymi zbiór ograniczeń  $C_X^{(l)}$  determinowany (funkcja  $\text{CreateConstraintsM}$ ) przez harmonogram poziomu niższego  $m^{(l-1)} X_{DC}$ . W efekcie, wynikiem Algorytmu 3 jest (o ile istnieje) **harmonogram cykliczny  $X^{lp}$**  uwzględniający harmonogramy  $m^{(l)} X_{DC}$  uzyskane dla każdego poziomu zachowań. Harmonogram definiowany jest następująco:

$$X^{lp} = \left( \left( \left( \left( (X'_{DC}, \alpha), (m^1 X'_{DC}, m^1 \alpha) \right), \dots \right), \dots (m^l X'_{DC}, m^l \alpha) \right) \dots (m^{lp} X'_{DC}, m^{lp} \alpha) \right) \quad (95)$$

gdzie:

$X'_{DC}, \alpha$  – zbiór momentów rozpoczęcia operacji procesów lokalnych dla cyklu zerowego  $k = 0$  oraz okres przebiegu cyklicznego – definiowane jak w (87),  
 $m^l X'_{DC}, m^l \alpha$  – zbiór momentów rozpoczęcia operacji procesów multimodalnych  $l$ -tego poziomu zachowań oraz okres przebiegu cyklicznego – definiowane jak w (91).

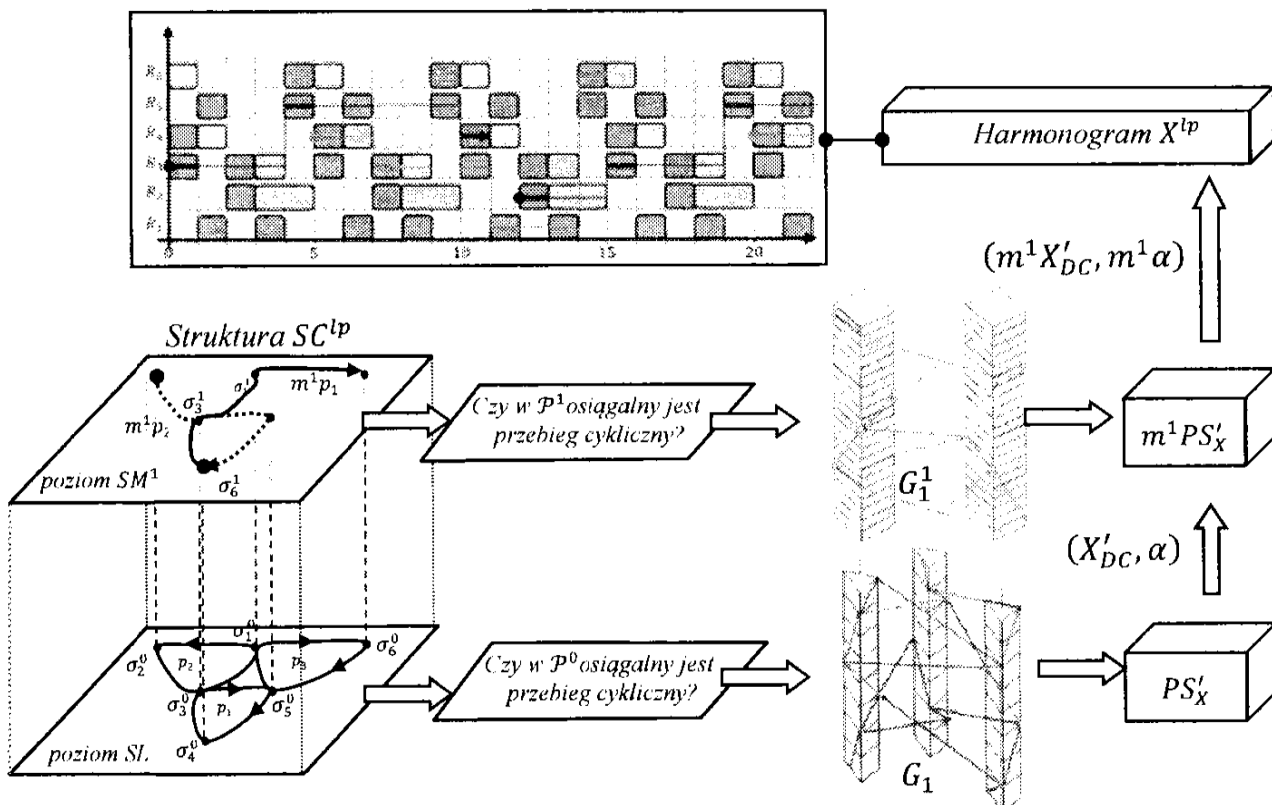
Rozważane podejście, przedstawione na rys. 3.15, można sprowadzić do postaci jak na rys. 3.24, gdzie harmonogram  $X^{lp}$  realizacji procesów systemu uzyskiwany jest na podstawie wcześniej wyznaczonych harmonogramów  $(X'_{DC}, \alpha)$ ,  $(m^1 X'_{DC}, m^1 \alpha)$  reprezentujących zachowanie procesów lokalnych (poziom  $SL$ ) i multimodalnych (poziom  $SM^1$ ). Zgodnie z (95) harmonogram  $X^{lp}$  ma postać:

$$X^{lp} = ((X'_{DC}, \alpha), (m^1 X'_{DC}, m^1 \alpha)), \tag{96}$$

gdzie wartości sekwencji  $(X'_{DC}, \alpha)$  określają harmonogram procesów lokalnych (87) (patrz tab. 3.5), a wartości sekwencji  $(m^1 X'_{DC}, m^1 \alpha)$  harmonogram procesów multimodalnych (patrz tab. 3.7).

Należy podkreślić, że istnienie harmonogramu  $m^{l+1} X_{DC}$  z  $(l + 1)$ -tego poziomu zachowań jest warunkowane spełnieniem ograniczeń  $C_X^{(l+1)}$  determinowanych przez harmonogram  $m^l X_{DC}$ . **Zbiór  $C'_X \cup C^1_X \cup \dots \cup C^{lp}_X$  można więc traktować jako warunki wystarczające istnienia harmonogramu  $X^{lp}$  (95).**

Ograniczenia tego typu są wyznaczane w każdej iteracji Algorytmu 3 i są konsekwencją wywołania funkcji CreateConstraintsM (lub CreateConstraintsL). W dotychczasowych rozważaniach ograniczenia  $C_X^{(l+1)}$  określone były w oparciu o digrafy pierwszeństwa operacji  $G_1, G_2$  lub  $G_1^1$  i reprezentowane w postaci (83) i (92).



Rys. 3.24. Schemat wyznaczania cyklicznych przebiegów ustalonych SWMPC z rys. 3.14

Postać digrafów pierwszeństwa operacji jest zwykle nieznana (problemy  $PS'_X/m^lPS'_X$  zorientowane są na ocenę cykliczności tego typu digrafów). Przyjmuje się więc, że ograniczenia charakteryzujące wzajemną kolejność wykonywania operacji determinowane są przez reguły priorytetowania  $\Theta$  oraz marszruty procesów lokalnych  $U$  i multimodalnych  $m^lU$  (18). Schemat wyznaczania ograniczeń  $C_X^{(l+1)}$  wykorzystujący parametry struktury  $\Theta$ ,  $U$ ,  $m^lU$  przedstawiono m.in. w pracy (Bocewicz i Banaśzak 2013b). Algorytmy implementowanych w nim funkcji CreateConstraintsL i CreateConstraintsM przedstawiono poniżej:

---

#### Algorytm 4

**function** CREATECONSTRAINTSL( $SL$ )

$C_X \leftarrow \emptyset$

$\Delta t \leftarrow 1$

**for**  $i \leftarrow 1$  **to**  $ln$

**for**  $h \leftarrow 1$  **to**  $ls(i)$

**for**  $j \leftarrow 1$  **to**  $lr(i)$

$(a, b, c, k'') \leftarrow \text{PREDECESSOR}(\Theta^0, (i, j, h))$

$b^* \leftarrow \text{modl}((b + 1), lr(a))$

**if**  $b + 1 > lr(a)$  **then**  $k'' \leftarrow k'' + 1$

$j^* \leftarrow \text{modl}((j - 1), lr(i))$

**if**  $j - 1 = 0$  **then**  $k' \leftarrow (-1)$  **else**  $k' \leftarrow 0$

$C_X \leftarrow C_X \cup \{x_{i,j}^h = \max\{x_{i,j^*}^h + k' \cdot \alpha + t_{i,j^*}\}; (x_{a,b^*}^c + k'' \cdot \alpha + \Delta t)\}$

**end**

**end**

**end**

**return**  $C_X$

**end**

---

#### Algorytm 5

**function** CREATECONSTRAINTSM( $l, m^lX'_{DC}, SM^{(l+1)}$ )

$C_X^{(l+1)} \leftarrow \emptyset$

$\Delta t \leftarrow 1$

**for**  $i \leftarrow 1$  **to**  $lw(l + 1)$

**for**  $h \leftarrow 1$  **to**  $lsm(i, l + 1)$

**for**  $j \leftarrow 1$  **to**  $lm(i, l + 1)$

$(a, b, c, k'') \leftarrow \text{PREDECESSOR}(\Theta^{(l+1)}, (i, j, h))$

$b^* \leftarrow \text{modl}((b + 1), lm(i, l + 1))$

**if**  $b + 1 > lr(a)$  **then**  $k'' = k'' + 1$

$j^* \leftarrow \text{modl}((j - 1), lm(i, l + 1))$

**if**  $j - 1 = 0$  **then**  $k' \leftarrow (-1)$  **else**  $k' \leftarrow 0$



---


$$C_X^{(l+1)} \leftarrow C_X^{(l+1)} \cup \left\{ "m^{(l+1)}x_{i,j}^h = [m^{(l+1)}x_{i,j}^h]^*_{x_{i,j}^{(l+1)}} " \right\}$$

$$C_X^{(l+1)} \leftarrow C_X^{(l+1)} \cup \left\{ "m^{(l+1)}x_{i,j}^h = \max\{(m^{(l+1)}x_{i,j}^h + k' \cdot m^{(l+1)}\alpha + m^{(l+1)}t_{i,j}); \right.$$

$$\left. (m^{(l+1)}x_{a,b}^c + k'' \cdot m^{(l+1)}\alpha + \Delta t)\} " \right\}$$

$$C_X^{(l+1)} \leftarrow C_X^{(l+1)} \cup \left\{ "x_{i,j}^{(l+1)} = \{m^{(l)}x_{i,j}^h(k) \mid m^{(l)}x_{i,j}^h(k) = m^{(l)}x_{i,j}^h + k \cdot m^{(l)}\alpha, k \in \mathbb{C}\} " \right\}$$

end  
end  
end  
return  $C_X^{(l+1)}$   
end

---

gdzie:

$l, m^l X'_{DC}, SL, SM^{(l+1)}$  – dane wejściowe:  $l$  – numer poziomów zachowań,  $m^l X'_{DC}$  – zbiór momentów rozpoczęcia operacji procesów multimodalnych  $l$ -tego poziomu zachowań (92),  $SL$  – struktura poziomu zachowań procesów lokalnych (18) określająca postać:  $\Theta^0, U, SM^{(l+1)}$  – struktura  $l+1$ -tego poziomu zachowań procesów multimodalnych określająca postać:  $\Theta^{(l+1)}, m^{(l+1)}U$ , itp.

$ln$  – liczba procesów lokalnych w SWMPC,

$ls(i)$  – liczba potoków  $P_i^h$  procesu lokalnego  $P_i$ ,

$lr(i)$  – liczba operacji potoku  $P_i^h$  procesu lokalnego  $P_i$ , długość marszruty  $p_i \in U$ ,

$lw(l)$  – liczba procesów multimodalnych  $m^l P_i$  należących do  $l$ -tego poziomu zachowań,

$lsm(i, l)$  – liczba potoków  $m^l P_i^h$  procesu multimodalnego  $m^l P_i$ ,

$lm(i, l)$  – liczba operacji potoku  $m^l P_i^h$  procesu multimodalnego  $m^l P_i$ , długość marszruty  $m^l p_i \in m^l U$ ,

$modl(a, b)$  – funkcja zwracająca wartość  $r = modl(a, b)$  gdy  $a = k \cdot b + r$  i  $0 < r \leq |b|$ ,  $a, b, c, k \in \mathbb{C}$ ,

$PREDECESSOR(\Theta^{(l+1)}, (i, j, h))$  – funkcja zwracająca wartości  $a, b, c, k''$ , gdzie:

$a, b, c$  okeślają indeksy operacji  $m^{(l+1)}o_{a,b}^c$  wykonywanej na wspólnym zasobie tuż przed operacją  $m^{(l+1)}o_{i,j}^h$ . Inaczej mówiąc, potok  $m^{(l+1)}P_a^c$  poprzedza potok  $m^{(l+1)}P_i^h$  w regule priorytetowania  $\sigma_d^{(l+1)}$  zasobu współdzielonego przez te potoki;  $k''$  – określa czy operacja  $m^{(l+1)}o_{a,b}^c$  jest wykonywana w ramach tego samego okresu co  $m^{(l+1)}o_{i,j}^h$ :  $k'' = 0$ , jeżeli  $\sigma_d^{(l+1)} = (\dots, m^{(l+1)}P_a^c, m^{(l+1)}P_i^h, \dots)$  oraz  $k'' = -1$ , jeżeli  $\sigma_d^{(l+1)} = (m^{(l+1)}P_i^h, \dots, m^{(l+1)}P_a^c)$  ( $m^{(l+1)}P_i^h$  – jest pierwszym elementem  $\sigma_d^{(l+1)}$  a  $m^{(l+1)}P_a^c$  ostatnim).

W obydwu algorytmach wykorzystana jest ta sama zasada wyznaczania ograniczeń. Ograniczenia odpowiadające wyrażeniom (83) (CREATECONSTRAINTSL) i (92) (CREATECONSTRAINTSM) są ograniczeniami typu *max*:

*moment rozpoczęcia operacji = max {moment ukończenia operacji poprzedniej; moment zwolnienia żadnego zasobu przez dotychczas zajmujący go potok +  $\Delta t$ }*

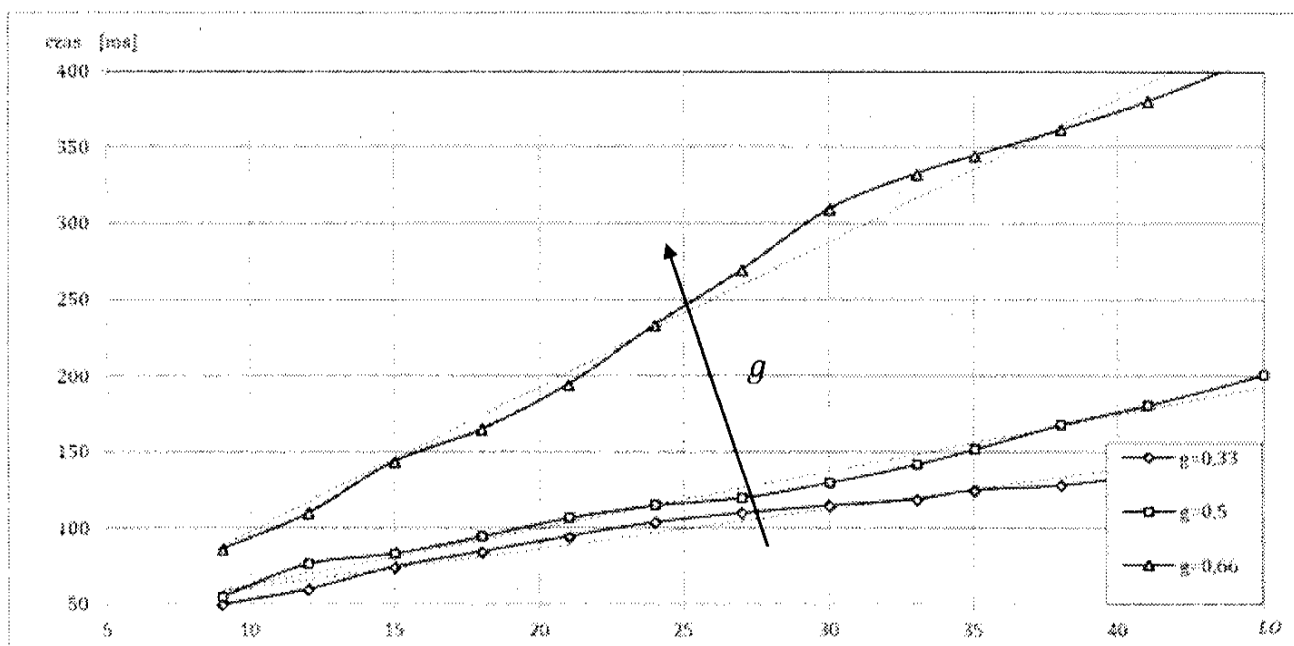
W odróżnieniu od ograniczeń rekonstruowanych na podstawie digrafu pierwszeństwa operacji w rozważanym przypadku operację poprzednią określa się na podstawie marszruty procesu ( $p_i \in U$ ,  $m^l p_i \in m^l U$ ) jako element poprzedzający operację ( $o_{i,j}^h$ ,  $m^{(l+1)} o_{i,j}^{h*}$ ), moment rozpoczęcia, której jest właśnie wyznaczany. Jest to operacja o indeksach  $o_{i,j^*}^h$ ,  $m^{(l+1)} o_{i,j^*}^h$  gdzie  $j^*$  określa numer operacji poprzedniej ( $j^* = \text{modl}((j-1), lr(i))$ ;  $j^* = \text{modl}((j-1), lm(i, l+1))$ ). Z kolei, potok dotychczas zajmujący żądany zasób i związaną z nim operację, wyznaczany jest w oparciu o regułę priorytetowania ( $\sigma_a^0$ ,  $\sigma_a^{(l+1)}$ ) przypisaną do żądanego zasobu. Wykorzystywana w tym celu funkcja Predecessor, pozwala określić indeksy  $a, b, c$  operacji:  $o_{a,b}^c$ ,  $m^{(l+1)} o_{c,b}^c$  realizowanej przez ten potok na żądanym zasobie. Przyjmuje się, że żądany zasób zostanie zwolniony z chwilą rozpoczęcia operacji kolejnej:  $o_{a,b^*}^c$ ,  $m^{(l+1)} o_{c,b^*}^c$  ( $b^* = \text{modl}((b+1), lr(a))$  lub  $b^* = \text{modl}((b+1), lm(i, l+1))$ ).

Oprócz momentów rozpoczęcia/ukończenia operacji występujących w ograniczeniach  $C_X^{(l+1)}$  sprawdzana jest również przynależność operacji do tego samego cyklu. Łatwo zauważyć, że w ramach jakiego cyklu (poprzedniego/następnego) realizowane są operacje decydujące o wartości zmiennych  $k'$  oraz  $k''$ :  $k' = -1$  oznacza cykl poprzedni względem cyklu realizacji operacji  $o_{i,j}^h$ ,  $m^{(l+1)} o_{i,j}^{h*}$ ,  $k'' = 1$  oznacza cykl następny względem cyklu realizacji operacji  $o_{i,j}^h$ ,  $m^{(l+1)} o_{i,j}^{h*}$ . Ponadto w przypadku CREATECONSTRAINTSM ograniczenia kolejnościowe typu  $max$  są uzupełnianie o ograniczenia określające zbiór  $\mathcal{X}_{i,j}^{(l+1)}$  dopuszczalnych wartości momentów rozpoczęcia operacji  $o^{(l+1)} x_{i,j}^{h*}$ , który określony jest tak samo jak w przypadku (92).

### 3.2.4. Złożoność obliczeniowa

Podobnie jak w przypadku Algorytmów 1 i 2, efektywność Algorytmu 3 jest zależna od przyjętego sposobu rozwiązywania problemów  $PS'_X$  (89) i  $m^l PS'_X$  (93). W literaturze przedmiotu znane są przypadki oceny spełnialności ograniczeń  $C'_X$  (ograniczenia  $PS'_X$ ) w których operator  $max$  zastępowany jest operatorem większości „>”. Przykładowo, w pracy (Smutnicki 2012) wykazano, że problem spełnialności tego typu ograniczeń, charakteryzuje się liniową złożonością obliczeniową  $\mathcal{O}(lk \cdot LO)$  ( $lk$  – liczba zasobów,  $LO$  – liczba operacji realizowanych w systemie). Do rozwiązania problemu wykorzystywana jest metoda wykorzystująca algorytm wyznaczania ścieżki krytycznej (zmodyfikowany algorytm Bellmana-Forda).

W przypadku problemów (89) i (93) sytuacja wygląda podobnie. Rys. 3.25 przedstawia funkcje czasu rozwiązania tego typu problemów w zależności od liczby operacji  $LO$  i gęstości struktury  $g$ . Funkcje te wyznaczone zostały w oparciu o eksperymenty przeprowadzone w środowisku programowania z ograniczeniami Oz Mozart (Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM). Czas rozwiązania problemu typu (89) i (93), w żadnym z analizowanych przypadków, nie przekracza 1 sekundy.



**Rys. 3.25.** Funkcje czasów rozwiązania problemów typu (89) i (93) (czas odpowiedzi funkcji  $\text{SEARCHONE}(PS)$ ) dla struktur o gęstościach:  $g = 0,33$ ;  $g = 0,5$ ;  $g = 0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM)

Z wykresu wynika, że ocena osiągalności cyklicznego przebiegu ustalonego na danym poziomie zachowań (przestrzeni stanów  $\mathcal{P}^l$ ) charakteryzuje się liniową złożonością obliczeniową (rys. 3.25).

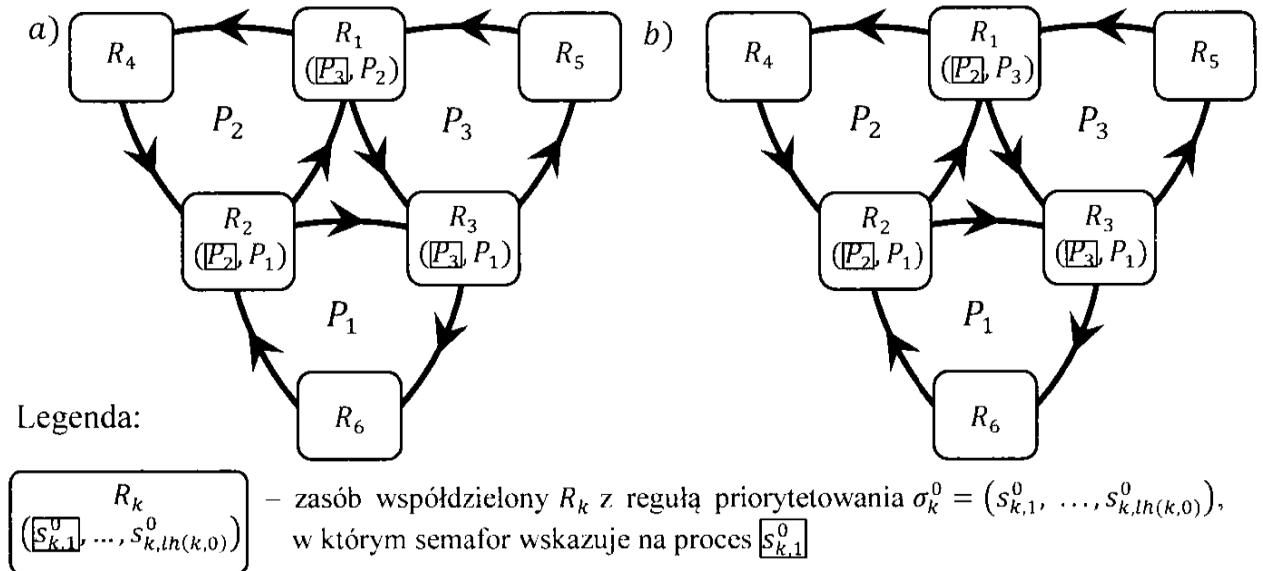
Należy zauważyć, że w problemach (89) i (93), stosuje się pewne uproszczenie polegające na tym, że operacje procesów realizowane są dokładnie w kolejności określonej przez reguły priorytetowania. Odpowiada to sytuacji, w której w stanie początkowym wszystkie semafony wskazują na pierwszy element związanych z nimi reguł priorytetowania. Tego typu ograniczenie może prowadzić do braku rozwiązania problemów (89) i (93) – nawet w przypadku istnienia poszukiwanego cyklicznego przebiegu ustalonego. Przykład tego typu systemów przedstawiono na rys. 3.26. Reguły priorytetowania zasobu  $R_1$ , przyjęte w strukturach systemów z rys. 3.26a i 3.26b, determinują ten sam porządek dostępu procesów. Oba systemy powinny więc charakteryzować się tym samym zachowaniem. W pierwszym przypadku rozwiązanie problemu (89) pozwala wyznaczyć przebieg cykliczny, w drugim natomiast prowadzi do sprzeczności ograniczeń. Wynika to z faktu, że w pierwszym przypadku, w stanie początkowym, dostęp do zasobu ma proces  $P_3$  (proces wskazany przez semafor), a w drugim proces  $P_2$ .

Oznacza to, że w celu określenia osiągalnych przebiegów cyklicznych wymagane jest uwzględnienie wszystkich możliwych przesunięć cyklicznych reguł priorytetowania. W rozważanym przypadku wymagane jest uwzględnienie reguł postaci:  $(P_3, P_2)$ ,  $(P_2, P_3)$  dla zasobu  $R_1$ ;  $(P_1, P_2)$ ,  $(P_2, P_1)$  dla zasobu  $R_2$  oraz  $(P_3, P_1)$ ,  $(P_1, P_3)$  dla zasobu  $R_3$ .

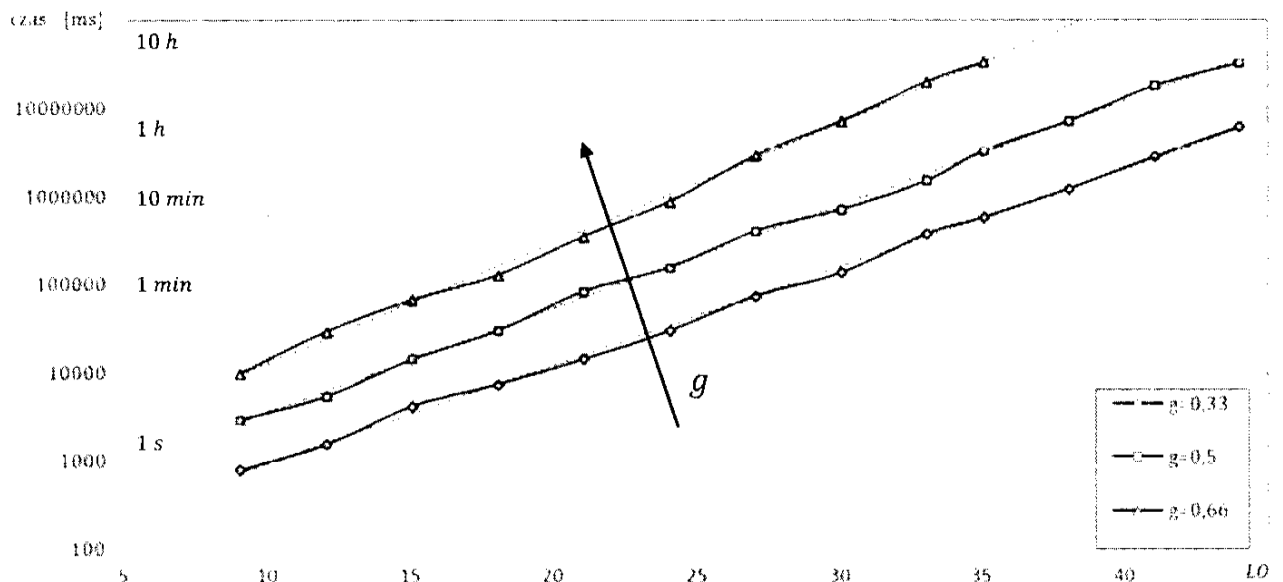
Funkcje czasów rozwiązania problemów (89) i (93) uwzględniających tego typu przesunięcia przedstawiono na rys. 3.27. Jak widać funkcje mają charakter wykładniczy. Skala zastosowań proponowanego algorytmu jest podobna jak w metodzie omawianej w poprzednim rozdziale. Czas rozwiązania nie przekraczający 60 minut jest

osiągalny w strukturach o gęstości  $g < 0,33$ , w których liczba  $LO$  nie przekracza 40 operacji.

Warto zauważyć, że w rozważanym podejściu występuje odwrotna tendencja niż ma to miejsce w metodzie omawianej w poprzednim rozdziale. Cechą charakterystyczną Algorytmów 1 i 2, jest to że ich efektywność rośnie wraz ze wzrostem gęstości analizowanych struktur, natomiast w przypadku Algorytmu 3 efektywność rośnie wraz z malejącą gęstością. Oznacza to, że proponowane metody wzajemnie się uzupełniają, tzn. pierwsza z nich jest dedykowana dla struktur „gęstych”, druga natomiast dla struktur „rzadkich”.



Rys. 3.26. SWMPC charakteryzujące się tym samym zachowaniem



Rys. 3.27. Funkcje czasów rozwiązania problemów typu (89) i (93) uwzględniające przesunięcia cykliczne reguł priorytetowania, dla struktur o gęstościach:  $g = 0,33$ ;  $g = 0,5$ ;  $g = 0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM)

Podsumowując wykorzystanie Algorytmu 3 ogranicza się do struktur rzadkich o gęstości  $g < 0,5$  należących do systemów typu  $MW_{1,1}$ ,  $MW_{2,1}$ ,  $MW_{3,1}$ ,  $MW_{4,1}$

w których sumaryczna liczba realizacji operacji nie przekracza 40 (czas rozwiązania poniżej 60 minut).

### 3.3. Metoda wyznaczania parametrów struktury systemów cyklicznych

Omawiane dotychczas metody odnosiły się do problemów analizy postaci (72), związanych z oceną istnienia przebiegów cyklicznych w systemach o znanych parametrach struktury  $SC^{lp}$ . Odrębną kwestią jest wyznaczanie parametrów struktury  $SC^{lp}$  gwarantujących osiągalność w systemie przebiegów cyklicznych. Zagadnienia tego typu związane są z problemami syntezy (omawianymi w podrozdziale 2.3), które w zależności od przyjętych założeń mogą być formułowane jako różne problemy spełniania ograniczeń (67), (71).

Jednym z wariantów problemu syntezy może być uszczegółowienie problemu (67) o ograniczania mówiące, że w systemie mają być osiągalne zachowania cykliczne. Problem ten (stosując notację przyjętą przy (67)) można sformułować następująco:

$$PS_{RE} = \left( (SC^{lp}, D_{SC^{lp}}), C_{PRE} \right), \quad (97)$$

gdzie zbiór zmiennych decyzyjnych obejmuje strukturę  $SC^{lp}$  a zbiór ograniczeń  $C_{PRE} = C_{SC} \cup C_{\mathcal{P}} \cup 'P$  zawiera przebiegi cykliczne'  $\cup C_{ST}$  zawiera:

$C_{SC}$  – ograniczenia struktury (założenia SWMPC patrz podrozdział 2.1.1 oraz tab. 2.1),

$C_{ST}$  – ograniczenia określające postać struktury  $SC^{lp}$  (18) zadanego SWMPC,

$C_{\mathcal{P}}$  – ograniczenia zachowania: warunki dopuszczalności stanów (22)–(31), warunki przejścia między stanami  $\Lambda^{lp}$  (51),

oraz ograniczenia dodatkowe ' $P$  zawiera przebiegi cykliczne' narzucające występowanie w przestrzeni  $\mathcal{P}$  cyklicznych przebiegów ustalonych (tj. wirów).

Problem (97) można więc interpretować jako problem, w którym dla częściowo określonej struktury (poprzez ograniczenia  $C_{ST}$ ) poszukiwane są jej pozostałe parametry  $SC^{lp}$  (reguły priorytetowania, czasy wykonania operacji, marszruty procesów, itp.) gwarantujące istnienie cyklicznego zachowania systemu. Innymi słowy poszukiwana jest odpowiedź na pytanie:

*Czy istnieje struktura  $SC^{lp}$  (opisana przez  $C_{SC}, C_{ST}$ ) gwarantująca istnienie w przestrzeni  $\mathcal{P}$  cyklicznych przebiegów ustalonych?*

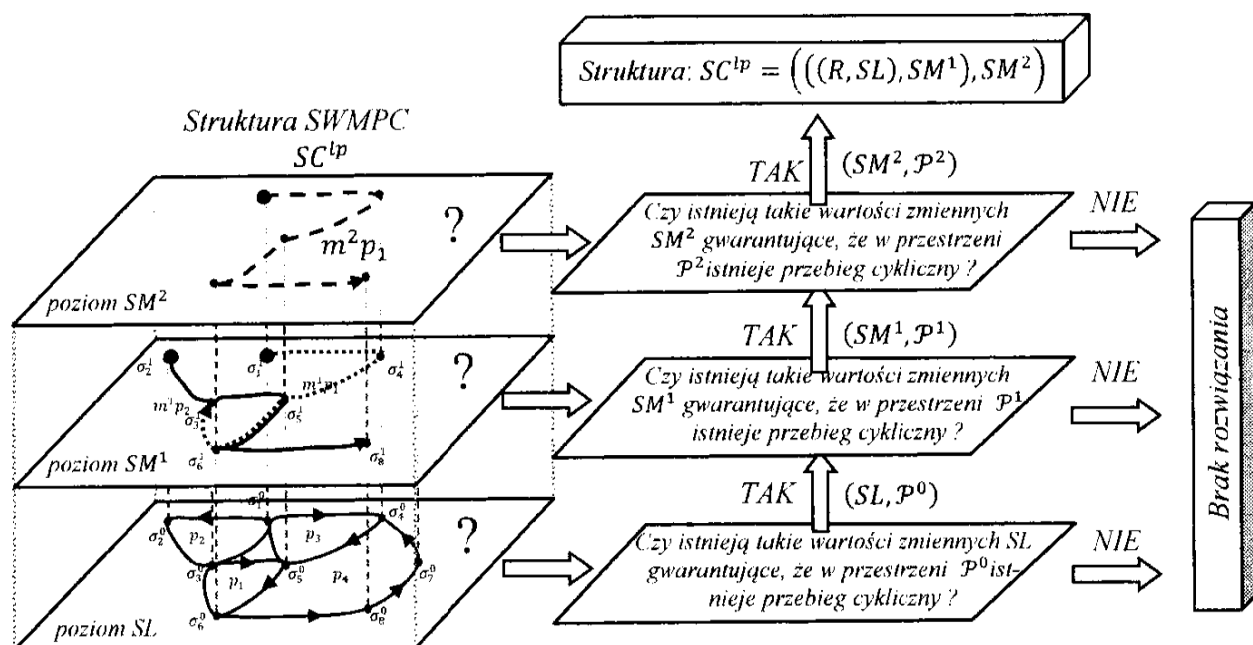
Podobnie jak w przypadku problemu (72), bezpośrednie rozwiązanie problemu (97) jest nieefektywne, gdyż sprowadza się do przeglądu pełnego potencjalnych wariantów struktur systemu. Jak już zauważono w podrozdziale 2.4 rozważane problemy syntezy są rozstrzygalne, nieznane są jednak algorytmy ich efektywnego (wielomianowego) rozwiązywania.

Wielopoziomowość struktury oraz współzależność przestrzeni przebiegów  $\mathcal{P}$  kolejnych poziomów zachowań (patrz podrozdział 2.2.3) pozwala przedstawić problem (97) jako kompozycję problemów spełniania ograniczeń dedykowanych dla pojedynczych poziomów zachowań. Oznacza to, że można wykorzystać sposób przyjęty w metodzie generowania harmonogramów cyklicznych  $X^{lp}$  prezentowany w po-

przednim podrozdziale (rys. 3.15 i 3.24). Graficzna ilustracja schematu wyznaczania struktury  $SC^{lp}$  została przedstawiona na rys. 3.28.

Podobnie jak poprzednio, istotą proponowanej metody jest analiza każdego z poziomów zachowań pod kątem istnienia cyklicznych przebiegów ustalonych. Jak przedstawiono to w podrozdziale 3.2.3 ocena tego typu odbywa się w oparciu o rozwiązanie odpowiednich problemów spełniania ograniczeń ( $PS'_X$  (89) i  $m^l PS'_X$  (93)), których zbiory ograniczeń determinowane są (odpowiedzialne są za to funkcje: CREATECONSTRAINTSL i CREATECONSTRAINTSM) przez zmienne charakteryzujące strukturę (reguły priorytetowania  $\Theta^0, \Theta^l$  marszruty procesów  $U, m^l U$  czasy wykonania operacji  $T, m^l T$ ).

Poszukiwanie struktury  $SC^{lp}$  (tzn. składających się na nią struktur poziomów zachowań  $SL, SM^l$  (18)), gwarantującej istnienie przebiegów cyklicznych, może więc prowadzić do poszukiwania parametrów determinujących ograniczenia  $C'_X, C^l_X$ , dla których istnieją dopuszczalne wartości zmiennych  $X'_{DC}, \alpha, m^l X'_{DC}, m^l \alpha$  (tzn. istnieją harmonogramy cykliczne).



**Rys. 3.28.** Sposób wyznaczania struktury  $SC^{lp}$  gwarantującej istnienie cyklicznych przebiegów ustalonych

Oznacza to, że każdemu wariantowi struktury  $SL$  (lub w przypadku procesów multimodalnych struktury  $SM^l$ ), odpowiada inny zestaw ograniczeń  $C'_X$  (lub  $C^l_X$ ). Nieśprzeczność ograniczeń otrzymanych dla określonej postaci  $SL$  (lub  $SM^l$ ) oznacza osiągalność w systemie zachowania cyklicznego. Problem spełniania ograniczeń pozwalający wyznaczać postać  $SL$  (lub  $SM^l$ ) powinien zatem uwzględniać zmieniający się, wraz ze zmianą  $SL, SM^l$ , charakter zbiorów ograniczeń  $C'_X$  (lub  $C^l_X$ ). Problemy tego typu można sformułować następująco:

- dla poziomu procesów lokalnych  $SL$  (odpowiednik problemu  $PS'_X$  (89)):

$$PS'_{REX} = ((V_{REX}, D_{REX}), C_{REX}), \quad (98)$$

gdzie:

$V_{REX} = X'_{DC} \cup \{\alpha, SL\}$  – zbiór zmiennych decyzyjnych:  $X'_{DC}$ ,  $\alpha$  – definiowane jak w (89),  $SL = (ST_L, BE_L)$  – struktura poziomu procesów lokalnych (zgodnie z (18)),

$D_{REX} = \{D_X, D_\alpha, D_{SL}\}$  – zbiór dziedzin zmiennych decyzyjnych:  $D_X$ ,  $D_\alpha$  – dziedziny definiowane jak w (89),  $D_{SL}$  – dziedzina określająca dopuszczalne wartości zmiennych charakteryzujących strukturę  $SL$ ,

$C_{REX} = C'_X(SL) \cup C_{STL}$  – zbiór ograniczeń:  $C'_X(SL)$  – zbiór ograniczeń określających związki między momentami  $x^h_{i,j}$  rozpoczęcia operacji (83) determinowane przez wartości  $SL$ . Postać zbioru  $C'_X(SL)$  wyznaczana jako wynik funkcji  $CREATECONSTRAINTSL: C'_X(SL) = CREATECONSTRAINTSL(SL)$ ,  $C_{STL}$  – ograniczenia określające postać struktury  $SL$  poziomu zachowań procesów lokalnych (18),

- dla poziomów procesów multimodalnych  $SM^l$  (odpowiednik problemu  $m^lPS'_X$  (93)):

$$m^lPS'_{REX} = \left( (m^lV_{REX}, D^l_{REX}), C^l_{REX} \right), \quad (99)$$

gdzie:

$m^lV_{REX} = m^lX'_{DC} \cup \{m^l\alpha, SM^l\}$  – zbiór zmiennych decyzyjnych:  $m^lX'_{DC}$ ,  $m^l\alpha$  – definiowane jak w (93),  $SM^l = (ST^l_M, BE^l_M)$  – struktura  $l$ -tego poziomu procesów multimodalnych (zgodnie z (18)),

$D^l_{REX} = \{D^l_X, D^l_\alpha, D^l_{SM}\}$  – zbiór dziedzin zmiennych decyzyjnych:  $D^l_{SM}$  – dziedzina określająca dopuszczalne wartości zmiennych charakteryzujących strukturę  $SM^l$ ,  $D^l_X$ ,  $D^l_\alpha$  – dziedziny definiowane jak w (93),

$C^l_{REX} = C^l_X(SM^l, m^{(l-1)}X'_{DC}) \cup C^l_{STM}$  – zbiór ograniczeń:  $C^l_X(SM^l, m^{(l-1)}X'_{DC})$  – ograniczenia określające związki między momentami  $m^l x^h_{i,j}$  rozpoczęcia operacji  $l$ -tego poziomu zachowań (92) jak i ich dopuszczalne wartości  $X^l_{i,j}$ . Postać zbioru  $C^l_X(SM^l, m^{(l-1)}X'_{DC})$  wyznaczana jako wynik funkcji  $CREATECONSTRAINTSM$ :

$$C^l_X(SM^l, m^{(l-1)}X'_{DC}) = CREATECONSTRAINTSM \left( (l-1), m^{(l-1)}X'_{DC}, SM^l \right),$$

$C^l_{STM}$  – ograniczenia określające postać struktury  $SM^l$   $l$ -tego poziomu zachowań (18).

Rozwiązaniem powyższych problemów są postacie struktur  $SL$ ,  $SM^l$  poziomów zachowań odpowiadających procesom lokalnym i multimodalnym składającym się na strukturę  $SC^{lp}$  (18), strukturę gwarantującą osiągalność w systemie cyklicznych przebiegów ustalonych.

W odróżnieniu od dotychczas formułowanych problemów spełniania ograniczeń, problemy (98), (99) wykorzystują tak zwane ograniczenia dynamiczne:  $C'_X(SL)$ ,  $C^l_X(SM^l, m^{(l-1)}X'_{DC})$  (Mądry 2006; Legierski i Parys 2004). Są to ograniczenia, których postać zmienia się w zależności od wartości zmiennych decyzyjnych. W rozważanych przypadkach ograniczenia zależą od wartości elementów sekwencji  $SL, SM^l$  charakteryzujących poszczególne poziomy zachowań (w szczególności od reguł priorytetowania  $\Theta^0, \Theta^l$  marszrut procesów  $U, m^lU$ ). Oznacza to, że każda zmiana

na reguł priorytetowania lub marszrut procesów skutkuje zmianą ograniczeń. Inaczej mówiąc, wraz z poszukiwaniem struktur  $SL$ ,  $SM^l$  poszukuje się niesprzecznej postaci zbiorów ograniczeń  $C'_X(SL)$ ,  $C'_X(SM^l, m^{(l-1)}X'_{DC})$ .

Rozwiązaniem problemów (98), (99) są więc, struktury  $SL$ ,  $SM^l$  determinujące niesprzeczne zbiory ograniczeń  $C'_X(SL)$ ,  $C'_X(SM^l, m^{(l-1)}X'_{DC})$ .

Przyjęto, że związek między strukturą (poziomami zachowań), a postacią ograniczeń jest określony w ten sam sposób jak miało to miejsce w przypadku metody prezentowanej w poprzednim podrozdziale (patrz CREATECONSTRAINTSL i CREATECONSTRAINTSM). Wykorzystanie tych samych funkcji opisujących związek między strukturą a ograniczeniami, jak i tego samego sposobu oceny istnienia przebiegów cyklicznych powoduje, że problemy  $PS'_X$  (89),  $m^l PS'_X$  (93) można traktować jako szczególny przypadek problemów (98), (99), w których postać struktury  $SC^{lp}$  systemu jest znana. Oznacza to, że problemy syntezy można traktować jako uogólnioną postać problemów typu analizy. Poniżej przedstawiono algorytm wyznaczania struktury  $SC^{lp}$  całego systemu, odpowiadający prezentowanej idei (rys. 3.28).

### Algorytm 6

```

function CYCLICSTRUCTUREPROTOTYPING( $D_X, D_\alpha, D_{SL}, \dots, D_{SM}^{lp}, C_{STL}, C_{STM}^1, \dots, C_{STM}^{lp}$ )
   $PS'_{REX} \leftarrow ((X'_{DC} \cup \{\alpha, SL\}, \{D_X, D_\alpha, D_{SL}\}), C'_X(SL) \cup C_{STL})$ 
   $(X'_{DC}, \alpha, SL) \leftarrow \text{SEARCHONE}(PS'_{REX})$ 
   $SC^{lp} \leftarrow (R, SL)$ 
   $m^0 X'_{DC} \leftarrow X'_{DC}$ 
  if  $SL \neq \emptyset$  and  $lp > 1$  then
    for  $l \leftarrow 1$  to  $lp$ 
       $m^l PS'_{REX} \leftarrow ((m^l X'_{DC} \cup \{m^l \alpha, SM^l\}, \{D_X^l, D_\alpha^l, D_{SM}^l\}), C'_X(SM^l, m^{(l-1)}X'_{DC}) \cup C_{STM}^l)$ 
       $(m^l X'_{DC}, m^l \alpha, SM^l) \leftarrow \text{SEARCHONE}(m^l PS'_{REX})$ 
      if  $SM^l \neq \emptyset$  then
         $SC^{lp} \leftarrow (SC^{lp}, SM^l)$ 
      else
         $SC^{lp} \leftarrow \emptyset$ 
      break
    end
  end
  return  $SC^{lp}$ 
end

```

gdzie:

$D_X, D_\alpha, D_{SL}, \dots, D_{SM}^{lp}, C_{STL}, C_{STM}^1, \dots, C_{STM}^{lp}$  – dane wejściowe definiowane jak w (89), (93), (99), (98),



$PS'_{REX}$  – problem spełniania ograniczeń (98),  
 $m^l PS'_{REX}$  – problem spełniania ograniczeń (99),  
 $SL$  – struktura poziomu zachowań procesów lokalnych,  
 $SM^l$  – struktura  $l$ -tego poziomu zachowań procesów multimodalnych,  
 $SEARCHONE(PS)$  – funkcja zwracająca pierwsze dopuszczalne rozwiązanie problemu  $PS$ , dla  $PS'_{REX}$  i  $m^l PS'_{REX}$  jako rozwiązanie zwracane są odpowiednio sekwencje  $(SL, X'_{DC}, \alpha)$  i  $(SM^l, m^l X'_{DC}, m^l \alpha)$  (w przypadku braku rozwiązania, każdy element sekwencji jest zbiorem pustym).

Zasada działania Algorytmu 6 jest analogiczna do działania Algorytmu 3. W pierwszej kolejności wyznaczana jest struktura procesów lokalnych  $SL$ . Jeżeli istnieją takie wartości reguł priorytetowania, marszrut, itp., które gwarantują istnienie cyklicznych przebiegów ustalonych ( $PS'_{REX}$  ma przynajmniej jedno rozwiązanie dopuszczalne) to przeprowadzana jest analiza kolejnego poziomu zachowań  $SM^1$ . Postępowanie takie jest powtarzane do momentu gdy osiągnięty zostanie ostatni ( $lp$ ) poziom zachowań lub też do momentu, gdy dla jednego z poziomów zachowań nie udaje się wyznaczyć dopuszczalnego (gwarantującego niesprzeczność ograniczeń  $C'_x$  i  $C'_y$ ) zestawu parametrów struktury.

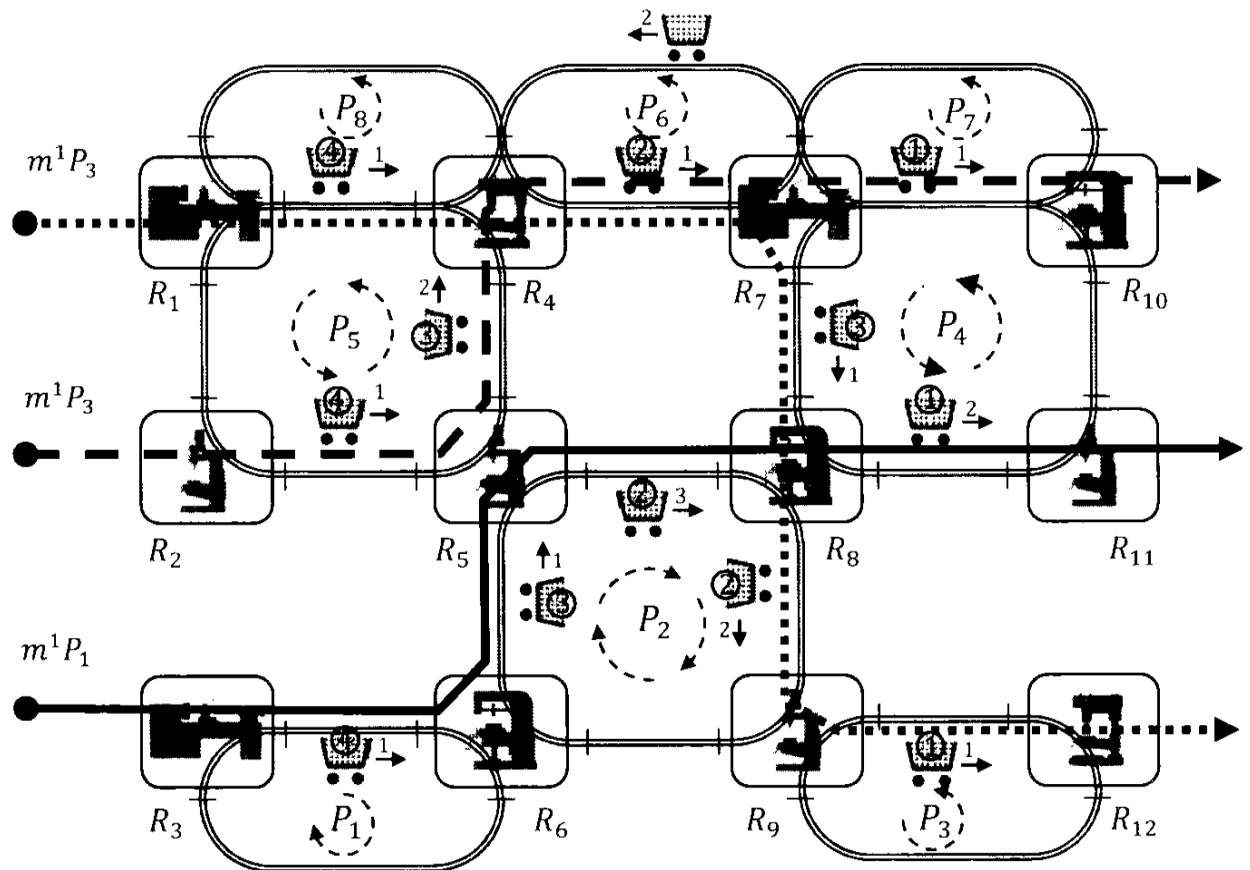
Proponowany algorytm wykorzystywany jest przy poszukiwaniu struktury SWMPC przy czym zakłada jej częściową bądź też całkowitą nieznaną. W rozważanej wersji implementuje on strategię bottom-up, tzn. proces wyznaczania struktury rozpoczyna się od najniższego poziomu zachowań i kończy na najwyższym. Jego alternatywę implementującą strategię top-down przedstawia poniższy przykład.

**Przykład 6.** Ilustracja rozwiązania problemu syntezy w oparciu o strategię top-down.

Dany jest system (patrz rys. 3.29) wielopotokowej produkcji 3 rodzajów wyrobów zgodnie z zadanymi marszrutami produkcyjnymi (marszruty oznaczone są liniami koloru: zielonego, pomarańczowego i niebieskiego). Produkcja jednego rodzaju wyrobu jest reprezentowana przez proces multimodalny  $m^1 P_i$  ( $i = 1, 2, 3$ ). Poszczególne wyroby oznaczane jako  $\bullet$ , są reprezentowane przez potoki procesów multimodalnych  $m^1 P_i^q$ . W systemie wykorzystywanych jest 13 wózków samojezdnych obsługujących transport wyrobów między stanowiskami. Operacje transportu wyrobów reprezentowane są przez potoki procesów lokalnych  $P_i^h$ .

Rozważany system, należy do klasy SWMPC (patrz rys. 3.30) typu  $MW_{3,1}$  (tab. 2.1). Oznacza to, że:

- dopuszcza się nie-jednostkowe czasy wykonania operacji (zarówno procesów lokalnych jak i multimodalnych),  $t_{i,j} \geq 1$ ,  $m^l t_{i,j} \geq 1$  (operacje obróbki/transportu mogą trwać dłużej niż 1 u.j.c.),
- procesy multimodalne synchronizowane są w oparciu o protokół wzajemnego wykluczania, (tylko jeden wyrób może być na stanowisku),
- pojemności zasobów są jednostkowe:  $C(R_k) = 1$ ,  $k = 1 \dots 33$ ,
- zapotrzebowanie procesów lokalnych i multimodalnych na wykorzystywane przez siebie zasoby jest jednostkowe:  $\omega_{i,j}^h = 1$ ,  $m^l \omega_{i,j}^h = 1$ , (tzn. w trakcie wykonywania operacji zajmowany jest cały zasób).



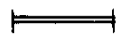
Legend:



$i$ -te stanowisko/maszyna,  
zasób  $R_i$



marszruty produkcyjne,  
procesy multimodalne,  
 $m^1P_i$



– sektor transportowy

$P_i$   $h$  – wózek samojezdny modelowany jako  $h$ -ty potok procesu lokalnego  $P_i$



$q$  – wyrób modelowany jako  $q$ -ty potok procesu multimodalnego  $m^1P_i$



$q$ -ty wyrób transportowany przez  $j$ -ty wózek ( $q$ -ty potok procesu multimodalnego, wykorzystujący  $h$ -ty potok procesu lokalnego)

Rys. 3.29. System produkcji potokowej z podsystemem transportowym AGV

Struktura SWMPC z rys. 3.30 (zgodnie z (18)) jest następująca:

$$SC^1 = ((R, SL), SM^1),$$

gdzie  $R = \{R_1, \dots, R_{33}\}$  to zbiór 33 zasobów, każdy o pojemności  $C(R_k) = 1$ ,  $R_k \in R$ . Zasoby  $R_1 - R_{12}$  reprezentują stanowiska pracy, ( $R_1 - R_3$  są zasobami wejściowymi, a zasoby  $R_{10} - R_{12}$  zasobami wyjściowymi), zasoby  $R_{13} - R_{33}$  reprezentują sektory transportowe, wzdłuż których przemieszczają się wózki.

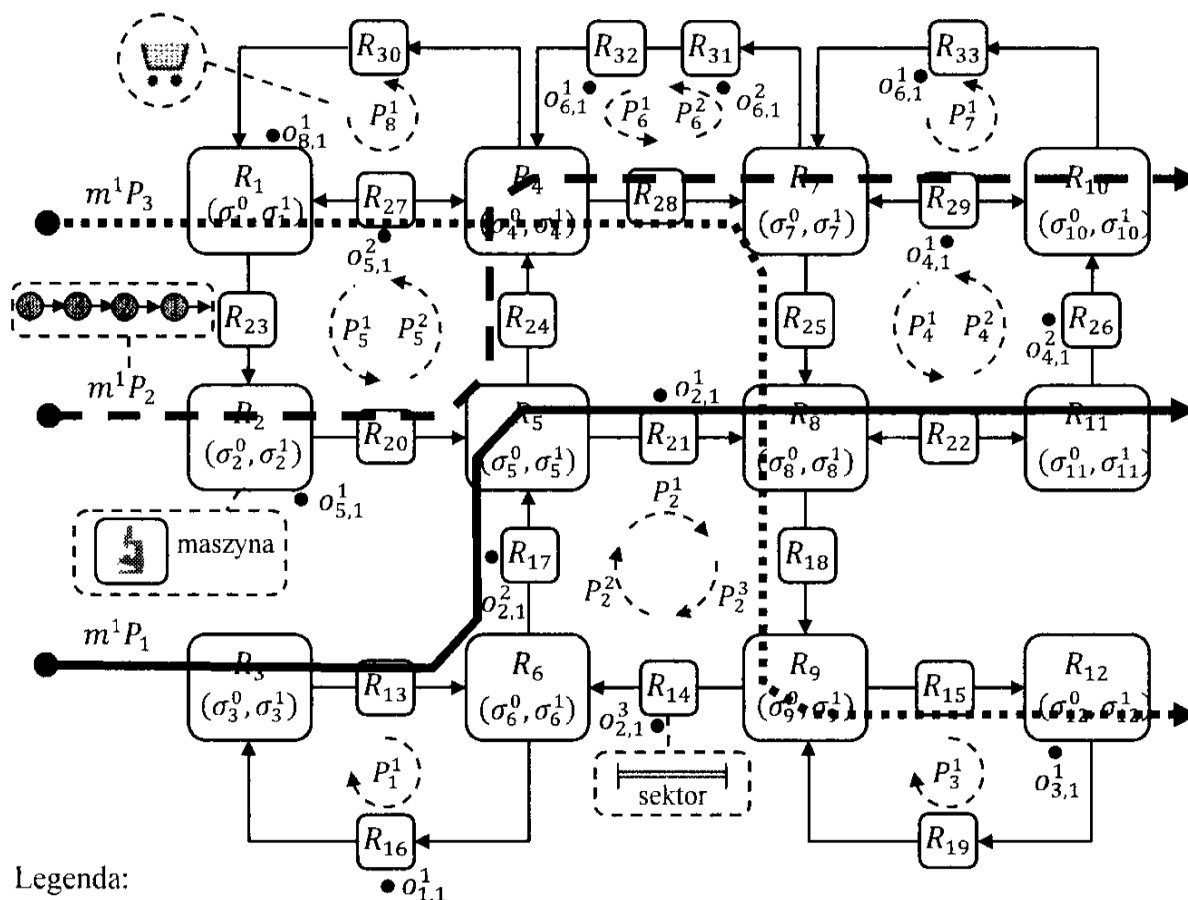
Struktura  $SC^1$  zawiera dwa poziomy zachowań (rys. 3.31): poziom procesów lokalnych  $SL$  (warstwa transportu między-stanowiskowego) oraz poziom procesów multimodalnych  $SM^1$  (warstwa produkcji wielowersyjnej).

Poziom procesów lokalnych  $SL$  charakteryzowany jest przez zbiór 8 procesów:

$$P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$$

$$P_1 = \{P_1^1\}, P_2 = \{P_2^1, P_2^2, P_2^3\}, P_3 = \{P_3^1\}, P_4 = \{P_4^1, P_4^2\}, P_5 = \{P_5^1, P_5^2\}, P_6 = \{P_6^1, P_6^2\}, P_7 = \{P_7^1\}, P_8 = \{P_8^1\},$$

z czego procesy:  $P_4, P_5, P_6$  są procesami dwupotokowymi (tzn. zakłada się, że wzdłuż jednej marszruty poruszają się dwa wózki) a proces  $P_2$  jest procesem trójpotokowym (tzn. wzdłuż jednej marszruty poruszają się trzy wózki).



Legenda:

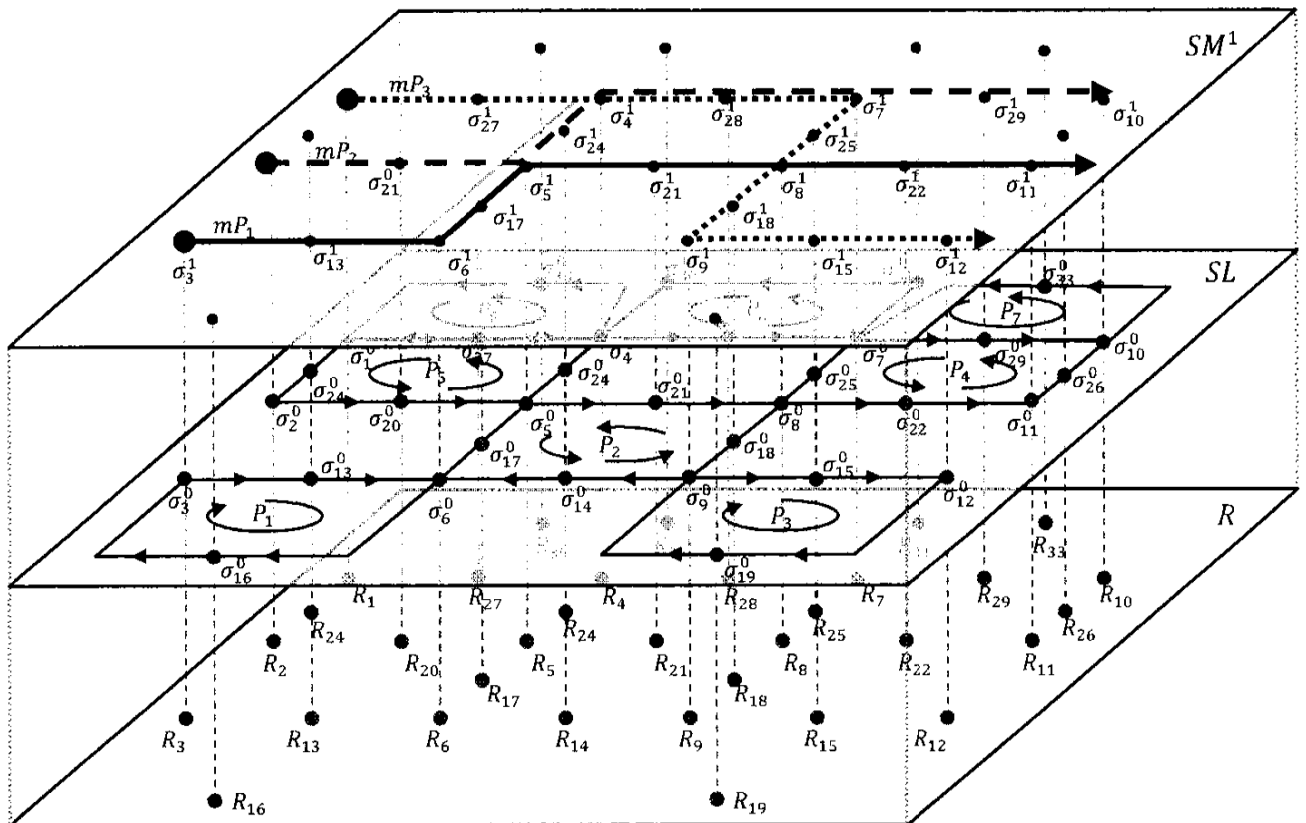
- $R_k$  – zasób współdzielony  $R_k$  z regułami priorytetowania  $\sigma_k^0 = (s_{k,1}^0, \dots, s_{k,lh(k,0)}^0)$ ,  
 $\sigma_k^1 = (s_{k,1}^1, \dots, s_{k,lh(k,1)}^1)$
- $R_k$  – zasób  $R_k$  reprezentujący sektor transportowy
- $P_i^h$  – potok  $P_i^h$  procesu lokalnego
- $m^1P_i$  – procesy multimodalne  $m^1P_i$
- $\dots$  – kolejne potoki (wyroby) procesów multimodalnych
- $o_{i,1}^h$  – pierwsza operacja potoku  $P_i^h$

Rys. 3.30. SWMPC przedstawiający system z rys. 3.29

Procesom odpowiadają następujące marszruty:

$$\begin{aligned}
 p_1 &= (R_{16}, R_3, R_{13}, R_6), & p_2 &= (R_{21}, R_8, R_{18}, R_9, R_{14}, R_6, R_{17}, R_5), \\
 p_3 &= (R_{12}, R_{19}, R_9, R_{15}), & p_4 &= (R_7, R_{25}, R_8, R_{22}, R_{11}, R_{26}, R_{10}, R_{29}), \\
 p_5 &= (R_2, R_{20}, R_5, R_{24}, R_4, R_{27}, R_1, R_{23}), & p_6 &= (R_{32}, R_4, R_{28}, R_7, R_{31}), \\
 p_7 &= (R_{33}, R_7, R_{29}, R_{10}), & p_8 &= (R_1, R_{27}, R_4, R_{30}).
 \end{aligned}$$

Operacje potoków wykonywanych wzdłuż tak określonych marszrut oznaczane są przez:  $O_i^h = (o_{i,1}^h, \dots, o_{i,lr(i)}^h)$ ,  $i = 1 \dots 8$ ,  $h = 1 \dots ls(i)$ , gdzie każdej operacji  $o_{i,j}^h$  odpowiada moment jej rozpoczęcia  $x_{i,j}^h$  oraz czas trwania  $t_{i,j}^h$  (dopuszcza się, że operacje z różnych potoków tego samego procesu mogą mieć różne czasy wykonania).



Rys. 3.31. Poziomy zachowań SWMPC systemu z rys. 3.30

Operacje początkowe  $\sigma_{i,1}^h$  potoków  $P_i^h$  oznaczone zostały na rys. 3.30 symbolem  $\bullet$ . Przyjęto, że operacje początkowe  $\sigma_{i,1}^h$  potoków tego samego procesu mogą być realizowane na różnych zasobach. Przykładowo operacje początkowe potoków procesu  $P_2$  są wykonywane na zasobach  $R_{21}$ ,  $R_{17}$ ,  $R_{14}$ .

Dostęp potoków (wózków) do zasobów jest określony przy pomocy reguł priorytetowania  $\theta^0 = \{\sigma_1^0, \dots, \sigma_{33}^0\}$  przedstawionych w tab. 3.8. Reguły  $\theta^0$  zostały tak dobrane by każdy proces był wykonywany dokładnie raz na jedno wykonanie każdego z pozostałych procesów:  $\psi^0 = (1,1,1,1,1,1,1)$ .

Tabela 3.8. Reguły priorytetowania procesów poziomego SL systemu z rys. 3.30

Reguły priorytetowania $\theta^0$		
$\sigma_1^0 = (P_5^2, P_8^1, P_5^1)$	$\sigma_{12}^0 = (P_3^1)$	$\sigma_{23}^0 = (P_5^2, P_5^1)$
$\sigma_2^0 = (P_5^1, P_5^2)$	$\sigma_{13}^0 = (P_1^1)$	$\sigma_{24}^0 = (P_5^1, P_5^2)$
$\sigma_3^0 = (P_1^1)$	$\sigma_{14}^0 = (P_2^3, P_2^1, P_2^2)$	$\sigma_{25}^0 = (P_4^1, P_4^2)$
$\sigma_4^0 = (P_6^1, P_8^1, P_6^2, P_5^1, P_5^2)$	$\sigma_{15}^0 = (P_3^1)$	$\sigma_{26}^0 = (P_4^2, P_4^1)$
$\sigma_5^0 = (P_2^2, P_2^3, P_5^1, P_5^2, P_2^1)$	$\sigma_{16}^0 = (P_1^1)$	$\sigma_{27}^0 = (P_5^2, P_8^1, P_5^1)$
$\sigma_6^0 = (P_2^3, P_2^1, P_1^1, P_2^2)$	$\sigma_{17}^0 = (P_2^2, P_2^3, P_2^1)$	$\sigma_{28}^0 = (P_6^1, P_6^2)$
$\sigma_7^0 = (P_4^1, P_4^2, P_1^1, P_1^2, P_6^2)$	$\sigma_{18}^0 = (P_2^1, P_2^2, P_2^3)$	$\sigma_{29}^0 = (P_4^1, P_4^2, P_1^1)$
$\sigma_8^0 = (P_2^1, P_4^1, P_2^2, P_2^3, P_4^2)$	$\sigma_{19}^0 = (P_3^1)$	$\sigma_{30}^0 = (P_8^1)$
$\sigma_9^0 = (P_2^1, P_2^2, P_1^1, P_2^3)$	$\sigma_{20}^0 = (P_5^1, P_5^2)$	$\sigma_{31}^0 = (P_6^2, P_6^1)$
$\sigma_{10}^0 = (P_4^2, P_7^1, P_4^1)$	$\sigma_{21}^0 = (P_2^1, P_2^2, P_2^3)$	$\sigma_{28}^0 = (P_6^1, P_6^2)$
$\sigma_{11}^0 = (P_4^1, P_4^2)$	$\sigma_{22}^0 = (P_4^1, P_4^2)$	$\sigma_{30}^0 = (P_7^1)$

Poziom  $SM^1$  (rys. 3.31) charakteryzowany jest z kolei przez zbiór trzech czteropotokowych procesów multimodalnych:

$$m^1P = \{m^1P_1, m^1P_2, m^1P_3\},$$

$$\text{gdzie: } m^1P_i = \{m^1P_i^1, m^1P_i^2, m^1P_i^3, m^1P_i^4\}, i = 1, 2, 3.$$

Założenie wielopotokowości procesów multimodalnych wynika z faktu, że w danym momencie w systemie może być realizowana jednoczesna produkcja wielu wyrobów jednego rodzaju.

Każdy potok  $m^1P_i^h$  procesów multimodalnych jest wykonywany wzdłuż jednej z następujących marszrut:

$$m^1p_1 = ((R_3, R_{13}, R_6) \wedge (R_{17}, R_5) \wedge (R_{21}, R_8) \wedge (R_{22}, R_{11})) = \\ = (R_3, R_{13}, R_6, R_{17}, R_5, R_{21}, R_8, R_{22}, R_{11}),$$

$$m^1p_2 = ((R_2, R_{20}, R_5) \wedge (R_{24}, R_4) \wedge (R_{28}, R_7) \wedge (R_{29}, R_{10})) = \\ = (R_2, R_{20}, R_5, R_{24}, R_4, R_{28}, R_7, R_{29}, R_{10}),$$

$$m^1p_3 = ((R_1, R_{27}, R_4) \wedge (R_{28}, R_7) \wedge (R_{25}, R_8) \wedge (R_{18}, R_9), (R_{15}, R_{12})) = \\ = (R_1, R_{27}, R_4, R_{28}, R_7, R_{25}, R_8, R_{18}, R_9, R_{15}, R_{12}),$$

gdzie:

$(R_3, R_{13}, R_6), (R_{17}, R_5), (R_{21}, R_8), (R_{22}, R_{11})$  – fragmenty marszrut potoków procesów lokalnych  $P_1^1, P_2^1, P_2^3$ , i  $P_4^2$  (wybranych arbitralnie wózków) kolejno wykorzystywanych przy realizacji potoków (transporte wyrobów) procesu  $m^1P_1$  (oznaczonego na rys. 3.29 linią ciągłą),

$(R_2, R_{20}, R_5), (R_{24}, R_4), (R_{28}, R_7), (R_{29}, R_{10})$  – fragmenty marszrut potoków procesów lokalnych  $P_5^1, P_5^2, P_6^1$  i  $P_7^1$  (wózków) kolejno wykorzystywanych przy realizacji potoków (transporte wyrobów) procesu  $m^1P_2$  (oznaczonego na rys. 3.29 linią przerywaną),

$(R_1, R_{27}, R_4), (R_{28}, R_7), (R_{25}, R_8), (R_{18}, R_9), (R_{15}, R_{12})$  – fragmenty marszrut potoków procesów lokalnych  $P_8^1, P_6^2, P_4^1, P_2^2$  (wózków) kolejno wykorzystywanych przy realizacji potoków (transporte wyrobów) procesu  $m^1P_3$  (oznaczonego na rys. 3.29 linią kropkowaną).

Operacje potoków wykonywanych wzdłuż tak określonych marszrut oznaczane są przez:  $m^1O_i^h = (m^1o_{i,1}^h, \dots, m^1o_{i,lm(i,1)}^h)$ ,  $i = 1 \dots 3$ ,  $h = 1 \dots 4$ , gdzie każdej operacji  $m^1o_{i,j}^h$  odpowiada moment jej rozpoczęcia  $x_{i,j}^h$ .

Zbiór reguł priorytetowania  $\theta^1$  dla rozważanych procesów multimodalnych (poziom  $SM^1$ ) przedstawia tab. 3.9.

W rozważanym systemie przyjmuje się, że znane są niektóre z elementów jego struktury, tzn. marszruty procesów lokalnych (wózków) i multimodalnych (produkowanych wyrobów) (rys. 3.31), reguły priorytetowania (tab. 3.8 i 3.9) oraz czasy wykonania operacji procesów multimodalnych (tab. 3.10). Nieznane są natomiast czasy wykonania operacji procesów lokalnych  $T$ . Innymi słowy nie wiadomo jaki jest czas przebywania wózków na określonych stanowiskach, jak i nieznany jest czas ich przejazdów wzdłuż sektorów transportowych.

**Tabela 3.9.** Reguły priorytetowania procesów poziomu  $SM^1$  systemu z rys. 3.30

Reguły priorytetowania $\theta^1$	
$\sigma_1^1 = (mP_3^1, mP_3^2, mP_3^3, mP_3^4)$	$\sigma_{13}^1 = (mP_1^1, mP_1^2, mP_1^3, mP_1^4)$
$\sigma_2^1 = (mP_2^1, mP_2^2, mP_2^3, mP_2^4)$	$\sigma_{15}^1 = (mP_3^3, mP_3^4, mP_3^1, mP_3^2)$
$\sigma_3^1 = (mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	$\sigma_{17}^1 = (mP_1^1, mP_1^2, mP_1^3, mP_1^4)$
$\sigma_4^1 = (mP_3^4, mP_2^4, mP_3^1, mP_2^1, mP_3^2, mP_2^2, mP_3^3, mP_2^3)$	$\sigma_{18}^1 = (mP_3^3, mP_3^4, mP_3^1, mP_3^2)$
$\sigma_5^1 = (mP_1^3, mP_2^4, mP_1^4, mP_2^1, mP_1^1, mP_2^2, mP_1^2, mP_2^3)$	$\sigma_{20}^1 = (mP_2^1, mP_2^2, mP_2^3, mP_2^4)$
$\sigma_6^1 = (mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	$\sigma_{21}^1 = (mP_1^3, mP_1^4, mP_1^1, mP_1^2)$
$\sigma_7^1 = (mP_2^3, mP_3^4, mP_2^4, mP_3^1, mP_2^1, mP_3^2, mP_2^2, mP_3^3)$	$\sigma_{22}^1 = (mP_1^3, mP_1^4, mP_1^1, mP_1^2)$
$\sigma_8^1 = (mP_3^3, mP_1^3, mP_3^4, mP_1^4, mP_3^1, mP_1^1, mP_3^2, mP_1^2)$	$\sigma_{24}^1 = (mP_2^1, mP_2^2, mP_2^3, mP_2^4)$
$\sigma_9^1 = (mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	$\sigma_{25}^1 = (mP_3^3, mP_3^4, mP_3^1, mP_3^2)$
$\sigma_{10}^1 = (mP_2^3, mP_2^4, mP_2^1, mP_2^2)$	$\sigma_{27}^1 = (mP_3^1, mP_3^2, mP_3^3, mP_3^4)$
$\sigma_{11}^1 = (mP_1^3, mP_1^4, mP_1^1, mP_1^2)$	$\sigma_{28}^1 = (mP_3^4, mP_2^4, mP_3^1, mP_2^1, mP_3^2, mP_2^2, mP_3^3, mP_2^3)$
$\sigma_{12}^1 = (mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	$\sigma_{29}^1 = (mP_2^3, mP_2^4, mP_2^1, mP_2^2)$

Oprócz wielkości charakteryzujących strukturę znany jest również sposób realizacji marszrut produkcyjnych (procesów multimodalnych). Realizacja marszrut jest określona z góry i wynika z założeń przyjętego planu produkcyjnego zadanego harmonogramem cyklicznym przedstawionym na rys. 3.32.

Harmonogram (będący graficzną ilustracją  $m^1 X'_{DC}$ ) przedstawia realizację operacji (obróbki wyrobów na zasobach  $R_1-R_{12}$ ) czterech potoków  $mP_i^1, mP_i^2, mP_i^3, mP_i^4$ , każdego z procesów multimodalnych ( $i = 1, 2, 3$ ). Realizacja operacji odbywa się zgodnie z przyjętymi regułami (tab. 3.9), procesy pracują z okresem:  $m^1 \alpha = 320$  u.j.c., gdzie każdy wyrób opuszcza system w odstępach wynoszących 80 u.j.c. (czasy wykonania operacji przedstawia tab. 3.10).

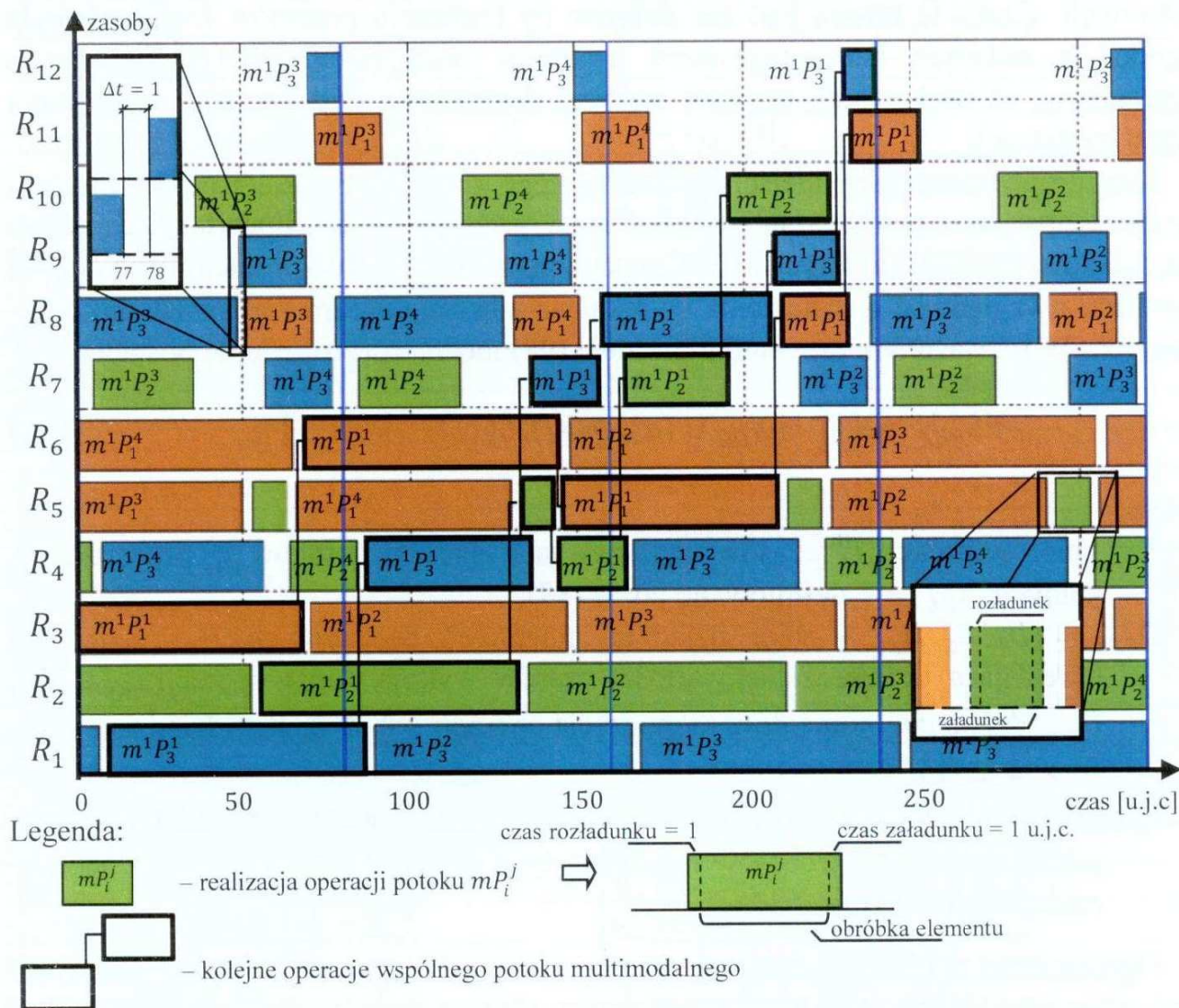
W kontekście tak zadanego, oczekiwanego zachowania systemu poszukiwana jest odpowiedź na następujące pytanie:

*Czy w systemie z rys. 3.29 możliwa jest taka organizacja pracy wózków transportowych, która gwarantuje produkcję wyrobów zgodnie z przyjętym planem produkcyjnym?*

Przez organizację pracy wózków rozumie się w tym przypadku taki dobór czasów wykonywania poszczególnych operacji (zbiór  $T$ ) procesów lokalnych, który gwarantuje transport elementów między stanowiskami zapewniający zadaną, terminową realizację planu produkcyjnego.

Przyjęty harmonogram uwzględnia zadania związane z transportem oraz załadunkiem/rozładunkiem elementów. Przyjęto, że transport elementu między stanowiskami wymaga 1 jednostki czasu ( $\Delta t = 1$ ), a zadania załadunku/rozładunku obejmują kolejno pierwszą i ostatnią jednostkę przedziału czasu wykonania operacji (rys. 3.32). W praktyce oznacza to, że na stanowisku, na którym realizowana jest obróbka elementu wymagana jest obecność odpowiedniego wózka w momencie rozpoczynania

operacji (rozładunek na stanowisko) jak i jej ukończenia (załadunek na wózek). Wyjątek stanowią stanowiska wejściowe ( $R_1, R_2, R_3$ ) i wyjściowe ( $R_{10}, R_{11}, R_{12}$ ), w których pobranie/odbiór elementów odbywa się w sposób automatyczny (bez udziału wózków transportowych).



Rys. 3.32. Harmonogram cykliczny procesów multimodalnych reprezentujący przyjęty plan produkcyjny

Tabela 3.10. Czasy wykonania operacji procesów multimodalnych z rys. 3.32

$k = 1 \dots 4$	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$	$R_7$	$R_8$	$R_9$	$R_{10}$	$R_{11}$	$R_{12}$	pozostałe wykorzystywane zasoby
$mt_1^k$	-	-	77	-	64	77	-	20	-	-	20	-	1
$mt_2^k$	-	77	-	20	10	-	30	-	-	30	-	-	1
$mt_3^k$	77	-	-	49	-	-	20	50	20	-	-	10	1

Odpowiedź na postawione pytanie sprowadza się do rozwiązania problemu syntezy (97) a w szczególności do problemów spełniania ograniczeń typu:  $PS'_{REX}$  (98) i  $m^1PS'_{REX}$  (99). W odróżnieniu od podejścia proponowanego na rys. 3.28, w rozważanym przypadku przyjmuje się, że znane są parametry charakteryzujące poziom wyższy. Znana jest zarówno struktura  $SM^1$  procesów multimodalnych jak i ich za-

chowanie  $(m^1 X'_{DC}, m^1 \alpha)$  (reprezentowane przez harmonogram cykliczny). Innymi słowy znane jest rozwiązanie problemu  $m^1 PS'_{REX}$  (99).

Oznacza to, że poszukiwane rozwiązanie sprowadza się do określenia postaci warstwy niższej, a ściślej mówiąc zmiennych czasów wykonywania operacji  $T$  procesów lokalnych. Czasy te muszą być tak dobrane by realizacja procesów lokalnych była zgodna z zadanym harmonogramem procesów multimodalnych  $(m^1 X'_{DC}, m^1 \alpha)$ . Oznacza to, że zachowanie warstwy wyższej determinuje dopuszczalne zachowania warstwy niższej.

Ideę proponowanego podejścia poszukującego rozwiązań postaci poziomów niższych w oparciu o znane parametry poziomów wyższych, ideę odpowiadającą strategii top-down, przedstawia rys. 3.33. Oznacza to, że rozwiązanie problemu  $PS'_{REX}$  (98) powinno uwzględniać ograniczenia wynikające z harmonogramu procesów multimodalnych. W rozważanym przypadku problem (98) można sprowadzić do postaci:

$$PS'_{REX} = \left( (T \cup X'_{DC} \cup \{\alpha\}, D_{REX}), C'_X(SM^1, m^1 X'_{DC}) \right), \quad (100)$$

gdzie:

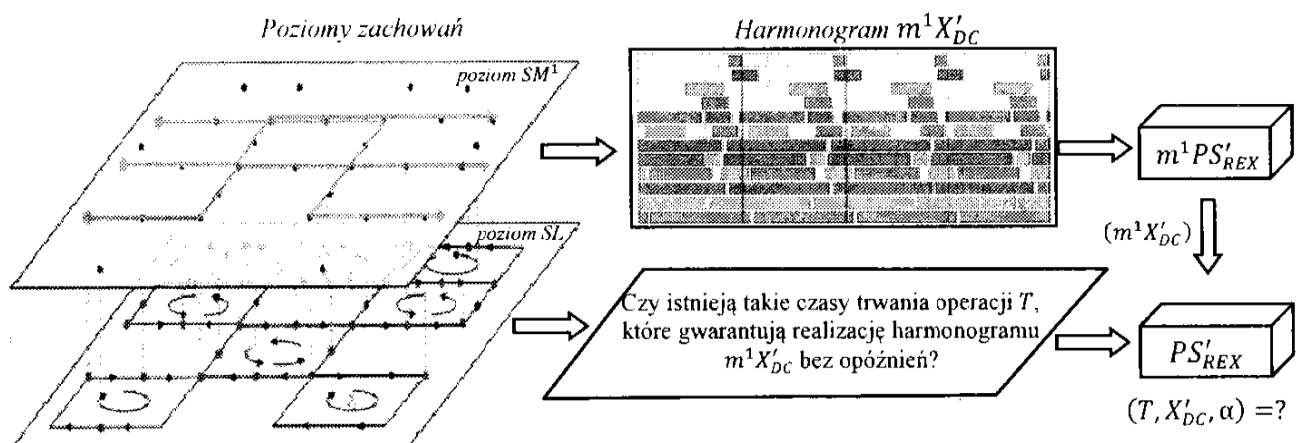
$T$  – zbiór sekwencji  $T_i^h$  czasów wykonywania operacji potoków  $P_i^h$  procesów lokalnych,  $X'_{DC}$ ,  $\alpha$  – definiowane jak w (89),

$D_{REX} = \{D_T, D_X, D_\alpha\}$  – zbiór dziedzin zmiennych decyzyjnych:  $D_T$  – dziedzina określająca dopuszczalne wartości czasów wykonywania operacji:  $t_{i,j} \in \mathbb{N}^+$ ,

$D_X$  – dziedzina momentów rozpoczęcia operacji:  $x_{i,j}^h \in \mathbb{C}$ ,  $D_\alpha$  – dziedzina okresu  $\alpha$ :  $\alpha \in \mathbb{N}^+$ ,

$C'_X(SM^1, m^1 X'_{DC})$  – ograniczenia określające związki między momentami  $x_{i,j}^h$  rozpoczęcia operacji procesów lokalnych typu (83) (tab. 3.11) jak i ich dopuszczalne wartości (tab. 3.12).

Ograniczenia z tab. 3.11 odpowiadają ograniczeniom  $C'_X$  z (89) i wyznaczone zostały przy użyciu algorytmu  $CREATECONSTRAINTSL(SL)$  (w oparciu o marszruty procesów lokalnych  $p_i$ , multimodalnych  $m^1 p_i$  oraz przyjęte reguły priorytetowania  $\theta^0$  i  $\theta^1$ ) Ograniczenia z tab. 3.12 wyznaczane są w analogiczny jak w przypadku (92).



Rys. 3.33. Wyznaczanie parametrów struktury  $SC^1$  systemu z rys. 3.29 w strategii top-down



Tabela 3.11. Ograniczenia określające zachowanie procesów lokalnych SWMPC z rys. 3.29

Ograniczenia $C'_x(SM^1, m^1 X'_{DC})$	
$x_{1,1}^1 = x_{1,4}^1 + t_{1,4}^1 - \alpha$	$x_{5,6}^1 = \max\{(x_{8,3}^1 + \Delta t), (x_{5,5}^1 + t_{5,5}^1)\}$
$x_{1,2}^1 = x_{1,1}^1 + t_{1,1}^1$	$x_{5,7}^1 = \max\{(x_{8,2}^1 + \Delta t), (x_{5,6}^1 + t_{5,6}^1)\}$
$x_{1,3}^1 = x_{1,2}^1 + t_{1,2}^1$	$x_{5,8}^1 = \max\{(x_{5,4}^1 + \Delta t), (x_{5,7}^1 + t_{5,7}^1)\}$
$x_{1,4}^1 = \max\{(x_{2,7}^1 + \Delta t), (x_{1,3}^1 + t_{1,3}^1)\}$	-----
-----	$x_{3,1}^1 = x_{3,4}^1 + t_{3,4}^1 - \alpha$
$x_{2,1}^1 = \max\{(x_{2,6}^1 + \Delta t - \alpha), (x_{2,8}^1 + t_{2,8}^1 - \alpha)\}$	$x_{3,2}^1 = x_{3,1}^1 + t_{3,1}^1$
$x_{2,2}^1 = \max\{(x_{4,7}^1 + \Delta t - \alpha), (x_{2,1}^1 + t_{2,1}^1)\}$	$x_{3,3}^1 = \max\{(x_{2,7}^1 + \Delta t), (x_{3,2}^1 + t_{3,2}^1)\}$
$x_{2,3}^1 = \max\{(x_{3,8}^1 + \Delta t - \alpha), (x_{2,2}^1 + t_{2,2}^1)\}$	$x_{3,4}^1 = x_{3,3}^1 + t_{3,3}^1$
$x_{2,4}^1 = \max\{(x_{2,1}^1 + \Delta t), (x_{2,3}^1 + t_{2,3}^1)\}$	-----
$x_{2,5}^1 = \max\{(x_{3,2}^1 + \Delta t), (x_{2,4}^1 + t_{2,4}^1)\}$	$x_{4,1}^1 = \max\{(x_{7,4}^1 + \Delta t - \alpha), (x_{4,8}^1 + t_{4,8}^1 - \alpha)\}$
$x_{2,6}^1 = \max\{(x_{2,3}^1 + \Delta t), (x_{2,5}^1 + t_{2,5}^1)\}$	$x_{4,2}^1 = \max\{(x_{6,1}^1 + \Delta t), (x_{4,1}^1 + t_{4,1}^1)\}$
$x_{2,7}^1 = \max\{(x_{3,4}^1 + \Delta t), (x_{2,6}^1 + t_{2,6}^1)\}$	$x_{4,3}^1 = \max\{(x_{4,6}^1 + \Delta t - \alpha), (x_{4,2}^1 + t_{4,2}^1)\}$
$x_{2,8}^1 = \max\{(x_{5,7}^1 + \Delta t), (x_{2,7}^1 + t_{2,7}^1)\}$	$x_{4,4}^1 = \max\{(x_{2,3}^1 + \Delta t), (x_{4,3}^1 + t_{4,3}^1)\}$
-----	$x_{4,5}^1 = \max\{(x_{4,8}^1 + \Delta t - \alpha), (x_{4,4}^1 + t_{4,4}^1)\}$
$x_{2,1}^2 = \max\{(x_{2,8}^1 + \Delta t - \alpha), (x_{2,8}^2 + t_{2,8}^2 - \alpha)\}$	$x_{4,6}^1 = \max\{(x_{4,1}^1 + \Delta t), (x_{4,5}^1 + t_{4,5}^1)\}$
$x_{2,2}^2 = \max\{(x_{2,1}^1 + \Delta t), (x_{2,1}^2 + t_{2,1}^2)\}$	$x_{4,7}^1 = \max\{(x_{4,2}^1 + \Delta t), (x_{4,6}^1 + t_{4,6}^1)\}$
$x_{2,3}^2 = \max\{(x_{2,2}^1 + \Delta t), (x_{2,2}^2 + t_{2,2}^2)\}$	$x_{4,8}^1 = \max\{(x_{7,1}^1 + \Delta t - \alpha), (x_{4,7}^1 + t_{4,7}^1)\}$
$x_{2,4}^2 = \max\{(x_{4,5}^1 + \Delta t), (x_{2,3}^2 + t_{2,3}^2)\}$	-----
$x_{2,5}^2 = \max\{(x_{2,4}^1 + \Delta t), (x_{2,4}^2 + t_{2,4}^2)\}$	$x_{5,1}^2 = \max\{(x_{5,7}^1 + \Delta t - \alpha), (x_{5,8}^2 + t_{5,8}^2 - \alpha)\}$
$x_{2,6}^2 = \max\{(x_{2,5}^1 + \Delta t), (x_{2,5}^2 + t_{2,5}^2)\}$	$x_{5,2}^2 = \max\{(x_{5,8}^1 + \Delta t - \alpha), (x_{5,1}^2 + t_{5,1}^2)\}$
$x_{2,7}^2 = \max\{(x_{2,1}^1 + \Delta t), (x_{2,6}^2 + t_{2,6}^2)\}$	$x_{5,3}^2 = \max\{(x_{5,1}^1 + \Delta t), (x_{5,2}^2 + t_{5,2}^2)\}$
$x_{2,8}^2 = \max\{(x_{1,1}^1 + \Delta t - \alpha), (x_{2,7}^2 + t_{2,7}^2)\}$	$x_{5,4}^2 = \max\{(x_{5,2}^1 + \Delta t), (x_{5,3}^2 + t_{5,3}^2)\}$
-----	$x_{5,5}^2 = \max\{(x_{5,3}^1 + \Delta t), (x_{5,4}^2 + t_{5,4}^2)\}$
$x_{2,1}^3 = \max\{(x_{2,8}^2 + \Delta t - \alpha), (x_{2,8}^3 + t_{2,8}^3 - \alpha)\}$	$x_{5,6}^2 = \max\{(x_{5,4}^1 + \Delta t), (x_{5,5}^2 + t_{5,5}^2)\}$
$x_{2,2}^3 = \max\{(x_{2,1}^2 + \Delta t), (x_{2,1}^3 + t_{2,1}^3)\}$	$x_{5,7}^2 = \max\{(x_{5,5}^1 + \Delta t), (x_{5,6}^2 + t_{5,6}^2)\}$
$x_{2,3}^3 = \max\{(x_{2,2}^2 + \Delta t), (x_{2,2}^3 + t_{2,2}^3)\}$	$x_{5,8}^2 = \max\{(x_{5,6}^1 + \Delta t), (x_{5,7}^2 + t_{5,7}^2)\}$
$x_{2,4}^3 = \max\{(x_{2,3}^2 + \Delta t), (x_{2,3}^3 + t_{2,3}^3)\}$	-----
$x_{2,5}^3 = \max\{(x_{2,4}^2 + \Delta t), (x_{2,4}^3 + t_{2,4}^3)\}$	$x_{6,1}^1 = \max\{(x_{6,3}^2 + \Delta t - \alpha), (x_{6,5}^1 + t_{6,5}^1 - \alpha)\}$
$x_{2,6}^3 = \max\{(x_{2,5}^2 + \Delta t), (x_{2,5}^3 + t_{2,5}^3)\}$	$x_{6,2}^1 = \max\{(x_{5,1}^2 + \Delta t), (x_{6,1}^1 + t_{6,1}^1)\}$
$x_{2,7}^3 = \max\{(x_{2,6}^2 + \Delta t), (x_{2,6}^3 + t_{2,6}^3)\}$	$x_{6,3}^1 = \max\{(x_{6,5}^2 + \Delta t - \alpha), (x_{6,2}^1 + t_{6,2}^1)\}$
$x_{2,8}^3 = \max\{(x_{2,7}^2 + \Delta t), (x_{2,7}^3 + t_{2,7}^3)\}$	$x_{6,4}^1 = \max\{(x_{4,5}^2 + \Delta t), (x_{6,3}^1 + t_{6,3}^1)\}$
-----	$x_{6,5}^1 = \max\{(x_{6,2}^2 + \Delta t), (x_{6,4}^1 + t_{6,4}^1)\}$
$x_{4,1}^2 = \max\{(x_{4,8}^1 + \Delta t - \alpha), (x_{4,8}^2 + t_{4,8}^2 - \alpha)\}$	-----
$x_{4,2}^2 = \max\{(x_{4,1}^1 + \Delta t), (x_{4,1}^2 + t_{4,1}^2)\}$	$x_{6,1}^2 = \max\{(x_{6,3}^2 + \Delta t), (x_{6,5}^2 + t_{6,5}^2 - \alpha)\}$
$x_{4,3}^2 = \max\{(x_{4,2}^1 + \Delta t), (x_{4,2}^2 + t_{4,2}^2)\}$	$x_{6,2}^2 = \max\{(x_{6,2}^1 + \Delta t), (x_{6,1}^2 + t_{6,1}^2)\}$
$x_{4,4}^2 = \max\{(x_{4,3}^1 + \Delta t), (x_{4,3}^2 + t_{4,3}^2)\}$	$x_{6,3}^2 = \max\{(x_{8,4}^1 + \Delta t), (x_{6,2}^2 + t_{6,2}^2)\}$
$x_{4,5}^2 = \max\{(x_{4,4}^1 + \Delta t), (x_{4,4}^2 + t_{4,4}^2)\}$	$x_{6,4}^2 = \max\{(x_{6,4}^1 + \Delta t), (x_{6,3}^2 + t_{6,3}^2)\}$
$x_{4,6}^2 = \max\{(x_{2,7}^3 + \Delta t), (x_{4,5}^2 + t_{4,5}^2)\}$	$x_{6,5}^2 = \max\{(x_{7,3}^1 + \Delta t), (x_{6,4}^2 + t_{6,4}^2)\}$
$x_{4,7}^2 = \max\{(x_{4,6}^1 + \Delta t), (x_{4,6}^2 + t_{4,6}^2)\}$	-----
$x_{4,8}^2 = \max\{(x_{4,7}^1 + \Delta t), (x_{4,7}^2 + t_{4,7}^2)\}$	$x_{7,1}^1 = x_{7,4}^1 + t_{7,4}^1 - \alpha$
-----	$x_{7,2}^1 = \max\{(x_{6,5}^1 + \Delta t), (x_{7,1}^1 + t_{7,1}^1)\}$
$x_{5,1}^1 = \max\{(x_{5,5}^2 + \Delta t - \alpha), (x_{5,8}^1 + t_{5,8}^1 - \alpha)\}$	$x_{7,3}^1 = \max\{(x_{4,4}^2 + \Delta t), (x_{7,2}^1 + t_{7,2}^1)\}$
$x_{5,2}^1 = \max\{(x_{5,6}^2 + \Delta t - \alpha), (x_{5,1}^1 + t_{5,1}^1)\}$	$x_{7,4}^1 = \max\{(x_{4,3}^2 + \Delta t), (x_{7,3}^1 + t_{7,3}^1)\}$
$x_{5,3}^1 = \max\{(x_{2,5}^3 + \Delta t), (x_{5,2}^1 + t_{5,2}^1)\}$	-----
$x_{5,4}^1 = \max\{(x_{5,8}^2 + \Delta t - \alpha), (x_{5,3}^1 + t_{5,3}^1)\}$	$x_{8,1}^1 = \max\{(x_{5,3}^2 + \Delta t), (x_{8,4}^1 + t_{8,4}^1 - \alpha)\}$

**Tabela 3.12.** Ograniczenia determinujące dopuszczalne wartości momentów rozpoczęcia procesów lokalnych SWMPC z rys. 3.29

$m^1 P_1^h$	$m^1 P_2^h$	$m^1 P_3^h$
$x_{1,2}^1 = m^1 x_{1,1}^h + l \cdot \alpha$	$x_{5,1}^1 = m^1 x_{2,1}^h + l \cdot \alpha$	$x_{8,1}^1 = m^1 x_{3,1}^h + l \cdot \alpha$
$x_{1,3}^1 = m^1 x_{1,2}^h + l \cdot \alpha$	$x_{5,2}^1 = m^1 x_{2,2}^h + l \cdot \alpha$	$x_{8,2}^1 = m^1 x_{3,2}^h + l \cdot \alpha$
$x_{1,4}^1 = m^1 x_{1,3}^h + l \cdot \alpha$	$x_{5,3}^1 = m^1 x_{2,3}^h + l \cdot \alpha$	$x_{8,3}^1 = m^1 x_{3,3}^h + l \cdot \alpha$
$x_{2,7}^1 = m^1 x_{1,4}^h + l \cdot \alpha$	$x_{5,7}^2 = m^1 x_{2,4}^h + l \cdot \alpha$	$x_{6,4}^2 = m^1 x_{3,4}^h + l \cdot \alpha$
$x_{2,8}^1 = m^1 x_{1,5}^h + l \cdot \alpha$	$x_{5,8}^2 = m^1 x_{2,5}^h + l \cdot \alpha$	$x_{6,5}^2 = m^1 x_{3,5}^h + l \cdot \alpha$
$x_{2,5}^3 = m^1 x_{1,6}^h + l \cdot \alpha$	$x_{6,3}^1 = m^1 x_{2,6}^h + l \cdot \alpha$	$x_{4,3}^1 = m^1 x_{3,6}^h + l \cdot \alpha$
$x_{2,6}^3 = m^1 x_{1,7}^h + l \cdot \alpha$	$x_{6,4}^1 = m^1 x_{2,7}^h + l \cdot \alpha$	$x_{4,4}^1 = m^1 x_{3,7}^h + l \cdot \alpha$
$x_{4,7}^2 = m^1 x_{1,8}^h + l \cdot \alpha$	$x_{7,3}^1 = m^1 x_{2,8}^h + l \cdot \alpha$	$x_{2,5}^2 = m^1 x_{3,8}^h + l \cdot \alpha$
$x_{4,8}^2 = m^1 x_{1,9}^h + l \cdot \alpha$	$x_{7,4}^1 = m^1 x_{2,9}^h + l \cdot \alpha$	$x_{2,6}^2 = m^1 x_{3,9}^h + l \cdot \alpha$
$l \in \mathbb{C}, h = 1 \dots 4$	$l \in \mathbb{C}, h = 1 \dots 4$	$x_{3,3}^1 = m^1 x_{3,10}^h + l \cdot \alpha$
		$x_{3,4}^1 = m^1 x_{3,11}^h + l \cdot \alpha$
		$l \in \mathbb{C}, h = 1 \dots 4$
$mod(m^1 \alpha, \alpha) = 0$		

W ramach ograniczeń przyjęto, że moment rozpoczęcia operacji  $x_{i,j}^h$  potoku  $P_i^h$  procesu lokalnego (wózka) musi mieć taką samą wartość jak moment rozpoczęcia operacji  $m^1 x_{a,b}^c$  procesu multimodalnego, wymagającego do swojej realizacji potoku  $P_i^h$ . Innymi słowy, moment rozpoczęcia pracy wózka (operacji rozładunku/transportu) na danym zasobie pokrywa się z momentem rozpoczęcia odpowiedniej operacji procesu produkcji danego elementu. Przykładowo:  $x_{1,2}^1 = m^1 x_{1,1}^h + l \cdot \alpha$  oznacza, że druga operacja potoku  $P_1^1$  (realizowana na zasobie  $R_3$ ) rozpoczyna się wraz operacją  $m^1 o_{1,1}^h$   $h$ -tego potoku procesu multimodalnego  $m^1 P_1$ . Ponadto cykliczna realizacja operacji procesów lokalnych i multimodalnych wymaga spełniania ograniczenia wzajemnej wielokrotności okresów  $m^1 \alpha$ ,  $\alpha$  (Bocewicz et al. 2013b):  $mod(m^1 \alpha, \alpha) = 0$ .

Rozważany problem  $PS'_{REX}$ , zaimplementowany w środowisku programowania z ograniczeniami OzMozart (Intel Core Duo2 3.00 GHz, 4 GB RAM), został rozwiązany w czasie nie przekraczającym 7 s.

Wyznaczone wartości czasów wykonywania  $T$  operacji procesów lokalnych (wózków), gwarantujących wykonanie planu produkcyjnego z rys. 3.32, mają postać:

$$T = \{T_1^1, T_2^1, T_2^2, T_2^3, T_3^1, T_4^1, T_4^2, T, T_5^2, T_6^1, T_6^2, T_7^1, T_8^1\}$$

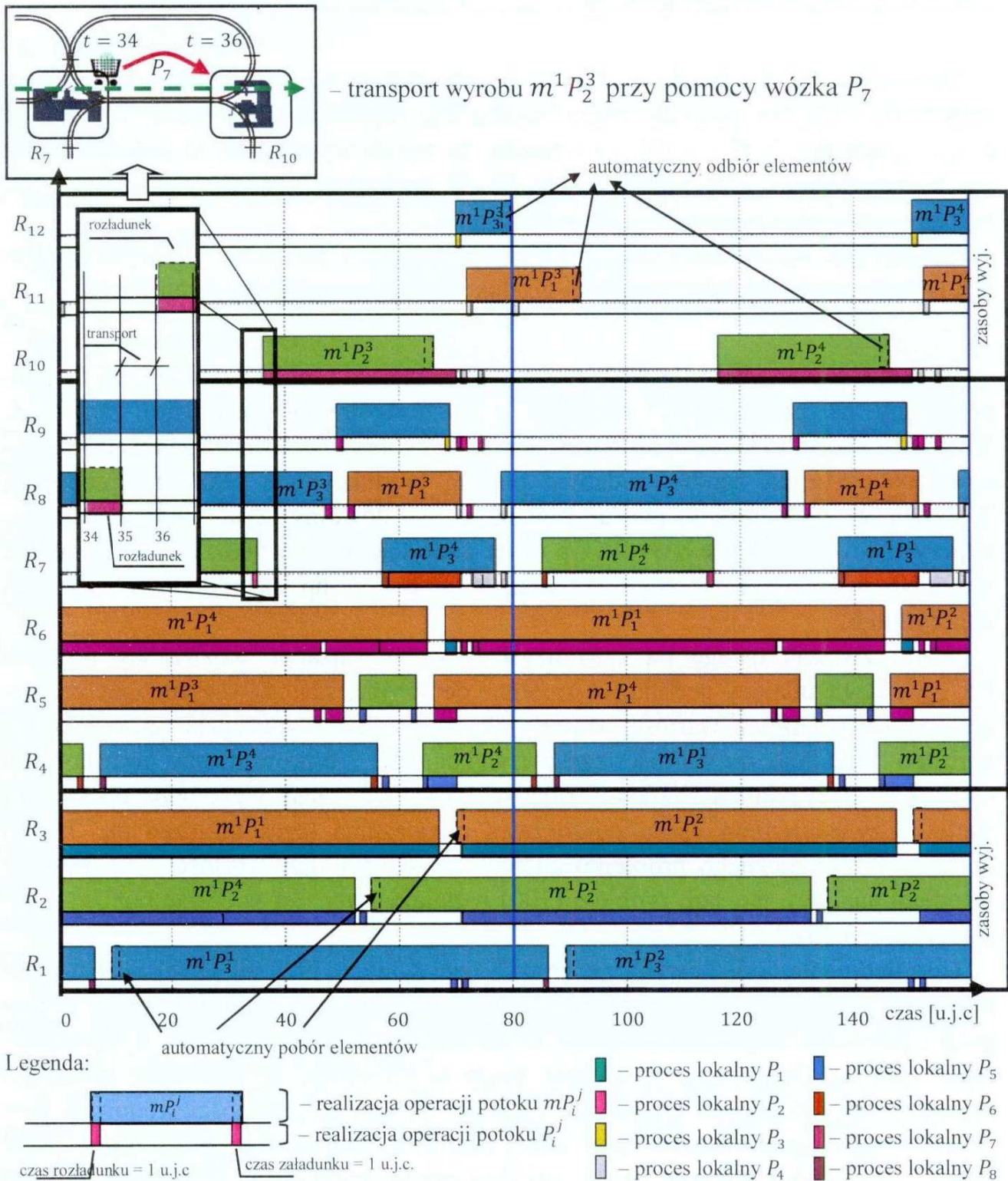
$$\begin{array}{lll} T_1^1 = (1, 76, 1, 2) & T_4^1 = (1, 4, 1, 1, 1, 1, 1, 1) & T_6^1 = (13, 1, 1, 1, 64) \\ T_2^1 = (1, 1, 1, 1, 1, 18, 1, 4) & T_4^2 = (1, 1, 1, 1, 71, 1, 1, 1) & T_6^2 = (1, 51, 1, 1, 1) \\ T_2^3 = (53, 1, 1, 1, 1, 1, 1, 1) & T_5^1 = (61, 1, 1, 1, 1, 1, 1, 1) & T_7^1 = (44, 1, 1, 34) \\ T_2^2 = (1, 1, 1, 3, 1, 1, 1, 1) & T_5^2 = (1, 1, 1, 1, 8, 1, 1, 1) & T_8^1 = (1, 1, 1, 77) \\ T_3^1 = (1, 77, 1, 1) & & \end{array}$$

gdzie:

$T_i^h = (t_{i,1}^h, \dots, t_{i,j}^h, \dots, t_{i,lr(i)}^h)$  – sekwencja czasów wykonywania operacji potoku  $P_i^h$ ,  
 $t_{i,j}^h$  – czas wykonywania  $j$ -tej operacji  $h$ -tego potoku procesu  $P_i$ .

Uzyskane wartości momentów  $x_{i,j}^h$  (sekwencji  $X'_{DC}$ ) gwarantujących cykliczną realizację procesów lokalnych zostały przedstawione w tab. 3.13.

Harmonogram z rys. 3.34 ilustruje jednoczesną realizację procesów lokalnych i multimodalnych. Jak widać operacje wykonywane są zgodnie z przyjętymi założeniami. Przykładowo, wyrób realizowany w procesie  $m^1P_2^3$  rozpoczynający swoją operację na zasobie  $R_{10}$ , w 36 jednostce czasu wymaga transportu (przy użyciu wózka  $P_7^1$ ) z zasobu  $R_7$ . Wymagane jest zatem aby wózek  $P_7^1$  był obecny w momencie załadunku wyrobu na zasobie  $R_7$  (ostatnia jednostka operacji  $m^1P_2^3$  na  $R_7$ ) oraz w momencie rozładunku na zasobie  $R_{10}$  (pierwsza jednostka operacji  $m^1P_2^3$  na  $R_{10}$ ).



Rys. 3.34. Harmonogram cykliczny systemu z rys. 3.29

**Tabela 3.13.** Momenty rozpoczęcia operacji  $X'_{DC}$  procesów lokalnych gwarantujących wykonanie harmonogramu z rys. 3.32

$x_i^h = (x_{i,1}^h, \dots, x_{i,j}^h, \dots, x_{i,lr(i)}^h)$	
$x_1^1 = (-10, -9, 67, 68)$	$x_5^1 = (-9, 52, 53, 54, 57, 58, 69, 70)$
$x_2^1 = (-10, -8, -7, -6, -5, 47, 65, 66)$	$x_5^2 = (-10, -9, -8, 53, 54, 62, 63, 64)$
$x_2^2 = (-8, 45, 46, 47, 48, 49, 50, 71)$	$x_6^1 = (-10, 3, 4, 5, 6)$
$x_2^3 = (-8, -7, 46, 47, 50, 51, 52, 70)$	$x_6^2 = (-9, 4, 55, 56, 57)$
$x_3^1 = (-10, -9, 68, 69)$	$x_7^1 = (-10, 34, 35, 36)$
$x_4^1 = (-8, -7, -3, -2, -1, 0, 1, 71)$	$x_8^1 = (5, 6, 7, 8)$
$x_4^2 = (-7, -6, -5, -2, -1, 70, 71, 72)$	$\alpha = 80$

Oznacza to, że oba potoki  $m^1P_2^3$  i  $P_7^1$  przemieszczają się równocześnie między zasobami  $R_7$  i  $R_{10}$  (za pośrednictwem zasobu  $R_{29}$  reprezentującego sektor łączący  $R_7$  z  $R_{10}$  – patrz rys. 3.29 i 3.30), co oznacza, że wyrób wytwarzany w procesie  $m^1P_2^3$  jest transportowany przy użyciu wózka  $P_7^1$ . W analogiczny sposób realizowany jest transport pomiędzy pozostałymi stanowiskami.

Reasumując wyznaczone czasy wykonania operacji  $T$  procesów lokalnych pozwalają na wykonanie zakładanego planu produkcyjnego bez opóźnień i konfliktów. ■

Przestawiony przykład ilustruje przypadek, w którym problem syntezy jest dekomponowany na podproblemy związane z poszczególnymi poziomami zachowań. W ramach znanej częściowo struktury poszukiwane były wartości zmiennych  $T$ , gwarantujące cykliczną zgodną zadanym planem produkcyjnym realizację procesów. W ogólności, przedmiotem poszukiwań może być dowolny zestaw zmiennych charakteryzujących strukturę SWMPC, a użyte podejście może wykorzystywać strategię top-down, jak w prezentowanym przykładzie lub bottom-up jak zakłada się w ramach Algorytmu 6.

Warto zwrócić uwagę na czas uzyskanego rozwiązania. Rozważany problem  $PS'_{REX}$ , uwzględniający dodatkowe zmienne decyzyjne, stanowi rozszerzenie problemu  $PS'_X$  (89). Oprócz wartości zmiennych momentów rozpoczęcia operacji  $X'_{DC}$  i okresu  $\alpha$  poszukuje się wartości czasów wykonania  $T$ . Wprowadzenie dodatkowych zmiennych decyzyjnych charakteryzujących strukturę (czasy wykonania operacji  $T$ ) skutkuje zwiększeniem czasu poszukiwań rozwiązania (czas obliczeń ok. 7 sek) względem analogicznego problemu analizy. W sytuacji gdy wartości  $T$  są ustalone (problem analizy), ten sam problem (100) rozwiązywany jest w czasie poniżej 1 sekundy (rys. 3.25).

Uwzględnianie dodatkowych zmiennych decyzyjnych, przy tej samej liczbie ograniczeń co w analogicznym problemie analizy, osłabia efektywności procedur propagacji ograniczeń implementowanych w środowiskach programowania z ograniczeniami (jak np. OzMozart). W efekcie, może to prowadzić do przeglądu pełnego przestrzeni potencjalnych rozwiązań. Zjawisko niskiej efektywności środowisk programowania z ograniczeniami przy dużej liczbie zmiennych decyzyjnych jest znane (Sitek 2013; Sitek i Wikarek 2013a, 2013b) i często spotykane w problemach o dużej skali (np. problemy zarządzania łańcuchami dostaw itp.). W przypadkach tych stoso-

wane są podejścia oparte na wprowadzaniu dodatkowych ograniczeń heurystycznych, czy też wykorzystaniu technik programowania matematycznego (MIP/ MILP).

Wykładniczy charakter procedur rozwiązywania problemu syntezy potwierdza wykres przedstawiony na rys. 3.27. Przedstawiony wykres odpowiada szczególnemu przypadkowi, w którym rozważane są różne warianty reguł priorytetowania, rozwiązywania problemu wstecz w oparciu o Algorytm 6. Skala problemów możliwych do rozwiązania w praktyce przy użyciu proponowanego podejścia (stosując bezpośrednio środowiska programowania z ograniczeniami) ogranicza się do systemów, w których liczba operacji nie przekracza 40.

### 3.4. Podsumowanie

W rozdziale przedstawione zostały dwie metody harmonogramowania cyklicznego SWMPC:

- metoda redukcji stanów początkowych, oparta na idei odtwarzania przebiegów cyklicznych w wyniku usuwania prowadzących do nich przebiegów przejściowych,
- metoda generowania harmonogramów, oparta na idei analizy acykliczności digrafów pierwszeństwa operacji.

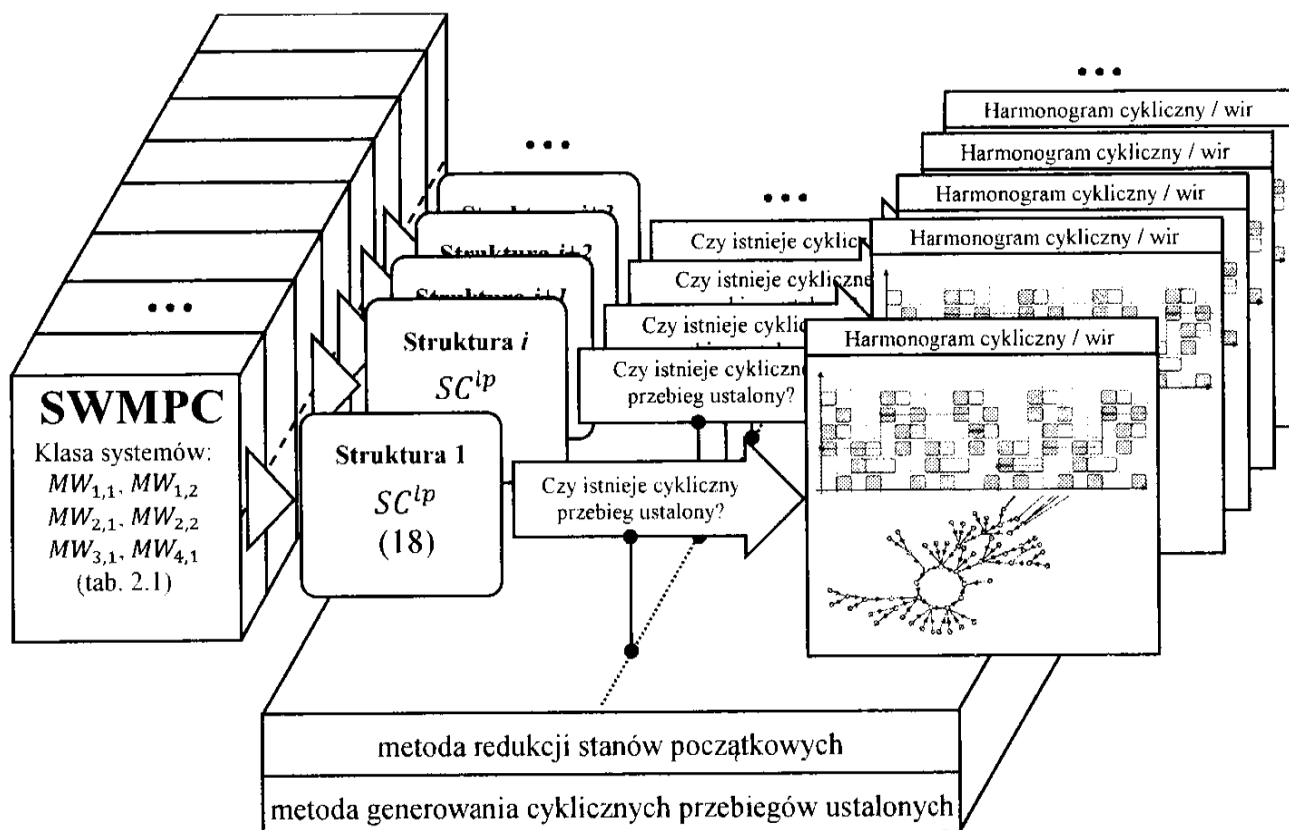
Proponowane metody wpisują się w koncepcję podejścia analizy SWMPC (rys. 3.35), w której dla zadanej struktury poszukiwane jest cykliczne zachowanie systemu.

Obie metody mają charakter deklaracyjny to znaczy, że rozważany problem wyrażony jest w kategoriach (jednego lub wielu) problemów spełniania ograniczeń (zbioru zmiennych, dziedzin i ograniczeń). Do implementacji tych metod można więc wykorzystywać komercyjnie dostępne środowiska programowania z ograniczeniami (ILOG, ECLiPS<sup>e</sup>, OzMozart, itp.)

Pierwsza z zaproponowanych metod umożliwia analizę zachowania systemu jednocześnie dla wszystkich poziomów struktury  $SC^{lp}$ . Wyznaczenie cyklicznych przebiegów ustalonych odbywa się na drodze redukcji (usuwania) z przestrzeni  $\mathcal{P}$  przebiegów przejściowych. W związku z tym metoda pozwala wyznaczyć przebiegi cykliczne oraz wszystkie prowadzące do nich przebiegi przejściowe (wyznaczając w ten sposób wiry przestrzeni  $\mathcal{P}$ ). Cechą szczególną jest to, że w oparciu wyznaczany zbiór odciętych stanów można wyznaczyć warunki gwarantujące cykliczne zachowanie systemów. Przez warunki tego typu rozumie się zbiór ograniczeń wykluczających z przestrzeni stanów stany składające się na przebiegi przejściowe.

Możliwość wyznaczania wirów jest największą zaletą proponowanej metody. Znajomość przebiegów cyklicznych wraz z towarzyszącymi im przebiegami przejściowymi ma istotne znaczenie praktyczne.

W ogólnym przypadku zastosowanie proponowanej metody ogranicza się do systemów o przestrzeniach  $\mathcal{P}$  zawierających niewielką liczbę stanów dopuszczalnych tzw. systemów gęstych.



**Rys. 3.35.** Analiza SWMPC (zgodna z koncepcją rys. 1.23), uwzględniająca opracowane metody harmonogramowania

Druga z proponowanych metod pozwala dokonać oceny zachowania systemu w oparciu o ocenę istnienia harmonogramów cyklicznych. Istnienie harmonogramów cyklicznych warunkowane jest acyklicznością tak zwanych digrafów pierwszeństwa operacji. O ile w pierwszej metodzie ocena zachowań cyklicznych odnosi się do analizy stanów systemu o tyle w drugim przypadku ocenie podlegają wartości momentów rozpoczęcia operacji procesów realizowanych w systemie. Wyznaczanie harmonogramów cyklicznych dla systemów z ustalonymi semaforami (wskazującymi na pierwszy element każdej reguły) okazuje się zadaniem o liniowej złożoności obliczeniowej. Przegląd potencjalnych wariantów reguł priorytetowania (cyklicznych przesunięć) charakteryzuje się wykładniczą złożonością obliczeniową.

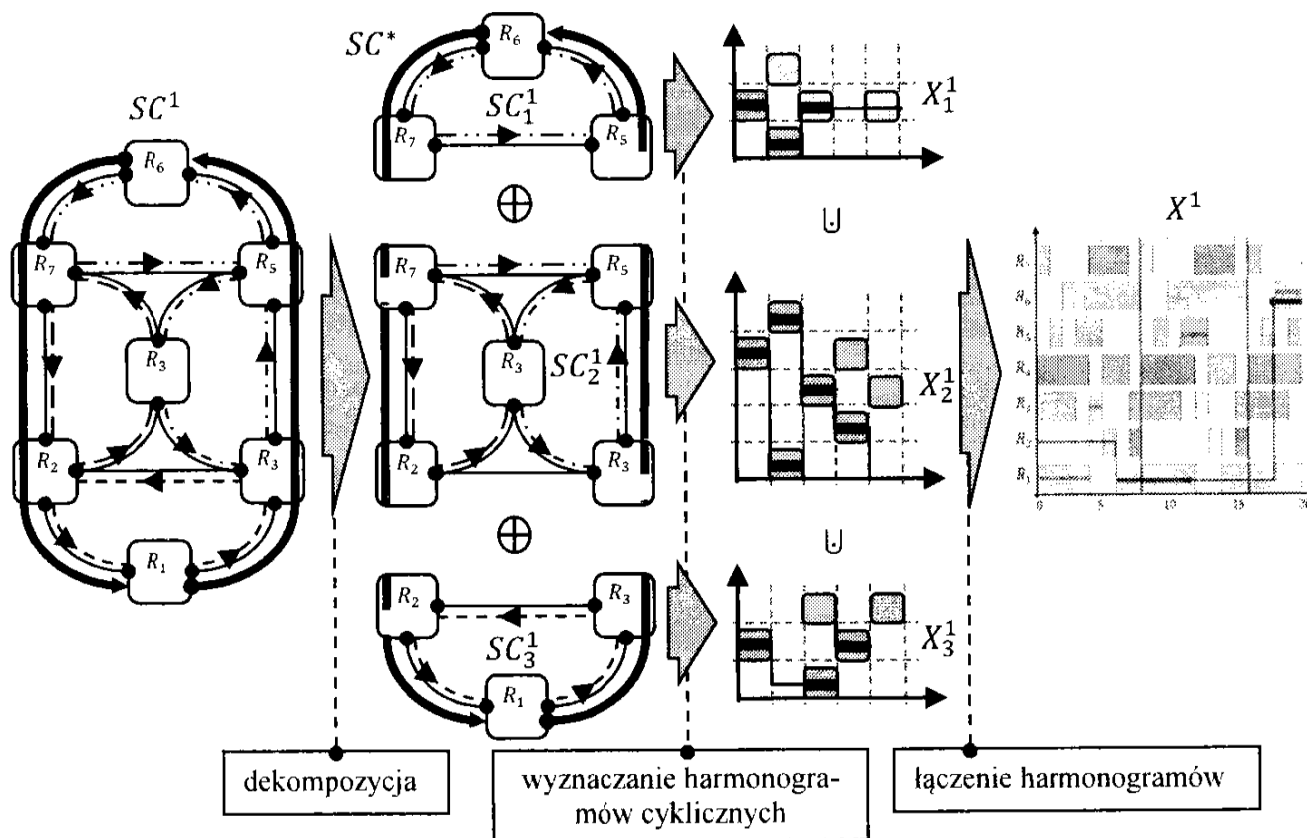
Przyjęty sposób reprezentacji problemu w przypadku metody generowania harmonogramów cyklicznych pozwala na wykorzystanie tej metody również w podejściu syntezy. Przeprowadzone eksperymenty wskazują, że metoda ta może znaleźć zastosowanie w systemach o rozmiarach nie przekraczających 40 realizowanych operacji.

## 4. Synteza SWMPC

Wspólną cechą omawianych problemów analizy i syntezy systemów klasy SWMPC jest „całościowe” postrzeganie ich struktur, sprowadzające się do „jednoczesnego” postrzegania związków łączących wszystkie elementy struktury ze wszystkimi przejawami zachowania systemu. Oznacza to, że zarówno w przypadku oceny cyklicznych zachowań systemu, jak i parametrów gwarantujących żądane zachowania, przedmiotem analizy jest cała struktura (obejmująca wszystkie zasoby i procesy realizowane w systemie).

Alternatywą takiego podejścia, jest poszukiwanie warunków wystarczających spełnienie których gwarantuje oczekiwane zachowanie łączonych podsystemów. W szczególności, oznacza to poszukiwanie odpowiedzi na pytanie jakie klasy struktur, jak łączonych ze sobą, składają się na strukturę systemu realizującego oczekiwane zachowanie? Do zalet tego podejścia należy zaliczyć możliwość zmniejszenia efektu skali (dekompozycja systemów) oraz wariantowania rozbudowy planowanych systemów.

Łatwo zauważyć, że skala systemu determinująca złożoność jego struktury, silnie ogranicza zakres zastosowań wcześniej proponowanych metod (patrz rys. 3.12 oraz rys. 3.27). Często też występują sytuacje, w których decyzja o zachowaniu systemu podejmowana jest w oparciu o analizę jego fragmentu lub też odwrotnie, gdy parametry rozbudowywanej części systemu dobierane są pod kątem jego docelowego zachowania, np. gwarantującego cykliczność przebiegu realizowanych w nim procesów. Sytuacje tego typu występują często przy planowaniu rozkładów jazdy autobusów komunikacji miejskiej, rozkładów uwzględniających np. istniejące harmonogramy linii metra, pociągów, czy rejsy promów.



Rys. 4.1. Ogólna idea składania struktur SWMPC

Reasumując, zgodnie z przykładem 3, z rozdziału 2, ocena pracy systemu może być podejmowana na podstawie oceny zachowań jego fragmentów. W konsekwencji oznacza to, że ocena pożądanego, np. cyklicznego zachowania systemu może być uzyskana na drodze wieloetapowego łączenia fragmentów jego struktury i wyboru stosownych (gwarantujących cykliczne zachowania) reguł priorytetowania. Przedstawione postępowanie, sprowadzające się do budowy (lub odtwarzania) struktury spełniającej zadane, oczekiwane zachowanie systemu, ilustruje istotę podejścia wykorzystywanego przy rozwiązywaniu problemu składania (69), (70). Może być ono traktowane jako szczególny przypadek syntezy SWMPC powstałego w wyniku składania jego składowych podstruktur. Analizowany dalej problem składania struktur SWMPC koncentruje się na syntezie struktur regularnych. Ogólną ideę składania struktur SWMPC przedstawia rys. 4.1.

#### 4.1. Problem składania struktur

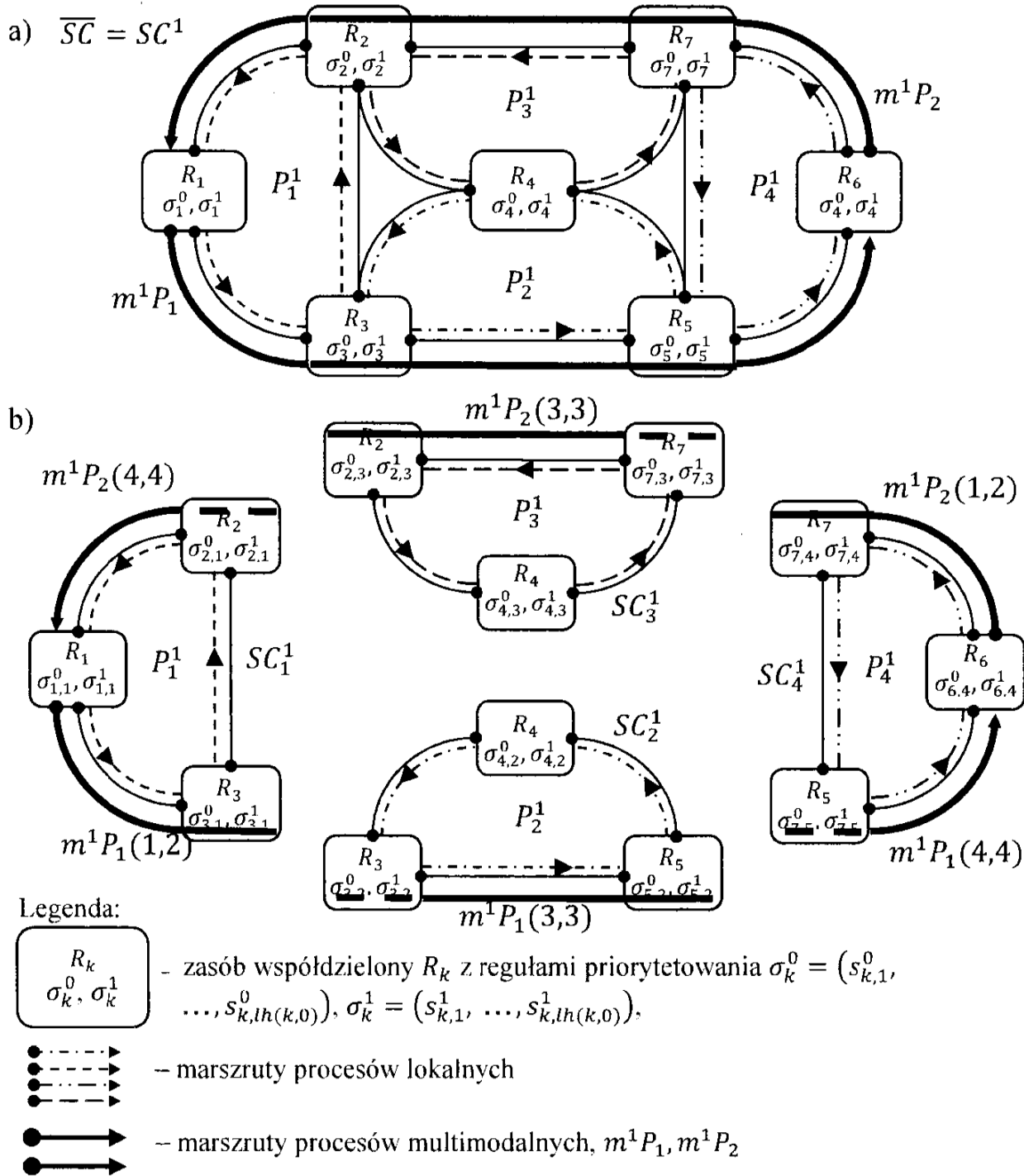
Problem składania, zdefiniowany w podrozdziale 2.3 (68) – (70), sprowadza się do poszukiwania, w zadanym zbiorze struktur  $SC^*$ , takiej ich kompozycji  $\overline{SC}$ , która gwarantuje cykliczną realizację procesów (lokalnych i multimodalnych). W szczególnym przypadku, problem ten odnosi się do sytuacji, w której topologia struktury wynikowej  $\overline{SC}$  jest znana, a do jej rekonstrukcji wykorzystywane są wszystkie elementy zbioru  $SC^*$ , zbioru podstruktur otrzymanych w wyniku dekompozycji  $\overline{SC}$ . Przykład takiej sytuacji ilustruje rys. 4.2 przedstawiający system o dwupoziomowej strukturze  $SC^1$  (rys. 4.2a), w którym realizowane są 4 jednopotokowe procesy lokalne:  $P_1, P_2, P_3, P_4$  i dwa jednopotokowe procesy multimodalne  $m^1P_1, m^2P_2$ . Struktura  $SC^1$  została zdekomponowana na 4 podstruktury:  $SC^* = \{SC_1^1, SC_2^1, SC_3^1, SC_4^1\}$  (rys. 4.2b), gdzie każda podstruktura  $SC_i^1$  gwarantuje cykliczną realizację wchodzących w jej skład procesów. Cykliczność systemu końcowego wynika z „niekolidującego” sposobu składania fragmentów procesów multimodalnych  $m^1P_{i,j}$  i ich nośników – podstruktur  $SC_i^1$  obejmujących pojedyncze procesy lokalne.

Rozważany problem składania systemów klasy SWMPC, sprowadza się do odpowiedzi na pytanie: Czy istnieje sposób składania podstruktur zbioru  $SC^*$ , gwarantujący cykliczne zachowanie systemu  $SC^1$ ?

W rozważanym podejściu przyjmuje się, że znane są punkty łączenia podstruktur (zasoby  $R_2, R_3, R_4, R_5, R_7$ ), nieznane są natomiast reguły priorytetowania związane z tymi punktami (zasobami). Pytanie o sposób składania podstruktur sprowadza się więc do sposobu wyznaczania reguł priorytetowania. Przykładowo, składanie podstruktur  $SC_1^1, SC_3^1$  wymaga podania sposobu łączenia reguł  $\sigma_{2,1}^0, \sigma_{2,3}^0$  na zasobie  $R_2$ , w regułę  $\sigma_2^0$  gwarantującą cykliczną realizację procesów struktury wynikowej.

Tak widziany problem składania może być postrzegany jako szczególny przypadek problemu syntezy, w którym dla częściowo określonej struktury  $SC^1$  poszukiwane są postaci reguł gwarantujących cykliczną pracę całego systemu. Okazuje się (co będzie pokazane w dalszej części rozdziału), że możliwość postrzegania struktury  $SC^1$  jako kompozycji wielu podstruktur  $SC^*$  może prowadzić do sytuacji, w których tak formułowane problemy syntezy będą rozwiązywane w sposób bardziej efektywny niż ma to np. miejsce w przypadku Algorytmu 6.





Rys. 4.2. Przykład SWMPC a) zdekomponowanego na 4 podsystemy b)

W celu formalnego zdefiniowania problemu składania podstruktur  $SC^*$  systemu o znanej strukturze  $\overline{SC} = SC^{lp}$  wprowadza się następujące założenia:

i) podstruktura  $SC_i^{lp}$  struktury  $SC^{lp}$  definiowana jest w analogiczny sposób jak (18):

$$SC_i^{lp} = \left( \left( \left( \left( \left( \left( R p_i, S L p_i \right), S M p_i^1 \right), S M p_i^2 \right), \dots, S M p_i^l \right), \dots \right), S M p_i^{lp} \right), \quad (101)$$

gdzie:

$R p_i$  – zbiór zasobów podstruktury  $SC_i^{lp}$ ,  $R p_i \subset R$ ,

$SLp_i = \left( (Pp_i, Up_i, Op_i, Tp_i), (\Theta_i^0, \Psi_i^0) \right)$  – struktura poziomu zachowań procesów lokalnych podstruktury  $SC_i^{lp}$ , w skład której wchodzi procesy lokalne  $Pp_i \subset P$  oraz odpowiadające im sekwencje marszrut  $Up_i \subset U$ , operacji  $Op_i \subset O$ , czasów wykonania operacji  $Tp_i \subset T$ . Zbiór marszrut  $Up_i$  obejmuje wszystkie zasoby  $Rp_i$  podstruktury. Zbiór reguł priorytetowania podstruktury  $SC_i^{lp}$  jest z kolei charakteryzowany następująco  $\Theta_i^0 = \{ \sigma_{k,i}^0 = (s_{k,1,i}^0, \dots, s_{k,d,i}^0, \dots, s_{k,lh(k,i),i}^0) \mid R_k \in Rp_i \}$ , gdzie:  $\sigma_{k,i}^0$  – reguła priorytetowania dla zasobu  $R_k$  w  $i$ -tej podstrukturze  $SC_i^{lp}$ ,  $s_{k,d,i}^0$  – potok procesu lokalnego  $Pp_i$ ,  $lh(k,i)$  – długość reguły  $\sigma_{k,i}^0$ . Sekwencja wzajemnych wykonań podstruktury  $SC_i^{lp}$  przyjmuje postać:  $\Psi_i^0 = (\psi_{i_1}^0, \psi_{i_2}^0, \dots, \psi_{i_j}^0, \dots, \psi_{i_{ln(i)}}^0)$ , gdzie:  $\psi_{i_j}^0$  – liczba wykonań procesu  $P_{i_j} \in Pp_i$ ,  $ln(i) = |Pp_i|$  – liczba procesów lokalnych podstruktury  $SC_i^{lp}$ .

$SMp_i^l = \left( (m^l Pp_i, m^l Up_i, m^l Op_i, m^l Tp_i), (\Theta_i^l, \Psi_i^l) \right)$  – struktura  $l$ -tego poziomu zachowań procesów multimodalnych podstruktury  $SC_i^{lp}$ , w skład której wchodzi **fragmenty  $m^l P_j(a, b)$  procesów multimodalnych** tworzących zbiór  $m^l Pp_i$ , gdzie:  $m^l P_j(a, b)$  – fragment procesu  $m^l P_j$  związany z wykonaniem operacji:  $m^l o_{j,a}^h, m^l o_{j,a+1}^h, \dots, m^l o_{j,b}^h$  wszystkich należących do tego procesu potoków  $m^l P_j^h$  ( $h = 1 \dots lsm(j, l)$ ).

Fragmenty procesów multimodalnych charakteryzowane są przez zbiory operacji  $m^l Op_i$  wykonywanych na zasobach  $Rp_i$  oraz fragmenty marszrut  $m^l Up_i$  (definiowanych jak w (17)). W skład podstruktury wchodzi wszystkie fragmenty  $m^l Pp_i$  procesów multimodalnych  $m^l P$  wykonywanych na zasobach  $Rp_i$ . Przykładowo podstruktura  $SC_1^l$  z rys. 4.2 zawiera fragment  $m^1 P_2(4,4)$  procesu  $m^1 P_2$  oraz fragment  $m^1 P_1(1,2)$  procesu  $m^1 P_1$ , tzn. fragmenty charakteryzowane odpowiednio przez marszruty:  $m^1 p_{2,3} = m^1 pr_2(4,4) = (R_1)$  oraz  $m^1 p_{1,1} = m^1 pr_1(1,2) = (R_1, R_3)$ .

Ponadto, analogicznie jak w przypadku  $SLp_i$ , wielkości  $m^l Tp_i$ ,  $\Theta_i^l$ ,  $\Psi_i^l$ , określają odpowiednio czasy wykonania operacji  $m^l Op_i$ , reguły priorytetowania procesów  $m^l Pp_i$  dla zasobów  $Rp_i$  i wzajemną liczbę wykonań  $\Psi_i^l$  procesów  $m^l Pp_i$ .

ii) zbiór  $SC^* = \{SC_1^{lp}, \dots, SC_i^{lp}, \dots, SC_{lc}^{lp}\}$  jest zbiorem podstruktur  $SC^{lp}$  jeżeli:

- podstruktury obejmują wszystkie zasoby struktury  $SC^{lp}$ :  $\bigcup_{i=1}^{lc} Rp_i = R$ ,
- podstruktury opisują wszystkie procesy lokalne:  $\bigcup_{i=1}^{lc} Pp_i = P$ ;  $\bigcup_{i=1}^{lc} Up_i = U$  przy czym każdemu procesowi odpowiada tylko jedna podstruktura,
- każdy fragment procesu multimodalnego jest elementem tylko jednej podstruktury, a poszczególne podstruktury pokrywają wszystkie fragmenty procesów multimodalnych.

Niech podstruktury  $SC_a^{lp}, SC_b^{lp} \in SC^*$  o wspólnych zasobach ( $Rp_a \cap Rp_b = Rk_{a,b} \neq \emptyset$ ), determinujące cykliczne zachowanie reprezentowanych podsystemów (w prze-

strzeniach stanów istnieją cykliczne przebiegi ustalone) są podstrukturami struktury  $SC^{lp}$ . **Operacja składania podstruktur**  $SC_a^{lp}$ ,  $SC_b^{lp}$  (oznaczana operatorem  $\oplus$ ) po zbiorze zasobów  $Rk_{a,b}$  jest definiowana jako:  $SC_a^{lp} \oplus SC_b^{lp} = SC_c^{lp}$ , gdzie:  $SC_c^{lp}$  jest również strukturą systemu o zachowaniu cyklicznym, postaci:

$$SC_c^{lp} = \left( (Rp_c, SLp_c), SMP_c^{lp} \right), \quad (102)$$

gdzie:

$Rp_c = Rp_a \cup Rp_b$  – zbiór zasobów podstruktury  $SC_c^{lp}$ ,

$SLp_c = ((Pp_c, Up_c, Op_c, Tp_c), (\theta_c^0, \psi_c^0))$  – struktura poziomu zachowań procesów lokalnych o elementach:

$Pp_c = Pp_a \cup Pp_b$ ;  $Up_c = Up_a \cup Up_b$ ;  $Op_c = Op_a \cup Op_b$ ;  $Tp_c = Tp_a \cup Tp_b$ ;  
 $\psi_c^0 = \psi_a^0 \wedge \psi_b^0$ ;

$\theta_c^0 = \{\sigma_{k,c}^0 | R_k \in Rp_c\}$ , gdzie:

$$\sigma_{k,c}^0 = \begin{cases} \sigma_{k,a}^0 & \text{dla } R_k \in Rp_a \text{ i } R_k \notin Rp_b \\ \sigma_{k,b}^0 & \text{dla } R_k \in Rp_b \text{ i } R_k \notin Rp_a, \\ \vartheta(\sigma_{k,a}^0, \sigma_{k,b}^0) & \text{dla } R_k \in Rp_a \text{ i } R_k \in Rp_b \end{cases} \quad (103)$$

$\vartheta(\sigma_{k,a}^0, \sigma_{k,b}^0)$  – funkcja określająca regułę priorytetowania dla wspólnego zasobu  $R_k \in Rk_{a,b}$  składanych podstruktur, która nie prowadzi do blokady procesów. W ogólności funkcja  $\vartheta$  definiowana jest jako:

$$\vartheta: Pp_{k,a}^l \stackrel{lh(k,a,l)}{\times} Pp_{k,b}^l \stackrel{lh(k,b,l)}{\times} \mapsto (Pp_{k,a}^l \cup Pp_{k,b}^l) \stackrel{lh(k,c,l)}{\times},$$

gdzie:  $A \times B$  – iloczyn kartezjański zbiorów  $A$  i  $B$ ,  $A^n = \underbrace{A \times A \times \dots \times A}_n$ ,

$Pp_{k,a}^l, Pp_{k,b}^l$  – zbiory potoków procesów lokalnych ( $l = 0$ ) lub multimodalnych ( $l > 0$ ) realizowanych na zasobie  $R_k$  odpowiednio w podstrukturze  $SC_a^{lp}$  i  $SC_b^{lp}$ ,

$lh(k, a, l), lh(k, b, l), lh(k, c, l)$  – liczba elementów wchodzących w skład sekwencji  $\sigma_{k,a}^l, \sigma_{k,b}^l, \sigma_{k,c}^l$ .

Zakłada się, że sekwencja  $\sigma_{k,c}^l$  będąca wartością funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$ , składa się ze wszystkich elementów sekwencji  $\sigma_{k,a}^l, \sigma_{k,b}^l$ .

$SMP_c^l = ((m^l Pp_c, m^l Up_c, m^l Op_c, m^l Tp_c), (\theta_c^l, \psi_c^l))$  – struktura poziomu zachowań procesów multimodalnych o elementach:

$m^l Pp_c$  – zbiór zawierający wszystkie fragmenty procesów multimodalnych zbiorów  $m^l Pp_a$  i  $m^l Pp_b$ . Przy czym każde dwa fragmenty:  $m^l P_j(a_{j_1}, a_{j_2}), m^l P_j(b_{j_1}, b_{j_2})$ , dla których spełnione jest  $b_{j_1} = a_{j_2} + 1$ , zastępowane są fragmentem  $m^l P_j(a_{j_1}, b_{j_2})$  (fragment będący złożeniem fragmentów  $m^l P_j(a_{j_1}, a_{j_2}), m^l P_j(b_{j_1}, b_{j_2})$ ). Tak uzyskany zbiór  $m^l Pp_c$  determinuje zbiór marszrut  $m^l Up_c$  (zbiór zawierający marszruty fragmentów procesów multimodalnych  $m^l Pp_c$ ), operacji  $m^l Op_c$  oraz czasów ich wykonania  $m^l Tp_c$ ,

$\psi_c^l$  – sekwencja określająca wzajemną liczbę wykonań procesów zbioru  $m^l Pp_c$  równą liczbie wykonań określoną w sekwencjach  $\psi_a^l$  i  $\psi_b^l$ .

$\theta_c^l = \{\sigma_{k,c}^l | R_k \in Rp_c\}$ , gdzie  $\sigma_{k,c}^l$  wyznaczana jest analogicznie jak (103):

$$\sigma_{k,c}^l = \begin{cases} \sigma_{k,a}^l & \text{dla } R_k \in Rp_a \text{ i } R_k \notin Rp_b \\ \sigma_{k,b}^l & \text{dla } R_k \in Rp_b \text{ i } R_k \notin Rp_a \\ \vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l) & \text{dla } R_k \in Rp_a \text{ i } R_k \in Rp_b \end{cases} \quad (104)$$

$\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  – funkcja określająca regułę priorytetowania dla wspólnego zasobu  $R_k \in Rk_{a,b}$  na  $l$ -tym poziomie zachowań. Analogicznie jak w (103) wynikiem funkcji jest reguła zawierająca wszystkie elementy sekwencji  $\sigma_{k,a}^l, \sigma_{k,b}^l$ .

Przykład ilustracji składania dwóch podstruktur  $SC_1^1 \oplus SC_2^1$  z rys. 4.2 po jednoelementowym zbiorze wspólnych zasobów  $Rk = \{R_3\}$  przedstawia rys. 4.3.

Cechą charakterystyczną tak zdefiniowanej operacji składania jest to, że jest ona możliwa tylko w przypadku pewnych określonych podstruktur, tzn. składanie nie każdej pary  $SC_a^{lp}, SC_b^{lp}$  podstruktur jest dopuszczalne. Po pierwsze składanie odbywa się po zbiorze wspólnych zasobów ( $Rk_{a,b} = Rp_a \cap Rp_b$ ), który musi być niepusty ( $Rk_{a,b} \neq \emptyset$ ). Warto zwrócić uwagę, że w ogólności dopuszcza się składanie struktur po dowolnym zbiorze zasobów (niekoniecznie wspólnych). W takich przypadkach (operacje tego typu będą dalej stosowane w składaniu struktur regularnych)  $Rk_{a,b}$  jest definiowane jako rodzina zbiorów zasobów podlegających łączeniu. Po drugie, składanie podstruktur powinno skutkować strukturą systemu o zachowaniu cyklicznym. W tym ujęciu istotna staje się przyjęta postać funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$ . Funkcja ta będąca składową wyrażen (103), (104) determinuje kolejność dostępu procesów do zasobów zbioru  $Rk_{a,b}$  i tym samym determinuje możliwość cyklicznej (bezblokadowej) ich realizacji.

W związku z powyższym **problem składania podstruktur  $SC^*$**  systemu o strukturze  $SC^{lp}$ , będący szczególnym przypadkiem problemu syntezy (charakteryzowanego przez (68) – (70)), formułowany jest następująco:

*Dana jest struktura  $SC^{lp}$  SWMPC zdekomponowana na zbiór podstruktur  $SC^* = \{SC_1^{lp}, \dots, SC_i^{lp}, \dots, SC_{lc}^{lp}\}$  reprezentujących podsystemy o znanych zachowaniach cyklicznych.*

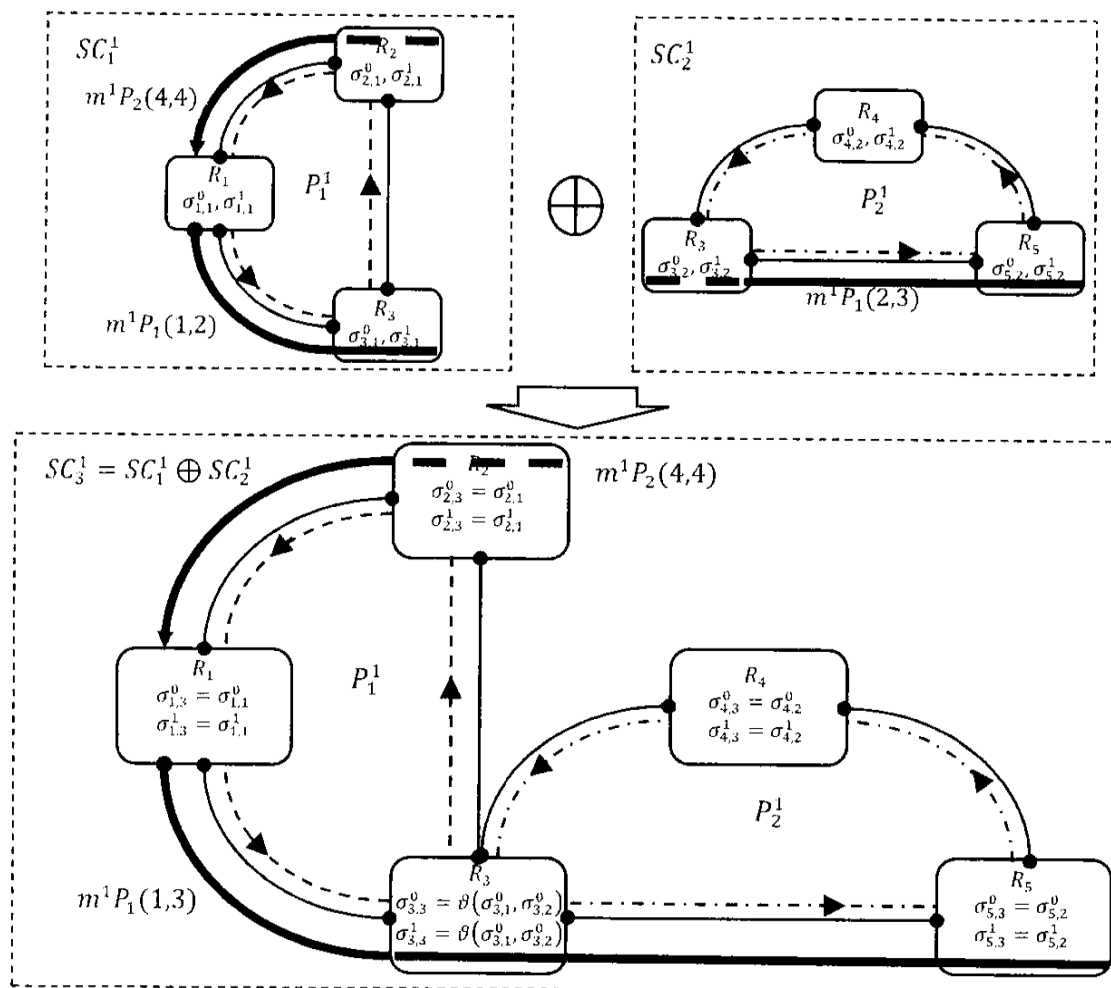
*Czy możliwe jest :  $\left( \left( \left( \left( SC_1^{lp} \oplus SC_2^{lp} \right) \oplus SC_3^{lp} \dots \right) \oplus SC_i^{lp} \right) \dots \right) \oplus SC_{lc}^{lp}$ ?*

Innymi słowy poszukiwana jest odpowiedź na pytanie czy w wyniku składania podstruktur generujących zachowania cykliczne, można uzyskać strukturę  $SC^{lp}$  systemu o zachowaniu cyklicznym?

To jakie zachowanie jest osiągnięte zależy od wartości zmiennych charakteryzujących strukturę (patrz podrozdział 2.2). Zgodnie z przyjętą definicją w wyniku składania  $SC_c^{lp} = SC_a^{lp} \oplus SC_b^{lp}$  zarówno marszruty procesów jaki i czasy wykonywania operacji nie ulegają zmianie. Tak więc, zachowanie systemu o strukturze wynikowej  $SC_c^{lp}$  jest determinowane wyłącznie przez przyjętą postać reguł priorytetowania  $\theta_c$ .

Przedstawiony (103), (104) sposób wyznaczania zbioru reguł  $\theta_c$  w strukturze wynikowej  $SC_c^{lp}$ , ogranicza się do zmiany reguł odnoszących się do zasobów, na których

dochodzi do składania podstruktur (tzn.  $R_k \in Rp_a$  i  $R_k \in Rp_b$ ). To jakie reguły zostaną przypisane do tych zasobów determinuje postać przyjętej funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$ . Oznacza to, że pozytywna odpowiedź na postawione pytanie warunkowana jest „odpowiednią” postacią funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  tzn. gwarantującą cykliczne zachowanie systemu o strukturze wynikowej  $SC_c^{lp}$ .



Legenda:

- $R_k$   
 $\sigma_{k,i}^0, \sigma_{k,i}^1$  – zasób współdzielony  $R_k$  z regułami priorytetowania  $\sigma_{k,i}^0, \sigma_{k,i}^1$
- - - - - -> – marszruty potoków procesów lokalnych  $P_1^1$  i  $P_2^1$
- ———> – fragmenty marszrut procesów multimodalnych  $m^1P_1, m^1P_2$

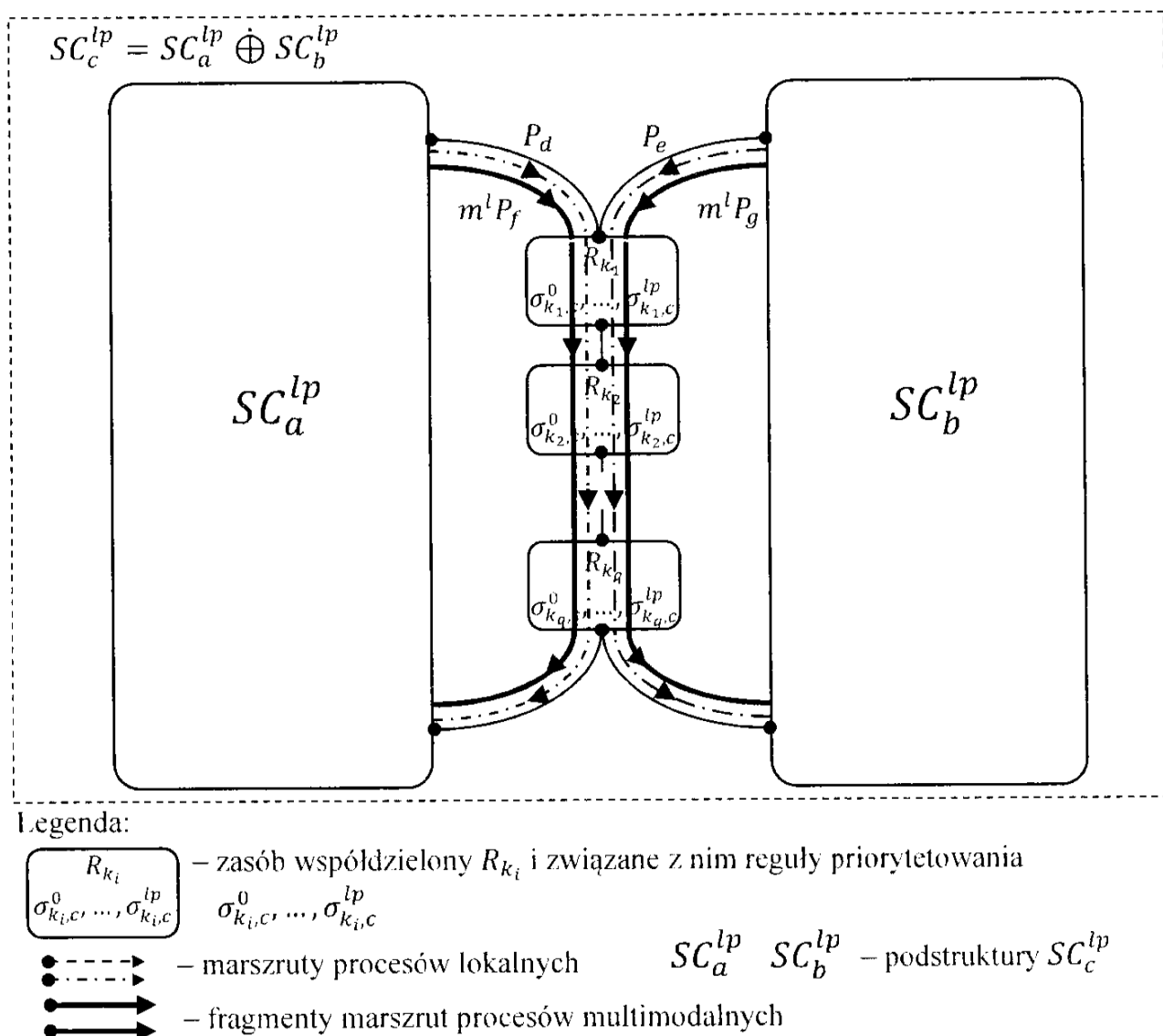
Rys. 4.3. Przykład składania dwóch podstruktur SWMPC:  $SC_3^1 = SC_1^1 \oplus SC_2^1$

#### 4.1.1. Warunki składania struktur SWMPC

Różnorodność wariantów operatorów składania struktur SWMPC, określonych na ciele struktur systemów cyklicznych, warunkowana jest ograniczeniami narzucanymi na definiującą je funkcję  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$ .

Jako pierwszy przykład formułowania ograniczeń narzucanych  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  rozważmy przypadek składania dwóch podstruktur wzdłuż łańcucha zasobów:  $Lk = (R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q})$ . Dwie podstruktury, reprezentujące systemy o zachowa-

niach cyklicznych z rys. 4.4, składane są po zasobach łańcucha  $Lk$ . Kolejność zasobów występujących w łańcuchu odpowiada kierunkom (zwrotom) marszrut składanych procesów, na wszystkich zasobach łańcucha obowiązują te same reguły priorytetowania:  $\sigma_{k_1,c}^l = \sigma_{k_2,c}^l = \dots = \sigma_{k_i,c}^l = \dots = \sigma_{k_q,c}^l$ , dla  $l = 1 \dots lp$ .



Rys. 4.4. Składanie podstruktur  $SC_a^{lp} \oplus SC_b^{lp}$  wzdłuż łańcucha zasobów wykorzystanych w procesach o tym samym zwrocie

Zgodnie z warunkami koniecznymi istnienia blokady (patrz podrozdział 1.2) do blokady nie dochodzi jeżeli w trakcie realizacji procesów nie powstanie zamknięty łańcuch żądań zasobowych (np. rys. 1.19a i 1.19b). W systemach o podstrukturach  $SC_a^{lp}$ ,  $SC_b^{lp}$  taki łańcuch nie występuje (podstruktury generują cykliczne zachowania). W tym ujęciu przyjęty sposób ich składania wzdłuż jednego wspólnego łańcucha również nie może doprowadzić do wystąpienia cyklu żądań zasobowych w systemie o strukturze wynikowej. Ponadto, przyjęcie tych samych reguł priorytetowania dla wszystkich zasobów, na których dochodzi do złożenia podstruktur powoduje, że na zasobach należących do łańcucha  $Lk$ , gdzie realizacja procesów realizowanych wzdłuż różnych struktur jest naprzemienna, również nie dochodzi do blokady.

Skoro wiadomo, że w tego typu podstrukturach nie dochodzi do blokady to powstaje pytanie: jaką postać przyjmuje funkcja  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  wykorzystana przy skła-

daniu  $SC_a^{lp} \oplus SC_b^{lp}$  z rys. 4.4? W celu rozróżnienia, operator składania wykorzystujący zasadę łączenia wzdłuż jednego wspólnego łańcucha zasobów, jest dalej oznaczony symbolem  $\oplus$ .

Naprzemienną, wzdłuż łańcucha  $Lk$ , realizację procesów zapewniają różne postaci funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$ , m.in. poniższa:

$$\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l) = (s_{k,1,a}^l, \dots, s_{k,lh_{a,a}}^l, s_{k,1,b}^l, \dots, s_{k,lh_{b,b}}^l), \quad (105)$$

gdzie:

$\sigma_{k,a}^l = (s_{k,1,a}^l, \dots, s_{k,lh_{a,a}}^l)$  – reguła przypisana do zasobu  $R_k$  podstruktury  $SC_a^{lp}$  przed składaniem  $SC_a^{lp} \oplus SC_b^{lp}$ ,

$\sigma_{k,b}^l = (s_{k,1,b}^l, \dots, s_{k,lh_{b,b}}^l)$  – reguła przypisana do zasobu  $R_k$  podstruktury  $SC_b^{lp}$  przed składaniem  $SC_a^{lp} \oplus SC_b^{lp}$ .

W przyjętej definicji funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  (105) zakłada się, że wzdłuż łańcucha  $Lk$  w pierwszej kolejności wykonywany jest proces podstruktury  $SC_a^{lp}$ , a następnie proces podstruktury  $SC_b^{lp}$ . Ograniczenie to determinuje warunki składania podstruktur systemów cyklicznych:

#### Twierdzenie 6

Dane są podstruktury  $SC_a^{lp}, SC_b^{lp} \in SC^*$  (101) systemów cyklicznych, składane po zbiorze zasobów  $Rk_{a,b} = Rp_a \cap Rp_b = \{R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q}\}$  występujących w łańcuchu  $Lk = (R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q})$ . Jeżeli poniższe warunki są spełnione, to system o strukturze  $SC_c^{lp} = SC_a^{lp} \oplus SC_b^{lp}$  jest systemem cyklicznym.

- i) każdy proces lokalny  $P_i$  (multimodalny  $m^l P_i, l = 1 \dots lp$ ) występujący w  $SC_a^{lp}$  lub  $SC_b^{lp}$ , który realizuje operacje na zasobie  $R_c \in Rk$ , realizuje również operacje na pozostałych zasobach  $Rk$  zgodnie z porządkiem (marszrutą) określonym przez  $Lk = (R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q})$ ,
- ii) funkcja  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  spełnia warunki definicji (105).

Dowód: Jeżeli w SWMPC o zadanych podstrukturach  $SC_a^{lp}, SC_b^{lp}$  osiągalne są zachowania cykliczne to zgodnie z Twierdzeniem 5 zachowania te mogą być reprezentowane przez acykliczne digrafy pierwszeństwa operacji:  $G_a, G_b$ . System o strukturze otrzymanej z  $SC_a^{lp} \oplus SC_b^{lp}$  charakteryzuje zachowanie opisane przez digraf  $G_c = G_a \dot{\cup} G_b = (\mathbb{V}_a \cup \mathbb{V}_b, \mathbb{E}_a \cup \mathbb{E}_b)$ , będący sumą (Ross i Wrigh 2008) digrafów  $G_a \dot{\cup} G_b$ . Digrafy  $G_a, G_b$  są sumowane po wspólnej ścieżce utworzonej z wierzchołków związanych z operacjami realizowanymi kolejno na zasobach  $Lk = (R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q})$ . Suma dwóch acyklicznych digrafów po wspólnej ścieżce, jest również digrafem acyklicznym. Acykliczność digrafu  $G_c$  oznacza, że w systemie o strukturze  $SC_c^{lp} = SC_a^{lp} \oplus SC_b^{lp}$  osiągalne jest zachowanie cykliczne – co kończy dowód.

□

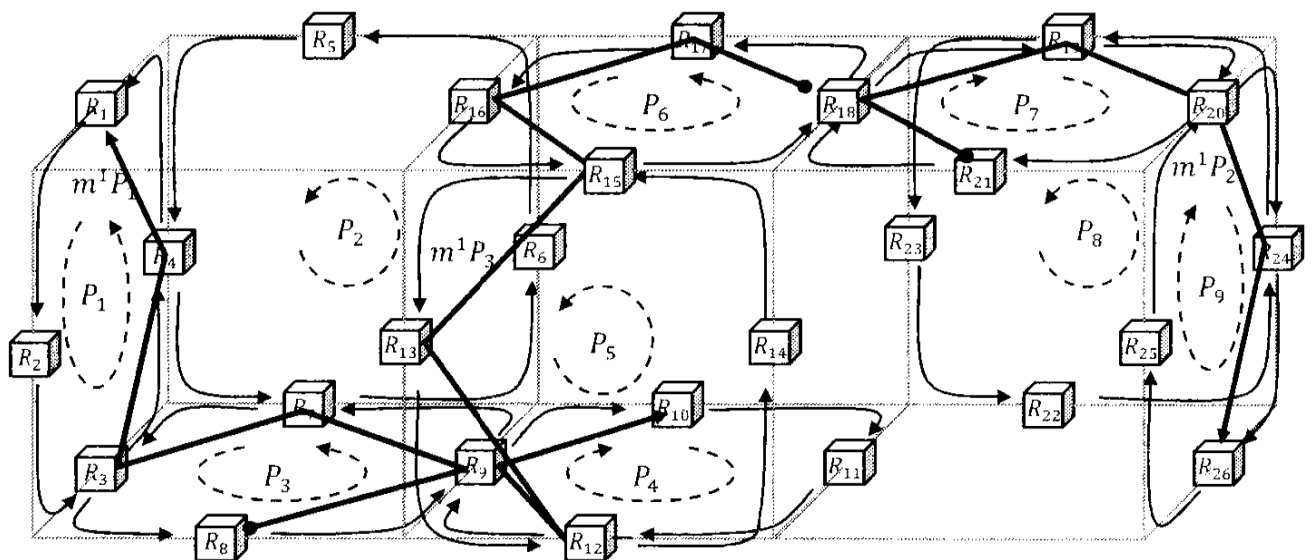
Oznacza to, że przyjęte w definicji (105) ograniczenie pozwala wyznaczyć warunki Twierdzenia 6 (obejmujące podstruktury składane wzdłuż wspólnego łańcucha zasobów  $Lk$ ) gwarantujące, że zbiór podstruktur systemów cyklicznych jest zbiorem zamkniętym ze względu na operator  $\oplus$ .

Przydatność tego twierdzenia w sytuacjach związanych z syntezą struktur SWMPC ilustruje poniższy przykład.

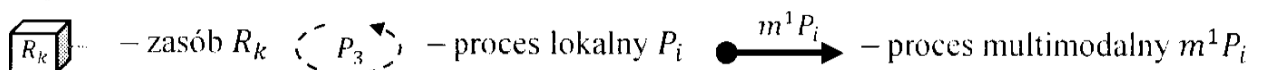
**Przykład 7.** Rozwiązanie problemu syntezy systemu SWMPC metodą składania podstruktur uzyskanych w wyniku dekompozycji jego struktury.

Dany jest SWMPC jak na rys. 4.5 o wyróżnionych dwóch poziomach zachowań:

- poziom procesów lokalnych, zawierający 9 procesów jednopotokowych:  $P_1 - P_9$ ,
- poziom procesów multimodalnych, zawierający 3 procesy jednopotokowe:  $m^1P_1, m^1P_2, m^1P_3$ .



Legenda:



**Rys. 4.5.** SWMPC z przykładu 7

System o strukturze  $SC^1$  należy do klasy  $MW_{3,1}$  (tab. 2.1). Znale są marszruty procesów oraz czasy wykonania operacji (tab. 4.1). Czasy wykonania operacji procesów multimodalnych są takie same jak czasy wymaganych do ich realizacji operacji procesów lokalnych. Poszukiwana jest odpowiedź na pytanie:

*Czy istnieje taka postać reguł priorytetowania  $\theta^0, \theta^1$ , która gwarantuje osiągalność cyklicznego przebiegu ustalonego SWMPC?*

Analogicznie jak w Algorytmie 6 rozwiązywane są kolejno problemy  $PS'_{REX}$  (98) oraz  $m^1PS'_{REX}$  (99), w ramach których poszukiwane są postacie struktur poziomów procesów lokalnych  $SL$  i multimodalnych  $SM^1$ . Poszukiwanie tego typu struktur sprowadza się do określenia postaci reguł priorytetowania  $\theta^0, \theta^1$  – znane bowiem są pozostałe wielkości charakteryzujące struktury (takie jak: marszruty, czasy wykonania operacji itp.).



Tabela 4.1. Czasy wykonania operacji procesów lokalnych SWMPC z rys. 4.5

	$i$	$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$		$i$	$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$
$P_1$	1	1	2	1	2	$P_6$	6	2	1	4	1
$P_2$	2	3	2	2	3	$P_7$	7	1	2	3	4
$P_3$	3	3	1	4	1	$P_8$	8	1	2	3	3
$P_4$	4	1	3	2	4	$P_9$	9	2	4	1	1
$P_5$	5	3	2	1	3	-	-	-	-	-	-

W strukturze z rys. 4.5 występuje 26 zasobów z czego: 10 jest współdzielonych przez 2 procesy lokalne, a 2 zasoby współdzielone przez 2 procesy multimodalne. Przyjmując, że każda reguła, zasobu współdzielonego, jest sekwencją dwuelementową (zakłada się wzajemną liczbę wzajemnych wykonań procesów równą 1), otrzymuje się  $2^{12}$  potencjalnych wariantów reguł priorytetowania  $\theta$  ( $2^{10}$  wariantów reguł  $\theta^0$  i  $2^2$  wariantów reguł  $\theta^1$ ). Oznacza to, że w najgorszym przypadku rozwiązanie  $PS'_{REX}$  (98) oraz  $m^1PS'_{REX}$  wymagać będzie 4096 kroków.

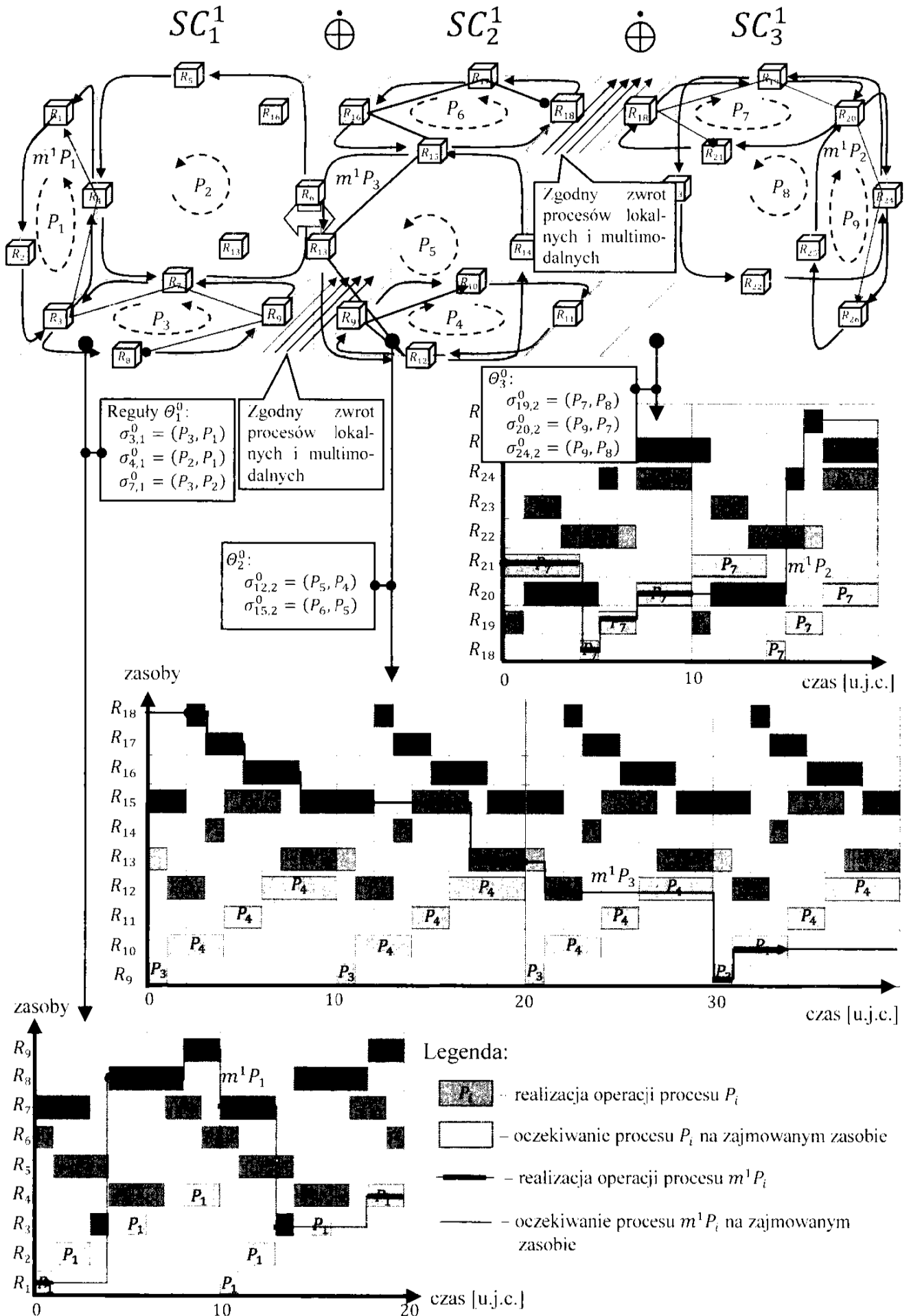
Rozmiar przeszukiwanej przestrzeni można ograniczyć jeszcze przed rozpoczęciem obliczeń, dokonując dekompozycji  $SC^1$  na podstruktury. W rozważanym przypadku struktura  $SC^1$  zdekomponowana została na 3 podstruktury:  $SC^* = \{SC_1^2, SC_2^2, SC_3^2\}$  (patrz rys. 4.6). Składanie podstruktur  $SC^*$  odbywa się na wspólnych zasobach  $R_9$  i  $R_{18}$  zgodnie z warunkami Twierdzenia 6: składanie kolejnych dwóch podstruktur odbywa się wzdłuż jednego łańcucha (w tym przypadku jednoelementowego), zwroty procesów na wspólnych zasobach są zgodne.

Spełnienie tych warunków pozwala rozpocząć poszukiwanie reguł priorytetowania gwarantujących cykliczną pracę systemu poprzez analizę każdej podstruktury, niezależnie. Jeżeli istnieją takie reguły, dla których zachowanie systemów podstruktur  $SC_1^1$ ,  $SC_2^1$ ,  $SC_3^1$  jest cykliczne, to zgodnie z Twierdzeniem 6 również SWMPC o strukturze  $SC^1 = (SC_1^1 \oplus SC_2^1) \oplus SC_3^1$  jest systemem cyklicznym.

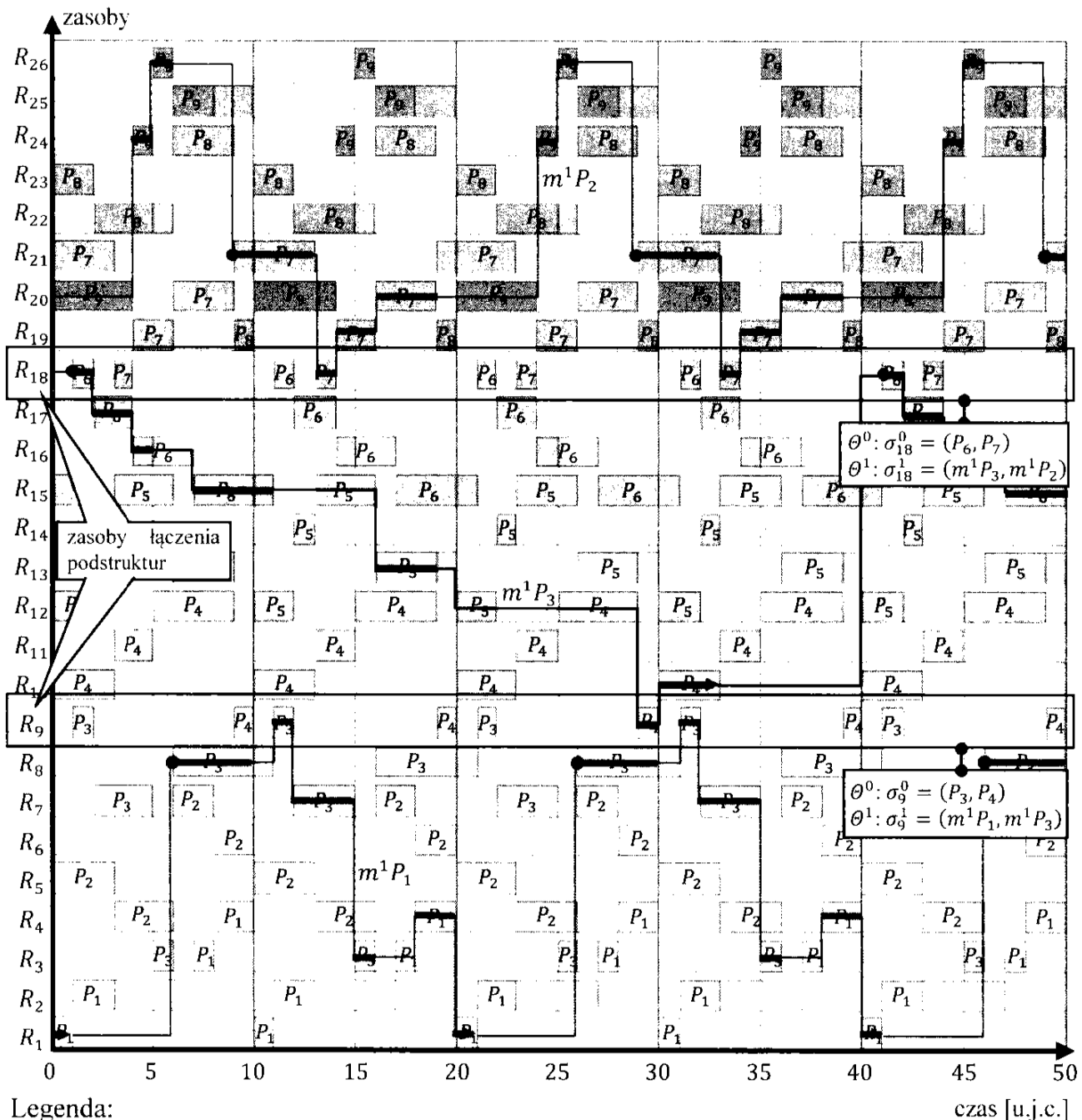
Do wyznaczenia reguł priorytetowania  $\theta_1^0, \theta_1^1; \theta_2^0, \theta_2^1; \theta_3^0, \theta_3^1$  odpowiednio podstruktur  $SC_1^1, SC_2^1, SC_3^1$  wykorzystany został Algorytm 6, który w rozważanym przypadku wymaga trzykrotnego rozwiązania problemów  $PS'_{REX}$  (98) i  $m^1PS'_{REX}$  (99).

Ze względu na możliwość analizy każdej podstruktury niezależnie wyznaczenie reguł priorytetowania charakteryzuje się znacznie mniejszą liczbą potencjalnych wariantów:  $2^3 + 2^2 + 2^3 = 20$  (tak jak poprzednio przyjmuje się liczbę wzajemnych wykonań procesów równą 1).

Otrzymane zbiory reguł priorytetowania wraz z odpowiadającymi im harmonogramami cyklicznymi przedstawia rys. 4.6. Każda podstruktura generuje zachowanie cykliczne. Oznacza to, że jeżeli na zasobach  $R_9$  i  $R_{18}$  przyjęte zostaną reguły zgodne z funkcją (105), to w systemie o strukturze otrzymanej w wyniku złożenia  $(SC_1^1 \oplus SC_2^1) \oplus SC_3^1$  osiągalne jest również zachowanie cykliczne.



Rys. 4.6. Podstruktury systemu z rys. 4.5 wraz z odpowiadającymi im harmonogramami cyklicznymi



Legenda:

- realizacja operacji procesu  $P_i$
- oczekiwanie procesu  $P_i$  na zajmowanym zasobie
- realizacja operacji procesu  $m^1P_i$
- oczekiwanie procesu  $m^1P_i$  na zajmowanym zasobie

Rys. 4.7. Harmonogram cykliczny realizacji procesów systemu z rys. 4.5

Ostatecznie wyznaczone, zgodnie z definicją (105), reguły priorytetowania zasobów  $R_9$  i  $R_{18}$  przyjmują postać:

$$\sigma_9^0 = (P_3, P_4), \sigma_9^1 = (m^1P_1, m^1P_3), \sigma_{18}^0 = (P_6, P_7), \sigma_{18}^1 = (m^1P_3, m^1P_2) \quad (106)$$

Harmonogram cykliczny realizacji procesów (patrz rys. 4.7) odpowiadający cyklicznemu przebiegowi ustalonemu systemu (zastosowano reguły z rys. 4.6 i (106)) został wyznaczony przez Algorytm 3 (patrz podrozdział 3.2.3).



Przedstawiony wyżej przykład ilustruje możliwości ograniczania rozmiaru przestrzeni przeszukiwań – w wyniku składania podstruktur uzyskanych w wyniku wcześniej dokonanej dekompozycji struktury docelowej. W rozważanym przypadku, liczba wariantów reguł priorytetowania (koniecznych do weryfikacji) zmniejszyła się z 4096 do 20 co skutkuje 99% oszczędnością.

W przypadkach składania par podstruktur wzdłuż wspólnego łańcucha zasobów, ale nie spełniających wymagań zgodności zwrotów marszrut (patrz rys. 4.2), poszukiwana jest inna postać operatora  $\oplus$ .

#### 4.1.2. Reguła „zamka błyskawicznego”

Przedstawiony wyżej sposób składania podstruktur  $SC^{lp}$  wykorzystuje specyficzne cechy struktury związane z wymaganiem zgodności zwrotów marszrut składanych procesów. Alternatywne, akcentujące specyficzne cechy zachowań osiągalnych w składanych podstrukturach  $SC_a^{lp}$ ,  $SC_b^{lp}$ , warunki wiążą się z tzw. regułą „zamka błyskawicznego”.

Na rys. 4.8 przedstawiono dwie podstruktury  $SC_a^{lp}$  i  $SC_b^{lp}$  składane poprzez zbiór wspólnych zasobów:  $Rk_{a,b} = Rp_a \cap Rp_b = \{R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q}\}$ . W celu różniczenia, operator składający wykorzystujący regułę „zamka błyskawicznego”, jest dalej oznaczony symbolem  $\oplus$ . W odróżnieniu od poprzedniej sytuacji (rys. 4.4), w rozważanym przypadku zbiór zasobów nie tworzy jednego wspólnego łańcucha. W ogólności może istnieć wiele łańcuchów o dowolnym zwrocie marszrut procesów. Zakłada się, że zachowania cykliczne osiągalne w systemach o podstrukturach  $SC_a^{lp}$  i  $SC_b^{lp}$  są reprezentowane przez harmonogramy cykliczne  $X^{lp}$  (95) (stanowiące rozwiązanie problemów  $PS'_X$  (89) oraz  $m^l PS'_X$  (93)).

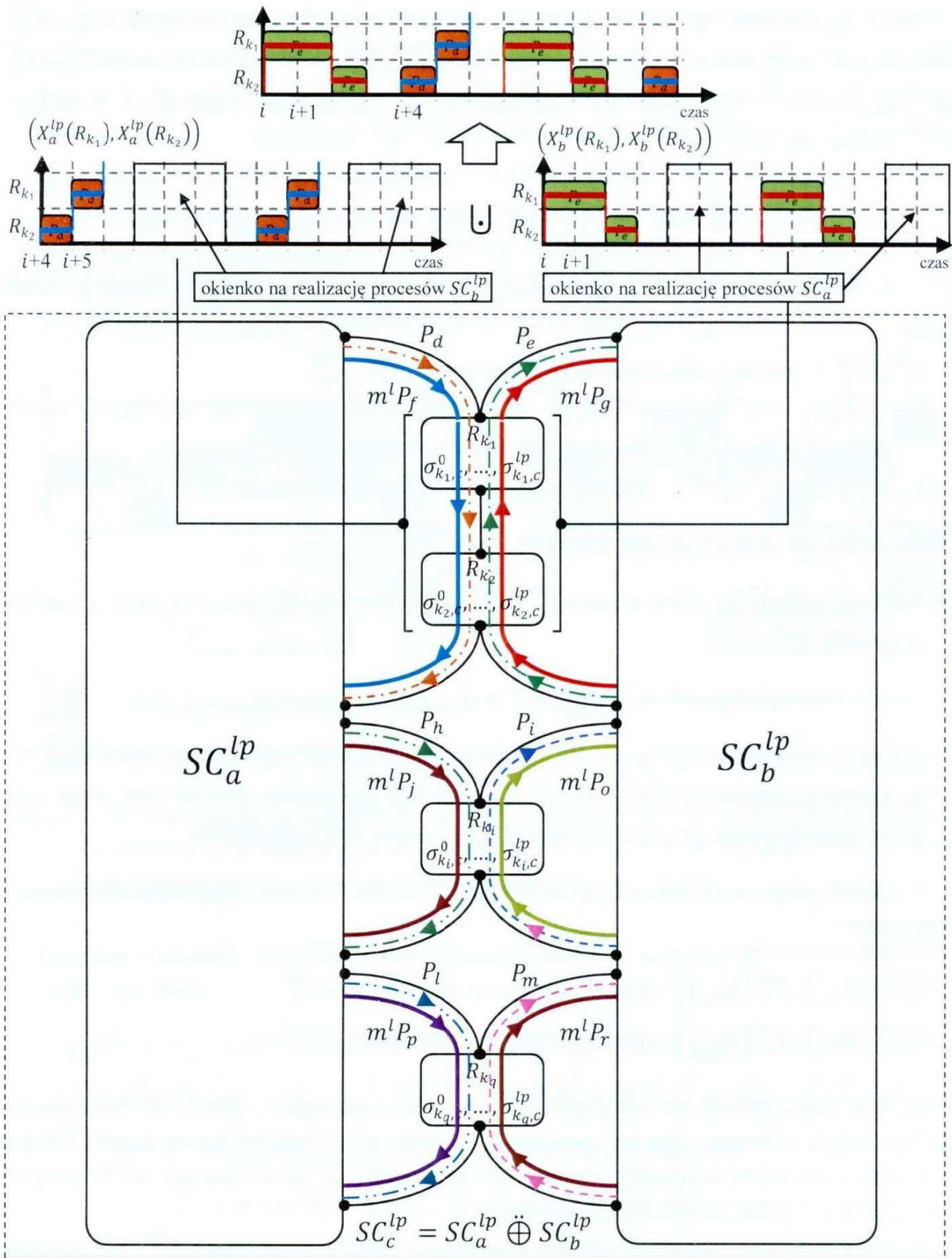
W celu wyjaśnienia ww. reguły „zamka błyskawicznego” wprowadza się pojęcie harmonogramu  $X^{lp}(R_k)$  opisującego zachowanie procesów na pojedynczym zasobie  $R_k$ :

$$X^{lp}(R_k) = \left( \left( \left( (X'_{DC}(R_k), \alpha), (m^1 X'_{DC}(R_k), m^1 \alpha) \right) \dots \right) \dots (m^{lp} X'_{DC}(R_k), m^{lp} \alpha) \right), \quad (107)$$

gdzie:  $X'_{DC}(R_k), m^1 X'_{DC}(R_k), \dots, m^{lp} X'_{DC}(R_k)$  – określają odpowiednio zbiory momentów rozpoczęcia operacji procesów lokalnych i multimodalnych;  $X'_{DC}(R_k) \subset X'_{DC}, m^1 X'_{DC}(R_k) \subset m^1 X'_{DC}, \dots, m^{lp} X'_{DC}(R_k) \subset m^{lp} X'_{DC}$ ,

Harmonogram  $X^{lp}(R_k)$  (dla  $R_k \in R$ ) należy traktować jako fragment harmonogramu  $X^{lp}$  (95) zawierający momenty rozpoczęcia wszystkich operacji realizowanych na zasobie  $R_k$ . Przykłady takich harmonogramów przedstawia rys. 4.8.

W tym ujęciu podstrukturom  $SC_a^{lp}$  i  $SC_b^{lp}$  (rys. 4.8) odpowiadają cykliczne harmonogramy  $X_a^{lp}, X_b^{lp}$  i tym samym harmonogramy związane z zasobami współdzielonymi:  $X_a^{lp}(R_{k_i}), X_b^{lp}(R_{k_i}), R_{k_i} \in Rk_{a,b}$ .



Legenda:

- $\begin{matrix} R_{k_i} \\ \sigma_{k_i,c}^0, \dots, \sigma_{k_i,c}^{lp} \end{matrix}$  – zasób współdzielony  $R_k$  z regułami priorytetowania  $\sigma_{k_i,c}^0, \dots, \sigma_{k_i,c}^{lp}$
- $\begin{matrix} \text{---} \\ \text{---} \end{matrix}$  – marszrut procesów lokalnych
- $\begin{matrix} \text{---} \\ \text{---} \end{matrix}$  – fragmenty marszrut procesów multimodalnych
- $SC_a^{lp} \quad SC_b^{lp}$  – podstruktury  $SC_c^{lp}$

Rys. 4.8. Składanie struktur według reguły „zamka błyskawicznego”

Niech  $\cup$  oznacza **operator łączenia harmonogramów cyklicznych**  $X_a^{lp}$ ,  $X_b^{lp}$ , osiągalnych w systemach o podstrukturach  $SC_a^{lp}$ ,  $SC_b^{lp}$ . Wynikiem łączenia  $X_a^{lp} \cup X_b^{lp}$  jest harmonogram cykliczny  $X_c^{lp}$  reprezentujący zachowanie systemu o strukturze  $SC_c^{lp} = SC_a^{lp} \dot{\cup} SC_b^{lp}$ :

$$X_c^{lp} = X_a^{lp} \cup X_b^{lp} = \left( \left( \left( X'_{DC_a} \cup X'_{DC_b}, nww(\alpha_a, \alpha_b) \right), \dots \right), \dots, \left( m^{lp} X'_{DC_a} \cup m^{lp} X'_{DC_b}, nww(m^{lp} \alpha_a, m^{lp} \alpha_b) \right) \right), \quad (108)$$

gdzie:

$X_a^{lp} \cup X_b^{lp}$  – operacja łączenia harmonogramów  $X_a^{lp}$ ,  $X_b^{lp}$ ,  
 $X'_{DC_a}$ ,  $X'_{DC_b}$  i  $m^{lp} X'_{DC_a}$ ,  $m^{lp} X'_{DC_b}$  – zbiory momentów rozpoczęcia operacji procesów lokalnych i multimodalnych podstruktur  $SC_a^{lp}$  i  $SC_b^{lp}$ ,  
 $\alpha_a$ ,  $\alpha_b$  i  $m^{lp} \alpha_a$ ,  $m^{lp} \alpha_b$  – okresy harmonogramów cyklicznych  $X_a^{lp}$ ,  $X_b^{lp}$ ,

jeżeli spełnione są następujące warunki:

(a) wartość okresu harmonogramu  $X_a^{lp}$  jest całkowitą wielokrotnością okresu harmonogramu  $X_b^{lp}$ :

$$\text{mod}(m^l \alpha_a, m^l \alpha_b) = 0 \text{ dla } l = 1 \dots lp; \text{ oraz } \text{mod}(\alpha_a, \alpha_b) = 0,$$

(b) operacje zasobów wspólnych  $Rk_{a,b}$  realizowane są bez wzajemnych zakłóceń, tzn. w harmonogramach  $X_a^{lp}(R_{k_i})$ ,  $X_b^{lp}(R_{k_i})$  istnieją cyklicznie powtarzane okna czasowe umożliwiające realizację operacji procesów obu podstruktur.

Przykład połączenia harmonogramów przedstawia rys. 4.8. Operacje obu harmonogramów:

$$\left( X_a^{lp}(R_{k_1}), X_a^{lp}(R_{k_2}) \right) \text{ – harmonogram podstruktury } SC_a^{lp},$$

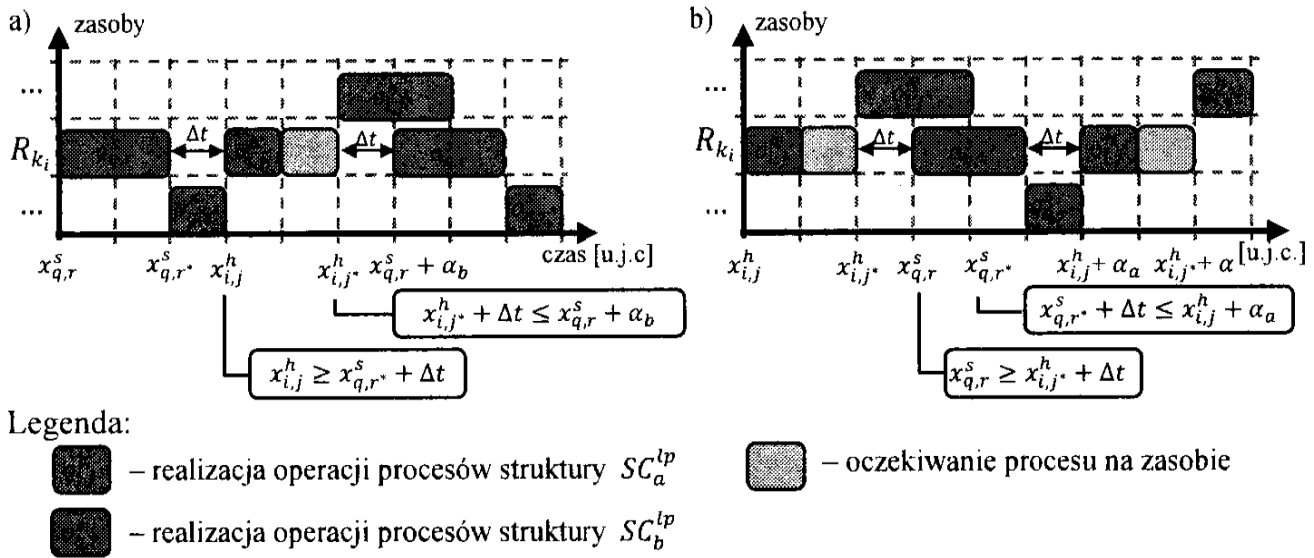
$$\left( X_b^{lp}(R_{k_1}), X_b^{lp}(R_{k_2}) \right) \text{ – harmonogram podstruktury } SC_b^{lp},$$

mogą być realizowane na zasobach  $R_{k_1}$  i  $R_{k_2}$  naprzemiennie, dzięki powtarzającym się, ze stałym okresem, oknom czasowym, w których te zasoby nie są zajęte. Można powiedzieć, że harmonogramy wzajemnie się „zazębiają” podobnie jak ma to miejsce w przypadku zębów zamka błyskawicznego.

**Inaczej mówiąc, jeżeli przebiegi harmonogramów zachowań  $X_a^{lp}$ ,  $X_b^{lp}$  generowanych przez podstruktury  $SC_a^{lp}$  i  $SC_b^{lp}$  na wszystkich współdzielonych zasobach  $Rk_{a,b}$ , nie kolidują ze sobą, tzn. wzajemnie się „zazębiają”, to istnieje harmonogram cykliczny  $X_c^{lp}$  przebiegu zachowania struktury  $SC_c^{lp} = SC_a^{lp} \dot{\cup} SC_b^{lp}$ , będący wynikiem połączenia harmonogramów  $X_a^{lp} \cup X_b^{lp}$  (108).**

Warunki (b) gwarantujące brak kolizji procesów na wspólnych zasobach składanych podstruktur procesów klasy SWMPC są definiowane następująco:

- **Poziom operacji procesów lokalnych.** Aby zagwarantować bezkolizyjną realizację procesów na zasobie  $R_{k_i} \in Rk_{a,b}$  przyjmuje się, że operacje  $o_{i,j}^h, o_{q,r}^s$  nie kolidują ze sobą (na współdzielonym zasobie  $R_{k_i}$ ) jeżeli operacja  $o_{i,j}^h$  rozpoczyna się (moment  $x_{i,j}^h$ ) po zwolnieniu (ze zwłoką  $\Delta t$ ) zasobu przez operację  $o_{q,r}^s$  (moment  $x_{q,r}^s$  rozpoczęcia operacji następnej) i zwalnia zasób (moment  $x_{i,j}^h$  rozpoczęcia operacji następnej) przed rozpoczęciem kolejnego wykonania operacji  $o_{q,r}^s$  (moment  $x_{q,r}^s + \alpha_b$ ) (Bocewicz et al. 2009c). Sytuację tego typu ilustruje rys. 4.9a. Oczywiście, dopuszczalna jest również sytuacja odwrotna gdy to operacja  $o_{q,r}^s$  realizowana jest pomiędzy kolejnymi wykonaniami operacji  $o_{i,j}^h$  – rys. 4.9b).



**Rys. 4.9.** Harmonogramy bezkolizyjnej realizacji operacji procesów lokalnych na wspólnym zasobie  $R_{k_i}$ : realizacja operacji  $o_{i,j}^h$  pomiędzy kolejnymi wykonaniami  $o_{q,r}^s$  a), realizacja operacji  $o_{q,r}^s$  pomiędzy kolejnymi wykonaniami  $o_{i,j}^h$  b)

Poniższy warunek implikuje bezkolizyjną realizację operacji procesów lokalnych (patrz rys. 4.9):

$$\left[ (x_{i,j}^h \geq x_{q,r}^s + k'' \cdot \alpha_b + \Delta t) \wedge (x_{i,j}^h + k' \cdot \alpha_a + \Delta t \leq x_{q,r}^s + \alpha_b) \right] \vee \left[ (x_{q,r}^s \geq x_{i,j}^h + k' \cdot \alpha_a + \Delta t) \wedge (x_{q,r}^s + k'' \cdot \alpha_b + \Delta t \leq x_{i,j}^h + \alpha_a) \right] \quad (109)$$

$$\forall x_{i,j}^h \in X'_{DC_a}(R_{k_i}), \forall x_{q,r}^s \in X'_{DC_b}(R_{k_i}), \forall R_{k_i} \in Rk,$$

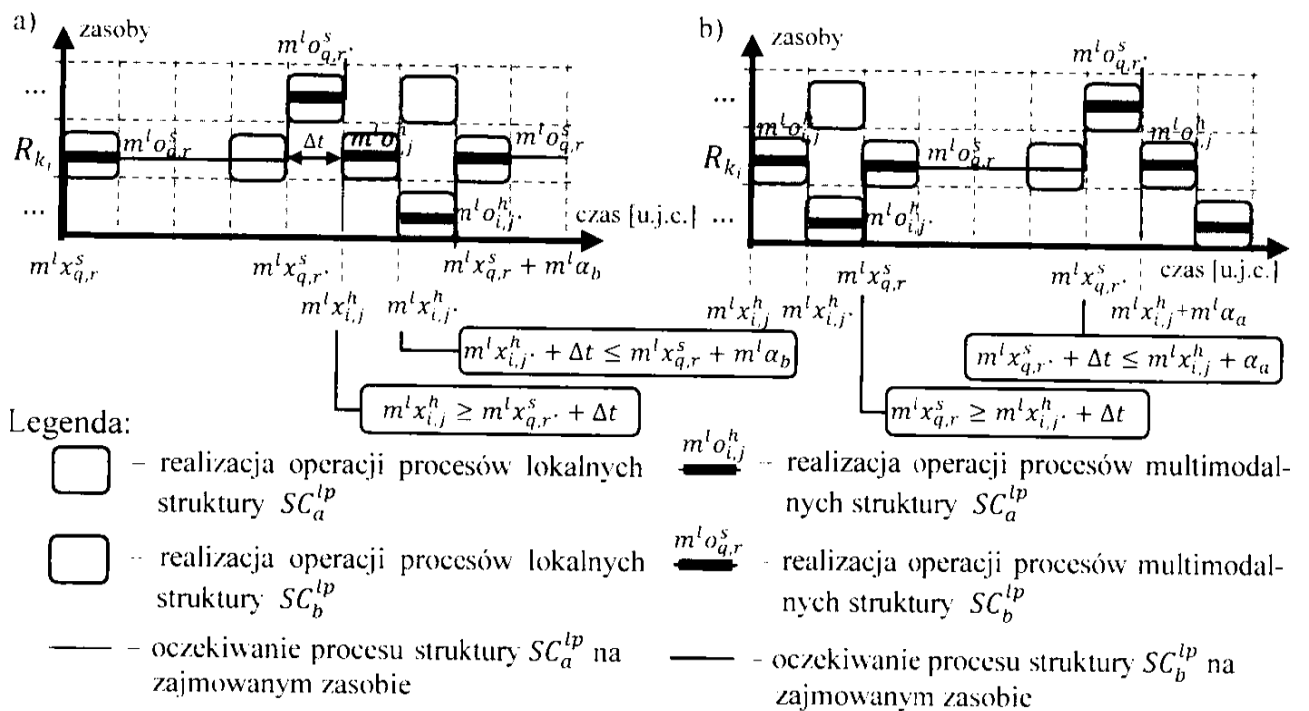
$$j^* = \text{modl}((j+1), lr(i)), r^* = \text{modl}((r+1), lr(q)),$$

$$k' = \begin{cases} 0 & \text{gdy } j+1 > lr(i) \\ 1 & \text{gdy } j+1 \leq lr(i) \end{cases} \quad i \quad k'' = \begin{cases} 0 & \text{gdy } r+1 > lr(q) \\ 1 & \text{gdy } r+1 \leq lr(q) \end{cases}$$

gdzie:  $\alpha_a, \alpha_b$  – okresy harmonogramów  $X'_{DC_a}(R_k), X'_{DC_b}(R_k)$ ;  $lr(i), lr(q)$  – długości marszrut procesów  $P_i, P_q$ ;  $x_{i,j}^h, x_{q,r}^s$  – momenty rozpoczęcia operacji  $o_{i,j}^h, o_{q,r}^s$  procesów podstruktur  $SC_a^{lp}, SC_b^{lp}$ ;  $x_{i,j}^h, x_{q,r}^s$  – momenty rozpoczęcia operacji realizowanych po  $o_{i,j}^h, o_{q,r}^s$ ;  $\text{modl}(a,b)$  – funkcja zwracająca wartość  $r = \text{modl}(a,b)$ , gdy  $a = k \cdot b + r$  i  $0 < r \leq |b|, a,b,c,k \in \mathbb{C}$ .

Sposób realizacji operacji  $o_{i,j}^h, o_{q,r}^s$  spełniających warunek (109) ilustruje rys. 4.9. Jego spełnienie oznacza, że na każdym wspólnym zasobie składanych podstruktur  $SC_a^{lp}, SC_b^{lp}$  procesy lokalne są realizowane naprzemiennie – tzn. są bezkolizyjne.

- **Poziom operacji procesów multimodalnych.** Aby zagwarantować bezkolizyjną realizację procesów multimodalnych (synchronizowanych protokołem wzajemnego wykluczania) przyjmuje się podobnie jak w przypadku procesów lokalnych, że operacje  $m^l o_{i,j}^h, m^l o_{q,r}^s$  współdzielące zasób  $R_{k_i} \in Rk_{a,b}$  są wykonywane bezkonfliktowo, jeżeli jedna z nich wykonywana jest pomiędzy kolejnymi realizacjami drugiej (patrz rys. 4.10).



**Rys. 4.10.** Harmonogramy bezkolizyjnej realizacji operacji procesów multimodalnych na wspólnym zasobie  $R_{k_i}$ : realizacja operacji  $m^l o_{i,j}^h$  pomiędzy kolejnymi wykonaniami  $m^l o_{q,r}^s$  a), realizacja operacji  $m^l o_{q,r}^s$  pomiędzy kolejnymi wykonaniami  $m^l o_{i,j}^h$  b)

Poniższy warunek implikuje bezkolizyjną realizację operacji procesów multimodalnych (patrz rys. 4.10):

$$\begin{aligned} & [(m^l x_{i,j}^h \geq m^l x_{q,r}^s + k'' \cdot m^l \alpha_b + \Delta t) \wedge (m^l x_{i,j}^h + k' \cdot m^l \alpha_a + \Delta t \leq m^l x_{q,r}^s + m^l \alpha_b)] \vee \\ & \vee [(m^l x_{q,r}^s \geq m^l x_{i,j}^h + k' \cdot m^l \alpha_a + \Delta t) \wedge (m^l x_{q,r}^s + k'' \cdot m^l \alpha_b + \Delta t \leq m^l x_{i,j}^h + m^l \alpha_a)] \quad (110) \end{aligned}$$

$$\forall m^l x_{i,j}^h \in m^l X'_{DC_a}(R_{k_i}), \forall m^l x_{q,r}^s \in m^l X'_{DC_b}(R_{k_i}), \forall R_{k_i} \in Rk, l = 1 \dots lp,$$

$$j^* = \text{mod}l((j + 1), lm(i, l)), r^* = \text{mod}l((r + 1), lm(q, l)),$$

$$k' = \begin{cases} 0 & \text{gdy } j + 1 \leq lm(i, l) \\ 1 & \text{gdy } j + 1 > lm(i, l) \end{cases} \quad i \quad k'' = \begin{cases} 0 & \text{gdy } r + 1 \leq lm(q, l) \\ 1 & \text{gdy } r + 1 > lm(q, l) \end{cases}$$



gdzie:  $m^l \alpha_a, m^l \alpha_b$  – okresy harmonogramów  $m^l X'_{DC_a}(R_{k_i}), m^l X'_{DC_b}(R_{k_i})$ ;  $lm(u, l), lm(q, l)$  – długości marszrut procesów  $m^l P_i, m^l P_q$ ;  $m^l x_{i,j}^h, m^l x_{q,r}^s$  – momenty rozpoczęcia operacji  $m^l o_{i,j}^h, m^l o_{q,r}^s$  podstruktur  $SC_a^{lp}, SC_b^{lp}$ ;  $m^l x_{i,j}^{h*}, m^l x_{q,r}^{s*}$  – momenty rozpoczęcia operacji realizowanych po  $m^l o_{i,j}^h, m^l o_{q,r}^s$ ;  $modl(a, b)$  – funkcja zwracająca wartość  $r = modl(a, b)$  gdy  $a = k \cdot b + r$  i  $0 < r \leq |b|$ ,  $a, b, c, k \in \mathbb{C}$ .

**Spełnienie warunku (110) oznacza, że na każdym wspólnym zasobie składanych podstruktur  $SC_a^{lp}, SC_b^{lp}$  procesy multimodalne wszystkich poziomów zachowań ( $l = 1 \dots lp$ ) są realizowane naprzemiennie, tzn. bezkonfliktowo.**

Reasumując, spełnienie ograniczeń (109) i (110) warunkuje cykliczność systemów o strukturach składanych z podstruktur systemów cyklicznych. Łatwo zauważyć, że warunki te determinują jednocześnie postać reguł priorytetowania  $\sigma_{k,c}^l$  (104), a dokładniej postać funkcji  $\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l)$  definiowanej w oparciu o wartości momentów rozpoczęcia operacji realizowanych na zasobie  $R_k$ :

$$\vartheta(\sigma_{k,a}^l, \sigma_{k,b}^l) = (s_{k,1,c}^l, \dots, s_{k,j,c}^l, \dots, s_{k,lh_c,c}^l) \text{ gdy } x_{k,1,c}^l < \dots < x_{k,j,c}^l < \dots < x_{k,lh_c,c}^l, \quad (111)$$

$$l = 0 \dots lp,$$

gdzie:  $s_{k,j,c}^l$  –  $j$ -ty element reguły  $\sigma_{k,c}^l$  determinujący potok ( $l$ -tego poziomu zachowań), który rozpoczyna operację na zasobie  $R_k$  w momencie:  $x_{k,j,c}^l$ ;

$s_{k,j,c}^l$  jest jednym z elementów reguł  $\sigma_{k,a}^l, \sigma_{k,b}^l$ ;

$x_{k,j,c}^0 \in X'_{DC_a}(R_k) \cup X'_{DC_b}(R_k)$ ;  $x_{k,j,c}^l \in m^l X'_{DC_a}(R_k) \cup m^l X'_{DC_b}(R_k)$  dla  $l = 1 \dots lp$ .

Oznacza to, że kolejność występowania potoków w regule priorytetowania zasobu  $R_k$  jest determinowana przez kolejność wykonywania operacji tych potoków na tym zasobie (momenty rozpoczęcia operacji określone są przez harmonogramy  $X_a^{lp}, X_b^{lp}$  spełniające warunki (109) i (110)):

*Twierdzenie 7*

*Jeżeli podstruktury  $SC_a^{lp}, SC_b^{lp} \in SC^*$  (101) systemów o cyklicznych harmonogramach  $X_a^{lp}, X_b^{lp}$  składane po współdzielących je zasobach  $Rk_{a,b} = Rp_a \cap Rp_b = \{R_{k_1}, R_{k_2}, \dots, R_{k_i}, \dots, R_{k_q}\}$ , spełniają warunki (109), (110) i (111), to w systemie o strukturze  $SC_c^{lp} = SC_a^{lp} \ddot{\oplus} SC_b^{lp}$  osiągalny jest harmonogram cykliczny  $X_c^{lp} = X_a^{lp} \cup X_b^{lp}$  (108).*

**Dowód:** Operacje procesów podstruktur  $SC_a^{lp}, SC_b^{lp}$  realizowane w zgodzie z  $X_a^{lp}, X_b^{lp}$ , mogą wchodzić we wzajemne interakcje jedynie na zasobach wspólnych  $Rk_{a,b}$ . Jeżeli harmonogramy  $X_a^{lp}, X_b^{lp}$  są tak dobrane, że w przypadku realizacji operacji obu struktur jednocześnie, na zasobach  $Rk$  nie dochodzi do kolizji ani wymuszonych przestojów, (tzn. spełnione są warunki (109), (110)), to operacje tych harmonogramów wykonywane są naprzemiennie. Oznacza to, że harmonogramy  $X_a^{lp}, X_b^{lp}$  wza-

jemnie ze sobą nie kolidują. W systemie o strukturze  $SC_c^{lp}$  osiągalne jest więc zachowanie cykliczne reprezentowane przez harmonogram  $X_c^{lp}$  będący „złożeniem” harmonogramów  $X_a^{lp} \cup X_b^{lp}$  (108). □

Oznacza to, że przyjęte w definicji (111) ograniczenie pozwala wyznaczyć warunki Twierdzenia 7 (obejmującego harmonogramy cykliczne spełniające ograniczenia (109), (110)) gwarantujące, że zbiór podstruktur systemów cyklicznych jest zbiorem zamkniętym ze względu na operator  $\dot{\Theta}$ . Jego przydatność w sytuacjach związanych z iteracyjnym doбором parametrów podstruktur systemu SWMPC ilustruje poniższy przykład.

**Przykład 8.** Rozwiązanie problemu syntezy systemu SWMPC metodą składania podstruktur wg warunków Twierdzenia 7.

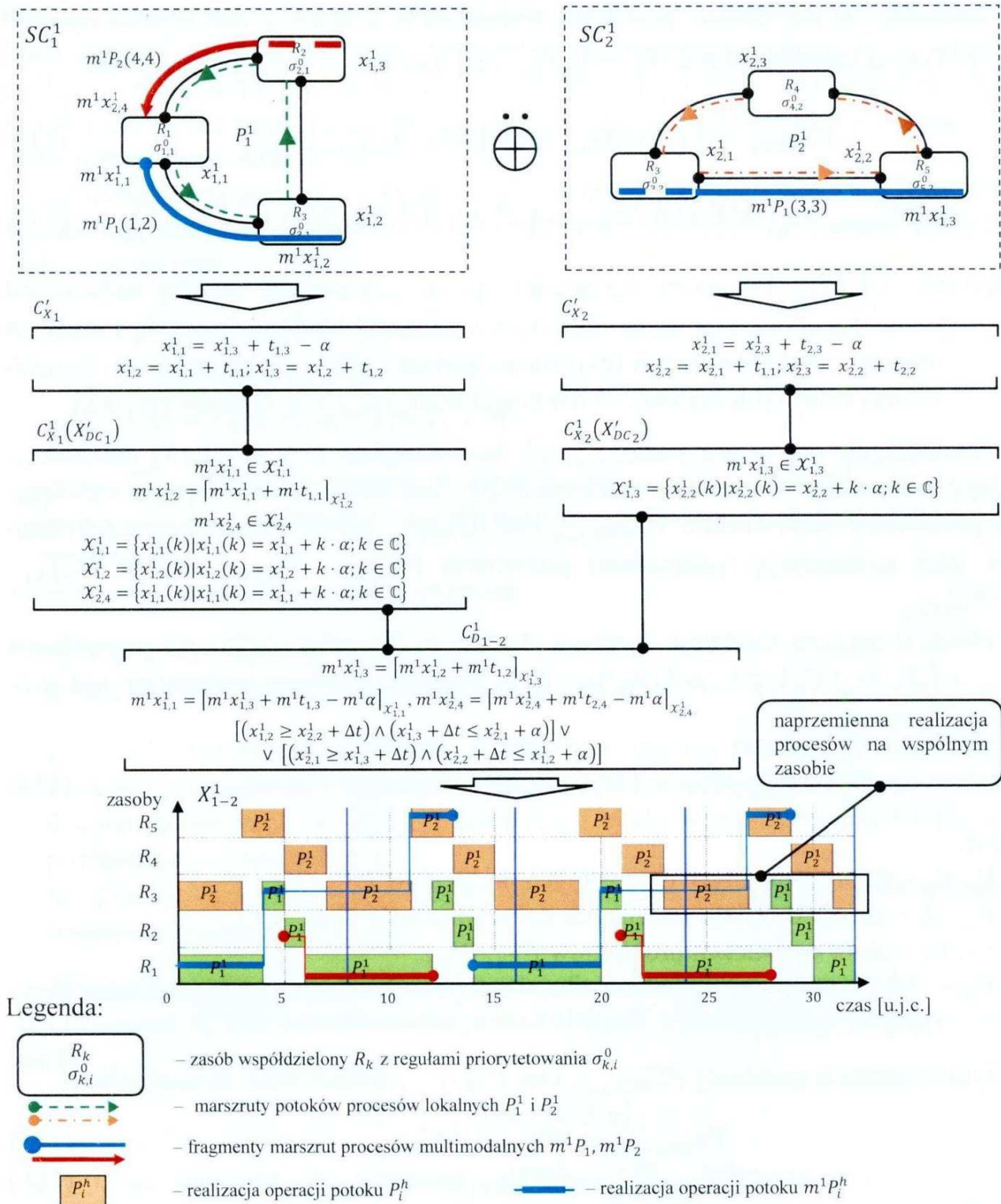
Dany jest SWMPC z rys. 4.2. Struktura systemu  $SC^1$  (typu  $MW_{4,1}$  (tab. 2.1)) dopuszcza jednocześnie współdzielenie zasobów przez procesy multimodalne (zbiór reguł priorytetowania poziomu procesów multimodalnych jest zbiorem pustym:  $\theta^1 = \emptyset$ ). Znane są marszruty procesów, nieznane są natomiast czasy wykonywania operacji oraz reguły priorytetowania procesów lokalnych  $\theta^0$ . Poszukiwana jest odpowiedź na pytanie:

*Czy istnieje zbiór reguł priorytetowania  $\theta^0$  oraz wartości czasów wykonywania operacji  $T$  gwarantujące istnienie cyklicznego przebiegu ustalonego w SWMPC z rys. 4.2?*

W celu uproszczenia rozważań przyjęto, że czasy wykonywania operacji procesów multimodalnych są takie same jak czasy operacji procesów lokalnych wymaganych do ich realizacji.

Jedno z możliwych rozwiązań, w sposób naturalny, kojarzy się z wykorzystaniem Algorytmu 6, w którym struktura  $SC^1$  traktowana jest jako jedna niepodzielna całość. Uzyskane w ten sposób rozwiązanie (tzn. rozwiązanie  $PS'_{REX}$  (98) oraz  $m^1 PS'_{REX}$  (99)) wymaga przeszukania przestrzeni obejmującej  $2^5$  potencjalnych wariantów reguł priorytetowania (5 zasobów współdzielonych przez dwa procesy lokalne – przyjęto, że reguły są dwuelementowe) i  $t_M^{12}$  wariantów czasów wykonania operacji ( $t_M$  – górny zakres wartości czasów wykonania operacji:  $0 \leq t_{ij} \leq t_M$ ,  $t_M = 10$ ). W tym ujęciu rozmiar przestrzeni przeszukiwań wynosi  $2^5 \cdot 10^{12}$ .

W alternatywnym podejściu zakładającym możliwość dekompozycji struktury systemu wyróżniono 4 podstruktury:  $SC^* = \{SC_1^1, SC_2^1, SC_3^1, SC_4^1\}$  (rys. 4.2b). Reguły priorytetowania  $\theta^0$  oraz czasy wykonywania operacji  $T$ , wyznaczano w kolejnych etapach składania podstruktur  $SC^*$ . W pierwszej kolejności złożono podstruktury  $SC_1^1$  i  $SC_2^1$  generujące cykliczne harmonogramy  $X_1^1, X_2^1$  (rys. 4.11). Harmonogramy te nie są znane tak jak nieznane są czasy wykonania poszczególnych operacji.



Rys. 4.11. Składanie  $SC_1^1 \oplus SC_2^1$

Zgodnie z Twierdzeniem 7, w systemie o strukturze  $SC_{1-2}^1 = SC_1^1 \oplus SC_2^1$  osiągalne jest zachowanie cykliczne  $X_{1-2}^1 = X_1^1 \cup X_2^1$  jeżeli operacje realizowane (zgodnie z  $X_1^1, X_2^1$ ) na zasobie  $R_3$  spełniają ograniczenia (109) i (110). **Oznacza to konieczność poszukiwania harmonogramów (wartości czasów wykonania operacji), gwarantujących spełnienie tych ograniczeń.**

Istnienie harmonogramów cyklicznych  $X_1^1, X_2^1$  determinują (patrz podrozdział 3.3) rozwiązania odpowiednich problemów spełniania ograniczeń:  $PS'_{REX}$  (98) (dla operacji procesów lokalnych),  $m^1 PS'_{REX}$  (99) (dla operacji procesów multimodalnych).

W rozważanym przypadku, problemy wyznaczania czasów wykonywania operacji  $T_i$ ,  $m^1T_i$  oraz harmonogramu:  $X_i^1 = \left( (X'_{DC_i}, \alpha_i), (m^1X'_{DC_i}, m^1\alpha_i) \right)$ , mają postać:

$$PS'_{REX_i} = \left( (T_i \cup X'_{DC_i} \cup \{\alpha_i\}, \{D_T, D_X, D_\alpha\}), C'_{X_i} \right) \quad (112)$$

$$m^1PS'_{REX_i} = \left( (m^1T_i \cup m^1X'_{DC_i} \cup \{m^1\alpha_i\}, \{D_T^1, D_X^1, D_\alpha^1\}), C_{X_i}^1(X'_{DC_i}) \right), \quad (113)$$

gdzie:  $C'_{X_i}$ ,  $C_{X_i}^1(X'_{DC_i})$  – zbiory ograniczeń opisujących związki między momentami rozpoczęcia operacji procesów lokalnych (określane zgodnie z (78) i Algorytmem 4) i multimodalnych (określane zgodnie z (92) i Algorytmem 5). Uproszczoną postać tych ograniczeń dla podstruktur  $SC_1^1$  i  $SC_2^1$  ilustruje rys. 4.11.

Analogicznie jak w przypadku  $X_1^1$ ,  $X_2^1$ , harmonogram  $X_{1-2}^1 = X_1^1 \cup X_2^1$  charakteryzujący zachowanie systemu o strukturze  $SC_{1-2}^1$ , jest determinowany przez rozwiązania problemów odpowiednio:  $PS'_{REX_{1-2}}$  i  $m^1PS'_{REX_{1-2}}$ . Problemy te można przedstawić jako kompozycję (połączenie) problemów  $PS'_{REX_1}$ ,  $PS'_{REX_2}$  oraz  $m^1PS'_{REX_1}$ ,  $m^1PS'_{REX_2}$ .

Niech  $\cup$  oznacza **operator łączenia dwóch problemów spełniania ograniczeń**  $PS_a = (V_a, D_a, C_a)$ ,  $PS_b = (V_b, D_b, C_b)$ . Wynikiem takiego połączenia jest problem  $PS_c$  postaci:

$$PS_c = PS_a \cup PS_b = (V_a \cup V_b, D_a \cup D_b, C_a \cup C_b \cup C_D), \quad (114)$$

gdzie:

$V_a, V_b$  – zbiory zmiennych decyzyjnych problemów  $PS_a, PS_b$ ,

$D_a, D_b$  – zbiory dziedzin zmiennych decyzyjnych  $V_a, V_b$ ,

$C_a, C_b$  – zbiór ograniczeń problemów  $PS_a, PS_b$ ,

$C_D$  – zbiór ograniczeń dodatkowych, określających relacje łączące zmienne decyzyjnymi  $V_a$  i  $V_b$ .

W tym kontekście problemy  $PS'_{REX_{1-2}}$  i  $m^1PS'_{REX_{1-2}}$  definiowane są następująco:

$$PS'_{REX_{1-2}} = PS'_{REX_1} \cup PS'_{REX_2}, \quad (115)$$

$$m^1PS'_{REX_{1-2}} = m^1PS'_{REX_1} \cup m^1PS'_{REX_2}, \quad (116)$$

gdzie:  $PS'_{REX_1}$ ,  $m^1PS'_{REX_1}$  oraz  $PS'_{REX_2}$ ,  $m^1PS'_{REX_2}$  – problemy spełniania ograniczeń pozwalające wyznaczyć harmonogramy  $X_1^1$ ,  $X_2^1$ .

W wyrażeniach (115) i (116) wykorzystywane są ograniczenia dodatkowe  $C_D$  (warunki (109) i (110)) oraz ograniczenia typu  $C_X^1$  (92) (determinujące kolejność wykonywania operacji procesów multimodalnych).

Na rys. 4.11 przedstawiono ograniczenia charakteryzujące zachowanie osiągalne w strukturze  $SC_{1-2}^1$  otrzymanej w wyniku składania  $SC_1^1 \dot{\cup} SC_2^1$ . Ograniczenia  $C'_{X_1}$  i  $C'_{X_2}$  należą do problemu  $PS'_{REX_{1-2}}$  (115), natomiast ograniczenia  $C_{X_1}^1(X'_{DC_1})$ ,  $C_{X_2}^1(X'_{DC_2})$  do problemu  $m^1PS'_{REX_{1-2}}$  (116). Ograniczenia  $C_{D_{1-2}}^1$  (w problemie  $PS'_{REX_{1-2}}$ ) sta-

nowią ograniczenia dodatkowe gwarantujące przemienną realizację procesów lokalnych na zasobie  $R_3$  (ograniczenia typu (109)) oraz wykonanie operacji  $m^1 o_{1,3}$  potoku  $m^1 P_1$ , po ukończeniu operacji  $m^1 o_{1,2}$  (ograniczenie typu (92)). Dla procesów multimodalnych ograniczenia tego typu nie są konieczne (na zasobie  $R_3$  realizowany jest tylko jeden proces multimodalny).

Zachowanie systemu o strukturze  $SC_{1-2}^1$  ilustruje harmonogram  $X_{1-2}^1$  (rys. 4.11) będący rozwiązaniem problemów (115) i (116). Poszukiwane zachowanie cykliczne jest osiągalne gdy:

- czasy wykonania operacji wynoszą:

$$T_1^1 = (6, 1, 1), T_2^1 = (4, 2, 2), \\ m^1 T_1^1(1, 3) = (6, 1, 1), m^1 T_2^1(4, 4) = (6),$$

gdzie:  $T_i^h$  – sekwencja czasów wykonania operacji potoku  $P_i^h$ ,  $m^l T_i^h(a, b)$  – sekwencja czasów wykonania operacji realizowanych w ramach fragmentu potoku  $m^l P_i^h(a, b)$ ,

- momenty rozpoczęcia operacji wynoszą:

$$x_1^1 = (-2, 4, 5), x_2^1 = (-1, 3, 5), \alpha = 8, \\ m^1 x_1^1(1, 3) = (-2, 4, 11), m^1 x_2^1(4, 4) = (6), m^1 \alpha = 16,$$

gdzie:  $x_i^h$  – sekwencja momentów rozpoczęcia operacji potoku  $P_i^h$ . Przykładowo z sekwencji  $x_1^1$  wynika, że momenty rozpoczęcia operacji potoku  $P_1^1$  (wykonywane kolejno na zasobach  $R_1, R_3, R_2$ ) pierwszego cyklu wynoszą:  $-2, 4, 5$  u.j.c. i są powtarzane z okresem  $8$  u.j.c.

$m^l x_i^h(a, b)$  – sekwencja momentów rozpoczęcia operacji realizowanych w ramach fragmentu potoku  $m^l P_i^h(a, b)$ .

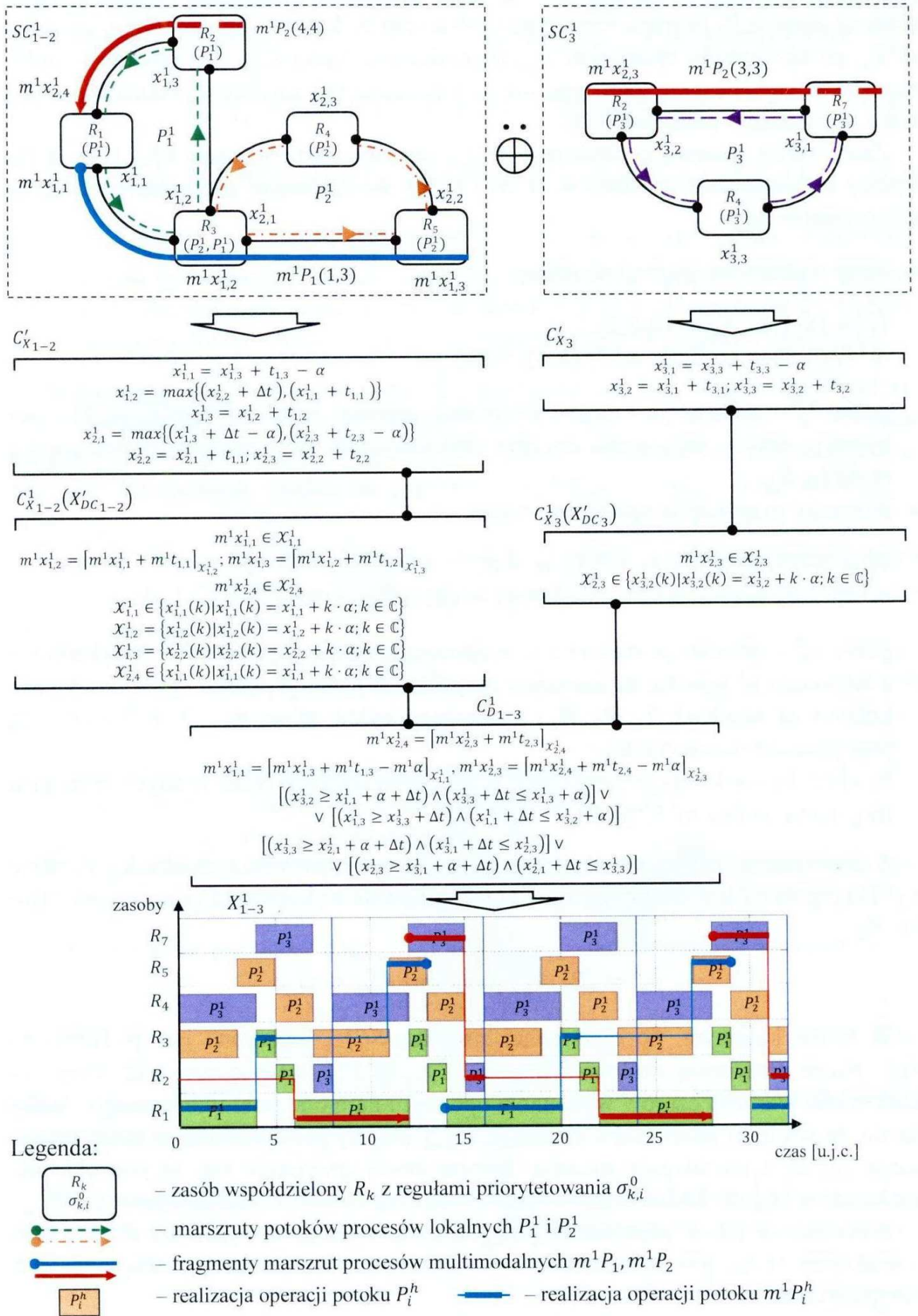
Z harmonogramu  $X_{1-2}^1$  wynika również reguła priorytetowania zasobu  $R_3$ . Zgodnie z (111) reguła  $\sigma_3^0$  jest determinowana przez kolejność wykonywania operacji na zasobie  $R_3$ :

$$\sigma_3^0 = (P_2^1, P_1^1) \text{ ponieważ } x_{2,1}^1 < x_{1,2}^1.$$

W kroku kolejnym, do otrzymanej struktury  $SC_{1-2}^1$  dołączana jest podstruktura  $SC_3^1$ . Nowo otrzymaną strukturę  $SC_{1-3}^1 = SC_{1-2}^1 \dot{\oplus} SC_3^1$  ilustruje rys. 4.12. Przy poszukiwaniu wartości czasów wykonania operacji i harmonogramu końcowego, zakłada się, że wartości parametrów struktury  $SC_{1-2}^1$  (reguły priorytetowania, czasy wykonania operacji) nie ulegają zmianie. Innymi słowy przyjmuje się, że poszukiwane zachowanie  $SC_3^1$  nie koliduje z już wyznaczonym sposobem realizacji operacji  $SC_{1-2}^1$ .

Analogicznie jak w poprzednim przypadku, harmonogram cykliczny dla systemu o strukturze  $SC_{1-3}^1$  jest wyznaczany w wyniku rozwiązania problemów będących kompozycjami:

$$PS'_{REX_{1-3}} = PS'_{REX_{1-2}} \cup PS'_{REX_3} \text{ oraz } m^1 PS'_{REX_{1-3}} = m^1 PS'_{REX_{1-2}} \cup m^1 PS'_{REX_3}.$$

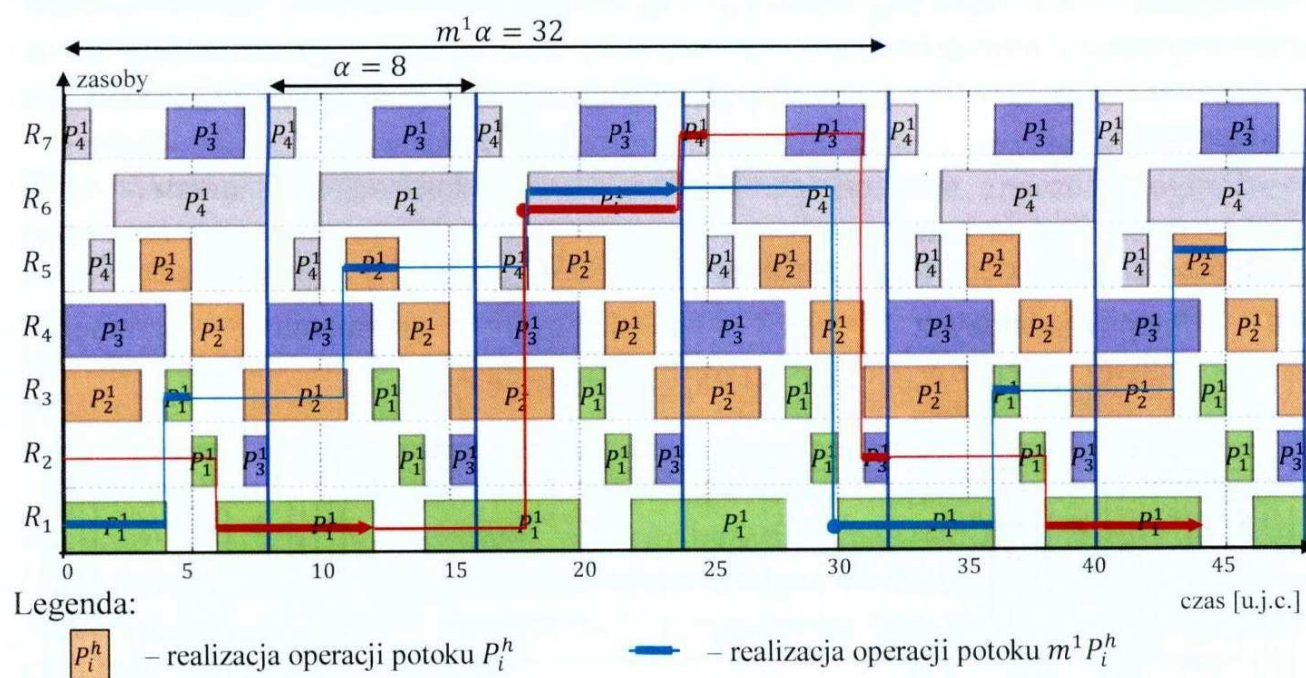


Rys. 4.12. Składanie  $SC_{1-2}^1 \oplus SC_3^1$

Harmonogram  $X_{1-3}^1$  ilustrujący cykliczne zachowanie systemu o strukturze  $SC_{1-3}^1$  przedstawia rys. 4.12, wynikające z niego reguły priorytetowania dla zasobów  $R_2$  i  $R_4$  mają postać:

$$\sigma_2^0 = (P_1^1, P_3^1), \sigma_4^0 = (P_2^1, P_3^1).$$

W kolejnym kroku, do struktury  $SC_{1-3}^1$  dołączono podstrukturę  $SC_4^1$ . W celu wyznaczenia zachowania systemu o strukturze:  $SC_{1-4}^1 = SC_{1-3}^1 \dot{\oplus} SC_4^1$ , rozwiązane zostały problemy spełniania ograniczeń odpowiadające kompozycjom:  $PS'_{REX_{1-4}} = PS'_{REX_{1-3}} \dot{\cup} PS'_{REX_4}$ ,  $m^1 PS'_{REX_{1-4}} = m^1 PS'_{REX_{1-3}} \dot{\cup} m^1 PS'_{REX_4}$ . Harmonogram cyklicznej realizacji procesów systemu o strukturze  $SC_{1-4}^1$  przedstawia rys. 4.13.



Rys. 4.13. Harmonogram cykliczny systemu z rys. 4.2

Czasy wykonania i momenty rozpoczęcia operacji, a także wynikające z nich reguły priorytetowania przedstawiają odpowiednio tab. 4.2 – 4.4.

Tabela 4.2. Czasy wykonania operacji procesów SWMPC z rys. 4.2

	$i$	$t_{i,1}$	$t_{i,2}$	$t_{i,3}$
$P_1^1$	1	6	1	1
$P_2^1$	2	4	2	2
$P_3^1$	3	3	1	4
$P_4^1$	4	6	1	1

	$i$	$m^1 t_{i,1}$	$m^1 t_{i,2}$	$m^1 t_{i,3}$	$m^1 t_{i,4}$
$m^1 P_1^1$	1	6	1	2	6
$m^1 P_2^1$	2	6	1	2	6

Tabela 4.3. Momenty rozpoczęcia operacji procesów lokalnych SWMPC z rys. 4.2

	$i$	$x_{i,1}^1$	$x_{i,2}^1$	$x_{i,3}^1$
$P_1^1$	1	-2	4	5
$P_2^1$	2	-1	3	5
$P_3^1$	3	4	7	8
$P_4^1$	4	2	8	9

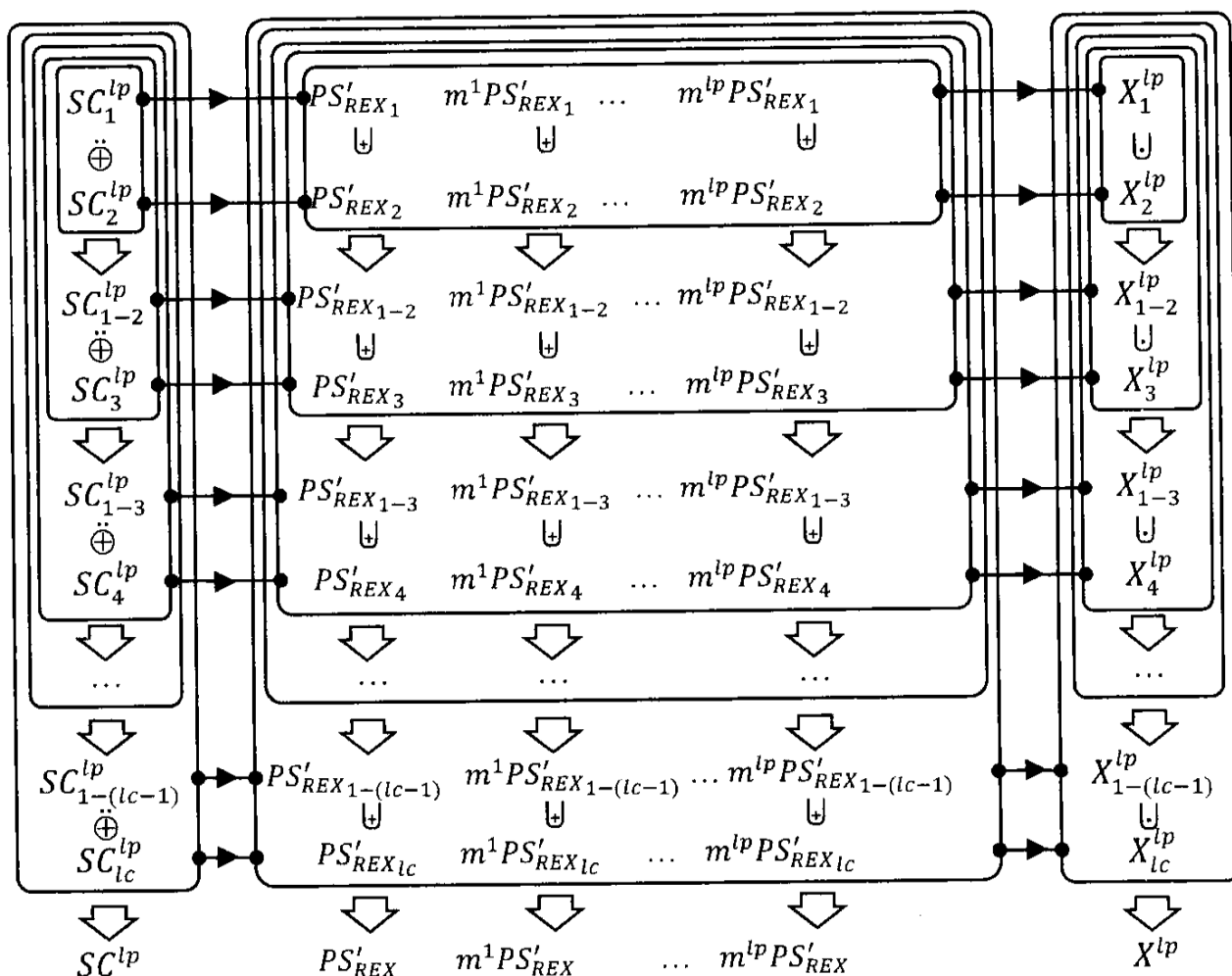
	$i$	$m^1 x_{i,1}^1$	$m^1 x_{i,2}^1$	$m^1 x_{i,3}^1$	$m^1 x_{i,4}^1$
$m^1 P_1^1$	1	-2	4	11	18
$m^1 P_2^1$	2	-15	-8	-1	6

Tabela 4.4. Reguły priorytetowania SWMPC z rys. 4.2

reguły dla zasobów $R_2, R_3$ i $R_4$		reguły dla zasobów $R_5$ i $R_7$	
$\sigma_2^0$	$(P_1^1, P_3^1)$	$\sigma_5^0$	$(P_2^1, P_4^1)$
$\sigma_3^0$	$(P_2^1, P_1^1)$	$\sigma_7^0$	$(P_3^1, P_4^1)$
$\sigma_4^0$	$(P_2^1, P_3^1)$		

Wyznaczanie wartości poszukiwanych parametrów, wymagało trzykrotnego rozwiązania problemów typu (112), (113). W odróżnieniu od pierwotnie rozważanej sytuacji zakładającej brak dekompozycji struktury systemu (wymagającej zastosowanie Algorytmu 6), rozwiązanie ww. problemów wymagało przeszukania przestrzeni obejmującej  $2 \cdot t_M^6 + 2^2 \cdot t_M^3 + 2^2 \cdot t_M^3$  ( $t_M = 10$ ) potencjalnych wariantów reguł priorytetowania i wartości czasów wykonania operacji. Oznacza to, że zaproponowana dekompozycja struktury pozwoliła zmniejszyć rozmiar przeszukiwanej przestrzeni o ponad 99%.

Ogólną zasadę składania podstruktur  $((SC_1^{lp} \oplus SC_2^{lp}) \oplus SC_3^{lp}) \oplus \dots \oplus SC_{lc}^{lp}$  jak i łączenia odpowiadających im problemów spełniania ograniczeń  $((PS'_{REX_1} \cup PS'_{REX_2}) \cup PS'_{REX_3}) \cup \dots \cup PS'_{REX_{lc}}$  ilustruje rys. 4.14.



Rys. 4.14. Idea wyznaczania harmonogramu cyklicznego  $X^{lp}$  polegająca na składaniu podstruktur struktury  $SC^{lp}$



Zgodnie z przyjętą metodyką, każda wyjściowa (aktualnie rozważana) struktura systemu SWMPC jest dekomponowana na zbiór podstruktur i związanych z nimi zachowań determinowanych poprzez rozwiązania problemów typu (112), (113).

Rozwiązania tych problemów pozwalają wyznaczyć parametry gwarantujące osiągalność zachowania cyklicznego (np. harmonogramu  $X_{1-2}^{lp}$ ) w podsystemach wyznaczanych przez struktury składanych podstruktur (np.  $SC_{1-2}^{lp}$ ). Dla raz ustalonych parametrów jednej struktury, w kolejnych krokach, poszukiwane są parametry składanej z nią kolejnej podstruktury (np.  $SC_3^{lp}$ ). Ocena tego czy możliwe jest złożenie podstruktury ze strukturą o już ustalonych parametrach odbywa się w oparciu o regułę „zamka błyskawicznego” – warunki (109), (110). Składanie kolejnych podstruktur jest realizowane do momentu wyczerpania wszystkich podstruktur lub stwierdzenia, że tego typu złożenie nie jest możliwe (sprzeczność ograniczeń co najmniej jednego z problemów  $PS'_{REX_1}$ ,  $m^1PS'_{REX_1}$ , ...,  $m^{lp}PS'_{REX_1}$ ).

Przedstawione postępowanie odpowiada metodzie „dziel i zwyciężaj”. Dekompozycja systemu na odpowiednio małe podstruktury pozwala znaczenie ograniczyć przestrzeń przeszukiwań (jeszcze przed etapem wykorzystania technik programowania z ograniczeniami), a tym samym skrócić czas obliczeń. Umożliwia również jednoczesne, niezależnie prowadzone wyznaczanie zachowań poszczególnych podstruktur. Oznacza to, że zakres zastosowań proponowanego podejścia determinuje rozmiar podstruktur rozważanego SWMPC. Liczba operacji  $LO$  realizowanych w ramach każdej podstruktury nie powinna przekraczać 40 (rys. 3.27).

Podobnie jak w przypadku wcześniej omawianych warunków (np. acykliczności digrafów pierwszeństwa operacji) warunki bezkolizyjnej realizacji procesów (109), (110) należy traktować jako **warunki wystarczające** do istnienia cyklicznych zachowań SWMPC. W praktyce oznacza to, że niespełnienie tych warunków nie musi oznaczać braku zachowań cyklicznych.

Podsumowując dekompozycja zadanej struktury systemu znaczenie skraca czas obliczeń kosztem zmniejszenia przestrzeni rozwiązań dopuszczalnych.

## 4.2. Synteza struktur regularnych

Reguła „zamka błyskawicznego” może być również użyta w problemach syntezy struktur regularnych. Ilustrację tej możliwości przedstawiono na następującym przykładzie.

Dany jest zbiór  $SC^* = \{SC_1^{lp}, \dots, SC_i^{lp}, \dots, SC_{lc}^{lp}\}$  podstruktur izomorficznych. Struktury  $SC_a^{lp}$ ,  $SC_b^{lp}$  są izomorficzne jeżeli:

- każdemu zasobowi  $R_a \in Rp_a$  podstruktury  $SC_a^{lp}$  odpowiada dokładnie jeden zasób  $R_b \in Rp_b$  podstruktury  $SC_b^{lp}$ :  $R_b = f(R_a)$ ,
- każdemu procesowi  $P_a$ ,  $m^l P_a$  (lokalnemu jak i multimodalnemu) podstruktury  $SC_a^{lp}$  odpowiada dokładnie jeden proces  $P_b$ ,  $m^l P_b$  podstruktury  $SC_b^{lp}$ :  $P_b = f(P_a)$ ,  $m^l P_b = f(m^l P_a)$ ,
- marszruty  $p_b$ ,  $m^l p_b$  i  $p_a$ ,  $m^l p_a$ , odpowiadających sobie procesów, są sekwencjami składającymi się z odpowiadających sobie zasobów,

- sekwencje operacji potoków  $O_a^h, m^l O_a^h$  i  $O_b^h, m^l O_b^h$ , odpowiadających sobie procesów, składają się z odpowiadających sobie operacji:  $o_{a,j}^h = f(o_{b,j}^h)$ ,  $m^l o_{a,j}^h = f(m^l o_{b,j}^h)$ , wykonywanych w tym samym czasie  $t_{a,j} = t_{b,j}$ ,  $m^l t_{a,j} = m^l t_{b,j}$ ,
- reguły priorytetowania  $\sigma_a^l, \sigma_b^l$ , odpowiadających sobie zasobów, są sekwencjami składającymi się z elementów  $s_{a,d}^l, s_{b,d}^l$ , wskazujących na potoki odpowiadających sobie procesów.

Struktura złożona z podstruktur izomorficznych nazywana jest dalej **strukturą regularną**. Przykładem takiej struktury jest  $SC^1$  z rys. 4.15a. Podstruktury występujące w  $SC^1$  składają się z 12 zasobów, 3 procesów lokalnych i dwóch fragmentów procesów multimodalnych (przyjmuje się, że z każdym fragmentem związany jest jeden potok procesu multimodalnego). Cechą charakterystyczną struktury  $SC^1$  jest to, że każda z wchodzących w jej skład podstruktur izomorficznych może zostać przedstawiona przez jedną podstrukturę  $(i)SC^1$  (jak na rys. 4.15b), w której elementy (zasoby, procesy, operacje, momenty ich rozpoczęcia, itp.) wyróżniane są przez lewy indeks górny  $(i)$ .

W rozważanym kontekście, problem składania struktury regularnej sprowadza się do pytania:

*Czy istnieją takie parametry (reguły priorytetowania, czasy wykonywania operacji itp.) podstruktur izomorficznych  $(i)SC^1$ , które gwarantują, że w systemie o strukturze  $SC^1$  złożonej z tych podstruktur osiągalne jest zachowanie cykliczne (harmonogram cykliczny  $X^1$ )?*

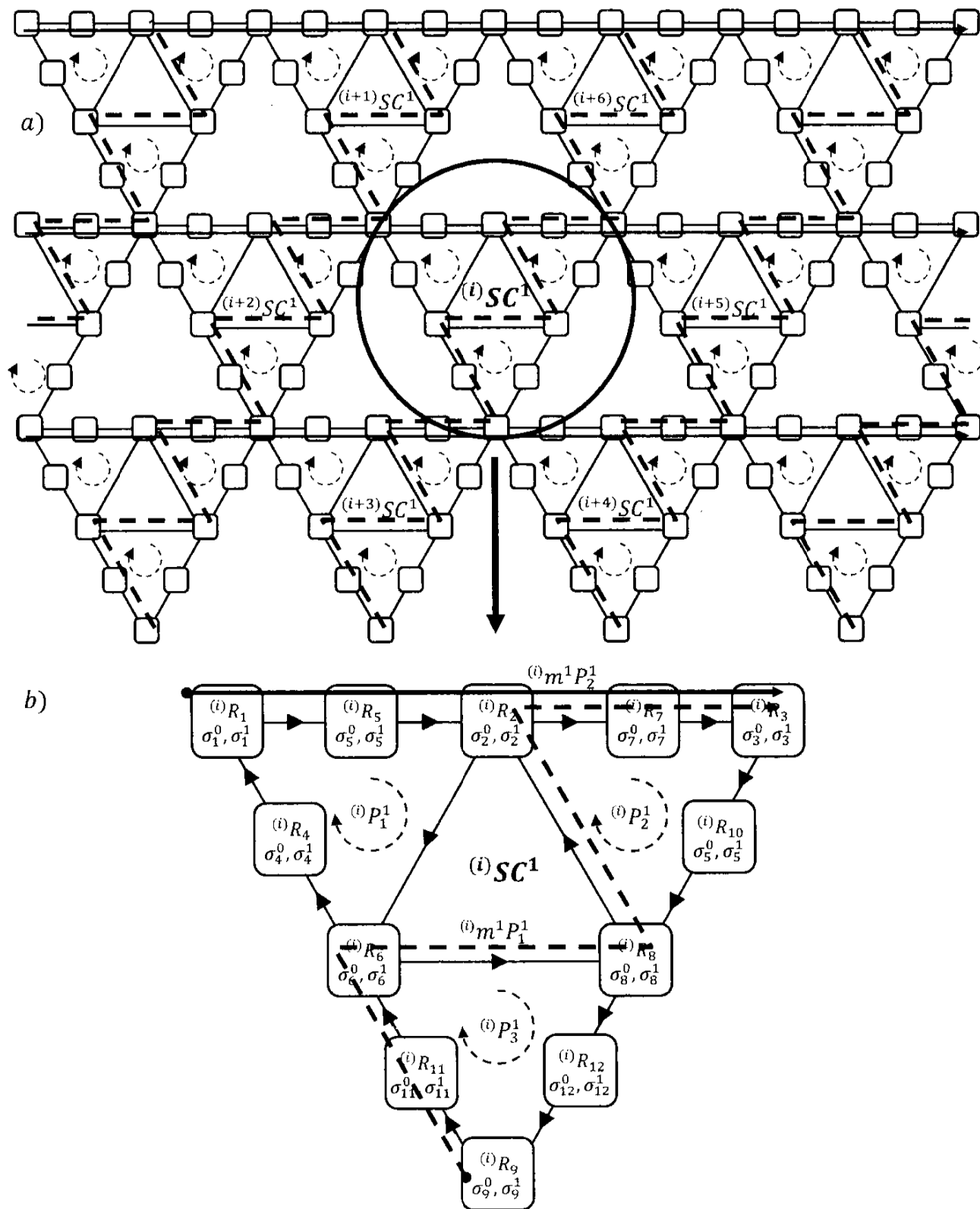
Przyjmując, że do składania podstruktur wykorzystywana jest reguła „zamka błyskawicznego” zachowanie systemu o strukturze regularnej  $SC^1$  wyznacza schemat postępowania przedstawiony na rys. 4.14. Wynikowy harmonogram cykliczny  $X^1$  stanowi efekt wielokrotnego rozwiązywania problemów  $PS'_{REX_{i-j}} = PS'_{REX_i} \cup PS'_{REX_j}$  oraz  $m^1 PS'_{REX_{i-j}} = m^1 PS'_{REX_i} \cup m^1 PS'_{REX_j}$ .

Wyznaczanie harmonogramu  $X^1$  (determinującego reguły priorytetowania oraz czasy wykonywania operacji) można uprościć wykorzystując izomorficzny charakter łączonych podstruktur.

W każdej podstrukturze reprezentowanej przez  $(i)SC^1$  (rys. 4.15b) procesy realizowane są w ten sam sposób – operacje wykonywane są wzdłuż tych samych marszrut, stosowane są te same reguły priorytetowania itp. Struktura  $SC^1$  stanowi więc efekt wielokrotnego złożenia podstruktur  $(i)SC^1$ :

$$SC^1 = \ddot{\oplus}_{i=1}^{lc} (i)SC^1, \quad (117)$$

gdzie:  $\ddot{\oplus}_{i=1}^{lc} (i)SC^1 = (1)SC^1 \ddot{\oplus} \dots \ddot{\oplus} (i)SC^1 \ddot{\oplus} \dots \ddot{\oplus} (lc)SC^1$  – składanie podstruktur  $(i)SC^1$  dla  $i = 1 \dots lc$ , gdzie:  $\ddot{\oplus}$  oznacza składanie dwóch podstruktur zgodnie z regułą „zamka błyskawicznego” (funkcja  $\ddot{\oplus}$  definiowana jak w (111)) po odpowiadających sobie zasobach  $(i)Rk$ .



**Rys. 4.15.** Przykład struktury złożonej z podstruktur izomorficznych a) pojedyncza podstruktura izomorficzna  $(i)SC^1$  b)

Przykład takiego składania przedstawia rys. 4.16, w którym podstruktura  $(i)SC^1$  jest składana z pozostałymi poprzez zasoby  $(i)R_1$ ,  $(i)R_3$ ,  $(i)R_9$ . Zasób  $(i)R_1$  pełni rolę zasobu  $(i+2)R_3$  podstruktury  $(i+2)SC^1$  oraz zasobu  $(i+1)R_9$  podstruktury  $(i+1)SC^1$ .

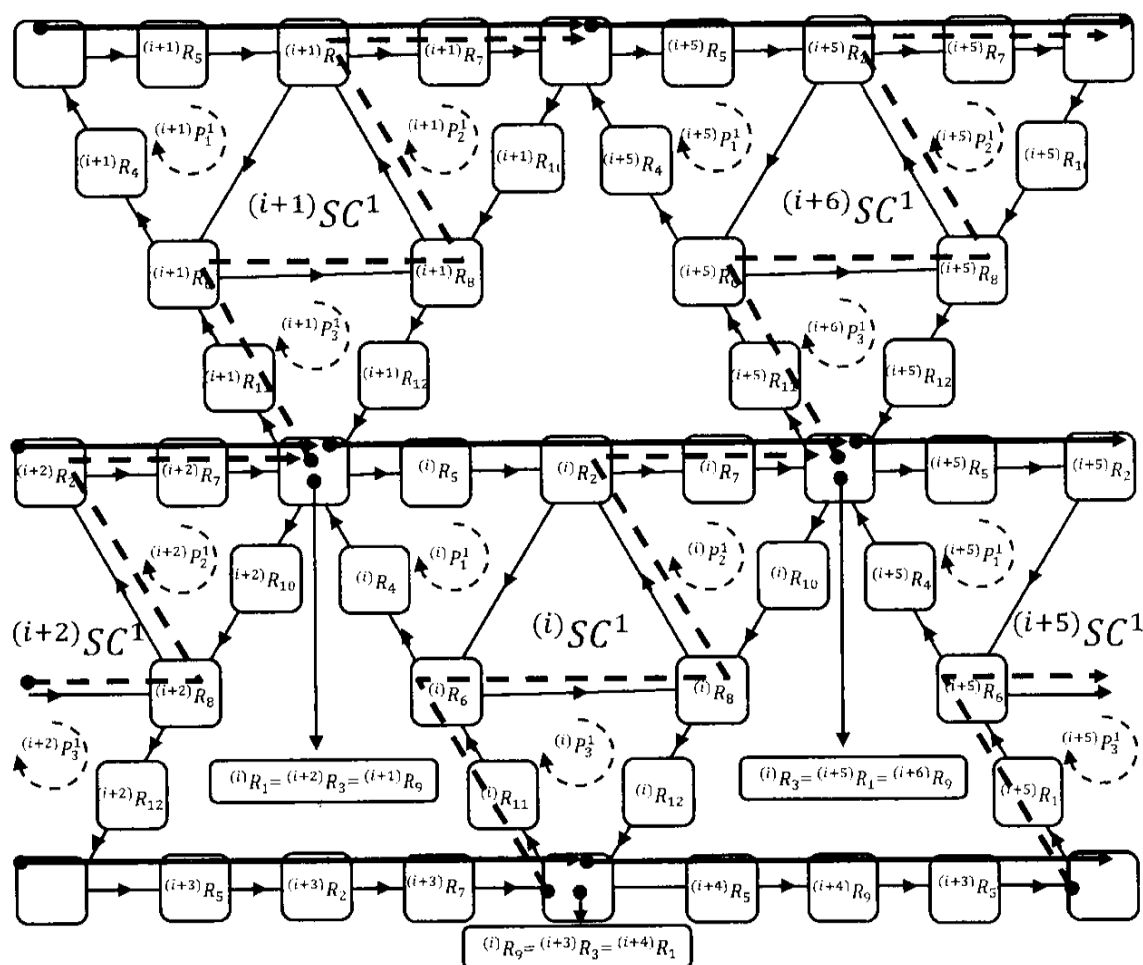
W rozważanym przypadku  $(i)Rk$  stanowi rodzinę odpowiadających sobie zbiorów zasobów – zasobów współdzielonych z sąsiednimi podstrukturami izomorficznymi. Każda podstruktura izomorficzna typu  $(i)SC^1$  (rys. 4.15b) współdzieli z sąsiednimi podstrukturami zasoby:

- ${}^{(i)}R_1$  pełniące rolę zasobów  ${}^{(i+2)}R_3$  i  ${}^{(i+1)}R_9$  (sąsiedztwo z  ${}^{(i+1)}SC^1$  i  ${}^{(i+2)}SC^1$ ),
- ${}^{(i)}R_3$  pełniące rolę zasobów  ${}^{(i+5)}R_1$  i  ${}^{(i+6)}R_9$  (sąsiedztwo z  ${}^{(i+5)}SC^1$  i  ${}^{(i+6)}SC^1$ ),
- ${}^{(i)}R_9$  traktowane jako  ${}^{(i+3)}R_3$  i  ${}^{(i+4)}R_1$  (sąsiedztwo z  ${}^{(i+3)}SC^1$  i  ${}^{(i+3)}SC^1$ ).

W tym ujęciu rodzina  ${}^{(i)}Rk$  wykorzystana przy składaniu podstruktur (117) przyjmuje postać:

$${}^{(i)}Rk = \left\{ \left\{ {}^{(i)}R_1, {}^{(i+2)}R_3, {}^{(i+1)}R_9 \right\}, \left\{ {}^{(i)}R_3, {}^{(i+5)}R_1, {}^{(i+6)}R_9 \right\}, \left\{ {}^{(i)}R_9, {}^{(i+3)}R_3, {}^{(i+4)}R_1 \right\} \right\}.$$

Jak widać na rys. 4.16 zasoby należące do jednego zbioru rodziny  ${}^{(i)}Rk$  są ze sobą łączone.



Rys. 4.16. Składanie podstruktur  ${}^{(i)}SC^1$  poprzez zasoby  ${}^{(i)}R_1$ ,  ${}^{(i)}R_3$  oraz  ${}^{(i)}R_9$

Przyjęcie tego samego sposobu realizacji procesów, jak i tego samego sposobu składania podstruktur izomorficznych, pozwala postrzegać harmonogram cykliczny reprezentujący zachowanie całej struktury jako kompozycję odpowiadających sobie (izomorficznych) harmonogramów:

$$X^1 = \dot{\cup}_{i=1}^{lc} ({}^{(i)}X^1), \quad (118)$$

gdzie:

${}^{(i)}X^1$  – harmonogram cykliczny podstruktury  ${}^{(i)}SC^1$ ,

$\dot{\cup}_{i=1}^{lc} ({}^{(i)}X^1) = ({}^{(1)}X^1 \dot{\cup} ({}^{(2)}X^1 \dot{\cup} \dots \dot{\cup} ({}^{(lc)}X^1$  – łączenie harmonogramów  ${}^{(i)}X^1$ , dla  $i = 1 \dots lc$ , zgodnie z (108).



**Tabela 4.6.** Momenty rozpoczęcia operacji procesów lokalnych i multimodalnych  $(i)SC^1$  z rys. 4.17a

	$j$	$(i)x_{j,1}^1$	$(i)x_{j,2}^1$	$(i)x_{j,3}^1$	$(i)x_{j,4}^1$	$(i)x_{j,5}^1$		$j$	$(i)m^1x_{j,1}^1$	$(i)m^1x_{j,2}^1$	$(i)m^1x_{j,3}^1$	$(i)m^1x_{j,4}^1$	$(i)m^1x_{j,5}^1$	$(i)m^1x_{j,6}^1$	$(i)m^1x_{j,7}^1$
$(i)P_1^1$	1	2	3	4	5	6	$(i)m^1P_1^1$	1	6	7	8	9	13	14	16
$(i)P_2^1$	2	4	5	6	7	8	$(i)m^1P_2^1$	2	2	3	4	8	10	-	-
$(i)P_3^1$	3	6	7	8	9	10									

Do wyznaczenia harmonogramu  $X^1$  wystarczy więc znajomość harmonogramu  $(i)X^1$  pojedynczej podstruktury. Aby połączenie (118) było możliwe, operacje realizowane zgodnie z  $(i)X^1$  na zasobach wspólnych nie mogą kolidować, tzn. muszą być realizowane naprzemiennie. Zgodnie z wykorzystywaną regułą „zamka błyskawicznego” (patrz rys. 4.16), operacje potoków  $(i)P_1^1, (i+2)P_2^1, (i+1)P_3^1$  (na zasobie  $(i)R_1$ ),  $(i)P_2^1, (i+2)P_1^1, (i+6)P_3^1$  (na zasobie  $(i)R_3$ ) oraz  $(i)P_3^1, (i+3)P_2^1, (i+4)P_1^1$  (na zasobie  $(i)R_9$ ), powinny być realizowane bezkolizyjnie.

Ponieważ realizacja potoków  $(i)P_1^1, (i+2)P_1^1, (i+4)P_1^1$  (jak i  $(i)P_2^1, (i+2)P_2^1, (i+3)P_2^1$  oraz  $(i)P_3^1, (i+1)P_3^1, (i+6)P_3^1$ ) podstruktur sąsiadujących  $(i)SC^1, (i+1)SC^1, \dots, (i+6)SC^1$ , jest taka sama we wszystkich podstrukturach, wystarczy zatem wyznaczyć parametry tylko jednej z nich, przedstawionej na rys. 4.17a. W strukturze tej, liniami przerywanymi oznaczono ograniczenia, gwarantujące, że na zasobach połączonych przez te linie nie dochodzi do jednoczesnej realizacji operacji potoków  $(i)P_1^1, (i)P_2^1, (i)P_3^1$ . Bezkolizyjna realizacja potoków podstruktury z rys. 4.17a oznacza bezkolizyjną realizację potoków z wszystkich podstruktur  $(i)SC^1$ .

Do wyznaczenia parametrów (reguł priorytetowania i czasów wykonania operacji) podstruktury  $(i)SC^1$  (rys. 4.15a), można wykorzystać rozszerzoną postać problemów (113) i (114):

$$PS'_{REX_i} = \left( (T_i \cup X'_{DC_i} \cup \{\alpha_i\}, \{D_T, D_X, D_\alpha\}), C'_{X_i} \cup C_{D_i} \right), \quad (119)$$

$$m^1PS'_{REX_i} = \left( (m^1T_i \cup m^1X'_{DC_i} \cup \{m^1\alpha_i\}, \{D_T^1, D_X^1, D_\alpha^1\}), C'_{X_i}(X'_{DC_i}) \cup C_{D_i} \right), \quad (120)$$

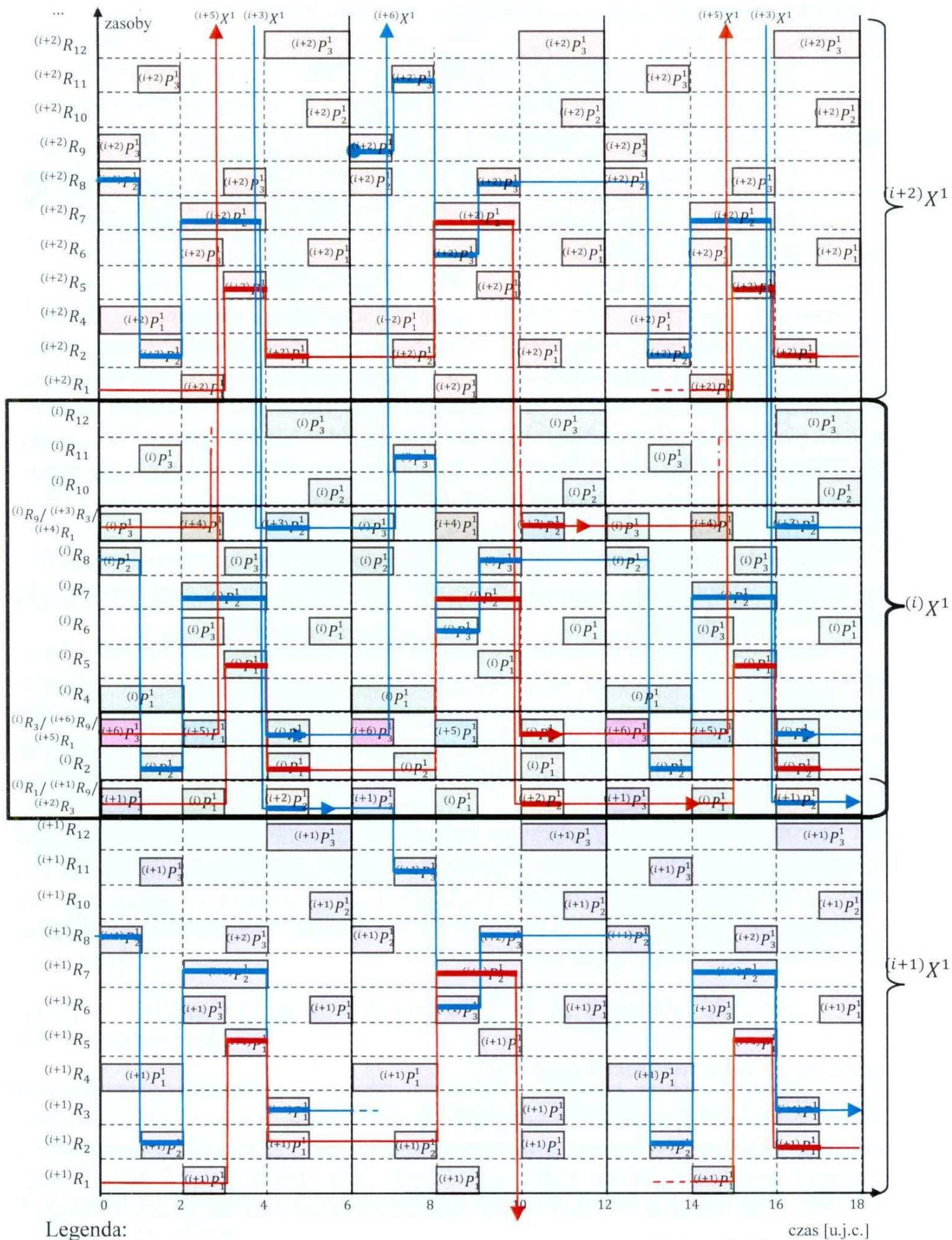
gdzie:

$C'_{X_i}, C'_{X_i}(X'_{DC_i})$  – zbiory ograniczeń opisujące związki między momentami rozpoczęcia operacji procesów lokalnych (zgodnie z (78) i Algorytmem 4) i multimodalnych (zgodnie z (92) i Algorytmem 5),

$C_{D_i}, C_{D_i}$  – zbiory ograniczeń (109) i (110) niedopuszczających do jednoczesnej realizacji potoków  $(i)P_1^1, (i)P_2^1, (i)P_3^1$  na zasobach łączonych  $(i)R_1, (i)R_3$  i  $(i)R_9$ ). Ograniczenia charakteryzujące rozważaną strukturę zostały przedstawione rys. 4.17a w postaci przerywanych strzałek.

Rozwiązanie problemów (119) i (120) dla podstruktury z rys. 4.17a pozwoliło wyznaczyć czasy wykonania operacji oraz momenty ich rozpoczęcia przedstawione w tab. 4.5 i 4.6.

Harmonogram cykliczny zachowania osiągalnego w systemie o podstrukturze  $(i)SC^1$  ilustruje rys. 4.17b. Przedstawione na nim operacje realizowane na zasobach łączonych  $(i)R_1, (i)R_3$  i  $(i)R_9$ , nie nakładają się na siebie.



Rys. 4.18. Harmonogram cykliczny osiągalny w systemie o strukturze regularnej SC<sup>1</sup> z rys. 4.15a

Tabela 4.7. Reguły priorytetowania podstruktury  ${}^{(i)}SC^1$  z rys. 4.17a

reguły dostępu procesów lokalnych			
${}^{(i)}\sigma_1^0$	$({}^{(i)}P_1^1, {}^{(i+2)}P_2^1, {}^{(i+1)}P_3^1)$	${}^{(i)}\sigma_6^0$	$({}^{(i)}P_1^1, {}^{(i)}P_3^1)$
${}^{(i)}\sigma_2^0$	$({}^{(i)}P_1^1, {}^{(i)}P_2^1)$	${}^{(i)}\sigma_8^0$	$({}^{(i)}P_2^1, {}^{(i)}P_3^1)$
${}^{(i)}\sigma_3^0$	$({}^{(i)}P_2^1, {}^{(i+2)}P_3^1, {}^{(i+5)}P_1^1)$	${}^{(i)}\sigma_9^0$	$({}^{(i)}P_3^1, {}^{(i+4)}P_1^1, {}^{(i+2)}P_2^1)$
reguły dostępu procesów multimodalnych			
${}^{(i)}\sigma_1^1$	$({}^{(i+2)}m^1P_2^1, {}^{(i+2)}m^1P_1^1)$	${}^{(i)}\sigma_7^0$	$({}^{(i)}m^1P_2^1, {}^{(i)}m^1P_1^1)$
${}^{(i)}\sigma_2^1$	$({}^{(i)}m^1P_2^1, {}^{(i)}m^1P_1^1)$	${}^{(i)}\sigma_9^0$	$({}^{(i+3)}m^1P_1^1, {}^{(i+3)}m^1P_2^1)$
${}^{(i)}\sigma_3^0$	$({}^{(i)}m^1P_2^1, {}^{(i)}m^1P_1^1)$		

Uzyskany, zgodnie z (118), harmonogram stanowi składową harmonogramu  $X^1$  charakteryzującego zachowanie całej struktury  $SC^1$ . Harmonogram  $X^1$ , będący wielokrotnym złożeniem harmonogramów  ${}^{(i)}X^1$ , przedstawia rys. 4.18. Łatwo zauważyć, że połączenie harmonogramów  ${}^{(i)}X^1$  wszystkich podstruktur izomorficznych struktury  $SC^1$  nie prowadzi do konfliktów między operacjami – harmonogramy  ${}^{(i)}X^1$  „zazębiają” się na zasobach  ${}^{(i)}R_1$ ,  ${}^{(i)}R_3$  i  ${}^{(i)}R_9$ . Zgodnie też z (111), na podstawie harmonogramu  $X^1$ , dla wszystkich zasobów struktury  $SC^1$  można wyznaczyć związane z nimi reguły priorytetowania; zestawione w tab. 4.7.

Podsumowując, zachowanie cykliczne jest osiągalne w systemie o strukturze  $SC^1$ , jeżeli czasy wykonywania operacji oraz reguły priorytetowania przyjmą wartości jak pokazane w tab. 4.5 i 4.7.

Przestawiony przykład pokazuje jak poprzez ocenę tylko fragmentu struktury regularnej można dokonać oceny zachowania cyklicznego całej struktury. W ogólnym przypadku, w analogiczny sposób można rozważać struktury regularne składające się z wielu grup różnych typów podstruktur izomorficznych łączonych na wiele sposobów. Ilustrację tej możliwości przedstawia przykład czterech grup podstruktur izomorficznych  ${}^{(i)}SC_I^1$ ,  ${}^{(i)}SC_{II}^1$ ,  ${}^{(i)}SC_{III}^1$ ,  ${}^{(i)}SC_{IV}^1$  (patrz rys. 4.19a). Podstruktury te mogą być łączone poprzez zasoby określone kolorem szarym. To jakie połączenia są dopuszczalne oznaczono dodatkowo strzałkami. Przykład struktury regularnej zbudowanej z tych podstruktur przedstawia rys. 4.19b.

W ogólnym przypadku można powiedzieć, że w strukturze zbudowanej z podstruktur należących do różnych grup izomorficznych:

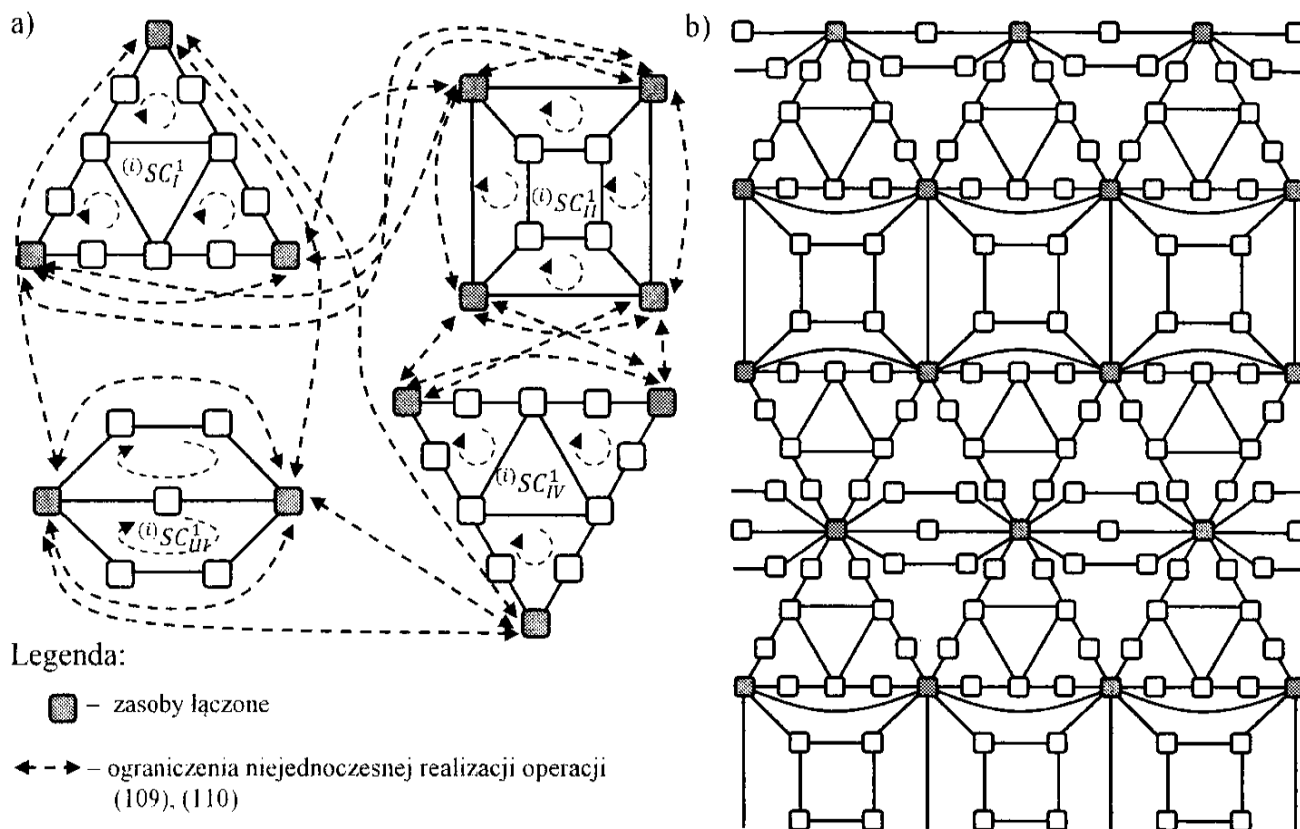
$$SC^{lp} = \ddot{\oplus}_{i=1}^{lc_I} ({}^{(i)}SC_I^1) \ddot{\oplus} \ddot{\oplus}_{i=1}^{lc_{II}} ({}^{(i)}SC_{II}^1) \ddot{\oplus} \dots \ddot{\oplus} \ddot{\oplus}_{i=1}^{lc_q} ({}^{(i)}SC_q^1), \quad (121)$$

gdzie:  $lc_j$  – liczba podstruktur typu  ${}^{(i)}SC_j^1$ ,

$$\ddot{\oplus}_{i=1}^{lc} ({}^{(i)}SC^1) = ({}^1)SC^1 \ddot{\oplus} \dots \ddot{\oplus} ({}^{(i)}SC^1) \ddot{\oplus} \dots \ddot{\oplus} ({}^{lc})SC^1,$$

cykliczny przebieg ustalony o harmonogramie  $X^{lp}$  jest osiągalny, jeżeli istnieje rozwiązanie problemów (119), (120) opisujących zbiór podstruktur:  ${}^{(i)}SC_I^1$ ,  ${}^{(i)}SC_{II}^1$ , ...,  ${}^{(i)}SC_q^1$ , rozwiązanie uwzględniające ograniczenia narzucane na sposób ich łączenia (np. jak na rys. 4.19a).





**Rys. 4.19.** Zbiór czterech grup podstruktur izomorficznych  $(i)SC_I^1$ ,  $(i)SC_{II}^1$ ,  $(i)SC_{III}^1$ ,  $(i)SC_{IV}^1$  a) oraz przykład struktury regularnej zbudowanej z tych podstruktur b)

Podsumowując, wyznaczanie parametrów struktury regularnej poprzez ocenę jej fragmentu jest możliwe dzięki wykorzystaniu reguły „zamka błyskawicznego”. Zasada ta pozwala wyznaczyć harmonogram danej podstruktury (jak i parametry go gwarantujące), realizowany naprzemiennie z odpowiadającymi mu harmonogramami podstruktur sąsiadujących. Wyznaczanie harmonogramu odbywa się przy użyciu metody przedstawionej w poprzednim rozdziale (Algorytm 6).

Zakres zastosowań proponowanego podejścia ogranicza się do przypadków struktur regularnych, w których liczba  $LO$  operacji procesów realizowanych w ramach zbioru grup podstruktur izomorficznych (rys. 4.19a) nie przekracza 40 (zgodnie z rys. 3.27).

### 4.3. Podsumowanie

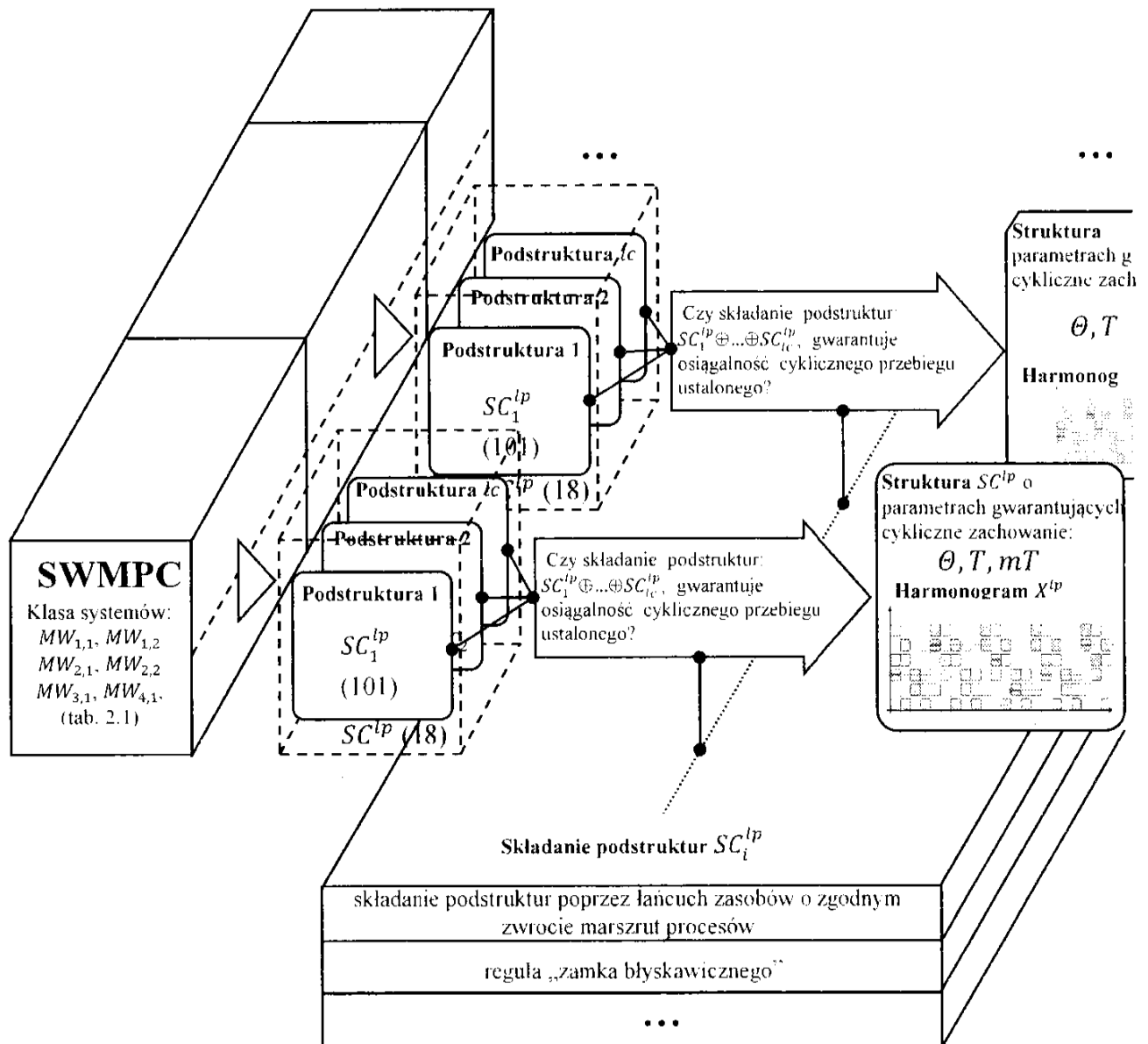
Przedstawione, szczególne przypadki problemów składania struktur SWMPC sprowadzają się do określania postaci operatora  $\oplus$  gwarantującego cykliczność systemu o strukturze wynikowej. W tym celu zaproponowane zostały dwa podejścia oparte na:

- warunkach składania podstruktur wzdłuż łańcucha zasobów o zgodnym zwrocie marszrut procesów,
- regule „zamka błyskawicznego”.

Pierwsze podejście, wykorzystujące specyficzne cechy struktur składanych, pozwala ocenić istnienie zachowania cyklicznego w strukturze wynikowej bez koniecz-

ności wyznaczania harmonogramu cyklicznego. Zakres zastosowań takiego podejścia jest jednak ograniczony do struktur, w których dochodzi do zgodnej realizacji procesów (marszruty procesów na łączonych zasobach mają ten sam zwrot).

Drugie podejście, wykorzystujące zasadę „zamka błyskawicznego”, pozwala określać zachowanie struktury wynikowej w oparciu o harmonogramy cykliczne osiągalne w jej podstrukturach. Zgodnie z użytą regułą, w strukturze wynikowej osiągalne jest zachowanie cykliczne, jeżeli operacje realizowane na zasobach łączonych są wykonywane naprzemiennie – harmonogramy „zazębiają” się. Uzyskany w ten sposób harmonogram cykliczny stanowi połączenie harmonogramów łączonych podstruktur.



Rys. 4.20. Składanie podstruktur SWMPC, uwzględniające proponowane podejścia

Reguła „zamka błyskawicznego” może być również wykorzystywana przy rozwiązywaniu problemu składania, który może być postrzegany jako szczególny przypadek problemu syntezy. Prezentowane przykłady pokazują, że stosowanie dekompozycji pozwala na znaczną redukcję przestrzeni poszukiwań i tym samym zwiększenie efektywności proponowanych algorytmów (np. Algorytm 6).

Omawiane podejścia do składania struktur systemów SWMPC wpisują się metodykę (patrz rys. 4.20), której wstępne założenia przedstawiono w rozdziale pierwszym

(rys. 1.25). Wykorzystane w ramach tych podejść warunki, m.in. Twierdzenie 6 i Twierdzenie 7 a także i (105), (109) – (111), pełnią rolę warunków wystarczających istnienia cyklicznych zachowań SWMPC. Oznacza to, że zakres ich zastosowań (jak i związana z nimi efektywność obliczeń) ogranicza się tylko do struktur spełniających specyficzne założenia (związane np. z naprzemienną realizacją potoków na wspólnych zasobach, itp.).

Jednym z przykładów wykorzystania podejścia bazującego na regule „zamka błyskawicznego” jest problem syntezy struktur regularnych. Ze względu na możliwość oceny osiągalności zachowania cyklicznego całej struktury poprzez ocenę zachowania tylko pojedynczej struktury izomorficznej (lub zbioru grup struktur izomorficznych), skala rozwiązywalnych problemów jest znacznie większa niż w przypadku podejść prezentowanych w rozdziale 3.

## 5. Re-harmonogramowanie SWMPC

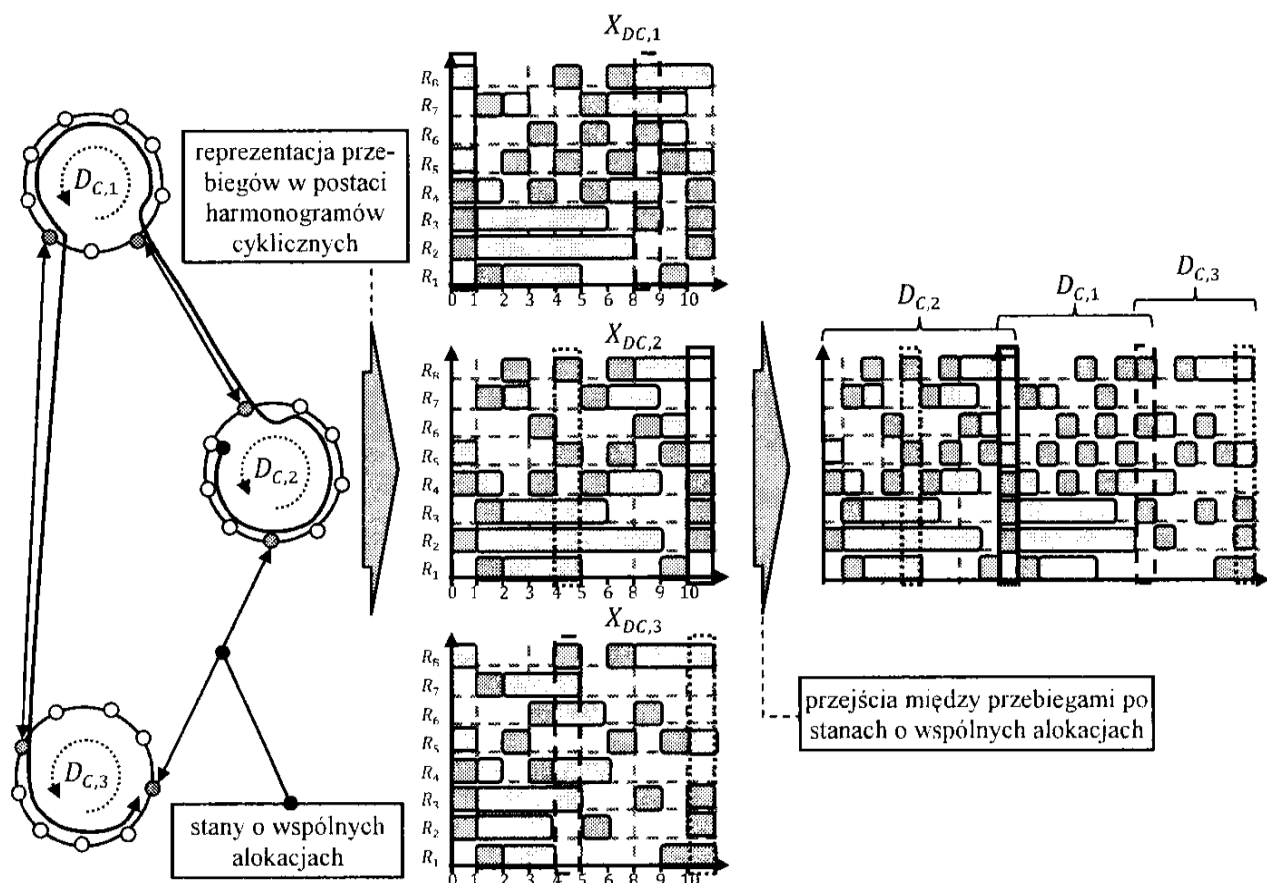
Przyjęcie struktury  $SC^{lp}$  i zachowania  $\mathcal{P}$  jako podstawowych składowych deklarytywnego modelu SWMPC, pozwala rozważać problemy koncentrujące się na istnieniu zachowań cyklicznych oraz poszukiwaniu wartości parametrów struktur warunkujących takie zachowania. Przyjmując, że zachowanie systemu o strukturze  $SC^{lp}$  (18) (scharakteryzowanej przez zbiory zasobów, zbiory marszrut procesów lokalnych i multimodalnych, czasy realizacji operacji, reguły priorytetowania, itd.), wyrażane jest przez przestrzeń stanów  $\mathcal{P}$ , na którą składają się komponenty spójności typu wir lub drzewo, zakłada się jednocześnie, że każdy stan dopuszczalny przestrzeni  $\mathcal{P}$  generuje pewną instancję zachowania (komponent spójności). Oznacza to, że zachowanie systemu jest determinowane przez przyjęty stan początkowy i, zgodnie z Własnością 6 raz osiągnięte, nie może ulec zmianie. W tym ujęciu interesująca staje się kwestia wzajemnej osiągalności zachowań cyklicznych. Ograniczając się w dalszych rozważaniach do zachowań obejmujących zbiory cyklicznych przebiegów ustalonych, poszukiwane są odpowiedzi inspirowane poniższymi pytaniami:

- Spełnienie jakich warunków gwarantuje osiągalność żądanego zachowania cyklicznego z zadanego zachowania cyklicznego?
- Spełnienie jakich warunków gwarantuje osiągalność żądanego zachowania cyklicznego z zadanego zachowania cyklicznego bez pośrednictwa przebiegu przejściowego?
- Spełnienie jakich warunków gwarantuje, że w przestrzeni cyklicznych przebiegów ustalonych istnieją cykliczne przebiegi ustalone złożone z fragmentów „lokalnych” cyklicznych przebiegów ustalonych?
- ...

Pytania tego typu wiążą się z tzw. **problemem wzajemnej osiągalności cyklicznych przebiegów ustalonych**. Przyjmując, że w systemie o danej strukturze osiągalnych jest wiele zachowań cyklicznych problem ten sprowadza się do pytania: spełnienie jakich warunków zapewnia przejście pomiędzy jego wybranymi cyklicznymi zachowaniami?

Możliwość zmiany przebiegu cyklicznego i tym samym zmiany harmonogramu w trakcie jego realizacji wpisuje się w obszar problemów re-harmonogramowania. W tym ujęciu poszukiwane warunki mogą gwarantować istnienie bądź to bezpośredniego, bądź też pośredniego przejścia pomiędzy zadanymi instancjami zachowań cyklicznych. W dalszych rozważaniach przyjmuje się, że stan umożliwiający przejście pomiędzy cyklicznymi zachowaniami systemu zachowuje przyjętą w tym stanie alokację procesów na zasobach. Oznacza to, że zakładana zmiana zachowania odbywać się będzie na drodze zmian parametrów struktury modyfikujących działanie reguł priorytetowania.

Poszukiwane warunki gwarantujące przejścia pomiędzy wybranymi zachowaniami wyznaczane są w procesie analizy rzutów stanów wchodzących w skład przebiegów cyklicznych. Ogólną ideę proponowanego podejścia ilustruje rys. 5.1.



Rys. 5.1. Analiza potencjalnych przejść pomiędzy cyklicznymi przebiegami ustalonymi

### 5.1. Problem wzajemnej osiągalności cyklicznych przebiegów ustalonych

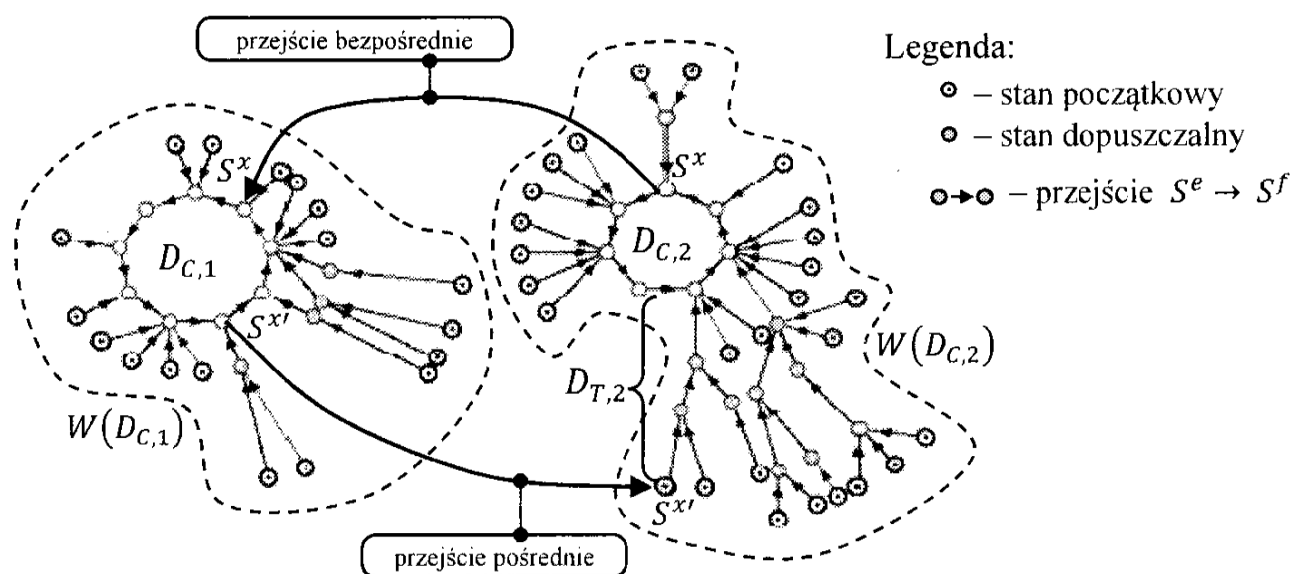
Z dotychczasowych rozważań wynika, że instancje zachowań osiągalnych w SWMPC mogą być wyrażane w różnych reprezentacjach. W przypadku zachowania cyklicznego, może być ono reprezentowane przez harmonogram cykliczny  $X^{lp}$  (95) lub cykliczny przebieg ustalony  $D_C$  (reprezentowany przez digraf  $G(D_C)$  (57) będący elementem wiru  $W(D_C)$  (59)). Jak pokazano to na przykładzie digrafów pierwszeństwa operacji (podrozdział 3.2.1 – rys. 3.17) reprezentacje te są równoważne. Każdemu harmonogramowi cyklicznemu odpowiada cykliczny przebieg ustalony i odwrotnie każdemu przebiegowi cyklicznemu odpowiada harmonogram cykliczny.

W rozważanym kontekście problem wzajemnej osiągalności zachowań cyklicznych (inaczej re-harmonogramowania) odnosi się do wzajemnej osiągalności cyklicznych przebiegów ustalonych. Oznacza to, że osiągalność przebiegu ustalonego  $D_{C,2}$  z przebiegu  $D_{C,1}$  rozumiana jest jako możliwość przejścia z jednego ze stanów zbioru  $W(D_{C,1})$  (zbioru stanów składających się na przebieg  $D_{C,1}$ ) do stanu ze zbioru  $W(D_{C,2})$  (zbioru stanów składających się na przebieg  $D_{C,2}$ ) – jest to tzw. przejście bezpośrednie, albo do stanu ze zbioru  $V_W(D_{C,2})$  (zbiór stanów należących do wiru  $W(D_{C,2})$ ) prowadzącego do przebiegu  $D_{C,2}$  – jest to tzw. przejście pośrednie. Problem wzajemnej osiągalności cyklicznych przebiegów ustalonych jest więc definiowany następująco:

*Dana jest struktura  $SC^{lp}$  (18) oraz generowana przez nią przestrzeń stanów  $\mathcal{P} = (\mathbb{S}^{lp}, \mathbb{E}^{lp})$ , zawierająca dwa cykliczne przebiegi ustalone  $D_{C,1}, D_{C,2} \in DC$ . Poszuki-*

wana jest odpowiedź na następujące pytanie: Czy przebiegi  $D_{C,1}$ ,  $D_{C,2}$  są wzajemnie osiągalne (rys. 5.2)? A w szczególności: Jakie warunki muszą być spełnione aby istniały pośrednie i/lub bezpośrednie przejścia między tymi przebiegami?

Przykład ilustrujący bezpośrednie i pośrednie przejścia pomiędzy zadanymi instancjami zachowań cyklicznych przedstawia rys. 5.2. W rozważanym przypadku przejście pośrednie sprowadza się do zmiany przebiegu  $D_{C,1}$  na przebieg  $D_{C,2}$ , będącej wynikiem przejścia ze stanu  $S^{x'}$  do przebiegu przejściowego  $D_{T,2}$ , prowadzącego do  $D_{C,2}$ . Z kolei, przejście bezpośrednie polega na zmianie przebiegu  $D_{C,2}$  na przebieg  $D_{C,1}$  w tym samym stanie  $S^x$ .



Rys. 5.2. Wzajemna osiągalność cyklicznych przebiegów ustalonych  $D_{C,1}$ ,  $D_{C,2}$

Łatwo zauważyć, że rozważane przejścia są warunkowane istnieniem stanu  $S^x$  (lub  $S^{x'}$ ) będącego jednocześnie elementem dwóch digrafów zachowań  $W(D_{C,1})$  oraz  $W(D_{C,2})$  (wirów, w skład których wchodzi stany przebiegów cyklicznych  $D_{C,1}$ ,  $D_{C,2}$ ). Inaczej mówiąc, w przypadku bezpośredniej osiągalności przebiegów  $D_{C,1}$ ,  $D_{C,2}$  wymagane jest by dochodziło do następujących przejść między stanami:

- $\exists S^k \in V(D_{C,1}), \exists S^x \in V(D_{C,2}) (S^k \rightarrow \dots \rightarrow S^x \rightarrow \dots \rightarrow S^k)$ ,
- $\exists S^l \in V(D_{C,2}), \exists S^x \in V(D_{C,1}) (S^l \rightarrow \dots \rightarrow S^x \rightarrow \dots \rightarrow S^l)$ .

Przejścia tego typu nie są dopuszczalne ponieważ istnienie stanu  $S^x$  należącego jednocześnie do dwóch digrafów zachowań jest sprzeczne z Własnością 6, a także z Własnością 1, mówiącą że każdy stan ma co najwyżej jednego potomka (patrz podrozdział 2.2.2).

O ile brak wspólnych stanów nie jest możliwy na najwyższym poziomie przestrzeni zachowań  $\mathcal{P}^{lp}$  o tyle ich rzuty na niższe poziomy  $\mathcal{P}^l$  jak też i ich składowe (alokacje, semafor, indeksy) mogą być wspólne dla wielu stanów jednocześnie.

Przykład takiej sytuacji przedstawia rys. 5.3 ilustrujący kolejne rzuty  $\mathcal{P}^1$  i  $\mathcal{P}^0 = Ev(\mathcal{P}^1)$  (patrz podrozdział 2.2.3) pewnej przestrzeni stanów  $\mathcal{P}$ . W celu lepszego zo-

brazowania rzutowanych stanów, związane z nimi instancje przebiegów cyklicznych przestrzeni  $\mathcal{P}^1$  zilustrowano spiralo-kształtnymi postaciami digrafów, w których „wysokości” rozmieszczonych stanów  $S^r$  odpowiadają momentom ich wystąpienia.

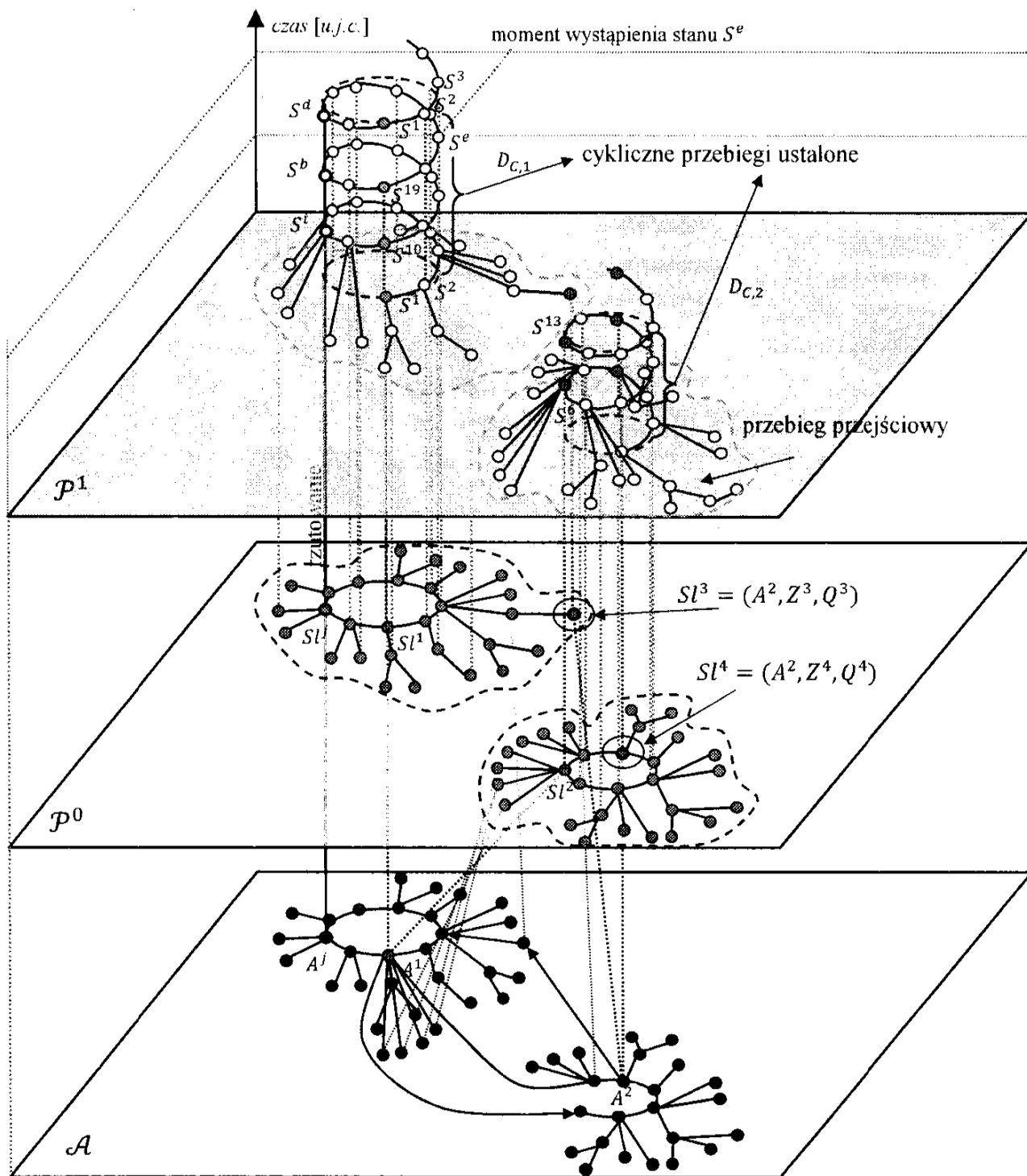
Niech  $\mathcal{A}$  oznacza przestrzeń będącą rzutem przestrzeni  $\mathcal{P}^0 = (\mathbb{S}^0, \mathbb{E}^0)$  względem alokacji stanów  $\mathbb{S}^0$ :  $\mathcal{A} = (\mathbb{A}, \mathbb{E}^{\mathcal{A}})$ , gdzie:  $\mathbb{A} = \{A^r \mid Sl^r = (A^r, Z^r, Q^r) \in \mathbb{S}^0\}$  – jest zbiorem alokacji, a  $\mathbb{E}^{\mathcal{A}} = \{(A^r, A^q) \mid (S^r, S^q) \in \mathbb{E}^0, Sl^r = (A^r, Z^r, Q^r) \in \mathbb{S}^0, Sl^q = (A^q, Z^q, Q^q) \in \mathbb{S}^0\}$  – jest zbiorem łuków. Przestrzeń  $\mathcal{A}$  stanowi najniższy poziom modelu z rys. 5.3.

Elementami tak zdefiniowanej przestrzeni są alokacje  $A^r$  stanów  $Sl^r = (A^r, Z^r, Q^r)$  występujących w przestrzeni  $\mathcal{P}^0 = (\mathbb{S}^0, \mathbb{E}^0)$  (gdzie  $Sl^r \in \mathbb{S}^0$ ). Łatwo zauważyć, że niektóre z alokacji przestrzeni  $\mathcal{A}$  są wspólne dla kilku różnych stanów. Przykładowo, alokacja  $A^2 \in \mathbb{A}$  jest jednocześnie elementem stanu  $Sl^3 = (A^2, Z^3, Q^3)$  i stanu  $Sl^4 = (A^2, Z^4, Q^4)$ . W praktyce oznacza to, że w obu stanach procesy lokalne (reprezentujące np. środki transportu typu: pociągi, wózki, itp.) są zaalokowane na tych samych zasobach. Ze względu jednak na różne prawa dostępu do zasobów (określone przez  $Z^3, Q^3$  oraz  $Z^4, Q^4$ ) ich dalsza realizacja ma różny przebieg.

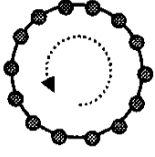
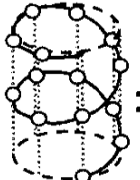
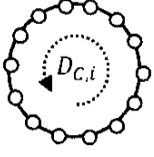
Warto zwrócić uwagę, że istnienie stanów o wspólnej alokacji może zostać wykorzystane do przejścia między różnymi przebiegami cyklicznymi. Stany  $Sl^3$  i  $Sl^4$  różnią się wyłącznie sekwencją semaforów i indeksów, jeżeli więc w stanie  $Sl^3$  o alokacji  $A^2$  zmianie ulegnie postać  $Z^3, Q^3$  np. w ich miejsce wykorzystane zostaną  $Z^4, Q^4$  to otrzymany w wyniku tego stan  $Sl^4$  prowadzi do innego (nowego) przebiegu cyklicznego. Oznacza to, że w wyniku modyfikacji stanu  $Sl^3$  na  $Sl^4$ , modyfikacji polegającej na zastąpieniu semaforów  $Z^3$  i indeksów  $Q^3$  odpowiednio na  $Z^4, Q^4$  możliwe jest przejście pośrednie pomiędzy przebiegami cyklicznymi przestrzeni  $\mathcal{P}^0$ . Przejście bezpośrednie jest z kolei możliwe w wyniku modyfikacji semaforów i indeksów stanów  $Sl^1 = (A^1, Z^1, Q^1)$  i  $Sl^2 = (A^1, Z^2, Q^2)$  współdzielących wspólną alokację  $A^1$ . Widać więc, że istnienie stanów o wspólnej alokacji implikuje możliwość przejść pomiędzy ustalonymi przebiegami cyklicznymi.

Rozważane przejścia między stanami  $Sl^3$  i  $Sl^4$  oraz stanami  $Sl^1$  i  $Sl^2$  odnoszą się do przebiegów występujących jedynie na poziomie lokalnym (przestrzeń  $\mathcal{P}^0$ ). W przypadku poszukiwania przejść łączących przebiegi cykliczne przestrzeni  $\mathcal{P}^1$  stosowne postępowanie przebiega analogicznie. Stany  $Sl^3, Sl^4, Sl^1$  oraz  $Sl^2$  są rzutami odpowiednich stanów z przestrzeni  $\mathcal{P}^1$ . Stan  $Sl^1$  jest rzutem trzech stanów przebiegu  $D_{C,1}$ :  $S^1 = (Sl^1, m^1S^1)$ ,  $S^{10} = (Sl^1, m^1S^{10})$  oraz  $S^{19} = (Sl^1, m^1S^{19})$ , a  $Sl^2$  jest rzutem dwóch stanów  $D_{C,2}$ :  $S^6 = (Sl^2, m^1S^6)$ ,  $S^{13} = (Sl^2, m^1S^{13})$ . Jeżeli więc wśród stanów o rzutach postaci  $Sl^1$  i  $Sl^2$  występują stany (np. stany  $S^{10}$  i  $S^6$ ), posiadające wspólną alokację  $m^1A^x$  procesów multimodalnych, to możliwe jest przejście pomiędzy  $D_{C,1}$  a  $D_{C,2}$  – przejście uzyskane w wyniku zastąpienia semaforów i odpowiadających im indeksów z postaci  $m^1Z^{10}, m^1Q^{10}$  na odpowiadające im postaci  $m^1Z^6, m^1Q^6$ .

Stany  $S^{10}, S^6$  są stanami, które charakteryzują się wspólną alokacją zarówno procesów lokalnych jak i multimodalnych. Dokonując zmiany semaforów i indeksów w stanie  $S^6$  można przejść bezpośrednio z przebiegu  $D_{C,1}$  do przebiegu  $D_{C,2}$ .



Legenda:

- ⊙ –  $r$ -ty stan procesów lokalnych:  $Sl^r = (A^r, Z^r, Q^r)$     ○ –  $r$ -ty stan SWMPC  $S^r = (Sl^r, m^1 S^r)$
- ⊙ → ⊙ – przejście  $Sl^a \rightarrow Sl^b$     ○ → ○ – przejście  $S^a \rightarrow S^b$     ● –  $r$ -ta alokacja  $A^r$
-  – rzut przebiegu cyklicznego  $D_{C,i}$  na przestrzeń  $P^0$      =  – cykliczny przebieg ustalony  $D_{C,i}$

Rys. 5.3. Przykład rzutów przestrzeni zachowań:  $P^1$ ,  $P^0$  oraz  $\mathcal{A}$



W praktyce, przejście tego typu sprowadza się do zmiany sygnalizacji (będącej skutkiem zmiany semaforów i/lub indeksów) lub też do zmiany przyjętego sposobu sterowania (reguł priorytetowania) dokonywanego w momentach odpowiadających stanom o odpowiednio zaalokowanych procesach (np. zgodnego z  $S^6$ ). Oznacza to, że do zmiany zachowań systemu jest wymagana korekta jego struktury, w szczególności nie jest konieczne zatrzymywanie systemu celem przeprowadzenia odpowiedniej zmiany alokacji procesów (np. zmiana pozycji wózków samojezdnych, pociągów itp.). W rozważanym przypadku, modyfikacja stanu  $S^6$ , przejście pomiędzy instancjami zachowań dotyczy tylko korekty parametrów sterowania systemu.

Reasumując, przejście pomiędzy dwoma przebiegami cyklicznymi  $D_{C,1}$  i  $D_{C,2}$  jest możliwe jeżeli w skład tych przebiegów (lub przebiegów przejściowych do nich prowadzących) wchodzi stany charakteryzujące się wspólną alokacją na każdym poziomie zachowań. Spostrzeżenie to prowadzi do następujących dwóch własności:

#### Własność 8

Jeżeli istnieją stany  $S^a \in V(D_{C,1})$ ,  $S^b \in V_W(D_{C,2})$  (gdzie:  $V_W(D_{C,2})$  – zbiór stanów należących do wiru  $W(D_{C,2})$ ) posiadające wspólną alokację procesów lokalnych i multimodalnych:  $A^a = A^b$ ;  $m^l A^a = m^l A^b$  dla  $l = 1 \dots lp$ , to przebieg cykliczny  $D_{C,2}$  jest osiągalny z przebiegu  $D_{C,1}$  (oznaczane przez  $D_{C,1} \rightarrow D_{C,2}$ ).

□

#### Własność 9

Dwa przebiegi cykliczne  $D_{C,1}$  i  $D_{C,2}$  są wzajemnie osiągalne (oznaczane przez  $D_{C,1} \leftrightarrow D_{C,2}$ ) jeżeli  $D_{C,1} \rightarrow D_{C,2}$  oraz  $D_{C,2} \rightarrow D_{C,1}$ .

□

Tak więc, rozważany problem osiągalności cyklicznych przebiegów ustalonych sprowadza się do pytania: Czy istnieją stany:  $S^a \in V(D_{C,1})$  i  $S^b \in V_W(D_{C,2})$ , współdzielące te same alokacje:  $A^a = A^b$ ;  $m^l A^a = m^l A^b$  dla  $l = 1 \dots lp$ ?

Jeżeli stany takie istnieją to przejście pomiędzy nimi (a tym samym pomiędzy instancjami przebiegów) wiąże się z zmianą odpowiednich semaforów i wartości indeksów. Poniżej przedstawiono formalną definicję tego typu przejścia.

Dane są dopuszczalne stany systemu SWMPC (19):  $S^a \neq S^b$ ,  $S^a, S^b \in \mathbb{S}^{lp}$ . Stan  $S^b$  jest osiągalny ze stanu  $S^a$  w wyniku zmiany semaforów i indeksów (oznaczane jako:  $S^a \rightsquigarrow S^b$ ), jeżeli stany te współdzielą te same alokacje na każdym poziomie zachowań  $A^a = A^b$ ;  $m^l A^a = m^l A^b$  dla  $l = 1 \dots lp$ .

W szczególności oznacza to, że przejście bezpośrednie jest określane jako:

$$\dots \rightarrow S^{a_1} \rightarrow S^{a_2} \rightarrow \dots \rightarrow S^{a_{(x-1)}} \rightarrow (S^{a_x} \rightsquigarrow S^{b_x}) \rightarrow S^{b_{(x+1)}} \rightarrow \dots \quad (122)$$

a przejście pośrednie jako:

$$\dots \rightarrow S^{a_1} \rightarrow \dots \rightarrow (S^{a_x} \rightsquigarrow S^{d_x}) \rightarrow \dots \rightarrow S^{d_{ld}} \rightarrow S^{b_j} \rightarrow S^{b_{(j+1)}} \dots \quad (123)$$

gdzie:

$S^{a_1}, \dots, S^{a_{(x-1)}}, S^{a_x}, \dots, S^{a_{ld}}$  – stany składające się na przebieg  $D_{C,1}$ ,  $S^{a_x} \in V(D_{C,1})$ ,  
 $S^{b_1}, \dots, S^{b_{(x-1)}}, S^{b_x}, \dots, S^{b_{ld}}$  – stany składające się na przebieg  $D_{C,2}$ ,  $S^{b_x} \in V(D_{C,2})$ ,

$S^{d_1}, \dots, S^{d_x}, S^{d_{(x+1)}}, \dots, S^{d_{lb}}$  – stany składające się na przebieg przejściowy  $D_T$ ,  
 prowadzący do przebiegu  $D_{C,2}$ ,  $S^{d_x} \in V_W(D_{C,2})$ ,  
 $S^{a_1} \rightarrow S^{a_2}$  – przejście między stanami zgodne z (32) – (50),  
 $S^{a_x}, S^{b_x}$  – stany o wspólnej alokacji,  $A^{a_x} = A^{b_x}$ ;  $m^l A^{a_x} = m^l A^{b_x}$  dla  $l = 1 \dots lp$ ,  
 $S^{a_x} \rightsquigarrow S^{b_x}$  – przejście ze stanu  $S^{a_x}$  do  $S^{b_x}$  w wyniku zmiany semaforów i indeksów z:  $Z^{a_x}, Q^{a_x}, m^1 Z^{a_x}, m^1 Q^{a_x}, \dots, m^{lp} Z^{a_x}, m^{lp} Q^{a_x}$  na  $Z^{b_x}, Q^{b_x}, m^1 Z^{b_x}, m^1 Q^{b_x}, \dots, m^{lp} Z^{b_x}, m^{lp} Q^{b_x}$ .

## 5.2. Procedura wyznaczania stanów o wspólnej alokacji

W sytuacji gdy znane są cykliczne przebiegi ustalone  $D_{C,1}$ ,  $D_{C,2}$  oraz prowadzące do nich przebiegi przejściowe, wyznaczanie stanów o wspólnej alokacji sprowadza się do przeglądu wszystkich dopuszczalnych w ramach tych przebiegów par stanów, tzn. elementów zbioru  $V(D_{C,1}) \times V_W(D_{C,2})$ . Odpowiedni algorytm ma postać:

### Algorytm 7

```

function STATESCOALL ( $G(D_{C,1}) = (V(D_{C,1}), E(D_{C,1})), W(D_{C,2}) = (V_W(D_{C,2}), E_W(D_{C,2}))$ )
  AC  $\leftarrow \emptyset$ 
  forall  $S^a \in V(D_{C,1})$ 
    forall  $S^b \in V_W(D_{C,2})$ 
      if ( $A^a = A^b$ ) & ( $m^1 A^a = m^1 A^b$ ) &...& ( $m^{lp} A^a = m^{lp} A^b$ ) then
        AC  $\leftarrow AC \cup (S^a, S^b)$ 
      end
    end
  end
  return AC
end
  
```

gdzie:

$G(D_{C,1}) = (V(D_{C,1}), E(D_{C,1})), W(D_{C,2}) = (V_W(D_{C,2}), E_W(D_{C,2}))$  – dane wejściowe: digraf odwzorowujący przebieg  $D_{C,1}$  i wir związany z przebiegiem  $D_{C,2}$ ,  
 $S^a, S^b$  – stany (19) należące do  $V(D_{C,1}), V_W(D_{C,2})$ ,  
 AC – zbiór par  $(S^a, S^b)$  stanów o wspólnej alokacji.

Wynikiem Algorytmu 7 jest zbiór AC par stanów  $(S^a, S^b)$  posiadających wspólne alokacje:  $AC \subseteq V(D_{C,1}) \times V_W(D_{C,2})$ . Stany te mogą zostać wykorzystane przy poszukiwaniu przejść bezpośrednich lub pośrednich pomiędzy przebiegami cyklicznymi  $D_{C1}$ ,  $D_{C2}$ . Przyjmując dla uproszczenia, że rozważane są przejścia bezpośrednie, a przebiegi cykliczne posiadają tę samą liczbę stanów (oznaczaną przez  $ld$ ) złożoność obliczeniową Algorytmu 7 szacuje funkcja kwadratowa  $f(ld) = \mathcal{O}(ld^2)$ .

Algorytm 7 umożliwia ocenę wzajemnej osiągalności tylko dwóch wybranych przebiegów cyklicznych  $D_{C,1}, D_{C,2} \in DC$  ( $DC$  – zbiór wszystkich przebiegów cyklicz-

nych przestrzeni  $\mathcal{P}$ ). Badanie wzajemnej osiągalności wszystkich przebiegów zbioru  $DC$  sprowadza się do oceny osiągalności każdej pary tego zbioru. Złożoność obliczeniowa tego typu procedury wynosi:  $f(ld, dc) = \frac{1}{2}(dc^2 - dc) \cdot ld^2$  ( $dc = |DC|$ ).

Wielomianowy charakter funkcji złożoności obliczeniowej sprawia, że problem oceny wzajemnej osiągalności przebiegów cyklicznych jest problemem łatwym. Oznacza to, że główny wysiłek obliczeniowy związany jest wyłącznie z etapem wyznaczenia przebiegów cyklicznych (zbioru  $DC$ ), w którym wykorzystywane są metody opisane w rozdziale 3. Wśród tych metod jedynie metoda redukcji stanów początkowych (podrozdział 3.1) pozwala wyznaczać wiry niezbędne przy przejściach pośrednich.

Metoda generowania harmonogramów cyklicznych (podrozdział 3.2), nie pozwala wyznaczać przebiegów przejściowych, co ogranicza jej wykorzystanie tylko do przypadków związanych z poszukiwaniem przejść bezpośrednich. Wykorzystanie tej metody przy ocenie wzajemnej osiągalności przebiegów cyklicznych ilustruje poniższy przykład.

**Przykład 8.** Ilustracja osiągalności przebiegów cyklicznych wykorzystująca metodę generowania harmonogramów cyklicznych.

Dany jest system z rys. 1.29 reprezentowany przez SWMPC z rys. 1.30. W systemie wyróżnia się dwa poziomy zachowań:

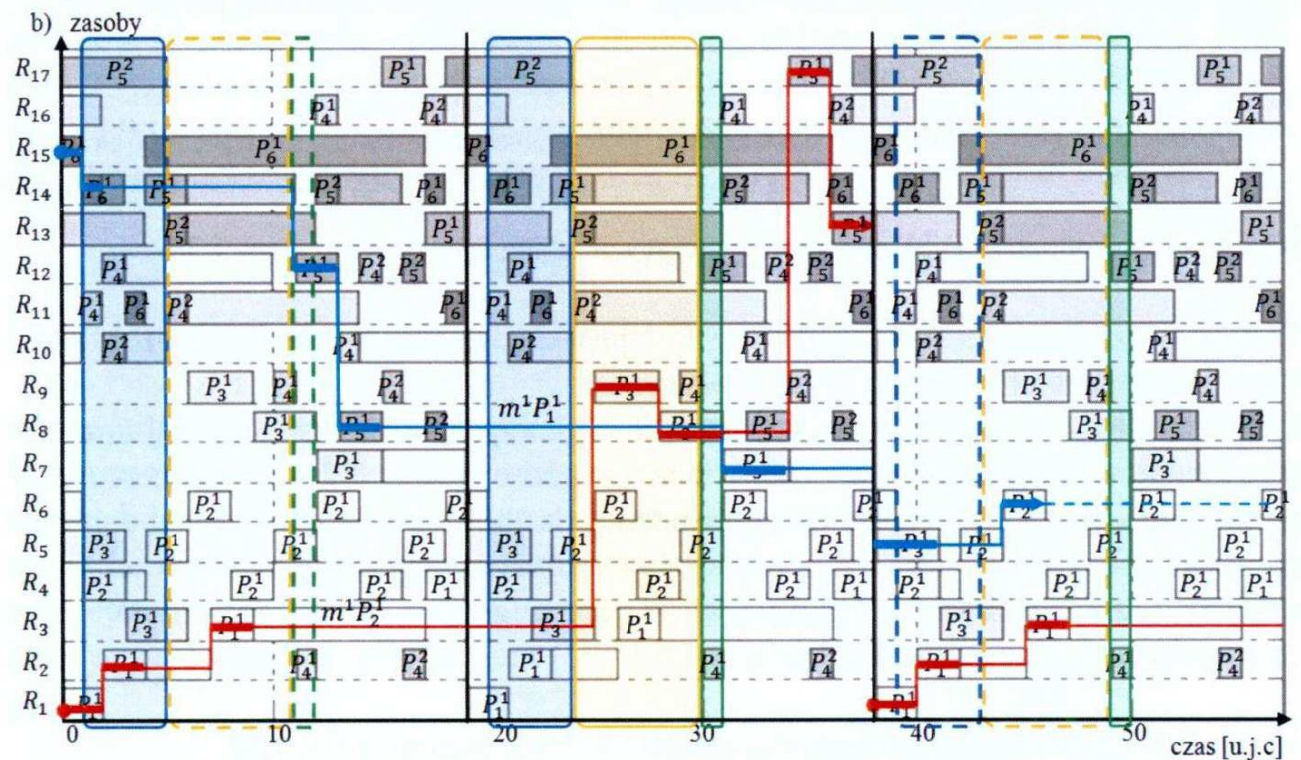
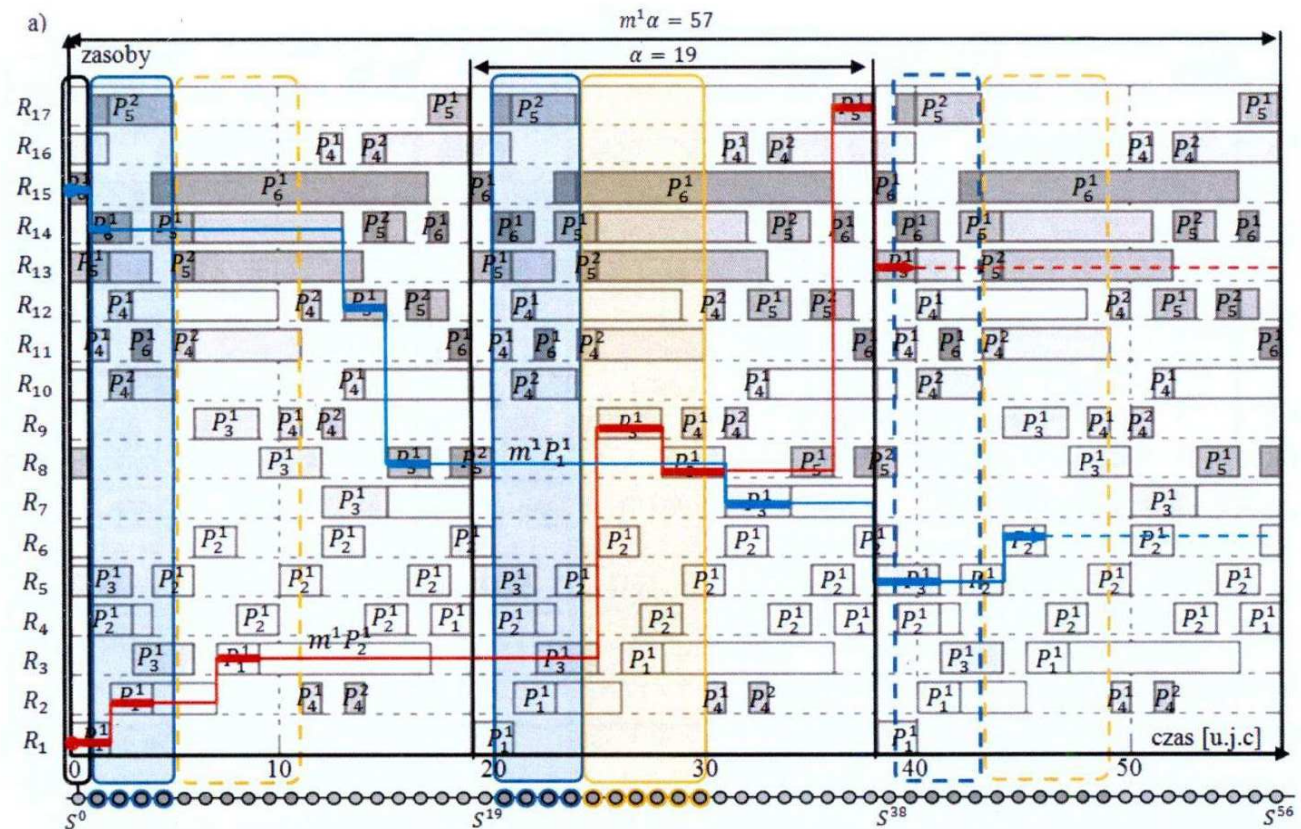
- poziom procesów lokalnych zawierający 6 procesów wielopotokowych:  $P_1 = \{P_1^1\}$ ,  $P_2 = \{P_2^1\}$ ,  $P_3 = \{P_3^1\}$ ,  $P_4 = \{P_4^1, P_4^2\}$ ,  $P_5 = \{P_5^1, P_5^2\}$ ,  $P_6 = \{P_6^1\}$ , reprezentujących pracę wózków samojezdnych,
- poziom procesów multimodalnych zawierający 2 procesy jednopotokowe:  $m^1P_1$ ,  $m^1P_2$ , reprezentujące marszruty transportowe dwóch rodzajów palet.

Ze względu na przyjęte założenie, dopuszczające jednoczesną realizację procesów multimodalnych na wspólnych zasobach, rozważany system jest typu  $MW_{4,2}$  (tab. 2.1). Przyjętą postać reguł priorytetowania  $\theta^0$  przedstawia tab. 5.1. Przyjęto również, że czasy wykonania operacji procesów multimodalnych są takie same jak czasy wymaganych do ich realizacji operacji procesów lokalnych (tab. 5.2).

**Tabela 5.1.** Reguły priorytetowania dla systemu z rys. 1.30

reguły priorytetowania procesów lokalnych					
$\sigma_1^0$	$(P_1^1)$	$\sigma_7^0$	$(P_3^1)$	$\sigma_{13}^0$	$(P_5^1, P_5^2)$
$\sigma_2^0$	$(P_1^1, P_4^1, P_4^2)$	$\sigma_8^0$	$(P_5^1, P_5^2, P_3^1)$	$\sigma_{14}^0$	$(P_5^1, P_5^2, P_6^1, P_6^1)$
$\sigma_3^0$	$(P_1^1, P_3^1)$	$\sigma_9^0$	$(P_3^1, P_4^1, P_4^2)$	$\sigma_{15}^0$	$(P_6^1, P_6^1)$
$\sigma_4^0$	$(P_1^1, P_2^1, P_2^1, P_2^1)$	$\sigma_{10}^0$	$(P_4^1, P_4^2)$	$\sigma_{16}^0$	$(P_4^1, P_4^2)$
$\sigma_5^0$	$(P_2^1, P_2^1, P_2^1, P_3^1)$	$\sigma_{11}^0$	$(P_4^1, P_6^1, P_4^2, P_6^1)$	$\sigma_{17}^0$	$(P_5^1, P_5^2)$
$\sigma_6^0$	$(P_2^1, P_2^1, P_2^1)$	$\sigma_{12}^0$	$(P_4^1, P_4^2, P_5^1, P_5^2)$		

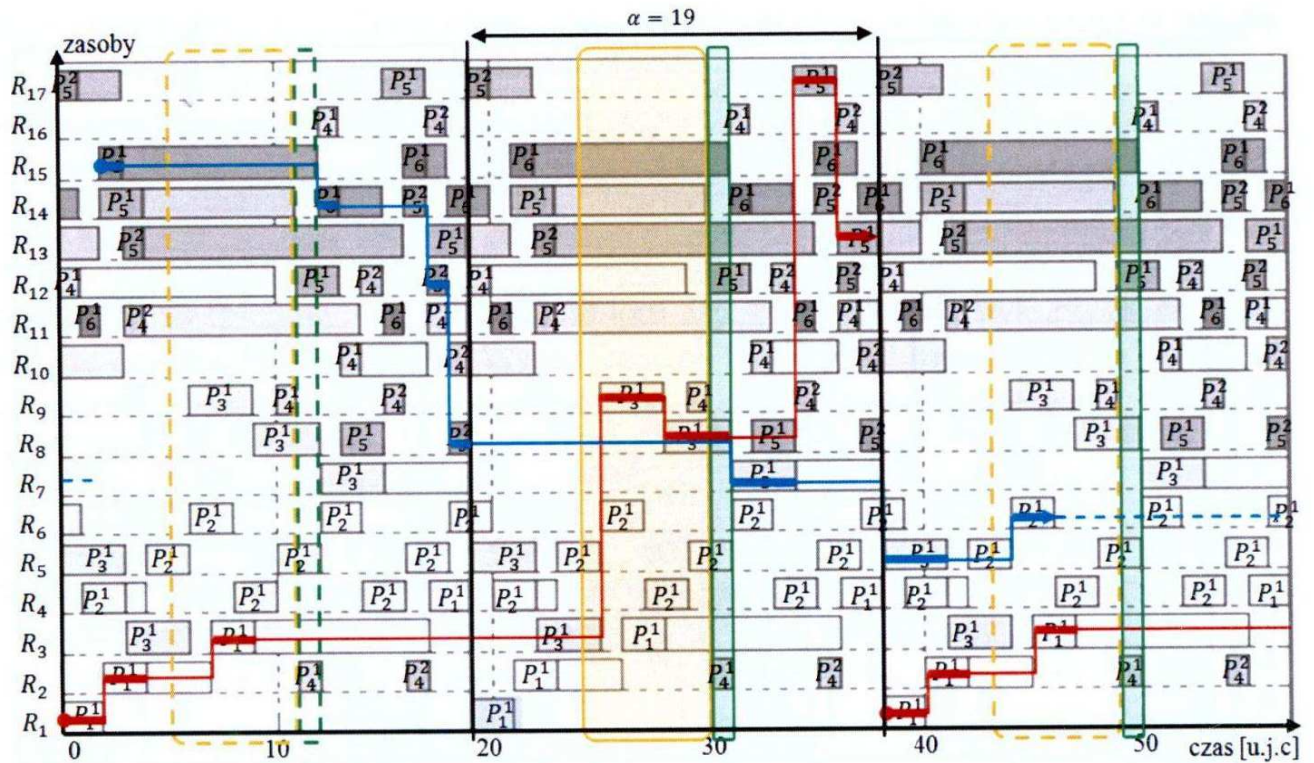
W odróżnieniu od dotychczas rozważanych systemów zakłada się, że niektóre procesy lokalne wykonywane są wielokrotnie względem pozostałych. Takim procesem jest proces  $P_2$ , którego potok jest wykonywany trzykrotnie względem pozostałych.



Legenda:

- $P_i^j$  – realizacja operacji potoku  $P_i^j$
- oczekiwanie potoku  $P_i^j$
- $\rightarrow \odot \leftarrow$  –  $i$ -ty stan SWMPC
- realizacja operacji potoku  $m^1 P_i^h$
- oczekiwanie operacji potoku  $m^1 P_i^h$
- stany przebiegów  $D_{C,1}$  i  $D_{C,2}$  współdzielące alokacje procesów lokalnych i multimodalnych
- stany przebiegów  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$  współdzielące alokacje procesów lokalnych i multimodalnych
- obszar stanów współdzielących tylko alokacje procesów lokalnych przebiegów  $D_{C,1}$  i  $D_{C,2}$
- obszar stanów współdzielących tylko alokacje procesów lokalnych przebiegów  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$

Rys. 5.4. Harmonogram cykliczny reprezentujący przebieg:  $D_{C,1}$  a) i  $D_{C,2}$  b) systemu z rys. 1.30



Legenda:

- obszar stanów współdzielących alokacje procesów lokalnych i multimodalnych przebiegów  $D_{C,3}$  i  $D_{C,2}$
- obszar stanów współdzielących alokacje procesów lokalnych i multimodalnych przebiegów  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$
- obszar stanów współdzielących tylko alokacje procesów lokalnych przebiegów  $D_{C,3}$  i  $D_{C,2}$
- obszar stanów współdzielących tylko alokacje procesów lokalnych przebiegów  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$

**Rys. 5.5.** Harmonogram cykliczny reprezentujący przebieg  $D_{C,3}$  systemu z rys. 1.30

Podobnie potok procesu  $P_6$  wykonywany jest dwukrotnie względem pozostałych. Liczba wzajemnych wykonań procesów wynosi:  $\psi^0 = (1,3,1,1,1,2)$ .

Do wyznaczenia przebiegów cyklicznych wykorzystano metodę generowania harmonogramów cyklicznych przedstawioną w podrozdziale 3.2 i implementowaną w niej Algorytm 3. Wynikiem przeprowadzonych obliczeń (algorytm został zaimplementowany w środowisku OzMozart) są trzy harmonogramy cykliczne reprezentujące kolejno przebiegi  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$ . Harmonogramy te zostały zilustrowane na rys. 5.4a, 5.4b oraz rys. 5.5.

**Tabela 5.2.** Czasy wykonania operacji procesów lokalnych systemu z rys. 1.30

potoki	$i, k$	$t_{i,1}^k$	$t_{i,2}^k$	$t_{i,3}^k$	$t_{i,4}^k$	$t_{i,5}^k$	$t_{i,6}^k$
$P_1^1$	1,1	2	2	2	2	-	-
$P_2^1$	2,1	2	2	2	-	-	-
$P_3^1$	3,1	3	3	3	3	-	-
$P_4^1$	4,1	1	1	1	1	1	1
$P_4^2$	4,2	1	1	1	1	1	1
$P_5^1$	5,1	2	2	2	2	2	-
$P_5^2$	5,2	1	1	1	1	1	-
$P_6^1$	6,1	1	1	1	-	-	-

Wszystkie trzy harmonogramy charakteryzują się tym samym okresem realizacji procesów przebiegów lokalnych  $\alpha = 19$  u.j.c. lecz różnymi okresami realizacji procesów multimodalnych, wynoszącymi odpowiednio  $m^1\alpha = 57$  u.j.c. (dla  $D_{C,1}$ ) oraz  $m^1\alpha = 114$  u.j.c. (dla  $D_{C,2}$  i  $D_{C,3}$ ). Na uwagę zasługuje sposób realizacji poszczególnych procesów multimodalnych (związanych z transportem palet) – uwzględniając czas transportu palet, pierwszy przypadek jest najmniej korzystny (rys. 5.4a)

Realizacja procesów zgodnie z tym harmonogramem powoduje, że w przedziale obejmującym 6 okresów procesów lokalnych (114 u.j.c.), każda z palet zostanie dwukrotnie przetransportowana odpowiednio z  $R_1$  do  $R_{13}$  (paleta czerwona –  $m^1P_1$ ) oraz z  $R_{15}$  do  $R_6$  (paleta niebieska –  $m^1P_2$ ). Czasy transportu w takim przypadku wynoszą odpowiednio: 40 i 46 u.j.c. Realizacja procesów zgodnie z przebiegami  $D_{C,2}$  i  $D_{C,3}$  pozwala w tym samym przedziale czasu (114 u.j.c.) już na trzykrotne przetransportowanie palety  $m^1P_1$ . Ponadto w takim przypadku czasy transportów są krótsze i wynoszą odpowiednio 38 u.j.c., 46 u.j.c. (dla  $D_{C,2}$ ) i 38 u.j.c., 44 u.j.c. (dla  $D_{C,3}$ ).

Niech dany będzie przypadek, w którym realizacja procesów odbywa się zgodnie z przebiegiem  $D_{C,1}$  (rys. 5.4a) i związane z tym pytanie: Czy możliwa jest zmiana tego sposobu na przebieg  $D_{C,3}$  (rys. 5.5), a następnie na  $D_{C,2}$  (rys. 5.4b)?

W procesie poszukiwania rozwiązania tak sformułowanego problemu wzajemnej osiągalności przebiegów cyklicznych  $D_{C,1}$ ,  $D_{C,2}$  i  $D_{C,3}$  wykorzystany został Algorytm 7. Uzyskane przy jego pomocy, zbiory  $AC$  dla każdej pary przebiegów ( $D_{C,1}$ ,  $D_{C,2}$ ), ( $D_{C,1}$ ,  $D_{C,3}$ ), ( $D_{C,3}$ ,  $D_{C,2}$ ) nie są zbiorami pustymi. Oznacza to, że badane przebiegi są wzajemnie osiągalne:

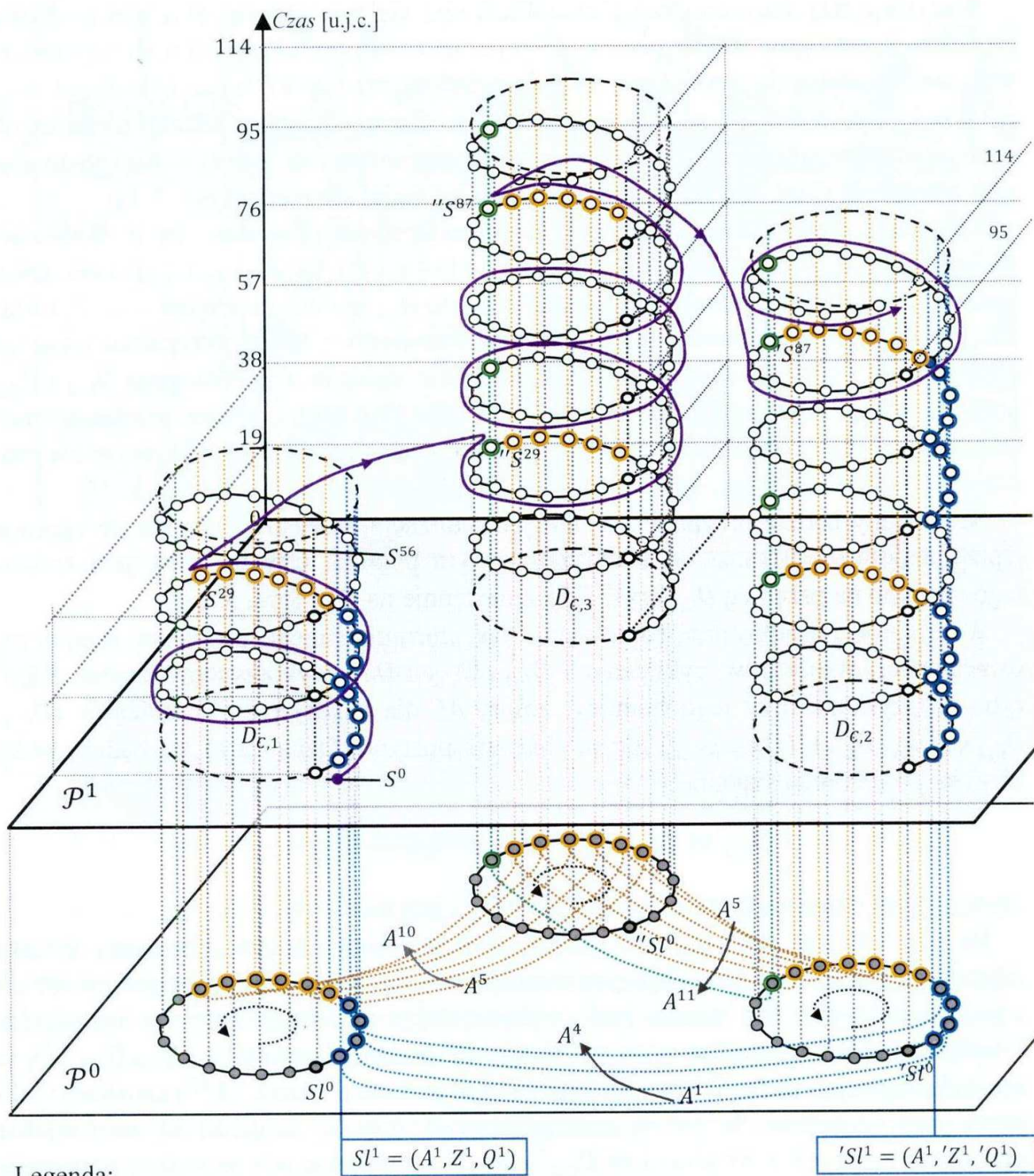
$$D_{C,1} \leftrightarrow D_{C,2}, D_{C,2} \leftrightarrow D_{C,3}, D_{C,3} \leftrightarrow D_{C,2},$$

i tym samym przejście typu:  $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2}$  jest możliwe.

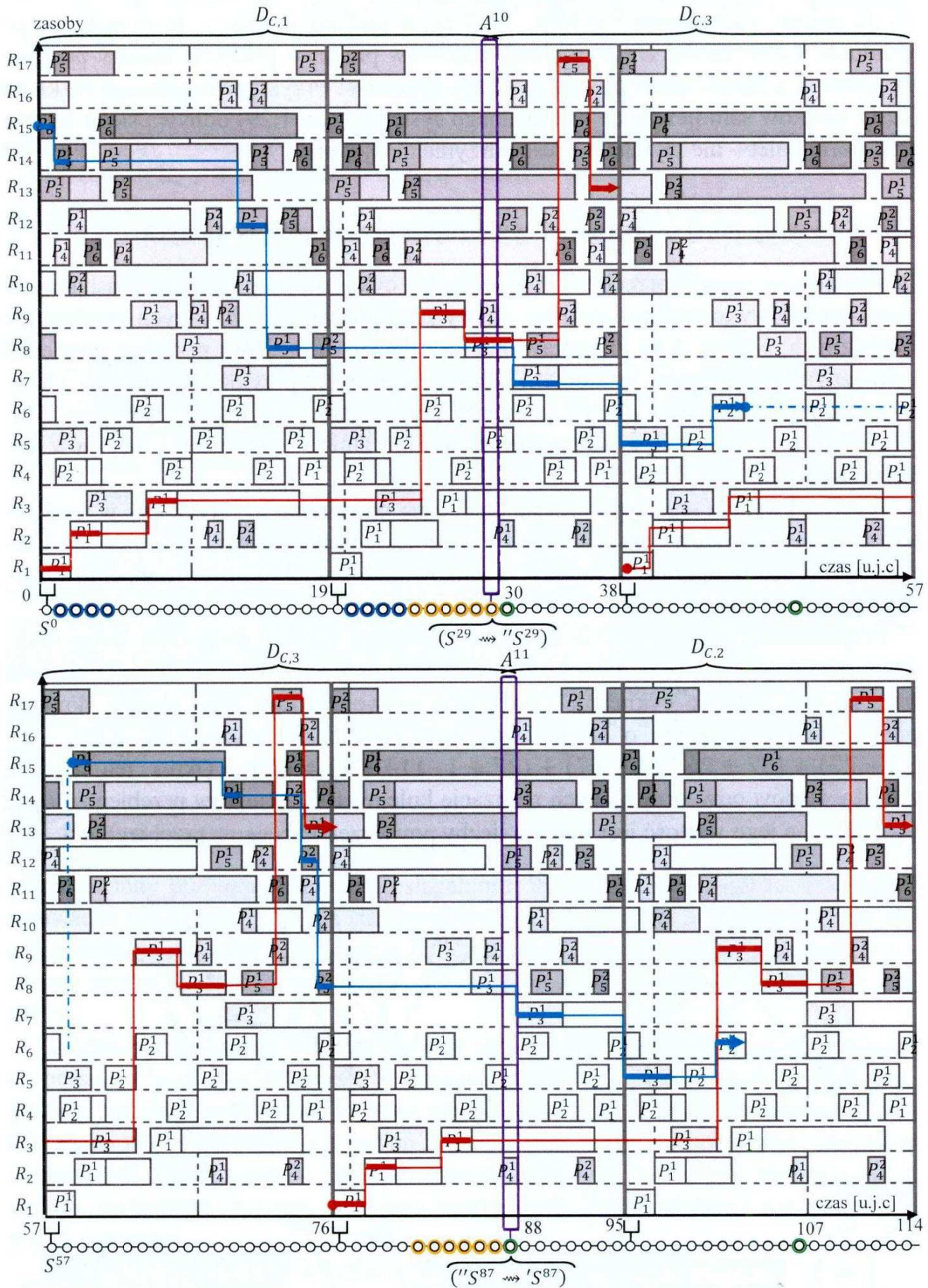
Na rys. 5.4 i rys. 5.5 kolorami żółtym, zielonym i niebieskim oznaczono obszary odpowiadające stanom posiadającym wspólne alokacje (zarówno procesów lokalnych i multimodalnych). Dla stanów tych, wykorzystując możliwości zmiany semaforów i wartości indeksów, można wyznaczyć sposób przejścia pomiędzy  $D_{C,1}$ ,  $D_{C,3}$  i  $D_{C,2}$ , np. wykorzystując zamianę semaforów i indeksów stanu  $S^{29}$  oraz " $S^{87}$ " (apostrofy: ', " , użyte przy oznaczeniach stanów zostały wprowadzone ze względu na rozróżnienie stanów związanych z przebiegiem  $D_{C,2}'$  i  $D_{C,3}''$ ). Odpowiednie przejście przyjmuje postać:

$$S^0 \rightarrow \dots \rightarrow S^{28} \rightarrow (S^{29} \rightsquigarrow ''S^{29}) \rightarrow \dots \rightarrow ''S^{87} \rightarrow (''S^{87} \rightsquigarrow 'S^{87}) \rightarrow 'S^{88} \rightarrow \dots \quad (124)$$

Przejście (124) zilustrowane zostało na rys. 5.6 i rys. 5.7. Jak widać na harmonogramie z rys. 5.7 przejście pomiędzy rozważanymi przebiegami instancji zachowań systemu nie wymaga zmiany alokacji realizowanych w nim procesów – zmiana semaforów i indeksów nie wstrzymuje realizacji procesów multimodalnych. ■



Rys. 5.6. Rzuty przestrzeni zachowań  $\mathcal{P}^1$  i  $\mathcal{P}^0$  systemu z rys. 1.30, wraz z przejściem  $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2}$



Rys. 5.7. Harmonogram ilustrujący przejście:  $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2}$



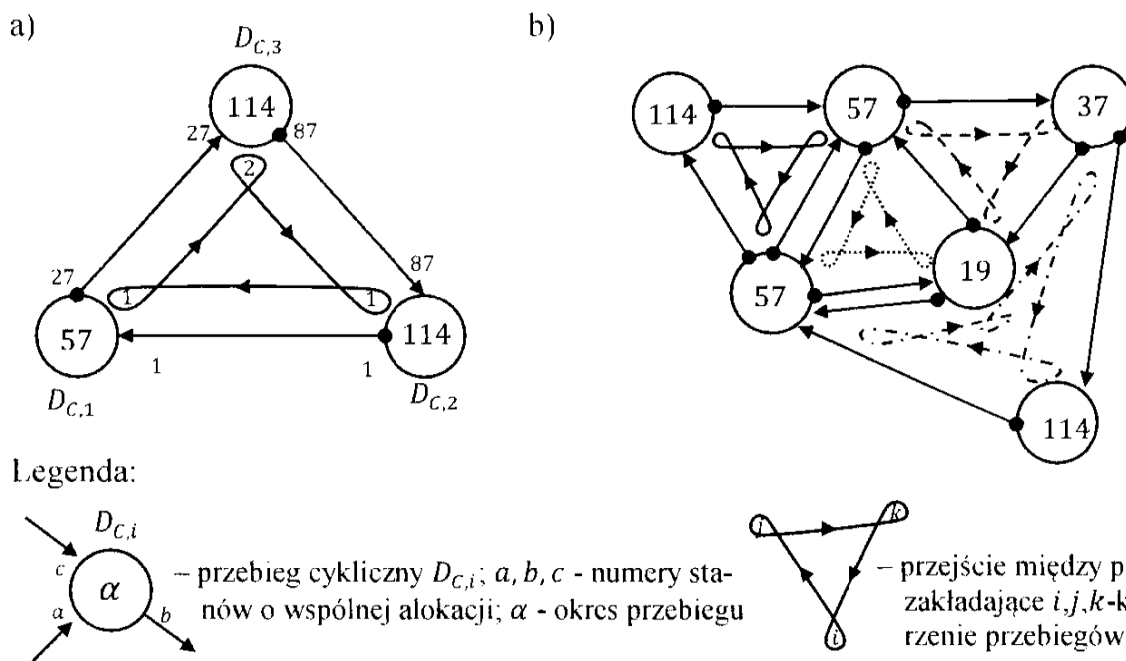
Przedstawiony przykład pokazuje jak można wykorzystać stany o wspólnej alokacji do zmiany zachowania SWMPC. Przy czym podkreślić należy, że sposób ten nie wymusza konieczności zmian alokacji procesów przez co przejścia między przebiegami mogą odbywać się w trakcie realizacji procesów. Przykładowo, zmiana rozkładu jazdy wózków samojezdnych rozważanego systemu (rys. 1.29) odbywa się w trakcie transportu palet – nie wymagane jest zatrzymanie systemu.

### 5.3. Sieć wzajemnej osiągalności cyklicznych przebiegów ustalonych

Przedstawiony sposób przechodzenia pomiędzy cyklicznymi przebiegami ustalonymi może zostać wykorzystany również do wyznaczania kolejnych, innych przebiegów cyklicznych. Na rys. 5.8a przedstawiono sieć opisującą możliwe przejścia pomiędzy cyklicznymi przebiegami ustalonymi przestrzeni  $\mathcal{P}$  (patrz przykład 8). Jednym z wielu możliwych wariantów przejść pomiędzy przebiegami, jest wariant tworzący cykl:  $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2} \rightarrow D_{C,1}$  (otrzymany w wyniku zmian semaforów i indeksów kolejno stanów  $S^{29}$ ,  $S^{87}$  i  $S^1$ ).

Jeżeli sekwencja tego typu przejść będzie się powtarzała to otrzymany w ten sposób przebieg wynikowy również będzie przebiegiem cyklicznym. Przebieg ten charakteryzuje harmonogram „sklejony” z całkowitych wielokrotności fragmentów harmonogramów przebiegów cyklicznych.

Przykładowo, jeżeli przejście między przebiegami zakłada dwukrotną, pełną realizację przebiegu  $D_{C,3}$ , przy jednokrotnej realizacji przebiegów pozostałych (fioletowa linia na rys. 5.8a, to cyklicznie powtarzana sekwencja tego typu przejść skutkuje przebiegiem cyklicznym o okresie (liczonym jako liczba stanów):  $\alpha_S = (56 + 1 \cdot 57 + 27) + (87 + 2 \cdot 114 + 87) + (27 + 1 \cdot 114 + 1) = 627$ . Okres ten wynika z liczby stanów odwzorowujących realizację kolejnych fragmentów przebiegów  $D_{C,1}$ ,  $D_{C,3}$  i  $D_{C,2}$ , a jego wartość jest funkcją liczby powtórzeń każdego z przebiegów.



Rys. 5.8. Sieć wzajemnych przejść między przebiegami cyklicznymi: wariant z przykładu 8 a), wariant ogólny b)

Oznacza to, że w strukturze przejeść cyklicznych przebiegów ustalonych z rys. 5.8a można utworzyć właściwie nieograniczoną liczbę przebiegów cyklicznych o okresach zadanych wyrażeniem:

$$\alpha_S = (q \cdot 57 + 27) + (87 + q'' \cdot 114 + 87) + (27 + q' \cdot 114), \quad (125)$$

gdzie:  $q, q', q''$  – liczba pełnych wykonań każdego z przebiegu w trakcie przejścia  $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2} \rightarrow D_{C,1}$ ,  $q, q', q'' \in \mathbb{N}$ .

Wyrażenie (125) można uogólnić do postaci:

$$\alpha_S = \sum_{D_{C,i} \in DS} ((q_i + 1) \cdot \alpha_i + b_i - a_i), \quad (126)$$

gdzie:

$\alpha_S$  – okres przebiegu, liczony jako liczba wchodzących w jego skład stanów,

$DS$  – zbiór przebiegów cyklicznych wykorzystanych w ramach przejścia,

$\alpha_i$  – okres  $i$ -tego przebiegu cyklicznego,

$q_i$  – liczba pełnych wykonań  $i$ -tego przebiegu cyklicznego,

$a_i$  – numer stanu, od którego rozpoczyna się  $i$ -ty przebieg cykliczny,

$b_i$  – numer stanu, w którym dochodzi do przejścia do następnego przebiegu.

Wyrażenia (125) i (126) pozwalają określić okres przebiegów uzyskanych tylko z jednego rodzaju przejścia (rys. 5.8a). Liczba przejść tego typu jest równa liczbie cykli występujących w strukturze przejeść cyklicznych przebiegów ustalonych. Liczba ta rośnie eksponentalnie wraz ze wzrostem rozmiaru struktury.

W ogólności oznacza to, że nawet niewielka struktura przejeść cyklicznych przebiegów ustalonych (jak np. na rys. 5.8b) może generować wiele nowych przebiegów cyklicznych.

#### 5.4. Podsumowanie

Przedstawiony problem wzajemnej osiągalności cyklicznych przebiegów ustalonych sprowadza się do poszukiwania stanów posiadających wspólną alokację. Problem ten charakteryzuje się niską złożonością obliczeniową warunkowaną znajomością przestrzeni stanów  $\mathcal{P}$  (oraz jej rzutów  $\mathcal{P}^1$ ,  $\mathcal{P}^0$  i  $\mathcal{A}$ ), a także liczbą występujących w niej wirów.

Założenie przyjmujące wspólną alokację procesów w stanach, w których dochodzi do przejścia pomiędzy zadanymi instancjami zachowań systemu (zmiany harmonogramów realizacji procesów), pozwala na realizację przejść między przebiegami cyklicznymi nie wymuszających zmian alokacji procesów. Ma to istotne znaczenie w systemach spotykanych w praktyce dnia codziennego, w których wymagana jest zmiana zachowania systemu bez jego wstrzymania i/lub realokacji występujących w nim procesów.

Znajomość stanów charakteryzujących się wspólną alokacją pozwala wyznaczyć sieć wzajemnych przejść cyklicznych przebiegów ustalonych. Koncepcja wyznaczania nowych przebiegów cyklicznych oparta o dopuszczalne w tego typu sieci przejścia, może stanowić alternatywę do opisanych w poprzednich rozdziałach metod.

## 6. Zakończenie

Omawiane w pracy problemy analizy, syntezy, składania oraz wzajemnej osiągalności zachowań cyklicznych, formułowane są w kategorii problemów decyzyjnych. Założenie takie wynika z diofantycznego charakteru modeli systemów klasy SWMPC, deterministycznych modeli całkowitoliczbowych. Poszukiwane w nich odpowiedzi dotyczą pytań związanych z istnieniem rozwiązań dopuszczalnych w szczególności zachowań cyklicznych (przebiegów i harmonogramów cyklicznych) lub parametrów struktury (czasów wykonania operacji, reguł priorytetowania, itp.) je gwarantujących. Do wybranych aspektów optymalizacji rozwiązań, np. nawiązujących do minimalizacji cyklu, czy też maksymalizacji ruchu na wybranych odcinkach sieci multimodalnej, nawiązują stosunkowo nieliczne wątki. Przykładem tego jest poszukiwanie przebiegu cyklicznego maksymalizującego prędkość transmisji danych w routerze z rys. 1.31 (patrz przykład 5).

Jak pokazano (rys. 3.12 i rys. 3.27), problem rozstrzygnięcia o istnieniu lub nie cyklicznych przebiegów ustalonych SWMPC, charakteryzuje się wykładniczą złożonością obliczeniową. Liczności uzyskiwanych zbiorów rozwiązań dopuszczalnych nie są zwykle zbyt duże (patrz liczba przebiegów cyklicznych w systemach gęstych), nie oznacza to jednak rezygnacji z poszukiwania algorytmów optymalizacji dedykowanych wybranym funkcjom celu.

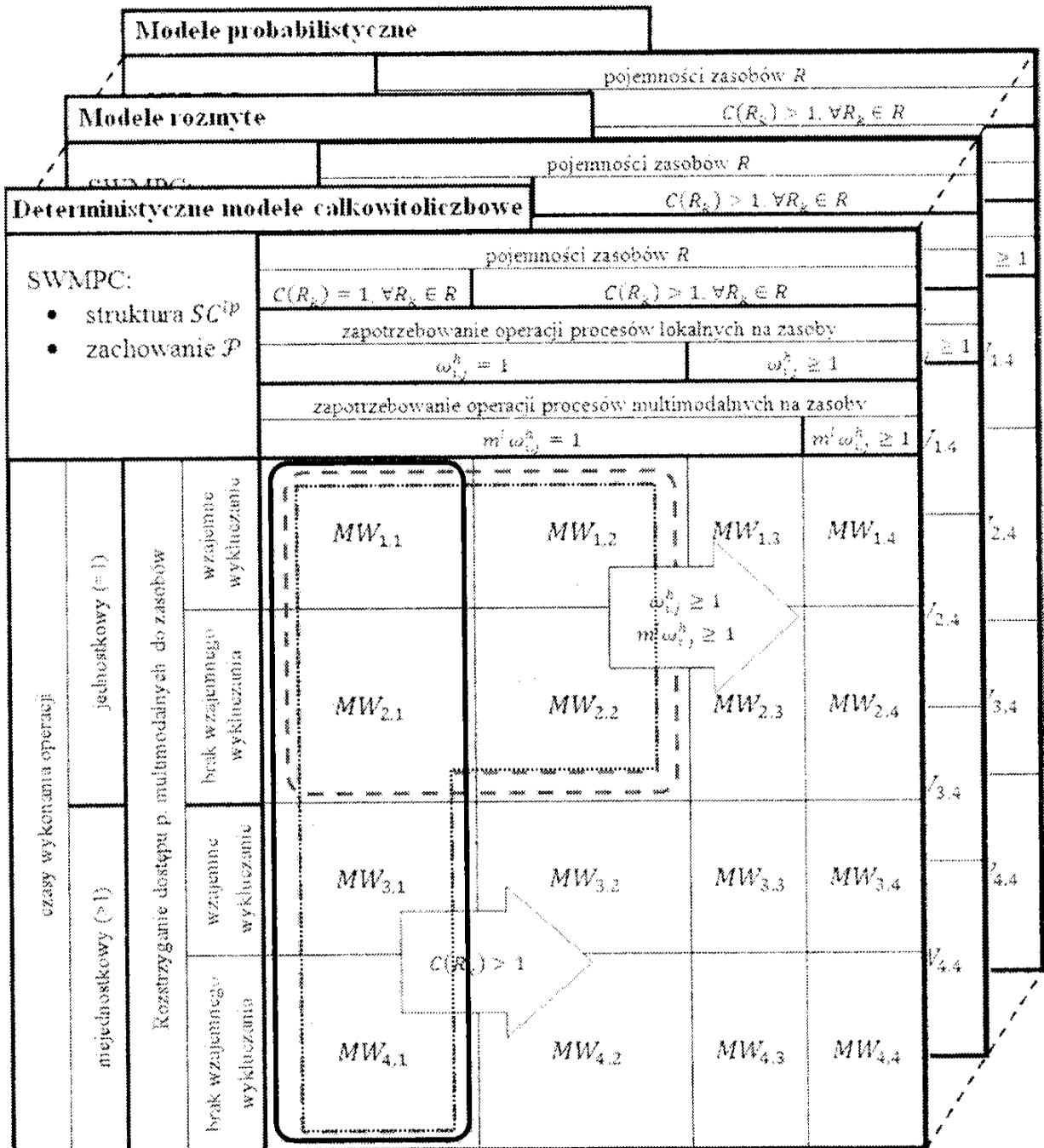
Reasumując, uzyskane wyniki obejmują metody rozwiązywania:

- problemu analizy SWMPC, a w szczególności metody:
  - redukcji stanów początkowych (podrozdział 3.1),
  - generowania harmonogramów cyklicznych (podrozdział 3.2),
- problemu syntezy SWMPC, a w szczególności metodę:
  - generowania harmonogramów cyklicznych (podrozdział 3.3),
- problemu składania struktur SWMPC, a w szczególności metody składania podstruktur:
  - wzdłuż łańcucha zasobów o zgodnym zwrocie marszrut procesów (podrozdział 4.1.1),
  - zgodne z regułą „zamka błyskawicznego” (podrozdział 4.1.2),
- problemu wzajemnej osiągalności cyklicznych przebiegów ustalonych SWMPC, a w szczególności metodę:
  - wyznaczania stanów posiadających wspólną alokację.

Pierwsze dwie metody (bazujące na wyprowadzonych w pracy Twierdzeniach i Własnościach): redukcji stanów początkowych i generowania harmonogramów cyklicznych, wykorzystywane są w pozostałych metodach. Metoda redukcji stanów początkowych ujmuje zachowania cykliczne w kontekście zdarzeniowym – zachowania cykliczne reprezentowane są w postaci przebiegów cyklicznych (definiowanych jako sekwencje kolejno osiągalnych stanów systemu). Z kolei, metoda generowania

harmonogramów cyklicznych ujmuje zachowania w kontekście czasowym – zachowania cykliczne reprezentowane są w postaci harmonogramów cyklicznych. W rozdziale 3 wykazano, że wymienione reprezentacje są sobie równoważne.

Opracowane metody dedykowane są dla różnych klas SWMPC, patrz rys. 6.1. Jak widać, ograniczają się głównie do systemów typu  $MW_{1,1}$ ,  $MW_{1,2}$ ,  $MW_{2,1}$ ,  $MW_{2,2}$ ,  $MW_{3,1}$ ,  $MW_{4,1}$  (pola oznaczone szarym kolorem na rys. 6.1), charakteryzujących się jednostkowym zapotrzebowaniem operacji na zasoby ( $\omega_{i,j}^h = 1$ ,  $m^l \omega_{i,j}^h = 1$ ).



- obszar wykorzystania metody generowania cyklicznych przebiegów ustalonych
- obszar wykorzystania metody redukcji stanów początkowych
- obszar wykorzystania metod rozwiązywania problemów składania i wzajemnej osiągalności cyklicznych przebiegów ustalonych

Rys. 6.1. Obszary wykorzystania i rozwoju opracowanych metod oceny zachowań cyklicznych SWMPC

W kontekście rozważanego modelu struktury  $SC^{lp}$  i przestrzeni stanów  $\mathcal{P}$ , przedstawione metody mogą być dalej rozwijane, m.in. w kierunku możliwości modelowania zasobów o nie jednostkowej pojemności ( $C(R_k) > 1$ ) i/lub operacji procesów o nie jednostkowym zapotrzebowaniu na zasoby ( $\omega_{i,j}^h \geq 1, m^l \omega_{i,j}^h \geq 1$ ). Umożliwi to ich wykorzystanie przy ocenie zachowań cyklicznych systemów transportowych z wielostanowiskowymi stacjami (tzn. zasobami pozwalającymi na jednoczesną realizację operacji wielu procesów), występującymi np. w systemach sieci kolei pasażerskich, sieci transmisji danych z buforami, itp.

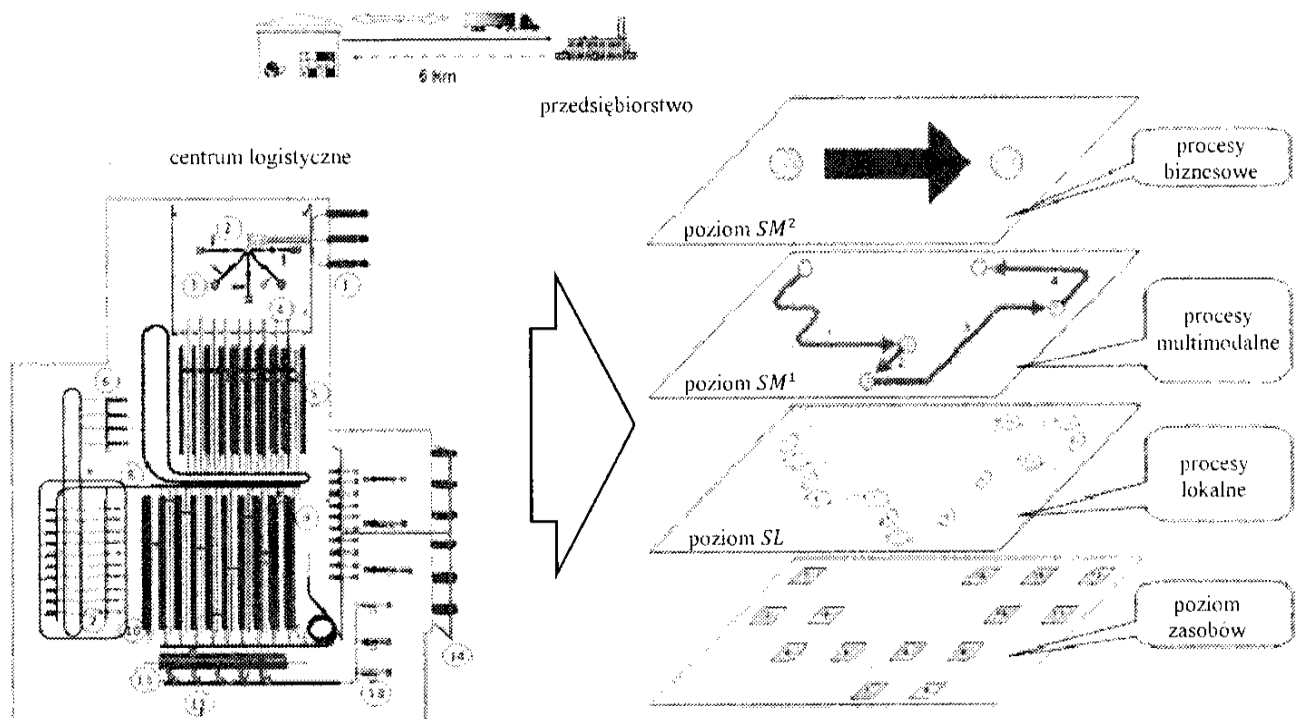
Warto zauważyć, że w przypadku metody redukcji stanów początkowych (wykorzystującej funkcję przejścia między stanami  $\delta^{lp}$  definiowaną w oparciu o zbiór predykatów (51)) rozszerzenie takie jest stosunkowo proste. Dostosowanie jej do nowych założeń (np. możliwości uwzględniania zasobów o wielo-jednostkowej pojemności itp.) wiąże się wyłącznie z odpowiednią zmianą (o ile jest ona możliwa) predykatów zbioru (51), a zatem poszukiwaniem odpowiednich warunków gwarantujących dopuszczalne (gwarantujące bezkolizyjne i bezblokadowe zachowanie systemu) przejścia między stanami. Złożoność obliczeniowa tak przedstawianej metody ma charakter wykładniczy (rys. 3.12) – badania efektywności opracowanych metod były prowadzone m.in. w ramach projektu TAPAS no 260026 (Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future). Duża złożoność obliczeniowa oznacza dalszą potrzebę poszukiwania wariantów proponowanej metody umożliwiających rozwiązywanie problemów o skali spotykanych w praktyce.

Prowadzone wyżej rozważania ograniczają się do deterministycznego modelu SWMPC (dopuszczającego wyłącznie precyzyjne wartości zmiennych decyzyjnych). Natura wielu systemów spotykanych w praktyce wymusza konieczności uwzględniania przybliżonego (nieprecyzyjnego, rozmytego) charakteru zmiennych. Wielkości takie jak np.: czas transportu towarów między stacjami, czas przejazdu autobusu przez centrum miasta, żądanie dostępu do danych na serwerze, czas ręcznej obróbki materiału, itp., są modelowane w postaci zmiennych losowych (charakteryzowanych przez odpowiednie rozkłady gęstości prawdopodobieństwa i dystrybuanty) lub też w postaci zmiennych rozmytych (charakteryzowanych przez odpowiednie funkcje przynależności). Możliwość analizy, syntezy, składania systemów dopuszczających nieprecyzyjny charakter zmiennych decyzyjnych, wyznacza zatem kolejny kierunek rozwoju proponowanych metod.

W przypadku modelowania rozmytego, deklaratywne ujęcie rozważanych problemów pozwala na wykorzystanie koncepcji Rozmytych Problemów Spełniania Ograniczeń (RPSO) przedstawionej m.in. w pracach (Bocewicz et al. 2009b, 2009c; Bach et al. 2010). Przyjęta tam koncepcja zakłada, że zarówno zmienne decyzyjne jak i wiążące je ograniczenia przyjmują postać rozmytą. W odróżnieniu od wielu istniejących podejść formułowania RPSO (Ruttkay 1994; Kowalczyk 1998; Moura-Pires 1998) pozwala ona na modelowanie ograniczeń rozmytych przyjmujących tak postaci zdań logicznych, jaki i wyrażeń algebraicznych. Dzięki temu możliwe jest uzyskiwanie rozwiązań w postaci liczb rozmytych (np. rozmytych momentów rozpoczęcia operacji) i tym samym determinowanych przez nie rozmytych harmonogramów cyklicznych.

W przypadku modelowania uwzględniającego losowy charakter zmiennych decyzyjnych, wykorzystane mogą być również techniki symulacji komputerowej. W literaturze co raz częściej pojawiają się tego typu opracowania. Przykładowo, w pracy (Pawlewski 2013) przedstawiona została koncepcja SWMPC, modelującego centrum logistyczne łańcucha dostaw. Prezentowany model bazuje na wykorzystaniu środowiska symulacyjnego FlexSim (rys. 6.2). W rozważanym systemie założono deterministyczną (z góry ustaloną) realizację procesów lokalnych (poziom  $SL$ ), reprezentujących operacje transportowe, rozładunku, przepakowywania i paletyzacji towarów oraz niedeterministyczną (losową) realizację procesów multimodalnych reprezentujących strumień towarów (poziom  $SM^1$ ). W systemie tym, zmienne losowe (charakteryzowane normalnym rozkładem gęstości prawdopodobieństwa) wykorzystywane są, tak do opisu odstępów czasowych między kolejnymi dostawami towarów, jak i rozmiaru dostawy (liczby palet). Na szczególną uwagę zasługuje fakt wykorzystania koncepcji procesów multimodalnych wyższego poziomu (poziom  $SM^2$ ) do reprezentacji procesów biznesowych.

Wielopoziomowy model zachowania systemu pozwala ocenić wzajemne oddziaływanie między poziomami zachowań procesów, tzn. oceniać wpływ realizacji procesów lokalnych na tempo realizacji procesów multimodalnych (oceniać przepustowość centrum logistycznego) i odwrotnie, oceniać wpływ wielkości charakteryzujących zmienne losowe procesów multimodalnych na realizację procesów lokalnych (wąskie gardła itp.). Analogiczne podejście można również wykorzystywać przy ocenie zachowań systemów transmisji danych.



**Rys. 6.2.** Centrum logistyczne łańcucha dostaw a), wielopoziomowy model struktury reprezentującego go SWMPC b) (Pawlewski 2013)

Techniki symulacji komputerowej umożliwiają badanie zachowań systemów klasy SWMPC o skali znacznie większej niż te proponowane przez metody omawiane w niniejszej pracy (rys. 3.12 i rys. 3.27). Wykorzystanie tych technik ogranicza się

jednak tylko do problemów analizy. Synteza SWMPC związana z przeglądem poszczególnych wariantów parametrów struktury, w tym przypadku, okazuje się nieskuteczna.

Jak pokazano, chociażby na przykładzie problemu wzajemnej osiągalności zachowań SWMPC, kluczową rolę przy ocenie potencjalnych przejść pomiędzy wybranymi, cyklicznymi przebiegami ustalonymi odgrywa znajomość tych zachowań reprezentowanych w postaci harmonogramów lub przebiegów cyklicznych (wchodzących w skład wirów przestrzeni  $\mathcal{P}$ ). Oznacza to, że metody zorientowane na „składanie” wybranych fragmentów cyklicznych przebiegów ustalonych nabierają istotnego znaczenia przy ocenie zachowań SWMPC, w szczególności w kontekście problemów analizy, syntezy i wzajemnej osiągalności zachowań. Możliwości związane z tym kierunkiem ilustrują przedstawione wcześniej metody (jak i związaną z nimi warunki) składania struktur SWMPC (w szczególności składania struktur regularnych) stanowiące podstawę do budowy algebry SWMPC.

Opracowanie modelu algebraicznego struktur SWMPC, umożliwiającego wykorzystanie operatorów odpowiadających intuicjom dodawania, usuwania i multiplikacji struktur, przedstawia nowy, alternatywny i perspektywistyczny kierunek badań. Nowe perspektywy obejmują m.in. rozszerzony wachlarz potencjalnych pytań jak np.:

- Jakie struktury SWMPC, jak komponowane (składane) prowadzą do struktur systemów cyklicznych?
- Czy multiplikacja struktury (rozumiana jako jej wielokrotne łączenie) gwarantuje uzyskanie cyklicznego zachowania systemu o zadanym okresie realizowanych procesów?
- Czy istnieje struktura, którą można dodać do istniejącej już struktury (złożyć z istniejącą strukturą) SWMPC, tak aby uzyskany w jej wyniku system był cykliczny?
- Jaki zbiór podstruktur należy dodać do danej struktury SWMPC, aby wynikowy system był systemem cyklicznym o zadanym okresie?
- Jaki zbiór podstruktur należy usunąć (tzn. usunąć procesy i związane z nimi zasoby) z danej struktury SWMPC, aby wynikowy system był systemem cyklicznym o zadanym okresie?
- Dodawanie/usuwanie jakiego zbioru podstruktur umożliwia przejście między zadanymi cyklicznymi przebiegami ustalonymi?
- ...

Szczególne perspektywy wykorzystania modelu algebraicznego struktur SWMPC wiążą się z ich implementacjami, obejmującymi procesy multimodalne, oraz warunkami umożliwiającymi szacowanie zachowań tych procesów w oparciu o algebraiczne właściwości zachowań lokalnych procesów cyklicznych.

## Literatura

- Abdallah IB, Elmaraghy A. *Deadlock prevention and avoidance in FMS: a Petri net based approach*. Int J Adv Manuf Technology. 1998, Vol. 14(10), s. 704–15.
- Aizenshtat VS. *Multi-operator cyclic processes*. Doklady of the Byelorussian Academy of Sciences. 1963, Vol. 7(4), s. 224–7.
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M. *A mixed linear program for a multi-part cyclic hoist scheduling problem*. Int J of Sciences and Techniques of Automatic control & computer engineering. 2008. s. 612–23.
- Araki T, Sugiyama Y, Kasami T. *Complexity of the deadlock avoidance problem*. 2nd IBM Symp Math Found Computer Sci. IBM. 1977, s. 229–57.
- Baccelli FL, Cohen G, Olsder GJ, Quadrat JP. *Synchronization and Linearity*. Wiley. 1992, s. 489.
- Bach I, Bocewicz G, Banaszak Z, Muszyński W. *Knowledge based and CP-driven approach applied to multi product small-size production flow*. Control and Cybernetics. 2010, Vol. 39(1), s. 69–95.
- Banaszak Z. *Modelling and control of FMS: Petri net approach*. Press, Wrocław Technical University. 1991.
- Banaszak Z, Bocewicz G. *Decision Support Driven Models and Algorithms of Artificial Intelligence*. Warsaw University of Technology. 2011, s. 237.
- Banaszak Z, Krogh B. *Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows*. IEEE Trans Robot Automat. 1990, Vol. 6, s. 724–34.
- Banaszak Z, Majdzik P, Wójcik R. *Procesy Współbieżne: Modele efektywności funkcjonowania*. Wydawnictwo Uczelniane Politechniki Koszalińskiej. 2008, s. 222.
- Ben-Ari M. *Principles of Concurrent and Distributed Programming*. Addison Wesley. 2005, s. 384.
- Bertalanffy L. *Ogólna teoria systemów. Podstawy, rozwój, zastosowania*. PWN. 1984, s. 339.
- Bielli M, Boulmakoul A, Mouncif H. *Object modeling and path computation for multimodal travel systems*. European Journal of Operational Research. 2006, Vol. 175(3), s. 1705–30.
- Błażewicz J. *Złożoność obliczeniowa problemów kombinatorycznych*. WNT. 1988, s. 194.
- Błażewicz J, Bovet DP, Brzeziński J, Gambosi G, Talamo M. *Optimal centralized algorithms for store-and-forward deadlock avoidance*. IEEE Trans On Computers. 1994, Vol. 43(11), s. 1333–8.
- Błażewicz J, Ecker K, Pesch E, Schmidt G, Węglarz J. *Handbook on Scheduling: From Theory to Applications*. Springer. 2007, s. 647.
- Bocewicz G, Bach I, Banaszak ZA. *Logic-algebraic method based and constraints programming driven approach to AGVs scheduling*. International Journal of Intelligent Information and Database Systems. 2009a, Vol. 3(1), s. 56–74.
- Bocewicz G, Bach I, Wójcik R. *Production flow prototyping subject to imprecise activity specification*. Kybernetes. 2009b, Vol. 38(7/8), s. 1298–316.
- Bocewicz G, Bach-Dąbrowska I, Banaszak Z. *Deklaratywne projektowanie systemów komputerowego wspomaganie planowania przedsięwzięć*. Akademicka Oficyna Wydawnicza Exit. 2009c, s. 302.
- Bocewicz G, Banaszak Z. *Abductive Inference Based Approach to DSS Designing for Project Portfolio Planning*. In: Nguyen NT, Szczerbicki E, (eds.). Intelligent System for Knowledge Management, series: Studies in Computational Intelligence, Vol. 252. Springer Berlin Heidelberg. 2009, s. 107–29.
- Bocewicz G, Banaszak Z. *Cyclic processes scheduling*. Applied Computer Science, Production engineering IT-driven concepts. 2010, Vol. 6(2), s. 41–70.



- Bocewicz G, Banaszak Z. *Declarative modeling of multimodal cyclic processes*. In: Golińska P, Fertsch M, Marx-Gomez J, (eds.). Information Technologies in Environmental Engineering, series: Environmental Science and Engineering, Vol. 3. Springer Berlin Heidelberg. 2011, s. 551–68.
- Bocewicz G, Banaszak Z. *Declarative approach to cyclic scheduling of multimodal processes*. In: Golinska P, editor. EcoProduction and Logistics, Emerging Trends and Business Practices, series: EcoProduction, Vol. 3. Springer Berlin Heidelberg. 2013a, s. 203–35.
- Bocewicz G, Banaszak Z. *Declarative approach to cyclic steady state space refinement: periodic process scheduling*. The International Journal of Advanced Manufacturing Technology. 2013 b, Vol. 67(1-4), s. 137–55.
- Bocewicz G, Banaszak Z. *Scheduling for Multimodal Cyclic Transport Systems*. Pomiary Automatyka, Robotyka. 2013c. Vol. 2, s. 106–13.
- Bocewicz G, Banaszak Z, Nielsen I. *A declarative approach to cyclic processes coupling and scheduling*. Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation. 2012a, s. 1–8.
- Bocewicz G, Banaszak Z, Nielsen I, Dang VQ. *Multimodal processes rescheduling*. Proceedings of the International Conference on Advances in Production Management Systems, APMS 2012. 2012b.
- Bocewicz G, Bzdyra K. *Graficzna metoda planowania zajęć*. Materiały konferencyjne VI międzynarodowej konferencji-naukowo technicznej, Technologiczne Systemy Informacyjne w Inżynierii Produkcji i Kształceniu Technicznym. 2005, s. 27–34.
- Bocewicz G, Muszyński W, Banaszak Z. *CP-driven multimodal processes scheduling*. Proceedings of the 16th IEEE International Conference on Emerging Technologies and Factory Automation. 2011a.
- Bocewicz G, Nielsen P, Banaszak Z, Dang VQ. *Cyclic steady state refinement: multimodal processes perspective*. Proceedings of the International Conference on Advances in Production Management Systems. 2011b.
- Bocewicz G, Wójcik R, Banaszak Z. *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach*. In: Nguyen NT, Grzech A, Howlett R, Jain L, (eds.). Agent and Multi-Agent Systems: Technologies and Applications, series: Lecture Notes in Computer Science, Vol. 4496. Springer Berlin Heidelberg. 2007, s. 578–87.
- Bocewicz G, Wójcik R, Banaszak Z. *Planowanie pracy zespołu robotów wielofunkcyjnych w systemach potokowej produkcji wieloasortymentowej (Interakcyjne wspomaganie decyzji)*. Problemy Robotyki, Prace naukowe, Elektronika. Oficyna Wydawnicza Politechniki Warszawskiej. 2008, s. 647–60.
- Bocewicz G, Wójcik R, Banaszak Z. *On undecidability of cyclic scheduling problems*. In: Karagiannis D, Jin Z, (eds.). Knowledge Science, Engineering and Management, series: Lecture Notes in Computer Science Vol. 5914. Springer Berlin Heidelberg. 2009d, s. 310–21.
- Bocewicz G, Wójcik R, Banaszak Z. *Cyclic Steady State Refinement*. In: Abraham A, Corchado JM, Gonzales SR, De Paz Santana JF, (eds.). International Symposium on Distributed Computing and Artificial Intelligence, series: Advances in Intelligent and Soft Computing, Vol. 91. Springer Berlin Heidelberg. 2011c, s. 191–8.
- Bocewicz G, Wójcik R, Banaszak Z. *Toward cycling scheduling of concurrent multimodal processes*. In: Jędrzejowicz P, Nguyen NT, Hoang K, (eds.). Computational Collective Intelligence: Technologies and Applications, series: Lecture Notes in Artificial Intelligence, Vol. 6922. Springer Berlin Heidelberg. 2011d, s. 448–57.
- Bocewicz G, Wójcik R, Banaszak Z. *Cyclic Scheduling of Multimodal Concurrently Flowing Processes*. In: Świątek J, Grzech A, Świątek P, Tomczak JM, (eds.). Advances in Systems

- Science, series: Advances in Intelligent Systems and Computing, Vol. 240. Springer Berlin Heidelberg. 2014, s. 587–98.
- Bocewicz G, Wójcik R, Banaszak Z. *A Reachability of Cyclic Steady States Space: Declarative Modeling Approach*. In: Nguyen N-T, Hoang K, Jędrzejowicz P, (eds.). Computational Collective Intelligence Technologies and Applications, series: Lecture Notes in Artificial Intelligence, Vol. 7654. Springer. 2012c, s. 233–43.
- Brucker P, Dhaenens-Flipo C, Knust S, Kravchenko SA, Werner F. *Complexity results for parallel machine problems with a single server*. Journal of Scheduling. 2002, Vol. 5, s. 429–57.
- Brucker P, Kampmeyer T. *Tabu search algorithms for cyclic machine scheduling problems*. Journal of Scheduling. 2005, Vol. 8(4), s. 303–22.
- Brzeziński J. *Ocena stanu globalnego w systemach rozproszonych*. OWN. 2001, s. 152.
- Bubnicki Z. *Learning processes and logic-algebraic method for the systems with knowledge representation*. Systems analysis and management. PAS. 1999.
- Castro PM, Erdirik-Dogan M, Grossmann IE. *Simultaneous batching and scheduling of single stage batch plants with parallel units*. AIChE Journal. 2008, Vol. 54(1), s. 183–93.
- Cellary W. *Rozdział zasobów w systemach komputerowych – próba podejścia globalnego*. Seria Rozprawy 127. Poznań: Wydawnictwo Politechniki Poznańskiej. 1981.
- Cellary W. *Admission test for deadlock avoidance in single-resource systems*. Computer Performance. 1982, Vol. 3(4), s. 240–6.
- Che A, Chu C. *A polynomial algorithm for no-wait cyclic hoist scheduling in an extended electroplating line*. Operations Research Letters. 2005, Vol. 33(3), s. 274–84.
- Che A, Chu C, Levner E. *A polynomial algorithm for 2-degree cyclic robot scheduling*. European Journal of Operational Research. 2003, Vol. 145(1), s. 31–44.
- Chen H, Chu C, Proth JM. *Cyclic scheduling of a hoist with time window constraints*. IEEE Transactions on Robotics and Automation. 1998, Vol. 14(1), s. 144–52.
- Cho H, Kumaran TK, Wysk RA. *Graph-theoretic deadlock detection and resolution for flexible manufacturing systems*. IEEE Transactions on Robotics and Automation. 1995, Vol. 11(3), s. 413 – 421.
- Chrétienne P. *The basic cyclic scheduling problem with deadlines*. Discrete Applied Mathematics. 1991, Vol. 30(2-3), s. 109–23.
- Chu F, Xie XL. *Deadlock analysis of Petri nets using siphons and mathematical programming*. IEEE Transactions on Robotics and Automation. 1997, Vol. 13(6), s. 793–804.
- Coffman EGJ, Denning PJ. *Operating systems theory*. Prentice-Hall. 1973, s. 331.
- Coffman EGJ, Elphick MJ, Shoshani A. *System deadlocks*. ACM Comput Surveys. 1971, Vol. 3(2), s. 67–78.
- Cohen G, Dubois D, Quadrat JP, Viot M. *A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing*. IEEE Transactions on Automatic Control. 1985, Vol. 30(1), s. 210–20.
- Commoner FG. *Deadlocks in Petri nets*. Applied Data Research Inc., Wakeeld, Massachusetts 01880, Report Nr CA-7206-2311. 1972.
- Cuninghame-Greene RA. *Process synchronization in a steelworks - a problem of feasibility*. Proceedings of the 2nd International Conference on Operational Research. 1960.
- Cuninghame-Greene RA. *Describing industrial processes with interference and approximation their steady-state behaviour*. Operational Research Quarterly. 1962, Vol. 13(1), s. 95–100.
- Cuninghame-Greene RA. *Minimax Algebra*. Springer-Verlag. 1979, s. 136.
- Dang VQ, Nielsen I, Bocewicz G. *A Genetic Algorithm-Based Heuristic for Part-Feeding Mobile Robot Scheduling Problem*. In: Corchado Rodriguez JM, Perez J, Golińska P, Giroux S, Corchuleo R, (eds.). Trends in Practical Applications of Agents and Multiagent

- Systems, series: *Advances in Intelligent and Soft Computing*, Vol. 157. Berlin, Heidelberg: Springer. 2012, s. 85–92.
- Dasdan A, Gupta RK, Irani SS. *An Experimental Study of Minimum Mean Cycle Algorithms*. UCI-ICS Technical Report, #98-32. 1998.
- Dawande MN, Geismar HN, Sethi SP, Sriskandarajah C. *Throughput optimization in robotic cells*. International Series in Operations Research & Management Science, Vol. 101, Springer. 2007, s. 417.
- Degtyarev YI, Timkovsky VG. *On a model of optimal planning systems of flow type*. Automation Control Systems. 1976, Vol. 1, s. 69–77.
- Deitel HM, Deitel PJ, Choffnes DR. *Operating Systems*. Prentice Hall. 2003, s. 1209.
- Dijkstra EW. *Cooperating sequential processes*. New York: Academic Press. 1965.
- Ezpeleta J, Colom JM, Martínez J. *A Petri net based deadlock prevention policy for flexible manufacturing systems*. IEEE Transactions on Robotics and Automation. 1995, Vol. 11(2), s. 173–84.
- Ezpeleta J, Recalde L. *Deadlock avoidance approach for non-sequential resource allocation systems*. IEEE Transactions on Systems, Man and Cybernetics Part-A: Systems and Humans. 2004, Vol. 34(1), s. 93–101.
- Fanti MP. *Deadlock resolution strategy for automated manufacturing systems including conjunctive resource service*. IEEE Transactions on Systems, Man and Cybernetics Part-A: Systems and Humans. 2004, Vol. 34(1), s. 80–92.
- Fanti MP, Maione B, Mascolo S, Turchiano B. *Performance of deadlock avoidance algorithms in flexible manufacturing systems*. J Manufact Syst. 1996 a, Vol. 15(3), s. 164–78.
- Fanti MP, Maione B, Mascolo S, Turchiano B. *Low-cost deadlock avoidance policies for flexible production systems*. Int J Model Simul. 1997, Vol. 17(4), s. 310–6.
- Fanti MP, Maione B, Turchiano B. *Deadlock detection and recovery in flexible production systems with multiple capacity resources*. Proceedings of 8th Mediterranean Electrotechnical. 1996b, s. 237–41.
- Fanti MP, Maione B, Turchiano B. *Comparing digraph and Petri net approaches to deadlock avoidance in FMS*. IEEE Transactions on Systems, Man and Cybernetics, Part B. 2000, Vol. 30(5), s. 783–98.
- Fanti MP, Zhou M. *Deadlock control methods in automated manufacturing systems*. IEEE Transactions on Systems, Man and Cybernetics, Part A. 2004, Vol. 34(1), s. 5–22.
- Fokkink W. *Introduction to process algebra*. Springer-Verlag. 1999, s. 174.
- Foo HM, Leong HW, Lao Y, Lau HC. *A Multi-Criteria, Multi-Modal Passenger Route Advisory System*. Proceedings of the IES-CTR. 1999.
- Friedrich M. *A multi-modal transport model for integrated planning*. Proceedings of 8th World Conference on Transport Research. 1999, s. 1–14.
- Gebraeel N, Lawley M. *Deadlock detection, prevention, and avoidance for automated tool sharing systems in flexible manufacturing*. IEEE Transactions on Robotics and Automation. 2001, Vol. 17(3), s. 342–56.
- Gerla M, Kleinrock L. *Flow control, a comparative study*. IEEE Trans Comm. 1980, Vol. 28(4), s. 553–74.
- Graves SC, Meal HC, Stefek D, Zeghmi AH. *Scheduling of re-entrant flow shops*. Journal of Operations Management. 1983, Vol. 3, s. 197–207.
- Guo Z. *Transfers and path choice in urban public transport systems*. Massachusetts Institute of Technology. 2008, s. 295.
- Gürel A, Bogdan S, Lewis FL. *Matrix approach to deadlock-free dispatching in multi-class finite buffer flowlines*. IEEE Transactions on Automatic Control. 2000, Vol. 45(11), s. 2086–90.

- Habermann AN. *Prevention of system deadlocks*. Communication of the ACM. 1969, Vol. 12(7), s. 373–7.
- Hall NG, Kamoun H, Sriskandarajah C. *Scheduling in robotic cells: Complexity and steady state analysis*. European Journal of Operational Research. 1998, Vol. 109(1), s. 43–65.
- Hall NG, Lee TE, Posner ME. *The complexity of cyclic shop scheduling problems*. Journal of Scheduling. 2002, Vol. 5(4), s. 307–27.
- Hall NG, Potts CN, Sriskandarajah C. *Parallel machine scheduling with a common server*. Discrete Applied Mathematics. 2000, Vol. 102(3), s. 223–43.
- Hanan C. *Study of a NP-hard cyclic scheduling problem: The recurrent jobshop*. European Journal of Operational Research. 1994, Vol. 72(1), s. 82–101.
- Hanan C, Munier A. *Cyclic scheduling on parallel processors: on overview*. In: Chretienne P, Coffman EG, Lenstra JK, Liu Z, (eds.). Scheduling Theory and Its Applications. John Wiley and Sons. 1995, s. 194–226.
- Havender JW. *Avoiding deadlock in multitasking systems*. IBM Systems Journal. 1968, Vol. 7(2), s. 74–84.
- Hilbert D. *Mathematische Probleme*. Internationalen Mathematiker-Kongreß. Paris. 1900.
- Hindi K, Fleszar K. *A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem*. Computers & Industrial Engineering. 2004, Vol. 47(1), s. 91–101.
- Hopcroft JE, Motwani R, Ullman JD. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley. 2006, s. 750.
- Horn MET. *Multi-modal and demand-responsive passenger transport systems: a modelling framework with embedded control systems*. Transportation Research Part A: Policy and Practice. 2002, Vol. 36(2), s. 167–88.
- Hsieh FS. *Fault-tolerant deadlock avoidance algorithm for assembly processes*. IEEE Transactions on Systems, Man and Cybernetics Part-A: Systems and Humans. 2004, Vol. 34(1), s. 65–78.
- Huang Y, Jeng M, Xie X, Chung S. *Deadlock prevention policy based on Petri nets and siphons*. Int J Of Production Research. 2001, Vol. 39(2), s. 283–305.
- Ioachim I, Soumism F. *Schedule efficiency in a robotic production cell*. International Journal of Flexible Manufacturing Systems. 1995, Vol. 7, s. 5–26.
- Iszkowski W, Maniecki M. *Programowanie współbieżne*. WNT. 1982, s. 286.
- Jeng MD, DiCesare F. *A review of synthesis techniques for Petri nets with applications to automated manufacturing systems*. IEEE Transactions on Systems Man and Cybernetics. 1993, Vol. 23(1), s. 301–12.
- Jones ND, Landweber LH, Lien YE. *Complexity of some problems in Petri nets*. Theoretical Computer Science. 1977, Vol. 4, s. 277–99.
- Kampmeyer T. *Cyclic scheduling problems, rozprawa doktorska*. University of Osnabrück. 2006.
- Kats V, Lei L, Levner E. *Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence and time-window constraints*. European Journal of Operational Research. 2008, Vol. 187(3), s. 1196–211.
- Kats V, Levner E. *A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling*. Operations Research Letters. 1997, Vol. 21(4), s. 171–9.
- Kats V, Levner E. *Cyclic scheduling of operations for a part type in an FMS handled by a single robot: A parametric critical-path approach*. International Journal of Flexible Manufacturing Systems. 1998, Vol. 10(2), s. 129–38.
- Kats V, Levner E, Meyzin L. *Multiple-part cyclic hoist scheduling using a sieve method*. IEEE Transactions on Robotics and Automation. 1999, Vol. 15(4), s. 704–13.

- Kise H. *On an automated two-machine flowshop scheduling problem with infinite buffer*. Journal of the Operations Research Society of Japan. 1991, Vol. 34(3), s. 354–61.
- Kowalczyk R. *On solving fuzzy constraints satisfaction problems with genetic algorithms*. Evolutionary Computation Proceedings IEEE World Congress on Computational Intelligence. 1998. s. 758–62.
- Krishnamurthy EV. *Parallel processing, principles and practice*. Addison-Wesley. 1989, s. 343.
- Kubale M, Nadolski A. *Chromatic scheduling in a cyclic open shop*. European Journal of Operational Research. 2005, Vol. 164(3), s. 585–91.
- Lautenbach K, Thiagarajan PS. *Analysis of a resource allocation problem using Petri nets*. In: Syre JC, editor. Proceedings 1st European Conference on Parallel and Distributed Processing. Toulouse. 1979, s. 260–6.
- Lawley M, Reveliotis SA, Ferreira P. *Structural control of flexible manufacturing systems and the neighborhood policy: Part 2 generalization, optimization, and efficiency*. IIE Transactions. 1997, Vol. 29(10), s. 888–99.
- Lawley M, Reveliotis SA, Ferreira P. *A correct and scalable deadlock avoidance policy for flexible manufacturing systems*. IEEE Transactions on Robotics and Automation. 1998, Vol. 14(5), s. 796–809.
- Lawley M, Sulistyono W. *Robust supervisory control policies for manufacturing systems with unreliable resources*. IEEE Transactions on Robotics and Automation. 2002, Vol. 18(3), s. 346–59.
- Lee TE. *Stable earliest starting schedules for cyclic job shops: A linear system approach*. The International Journal of Flexible Manufacturing Systems. 2000, Vol. 12(1), s. 59–80.
- Lee TE, Posner ME. *Performance measures and schedules in periodic job shops*. Operations Research. 1997, Vol. 45(1), s. 72–91.
- Legierski W, Parys P. *Planowanie zajęć metodami programowania z ograniczeniami*. Automatyizacja Procesów Dyskretnych, tom: Optymalizacja Dyskretna Robotyka i sterowniki programowalne. WNT. 2004, s. 125 – 132.
- Lei L. *Determining the optimal starting times in a cyclic schedule with a given route*. Computers & Operations Research. 1993, Vol. 20(8), s. 807–16.
- Lei L, Liu Q. *Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints*. INFOR Journal. 2001, Vol. 39(2), s. 185–99.
- Lei L, Wang TJ. *A proof: The cyclic scheduling problem is NP-complete*. Working paper no 89-0016, Rutgers University. 1989.
- Levner E, Kats V, Alcaide López de Pablo D. *Cyclic Scheduling in Robotic Cells: An Extension of Basic Models in Machine Scheduling Theory*. In: Levner E, editor. Multiprocessor Scheduling: Theory and Applications. Itech Education and Publishing. 2007, s. 1–20.
- Levner E, Kats V, Alcaide López de Pablo D, Cheng TCE. *Complexity of cyclic scheduling problems: A state-of-the-art survey*. Computers & Industrial Engineering. 2010, Vol. 59, s. 352–61.
- Levner E, Kats V, Levit V. *An improved algorithm for cyclic flowshop scheduling in a robotic cell*. European Journal of Operational Research. 1997, Vol. 197, s. 500–8.
- Livshits EM, Mikhailetsky ZN, Chervyakov EV. *A scheduling problem in an automated flow line with an automated operator*. Computational Mathematics and Computerized Systems. 1974, Vol. 5, s. 151–5.
- Mahadev NVR, Solot P, Werra D. *The cyclic compact open-shop scheduling problem*. Discrete Mathematics. 1993, Vol. 111(1-3), s. 361 – 366.
- Majdzik P, Banaszak Z. *Methods and Models in Automation and Robotics*. Proceedings of the 12th IEEE International Conference Methods and Models In Automations and Robotics MMAR. 2006, s. 1139–44.

- Matijasiewicz JW. *Diofantowost' pierieczislmych mnozestw*. Dokl AkadNauk SSSR. 1970, Vol. 191, s. 278–82.
- Matsuo H, Shang JS, Sullivan RS. *A crane scheduling problem in a computer-integrated manufacturing environment*. Management Science. 1991, Vol. 17, s. 587–606.
- Mayer E, Haus UU, Raish J, Weismantel R. *Throughput-optimal sequences for cyclically operated plants*. Discrete Event Dynamical Systems. 2008, Vol. 18(3), s. 355–83.
- Mądry M. *Planowanie obsługi sieci dystrybucji odpadów poprodukcyjnych z wykorzystaniem technik programowania ograniczeń*, rozprawa doktorska. Politechnika Poznańska. 2006.
- Merlin PM, Schweitzer PJ. *Deadlock avoidance in store-and-forward networks - I: Store-and-forward deadlock*. IEEE Transactions on Communications. 1980, Vol. 28(3), s. 345–56.
- Milner R. *Communication and concurrency*. Prentice Hall. 1995. s. 300.
- Modarres M, Ghandehari M. *Generalized Cyclic Open Shop Scheduling and a Hybrid Algorithm*. Journal of Industrial and Systems Engineering. 2008, Vol. 1(4), s. 345–59.
- Moura-Pires J. *Starting points in optimization algorithms for solving discrimin and leximin Fuzzy - CSP*. Fuzzy Systems Proceedings IEEE World Congress on Computational Intelligence. 1998, s. 857–62.
- Munier A. *The basic cyclic scheduling problem with linear precedence constraints*. Discrete Applied Mathematics. 1996, Vol. 64(3), s. 219–38.
- Narahari Y, Viswanadham N. *Petri net approach to the modelling and analysis of flexible manufacturing systems*. Annals of Operations Research. 1985, Vol. 3(8), s. 449–72.
- Nęcka E, Orzechowski J, Szymura B. *Psychologia poznawcza*. Warszawa: PWN I ACADEMICA. 2012, s. 720.
- Papadimitrou CH. *Computational Complexity*. Addison Wesley Longman. 1994, s. 540.
- Park J, Reveliotis SA. *Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems*. IEEE Transactions on Robotics and Automation. 2000, Vol. 16(2), s. 190–5.
- Park J, Reveliotis SA. *Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings*. IEEE Transactions On Automatic Control. 2001, Vol. 46(10), s. 1572–83.
- Park J, Reveliotis SA. *Liveness enforcing supervision for resource allocation systems with forbidden states and uncontrollable events*. IEEE Transactions on Robotics and Automation. 2002, Vol. 18(2), s. 234–9.
- Pawlewski P. *Multimodal Approach to Model and Design Supply Chain (I)*. Manufacturing Modelling, Management, and Control, Proceedings of the IFAC Conference on Manufacturing Modeling Management and Control. 2013, s. 2110–5.
- Phillips LW, Unger PS. *Mathematical programming solution of a hoist scheduling program*. AIIE Transactions. 1976, Vol. 8, s. 219–25.
- Pinedo M. *Scheduling: theory, algorithms, and systems*. Prentice Hall. 1995, s. 693.
- Pinto T, Barbosa-Povoa APFD, Novais AQ. *Optimal design and retrofit of batch plants with a periodic mode of operation*. Computers and Chemical Engineering. 2005, Vol. 29, s. 1293–303.
- Polak M. *Algorytmy automatycznej syntezy procedur sterowania rozproszonego w systemach współbieżnych procesów cyklicznych*, rozprawa doktorska. Politechnika Poznańska. 2004.
- Polak M, Majdzik P, Banaszak Z, Adamski M. *Prototyping of distributed control procedures in concurrent cyclic processes systems*. Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation. 2003, s. 261–8.
- Polak M, Majdzik P, Banaszak Z, Wójcik R. *The Performance Evaluation Tool For Automated Prototyping of Concurrent Cyclic Processes*. Fundamenta Informaticae. 2004, Vol. 60(1-4), s. 269–89.

- Post E. *A variant of a recursively unsolvable problem*. Bulletin of the AMS. 1946, Vol. 52, s. 264–8.
- Raynal M. *Algorithms for mutual exclusion*. MIT Press, editor. 1986, s. 107.
- Raynal M. *Distributed algorithms and protocols*. John Wiley & Sons. 1988, s. 176.
- Reisig W. *Petri nets: An introduction*. Springer. 1985, s. 180.
- Reveliotis SA. *Real-time management of resource allocation systems: a discrete-event systems approach*. New York: Springer. 2005, s. 246.
- Reveliotis SA, Ferreira PM. *Deadlock avoidance policies for automated manufacturing cells*. IEEE Transactions on Robotics and Automation. 1996, Vol. 12(6), s. 845–57.
- Reveliotis SA, Lawley MA, Ferreira PM. *Polynomial complexity deadlock avoidance policies for sequential resource allocation systems*. IEEE Transactions On Automatic Control. 1997, Vol. 42(10), s. 1344–57.
- Romanovskii IV. *Optimization of stationary control of a discrete deterministic process*. Kybernetika (Cybernetics). 1967, Vol. 3(2), s. 66–78.
- Ross AK, Wright CRB. *Matematyka Dyskretna*. PWN. 2008, s. 899.
- Roszkowska E. *Deadlock avoidance in concurrent compound pipeline processes*. Archiwum informatyki teoretycznej i stosowanej. 1990, Vol. 2, s. 227–42.
- Roszkowska E. *Zastosowanie sieci Petriego do modelowania i oceny efektywności funkcjonowania ESP*, praca doktorska. Politechnika Wrocławska. 1991.
- Roszkowska E. *Preventing second level and avoiding first level deadlocks in FMS*. Proceedings ICRA, IEEE International Conference on Robotics and Automation, Vol. 3. 2000, s. 2859–64.
- Roszkowska E. *Supervisory control for deadlock avoidance in compound processes*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 2004, Vol. 34(1), s. 52–64.
- Roszkowska E, Banaszak Z. *Problems of deadlock handling in pipeline processes*. Computer and Information Sciences VI. 1991, Vol. 2, s. 1185–94.
- Roszkowska E, Jentink J. *Minimal restrictive deadlock avoidance in fmss*. Proceedings of European Control Conference, Vol. 2. 1993, s. 539–43.
- Roszkowska E, Wójcik R. *Problems of process flow feasibility in FAS*. In: Leiviska K, editor. Proceedings of IFAC Workshop on CIM in Process and Manufacturing Industries. Espoo, Finland: Pergamon Press Ltd.. 1992, s. 115–20.
- Roundy RO. *Cyclic schedules for job-shops with identical jobs*. Mathematics of Operations Research. 1992, Vol. 17, s. 842–65.
- Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*, Third Edition. Prentice Hall. 2009, s. 1152.
- Ruttikay Z. *Fuzzy Constraint Satisfaction*. In Proceedings 3rd IEEE International Conference on Fuzzy Systems. 1994.
- Seiche W, Abel D, Rake H. *Synthesis of deadlock-free control structures using Petri nets*. In: Franke D, Kraus F, (eds.). Preprints Of the FIRST IFAC Symposium on “Design Methods of Control Systems”, Vol. 2. Pergamon Press Ltd.. 1991. s. 697–702.
- Silberschatz A, Galvin PB, Gagne G. *Operating systems concepts*. Amsterdam: John Wiley & Sons. 2004. s. 921.
- Sitek P. *Application of constraint logic programming to decision support for the supply chain management*. Pomiary Automatyka Robotyka. 2013, Vol. 2, s. 129–34.
- Sitek P, Wikarek J. *A Hybrid Approach to Modeling and Optimization for Supply Chain Management with multimodal transport*. Proceedings of the 18th International Conference on Methods and Models in Automation and Robotics. 2013a, s. (w druku).

- Sitek P, Wikarek J. *A hybrid method for modeling and solving constrained search problems*. Proceedings of Federated Conference on Computer Science and Information Systems. 2013b, s. (w druku).
- Smutnicki C. *Wytwarzanie cykliczne*. In: Bożejko W, Pempera J, (eds.). *Optymalizacja dyskretna w informatyce, automatyce i robotyce*. Oficyna Wydawnicza Politechniki Wrocławskiej. 2012, s. 147–65.
- Smutnicki C, Smutnicki A. *Nowe własności harmonogramów cyklicznych w systemie przepływowym*. AGH Automatyka. 2007, Vol. 11(1-2), s. 275–85.
- Sprindzuk VG. *Classical Diophantine Equations*. Lecture Notes in Mathematics. 1993, Vol. 1559, s. 228.
- Suprunenko DA, Aizenshtat VS, Metel'sky AS. *A multistage technological process*. Doklady Academy Nauk BSSR, Vol. 6. 1962, s. 522–41.
- Suraj Z. *On certain problem of resource allocation in systems of concurrent programs*. Prace IPI PAN, ICS PAS Reports 549, 1984.
- Tazza M. *Quantitative analysis of a resource allocation problem: a net theory based proposal*. Concurrency and Nets, Advances in Petri Nets. Springer-Verlag. 1987, s. 511–32.
- Timkovsky VG. *On transition processes in systems of flow type*. Automation Control Systems. 1977, Vol. 1(3), s. 46–9.
- Tricas F, Ezpeleta J. *Some results on siphon computation for deadlock prevention in resource allocation systems modeled with Petri nets*. Proceedings IEEE Conference on Emerging Technologies and Factory Automation. 2003, s. 322–9.
- Tsang E. *Foundations of Constraint Satisfaction*. Academic Press. 1993, s. 300.
- Ułasiewicz J. *Systemy czasu rzeczywistego QNX6 Neutrino*. Warszawa. 2007, s. 301.
- Viswanadham N, Narahari Y. *Performance modeling of automated manufacturing systems*. Englewood Cliffs: Prentice Hall. 1992, s. 350.
- Wang F, Gao Y, Zhou M. *A modified reachability tree approach to analysis of unbounded Petri nets*. IEEE Transactions on Systems, Man and Cybernetics, Part B. 2004, Vol. 34(1), s. 303–8.
- Wójcik R. *Constraint programming approach to designing conflict-free schedules for repetitive manufacturing processes*. In: Cunha PF, Maropoulos PG, (eds.). *Digital enterprise technology Perspectives and future challenges*. Springer. 2007, s. 267–74.
- Wójcik R, Banaszak Z, Roszkowska E. *Automation of self-recovery resource allocation procedures synthesis in FMS*. In: Leiviska K, editor. *Proceedings of IFAC Workshop on CIM in Process and Manufacturing Industries*. Espoo, Finland: Pergamon Press Ltd. 1992, s. 127–32.
- Wójcik R, Bzdyra K, Crisostomo MM, Banaszak Z. *Constraint programming approach to design of deadlock-free schedules in concurrent production systems*. In: Lo Bello L, Sauter T, (eds.). *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation, E*. 2005, s. 135–42.
- Wysk RA, Yang NS, Yoshi Y. *Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches*. Journal of Manufacturing Systems. 1994, Vol. 13, s. 128–38.
- Xie X, Jeng M. *ERCN-merged nets and their analysis using siphons*. IEEE Transactions on Robotics and Automation. 1999, Vol. 15(4), s. 692–703.
- Xing KY, Hu BS, Chen HX. *Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources*. IEEE Transactions on Automatic Control. 1996, Vol. 41(2), s. 289–95.



- Xing KY, Jin X, Feng Y. *Deadlock avoidance Petri net controller for manufacturing systems with multiple resource service*. Proceedings Of the 2005 IEEE International Conference on Robotics and Automation. 2005, s. 4757–61.
- Yalcin A, Boucher TO. *Deadlock avoidance in flexible manufacturing systems using finite automata*. IEEE Transactions on Robotics and Automation. 2000,16(4), s. 424–9.
- Zhou M, Fanti MP. *Deadlock resolution in computer-integrated systems*. Marcel Dekker, Inc. 2004, s. 696.

## Spis rysunków

Rys. W.1. Galop konny ( <a href="http://whitmanhansonphoto.wordpress.com">http://whitmanhansonphoto.wordpress.com</a> ) a), ilość plam słonecznych oraz występowanie recesji w USA (od roku 1900) ( <a href="ftp://ftp.ngdc.noaa.gov">ftp://ftp.ngdc.noaa.gov</a> ) b) .....	13
Rys. W.2. Sieć kolejowa PKP ( <a href="http://rozklad.sitkol.pl">http://rozklad.sitkol.pl</a> ) .....	14
Rys. W.3. System składający się z współbieżnie realizowanych procesów cyklicznych .....	15
Rys. W.4. Madryt – połączenia linii metra ( <a href="maps.google.pl">maps.google.pl</a> ) .....	16
Rys. W.5. Sieć komputerowa .....	17
Rys. W.6. Przyjęty proces podejmowania decyzji.....	19
Rys. 1.1. Struktura rozdziału 1 .....	23
Rys. 1.2. Diagram Gantta reprezentujący harmonogram cykliczny X .....	25
Rys. 1.3. Sieć czynności AOA (activity-on-arrow) przykładowego przedsięwzięcia.....	26
Rys. 1.4. Digraf G dla sieci z rys. 1.3 .....	27
Rys. 1.5. Harmonogram X cyklicznej realizacji przedsięwzięcia z rys. 1.3 .....	29
Rys. 1.6. Cykliczny problem gniazdowy (cyclic jobshop problem) .....	31
Rys. 1.7. Cykliczny harmonogram dla problemu z rys. 1.6 a), fragment harmonogramu odpowiadający sekwencji X' b) .....	33
Rys. 1.8. Przykład problemu przepływowego a), optymalny harmonogram cyklicznej realizacji operacji b), fragment harmonogramu odpowiadający sekwencji X' c).....	34
Rys. 1.9. Przykład harmonogramowania przedsięwzięcia z uwzględnieniem maszyn, (El Amraoui et al. 2008), digraf ograniczeń kolejnościowych a), optymalny harmonogram b), fragment harmonogramu odpowiadający sekwencji X' c).....	35
Rys. 1.10. Przykład systemu wielowariantowej produkcji potokowej (Bocewicz et al. 2009b).....	36
Rys. 1.11. Przykład gniazda produkcyjnego z robotem mobilnym (Dang et al. 2012).....	37
Rys. 1.12. Przykład zrobotyzowanego systemu przepływowego (El Amraoui et al. 2008).....	38
Rys. 1.13. Klasyfikacja problemów harmonogramowania cyklicznego .....	40
Rys. 1.14. Możliwe rozkłady zajęć preferujące grupę $E_2$ wraz z odpowiadającymi im diagramami Gantta .....	42
Rys. 1.15. Przestrzenie obszarów zabronionych preferujące grupę $E_1$ .....	43
Rys. 1.16. Przykład problemu CJP z trzema zadaniami a), harmonogram bez oczekiwania b), harmonogram bez blokad c).....	44
Rys. 1.17. System współbieżnych procesów cyklicznych z trzema procesami a) dopuszczalny harmonogram cyklicznej realizacji operacji b).....	48
Rys. 1.18. Ilustracja blokady procesów (Banaszak et al. 2008).....	50
Rys. 1.19. Przykłady blokad: w której uczestniczą dwa procesy a) w której uczestniczą trzy procesy b).....	50
Rys. 1.20. SWPC charakteryzujący się częściową cyklicznością: struktura a), harmonogram realizacji operacji b).....	51
Rys. 1.21. Przykłady SWPC prowadzących do blokad dla każdego stanu początkowego i dowolnego mechanizmu synchronizacji: system z czterema procesami i czterema zasobami a), system z ośmioma procesami i dwunastoma zasobami (Bocewicz et al. 2012a) b).....	52
Rys. 1.22. Przykłady SWPC wraz z opisującymi je układami równań diofantycznych (Bocewicz i Banaszak 2010) .....	53
Rys. 1.23. Analiza SWPC.....	59
Rys. 1.24. Synteza SWPC .....	60
Rys. 1.25. Składanie SWPC .....	61
Rys. 1.26. Składanie SWMPC, elementy składowe a), system wynikowy b) harmonogram realizacji operacji systemu wynikowego c).....	62
Rys. 1.27. Składanie SWMPC, elementy składowe a), system wynikowy c).....	63

<b>Rys. 1.28.</b> Harmonogram realizacji operacji systemu wynikowego przedstawionego na rys. 1.27b.....	63
<b>Rys. 1.29.</b> Przykład systemu transportowego ESP (Bocewicz i Banaszak 2013b).....	65
<b>Rys. 1.30.</b> SWMPC dla systemu z rys. 1.29 (Bocewicz i Banaszak 2013b).....	66
<b>Rys. 1.31.</b> Przykład czteroportowego routera transmisji danych dla usługi VOD (ang. Video On Demand) wspomagający stosowanie QoS (ang. Quality of Service) (Polak 2004) ..	66
<b>Rys. 1.32.</b> Reprezentacja routera z rys. 1.31 jako systemu współbieżnych multimodalny procesów cyklicznych .....	67
<b>Rys. 1.33.</b> Cykliczny harmonogram realizacji procesów lokalnych i multimodalnych systemu z rys. 1.30.....	69
<b>Rys. 2.1.</b> Związek problemów analizy i syntezy SWMPC z jego zachowaniem i strukturą.....	73
<b>Rys. 2.2.</b> Model wielopoziomowy multimodalnego transportu publicznego (Londyn, maps.google.pl).....	75
<b>Rys. 2.3.</b> Etapy rozpoczęcia operacji $o_{2,3}^1$ potoku $P_2^1$ procesu lokalnego $P_2$ .....	80
<b>Rys. 2.4.</b> Etapy rozpoczęcia operacji $m^2 o_{2,3}^1$ potoku $m^2 P_2^1$ procesu multimodalnego $m^2 P_2$ .....	82
<b>Rys. 2.5.</b> Obszar zastosowań przyjętej klasy SWMPC w kontekście klasy WPP (tab. 1.4) .....	83
<b>Rys. 2.6.</b> Przykład SWMPC z trzema poziomami zachowań: jednym poziomem procesów lokalnych i dwoma poziomami procesów multimodalnych .....	83
<b>Rys. 2.7.</b> Graficzna ilustracja struktury (18) na przykładzie SWMPC z rys. 2.6.....	87
<b>Rys. 2.8.</b> Podsystem transportowy z 6 wózkami samojezdnymi a) jego model SWMPC b) (Bocewicz et. al 2014).....	90
<b>Rys. 2.9.</b> Graficzna ilustracja struktury (18) dla SWMPC z rys. 2.8 .....	92
<b>Rys. 2.10.</b> Cykliczny harmonogram realizacji potoków procesów lokalnych systemu z rys. 2.6 a), fragment harmonogramu odpowiadający sekwencji $X'$ (11) b) .....	94
<b>Rys. 2.11.</b> Przebieg przejściowy prowadzący do cyklu.....	96
<b>Rys. 2.12.</b> Przebieg prowadzący do blokady a), żądania zasobowe w stanie blokady b).....	96
<b>Rys. 2.13.</b> Ilustracja wstrzymywania procesów sąsiadujących.....	98
<b>Rys. 2.14.</b> Niespójny digraf jako przestrzeń stanów $P$ systemu z rys. 2.6 .....	99
<b>Rys. 2.15.</b> Przestrzeń stanów $P$ systemu z rys. 2.6, dla reguł: $\sigma_1^0 = (P_2^1, P_3^1)$ , $\sigma_2^0 = (P_3^1)$ , $\sigma_3^0 = (P_2^1, P_1^1)$ , $\sigma_4^0 = (P_4^1, P_3^1, P_4^2)$ , $\sigma_5^0 = (P_4^1, P_3^1, P_1^1, P_4^2)$ , $\sigma_6^0 = (P_4^1, P_1^1, P_4^2)$ , $\sigma_7^0 = (P_4^1, P_4^2)$ , $\sigma_8^0 = (P_4^1, P_4^2)$ .....	100
<b>Rys. 2.16.</b> Relacja struktura – zachowanie .....	101
<b>Rys. 2.17.</b> Cykliczny harmonogram realizacji procesów systemu z rys. 2.6 .....	102
<b>Rys. 2.18.</b> Dwa kolejne stany systemu z rys. 2.6: stan $S^1$ a), stan $S^2$ b).....	107
<b>Rys. 2.19.</b> Stan niedopuszczalny w systemie z rys. 2.6 .....	109
<b>Rys. 2.20.</b> Przykładowa przestrzeń stanów $P$ SWMPC (Bocewicz et al. 2010) .....	122
<b>Rys. 2.21.</b> Przebiegi SWMPC: ustalony przebieg cykliczny $D_C$ a), przebieg przejściowy prowadzący do cyklu $D_T$ b), przebieg przejściowy prowadzący do blokady $D_T$ c).....	124
<b>Rys. 2.22.</b> Przykłady przebiegów niedopuszczalnych .....	125
<b>Rys. 2.23.</b> Digrafy zachowań: wir a), drzewo b) .....	127
<b>Rys. 2.24.</b> Przestrzeń $P$ reprezentująca zachowanie SWMPC .....	129
<b>Rys. 2.25.</b> Przestrzenie $P^0$ , $P^1$ , $P^2$ na przykładzie SWMPC z rys. 2.6 .....	131
<b>Rys. 2.26.</b> SWMPC bez określonych reguł priorytetowania.....	137
<b>Rys. 2.27.</b> Harmonogram systemu z rys. 2.26 .....	138
<b>Rys. 2.28.</b> Digraf osiągalności alokacji systemu z rys. 2.26.....	140
<b>Rys. 2.29.</b> Przykład harmonogramu cyklicznego systemu z rys. 2.26 .....	140
<b>Rys. 2.30.</b> Przykładowe reguły priorytetowania uzyskane z digrafu z rys. 2.28: dla przebiegu oznaczonego kolorem zielonym a), dla przebiegu oznaczonego kolorem pomarańczowym b), dla przebiegu oznaczonego kolorem niebieskim c) .....	141
<b>Rys. 2.31.</b> Fragment digrafu osiągalności odpowiadający przebiegowi związanemu z harmonogramem przedstawionym na rys. 2.11 .....	143

<b>Rys. 2.32.</b> Reguły priorytetowania dla przebiegów z rys. 2.31: przebiegu przejściowego a), cyklicznego przebiegu ustalonego b) .....	144
<b>Rys. 3.1.</b> Metoda redukcji stanów początkowych a), metoda generowania harmonogramów cyklicznych w oparciu o digrafy pierwszeństwa operacji .....	147
<b>Rys. 3.2.</b> Redukcja stanów początkowych $SB_{(1)}$ zbioru $S^{lp}$ – pierwszy krok procedury wyznaczania cyklicznych przebiegów ustalonych (dla systemu z rys. 2.6).....	149
<b>Rys. 3.3.</b> Kolejne kroki redukcji stanów początkowych (kontynuacja rys. 3.2) .....	150
<b>Rys. 3.4.</b> Odtworzenie wiru ze stanów przebiegu cyklicznego $S_{(8)}^{lp}$ oraz stanów początkowych $SB_{(1)} - SB_{(7)}$ .....	151
<b>Rys. 3.5.</b> Przestrzeń stanów początkowych $SB_{(1)}$ (pierwszy krok Algorytmu 1: $SB_{(1)} =$ $SEARCHALL(PS_{SB_{(1)}})$ ).....	155
<b>Rys. 3.6.</b> Przestrzeń stanów $SB_{(1)} \cup SB_{(2)}$ (drugi krok Algorytmu 1: $SB_{(2)} =$ $SEARCHALL(PS_{SB_{(2)}})$ ).....	155
<b>Rys. 3.7.</b> Przestrzeń stanów $SB_{(1)} \cup SB_{(2)} \cup SB_{(3)}$ (krok 3: $SB_{(3)} = SEARCHALL(PS_{SB_{(3)}})$ ).....	156
<b>Rys. 3.8.</b> Osiągalne cykliczne przebiegi ustalone (krok 4: $V(DC) = SEARCHALL(PS_C)$ ).....	156
<b>Rys. 3.9.</b> Harmonogramy cykliczne dla przebiegów: $D_{C,1}$ a), $D_{C,2}$ b), $D_{C,3}$ c), z rys. 3.8.....	157
<b>Rys. 3.10.</b> Przykładowe przebiegi powstałe w przestrzeni stanów P systemu z rys. 1.32 w wyniku wykorzystania Algorytmu 2 .....	160
<b>Rys. 3.11.</b> Przykłady SWMPC o różnych gęstościach: $g \approx 0,33$ a), $g \approx 0,42$ b), $g = 0,5$ c). 162	
<b>Rys. 3.12.</b> Funkcje czasów rozwiązania problemów typu (73) i (74) (czas odpowiedzi funkcji $SEARCHALL(PS_{SB})$ ) dla struktur o gęstościach: $g = 0,33$ ; $g = 0,5$ ; $g = 0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM) .....	162
<b>Rys. 3.13.</b> SWMPC ( $g = 0,6875$ ) reprezentujący router VOD wielopotokowej transmisji danych między portami: A–C, C–A, B–D, D–B, D–C.....	163
<b>Rys. 3.14.</b> Przykład SWMPC opisanego regułami priorytetowania: $\sigma_1^0 = (P_2^1, P_3^1)$ , $\sigma_2^0 =$ $(P_2^1)$ , $\sigma_3^0 = (P_2^1, P_1^1)$ , $\sigma_4^0 = (P_1^1)$ , $\sigma_5^0 = (P_3^1, P_2^1)$ , $\sigma_6^0 = (P_3^1)$ oraz $\sigma_4^1 =$ $(m^1P_1, m^1P_2)$ , $\sigma_3^1 = (m^1P_2, m^1P_1)$ .....	164
<b>Rys. 3.15.</b> Idea wyznaczania cyklicznych przebiegów ustalonych.....	165
<b>Rys. 3.16.</b> Przestrzeń przebiegów $P^0$ dla systemu z rys. 3.14 .....	166
<b>Rys. 3.17.</b> Cykliczny przebieg ustalony $D_{C,1}$ systemu z rys. 3.14 a), digraf pierwszeństwa operacji $G_1$ reprezentujący $D_{C,1}$ b), $G_1^i$ – rzut digrafu $G_1$ na płaszczyznę c).....	167
<b>Rys. 3.18.</b> Ograniczenie: $x_{i,j}^h(k) = \max\{x_{i,(j-1)}^h(k') + t_{i,(j-1)}; x_{a,b}^c(k'') + \Delta t\}$ .....	170
<b>Rys. 3.19.</b> Związki łączące cykliczny przebieg ustalony $D_C$ , digraf pierwszeństwa operacji i harmonogram cykliczny $X_{DC}$ .....	172
<b>Rys. 3.20.</b> Przykład SWMPC z trzema procesami lokalnymi a), odpowiadający mu cykliczny digraf pierwszeństwa operacji $G_2$ b), $G_2^i$ – rzut digrafu $G_2$ na płaszczyznę c) .....	173
<b>Rys. 3.21.</b> Harmonogram realizacji operacji ustalonego przebiegu cyklicznego $D_{C,1}$ systemu z rys. 3.14.....	178
<b>Rys. 3.22.</b> Digraf pierwszeństwa operacji $G_1^1$ dla procesów multimodalnych $m^1P_1$ , $m^1P_2$ systemu z rys. 3.14.....	179
<b>Rys. 3.23.</b> Harmonogram realizacji operacji procesów multimodalnych i lokalnych systemu z rys. 3.14.....	183
<b>Rys. 3.24.</b> Schemat wyznaczania cyklicznych przebiegów ustalonych SWMPC z rys. 3.14 ...	185
<b>Rys. 3.25.</b> Funkcje czasów rozwiązania problemów typu (89) i (93) (czas odpowiedzi funkcji $SEARCHONE(PS)$ ) dla struktur o gęstościach: $g = 0,33$ ; $g = 0,5$ ; $g =$ $0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM) .....	189
<b>Rys. 3.26.</b> SWMPC charakteryzujące się tym samym zachowaniem .....	190

<b>Rys. 3.27.</b> Funkcje czasów rozwiązania problemów typu (89) i (93) uwzględniające przesunięcia cykliczne reguł priorytetowania, dla struktur o gęstościach: $g = 0,33$ ; $g = 0,5$ ; $g = 0,66$ , (Oz Mozart, Windows 7, Intel Core Duo2 3.00 GHz, 4 GB RAM).....	190
<b>Rys. 3.28.</b> Sposób wyznaczania struktury $SC^{lp}$ gwarantującej istnienie cyklicznych przebiegów ustalonych.....	192
<b>Rys. 3.29.</b> System produkcji potokowej z podsystemem transportowym AGV .....	196
<b>Rys. 3.30.</b> SWMPC przedstawiający system z rys. 3.29.....	197
<b>Rys. 3.31.</b> Poziomy zachowań SWMPC systemu z rys. 3.30 .....	198
<b>Rys. 3.32.</b> Harmonogram cykliczny procesów multimodalnych reprezentujący przyjęty plan produkcyjny.....	201
<b>Rys. 3.33.</b> Wyznaczania parametrów struktury $SC^1$ systemu z rys. 3.29 w strategii top-down.....	202
<b>Rys. 3.34.</b> Harmonogram cykliczny systemu z rys. 3.29.....	205
<b>Rys. 3.35.</b> Analiza SWMPC (zgodna z koncepcją rys. 1.23) uwzględniająca opracowane metody harmonogramowania.....	208
<b>Rys. 4.1.</b> Ogólna idea składania struktur SWMPC.....	209
<b>Rys. 4.2.</b> Przykład SWMPC a) zdekomponowanego na 4 podsystemy b).....	211
<b>Rys. 4.3.</b> Przykład składania dwóch podstruktur SWMPC: $SC_3^1 = SC_1^1 \oplus SC_2^1$ .....	215
<b>Rys. 4.4.</b> Składanie podstruktur $SC_a^{lp} \oplus SC_b^{lp}$ wzdłuż łańcucha zasobów wykorzystanych w procesach o tym samym zwrocie.....	216
<b>Rys. 4.5.</b> SWMPC z przykładu 7.....	218
<b>Rys. 4.6.</b> Podstruktury systemu z rys. 4.5 wraz z odpowiadającymi im harmonogramami cyklicznymi.....	220
<b>Rys. 4.7.</b> Harmonogram cykliczny realizacji procesów systemu z rys. 4.5 .....	221
<b>Rys. 4.8.</b> Składanie struktur według reguły „zamka błyskawicznego”.....	224
<b>Rys. 4.9.</b> Harmonogramy bezkolizyjnej realizacji operacji procesów lokalnych na wspólnym zasobie $R_{k_i}$ : realizacja operacji $o_{i,j}^h$ pomiędzy kolejnymi wykonaniami $o_{q,r}^s$ a), realizacja operacji $o_{q,r}^s$ pomiędzy kolejnymi wykonaniami $o_{i,j}^h$ b).....	225
<b>Rys. 4.10.</b> Harmonogramy bezkolizyjnej realizacji operacji procesów multimodalnych na wspólnym zasobie $R_{k_i}$ : realizacja operacji $m^1 o_{i,j}^h$ pomiędzy kolejnymi wykonaniami $m^1 o_{q,r}^s$ a), realizacja operacji $m^1 o_{q,r}^s$ pomiędzy kolejnymi wykonaniami $m^1 o_{i,j}^h$ b).....	226
<b>Rys. 4.11.</b> Składanie $SC_1^1 \oplus SC_2^1$ .....	229
<b>Rys. 4.12.</b> Składanie $SC_{1-2}^1 \oplus SC_3^1$ .....	232
<b>Rys. 4.13.</b> Harmonogram cykliczny systemu z rys. 4.2.....	233
<b>Rys. 4.14.</b> Idea wyznaczania harmonogramu cyklicznego $X^{lp}$ polegająca na składaniu podstruktur struktury $SC^{lp}$ .....	234
<b>Rys. 4.15.</b> Przykład struktury złożonej z podstruktur izomorficznych a) pojedyncza podstruktura izomorficzna $(i)SC^1$ b).....	237
<b>Rys. 4.16.</b> Składanie podstruktur $(i)SC^1$ poprzez zasoby $(i)R_1$ , $(i)R_3$ oraz $(i)R_9$ .....	238
<b>Rys. 4.17.</b> Podstruktura $(i)SC^1$ z ograniczeniami gwarantującymi naprzemienną realizację potoków $(i)P_1^1$ , $(i)P_2^1$ , $(i)P_3^1$ a), harmonogram cykliczny $(i)X^1$ podstruktury $(i)SC^1$ b) .....	239
<b>Rys. 4.18.</b> Harmonogram cykliczny osiągalny w systemie o strukturze regularnej $SC^1$ z rys. 4.15a.....	241
<b>Rys. 4.19.</b> Zbiór czterech grup podstruktur izomorficznych $(i)SC_I^1$ , $(i)SC_{II}^1$ , $(i)SC_{III}^1$ , $(i)SC_{IV}^1$ a) oraz przykład struktury regularnej zbudowanej z tych podstruktur b)....	243
<b>Rys. 4.20.</b> Składanie podstruktur SWMPC uwzględniające proponowane podejścia.....	244

<b>Rys. 5.1.</b> Analiza potencjalnych przejść pomiędzy cyklicznymi przebiegami ustalonymi .....	248
<b>Rys. 5.2.</b> Wzajemna osiągalność cyklicznych przebiegów ustalonych $D_{C,1}$ , $D_{C,2}$ .....	249
<b>Rys. 5.3.</b> Przykład rzutów przestrzeni zachowań: $P^1$ , $P^0$ oraz $A$ .....	251
<b>Rys. 5.4.</b> Harmonogram cykliczny reprezentujący przebieg: $D_{C,1}$ a) i $D_{C,2}$ b) systemu z rys. 1.30 .....	255
<b>Rys. 5.5.</b> Harmonogram cykliczny reprezentujący przebieg $D_{C,3}$ systemu z rys. 1.30 .....	256
<b>Rys. 5.6.</b> Rzuty przestrzeni zachowań $P^1$ i $P^0$ systemu z rys. 1.30, wraz z przejściem $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2}$ .....	258
<b>Rys. 5.7.</b> Harmonogram ilustrujący przejście: $D_{C,1} \rightarrow D_{C,3} \rightarrow D_{C,2}$ .....	259
<b>Rys. 5.8.</b> Sieć wzajemnych przejść między przebiegami cyklicznymi: wariant z przykładu 8 a), wariant ogólny b) .....	260
<b>Rys. 6.1.</b> Obszary wykorzystania i rozwoju opracowanych metod oceny zachowań cyklicznych SWMPC .....	264
<b>Rys. 6.2.</b> Centrum logistyczne łańcucha dostaw a), wielopoziomowy model struktury reprezentującego go SWMPC b) (Pawlewski 2013) .....	266

## Spis tabel

<b>Tabela 1.1.</b> Ograniczenia determinowane przez digraf $G$ z rys. 1.4.....	27
<b>Tabela 1.2.</b> Uproszczenie ograniczeń z tab. 1.1. zgodnie z (5) i (6).....	28
<b>Tabela 1.3.</b> NP-trudne problemy harmonogramowania cyklicznego (Levner et al. 2010).....	39
<b>Tabela 1.4.</b> Klasy WPP (Banaszak et al. 2008).....	57
<b>Tabela 2.1.</b> Podział SWMPC ( $WPP ol, ks, ju, WPP ol, ks, jg$ ) uwzględniający kryteria: pojemności zasobów $C(R_k)$ , zapotrzebowania operacji procesów na zasoby $\omega_{i,j}^h / m^l \omega_{i,j}^h$ oraz czasów wykonania operacji $t_{i,j} / m^l t_{i,j}$ .....	89
<b>Tabela 2.2.</b> Stany: $S^1$ i $S^2$ w reprezentacji (19) .....	107
<b>Tabela 3.1.</b> Momenty rozpoczęcia operacji realizowanych w kolejnych stanach cyklicznego przebiegu ustalonego $D_{C,1}$ .....	168
<b>Tabela 3.2.</b> Ograniczenia typu (76) dla digrafu $G_1$ z rys. 3.17 .....	170
<b>Tabela 3.3.</b> Ograniczenia opisujące związki między momentami $x_{i,j}^h(k)$ rozpoczęcia operacji determinowane przez digraf pierwszeństwa operacji $G_2$ (rys. 3.20b).....	174
<b>Tabela 3.4.</b> Ograniczenia opisujące związki między momentami $x_{i,j}$ rozpoczęcia operacji determinowane przez digraf pierwszeństwa operacji $G'_1$ (rys. 3.17c) .....	176
<b>Tabela 3.5.</b> Wartości elementów zbioru $X_{DC}$ i okresu $\alpha$ dla SWMPC z rys. 3.14.....	177
<b>Tabela 3.6.</b> Ograniczenia opisujące związki między momentami $m^1 x_{i,j}^h$ rozpoczęcia operacji procesów $m^1 P_1, m^1 P_2$ determinowane przez digraf pierwszeństwa operacji $G_1^1$ (rys. 3.22) .....	182
<b>Tabela 3.7.</b> Wartości elementów zbioru $m^1 X_{DC}'$ i okresu $m^1 \alpha$ dla SWMPC z rys. 3.14 .....	182
<b>Tabela 3.8.</b> Reguły priorytetowania procesów poziomu $SL$ systemu z rys. 3.30 .....	198
<b>Tabela 3.9.</b> Reguły priorytetowania procesów poziomu $SM^1$ systemu z rys. 3.30 .....	200
<b>Tabela 3.10.</b> Czasy wykonania operacji procesów multimodalnych z rys. 3.32 .....	201
<b>Tabela 3.11.</b> Ograniczenia określające zachowanie procesów lokalnych SWMPC z rys. 3.29 .....	203
<b>Tabela 3.12.</b> Ograniczenia determinujące dopuszczalne wartości momentów rozpoczęcia procesów lokalnych SWMPC z rys. 3.29.....	204
<b>Tabela 4.1.</b> Czasy wykonania operacji procesów lokalnych SWMPC z rys. 4.5 .....	219
<b>Tabela 4.2.</b> Czasy wykonania operacji procesów SWMPC z rys. 4.2.....	233
<b>Tabela 4.3.</b> Momenty rozpoczęcia operacji procesów lokalnych SWMPC z rys. 4.2.....	233
<b>Tabela 4.4.</b> Reguły priorytetowania SWMPC z rys. 4.2 .....	234
<b>Tabela 4.5.</b> Czasy wykonania operacji procesów lokalnych i multimodalnych ${}^{(i)}SC^1$ z rys. 4.17a.....	239
<b>Tabela 4.6.</b> Momenty rozpoczęcia operacji procesów lokalnych i multimodalnych ${}^{(i)}SC^1$ z rys. 4.17a.....	240
<b>Tabela 4.7.</b> Reguły priorytetowania podstruktury ${}^{(i)}SC^1$ z rys. 4.17a.....	242
<b>Tabela 5.1.</b> Reguły priorytetowania dla systemu z rys. 1.30.....	254
<b>Tabela 5.2.</b> Czasy wykonania operacji procesów lokalnych systemu z rys. 1.30.....	256

## Skorowidz

<b>A</b>		<b>Ł</b>	
AGV	65	łuk	25,101
alokacja	104	<b>M</b>	
analiza	59,139	maszyna	31
<b>B</b>		marszruta	47,77
blokada	50, 96	(max,+) algebra	18
bottom-up	88,165	model	
<b>C</b>		- wielopoziomowy	75
centrum logistyczne	266	- deklaracyjny	133,134
czynność	26	<b>O</b>	
<b>D</b>		okres	24
diagram Gantta	24	- procesów lokalnych	49
digraf		- procesów multimodalnych	69,102
- osiągalności alokacji	139,140,142	ograniczenia	
- pierwszeństwa operacji	166,179	- kolejnościowe	25
- relacji kolejnościowych	26,27	- realizacji cyklicznej	25
- zachowań	128	- typu max	176,182
drzewo	127	operacja	24,31,47,80
dziedzina	133	operator składania	134,213,217,222
<b>E</b>		OzMozart	20,153,207
ECLIPSe	20,153	osiągalność	
ESP	56,88	- stanów	112
<b>F</b>		- przebiegów	252
funkcja przejścia	119	<b>P</b>	
FIFO	51,97	pojemność zasobów	76
fragment procesu multimodalnego	212	podstruktura	
<b>G</b>		- struktury SWMPC	211
gęstość struktury	161	- izomorficzna	235
graf skierowany	25	potok	77
<b>H</b>		poziom zachowań	76
harmonogram cykliczny	24,171,184,	procesy	
<b>I</b>		- cykliczne	46
indeksy	105	- ciągle	46
ILOG	20,153,207	- dyskretne	46
<b>K</b>		- lokalne	64,76
klasa WPP	57	- multimodalne	64,76
klasa PR	55	- równoległe	46
konkatenacja	68	- sąsiadujące	97
<b>L</b>		- współbieżne	46
LIFO	97	- podejmowania decyzji	19
		predykat	119



prawo alokacji	138	- potomek	120
problem		- rodzic	120
- BCSP	2	- startowy	123
- CJP	32	struktura	
- COP	35	- SWMPC	85
- CPSP	35	- regularna	236
- analizy	45,58,133	strumień	65
- syntezy	45,60,133	synteza	60,134
- składania	60,134	system	
- spełniania ograniczeń	133	- SWPC	47
- wzajemnej osiągalności	248	- SWMPC	66
- gniazdowy	32	- zrobotyzowany	37
- przepływowy	34	- rozproszony	55
- otwarty	35	<b>T</b>	
- optymalizacyjny	24	top-down	88,130,195
- decyzyjny	45	<b>W</b>	
programowanie z ograniczeniami	153	warunek wzajemnego wykluczania	44
przebieg		wir	127
- przejściowy	124	wierzchołek	25,101,166
- cykliczny	124	wózek samojezdny	90
przedsięwzięcie	26	„w przód”	45
przestrzeń		„wstecz”	45
-stanów	94,100	<b>Z</b>	
-alokacji	250	zadanie	32
<b>R</b>		zachowanie	94
reguły priorytetowania	52,78	zagłodzenie	50
reguła „zamka błyskawicznego”	222	zasób	47
redukcja stanów	139	złożoność obliczeniowa	160,188
robot mobilny	36	zmiennie	
rozstrzygalność	135	- decyzyjne	133
router	66	- rozmyte	78,265
równania diofantyczne	29	- losowe	78,265
rzut		<b>Ż</b>	
- przestrzeni	130	żądania zasobowe	50
- stanu	249		
<b>S</b>			
semafory	105		
sieć			
- czynności	26		
- wzajemnej osiągalności	260		
składanie	61,134		
stan	105		
- blokadowy	122		
- dopuszczalny	109		
- początkowy	121		