C3

na prawach rękopisu

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ

Komunikat nr 91

PEWNE PROBLEMY ROZDZIAŁU ZADAŃ W WIELOPROCESOWYM SYSTEMIE CYFROWYM W WARUNKACH PROBABILISTYCZNYCH

/ Rozprawa doktorska/

Zbigniew Huzar

Promotor:
Prof.dr Jerzy Bromirski

Słowa kluczowe: system operacyjny, praca równoległa, rozdział zadań.

Wrocław 1974

mgr inż. Zbigniew Huzar
Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej
Wrocław, ul. Janiszewskiego 11/17
budynek C-3

Komunikat wpłynął 15.I.1974r.

SPIS TREŚCI

WYKAZ WAŻNIEJSZYCH SYMBOLI I OZNACZEŃ

$\wedge$	- konjunkcja
$\Longrightarrow$	- implikacja
$\forall$	- kwantyfikator ogólny
$\exists$	- kwantyfikator szczegółowy
$a \in A$ $(a \notin A)$	- element a należy (nie należy) do zbioru A
$A \cup B$	- suma zbiorów A i B
$A \setminus B$	- różnica zbiorów A i B
$\bigcup_{i \in I} A_i$	- suma zbiorów A_i, $i \in I$
Θ	- zbiór pusty
$A \subset B$	- A jest podzbiorem B
$\{z : \mathcal{P}(z)\}$	- zbiór z posiadających własność $\mathcal{P}(z)$
$A \times B$	- iloczyn kartezjański zbiorów A i B
$\lvert A \rvert$	- moc zbioru A
$\prod_{i \in I} W_i$	- iloczyn liczbowy W_i, $i \in I$
$\sum_{i \in I} W_i$	- suma liczbowa W_i, $i \in I$
$\mathbb{R}^+$	- zbiór nieujemnych liczb rzeczywistych
$pr\{A\}$	- prawdopodobieństwo zdarzenia A
$E\,t$	- wartość oczekiwana zmiennej losowej t
$t \mid b$	- warunkowa zmienna losowa $(t \geqslant b)$
$F_i(F_i^b)$	- dystrybuanta zmiennej losowej t_i $(t_i \mid b)$, $i = 1,2,\ldots,m$
$f_i(f_i^b)$	- rozkład prawdopodobieństwa określony przez $F_i(F_i^b)$, $i = 1,2,\ldots, m$
W	- system obsługi

W^i	- podsystem obsługi, stanowiska obsługi i-tego rodzaju $i = 1,2,\ldots, K$
G	- graf skierowany
Z	- zbiór zadań elementarnych
Q	- matryca incydencji
P	- wektor rozkładów prawdopodobieństw
R	- wektor rodzaju obsługi
T	- wektor losowy
$\mathcal{Z}$	- zadanie
s_k	- stan wykonania zadania
S^k	- zbiór stanów wykonania zadania
S^*	- zbiór dopuszczalnych stanów wykonania zadania
S_{odp} (S_{dec})	- zbiór stanów odpowiedzi (decyzji)
SO_i (SD_i)	- i-ta warstwa stanów odpowiedzi (decyzji)
PO_i (PD_i)	- i-te piętro stanów odpowiedzi (decyzji)
$\prec$	- relacja poprzedzania stanów
$\overset{d}{\prec}$ $(\overset{q}{\prec})$	- relacja d-następowania (q-następowania) stanów
$D: S_{odp} \rightarrow S_{dec}$	- funkcja rozdziału
$\mathcal{D}$	- zbiór funkcji rozdziału
$\mathcal{T}_D$ $(\mathcal{T}_b)$	- drzewo wykonania (bieżące) zadania
$\mathcal{T}_{\mathcal{D}}$	- zbiór drzew wykonania
ł	- droga w grafie G
$Ł(G, s_k)$	- zbiór dróg w zredukowanym grafie G
$DG(s_k)$	- dolne ograniczenie stanu decyzji
$PDG(s_k)$ $(WDG(s_k))$	- pierwotne (wtórne) dolne ograniczenie stanu decyzji
$\square$	- znak końca definicji, twierdzenia, dowodu itp.

1. PRZEGLAD MODELI I METOD ROZWIĄZYWANIA ZAGADNIENIA ROZDZIAŁU ZADAŃ

1. Wprowadzenie

Przedstawiona praca dotyczy jednego z podstawowych problemów teorii systemów operacyjnych współczesnych systemów cyfrowych, jakim jest kompleks zagadnień określonych mianem zagadnienia rozdziału resursów systemu. Wiele istotnych problemów zarządzania pracą elementów systemu cyfrowego, takich jak np. zarządzanie pamięcią, planowanie obciążeń procesorów i urządzeń wejścia/wyjścia itp., mogą być sprowadzone do zagadnienia rozdziału resursów [20, 28, 41, 50]. W pracy zajęto się problemem zarządzania pracą równolegle pracujących stanowisk obsługi - procesorów w wieloprocesorowym systemie cyfrowym. Zagadnienie to, znane w literaturze angielskiej jako "parallel processing" lub "parallel computation", jest obecnie szeroko rozwijane. Idea prowadzenia równoległych obliczeń sięga połowy XIX w. - czasów Ch. Babbage'a [38]. Szczególnie intensywnie jest rozwijana od początku lat 60-tych naszego stulecia, kiedy powstały pierwsze systemy równolegle pracujących maszyn cyfrowych, a zwłaszcza kiedy uświadomiono sobie, że w związku ze zbliżeniem się szybkości sekwencyjnie wykonywanych obliczeń do pewnego nieprzekraczalnego poziomu, jedyna możliwość ich przyśpieszenia polega na jednoczesnym wykonywaniu niezależnych fragmentów programu na różnych maszynach cyfrowych.

Wśród aktualnie prowadzonych prac z zakresu teorii obliczeń równoległych można wyróżnić dwa podstawowe kierunki [42, 52]:

- poszukiwanie metod przedstawiania algorytmów obliczeniowych w postaci "równoległych", niezależnych fragmentów, najbardziej dogodnych do wykonania w wieloprocesorowym systemie cyfrowym;
- poszukiwanie metod optymalnego sterowania wykonywaniem "równolegle" przedstawionych algorytmów.

Do zakresu pierwszego kierunku należą np. prace [40, 45], zajmujące się "zrównolegnieniem" pewnych konkretnych problemów obliczeniowych, czy też prace [47, 50, 62, 65], zajmujące się ogólnymi modelami przedstawiania "równoległości" algorytmów. Szczególnie interesująca jest praca [42], stanowiąca autorytatywny przegląd modeli i prac z tego kierunku. Pochodnymi kierunkami prowadzonych prac są:

- prace nad językami umożliwiającymi zapis "równoległości" np. klasyczne operatory FORK, JOIN E.S. Schwartza [22], języki macierzowe [34], język PL/1 [7];
- prace nad automatycznym rozpoznawaniem "równoległości" na etapie translacji sekwencyjnie pisanych programów [23], czy też badanie występowania "równoległości" w klasycznie pisanych programach [38];
- prace nad organizacją wewnętrznej struktury logicznej systemu cyfrowego np. [8, 10, 39, 56].

Przedstawiona praca należy do drugiego z wymienionych wyżej podstawowych kierunków prac, których omówienie jest zawarte w następnym punkcie (1.2).

1.2. Problemy sterowania rozdziałem zadań

Problemy sterowania rozdziałem zadań są znane w literaturze radzieckiej m.in. jako zagadnienia "kalendarnovo płanirovanija", zaś w literaturze angielskiej jako "the machine assigment problems" bądź jako "the job sequencing (scheduling) problems". Zagadnieniom tym jest poświęcona obszerna literatura np. [13, 26, 54, 63, 69, 70, 71] a szczególnie [9, 33, 51], które stanowią szerokie przeglądy modeli i metod ich rozwiązywania.

Wiele spośród deterministycznych modeli planowania kalendarzowego było formułowanych bezpośrednio dla pewnych interpretacji pracy systemów cyfrowych. Są to, począwszy od klasycznej pracy [29], np. [2, 3, 5, 16, 27, 44, 46, 50, 52, 59].

Do modeli probabilistycznych systemu cyfrowego [4, 14, 37, 41,50] zalicza się klasyczne modele kolejkowe [6, 21, 60], jak również pewne ich modyfikacje [11, 17, 30, 31, 32, 37, 43, 53, 55, 61, 67]. Ostatnia z wymienionych prac [67] daje przegląd modeli pracy procesora a ponadto przedstawia modele symulacyjne pracy systemu cyfrowego. Pośrednio interesujące są prace związane z planowaniem pewnych przedsięwzięć w warunkach probabilistycznych np. [25, 35, 58].

Do rozwiązywania formułowanych tu zagadnień stosuje się szereg metod z zakresu badań operacyjnych. Należą do nich m.in. metody programowania nieliniowego [1], metody optymalizacyjne [49, 68], sieciowe [18], wykorzystuje się aparat teorii grafów [12, 72], gier [48] itp. Obszerne przeglądy i klasyfikacje stosowanych metod zawierają częściowo prace [9, 51].

W przedstawionych dalej rozważaniach wykorzystuje się klasyczny rachunek prawdopodobieństwa [15], metodę programowania dynamicznego [68] oraz metodę podziału i ograniczeń [57, 64, 66].

1.3. Cel i zakres pracy

Podstawowy problem jaki formułuje się w pracy jest następujący. Dany jest pewien model zadania jako zbiór programów, które mogą być wykonywane w pewnej ograniczonej względem siebie kolejności (zadanej grafem skierowanym), zaś czasy wykonywania pojedynczych programów są zmiennymi losowymi o zadanych rozkładach prawdopodobieństwa. Zadanie ma być wykonywane w systemie złożonym z pewnej liczby jednakowych, niezależnie pracujących procesorów. Celem pracy jest znalezienie dynamicznej reguły decyzyjnej, kierującej odpowiednie programy do wykonania w systemie, takiej aby minimalizować wartość oczekiwaną czasu wykonania całości zadania.

W rozdziale 2 pracy przedstawia się formalne modele zadania, systemu wykonawczego, a następnie formułuje się problem optymalizacyjny. Sformułowany problem jest adaptacją i rozszerzeniem analogicznych zagadnień deterministycznych, którym poświęcono szereg rozważań [2, 16, 24, 29, 44].

Rozdział 3 jest poświęcony rozwiązaniu szczególnego przypadku problemu postawionego w rozdziale 2, mianowicie optymalnemu rozdziałowi zadań o wykładniczych rozkładach prawdopodobieństwa czasów wykonywania zadań elementarnych (pojedynczych programów).

Rozdział 4 zawiera rozważania na temat optymalnego rozdziału zadań o dowolnych rozkładach czasów wykonywania zadań elementarnych, a w rozdziale 5 przedstawia się wnioski i ocenę przydatności proponowanych rozwiązań.

Przedstawiony spis literatury zawiera podstawowe, dostępne autorowi pozycje. Należy jednak zaznaczyć, że ze względu na intensyfikację prowadzonych badań w omawianym zakresie, istnieje wiele prac trudno dostępnych, zawartych zwłaszcza w amerykańskich materiałach konferencyjnych i rozprawach doktorskich, a także wiele prac publikowanych tylko informacyjnie (np. prace prowadzone w ramach amerykańskich programów rządowych).

W dodatku przedstawiono program napisany w ALGOLu-60 na m.c.ODRA-1204, stanowiący załącznik do rozdziału 3, który wyszukuje funkcję optymalnego rozdziału (bez przerwań) zadań o wykładniczych rozkładach prawdopodobieństwa czasów wykonywania zadań elementarnych w jednofazowym systemie obsługi.

2. PRZEDSTAWIENIE PROBLEMU

2.1. Model systemu wykonawczego i model zadania

Definicja 2.1. Systemem wykonawczym (obsługi) będziemy nazywać zbiór

$$W = \{W^1, W^2, \ldots, W^K\}, \tag{2.1}$$

gdzie

$$W^i = \{w_1^i, w_2^i, \ldots, w_{n_i}^i\} \quad (i = 1,2,\ldots, K), \tag{2.2}$$

Podzbiory W^i są zestawami n_i-jednorodnych niezależnie pracujących stanowisk wykonawczych (obsługi) typu i. System wykonawczy W będziemy nazywać jednofazowym, jeżeli $K = 1$, albo wielofazowym, jeżeli $K > 1$. □

Niech $G = (Z,Q)$, gdzie Z - jest zbiorem wierzchołków, a Q - matrycą incydencji, będzie grafem skierowanym, nie posiadającym dróg cyklicznych ani pętli, z jednym wierzchołkiem początkowym i z pewnym wierzchołkiem końcowym. Elementy zbioru $Z = \{z_0, z_1, \ldots, z_{m+1}\}$ będziemy nazywać zadaniami elementarnymi; z_0 - elementarnym zadaniem początkowym, z_{m+1} - elementarnym zadaniem końcowym. Łuki wychodzące z wierzchołka $z_i \in Z$ mogą być podzielone na co najwyżej M rozłącznych podzbiorów. Zadania elementarne, którym odpowiada jeden podzbiór wychodzących łuków nazywamy elementarnymi zadaniami operacyjnymi, zaś pozostałe, którym odpowiada więcej niż jeden podzbiór - elementarnymi zadaniami decyzyjnymi.

Każdemu zadaniu elementarnemu $z_i \in Z$ przyporządkowuje się wektor parametrów (r_i, t_i, p_i), którego poszczególne składowe określają:

r_i - rodzaj obsługi (typ stanowiska obsługi) wymagany przez z_i w systemie W, $r_i \in \{1,2,\ldots, K\}$;

t_i - nieujemną zmienną losową, o znanej dystrybuancie F_i, czasu wykonywania z_i przez stanowisko wykonawcze rodzaju r_i;

$p_i = (p_{i1}, p_{i2}, \ldots, p_{iM})$ - rozkład prawdopodobieństwa wyboru odpowiedniego typu podzbioru łuków wychodzących z wierzchołka z_i.

O zmiennych losowych t_i $(i = 0,1,\ldots,m+1)$ zakładamy, że są niezależne oraz $Et_i < \infty$. Ponadto, bez utraty ogólności, zakłada się, że $t_0 = t_{m+1} = 0$, oraz że z_0, z_{m+1} są elementarnymi zadaniami operacyjnymi. Oczywiście, dla zadań operacyjnych $p_{i1} = 1$, $p_{i2} = \ldots = p_{iM} = 0$.

Elementy q_{ij} $(i,j = 0,1,\ldots, m+1)$ matrycy incydencji Q przyjmują wartości ze zbioru $\{-M,\ldots, -1,0,1,\ldots, M\}$. Znaczenie przyjmowanych wartości jest następujące: $q_{ij} = 0$ określa brak łuków pomiędzy wierzchołkami z_i a z_j; $q_{ij} = 1, 2, \ldots, M$ określa, że z wierzchołka z_i prowadzi się łuk do wierzchołka z_j; znaczenie $q_{ij} = -1, -2, \ldots, -M$ wynika z własności $q_{ij} = -q_{ji}$.

Definicja 2.2. Uporządkowaną czwórkę

$$\mathcal{Z} = (G, R, T, P) \tag{2.3}$$

gdzie G jest wyżej określonym grafem, zaś

$$\begin{aligned} R &= (r_o, r_1, \ldots, r_{m+1}), \\ T &= (t_o, t_1, \ldots, t_{m+1}), \\ P &= (p_o, p_1, \ldots, p_{m+1}), \end{aligned} \tag{2.4}$$

są wektorami o składowych określonych wyżej, będziemy nazywać zadaniem dla systemu obsługi W. $\square$

2.2. Wykonanie zadania w systemie obsługi

Wykonanie zadania $\mathcal{Z}$ w systemie obsługi W polega na wykonaniu pewnego podzbioru Z_o zbioru zadań elementarnych Z. Oczywiście z_o, $z_{m+1} \in Z_o$. Kolejność wykonywania zadań elementarnych jest określona przez pewne zasady. Dla ich przedstawienia wprowadza się następujące pojęcia. O wszystkich łukach grafu G przed rozpoczęciem wykonywania zadania $\mathcal{Z}$ mówimy, że są nieaktywne. O łukach wychodzących z wierzchołka elementarnego zadania operacyjnego mówimy, że stają się aktywne, gdy zostanie zakończone jego wykonywanie. O łukach wychodzących z wierzchołka elementarnego zadania decyzyjnego mówimy, że aktywne stają się łuki typu wylosowanego w trakcie jego wykonywania oraz, że pozostałe łuki stają się obojętne.

Wykonanie zadania $\mathcal{Z}$ rozpoczyna się od wykonania elementarnego zadania początkowego z_o. Następnie mogą być wykonywane inne zadania elementarne. Dane zadanie elementarne $z_i \in Z$ może być wykonywane, jeżeli w zbiorze łuków wchodzących do tego zadania znajdują się tylko łuki aktywne lub obojętne. Wykonanie zadania elementarnego z_i polega na skierowaniu go na dowolne stanowisko obsługi typu r_i, na którym pozostaje nieprzerwanie aż do czasu jego zakończenia t, który jest realizacją zmiennej losowej t_i - obsługa bez przerwań, bądź na którym wykonuje się częściowo przez pewną liczbę odcinków czasu, których sumaryczna długość wynosi t - obsługa z przerwami. Zadanie elementarne z_i wchodzi do podzbioru Z_o, jeżeli w trakcie wykonywania zadania $\mathcal{Z}$ pojawi się chociaż jeden aktywny łuk wchodzący do wierzchołka z_i.

2.3. Przestrzeń stanów wykonywania zadania

Dla opisu kolejności wykonywania bez przerwań zadań elementarnych, wprowadza się pojęcie stanu wykonania zadania $\mathcal{Z}$.

Definicja 2.3. Stanem wykonania zadania $\mathcal{Z}$ nazywa się matrycę

$$s_k = \begin{bmatrix} a_o^k, & a_1^k, & \ldots, & a_{m+1}^k \\ b_o^k, & b_1^k, & \ldots, & b_{m+1}^k \end{bmatrix} \tag{2.5}$$

gdzie

$$a_i^k \in A = \{-1,0,1, \ldots, M\},$$

$$b_i^k \in \mathbb{R}^+ \quad (i = 0,1,\ldots, m+1).$$

Znaczenie poszczególnych elementów matrycy jest następujące: $a_i^k = -1$ określa, że zadanie elementarne z_i nie było jeszcze wykonywane; $a_i^k = 0$ - że z_i jest w trakcie wykonywania; $a_i^k = 1,2,\ldots,M$ - że z_i zostało wykonane, i że podczas jego wykonywania wylosowano odpowiednio $1,2,\ldots,M$ typ łuków wychodzących z wierzchołka z_i. Natomiast b_i^k określa sumaryczną długość czasu, w którym wykonywało się zadanie elementarne z_i. □

Początkowym stanem wykonania zadania $\mathcal{Z}$ jest więc stan

$$s_o = \begin{bmatrix} -1, & -1, & \ldots, & -1 \\ 0, & 0, & \ldots, & 0 \end{bmatrix}, \tag{2.6}$$

natomiast stan końcowy s_L nie jest określony jednoznacznie. Jego postać zależy od wylosowanego w trakcie wykonywania zadania $\mathcal{Z}$ zbioru zadań elementarnych Z_o. Elementy a_i dla $z_i \in Z_o$ będą przyjmować wartości $a_i^L \geqslant 1$, zaś dla $z_i \notin Z_o$ pozostaną $a_i = -1$. Elementy b_i^L dla $z_i \in Z_o$ przyjmą wartości realizacji odpowiednich zmiennych losowych t_i, zaś dla $z_i \notin Z_o$ pozostaną $b_i = 0$. Liczba stanów końcowych, różnych ze względu na wartości elementów a_i, nie przekracza M^h, gdzie h jest liczbą elementarnych zadań decyzyjnych.

Matryce s_k stanów wykonania zadania $\mathcal{Z}$ przyjmują wartości ze zbioru S postaci

$$\underbrace{\begin{bmatrix} A \times A \times \dots \times A \\ \mathbb{R}^+ \times \mathbb{R}^+ \times \dots \times \mathbb{R}^+ \end{bmatrix}}_{(m+2) \text{ razy}}$$

W zbiorze S należy wyróżnić pewien podzbiór S^*, nazwany zbiorem dopuszczalnych stanów wykonania zadania $\mathfrak{Z}$, tj. takich stanów w jakich możliwe jest znalezienie się zadania $\mathfrak{Z}$ w trakcie wykonywania. Zbiór S^* jest generowany przez $\mathfrak{Z}$, W oraz dyscyplinę obsługi.

Wykonanie zadania $\mathfrak{Z}$, jako realizację pewnej funkcji losowej, opisuje się pewną trajektorią stanów s_o, $s_1, \dots, s_{2k}$ ze zbioru S^* takich, że

$$s_o \prec s_1 \prec \dots \prec s_{2k}, \qquad (2.7)$$

gdzie $\prec$ jest symbolem relacji słabego porządku określonej poniżej, zaś s_o jest stanem początkowym postaci (2.6), a s_{2k} jest jednym z możliwych stanów końcowych. Stany s_i $(i = 0,2,\dots,2k)$ nazywa się stanami odpowiedzi i zbiór wszystkich możliwych stanów odpowiedzi oznacza się przez S_{odp}, zaś stany s_j $(j = 1,3,\dots, 2k-1)$ nazywa się stanami decyzji i zbiór wszystkich możliwych stanów decyzji oznacza się przez S_{dec}; oczywiście $S^* = S_{odp} \cup S_{dec}$. Relację słabego porządku zwaną relacją poprzedzania stanów określa się następująco:

<u>Definicja 2.4.</u> Mówimy, że stan s_l następuje po stanie s_k (albo, że s_k poprzedza s_l), co notujemy $s_k \prec s_l$, jeżeli $a_i^k \geqslant 1$, to $a_i^k = a_i^l$ oraz jeżeli $a_i^k < 1$, to $a_i^k \leqslant a_i^l$ dla $i = 0,1,\dots, m+1$. $\square$

Wprowadza się następujące oznaczenia. Dla dowolnego $s_k \in S$ utworzymy zbiory

$$\mathcal{X}_a(s_k) = \{ z_i \in Z : a_i^k = a \} \qquad (2.8)$$

gdzie $a \in A$. Wówczas zbiór

$$\mathcal{X}^+(s_k) = \{ z_i \in \mathcal{X}_{-1}(s_k) : \forall_{q_{ij} < 0} \; z_j \in \bigcup_{l=1}^{M} \mathcal{X}_l(s_k) \} \qquad (2.9)$$

zawiera podzbiór tych żadań elementarnych, które mogą być wykonane w następnym momencie czasu, po osiągnięciu przez zadanie $\mathfrak{Z}$ stanu s_k. Natomiast zbiór

$$\mathcal{X}^-(s_k) = \{ z_i \in \bigcup_{l=1}^{M} \mathcal{X}_l(s_k) : \forall_{q_{ij} > 0} \; z_j \in \mathcal{X}_{-1}(s_k) \} \qquad (2.10)$$

zawiera podzbiór tych zadań elementarnych, które mogły być wykonane bezpośrednio przed momentem osiągnięcia przez zadanie $\mathcal{Z}$ stanu s_k. Ponadto, dla $s_k, s_l \in S$, będziemy oznaczać

$$\chi(s_k \otimes s_l) = \{z_i \in Z : a_i^k \neq a_i^l\}. \tag{2.11}$$

Wśród zbiorów $\chi_a(s_k)$, $\chi^+(s_k)$, $\chi^-(s_k)$ będziemy wyróżniać podzbiory zadań elementarnych o tym samym rodzaju obsługi r $(r = 1,2,\dots, K)$. Więc

$$\chi_{ar}(s_k) = \{z_i \in \chi_a(s_k) : r_i = r\}, \tag{2.12}$$

podobnie określa się $\chi_r^+(s_k)$ i $\chi_r^-(s_k)$.

Definicja 2.5. Mówimy, że stan $s_l \in S_{dec}$ d-następuje po stanie $s_k \in S_{odp}$ (albo, że s_k d-poprzedza s_l), co notujemy $s_k \overset{d}{\prec} s$, jeżeli spełnione są następujące warunki:

(i) $\chi^+(s_k) \cup \chi_o(s_k) \neq \Theta \Rightarrow \chi_o(s_l) \neq \Theta$,

(ii) $\chi_o(s_l) \subset \chi^+(s_k) \cup \chi_o(s_k)$,

(iii) $|\chi_{o,r}(s\ )| \leqslant \min(|W^r|, |\chi_r^+(s_k)| + |\chi_{o,r}(s_k)|)$

dla $r = 1,2,\dots, K$. □

Warunek (i) oznacza, że nie dopuszcza się postoju wszystkich stanowisk obsługi, gdy istnieją zadania elementarne gotowe do wykonania; warunek (ii) dopuszcza wykonywanie zadań elementarnych zgodnie ze strukturą grafu zadanie; warunek (iii) określa maksymalną liczbę zadań elementarnych, które mogą być przeznaczone do wykonania na określonych rodzajach stanowisk obsługi.

Łatwo sprawdzić, że jeżeli $s_k \overset{d}{\prec} s$, to $s_k \prec s_l$. Zakłada się, że d-przejście tj. czasy podejmowania decyzji wyboru zadań elementarnych do wykonania, odbywają się natychmiastowo, bez strat czasu.

Definicja 2.6. Funkcję $D : S_{odp} \rightarrow S_{dec}$ nazywamy funkcją rozdziału zadań elementarnych Z do wykonania w systemie obsługi W, jeżeli dla każdych $s_k \in S_{odp}$, $s_l \in S_{dec}$ takich, że $s_l = D(s_k)$ spełniony jest warunek $s_k \overset{d}{\prec} s$. Zbiór wszystkich możliwych funkcji rodziału oznaczamy przez $\mathcal{D}$. □

Definicja 2.7. Niech q będzie liczbą naturalną taką, że $1 \leqslant q \leqslant$ $\leqslant |\mathcal{X}_o(s_k)|$. Mówimy, że stan $s_l \in S_{odp}$ q-następuje po stanie $s_k \in S_{dec}$ (albo, że s_k q-poprzedza s_l), co notujemy $s_q \overset{q}{\prec} s$, jeżeli istnieje q elementowy podzbiór $I_q = \{i_1, i_2, \dots, i_q\}$ zbioru wskaźników $I = \{0, 1, \dots, m+1\}$ taki, że spełnione są warunki:

$$\text{(i)} \quad \forall_{i \in I_q} \; z_i \in \mathcal{X}_o(s_k) \wedge z_i \in \bigcup_{j=1}^{M} \mathcal{X}_j(s_l),$$

$$\text{(ii)} \quad \forall_{i \notin I_q} \; (z_i \in \mathcal{X}_o(s_k) \Rightarrow b_i^l = b_i^k + b) \wedge$$

$$\wedge \; (z_i \notin \mathcal{X}_o(s_k) \Rightarrow b_i = b_i^k),$$

gdzie b jest wartością realizacji zmiennej losowej określonej wzorem

$$t(s_k, s\ ) = \min_{z_j \in \mathcal{X}_o(s_k)} (t_j | b_j^k) \;, \tag{2.13}$$

z kolei $t_j | b_j^k$ oznacza warunkową zmienną losową powstałą ze zmiennej losowej t_j po dołączeniu warunku, że $t_j \geqslant b_j^k$. $\square$

Ostatnia definicja określa sposób przejścia ze stanu $s_k \in S_{dec}$ do stanu $s_l \in S_{odp}$ polegający na zakończeniu wykonywania q zadań elementarnych. Czas q-przejścia jest zmienną losową $t(s_k, s\ )$ (2.13). Prawdopodobieństwo q-przejścia z s_k do s_l będziemy oznaczać przez $pr\{s_k \overset{q}{\prec} s\}$. Łatwo również sprawdzić, że jeżeli $s_k \overset{q}{\prec} s_l$, to $s_k \prec s_l$.

Uściślając wprowadzone pojęcie trajektorii wykonania zadania $\mathfrak{Z}$, można stwierdzić, że każda trajektoria (2.7) jest ciągiem stanów z S^* takich, że

$$s_o \overset{d}{\prec} s_1 \overset{q_1}{\prec} s_2 \overset{d}{\prec} \dots \overset{d}{\prec} s_{2k-1} \overset{q_k}{\prec} s_{2k} \tag{2.14}$$

gdzie $q_1 + q_2 + \dots + q_k = |Z_o|$ $(k \leqslant m+2)$ oraz $\mathcal{X}^+(s_{2k}) = \Theta$.

Definicja 2.8. Zbiór stanów dopuszczalnych S^* z rodziną trajektorii (2.14) będziemy nazywać przestrzenią stanów wykonania zadania $\mathfrak{Z}$. $\square$

2.4. Kryterium optymalnego wyboru funkcji rozdziału

Definicja 2.6 określa zbiór $\mathfrak{D}$ wszystkich sensownych funkcji rozdziału zadań elementarnych Z do wykonania w systemie obsługi W. Często w rozwiązaniach praktycznych poszukiwanie optymalnego rozwiązania zawęża się do pewnej klasy funkcji. Spotykana jest na przykład zasada, że nie mogą w systemie obsługi W istnieć niepracujące stanowiska obsługi, jeżeli istnieją czekające zadania elementarne, które mogą być na nich wykonywane. Zastosowanie tej zasady w rozpatrywanym modelu wymaga zastąpienia równością nierówności w warunki (iii) definicji 2.5.

Jako kryterium wyboru optymalnej funkcji rozdziału przyjmuje się wartość oczekiwaną czasu zakończenia wykonywania zadania $\mathfrak{Z}$. Funkcję $D_0 \in \mathfrak{D}$ uważamy za optymalną jeżeli minimalizuje tę wartość.

Zastosowanie przyjętego kryterium jest sensowne wówczas, gdy zadanie $\mathfrak{Z}$ ma być wykonywane wielokrotnie. Z taką sytuacją spotykamy się np. gdy zadanie $\mathfrak{Z}$ jest zestawem programów w cyfrowych systemach sterowania procesami technologicznymi, a także wtedy, gdy $\mathfrak{Z}$ traktujemy jako pewien biblioteczny, standardowy zestaw programów systemu cyfrowego.

Konsekwencją przyjęcia powyższego kryterium jest ustalenie wielkości b w warunku (ii) definicji 2.7 jako wartości oczekiwanej zmiennej losowej (2.13). Warto zauważyć, że wartość oczekiwana czasu wykonania zadania $\mathfrak{Z}$ wg pewnej trajektorii (2.14) wyraża się wzorem

$$E\,t\,(s_0, s_1, \ldots, s_{2k}) = \sum_{i=1}^{k} E\,t(s_{2i-1}, s_{2i}), \qquad (2.15)$$

czyli jest sumą wartości oczekiwanych niezależnych funkcji losowych (p. lemat 3.1) $t(s_{2i-1}, s_{2i})$ $(i = 1,2,\ldots, k)$.

Warto również zwrócić uwagę, że wprowadzony model opisu - przestrzeń wykonania zadania $\mathfrak{Z}$, pozwala, drogą małych modyfikacji, na uwzględnienie czasów podejmowania decyzji, tj. czasów d-przejść. Modyfikacje polegałyby na wprowadzeniu pewnej funkcji $t'(s_k, s_l)$ czasów d-przejścia ze stanu $s_k \in S_{odp}$ do stanu $s_l \in S_{dec}$ oraz na rozbudowie definicji 2.5. Rozbudowa polegałaby na innym określeniu wartości b_i $(i = 0,1,\ldots, m+1)$, a mianowicie

$$b_i^l = b_i^k + b'$$

gdzie $b' = t'(s_k, s_l)$. Wprowadzone modyfikacje nie wpływają na istotę dalej prowadzonych rozważań.

3. ROZDZIAŁ ZADAŃ O WYKŁADNICZYCH ROZKŁADACH PRAWDOPODOBIEŃSTWA CZASÓW WYKONYWANIA ZADAŃ ELEMENTARNYCH

3.1. Wprowadzenie

W bieżącym rozdziale rozważa się problemy rozdziału zadań $\mathfrak{Z}$ o wykładniczych rozkładach prawdopodobieństwa zmiennych losowych $t_1, t_2, \ldots, t_m$, tzn. o dystrybuantach postaci

$$F_i(t) = \begin{cases} 1 - \exp(-\lambda_i t) & \text{dla } t \geqslant 0, \\ 0 & \text{dla } t < 0, \end{cases} \tag{3.1}$$

gdzie $\lambda_i > 0$ $(i = 1,2,\ldots, m)$. Przyjęta klasa zadań zasługuje na uwagę ze względu na możliwość uproszczenia rozważań, wynikającą ze szczególnych własności wykładniczego rozkładu prawdopodobieństwa. Taką metodologię stosuje się powszechnie w literaturze, np. [11, 43, 60, 61]. Rzeczywiste rozkłady prawdopodobieństwa dają się czasem, z wystarczającą dla praktyki dokładnością aproksymować rozkładami wykładniczymi. Ponadto przyjęcie konkretnej postaci rozkładu pozwala na efektywne wyliczenie potrzebnych zależności.

Ze względu na dalsze potrzeby przedstawia się następujące lematy:

Lemat 3.1. Niech $X_0, X_1, \ldots, X_L$ będą dowolnymi, niezależnymi zmiennymi losowymi o wartościach rzeczywistych. Wówczas funkcje losowe $X_0-X_1, X_0-X_2, \ldots, X_0-X_L$ są również niezależne.

Dowód. Wystarczy rozpatrzeć następujące prawdopodobieństwa:

$$\mathrm{pr}\{X_0-X_1 < c_1, X_0-X_2 < c_2, \ldots, X_0-X_L < c_L\} =$$

$$= \mathrm{pr}\{X_1 > X_0-c_1, X_2 > X_0-c_2, \ldots, X_L > X_0-c_L\},$$

gdzie $c_1, c_2, \ldots, c_L$ dą dowolnymi liczbami rzeczywistymi. Dla każdej wartości x_0 jaką przyjmuje X_0 zachodzi równość:

$$\mathrm{pr}\{X_1 > x_0-c_1, X_2 > x_0-c_2, \ldots, X_L > x_0-c_L\} =$$

$$= \mathrm{pr}\{X_1 > x_0-c_1\}\ \mathrm{pr}\{X_2 > x_0-c_2\} \ldots \mathrm{pr}\{X_L > x_0-c_L\}$$

na mocy niezależności $X_o, X_1, \ldots, X_L$. A stąd

$$\mathrm{pr}\{X_o-X_1 < c_1,\ X_o-X_2 < c_2, \ldots, X_o-X_L < c_L\} =$$

$$= \mathrm{pr}\{X_o-X_1 < c_1\}\ \mathrm{pr}\{X_o-X_2 < c_2\} \ldots \mathrm{pr}\{X_o-X_L < c_L\},$$

co kończy dowód. □

Lemat 3.2. Niech $X_1, X_2, \ldots, X_L$ $(1 \leqslant L \leqslant m)$ będą niezależnymi zmiennymi losowymi o rozkładach (3.1). Wówczas wartość oczekiwana funkcji losowej

$$Y = \min(X_1, X_2, \ldots, X_L) \tag{3.2}$$

ma wartość

$$EY = \left(\sum_{i=1}^{L} \lambda_i\right)^{-1}. \tag{3.3}$$

Dowód. Dystrybuanta F funkcji losowej Y ma postać

$$F(t) = \mathrm{pr}\{\min(X_1,X_2,\ldots,X_L) < t\} = 1 - \mathrm{pr}\{\min(X_1,X_2,\ldots,X_L) \geqslant t\} =$$
$$= 1 - \prod_{i=1}^{L} \mathrm{pr}\{X_i \geqslant t\} = 1 - \prod_{i=1}^{L} (1 - F_i(t)).$$

Stąd

$$EY = \int_0^\infty t\, dF(t) = -\int_0^\infty t \left(\prod_{i=1}^{L} (1 - F_i(t))\right)' dt =$$
$$= \int_0^\infty t \sum_{i=1}^{L} \lambda_i \exp\left(-\sum_{j=1}^{L} \lambda_j t\right) dt =$$
$$= \sum_{i=1}^{L} \lambda_i \int_0^\infty t \exp\left(-\sum_{j=1}^{L} \lambda_j t\right) dt =$$
$$= \sum_{i=1}^{L} \lambda_i \left(\sum_{j=1}^{L} \lambda_j\right)^{-2} = \left(\sum_{i=1}^{L} \lambda_i\right)^{-1}.$$ □

Lemat 3.3. Niech $X_o, X_1, \ldots, X_L$ $(1 \leqslant L \leqslant m)$ będą niezależnymi zmiennymi losowymi o rozkładach (3.1). Wówczas

$$\mathrm{pr}\{X_o < \min(X_1,X_2,\ldots,X_L)\} = \lambda_o^L \prod_{i=1}^{L} (\lambda_o + \lambda_i)^{-1} \tag{3.4}$$

Dowód. Korzystając z lematu 3.1 mamy

$$\mathrm{pr}\{X_0 < \min(X_1, X_2, \ldots, X_L)\} = \prod_{i=1}^{L} \mathrm{pr}\{X_0 - X_i < 0\}.$$

Dalej dla $i = 1,2,\ldots, L$ zachodzą równości

$$\mathrm{pr}\{X_0 - X_i < 0\} = \int_0^\infty \lambda_0 \exp(-\lambda_0 x) \int_x^\infty \lambda_i \exp(-\lambda_i y)\, dy\, dx =$$

$$= \lambda_0 \int_0^\infty \exp(-\lambda_0 x - \lambda_i x)\, dx = \lambda_0 / (\lambda_0 + \lambda_i),$$

skąd wynika (3.4). □

Korzystając z podstawowej własności rozkładu wykładniczego, że rozkład warunkowy ma taką samą postać jak rozkład bezwarunkowy, nasuwa się możliwość uproszczenia postaci stanu wykonania zadania $\mathcal{Z}$. Jeżeli rozkłady czasów wykonania zadań elementarnych mają charakter wykładniczy, to matryca stanu s_k (2.5) redukuje się do wektora

$$s_k = (a_0^k, a_1^k, \ldots, a_{m+1}^k), \tag{3.5}$$

bez powodowania utraty informacji, potrzebnej do podjęcia odpowiednich decyzji.

Zauważmy też, że ponieważ prawdopodobieństwo jednoczesnego zakończenia wykonywania zadań elementarnych o dowolnym rozkładzie ciągłym jest równe zeru, więc q-następowanie stanów określone definicją 2.7 uprasza się do 1-następowania.

Definicja 3.1. Wprowadza się teraz rozbicie zbioru dopuszczalnych stanów wykonania S^* na następujące warstwy stanów:

$$\begin{aligned} SO_i &= \{s_k \in S_{odp} : \sum_{l=1}^{M} |\chi_l(s_k)| = i\} \\ &(i = 0,1,2,\ldots), \\ SD_j &= \{s_k \in S_{dec} : s_l \overset{d}{\prec} s_k,\ s_l \in SO_{j-1}\} \\ &(j = 1,2,3,\ldots). \end{aligned} \tag{3.6}$$

Pierwsze warstwy nazywa się warstwami odpowiedzi, drugie - warstwami decyzji. □

Bezpośrednio z (3.6) wynikają proste własności warstw stanów:

(i) Dla każdych, różnych od siebie s_k, s_l należących do tej samej warstwy nie zachodzi $s_k \prec s_l$, ani $s_l \prec s_k$.

(ii) Warstwy decyzji (odpowiedzi) są rozłączne.

(iii) $SO_0 = \{s_0\}$, gdzie s_0 jest postaci (2.6) oraz $SO_i = SD_i = \Theta$ dla $i > m+1$.

(iv) Dla dowolnego $s_k \in SO_i$ $(i = 0,1,2,\ldots)$

$$\{s_l \in S_{dec} : s_k \overset{d}{\prec} s_l\} \subset SD_{i+1},$$

oraz dla dowolnego $s_k \in SD_j$ $(j = 1,2,3,\ldots)$

$$\{s_l \in S_{odp} : s_k \overset{1}{\prec} s_l\} \subset SO_j.$$

Warstwę odpowiedzi SO_r mającą tę własność, że $SO_r \neq \Theta$ i $SO_{r+1} = \Theta$ będziemy nazywać najniższą warstwą odpowiedzi.

3.2. Optymalne wykonanie bez przerwań w jednofazowym systemie obsługi

Poniżej podaje się algorytm wyznaczania optymalnej funkcji rozdziału $D \in \mathfrak{D}$. W ogólnym zarysie, polega ona na przypisaniu każdemu stanowi $s_k \in S^*$ pewnego wskaźnika liczbowego $w(s_k)$ zwanego wagą stanu s_k. Wyznaczenie wag odbywa się w warstwach ustawionych w następującym porządku

$$SO_r, SD_r, SO_{r-1}, \ldots, SD_1, SO_0$$

gdzie SO_r jest najniższą warstwą odpowiedzi. Wprowadza się jeszcze oznaczenie $ind(i,j)$, określające wskaźnik zadania elementarnego, którego wykonanie zostało zakończone przy przejściu ze stanu $s_i \in S_{dec}$ do stanu $s_j \in S_{odp}$. Przy założeniu jednofazowego systemu wykonawczego $W = W^1 = \{w_1^1, w_2^1, \ldots, w_n^1\}$ $(m > n)$ algorytm przyjmie postać.

Algorytm 3.1. K r o k 0. Na podstawie matrycy incydencji Q oraz rozkładu P zadania $\mathfrak{Z}$ wyznacza się zbiór SK stanów końcowych i przypisuje się im zerowe wagi. Każdy ze stanów $s_k \in SK$ kwalifikuje się do odpowiedniej warstwy odpowiedzi SO_l (3.6), przez obliczenie wskaźnika

$$l = \sum_{i=1}^{M} |\chi_i(s_k)|. \tag{3.7}$$

Przez SK_l będziemy oznaczać podzbiór zbioru SK tych stanów końcowych, które posiadają wartość l wskaźnika (3.7). Zbiór SK_r o największej wartości wskaźnika (3.7) wyznacza najniższą warstwę odpowiedzi, tzn.

$$SO_r = SK_r \, . \tag{3.8}$$

K r o k i 1, 3, ..., 2r-1. Na podstawie ustalonej warstwy odpowiedzi SO_l, gdzie l w kolejnych, rozważanych tu krokach przyjmuje wartości r, r-1, ..., 1, wyznacza się warstwą decyzji SD_l w sposób następujący. Dla każdego $s_k \in SO_l$ należy:

a) utworzyć rodzinę $R_1(\chi^-(s_k))$ jednoelementowych podzbiorów zbioru $\chi^-(s_k)$;

b) utworzyć zbiór

$$\{s_j \in S_{dec} : s_j \overset{1}{\prec} s_k\} =$$

$$= \{s_j \in S : \chi(s_k \otimes s_j) \in R_1(\chi^-(s_k)) \wedge \chi(s_k \otimes s_j) \subset \chi_0(s_j)\}. \tag{3.9}$$

Stąd

$$SD_l = \bigcup_{s_k \in SO_l} \{s_j \in S_{dec} : s_j \overset{1}{\prec} s_k\} \, . \tag{3.10}$$

Każdemu ze stanów $s_j \in SD_l$ przypisuje się wagę zgodnie ze wzorem

$$w(s_j) = \sum_{s_j \overset{1}{\prec} s_k} pr\{s_j \overset{1}{\prec} s_k\} (E\, t(s_j, s_k) + w(s_k)) \, , \tag{3.11}$$

gdzie $t(s_j, s_k)$ jest funkcją losową (2.13), zaś $w(s_k)$ - wagą stanu $s_k \in SO_l$, wyznaczoną w poprzednim kroku.

K r o k i 2,4,..., 2r. Na podstawie ustalonej warstwy decyzji SD_l, gdzie l w kolejnych, rozważanych tu krokach przyjmuje wartości r, r-1, ..., 1, wyznacza się warstwą odpowiedzi SO_{l-1} w sposób następujący. Dla każdego $s_k \in SD_l$ należy:

a) utworzyć rodzinę $R_2(\chi_0(s_k))$ wszystkich niepustych podzbiorów zbioru $\chi_0(s_k)$; jeżeli $|\chi_0(s_k)| < |W|$, to do $R_2(\chi_0(s_k))$ dołącza się zbiór pusty;

b) utworzyć zbiór

$$\{s_j \in S_{odp} : s_j \overset{d}{\prec} s_k\} =$$

$$= \{s_j \in S : \chi(s_j \otimes s_k) \in R_2(\chi_0(s_k)) \wedge \chi(s_j \otimes s_k) \subset \chi_{-1}(s_k)\}. \tag{3.12}$$

Stąd

$$SO_{l-1} = SK_{l-1} \cup \bigcup_{s_k \in SD} \{s_j \in S_{odp} : s_j \overset{d}{\prec} s_k\}. \tag{3.13}$$

Każdemu ze stanów $s_j \in SO_{l-1}$, różnym od stanów końcowych przypisuje się wagi zgodnie ze wzorem

$$w(s_j) = \min_{s_j \overset{d}{\prec} s_k} w(s_k). \tag{3.14}$$

Stan $s_{j^*} \in SD_l$ minimalizujący wyrażenie (3.14) nazywa się stanem optymalnej decyzji podjętej w stanie $s_j \in SO_{l-1}$. Wyznacza się zredukowaną warstwę decyzji

$$SD_l^* = \{ s_k \in SD_l : \exists_{s_j \in SO_{l-1}} \; s_k = s_{j^*} \}. \tag{3.15}$$

$\square$

Programową realizację podanego algorytmu, napisaną w języku ALGOL-60 dla m.c. ODRA-1204 zawiera dodatek. Algorytm wyznacza optymalną, ze względu na ustalone kryterium, kolejność wykonywania zadań elementarnych zadania $\mathfrak{Z}$. Zachodzi bowiem twierdzenie:

Twierdzenie 3.1. Funkcja rozdziału $D \in \mathfrak{D}$ jest optymalna (w sensie najmniejszej wartości oczekiwanej czasu wykonania zadania $\mathfrak{Z}$) wtedy i tylko wtedy, gdy przyjmuje wartości ze zbioru

$$S_{dec}^* = \bigcup_{l=1}^{r} SD_l^*, \tag{3.16}$$

zwanego zredukowanym zbiorem stanów decyzji, w taki sposób, że

$$D(s_j) = s_{j^*} \quad \text{dla} \quad s_j \in S_{odp}. \tag{3.17}$$

Dowód. Zauważymy najpierw, że wagi stanów $s_k \in S^*$ wyznaczane przez algorytm 3.1 są wartościami oczekiwanymi czasów wykonania podzbiorów $\mathcal{X}_{-1}(s_k)$ zadań elementarnych, przy założeniu, że ich wykonywanie będzie przebiegać zgodnie z określoną regułą - wzór (3.17).

Każda funkcja rozdziału $D \in \mathfrak{D}$ spełniająca (3.17) określa jednoznacznie wartość $w(s_k)$ dla $s_k \in S_{odp}$. Tym samym określona jest jednoznacznie wartość oczekiwana czasu wykonania zadania $\mathfrak{Z}$ - $w(s_o)$, gdzie s_o jest początkowym stanem wykonania (2.6). Zatem warunek (3.17) jest wystarczający.

Konieczność warunku (3.17) wynika z następującego wnioskowania. Jeżeli założymy, że znana jest optymalna kolejność wykonania zadania

$\mathcal{Z}$, znajdującego się w stanie $s_k \in SO$ oraz, że znane są wszystkie możliwe przejścia do tego stanu ze stanów SO_{l-1} poprzez stany SD_l, to każda funkcja $D_1 \in \mathcal{D}$ nie spełniająca (3.17) daje wartość wagi $w(s_j)$, dla $s_j \in SO_{l-1}$ nie mniejszą niż funkcja D. Stwierdzenie to wynika ze wzorów (3.11) i (3.14). Przeglądając indukcyjnie warstwy odpowiedzi SO_l w kolejności $l = r, r-1, \ldots, 0$, stwierdzamy konieczność warunku (3.14) a zatem i (3.17). $\square$

Korzystając z lematów przedstawionych w p. 3.1 można podać jawne wyrażenia na elementy występujące we wzorze (3.11). Mianowicie

$$pr\{s_j \overset{1}{\prec} s_k\} = \lambda_{ind(j,k)}^{|A_{jk}|}\, p_{ind(j,k)}\, a^j_{ind(j,k)} \prod_{l \in A_{jk}} (\lambda_{ind(j,k)} + \lambda_l)^{-1}, \tag{3.18}$$

gdzie $A_{jk} = \{ l : z_l \in \mathcal{X}_0(s_j)\} \setminus \{ind(j,k)\}$; ponadto

$$E\, t(s_j, s_k) = \Big(\sum_{z_l \in \mathcal{X}_0(s_j)} \lambda_l \Big)^{-1}. \tag{3.19}$$

3.3. Optymalne wykonanie bez przerwań w wielofazowym systemie obsługi

Algorytm przedstawiony w p. 3.2 w naturalny sposób, po uwzględnieniu małych zmian, rozszerza się dla określenia optymalnej funkcji rozdziału w wielofazowym systemie obsługi. Stan s_k wykonania zadania $\mathcal{Z}$ ma postać (3.5) z tym, że składowe wektorowe stanu są uporządkowane w taki sposób, że $a_1^k, \ldots, a_{m_1}^k$ odnoszą się do zadań elementarnych wykonywanych na stanowiskach W^1, $a^k_{m_1+1}, \ldots a^k_{m_1+m_2}$ - na stanowiskach W^2 itd. Oczywiście $m_1 + m_2 + \ldots + m_K = m$ oraz, aby rozdział nie był trywialny $m_i > n_i$ $(i = 1,2,\ldots, K)$.

Algorytm podaje się przez porównanie różnic z algorytmem 3.1.

<u>Algorytm 3.2.</u> K r o k 0. Bez zmian.

K r o k i 1, 3, ..., 2r-1. Bez zmian.

K r o k i 2, 4, ..., 2r. W punkcie a) zamiast $R_2(\mathcal{X}_0(s_k))$ tworzy ciąg rodzin $R_2(\mathcal{X}_{0,i}(s_k))$ niepustych podzbiorów zbioru $\mathcal{X}_{0,i}(s_k)$ dla $i = 1,2,\ldots, K$. Do każdej z tych rodzin dołącza się element pusty, jeżeli spełniony jest warunek

$$|\mathcal{X}_0(s_k)| = \sum_{i=1}^{K} |\mathcal{X}_{0,i}(s_k)| < \sum_{i=1}^{K} |W^i|.$$

W punkcie b) tworzy się zbiór

$$\{s_l \in S_{odp} : s_l \overset{d}{\prec} s_k\} =$$

$$= \{s_l \in S : \mathcal{X}_i(s_l \otimes s_k) \in R_2(\mathcal{X}_{o,i}(s_k)) \wedge \qquad (3.12a)$$

$$\wedge\ \mathcal{X}_i(s_l \otimes s_k) \subset \mathcal{X}_{-1,i}(s_k)\ ,\ i = 1, 2, \ldots, K\}.$$

Dalsze postępowanie pozostaje bez zmian. □

Jak widać pomiędzy algorytmami 3.1 i 3.2 zachodzą jedynie różnice ilościowe. Zatem funkcja rozdziału zadań elementarnych, wyznaczona przez algorytm 3.2 jest optymalna w sensie rozważanego kryterium. Stwierdzenie to jest analogonem twierdzenia 3.1.

3.4. Optymalne wykonanie z przerwaniami w wielofazowym systemie obsługi

Rozważany dotychczas sposób wykonywania pojedynczych zadań elementarnych polegał na ich przesłaniu, w określonym momencie, na stanowisko obsługi odpowiedniego typu i pozostawieniu ich tam, aż do momentu zakończenia ich wykonywania. Obecnie dopuszcza się obsługę z przerwami, tzn. wykonywanie, na określonym stanowisku zadania elementarnego może być zatrzymane w arbitralnie określonym momencie czasu, a wykonywanie pozostałej jego części może być wznowione po upływie dowolnie określonego odcinka czasu.

Ogólny model stanu wykonania zadania $\mathcal{Z}$ - definicja 2.3, jest przewidywany do opisu wykonywania zadania bez przerwań. Ze względu na szczególne własności wykładniczego rozkładu prawdopodobieństwa wystarcza on jednak dla opisu wykonywania z przerwami rozważanej w bieżącym rozdziale klasy zadań.

W ogólnym przypadku wykonywania zadań z przerwami, momenty przerwań określone są momentami zakończenia wykonywania zadań elementarnych oraz dowolnie przyjętym ciągiem momentów np. po upływie stałego odcinka czasu. Zauważmy, że w przypadku wykładniczego rozkładu prawdopodobieństwa czasu wykonywania zadań elementarnych, jedynymi sensownymi momentami przerwań są momenty zakończenia wykonywania zadań elementarnych. Istotnie, bowiem, stan wykonywania zadania $\mathcal{Z}$ w dowolnie,inaczej wprowadzonym momencie przerwania jest identyczny ze stanem wykonania - stanem decyzji, podjętym po ostatnim przerwaniu polegającym na zakończeniu wykonywania jednego z zadań elementarnych. Własność ta wynika z "bezpamięciowego" charakteru wykładniczego rozkładu prawdopodobieństwa. Pokazaliśmy więc lemat.

Lemat 3.4. Jeżeli zadania elementarne zadania $\mathcal{Z}$ mają wykładnicze rozkłady prawdopodobieństwa czasów wykonywania, to zbiór S stanów wykonania zadania $\mathcal{Z}$ z przerwaniami jest identyczny ze zbiorem stanów wykonywania zadania $\mathcal{Z}$ bez przerwań. □

Oczywisty jest fakt, że wartość oczekiwana czasu optymalnego wykonania zadania z przerwaniami nie jest większa od wartości oczekiwanej czasu optymalnego wykonania zadania bez przerwań. Opis wykonania zadania $\mathcal{Z}$ z przerwaniami różni się nieco od opisu wykonania zadania bez przerwań, który był określony przez trajektorię

$$s_0 \overset{d}{\prec} s_1 \overset{1}{\prec} s_2 \overset{d}{\prec} \ldots \overset{d}{\prec} s_{2r-1} \overset{1}{\prec} s_{2r}, \qquad (3.20)$$

$$r = |Z_0|, \qquad r \leqslant m+2.$$

Powodem różnicy jest fakt, że teraz d-przejście ze stanu $s_k \in S_{odp}$ dokonuje się przez wybranie do wykonania pewnego podzbioru zadań elementarnych ze zbioru $\mathcal{X}_0(s_k) \cup \mathcal{X}^+(s_k)$, a nie jak poprzednio przez dołączenie do zbioru $\mathcal{X}_0(s_k)$ pewnego podzbioru zadań elementarnych ze zbioru $\mathcal{X}^+(s_k)$. Aby wykorzystać wprowadzone poprzednio definicje d-poprzedzania i q-poprzedzania stanów będziemy, przed wyborem do wykonania podzbioru zadań elementarnych, dokonywać następującej korekcji stanów $s_k \in S_{odp}$.

Definicja 3.2. Dla stanu $s_k \in S_{odp}$, skorygowanym stanem odpowiedzi będziemy nazywać stan $s_l \in S$ taki, że spełnione są warunki:

(i) $\mathcal{X}(s_k \otimes s_l) = \mathcal{X}_0(s_k)$,

(ii) $\mathcal{X}_{-1}(s\ ) = \mathcal{X}_{-1}(s_k) \cup \mathcal{X}_0(s_k)$. □

Stany s_k oraz s_l są sobie równoważne ze względu na informację potrzebną do podjęcia decyzji o d-przejściu (z przerwaniami) do nowego stanu - lemat 3.4. W związku z nową definicją, przejście od nieskorygowanego stanu odpowiedzi $s_k \in S_{odp}$, poprzez stan skorygowany s_l, do stanu decyzji s_j w wykonaniu z przerwaniami, będziemy oznaczać przez $s_k \overset{pd}{\prec} s_j$, a zbiór stanów skorygowanych przez $S_{odp\ skor}$.

Lemat 3.5. Jeżeli funkcja rozdziału $D : S_{odp\ skor} \longrightarrow S_{dec}$ jest optymalna, to d-przejścia spełniają wprowadzone wyżej ograniczenie.

Dowód. Wartość oczekiwana czasu wykonania zadania $\mathcal{Z}$ z przerwaniami po pewnej trajektorii wyraża się jako suma wartości oczekiwanych funkcji losowych postaci (2.13). Ponieważ zaś

$$E \min_{z_j \in \mathcal{X}'} (t_j) \geqslant E \min_{z_j \in \mathcal{X}''} (t_j)$$

dla $\mathcal{X}' \subset \mathcal{X}'' \subset Z$, więc prawa strona nierówności może osiągnąć wartość najmniejszą tylko dla podzbioru $\mathcal{X}''$ posiadającego największą liczbę zadań elementarnych spośród zadań elementarnych gotowych do wykonania, co pokrywa się z sensem wprowadzonego ograniczenia. $\square$

Algorytm znajdowania optymalnej funkcji rozdziału, przez porównanie z algorytmem 3.1, przedstawia się następująco.

Algorytm 3.3. K r o k 0. Bez zmian.

K r o k i 1, 3, ..., 2r-1. Bez zmian.

K r o k i 2, 4, ..., 2r. Na podstawie ustalonej warstwy stanów decyzji SD_l, gdzie l w rozważanych tu krokach przyjmuje kolejno wartości r, r-1, ..., 1, wyznacza się warstwę stanów odpowiedzi SO_{l-1} w sposób następujący. Dla każdego $s_k \in SD_l$:

a) tworzymy skorygowany stan odpowiedzi $s_j \in S$, d-poprzedzający s_k, zgodnie z formułą

$$\begin{aligned} &\mathcal{X}(s_j \otimes s_k) = \mathcal{X}_0(s_k), \\ &\mathcal{X}_{-1}(s_j) = \mathcal{X}_{-1}(s_k) \cup \mathcal{X}_0(s_k) ; \end{aligned} \tag{3.21}$$

b) stan s_j wyznacza zbiór stanów SO_{l-1}, które mogły być skorygowane do jego postaci; tworzymy w tym celu rodzinę $R_3(\mathcal{X}^+(s_j))$ podzbiorów zbioru $\mathcal{X}^+(s_j)$, o mocy

$$\min (|\mathcal{X}^+(s_j)|, n) - 1,$$

gdzie

$$n = n_1 + n_2 + \ldots + n_K ; \tag{3.22}$$

c) tworzymy zbiór

$$\{s_i \in S_{odp} : s_i \overset{pd}{\prec} s_k\} =$$

$$= \{s_i \in S : \mathcal{X}_0(s_i) = \mathcal{X}(s_i \otimes s_j) \in R_3(\mathcal{X}^+(s_j)) \ . \tag{3.23}$$

Stąd

$$SO_{l-1} = SK_{l-1} \cup \bigcup_{s_k \in SD_l} \{s_i \in S_{odp} : s_i \overset{pd}{\prec} s_k\} \tag{3.24}$$

Każdemu ze stanów $s_i \in SO_{l-1}$, różnemu od stanu końcowego, przypisuje się wagę zgodnie ze wzorem

$$w(s_i) = \min_{s_i \xrightarrow{pd} s_k} w(s_k). \qquad (3.25)$$

Stan $s_i^* \subset SD_l$ nazywa się stanem decyzji optymalnej, podjętej w stanie $s_i \in SO_{l-1}$, jeżeli minimalizuje wartość wyrażenia (3.25). Wyznacza się zredukowaną warstwę decyzji - wzór (3.15). $\square$

Uwzględniając twierdzenie 3.1 oraz lemat 3.5 można sformułować twierdzenie.

Twierdzenie 3.2. Funkcja rozdziału $D \in \mathfrak{D}_1$, $\mathfrak{D}_1$ - zbiór funkcji rozdziału z przerwaniami, jest optymalna (w sensie najmniejszej wartości oczekiwanej czasu wykonania zadania $\mathfrak{T}$) wtedy i tylko wtedy, gdy przyjmuje wartości ze zbioru (3.16), gdzie SD_l^* są wyznaczone przez algorytm 3.3, w taki sposób, że spełniony jest warunek (3.17).

$\square$

3.5. Oszacowanie wymiaru zagadnienia

Z realizacją każdego algorytmu na maszynie cyfrowej wiążą się problemy oszacowania czasu obliczeń oraz wielkości zajętego obszaru pamięci. Czas obliczeń - t_{obl} oraz pojemność wymaganej pamięci - v_{pam} określa się w postaci funkcji, których argumentem - x, jest wymiar problemu rozwiązywanego przez algorytm. Więc

$$t_{obl} = t_{obl}(x), \qquad v_{pam} = v_{pam}(x). \qquad (3.26)$$

Celem obliczeń algorytmów 3.1, 3.2, 3.3 jest określenie funkcji rozdziału D tzn. zbioru argumentów - S_{odp} oraz zbioru wartości S_{dec}. Naturalnym wydaje się przyjęcie jako wymiaru zagadnienia x - liczbę sytuacji, w których należy podejmować decyzje, tj. moc zbioru S_{odp}. Więc

$$x = |S_{odp}| . \qquad (3.27)$$

Ponieważ $S_{odp} \subset S^*$, zaś $|S^*| \leqslant (M+2)^m$, można więc przyjąć, że

$$x = c_o(M+2)^m, \quad 0 < c_o \leqslant 1. \qquad (3.28)$$

Przy takim założeniu analiza algorytmu 3.1 daje oszacowania

$$t_{obl} = c_1x + (c_2m + c_3 2^m)x \approx c_4x, \qquad (3.29)$$
$$v_{pam} = c_5 + c_6x,$$

gdzie $c_1, c_2, \ldots, c_6$ są pewnymi stałymi proporcjonalnymi odpowiednio: c_1 - do ilości operacji w kroku 0, c_2 - w krokach 1, 3, ..., 2r-1 c_3 - w krokach 2, 4, ..., 2r, c_5 - do stałego obszaru pamięci, c_6 - do roboczego obszaru pamięci.

Zgadzając się na zaproponowane określenie wymiaru zagadnienia, można stwierdzić, że algorytm 3.1 jest efektywny w sensie Edmondsa [36], tzn. funkcje t_{obl}, v_{pam} dają się przedstawić w postaci pewnych wielomianów od wymiaru zagadnienia x.

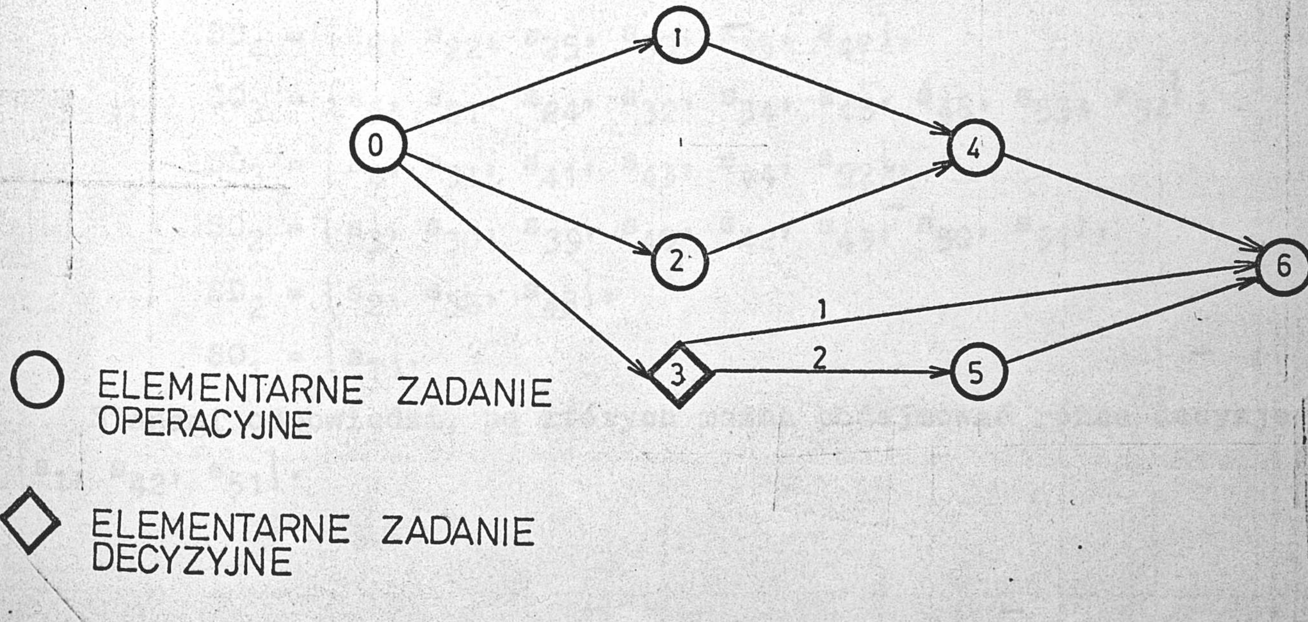
Podobne uwagi i oszacowania odnoszą się do algorytmów 3.2, 3.3. Niestety, wymiar problemów jest wykładniczą funkcją liczby zadań elementarnych. Traktując zatem m jako wymiar zagadnienia (pomijając inne parametry) algorytmy przestają być efektywne.

3.6. Przykład

Dla ilustracji omawianych zagadnień podamy postać przestrzeni stanów wykonania dla prostego przykładu zadania $\mathfrak{Z}$. Zakładamy, że rozpatrywane jest wykonywanie zadania $\mathfrak{Z}$ w jednofazowym systemie obsługi złożonym z dwóch stanowisk obsługi

$$W = W^1 = \{w_1^1, w_2^1\}.$$

Graf G zadania $\mathfrak{Z}$ przedstawia rys. 3.1.



Rys. 3.1. Graf zadania $\mathfrak{Z}$

Matryca incydencji Q rozpatrywanego grafu ma postać:

$$Q = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

Zbiór S^* dopuszczalnych stanów wykonania zadania $\mathcal{Z}$ zawiera tablica 3.1, zaś na rys. 3.2 przedstawiono przestrzeń stanów wykonania zadania $\mathcal{Z}$ (numery stanów są zakodowane zgodnie z tablicą 3.1). Ze względu na formalne tylko znaczenie stanu (-1 -1 -1 -1 -1 -1 -1) został on pominięty. Jak wynika z rys. 3.2 stanami końcowymi są

$$SK = \{ s_{11}, s_{18} \}.$$

Poszczególne warstwy stanów układają się następująco

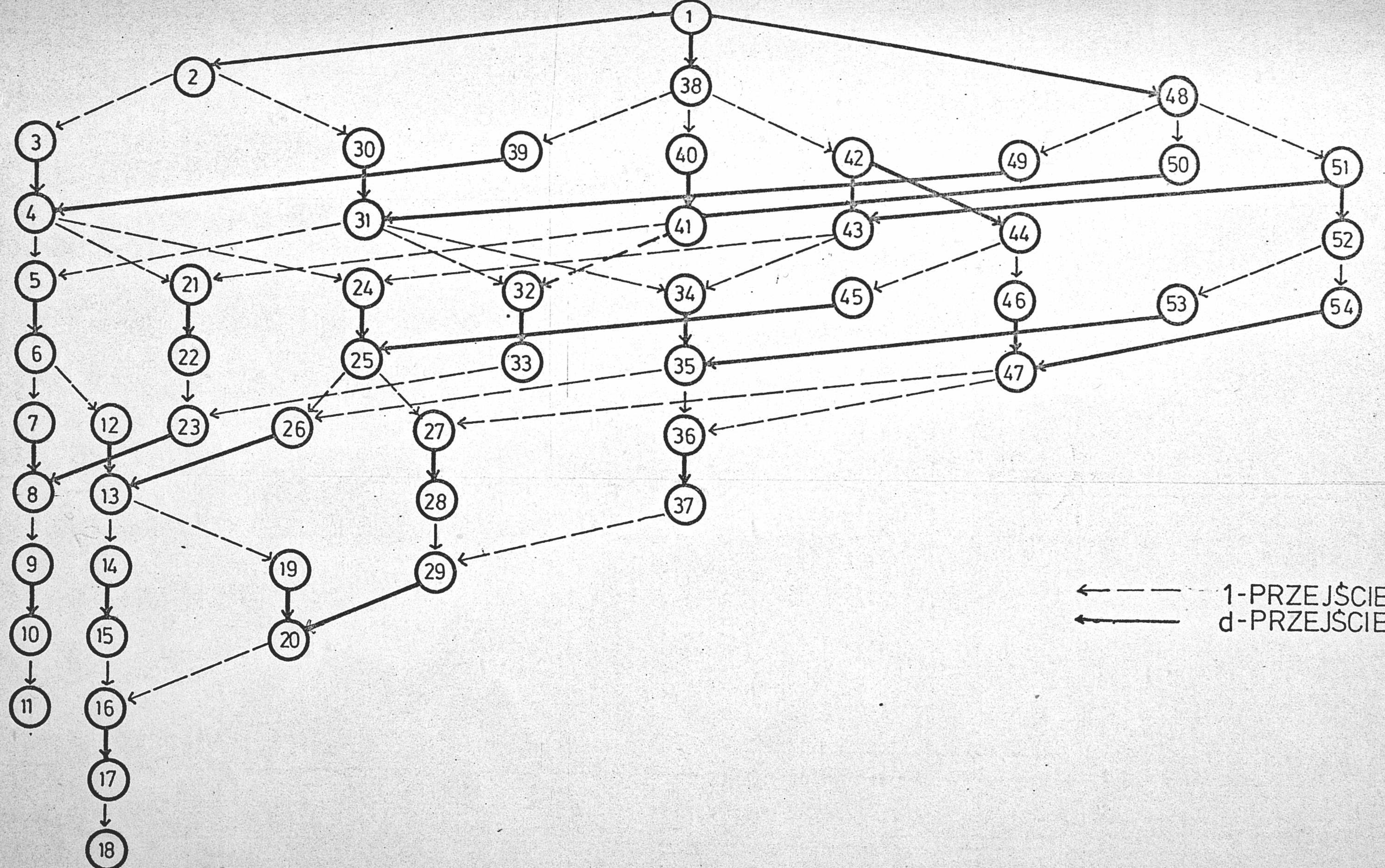
$$SO_7 = \{ s_{18} \},$$
$$SD_7 = \{ s_{17} \},$$
$$SO_6 = \{ s_{11}, s_{16} \},$$
$$SD_6 = \{ s_{10}, s_{15}, s_{20} \},$$
$$SO_5 = \{ s_9, s_{14}, s_{19}, s_{29} \},$$
$$SD_5 = \{ s_8, s_{13}, s_{28}, s_{37} \},$$
$$SO_4 = \{ s_7, s_{12}, s_{23}, s_{26}, s_{27}, s_{36} \},$$
$$SD_4 = \{ s_6, s_{22}, s_{25}, s_{33}, s_{35}, s_{47} \},$$
$$SO_3 = \{ s_5, s_{21}, s_{24}, s_{32}, s_{34}, s_{45}, s_{46}, s_{53}, s_{54} \},$$
$$SD_3 = \{ s_4, s_{31}, s_{41}, s_{43}, s_{44}, s_{52} \},$$
$$SO_2 = \{ s_3, s_{30}, s_{39}, s_{40}, s_{42}, s_{49}, s_{50}, s_{51} \},$$
$$SD_2 = \{ s_2, s_{38}, s_{48} \},$$
$$SO_1 = \{ s_1 \}.$$

Stanami odpowiedzi, po których można podejmować różne decyzje są $\{ s_1, s_{42}, s_{51} \}$.

T a b l i c a 3.1

Zbiór dopuszczalnych stanów wykonania zadania

Nr	Wektor stanu						
1	1	-1	-1	-1	-1	-1	-1
2	1	0	0	-1	-1	-1	-1
3	1	1	0	-1	-1	-1	-1
4	1	1	0	0	-1	-1	-1
5	1	1	1	0	-1	-1	-1
6	1	1	1	0	0	-1	-1
7	1	1	1	1	0	-1	-1
8	1	1	1	1	0	-1	-1
9	1	1	1	1	1	-1	-1
10	1	1	1	1	1	-1	0
11	1	1	1	1	1	-1	1
12	1	1	1	2	0	-1	-1
13	1	1	1	2	0	0	-1
14	1	1	1	2	1	0	-1
15	1	1	1	2	1	0	-1
16	1	1	1	2	1	1	-1
17	1	1	1	2	1	1	0
18	1	1	1	2	1	1	1
19	1	1	1	2	0	1	-1
20	1	1	1	2	0	1	-1
21	1	1	0	1	-1	-1	-1
22	1	1	0	1	-1	-1	-1
23	1	1	1	1	-1	-1	-1
24	1	1	0	2	-1	-1	-1
25	1	1	0	2	-1	0	-1
26	1	1	1	2	-1	0	-1
27	1	1	0	2	-1	1	-1
28	1	1	0	2	-1	1	-1
29	1	1	1	2	-1	1	-1
30	1	0	1	-1	-1	-1	-1
31	1	0	1	0	-1	-1	-1
32	1	0	1	-1	-1	-1	-1
33	1	0	1	1	-1	-1	-1
34	1	0	1	2	-1	-1	-1
35	1	0	1	2	-1	0	-1
36	1	0	1	2	-1	1	-1
37	1	0	1	2	-1	1	-1
38	1	0	-1	0	-1	-1	-1
39	1	1	-1	0	-1	-1	-1
40	1	0	-1	1	-1	-1	-1
41	1	0	0	1	-1	-1	-1
42	1	0	-1	2	-1	-1	-1
43	1	0	0	2	-1	-1	-1
44	1	0	-1	2	-1	0	-1
45	1	1	-1	2	-1	0	-1
46	1	0	-1	2	-1	1	-1
47	1	0	0	2	-1	1	-1
48	1	-1	0	0	-1	-1	-1
49	1	-1	1	0	-1	-1	-1
50	1	-1	0	1	-1	-1	-1
51	1	-1	0	2	-1	-1	-1
52	1	-1	0	2	-1	0	-1
53	1	-1	1	2	-1	0	-1
54	1	-1	0	2	-1	1	-1



RYS. 32. PRZESTRZEŃ STANÓW WYKONANIA ZADANIA

4. ROZDZIAŁ ZADAŃ O DOWOLNYCH DYSKRETNYCH ROZKŁADACH PRAWDOPODOBIEŃSTWA CZASÓW WYKONYWANIA ZADAŃ ELEMENTARNYCH

4.1. Wprowadzenie

W niniejszym rozdziale ograniczamy się do zadań o dowolnych dyskretnych dyskretnych rozkładach prawdopodobieństwa czasów wykonywania zadań elementarnych. Podejście takie jest wystarczająco ogólne, gdyż rozkłady ciągłe można przybliżać rozkładami dyskretnymi, a ponadto pozwala na wprowadzenie jednolitej formy zapisu odpowiednich wzorów. O zmiennych losowych t_j $(j = 1,2,\ldots, m)$ zakładamy, że mogą przyjmować wartości $0, 1, 2, \ldots, N$.

Ponadto, ograniczamy się do rozpatrywania obsługi w jednofazowym systemie wykonawczym. Jest to typowe ograniczenie ilościowe, które, podobnie jak w rozdziale 3, nie wpływa na zasadniczy tok rozważań.

Podobnie jak w p. 3.1 podaje się podstawowe wzory wykorzystywane dla znalezienia optymalnej funkcji rozdziału zadań elementarnych. Oznaczając, jak poprzednio, przez F_j dystrybuantę zmiennej losowej t_j $(j = 1,2,\ldots, m)$, definiuje się

$$F_{ji} = F_j(i)$$

oraz

$$f_{ji} = F_{j,i+1} - F_{ji} \quad (i = 0,1,\ldots, N), \tag{4.1}$$

f_{ji} jest więc prawdopodobieństwem przyjęcia przez zmienną losową t_j wartości i.

Dla warunkowych zmiennych losowych $t_j|l$, $(j = 1,2,\ldots, m,\ l = 0,1,\ldots, N)$, warunkowe dystrybuanty i warunkowe rozkłady prawdopodobieństwa będą miały postać

$$F_{ji}^{l} = (F_{ji} - F_{jl}) / (1 - F_{jl})$$

oraz

$$f_{ji}^{l} = F_{j,i+1}^{l} - F_{ji}^{l} = f_{ji} / (1 - F_{jl}), \tag{4.2}$$

gdzie $i = 0, 1, \ldots, N$.

Lemat 4.1. Jeżeli $X_1, X_2, \ldots, X_L$ są niezależnymi zmiennymi losowymi o rozkładzie (4.1), to wartość oczekiwana zmiennej losowej

$$Y = \min (X_1, X_2, \ldots, X_L) \tag{4.3}$$

ma postać

$$EY = \sum_{i=0}^{N} i \, (F_{min,i+1} - F_{min,i}) \tag{4.4}$$

gdzie

$$F_{min,i} = 1 - \prod_{j=1}^{L} (1 - F_{ji}). \tag{4.5}$$

Dowód. Wzory wynikające bezpośrednio z definicji EY i z pierwszej części dowodu lematu 3.2. □

Wniosek 4.1. Jeżeli zamiast $X_1, X_2, \ldots, X_L$ w lemacie 4.1 rozważymy warunkowe zmienne losowe $X_1 | b_1, X_2 | b_2, \ldots, X_L | b_L$, to wzór (4.5) przekształci się do postaci

$$F_{min,i} = 1 - \prod_{j=1}^{L} (1 - F_{ji}^{b_i}). \tag{4.5a}$$

Lemat 4.2. Jeżeli $X_o, X_1, \ldots, X_L$ jest ciągiem niezależnych zmiennych losowych o rozkładzie (4.1), to

$$\mathrm{pr}\{X_o < \min(X_1, X_2, \ldots, X_L)\} = \prod_{i=1}^{L} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} f_{oj} f_{ik} \tag{4.6}$$

Dowód. Dowód wynika ze wzoru

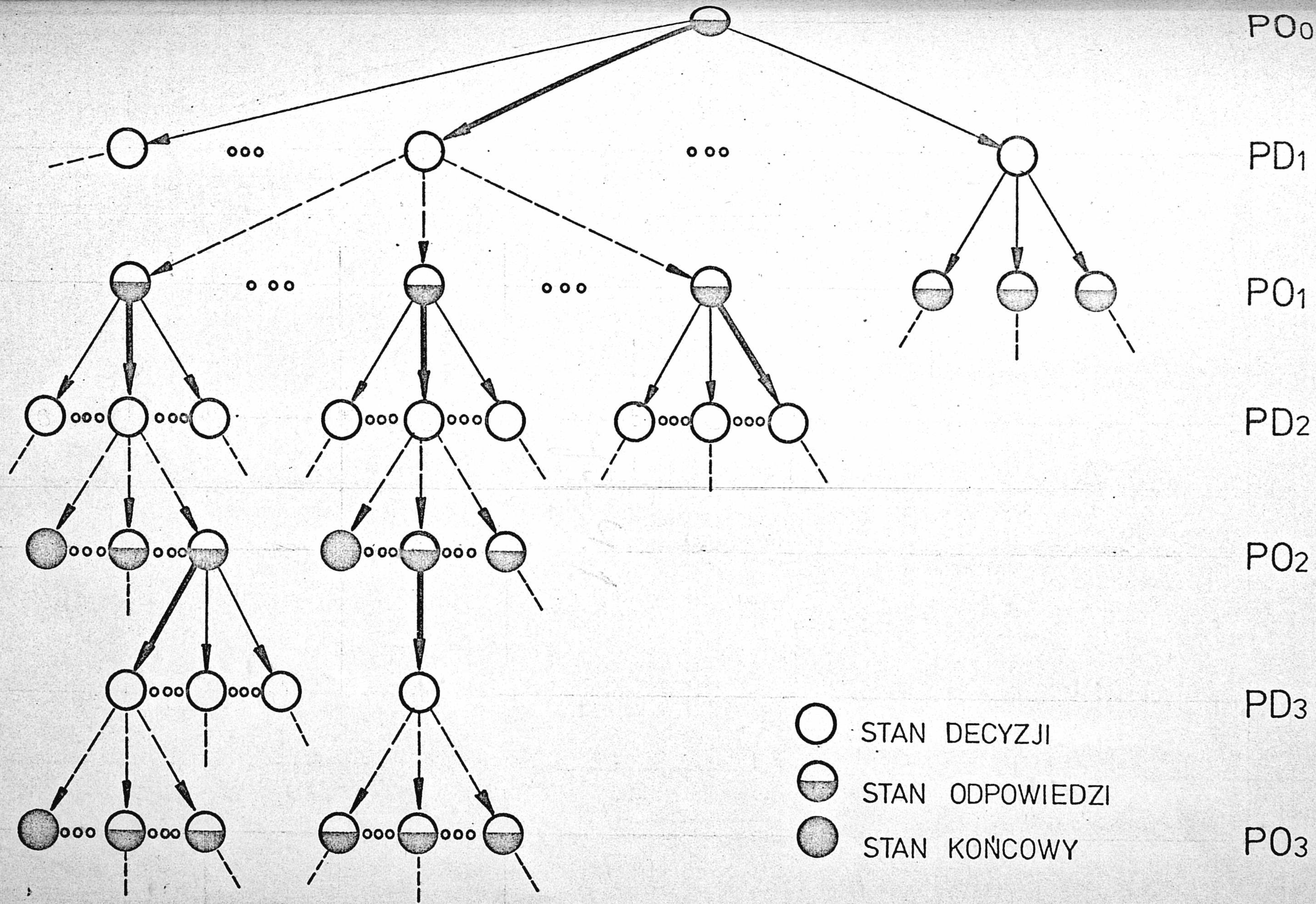
$$\mathrm{pr}\{X_o < X_i\} = \sum_{j<k} f_{oj} f_{ik} = \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} f_{oj} f_{ik}$$

oraz z tego, że

$$\mathrm{pr}\{X_o < \min (X_1, X_2, \ldots, X_L)\} = \prod_{i=1}^{L} \mathrm{pr}\{X_o < X_i\} . \qquad \square$$

Wniosek 4.2. Jeżeli zamiast $X_o, X_1, \ldots, X_L$ rozpatrzymy ciąg warunkowych zmiennych losowych $X_o | b_o, X_1 | b_1, \ldots, X_L | b_L$, to wzór (4.6) przekształci się do postaci

$$\mathrm{pr}\{X_o | b_o < \min (X_1 | b_1, X_2 | b_2, \ldots, X_L | b_L) = \prod_{i=1}^{L} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} f_{oj}^{b_o} f_{ik}^{b_i} . \tag{4.6a}$$



RYS. 4.1. STRUKTURA PODZIAŁU ZBIORU S*

Ze względu na, bardziej niż dotychczas, złożony charakter rozkładów prawdopodobieństwa dokonamy innego podziału zbioru stanów dopuszczalnych S^*. Analogonem warstw stanów będą piętra stanów (decyzji i odpowiedzi) określone następująco.

Definicja 4.1. Niech będzie dana dowolna trajektoria (2.15) wykonania zadania

$$s_0 \overset{d}{\prec} s_1 \overset{q_i}{\prec} s_2 \overset{d}{\prec} \dots \overset{d}{\prec} s_{2k-1} \overset{q_k}{\prec} s_{2k}, \qquad (2.15)$$

$$q_1 + q_2 + \dots + q_k = |Z_0|, \quad \chi^+(s_{2k}) = \Theta,$$

Wówczas piętrem stanów odpowiedzi PO_i $(i = 0,1,\dots, m+2)$ nazywa się podzbiór stanów odpowiedzi, mający tę własność, że zawiera stany s_{2i} ze wszystkich możliwych trajektorri (2.15) wykonania zadania $\mathfrak{Z}$. Podobnie, do piętra stanów decyzji PD_j $(j = 1,2,\dots, m+2)$ zalicza się odpowiednie stany s_{2j-1}. □

Zbiór S^* dopuszczalnych stanów wykonania zadania $\mathfrak{Z}$, uporządkowany według trajektorii wykonania, wyznacza pewne drzewo - rys. 4.1, którego wierzchołkiem początkowym jest stan początkowy, ścieżkami - trajektorie wykonania, biegnące od stanu początkowego do stanu końcowego. Drzewo z rys. 4.1 będziemy oznaczać przez $\mathcal{T}_{\mathfrak{D}}$ i będziemy nazywać drzewem wszystkich wykonań zadania $\mathfrak{Z}$.

Stawianym celem jest wyznaczenie pewnego poddrzewa $\mathcal{T}_{D_0}$ drzewa $\mathcal{T}_{\mathfrak{D}}$, odpowiadającego optymalnej funkcji wykonania D_0 zadania $\mathfrak{Z}$.

4.2. Adaptacja metody podziału i ograniczeń

Metoda podziału i ograniczeń (ang. branch-and-bound method) [9, 51, 57, 64, 66] polega na podziale zbioru dopuszczalnych rozwiązań $\mathfrak{D}$ na pewne rozłączne podzbiory $\mathfrak{D}^1_1, \mathfrak{D}^1_2, \dots, \mathfrak{D}^1_{k1}$, a następnie na wyborze jednego z nich do nowego podziału itd., aż do znalezienia jednoelementowego podzbioru zawierającego rozwiązanie optymalne D_0. Wybór jednego z podzbiorów do dalszego podziału dokonuje się na podstawie oszacowania kresów (dolnych lub górnych) "jakości" rozwiązań zawartych w tych podzbiorach. W naszym przypadku będziemy korzystać z oszacowania kresu dolnego.

Wprowadza się pojęcia: pierwotnego dolnego ograniczenia i wtórnego dolnego ograniczenia stanu decyzji s_k, oznaczanych odpowiednio przez $PDG(s_k)$ i $WDG(s_k)$. Mówiąc "dolne ograniczenie stanu decyzji s_k" (bez przymiotnika pierwotne lub wtórne) mamy na uwadze znane pierwotne lub wtórne dolne ograniczenie stanu s_k; będziemy je oznaczać przez $DG(s_k)$.

Niech G będzie grafem zadania $\mathfrak{Z}$ oraz niech zadanie znajduje się w stanie s_k. Wówczas przez $Ł(G, s_k)$ będziemy oznaczać zbiór dróg w grafie G takich, że prowadzą one przez wierzchołki $\mathcal{X}_{-1}(s_k) \cup \mathcal{X}_o(s_k)$ do wierzchołka końcowego z_{m+1}. Przez $t_ł$ będziemy oznaczać funkcję losową, która jest sumą zmiennych losowych odpowiadających wierzchołkom, przez które prowadzi droga $ł \in Ł(G, s_k)$.

<u>Definicja 4.2.</u> Pierwotnym dowolnym ograniczeniem stanu decyzji $s_k \in S_{dec}$ będziemy nazywać wyrażenie

$$PDG(s_k) = \max \Big(\sum_{z_j \in \mathcal{X}_o(s_k) \cup \mathcal{X}_{-1}(s_k)} E(t_j | b_j^k) \,/\, |W| , \quad \max_{ł \in Ł(G, s_k)} E\, t_ł \Big) . \tag{4.7}$$

$\square$

Podane ograniczenie wyraża fakt, że wartość oczekiwana czasu wykonania podzbioru $\mathcal{X}_o(s_k) \cup \mathcal{X}_{-1}(s_k)$ zbioru Z zadań elementarnych, przy założeniu podjęcia decyzji s_k, będzie nie mniejsze niż $PDG(s_k)$. Sens wartości $PDG(s_k)$ jest intuicyjnie prosty, stanowi on wniosek z lematu McNaughtona [44]. Pierwszy składnik wyrażenia (4.7) określa wartość oczekiwaną czasu wykonania nie wykonywanych i nie zakończonych dotychczas zadań elementarnych, bez przestoju każdego ze stanowisk obsługi; drugi składnik określa wartość oczekiwaną "krytycznej" drogi w częściowo zredukowanym grafie G.

<u>Definicja 4.3.</u> Wtórnym dolnem ograniczeniem stanu decyzji $s_k \in S_{dec}$ będziemy nazywać wyrażenie

$$WDG(s_k) = \sum_{s_k \overset{q}{\prec} s_j} pr\{s_k \overset{q}{\prec} s_j\} \left(E\, t(s_k, s_j) + DG(s_{i_j})\right), \tag{4.8}$$

gdzie $s_{i_j} \in S_{dec}$ jest takim stanem, że

$$s_j \overset{d}{\prec} s_{i_j}$$

oraz (4.9)

$$DG(s_{i_j}) = \min_{s_j \overset{d}{\prec} s_i} DG(s_i) .$$

$\square$

Wtórne dolne ograniczenie stanu $s_k \in S_{dec}$ wyraża dokładniejsze oszacowanie wartości oczekiwanej czasu wykonania zadania $\mathfrak{Z}$, przy podjętej decyzji, na podstawie znajomości dolnych oszacowań decyzji podjętych w kroku następnym.

Lemat 4.3. Wtórne dolne ograniczenie stanu decyzji $s_k \in S_{dec}$ jest dobrze określone, tzn. wartość oczekiwana czasu optymalnego wykonania zadań elementarnych $\mathcal{X}_{-1}(s_k) \cup \mathcal{X}_0(s_k)$ nie jest mniejsza od $WDG(s_k)$.

Dowód. Rozważmy przypadek gdy dolne ograniczenia stanów s_l we wzorze (4.8) są pierwotnymi dolnymi ograniczeniami. Rozkład prawdopodobieństwa $pr\{s_k \overset{q}{\prec} s_j\}$ oraz wartości $E\,t(s_k, s_j)$ są niezależne od stanów s_{lj}. Ponieważ $PDG(s_{lj})$ jest dobrze określone w rozważanym sensie, zatem i $WDG(s_k)$ jest dla tego przypadku dobrze określone. Przedstawione rozumowanie przenosi się indukcyjnie na dowolne sytuacje określone definicją 4.3. $\square$

Dla opisu algorytmu wykorzystamy pojęcie bieżącego poddrzewa wykonania $\mathcal{T}_b$. Przez $\mathcal{T}_b$ będziemy rozumieć poddrzewo drzewa wykonań $\mathcal{T}$ takie, że spełnione są warunki:

(i) wierzchołkiem początkowym $\mathcal{T}_b$ jest stan początkowy wykonania zadania, zaś stanami końcowymi stany decyzji;

(ii) jeżeli do $\mathcal{T}_b$ należy pewien stan $s_k \in S_{dec}$, różny od wierzchołka końcowego $\mathcal{T}_b$, to do $\mathcal{T}_b$ należą także wszystkie stany $s_j \in S_{odp}$ takie, że $s_k \overset{q}{\prec} s_j$;

(iii) jeżeli $s_j \in S_{odp}$ należy do $\mathcal{T}_b$, to istnieje dokładnie jeden $s_l \in S_{dec}$, należący do $\mathcal{T}_b$, taki, że $s_j \overset{d}{\prec} s_l$.

Za początkowe drzewo bieżące $\mathcal{T}_b$ przyjmujemy drzewo złożone z $s_0 \in PO_0$ i dowolnego $s_1 \in PD_1$. Do bieżącego drzewa $\mathcal{T}_b$ będą dołączane nowe stany decyzji zgodnie ze sformułowaną poniżej zasadą.

R e g u ł a d o ł ą c z a n i a. Dla każdego $s_k \in PD_L$, gdzie PD_L jest najniższym piętrem bieżącego drzewa $\mathcal{T}_b$, wyznacza się zbiór

$$B(s_k) = \{ s_j \in S_{odp} : s_k \overset{q}{\prec} s_j \}.$$

Zbiór

$$PO_L = \bigcup_{s_k \in PD_L} B(s_k)$$

wyznacza nowe piętro odpowiedzi. Do każdego $s_j \in PO_L$ dołącza się dowolny $s_l \in PD_{L+1}$ taki, że $s_j \overset{d}{\prec} s_l$.

Algorytm 4.1. K r o k 0. Wyznacza się początkowe bieżące drzewo wykonania $\mathcal{T}_b$.

K r o k 1. Jeżeli bieżące drzewo wykonania $\mathcal{T}_b$ nie jest drzewem kompletnym, tzn. nie wszystkie końcowe stany decyzji są stanami pustymi, to - zgodnie z podaną wyżej regułą - dołącza się do $\mathcal{T}_b$ nowe piętra stanów odpowiedzi PO_L i decyzji PD_{L+1}. L - oznacza tu indeks najniższego piętra bieżącego drzewa $\mathcal{T}_b$. Obliczenie $PDG(s_k)$ dla $s_k \in PD_{L+1}$. Ustalenie $l = L$.

K r o k 2. Obliczenie $WDG(s_k)$ dla $s_k \in PD_l$. Dla każdego stanu odpowiedzi s_i z piętra PO_{l-1} sprawdza się, czy $DG(s_k)$ dla s_j z piętra PD_l, takiego, że w aktualnie utworzonym $\mathcal{T}_b$ $s_i \overset{d}{\prec} s_j$, posiada wartość minimalną (najmniejszą w stosunku do innych możliwych stanów, które mogą d-następować po rozważanym stanie s_i). Jeżeli warunek jest spełniony powtarza się czynności kroku 2 dla $l = l-1$ ($l = L, L-1, \ldots, 1$), w przypadku przeciwnym przechodzi się do kroku 3.

K r o k 3. Jeżeli dla pewnego $s_i \in PO_{l-1}$ d-następujący po nim stan $s_j \in PD_j$ nie minimalizuje wartości dolnego ograniczenia $DG(s_k)$ dla $s_i \overset{d}{\prec} s_k$, wówczas stan s_j i następujący po nim fragment bieżącego drzewa $\mathcal{T}_b$ zostają usunięte z $\mathcal{T}_b$. Stan s_j zostaje zastąpiony przez s_j', który minimalizuje dolne ograniczenie stanów d-następujących po s_i. Odtwarza się usunięty fragment drzewa $\mathcal{T}_b$, traktując go jako pewne poddrzewo drzewa $\mathcal{T}_b$, tzn. stosując do niego zasady kroku 1, 2, 3 algorytmu. Dołącza się odtworzony fragment do $\mathcal{T}_b$ i powraca do czynności kroku 2.

Obliczenia zostają zakończone, gdy uzyska się drzewo $\mathcal{T}_b$, które jest kompletnym drzewem wykonania zadania. □

Twierdzenie 4.1. Algorytm 4.1 zapewnia zbieżność bieżącego drzewa wykonania $\mathcal{T}_b$ zadania $\mathcal{Z}$ do drzewa $\mathcal{T}_{D_o}$, odpowiadającego funkcji $D_o \in \mathcal{D}$ optymalnego rozdziału zadania $\mathcal{Z}$.

Dowód. Załóżmy, że algorytm jest zbieżny do pewnego drzewa $\mathcal{T}_{D_1}$ różnego od $\mathcal{T}_{D_o}$. Drzewa $\mathcal{T}_{D_1}$ i $\mathcal{T}_{D_o}$ mają wspólne stany odpowiedzi np. s_o. Istnieją więc stan odpowiedzi $s_i \in S_{odp}$ oraz s_j, $s_j' \in S_{dec}$ takie, że $s_j \neq s_j'$ i

$$s_i \overset{d}{\prec} s_j, \quad s_j \in \mathcal{T}_{D_o},$$

oraz

$$s_i \overset{d}{\prec} s_j', \quad s_j' \in \mathcal{T}_{D_1}.$$

Wówczas zgodnie z algorytmem

$$DG(s_j') \leqslant DG(s_j) \ ,$$

przy czym $DG(s_j)$, $DG(s_j')$ wyrażają tu (uwzględniając korekcje na podstawie następnych decyzji w następnych piętrach drzew) wartości analizowane czasów wykonania zadań elementarnych $\mathcal{X}_{-1}(s_i) \cup \mathcal{X}_o(s_i)$. Wynika stąd, że wartość oczekiwana czasu wykonania zadania $\mathcal{Z}$ zgodnie z funkcją rozdziału D_1 nie jest większa od wartości oczekiwanej czasu wykonania zadania zgodnie z funkcją rozdziału D_o. Ponieważ D_o jest optymalną funkcją rozdziału, otrzymaliśmy sprzeczność, co dowodzi twierdzenia. $\square$

Dla pełności rozważań należy podać wzory obliczania elementów występujących w wyrażeniu (4.8). Niech $s_k \in S_{dec}$, $s_j \in S_{odp}$ mają postać

$$s_k = \begin{bmatrix} a_o^k & a_1^k & \dots & a_{m+1}^k \\ b_o^k & b_1^k & \dots & b_{m+1}^k \end{bmatrix} ,$$

$$s_j = \begin{bmatrix} a_o^j & a_1^j & \dots & a_{m+1}^j \\ b_o^j & b_1^j & \dots & b_{m+1}^j \end{bmatrix} ,$$

wówczas

$$\begin{aligned} pr\{s_k \overset{q}{\prec} s_j\} = & \Big(\prod_{z_i \in \mathcal{X}_o(s_k) \setminus \mathcal{X}_o(s_j)} p_i \, a_i^j \Big) \, . \\ & \cdot \, pr\Big\{ t_i^{b_i^k} = l, \ z_i \in \mathcal{X}_o(s_k) \setminus \mathcal{X}_o(s_j) \ \wedge \\ & \wedge \ l < \min_{z_i \in \mathcal{X}_o(s_j)} (t_i^{b_i^k}) \ , \\ & l = 1, 2, \dots, N - \max_{z_i \in \mathcal{X}_o(s_j)} (b_i^k) \Big\} \end{aligned} \tag{4.10}$$

Oznaczmy drugi czynnik po prawej stronie wzoru (4.10) przez $pr\{s_k \overset{q}{\prec} s_j\}'$. Korzystając ze wzorów (4.2)-(4.6) mamy

$$pr\{s_k \overset{q}{\prec} s_j\}' = \sum_{l=1}^{N} pr\{s_k \overset{q}{\prec} s_j\}'_l \tag{4.11}$$

gdzie

$$N' = N - \max_{z_i \in \chi_o(s_j)} (b_i^k) \tag{4.12}$$

oraz

$$pr\{s_k \overset{q}{\prec} s_j\}'_l = \Big(\prod_{z_i \in \chi_o(s_k) \setminus \chi_o(s_j)} f_{il}^{b_i^k}\Big) \cdot$$

$$\cdot \Big(\prod_{z_i \in \chi_o(s_j)} (1 - F_{il}^{b_i^k})\Big), \tag{4.13}$$

Dla obliczenia $E\,t(s_k, s_j)$ korzysta się ze wzoru

$$E\,t(s_k, s_j) = \sum_{l=1}^{N'} l\, pr\{s_k \overset{q}{\prec} s_j\}'_l\,. \tag{4.14}$$

Ostatnim, pozostałym problemem obliczeniowym jest określanie maksymalnej wartości $Et_ł$ dla $ł \in Ł(G, s_k)$. Jest to typowy problem numeryczny, którego rozwiązanie w formie programowej zawiera np. biblioteka procedur m.c. ODRA-1204 (1204-VIII-11).

4.3. Oszacowanie wymiaru zagadnienia

Prowadząc rozważania analogicznie jak w p. 3.5, za wymiar x rozpatrywanego wyżej zagadnienia można przyjąć moc wierzchołków drzewa wykonania $\mathcal{T}_D$ wyznaczonego przez algorytm 4.1. Przybliżone oszacowanie tej wielkości można wyrazić przez

$$x = (c_1 2^m)^{c_2 m} \tag{4.15}$$

gdzie m jest liczbą stanowisk obsługi, c_1, c_2 - stałe ($c_1 \leqslant 1$, $c_2 \leqslant 1$).

Czas prowadzenia obliczeń t_{obl} algorytmu trudno oszacować ze względu na możliwości wielokrotnego powtarzania: usuwania i tworzenia nowych fragmentów bieżącego drzewa wykonania $\mathcal{T}_b$. Maksymalny czas obliczeń jest rzędu

$$t_{obl} = c_3 x^2, \tag{4.16}$$

gdzie c_3 jest pewną stałą, tj. proporcjonalny do rzędu mocy wierzchołków zbioru wszystkich drzew wykonań $\mathcal{T}_{\mathfrak{D}}$. Natomiast zapotrzebowanie na pojemność pamięci v_{pam} daje się oszacować

$$v_{pam} = c_4 x + c_5, \qquad (4.17)$$

gdzie c_4, c_5 - pewne stałe ($c_4 \approx 1$), co stanowi istotną zaletę algorytmu.

Już z tak wstępnie przeprowadzonej analizy, ze względu na dużą czasochłonność, nasuwa się konieczność uproszczeń algorytmu,kosztem rezygnacji z rozwiązania absolutnie optymalnego. Do algorytmu 4.1 można wprowadzić uproszczenia polegające na zawężeniu liczby pięter decyzji, dla których ma być prowadzona korekcja wartości dolnych ograniczeń stanów decyzji. Krok 2 algorytmu wymaga przeprowadzenia korekcji wartości dolnych ograniczeń począwszy od PD_L (najniższego piętra decyzji bieżącego drzewa $\mathcal{T}_b$) aż do PD_1 (najwyższego piętra bieżącego drzewa $\mathcal{T}_b$). Uproszczenie może polegać na korekcji od PD_L do PD_{L-p} gdzie p jest liczbą naturalną, określającą liczbę objętych korekcją pięter decyzji. Konsekwencją takiego uproszczenia jest przyjęcie, że decyzje określone w piętrach decyzji PD_1, PD_2, ..., PD_{L-p} są "optymalne".

5. ZAKOŃCZENIE

Zasadniczą część pracy stanowią rozdziały 3, 4, w których przedstawiono oryginalne propozycje rozwiązań problemu, sformułowanego w rozdziale 2. Osiągnięciem autora wydaje się być opis i formalizacja modelu wykonania zadania (przestrzeń stanów wykonania), które pozwoliły na przedstawienie rozwiązania stawianych zagadnień i wskazania na powstałe problemy obliczeniowe. Należy też podkreślić, że wprowadzany model opisu pozwala, na co zwrócono uwagę w p. 2.4, na postawienie pewnego szerszego zagadnienia, mianowicie szukania optymalnej funkcji wykonania zadania, przy uwzględnieniu czasów podejmowania decyzji (czasów d-przejść).

Rozważania zawarte w rozdziale 3 najbardziej odpowiadają pracy [11], powstałej niezależnie (por. pracę autora [31] z rosyjskim opracowaniem [11] zawartym w VT 32/73). W pracy [11] informuje się o rozwiązaniu problemu optymalnego rozdziału (bez przerwań) zadania $\mathfrak{Z}$ o wykładniczych rozkładach prawdopodobieństwa czasów wykonywania zadań elementarnych w jednofazowym niejednorodnym systemie obsługi.

Niniejsza praca rozwiązuje problem rozdziału (z przerwaniami i bez przerwań) powyższej klasy zadań w jednorodnym wielofazowym systemie obsługi. (W przedstawionych algorytmach tkwią możliwości rozszerzenia rozwiązań dla niejednorodnych systemów wykonawczych. Np. w algorytmie 3.1 należałoby rozpatrzeć dodatkowo permutacje elementów ze zbiorów należących do rodziny $R_2(\mathcal{X}_o(s_k))$ - kroki 2,4, ..., 2r). Zbieżność tematów omawianych prac wydaje się świadczyć o właściwym postawieniu problemu i potrzebie jego rozwiązania.

Rozwiązania rozdziału 4 dotyczą rozdziału zadań o dowolnych, dyskretnych rozkładach prawdopodobieństw czasów wykonywania zadań elementarnych. Autorowi nie są znane z literatury inne próby rozwiązań tego zagadnienia. Omawiając zastosowanie metody podziału i ograniczeń można wyciągnąć następujące wnioski:

(i) Ze względu na wymiar zagadnienia można je rozwiązywać tylko w ograniczonym zakresie.

(ii) Istnieje potrzeba poszukiwania rozwiązań przybliżonych. Sposób ten, oparty o przedstawioną metodę, został zarysowany w p. 4.3.

(iii) Istnieje potrzeba budowania innych modeli zadania i sposobu wykonania zadania w systemie obsługi. Przykładem tego typu modeli są rozwinięcia klasycznych modeli kolejkowych, w których rozpatruje się strumienie niezależne przychodzących zadań elementarnych [6, 67], czy też opis zadania pewnymi łańcuchami Markowa [5, 22, 61]. Ocena i porównanie z tymi modelami oraz ewentualne poszukiwanie nowych leżą jednak poza zakresem przedstawionej pracy.

Na zakończenie pragnę wyrazić wdzięczność prof. dr Jerzemu Bromirskiemu za życzliwą opiekę w trakcie przygotowywania pracy, oraz koleżankom i kolegom z Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, których uwagi i sugestie wpłynęły na ostateczny jej kształt.

LITERATURA

[1] A b a d i e J. (ed.), Integer and nonlinear programming. North-Holland Publishing Company, Amsterdam-London 1970.

[2] A n t i p o v V.J., Rešenije častnoj zadačy kalendarnovo planirovalija w: Mojsejev N.N. (red.), Sistemy rozpredelinije resursov na grafach. Sbornik. Vyčislitelnyj Centr AN SSSR,Moskva 1970.

[3] A r c h i p o v a T.T., R o š i n V.A., S i e r g i e n k o I.V., O rešenii odnovo klassa zadač organizacii vyčislitelnovo processa v sistemach obrabotki informacji. Kibernetika No 2,1973.

[4] A r t a m o n o v G.T., Cifrova vyčeslitelnaja mašina. w: Massovoe obsłuživanie v sistemach peredači informacji - Sbornik. Izd. "Nauka", Moskva 1969.

[5] A v e n O.I., K o g a n J.A., Matematičéskie modeli zložnych vyčislitelnych sistem (obzor).Avtomatika i Telemechanika,No 1, 1971.

[6] B o r o v k o v A.A., Verojatnostnye processy v teorii massovo obsluživanija. Izd. "Nauka", Moskva 1972.

[7] B o r o w i e c J., Wprowadzenie do języka PL/1, w: Marczyński R. (red.),Problemy przetwarzania informacji.T.1,WNT,Warszawa 1970.

[8] B r o w n J.C., C h a n d y K.M., H o g a r t h J., L e e C. C.-A: The effect on throughput of multiprocessing in a multiprogramming envoirment. IEEE Trans.Comput., vol.C-22,No.8, 1973.

[9] B u d z y ń s k i S., C h o r o b i ń s k i A.; Matematyczne podstawy optymalizacyjne w projektowaniu zakładów przemysłowych. Zagadnienie obciążenia maszyn i urządzeń. CO PAN, Warszawa 1968.

[10] C h a n g S.K., The computation of window operation in a parallel organized computer - a case study. IEEE Trans., Comput., vol.C-22, No.1, 1973.

[11] C h a n d y K.M., D i c k s o n J.R., Scheduling unindentical processors in a stochastic envoirment."COMPCON 72, 6-th Annual IEEE Comput.Soc.Int.Conf., San Francisco, Dig.Rap. 1972: Innov. Comput. Archit.", New York 1972.(Tłum.na ros. w Ekspress Informacja No.32, 1973).

[12] C h e n W.K., Applied graph theory. North-Holland Publishing Company, Amsterdam-London 1971.

[13] C i c h o c k i K., Optymalizacja w deterministycznych modelach produkcyjnych, Arch.Aut.Telem., t. XVII, z. 3, 1972.

[14] D e g t a r i e v E.K., Stochastičeskie modeli funkcjonirovanija ACVM. Techničeskaja Kibernatika No.2, 1967.

[15] F e l l e r W., Wstęp do rachunku prawdopodobieństwa. (Tłum. z ang.), T.I, t.2, PWN, Warszawa 1966, 1969.

[16] F e r n a n d e z E.B., B u s s e l B., Bounds on the number of processors and time for multiprocessor optimal schedules. IEEE Trans. Comput., vol.C-22, No.8, 1973.

[17] F l y n n M.J., Shared internal resources in a multiprocessor w: [19],vol. 1.

[18] F o r d L.R., F u l k e r s o n D.R., Przepływy w sieciach. Tłum. z ang., WNT, Warszawa 1966.

[19] F r e i m a n C.V. (ed.), Information Processing. Proceed. IFIP Congress 71. Vol.1, vol.2, North-Holland Publishing Company, Amsterdam 1972.

[20] G l u š k o v V.M., B a r a b a n o v A.A., K a l i č e n k o A.A., M i c h n o v s k i j S.D., R a b i n o v i č Z.L., Vyčislitelnye mašiny s razvitymi sistemami interpretacii. Izd. "Naukova Dumka", Kijev 1970.

[21] G n i e d e n k o B.V., K o v a l e n k o I.N., Vvedenie w teoriju massovo obslužіvanija. Izd. "Nauka", Moskva 1966.

[22] G o ł o v k i n B.A., Postrojenie verojatnostnych modeli i analiz parallelnych vyčislitelnych procesorov. Techničeskaja Kibernetika, No.3, 1973.

[23] G o n z a l e z M.J., Jr., R a m a m o o r t h y C.V., Program suitability for parallel processing. IEEE Trans. Comput., vol. C-20, No.6, 1971.

[24] G o n z a l e z M.J., Jr., R a m a m o o r t h y C.V., Parallel task execution in a decentralized system. IEEE Trans.Comput.,vol. C-21, No.12, 1972.

[25] G o r e n s t e i n S., An algorithm for project (job) sequencing with resources constraints. Oper. Res., vol.20, No.4, 1972.

[26] G r a b o w s k i J., S y s ł o M.M., On a machine sequencing problem (I). Applic. Math. t.XII, z.3, 1973.

[27] H e b a l k a r P.G., A graph model for analysis of deadlock prevantion in system with parallel computations, w: [19],vol.1.

[28] H o a r e C.A.R., P e r r o t t R.H. (eds), Operating systems techniques. Academic Press, London-New York 1972.

[29] H u T.C., Parallel sequencing and assembly line problems,Oper. Res., vol.9, Nov.-Dec., 1961.

[30] H u z a r Z., Rozdział zadań niezależnych w systemie obsługi w warunkach probabilistycznych. ICT Politechniki Wrocławskiej, Kom. nr 25, Wrocław 1973.

[31] H u z a r Z., Optymalny rozdział pewnej klasy zadań w systemie obsługi w warunkach probabilistycznych. ICT Politechniki Wrocławskiej. Kom. nr 53, Wrocław 1973.

[32] H u z a r Z., Zastosowanie metody podziału i ograniczeń dla optymalnego rozdziału zadań o dowolnych rozkładach prawdopodobieństwa czasów wykonania. ICT Politechniki Wrocławskiej, Kom. nr 75, Wrocław 1973.

[33] J a n k o w s k a - Z a r y c h t a Z., Modele sekwencyjne i ich zastosowanie w planowaniu optymalnej organizacji o dyskretnych procesach produkcji. Praca doktorska, CO, PAN, Warszawa 1973.

[34] J e v r e i n o v E.V., K o s a r i e v J.G., Matričnyj p-jazyk dla opisanija parallelnych avtomatov. Vyčislitelnye Sistemy No. 17, 1965.

[35] K a m b u r a c i T.E., Precedence diagramming and analysis in assembly line-balancing. w: [19], vol. 2.

[36] K a r p R.M., Reducibility amoung combinatorial problems. w: Miller R.E. (ed.), Complexity of computer computation. Plenum Press, New York 1972.

[38] K u c k D.J., M u r a o k a Y., C h e n S. C., On the number of operations simultaneously executable in Fortran-like programs and their resultating speedup. IEEE Trans.Comput., vol.C-21, No.12, 1972.

[39] K u r t z b e r g J.M., V i l l a n i R.D., A balanced pipelining approach to multiprocessing on an instruction stream level. IEEE Trans. Comput., vol. C-22, No.2, 1973.

[40] L a r s o n R.E., T s e E., Parallel processing algorithms for the optimal control of nonlinear systems. IEEE Trans. Comput. No. 8, 1973.

[41] L i p a e v V.V., K o l i n K.K., S e r e b r o v s k i j L.M Matematičeskoe obezpečenie upravlajuščich C.V.M., Izd. "Sovetskoe Radio", Moskva 1972.

[42] M i l l e r R.E., A comparison of some theoretical models of parallel computation. IEEE Trans. Comput., vol.C-22, No.8, 1973

[43] M o o r e F.R., Computational model of a closed quening network with exponential servers. IBM J. Res. Devel., November 1972.

[44] M u n t z R.R., C o f f m a n E.G., Jr., Optimal preemtive scheduling on two-processor system. IEEE Trans.Comput., vol.C-18, No.11, 1969.

[45] M u n r o P., P a t e r s o n M., Optimal algorithms for parallel polynomial evaluation. J. Comput. Syst. Sci., No 7, 1973.

[46] N i e m i n e n J., An optimal processing unit graph. Int. J. Syst. Sci., vol4, No.3, 1973.

[47] N o e J.D., N u t t G.J., Macro E-nets for representation of parallel systems. IEEE Trans.Comput., vol.C-22, No.8, IEEE, Trans. Comput., vol. C-22, No.8, 1973.

[48] O w e n G., Teorija igr. Tłum. z ang., Izd. "Mir", Moskva 1971.

[49] P o l a k E., Computational methods in optimization. A unifield, approach. Academic Press, New York 1971.

[50] P o s p i e l o v D.A., Vvedenie v teoriju vyčislitelnych sistem. Izd. "Sovetskoe Radio", Moskva 1972.

[51] P o t r z e b o w s k i H., Planowanie kalendarzowe (Przegląd zagadnień). Prace Instytutu Cybernetyki Stosowanej PAN, z.13, 1973.

[52] R a m a m o o r t h y C.V., C h a n d y K.M., G o n z a l e z J.A., Jr., Optimal scheduling strategies in a multiprocessor system. IEEE Trans.Comput., vol. C-21, No.2, 1972.

[53] R a s c h P.J., A quening theory study of round-robin scheduling of time-shared computer systems. J. ACM , vol.17, No.1,1970.

[54] R e d d i S.S., R a m a m o o r t h y C.V., A scheduling problem. Oper. Res. Q., vol. 24, No.3, 1973.

[55] R e g i s R.C., Multiserver quening models of multiprossing systems. IEEE Trans. Comput., vol.C-22, No.8, 1973.

[56] R o s e n f e l d J.L., V i l l a n i R.D., An approach to multiprocessing of the level of very small tasks. IEEE Trans. Comput., vol. C-22, No.2,1973.

[57] R o y B., B e n a y o u n B., T e r g n y I., From S.E.P. procedure to the mixed Ophelie program. w: [1].

[58] R u b i n o v i t c h M., A probabilistic model for optimal project scheduling. Oper. Res., vol 20, No.2, 1972.

[59] R y ž k o v A.P., Vybor optimalnych posledovatelnosti vypolnienija program na E.V.M. Techničeskaja Kibernetika, No.2, 1973.

[60] S a a t y T.L., Elementy teorii massovo obsluživanija i jejo priloženija. Tłum. z ang., Izd. "Sovetskoe Radio", Moskva 1971.

[61] S c h e r r A.L., An analysis of time-shared computer systems. The MIT Press, Cambridge, Massachusetts 1967.

[62] S i l v e r m a n G.J., Primal decomposition of mathematical programs by resource allocation: I-Basic theory and divection -funding procedure. II - Computational algorithm with an application to the modular design problem. Oper.Res., vol.20, No.1, 1972.

[63] S o l i c h R., Zadanie planowania kalendarzowego dla grup detali. Prace CO PAN, nr 109, 1973.

[64] T a n g D.T., W o n g C.K., A modified branch-and-bound strategy. Inform. Processing Letters, vol. 2, No.8, 1973.

[65] T j a d e n G.S., F l y n n M.J., Representation of concurrency with ordering matrices. IEEE Trans.Comput., vol. C-22, No.8, 1973.

[66] T o m l i n J.A., Branch and bound methods for integer and nonconvex programming. w: [1].

[67] T u r s k i W.M. (red.), Wybrane zagadnienia systemów operacyjnych. CO PAN, PWN, Warszawa 1971.

[68] V a r a i y a P.P., Notes on optimization. Van Nostrand Reinhold Company, New York 1972.

[69] W a l u k i e w i c z S., Twierdzenia o istnieniu cyklicznych i acyklicznych marszrut Hamiltona w grafie skończonym. (Algebra dróg elementarnych). Prace V KKA sekcja 9, Gdańsk 1971.

[70] W a r m u s M., Optymalny podział zadań planowych w branży przemysłowej. CO PAN, Warszawa 1971.

[71] W i s m e r D.A., Solution of the flowshop-scheduling problem with no intermediate quenes. Oper.Res., vol.20, No.3, 1972.

[72] Z y k o v A.A., Teorija końečnych grafov. Izd. "Nauka", Novosibirsk 1969.

DODATEK

```
begin
 comment Program wyznaczania optymalnej funkcji rozdziału dla zadań o
         wykładniczym rozkładzie prawdopodobieństwa czasów wykonywa-
         nia zadań elementarnych w jednofazowym systemie obsługi;
 integer m,L,n;
 setinput (1);
 read (m,L,n);
 begin
  integer i,j,k,L,m,n,MM,moc SKON,moc PWD,moc WD,moc PWO,moc WO,moc
          INDSO,OPTY;
  real  wS, wag, wag 1, lam, pr;
  boolean alfa;
  integer array M,S,SS,D[1:m],NRZADEL,INDSO,NASTSC[1:100],IND[1:50],
                Q[0:m+1, 0:m+1],PWD,PWO[1:100, 1:m], WD,WO[1:60,1:m],
                SKON[1:50, 1:m];
  real array  WAGWO,WAGWD[1:60], P[1:m, 1:L], lambda [1:m];
  boolean array  FI [1:100];
  comment 1. Blok wprowadzenia danych i wyprowadzenie nagłówka
             Oznaczenia w całym programie:
             i,j,k - indeksy pomocnicze,
             m - ilość zadań elementarnych,
             n - ilość stanowisk obsługi,
             L - ilość maksymalna typów łuków wychodzących z wierz-
             chołka zadania elementarnego,
             moc SKON,moc PWD,... - indeksy bieżące tablic o odpo-
                                    wiednich nazwach,
             OPTY - patrz komentarz 8,
             wS,wag,wag 1,lam,pr - zmienne pomocnicze przy obliczaniu
                                   wag stanów decyzji,
             Q - matryca incydencji,
             SKON - tablica stanów końcowych,
             PWO (PWD) - pierwotna warstwa stanów odpowiedzi (decyzji)
             WO (WD) - warstwa stanów odpowiedzi decyzji,
             WAGWO (WAGWD) - wagi stanów należących do warstwy odpo-
                             wiedzi (decyzji),
             FI - tablica pomocnicza
             pozostałe tablice opisane są w komentarzach 2-7;
  read (lambda, P, Q);
  setoutput (3);
  comment Wyjście na drukarkę wierszową;
  print (‘??? Parametry  zadania : lambda, P, Q??’);
  for  i:=1 step 1 until m  do
   begin
    format (‘␣123.12␣␣␣’);
    print (lambda [i]);
    format (‘␣0.12␣’);
    for j:=1 step  1  until  L  do  print  (P[i,j])
   end;
```

```
line (1);
format ('␣-1');
k:=m+1;
for j:= 0  step  1  until  k  do
 begin
  line (1);
  for  i:= 0  step  1  until  k  do  print (Q[j,i])
 end;
  comment 2. Generacja stanów końcowych
             Oznaczenia:
              M[1:m]  - maksymalna liczba typów łuków wychodzących
                        z i-tego wierzchołka ,
              S[1:m]  - kolejny generowany stan końcowy,
             SS[1:m]  - stan pomocniczy służący do badania czy S[i]
                        należy do zbioru stanów dopuszczalnych,
              D[1:m]  - wektor o składowych równych 1 odpowiadają-
                        cych elementarnym zadaniom operacyjnym wcho-
                        dzących w skład każdego stanu końcowego,
      SKON[1:50, 1:m] - tablica stanów końcowych,
           IND [1:50] - liczba wykonanych zadań elementarnych wcho-
                        dzących w skład odpowiedniego stanu końcowe-
                        go,
            FI [1:m]  - boolowska tablica pomocnicza;
  for i:=1  step  1  until  m  do
   begin
    MM:=1; k:=m+1;
    for j:=1  step  1  until  k  do
     if Q [i,j] gt  1  then  MM:=Q[i,j];
    M[i]:= MM
   end;
  for  i:=1  step  1  until  m  do
   begin
    D[i]:= if Q[0,i]=1  then  1  else 0;
    FI[i]:= false
   end;
E1:alfa:= false;
   for i:=1  step  1  until  m  do
    if  D[i] = 1  then
     begin
      if  FI[i]  eqv false then
       begin
        FI[i]:=alfa:= true;
        for j:=1 step 1 until m do
         if Q[i,j] =1 then D[j] :=1
       end
     end;
```

```
    if alfa eqv true then go to E1;
    moc SKON:=0;
    for i:=1 step 1 until m do S[i]:=D[i];
 E2:for i:=1 step 1 until m do
     if D[i]=0 then
      begin
       if S[i]=M[i] then S[i]:=0
                     else
                      begin
                       S[i]:=S[i]+1;
                       for k:=1 step 1 until m do
                        begin
                         SS[i]:=D[i];
                         FI[i]:= false
                        end;
                  E3:alfa:=false
                     for k:=1 step 1 until m do
                      if SS[k] ≠ 0 then
                       begin
                        if FI[k]  eqv false then
                         begin
                          FI[k]:=alfa:=true;
                          for j:=1 step 1 until m do
                           if Q[k,j] =S[k]  then SS[j] :=S[k]
                         end
                       end k;
                     if alfa eqv true then go to E3; j:=0;
                     for k:=1 step 1 until m do
                      if S[k]≠ SS[k]  then go to E2
                                      else if S[k] =0 then j:=j+1;
                     moc SKON:=moc SKON+1;
                     for k:=1 step 1 until m do SKON[mocSKON,k]:=S[k];
                     IND[moc SKON]:= j;
                     go to E2
                    end ;
comment 3. Tworzenie najgłębszej warstwy odpowiedzi
           Oznaczenia:
           WO[1: moc WO,1:m] - tablica stanów końcowych odpowiedzi,
            WAGWO[1:moc WO ] - tablica wag stanów odpowiedzi;
```

```
MM:=1;
for i:=1 step 1 until moc SKON do
  MM:=if MM lt IND[i] then IND[i];
moc WO:=0;
for i:=1 step 1 until moc SKON do
 if IND[i]=MM then
  begin
   moc WO:=moc WO+1;
   WAGWO[moc WO]:=.0;
   for j:=1 step 1 until m do WO[moc WO,j]:=SKON[i,j]
  end;
comment 4. Tworzenie pierwotnej warstwy stanów decyzji
           Oznaczenia:
           PWD[1:moc PWD,1:m]   - tablica stanów odpowiedzi,
           NASTSO[1:moc PDW]    - parametr określający pozycję stanu
                                  odpowiedzi następującego po odpo-
                                  wiednim stanie decyzji;
Epowt:moc PDW:=0;
    for i:=1 step 1 until moc WO do
     for j:=1 step 1 until m do
      if WO[i,j] ge 1 then
       begin
        for k:=1 step 1 until m do
         if Q[j,k] ge 1 then
          begin
           if WO[i,k] gt - 1 then go to E4
          end;
       moc PWD:= moc PWD+1
       PWD[moc PWD,j]:=0;
       for k:=1 step 1 until j-1, j+1 step 1 until m do
        PDW [ moc PDW,k]:=WO[i,k];
       NASTSO[mocPWD] :=i;
  E4:end;
  comment 5. Tworzenie warstwy stanów decyzji oraz odpowiadających
             im wag.
             Oznaczenia:
             WD[1:moc WD,1:m] - tablica stanów decyzji,
              WAGWD[1:moc WD] - tablica wag stanów decyzji,
           INDSO[1:moc INDSO] - tablica indeksów (nr w tablicy WO)
                                stanów odpowiedzi następujących po
                                danym stanie decyzji
       NRZADEL[1:mocINDSO] - numery zadań elementarnych, które
                                mogą być wykonane po wyjściu z da-
                                nego stanu decyzji;
```

```
moc WD:=0,
for i:=1 step 1 until moc PWD do FI[i]:= false;
for i:=1 step 1 until moc PWD do
 if FI[i] eqv false then
  begin
   moc WD:=moc WD+1;
   for j:=1 step 1 until m do WD[moc WD,j] :=PWD[i,j];
   moc INDSO:=1;
   INDSO[1]:=NASTSO[i];
   for j:=i+1 step 1 until moc PWD do
    if FI[j] eqv false then
     begin
      for k:=1 step 1 until m do
       if PWD[i,k] ≠ PWD[j,k] then go to E5;
      FI[j]:= true;
      moc INDSO:= moc INDSO+1;
      INDSO[moc INDSO]:=NASTSO[j];
  E5:end;
   comment 6. Obliczenie wagi pojedynczego stanu decyzji;
   for k:=1 step 1 until moc INDSO do
    begin
     for j:=1 step 1 until m do
      if PDW[INDSO[k],j] ≠ WD [ moc WD,j] then
       begin
        NRZADEL[k]:=j;
        go to E6
       end;
     print (‘??błąd?’);
     go to E koniec ;
 E6:end;
  wS:=.0;
   for k:=1 step 1 until moc INDSO do wS:=wS+lambda [NRZADEL[k]];
  wS:=1.0/wS;
   wag:=.0;
   for k:= step 1 until moc INDSO do
    begin
     wag 1:=wS+WAGWO[INDSO[k]] ;
     lam:=lambda[NRZADEL[k]];
     pr:=lam ↑ (moc INDSO-1)×P[NRZADEL[k],WO[INDSO[k],NRZADEL[k]]];
     for j:=1 step 1 until k-1,k+1 step 1 until moc INDSO do
      pr:=pr /(lam + lambda[NRZADEL[j]]) ;
```

```
        wag:=wag+pr×wag 1
       end;
      WAGWD[moc WD]:=wag
    end i;
  comment 7. Tworzenie pierwotnej tablicy stanów odpowiedzi
             Oznaczenia:
             PWO[1:moc PWO,1:m] - pierwotna warstwa odpowiedzi,
             NAST SO[1:moc PWO] - zawiera indeks (numer w tablicy
                                  WD) stanu decyzji, który nastę-
                                  puje po odpowiednim stanie od-
                                  powiedzi;
  moc PWO:=0;
  for i:=1 step 1 until moc WD do
   begin
    k:=0;
    for j:=1 step 1 until m do
     it WD[i,j]=0 then
                    begin
                     D[j]:=0;
                     k:=k+1
                    end
                   else D[j]:=1;
    if k lt n then
     begin
      moc PWO:=moc PWO+1; FI[moc PWO]:=false;
      for j:=1 step 1 until  m do PWO[moc PWO,j]:=WD[i,j];
      NASTSO[moc PWO]:=i
     end;
E7:for j:=1 step 1 until m do
    if D[j]≠ 1 then
     begin
      if D[j] =0 then
                    begin
                     D[j]:=-1;
                     go to E8
                    end
                   else D[j]:=0
     end;
   go to E9;
E8:moc PWO:=moc PWO+1; FI[moc PWO]:=false;
   for j:=1 step 1 until m do
    PWO[moc PWO,j]:=if D[j]=1 then WD[i,j]
                                else D[j];
   go to E7;
```

```
E9: end;
   comment 8. Tworzenie tablicy stanów odpowiedzi
              Oznaczenia:
              WO[1:moc WO,1:m] - tablica stanów odpowiedzi,
              WAGWO[1:mocWO]   - tablica wag stanów odpowiedzi
                        OPTY   - numer stanu (w tablicy stanów od-
                                 powiedzi)optymalnej decyzji;
   moc WO:=0;
   for i:=1 step 1 until moc PWO do
    if FI[i] eqv false then
     begin
      moc WO:=moc WO+1;
      for j:=1 step 1 until m do WO[moc WO,j]:=PWO[i,j];
      OPTY:=NASTSO[i];
      WAGWO[moc WO]:=WAGWD[OPTY];
      for j:=i+1 step 1 until moc PWO do
       if FI[j]  eqv false then
        begin
         for k:=1 step 1 until m do
          if PWO[i,k] ≠ PWO[j,k] then go to E10;
         FI[j]:=true;
         if WAGWO[moc WO] gt WAGWD[NASTSO[j]] then
          begin
           OPTY:=NASTSO[j];
           WAGWO[moc WO]:=WAGWD[OPTY]
          end ;
    E10:end:
     comment 9. Wyprowadzenie wyników w postaci:
                stan odpowiedzi - stan decyzji - waga decyzji;
     format ('??␣␣␣␣␣warstwa␣␣␣123?');
     print  (MM);
     format ('-1');
     for k:=1 step 1 until m do print (WO[moc WO,k]);
     print ('␣␣␣␣␣␣');
     for k:=1 step 1 until m do print (WD[OPTY,k]);
     format ('␣␣␣1234');
     print (WAGWO[moc WO])
    end;
   comment 10. Dołączenie do warstwy odpowiedzi - stanów końcowych;
   if MM=1 then go to Ekoniec;
   MM:=MM-1;
   for i:1 step 1 until moc SKON do
    if IND[i] = MM then
```

```
          begin
           moc WO:=moc WO+1;
           for j:=1 step 1 until m do WO[moc WO,j]:=SKON[i,j];
           WAGWO[moc WO]:=.0
          end; go to Epowt ;
Ekoniec:end program
```