

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ
Raport Serii: PREPRINTY Nr 2/81

ZAGADNIENIA OPTYMALIZACJI SEKWENCJI
OPERACJI W DYSKRETNYCH SYSTEMACH
PRODUKCYJNYCH Z KRYTERIUM MINIMALNO-
KOSZTOWYM

/praca doktorska/

Czesław Smutnicki

Promotor: doc. dr hab. inż. Józef Grabowski

Słowa kluczowe: szeregowanie,
optymalizacja,
algorytm

Wydawnictwo Politechniki Wrocławskiej

Wrocław 1981

SPIS TRESCI

1. Wprowadzenie	8
2. Cel i zakres pracy	9
3. Sterowanie dyskretnym procesem produkcyjnym	13
4. Aktualny stan literatury. Klasyfikacja zagadnień kolejnościowych	17
5. Minimalizacja kosztu maksymalnego Zagadnienie $n m F, r_i \geq 0, \prec F_{\max}$	24
5.1. Pewne własności zagadnienia $n m P, r_i \geq 0, \prec f_{\max}$	26
5.2. Schemat przebiegu algorytmu	30
5.3. Wyznaczenie dolnych ograniczeń	33
5.4. Zagadnienie wzmocnienia dolnych ograniczeń	39
5.5. Zasada podziału	46
5.6. Algorytm	47
5.7. Realizacja programowa algorytmu	49
5.8. Eksperymenty obliczeniowe	50
5.9. Przykłady testujące. Wyniki obliczeniowe	52
5.10. Wnioski i uwagi	57
6. Minimalizacja kosztu maksymalnego Zagadnienie $n m F, \prec h_{\max}$	58
6.1. Pewne własności zagadnienia $n m P, \prec h_{\max}$	60
6.2. Dolne ograniczenia	72
6.3. Algorytm	73
7. Minimalizacja kosztu maksymalnego. Uogólnienie	76
8. Minimalizacja kosztu całkowitego Zagadnienie $n 1 \prec \sum h_i$	83
8.1. Sformułowanie problemu	83
8.2. Modele i metody PD i PC do rozwiązywania problemu	86
8.3. Rozwiązanie optymalne i biooptymalne	89
8.4. Kryteria eliminacji	102
8.5. Schemat przebiegu algorytmu	110
8.6. Wyznaczanie dolnych ograniczeń	114
8.7. Zasada podziału	133
8.8. Algorytm	134

8.9. Realizacja programowa algorytmu. Przykłady testujące.	139
8.10. Eksperymenty obliczeniowe. Wyniki.	140
8.11. Wnioski i uwagi	145
9. Minimalizacja kosztu całkowitego	
Zagadnienie $n m J, \tau \gg 1, m_w^k \gg 1, RT \sum h_i$	146
9.1. Sformułowanie problemu	146
9.2. Pewne własności zagadnienia $n m J, \tau \gg 1, m_w^k \gg 1,$ $RT \sum h_i$	149
9.3. Schemat przebiegu algorytmu	155
9.4. Dolne ograniczenia	160
9.5. Zasady podziału.	172
9.6. Algorytm	174
9.7. Eksperymenty obliczeniowe	176
9.8. Wnioski i uwagi	179
10. Zastosowanie	179
11. Zakończenie	183
12. Literatura	186

WYKAZ WAŻNIEJSZYCH SYMBOLI I OZNACZEN

- A_j, A_{π}^j - zbiór następników zadania J_j /ustalony w permutacji π /,
 a_i, a_{ik} - najwcześniejszy dopuszczalny termin rozpoczęcia realizacji zadania /operacji/,
 B_j, B_{π}^j - zbiór poprzedników zadania J_j /ustalony w permutacji π /,
 b_i, b_{ik} - najpóźniejszy dopuszczalny termin zakończenia realizacji zadania /operacji/,
 C_i, C_{ik} - termin zakończenia wykonywania zadania J_i /operacji/ na maszynie M_k ,
 $C_{\max}(\pi), L_{\max}(\pi), T_{\max}(\pi), f_{\max}(\pi)$ - wartość minimaksowej funkcji celu dla permutacji π ,
 d_i, d_{ik} - pożądaný termin zakończenia wykonywania zadania /operacji/,
 E_i - przyspieszenie zadania /operacji/,
 $F(\bar{S}; \pi)$ - wartość funkcji celu, dla permutacji π ,
 $G_t(\bar{S}), G_{tk}(\bar{S})$ - zbiór operacji realizowanych w przedziale $[t-1, t]$ na maszynie k -tej,
 $g_t(\bar{S}), g_{tk}(\bar{S})$ - liczba operacji w zbiorach $G_t(\bar{S})$ lub $G_{tk}(\bar{S})$,
 $g_i(t), f_i(t), h_i(t)$ - funkcje kosztu realizacji zadań,
 $g_{ik}(t), f_{ik}(t), h_{ik}(t)$ - funkcje kosztu realizacji operacji,
 $H, \bar{H}, \hat{H}$ - drzewo rozwiązań,
 J - zbiór zadań,
 J_i - zadanie,
 L_i - nieterminowość realizacji zadania /operacji/,
 $L(\bar{S}, \bar{u}, \bar{\lambda}), L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda})$ - funkcja Lagrange'a,
 $L_i(S_i, \bar{u}, \bar{\lambda}), L_{ik}(S_i, k, \bar{u}, \bar{\lambda})$ - częściowa funkcja Lagrange'a,
 λ_{ij} - zmienna dualna,
 LB - dolne ograniczenie,
 M - zbiór maszyn,
 m - liczba maszyn,
 M_k, k - maszyna,

N	- zbiór operacji,
n	- liczba operacji,
$N_{k\mu}$	- zbiór operacji realizowanych na maszynie k -tej przy przydziale μ ,
$n_{k\mu}$	- liczba operacji w zbiorze N_k ,
$\bar{\mu}$	- wektor przydziałów operacji,
$\mu(i)$	- numer maszyny, na której realizowana jest operacja i -ta,
$\mathcal{M}$	- zbiór dopuszczalnych wektorów przydziału,
O_{ik}	- operacja k -ta zadania J_j ,
Ω	- ciąg definiujący typ ograniczenia,
p_i, p_{ik}	- czas realizacji operacji,
π, β, α	- permutacja wskaźników zadań /operacji/,
π_k, β_k	- permutacja wskaźników operacji na maszynie k -tej,
$\bar{\pi}, \bar{\beta}$	- wektor permutacji,
Π	- zbiór wszystkich permutacji,
$\Pi(\mathcal{R})$	- zbiór dopuszczalnych permutacji dla przyjętej relacji $\mathcal{R}$ w zbiorze zadań /operacji/,
Π_μ	- zbiór wektorów permutacji do przydziału $\bar{\mu}$,
Q	- zbiór typów maszyn,
q_i, q_{ik}	- końcówka zadania /operacji/,
r_i, r_{ik}	- pożądaný termin rozpoczęcia zadania /operacji/,
$\mathcal{R}, RT$	- relacja poprzedzania w zbiorze zadań /operacji/,
$\mathcal{R}_\pi, \mathcal{R}_{\mu\pi}$	- relacja ustalona w permutacji π ; relacja ustalona dla wektora π i przydziału μ ,
S_i, S_{ik}	- termin rozpoczęcia wykonywania zadania /operacji/,
$\bar{S}, \bar{S}_\pi, \bar{S}_{\mu\pi}$	- zbiory dopuszczalnych momentów rozpoczęcia wykonywania zadań /operacji/,
T_i	- spóźnienie w wykonywaniu zadania /operacji/,
T	- horyzont czasowy,
t_0	- najwcześniejszy dopuszczalny moment rozpoczęcia realizacji wszystkich zadań /operacji/,
UB	- górne ograniczenia,
u, u_t, u_{tk}	- zmienne dualne,
$u_1, u_2, \dots, u_{m+1}$	- ciąg krytyczny,

$v_i, w_i, z_i, v_{ik}, w_{ik}, z_{ik}$ - wagi,
 $W(\bar{u}, \bar{\lambda})$ - funkcja dualna,

Problems of Operations Sequencing
 Optimization in Discrete Production
 Systems with Minimum- Cost Criterion

Abstract

This paper deals with the theoretical and practical aspects of some sequencing problems. Two general classes of sequencing problems with cost function /penalty function/ are considered. There are job-shop problems with maximum penalty criterion and total penalty criterion. The penalties representing: job store cost, job waiting cost, job deferral cost, machine idle cost, earliness and tardiness job penalties. The general problems classification according to goal function, type of problem and additional constraints are presented. Four new sequencing problems are formulated. There are: minimizing maximum penalty in flow-shop problem, minimizing maximum earliness and tardiness penalties in flow-shop problem, minimizing total earliness and tardiness costs in one-machine problem, minimizing total earliness and tardiness penalties in job-shop problem with parallel machines. All these problems are NP-complete.

Various mathematical models of the problems are formulated. A new properties are shown. Alternative approaches to problems solution are considered. A branch-and-bound algorithm with stopped fathoming is used to find optimal or suboptimal sequence. The proposed methods combine branch-and-bound scheme with heuristic algorithms. In each case several lower bounds are derived and discussed. The general methodology for computing lower bounds are presented and developed. Some lower bounds relaxation are described. Subgradient optimization method is used in minimizing total cost problem to maximize the value of lower bound into search tree algorithms.

The valid extensions of "critical sequence approach" and "dual approach" into general sequencing problems are introduced.

The computational experiences with these algorithms are reported. An practical application examples are shown.

W pracy rozważane są problemy szeregowania zadań z kryterium w postaci funkcji kosztu /kary/ związanej z nieterminowym wykonywaniem zadań. Wprowadzone funkcje kryterialne reprezentują zarówno koszty związane ze składowaniem zadań, koszty przestoju maszyn jak i kary za przyspieszenie i spóźnienie realizacji zadań w odniesieniu do żądanych terminów rozpoczęcia i zakończenia. W pracy przedstawiono ogólną klasyfikację problemów szeregowania ze względu na przyjęte postacie funkcji celu, typ problemu i rodzaj ograniczeń. Sformułowano, zbadano i rozwiązano cztery nowe problemy szeregowania. Dwa z nich dotyczą tzw. problemu taśmowego z różnymi, ogólnymi minimaksowymi funkcjami celu; problem, trzeci jest jednomaszynowym problemem szeregowania z najogólniejszą addytywną funkcją celu i stanowi punkt wyjścia do problemu czwartego - tzw. zagadnienia kolejnościowego gniazdowego z maszynami równoległymi. W przypadku podanych problemów udowodniono szeregowych własności oraz sformułowano algorytmy rozwiązania w oparciu o metodę podziału i ograniczeń. Zaproponowano i zanalizowano różne sposoby wyznaczania dolnych ograniczeń w algorytmach. Przedstawiono i oceniono wyniki badań numerycznych skonstruowanych algorytmów. Podano także przykłady zastosowań w wybranych procesach produkcyjnych.

1. WPROWADZENIE

W ostatnich latach można zaobserwować wzrost znaczenia problematyki optymalnego sterowania i automatyzacji procesów w dyskretnych systemach produkcyjnych. Jest to wynikiem praktycznych potrzeb w istniejących złożonych procesach technologicznych. Szybki rozwój teorii szeregowania oraz większe możliwości nowoczesnego sprzętu komputerowego umożliwiają zaspokojenie tych potrzeb w coraz większym zakresie.

W praktyce, przez pojęcie dyskretnego systemu produkcyjnego rozumie się takie oddziały wytwórcze, w których proces produkcji charakteryzuje się "przepływem" materiałów i półproduktów w postaci pojedynczych elementów, podzespołów lub zespołów. Elementami mogą być detale kęsiska, wlewki, odkuwki, a także programy maszyn cyfrowych, itp. Elementy te są poddawane operacjom obróbki na kolejnych stanowiskach /gniazdach/ wyposażonych w maszyny. Przez maszyny należy rozumieć piece grzewcze, maszyny do obróbki mechanicznej, klatki walcownicze, stanowiska montażu, brygady pracowników, procesory maszyn cyfrowych, itd. Operacjami mogą być: obróbka mechaniczna, czynności montażu, wytop stali, walcowanie, podgrzewanie, obróbka ręczna, proces obliczania, itp.

Proces przetwarzania elementów jest określony przez ciąg

kolejno po sobie następujących operacji uwarunkowanych czasowo, nazywany w dalszym ciągu zadaniem produkcyjnym.

Wszystkie zdarzenia ściśle związane z zadaniem produkcyjnym określają zbiór zdarzeń. Przebieg procesu produkcyjnego polega na czasowo uporządkowanym następowaniu zdarzeń z tego zbioru. Problem optymalnego sterowania procesem polega na ustaleniu takiej dopuszczalnej kolejności następowania zdarzeń z tego zbioru, aby optymalizować przyjęty wskaźnik efektywności. Tego rodzaju problemy optymalizacji są nazywane zagadnieniami kolejnościowymi lub szeregowania.

2. Cel i zakres pracy

Zakres pracy obejmuje deterministyczne zagadnienia szeregowania zadań, tzn. takie w których sformułowaniu nie występuje żadna zmienna o opisie probabilistycznym.

Z punktu widzenia teorii szeregowania zadań praca zawiera algorytmy rozwiązania zagadnień kolejnościowych z funkcją celu w najogólniejszej postaci. Klasa rozważanych w pracy problemów zawiera prawie wszystkie zagadnienia kolejnościowe sformułowane do tej pory w literaturze oraz inne dla których nie został podany jeszcze żaden algorytm.

Z matematycznego punktu widzenia praca prezentuje nowe własności i algorytmy pewnych klas zagadnień nieliniowego programowania dyskretnego.

Z technicznego punktu widzenia praca zawiera nowe algorytmy optymalizacji procesu w dyskretnych systemach produkcyjnych, przeznaczone do rozwiązywania zarówno zagadnień projektowania, planowania jak i optymalnego sterowania. Prezentowane zagadnienie i metody stanowią podstawową problematykę systemów sterowania złożonymi procesami produkcyjnymi.

Cele prezentowanej pracy są następujące:

- sformułowanie zagadnień kolejnościowych z funkcjami celu w ogólnej postaci,
- zbadanie własności sformułowanych zagadnień szeregowania,
- skonstruowanie algorytmów rozwiązywania wybranych klas zagadnień z funkcjami celu w ogólnej postaci,

- zrealizowanie eksperymentów obliczeniowych w zakresie zbada-
nia numerycznych własności zaproponowanych algorytmów,
- ocena przydatności zaproponowanych algorytmów dla optymaliza-
cji dyskretnego procesu produkcyjnego,
- zaproponowanie jednolitego podejścia do rozwiązywania zagad-
nień kolejnościowych z funkcjami celu w ogólnej postaci.

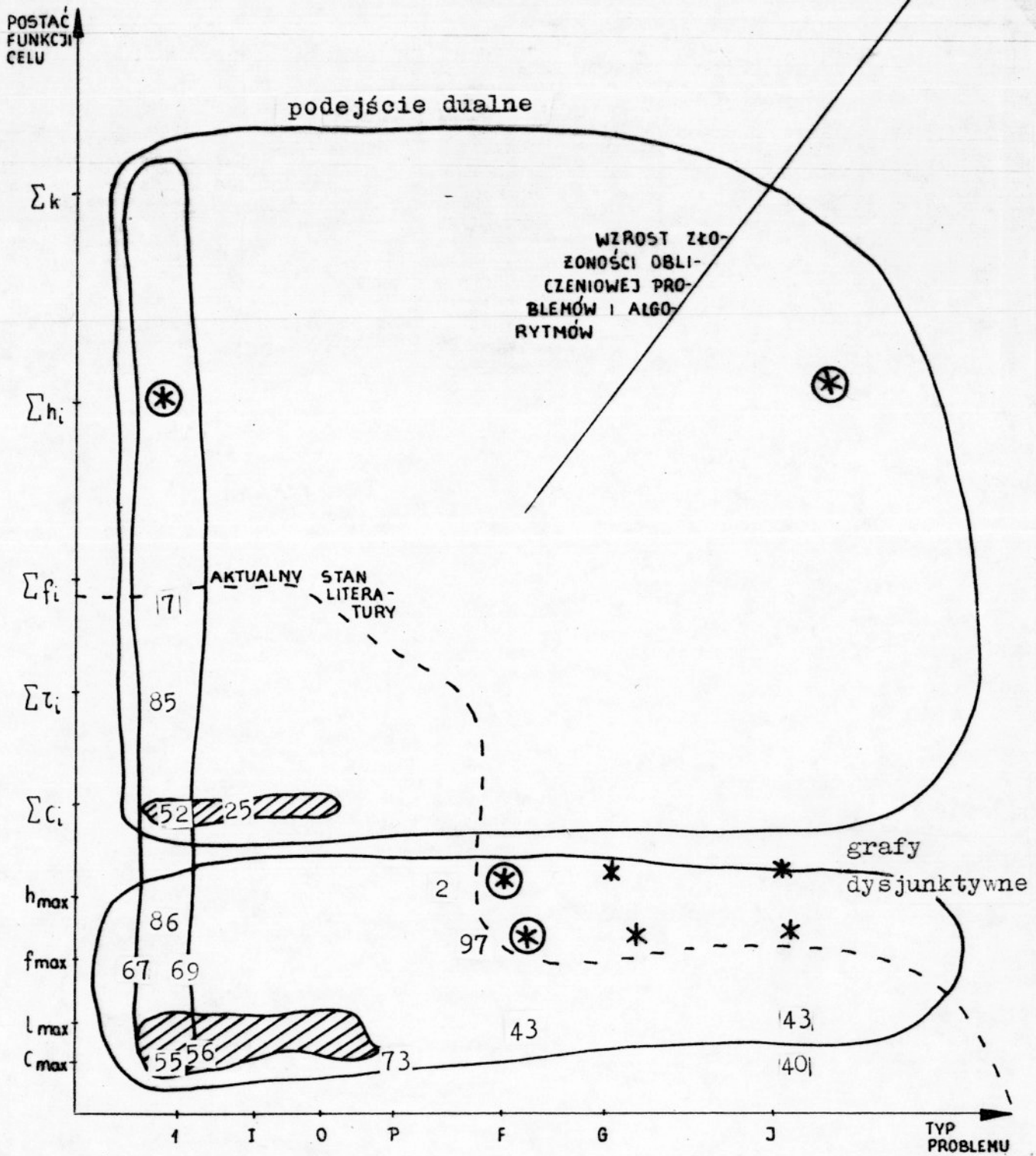
Praca niniejsza stanowi również próbę syntetycznego ujęcia i uogólnienia wszystkich rezultatów uzyskanych dla problemów szeregowania z minimaxową funkcją celu oraz z funkcjami celu typu sumacyjnego.

Praca zawiera:

- (i) wprowadzenie w problematykę optymalnego sterowania procesem w dyskretnych systemach produkcyjnych oraz ogólne koncepcje systemów sterowania tym procesem /rozd. 3 i 4/,
- (ii) sformułowanie 4 oryginalnych problemów szeregowania. Dwa z nich dotyczą tzw. problemu taśmowego z różnymi minimak-
sowymi funkcjami celu; problem trzeci jest jednomaszynowym problemem szeregowania z najogólniejszą addytywną funkcją celu i stanowi punkt wyjścia do problemu czwartego tzw. zagadnienia kolejnościowego gniazdowego z maszynami równoległymi /rozd. 5,6,8,9/.
- (iii) określenie i dowody własności sformułowanych zagadnień,
- (iv) opracowanie algorytmów rozwiązania,
- (v) implementację programową opracowanych algorytmów,
- (vi) wyniki eksperymentów obliczeniowych dla obszernego zestawu przykładów oraz ocenę efektywności algorytmów,
- (vii) uogólnienie udowodnionych własności na szeroką klasę problemów szeregowania /rozd. 7/,
- (viii) ilustrację zastosowań praktycznych na wybranych przykładach /rozd. 10/,
- (ix) podsumowanie i krytyczną ocenę uzyskanych rezultatów /rozd. 11/.

W celu bardziej precyzyjnego określenia relacji problemów rozwiązanych w pracy do problemów występujących w literaturze posłużono się ilustracyjnym diagramem /rys. 2.1/. Diagram ten przedstawia subiektywną ocenę złożoności problemów szeregowania dokonaną na podstawie:

- liczby i złożoności problemów częściowych, których rozwiązanie



Rys. 2.1. Przybliżony diagram złożoności obliczeniowej problemów i algorytmów szeregowania

P_D - programowanie dynamiczne

problemy P-klasy

(*) - problemy rozwiązane przez autora i zamieszczone w pracy

* - problemy rozwiązane przez autora

- jest wymagane dla rozwiązania problemu wyjściowego,
- stopnia ogólności problemu /ze względu na przyjęty model matematyczny/,
 - czasu działania algorytmu /dla ustalonego rozmiaru zadania/.

Na osi poziomej układu zaznaczano znane w literaturze typy zagadnień kolejnościowych: 1 - zagadnienie jednomaszynowe, I - zagadnienie jednogniazdowe z równoległymi maszynami, P - permutacyjne zagadnienie taśmowe, F - niepermutacyjne zagadnienie taśmowe, G - zagadnienie gniazdowe, J - zagadnienie gniazdowe z równoległymi maszynami. Wszystkie zagadnienia /z wyjątkiem I/ uszeregowano zgodnie ze wzrostem ogólności, tzn. zagadnienie P jest szczególnym przypadkiem zagadnienia F. Zagadnienie I jest szczególnym przypadkiem zagadnienia J, ale brak jest wyraźnej relacji tego zagadnienia do pozostałych. Jednakże ze względu na charakter metod używanych do rozwiązania zagadnienia typu I dokonano przybliżonej jego lokalizacji pomiędzy 1 i P.

Na osi pionowej układu zaznaczano w sposób przybliżony znane w literaturze typy funkcji celu /patrz rozdz. 4/. Typy funkcji celu zostały podzielone na 2 grupy: funkcje minimaksowe /typy funkcji są uszeregowane zgodnie ze wzrostem ogólności/ i funkcje o charakterze addytywnym /uszeregowane w analogiczny sposób/.

Podobnie jak poprzednio nie wszystkie minimaksowe funkcje celu są w wyraźnej relacji do funkcji addytywnych. Tak na przykład problemy z funkcją celu $C_{\max}$ i $L_{\max}$ są szczególnymi przypadkami problemów z funkcją celu $\sum C_i$, $\sum T_i$, $\sum f_i$, $\sum h_i$, $\sum K$ jednakże brak jest relacji między $f_{\max}$, $h_{\max}$ i $\sum C_i$, $\sum T_i$, $\sum h_i$, $\sum K$. W ogólnym przypadku niemożliwe jest również uwzględnienie szeregu innych elementów takich jak: dodatkowe założenia zmniejszające złożoność algorytmu, sposób implementacji programowej itp, dlatego też diagram posiada charakter orientacyjny i subiektywny.

Na diagramie zaznaczano cyframi lokalizację głównych rezultatów zawartych w odpowiednich pozycjach literaturowych oraz gwiazdką * problemy sformułowane i rozwiązane w prezentowanej pracy. Problemy oznaczone * zostały przez autora rozwiązane jednakże ich prezentacja przekracza objętość niniejszej pracy. Linia ciągłą objęto obszar stosowalności dwóch głównych metod rozwiązywania zagadnień kolejnościowych rozwijanych przez autora: metody podziału i ograniczeń z elementami grafów dysjunktywnych oraz metody podzia-

ku i ograniczeń z podejściem dualnym dla wyznaczania dolnych ograniczeń.

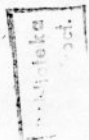
3. STEROWANIE DYSKRETNYM PROCESEM PRODUKCYJNYM

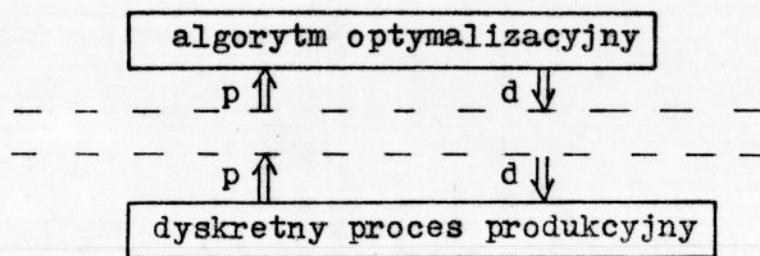
W pracy przyjęto definicje systemu produkcyjnego, procesu produkcyjnego i optymalnego procesu produkcyjnego podane w [43]. Przyjęto również wynikające z podanych definicji pojęcia zagadnienia kolejnościowego z równoległymi maszynami (J) oraz zagadnienia gniazdowego (G).

Optymalne sterowanie /oddziaływanie/ procesem w dyskretnym systemie produkcyjnym polega na ogół na takim określeniu przydziału operacji do maszyn oraz terminów rozpoczęcia i zakończenia wszystkich operacji na maszynach, by przyjęty wskaźnik jakości był optymalny. Optymalne sterowanie procesem może odbywać się w a/ układzie otwartym bez pomiaru zaburzeń /rys. 3.1/, b/ w układzie zamkniętym z pomiarem zaburzeń /rys. 3.2, 3.3, 3.4/.

Sterowanie w układzie otwartym /rys. 3.1/ charakteryzuje się brakiem możliwości bieżącej kontroli pomiaru parametrów zadania produkcyjnego /np. czasów realizacji czynności, itp/ oraz brakiem bieżącego oddziaływania sterowania d na przebieg procesu produkcyjnego. W tej sytuacji zadania optymalnego sterowania procesem sprowadza się do jednorazowego wyznaczenia wektora sterowań d. Czynność tę wykonuje się przed rozpoczęciem procesu produkcyjnego i nosi ona nazwę planowania.

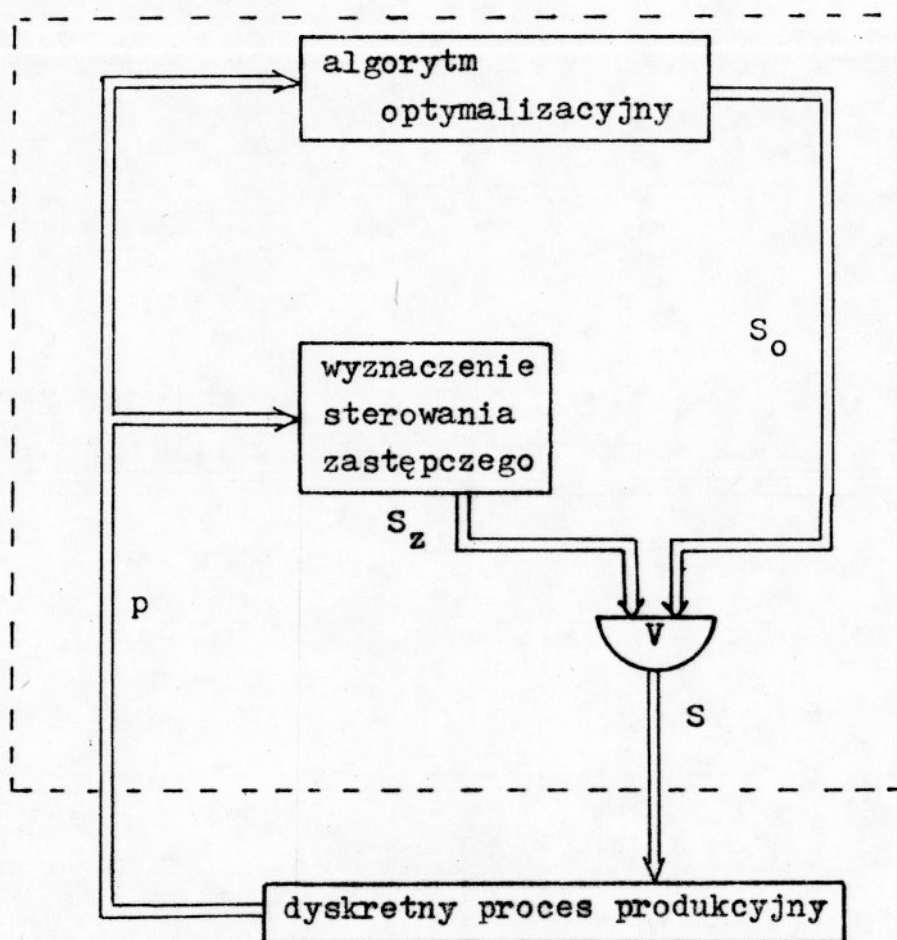
Sterowanie w układzie zamkniętym z pomiarem zaburzeń /rys. 3.2/ charakteryzuje się możliwością bieżącej kontroli i bieżącego wyznaczania optymalnego wektora sterowania. Ze względu na zakłócenie, jakie mogą wystąpić w procesie, mogą ulec zmianie parametry zadania produkcyjnego p /np. czasy trwania operacji, kolejność ich wykonania, przydział maszyn, itp./. Zadanie optymalnego sterowania procesem w układzie zamkniętym polega na pomiarze składowych sterowania zrealizowanego w procesie, wyrażonych przez parametry p zadania, a następnie na wyznaczeniu optymalnego wektora sterowań S dla operacji jeszcze nie zrealizowanych. Ponieważ zastosowanie złożonych algorytmów optymalizacyjnych w systemach uwarunkowanych





Rys. 3.1. Sterowanie w układzie otwartym

/p - parametry zadania produkcyjnego, d - decyzje/



Rys. 3.2. Sterowanie w układzie zamkniętym z pomiarem

zaburzeń /p - parametry zadania produkcyjnego,

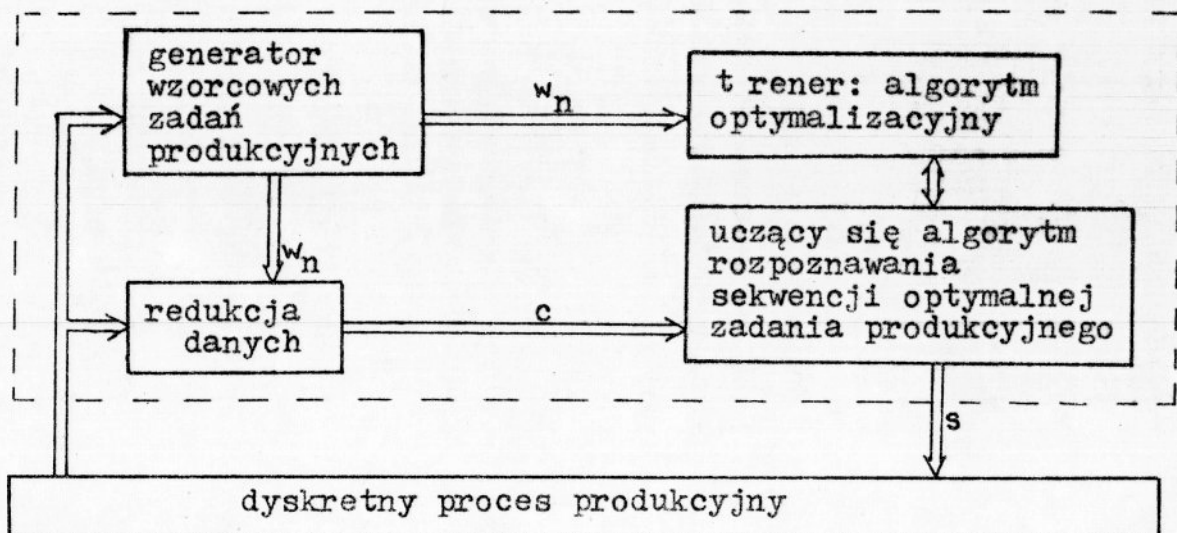
S_0 - sterowanie optymalne, S_z - sterowanie zastępcze,

s - sterowanie procesu/

czasowo nie zawsze gwarantuje otrzymanie rozwiązania optymalnego w wymaganym czasie, to należałoby w takich sytuacjach przewidzieć możliwość wyznaczenia sterowania zastępczego S_z dla obiektu.

System sterowania z uczeniem /rys. 3.3/ charakteryzuje się możliwością bieżącej kontroli i bieżącego wyznaczania wektora sterowania S . Sterowanie podawane na obiekt są wyznaczane przez algorytm decyzyjny [80] /funkcjonujący "analogicznie" jak algorytm rozpoznawania/ na podstawie pomiaru pewnych charakterystycznych cech zadania produkcyjnego c [80]. Funkcjonowanie opisanego systemu sterowania można podzielić na dwie fazy. Faza I - na podstawie pomiaru parametrów p_n ciągu rzeczywistych zadań produkcyjnych wyznacza się ciąg uczący w_n , a następnie przeprowadza się proces "uczenia" algorytmu decyzyjnego z algorytmem optymalizacyjnym jako trenerem. Faza II - na podstawie pomiaru parametrów p zadania produkcyjnego algorytm decyzyjny wyznacza sterowanie S . Zaproponowana koncepcja systemu sterowania umożliwia stosowanie dowolnie złożonych /czasochłonnych/ algorytmów optymalizacyjnych w systemach uwarunkowanych czasowo.

Dwupoziomowy system sterowania z wyborem decydenta na górnym poziomie /rys. 3.4/. Przypuśćmy, że dla wyznaczenia sterowania optymalnego pewnego dyskretnego procesu produkcyjnego możemy użyć jednego z dostępnych algorytmów różniących się między sobą pewnymi własnościami /np. szybkością działania/. Dodatkowo założymy, że własności algorytmów zależą nie tylko od typu zadania, lecz także od wartości liczbowych parametrów zadania /wszystkie algorytmy oparte na metodzie podziału i ograniczeń tym się charakteryzują/. Podczas realizacji sterowania w czasie rzeczywistym należy wybrać taki algorytm optymalizacyjny, który dla danego wektora parametrów p zadania produkcyjnego "gwarantuje" najszybsze wyznaczenie sterowania optymalnego. Wybór ten realizuje algorytm decyzyjny górnego poziomu.



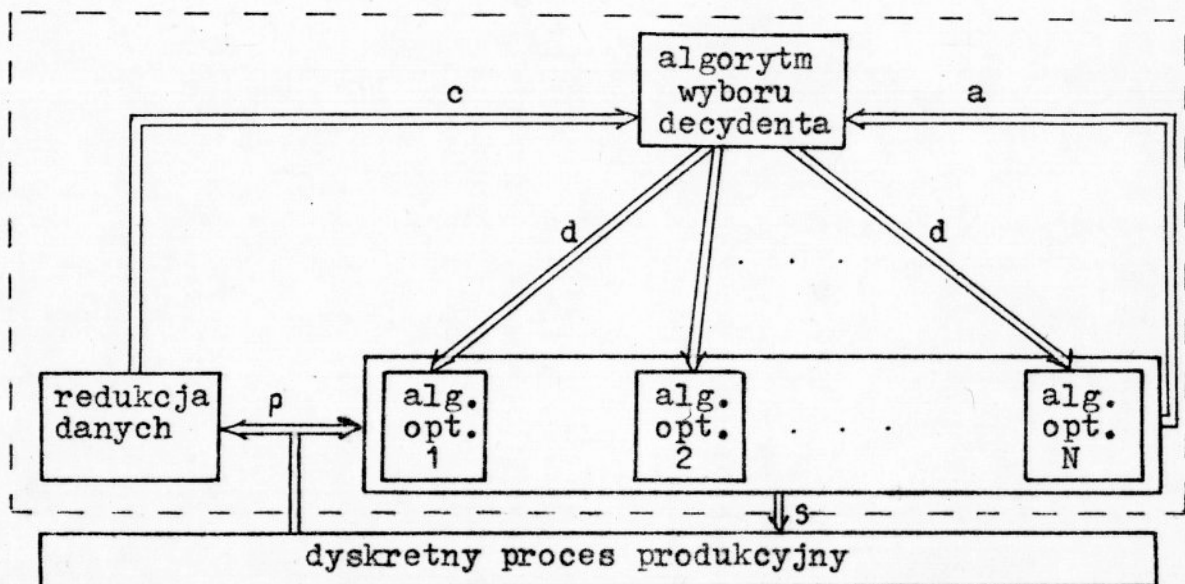
Rys. 3.3. System sterowania z uczeniem

/p - parametry zadania produkcyjnego,

c - wektor cech zadania produkcyjnego,

w_n - ciąg wzorcowych zadań produkcyjnych,

s - sterowanie procesu/



Rys. 3.4. Dwupoziomowy system sterowania z wyborem decydenta

na górnym poziomie /p - parametry zadania produkcyjnego, c - wektor cech zadania produkcyjnego, a - wektor cech algorytmów, d - wybór decydenta, s - sterowanie procesu/

4. AKTUALNY STAN LITERATURY. KLASYFIKACJA ZAGADNIEN KOLEJNOSCIOWYCH

W ostatnim dwudziestoleciu w literaturze światowej pojawiło się wiele prac zawierających sformułowania i rozwiązania nowych problemów szeregowania. Dotyczą one problemów o różnym poziomie ogólności. W celu ich rozróżnienia, w literaturze używa się symbolicznego zapisu. Przedstawiono go w pracy [26], a następnie rozszerzono w pracach [82] oraz [46]. Kolejnych rozszerzeń dokonano w pracach [43] oraz [16]. W niniejszej pracy używana jest symbolika zgodna z przyjętą w pracy [43].

Szybko rozwijająca się dziedzina szeregowania zadań obejmuje swoim zakresem coraz większą klasę praktycznych przypadków przy jednoczesnym rozpatrywaniu coraz ogólniejszych modeli problemów szeregowania, adekwatnych do złożonych sytuacji praktycznych. Realizowane uogólnienia zmierzają zarówno w kierunku uwzględniania coraz większej liczby ograniczeń /ogr. zasobowe, czasowe/ [25], [16], jak i w kierunku uogólnienia rozważanych funkcji celu /czas, koszt maksymalny, koszt całkowity/ [40], [97], [71], Kolejnymi uogólnieniami są podejścia wielokryterialne [88][29].

Istniejące różnice metodologiczne /szczególnie w przypadku problemów NP-zupełnych/ powodują rozproszenie wyników badań oraz rozwiązywanie szeregu szczególnych przypadków ogólnych problemów szeregowania, częstokroć dawno rozwiązanych. Stąd uzasadnione są próby poszukiwania ogólnej metody rozwiązywania szerokiej klasy zagadnień szeregowania.

Prezentowana praca stanowi próbę wprowadzenia dwóch jednolitych metod do rozwiązywania dwóch ogólnych klas zagadnień szeregowania.

Prawie wszystkie prace z klasycznej teorii szeregowania zadań [56], [55], [26] oraz szereg prac aktualnych [71], [43], [12] koncentrują się na problemach z kryterium minimalno-czasowym - $C_{\max}$. Jednakże większość problemów praktycznych wymaga przyjęcia kryterium w postaci pewnej funkcji zależnej od momentów zakończenia poszczególnych zadań. Funkcje te posiadają określone własności i reprezentują koszt związany z realizacją poszczególnych zadań. Tak określone problemy noszą nazwę problemów szeregowania

z funkcją kosztów lub karą /deferral cost, cumulative cost, penalties/.

Dla pewnych szczególnych zagadnień tego typu były formułowane w literaturze algorytmy /patrz tabele 4.1 - 4.3 ; kryteria $\sum C_i$, $\sum w_i C_i$, $\sum T_i$, $\sum w_i T_i$ i inne/.

Analiza praktycznych sytuacji w systemach produkcyjnych doprowadziła do wyodrębnienia następujących grup kosztów [30] .

- koszt oczekiwania zadania /job waiting cost/. Jest to koszt związany z oczekiwaniem na realizację lub wstrzymaniem realizacji zadania /także koszt zapasu/. Przyjmuje się, że opłata za jednostkę czasu oczekiwania zadania jest proporcjonalna do wielkości /wartości/ zadania.
- koszt przestoju maszyn /machine idle cost/. Jest to koszt występujący opcjonalnie. Uwzględnia on fakt, że maszyny są nieobciążone w różnych przedziałach czasu, w których mogłoby wykonywać inne czynności. Zwykle zakłada się, że wszystkie maszyny są dostępne od pewnego momentu początkowego /zerowego/ aż do czasu zakończenia ostatniego zadania /możliwe są wyjątki [35] /. Jest to nierzeczywiste dla szeregowania wsadowego, gdzie zadania oczekują w kolejce.
- koszty kar /job penalty costs/. Najtrudniejszym do ilościowego określenia jest opłata z powodu nieterminowej realizacji zadań ponieważ jest zależna od stosunków klient - dostawca. Gere [38] proponuje, aby w koszt spóźnienia były włączone: kary umowne w klauzulach kontraktu, koszty ekspedycji zadań spóźnionych lub przechowywania zadań przyspieszonych oraz straty w interesach spowodowane niezadowoleniem klientów. Mc Naughton [76] proponuje użyć kwadratowej funkcji strat ponieważ pilność realizowanego zadania silnie rośnie po przekroczeniu żadanego terminu zakończenia. Lawler [70] proponuje funkcję monotonicznie niemalejącą. Chociaż nierozsądnym byłoby zakładać, że wielkość strat maleje ze wzrostem spóźnienia, to wydaje się, że przyjęte założenia ogranicza klasę zagadnień. Sidney [86] proponuje uwzględnienie kosztów przyspieszenia zadań /np. przyspieszenie ekspedycji, przechowywanie zadań przyspieszonych, straty w surowcu na skutek magazynowania, przyspieszenie czynności przygotowawczych przed realizacją zadania inwestycyjnego/.

zadania inwestycyjnego/.

Całkowity koszt realizacji sekwencji zadań jest zwykle wielkością maksymalną [2] , [67] lub sumą wielkości kosztów [6] , [71] , oczekiwania zadań, przestoju maszyn i kar przy założeniu, że:

- (i) koszty wykonywania i czasy realizacji zadań są niezależne od sekwencji,
- (ii) koszty oczekiwania są proporcjonalne do wielkości zadania /ilości produktu/ w każdym momencie czasu,
- (iii) zadania zakończone przed swoimi żądanymi terminami zakończenia są przechowywane do momentu ich upływu,
- (iv) koszt przestoju maszyn jest wyliczany w przedziale $[0, C_{\max}]$,
- (v) koszty kar są wielomianową funkcją spóźnienia /przyspieszenia/.

W dalszym ciągu będziemy się posługiwać następującymi oznaczeniami [46] i [43] :

J - zbiór zadań $J = \{J_1, J_2, \dots, J_n\}$,

M - zbiór maszyn $M = \{M_1, M_2, \dots, M_m\}$,

p_{ik} - czas realizacji zadania J_i na maszynie M_k ,

S_i - termin rozpoczęcia wykonywania zadania J_i ,

C_i - termin zakończenia wykonywania zadania J_i ,

r_i - najwcześniej pożądaný termin rozpoczęcia zadania J_i ,

d_i - pożądaný termin zakończenia zadania J_i ,

$L_i \triangleq C_i - d_i$ nieterminowość wykonania zadania,

$T_i \triangleq \max [0, L_i]$ spóźnienie wykonania zadania,

$E_i \triangleq \max [0, r_i - S_i]$ przyspieszenie wykonania zadania,

$g_i(t)$, $f_i(t)$ - funkcje kosztu; niemalejące funkcje argumentu t .

Uwzględnienie wszystkich rodzajów kosztów doprowadziło do zdefiniowania następującego zbioru funkcji celu występujących najczęściej w praktyce [82] , [25] :

$$C_{\max} \triangleq \max_{1 \leq i \leq n} C_i \text{ /completion time, makespan [73] /,$$

$$\sum w_i C_i \triangleq \sum_{i=1}^n w_i C_i \text{ /defferal cost [69] , [87] /,$$

$$L_{\max} \triangleq \max_{1 \leq i \leq n} L_i \text{ /maksimum lateness [74] , [45] , [64] /,$$

$$\sum w_i T_i \triangleq \sum_{i=1}^n w_i T_i \text{ /weighted tardiness [85] , [79] /;$$

$$f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(C_i) \text{ /maximum penalty cost [4] /,$$

$$h_{\max} \triangleq \max \left[\max_{1 \leq i \leq n} g_i(E_i) , \max_{1 \leq i \leq n} f_i(T_i) \right] \text{ /maximum cost with earliness and tardiness penalties [86], [2],/}$$

$$\sum f_i \triangleq \sum_{i=1}^n f_i(C_i) \text{ /total cost [71] /,$$

$$\sum h_i \triangleq \sum_{i=1}^n [g_i(E_i) + f_i(T_i)] \text{ /total cost with earliness and tardiness penalties/.$$

Analogiczne klasy funkcji kryterialnych można określić w przypadku gdy koszt jest związany z realizacją poszczególnych operacji [43] . Z rezultatów zawartych w pracach [71] , [46] wynika, że podane kryteria zawierają /ze względu na równoważność / prawie wszystkie pozostałe kryteria /weighted machine idle time, machine utilization, flow time, mean flow time, maximum tardiness/. Zatem wystarczy ograniczyć rozważania do problemów z podanymi funkcjami celu. Bardziej precyzyjnie mówiąc wystarczy rozważania ograniczyć do problemów z funkcją celu $f_{\max}$ lub $h_{\max}$ /maximum penalty cost/ oraz z funkcją celu $\sum f_i$ lub $\sum h_i$ /total penalty cost/. To spostrzeżenie stało się podstawą wprowadzania przez autora jednolitych ogólnych technik rozwiązywania podanych klas zagadnień.

Pełny przegląd problemów szeregowania przedstawiono w tabl. 4.1 - 4.3 przy użyciu symbolicznego zapisu [71] , [43] , [46] . W wykazie ujęto tylko te zagadnienia, dla których skonstruowano algorytm znajdujący rozwiązanie optymalne /nie rozwiązanie przybliżone/.

Tablica 4.1.

Zagadnienia postaci $n|1|\gamma|\delta$

		Autor algorytmu	Złożoność	Źródło
$C_{\max}$	$r_i \geq 0$	Jackson [1955]	$O(n \log n)$	[55]
	$r_i \geq 0, \prec$	Lawler [1973]	$O(n^2)$	[70]
$L_{\max}$	-	Jackson [1955]	$O(n \log n)$	[55]
	$r_i \geq 0, p_i = 1$	Horn [1974]	$O(n^2)$	[53]
	$r_i \geq 0, p_i = 1, \prec$	Lageweg i inni [1976]	$O(n^2)$	[64]
	$r_i \geq 0$	Mc Mahon, Florian [1975]	NP	[74]
	$r_i \geq 0, \prec$	Lageweg i inni [1976]	NP	[64]
		Lawler [1973]	$O(n^2)$	[70]
$f_{\max}$	$\prec$	Lawler [1973]	$O(n^2)$	[70]
$\sum w_i C_i$	-	Smith [1956]	$O(n \log n)$	[91]
	drzewo	Horn [1972]	$O(n \log n)$	[52]
	graf szeregowo-równoległy	Sidney [1975]	$O(n^3)$	[87]
	$r_i \geq 0, p_i = 1$	Rinnooy [1976]	$O(n \log n)$	[82]
$\sum w_i T_i$	-	Shwimer [1972]	NP	[85]
	$r_i \geq 0, p_i = 1$	Lawler [1964]	$O(n^3)$	[69]
$\sum f_i$	-	Schild, Fredman [1962]	NP	[83]
	$\prec$	Lenstra [1977]	NP	[71]
	$f_i(t) = a_i \varphi(t) + b_i$	Lawler, Sivazlian [1978]	$O(n \log n)$	[67]
$h_{\max}$	$r_i \leq r_j \Rightarrow d_i \leq d_j$ $g_i(t) = g(t), f_i(t) = f(t)$	Sidney [1977]	$O(n \log n)$	[86]
	$\prec$	Smutnicki	NP	(1)
$\sum h_i$	$\prec$	Smutnicki	NP	(1)

Tablica 4.2.

Zagadnienia postaci $n|m|\gamma, r|\delta$

δ	γ, r	Autor algorytmu	Złożoność	Zródło
$C_{\max}$	$F, m = 2$	Johnson [1954]	$O(n \log n)$	[56]
	$F, m = 2$, bez czekania	Gilmore, Gomory [1964]	$O(n^2)$	[39]
	P	Brown, Lomnicki [1966]	NP	[17]
	$P(F), \prec$	Grabowski [1978]	NP	[44]
	P , bez czekania	Grabowski, Sysko [1972]	NP	[47]
	G	Greenberg [1968]	NP	[48]
	G, RT	Balas [1969]	NP	[11]
	G, RT , bez magazynu	Grabowski [1975]		[42]
$L_{\max}$	$P(F), r_i \geq 0, \prec$	Grabowski [1979]	NP	[45]
$f_{\max}$	P	Townsend	NP	[97]
	P	Smutnicki	NP	(1)
	$F, G, \prec$		NP	(2)
$\sum w_i C_i$	$F, w_i = 1, m = 2$	Kohler, Steiglitz [1975]	NP	[58]
	-	-		-
$\sum w_i T_i$		-		-
$\sum f_i$	$F, m = 2$	Rinnooy [1976]	NP	[82]
	-	-		-
$h_{\max}$	$P, g_i(t) = g(t), f_i(t) = f(t)$	Achuthan, Grabowski, Sidney [1978]	NP	[2]
	$P, \prec$	Smutnicki	NP	(1)
	$F, G, \prec$	Smutnicki	NP	(2)
$\sum h_i$		-		-

Tablica 4.3

Zagadnienia z równoległym wykorzystaniem maszyn $n|m|J, \Gamma|\delta$

δ	m	Γ	Autor algorytmu	Złożoność	Źródło
$C_{\max}$	2	$\tau = 1, r_j \geq 0, p_{jw} = 1, RT$	Garey, Johnson [1975]	$O(n^3)$	[33]
t_z	m	$\tau = 1, m_w^k = 1, p_{jw} = p_j$ bez magazynu, RT	Grabowski [1973]	?	[40]
	m	$\tau = 1, m_w^k = 1, p_{jw} = p_j, RT$	Nabeshima [1973]	NP	[78]
	m	$\tau = 1, m_w^k \geq 1, RT$	Grabowski [1977]	NP	[40]
$L_{\max}$	m	$\tau = 1, m_w^k \geq 1, r_j \geq 0, RT$	Grabowski [1979]	NP	[43]
$f_{\max}$	m	$\tau = 1, m_w^k \geq 1, r_j \geq 0, RT$		NP	(2)
$\sum w_i C_i$ $\sum w_j t_j$			-		-
$\sum w_i \tau_i$			-		-
$\sum f_i$			-		-
$h_{\max}$	m	$\tau = 1, m_w^k \geq 1, RT$			(2)
$\sum h_i$	m	$\tau = 1, m_w^k \geq 1, RT$	Smutnicki	NP	(1)

(1) Zagadnienia te są przedmiotem niniejszej pracy

(2) Zagadnienia te zostały rozwiązane przez uogólnienie rezultatów pracy /rozdz. 7/, jednakże ich szczegółowa prezentacja przekracza objętość pracy.

5. MINIMALIZACJA KOSZTU MAKSYMALNEGO

Zagadnienie $n|m|F, r_i \geq 0, \sum f_{\max}$

Zagadnienie kolejnościowe taśmowe sformułowane jest następująco. Dany jest zbiór zadań $J = \{J_1, J_2, \dots, J_n\}$. Każde zadanie J_i $/i = 1, 2, \dots, n/$ składa się z ciągu operacji $\langle O_{i1}, O_{i2}, \dots, O_{im} \rangle$ realizowanych kolejno na maszynach $M_1, M_2, \dots, M_m$. Czas realizacji operacji O_{ik} na maszynie M_k wynosi p_{ik} $/i = 1, 2, \dots, n, k = 1, 2, \dots, m/$. Każda z maszyn może realizować co najwyżej jedno zadanie w dowolnej chwili. Zagadnienie optymalizacji polega na ustaleniu takiej kolejności czasowej wykonywania zadań na poszczególnych maszynach aby przyjęta funkcja celu osiągała wartość optymalną. W przypadku wprowadzenia dodatkowego wymagania, aby kolejność wykonywania zadań na wszystkich maszynach była jednakowa, otrzymujemy permutacyjne zagadnienie taśmowe oznaczone $n|m|P, \Gamma|\delta$.

Zagadnienia kolejnościowe tego rodzaju mogą wystąpić w procesach walcowania [13], wytopu i odlewu stali [24], w produkcji taśmowej zespołów i podzespołów [23], a także w produkcji detali [14]. Zagadnienia tego rodzaju mogą pojawić się również w systemach cyfrowych [25].

W literaturze najczęściej spotyka się rozwiązanie zagadnień taśmowych z funkcją $C_{\max}$ [56], [73], [47].

W pracy [44] autor przedstawił algorytm rozwiązania problemu z funkcją celu $L_{\max}$. W pracy [97] autor przedstawił rozwiązanie problemu $n|m|P | f_{\max}$ przy wykorzystaniu rezultatów Lawlera i Łomnickiego [69], [73].

W ciągu ostatnich lat najwięcej uwagi poświęcono zagadnieniu $n|m|P|C_{\max}$. Pierwszy algorytm tego rozwiązania podano w pracy [56]. Większość późniejszych prac [10], [12], [15] koncentrowała się na wykazaniu nowych własności problemu oraz na konstrukcji silniejszych danych ograniczeń dla metody podziału i ograniczeń. W pracach [63], [71], [82] autorzy przedstawili elegancką koncepcję konstruowania dolnych ograniczeń problemu $n|m|F|C_{\max}$. Koncepcja ta została rozszerzona następnie w pracy [43]. Koncepcje te zawierają w sobie wszystkie znane w literaturze metody wyznaczania dolnych ograniczeń problemu $n|m|F|C_{\max}$.

Celem niniejszego rozdziału jest rozwiązanie zagadnienia postaci $n|m|P, r_i \geq 0, \prec | f_{\max}$. Podana metoda rozwiązania zagadnienia $n|m|P, r_i \geq 0, \prec | f_{\max}$ jest ilustracyjnym przykładem zaproponowanego przez autora ogólnego schematu rozwiązania zagadnień kolejnościowych z minimaxową funkcją celu.

Podstawowymi elementami tej metody są: schemat metody podziału i ograniczeń oraz eliminacyjne właściwości ciągu krytycznego /pojęcie to jest wyjaśnione szczegółowo w rozdziałach 3.1, 6.1, 7/. Zastosowanie tego typu podejścia doprowadziło do uzyskania bardzo dobrych rezultatów obliczeniowych. Proces wykonywania zadań dla zagadnienia $n|m|P, r_i \geq 0, \prec | f_{\max}$ jest taki sam jak dla zagadnienia $n|m|F, \prec | f_{\max}$. Dodatkowo występuje wymaganie, że każde zadanie J_i / $i = 1, 2, \dots, n$ / nie może się rozpocząć wcześniej niż jego najwcześniejszy możliwy termin rozpoczęcia r_i . Zakończenie realizacji zadania w terminie C_i pociąga za sobą koszt realizacji zadania w wysokości $f_i(C_i)$, gdzie $f_i(C_i)$ - niemalejąca funkcja kosztu realizacji zadania $i = 1, 2, \dots, n$. Jeżeli kolejność wykonywania zadań na poszczególnych maszynach jest określone, to dla każdego zadania J_i możemy wyznaczyć terminy rozpoczęcia wykonywania S_{ik} oraz zakończenie wykonywania C_{ik} operacji O_{ik} na maszynie M_k , przy czym oczywiście zachodzi $r_i \leq S_{i1}$. Zagadnienie optymalizacyjne polega na ustaleniu takiej kolejności wykonywania zadań na poszczególnych maszynach, aby koszt maksymalny $f_{\max} = \max_{1 \leq i \leq n} f_i(C_i)$ osiągnął wartość minimalną.

Zagadnienie $n|m|P, r_i \geq 0, \prec | f_{\max}$ jest NP-zupełne [72], co usprawiedliwia stosowanie metody podziału i ograniczeń do jego rozwiązania.

Należy zaznaczyć, że przyjmując $r_i = 0$ oraz $f_i(C_i) = C_i$ dla każdego zadania J_i , otrzymujemy zagadnienie postaci $n|m|F|C_{\max}$ [63], zaś przyjmując $f_i(C_i) = C_i - d_i$ otrzymujemy zagadnienie postaci $n|m|F, r_i \geq 0|L_{\max}$ [43], zaś przyjmując $r_i = 0$ otrzymujemy $n|m|F|f_{\max}$ [97]. Dlatego też przedstawiany w dalszej części pracy algorytm zagadnienia $n|m|F, r_i \geq 0|f_{\max}$ może rozwiązywać również podane zagadnienie jako szczególne przypadki.

5.1. Pewne własności zagadnienia $n|m|P, r_i \geq 0, \forall i, f_{\max}$

Niech $\pi = \langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ będzie dowolną permutacją wskaźników zadań $1, 2, \dots, n$. Niech Π będzie zbiorem wszystkich tych permutacji. Jeśli pośród zadań określono dodatkowo relację poprzedzania $\mathcal{R}$ to niech $\Pi(\mathcal{R})$ będzie zbiorem dopuszczalnych permutacji /dla danej relacji $\mathcal{R}$ /, tzn. jeśli $\langle \pi(i), \pi(j) \rangle \in \mathcal{R}$, to $i < j$, $i, j = 1, 2, \dots, n$, $i \neq j$. Każda permutacja $\pi \in \Pi(\mathcal{R})$ określa dopuszczalną kolejność wykonywania zadań $J_{\pi(1)}, J_{\pi(2)}, \dots, J_{\pi(n)}$ na każdej maszynie. Zagadnienie optymalizacji polega na znalezieniu takiej permutacji π^* , dla której mamy

$$f_{\max}(\pi^*) = \min_{\pi \in \Pi(\mathcal{R})} f_{\max}(\pi), \quad (5.1)$$

gdzie

$$f_{\max}(\pi) = \max_{1 \leq j \leq n} f_{\pi(j)}(C_{\pi(j)m}). \quad (5.2)$$

Twierdzenie 5.1.

Niech $\pi \in \Pi(\mathcal{R})$ będzie dowolną permutacją dopuszczalną określającą kolejność wykonywania zadań na maszynach oraz niech $f_i(C_i)$, $i = 1, 2, \dots, n$ będzie zbiorem funkcji niemalejących, wtedy otrzymujemy

$$f_{\max}(\pi) = \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{m+1} \leq n} f_{\pi(i_{m+1})} \left(r_{\pi(i_1)} + \sum_{i=i_1}^{i_2} p_{\pi(i)1} + \sum_{i=i_2}^{i_3} p_{\pi(i)2} + \dots + \sum_{i=i_m}^{i_{m+1}} p_{\pi(i)m} \right). \quad (5.3)$$

Dowód

Każda permutacja $\pi \in \Pi(\mathcal{R})$ określa terminy zakończenia $C_{\pi(j)k}$ zadania $J_{\pi(j)}$ na maszynie M_k $j = 1, 2, \dots, n$, $k = 1, 2, \dots, m$. Z dowodu tw. 3.1 w pracy [43] wynika, że

$$C_{\pi(j)k} = \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_k \leq j} \left(r_{\pi(i_1)} + \sum_{i=i_1}^{i_2} p_{\pi(i)1} + \sum_{i=i_2}^{i_3} p_{\pi(i)2} + \dots + \sum_{i=i_k}^j p_{\pi(i)k} \right) \quad (5.4)$$

wstawiając zatem do wyrażenia (5.2) zależność (5.4) dla $k = m$, przy założeniu, że $f_i(t)$ są niemalejące, otrzymujemy

$$\begin{aligned} f_{\max}(\pi) &= \max_{1 \leq j \leq n} f_{\pi(j)}(C_{\pi(j)m}) = \\ &= \max_{1 \leq j \leq n} f_{\pi(j)} \left(\max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq j} \left(r_{\pi(i_1)} + \sum_{i=i_1}^{i_2} p_{\pi(i)1} + \sum_{i=i_2}^{i_3} p_{\pi(i)2} + \dots + \right. \right. \end{aligned}$$

$$\begin{aligned}
& + \dots + \sum_{i=i_m}^j P_{\pi(i)m}) = \max_{1 \leq j \leq n} \left(\max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq j} f_{\pi(j)}(r_{\pi(i_1)} + \right. \\
& + \sum_{i=i_1}^{i_2} P_{\pi(i)1} + \sum_{i=i_2}^{i_3} P_{\pi(i)2} + \dots + \sum_{i=i_m}^j P_{\pi(i)m}) \Big) = \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq i_{m+1} \leq n} \\
& f_{\pi(i_{m+1})}(r_{\pi(i_1)} + \sum_{i=i_1}^{i_2} P_{\pi(i)1} + \sum_{i=i_2}^{i_3} P_{\pi(i)2} + \dots + \sum_{i=i_m}^{i_{m+1}} P_{\pi(i)m})). \quad (5.5)
\end{aligned}$$

Ostatnia równość we wzorze (5.5) została zapisana po przyjęciu $i_{m+1} \triangleq j$.

Ciąg liczb $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ spełniających warunek $1 \leq u_1^\pi \leq u_2^\pi \leq \dots \leq u_{m+1}^\pi \leq n$ oraz

$$f_{\max}(\pi) = f_{\pi(u_{m+1}^\pi)}(r_{\pi(u_1^\pi)} + \sum_{i=u_1^\pi}^{u_2^\pi} P_{\pi(i)1} + \sum_{i=u_2^\pi}^{u_3^\pi} P_{\pi(i)2} + \dots + \sum_{i=u_m^\pi}^{u_{m+1}^\pi} P_{\pi(i)m}). \quad (5.6)$$

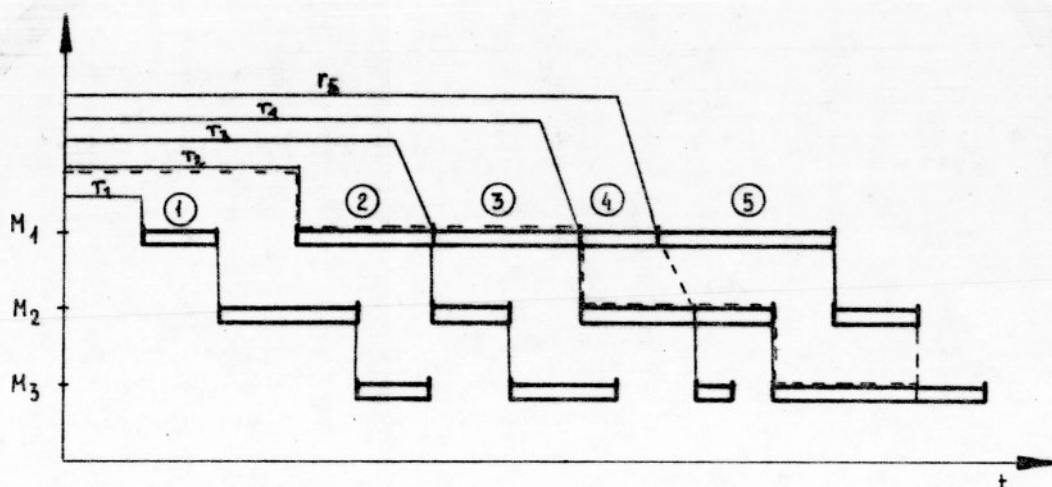
będziemy nazywać ciągiem krytycznym. Z zależności (5.3) wynika, że dla każdej mamy

$$\begin{aligned}
f_{\max}(\pi) \geq \hat{f}_{\max}(\pi) = f_{\pi(u_{m+1}^\pi)}(r_{\pi(u_1)} + \sum_{i=\hat{u}_1}^{\hat{u}_2} P_{\pi(i)1} + \sum_{i=\hat{u}_2}^{\hat{u}_3} P_{\pi(i)2} + \dots + \\
+ \dots + \sum_{i=\hat{u}_m}^{\hat{u}_{m+1}} P_{\pi(i)m}) \quad (5.7)
\end{aligned}$$

gdzie $\langle \hat{u}_1, \hat{u}_2, \dots, \hat{u}_{m+1} \rangle$ jest dowolnym ciągiem, niekoniecznie krytycznym, w π . Wartość $\hat{f}_{\max}(\pi)$ jest kosztem związanym z tym ciągiem /dokładniej kosztem pracy $J_{\pi(\hat{u}_{m+1})}$, gdzie $\hat{u}_{m+1}$ jest ostatnim elementem ciągu/.

Zdefiniowane zostanie dalej, dla każdej permutacji π , kilka dodatkowych pojęć.

Ciąg zadań $\langle J_{\pi(u_k)}, J_{\pi(u_{k+1})}, \dots, J_{\pi(u_{k+1})} \rangle$ będziemy nazywać k-tym blokiem w π , $k = 1, 2, \dots, m$. Blok k-ty odpowiada ciągowi zadań realizowanych na maszynie k-tej bez przerw pomiędzy zadaniami /rys. 5.1/.



Rys. 5.1. Wykres Gantta dla problemu $n/m/P$, $r_i \geq 0 / f_{\max}$

Dla przykładu przedstawionego na rys. 5.1 mamy $f_{\max} = \max_{1 \leq i \leq n} (w_i C_i)$

$w_1 = w_2 = w_3 = w_5 = 1$, $w_4 = 10$, oraz $u_1^\pi = 2$, $u_2^\pi = 3$, $u_3^\pi = 4$, $u_4^\pi = 4$. Ciągi zadań $\langle J_2, J_3 \rangle$, $\langle J_3, J_4 \rangle$, $\langle J_4 \rangle$ są odpowiednio blokami na maszynie 1-szej, 2-giej, 3-ciej.

Zadanie $J_{\pi(u_k)}$ jest pierwszym zadaniem w k-tym bloku /i ostatnim zadaniem w k-1-szym bloku/. Zadanie $J_{\pi(u_{k+1})}$ jest ostatnim zadaniem w k-tym bloku /i pierwszym zadaniem w k+1-szym bloku/. Ciąg częściowy zadań $\langle J_{\pi(u_{k+1})}, J_{\pi(u_{k+2})}, \dots, J_{\pi(u_{k+1}-1)} \rangle$

będziemy nazywać k-tym wewnętrznym blokiem. Niech

$$P_k^\pi = \{ \pi(j) \mid u_k^\pi \leq j \leq u_{k+1}^\pi \} \quad (5.8)$$

będzie zbiorem wskaźników zadań k-tego bloku w π . W dalszym ciągu, dla uproszczenia zapisu, wskaźnik będzie opuszczany w oznaczeniach u_k^π , itd.

Twierdzenie 5.2.

Niech $\pi \in \Pi$ będzie dowolną permutacją określającą kolejność wykonywania zadań oraz niech $\langle u_1, u_2, \dots, u_{m+1} \rangle$ będzie ciągiem krytycznym w tej permutacji. Jeżeli istnieje permutacja $\beta \in \Pi$ taka,

że $f_{\max}(\beta) < f_{\max}(\pi)$ wtedy w β przynajmniej jedno zadanie dla przynajmniej jednego bloku z π poprzedza pierwsze zadanie lub występuje za ostatnim zadaniem tego bloku. ■

Dowód

Bez zmniejszania ogólności możemy założyć, że $\pi(i) = i$, $i = 1, 2, \dots, n$. Niech $\langle u_1, u_2, \dots, u_{m+1} \rangle$ będzie ciągiem krytycznym w π , $\pi(u_k) = u_k$, $k = 1, 2, \dots, m+1$.

Dowód przeprowadzony zostanie przez zaprzeczenie. Niech $\beta \in \Pi$ będzie permutacją taką, że $f_{\max}(\beta) < f_{\max}(\pi)$ oraz teza twierdzenia nie jest spełniona. Oznacza to, że β jest permutacją postaci

$$\beta = \langle \beta(1), \beta(2), \dots, \beta(u_1-1), u_1, \beta(u_1+1), \beta(u_1+2), \dots, \beta(u_2-1), u_2, \\ \beta(u_2+1), \dots, \beta(u_3-1), u_3, \dots, u_k, \dots, u_{k+1}, \dots, u_{m+1}, \beta(u_{m+1}+1), \\ \dots, \beta(n) \rangle,$$

gdzie $\beta u_k = u_k$, $k = 1, 2, \dots, m+1$ oraz $\langle \beta(u_k+1), \beta(u_k+2), \dots, \dots, \beta(u_{k+1}-1) \rangle$ $k = 1, 2, \dots, m$ jest dowolną permutacją zadań k -tego wewnętrznego bloku, tzn. permutację zadań ze wskaźnikami $u_k+1, u_k+2, \dots, u_{k+1}-1$ zaś $\langle \beta(1), \dots, \beta(u_1-1) \rangle$, $\langle \beta(u_{m+1}+1), \dots, \beta(n) \rangle$ są odpowiednio permutacjami zadań ze wskaźnikami $1, 2, \dots, u_1-1$ i $u_{m+1}+1, \dots, n$.

Ciąg krytyczny w π $\langle u_1, u_2, \dots, u_{m+1} \rangle$ jest pewnym ciągiem /niekoniecznie krytycznym/ w β , możemy więc zapisać

$$f_{\max}(\beta) \geq \hat{f}_{\max}(\beta) = f_{u_{m+1}}(r_{u_1} + \sum_{i=u_1}^{u_2} P_{\beta(i)1} + \sum_{i=u_2}^{u_3} P_{\beta(i)2} + \dots + \sum_{i=u_m}^{u_{m+1}} P_{\beta(i)m}) \quad (5.9)$$

ale

$$\sum_{i=u_k}^{u_{k+1}} P_{\beta(i)k} = p_{u_k k} + \sum_{i=u_k+1}^{u_{k+1}-1} P_{\beta(i)k} + p_{u_{k+1} k} = p_{u_k k} + \sum_{i=u_k+1}^{u_{k+1}-1} p_{ik} + \\ + p_{u_{k+1} k} = p_{u_k k} + \sum_{i=u_k+1}^{u_{k+1}-1} P_{\pi(i)k} + p_{u_{k+1} k}, \quad (5.10)$$

bowiem $\sum_{i=u_k+1}^{u_{k+1}-1} P_{\beta(i)n}$ jest wielkością stałą, niezależną od kolejności sumowania i równą $\sum_{i=u_k+1}^{u_{k+1}-1} p_{ik}$. W konsekwencji otrzymujemy

$$f_{\max}(\beta) \geq \hat{f}_{\max}(\beta) = f_{\max}(\pi) = f_{u_{m+1}}(r_{u_1} + \sum_{i=u_1}^{u_2} p_{\pi(i)_1} + \sum_{i=u_2}^{u_3} p_{\pi(i)_2} + \dots + \\ + \dots + \sum_{i=u_m}^{u_{m+1}} p_{\pi(i)_m} \quad , \quad (5.11)$$

co przeczy założeniu i kończy dowód twierdzenia. ■

Z twierdzenia wynika, że jeśli permutacja β została otrzymana z permutacji π przez zmianę kolejności wykonywania zadań w π oraz jeżeli $f_{\max}(\beta) < f_{\max}(\pi)$, wtedy w przynajmniej jedno zadanie dla przynajmniej jednego bloku z π poprzedza pierwsze zadanie /lub następuje za ostatnim zadaniem/ tego bloku. Wniosek ten stał się podstawą konstrukcji algorytmu opartego na metodzie podziału i ograniczeń.

Dla przedstawionego problemu prawdziwe są twierdzenia 3.3 oraz wnioski 3.2, 3.3, 3.4 z pracy [43] .

5.2. Schemat przebiegu algorytmu [45]

Zasada działania algorytmu jest oparta na metodzie podziału i ograniczeń z mieszaną strategią podziału. W celu pełnego wykorzystania eliminacyjnych własności ciągu krytycznego zastosowano specjalną technikę przeglądania drzewa rozwiązań, wprowadzoną przez Grabowskiego [43] . Opisany w tym rozdziale schemat przebiegu algorytmu nie różni się istotnie od schematu podanego w pracy [45], jednakże jego zamieszczenie jest pożądane dla pełnego scharakteryzowania proponowanego algorytmu.

Rozpoczynając od permutacji początkowej $\pi^0 \in \Pi(\mathcal{R})$ generujemy ciąg permutacji $\pi \in \Pi(\mathcal{R})$. Dla każdej permutacji z ciągu obliczamy wartość $f_{\max}(\pi)$, wyznaczamy ciąg krytyczny $\langle u_1, u_2, \dots, u_{m+1} \rangle$ oraz określamy bloki zadań. W przypadku, gdy permutacja ma więcej niż jeden ciąg krytyczny to bierzemy pod uwagę dowolny z nich. Każdą nową permutację $\beta \in \Pi(\mathcal{R})$ otrzymujemy z pewnej permutacji π z ciągu przez przesunięcie jednego zadania J_j z k -tego bloku w permutacji π . Oczywiście takie przesunięcie będzie realizowane tylko wtedy, gdy nowo otrzymana permutacja π będzie dopuszczalna, tzn. $\beta \in \Pi(\mathcal{R})$. Zadanie J_j z k -tego wewnętrznego bloku jest przesuwana w π na pozycję u_k /lub u_{k+1} /. Oznacza to, że w permutacji β zadanie to jest

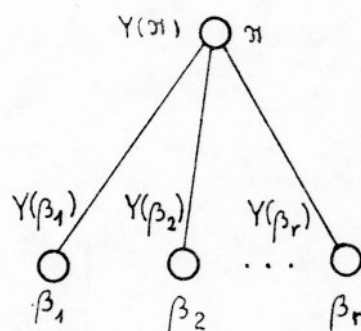
wykonywane jako pierwsze /lub ostatnie/ z wszystkich zadań k-tego bloku z π . Pierwsze /lub ostatnie/ zadanie k-tego bloku jest przesuwane na pozycję u_{k+1} /lub u_k / w π . Jeżeli będziemy mogli wykazać, że dla pewnej permutacji β przesuwanie prac nie poprawi wartości $f_{\max}(\beta)$ lub, że w wyniku przesuwania prac nie otrzymamy permutacji dopuszczalnej, wtedy odrzucamy rozpatrywaną permutację β i cofamy się do π , z której wygenerowaliśmy β . Proces generowania permutacji przedstawimy w postaci drzewa $H = \langle V, E \rangle$ gdzie V - zbiór węzłów drzewa reprezentujący wszystkie permutacje, E - zbiór gałęzi drzewa postaci $\langle \pi, \beta \rangle$ oraz β została otrzymana przez przesunięcie jednego zadania w π .

Wygenerowanie nowej permutacji /węzła H / wiąże się z wyborem pewnego zadania do przesunięcia w π . Wybór ten jest związany z zasadą podziału węzłów H . Jeżeli nową permutację β otrzymaliśmy przez przesunięcie w π zadanie J_j z k-tego bloku, ustalamy wymagania częściowego porządku

$$W_k^j = \{J_j \prec J_h \mid h \in P_k - \{j\}\} \quad \text{/lub } Z_k^j = \{J_h \prec J_j \mid h \in P_k - \{j\}\} \quad (5.14)$$

Wymagania te muszą być respektowane w każdym następniku β względem β w H .

Generowanie bezpośrednich następników β_r /rys. 5.2/ względem π powoduje, że zbiór rozwiązań dopuszczalnych odpowiadających węzłowi π zostaje rozbitý na podzbiory $Y(\beta_r)$ /niekoniecznie rozłączne/. Z każdym z podzbiorów $Y(\beta_r)$ jest związana relacja postaci (5.14). Niech β_1 będzie pierwszym następnikiem względem π wygenerowanym przez przesunięcie J_j na pierwszą pozycję k-tego bloku w π . Po dokonaniu cofania z β_1 do π otrzymujemy zdanie logiczne dopełniające, że J_j nie może być wykonywane jako pierwsze z rozważanego bloku. Dla zapewnienia rozłączności zbiorów rozwiązań $Y(\beta_r)$ $r = 1, 2, \dots$, zdanie to powinno być przestrzegane w kolejnych następnikach generowanych z π , tzn. we



Rys. 5.2. Drzewo rozwiązań

wszystkich β_r , $r \geq 2$. Trudności w realizacji tego typu wymagań przy użyciu relacji częściowego porządku zmuszają nas do zrezygno-

wania z rozłączności podzbiorów $Y(\beta_r)$. Dalej niech β_2 będzie kolejnym następnikiem względem π wygenerowanym przez przesunięcie J_j na ostatnią pozycję k -tego bloku w π . Wykonanie cofania z β_2 do π , w połączeniu ze zdaniem logicznym dopełniającym dla β_1 określa nowe zdanie logiczne dopełniające, że zadanie J_j nie może być wykonywane ani jako pierwsze ani jako ostatnie w k -tym bloku. Jeżeli wszystkie zadania k -tego bloku były przesuwane, to otrzymuje zdanie logiczne dopełniające, że żadne zadanie z k -tego bloku wewnętrznego nie może być wykonywane ani jako pierwsze, ani jako ostatnie, natomiast zadanie pierwsze tego bloku nie może być wykonywane jako ostatnie, a zadanie ostatnie jako pierwsze. Zdanie to można zrealizować w postaci wymagań częściowego porządku postaci

$$\{J_{\pi(u_k)} \prec J_h \mid h \in P_k - \{\pi(u_k)\}\} \cup \{J_h \prec J_{\pi(u_{k+1})} \mid h \in P_k - \{\pi(u_{k+1})\}\} \quad (5.15)$$

Jeżeli β w algorytmie jest otrzymywane przez przesunięcia zadania z k -tego bloku w π , to dla β ustalamy arbitralnie wymagania częściowego porządku zadań dla pierwszych $k-1$ bloków. Zapewnia to rozłączność następników /podzbiorów rozwiązań/ generowanych przez przesuwanie zadań w różnych blokach. W wyniku dokonanych ustaleń relacji częściowego porządku otrzymujemy dla każdej permutacji relację $\mathcal{R}_\pi$. Relację tą możemy również przedstawić w postaci zbiorów poprzedników i następników zadań

$$A_\pi^j = \{h \mid \langle J_j, J_h \rangle \in \mathcal{R}_\pi\}, \quad B_\pi^j = \{h \mid \langle J_h, J_j \rangle \in \mathcal{R}_\pi\}, \quad j = 1, 2, \dots, n \quad (5.16)$$

W procesie generowania permutacji, węzeł π w H możemy odrzucić, jeżeli

- (i) dolne ograniczenie wartości $f_{\max}(\beta)$ dla wszystkich następników w β wygenerowanych z π jest nie mniejsze niż aktualne górne ograniczenie,
- (ii) żadne z zadań nie może być przesuwane na pierwszą lub ostatnią pozycję swojego bloku bez naruszania relacji $\mathcal{R}_\pi$,
- (iii) każdy z bloków zawiera dokładnie jedno zadanie, tzn.

$$u_1 = u_2 = \dots = u_{m+1}.$$

Przypadki (ii), (iii) wynikają z twierdzenia 5.2.

5.3. Wyznaczanie dolnych ograniczeń

Dla każdego węzła π należy znaleźć dolne ograniczenie wartości $f_{\max}(\beta)$ dla wszystkich następników β , które można wygenerować z π . Problem ten jest równoważny rozwiązaniu zagadnienia kolejnościowego postaci $n|m|P, r_i \geq 0, R_\pi | f_{\max}$ gdzie R_π jest relacją częściowego porządku ustaloną w każdym następniku β względem π .

Dla rozpatrywanego zagadnienia dolne ograniczenia zostanie wyznaczone poprzez relaksację możliwości wykonawczych maszyn oraz relaksację funkcji kosztów. Wyznaczanie dolnego ograniczenia poprzez relaksację możliwości wykonawczych maszyn w odniesieniu do zagadnień z funkcją celu $C_{\max}$ i $L_{\max}$ prezentowano w [63], [71], [43]. Istota relaksacji polega na przyjęciu, że dla pewnych wybranych maszyn liczba operacji realizowanych jednocześnie może być dowolnie duża /zadania nie muszą być szeregowane na tych maszynach/. Zadania dotyczące pozostałych maszyn nie zmieniają się, tzn. każda z tych maszyn może realizować nie więcej niż jedno zadanie w dowolnej chwili czasowej. Optymalna wartość funkcji celu /lub dolne ograniczenie/ dla problemu zrelaksowanego jest dolnym ograniczeniem wartości funkcji celu problemu wyjściowego.

W dalszym ciągu została przeprowadzona analiza możliwości wyznaczania dolnego ograniczenia poprzez różne relaksacje problemu wyjściowego, które sprowadzają rozpatrywany problem do problemu kolejnościowego z algorytmem o złożoności wielomianowej. Takie ograniczenie klasy zagadnień wydaje się być rozsądne, bowiem problem wyznaczania dolnego ograniczenia jest rozwiązywany wielokrotnie w czasie działania algorytmu, w każdym węźle drzewa rozwiązań, co wpływa w istotny sposób na czas działania algorytmu.

Z analizy złożoności obliczeniowej algorytmów kolejnościowych wynika, że algorytmy efektywne istnieją tylko dla zagadnień z co najwyżej dwiema maszynami. Niech $M_u, M_v, 1 \leq u \leq v \leq m$ będą maszynami o ograniczonej przepustowości, zaś $M_i, i = 1, 2, \dots, m, u \neq i \neq v$, będą maszynami o nieograniczonej przepustowości. Podzbiór kolejnych maszyn o nieograniczonej przepustowości możemy zastąpić jedną maszyną o takiej samej własności. Z powyższego wynika, że wystarczy rozważać zagadnienie z co najwyżej pięcioma maszynami $M_u, M_v, M_{uv}, M_u, M_v$, z których co najwyżej trzy M_u, M_{uv}, M_v mają nieograniczoną przepustowość. Czasy wykonywania zadań na tych maszynach mają

postać

$$p_{i \cdot u} = r_i + \sum_{k=1}^{u-1} p_{ik} \quad , \quad J_i \in J \quad , \quad (5.17)$$

$$p_{iuv} = \sum_{k=u+1}^{v-1} p_{ik} \quad , \quad J_i \in J \quad , \quad (5.18)$$

$$p_{iv} = \sum_{k=v+1}^m p_{ik} \quad , \quad J_i \in J \quad , \quad (5.19)$$

oraz czasy p_{iu} , p_{iv} nie ulegają zmianie. Oznaczmy odpowiednio $S_{i \cdot u}$, S_{iu} , S_{iuv} , S_{iv} , S_{iv} oraz $C_{i \cdot u}$, C_{iu} , C_{iuv} , C_{iv} , C_{iv} terminy rozpoczęcia i zakończenia realizacji zadań na tych maszynach. W wyniku przyjętej relaksacji otrzymaliśmy zagadnienie kolejnościowe postaci $n|5|P, M_1, M_2, M_5$ - bez ograniczeń, $\mathcal{R}_n|f_{\max}$ gdzie $f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(C_{iv})$. Każde dolne ograniczenie funkcji celu tego zagadnienia jest dolnym ograniczeniem wszystkich następników względem Π .

Wartość $p_{i \cdot u}$ może być interpretowane jako najwcześniejszy termin rozpoczęcia wykonywania zadania $J_i \in J$ na maszynie M_u tzn. $r_{iu} \triangleq p_{i \cdot u}$ oraz $r_{iu} \leq S_{iu}$; wartość p_{iuv} - jako zwłoka czasowa pomiędzy wykonywaniem operacji O_{iu} na maszynie M_u oraz O_{iv} na maszynie M_v , tzn. $C_{iu} + p_{iuv} \leq S_{iv}$; wartość p_{iv} - jako końcówka zadania $J_i \in J$ na maszynie M_v związana z kosztem realizacji zadania tzn. $q_{iv} = p_{iv}$ oraz $f_i(C_{iv}) = f_i(C_{iv} + q_{iv}) \triangleq f_{iv}(C_{iv})$. Tak więc problem $n|5|P, M_1, M_3, M_5$ - bez ograniczeń, $\mathcal{R}_n|f_{\max}$ możemy również oznaczyć $n|2|P, r_i \geq 0, c_i \geq 0, \prec |f_{\max}, f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(C_{i4} + q_i)$.

Zagadnienie to jest NP-zupełne /bowiem zagadnienie $n|2|P, r_i \geq 0, c_i \geq 0, \prec |C_{\max}$ jest NP-zupełne. Dla opisanego problemu możemy przeprowadzić eliminację maszyny o nieograniczonej przepustowości. Dla powstałego w ten sposób zagadnienia możemy wyznaczyć nowe funkcje kosztu $f_{iu}(C_{iv})$, $f_{iuv}(C_{iv})$, $f_{iv}(C_{iv})$,

$$f_{iu}(C_{iv}) \triangleq f_i(C_{iv} + \min_{J_i \in J} r_{iu}) \quad , \quad (5.20)$$

$$f_{iuv}(C_{iv}) \triangleq f_i(C_{iv} + \min_{J_i \in J} p_{iuv}) \quad , \quad (5.21)$$

$$f_{iv}(C_{iv}) \triangleq f_i(C_{iv} + \min_{J_i \in J} q_{iv}), \quad (5.22)$$

oraz rozwiązać zagadnienie. Możemy również przeprowadzić eliminację kosztów z problemu przez przyjęcie

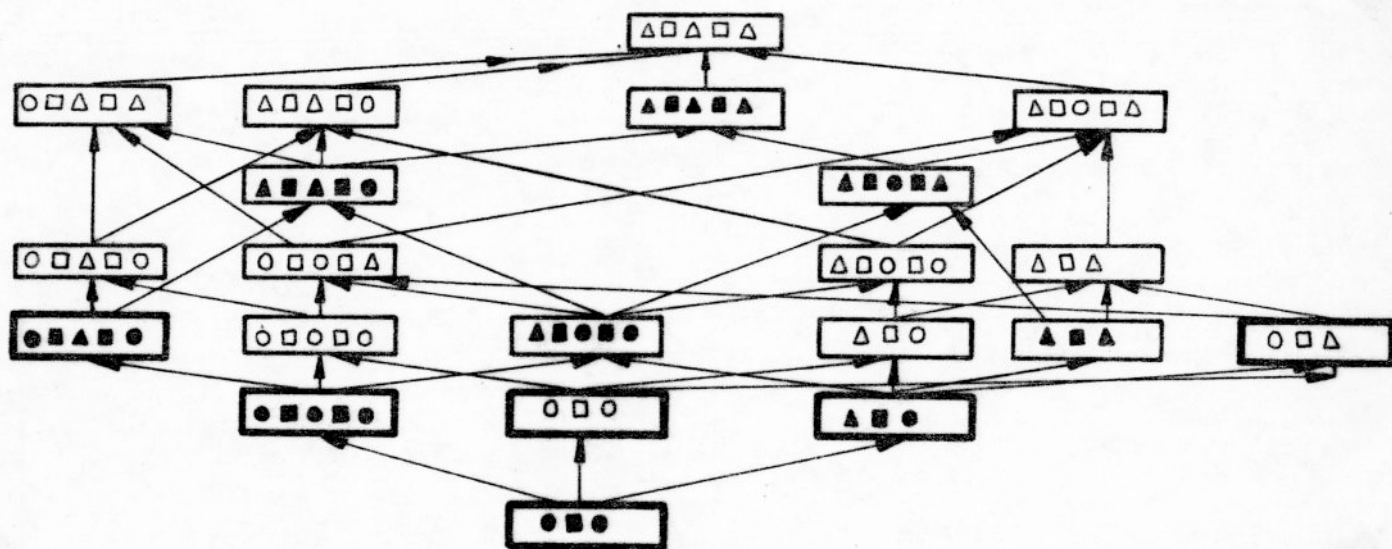
$$f_*(t) \triangleq \min_{J_i \in J} f_i(t), \quad 0 \leq t < \infty. \quad (5.23)$$

Dla powstałego w ten sposób zagadnienia możemy wyznaczyć rozwiązanie z kryterium $C_{\max}$ a następnie przyjąć koszt równy $f_*(C_{\max})$.

Wszystkie metody wyznaczania dolnych ograniczeń przedstawiono za pomocą ciągu Ω zawierającego co najwyżej pięć symboli ze zbioru $\{\square, \Delta, \circ\}$. Poszczególne symbole oznaczają

- $\square$ - maszynę o nieograniczonej przepustowości,
- Δ - maszynę o ograniczonej przepustowości,
- $\circ$ - maszynę o nieograniczonej przepustowości, którą wyeliminowa-
no w wyżej podany sposób.

Niech $\blacksquare, \blacktriangle, \bullet$ oznaczają odpowiednie maszyny w problemie, w którym eliminowano koszty.



Rys. 5.3. Graf relacji dominowania ciągów Ω związanych z dolnymi ograniczeniami

Dolne ograniczenie związane z ciągiem Ω i maszynami M_u, M_v oznaczmy symbolem $LB(u, v, \Omega, \mathcal{R}_\pi)$. Dalej niech $LB^*(u, v, \Omega, \mathcal{R}_\pi)$ oznacza wartości funkcji celu rozwiązanie optymalnego dla zagadnienia kolejnościowego z maszynami M_u i M_v postaci $\square, \Delta, \blacksquare, \blacktriangle$. Ponieważ $LB(u, v, \Omega, \mathcal{R}_\pi)$ jest dolnym ograniczeniem dla dowolnych par $u, v, 1 \leq u \leq v \leq m$ możemy więc przyjąć

$$LB(\Omega, \mathcal{R}_\pi) = \max_{1 \leq u \leq v \leq m} LB(u, v, \Omega, \mathcal{R}_\pi) \quad (5.24)$$

jako dolne ograniczenie związane z ciągiem Ω .

Pomiędzy pary ciągów Ω i Ω' wprowadza się relację dominowania. Ciąg Ω' dominuje nad ciągiem Ω jeśli dla każdej pary $\langle u, v \rangle$ możemy znaleźć taką parę $\langle u', v' \rangle$ że spełniona jest nierówność

$$LB(u', v', \Omega', \mathcal{R}_\pi) \geq LB(u, v, \Omega, \mathcal{R}_\pi)$$

Relacja dominowania jest relacją przechodnią. Na rys. 5.3 przedstawiono graf dominowania wszystkich ciągów Ω . Każdy wierzchołek grafu reprezentuje pewien ciąg Ω zaś każdy łuk grafu skierowanego $\langle \Omega', \Omega \rangle$ oznacza, że ciąg Ω' dominuje ciąg Ω . Z własności problemów kolejnościowych wynika, że problemy odpowiadające ciągom symetrycznym z eliminowanymi kosztami /np. ciągi $\bullet \blacksquare \blacktriangle$ i $\blacktriangle \blacksquare \bullet$ są symetryczne/ są równoważne.

Wobec powyższego wyróżniono 21 różnych ciągów, w tym 9 odpowiadających zagadnieniom z wyeliminowanymi kosztami. Z analizy złożoności obliczeniowej algorytmów podanych 21 zagadnień wynika, że tylko 6 zagadnień posiada algorytmy efektywne. Problemy te zostały krótko opisane.

- (i) $\bullet \blacksquare \bullet$. Pomijając pierwszą i ostatnią maszynę o nieograniczonej przepustowości oraz eliminując koszty z problemu otrzymujemy zagadnienie minimalizacji $C_{\max} \triangleq \max_{1 \leq i \leq n} C_{iv}$ na maszynie $M_u = M_v$, tzn. zagadnienie postaci $n | 1 | \prec | C_{\max}$. Ponieważ $LB^*(u, u, (\bullet \blacksquare \bullet), \mathcal{R}_\pi) = \sum_{i=1}^n P_{iu}$, więc

$$LB(u, u, (\bullet \blacksquare \bullet), \mathcal{R}_\pi) = f_* \left(\sum_{i=1}^n P_{iu} + \min_{\substack{i,j \\ i \neq j}} [r_{iu} + q_{jv}] \right). \quad (5.25)$$

Złożoność obliczeniowa algorytmu jest $O(n)$.

Jest to najskłabsze dolne ograniczenie.

- (ii) (○□○). Pomijając pierwszą i ostatnią maszynę o nieograniczonej przepustowości otrzymujemy zagadnienie minimalizacji kosztu maksymalnego

$$f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(C_{iu} + \min_{1 \leq j \leq n} r_{ju} + \min_{1 \leq j \leq n} q_{jv}) \triangleq \max_{1 \leq i \leq n} f_i'(C_{iu}), \quad (5.26)$$

na jednej maszynie $M_u = M_v$, tzn. zagadnienie $n|1|<|f_{\max}$. Algorytm rozwiązania tego zagadnienia przedstawiono w pracy

[70] i wymaga $O(n^2)$ iteracji. Optymalna wartość funkcji celu dla tego zagadnienia jest wartością $LB(u, u, (○□○), \mathcal{R}_n)$

- (iii) (▲■●). Pomijając ostatnią maszynę o nieograniczonej przepustowości oraz eliminując koszty z problemu otrzymujemy zagadnienie minimalizacji $C_{\max}$ na jednej maszynie $M_u = M_v$ z niezerowymi terminami rozpoczęcia wykonywania zadań, tzn. zagadnienie $n|1|r_i \geq 0, <|C_{\max}$. Optymalna wartość funkcji celu $C_{\max}$ dla tego zagadnienia jest wartością $LB^*(u, u, (▲■●), \mathcal{R}_n)$. Algorytm rozwiązania tego zagadnienia [69] wymaga $O(n^2)$ iteracji. W rezultacie końcowym otrzymujemy

$$LB(u, u, (▲■●), \mathcal{R}_n) = f_*(LB^*(u, u, (▲■●), \mathcal{R}_n) + \min_{1 \leq i \leq n} q_{iv}) \quad (5.27)$$

(●■▲). Pomijając pierwszą maszynę o nieograniczonej przepustowości oraz eliminując koszty z problemu otrzymujemy zagadnienie minimalizacji maksymalnej nieterminowości z końcówką q_{iv}

$$L'_{\max} = \max_{1 \leq i \leq n} (C_{iu} + q_{iv}), \quad (5.28)$$

na jednej maszynie $M_u = M_v$ /przyjmując $K = \max_{1 \leq i \leq n} q_{iv}$ oraz $d_i = K - q_{iv}$. Optymalna wartość funkcji $L'_{\max}$ celu dla tego zagadnienia jest wartością $LB^*(u, u, (●■▲), \mathcal{R}_n)$. Algorytm rozwiązań tego zagadnienia [69] wymaga $O(n^2)$ iteracji. W końcowym rezultacie otrzymujemy

$$LB(u, u, (●■▲), \mathcal{R}_n) = f_*(LB^*(u, u, (●■▲), \mathcal{R}_n) + \min_{1 \leq j \leq n} r_{ju}). \quad (5.29)$$

- (iv) (○□△). Pomijając pierwszą maszynę o nieograniczonej przepustowości otrzymujemy zagadnienie minimalizacji

$$f_{\max} \triangleq \max_{1 \leq i \leq n} (f_i(C_{iu} + q_{iv} + \min_{1 \leq j \leq n} r_{ju})) \triangleq \max_{1 \leq i \leq n} f_i''(C_{iu}) \quad (5.30)$$

na jednej maszynie $M_u = M_v$, tzn. zagadnienie postaci $n|1||f_{\max}$. Rezultat jest analogiczny jak w (ii).

Pozostałe zagadnienia w ogólnym przypadku są NP-zupełne.

Przy założeniu $R_n = \emptyset$ istnieją algorytmy efektywne dla niektórych z pozostałych problemów. Są to:

(v) $(\bullet \blacksquare \bullet \blacksquare \bullet), R_n = \emptyset$. Pomijając wszystkie maszyny w nieograniczonej przepustowości oraz eliminując koszty z problemu otrzymujemy zagadnienie minimalizacji

$$C_{\max} \triangleq \max_{1 \leq i \leq n} C_{iv} \quad (5.31)$$

maksymalnego czasu zakończenia na dwóch maszynach M_u i M_v , tzn. zagadnienie $n|2|P|C_{\max}$. Optymalna wartość $C_{\max}$ jest optymalną wartością $LB(u, v, (\bullet \blacksquare \bullet \blacksquare \bullet), \emptyset)$. Zagadnienie to można rozwiązać algorytmem Johnsona [56] w $O(n \log n)$ iteracjach. W rezultacie otrzymujemy

$$LB(u, v, (\bullet \blacksquare \bullet \blacksquare \bullet), \emptyset) = f_*(LB^*(u, v, (\bullet \blacksquare \bullet \blacksquare \bullet), \emptyset) + \min_{\substack{i, j \\ i \neq j}} [r_{iu} + q_{jv}] + \min_{1 \leq i \leq n} p_{iuv}) \quad (5.32)$$

Złożoność obliczeniowa tego algorytmu jest $O(n \log n)$.

(vi) $(\bullet \blacksquare \blacktriangle \bullet \blacksquare \bullet), R_n = \emptyset$. Pomijając pierwszą i ostatnią maszynę o nieograniczonej przepustowości otrzymujemy zagadnienie $n|3|P, M_2 - \text{bez ograniczeń}|C_{\max}$. Rozwiązanie tego zagadnienia możemy otrzymać poprzez rozwiązanie algorytmem Johnsona zagadnienia na dwóch maszynach M_u i M_v z czasami wykonywania zadań $p_{iu} + p_{iuv}$ oraz $p_{iv} + p_{iuv}$, $J_i \in J$ [82]. Otrzymana w ten sposób optymalna kolejność jest optymalną kolejnością dla $n|3|P, M_2 - \text{bez ograniczeń}|C_{\max}$. Optymalna wartość $C_{\max}$ jest wartością $LB^*(u, v, (\bullet \blacksquare \blacktriangle \bullet \blacksquare \bullet), \emptyset)$. Mamy zatem

$$LB(u, v, (\bullet \blacksquare \blacktriangle \bullet \blacksquare \bullet), \emptyset) = f_*(LB^*(u, v, (\bullet \blacksquare \blacktriangle \bullet \blacksquare \bullet), \emptyset) + \min_{\substack{i, j \\ i \neq j}} [r_{iu} + q_{jv}]) \quad (5.33)$$

Złożoność obliczeniowa tego algorytmu jest $O(n \log n)$.

5.4. Zagadnienie wzmocnienia dolnych ograniczeń

We wszystkich metodach wyznaczania dolnych ograniczeń dla ciągów Ω zawierających maszyny postaci $o, \bullet$ możemy zwiększyć wartości dolnego ograniczenia $LB(\Omega, \mathcal{R}_n)$. Ogólna idea polega na wyznaczeniu dolnego ograniczenia dla pewnego podzbioru zadań $J' \subset J$. Oczywiście podzbiór J' należy tak wybrać, aby nastąpiło zwiększenie wartości $LB(\Omega, \mathcal{R}_n)$. Zasady wyznaczania podzbioru J' oparto na niżej podanych własnościach.

Rozpatrzmy dolne ograniczenia związane z ciągiem $\Omega = (\blacktriangle \blacksquare \blacktriangle)$. Ponieważ optymalna wartość funkcji celu problemu $n/1/r_i \geq 0/C'_{\max}$, $C'_{\max} \triangleq \max_{1 \leq i \leq n} (C_i + q_i)$ jest także optymalną wartością funkcji celu problemu "symetrycznego" tzn. $n/1/q_i \geq 0/C''_{\max}$, $C''_{\max} \triangleq \max_{1 \leq i \leq n} C_i + r_i$ a więc rozważanie dotyczące ciągów $(\blacktriangle \blacksquare \bullet)$ oraz $(\bullet \blacksquare \blacktriangle)$ wystarczy ograniczyć do jednego z nich. Oczywiście

$$LB(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_n) \geq \max [LB(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_n), LB(u, u, (\bullet \blacksquare \blacktriangle), \mathcal{R}_n)] \quad (5.34)$$

oraz rozwiązanie obu zagadnień można otrzymać stosując algorytm opisany w pracy [69].

Twierdzenie 5.3.

Niech $C_{\max}$ będzie optymalną wartością funkcji celu, $\mathcal{G}$ - permutacją optymalną, $\langle u^{\mathcal{G}}, n \rangle$ - ciągiem krytycznym w $\mathcal{G}$ dla problemu $n/1/r_i \geq 0, \prec / C_{\max}$. Wówczas zachodzi

$$LB(u, u, (\blacktriangle \blacksquare \blacktriangle), \mathcal{R}_n) \geq f_*(LB(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_n) + \min_{u^{\mathcal{G}} \leq i \leq n} q_{\mathcal{G}(i)}) \geq LB(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_n). \quad (5.35)$$

Dowód

Zauważmy, że funkcja $f_*(t)$, $0 \leq t < \infty$ jest niemalejąca bowiem funkcje $f_i(t)$, $J_i \in J$ są niemalejące. Tak więc wystarczy pokazać, że

$$L_{\max}^* \geq C_{\max}^* + \max_{u^{\mathcal{G}} \leq i \leq n} q_{\mathcal{G}(i)} \geq C_{\max}^* + \max_{1 \leq i \leq n} q_i \quad (5.36)$$

gdzie $L_{\max}^*$, $C_{\max}^*$ są odpowiednio optymalnymi wartościami funkcji celu problemów $n/1/r_i \geq 0, \prec / L_{\max}$, $L_{\max} \triangleq \max_{1 \leq i \leq n} (C_i + q_i)$ oraz $n/1/r_i \geq 0, \prec / C_{\max}$, $C_{\max} \triangleq \max_{1 \leq i \leq n} C_i$.

Rozważmy zbiór zadań $J_G \subset J$, $J_G = \{J_{\sigma(i)} \in J \mid u^\sigma \leq i \leq n\}$.
 Optymalna wartość $L_{\max}$ problemu $n \mid 1 \mid r_i \geq 0, \prec \mid L_{\max}$ dla zadań ze zbioru J jest nie mniejsza niż optymalna wartość $L_{\max, G}$ problemu $n_G \mid 1 \mid r_i \geq 0, \prec \mid L_{\max, G}$ dla zadań ze zbioru $J_G \subset J$, tzn.

$$L_{\max}^* \geq L_{\max, G} \quad (5.37)$$

Z prezentacji dolnych ograniczeń wynika, że w odniesieniu do zadań zbioru J_G zachodzi

$$L_{\max, G}^* \geq C_{\max, G}^* + \min_{J_i \in J_G} q_i \quad (5.38)$$

gdzie $C_{\max, G}^*$ jest optymalną wartością funkcji celu problemu $n_G \mid 1 \mid r_i \geq 0, \prec \mid C_{\max, G}$. Ale $C_{\max, G}^* = C_{\max}^*$, $C_{\max}^* \triangleq \max_{J_i \in J} C_i^*$ w problemie $n \mid 1 \mid r_i \geq 0, \prec \mid C_{\max}$, więc korzystając z wzorów (5.37) i (5.38) możemy zapisać

$$L_{\max}^* \geq C_{\max, G}^* + \min_{J_i \in J_G} q_i = C_{\max}^* + \min_{u^\sigma \leq i \leq n} q_{\sigma(i)} \quad (5.39)$$

co daje spełnienie pierwszej nierówności w (5.35).

Nierówność druga wynika z faktu, że

$$\min_{u^\sigma \leq i \leq n} q_{\sigma(i)} \geq \min \left[\min_{1 \leq i \leq u^\sigma} q_{\sigma(i)}, \min_{u^\sigma \leq i \leq n} q_{\sigma(i)} \right] = \min_{1 \leq i \leq n} q_i. \quad (5.40)$$

Zauważmy, że z podanego twierdzenia i wzoru (5.34) wynika, że

$$LB(u, u, (\blacktriangle \blacksquare \blacktriangle), \mathcal{R}_\pi) \geq \max \left[f_*(LB^*(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_\pi) + \min_{u^\sigma \leq i \leq n} q_{\sigma(i)}), f_*(LB^*(u, u, (\bullet \blacksquare \blacktriangle), \mathcal{R}_\pi) + \min_{u^\beta \leq i \leq n} r_{\beta(i)}) \right] \quad (5.41)$$

gdzie $\langle u^\sigma, n \rangle$ oraz $\langle u^\beta, n \rangle$ są odpowiednimi ciągami krytycznymi. Rozpatrzmy dalej dolne ograniczenie postaci $f_*(LB^*(u, u, (\blacktriangle \blacksquare \bullet), \mathcal{R}_\pi) + \min_{u^\sigma \leq i \leq n} q_{\sigma(i)})$. Niech $C_{\max, G}^*$, $\langle u^\sigma, n \rangle$ będą odpowiednio optymalną wartością funkcji celu, permutacją optymalną, ciągiem krytycznym w G dla problemu $n \mid 1 \mid r_i \geq 0, \prec \mid C_{\max}$. Niech $J_k \in J$ będzie zadaniem takim, że $q_k = q_* = \min_{u^\sigma \leq i \leq n} q_{\sigma(i)}$, zaś J_1 zadaniem takim, że $f_1(C_{\max}^* + q_k) = \min_{J_i \in J} f_i(C_{\max}^* + q_*)$. Oznaczmy $J' = J - \{J_k\}$ ($J' = J - \{J_1\}$)

oraz

$$f'_*(t) \triangleq \min_{J_i \in J'} f_i(t), \quad 0 \leq t < \infty \quad (5.42)$$

Niech $C_{\max}^*$, σ , $\langle u^\sigma, n \rangle$ będą odpowiednio optymalną wartością funkcji celu, permutacją optymalną, ciągiem krytycznym w σ dla problemu $n-1|1|r_i \geq 0, \prec | C_{\max}$ zawierającego zadania ze zbioru J . Oznaczmy dodatkowo

$$q'_* \triangleq \min_{u^\sigma \prec i \leq n} q_{\sigma(i)}. \quad 5.43$$

Twierdzenie 5.4.

Niech $C_{\max}^*$, σ , $\langle u^\sigma, n \rangle$, $C_{\max}^{*'}$, σ' , $\langle u^{\sigma'}, n \rangle$, q_* , q'_* będą wielkościami zdefiniowanymi jak wyżej zaś $f_*(t)$, $f'_*(t)$ funkcjami określonymi wzorami (5.23) i (5.42). Niech $f_i(t)$, $J_i \in J$ będą funkcjami niemalejącymi.

Jeżeli istnieje zadanie $J_s \in J'$ takie, że $J_{\sigma(u^\sigma)} \prec J_s$ oraz

$$(i) \quad q_* + p_k < q_s = \min_{u^\sigma \prec i \leq n} q_{\sigma(i)}, \quad J' = J - \{J_k\} \quad (5.44)$$

lub

$$(ii) \quad f_1(C_{\max}^* + q_*) < f_s(C_{\max}^* - p_1 + q) = \\ = \min_{J_i \in J} f_i(C_{\max}^* - p_1 + q_*) \quad J' = J - \{J_1\}, \quad (5.45)$$

to zachodzi

$$f_*(C_{\max}^* + q_*) < f'_*(C_{\max}^{*'} + q'_*) \quad \blacksquare \quad (5.46)$$

Dowód

(i) Z własności problemu $n-1|1|r_i \geq 0, \prec | C_{\max}$ wynika, że optymalna kolejność dla problemu $n-1|1|r_i \geq 0, \prec | C_{\max}$ ze zbiorem zadań J jest także optymalną kolejnością dla problemu $n-1|1|r_i \geq 0, \prec | C_{\max}$ ze zbiorem zadań $J' = J - \{J_k\}$. Na mocy wzoru (5.3) i dla $m = 1$ oraz przy założeniu $J_{\sigma(u^\sigma)} \prec J_s$ zachodzi

$$C_{\max}^* - p_k \leq C_{\max}^{*'} \quad (5.47)$$

Korzystając z założenia (i) twierdzenia oraz z wzoru (5.47) możemy zapisać

$$C_{\max}^* - p_k + p_k + q_* = C_{\max}^* + q_* < C_{\max}^{*'} + q_s \quad (5.48)$$

Z konstrukcji dolnego ograniczenia wynika, że

$$\begin{aligned} f_*(C_{\max}^* + q_*) &= \min_{J_i \in J} f_i(C_{\max}^* + q_*) \leq \min_{J_i \in J'} f_i(C_{\max}^* + q_*) = \\ &= f'_*(C_{\max}^* + q_*). \end{aligned} \quad (5.49)$$

Ponieważ $f'_*(t)$ jest funkcją niemalejącą /bowiem wszystkie $f_i(t)$, $J_i \in J$ są funkcjami niemalejącymi/ oraz $u^G \leq u^G$ z własności problemu n/1/ $r_i \geq 0$, $\prec | C_{\max}$, więc

$$f_*(C_{\max}^* + q_*) \leq f'_*(C_{\max}^* + q_*) < f'_*(C_{\max}^{*'} + q_*) = f'_*(C_{\max}^{*'} + q'_*) \quad (5.50)$$

(ii) Z założenia (ii) otrzymujemy

$$\begin{aligned} f_*(C_{\max}^* + q_*) &= f_l(C_{\max}^* + q_*) < \min_{J_i \in J} f_i(C_{\max}^* - p_l + q_*) = \\ &= f'_*(C_{\max}^* - p_l + q_*). \end{aligned} \quad (5.51)$$

Uwzględniając zależność (5.47), (5.51) oraz fakt, że $f_i(t)$, $J_i \in J$ są funkcjami niemalejącymi otrzymujemy rezultat

$$\begin{aligned} f_*(C_{\max}^* + q_*) &< f'_*(C_{\max}^* - p_l + q_*) \leq f'_*(C_{\max}^{*'} + q_*) \leq \\ &\leq f'_*(C_{\max}^{*'} + q'_*), \end{aligned} \quad (5.52)$$

bowiem $q'_* \geq q_*$. ■

Zauważmy dodatkowo, że warunek (i) twierdzenia 5.4 można zastąpić warunkiem słabszym

$$q_* + p_k < q_s = \min_{\substack{u^G \leq i \leq n \\ i \neq k}} q_{G(i)}, \quad (5.53)$$

natomiast warunek (ii) w przypadku, gdy np. $f_i(t) = w_i t$, $i = 1, 2, \dots, n$, można zapisać w postaci

$$w_s > \frac{C_{\max}^* + q_*}{C_{\max}^* - p_l + q_*} w_l, \quad w_l = \min_{J_i \in J} w_i, \quad (5.54)$$

W wyniku zastosowania twierdzenia 5.4 otrzymujemy podzbiór $J^1 \subset J$ taki, że

$$LB^1(u, u, (O \square \Delta), \mathcal{R}_\pi) > LB(u, u, (O \square \Delta), \mathcal{R}_\pi), \quad (5.55)$$

gdzie

$$LB^1(u, u, (O \square \Delta), \mathcal{R}_\pi) = f'_*(C_{\max}^{*'} + q'_*) \quad (5.56)$$

Przez wielokrotne stosowanie twierdzenia możemy otrzymać ciąg podzbiorów $J \supset J^1 \supset J^2 \supset \dots$ oraz ciąg rosnących wartości dolnego ograniczenia $LB(\cdot) < LB^1(\cdot) < \dots$. Celem naszym będzie wyznaczenie takiego podzbioru J^S , że

$$LB(\cdot) - LB^S(\cdot) = \max_i [LB(\cdot) - LB^i(\cdot)] > 0. \quad (5.57)$$

Powyższe postępowanie może zostać rozszerzone na inne przypadki dolnych ograniczeń.

Rozpatrzmy dalej dolne ograniczenie postaci $LB(u, v, (\bullet \blacksquare \bullet), \emptyset)$. Z prezentacji dolnych ograniczeń /rozdz. 5.3/ wynika, że

$$LB(u, v, (\circ \square \square \circ), \emptyset) = f_*(LB^*(u, v, (\circ \square \square \circ), \emptyset) + r_* + q_*) \quad (5.58)$$

gdzie

$$f_*(t) \triangleq \min_{J_i \in J} f_i(t), \quad (5.59)$$

oraz

$$r_* + q_* = \min_{\substack{i, j \\ i \neq j}} [r_{iu} + q_{jv}]. \quad (5.60)$$

Wartość $LB^*(u, v, (\circ \square \square \circ), \emptyset)$ jest równa optymalnej wartości $C_{\max}$ dla problemu $n|2|P|C_{\max}$ rozwiązywanego algorytmem Johnsona [56].

Niech dalej $C_{\max}^*$, σ , < 1 , u^σ , $n >$ będą odpowiednio optymalną wartością funkcji celu, permutacją optymalną, ciągiem krytycznym w σ dla problemu $n|2|P|C_{\max}$. Niech $J_k \in J$ będzie zadaniem takim, że $r_k = r_*$ /lub $q_k = q_*$ /, zaś $J_1 \in J$ zadaniem takim, że $f_1(C_{\max} + r_* + q_*) = \min_{J_i \in J} f_i(C_{\max} + r_* + q_*)$. Oznaczmy $J' = J - \{J_k\}$ ($J' = J - \{J_1\}$) oraz

$$f'_*(t) \triangleq \min_{J_i \in J'} f_i(t), \quad 0 \leq t < \infty. \quad (5.61)$$

Niech $C_{\max}^{*'} bę d z e$ optymalną wartością funkcji celu problemu $n-1|2|P|C_{\max}$ zawierającego zadania ze zbioru J' . Oznaczmy dodatkowo

$$r'_* + q'_* = \min_{\substack{J_i, J_j \in J' \\ i \neq j}} [r_i + q_j]. \quad (5.62)$$

Twierdzenie 5.5.

Niech $C_{\max}^*$, σ , < 1 , u^σ , $n >$, $C_{\max}^{*}$, r_* , q_* , r'_* , q'_* będą wielkościami zdefiniowanymi jak wyżej zaś $f_*(t)$, $f'_*(t)$ funkcjami określonymi wzorami (5.59) i (5.61), gdzie $f_i(t)$, $i = 1, 2, \dots, n$ są funkcjami niemalejącymi.

Jeżeli istnieje zadanie $J_s \in J'$ /lub $J_p \in J'$ / takie, że $J_s \prec J_{G(u)}$ /lub $J_{G(u)} \prec J_p$ / oraz

$$(i) \quad p_{k1} + r_* < r_s = \min_{J_i \in J} r_i, \quad J' = J - \{J_k\}, \quad (5.63)$$

$$/lub \quad p_{k2} + q_* < q_p = \min_{J_i \in J} q_i, \quad J' = J - \{J_k\} / \quad (5.63)$$

lub

$$(ii) \quad f_1(r_* + C_{\max}^* + q_*) < f_s(r_* - p_{11} + C_{\max}^* + q_*) = \\ = \min_{J_i \in J'} f_i(r_* - p_{11} + C_{\max}^* + q_*), \quad J' = J - \{J_1\} \quad (5.64)$$

$$/lub \quad f_1(r_* + C_{\max}^* + q_*) < f_s(r_* - p_{12} + C_{\max}^* + q_*) = \\ = \min_{J_i \in J'} f_i(r_* - p_{12} + C_{\max}^* + q_*), \quad J' = J - \{J_1\} \quad (5.64)$$

to zachodzi

$$f_*(r_* + C_{\max}^* + q_*) < f'_*(r'_* + C_{\max}^{*'} + q'_*). \quad (5.65)$$

Dowód

(i) Z własności problemu $n|2|P|C_{\max}$ wynika, że optymalna kolejność dla problemu $n|2|P|C_{\max}$ ze zbiorem zadań J jest także optymalną kolejnością dla problemu $n-1|2|P|C_{\max}$ ze zbiorem zadań $J' = J - \{J_k\}$. Na mocy wzoru (5.3) dla $m = 2$ oraz przy założeniu $J_k \prec J_{G(u)}$ zachodzi

$$C_{\max}^* - p_{k1} \leq C_{\max}^{*'} \quad (5.66)$$

Korzystając z założenia (i) twierdzenia oraz z wzoru (5.66) możemy zapisać

$$C_{\max}^* - p_{k1} + p_{k1} + r_* + q_* = C_{\max}^* + r_* + q_* < C_{\max}^{*'} + r_s + q_* \quad (5.67)$$

Z konstrukcji dolnego ograniczenia wynika, że

$$f_*(r_* + C_{\max}^* + q_*) = \min_{J_i \in J} f_i(r_* + C_{\max}^* + q_*) \\ \leq \min_{J_i \in J'} f_i(r_* + C_{\max}^* + q_*). \quad (5.68)$$

Zauważmy, że $r_s = r'_*$, $r'_* + q_* \leq r'_* + q'_* = \min_{\substack{J_i, J_j \in J \\ i \neq j}} [r_i + q_j]$ oraz że funkcje $f_i(t)$, $J_i \in J$ są niemalejące więc funkcja $f'_*(t) = \min_{J_i \in J} f_i(t)$

jest również niemalejąca. W konsekwencji z wzoru (5.68) otrzymujemy

$$\begin{aligned} f_*(r_* + C_{\max}^* + q_*) &= \min_{J_i \in J} f_i(r_* + C_{\max}^* + q_*) \leq \\ &\leq \min_{J_i \in J} f_i(r_* + C_{\max}^* + q_*) < \min_{J_i \in J} f_i(r_*' + C_{\max}^{*'} + q_*) \leq \\ &\leq f_*'(r_*' + C_{\max}^{*'} + q_*'). \end{aligned} \quad (5.69)$$

(ii) Z założenia (ii) otrzymujemy

$$\begin{aligned} f_*(r_* + C_{\max}^* + q_*) &= f_l(r_* + C_{\max}^* + q_*) < f_s(r_* + C_{\max}^* - p_{l1} + q_*) \\ &= \min_{J_i \in J} f_i(r_* + C_{\max}^* - p_{l1} + q_*) = f_*'(r_* + C_{\max}^* - p_{l1} + q_*) \end{aligned} \quad (5.70)$$

Korzystając z faktu, że funkcja $f_*(t)$ jest niemalejąca oraz ze wzoru (5.66) możemy zapisać

$$\begin{aligned} f_*(r_* + C_{\max}^* + q_*) &< f_*'(r_* + C_{\max}^* - p_{l1} + q_*) \leq f_*'(r_* + C_{\max}^{*'} + q_*) \\ &\leq f_*'(r_*' + C_{\max}^{*'} + q_*'), \end{aligned} \quad (5.71)$$

gdzie $r_*' + q_*' = \min_{\substack{J_i, J_j \in J \\ i \neq j}} [r_i + q_j] > r_* + q_*$.

Dowód dla alternatywnego przypadku możemy otrzymać stosując analogiczny schemat postępowania. ■

Zauważmy dodatkowo, że w przypadku gdy np. $f_i(t) = w_i t$, $i = 1, 2, \dots, n$ to warunek (ii) twierdzenia 5.5 można zapisać w postaci

$$w_s > \frac{r_* + C_{\max}^* + q_*}{r_* + C_{\max}^* - p_{l1} + q_*} w_l, \quad w_l = \min_{J_i \in J} w_i \quad (5.72)$$

$$\text{/lub } w_s > \frac{r_* + C_{\max}^* + q_*}{r_* + C_{\max}^* - p_{l2} + q_*} w_l, \quad w_l = \min_{J_i \in J} w_i \quad (5.72)$$

Analogicznie jak poprzednio /tw. 5.4/, również i w tym przypadku możemy wyznaczyć /poprzez zastosowanie tw. 5.5/ ciąg podzbiorów $J \supset J^1 \supset J^2 \supset \dots$ oraz ciąg niemalejących wartości dolnych ograniczeń $LB(\cdot) < LB^1(\cdot) < LB^2(\cdot) \dots$, a w szczególności wyznaczyć podzbiór $J^S \subset J$ taki, że

$$LB^S(\cdot) - LB(\cdot) = \max_{J^1 \subset J} [LB^1(\cdot) - LB(\cdot)]. \quad (5.73)$$

Powyższe postępowanie może zostać rozszerzone na przypadek $LB(u, v, (\bullet \blacksquare \bullet \blacksquare \bullet), \emptyset)$ oraz inne.

5.5: Zasada podziału

Dana jest permutacja $\pi \in \Pi(\mathcal{R})$. Należy wskazać zadanie w π którego przesunięcie pozwoli na wygenerowanie następnika $\beta \in \Pi(\mathcal{R})$ w drzewie H. Będziemy przesuwac zadania ze zbioru kandydatów w π

$$E_{\pi}^{kb} \triangleq \{J_j \in J \mid j \in P_k - \{\pi(u_k)\}, P_k \cap B^j = \emptyset\}, \quad (5.74)$$

$$E_{\pi}^{ka} \triangleq \{J_j \in J \mid j \in P_k - \{\pi(u_{k+1})\}, P_k \cap A^j = \emptyset\}. \quad (5.75)$$

Zadanie ze zbioru E_{π}^{kb} będą przesuwane na pozycję u_k w π , zaś zadania ze zbioru E_{π}^{ka} na pozycję u_{k+1} . Wybór zadania do przesunięcia jest dokonywany na podstawie wyrażenia

$$\Delta_{\pi}^{kb}(j) \triangleq \begin{cases} \delta_{k-1}(\pi(u_k), j) & , j \neq \pi(u_{k+1}), \\ \delta_{k-1}(\pi(u_k), j) + \delta_{k+1}(j, j-1) & , j = \pi(u_{k+1}), \end{cases} \quad j \in E_{\pi}^{kb}, \quad (5.76)$$

$$\Delta_{\pi}^{ka}(j) \triangleq \begin{cases} \delta_{k+1}(\pi(u_{k+1}), j) & , j \neq \pi(u_k), \\ \delta_{k+1}(\pi(u_{k+1}), j) + \delta_{k+1}(j, j-1) & , j = \pi(u_k), \end{cases} \quad j \in E_{\pi}^{ka}, \quad (5.77)$$

gdzie $\delta_k(i, j) \triangleq p_{jk} - p_{ik}$, $k = 0, 1, 2, \dots, m$, $j = 1, 2, \dots, n$,
oraz $p_{j0} \triangleq r_j$.

Twierdzenie 5.7.

Niech π będzie dowolną permutacją zaś β permutacją otrzymamy z π przez przesunięcie J_j na pozycję u_k /lub u_{k+1} / w π . Niech $f_i(t)$, $J_i \in J$ będą funkcjami niemalejącymi wtedy dla $k = 1, 2, \dots, n$ otrzymujemy

$$(i) \quad f_{\max}(\beta) \geq f_{\pi(u_{m+1})}(C_{\pi(u_{m+1})} + \Delta_{\pi}^{kb}(j)), J_i \in E_{\pi}^{kb} \quad (5.78)$$

oraz

$$(ii) \quad f_{\max}(\beta) \geq f_{\pi(u_{m+1})}(C_{\pi(u_{m+1})} + \Delta_{\pi}^{ka}(j)), J_i \in E_{\pi}^{ka}. \quad \blacksquare \quad (5.79)$$

Dowód

Na mocy twierdzenia 3.5 z pracy [43] mamy

$$C_{\max}(\beta) \geq C_{\max}(\pi) + \Delta_{\pi}^{kb}(j), \quad J_j \in E_{\pi}^{kb}, \quad (5.80)$$

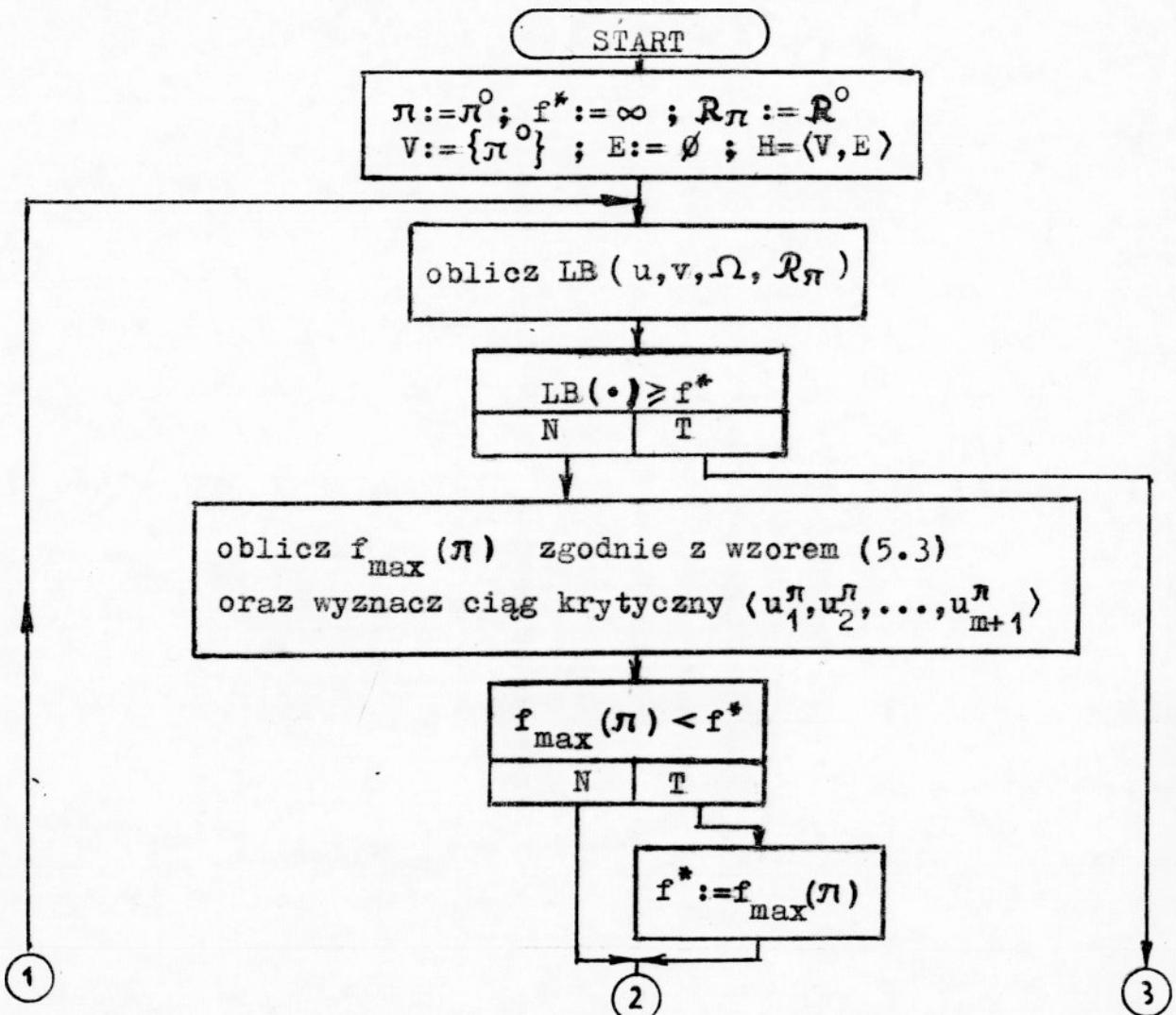
$$C_{\max}(\beta) \geq C_{\max}(\pi) + \Delta_{\pi}^{ka}(j), \quad J_j \in E_{\pi}^{ka}, \quad (5.81)$$

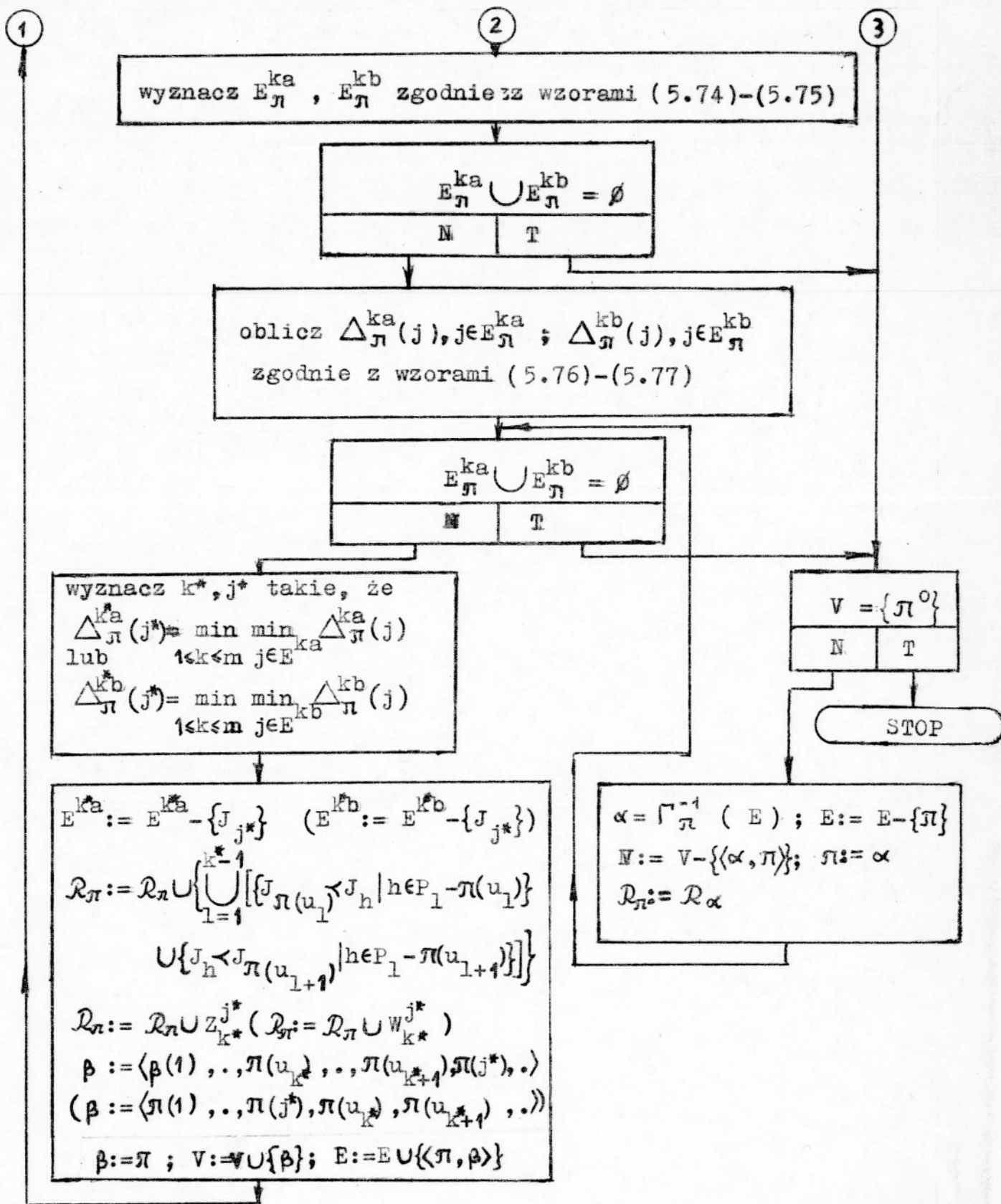
dla dowolnego problemu $n|m|P$, $r_i \geq 0 | C_{\max}$. Przyjmując $J' = \{J_{\pi(i)} \in J \mid 1 \leq i \leq u_{m+1}\}$, $|J'| = n'$ oraz korzystając z faktu że $f_i(t)$, $J_i \in J$ są funkcjami niemalejącymi otrzymujemy wzory (5.78), (5.79). ■

Do przesunięcia będziemy wybierać takie zadanie J_j , które posiada najmniejszą wartość $\Delta_{\pi}^{kb}(j)$ lub $\Delta_{\pi}^{ka}(j)$.

5.6. Algorytm

Niech $H = \langle V, E \rangle$ będzie drzewem rozwiązań / V - zbiór węzłów odpowiadających permutacjom, E - zbiór gałęzi drzewa/. Niech $\mathcal{R}^0$ będzie daną początkową relacją częściowego porządku realizacji zadań, π^0 - permutacją początkową, f^* - górnym ograniczeniem wartości funkcji celu, $\mathcal{R}_{\pi}$ - relacją częściowego porządku w węźle π





Rys. 5.4. Schemat blokowy algorytmu problemu $n|m|P, \alpha | f_{\max}$.

5.7. Realizacja programowa algorytmu

Opisany algorytm został zaprogramowany w języku ALGOL 1900 w postaci uniwersalnej procedury i był testowany na maszynie ODRA 1305. Struktura programowa procedury opiera się na zapamiętaniu zbiorów informacji π , $\bar{u}$, f^* , $LB(\cdot)$, $\mathcal{R}_\pi$, E_π^{ka} , E_π^{kb} , ..., odpowiadających kolejno generowanym węzłom, w postaci stosu oraz na zgodnym z algorytmem sposobem przetwarzania zapamiętanych informacji. Dzięki przechowywaniu całej potrzebnej informacji w pamięci operacyjnej m.c. przyspieszono maksymalnie czas przebiegu algorytmu.

W procedurze przewidziano następujące możliwości:

- dynamiczną rezerwację pamięci; pozwala to na rozwiązywanie problemów o dowolnym wymiarze oraz na maksymalne wykorzystanie pamięci do celów roboczych,
- przerywanie obliczeń w przypadku przekraczania dopuszczalnego czasu obliczeń i odczytanie najlepszego dotychczas znalezione-go rozwiązania,
- szukania rozwiązania suboptymalnego z błędem nie przekraczającym zadanej wartości ε ; $(f^* - LB(\cdot)) / LB(\cdot) \leq \varepsilon$,
- zadawanie przez użytkownika początkowego rozwiązania π^0 , w przypadku jego braku możliwe jest zastosowanie dowolnego algorytmu heurystycznego współpracującego z procedurą dla wyznaczenia π^0 ,
- kontynuowanie obliczeń po ich przerwaniu /w przypadku przydzielenia dodatkowego czasu na realizację/ bez straty informacji dotychczas uzyskanych.

W procedurze realizacja wymagań częściowego porządku $\mathcal{R}_\pi$ została zastąpiona zmianą wartości r_{ik} , q_{ik} , $k = 1, 2, \dots, n$, $k = 1, 2, \dots, m$. Taka technika postępowania dała korzystne rezultaty zarówno w pracach [64], [43], [45] jak i w szeregu eksperymentach obliczeniowych przeprowadzonych przez autora. W procedurze przyjęto następujący sposób wyznaczania wartości r_{ik} , q_{ik}

$$r_{ik} := \max[r_{ik}, r_{ik-1} + p_{ik-1}, \max_{j \in B_\pi^i}(r_{jk} + p_{jk}), \min_{j \in B_\pi^i}(r_{jk} + \sum_{j \in B_\pi^i} p_{jk})]$$

$$r_{i0} + p_{i0} = \emptyset, \quad r_{i1} = r_i \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, m \quad (5.82)$$

$$q_{ik} := \max[q_{ik}, q_{ik+1} + p_{ik+1}, \max_{j \in A_{\pi}^i}(q_{jk} + p_{jk}), \min_{j \in A_{\pi}^i}(q_{jk} + \sum_{j \in A_{\pi}^i} p_{jk})] \quad (5.83)$$

$$q_{im+1} + p_{im+1} = 0, \quad q_{im} = q_i \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, m$$

dla zadań należących dla bloku k-tego /jeśli zadanie w tym bloku są przesuwane/. Natomiast wartości r_{ik} , q_{ik} dla zadań należących do pierwszych $s-1$ bloków są w tym przypadku wyznaczane w/g zależności

$$r_{ik} := \max[r_{ik}, r_{ik-1} + p_{ik-1}, r_{\pi(u_l)} + p_{\pi(u_l)1}] \quad i \neq \pi(u_l), \quad (5.84)$$

$$r_{i0} + p_{i0} = 0, \quad r_{i1} = r_i, \quad i = 1, 2, \dots, n, \quad l = 1, 2, \dots, s-1$$

$$k = 1, 2, \dots, m$$

$$q_{ik} := \max[q_{ik}, q_{ik+1} + p_{ik+1}, q_{\pi(u_{l+1})} + p_{\pi(u_{l+1})}] \quad i \neq \pi(u_{l+1}) \quad (5.85)$$

$$q_{im+1} + p_{im+1} = 0, \quad q_{im} = q_i, \quad i = 1, 2, \dots, n, \quad l = 1, 2, \dots, s-1$$

$$k = 1, 2, \dots, m$$

$$r_{\pi(u_{l+1})k} := \max[r_{\pi(u_{l+1})k}, \sum_{j=u_l}^{u_{j+1}-1} p_{\pi(j)k} + r_{\pi(u_l)k+1}, \max_{u_l \leq j \leq u_{l+1}} (r_{\pi(j)k} + p_{\pi(j)k})] \quad k = 1, 2, \dots, m \quad (5.86)$$

$$q_{\pi(u_l)k} := \max[q_{\pi(u_l)k}, \sum_{j=u_l+1}^{u_{l+1}} p_{\pi(j)k} + q_{\pi(u_{l+1})k+1}, \max_{u_l \leq j \leq u_{l+1}} (q_{\pi(j)k} + p_{\pi(j)k})] \quad k = 1, 2, \dots, m \quad (5.87)$$

Wartości r_{ik} , q_{ik} tak wyznaczone są większe niż w pracy [43] co ma wyraźny wpływ na wartość $LB(\cdot)$.

5.8. Eksperymenty obliczeniowe

Eksperymenty miały na celu zbadanie numerycznych własności zaproponowanego algorytmu takich jak: liczba iteracji, czas obliczeń, szybkość zbiegania rozwiązania do rozwiązania optymalnego, błąd przybliżenia /w przypadku uzyskiwania rozwiązań suboptymalnych/. Elementami, które mają wpływ na podane parametry są: sposób wyznaczania dolnego ograniczenia, wartość rozwiązania początkowego,

rozmiar i wartości danych problemu, sposób realizacji wymagań częściowego porządku $\mathcal{R}_\pi$, obszar pamięci przydzielonej do celów roboczych i inne. W niniejszej pracy algorytmy testowane były z następującymi dolnymi ograniczeniami

$$LB_1 = \max [LB_0, \max_{1 \leq u \leq m} LB(u, u, (\circ \square \Delta), \mathcal{R}_\pi)], \quad (5.88)$$

$$LB_2 = \max [LB_0, \max_{1 \leq u \leq m-1} LB(u, m, (\bullet \blacksquare \blacktriangle \blacksquare \bullet), \emptyset)], \quad (5.89)$$

$$LB_3 = \max [LB_0, \max_{1 \leq u \leq n} \max_s LB^s(u, u, (\circ \square \Delta), \mathcal{R}_\pi)], \quad (5.90)$$

$$LB_4 = \max [LB_0, \max_{1 \leq u \leq m-1} \max_s LB^s(u, m, (\bullet \blacksquare \blacktriangle \blacksquare \bullet), \emptyset)], \quad (5.91)$$

gdzie

$$LB^0 = \max_{1 \leq i \leq n} f_i \left(r_i + \sum_{k=1}^m p_{ik} \right) \quad (5.92)$$

oraz $LB^s(\cdot)$ jest ciągiem dolnych ograniczeń otrzymanych przez zastosowanie twierdzenia 5.4 i 5.5.

Ponieważ problem $n|m|P, r_i \geq 0 |f_{\max}$ jest problemem nowym oraz dla problemu $n|m|P|f_{\max}$ [97] /będącego szczególnym przypadkiem/ brak jest danych numerycznych pozwalających na porównanie własności więc przeprowadzono ocenę własności algorytmu w odniesieniu do danych dla problemów $n|m|P|C_{\max}$ [81], [63] oraz $n|m|P, r_i \geq 0 |L_{\max}$ [43] będących również szczególnymi przypadkami problemu $n|m|P, r_i \geq 0 |f_{\max}$.

Rozwiązania początkowe dla algorytmu, we wszystkich przypadkach były wyznaczane następującymi metodami heurystycznymi:

- (i) reguła H1 dla problemu $n|m|P, r_i \geq 0 |L_{\max}$; uporządkować zadania zgodnie z niemalejącymi wartościami F_i gdzie

$$F_i = \sum_{k=1}^m p_{ik}^{m-n+1} + |r_i - q_i|^m \text{sign}(r_i - q_i) \quad (5.93)$$

- (ii) reguła H2 dla problemu $n|m|P|C_{\max}$; uporządkować zadania zgodnie z niemalejącymi wartościami F_i gdzie

$$F_i = \sum_{k=1}^m p_{ik}^{m-k+2}, \quad (5.94)$$

- (iii) reguła H3 dla problemu $n|m|P|C_{\max}$; uporządkować zadania

zgodnie z permutacją optymalną problemu $n|2|P|C_{\max}$ z czasami wykonywania P_{i1} , P_{i2} określonymi wzorami

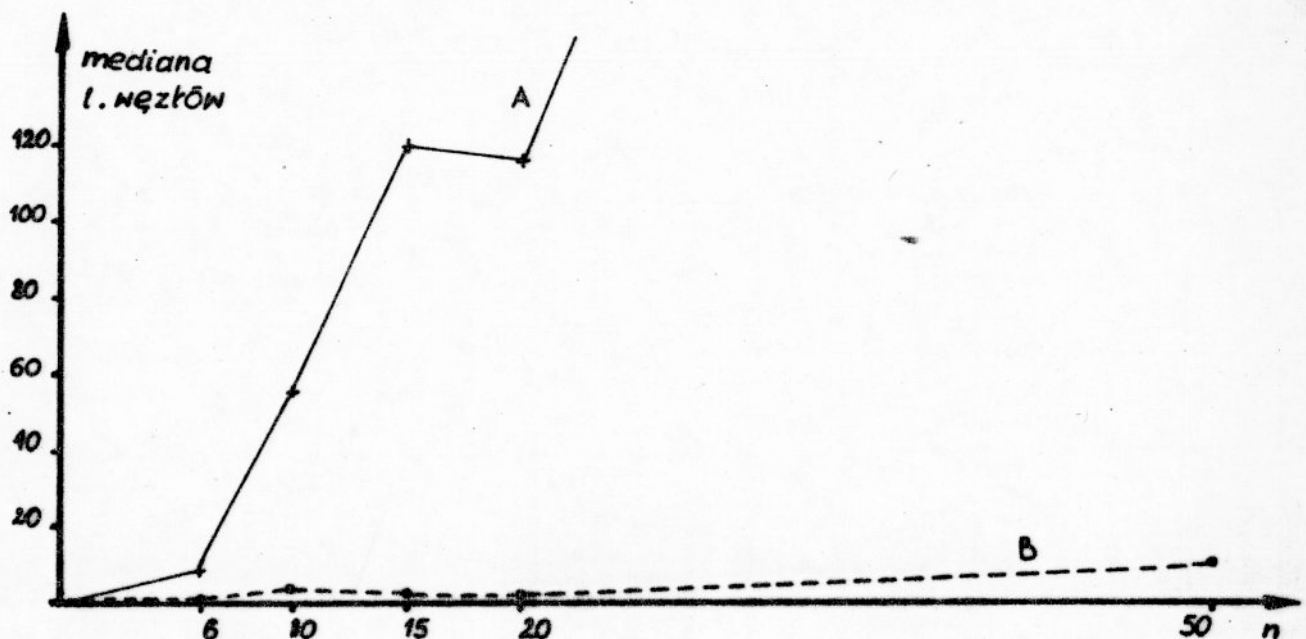
$$P_{i1} = \sum_{k=1}^{m-1} P_{ik}, \quad P_{i2} = \sum_{k=2}^m P_{ik} \quad (5.95)$$

5.9. Przykłady testujące. Wyniki obliczeniowe

Dla problemu $n|m|P|C_{\max}$ przeprowadzone testowanie na przykładach otrzymywanych od profesora Rinnooya Kana z Instytutu Ekonometriki /Rotterdam/ i dra Lenstry z Centrum Matematycznego /Amsterdam/. Zestaw obejmował 216 przykładów, których wyniki publikowano w pracach [63], [71], [82].

Każdy z testowanych przykładów był rozwiązywany 4-ro krotnie: przy użyciu reguły heurystycznej H2 oraz testów LB3 i LB4, przy użyciu reguły heurystycznej H3 oraz testów LB3 i LB4. Najlepsze uzyskane wyniki przedstawiono w tablicy 5.1. oraz na rys. 5.5.

Dla problemu $n|m|P, r_i \geq 0|L_{\max}$ przeprowadzono testowanie na 1120 przykładach, dla których wyniki publikowano w pracach [43], [44]. Każdy z przykładów był rozwiązywany 2-krotnie; przy użyciu reguły heurystycznej H1 oraz testów LB1, LB2. Wyniki przedstawiono w tablicach 5.2 - 5.5.



Rys. 5.5 .Mediana liczby węzłów dla przykładów $6|3$, $10|3$, $15|3$, $20|3$, $50|3$ problemu $n|m|P|C_{\max}$ /patrz tablica 5.1/. A- algorytm z pracy [63], B - nowy algorytm

Tablica 5.1. Wyniki obliczeń zagadnienia $n|m|P|C_{\max}$

$n m$	czas obliczeń [s]				mediana liczby węzłów*			
	algorytm [63]*		nowy algorytm**		[63] algorytm		nowy algorytm	
	b	e	H3/LB3	H3/LB4	b	e1	H3/LB3	H3/LB4
6 3	0.26	0.05	1	1	64	7	14	1
6 5	1.18	0.16	33	2	203	30	122	10
6 8	1.39	0.22	29	5	212	16	86	15
10 3		0.25		1		55		4
10 5		0.40		3		55		6
15 3		0.57		1		120		3
20 3		0.62		1		116		2
20 5		-		-		-		-
50 3		31.60		29		2356		10

Uwagi do tablicy 5.1:

* czas obliczeń na maszynie Control Data Cyber 73-28 ,

** czas obliczeń /z dokładnością do 1s/ na maszynie ODRA 1305 ,

podana liczba węzłów zawiera węzły eliminowane,

(b) zastosowano test $LB(Z, (\bullet \blacksquare \blacktriangle))$

(e1) zastosowano test $LB(W_1, (\bullet \blacksquare \blacktriangle \blacksquare \bullet))$

H3/LB3 zastosowano regułę heurystyczną H3 oraz test LB3,

H3/LB4 zastosowano regułę heurystyczną H3 oraz test LB4.

Każda informacja w tablicy dotyczy 24 przykładów testujących.

Tablica 5.2. Wyniki obliczeń zagadnienia $n|m|P, r_i \geq 0|L_{\max}$

n m	czas obliczeń [s]				liczba węzłów w drzewie H			
	LB1		LB2		LB1		LB2	
	M	S	M	S	M	S	M	S
6 2	0.1.	0.3	0.2	0.4	1	2.7	1	3.1
6 3	0.2	0.9	0.3	0.9	1	5.0	1	5.0
6 5	0.3	1.7	0.4	2.4	1	7.9	1	9.9
6 8	0.4	3.0	0.6	4.3	1	8.9	1	10.3
10 2	0.1	0.9	0.2	1.1	1	4.6	1	4.9
10 3	0.2	1.2	0.3	1.5	1	6.1	1	6.4
10 5	0.3	2.1	0.4	3.1	1	8.9	1	9.8
15 3	0.3	1.7	0.4	2.0	1	7.4	1	6.8
20 2	0.3	1.6	0.4	2.0	1	5.4	1	5.9
20 3	0.4	1.9	0.6	3.0	1	6.6	1	7.5
20 5	0.9	3.6	1.2	5.6	2	10.3	2	11.8
30 2	0.7	2.7	0.9	4.1	1	6.1	1	7.1
40 2	0.9	4.1	1.2	6.0	1	6.7	1	7.9
50 3	1.8	7.9	2.4	11.5	1	7.5	1	8.9

Uwagi do tablicy 5.2:

Każda informacja w tablicy dotyczy 80 przykładów testujących.

M - mediana, S - średnie /przykłady rozwiązane/.

Tablica 5.3. Liczba przykładów, które wygenerowały jeden węzeł w drzewie /zagadnienie

$$n|m|P, r_i \geq 0 \mid L_{\max} /$$

n m	6 2	6 3	6 5	6 8	10 2	10 3	10 5	15 3	20 2	20 3	20 5	30 2	40 2	50 3
LB1	50	50	44	41	50	50	42	42	45	42	33	53	51	49
LB2	50	50	44	41	50	50	42	42	44	41	31	52	51	49

Tablica 5.4. Liczba nierozwiązanych przykładów /w limicie czasu 60 sekund/ dla zagadnienia

$$n|m|P, r_i \geq 0 \mid L_{\max}$$

n m	6 2	6 3	6 5	6 8	10 2	10 3	10 5	15 3	20 2	20 3	20 5	30 2	40 2	50 3
LB1	-	-	-	4	2	4	16	7	7	12	18	6	8	20
LB2	-	-	-	6	8	11	20	17	12	17	20	12	13	22

Tablica 5.5. Czas obliczeń zagadnienia $n|m|P$, $r_i \geq 0$ $L_{\max}$: wpływ r i q /dla LB1/

$\frac{n m}{R Q}$	6 2	6 3	6 5	6 8	10 2	10 3	10 5	15 3	20 2	20 3	20 5	30 2	40 2	50 3
1 1	3.8	5.1	20.9	60:2	60:1	60:2	60:4	60:2	60:1	60:3	60:4	36.3	60:1	60:4
2	0.9	11.9	16.4	60:2	60:1	8:2	60:4	60:2	60:3	60:4	60:4	60:1	60:2	60:4
1n	0.2	0.8	0.9	2.0	0.4	0.8	0.9	7.2	5.9	17.8	18.9	1.9	7.8	60:2
2n	0.1	0.2	0.4	0.8	0.1	0.2	0.7	0.3	0.6	0.6	2.9	0.7	0.9	1.8
2 1	0.5	9.1	15.8	24.7	12.1	60:2	60:4	60:2	60:1	60:3	60:4	60:3	60:2	60:4
2	0.8	0.9	12.3	21.1	2.2	9.1	60:3	60:1	60:2	60:2	60:4	60:2	60:3	60:4
1n	0.2	0.2	0.6	0.8	0.3	0.2	5.3	6.2	0.5	11.4	23.1	1.8	6.4	60:2
2n	0.1	0.2	0.6	0.7	0.1	0.2	0.8	1.7	1.3	0.7	1.9	0.7	0.9	1.8
1n 1	0.8	3.5	7.1	9.9	0.2	4.2	41.3	2.9	2.6	15.9	14.1	5.9	24.9	49.1
2	0.7	0.7	0.8	5.1	7.9	5.7	60:1	0.6	3.9	0.7	60:2	3.8	30.6	32.1
1n	0.2	0.3	0.9	4.7	0.1	8.1	14.8	0.4	0.3	0.7	1.9	0.7	1.6	19.1
2n	0.2	0.3	0.4	0.7	0.1	0.2	0.3	0.3	0.3	0.4	0.5	0.7	1.6	1.8
2n 1	0.1	0.2	0.3	0.7	0.3	0.4	2.1	0.3	0.3	0.4	10.5	0.9	18.4	30.1
2	0.1	0.2	0.3	0.7	0.2	0.8	3.9	0.3	0.8	0.4	12.7	0.9	1.6	1.8
1n	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.9	0.6	0.7	2.7	0.7	1.6	1.8
2n	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4	0.3	0.7	1.8	0.9	1.9	4.9

Uwagi do tablicy 5.5:

Każda informacja w tablicy dotyczy 5 przykładów testujących

1 : k - limit czasu l został przekroczony k razy

5.10. Wnioski i uwagi

Porównanie wyników obliczeń w przypadku zagadnienia postaci $n|m|P|C_{\max}$ z wynikami obliczeń otrzymanymi przez Rinnooy Kana i Lenstrę wskazuje na znaczną przewagę zaproponowanego algorytmu. Jest to szczególnie widoczne w przykładach o dużym wymiarze, gdzie liczba iteracji jest znacznie mniejsza dla przedstawionego algorytmu. /Przy porównywaniu czasów obliczeń algorytmów należy wziąć pod uwagę różnicę szybkości, około kilkudziesięciu razy, maszyny CDC 7328 w stosunku do ODRY 1305/.

Dodatkową cechą pozytywną algorytmu jest nieduża /rzędu 7-9 %/ odległość rozwiązania początkowego od optymalnego. W przypadkach problemów nierozwiązanych w ograniczonym limicie czasu / 60 s/ wartość najlepszego rozwiązania w momencie przerwania obliczeń była odległa o rząd 3 - 5 % od dolnego ograniczenia w korzeniu drzewa rozwiązań /rozwiązania optymalnego/.

Wyniki obliczeń wykazały przewagę dolnego ograniczenia LB_3 oraz LB_4 nad pozostałymi ograniczeniami oraz przewagę algorytmu heurystycznego H3 nad H2.

Wyniki obliczeń przykładów zagadnienia $n|m|P, r_i \geq 0|L_{\max}$ pozwalają stwierdzić, że testowany algorytm /przy współpracy z procedurą heurystyczną H1/ dla 60 % przykładów wykonał tylko 1 iterację w celu znalezienia rozwiązania optymalnego. Jest to absolutnie najlepszy rezultat jakiego można się spodziewać po metodach typu podziału i ograniczeń. Dla około 9 % przykładów nie uzyskano rozwiązania optymalnego w ograniczonym limicie czasu /60 s/. Przykłady te mają rozwiązanie początkowe stosunkowo odległe od optymalnego /dolnego ograniczenia w korzeniu drzewa H/ - 6 - 12 %. Z analizy tablicy wynika, że liczba nierozwiązanych przykładów jest największa dla względnie małych wartości r_i i q_i . Może to być spowodowane zarówno gorszymi własnościami heurystyki H1 w tej klasie przykładów jak i większym "współczynnikiem trudności" problemów. Poprawa własności algorytmu jest możliwa poprzez określenie nowych, różnych procedur heurystycznych stosowanych alternatywnie w algorytmie w zależności od "współczynnika trudności" problemów.

W porównaniu do algorytmu opisanego w pracy [43] zaproponowany algorytm posiada dużo mocniejsze dostosowania wartości r_i i q_i

oraz mocniejsze dolne ograniczenia. Ma to wyraźny wpływ na zmniejszenie liczby iteracji oraz zmniejszenie liczby przykładów nierozwiązanych. Wyniki obliczeń wykazały przewagę dolnego ograniczenia LB_1 nad LB_2 .

Należy dodatkowo podkreślić, że zaprezentowany algorytm jest przeznaczony do rozwiązywania dużo ogólniejszej klasy zagadnień niż $n|m|P|C_{\max}$ i $n|m|P, r_i \geq 0|L_{\max}$.

6. MINIMALIZACJA KOSZTU MAKSYMALNEGO. ZAGADNIENIE $n|m|F, <|h_{\max}$

Dany jest zbiór zadań $J = \{J_1, J_2, \dots, J_n\}$. Każde zadanie J_i / $i = 1, 2, \dots, n$ / składa się z ciągu niepodzielnych operacji $J_i = \langle O_{i1}, O_{i2}, \dots, O_{im} \rangle$ realizowanych kolejno na maszynach $M_1, M_2, \dots, M_m$. Czas realizacji operacji O_{ik} na maszynie M_k wynosi p_{ik} / $i = 1, 2, \dots, n, k = 1, 2, \dots, m$ /. Każda maszyna może realizować co najwyżej jedno zadanie w dowolnej chwili czasowej. Rozpoczęcie realizacji zadania J_i w terminie S_i pociąga za sobą koszt realizacji zadania w wysokości $g_i(S_i)$, gdzie $g_i(S_i)$ - nierosnąca funkcja argumentu S_i / $i = 1, 2, \dots, n$ /. Zakończenie realizacji zadania J_i w terminie C_i pociąga za sobą koszt realizacji zadania w wysokości $f_i(C_i)$, gdzie $f_i(C_i)$ - niemalejąca funkcja argumentu C_i / $i = 1, 2, \dots, n$ /. Należy wyznaczyć kolejność wykonywania zadań, jednakową na wszystkich maszynach, która minimalizuje funkcję celu w postaci

$$h_{\max} = \max \left[\max_{1 \leq i \leq n} g_i(S_i), \max_{1 \leq i \leq n} f_i(C_i) \right]. \quad (6.1)$$

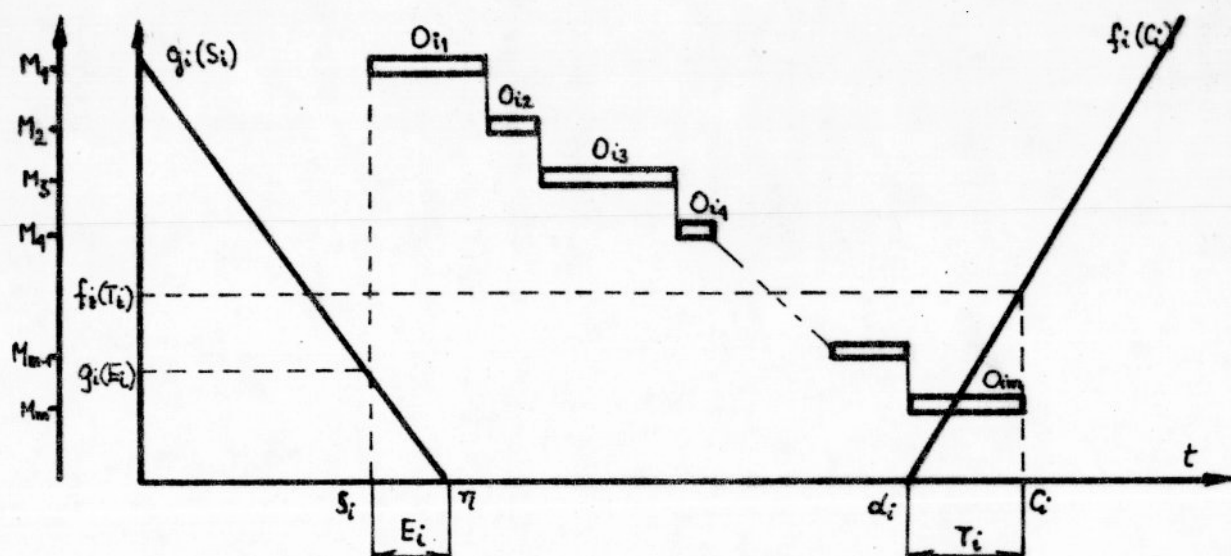
Określony sens praktyczny mają zagadnienia tego rodzaju, w których funkcje $g_i(t)$, $f_i(t)$ zależą od wartości przyspieszenia i spóźnienia, tzn. są funkcjami postaci $g_i(E_i)$ - niemalejąca funkcja argumentu E_i , $f_i(T_i)$ - niemalejąca funkcja argumentu T_i / $i = 1, 2, \dots, n$ / rys. 6.1.

W tym przypadku funkcja celu przyjmuje postać

$$h_{\max} = \max \left[\max_{1 \leq i \leq n} g_i(E_i), \max_{1 \leq i \leq n} f_i(T_i) \right]. \quad (6.2)$$

Zagadnienia kolejnościowe tego rodzaju mogą wystąpić w procesach chemicznych, lub procesach przetwórstwa spożywczego, w których surowce, półprodukty lub produkty zmieniają swoje właściwości w

czasie trwania procesu /dojrzewają, ulegają rozkładowi/. Mogą pojawić się również w procesach metalurgicznych związanych z nagrzewaniem i chłodzeniem elementów.



Rys. 6.1. Postać funkcji kosztu w problemie $n|m|F, \prec |h_{\max}$

W literaturze dotyczącej zagadnień z funkcją celu $h_{\max}$ znane są tylko 2 rezultaty szczególne: dla problemu $n|1|h_{\max}$ [B6] przy założeniach $r_i \leq r_j \Rightarrow d_i \leq d_j$ i funkcje kosztów są jednakowe, tzn. $g_i(E_i) = g(E_i)$, $f_i(T_i) = f(T_i)$, $g_i(0) = 0 = f_i(0)$, $i = 1, 2, \dots, n$ oraz dla problemu postaci $n|m|P|h_{\max}$ [2] przy założeniu, że funkcje kosztów są jednakowe, tzn. $g_i(E_i) = g(E_i)$, $f_i(T_i) = f(T_i)$, $g_i(0) = 0 = f_i(0)$, $i = 1, 2, \dots, n$.

W tym rozdziale podany zostanie ogólny algorytm rozwiązywania problemu $n|m|P, \prec |h_{\max}$ z funkcją celu postaci (6.2) /funkcje $g_i(E_i)$ $f_i(T_i)$ - są różne, $g_i(0) = 0 = f_i(0)$ /.

Zauważmy, że jeśli przyjmiemy

$$g_i(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases} \quad i = 1, 2, \dots, n, \quad (6.3)$$

oraz $d_i = 0$ i funkcje $f_i(T_i)$, $i = 1, 2, \dots, n$, są niemalejące, skończone, to problem $n|m|P, \prec |h_{\max}$ jest równoważny problemowi $n|m|P, r_i > 0, \prec |f_{\max}, f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(C_i)$ opisanemu w rozdziale 5. Istotnie, gdyby bowiem istniało rozwiązanie optymalne takie, że $E_j > 0$ dla pewnego $J_j \in J$, to $h_{\max} = \infty$; w przeciwnym przypadku

$E_i = 0$, $i = 1, 2, \dots, n$ oraz z założeń o funkcjach $f_i(T_i)$, $J_i \in J$ mamy $h_{\max} = f_{\max}$.

Zauważmy również, że w przypadku zagadnień z funkcją celu $h_{\max}$ zasada maksymalnego wykorzystania maszyn nie jest spełniona.

6.1. Pewne własności zagadnienia $n|m|P, \leq h_{\max}$

Niech π , $\Pi(Q)$ będą odpowiednio permutacją wskaźników zadań oraz zbiorem dopuszczalnych permutacji dla relacji Q , zdefiniowanymi jak w rozdz. 5.1. Niech $g_i(E_i)$, $f_i(T_i)$, $i = 1, 2, \dots, n$ będą funkcjami niemalejącymi swoich argumentów oraz $g_i(0) = 0 = f_i(0)$, $i = 1, 2, \dots, n$. Układ nierówności (6.4) - (6.6) wyznacza dopuszczalne momenty rozpoczęcia realizacji zadań S_{ik} / $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$ / na każdej maszynie, dla danej permutacji π .

$$S_{\pi(i-1)k} + p_{\pi(i-1)k} \leq S_{\pi(i)k} \quad i=1, 2, \dots, n, \quad k=1, 2, \dots, m, \quad (6.4)$$

$$S_{\pi(0)k} + p_{\pi(0)k} = t_0 \quad k = 1, 2, \dots, m, \quad (6.5)$$

$$S_{\pi(i)k} + p_{\pi(i)k} \leq S_{\pi(i)k+1} \quad i=1, 2, \dots, n, \quad k=1, 2, \dots, m-1. \quad (6.6)$$

Zagadnienie optymalizacji polega na znalezieniu takiej permutacji π^* oraz takich momentów rozpoczęcia wykonywania zadań na każdej maszynie S_{ik} aby

$$h_{\max}(\pi^*) = \min_{\pi \in \Pi(Q)} h_{\max}(\pi), \quad (6.7)$$

gdzie

$$h_{\max}(\pi) = \min_{S_{ik} \in S_{\pi}} \max [\max_{1 \leq i \leq n} g_i(E_i), \max_{1 \leq i \leq n} f_i(T_i)], \quad (6.8)$$

$$E_i \triangleq \max [0, r_i - S_{i1}], \quad (6.9)$$

$$T_i \triangleq \max [0, C_{im} - d_i] = \max [0, S_{im} + p_{im} - d_i], \quad (6.10)$$

zaś zbiór S_{π} jest wyznaczony wzorami (6.4) - (6.6).

Zauważmy dodatkowo, że problem wyznaczenia wartości S_{ik} / $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$ / minimalizujących koszt maksymalny (6.8) dla danej permutacji π jest problemem nietrywialnym, bowiem nie jest spełniona zasada maksymalnego wykorzystania maszyn. W dalszej części rozdziału zostanie sformułowany odpowiedni algorytm

rozwiązania tego problemu.

Niech π będzie permutacją. Z definicji przyspieszenia i spóźnienia mamy

$$E_{\pi(i)} = \max [0, r_{\pi(i)} - S_{\pi(i)1}] \geq r_{\pi(i)} - S_{\pi(i)1} \quad i = 1, 2, \dots, n \quad (6.11)$$

$$T_{\pi(j)} = \max [0, C_{\pi(j)m} - d_{\pi(j)}] \geq C_{\pi(j)m} - d_{\pi(j)} \quad j = 1, 2, \dots, n \quad (6.12)$$

Dodając (6.11) i (6.12) stronami otrzymujemy

$$E_{\pi(i)} + T_{\pi(j)} \geq r_{\pi(i)} + C_{\pi(j)m} - S_{\pi(i)1} - d_{\pi(j)}, \quad 1 \leq i \leq j \leq n \quad (6.13)$$

Korzystając z wzoru (5.5) oraz z zależności

$$S_{\pi(i)1} = \sum_{h=1}^{i-1} p_{\pi(h)1}, \quad i = 1, 2, \dots, n \quad (6.14)$$

otrzymujemy

$$\begin{aligned} E_{\pi(i)} + T_{\pi(j)} &\geq r_{\pi(i)} + \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq j} \left(\sum_{h=1}^{i_1} p_{\pi(h)1} + \dots + \sum_{h=i_m}^j p_{\pi(h)m} - \right. \\ &- \left. \sum_{h=1}^{i-1} p_{\pi(h)1} \right) - d_{\pi(j)} = r_{\pi(i)} + \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{m-1} \leq j} \left(\sum_{h=i_1}^{i_1} p_{\pi(h)1} + \dots + \right. \\ &+ \dots + \left. \sum_{h=i_{m-1}}^j p_{\pi(h)m} \right) - d_{\pi(j)} = \Delta_{\pi(i), \pi(j)}. \end{aligned} \quad (6.15)$$

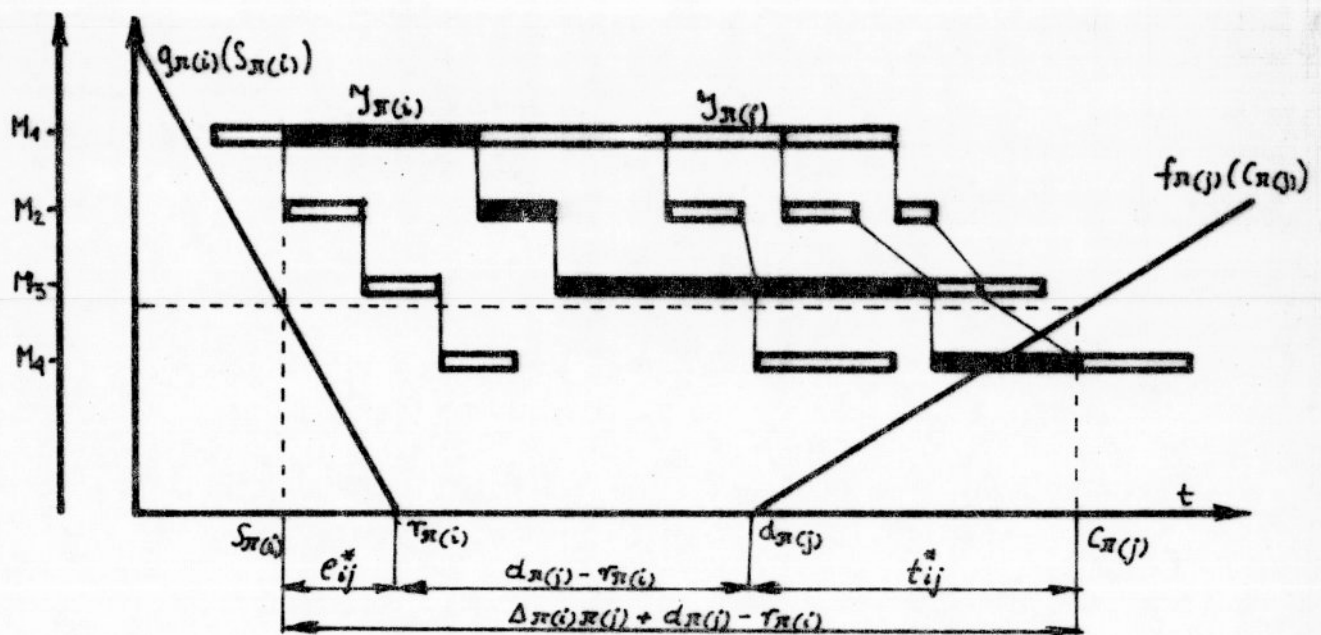
Rozpatrzmy dalej następujący problem: wyznaczyć e_{ij}^* , t_{ij}^* spełniające warunki (6.16) - (6.18), rys. 6.2.

$$g_{\pi(i)}(e_{ij}^*) = f_{\pi(j)}(t_{ij}^*) \quad , \quad (6.16)$$

$$e_{ij}^* + t_{ij}^* = \Delta_{\pi(i), \pi(j)} \quad , \quad (6.17)$$

$$t_{ij}^*, e_{ij}^* \geq 0 \quad . \quad (6.18)$$

Zauważmy, że jeśli $\Delta_{\pi(i), \pi(j)} \leq 0$ to $e_{ij}^* = 0 = t_{ij}^*$ oraz $g_{\pi(i)}(e_{ij}^*) = 0 = f_{\pi(j)}(t_{ij}^*)$ w sposób oczywisty.



Rys. 6.2. Interpretacja graficzna rozwiązania problemu
(6.16) - (6.18)

Twierdzenie 6.1.

Niech $\pi \in \Pi$ będzie dowolną permutacją określającą kolejność wykonywania zadań na maszynach. Niech $g_i(E_i)$, $f_i(T_i)$ będą funkcjami niemalejącymi swoich argumentów oraz $g_i(0) = 0 = f_i(0)$, $i = 1, 2, \dots, n$. Wówczas zachodzi

$$h_{\max}(\pi) = \max \left[\max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi) \right], \quad (6.19)$$

gdzie $f_{\max}(\pi)$ jest wartością funkcji celu problemu $n|m|P$, $\prec f_{\max}$
 $f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(T_i)$ dla permutacji π .

Dówód

Pokażemy wpierw, że prawa strona wzoru (6.19) jest dolnym ograniczeniem wartości $h_{\max}(\pi)$. Następnie pokażemy sposób wyznaczania dopuszczalnych momentów rozpoczęcia zadań, który zapewnia że otrzymana wartość funkcji celu będzie równa poprzednio wyznaczonemu dolnemu ograniczeniu.

>. Zauważmy, że nierówność

$$h_{\max}(\pi) \geq f_{\max}(\pi) \quad (6.20)$$

jest oczywista bowiem problem $n|m|P, \prec |f_{\max}$ jest relaksacją problemu $n|m|P, \prec |h_{\max}$.

Niech dalej i^*, j^* , $j^* \leq j^*$ będą takimi wskaźnikami dla których wartość $g_{\pi(i)}(e_{ij})$ /we wzorze 6.19/ osiąga maksimum. Wówczas zachodzi

$$E_{\pi(i^*)} + T_{\pi(j^*)} \geq \Delta_{\pi(i^*) \pi(j^*)} = e_{i^* j^*}^* + t_{i^* j^*}^*, \quad (6.21)$$

tzn., że spełniona jest co najmniej jedna z nierówności

$$E_{\pi(i^*)} \geq e_{i^* j^*}^* \quad \text{lub} \quad T_{\pi(j^*)} \geq t_{i^* j^*}^*. \quad (6.22), (6.22)$$

Ponieważ funkcje $g_i(E_i)$, $f_i(T_i)$ / $i = 1, 2, \dots, n$ / są niemalejące, więc spełniona jest, co najmniej jedna z nierówności

$$g_{\pi(i^*)}(E_{\pi(i^*)}) \geq g_{\pi(i^*)}(e_{i^* j^*}^*) \quad (6.23)$$

lub

$$f_{\pi(j^*)}(T_{\pi(j^*)}) \geq f_{\pi(j^*)}(t_{i^* j^*}^*). \quad (6.23)$$

Prawe strony nierówności (6.23) - (6.23) są sobie równe, czyli

$$g_{\pi(i)}(e_{i^* j^*}^*) \leq \max [g_{\pi(i^*)}(E_{\pi(i^*)}), f_{\pi(j^*)}(T_{\pi(j^*)})] \leq \\ \max [\max_{1 \leq i \leq n} g_i(E_i), \max_{1 \leq i \leq n} f_i(T_i)] = h_{\max}(\pi). \quad (6.24)$$

Łącząc otrzymany warunek z warunkiem (6.20) otrzymujemy

$$h_{\max}(\pi) \geq \max [\max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi)], \quad (6.25)$$

co kończy część pierwszą dowodu.

≤. Wyznaczmy wartości

$$\hat{E}_i = \min \{ h \mid g_i(h) = \max [g_{\pi(i^*)}(e_{i^* j^*}^*), f_{\max}(\pi)] \} \quad (6.26)$$

oraz dopuszczalne momenty rozpoczęcia zadań w następujący sposób

$$S_{\pi(j)1} = \max [S_{\pi(j-1)1} + p_{\pi(j-1)1}, r_{\pi(j)} - \hat{E}_{\pi(j)}] \\ j = 1, 2, \dots, n, \quad (6.27)$$

$$S_{\pi(0)1} + p_{\pi(0)1} = t_0, \quad (6.28)$$

$$S_{\pi(j)k} = \max [S_{\pi(j-1)k} + p_{\pi(j-1)k}, S_{\pi(j)k-1} + p_{\pi(j)k-1}],$$

$$j = 1, 2, \dots, n; \quad k = 2, 3, \dots, m \quad (6.29)$$

$$S_{\pi(0)k} = 0 \quad k = 2, 3, \dots, m. \quad (6.30)$$

Wzory (6.27) - (6.30) można również zapisać w formie

$$S_{\pi(j)1} = \max_{1 \leq i \leq j} \max [r_{\pi(i)}, t_0 + \sum_{h=1}^{j-1} p_{\pi(h)1}] \quad j = 1, 2, \dots, n, \quad (6.31)$$

$$C_{\pi(j)k} = \max_{1 \leq i_1 \leq \dots \leq i_k \leq j} (S_{\pi(i_1)1} + \sum_{h=1}^{i_1-1} p_{\pi(h)1} + \sum_{h=i_1}^{i_2-1} p_{\pi(h)2} + \dots +$$

$$+ \sum_{h=i_{k-1}}^j p_{\pi(h)k}), \quad j = 1, 2, \dots, m$$

$$k = 1, 2, \dots, m. \quad (6.32)$$

Z wzoru (6.27) wynika, że

$$S_{\pi(i)1} \geq r_{\pi(i)} - \hat{E}_{\pi(i)}, \quad i = 1, 2, \dots, n. \quad (6.33)$$

Uwzględniając powyższą nierówność otrzymujemy

$$E_{\pi(i)} = \max [0, r_{\pi(i)} - S_{\pi(i)1}] = \max [0, r_{\pi(i)} - r_{\pi(i)} + \hat{E}_{\pi(i)}] =$$

$$= \max [0, \hat{E}_{\pi(i)}] = \hat{E}_{\pi(i)} \quad (6.34)$$

oraz

$$g_{\pi(i)}(E_{\pi(i)}) \leq g_{\pi(i^*)}(E_{\pi(i^*)}) = g_{\pi(i^*)}(e_{i^*j^*}^*) \leq$$

$$\max [\max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi)], \quad i = 1, 2, \dots, n. \quad (6.35)$$

Podstawiając następnie wzór (6.31) do wzoru (6.32) otrzymujemy dla $k = m$

$$C_{\pi(j)m} = \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{m-1} \leq j} (\max [r_{\pi(i_1)} - \hat{E}_{\pi(i_1)}, t_0 + \sum_{h=1}^{i_1-1} p_{\pi(h)1} +$$

$$+ \dots + \sum_{h=i_{m-1}}^j p_{\pi(h)m}]), \quad j = 1, 2, \dots, n. \quad (6.36)$$

Niech i^* będzie taką wartością wskaźnika i dla którego wyrażenie z prawej strony wzoru (6.36) osiąga maksimum. Wówczas możliwe są dwa następujące przypadki:

$$(i) \quad r_{\pi(i^*)} - \hat{E}_{\pi(i^*)} > t_0 \quad \text{wtedy}$$

$$\begin{aligned} C_{\pi(j)m} &= \max_{i^* \leq i_1 \leq i_2 \leq \dots \leq i_{m-1} \leq j} (r_{\pi(i^*)} - \hat{E}_{\pi(i^*)} + \sum_{h=i^*}^{i_1} p_{\pi(h)1} + \dots \\ &\quad \dots + \sum_{h=i_{m-1}}^j p_{\pi(h)m}) = \Delta_{\pi(i^*)\pi(j)} - \hat{E}_{\pi(i^*)} + d_{\pi(j)} = \\ &= e_{i^*j}^* + t_{i^*j}^* - \hat{E}_{\pi(i^*)} + d_{\pi(j)} \quad j = 1, 2, \dots, n \quad (6.37) \end{aligned}$$

ale $\hat{E}_{\pi(i^*)} \geq e_{i^*j}^*$, więc

$$C_{\pi(j)m} \leq t_{i^*j}^* + d_{\pi(j)} \quad j = 1, 2, \dots, n \quad (6.38)$$

oraz

$$T_{\pi(j)} = \max [0, C_{\pi(j)m} - d_{\pi(j)}] \leq t_{i^*j}^* \quad j = 1, 2, \dots, n. \quad (6.39)$$

W konsekwencji otrzymujemy

$$\begin{aligned} f_{\pi(j)}(T_{\pi(j)}) &\leq f_{\pi(j)}(t_{i^*j}^*) \leq \max_{1 \leq i \leq j \leq n} f_{\pi(j)}(t_{ij}^*) = \\ &= \max_{1 \leq i \leq n} g_{\pi(i)}(e_{ij}^*) \leq \max \left[\max_{1 \leq i \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi) \right] \\ &\quad j = 1, 2, \dots, n, \quad (6.40) \end{aligned}$$

$$(ii) \quad r_{\pi(i^*)} - \hat{E}_{\pi(i^*)} \leq t_0 \quad \text{wtedy } i^* = 1 \quad \text{oraz}$$

$$\begin{aligned} C_{\pi(j)m} &= \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{m-1} \leq j} \left[t_0 + \sum_{h=1}^{i_1} p_{\pi(h)1} + \dots + \sum_{h=i_{m-1}}^j p_{\pi(h)m} \right] \\ &\quad j = 1, 2, \dots, n \quad (6.41) \end{aligned}$$

Uwzględniając (6.41) możemy zapisać

$$\begin{aligned} \max_{1 \leq j \leq n} f_{\pi(j)}(T_{\pi(j)}) &= \max_{1 \leq j \leq n} f_{\pi(j)} \left(\max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_{m-1} \leq j \leq n} \left[0, t_0 + \right. \right. \\ &+ \sum_{h=1}^{i_1} p_{\pi(h)1} + \dots + \sum_{h=i_{m-1}}^j p_{\pi(h)m} - d_{\pi(j)} \left. \right] \Big) = \max_{1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq n} f_{\pi(i_m)} \left(\max [0, t_0 + \right. \\ &+ \sum_{h=1}^{i_1} p_{\pi(h)1} + \sum_{h=i_{m-1}}^{i_m} p_{\pi(h)m} - d_{\pi(i_m)} \left. \right] \Big) \triangleq f_{\max}(\pi). \quad (6.42) \end{aligned}$$

Zauważmy, że wzór (6.42) określa koszt maksymalny w problemie $n|m|P, \prec |f_{\max}, f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(T_i)$. Tak więc prawdziwa ^{jest} zależność

$$\max_{1 \leq j \leq n} f_{\pi(j)}(T_{\pi(j)}) = f_{\max}(\pi) \leq \max_{1 \leq i \leq n} [\max_{1 \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi)] \quad (6.43)$$

Łącząc warunki (6.35), (6.40), (6.43) otrzymujemy

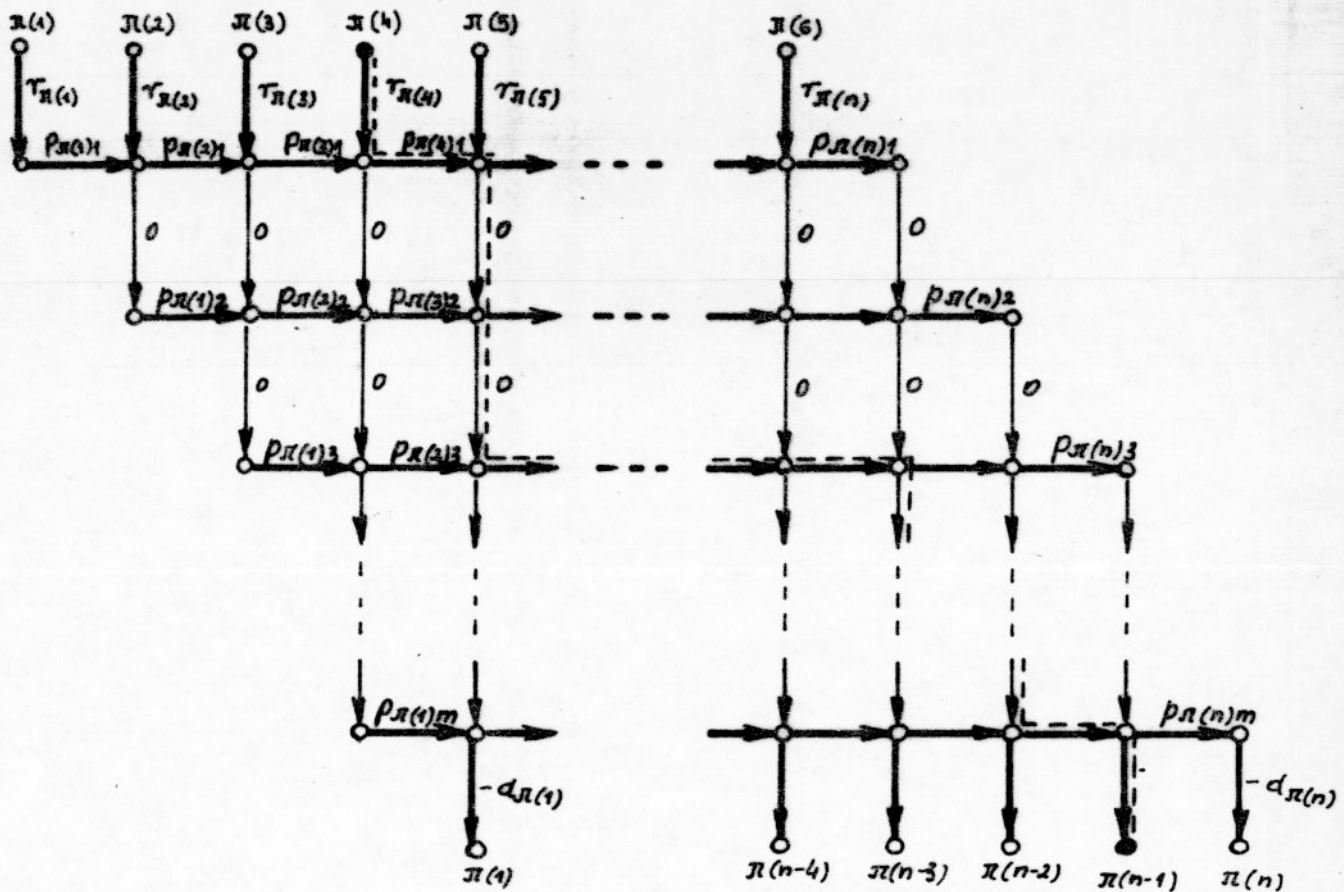
$$h_{\max}(\pi) = \max_{1 \leq i \leq n} [\max_{1 \leq j \leq n} g_i(E_j), \max_{1 \leq i \leq n} f_i(T_i)] \leq \max_{1 \leq i \leq n} [\max_{1 \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi)], \quad (6.44)$$

co dowodzi, że wartość funkcji celu w przypadku momentów rozpoczęcia zadań wyznaczonych wzorami (6.27) - (6.30) jest nie większa niż jej dolne ograniczenie. Uwzględniając więc warunki (6.25) i (6.44) otrzymujemy we wzorze (6.19) poszukiwaną równość. ■

Z dowodu twierdzenia 6.1 wynika następujący algorytm wyznaczania momentów rozpoczęcia zadań dla danej permutacji π . W tym celu należy kolejno:

- (i) wyznaczyć wartości $\Delta_{\pi(i)\pi(j)}$, wg wzoru (6.15) dla każdej pary $1 \leq i \leq j \leq n$; problem ten jest równoważny problemowi wyznaczenia najdłuższych dróg między wybranymi parami węzłów w sieci o specjalnej strukturze /rys. 6.3/ i jego rozwiązanie wymaga $O(n^2 m)$ iteracji,
- (ii) rozwiązać $n(n-1)/2$ prostych podproblemów postaci (6.16) - (6.18), z których każdy zawiera jedną zmienną niezależną; w przypadku gdy istnieje analityczne wyrażenie określające wartość rozwiązania w zależności od $\Delta_{\pi(i)\pi(j)}$ jako parametru, to każdy z podproblemów polega na wyliczeniu pojedynczej wielkości z odpowiedniego wzoru dla danej wartości $\Delta_{\pi(i)\pi(j)}$ /analogicznie jak na rys. 6.5/,
- (iii) wyznaczyć wartość $f_{\max}(\pi)$ problemu $n|m|P, \prec |f_{\max}$ zgodnie z (6.42); wyznaczenie $f_{\max}(\pi)$ wymaga $O(nm)$ iteracji,
- (iv) spośród wyznaczonych wartości $g_{\pi(i)}(e_{ij}^*) / 1 \leq i \leq j \leq n$ oraz $f_{\max}(\pi)$ wybrać wartość maksymalną $h_{\max}(\pi)$; $h_{\max}(\pi)$ wyznacza minimalną wartość kosztu maksymalnego dla permutacji π w problemie $n|m|P, \prec |h_{\max}$,
- (v) określić $\hat{E}_i / i = 1, 2, \dots, n$ zgodnie z (6.26),
- (vi) wyznaczyć, zgodnie z wzorami (6.27) - (6.30), momenty rozpoczęcia wykonywania zadań na wszystkich maszynach;

czynność ta wymaga $O(nm)$ iteracji.



Rys. 6.3. Struktura sieci do wyznaczenia wartości $\Delta_{\pi(i)\pi(j)}$.
Zaznaczono obciążenia łuków oraz przykładowo drogę
odpowiadającą $\Delta_{\pi(4)\pi(n-1)}$

W szczególnym przypadku, gdy przyjmiemy założenie, że $g_i(E_i) = g(E_i)$, $f_i(T_i) = f(T_i)$, $i = 1, 2, \dots, n$ /patrz [86], [2] /, to powyższy schemat postępowania upraszcza się do następującego schematu:

(i) wyznaczyć wartość

$$\Delta_{\max, \pi} = \max_{1 \leq i \leq j \leq n} \Delta_{\pi(i)\pi(j)} ; \quad (6.45)$$

wyznaczenie wartości $\Delta_{\max, \pi}$ wymaga $O(nm)$ iteracji, bowiem $\Delta_{\max, \pi} = L_{\max}(\pi)$, gdzie $L_{\max}(\pi)$ jest wartością funkcji celu problemu $n|n|P$, $r_i \geq 0$, $\forall i \in L_{\max}$,

(ii) jeśli $\Delta_{\max, \pi} \leq 0$ to przyjąć $E^* = 0 = T^*$, $g(E^*) = 0 = f(T^*)$,
jeśli $\Delta_{\max, \pi} > 0$ to wyznaczyć wartości E^* , T^* takie, że

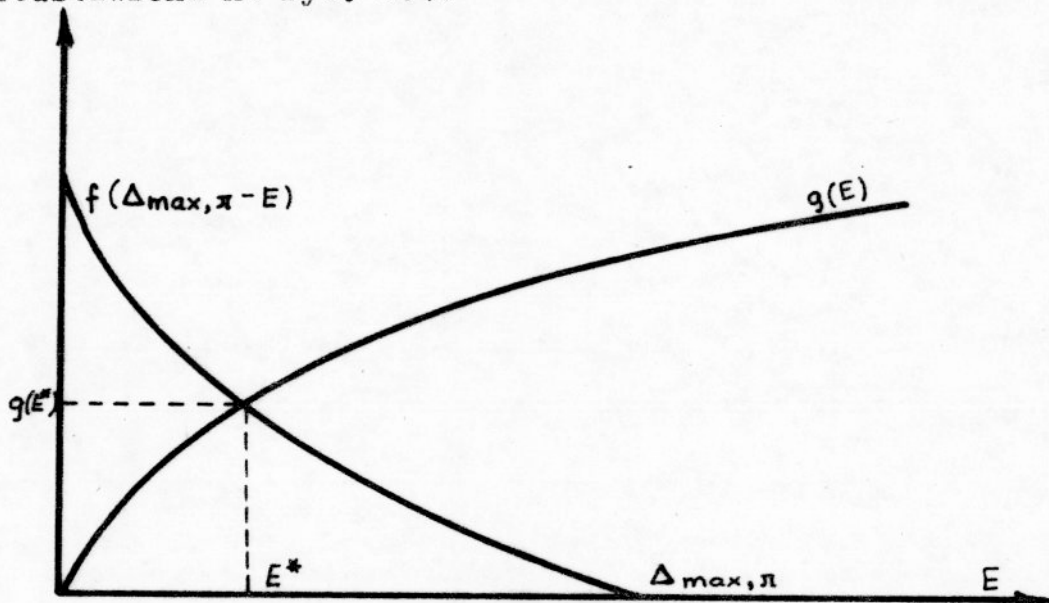
$$g(E^*) = f(T^*) \quad , \quad (6.46)$$

$$E^* + T^* = \Delta_{\max}, \quad , \quad (6.47)$$

$$E^*, T^* \geq 0 \quad , \quad (6.48)$$

- (iii) wyznaczyć wartość $T_{\max}(\pi)$ problemu $n|m|P$, $\prec |T_{\max}$, a następnie przyjmując $h_{\max}(\pi) = \max [g(E^*), f(T_{\max}(\pi))]$,
 (iv) określić $\hat{E}_i$, $i = 1, 2, \dots, n$ zgodnie z (6.26). Czynność ta wymaga $O(1)$ iteracji, bowiem wszystkie wartości E_i są sobie równe $\hat{E}_i = \hat{E}$, $/i = 1, 2, \dots, n/$,
 (v) wyznaczyć zgodnie z wzorami (6.27) - (6.30) momenty rozpoczęcia wykonywania zadań na wszystkich maszynach.

Ilustracja graficzna rozwiązania problemu (6.46) - (6.48) została przedstawiona na rys. 6.4.



Rys. 6.4. Graficzne rozwiązanie problemu (6.46) - (6.48)

Interesującym wydaje się fakt, że problem w szczególnym przypadku, gdy $g_i(E_i) = g(E_i)$, $f_i(T_i) = f(T_i)$, $i = 1, 2, \dots, n$ jest równoważny problemowi $n|m|P$, $r_i \geq 0$ $|T_{\max} [2]$, jednakże w ogólnym przypadku równoważności nie udowodniono. Ciąg $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ spełniający warunek $1 \leq u_1^\pi \leq u_2^\pi \leq \dots \leq u_{m+1}^\pi \leq n$ będziemy nazywać ciągami krytycznymi w π jeśli

- (i) $h_{\max}(\pi) = f_{\max}(\pi)$ oraz $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ jest ciągiem krytycznym problemu $n|m|P$, $f_{\max}$ /patrz wzór (5.4) dla $r_i = 0$, $i = 1, 2, \dots, n/$, lub

(ii) $h_{\max}(\pi) = \max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*) = g_{\pi(i^*)}(e_{i^*j^*}^*)$ oraz $u_1^\pi = i^*$,
 $u_{m+1}^\pi = j^*$, $u_2^\pi = \hat{u}_1^\pi$, $u_3^\pi = \hat{u}_2^\pi, \dots, u_m^\pi = \hat{u}_{m-1}^\pi$ gdzie
 $\langle \hat{u}_1^\pi, \hat{u}_2^\pi, \dots, \hat{u}_{m-1}^\pi \rangle$, $i^* \leq \hat{u}_1^\pi \leq \hat{u}_2^\pi \leq \dots \leq \hat{u}_{m-1}^\pi \leq j^*$ jest ciągiem
maksymalizującym wartość $\Delta_{\pi(i^*)\pi(j^*)}$ we wzorze (6.15).

Własność 6.1.

W przypadku każdej $\pi \in \Pi$ mamy

$$h_{\max}(\pi) \geq \tilde{h}_{\max}(\pi), \quad (6.49)$$

gdzie $\tilde{h}_{\max}(\pi)$ jest wartością funkcji celu dla dowolnego ciągu /niekoniecznie krytycznego/ w π . ■

Dowód

Niech $\langle \tilde{u}_1^\pi, \tilde{u}_2^\pi, \dots, \tilde{u}_{m+1}^\pi \rangle$ będzie dowolnym ciągiem /niekoniecznie krytycznym/ w π . Oznaczmy przez $\tilde{f}_{\max}(\pi)$ wartość funkcji celu problemu n|P, $\langle \tilde{f}_{\max}$ dla ciągu $\langle \tilde{u}_1^\pi, \tilde{u}_2^\pi, \dots, \tilde{u}_{m+1}^\pi \rangle$.

Niech dalej $g_{\pi(\tilde{i})}(\tilde{e}_{\tilde{i}\tilde{j}})$ będzie rozwiązaniem problemu (6.16) - (6.18) dla ciągu $\langle \tilde{u}_1^\pi, \tilde{u}_2^\pi, \dots, \tilde{u}_{m+1}^\pi \rangle$, tzn. $\tilde{u}_1^\pi = \tilde{i}$, $\tilde{u}_{m+1}^\pi = \tilde{j}$ oraz

$$g_{\pi(\tilde{i})}(\tilde{e}_{\tilde{i}\tilde{j}}) = f_{\pi(\tilde{j})}(\tilde{t}_{\tilde{i}\tilde{j}}), \quad (6.50)$$

$$\begin{aligned} \tilde{e}_{\tilde{i}\tilde{j}} + \tilde{t}_{\tilde{i}\tilde{j}} = \tilde{\Delta}_{\pi(\tilde{i})\pi(\tilde{j})} \triangleq r_{\pi(\tilde{i})} + \sum_{h=\tilde{i}}^{\tilde{u}_2^\pi} p_{\pi(h)1} + \dots + \sum_{h=\tilde{u}_m^\pi}^{\tilde{j}} p_{\pi(h)m} - \\ - d_{\pi(\tilde{j})}, \end{aligned} \quad (6.51)$$

$$\tilde{e}_{\tilde{i}\tilde{j}}, \tilde{t}_{\tilde{i}\tilde{j}} \geq 0. \quad (6.52)$$

Zauważmy, że ze wzoru (5.10) wynika następująca nierówność

$$h_{\max}(\pi) \geq \max_{1 \leq i \leq j \leq n} [g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi)] \geq f_{\max}(\pi) \quad (6.53)$$

Niech dalej $g_{\pi(\tilde{i})}(e_{\tilde{i}\tilde{j}}^*)$ będzie rozwiązaniem problemu (6.16) - (6.18) dla $\Delta_{\pi(\tilde{i})\pi(\tilde{j})}$ określonego wzorem (6.15).

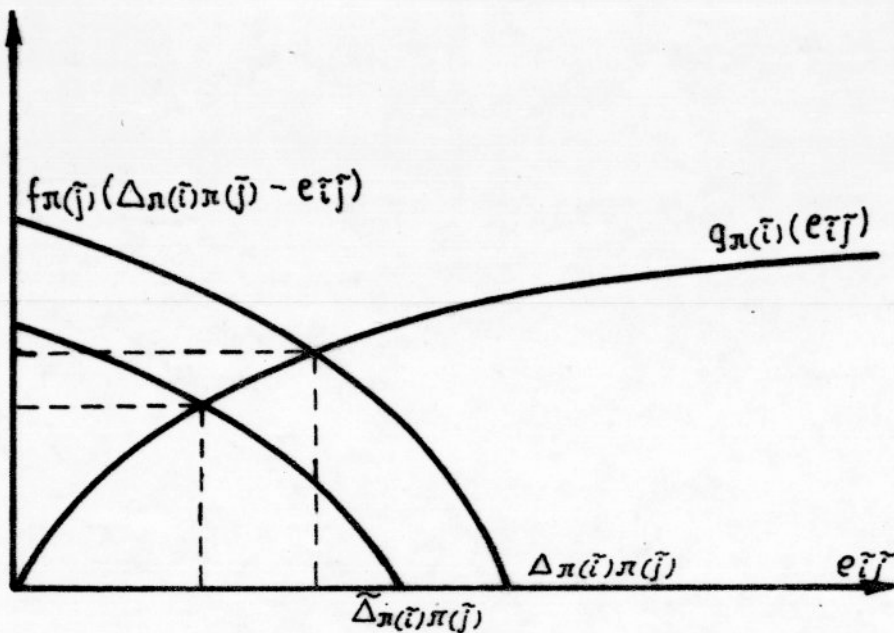
Wówczas na mocy wzoru (5.10) zachodzi

$$\Delta_{\pi(\tilde{i})\pi(\tilde{j})} \geq \Delta_{\pi(\tilde{i})\pi(\tilde{j})} \quad (6.54)$$

oraz z własności problemu (6.16) - (6.18) zachodzi

$$g_{\pi(\tilde{i})}(e_{\tilde{i}\tilde{j}}^*) \geq g_{\pi(\tilde{i})}(\tilde{e}_{\tilde{i}\tilde{j}}), \quad (6.55)$$

co zilustrowano na rys. 6.5.



Rys. 6.5. Wpływ parametru $\Delta \pi(i) \pi(j)$ na wartość rozwiązania problemu (6.16) - (6.18)

W konsekwencji możemy więc zapisać

$$\max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*) \geq g_{\pi(i)}(e_{ij}^*) \geq g_{\pi(i)}(\tilde{e}_{ij}), \quad (6.56)$$

co w połączeniu z wzorem (6.53) daje

$$h_{\max}(\pi) \geq \max[g_{\pi(i)}(\tilde{e}_{ij}), \tilde{f}_{\max}(\pi)] \triangleq \tilde{h}_{\max}(\pi). \quad (6.57)$$

Wartość $\tilde{h}_{\max}(\pi)$ jest kosztem związanym z ciągiem $\langle \tilde{u}_1^\pi, \tilde{u}_2^\pi, \dots, \tilde{u}_{m+1}^\pi \rangle$ /dokładniej kosztem zadania $J_\pi(\tilde{u}_{m+1}^\pi)$ /.

Przyjmijmy w dalszym ciągu definicje bloku, bloku wewnętrznego. Z rozdziału 5.1 wraz ze stosowanymi tam oznaczeniami.

Twierdzenie 6.2.

Niech π będzie permutacją z ciągiem krytycznym $\langle u_1, u_2, \dots, \dots, u_{m+1} \rangle$ w problemie $n|m|P$, $\prec |h_{\max}$. Niech $g_i(t)$, $f_i(t)$, $i = 1, 2, \dots, n$ będą funkcjami spełniającymi założenia twierdzenia 6.1. Jeżeli istnieje permutacja β taka, że

$$h_{\max}(\beta) < h_{\max}(\pi), \quad (6.58)$$

wtedy w β przynajmniej jedno zadanie dla przynajmniej jednego bloku z π poprzedza pierwsze zadanie /lub występuje za ostatnim

zadaniem/ tego bloku . ■

Dowód

Zauważmy, że jeśli $h_{\max}(\pi) = f_{\max}(\pi)$ /wzór 6.19/, to prawdziwość twierdzenia 6.2 wynika z twierdzenia 5.2. Rozpatrzmy zatem przypadek, gdy $h_{\max}(\pi) = \max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*)$. Załóżmy, bez straty ogólności rozważań, że $\pi(i) = i$, $i = 1, 2, \dots, n$ oraz $\pi(u_k) = u_k$, $k = 1, 2, \dots, m+1$. Dowód zostanie przeprowadzony przez zaprzeczenie. Załóżmy, że β jest permutacją taką, że $h_{\max}(\beta) < h_{\max}(\pi)$ oraz teza twierdzenia nie jest spełniona. Wówczas β jest permutacją postaci

$$\beta = \langle \beta(1), \dots, \beta(u_1-1), u_1, \beta(u_1+1), \dots, \beta(u_2-1), u_2, \beta(u_2+1), \dots, u_{k+1}, \dots, u_{m+1}, \dots, \beta(n) \rangle,$$

gdzie $\beta(u_k+1), \beta(u_k+2), \dots, \beta(u_{k+1}-1)$ $k = 1, 2, \dots, m$ są permutacjami zadań k -tego wewnętrznego bloku, tzn. permutacjami zadań o wskaźnikach $u_k+1, u_k+2, \dots, u_{k+1}-1$, zaś $\beta(1), \beta(2), \dots, \beta(u_1-1)$ oraz $\beta(u_{m+1}+1), \beta(u_{m+1}+2), \dots, \beta(n)$ są odpowiednio permutacjami zadań o wskaźnikach $1, 2, \dots, u_1-1$ i $u_{m+1}+1, \dots, n$. Zauważmy dalej, że ciąg krytyczny $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ w π jest tylko pewnym ciągiem, niekoniecznie krytycznym, w β . Wobec powyższego, zgodnie z własnością 6.1 możemy zapisać

$$h_{\max}(\beta) \geq \tilde{h}_{\max}(\beta) \quad (6.59)$$

oraz

$$\tilde{h}_{\max}(\beta) = h_{\max}(\pi), \quad (6.60)$$

bowiem zachodzi

$$\begin{aligned} \tilde{\Delta}_{\beta(u_1) \beta(u_{m+1})} &\triangleq r_{\beta(u_1)} + \sum_{h=u_1}^{u_2} p_{\beta(h)1} + \dots + \sum_{h=u_m}^{u_{m+1}} p_{\beta(h)m} - d_{\beta(u_{m+1})} \\ &= r_{\pi(u_1)} + \sum_{h=u_1}^{u_2} p_{\pi(h)1} + \dots + \sum_{h=u_m}^{u_{m+1}} p_{\pi(h)m} - d_{\pi(u_{m+1})} \triangleq \Delta_{\pi(u_1) \pi(u_{m+1})} \end{aligned} \quad (6.61)$$

Z wzorów (6.59) - (6.60) wynika, że $h_{\max}(\beta) \geq h_{\max}(\pi)$ co przeczy przyjętemu założeniu i kończy dowód twierdzenia. ■

Z twierdzenia 6.2 wynika wniosek, że jeśli permutacja β została otrzymana z permutacji π przez zmianę kolejności wykonywa-

nia zadań w π oraz $h_{\max}(\beta) < h_{\max}(\pi)$, to w β przynajmniej jedno zadanie dla przynajmniej jednego bloku z π poprzedza pierwsze zadanie /lub występuje za ostatnim zadaniem/ tego bloku. Wniosek ten jest w pełni analogiczny do wniosku z twierdzenia 5.2. Tak więc schemat działania algorytmu będzie identyczny z opisanymi w rozdziale 5.2.

W przypadku problemu $n|m|P, < | h_{\max}$ prawdziwe są również wnioski analogiczne do wniosków 3.1, 3.2, 3.3, 3.4 i tw. 3.3 z pracy [43] .

6.2. Dolne ograniczenia

Należy wyznaczyć dolne ograniczenie wartości funkcji celu problemu $n|m|P, \mathcal{R}_\pi | h_{\max}$, gdzie $\mathcal{R}_\pi$ jest relacją częściowego porządku w zbiorze zadań ustaloną w węźle π .

Zauważmy, że ze wzoru (6.19) wynika, że optymalna wartość funkcji celu problemu $n|m|P, < | f_{\max}$, $f_{\max} \triangleq \max_{1 \leq i \leq n} f_i(T_i)$ /lub jej dolne ograniczenie/ jest dolnym ograniczeniem kosztu maksymalnego w problemie $n|m|P, < | h_{\max}$. Do wyznaczenia dolnego ograniczenia problemu $n|m|P, < | f_{\max}$ możemy użyć dowolnej z metod opisanych w rozdziałach 5.3, 5.4.

Własność 6.2.

Niech LB_T oznacza dolne ograniczenie wartości funkcji celu problemu $n|m|P, r_i \geq 0, < | T_{\max}$. Niech $g(t) = \min_{1 \leq i \leq n} g_i(t)$, $f_*(t) = \min_{1 \leq i \leq n} f_i(t)$. Wówczas wartość $g_*(E^*)$ wyznaczona ze wzorów (6.62) - (6.64)

$$g_*(E^*) = f_*(T^*), \quad (6.62)$$

$$E^* + T^* = LB_T, \quad (6.63)$$

$$E^*, T^* \geq 0 \quad (6.64)$$

jest dolnym ograniczeniem kosztu maksymalnego w problemie $n|m|P, < | h_{\max}$. ■

Dowód

Problem $n|m|P, < | h'_{\max}$, $h'_{\max} \triangleq [\max_{1 \leq i \leq n} g_*(E_i), \max_{1 \leq i \leq n} f_*(T_i)]$ jest relaksacją problemu $n|m|P, < | h_{\max}$. Dodatkowo w problemie $n|m|P, < | h'_{\max}$ zachodzi $\Delta_{\max, \pi} = T_{\max}(\pi) \geq LB_T$ [2] . Stąd po

uwzględnieniu postaci problemu (6.62) - (6.64) wynika podana własność ■ . W celu wyznaczenia LB_T można zastosować dowolną z metod opisanych w rozdziałach 5.3, 5.4.

Wartość dolnego ograniczenia wyznaczona z własności 6.2 może zostać zwiększona przez zastosowanie twierdzeń analogicznych jak twierdzenia 5.5 i 5.6. Takie postępowanie może być szczególnie korzystne, gdy obie podane powyżej relaksacje okażą się zbyt silne.

6.3. Algorytm

Zasada działania algorytmu oparta jest na schemacie identycznym jak opisano w rozdziale 5.2. Istotne jednak różnice między podanym algorytmem /rozdz. 5.6/, a algorytmem formułowanym w tym rozdziale polegają na odmiennym i bardziej złożonym sposobie wyznaczania: $h_{\max}(\pi)$, momentów rozpoczęcia wykonywania zadań, ciągu krytycznego, dolnych ograniczeń oraz lokalnych oszacowań.

Niech dalej $\Delta_{\pi(j)}^{ka}$, $\Delta_{\pi(j)}^{kb}$, E_{π}^{ka} , E_{π}^{kb} będą wielkościami zdefiniowanymi, jak w rozdziale 5.5. Niech

$$\begin{aligned} h_{\max}(\pi) &= \max \left[\max_{1 \leq i \leq j \leq n} g_{\pi(i)}(e_{ij}^*), f_{\max}(\pi) \right] = \\ &= \max \left[g_{\pi(i^*)}(e_{i^*j^*}^*), f_{\max}(\pi) \right]. \end{aligned} \quad (6.65)$$

Rozważmy problem (6.16) - (6.18) w następującej postaci

$$g_{\pi(i^*)}(e_{i^*j^*}^a) = f_{\pi(j^*)}(t_{i^*j^*}^a). \quad (6.66)$$

$$e_{i^*j^*}^a + t_{i^*j^*}^a = \Delta_{\pi(i^*)\pi(j^*)}^{ka} \Delta_{\pi}^{ka}(j), \quad J_j \in E_{\pi}^{ka} \quad (6.67)$$

$$e_{i^*j^*}^a, t_{i^*j^*}^a \geq 0 \quad (6.68)$$

oraz

$$g_{\pi(i^*)}(e_{i^*j^*}^b) = f_{\pi(j^*)}(t_{i^*j^*}^b), \quad (6.66)$$

$$e_{i^*j^*}^b + t_{i^*j^*}^b = \Delta_{\pi(i^*)\pi(j^*)}^{kb} \Delta_{\pi}^{kb}(j), \quad J_j \in E_{\pi}^{kb} \quad (6.67)$$

$$e_{i^*j^*}^b, t_{i^*j^*}^b \geq 0. \quad (6.68)$$

Twierdzenie 6.3.

Niech $\pi \in \Pi$ będzie dowolną permutacją zaś $\rho \in \Pi$ permutacją otrzymaną z π przez przesunięcie zadania J_i na pozycję u_k /lub

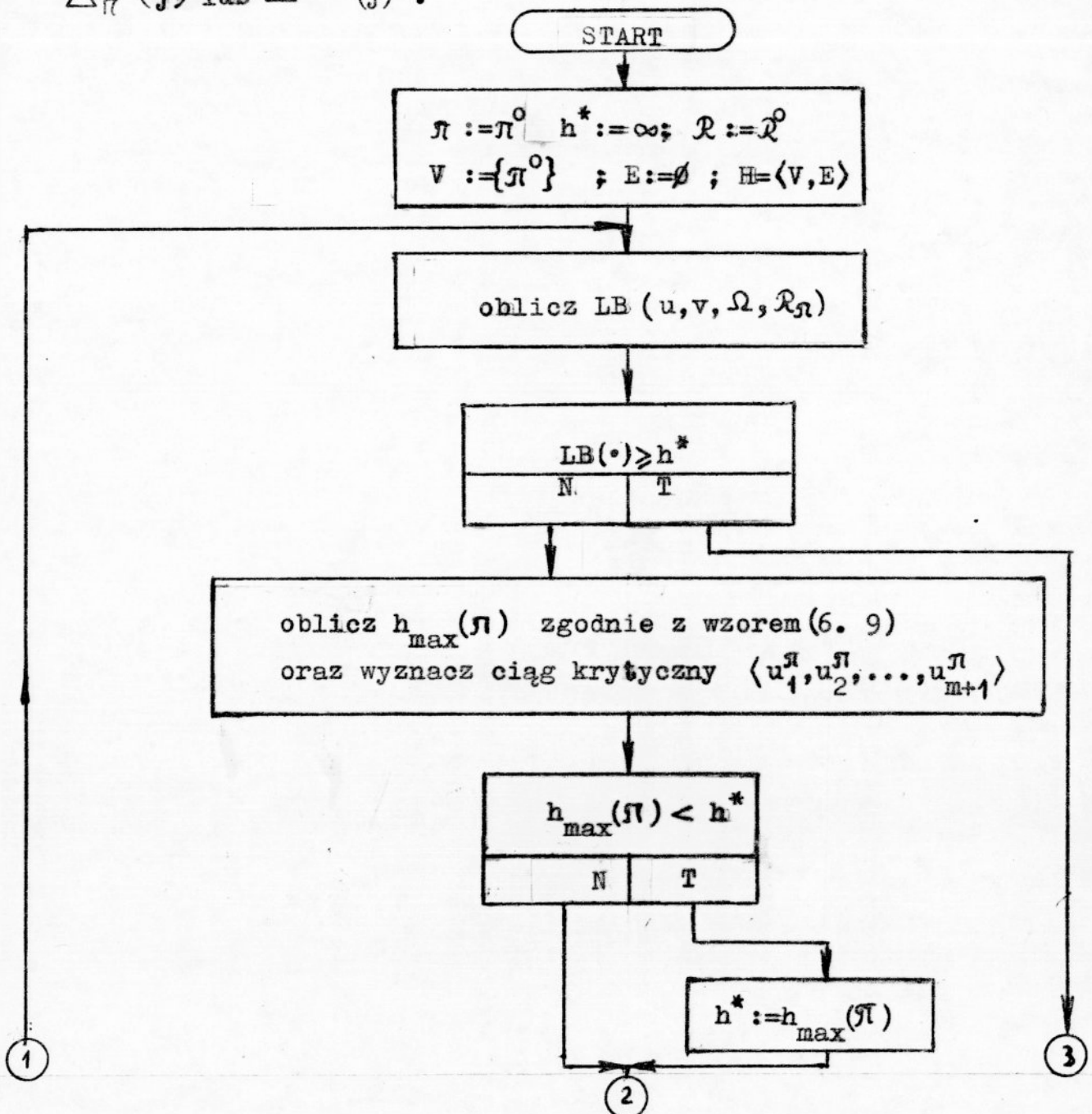
u_{k+1} / w π . Niech $g_i(t)$, $f_i(t)$, $i = 1, 2, \dots, n$ będą funkcjami spełniającymi założenia twierdzenia 6.1. Wtedy dla $k = 1, 2, \dots, m+1$ otrzymujemy

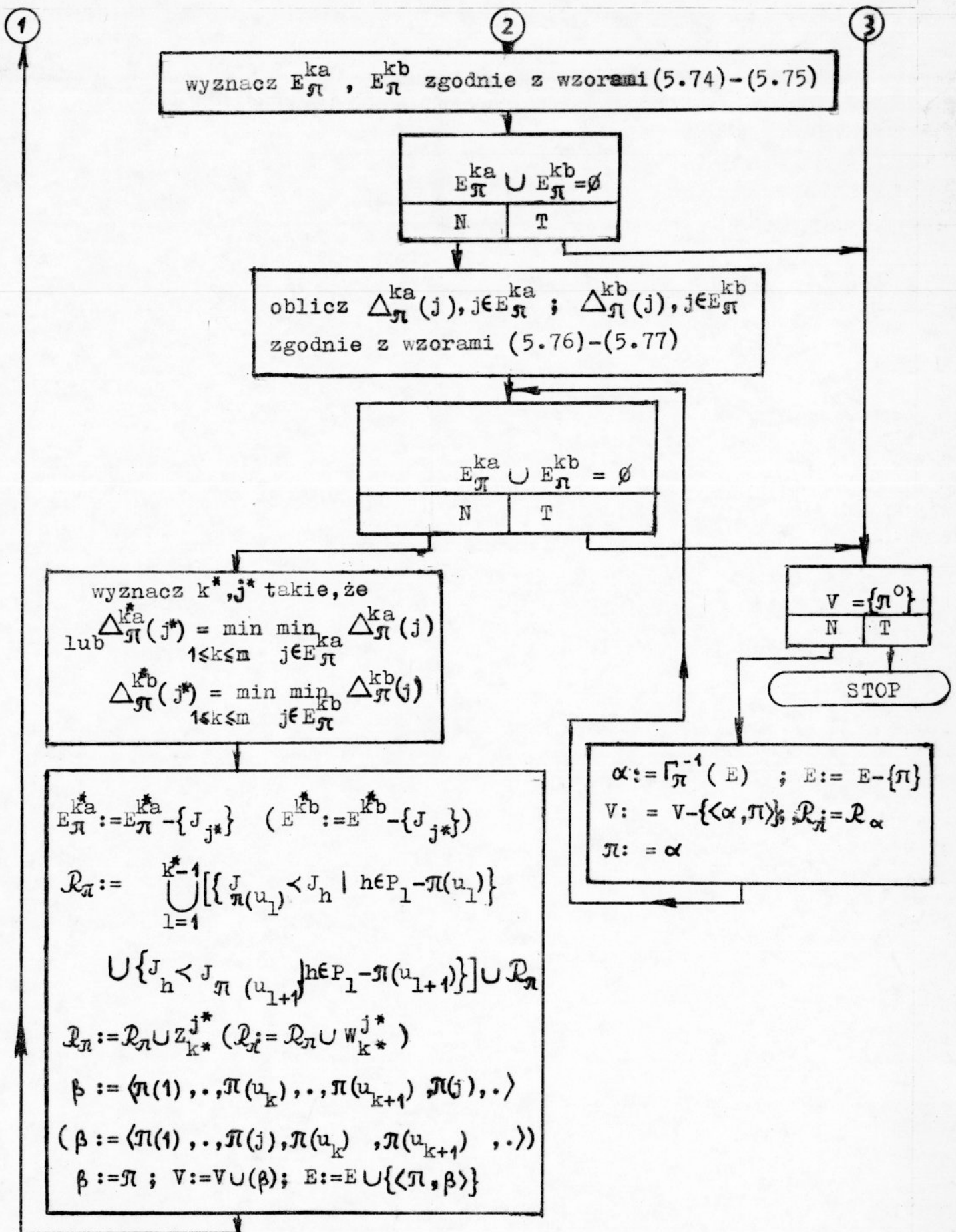
$$(i) \quad h_{\max}(\beta) \geq \max [g_{\pi(j)}(e_{i^*}^a j^*), f_{\pi(u_{m+1})}(C_{\pi(u_{m+1})} + \Delta_{\pi}^{ka}(j))] \\ J_j \in E_{\pi}^{ka} \quad (6.69)$$

oraz

$$(ii) \quad h_{\max}(\beta) \geq \max [g_{\pi(j)}(e_{i^*}^b j^*), f_{\pi(u_{m+1})}(C_{\pi(u_{m+1})} + \Delta_{\pi}^{kb}(j))] \\ J_j \in E_{\pi}^{kb} \quad (6.70)$$

Dowód jest analogiczny jak w przypadku twierdzenia 5.3. Do przesunięcia będziemy wybierać zadanie J_j o najmniejszej wartości $\Delta_{\pi}^{ka}(j)$ lub $\Delta_{\pi}^{kb}(j)$.



Rys. 6.6. Schemat blokowy algorytmu problemu $n|m|P, < h_{\max}$.

7. MINIMALIZACJA KOSZTU MAKSYMALNEGO. UOGÓLNIENIA.

Wyniki otrzymane w rozdziale 5 i 6 stały się podstawą daleko idących uogólnień. Ich podstawą było spostrzeżenie, że poprzez odpowiednie zdefiniowanie pojęcia ciągu krytycznego w problemach szeregowania, można udowodnić mocne twierdzenia eliminacyjne analogiczne do twierdzeń 5.3 i 6.3. W konsekwencji w przypadku szeregu ogólnych zagadnień z funkcją celu $f_{\max}$ i $h_{\max}$ /niepermutacyjne zagadnienie kolejnościowe taśmowe (F), problem gniazdowy (G), problem gniazdowy z równoległymi maszynami (J) i inne/ udało się autorowi zdefiniować pojęcie ciągu krytycznego oraz udowodnić odpowiednie twierdzenia eliminacyjne. Jednakże ze względu na obszerność dokonanych uogólnień zostaną zaprezentowane tylko podstawowe ich elementy.

Zagadnienie $n|m|F, r_i \geq 0, < f_{\max}$. Niech $\bar{\pi} = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$ $k = 1, 2, \dots, m$ jest permutacją /kolejnością/ wykonywania zadań na maszynie M_k . Należy znaleźć wektor $\bar{\pi}^*$ taki, że

$$f_{\max}(\bar{\pi}^*) = \min_{\bar{\pi}} f_{\max}(\bar{\pi}), \quad (7.1)$$

gdzie

$$f_{\max}(\bar{\pi}) = \max_{1 \leq j \leq n} f_{\pi_m(j)}(C_{\pi_m(j)m}). \quad (7.2)$$

Istotną różnicą między aktualnie rozpatrywanym zagadnieniem oraz zagadnieniem z rozdziału 5 jest przyjęcie, że kolejność wykonywania zadań na różnych maszynach może być dowolna. Liczba elementów w zbiorze rozwiązań jest w tym przypadku równa $(n!)^m$.

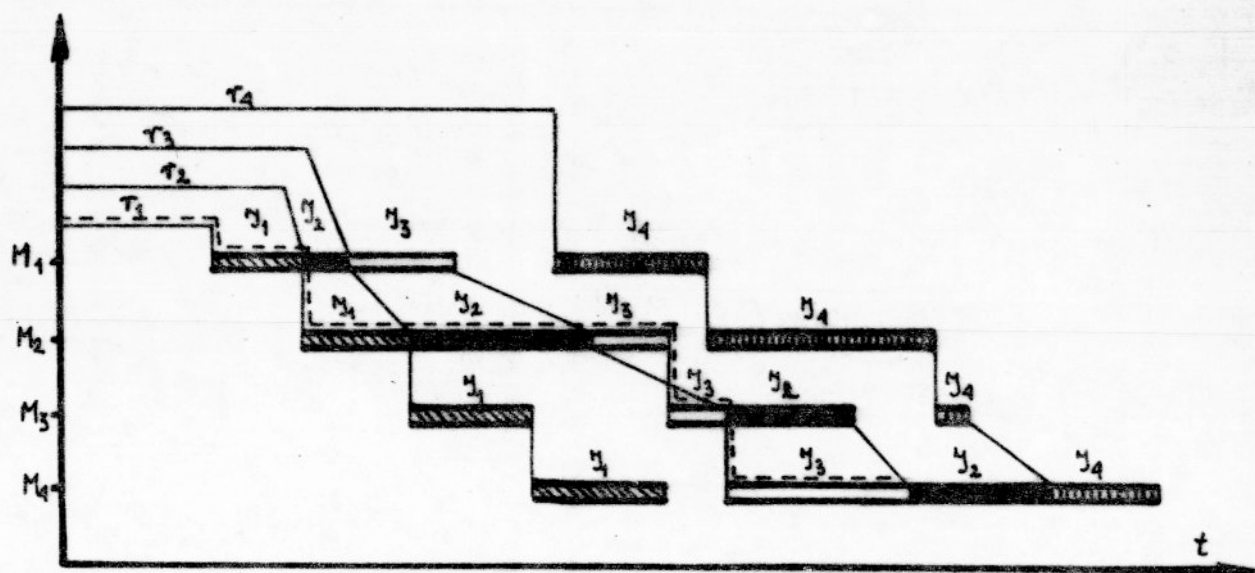
Niech dalej l_k będzie liczbą całkowitą taką, że

$$\pi_{k-1}(i) = \pi_k(i) + l_k \quad 1-n \leq l_k \leq n-1 \quad j = 1, 2, \dots, n \\ k = 2, 3, \dots, m. \quad (7.3)$$

Wówczas zachodzi /rys. 7.1/

$$f_{\max}(\pi) = \max_{1 \leq i_1, i_2, \dots, i_{m+1} \leq n} f_{\pi_m(i_{m+1})}(r_{\pi_1(i_1)} + \sum_{i=i_1}^{i_2} p_{\pi_1(i)} + \\ + \sum_{i=i_2+l_2}^{i_3} p_{\pi_2(i)} + \sum_{i=i_m+l_m}^{i_{m+1}} p_{\pi_m(i)}). \quad (7.4)$$

Zagadnienie kolejnościowe taśmowe (F) zostało sformułowane w rozdziale 5.



Rys. 7.1. Wykres Gantta problemu $n|m|F$, $r_i \geq 0$, $\leq |f_{\max}$
Zaznaczono ciąg krytyczny.

Ciąg wskaźników $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ spełniający warunek $1 \leq u_k^\pi \leq n$, $k = 1, 2, \dots, m+1$ i maksymalizujący prawą stronę wyrażenia (7.4) będziemy nazywać ciągiem krytycznym. Odpowiednio, ciągi zadań $\langle J_{\pi_k(u_k+1_k)}, J_{\pi_k(u_k+1_k+1)}, \dots, J_{\pi_k(u_{k+1})} \rangle$ będziemy nazywać blokami, $k = 1, 2, \dots, m$. Dla przykładu przedstawionego na rys. 7.1 mamy $f_i(C_i) = w_i C_i$, $i = 1, 2, 3, 4$, $w_1 = w_3 = w_4 = 1$, $w_2 = 10$ oraz $u_1^\pi = 1$, $u_2^\pi = 1$, $l_2^\pi = 0$, $u_3^\pi = 3$, $l_3^\pi = -1$, $u_4^\pi = 3$, $l_4^\pi = 0$, $u_5^\pi = 2$. Ciągi zadań $\langle J_1 \rangle$, $\langle J_1, J_2, J_3 \rangle$, $\langle J_4 \rangle$, $\langle J_5 \rangle$ są odpowiednio blokami na maszynie M_1 , M_2 , M_3 , M_4 .

Zagadnienie $n|m|F, \leq |h_{\max}$. Niech $\bar{\pi}$, $\pi_k/k = 1, 2, \dots, m$, $l_k/k = 2, 3, \dots, m$ będą wielkościami zdefiniowanymi jak wyżej. Problem $n|m|F, \leq |h_{\max}$ polega na wyznaczeniu wektora permutacji $\bar{\pi}^*$ oraz momentów rozpoczęcia zadań $S_{ik}/i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$ takie, że

$$h_{\max}(\bar{\pi}^*) = \min h_{\max}(\bar{\pi}) \quad , \quad (7.5)$$

gdzie

$$h_{\max}(\bar{\pi}) = \min_{S_{ik} \in \mathcal{S}} \max \left[\max_{1 \leq i \leq n} g_i(E_i), \max_{1 \leq i \leq n} f_i(T_i) \right] \quad , \quad (7.6)$$

zaś zbiór $\mathcal{S}_{\bar{\pi}}$ wyznaczony jest przez układ nierówności

$$S_{\pi_k(i-1)_k} + p_{\pi_k(i-1)_k} \leq S_{\pi_k(i)_k} \quad \begin{matrix} i = 1, 2, \dots, n, \\ k = 1, 2, \dots, m, \end{matrix} \quad (7.7)$$

$$S \pi_k(0)1 + p \pi_k(0)1 = t_0 \quad k = 1, 2, \dots, m, \quad (7.8)$$

$$S \pi_{k-1}(i)_{k-1} + p \pi_{k-1}(i)_{k-1} \leq S \pi_k(i+l_k)_k \quad (7.9)$$

$$i=1, 2, \dots, n$$

$$k=1, 2, \dots, m.$$

Podobnie jak poprzednio, problem ten różni się od problemu opisanego w rozdziale 6 przyjęciem, że kolejność wykonywania zadań na każdej z maszyn może być dowolna.

Zdefiniujmy dalej pewne pojęcia analogiczne do podanych w rozdziale 6. Niech

$$\Delta \pi_1(i) \pi_m(j) = \max_{i \leq i_1, i_2, \dots, i_{m-1} \leq j} (r \pi_1(i_1) + \sum_{h=1}^{i_1} p \pi_1(h)_1 +$$

$$+ \sum_{h=i_1+1}^{i_2} p \pi_2(h)_2 + \dots + \sum_{h=i_{m-1}+1}^j p \pi_m(h)_m - d \pi_m(j)) \quad (7.10)$$

$$1 \leq i \leq j \leq n.$$

Rozważmy ciąg problemów w postaci: znaleźć e_{ij}^* , t_{ij}^* , $1 \leq i \leq j \leq n$ z warunków (7.11) - (7.13)

$$g \pi_1(i)(e_{ij}^*) = f \pi_m(j)(t_{ij}^*) \quad (7.11)$$

$$e_{ij}^* + t_{ij}^* = \Delta \pi_1(i) \pi_m(j) \quad (7.12)$$

$$e_{ij}^*, t_{ij}^* \geq 0. \quad (7.13)$$

Wówczas zachodzi /rys. 7.2/

$$h_{\max}(\pi) = \max [\max_{1 \leq i \leq j \leq n} g \pi_1(i)(e_{ij}^*), f_{\max}(\pi)]$$

$$= \max [g \pi_1(i^*)(e_{i^*j^*}^*), f_{\max}(\pi)] \quad (7.14)$$

Ciąg wskaźników $\langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle$ będziemy nazywać ciągami krytycznym jeśli

$$(i) \quad h_{\max}(\pi) = f_{\max}(\pi) \text{ oraz } \langle u_1^\pi, u_2^\pi, \dots, u_{m+1}^\pi \rangle \text{ jest ciągiem}$$

$$\text{krytycznym problemu } n|m|F, < |f_{\max}, f_{\max} \hat{=} \max_{1 \leq i \leq n} f_i(T_i)$$

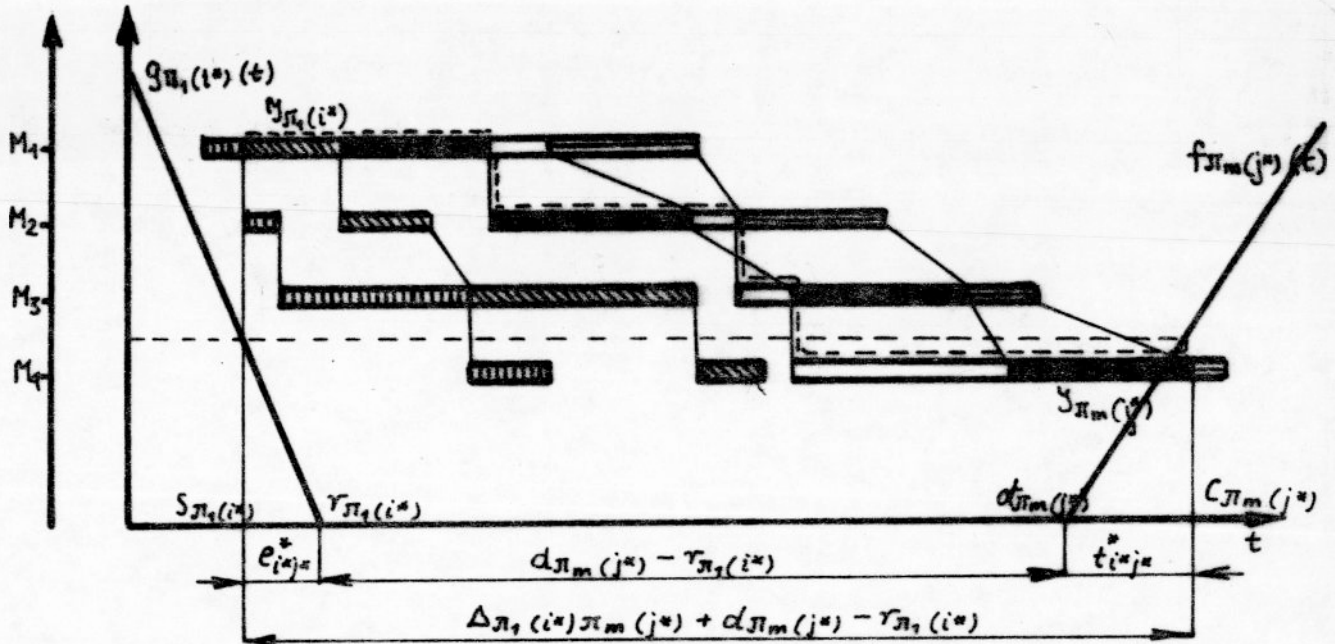
/wzór 7.4 dla $r_i \equiv 0$ $i = 1, 2, \dots, n$ / lub

$$(ii) \quad h_{\max}(\pi) = \max_{1 \leq i \leq j \leq n} g \pi_1(i)(e_{ij}^*) = g \pi_1(i^*)(e_{i^*j^*}^*),$$

$$u_1^\pi = i^*, u_{m+1}^\pi = j^*, 1 \leq i^* \leq j^* \leq n \text{ oraz } u_2^\pi = \hat{u}_1^\pi, u_3^\pi = \hat{u}_2^\pi, \dots,$$

$u_m^\pi = \hat{u}_{m-1}^\pi$, gdzie $\langle \hat{u}_1^\pi, \hat{u}_2^\pi, \dots, \hat{u}_{m-1}^\pi \rangle$, $1 \leq \hat{u}_k^\pi \leq n$,

$k = 1, 2, \dots, m-1$ jest ciągiem maksymalizującym prawą stronę wyrażenia (7.10).



Rys. 7.2. Wykres Gantta problemu $n|m|F, < h_{\max}$.

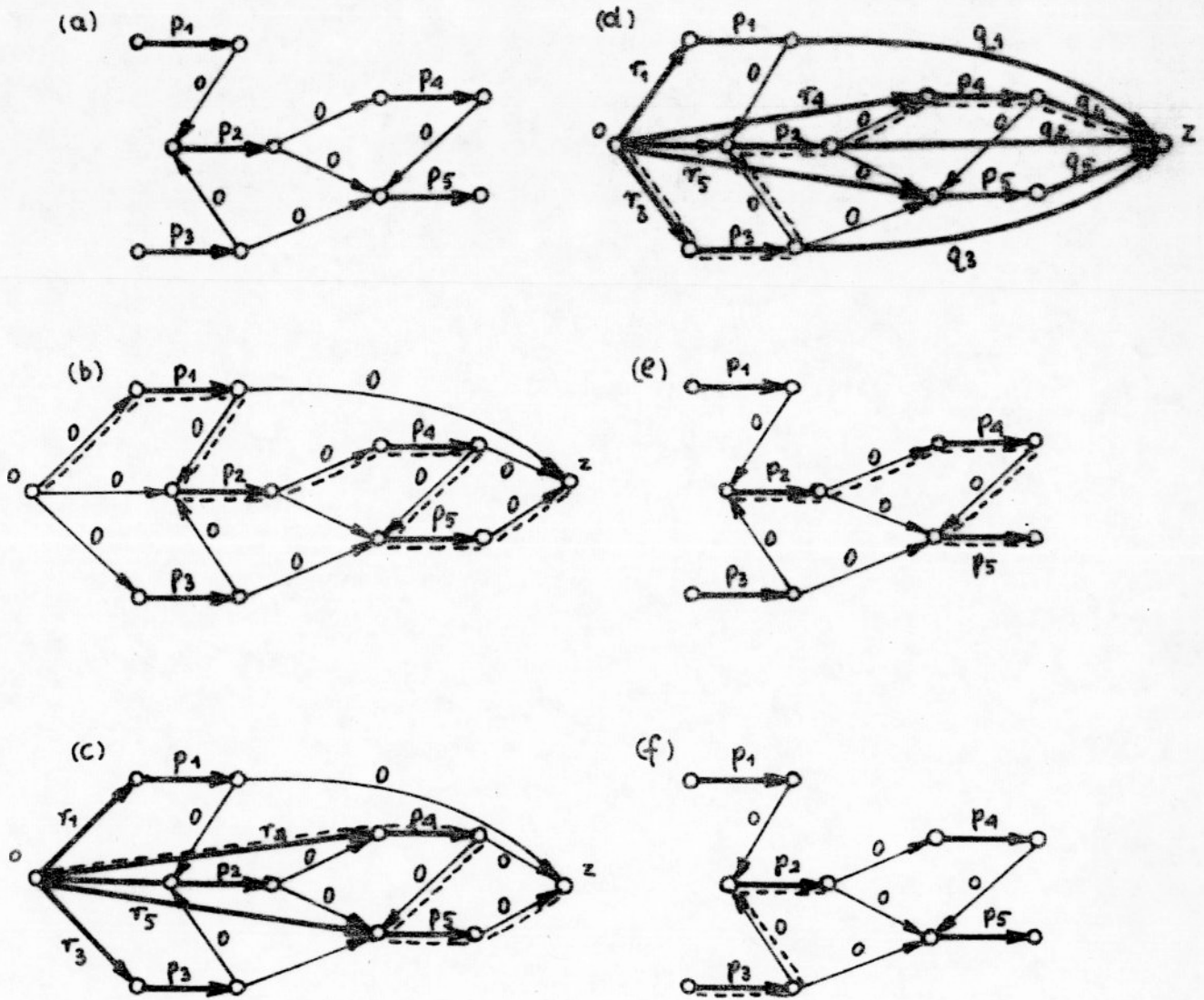
Zaznaczono funkcje kosztu maksymalnego oraz ciąg krytyczny.

Odpowiednio, ciągi zadań $\langle J_{\pi_k(u_k+1_k)}, \dots, J_{\pi_k(u_k+1)} \rangle$

będziemy nazywać blokami, $k = 1, 2, \dots, m$. Podobnie jak w poprzednich przypadkach bloki odpowiadają sekwencjom zadań wykonywanych na maszynie bez przerwy /rys. 7.2./.

Zagadnienie $n|m|J, r_j \geq 0, \tau \geq 1, m_w^k = 1, RT|f_{\max}$ i $h_{\max}$

Rozpatrzmy zagadnienie gniazdowe (G), w którym każde zadanie składa się tylko z jednej operacji, tzn. $J_i = \langle o_{i1} \rangle$. Załóżmy dodatkowo, że w zbiorze zadań /operacji/ istnieje relacja poprzedzania oznaczona RT. Problem ten oznaczony symbolem $n|m|G, r_i \geq 0, n_i = 1, < f_{\max}$ odpowiada problemowi $n|m|J, r_j \geq 0, \tau \geq 1, m_w^k = 1, RT|f_{\max}$ w notacji Grabowskiego [43] lub problemowi $n|m|J, r_j \geq 0, \tau \geq 1, m_w^k \geq 1, RT|f_{\max}$ z ustalonym przydziałem operacji do maszyn. Załóżmy, że dane /ustalone/ są czasy trwania poszczególnych operacji oraz, że dana /ustalona/ jest kolejność realizacji operacji, którą można przedstawić w postaci grafu skierowanego acyklicznego rys. 7.3 a). Węzły grafu odpowiadają zdarzeniom rozpoczęcia i zakończenia operacji, łuki obciążone - operacjom, łuki nieobciążone - wymaganiom kolejnościowym.



Rys. 7.3. Pojęcie ciągu krytycznego w różnych problemach szeregowania;

(a) graf kolejności realizacji operacji /zaznaczono łuki obciążone/,

(b) rozszerzony graf kolejności realizacji operacji /o - "fikcyjna operacja początkowa", z - "fikcyjna operacja końcowa"/, zaznaczono ciąg krytyczny (b) - (f):

(b) problem $n|m|G$, $n_1=1$, $< |C_{\max}$,

(c) problem $n|m|G$, $n_1=1$, $r_1 \geq 0$, $< |C_{\max}$,

(d) problem $n|m|G$, $n_1=1$, $r_1 \geq 0$, $< |L_{\max}$,

(e) problem $n|m|G$, $n_1=1$, $r_1 \geq 0$, $< |f_{\max}$,

(f) problem $n|m|G$, $n_1=1$, $< |h_{\max}$.

Zwykle tak skonstruowany graf jest rozszerzany o dwa dodatkowe węzły odpowiadające operacjom pustym: "początkowej" - poprzedzającej realizację wszystkich operacji, i "końcowej" - następującej za wszystkimi operacjami, rys. 7.3 b). Tradycyjne podejście "ścieżki krytycznej" interpretuje wartość $C_{\max}$ w problemie szeregowania jako najdłuższą drogę między węzłami operacji "początkowej" i "końcowej", rys. 7.3 b) [40]. Wprowadzenie ograniczenia typu $r_i \geq 0$ /różne czasy dostępności operacji/ nie zmienia w niczym przyjętej interpretacji $C_{\max}$ dodając jedynie obciążenia określonym łukom, rys. 7.3 c) [43]. Podobne podejście zastosowano również do problemów szeregowania z funkcją celu typu $L_{\max}$. Dzięki umiejętnemu przekształceniu funkcji celu $L_{\max}$ do funkcji $C_{\max}$ z użyciem końcówki q_i [43] pojęcie drogi krytycznej zachowało dotychczasowe znaczenie przy jednoczesnym dodaniu obciążenia nowym łukom, rys. 7.3 d). Jednakże w przypadku problemów z funkcją celu typu $f_{\max}$ i $h_{\max}$ tradycyjne pojęcie drogi krytycznej okazało się mało użyteczne i zostało zastąpione nowym. Nowo wprowadzone pojęcie można interpretować jako pewną drogę w grafie /niekoniecznie krytyczną/ posiadającą określone własności. Ponieważ w ogólnym przypadku nie istnieje bezpośredni związek między długością tej drogi a wartością funkcji celu /w problemach z funkcją celu $C_{\max}$ związek taki istnieje/ nowo wprowadzone pojęcie nazwano dla odróżnienia ciągiem krytycznym. Poniżej podano opisowe definicje ciągu krytycznego dla dwóch problemów szeregowania.

A. Zagadnienie $n|m|G$, $n_i = 1$, $r_i \geq 0$, $< |f_{\max}$

Ciąg krytyczny jest to droga w grafie kolejności realizacji operacji /rys. 7.3 e/ o następujących własnościach:

- (i) pierwsza operacja należąca do drogi rozpoczyna się w swoim momencie dostępności, tzn. $S_1 = r_1$,
- (ii) operacje leżące na drodze są wykonywane bez przerw czasowych /zwłok/ pomiędzy nimi,
- (iii) moment zakończenia ostatniej operacji leżącej na drodze wyznacza wartość kosztu maksymalnego $f_{\max}$,
- (iv) wyznaczona wartość $f_{\max}$ jest największa spośród kosztów ostatnich operacji leżącej na dowolnej drodze w grafie, spełniającej warunki (i) - (iii).

B. Zagadnienie $n|m|G$, $n_1 = 1$, $< |h_{\max}$.

Ciąg krytyczny jest to droga w grafie kolejności realizacji operacji /rys. 7.3 f/ o następujących własnościach.

- (i) operacje leżące na drodze są wykonywane bez przerw czasowych pomiędzy nimi,
- (ii) wartość kosztu związana z przyspieszeniem realizacji pierwszej operacji należącej do drogi jest równa wartości kosztu związanej ze spóźnieniem realizacji ostatniej operacji należącej do drogi,
- (iii) wyznaczona wartość kosztu (ii) jest największa spośród kosztów wszystkich dróg w grafie, spełniających warunki (i) - (ii) .

Na zakończenie warto podkreślić, że w wszystkich podanych powyżej przypadkach istnieje odpowiednie pojęcie bloku oraz prawdziwe są odpowiednie własności i twierdzenia eliminacyjne związane z ciągiem krytycznym. Zdaniem autora, kolejnym pożytecznym rozszerzeniem byłoby pojęcie i własności ciągu krytycznego w problemach szeregowania z ograniczeniami zasobowymi.

* * *

8. MINIMALIZACJA KOSZTU CAŁKOWITEGO. ZAGADNIENIE $n|1|<|\sum h_i$.

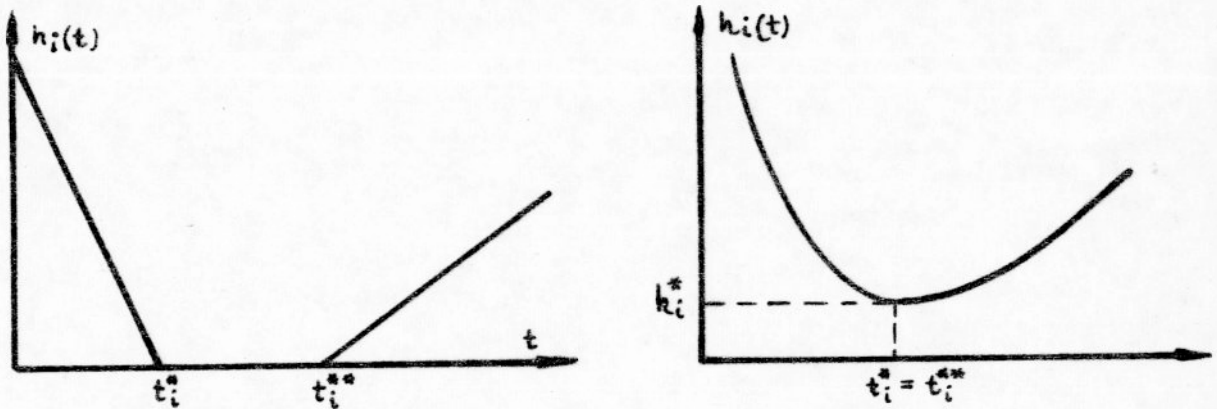
8.1. Sformułowanie problemu

Dany jest zbiór zadań $J = \{J_1, \dots, J_n\}$ przeznaczonych do wykonania na jednej maszynie. Dla każdego zadania określone są następujące wielkości:

p_i - czas realizacji zadania $J_i \in J$,

$h_i(t)$ - funkcja kosztu realizacji zadania $J_i \in J$.

O funkcjach $h_i(t)$, $J_i \in J$ zakłada się, że posiadają jedno minimum niekoniecznie właściwe, rys. 8.1.



Rys. 8.1. Przykład funkcji kosztu problemu $n|1|<|\sum h_i$

Dodatkowo określone są wielkości

t_0 - najwcześniejszy możliwy termin rozpoczęcia wszystkich zadań /zwykle $t_0 = 0$ /,

$\mathcal{R}$ - relacja częściowego porządku w zbiorze zadań, tzn.

$\langle J_i, J_j \rangle \in \mathcal{R}$ implikuje że zadania J_i -te musi być zakończone przed rozpoczęciem zadania J_j -tego; graf relacji $\mathcal{R}$ jest acykliczny.

Problem szeregowania polega na znalezieniu takich momentów rozpoczęcia wykonywania wszystkich zadań ze zbioru J , które minimalizują wskaźnik jakości w postaci

$$\sum h_i = \sum_{i=1}^n h_i(s_i), \quad (8.1)$$

przy następujących założeniach:

- maszyna może realizować jednocześnie co najwyżej jedno zadanie w dowolnym momencie czasowym,
- wykonywanie zadań nie może być przerywane,
- kolejność wykonywania zadań nie narusza przyjętej relacji $\mathcal{Q}$.

Problem tego rodzaju jest oznaczany symbolem klasyfikacyjnym $n|1|<|\sum h_i$ i jest najogólniejszym problemem jednomaszynowym ze względu na przyjętą postać funkcji celu.

Zwykle w praktyce wygodniejsze jest posługiwanie się innymi funkcjami kosztu $g_i(E_i)$, $f_i(T_i)$ wyznaczonymi na podstawie $h_i(S_i)$ w następujący sposób. Niech $[t_i^*, t_i^{**}]$ będzie przedziałem, w którym funkcja $h_i(t)$ osiąga minimum równe $h_i^* = h_i(t_i^*)$. Przyjmujemy najwcześniejszy pożądaný termin rozpoczęcia zadania $r_i \triangleq t_i^*$, najpóźniejszy pożądaný termin zakończenia zadania $d_i \triangleq t_i^{**} + p_i$. Definiujemy przyspieszenie $E_i \triangleq \max[0, r_i - S_i]$, spóźnienie $T_i \triangleq \max[0, S_i + p_i - d_i]$ oraz funkcje

$$g_i(E_i) = h_i(t_i^* - S_i) - h_i^*, \quad 0 \leq S_i \leq t_i^*, \quad (8.2)$$

$$f_i(T_i) = h_i(S_i - t_i^{**}) - h_i^*, \quad t_i^{**} \leq S_i < \infty. \quad (8.3)$$

Tak określone funkcje są niemalejące względem swoich argumentów $g_i(0) = 0 = f_i(0)$ oraz

$$h_i(S_i) = g_i(E_i) + f_i(T_i) + h_i^*, \quad 0 \leq S_i < \infty, \quad (8.4)$$

zaś funkcja celu (8.1) przyjmuje postać

$$\begin{aligned} \sum h_i &= \sum_{i=1}^n h_i(S_i) = \sum_{i=1}^n [g_i(E_i) + f_i(T_i) + h_i^*] = \\ &= \sum_{i=1}^n [g_i(E_i) + f_i(T_i)] + \sum_{i=1}^n h_i^*. \end{aligned} \quad (8.5)$$

Ponieważ składnik stały we wzorze (8.5) nie ma wpływu na rezultat minimalizacji, więc wystarczy rozważać problem z funkcją celu

$$\sum h_i = \sum_{i=1}^n [g_i(E_i) + f_i(T_i)], \quad (8.6)$$

gdzie $g_i(E_i)$, $f_i(T_i)$ - niemalejące funkcje swoich argumentów oraz $g_i(0) = 0 = f_i(0)$ / $i = 1, 2, \dots, n$ /.

Przyjmując odpowiednie założenia o funkcjach $g_i(E_i)$, $f_i(T_i)$, $J_i \in J$ otrzymamy pewne szczególne problemy szeregowania zestawione w tabeli 8.1.

Tabela 8.1.

Założenia	typ problemu	złożoność	autor algorytmu	źródło
$g_i(E_i) \equiv 0$ $f_i(T_i) \equiv 0$	$n 1 < \cdot$	$O(n)$	problem trywialny	
$g_i(E_i) \equiv 0$ $f_i(T_i) = w_i T_i$ $d_i \equiv 0$	$n 1 < \sum w_i C_i$		Horn ⁽¹⁾ Sidney ⁽²⁾	[52] [87]
$g_i(E_i) \equiv 0$ $f_i(T_i) = T_i, \mathcal{R} = \emptyset$	$n 1 \sum T_i$	NP	Fisher	[31]
$g_i(E_i) \equiv 0$	$n 1 < \sum f_i(T_i)$	NP	Lenstra ⁽³⁾	[71]
$g_i(E_i) \equiv 0$ $f_i(T_i) = \begin{cases} 0 & T_i = \emptyset \\ \infty & T_i > 0 \end{cases}$	$n 1 < , C_i \leq d_i \cdot$		(4)	
$g_i(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases}$ $f_i(T_i) \equiv 0$	$n 1 r_i \geq 0, < \cdot$	$O(n)$	problem trywialny	
$g_i(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases}$ $f_i(T_i) = w_i T_i, d_i \equiv 0$	$n 1 r_i \geq 0, < \sum w_i C_i$	NP	-	
$g_i(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases}$	$n 1 r_i \geq 0, < \sum f_i(T_i)$	NP	- (5)	
$g_i(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases}$ $f_i(T_i) = \begin{cases} 0 & T_i = 0 \\ \infty & T_i > 0 \end{cases}$	$n 1 r_i \geq 0, C_i \leq d_i, < \cdot$	NP	- (4)	
-	$n 1 < \sum h_i$	NP	- (6)	

- (1) $\mathcal{R}$ jest drzewem,
- (2) $\mathcal{R}$ jest grafem szeregowo-równoległym,
- (3) patrz także [7], [8], [68], [65],
- (4) rozwiązanie polega na wyznaczeniu rozwiązania dopuszczalnego,
- (5) dla tego zagadnienia autor pracy udowodnił nowe kryteria eliminacji w rozdz. 8.4,
- (5),(6) dla tych problemów sformułowano algorytm rozwiązania w rozdziale 8.7.

Dotychczas w literaturze najwięcej uwagi poświęcono problemom typu $n|1||\sum w_i C_i$ oraz $n|1||\sum w_i T_i$. Pierwszy algorytm dla zagadnienia $n|1||\sum w_i C_i$ podał Smith [91] zaś pierwszy algorytm dla zagadnienia $n|1||\sum w_i T_i$ podał Shwimer [85]. Późniejsze prace koncentrowały się głównie na badaniu własności zagadnień, formułowaniu kryteriów eliminacji oraz wyznaczaniu nowych mocniejszych oszacowań dla metody podziału i ograniczeń [79], [28], [27], [31]. Dla pewnych szczególnych klas zagadnień z funkcją celu typu $\sum f_i$ znaleziono algorytmy efektywne [52], [68]. W przypadku zagadnień NP-zupełnych stosowane są zwykle: metoda programowania dynamicznego PD [34], [7], [51], metoda podziału i ograniczeń [27], [79], [94], ... lub metody heurystyczne [98], [38].

Analizując tabelę 8.1 nietrudno zauważyć, że dla szeregu problemów nie sformułowano dotąd żadnego algorytmu. Powodem takiego stanu rzeczy jest silna NP-zupełność problemów oraz duże trudności związane z konstrukcją algorytmów.

8.2. Modele i metody PD i PC do rozwiązywania problemu

$$n|1|K|\sum h_i$$

Baker i Schrage [7], [84], [8] oraz Lawler [65] przedstawili algorytm rozwiązania problemu $n|1|K|\sum f_i$, $f_i(t)$ - niemalejące, $J_i \in J$ w oparciu o metodę programowania dynamicznego (PD) o złożoności $O(n2^n)$. Zaproponowany algorytm wykorzystuje eliminacyjne własności relacji $\mathcal{R}$, pozwalające zredukować liczbę dopuszczalnych sekwencji częściowych badanych w każdej iteracji algorytmu, a tym samym skrócić czas jego działania. Autorzy położyli główny nacisk na konstrukcję różnych oryginalnych metod adresacji pamięci roboczej [7], [34], [65] pozwalających na maksymalne jej wykorzysta-

nie /jest to problem nietrywialny, bowiem wielkość pamięci roboczej jest funkcją wykładniczą rozmiaru problemu/.

W zastosowaniu do zagadnień szeregowania /na podstawie podanego przykładu/ metoda PD ujawnia następujące wady: (i) duża liczba iteracji /długi czas obliczeń/, (ii) wymagany duży obszar pamięci roboczej algorytmu, oraz następujące zalety (iii) możliwość zastosowania metody w przypadku dowolnych innych niemalejących, nieaddytywnych postaci funkcji celu, (iv) niewielka złożoność pojedynczej iteracji, (v) łatwość realizacji programowej, (vi) możliwość redukcji czasu obliczeń przez zastosowanie eliminacyjnych własności relacji $\mathcal{R}$, (vii) możliwość redukcji wielkości obszaru roboczej pamięci przez zastosowanie odpowiednich procedur adresacji pamięci.

Jednakże zastosowanie PD jako ogólnej metody rozwiązywania jednomaszynowych problemów szeregowania wydaje się być sprawą dyskusyjną.

Rozważmy początkowo problem $n | 1 | r_i \geq 0, < | \sum f_i, f_i(t) - \text{niemalejące}, J_i \in J / \text{np. } f_i(t) = w_i T_i + z_i C_i$. Niech $I \subset J = \{J_1, J_2, \dots, J_n\}$ będzie pewnym podzbiorem zadań J , zaś $I' = J - I$ jego dopełnieniem. Załóżmy w dalszym ciągu, że konstruujemy sekwencję zadań ze zbioru I przy warunku, że realizacja każdego zadania ze zbioru I' poprzedza realizację wszystkich zadań ze zbioru I . Niech $F(I, t)$ oznacza minimalną wartość funkcji celu dla wszystkich dopuszczalnych sekwencji zadań ze zbioru I przy warunku $S_i \geq t, J_i \in I$.

Oznaczmy, a_I - najwcześniejszy dopuszczalny moment rozpoczęcia wykonywania zadań ze zbioru I , b_I - najpóźniejszy moment rozpoczęcia wykonywania zadań ze zbioru I .

Optymalna wartość funkcji celu dla wszystkich dopuszczalnych sekwencji zadań jest równa wartości $F(J, 0)$ wyznaczonej z następującego wzoru rekurencyjnego

$$F(I, t) = \min_{J_j \in B(I)} \{ f_j(\max[r_j, t + p_j] + F(I - \{J_j\}, \max[r_j, t + p_j])) \} \\ a_I \leq t \leq b_I, \quad (8.7)$$

$$F(\emptyset, t) = 0 \quad a_\emptyset \leq t \leq b_\emptyset, \quad (8.8)$$

$$a_I = \min_{J_i \in I'} r_i + \sum_{J_i \in I'} p_i, \quad (8.9)$$

$$b_I = \max_{J_i \in I'} r_i + \sum_{J_i \in I'} p_i, \quad (8.10)$$

$$B(I) = \{J_i \in I \mid \forall J_j \in I \quad \langle J_j, J_i \rangle \notin \mathcal{R}\}. \quad (8.11)$$

Wartość $F(I, t)$ jest wyznaczana dla wszystkich podzbiorów zbioru J oraz wszystkich momentów czasu $a_I' \leq t \leq b_I'$. Wielkości a_I' , b_I' są odpowiednio dolnym i górnym ograniczeniem wielkości a_I , b_I . Obliczenie ich wartości wymaga $O(n)$ iteracji. Możliwe jest wyznaczenie wielkości a_I , b_I w sposób bardziej dokładny, poprzez zastosowanie reguły Jacksona [55] oraz odwrotnej reguły Jacksona dla zadań należących do zbioru I' . Rozwiązanie takie wymaga użycia algorytmu o złożoności $O(n \cdot \log n)$ dla każdego etapu metody PD i wydaje się w końcowej ocenie zbyt czasochłonnym.

Rozważmy teraz problem $n|1|<|\sum h_i|/np$. $h_i(S_i) = v_i E_i + w_i T_i + z_i C_i$. Zachowując dotychczasowe oznaczenie możemy napisać wzór rekurencyjny

$$F(I, t) = \min_{t \leq s \leq b_I} \min_{J_j \in B(I)} \{f_j(s + p_j) + F(I - \{J_j\}, s + p_j)\}, \quad a_I' \leq t \leq b_I', \quad (8.12)$$

$$F(\emptyset, t) = 0 \quad a_\emptyset' \leq t \leq b_\emptyset', \quad (8.13)$$

$$a_I' = \sum_{J_i \in I'} p_i \quad (8.14)$$

oraz b_I , $B(I)$ są zdefiniowane wzorami (8.10), (8.11).

Optymalna wartość funkcji celu problemu $n|1|<|\sum h_i|$ jest równa wartości $F(J, 0)$.

Jeśli przyjmujemy w dalszym ciągu, że momenty czasowe przyjmują wartości dyskretne, tzn. $t = t_0, t_0+1, t_0+2, \dots, T$ to możemy określić złożoność obliczeniową metody PD dla problemu $n|1|r_i \geq 0$, $<|\sum f_i$ (8.7) - (8.11) równą $O(n(T-t_0)2^n)$ oraz dla problemu $n|1|<|\sum h_i$ (8.12) - (8.14) równą $O(nT^2 2^n)$.

Uzyskanie zadowalających wyników obliczeniowych w podanych przypadkach jest praktycznie niemożliwe ze względu na zbyt małą moc obliczeniową najlepszych maszyn cyfrowych znanych obecnie. Przykładowo, gdybyśmy dysponowali maszyną cyfrową o szybkości procesora ok. 10^6 operacji/s i chwilowo nieograniczonej wielkości pamięci operacyjnej, to dla następującego przykładu $n = 10$, $r = \langle 267, 68, 335, 31, 367, 26, 21, 47, 68, 115 \rangle$, $p = \langle 10, 21, 30, 50, 79, 28, 7, 34, 40, 73 \rangle$, $\mathcal{R} = \emptyset$ problemu $n|1|<|\sum h_i$ otrzymujemy:

- (a) wymagany obszar pamięci roboczej około 740 000 słów,
- (b) przewidywana liczba iteracji: około $3.7 \cdot 10^8$,
- (c) przewidywany czas obliczeń: około 2.53 godzin.

Wielkości (a) - (c) rosną w sposób wykładniczy, ze wzrostem rozmiaru problemu.

W pracach dotyczących zagadnień optymalizacji w dyskretnych systemach produkcyjnych spotykane są podejścia oparte na programowaniu całkowitoliczbowym lub binarnym [59], [61], [57]. Przyjęcie takiego sposobu postępowania jest uzasadnione wówczas, gdy:

- (i) model problemu jest liniowy, bowiem metody PCL są dość dobrze rozwinięte [34], [57],
- (ii) liczba zmiennych w przyjętym modelu oraz liczba ograniczeń nie jest zbyt duża /dotyczy to szczególnie modeli binarnych/.

Zauważmy, że w przypadku problemu $n|1| \sum h_i$ nie jest możliwe spełnienie warunków (i) - (ii) ze względu na przyjętą z definicji nieliniowość funkcji celu $\sum h_i = \sum \{g_i(\max[0, r_i - S_i]) + f_i(\max[0, c_i - d_i])\}$ a także ze względu na silną zależność /w przypadku modelu binarnego/ liczby zmiennych od wartości liczbowych danych problemu ($T - t_0$).

Problemy, których modele nie spełniają warunków (i) - (ii) są zwykle rozwiązywane metodami opartymi na metodzie podziału i ograniczeń [34], [61], metodami przeglądu leksykograficznego [59] lub metodami przybliżonymi [38]. W konsekwencji przyjęcie modelu całkowitoliczbowego nieliniowego, a następnie zastosowanie metody podziału i ograniczeń do rozwiązania zagadnienia PCN daje w praktyce gorszy rezultat niż bezpośrednie zastosowanie metody podziału i ograniczeń do rozwiązania problemu. Zwykle w tym drugim przypadku możliwe jest uwzględnienie szeregu specyficznych własności problemu pozwalających na poprawę własności algorytmu. Dlatego też podejście tego rodzaju jest stosowane w prezentowanej pracy.

8.3. Rozwiązanie optymalne i bioptymalne

Wprowadzone zostanie parę nowych pojęć. Niech $\pi = \langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ będzie permutacją wskaźników zadań ze zbioru J ; $\pi(i)$ jest numerem zadania wykonywanego jako i -te w kolejności. Permutacja π jest dopuszczalna /dla danej relacji $\mathcal{R}$ /, jeśli dla każdej pary zadań $\langle \pi(i), \pi(j) \rangle \in \mathcal{R}$ zachodzi $i < j$, $j = 1, 2, \dots, n$, $i \neq j$. Zbiór permutacji dopuszczalnych oznaczono $\Pi(\mathcal{R})$. Dla każdej

permutacji $\pi \in \Pi(\mathcal{R})$ zdefiniowano zbiór dopuszczalnych momentów rozpoczęcia zadań

$$S_{\pi} = \{ \bar{S} = \langle s_1, s_2, \dots, s_n \rangle \mid t_0 \leq s_{\pi(1)}, s_{\pi(1)} + p_{\pi(1)} \leq s_{\pi(2)}, \dots, s_{\pi(n-1)} + p_{\pi(n-1)} \leq s_{\pi(n)} \} \quad (8.15)$$

Para $\langle \pi, \bar{S} \rangle$, $\bar{S} \in S_{\pi}$, $\pi \in \Pi(\mathcal{R})$ określa rozwiązanie dopuszczalne. Odpowiednio wartość funkcji celu (8.6) dla rozwiązania $\langle \pi, \bar{S} \rangle$ oznaczmy $F(\bar{S}; \pi)$.

Rozwiązanie $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in S_{\pi}$, $\pi \in \Pi(\mathcal{R})$ jest optymalne /dla danej permutacji π /jeśli

$$F(\bar{S}^*; \pi) = \min_{\bar{S} \in S_{\pi}} F(\bar{S}; \pi) \quad (8.16)$$

Rozwiązanie $\langle \pi^*, \bar{S}^{**} \rangle$, $\bar{S}^{**} \in S_{\pi^*}$, $\pi^* \in \Pi(\mathcal{R})$ jest bioptymalne /globalnie optymalne/ jeśli

$$F(\bar{S}^{**}; \pi^*) = \min_{\pi \in \Pi(\mathcal{R})} F(\bar{S}^*; \pi) = \min_{\pi \in \Pi(\mathcal{R})} \min_{\bar{S} \in S_{\pi}} F(\bar{S}; \pi) \quad \bar{S}^* \in S_{\pi^*} \quad (8.17)$$

Problem szeregowania typu $n \mid 1 \mid \sum h_i$ polega na znalezieniu rozwiązania bioptymalnego $\langle \pi^*, \bar{S}^{**} \rangle$, $\bar{S}^{**} \in S_{\pi^*}$, $\pi^* \in \Pi(\mathcal{R})$ dla przyjętych funkcji kosztu $h_i(t)$, $J_i \in J$.

W ogólnym przypadku wyznaczenie rozwiązania optymalnego $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in S_{\pi}$, $\pi \in \Pi(\mathcal{R})$ wymaga rozwiązania problemu optymalizacji z nieliniową, nieróżniczkowalną funkcją celu i liniowymi ograniczeniami w postaci

$$\min_{\bar{S}} F(\bar{S}; \pi) = \min_{\bar{S}} \sum_{i=1}^n [g_i(E_i) + f_i(T_i)] \quad (8.18)$$

$$t_0 \leq s_{\pi(1)} \quad (8.19)$$

$$s_{\pi(i)} + p_{\pi(i)} \leq s_{\pi(i+1)} \quad i = 1, 2, \dots, n-1 \quad (8.20)$$

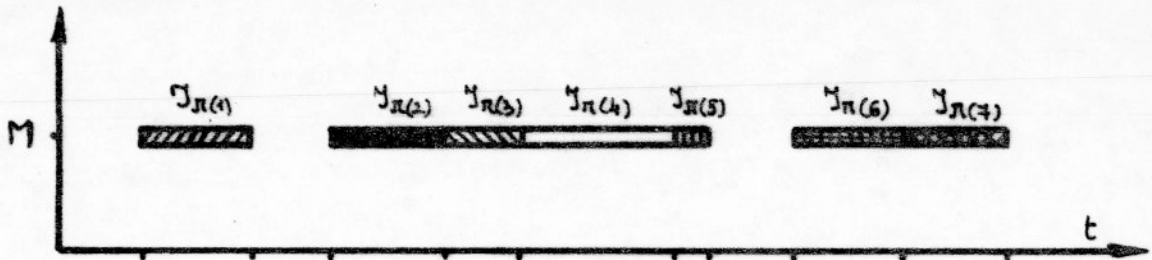
W przypadku, gdy funkcje $g_i(E_i)$, $f_i(T_i)$, $J_i \in J$ są wypukłe, to funkcją $F(\bar{S}; \pi)$ jest również wypukła.

Dalsze rozważania będą prowadzone pod kątem skonstruowania specjalizowanego, wielomianowego algorytmu dla rozwiązania problemu (8.18) - (8.20).

Permutację częściową $\beta(x, y) = \langle \pi(x), \pi(x+1), \dots, \pi(y) \rangle$, $x, y = 1, 2, 3, \dots, n$, $x \leq y$, będziemy nazywać blokiem w rozwiązaniu $\langle \pi, \bar{S}^* \rangle$ jeśli jest spełniony warunek

$$S_{\pi(i)}^* + p_{\pi(i)} = S_{\pi(i+1)}^* \quad x \leq i \leq y \quad (8.21)$$

Blok odpowiada sekwencji zadań wykonywanych na maszynie bez przerw pomiędzy zadaniami, rys. 8.2.



Rys. 8.2. Wykres Gantta dla problemu $n | 1 | < \sum h_i$

Blok $\beta(x, y)$ nazywamy maksymalnym w rozwiązaniu $\langle \pi, \bar{S} \rangle$ $\bar{S} \in \bar{S}_{\pi}$ jeśli $\beta(x, y+1)$ nie jest blokiem oraz $\beta(x-1, y)$ nie jest blokiem / $\beta(0, y)$ i $\beta(x, n+1)$ nie są blokami/.

Na rysunku 8.2 blokami maksymalnymi są odpowiednio $\langle \pi(1) \rangle$, $\langle \pi(2), \pi(3), \pi(4), \pi(5) \rangle$, $\langle \pi(6), \pi(7) \rangle$.

Niech, dalej $\langle i_0, i_1, \dots, i_l \rangle$ będzie ciągiem wszystkich wskaźników takich, że $i_j \in \{1, 2, \dots, n\}$, $j = 1, 2, \dots, l$, $l < n$, $0 = i_0 < i_1 < i_2 < \dots < i_l = n$ oraz ograniczenia $S_{\pi(i)}^* + p_{\pi(i)} \leq S_{\pi(i+1)}^*$ $i = i_1, i_2, \dots, i_{l-1}$ w problemie (8.18) - (8.20) są nieaktywne, tzn.

$$S_{\pi(i)}^* + p_{\pi(i)} < S_{\pi(i+1)}^* \quad i = i_1, i_2, \dots, i_{l-1} \quad (8.22)$$

Wówczas permutacje częściowe $\beta(i_j+1, i_{j+1})$, $j = 0, 1, 2, \dots, l-1$ są blokami maksymalnymi, zaś l - liczbą bloków w rozwiązaniu $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in \bar{S}_{\pi}$. Dodatkowo rozwiązanie optymalne problemu (8.18) - (8.20) jest sumą rozwiązań optymalnych problemów częściowych danych w postaci

$$F(\bar{S}^*; \pi) = \sum_{j=0}^{l-1} F(\bar{S}^*; \beta(i_j+1, i_{j+1})) \quad (8.23)$$

$$F(\bar{S}^*; \beta(i_j+1, i_{j+1})) \triangleq \min_{\bar{S}} \sum_{i=i_j+1}^{i_{j+1}} h_{\pi(i)}(S_{\pi(i)}) \quad (8.24)$$

przy ograniczeniach

$$t_0 \leq S_{\pi(i)} \quad i = i_j + 1, \quad (8.25)$$

$$S_{\pi(i)} + p_{\pi(i)} = S_{\pi(i+1)} \quad i = i_j + 1, i_j + 2, \dots, i_{j+1} - 1. \quad (8.26)$$

Własność tę wykorzystano w konstrukcji algorytmu opisanego w dalszej części rozdziału.

Rozwiązanie częściowe $\langle \beta(i_j + 1, i_{j+1}), \bar{S}^* \rangle$ wyznaczone z problemu (8.23) - (8.26) będziemy nazywać blokiem optymalnym.

Inaczej mówiąc blok optymalny jest to blok maksymalny w rozwiązaniu optymalnym. Jeśli rozwiązanie przedstawione na rys. 8.2 jest optymalne, to bloki $\langle \pi(1) \rangle, \langle \pi(2), \pi(3), \pi(4), \pi(5) \rangle, \langle \pi(6), \pi(7) \rangle$ są optymalne.

Rozpatrzmy teraz problem $n|1| < |\sum h_i|$ w przypadku, gdy $g_i(E_i) = 0$ dla $E_i = 0$ oraz $g_i(E_i) = \infty$ dla $E_i > 0$, $f_i(C_i)$ - funkcje niemalejące, skończone $J_i \in J$. Wówczas dla każdej $\pi \in \Pi(\mathcal{R})$ zachodzi $r_i \leq S_i^*$ /gdyby przyjąć, że $r_i > S_i^*$, to $F(\bar{S}^*; \pi) = \infty$; w przeciwnym przypadku, $F(\bar{S}^*; \pi) < \infty$ /. Ponieważ $f_i(C_i)$ - niemalejące więc $\langle \pi, \bar{S}^* \rangle$ można znaleźć z zasady maksymalnego wykorzystania maszyn. W tym przypadku momenty rozpoczęcia wykonywania zadań wyrażają się wzorem

$$S_{\pi(1)}^* = \max [t_0, r_{\pi(1)}] \quad (8.27)$$

$$S_{\pi(i+1)}^* = \max [r_{\pi(i+1)}, S_{\pi(i)}^* + p_{\pi(i)}] \quad i = 1, 2, \dots, n-1 \quad (8.28)$$

Optymalna wartość funkcji celu wynosi $F(\bar{S}^*; \pi) = \sum_{i=1}^n f_i(S_i^*)$, zaś ciąg $\langle i_0, i_1, \dots, i_l \rangle$ może być wyznaczony z warunków

$$S_{\pi(i)}^* + p_{\pi(i)} < S_{\pi(i+1)}^* \quad \text{dla } i = i_1, i_2, \dots, i_{l-1} \quad (8.30)$$

oraz $i_0 = 0, i_l = n$.

Zupełnie inaczej przedstawia się problem w przypadku ogólnej funkcji celu (8.6). Zasada maksymalnego wykorzystania maszyn nie jest w tym przypadku spełniona, a więc i znalezienie rozwiązania optymalnego $\langle \pi, \bar{S}^* \rangle$ jest zadaniem znacznie trudniejszym.

Rozważmy początkowo problem $n|1| < |\sum h_i|$ dla podstawowego przypadku, tzn. $h_i(S_i) = v_i E_i + w_i T_i + z_i C_i$ $v_i, w_i, z_i \geq 0$. Wprowadzenie współczynników stałych do funkcji kosztów nie zmienia w niczym rozważań i może być zastąpione przez dodanie tych wielkości do końcowej optymalnej wartości globalnego kosztu. Zdefiniujmy wielkości

$$D_i \triangleq S_i - r_i, \quad i = 1, 2, \dots, n, \quad (8.31)$$

$$L_i \triangleq S_i + p_i - d_i, \quad i = 1, 2, \dots, n, \quad (8.32)$$

Niech $\bar{S} \in \bar{S}_\pi$. Określmy zbiory: $J_E(\bar{S}) \triangleq \{J_i \in J \mid E_i > 0\}$ - zbiór zadań przyspieszonych, $J_D^0(\bar{S}) = \{J_i \in J \mid D_i = 0\}$ - zbiór zadań rozpoczynających się w terminie, $J_D(\bar{S}) = J - J_E(\bar{S}) - J_D^0(\bar{S})$ - zbiór zadań rozpoczynających się po terminie, $J_T(\bar{S}) = \{J_i \in J \mid T_i > 0\}$ - zbiór zadań spóźnionych, $J_L^0(\bar{S}) = \{J_i \in J \mid L_i = 0\}$ - zbiór zadań kończących się przed terminem.

Zdefiniujmy dalej

$$sv_{xy} = \sum_{i=x}^y \pi(i) \quad - \text{suma wag zadań z bloku } \beta(x,y) \text{ przyspieszonych lub rozpoczynających się w terminie w rozwiązaniu } \langle \pi, \bar{S}^* \rangle,$$

$$\overline{J_{\pi(i)}} \in \overline{J_D^0(\bar{S})} \cup \overline{J_E(\bar{S})}$$

$$sw_{xy} = \sum_{i=x}^y w_{\pi(i)} + \sum_{i=x}^y z_{\pi(i)} \quad - \text{suma wag zadań z bloku } \beta(x,y) \text{ spóźnionych w rozwiązaniu } \langle \pi, \bar{S}^* \rangle,$$

$$\overline{J_{\pi(i)}} \in \overline{J_T(\bar{S}^*)}$$

$$\bar{sv}_{xy} = \sum_{i=x}^y v_{\pi(i)} \quad - \text{suma wag zadań z bloku } \beta(x,y) \text{ przyspieszonych w rozwiązaniu } \langle \pi, \bar{S}^* \rangle,$$

$$\overline{J_{\pi(i)}} \in \overline{J_E(\bar{S}^*)}$$

$$\bar{sw}_{xy} = \sum_{i=x}^y w_{\pi(i)} + \sum_{i=x}^y z_{\pi(i)} \quad - \text{suma wag zadań z bloku } \beta(x,y) \text{ spóźnionych lub kończących się w terminie w rozwiązaniu } \langle \pi, \bar{S}^* \rangle,$$

$$\overline{J_{\pi(i)}} \in \overline{J_T(\bar{S}^*)} \cup \overline{J_L^0(\bar{S}^*)}$$

Twierdzenie 8.1.

Warunkiem koniecznym i dostatecznym na to, by blok maksymalny $\beta(x,y)$ był optymalny w rozwiązaniu $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in \bar{S}_\pi$, jest

$$(i) \quad sv_{xy} \geq sw_{xy} \quad \text{gdy} \quad t_0 < S_{\pi(x)}^*, \quad (8.33)$$

$$(ii) \quad \overline{sv}_{xy} \leq \overline{sw}_{xy} . \quad (8.34)$$

Dowód

→ Niech $\beta(x,y)$ będzie blokiem maksymalnym optymalnym w rozwiązaniu $\langle \pi, \bar{S}^* \rangle$ takim, że co najmniej jeden z warunków (i), (ii) nie jest spełniony. Niech nie będzie spełniony warunek (i), tzn, $\overline{sv}_{xy} < \overline{sw}_{xy}$ /rys. 8.3/. Wybierzmy wtedy wartość $\varepsilon > 0$ taką, że $\varepsilon < S_{\pi(x)}^* - S_{\pi(x-1)}^* - p_{\pi(x-1)}$ /albo $\varepsilon < S_{\pi(x)}^* - t_0$ /, $\varepsilon < D_{\pi(i)}^*, D_{\pi(i)}^* > 0$, $i = x, x+1, \dots, y$, $\varepsilon < T_{\pi(i)}^*, T_{\pi(i)}^* > 0$, $i = x, x+1, \dots, y$. Wartość taka istnieje, gdy $t_0 < S_{\pi(x)}^*$, bowiem z definicji bloku wynika, że $S_{\pi(x-1)}^* + p_{\pi(x-1)} < S_{\pi(x)}^*$. Wówczas zachodzi $J_E(\bar{S}^* - \varepsilon) = J_E(\bar{S}^*) \cup \cup J_D^0(\bar{S}^*)$, $J_T(\bar{S}^* - \varepsilon) = J_T(\bar{S}^*)$, $J_D(\bar{S}^* - \varepsilon) = J_D(\bar{S}^*)$ oraz

$$F(\bar{S}^* - \varepsilon; \beta(x,y)) - F(\bar{S}^*; \beta(x,y)) = \sum_{i=x}^y h_{(i)} (S_{(i)}^* - \varepsilon) - \sum_{i=x}^y h_{\pi(i)} (S_{\pi(i)}^*)$$

$$= \sum_{i=x}^y v_{\pi(i)} [E_{\pi(i)}^* + \varepsilon] + \sum_{i=x}^y w_{\pi(i)} [T_{\pi(i)}^* - \varepsilon] + \sum_{i=x}^y z_{\pi(i)} [C_{\pi(i)}^* - \varepsilon] -$$

$$\left[\begin{array}{l} J_{\pi(i)} \in J_E(\bar{S}^* - \varepsilon) \\ J_{\pi(i)} \in J_T(\bar{S}^* - \varepsilon) \end{array} \right]$$

$$- \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* - \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* - \sum_{i=x}^y z_{\pi(i)} C_{\pi(i)}^* = \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* +$$

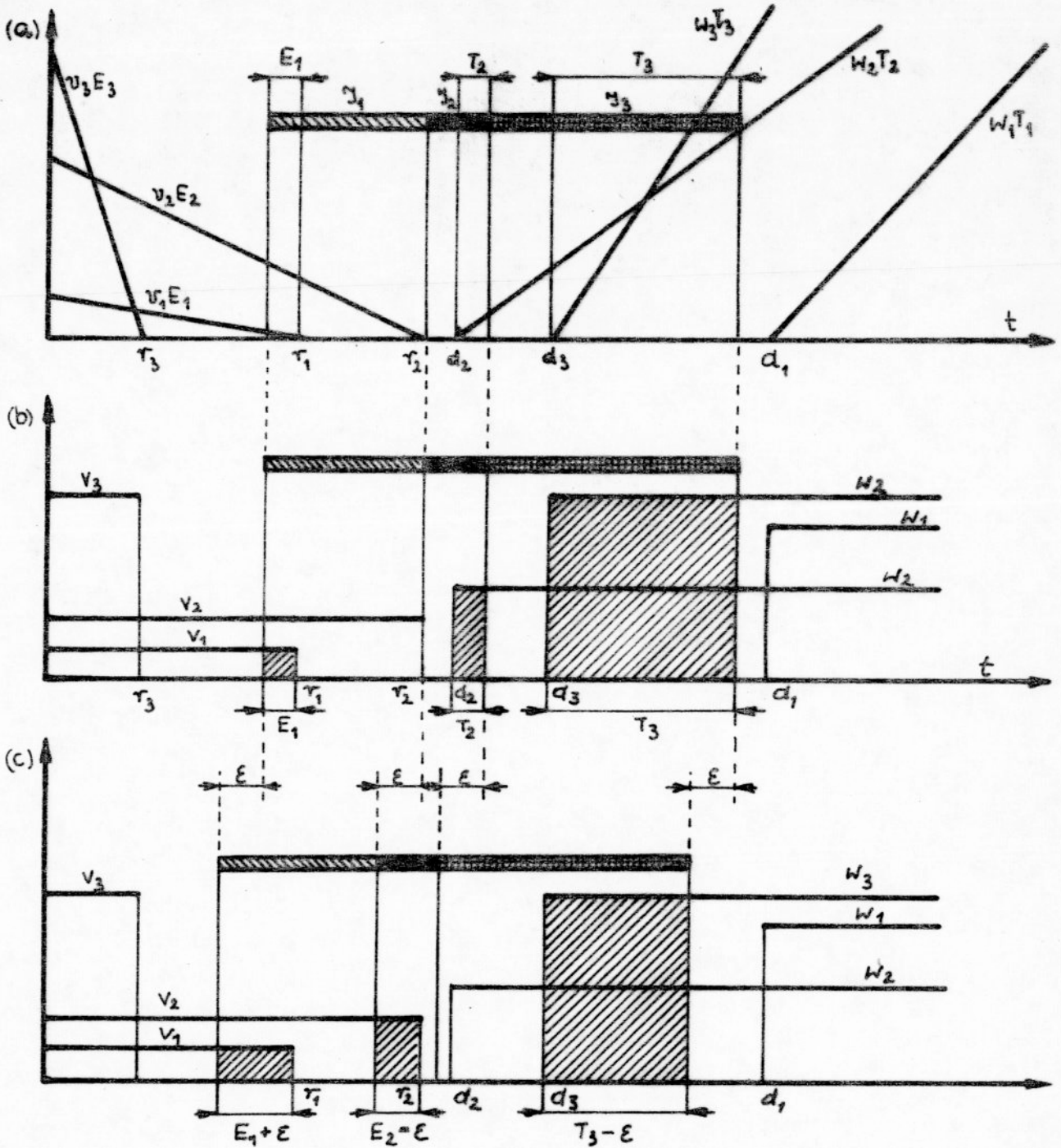
$$\left[\begin{array}{l} J_{\pi(i)} \in J_E(\bar{S}^*) \\ J_{\pi(i)} \in J_T(\bar{S}^*) \\ J_{\pi(i)} \in J_E(\bar{S}^*) \end{array} \right]$$

$$+ \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* + \sum_{i=x}^y v_{\pi(i)} \varepsilon + \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* - \sum_{i=x}^y w_{\pi(i)} \varepsilon -$$

$$\left[\begin{array}{l} J_{\pi(i)} \in J_D^0(\bar{S}^*) \\ J_{\pi(i)} \in J_E(\bar{S}^*) \cup J_D^0(\bar{S}^*) \\ J_{\pi(i)} \in J_T(\bar{S}^*) \\ J_{\pi(i)} \in J_T(\bar{S}^*) \end{array} \right]$$

$$- \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* - \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* - \sum_{i=x}^y z_{\pi(i)} \varepsilon =$$

$$\left[\begin{array}{l} J_{\pi(i)} \in J_E(\bar{S}^*) \\ J_{\pi(i)} \in J_T(\bar{S}^*) \end{array} \right]$$



Rys. 8.3. Ilustracja graficzna warunków optymalności bloku
 a/ zaznaczono blok zadań $\langle J_1, J_2, J_3 \rangle$ oraz odpowiednie funkcje kosztu $h_i(S_i) = v_i E_i + w_i T_i$,
 b/ brak spełnienia warunku $sv_{xy} \geq sw_{xy}$; pola zakreskowane odpowiadają kosztowi rozwiązania $\langle \pi, \bar{S}^* \rangle$, $\pi = \langle 1, 2, 3, \rangle$,
 c/ zmniejszenie kosztu poprzez "przesunięcie w czasie" bloku zadań

$$= \left[\sum_{i=x}^y v_{\pi(i)} - \sum_{i=x}^y w_{\pi(i)} - \sum_{i=x}^y z_{\pi(i)} \right] \xi = (sv_{xy} - sw_{xy}) \xi < 0 \quad (8.35)$$

$$\left[J_{\pi(i)} \in J_E(\bar{S}^*) \cup J_D^O(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^*) \right].$$

bowiem z założenia mamy $sv_{xy} < sw_{xy}$ oraz $\xi > 0$, co przeczy temu, że $\beta(x, y)$ jest blokiem optymalnym.

Założmy wobec tego, że nie jest spełniony warunek (ii), tzn. $\bar{sv}_{xy} > \bar{sw}_{xy}$. Możemy wówczas wybrać wartość $\xi > 0$ taką, że

$$\xi < S_{\pi(y+1)}^* - S_{\pi(y)}^* - p_{\pi(y)}, \quad \xi < -L_{\pi(i)}, \quad L_{\pi(i)}^* < 0, \quad i = x, x+1, \dots, y,$$

$$\xi < E_{\pi(i)}^*, \quad E_{\pi(i)}^* > 0, \quad i = x, x+1, \dots, y. \text{ Wartość taka istnieje bowiem}$$

$$S_{\pi(y)} + p_{\pi(y)} < S_{\pi(y+1)}. \quad \text{Wówczas zachodzi } J_T(S^* + \xi) =$$

$$= J_T(S^*) \cup J_L^O(S^*), \quad J_E(S^* + \xi) = J_E(S^*), \quad J_L(S^* + \xi) = J_L(S^*) \text{ oraz}$$

$$F(S^* + \xi; \beta(x, y)) - F(S^*; \beta(x, y)) = \sum_{i=x}^y h_{\pi(i)}(S_{\pi(i)}^* + \xi) - \sum_{i=x}^y h_{\pi(i)}(S_{\pi(i)}^*) =$$

$$= \sum_{i=x}^y v_{\pi(i)} (E_{\pi(i)}^* - \xi) + \sum_{i=x}^y w_{\pi(i)} (T_{\pi(i)}^* + \xi) + \sum_{i=x}^y z_{\pi(i)} (C_{\pi(i)}^* + \xi) -$$

$$\left[J_{\pi(i)} \in J_E(\bar{S}^* + \xi) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^* + \xi) \right]$$

$$- \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* - \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* - \sum_{i=x}^y z_{\pi(i)} C_{\pi(i)}^* = \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* -$$

$$\left[J_{\pi(i)} \in J_E(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_E(\bar{S}^*) \right]$$

$$- \sum_{i=x}^y v_{\pi(i)} + \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* + \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* + \sum_{i=x}^y w_{\pi(i)} -$$

$$\left[J_{\pi(i)} \in J_E(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_L^O(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^*) \cup J_L^O(\bar{S}^*) \right]$$

$$- \sum_{i=x}^y v_{\pi(i)} E_{\pi(i)}^* - \sum_{i=x}^y w_{\pi(i)} T_{\pi(i)}^* + \sum_{i=x}^y z_{\pi(i)} = \left[\sum_{i=x}^y w_{\pi(i)} + \right.$$

$$\left. \left[J_{\pi(i)} \in J_E(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_T(\bar{S}^*) \right] \left[J_{\pi(i)} \in J_L^O(\bar{S}^*) \right] \right]$$

$$+ \sum_{i=x}^y z_{\pi(i)} - \sum_{i=x}^y v_{\pi(i)} \Big] \xi = (\bar{s}_{xy} - \bar{s}_{xy}^*) \cdot \xi < 0, \quad (8.36)$$

$$| J_{\pi(i)} \in J_E(S^*)$$

bowiem z założenia mamy $\bar{s}_{xy}^* > \bar{s}_{xy}$ oraz $\xi > 0$, co przeczy temu, że $\langle \beta(x, y), \bar{S} \rangle$ jest blokiem optymalnym.

Niech $\beta(x, y)$ będzie blokiem maksymalnym, dla którego spełnione są warunki (i), (ii). Wówczas z wzorów (8.35), (8.36) wynika, że dla $\xi > 0$

$$F(S^* - \xi, \beta(x, y)) > F(S^*, \beta(x, y)) \text{ jeśli } t_0 < S_{\pi(k)} \quad (8.37)$$

oraz

$$F(S^*, \beta(x, y)) \leq F(S^* + \xi, \beta(x, y)). \quad (8.38)$$

Stąd wniosek, że $\langle \beta(x, y), S^* \rangle$ jest rozwiązaniem częściowym lokalnie optymalnym. Jest ono również globalnie optymalnym, bowiem przyjęta funkcja celu jest wypukła ■

Wykorzystując ideę dowodu zaproponowany został algorytm wyznaczania rozwiązania optymalnego $(\pi, \bar{S}^*)$, $\bar{S}^* \in \bar{S}_\pi$ dla danej permutacji $\pi \in \Pi(\mathcal{R})$, dla przypadku, gdy $h_i(S_i) = v_i E_i + w_i T_i + z_i C_i$.

Algorytm 8.1.

1/ $i := \emptyset$; $t := t_0$

2/ $i := i+1$

oblicz

$$S_{\pi(i)} := \max[t, r_{\pi(i)}],$$

$$C_{\pi(i)} := S_{\pi(i)} + p_{\pi(i)},$$

$$D_{\pi(i)} := S_{\pi(i)} - r_{\pi(i)},$$

$$E_{\pi(i)} := \max[0, r_{\pi(i)} - S_{\pi(i)}],$$

$$L_{\pi(i)} := C_{\pi(i)} - d_{\pi(i)},$$

$$T_{\pi(i)} := \max[0, L_{\pi(i)}].$$

3/ wyznacz wskaźnik x , $1 \leq x \leq i$ taki, że $\beta(x, i)$ jest blokiem maksymalnym, tzn.

$$x := \max \{ j \mid j = 1, 2, \dots, x, S_{\pi(j-1)} + p_{\pi(j-1)} < S_{\pi(j)} \}$$

$$S_{\pi(0)} + p_{\pi(0)} = -1$$

- 4/ Sprawdź warunek optymalności bloku $\beta(x,i)$ w następujący sposób: oblicz

$$sv_{xi} := \sum_{j=x}^i v_{\pi(j)}$$

$$J_{\pi(j)} \in J_E(\bar{S}) \cup J_D^0(\bar{S}),$$

$$sw_{xi} := \sum_{j=x}^i w_{\pi(j)} + \sum_{\substack{j=x \\ J_{\pi(j)} \in J_T(\bar{S})}}^i J_{\pi(j)}.$$

- 5/ Jeśli $x = 1$ i $S_{\pi(x)} = t_0$, to przejdź do kroku 6.

W przeciwnym przypadku oblicz wartość w/g wzoru

$$\delta := \min[\delta', \delta'', \delta'''],$$

gdzie

$$\delta' := \begin{cases} S_{\pi(x)} - t_0 & x = 1 \\ S_{\pi(x)} - C_{\pi(x-1)} & x = 2, 3, \dots, i \end{cases}$$

$$\delta'' := \min \{ D_{\pi(j)} \mid D_{\pi(j)} > 0 \quad j = k, x+1, \dots, i \}$$

$$\delta''' := \min \{ T_{\pi(j)} \mid T_{\pi(j)} > 0 \quad j = x, x+1, \dots, i \}.$$

Następnie należy wykonać

$$S_{\pi(j)} := S_{\pi(j)} - \delta \quad j = x, x+1, \dots, i$$

$$C_{\pi(j)} := C_{\pi(j)} - \delta \quad j = x, x+1, \dots, i$$

$$D_{\pi(j)} := D_{\pi(j)} - \delta \quad j = x, x+1, \dots, i$$

$$E_{\pi(j)} := \max[0, r_{\pi(j)} - S_{\pi(j)}] \quad j = x, x+1, \dots, i$$

$$L_{\pi(j)} := L_{\pi(j)} - \delta \quad j = x, x+1, \dots, i$$

$$T_{\pi(j)} := \max[0, L_{\pi(j)}] \quad j = x, x+1, \dots, i$$

$$t := C_{\pi(i)}$$

oraz przejdź do kroku 3.

- 6/ Jeśli $i < n$ to przejdź do kroku 2.

Jeśli $i = n$ to stop. Wektor $\bar{S} = \langle S_1, S_2, \dots, S_n \rangle$ wyznacza optymalne rozwiązanie $\langle \pi, \bar{S}^* \rangle$ dla danej

permutacji π . Ciąg $\langle i_0, i_1, \dots, i_l \rangle$ może być wyznaczony z warunków

$$S_{\pi(j)} + p_{\pi(j)} < S_{\pi(j+1)} \text{ dla } j = i_1, i_2, \dots, i_{l-1}$$

oraz $i_0 = 0, i_l = n$.

Ze względu na złożoność formalnego dowodu poprawności algorytmu 8.1 przedstawiony zostanie tylko krótki komentarz wyjaśniający zasadę jego działania.

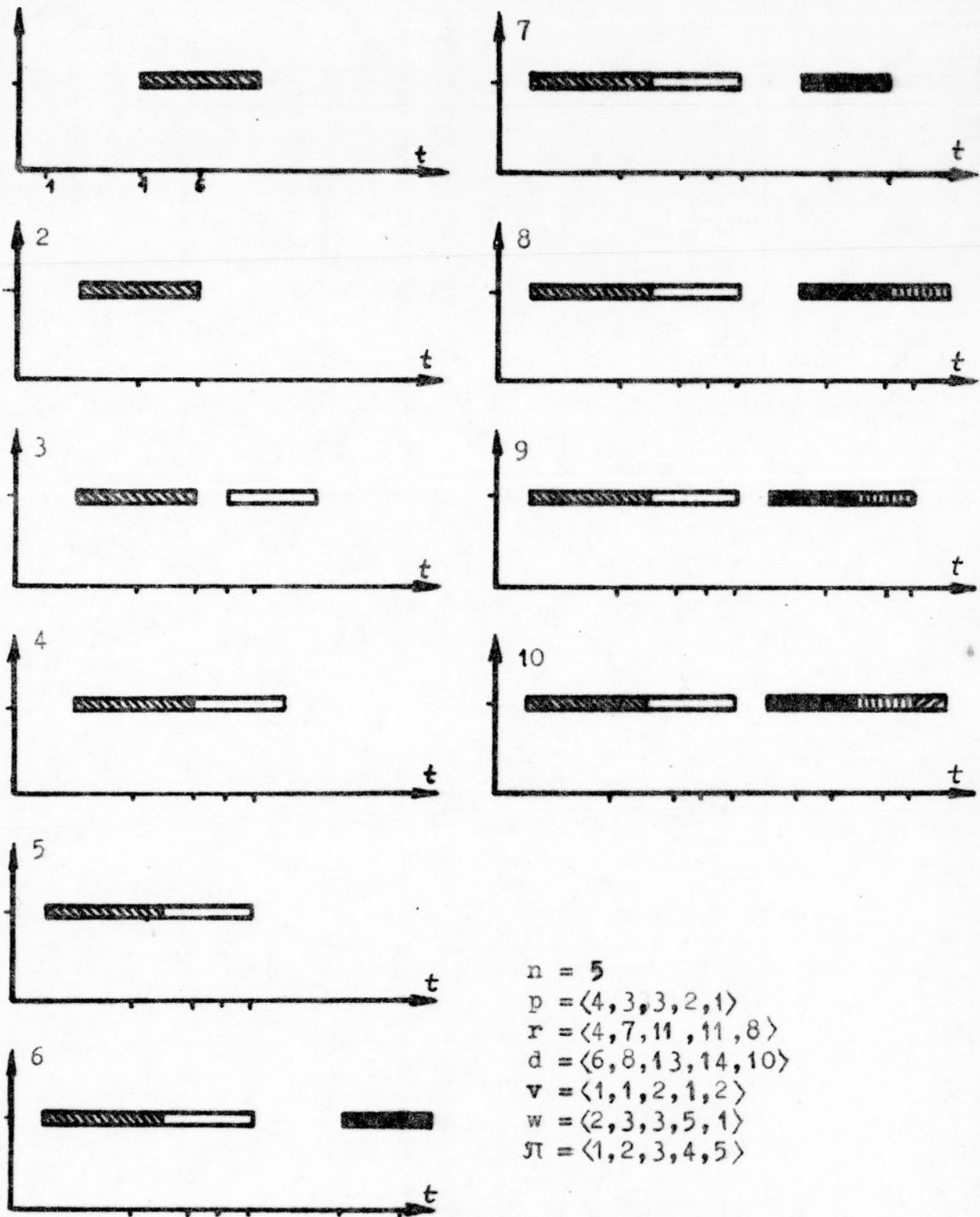
Zauważmy, że pełna realizacja kroków 2 ÷ 5 algorytmu odpowiada optymalnemu rozwiązaniu problemu częściowego zawierającego zadania $J_{\pi(1)}, J_{\pi(2)}, \dots, J_{\pi(i)}$ /dla ustalonego i /.

Niech $J_{\pi(1)}, J_{\pi(2)}, \dots, J_{\pi(i-1)}$ będzie sekwencją częściową z optymalnymi momentami rozpoczęcia zadań $S_{\pi(1)}^*, S_{\pi(2)}^*, \dots, S_{\pi(i-1)}^*$. Konstruujemy rozwiązanie optymalne dla problemu zawierającego i zadań $J_{\pi(1)}, J_{\pi(2)}, \dots, J_{\pi(i-1)}, J_{\pi(i)}$. Niech $\hat{S}_{\pi(i)}$ będzie optymalnym momentem rozpoczęcia zadania $J_{\pi(i)}$ /bez uwzględnienia pozostałych zadań/. Jeśli $S_{\pi(i-1)}^* + p_{\pi(i-1)} < \hat{S}_{\pi(i)}$ to optymalna "lokalizacja" pracy $J_{\pi(i)}$ nie ma wpływu na "lokalizację" prac wcześniejszych oraz $S_{\pi(i)}^* = \hat{S}_{\pi(i)}$ i praca $J_{\pi(i)}$ utworzy blok maksymalny. Jeśli $S_{\pi(i-1)}^* + p_{\pi(i-1)} \geq \hat{S}_{\pi(i)}$ to praca zostanie "włączona" w skład ostatniego bloku maksymalnego w permutacji częściowej $\pi(1), \pi(2), \dots, \pi(i+1)$. Oznaczmy ten blok $\beta(x, i)$. Zauważmy, że warunek (8.34) twierdzenia 8.1 jest dla $\beta(x, i)$ spełniony.

Wystarczy zatem sprawdzić warunek (8.33) twierdzenia. Jeśli jest on spełniony to $\beta(x, i)$ jest blokiem optymalnym. Ponieważ wszystkie bloki wcześniejsze /jeśli istnieją/ są optymalne więc na mocy (8.23) otrzymane rozwiązanie problemu częściowego z pracami $J_{\pi(1)}, J_{\pi(2)}, \dots, J_{\pi(i-1)}, J_{\pi(i)}$ jest również optymalne. Jeśli warunek (8.33) twierdzenia 8.1 nie jest spełniony, to korzystając z dowodu twierdzenia możemy wyznaczyć wartość $\delta > 0$ taką, że

$$F(S - \delta ; \beta(x, i)) \leq F(S ; \beta(x, i)) \quad (8.39)$$

oraz wyznaczyć nowe terminy, $S_{\pi(j)} := S_{\pi(j)} - \delta$ rozpoczęcia zadań z bloku $\beta(x, i)$. Proces wyznaczania wartości δ oraz zmiany terminów $S_{\pi(j)}$ są kontynuowane do momentu spełnienia warunku (8.33) /kroki 3 - 5 algorytmu/.



Rys. 8.4. Kolejne kroki algorytmu 8.1

Własność 8.1.

Algorytm 8.1 posiada złożoność $O(n^3)$. ■

Dowód

Dla ustalonego wskaźnika i / $i = 1, 2, \dots, n$ / wykonanie kroku 3 wymaga co najwyżej n iteracji, wykonanie kroku 4. wymaga co najwyżej n iteracji, wyznaczenia wartości $\tilde{\sigma}$ /krok 5/ wymaga co najwyżej $3n$ iteracji, oraz wyznaczenie pozostałych wartości w kroku 5 wymaga co najwyżej n iteracji. Niespełnienie warunku optymalności w kroku 4 powoduje powtórzenie kroków 3 - 5 co najwyżej $3n$ razy /maleje zbiór prac branych pod uwagę przy wyznaczaniu wartości $\tilde{\sigma}$ /. W konsekwencji złożoność całego algorytmu jest $O(n^2)$. ■

Rozważmy dalej problem $n \mid 1 \leq h_i$ dla przypadku, gdy $g_i(E_i)$, $f_i(C_i)$ $i = 1, 2, \dots, n$ są funkcjami wypukłymi, niemalejącymi. Założenie o wypukłości zapewnia istnienie optimum globalnego problemu (8.18) - (8.20). Założymy dodatkowo, że funkcje $g_i(E_i)$, $f_i(T_i)$, są ciągłe i różniczkowalne na półprostej $(0, \infty)$ $i = 1, 2, \dots, n$. Przyjmujemy $g'_i(0) \triangleq \lim_{E_i \rightarrow 0^+} g'_i(E_i)$, $f'_i(0) \triangleq \lim_{T_i \rightarrow 0^+} f'_i(T_i)$.

Zdefiniujmy dalej wielkości

$$SV_{xy} \triangleq \sum_{i=x}^J g_{\pi(i)}(E_{\pi(i)}^*). \quad (8.40)$$

$$J_{\pi(i)} \in J_D^O(\bar{S}^*) \cup J_E(\bar{S}^*)$$

$$SW_{xy} \triangleq \sum_{i=x}^Y f_{\pi(i)}(T_{\pi(i)}^*). \quad (8.41)$$

$$J_{\pi(i)} \in J_T(\bar{S}^*)$$

$$SV_{xy} \triangleq \sum_{i=x}^Y g_{\pi(i)}(E_{\pi(i)}^*). \quad (8.42)$$

$$J_{\pi(i)} \in J_E(\bar{S}^*)$$

$$SW_{xy} \triangleq \sum_{i=x}^Y f_{\pi(i)}(T_{\pi(i)}^*) \quad (8.43)$$

$$J_{\pi(i)} \in J_T(\bar{S}^*) \cup J_L^O(\bar{S}^*).$$

Wówczas warunki optymalności bloku /analogiczne do (8.33)-(8.34) / przyjmują postać

$$SV_{xy} \geq SW_{xy} \quad \text{gdy} \quad t_0 < S_{x(x)}^* , \quad (8.44)$$

$$\bar{SV}_{xy} \leq \bar{SW}_{xy} \quad (8.45)$$

Dowód poprawności warunków (8.44) - (8.45) jest analogiczny do dowodu twierdzenia 8.1.

8.4. Kryteria eliminacji

Przyjmujemy w dalszym ciągu, że $A_i \triangleq \{J_i \in J \mid < J_i, J_j > \in \mathcal{R}\}$
 $B_i \triangleq \{J_i \in J \mid < J_j, J_i > \in \mathcal{R}\}$ są zbiorami odpowiednio następników
 oraz poprzedników zadania J_i - tego. Niech a_i, b_i oznacza odpowied-
 nio najwcześniejszy moment rozpoczęcia i najpóźniejszy moment
 zakończenia realizacji zadania i -tego. Niech $\hat{A}_i = A_i \cup \{i\}$,
 $\hat{B}_i = B_i \cup \{i\}$.

Twierdzenia 8.2 - 8.5 formułują kryteria eliminacji dla
 problemu $n \mid 1 \mid r_i \geq 0, \prec \mid \sum f_i, f_i(S_i)$ - funkcje niemalejące $J_i \in J$.

Twierdzenie 8.2

Jeśli dla dwóch zadań $J_j, J_k \in J$ zachodzi

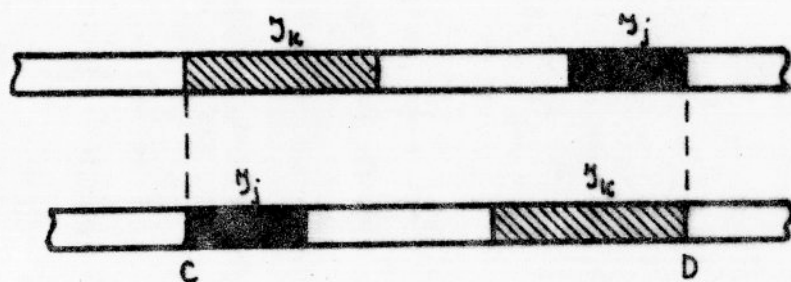
- (i) $f_j(t) - f_k t$ jest niemalejąca w przedziale $(a_k + p_k, b_j)$,
- (ii) $r_j \leq r_k$
- (iii) $p_j \leq p_k$
- (iv) $\max_{J_i \in J - \hat{A}_j - \hat{B}_k} r_i \leq r_k + p_j$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza
 J_k . ■

Dowód

Rozważmy sekwencję zadań taką, że zadanie J_k poprzedza zadanie
 J_j . Oznaczmy przez C moment rozpoczęcia wykonywania zadania J_k ,
 przez D moment zakończenia zadania J_j . Porównajmy tę sekwencję
 z sekwencją taką, że J_k poprzedza J_j , rys. 8.5.

Udział w koszcie całkowitym wszystkich prac z wyjątkiem J_k
 zmaleje lub pozostanie niezmienny ze względu na warunki (ii) oraz
 (iv). Natomiast dla J_k zachodzi



Rys. 8.5. Porównanie sekwencji zadań

$$a_k + p_k \leq C + p_k \leq D \leq b_j \quad . \quad (8.46)$$

korzystając z warunku (i) otrzymujemy

$$f_j(D) - f_k(D) \geq f_j(C + p_k) - f_k(C + p_k) , \quad (8.47)$$

zaś z warunku (iii)

$$f_j(C + p_k) \geq f_j(C + p_j) , \quad (8.48)$$

oraz w rezultacie

$$f_j(D) - f_k(C + p_k) \geq f_j(C + p_j) + f_k(D) . \quad (8.49)$$

Oznacza to, że udział zadań J_k, J_j w koszcie całkowitym również zmalał /lub pozostał niezmienny/.

Zauważmy, że jeśli $f_j(t), f_k(t)$ są funkcjami ciągłymi różniczkowalnymi w przedziale $(a_k + p_k, b_j)$ to warunek (i) twierdzenia może być zastąpiony warunkiem $df_j(t)/dt \geq df_k(t)/dt$ dla każdego $t \in (a_k + p_k, b_j)$

Własność 8.2.

Jeśli $r_i = r^*$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_i, J_k \in J$ zachodzi

$$(i) \quad f_j(t) - f_k(t) \text{ jest niemalejąca w przedziale } (r^* + \sum_{J_i \in \hat{B}_k} p_i, r^* + p_i)$$

$$(ii) \quad p_j \leq p_k ,$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k .

Dowód

Warunki (i), (iv) twierdzenia są oczywiste oraz $a_k + p_k =$

$$= r^* + \sum_{J_i \in \hat{B}_k} p_i + p_k = r^* + \sum_{J_i \in \hat{B}_k} p_i, \quad b_j = r^* + \sum_{J_i \in J} p_i - \sum_{J_i \in A_j} p_i = r^* + \sum_{J_i \in A_j} p_i.$$

Własność 8.2.

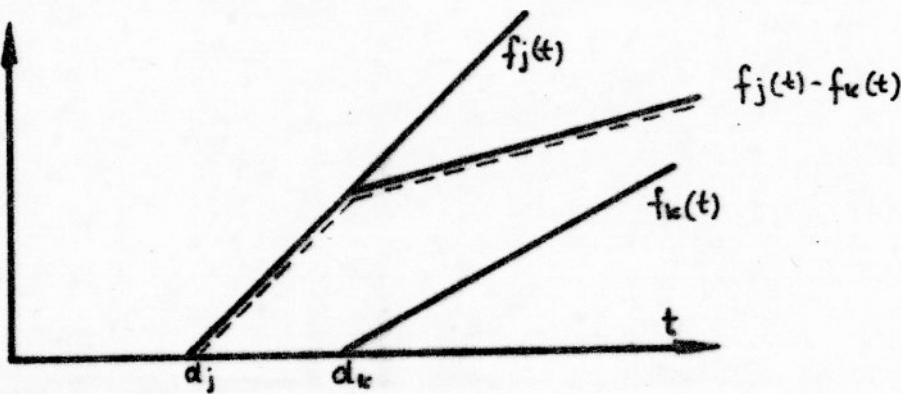
Jeśli $f_i(t) = w_i T_i$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

- (i) $d_j \leq d_k$
- (ii) $w_j \geq w_k$
- (iii) $p_j \leq p_k$
- (iv) $r_j \leq r_k$
- (v) $r_i \leq r_k + p_j \quad J_i \in J - \hat{A}_j - \hat{B}_k$

to istnieje optymalne uszeregowanie zadań, takie że J_j poprzedza J_k . ■

Dowód

Stosujemy twierdzenie 8.3 z $A_j = \emptyset = B_k$. Warunki (i), (ii) oznaczają, że funkcja $f_j(t) - f_k(t)$ jest niemalejąca w przedziale $(0, \max_{J_i \in J} b_i)$, rys. 8.6. ■



Rys. 8.6. Ilustracja warunków (i) - (ii) własności 8.2

Własność 8.2

Jeśli $f_i(t) = w_i T_i$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

- (i) $d_j \leq a_k + p_k$,

- (ii) $w_j \geq w_k$,
- (iii) $p_j \leq p_k$,
- (iv) $r_j \leq r_k$,
- (v) $r_i \leq r_k + p_j$, $J_i \in J - \hat{A}_j - \hat{B}_k$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód

Warunek (i) powoduje, że $f_j(t) > 0$ dla $t > a_k + p_k$ i z warunku (ii) wynika, że $f_j(t) - f_k(t)$ jest niemalejąca w wymaganym przedziale. ■

Twierdzenie 8.3

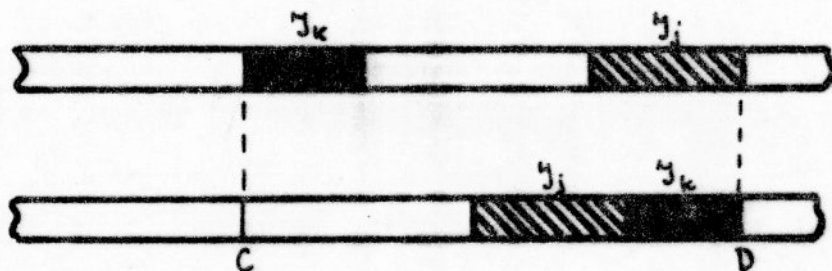
Jeśli dla dwóch zadań $J_j, J_k \in J$ zachodzi

- (i) $f_j(t) - f_k(t)$ niemalejąca w przedziale $(b_j - p_k, b_j)$,
- (ii) $\max_{J_i \in J - (A_j \cup \hat{B}_k)} r_i \leq r_k$,
- (iii) $f_k(a_k + p_k) = f_k(b_j - p_k)$,

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód

Z warunków (i) - (iii) wynika, że jeśli $f_j(t) - f_k(t)$ jest niemalejąca w przedziale $(a_k + p_k, b_j)$ oraz $p_j \leq p_k$ to możemy zastosować twierdzenie 8.2. Załóżmy dalej, że $p_j > p_k$. Rozważmy sekwencję zadań taką, że J_k poprzedza J_j . Oznaczmy przez C moment rozpoczęcia zadania J_k -tego oraz przez D moment zakończenia zadania J_j -tego. Porównajmy tę sekwencję z sekwencją otrzymaną przez umieszczenie J_k bezpośrednio za J_j , rys. 8.7.



Rys. 8.7. Porównanie sekwencji zadań

Udział w koszcie całkowitym wszystkich prac z wyjątkiem k -tej zmaleje lub pozostanie niezmienny na mocy warunku (ii). Dla prac J_k, J_j natomiast zachodzi

$$a_k + p_k \leq C + p_k < D - p_k \leq b_j - p_k. \quad (8.50)$$

Ze względu na warunek (iii) mamy

$$f_k(D - p_k) = f_k(C + p_k), \quad (8.51)$$

zaś ze względu na warunek (i) otrzymujemy

$$f_j(D) - f_k(D) \geq f_j(D - p_k) - f_k(D - p_k). \quad (8.52)$$

Stąd możemy zapisać

$$f_j(D) + f_k(C + p_k) \geq f_j(D - p_k) + f_k(D), \quad (8.53)$$

co oznacza, że udział zadań J_j, J_k w koszcie całkowitym zmaleje lub pozostanie zmienny. ■

Własność 8.3

Jeśli zachodzi $r_i = r^*$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

$$(i) \quad f_j(t) - f_k(t) \quad \text{niemalejąca w przedziale } (r^* + \sum_{J_i \notin A_j} p_i - p_k, r^* + \sum_{J_i \notin A_j} p_i)$$

$$(ii) \quad f_k(r^* + \sum_{J_i \in B_k} p_i) = f_k(r^* + \sum_{J_i \in A_j} p_i - p_k).$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód

Warunek (ii) jest oczywisty oraz $b_j = r^* + \sum_{J_i \in J} p_i - \sum_{J_i \in A_j} p_i =$

$$b_j - p_k = r^* + \sum_{J_i \notin A_j} p_i - p_k, \quad a_k + p_k = r^* + \sum_{J_i \in B_k} p_i + p_k = \sum_{J_i \in B_k} p_i.$$

Własność 8.3'

Jeśli $f_i(t) = w_i T_i$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

$$(i) \quad d_k \geq b_j - p_k,$$

$$(ii) \quad d_j \leq d_k,$$

$$(iii) w_j \geq w_k ,$$

$$(iv) r_i \leq r_k , \quad J_i \in J - A_j - \hat{B}_k ,$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód

Warunek (i) pociąga za $f_k(b_k + p_k) = f_k(b_j - p_k)$ oraz z (i), (iii) otrzymujemy, że $f_j(t) - f_k(t)$ jest niemalejąca w przedziale $(0, \max_{J_i \in J} b_i)$ /rys.8.7/. ■

Twierdzenie 8.4.

Jeśli dla dwóch zadań $J_j, J_k \in J$ zachodzi

$$(i) f_k(a_k + p_k) = f_k(b_j) ,$$

$$(ii) \max_{J_i \in J - A_j - \hat{B}_k} r_i \leq r_k ,$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód

Z (i) wynikają warunki (i), (iii) twierdzenia 8.3. Stosując więc twierdzenie 8.3 otrzymujemy powyższy rezultat. ■

Własność 8.4.

Jeśli zachodzi $r_i = r^*$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

$$(i) f_k(r^* + \sum_{J_i \in \hat{B}_k} p_i) = f_k(r^* + \sum_{J_i \in A_j} p_i)$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Dowód analogiczny jak w własności 8.3.

Własność 8.4'.

Jeśli $f_i(t) = w_i T_i$ dla każdego $J_i \in J$ oraz dla dwóch zadań $J_j, J_k \in J$ zachodzi

$$(i) d_k \geq b_j ,$$

$$(ii) r_i \leq r_k \quad J_i \in J - A_j - \hat{B}_k$$

to istnieje optymalne uszeregowanie zadań takie, że J_j poprzedza J_k . ■

Twierdzenie 8.5.

Jeśli dla pewnego zadania $J_k \in J$ zachodzi

- (i) $f_k(a_k + p_k) = f_k(b_k)$,
 (ii) $\max_{J_i \in J - \hat{A}_k} r_i \leq r_k$,

to istnieje optymalne uszeregowanie zadań takie, że zadanie J_k następuje za wszystkimi zadaniami ze zbioru $J - \hat{A}_k$. ■

Dowód

Warunek (i) pociąga warunek (i) z twierdzenia 8.4. Stosując twierdzenie 8.4 do każdej pary $j \in J - \hat{A}_k$, $k \in J$, $j \neq k$ otrzymujemy powyższy rezultat. ■

Własność 8.5.

Jeśli $r_i = r^*$ dla każdego $J_i \in J$ oraz dla pewnego zadania $J_k \in J$ zachodzi

$$(i) \quad f_k(r^* + \sum_{J_i \in \hat{B}_k} p_i) = f_k(r^* + \sum_{J_i \notin \hat{A}_k} p_i)$$

to istnieje optymalne uszeregowanie zadań takie, że J_k następuje za wszystkimi zadaniami ze zbioru $J - \hat{A}_k$. ■

Własność 8.5.

Jeśli $f_i(t) = w_i T_i$ dla każdego $J_i \in J$ oraz dla pewnego zadania $J_k \in J$ zachodzi

$$(i) \quad d_k \geq \max_{J_i \in J - \hat{A}_k} b_i + p_k$$

to istnieje optymalne uszeregowanie zadań takie, że J_k następuje za wszystkimi zadaniami ze zbioru $J - \hat{A}_k$. ■

Własności 8.2, 8.3, 8.4 dla przypadku $r_i = 0$ zostały udowodnione w pracy [71]

Głównym problemem powstającym przy zastosowaniu podanych kryteriów eliminacji jest możliwość stworzenia cykli poprzedzeń /w przypadku sprzecznych reguł poprzedzania wynikających z dwóch twierdzeń np. J_j poprzedza J_k , J_k poprzedza J_j /.

Trudność tą można ominąć poprzez wyznaczanie tranzytywnego domknięcia relacji poprzedzeń [71], [31] po każdym spełnieniu kryterium eliminacji.

Analizując postać warunków w kryteriach eliminacji /tw. i wł.

8.2 - 8.5/ można zauważyć, że ze wzrostem złożoności problemu rośnie liczba warunków, które należy sprawdzić dla spełnienia kryterium eliminacji. Jednocześnie maleje liczba rozwiązań potencjalnie eliminowanych przez zastosowanie podanych kryteriów

Prowadzone przez autora badania doprowadziły do sformułowania podobnych kryteriów eliminacji dla problemu $n|1| \prec |\sum h_i$. Jednakże liczba warunków występująca nawet w najprostszych kryteriach była tak duża, że "zysk" z ich zastosowania wydaje się wątpliwy.

W konsekwencji sformułowane zostało jedno ogólne twierdzenie eliminacyjne odnoszące się zarówno do problemu $n|1|r_i \geq 0, \prec |\sum f_i$ jak i do $n|1| \prec |\sum h_i$.

Twierdzenie 8.6.

Niech $\langle \pi, \bar{S}^* \rangle, \bar{S}^* \in \bar{S}_\pi$ będzie rozwiązaniem optymalnym, dla danej permutacji $\pi \in \Pi(\mathcal{R})$, z blokami wyznaczonymi przez ciąg $\langle i_0, i_1, \dots, i_l \rangle$. Jeśli zachodzi

$$\langle J_{\pi(i)}, J_{\pi(i+1)} \rangle \in \mathcal{R} \quad i = i_j + 1, i_j + 2, \dots, i_{j+1} - 1, j = 0, 1, 2, \dots, l-1 \quad (8.54)$$

to

$$F(\bar{S}^*; \pi) \leq F(\bar{S}; \sigma) \quad \forall \bar{S} \in \bar{S}_\sigma \quad \forall \pi, \sigma \in \Pi(\mathcal{R}). \quad (8.55)$$

Dowód

Rozwiązanie $\langle \pi, \bar{S}^* \rangle, \bar{S}^* \in \bar{S}_\pi$ można znaleźć przez rozwiązanie problemu postaci

$$F(\bar{S}^*; \pi) = \min_{\bar{S}} \sum_{i=1}^n h_i(S_i), \quad (8.56)$$

przy ograniczeniach

$$t_0 \leq S_{\pi(i)} \quad i = 1, 2, \dots, n, \quad (8.57)$$

$$S_{\pi(i)} + p_{\pi(i)} \leq S_{\pi(i+1)} \quad i = 1, 2, \dots, n-1, \quad (8.58)$$

Korzystając z definicji bloków maksymalnych możemy ograniczenia (8.58) zapisać w formie

$$S_{\pi(i)} + p_{\pi(i)} \leq S_{\pi(i+1)} \quad i = i_j + 1, i_j + 2, \dots, i_{j+1} - 1 \quad (8.59) \\ j = 0, 1, 2, \dots, l-1$$

$$S_{\pi(i_j)} + p_{\pi(i_j)} \leq S_{\pi(i_{j+1})} \quad j = 0, 1, 2, \dots, l-1. \quad (8.60)$$

Na mocy założeń twierdzenia ograniczenia (8.59) są ograniczeniami wprowadzanymi przez relację $\mathcal{R}$, zaś z definicji bloków ograniczenia (8.60) są nieaktywne, czyli w rezultacie możemy zapisać

$$F(\bar{S}^*; \pi) = \min_S \sum_{i=1}^n h_i(S_i) , \quad (8.61)$$

$$t_0 \leq S_{\pi(i)} \quad i = 1, 2, \dots, n , \quad (8.62)$$

$$S_{\pi(i)} + p_{\pi(i)} \leq S_{\pi(j)} < J_{\pi(i)}, J_{\pi(j)} > \in \mathcal{R} \quad (8.63)$$

Niech dalej $\sigma \in \Pi(\mathcal{Q})$ oraz

$$F(\hat{S} ; \sigma) = \min_S \sum_{i=1}^n h_i(S_i) , \quad (8.64)$$

$$t_0 \leq S_{\sigma(i)} \quad i = 1, 2, \dots, n , \quad (8.65)$$

$$S_{\sigma(i)} + p_{\sigma(i)} \leq S_{\sigma(i+1)} \quad i = 1, 2, \dots, n-1 . \quad (8.66)$$

Zauważmy, że ograniczenia (8.65) nie zależą od wyboru permutacji, zaś ograniczenia (8.66) zawierają ograniczenia (8.63) bowiem relacja $\mathcal{R}$ jest spełniona w σ , $\sigma \in \Pi(\mathcal{Q})$. Problem (8.61) - (8.63) jest więc relaksacją problemu (8.64) - (8.65) oraz

$$F(S^*; \pi) \leq F(\hat{S} ; \sigma) \leq F(S ; \sigma) , \quad (8.67)$$

dla $S^* \in \mathcal{S}_\pi$, $\hat{S}$, $S \in \mathcal{S}_\sigma$, co kończy dowód. ■

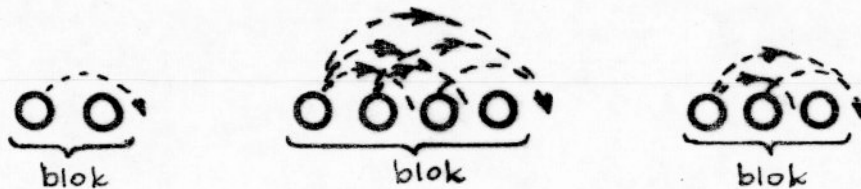
Z twierdzenia wynika, że jeśli relacja poprzedzeń $\mathcal{R}$ posiada własność (8.64) to otrzymanie rozwiązania jest bioptymalne. Wniosek ten stał się podstawą konstrukcji algorytmu opartego na metodzie podziału i ograniczeń.

8.5. Schemat przebiegu algorytmu

Zasada działania algorytmu jest oparta na metodzie podziału i ograniczeń z mieszaną strategią przeglądania drzewa rozwiązań.

Rozpoczynając od pewnej permutacji początkowej $\pi^0 \in \Pi(\mathcal{Q})$ generujemy ciąg permutacji $\pi \in \Pi(\mathcal{Q})$. Dla każdej permutacji z ciągu wyznaczamy wartość kosztu $F(S^*; \pi)$ oraz określamy bloki maksymalne zadań w rozwiązaniu $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in \mathcal{S}_\pi$. Każdą nową

permutację $\beta \in \Pi(\mathcal{Q})$ otrzymujemy z pewnej permutacji π z ciągu przez przesunięcie jednego zadania $J_{\pi(i)}$ -tego z bloku j -tego na s -tą pozycję w bloku $i_{j-1}+1 \leq i < s \leq i_j$, $j = 1, 2, \dots, l$ rys. 8.8.



Rys. 8.8. Schemat przesuwania zadań w blokach.

/Strzałki wskazują możliwe przesunięcia dla $\mathcal{Q} = \emptyset$ /

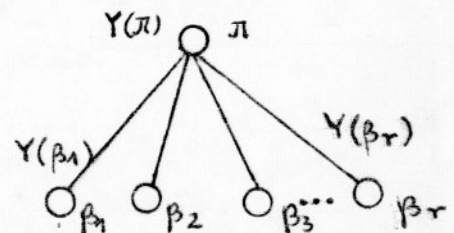
Jeżeli dla permutacji β będziemy mogli wykazać, że przesuwanie prac w β nie spowoduje zmniejszenia kosztu, to odrzucamy rozpatrywaną permutację β i cofamy się do permutacji π z której wygenerowaliśmy β . Proces generowania permutacji przedstawimy w postaci drzewa H . Każdy węzeł w H będzie przedstawiał pewną permutację π , a łuk w H będzie odpowiadał pewnej parze permutacji $\langle \pi, \beta \rangle$ takiej, że β została otrzymana przez przesunięcie jednego zadania w π .

Wygenerowanie nowej permutacji /węzła H / wiąże się z wyborem pewnego zadania do przesunięcia w π . Wybór ten jest związany z zasadą podziału węzłów w H . Jeżeli nową permutację β otrzymaliśmy przez przesunięcie w π zadania $J_{\pi(i)}$ -tego z pozycji i -tej w π /z j -tego bloku/ na pozycję s -tą, to ustalamy wymagania częściowego porządku

$$\cdot \{ J_{\pi(h)} < J_{\pi(i)} \mid i_{j-1} + 1 \leq h \leq s, h \neq i \} . \quad (8.68)$$

Wymagania te muszą być respektowane w każdym następniku względem β w H .

Generowanie bezpośrednich następników β_r /rys. 8.9/ względem π powoduje, że zbiór rozwiązań dopuszczalnych $Y(\pi)$ odpowiadających węzłowi π zostaje rozbity na rozłączne podzbiory $Y(\beta_r)$. Z każdym z tych



Rys. 8.9. Fragment drzewa rozwiązań

podzbiorów jest związany warunek logiczny, że pewne zadanie z pozycji i -tej w j -tym bloku, /którego przesunięcie w π wygenerowało β_r / ma być wykonywane jako s -te w j -tym bloku w π . Warunek ten jest realizowany poprzez ustalenie wymagań częściowego porządku (6.68). Dla ustalenia uwagi niech β_1 będzie pierwszym następnikiem względem π wygenerowanym przez przesunięcie zadania z $i = i_{j-1}+1$ -tej pozycji na pozycję $s = i_j$, w j -tym bloku w π . Po dokonaniu cofania z β_1 do π otrzymujemy warunek logiczny dopełniający, że zadanie z i -tej pozycji w π / $i = i_{j-1}+1$ / nie może być przesuwane na $s = i_j$ - tą pozycję w π . Warunek ten jest przestrzegany w kolejnych następnikach β_r generowanych z π powodując zachowanie rozłączności podzbiorów $Y(\beta_r)$. Dalej niech β_2 będzie permutacją otrzymaną przez przesunięcie zadania z pozycji $i = i_{j-1}+2$ -tej na pozycję $s = i$ -tą w j -tym bloku w π . Wtedy po cofaniu otrzymujemy kolejny warunek logiczny, że zadanie z $i = i_{j-1}+2$ -tej pozycji nie może być wykonywane na $s = i_j$ -tej pozycji w j -tym bloku w π . Łącząc ten warunek z poprzednim otrzymujemy, że zadania z $i = i_{j-1}+1$ -tej oraz $i = i_{j-1}+2$ -tej pozycji nie mogą być przesuwane na $s = i_j$ -tą pozycję w j -tym bloku w π . Warunek ten jest przestrzegany we wszystkich β_r , $r \geq 3$. Jeżeli wszystkie zadania z i -tej pozycji ($i = i_{j-1}+1, i_{j-1}+2, \dots, i_j-1$) były przesuwane na pozycję $s = i_j$, to otrzymujemy po cofaniu warunek logiczny dopełniający, że żadne zadanie i -te ($i = i_{j-1}+1, i_{j-1}+2, \dots, i_j-1$) nie może być przesuwane na $s = i_j$ -tą pozycję w j -tym bloku w π . Stąd wniosek, że zadanie $J_{\pi}(i)$, $i = i_j$ musi być umieszczone na $s = i_j$ -tej pozycji w j -tym bloku w π , czyli możemy ustalić wymagania częściowego porządku

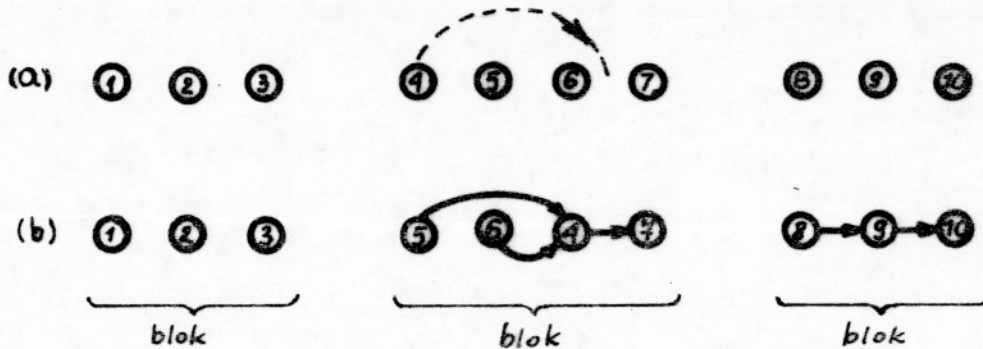
$$\{ J_{\pi}(h) \prec J_{\pi}(i_j) \mid i_{j-1}+1 \leq h < i_j \} . \quad (8.69)$$

Dalej niech β_t będzie permutacją otrzymaną przez przesunięcie zadania z pozycji $i = i_{j-1}+1$ na pozycję $s = i_j-1$ w j -tym bloku w π . Wtedy po cofaniu się otrzymujemy nowy warunek logiczny, że zadanie z pozycji $i = i_{j-1}+1$ nie może być przesuwane na pozycję $s = i_j-1$, w j -tym bloku w π . Postępując podobnie z pozostałymi zadaniami z pozycji $i = i_{j-1}+1, i_{j-1}+2, \dots, i_j-2$ otrzymamy po wykonaniu kolejnych cofań, że żadne z zadań z pozycji i -tej

$i = i_{j-1}+1, i_{j-1}+2, \dots, i_j-2$ nie może być przesuwane na pozycję $s = i_j-1$ w j -tym bloku w π . Stąd wniosek, że zadanie $J_{\pi}(i)$,

$i = i_j - 1$ musi być umieszczone na pozycji $s = i_j - 1$ w j -tym bloku w π . Łącząc ten wniosek z poprzednim otrzymujemy, że zadanie $J_{\pi(i)}$, $i = i_j - 1$ musi być umieszczone na pozycji $s = i_j - 1$ oraz zadanie $J_{\pi(i)}$, $i = i_j$ musi być umieszczone na pozycji $s = i_j$ w j -tym bloku w π , czyli możemy ustalić wymagania częściowego porządku /rys. 8.10/ w postaci

$$\{J_{\pi(h)} \prec J_{\pi(i_j-1)} \mid i_{j-1} + 1 \leq h < i_j - 1\} \cup \{J_{\pi(i_1-1)} \prec J_{\pi(i_j)}\}. \quad (8.70)$$



Rys. 8.10. Sposób ustalania relacji (b) po wykonaniu przesunięcia (a)

Postępując podobnie dla pozostałych zadań z bloku otrzymujemy po połączeniu wszystkich warunków dopełniających relację częściowego porządku w postaci

$$\bigcup_{i=i_{j-1}+1}^{i_j-1} \{J_{\pi(i)} \prec J_{\pi(i+1)}\}, \quad (8.71)$$

dla j -tego bloku w π .

Jeśli relacja częściowego porządku została ustalona w podanej postaci dla wszystkich bloków $j = 1, 2, \dots, l$ w π , to na mocy twierdzenia 8.6 rozwiązanie $\langle \pi, \bar{S}^* \rangle$, $\bar{S}^* \in \bar{S}_\pi$ jest bioptymalne.

Jeżeli permutacja β w algorytmie jest otrzymana przez przesunięcie zadania z j -tego bloku w π , to dla β ustalamy arbitralnie wymagania częściowego porządku postaci (8.71) dla ostatnich $j+1, j+2, \dots, l$ bloków /rys. 8.10/. Zapewnia to rozłączność następników /w sensie podzbiorów rozwiązań/, generowanych przez przesuwanie zadań w różnych blokach.

Węzeł π w H możemy odrzucić, jeżeli:

a/ dolne ograniczenie wartości kosztu $LB(\beta)$ dla wszystkich następ-

ników ρ wygenerowanych z π jest nie mniejsze niż aktualne górne ograniczenie,

b/ Każdy z bloków zawiera dokładnie jedno zadanie, tzn.

$$i_{j-1}+1 = i_j, \quad j = 1, 2, \dots, l,$$

c/ Żadne z zadań nie może być przesuwane w π bez naruszania ustalonych wymagań częściowego porządku,

d/ wszystkie bloki mają ustaloną relację postaci (8.54) .

Przypadki b/, c/, d/ wynikają z twierdzenia 8.6.

8.6. Wyznaczanie dolnych ograniczeń

Dla każdego wężła π w H należy wyznaczyć dolne ograniczenie wartości funkcji celu problemu $n|1|\mathcal{R}_\pi|\sum h_i$, gdzie $\mathcal{R}_\pi$ jest relacją częściowego porządku ustaloną w węźle π .

W tym rozdziale przedstawiono 3 metody wyznaczania dolnego ograniczenia oparte na

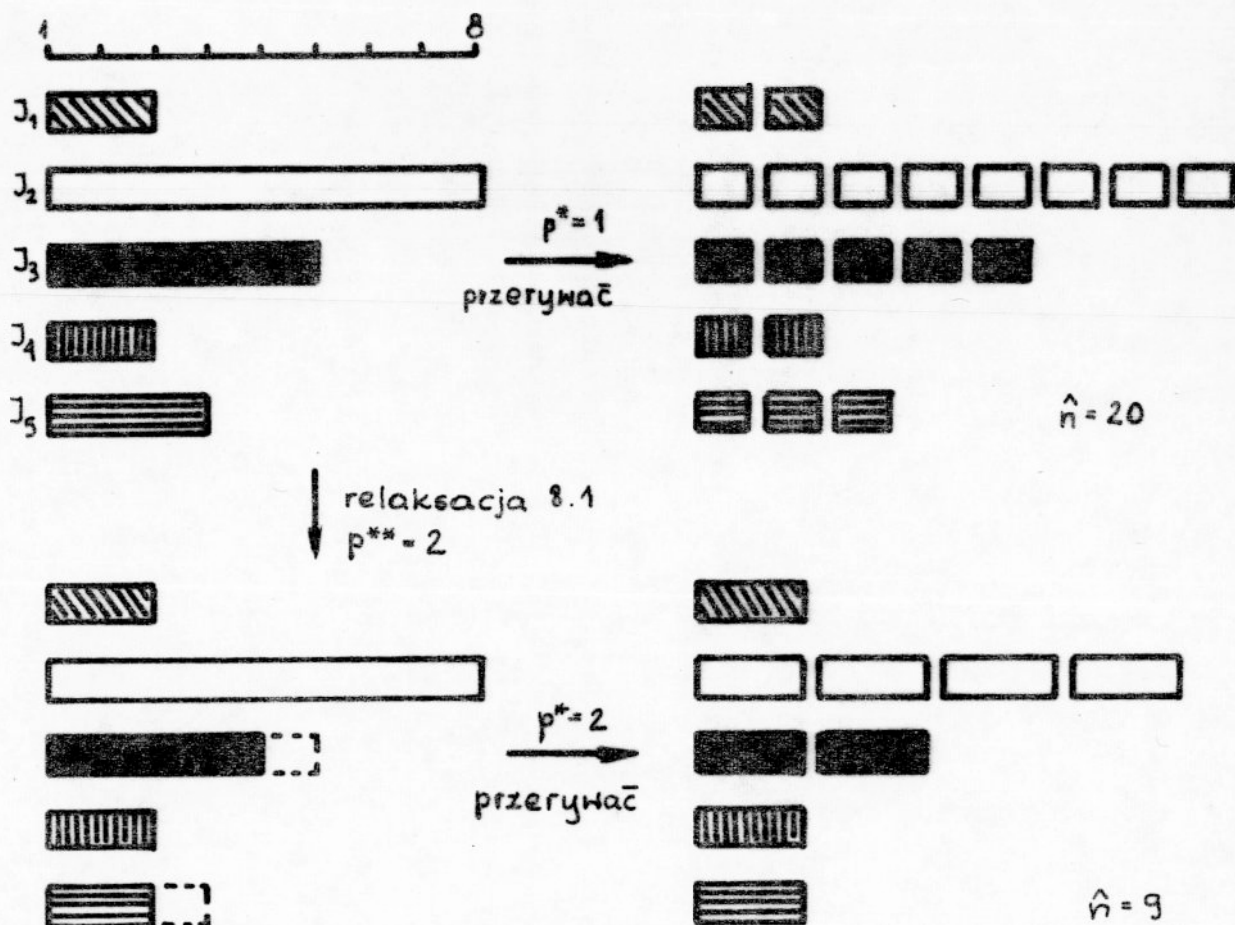
- (i) relaksacji dopuszczającej przerywanie zadań (LB1)
- (ii) rozwiązywaniu problemu dualnego programowania dyskretnego (LB2) ,
- (iii) rozwiązywaniu zrealizowanej postaci LB2 (LB3) .

Z przedstawionych dalszych ograniczeń najkorzystniejsze cechy posiada LB2, jednakże dużą przeszkodą w jego zastosowaniu jest złożoność pseudowielomianowa algorytmu mająca wyraźny wpływ na czas obliczeń. Dlatego też większy nacisk położony został na różnego rodzaju relaksacje prowadzące do otrzymania algorytmów o mniejszej złożoności (LB3).

Dolne ograniczenie LB1. Niech $p_i, r_i, d_i, i = 1, 2, \dots, n$ będą liczbami całkowitymi, zaś p^* niech będzie największym wspólnym dzielnikiem liczb $p_i, r_i, d_i, i = 1, 2, \dots, n$. Załóżmy, że poszczególne operacje są podzielne z dokładnością do p . Wobec tego każde zadanie J_i możemy traktować jak sekwencję p_i/p^* nowych zadań o jednakowej długości $p^*/\text{rys. 8.11/}$.

Zauważmy dalej, że rozwiązanie problemu $n|1|p_i = p^*|\sum f_i$ można otrzymać przez sprowadzenie go do zagadnienia przydziału (AP) posiadającego algorytm efektywny.

Założmy chwilowo, że $h_i(S_i) = v_i E_i + w_i T_i$. Niech $[0, T]$ będzie podziałem o tej własności, że nie spowoduje utraty rozwiązania optymalnego oraz T jest wielokrotnością p^* . Niech $J'_{x_i+1} J'_{x_i+2}, \dots,$



Rys. 8.11. Wyznaczanie dolnego ograniczenia poprzez dopuszczenie przerywania zadań

..., ..., $J_{y_i}^i$ oznacza zbiór zadań otrzymanych przez podzielenie zadania J_i -tego, $x_i \triangleq \sum_{j=1}^{i-1} p_j/p^*$, $y_i \triangleq \sum_{j=1}^i p_j/p^*$ $i = (1, 2, \dots, n)$. Niech $\hat{n} \triangleq \sum_{j=1}^n p_j/p^*$ oraz $\hat{T} = \lceil T/p^* \rceil$. Zdefiniujemy współczynniki kosztów

$$c_{jt} = \begin{cases} \infty & tp^* \leq a_i \\ \hat{c}_{jt} & a_i < tp^* \leq b_i \\ \infty & b_i < tp^* \end{cases} \quad \begin{matrix} x_i < j \leq y_i \\ i = 1, 2, \dots, n \\ t = 1, 2, \dots, \hat{T} \end{matrix} \quad (8.72)$$

gdzie

$$\hat{c}_{jt} = \begin{cases} v_i & tp^* \leq r_i \\ 0 & r_i < tp^* \leq d_i \\ w_i & d_i < tp^* \end{cases} \quad \begin{matrix} x_i < j \leq y_i \\ i = 1, 2, \dots, n \\ t = 0, 1, \dots, \hat{T}-1. \end{matrix} \quad (8.73)$$

W przypadku, gdy $\mathcal{R} = \emptyset$ rozwiązanie problemu $n|1|$ przerywać $|\sum h_i$ można otrzymać przez rozwiązanie następującego problemu transportowego [35]

$$\min_{x_{jt}} \sum_{j=1}^{\hat{n}} \sum_{t=1}^{\hat{T}} \hat{c}_{jt} x_{jt} \quad , \quad (8.74)$$

$$\sum_{j=1}^{\hat{n}} x_{jt} \leq 1 \quad t = 1, 2, \dots, \hat{T} \quad , \quad (8.75)$$

$$\sum_{t=1}^{\hat{T}} x_{jt} = 1 \quad j = 1, 2, \dots, \hat{n} \quad , \quad (8.76)$$

$$x_j \in \{0, 1\} \quad j = 1, 2, \dots, \hat{n} \quad , \quad t = 1, 2, \dots, \hat{T} \quad , \quad (8.77)$$

o wymiarze $(\hat{T} \times \hat{n})$. W celu rozwiązania problemu (8.74) - (8.77) można zastosować szczególnie efektywną metodę opisaną w [35].

Takie sformułowanie dolnego ograniczenia stwarza dodatkowe następujące problemy:

- (i) w przypadku, gdy $p^* = 1$ wymiar rozpatrywanego zagadnienia transportowego jest bardzo duży; jest praktycznie niemożliwym rozwiązywanie zbyt dużego problemu w każdym węźle drzewa rozwiązań,
- (ii) trudno jest zdefiniować współczynniki kosztu w przypadku ogólnych funkcji $h_i(t)$,
- (iii) uwzględnienie $\mathcal{R} \neq \emptyset$ zwiększa złożoność problemu (NP).

Rozwiązanie tych problemów jest możliwe poprzez zastosowanie odpowiednich relaksacji.

Relaksacja 8.1. /rys. 8.11/. Załóżmy, że $p^* = 1$. Niech p^{**} będzie liczbą całkowitą taką, że $p^* < p^{**} \leq \min_{1 \leq i \leq n} p_i$. Zdefiniujmy nowy problem szeregowania niepodzielnego z odpowiednimi wartościami

$$p_i' \triangleq \left[p_i / p^{**} \right] p^{**}$$

$$W(\bar{\lambda}) = \min_{x \in X} L(\bar{x}, \bar{\lambda}) + \sum_{\langle J'_i, J'_j \rangle \in \mathcal{R}} \lambda_{ij}, \quad (8.82)$$

$$X = \left\{ x \mid \sum_{j=1}^{\hat{n}} x_{jt} \leq 1, \quad t = 1, 2, \dots, \hat{T}, \quad \sum_{t=1}^{\hat{T}} x_{jt} = 1, \right. \\ \left. j = 1, 2, \dots, \hat{n} \right\}. \quad (8.83)$$

Można pokazać, że wartość $W(\bar{\lambda})$ jest dolnym ograniczeniem wartości funkcji celu problemu szeregowania $n|1|<|\sum h_i$ dla dowolnych wartości $\bar{\lambda} \geq 0$.

Najlepsza wartość dolnego ograniczenia jest określona przez zmienne dualne, które maksymalizują $W(\bar{\lambda})$, $\bar{\lambda} \geq 0$.

Zauważmy, że dla ustalonej wartości $\bar{\lambda}$ rozwiązania problemu (8.81) jest równoważne rozwiązaniu zagadnienia transportowego z odpowiednio określonymi współczynnikami kosztu.

Dolne ograniczenia LB2. Załóżmy w dalszym ciągu, że wybrany został odcinek czasu $[t_0, T]$ o tej własności, że $S_i^* + p_i \leq T$ $i = 1, 2, \dots, n$. Zwykle przyjmuje się $t_0 = 0$ / t_0 - dane/ oraz T - górne ograniczenie przedziału czasu zawierającego rozwiązanie optymalne, np. $T = \max_{1 \leq i \leq n} r_i + \sum_{i=1}^n p_i$.

Założmy dalej, że przedział $[t_0, T]$ został podzielony na rozłączne podprzedziały o długości jednostkowej $[t-1, t]$, $t = t_0 + 1, t_0 + 2, \dots, T$. Przyjętą jednostką czasu dla wyznaczania planu czynności może być minuta, godzina, dzień, miesiąc. Zdefiniujmy zbiór prac wykonywanych w przedziale $[t-1, t]$

$$G_t(\bar{S}) = \{ J_i \in J \mid t - p_i \leq S_i \leq t - 1 \} \quad (8.84)$$

oraz

$$g_t(\bar{S}) = |G_t(\bar{S})|, \quad (8.85)$$

liczbę prac wykonywanych w przedziale $[t-1, t]$. Wówczas ograniczenie na skończoną przepustowość maszyny można zapisać w formie $g_t(\bar{S}) \leq 1$, $t = t_0 + 1, t_0 + 2, \dots, T$ zaś problem szeregowania jako

$$\min_{\bar{S}} \sum_{i=1}^n h_i(S_i), \quad (8.86)$$

$$\begin{aligned} r_i^1 &\triangleq \left[r_i / p^{**} \right] p^{**} & g_i^1(t) &= g_i(t) \\ d_i^1 &\triangleq \left[d_i / p^{**} \right] p^{**} & f_i^1(t) &= f_i(t) \end{aligned} \quad i = 1, 2, \dots, n, \quad (8.78)$$

gdzie $[x]$ - oznacza część całkowitą x , zaś $\lceil x \rceil$ najmniejszą część całkowitą nie mniejszą niż x . Tak otrzymany problem jest relaksacją problemu $n | 1 | \sum h_i$ i wyznaczenie dolnego ograniczenia poprzez uszeregowanie podzielne wymaga rozwiązania zagadnienia transportowego o mniejszym wymiarze /rys. 8.11/. Właściwy wybór p^{**} jest uzależniony od wartości danych problemu.

Relaksacja 8.2. Ponieważ zdefiniowanie współczynników kosztu dla przypadku liniowego (8.72) - (8.73) nie jest trudne, więc przy określaniu współczynników kosztu dla przypadku ogólnego możemy skorzystać z odpowiedniej liniowej aproksymacji funkcji kosztu zadań.

Relaksacja 8.3. W przypadku, gdy $\mathcal{R} \neq \emptyset$ możemy wyznaczyć wartości $a_i = \sum_{j \in B_i} p_j$, $b_i = \max_{j \notin A_i} r_j + \sum_{j \in A_i} p_j$ $i = 1, 2, \dots, n$ oraz użyć je przy wyliczaniu c_{jt} ($j = 1, 2, \dots, \hat{n}$, $t = 1, 2, \dots, \hat{T}$). W przypadku wprowadzenia dodatkowych warunków ograniczających w postaci

$$\sum_{t=1}^{\hat{T}} tx_{jt} - \sum_{t=1}^{\hat{T}} tx_{it} - 1 \geq 0 \quad \langle J'_i, J'_j \rangle \in \mathcal{R}, \quad (8.79)$$

powoduje, że problem staje się NP zupełny. W tym przypadku rozwiązanie można otrzymać formułując odpowiednio problem dualny.

Zapiszmy warunek (8.79) w postaci

$$1 - \sum_{t=1}^{\hat{T}} t(x_{jt} - x_{it}) \leq 0 \quad \langle J'_i, J'_j \rangle \in \mathcal{R}. \quad (8.80)$$

zdefiniujmy funkcję Lagrange'a

$$L(\bar{x}, \bar{\lambda}) = \sum_{j=1}^{\hat{n}} \sum_{t=1}^{\hat{T}} c_{jt} x_{jt} - \sum_{\langle J'_i, J'_j \rangle \in \mathcal{R}} \lambda_{ij} \left(\sum_{t=1}^{\hat{T}} t (x_{jt} - x_{it}) \right) \quad (8.81)$$

i problem Lagrange'a

przy ograniczeniach

$$s_i + p_i \leq s_j \quad \langle J_i, J_j \rangle \in \mathcal{R}, \quad (8.87)$$

$$g_t(\bar{s}) \leq 1 \quad t = t_0+1, t_0+2, \dots, T \quad (8.88)$$

Niech $\mathcal{R}_1 \subseteq \mathcal{R}$ będzie relacją częściową taką, że graf relacji $\mathcal{R}_1$ jest drzewem /lasem/. Niech $\mathcal{R}_2 = \mathcal{R} - \mathcal{R}_1$. Określmy zbiór rozwiązań dopuszczalnych następująco

$$\mathcal{S} = \{ \bar{s} \mid a_i \leq s_i, s_i + p_i \leq b_i, i = 1, 2, \dots, n, s_i + p_i \leq s_j, \langle J_i, J_j \rangle \in \mathcal{R}_1 \}. \quad (8.89)$$

Wówczas problem (8.86) - (8.88) przyjmuje postać

$$\min_{\bar{s} \in \mathcal{S}} \sum_{i=1}^n h_i(s_i), \quad (8.90)$$

$$s_i + p_i \leq s_j \quad \langle J_i, J_j \rangle \in \mathcal{R}_2, \quad (8.91)$$

$$g_t(\bar{s}) \leq 1 \quad t = t_0+1, t_0+2, \dots, T. \quad (8.92)$$

Podział relacji $\mathcal{R}$ na $\mathcal{R}_1$ i $\mathcal{R}_2$ jest uwarunkowany istnieniem efektywnego algorytmu tylko w przypadku, gdy relacja poprzedzań spełnia warunek $|A_i| \leq 1, i = 1, 2, \dots, n$.

Ponieważ podział relacji $\mathcal{R}$ na $\mathcal{R}_1$ i $\mathcal{R}_2$ może być zrealizowany na wiele różnych sposobów więc pojawia się dodatkowy problem który z dokonanych podziałów będzie najlepszy. Fisher [31] proponuje, aby $\mathcal{R}_1$ wybierać w/g zasady zachowania najdłuższej drogi w grafie.

Wprowadzając zmienne dualne $\bar{u} \triangleq [u_{t_0+1}, u_{t_0+2}, \dots, u_T]$ związane z każdym ograniczeniem (8.92) oraz zmienne dualne λ_{ij} związane z każdym ograniczeniem (8.91) definiujemy funkcję Lagrange'a

$$\begin{aligned} L(\bar{s}, \bar{u}, \bar{\lambda}) &= \sum_{i=1}^n h_i(s_i) + \sum_{t=t_0+1}^T u_t g_t(\bar{s}) + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} (s_i - s_j) = \\ &= \sum_{i=1}^n \left[h_i(s_i) + \sum_{t=s_i+1}^{s_i+p_i} u_t + \sum_{J_j \in \Gamma_i(\mathcal{R}_2)} \lambda_{ij} s_i - \sum_{J_j \in \Gamma_i^{-1}(\mathcal{R}_2)} \lambda_{ji} s_i \right] \quad (8.93) \end{aligned}$$

i problem Lagrange'a

$$W(\bar{u}, \bar{\lambda}) = \min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \bar{u}, \bar{\lambda}) - \sum_{t=t_0+1}^T u_t + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} p_i \quad (8.94)$$

Własność 8.6.

$W(\bar{u}, \bar{\lambda})$ jest dolnym ograniczeniem wartości funkcji celu problemu szeregowania $n+1 \leq \sum h_i$ - dla dowolnych wartości $\bar{u} \geq 0$, $\bar{\lambda} \geq 0$.

Dowód

Niech $\bar{S}^* \in \mathcal{S}$ będzie optymalnym rozwiązaniem problemu (8.90)-(8.92). Zauważmy, że

$$\begin{aligned} W(\bar{u}, \bar{\lambda}) &= \min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \bar{u}, \bar{\lambda}) - \sum_{t=t_0+1}^T u_t + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} p_i \leq \\ &\leq L(\bar{S}^*, \bar{u}, \bar{\lambda}) - \sum_{t=t_0+1}^T u_t + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} p_i = \sum_{i=1}^n h_i(S_i^*) + \\ &+ \sum_{t=t_0+1}^T u_t (g_t(\bar{S}^*) - 1) + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} (S_i^* + p_i - S_j^*) \leq \sum_{i=1}^n h_i(S_i^*) \quad (8.95) \end{aligned}$$

bowiem $\sum_{t=t_0+1}^T u_t (g_t(\bar{S}^*) - 1) \leq 0$ oraz $\sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} (S_i^* + p_i - S_j^*) \leq 0$, co

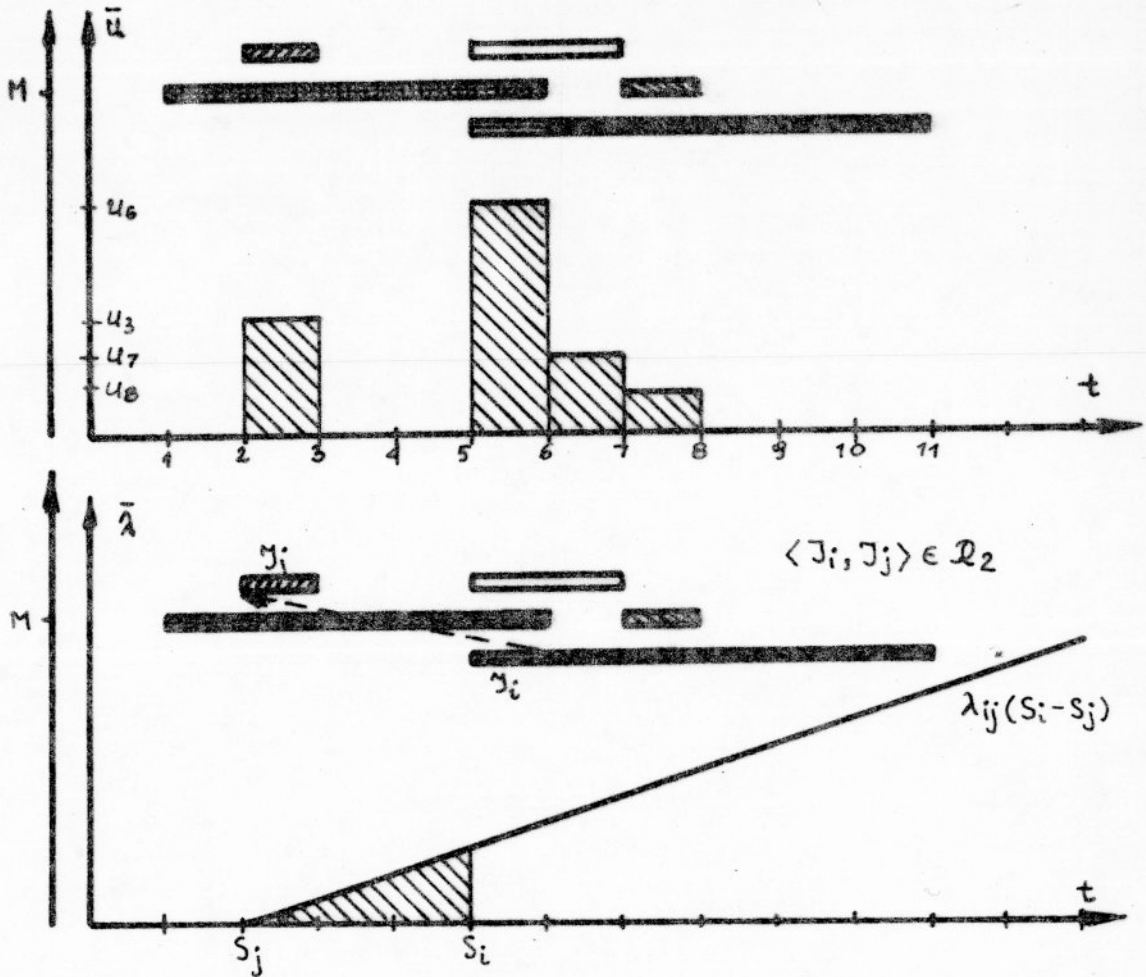
kończy dowód. ■

Najlepsza wartość dolnego ograniczenia jest określona przez zmienne dualne $\bar{u} \geq 0$, $\bar{\lambda} \geq 0$, które maksymalizują

$$\max_{\bar{u} \geq 0, \bar{\lambda} \geq 0} W(\bar{u}, \bar{\lambda}) \quad (8.96)$$

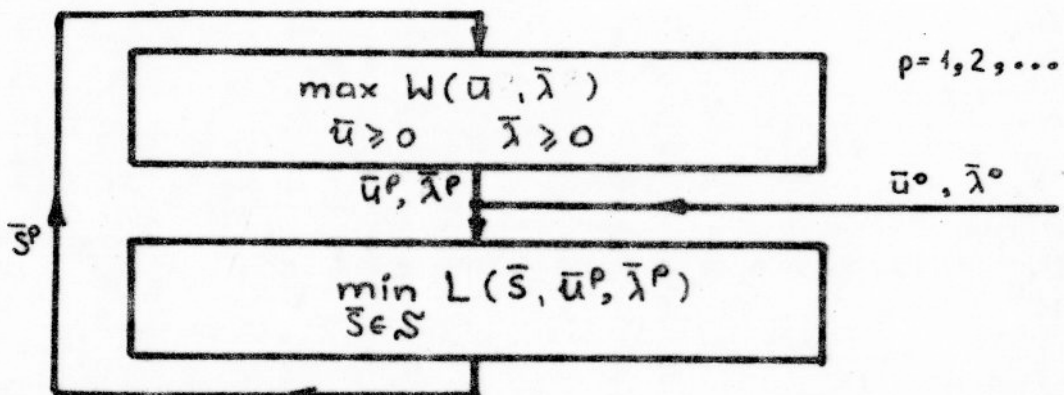
Zauważmy, że jeśli przyjmiemy u_t jako dodatkową opłatę za użycie maszyny w przedziale $[t-1, t]$ zaś λ_{ij} jako dodatkową opłatę spowodowaną wcześniejszym niż to wynika z relacji $\mathcal{R}_2$ rozpoczęciem zadania, to funkcja Lagrange'a przedstawia sobą sumę łącznych kosztów wykonywania zadań w przedziałach $[S_i, S_i + p_i]$, rys. 8.12.

W celu rozwiązania problemu dualnego zaproponowany został dwupoziomowy algorytm iteracyjny /rys. 8.13/. Najpierw określony zostanie algorytm rozwiązania problemu Lagrange'a $\min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \bar{u}, \bar{\lambda})$



Rys. 8.12. Graficzna interpretacja znaczenia zmiennych dualnych /zakreskowane pola odpowiadają dodatkowej opłacie za niespełnienie ograniczeń/

z ustaloną wartością $\bar{u}, \bar{\lambda}$. Następnie podany zostanie algorytm dla problemu dualnego $\max_{\bar{u} \geq 0, \bar{\lambda} \geq 0} W(\bar{u}, \bar{\lambda})$



Rys. 8.13. Dwupoziomowy algorytm problemu (8.90) - (8.92)

Niech $\bar{B}_i$ będzie zbiorem wszystkich poprzedników zadania J_i oraz $\hat{B}_i = \bar{B}_i \cup \{J_i\}$. Zauważmy, że

$$\begin{aligned} L(\bar{S}, \bar{u}, \bar{\lambda}) &= \sum_{i=1}^n (h_i(S_i) + \sum_{t=S_i+1}^{S_i+p_i} u_t + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} S_i - \sum_{\langle J_j, J_i \rangle \in \mathcal{R}_2} \lambda_{ji} S_i) = \\ &= \sum_{i=1}^n L_i(S_i, \bar{u}, \bar{\lambda}) \quad , \end{aligned} \quad (8.97)$$

gdzie

$$L_i(S_i, \bar{u}, \bar{\lambda}) = h_i(S_i) + \sum_{t=S_i+1}^{S_i+p_i} u_t + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} S_i - \sum_{\langle J_j, J_i \rangle \in \mathcal{R}_2} \lambda_{ji} S_i \quad (8.98)$$

Algorytm rozwiązania problemu (8.94) wyznacza rekurencyjnie optymalną wartość $K_i(t)$ częściowego problemu Lagrange'a zawierającego wszystkie zadania $J_j \in \hat{B}_i$ przy ograniczeniu, że czasy zakończenia tych zadań są nie większe niż t . Faza konstrukcji algorytmu bazuje na następującej własności, jeśli $S_i < t - p_i$ w optymalnym rozwiązaniu częściowego problemu Lagrange'a, to $K_i(t) = K_i(t-1)$; w przeciwnym przypadku, gdy $S_i = t - p_i$ to $K_i(t)$ jest równe sumie $L_i(S_i, \bar{u}, \bar{\lambda})$ oraz wartości funkcji celu pochodzących od zadań $J_j \in \bar{B}_i$.

Założmy, że zadania zostały ponumerowane tak, że $B_i \subset \{J_1, J_2, \dots, J_{i-1}\}$. Taka numeracja zawsze istnieje, gdy graf relacji $\mathcal{R}_1$ jest acykliczny.

Algorytm 8.2.

Algorytm wylicza wartość problemu Lagrange'a $\min_S L(\bar{S}, \bar{u}, \bar{\lambda})$ dla danych wartości $\bar{u}, \bar{\lambda}$.

krok 1 $i := \emptyset$,

krok 2 $i := i+1$

oblicz

$$K_i(t) = \begin{cases} \infty & t = t_0+1, t_0+2, \dots, a_i + p_i - 1 \\ \min[K_i(t-1), L_i(t-p_i, \bar{u}, \bar{\lambda}) + \sum_{J_j \in \bar{B}_i} K_j(t-p_i)] & t = a_i + p_i, \dots, b_i \\ K_i(b_i) & t = b_i+1, \dots, T \end{cases}$$

oraz

$$v_i(t) = \begin{cases} 0 & \text{jeśli } K_i(t) = K_i(t-1) \text{ lub } t = t_0, \\ 1 & \text{przeciwnie,} \end{cases}$$

krok 3 jeśli $i < n$ to przejdź do kroku 2 inaczej krok 4,

krok 4 wielkość $K(T) = \sum_{J_i \in \hat{\mathcal{R}}_1} K_i(T)$ gdzie

$\hat{\mathcal{R}}_1 = \{J_i \in J \mid \forall j \in J \langle J_i, J_j \rangle \notin \mathcal{R}_1\}$, jest optymalną wartością problemu Lagrange'a (8.94).

Dla znalezienia S_i^* , dla którego $L(\bar{S}^*, \bar{u}, \bar{\lambda}) = \min_{\bar{S} \in \bar{S}} L(\bar{S}, \bar{u}, \bar{\lambda})$ należy podstawić

$$S_i^* = t^* - p_i \quad \text{dla } J_i \in \hat{\mathcal{R}}_1, \quad (8.99)$$

gdzie t^* jest największą wartością dla której

$$v_i(t^*) = 1$$

Następnie dla pozostałych $i = n-1, n-2, \dots, 1$ przyjmujemy

$$S_i^* = \max t - p_i \quad (8.100)$$

przy ograniczeniach

$$t \leq S_j^* \quad J_j \in A_i, \quad (8.101)$$

$$v_i(t) = 1, \quad (8.102)$$

Każdy z tych problemów posiada rozwiązanie bowiem $v_i(a_i + p_i) = 1$ dla każdego $i = 1, 2, \dots, n$.

Własność 8.7.

Algorytm 8.2 posiada złożoność obliczeniową pseudowielomianową $O(n \cdot T)$. ■

Dowód

Kroki 1, 2, 4 wymagają co najwyżej $n(T - t_0)$ iteracji. Najbardziej złożona część kroku 2 wymaga $\sum_{\tau=t-p_i+1}^t u_\tau$ iteracji dla obliczenia

wartości $L_i(t - p_i, \bar{u}, \bar{\lambda})$ oraz $\sum_{J_j \in B_1} K_j(t - p_i)$ oraz $\sum_{J_j \in \Gamma_i(\mathcal{R}_1)} \lambda_{ij} t - \sum_{J_j \in \Gamma_i^{-1}(\mathcal{R}_1)} \lambda_{ij} t$

Zakładając, że znana jest wartość $\sum_{\tau=a_i+1}^{a_i+p_i} u_\tau$, to kolejne wartości

sum cząstkowych mogą być wyliczone rekurencyjnie

$$\sum_{\tau=t-p_i+1}^t u_{\tau} = \sum_{\tau=t-p_i}^{t-1} u_{\tau} + u_t - u_{t-p_i} \quad (8.103)$$

Stąd wniosek, że odpowiednia część kroku 2 wymaga $n(T-t_0)$ iteracji.

Ponieważ $|A_i| \leq 1$ dla $i = 1, 2, \dots, n$, $\sum_{i=1}^n |B_i| = \sum_{i=1}^n |A_i| \leq n$ więc wyliczenie $\sum_{J_j \in B_i} K_j(t-p_i)$ wymaga $n(T-t_0)$ iteracji. Dalej wyliczenie $\sum \lambda_{ij} - \sum \lambda_{ji}$ wymaga co najwyżej $n(n-1)/2$ iteracji i może być wykonane apriori. Stąd złożoność całego algorytmu $O(nT)$. ■

Rozpatrzmy teraz problem dualny

$$\max_{\bar{u}, \bar{\lambda}} W(\bar{u}, \bar{\lambda}), \quad (8.104)$$

$$\bar{u} \geq 0, \bar{\lambda} \geq 0. \quad (8.105)$$

Do rozwiązywania tego problemu zastosowana została metoda subgradientowa opisana w [50], [37]. Idea tej metody polega na generowaniu ciągu rozwiązań $\bar{u}^p, \bar{\lambda}^p$ w/g pewnego wzoru rekurencyjnego.

Niech $\bar{u}^0 = 0, \bar{\lambda}^0 = 0$. Ciąg wielkości $\bar{u}^p, \bar{\lambda}^p$ dla problemu (8.104) bez warunku (8.105) ma postać

$$u_t^{p+1} = u_t^p + E^p(g_t(S^p) - 1), t = t_0+1, t_0+2, \dots, T \quad (8.106)$$

$$\lambda_{ij}^{p+1} = \lambda_{ij}^p + E^p(s_i^p + p_i - s_j^p), \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad (8.107)$$

$$E^p = \gamma^p \frac{UB - W(\bar{u}_2^p, \bar{\lambda}^p)}{\sqrt{\sum_{t=t_0+1}^T (g_t(S^p) - 1)^2 + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} (s_i^p + p_i - s_j^p)^2}} \quad (8.108)$$

gdzie UB - górne ograniczenie, γ^p - skalar dobrany eksperymentalnie, zaś $\langle S_1^p, \dots, S_n^p \rangle$ - optymalne rozwiązanie problemu dolnego poziomu. Uwzględnienie warunku (8.105) wymaga zastosowania projekcji otrzymywanych wartości $\bar{u}^p, \bar{\lambda}^p$ na zbiór $U = \{\bar{u}, \bar{\lambda} \mid \bar{u} \geq 0, \bar{\lambda} \geq 0\}$, tzn. wyznaczenie takich wartości $\hat{u}^p, \hat{\lambda}^p$, że

$$\sum_{t=t_0+1}^T (u_t - \hat{u}_t)^2 + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} (\lambda_{ij} - \hat{\lambda}_{ij})^2 \rightarrow \min \quad (8.109)$$

$$\hat{u}_t \geq 0 \quad \hat{\lambda}_{ij} \geq 0 \quad (8.110)$$

Rozwiązanie problemu (8.109) – (8.110) można znaleźć z warunków Kuhna-Tackera i ma postać

$$\hat{u}_t = \max [0, u_t] \quad , \quad t = t_0+1, t_0+2, \dots, T \quad (8.111)$$

$$\hat{\lambda}_{ij} = \max [0, \lambda_{ij}] \quad , \quad \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad (8.112)$$

W konsekwencji otrzymujemy wzór procedury subgradientowej w postaci

$$\hat{u}_t^{p+1} = \max [0, \hat{u}_t^p + E^p (g_t(S^p) - 1)] \quad (8.113)$$

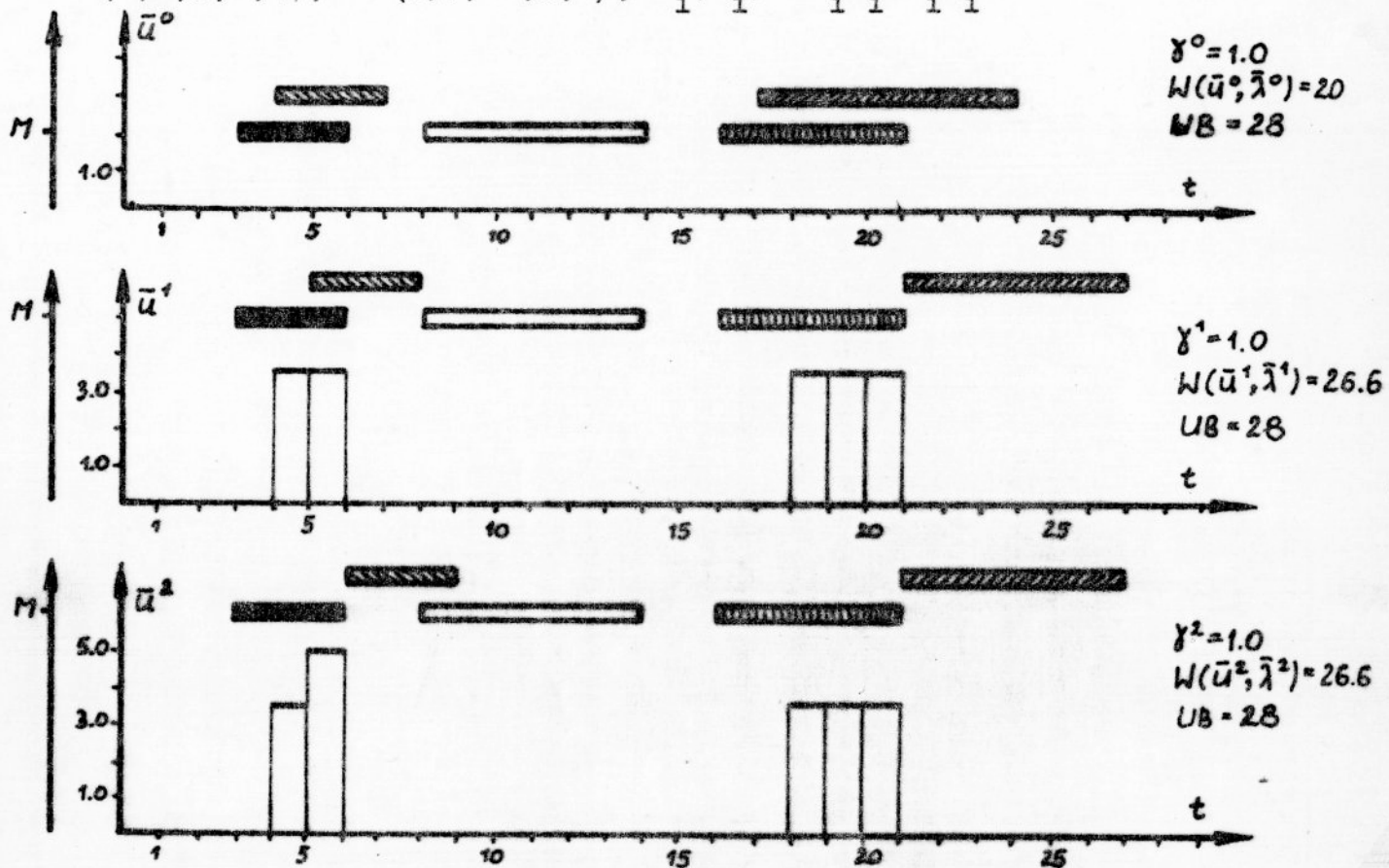
$$t = t_0+1, t_0+2, \dots, T,$$

$$\hat{\lambda}_{ij}^{p+1} = \max [0, \hat{\lambda}_{ij}^p + E^p (S_i^p + p_i - S_j^p)] \quad \langle J_i, J_j \rangle \in \mathcal{R} \quad (8.114)$$

gdzie E^p zdefiniowane jest wzorem (8.108).

Funkcjonowanie podanej procedury zilustrowano rysunkiem 8.14.

dla $n=5$, $r=\langle 3, 4, 11, 16, 18 \rangle$, $p=\langle 3, 3, 6, 5, 6 \rangle$, $d=\langle 5, 8, 14, 19, 22 \rangle$,
 $v=\langle 8, 2, 3, 6, 4 \rangle$, $w=\langle 3, 5, 10, 3, 1 \rangle$, $h_i(S_i) = v_i E_i + w_i T_i$.



Rys. 8.14. Kolejne iteracje algorytmu górnego poziomu
 $\max W(\bar{u}, \bar{\lambda})$, $\bar{u} \geq 0$, $\bar{\lambda} \geq 0$

Wartość górnego ograniczenia UB jest dana arbitralnie lub obliczana przy zastosowaniu heurystycznej reguły szeregowania zadań. Ponieważ rozsądnym wymaganiem jest, aby UB szacowało wartość kosztu z małym błędem, można zbadać szereg reguł heurystycznych dla wybrania reguły najlepszej do tego celu. Pewne algorytmy heurystyczne zaproponowano i opisano w rozdziale 8.9. Zastosowanie dowolnej reguły heurystycznej pozwala na bieżącą aktualizację UB.

Ciąg E^P powinien zapewniać zbieżność procedury subgradientowej lub co najmniej powinien gwarantować znaczny wzrost wartości $W(\bar{u}^P, \bar{\lambda}^P)$ w kolejnych krokach. Jednakże w przypadku zagadnień szeregowania procedura (8.113 - 8.114) nie gwarantuje zbieżności [50], [37] ciągu rozwiązań $W(\bar{u}^P, \bar{\lambda}^P)$ do rozwiązania optymalnego. Ponieważ jedynym elementem na wybór którego mamy wpływ jest skalar χ^P , więc odpowiedni jego wybór może zagwarantować szybki wzrost wartości dolnego ograniczenia. Wybór ten może być przeprowadzony jedynie na drodze eksperymentalnej i był przedmiotem badań opisanych w rozdziale 8.10.

Dolne ograniczenie LB3. Zauważmy, że złożoność obliczeniowa algorytmu dla LB1 jest stosunkowo duża, bowiem w praktyce $(T - t_0) \gg$ co powoduje, że silnie rośnie czas obliczeń problemu Lagrange'a. Trudności te mogą być eliminowane poprzez zastosowanie odpowiednich relaksacji.

Relaksacja 8.4. Niech t_k , $k = 1, \dots, s$ będzie ciągiem punktów takich, że $t_k \in \{t_0+1, t_0+2, \dots, T\}$, $t_0 < t_1 < t_2 < \dots < t_s = T$. Załóżmy, że w problemie (8.86)-(8.88) ograniczenia (8.88) zastępujemy ograniczeniami postaci

$$g_t(\bar{S}) \leq 1 \quad t = t_1, t_2, \dots, t_s. \quad (8.115)$$

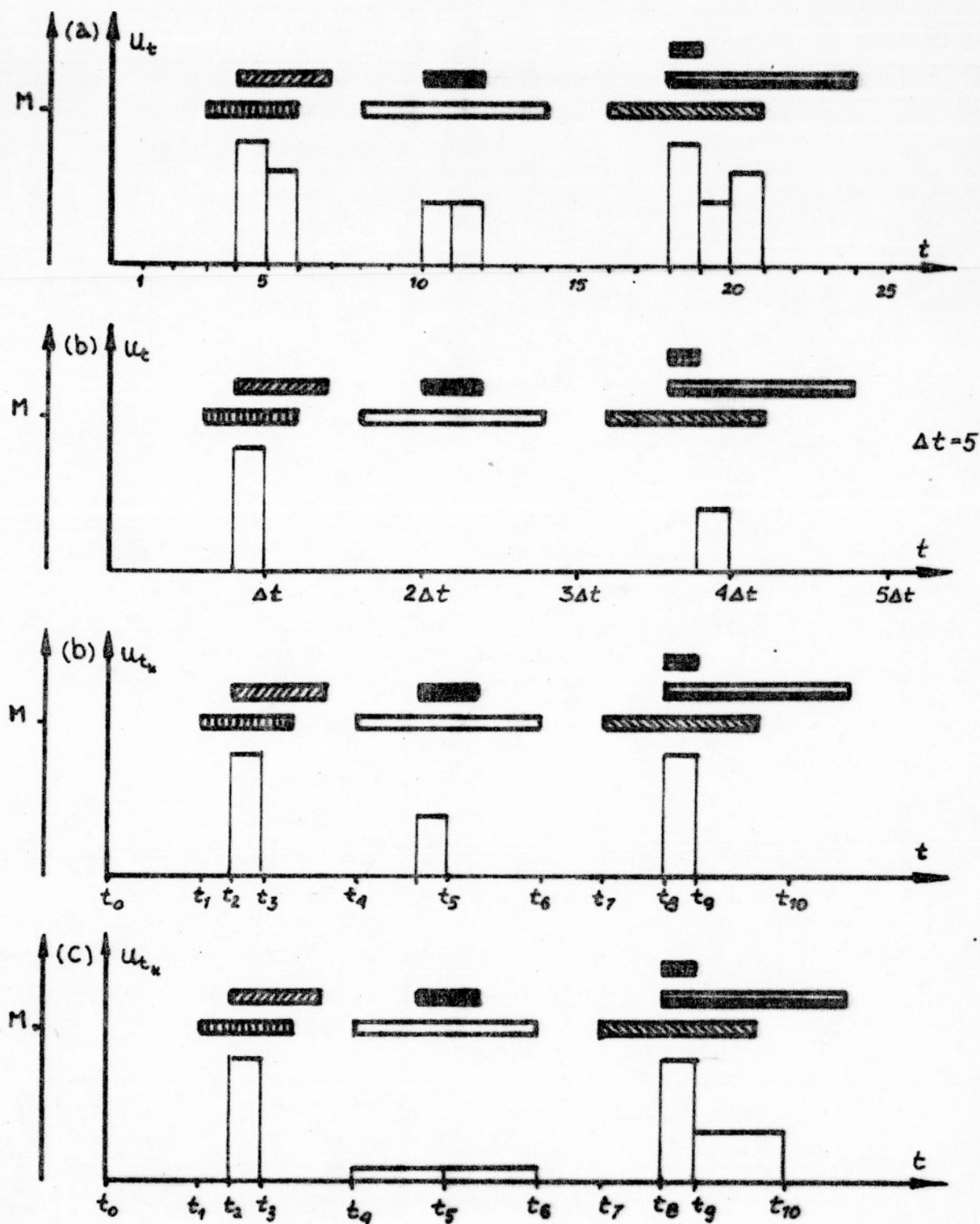
Odpowiedni problem dualny ma postać

$$\max_{\bar{u}, \bar{\lambda}} W(\bar{u}, \bar{\lambda}), \quad (8.116)$$

$$\bar{\lambda}, \bar{u} \geq 0, \quad (8.117)$$

$$u_t = 0 \quad t \notin \{t_1, t_2, \dots, t_s\}. \quad (8.118)$$

Rozpatrzmy początkowo szczególny rodzaj relaksacji zastosowany przez Fishera [31], tzn. $t_k = k \cdot \Delta t$, $0 < \Delta t \leq T$ prowadzący do algorytmu o złożoności nT/k , rys. 8.15. Zauważmy, że zachodzi



Rys. 8.15 Relaksacja 8.4 i 8.5 w dolnym ograniczeniu LB3
 /a/ dolne ograniczenie LB2, /b/ relaksacja 8.4 ,
 /b/ relaksacja 8.5

$$\sum_{\tau=t-p_i+1}^t u_{\tau} = \sum_{\tau=t-p_i}^{t-1} u_{\tau} - u_{t-p_i} + u_t = \sum_{\tau=t-p_i}^{t-1} u_{\tau} + u_t \geq \sum_{\tau=t-p_i}^{t-1} u_{\tau} \quad (8.119)$$

dla $t \neq t_k$ ponieważ $u_{t-p_i} = 0$ oraz $u_t \geq 0$. Dalej

$$K_i(t-1) \leq h_i(t-1) + \sum_{\tau=t-p_i}^{t-1} u_{\tau} \leq h_i(t-1) + \sum_{\tau=t-p_i+1}^t u_{\tau} \leq$$

$$\leq h_i(t) + \sum_{\tau=t-p_i+1}^t u_{\tau} = K_i(t), \quad t \neq t_1, t_2, \dots, t_s, \quad i = 1, 2, \dots, n \quad (8.120)$$

oraz nierówność ostatnią można zapisać tylko w przypadku, gdy $h_i(t-1) \leq h_i(t)$, czyli gdy funkcja jest niemalejąca. Ponieważ w ogólnym przypadku nie jest to spełnione, więc rozwiązanie problemu Lagrange'a będzie wymagało innego postępowania oraz większej niż nT/k liczby iteracji.

W ogólnym przypadku dla dowolnego ciągu $t_1, t_2, \dots, t_s$ rozwiązanie problemu Lagrange'a może być znalezione /przy założeniu $\mathcal{R}_1 = \emptyset$ / z następującego wzoru

$$\begin{aligned} \min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \bar{u}, \bar{\lambda}) &= \sum_{i=1}^n \min_{S_i \in \mathcal{S}} L_i(S_i, \bar{u}, \bar{\lambda}) = \\ &= \sum_{i=1}^n \min_{\tau \in \Psi_i} L_i(\tau, \bar{u}, \bar{\lambda}), \end{aligned} \quad (8.121)$$

gdzie

$$\begin{aligned} \Psi_i &= \{t_0, t_1, \dots, t_m\} \cup \{t_0+1, t_1+1, \dots, t_{s-1}+1\} \cup \{t_1-p_i, t_2-p_i, \dots, \\ &\quad t_{s-1}-p_i\} \cup \{t_1+1-p_i, t_2+1-p_i, \dots, t_{s-1}+1-p_i\} \cup \\ &\quad \cup \{a_i\} \cup \{b_i-p_i\} \cup \{r_i\} \cup \{d_i-p_i\} \quad i=1, 2, \dots, n, \end{aligned} \quad (8.122)$$

$$\Psi_i = \{t \in \Psi_i \mid a_i \leq t \leq b_i-p_i\} \quad i=1, 2, \dots, n. \quad (8.123)$$

Wyznaczenie rozwiązania optymalnego problemu (8.121) wymaga $(4 \cdot s + 4) \cdot n$ iteracji

Relaksacja 8.5. Niech $t_k, k = 1, \dots, s$ będzie ciągiem punktów takich, że $t_k \in \{t_0+1, t_0+2, \dots, T\}$, $t_0 < t_1 < t_2 < \dots < t_s = T$. Jeśli w problemie (8.104) - (8.105) spełniony jest dodatkowy warunek $u_t = u_{t_k}, t = t_{k-1}+1, t_{k-1}+2, \dots, t_k$, $k = 1, 2, \dots, s$ /rys. 8.14/, to rozwiązanie problemu Lagrange'a może być znalezione/przy założeniu, że $\mathcal{R}_1 = \emptyset$ /w/g następującego wzoru

$$\begin{aligned} \min_{\bar{S} \in \bar{S}} L(\bar{S}, \bar{u}, \bar{\lambda}) &= \sum_{i=1}^n \min_{S_i \in \bar{S}} L_i(S_i, \bar{u}, \bar{\lambda}) = \\ &= \sum_{i=1}^n \min_{\tau \in \Phi_i} L_i(\tau, \bar{u}, \bar{\lambda}), \end{aligned} \quad (8.124)$$

gdzie

$$\begin{aligned} \varphi_i &= \{t_0, t_1, \dots, t_s\} \cup \{t_1-p_i, t_2-p_i, \dots, t_s-p_i\} \cup \{b_i-p_i\} \cup \\ &\quad \{a_i\} \cup \{r_i\} \cup \{d_i - p_i\}, \end{aligned} \quad (8.125)$$

$$\Phi_i = \{t \in \varphi_i \mid a_i \leq t \leq b_i + p_j\}. \quad (8.126)$$

Złożoność obliczeniowa metody wyznaczania wartości (8.124) jest $O(ns)$. W szczególności, gdy $s \sim n$, to złożoność jest wielomianowa rzędu $O(n^2)$.

Rozpatrzmy teraz problem dualny z dodatkowym warunkiem ograniczającym w postaci

$$\max_{\bar{u}, \bar{\lambda}} W(\bar{u}, \bar{\lambda}), \quad (8.127)$$

$$\bar{u} \geq 0, \quad \bar{\lambda} \geq 0, \quad (8.128)$$

$$u_t = u_{t_k} \quad t = t_{k-1}+1, t_{k-1}+2, \dots, t_k, \quad k = 1, 2, \dots, s \quad (8.129)$$

Ze względu na warunek (8.129) można zredukować liczbę zmiennych u_t z $T-t_0$ do s . Oznaczmy nowe zmienne przez $\hat{u}_t$. Załóżmy w dalszym ciągu, że do rozwiązania stosujemy metodę subgradientową [50]. Wartości $\hat{u}_{t_k}$ możemy otrzymać poprzez projekcję wartości $\bar{u}^p, \bar{\lambda}^p$ na zbiór $U = \{\langle \bar{u}, \bar{\lambda} \rangle \mid \bar{u} \geq 0, \bar{\lambda} \geq 0, u_t = u_{t_k}, t = t_{k-1}+1, t_{k-1}+2, \dots, t_k, k = 1, 2, \dots, s\}$, tzn.

$$\sum_{k=1}^s \sum_{t=t_{k-1}+1}^{t_k} (u_t - \hat{u}_{t_k})^2 + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} (\lambda_{ij} - \hat{\lambda}_{ij})^2 \rightarrow \min \quad (8.130)$$

$$\hat{u}_{t_k} \geq 0, \quad k = 1, 2, \dots, m, \quad (8.131)$$

$$\hat{\lambda}_{ij} \geq 0, \quad \langle J_i, J_j \rangle \in \mathcal{R}_2. \quad (8.132)$$

Rozwiązanie problemu (8.130) - (8.132) można znaleźć z warunków Kuhna-Tackera i ma ono postać

$$\hat{u}_{t_k} = \max \left[0, \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} u_t \right], \quad (8.133)$$

$$\hat{\lambda}_{ij} = \max [0, \lambda_{ij}]. \quad (8.134)$$

Podstawiając wzór (8.113) do uzyskanego wyniku otrzymujemy

$$\begin{aligned} \hat{u}_{t_k}^{p+1} &= \max \left[0, \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} u_t^{p+1} \right] = \max \left[0, \frac{1}{t_k - t_{k-1}} \right. \\ &\quad \left. \sum_{t=t_{k-1}+1}^{t_k} (u_t^p + E^p(g_t(S^p) - 1)) \right] = \max \left[0, \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} u_t^p + \right. \\ &\quad \left. + \frac{1}{t_k - t_{k-1}} E^p \sum_{t=t_{k-1}+1}^{t_k} (g_t(S^p) - 1) \right] = \max \left[0, \hat{u}_{t_k}^p + E^p \frac{1}{t_k - t_{k-1}} \right. \\ &\quad \left. \sum_{t=t_{k-1}+1}^{t_k} (g_t(S^p) - 1) \right] = \max \left[0, \hat{u}_{t_k}^p + E^p(\hat{g}_{t_k}(S^p) - 1) \right] \\ &\quad k = 1, 2, \dots, s, \end{aligned} \quad (8.135)$$

gdzie

$$\hat{g}_{t_k}(S^p) = \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} g_t(S^p). \quad (8.136)$$

Stąd otrzymujemy, że wartość kary w przedziale $(t_{k-1}, t_k]$ jest równa średniej wartości kar w tym przedziale /lub zero/. Odpowiedni wzór procedury subgradientowej ma postać

$$\hat{u}_{t_k}^{p+1} = \max [0, \hat{u}_{t_k}^p + E^p(\hat{g}_{t_k}(S^p) - 1)] \quad k = 1, 2, \dots, s \quad (8.137)$$

gdzie $\hat{g}_{t_k}(S^p)$ - średnia wartość ograniczeń $g_t(S^p)$ w przedziale $(t_{k-1}, t_k]$ oraz

$$\hat{\lambda}_{ij}^{p+1} = \max [0, \hat{\lambda}_{ij}^p + E^p(S_i^p + p_i - S_j^p)] \quad \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad (8.138)$$

Rozpatrzmy teraz problem pierwotny (8.86) - (8.88) w zrelaksowanej postaci

$$\min_{\bar{S}} \sum_{i=1}^n h_i(S_i) \quad , \quad (8.138)$$

$$\hat{g}_{t_k}(\bar{S}) \leq 1 \quad , \quad k = 1, 2, \dots, s \quad (8.139)$$

$$S_i + p_i \leq S_j \quad \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad . \quad (8.140)$$

Funkcja Lagrange'a dla tego problemu ma postać

$$L(\bar{S}, \hat{u}, \hat{\lambda}) = \sum_{i=1}^n h_i(S_i) + \sum_{k=1}^s \hat{u}_{t_k} \hat{g}_{t_k}(\bar{S}) + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \hat{\lambda}_{ij} (S_i - S_j) \quad (8.141)$$

zaś problem Lagrange'a

$$W(\hat{u}, \hat{\lambda}) = \min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \hat{u}, \hat{\lambda}) = \sum_{k=1}^s \hat{u}_{t_k} + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \hat{\lambda}_{ij} p_i \quad (8.142)$$

Wzór procedury subgradientowej dla tego przypadku ma postać

$$\hat{u}_{t_k}^{p+1} = \max [0, \hat{u}_{t_k}^p + \hat{E}^p(\hat{g}_{t_k}(S^p) - 1)] \quad k = 1, 2, \dots, s \quad (8.143)$$

$$\hat{\lambda}_{ij}^{p+1} = \max [0, \hat{\lambda}_{ij}^p + \hat{E}^p(S_i^p + p_i - S_j^p)] \quad \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad (8.144)$$

gdzie

$$\hat{E}^p = \gamma^p \frac{UB - W(\hat{u}^p, \hat{\lambda}^p)}{\sqrt{\sum_{k=1}^s (\hat{g}_{t_k}(S^p) - 1)^2 + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} (S_i^p + p_i - S_j^p)^2}} \quad (8.145)$$

Jak łatwo zauważyć, obie relaksacje /problemu pierwotnego jak i dualnego/ prowadzą do zagadnienia o tej samej postaci, jednakże wzory (8.137) - (8.138), (8.108) oraz (8.143) - (8.145) określające rekurencyjnie rozwiązanie w metodzie subgradientowej różnią się między sobą postacią współczynnika $E^p \neq \hat{E}^p$. Wybór E^p ma wpływ na szybkość wzrostu wartości dolnego ograniczenia i powinien być przedmiotem badań eksperymentalnych.

Oczywiście o tym na ile jest silna relaksacja problemu (8.86), (8.87), (8.115) w odniesieniu do (8.86) - (8.88) decyduje wybór wielkości t_k , $k = 1, 2, \dots, s$.

Relaksacja 8.6. Niech t_k , $k = 1, 2, \dots, s$ będzie ciągiem liczb /niekoniecznie całkowitych/ spełniających warunek $t_0 < t_1 < t_2 < \dots < t_s = T$. Zdefiniujemy zbiór

$$G(t, S) \triangleq \{J_i \in J \mid t - p_i \leq S_i < t\} \quad (8.146)$$

oraz funkcję

$$g(t, \bar{S}) \triangleq |G(t, \bar{S})| \quad (8.147)$$

$$g_{t_k}(\bar{S}) \triangleq \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} g(\tau, \bar{S}) d\tau \quad (8.148)$$

Rozważmy problem analogiczny do (8.138) - (8.148) w postaci

$$\min_{\bar{S}} \sum_{i=1}^n h_i(S_i) \quad , \quad (8.149)$$

przy ograniczeniach

$$g_{t_k}(\bar{S}) \leq 1 \quad k = 1, 2, 3, \dots, s \quad , \quad (8.150)$$

$$S_i + p_i \leq S_j \quad \langle J_i, J_j \rangle \in \mathcal{R}_2 \quad . \quad (8.151)$$

Funkcja Lagrange'a oraz problem Lagrange'a mają w tym przypadku postać

$$L(\bar{S}, \bar{u}, \bar{\lambda}) = \sum_{i=1}^n \left[h_i(S_i) + \int_{S_i}^{S_i + p_i} u(\tau) d\tau \right] + \sum_{\langle J_i, J_j \rangle \in \mathcal{R}_2} \lambda_{ij} (S_i + p_i - S_j) \quad (8.152)$$

gdzie $u(t)$ jest funkcją przedziałami stałą, tzn.

$$u(t) = u_{t_k} \quad t \in (t_{k-1}, t_k] \quad , \quad k = 1, 2, \dots, s, \quad u(t_0) = 0 \quad .$$

$$W(\bar{u}, \bar{\lambda}) = \min_{\bar{S} \in \bar{S}} L(\bar{S}, \bar{u}, \bar{\lambda}) - \int_{t_0}^s u(\tau) d\tau + \sum_{\langle J_i, J_j \rangle \in \mathcal{L}_2} \lambda_{ij} p_i \quad . \quad (8.153)$$

Jak z powyższego widać, odrzucenie założenia o całkowitoliczbowości zmiennych S_i oraz parametrów p_i, t_k, r_i, d_i , itd. nie zmienia istotnie problemu, powoduje tylko zwiększenie pracochłonności obliczeniowej odpowiednich algorytmów.

8.7. Zasada podziału

Celem zasady podziału w π jest wskazanie zadania do przesunięcia i wygenerowania następnika β w drzewie H . Będziemy przesuwac zadania ze zbioru kandydatów w π .

$$E_{\pi}^s \triangleq \{ J_{\pi(i)} \in J \mid i_{j-1}+1 \leq i \leq s \leq i_j-1, \\ \langle J_{\pi(i)}, J_{\pi(h)} \rangle \notin \mathcal{R}_{\pi}, h=i+1, \dots, s \} \quad (8.154)$$

gdzie E_{π}^s - zbiór kandydatów do przesunięcia na pozycję s -tą w $/s = 1, 2, \dots, n/$ oraz $\mathcal{R}_{\pi}$ - relacje poprzedzania zadań ustalono dla π .

Jest zrozumiałe, że będziemy dążyli do wyboru takiego zadania, którego przesunięcie wygeneruje następnik o możliwie najmniejszym koszcie, lub którego przesunięcie jest najbardziej perspektywiczne /posiada najmniejsze dolne ograniczenie/. Dla oceny kandydatów wprowadzona została wielkość $LB'(i, s)$

$$LB'(i, s) = LB_{\pi} - L_{\pi(i)}^*(\bar{u}, \bar{\lambda}) + L_{\pi(i)}'^*(\bar{u}, \bar{\lambda}) \quad , \quad (8.155)$$

gdzie

$$L_i^*(\bar{u}, \bar{\lambda}) = \min_{a_i+p_i \leq S_i+p_i \leq b_i} L_i(S_i, \bar{u}, \bar{\lambda}) \quad , \quad (8.156)$$

$$L_i'^*(\bar{u}, \bar{\lambda}) = \min_{a_i'+p_i \leq S_i+p_i \leq b_i'} L_i(S_i, \bar{u}, \bar{\lambda}) \quad , \quad (8.157)$$

zaś a_i', b_i' są odpowiednio najwcześniejszym momentem rozpoczęcia i najpóźniejszym momentem zakończenia zadania $J_{\pi(i)}$ w permutacji β

otrzymanej z permutacji π przez przesunięcie zadania $J_{\pi(i)}$ na pozycję s -tą w π . Wielkości $\bar{u}, \bar{\lambda}$ są wartościami zmiennych dualnych, dla których wyliczano wartość LB.

Własność 8.8.

Dla każdej permutacji π niech β będzie permutacją otrzymaną z π przez przesunięcie pewnej zadania z pozycji i -tej na s -tą w π ; wtedy otrzymujemy

$$LB_{\beta} \geq LB'(i, s) \geq LB_{\pi} \quad (8.158)$$

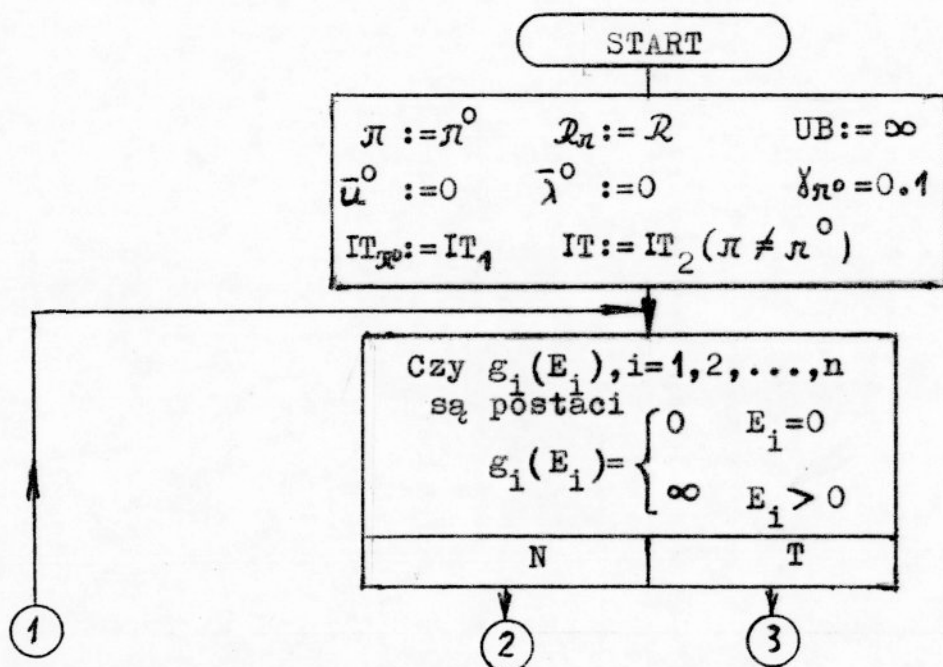
gdzie LB_{β} oraz LB_{π} są wartościami dolnych ograniczeń w węźle β i π odpowiednio. ■

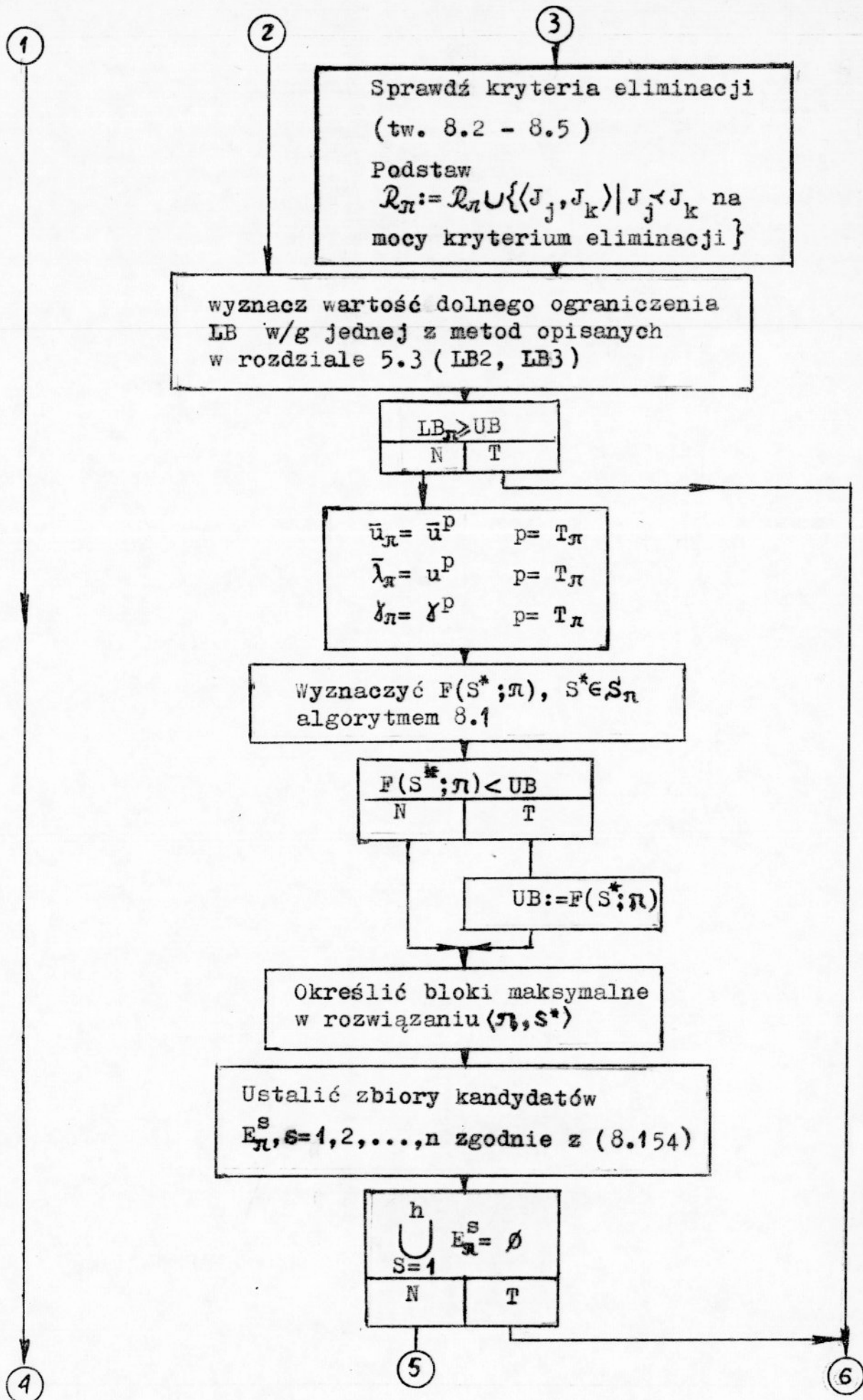
Dowód

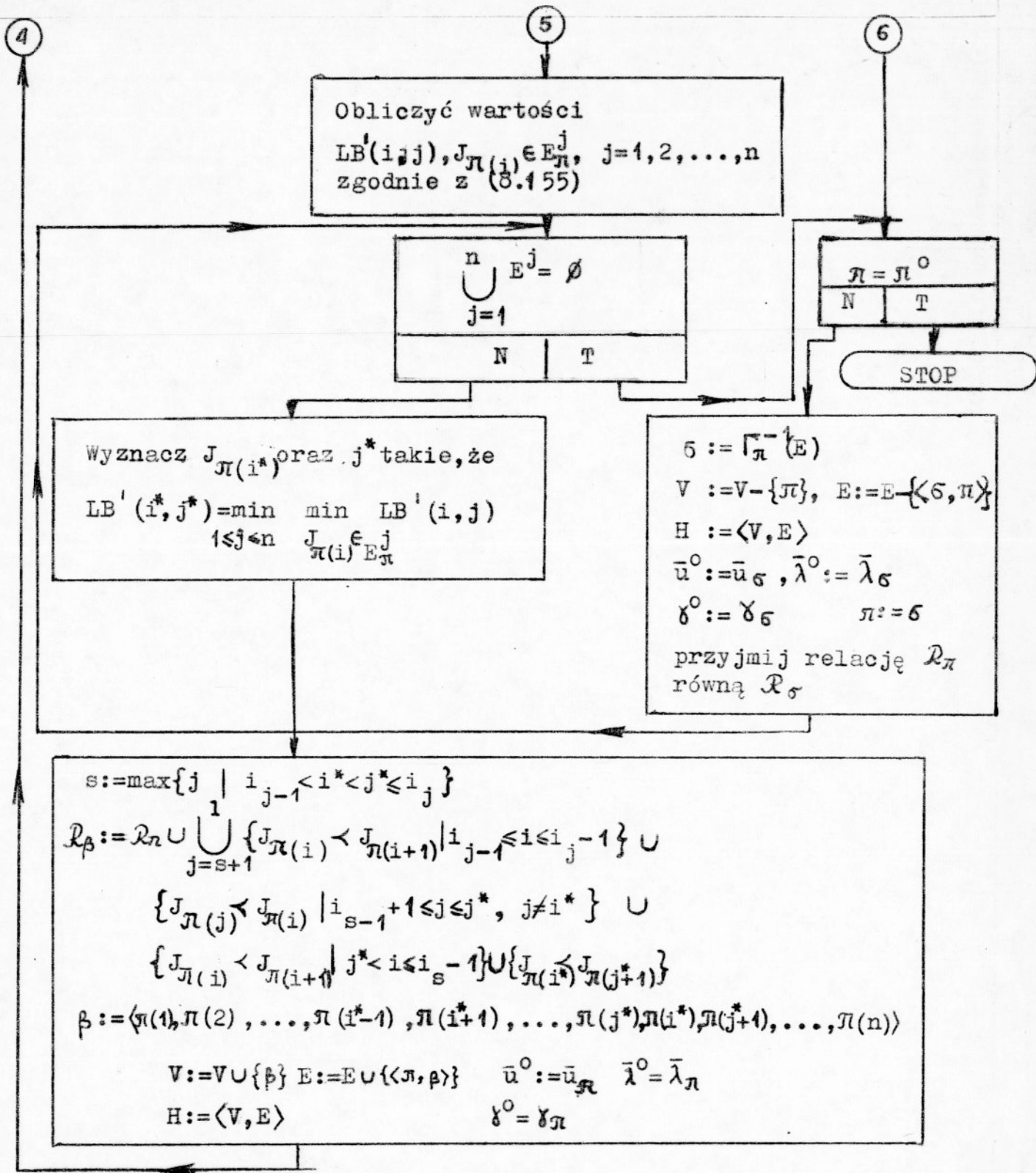
Zauważmy, że $\mathcal{R}_{\pi} \subseteq \mathcal{R}_{\beta}$ oraz $a_j \leq a_j^i, b_j^i \leq b_j \quad j = 1, 2, \dots, n$.
konsekwencji $L_i^*(\bar{u}, \bar{\lambda}) \leq L_i^*(\bar{u}, \bar{\lambda}^i)$ oraz wzór (8.158) jest spełniony w sposób oczywisty. ■

8.8. Algorytm

Niech π^0 będzie permutacją początkową, $\mathcal{R}$ - relacją poprzedzania w zbiorze zadań, γ^0 - skalar, IT_1, IT_2 - liczbami iteracji testu /metody subgradientowej/. Niech $\pi, \mathcal{R}_{\pi}, \bar{u}_{\pi}, \bar{\lambda}_{\pi}, \gamma_{\pi}$ będą odpowiednio permutacją, relacją, wartościami zmiennych dualnych oraz skalar w π -tej iteracji algorytmu. Oznaczmy przez LB_{π} wartość dolnego ograniczenia w π -tej iteracji algorytmu.

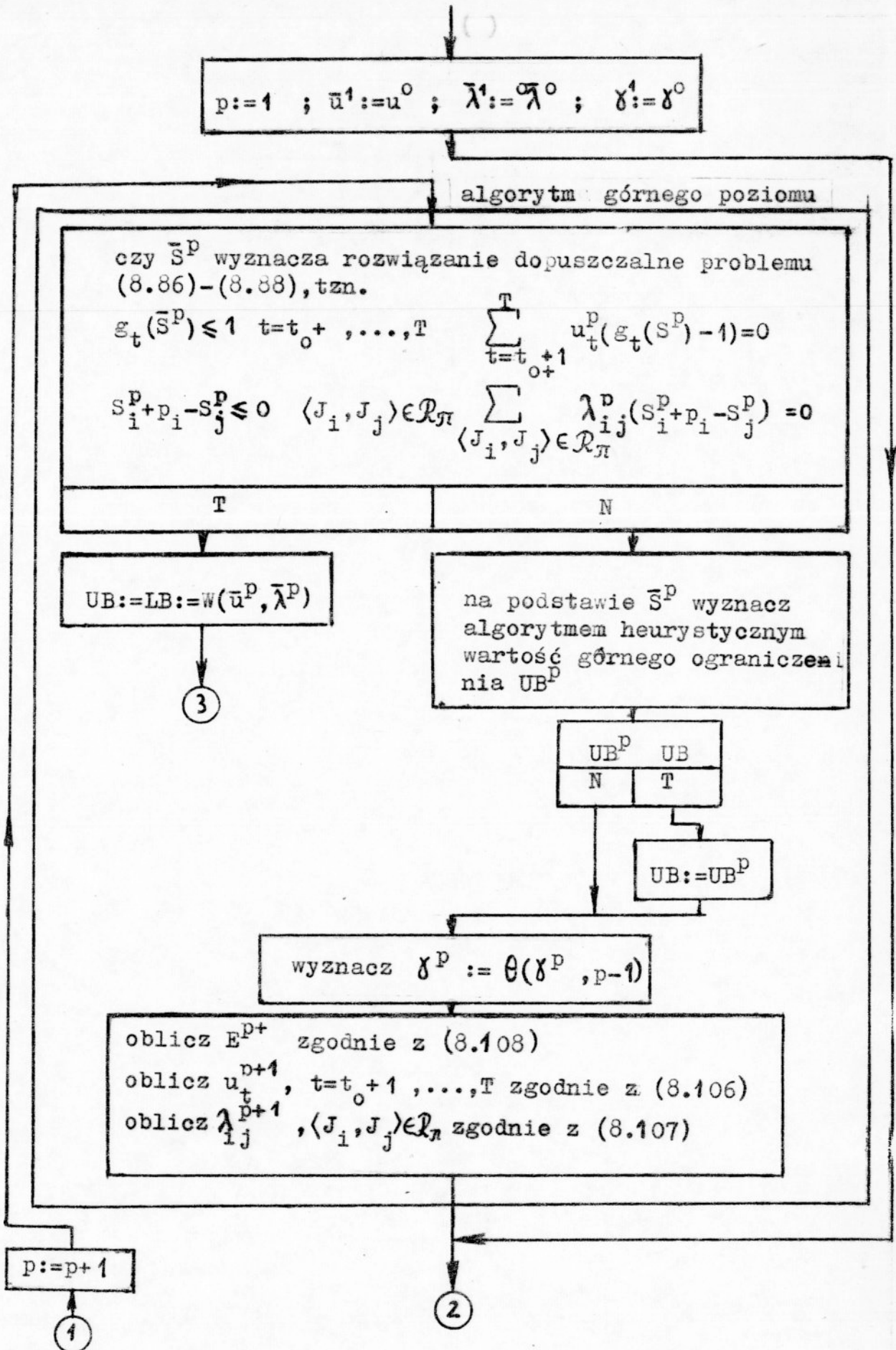


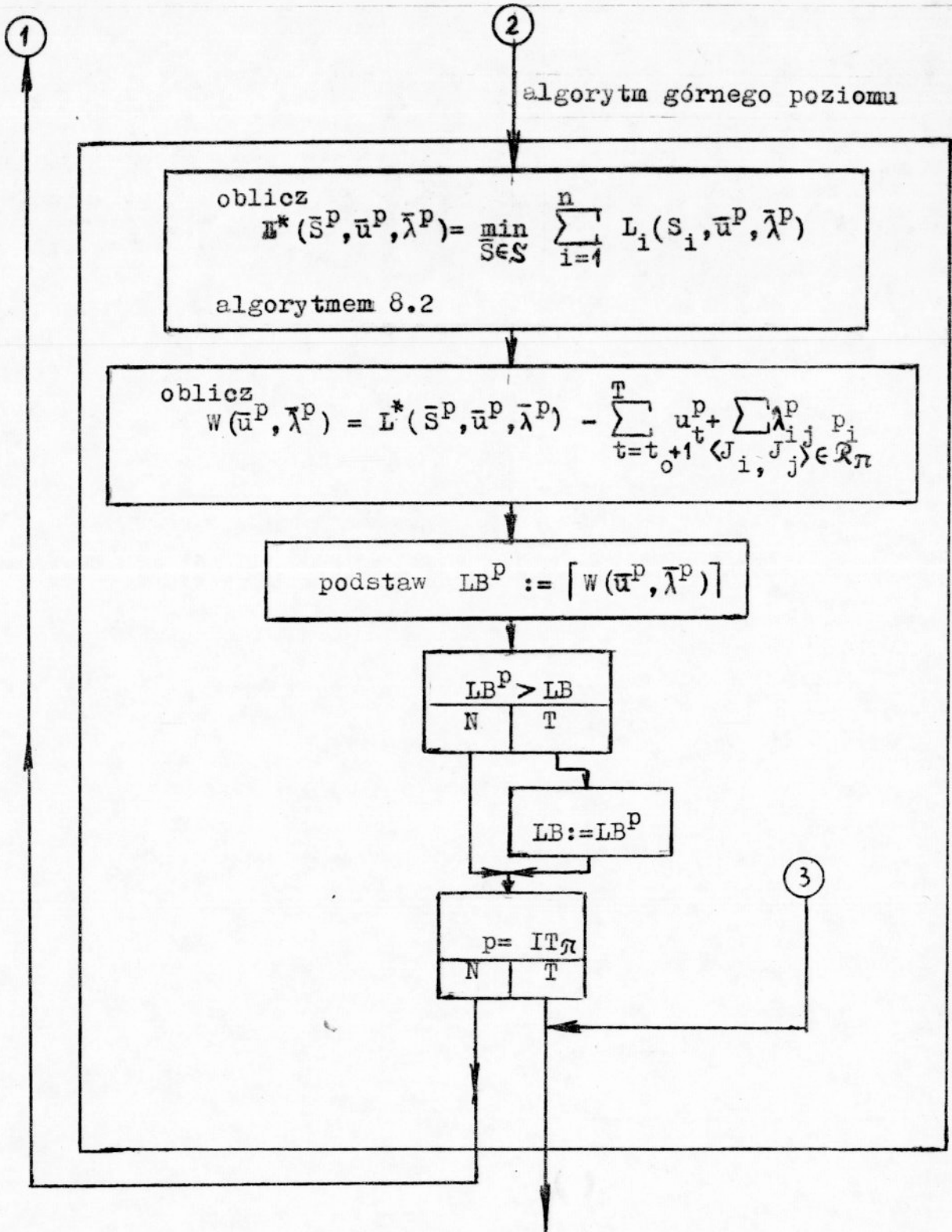




Rys. 8.16. Schemat blokowy algorytmu problemu $n | 1 | \sum h_i$.

Na rys. 8.17 przedstawiono szczegółowy schemat blokowy algorytmu wyznaczania dolnego ograniczenia





Rys. 8.17. Szczegółowy schemat blokowy dwupoziomowego algorytmu wyznaczania dolnego ograniczenia LB_π /na przykładzie LB_2 /

8.9. Realizacja programowa algorytmu. Przykłady testujące.

Algorytm zagadnienia $n \parallel \sum h_i$ opisany w rozdziale 8.8 został zaprogramowany w języku ALGOL 1900 i był testowany na maszynie ODRA 1305. Implementacja została zrealizowana przy założeniu liniowych funkcji kar, tzn. $g_i(E_i) = v_i E_i$, $f_i(T_i) = w_i T_i$, $i = 1, 2, \dots, n$. Realizacja algorytmu opiera się na programowo zorganizowanym stosie, w którym przechowywane są wszystkie wielkości. Dzięki przechowywaniu informacji w pamięci operacyjnej maksymalnie przyspieszono czas działania algorytmu. W celu zapamiętania relacji $\mathcal{R}_\pi$ /pamiętanej w każdej iteracji algorytmu/ zastosowano przedstawienie relacji w postaci elementów 0-1, które następnie są zapisywane w postaci binarnej na jednym słowie maszynowym /uzyskano dzięki temu ok. 23-krotną oszczędność pamięci/. W programie przewidziano dodatkowo możliwość przerywania obliczeń i odczytania najlepszego znalezionego do tej pory rozwiązania. Zrealizowano, także możliwość szukania rozwiązania ε -optymalnego, gdzie ε - zadany błąd przybliżenia $(UB - LB_{\pi^0})/LB_{\pi^0} \leq \varepsilon$.

W algorytmie zastosowano dwie reguły heurystyczne: do wyznaczania permutacji początkowej π^0 oraz do wyznaczenia górnego ograniczenia dla testu LB2 (LB3).

Algorytm wyznaczania π^0 był następujący:

- (i) wyznaczyć optymalne momenty rozpoczęcia każdego z zadań, przy założeniu, że maszyna ma nieograniczoną przepustowość, tzn. wyznaczyć

$$S_i^* = \min_{t_0 \leq S_i \leq T} h_i(S_i), \quad i = 1, 2, \dots, n$$

- (ii) uszeregować zadania w/g niemalejących wartości S_i^* .

Algorytm wyznaczania górnego ograniczenia UB w teście LB2

(LB3) był następujący:

- (i) wyznaczyć permutację $\hat{\pi}$ poprzez uszeregowanie zadań zgodnie z niemalejącymi czasami $C_i^* = S_i^* + p_i$, gdzie S_i^* - momenty rozpoczęcia zadań wyznaczono przez rozwiązanie problemu Lagrange'a (8.20) - (8.23).

- (ii) przyjąć $UB := \min_{\bar{S} \in S_{\hat{\pi}}} F(S; \hat{\pi})$.

W programie wykorzystano test LB3 z relaksacją 8.5. Liczba oraz wartości punktów t_k są zadawane z zewnątrz.

Badania i eksperymenty obliczeniowe zrealizowano dla ciągu losowo generowanych przykładów. Przyjęto następującą zasadę generacji danych do problemów.

- v_i, w_i ($i=1, \dots, n$) - liczby losowe o rozkładzie równomiernym z przedziału $[4.5, 15.5]$,
- p_i ($i = 1, \dots, n$) - liczby losowe o rozkładzie równomiernym z przedziału $[1, 100]$,
- r_i ($i = 1, \dots, n$) - liczby losowe o rozkładzie równomiernym z przedziału $\left[\sum_{i=1}^n p_i (1 - \tau - e_r/2), \sum_{i=1}^n p_i (1 - \tau_r + e_r/2) \right]$,
- d_i ($i = 1, \dots, n$) - liczby losowe o rozkładzie równomiernym z przedziału $\left[\left(\sum_{i=1}^n p_i + \frac{1}{n} \sum_{i=1}^n r_i \right) (1 - \tau_d - e_d/2), \left(\sum_{i=1}^n p_i + \frac{1}{n} \sum_{i=1}^n r_i \right) (1 - \tau_d + e_d/2) \right]$.

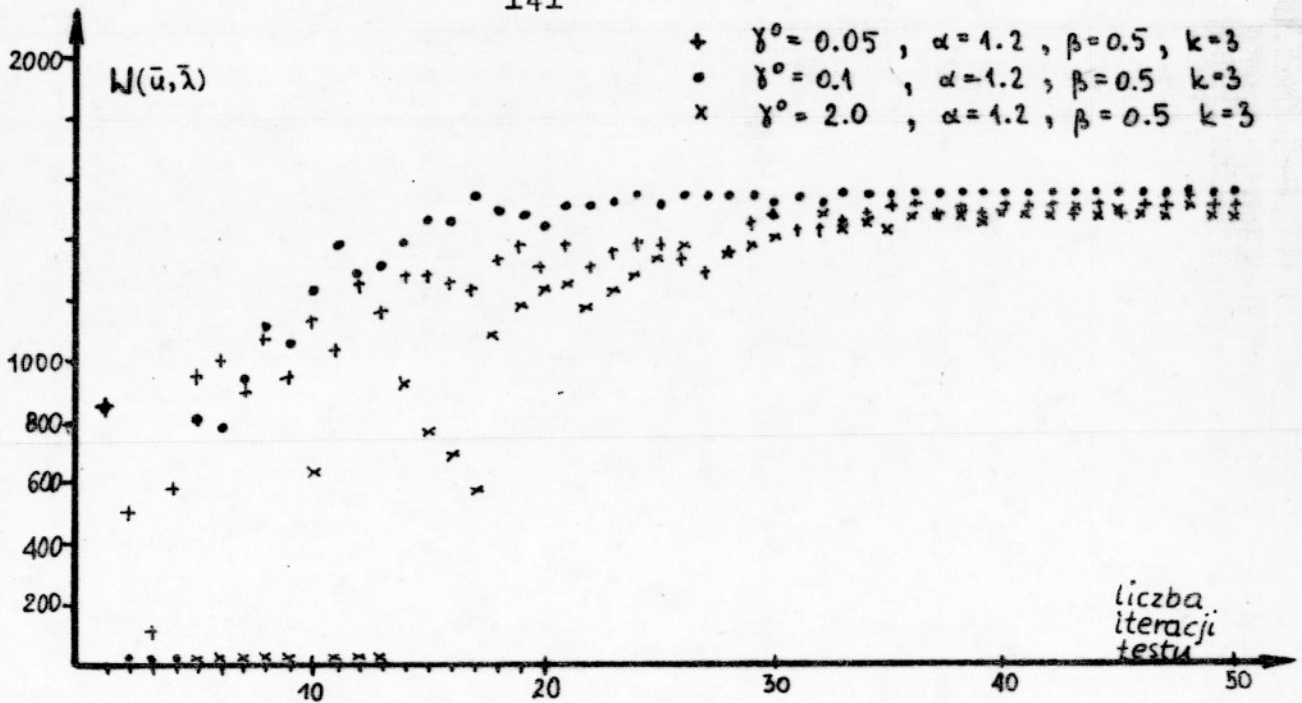
Wszystkie wylosowane wartości były zaokrąglane do najbliższych liczb całkowitych. Przyjęte wartości $(\tau_r, e_r, \tau_d, e_d)$ wynosiły odpowiednio:

$$(0.5, 1.0, 0.5, 1.0), \quad (0.8, 0.4, 0.8, 0.4), \\ (0.65, 0.2, 0.5, 1.0), \quad (0.65, 0.2, 0.65, 0.2),$$

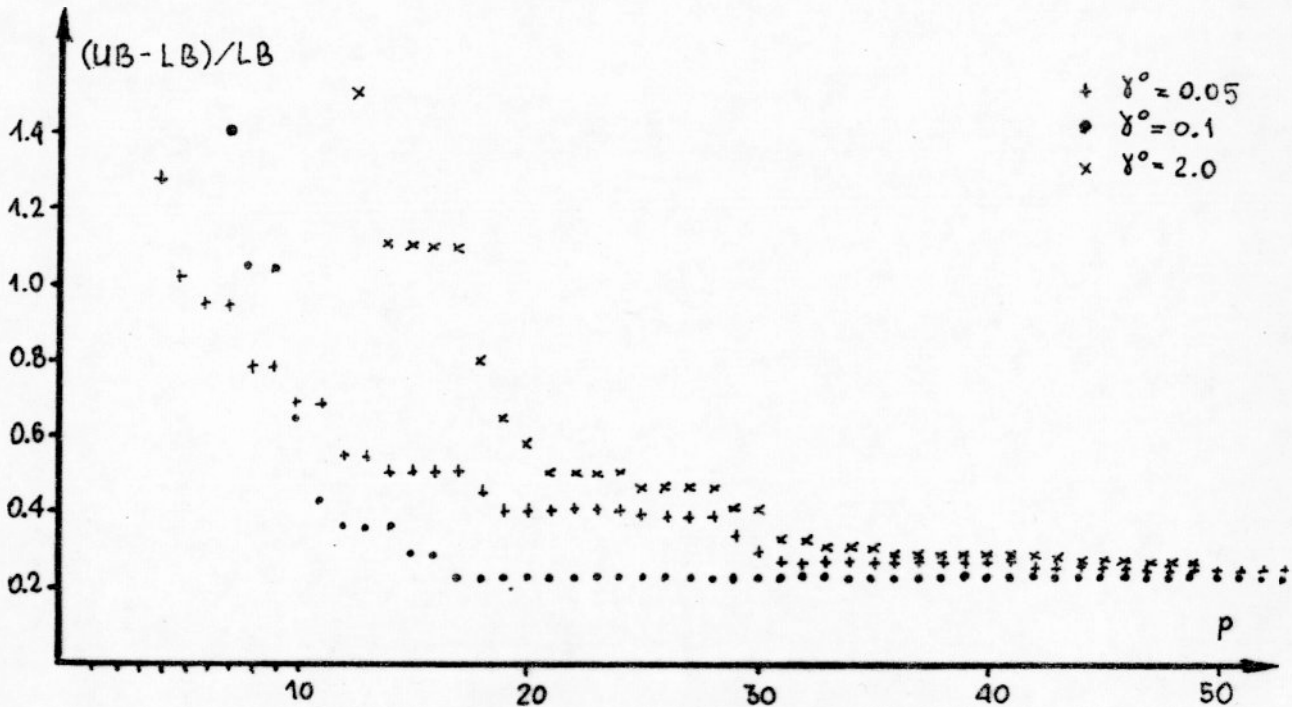
8.10. Eksperymenty obliczeniowe. Wyniki

Przy wykorzystaniu programu opisanego poprzednio zrealizowano szereg eksperymentów obliczeniowych w następującym zakresie:

A. Ocena wpływu początkowej wartości γ^0 na wartość testu LB3 z relaksacją 8.5. Badanie przeprowadzone na 24 przykładach z $n = 10$ i $n = 30$. W każdym przykładzie liczba elementów ciągu $\langle t_1, t_2, \dots, t_s \rangle$ w relaksacji 8.5 wynosiła $2n$, zaś ciąg γ^p wybierano w/g zasady $\gamma^p = \alpha \gamma^{p-1}$ ($\alpha > 1$), $p=1, 2, \dots$ gdy nastąpił wzrost wartości LB oraz $\gamma^p = \gamma^{p-1}$ gdy w kolejnych k iteracjach nie nastąpił wzrost wartości LB. Przyjmowano wartości $\gamma^0 = 2.4, 2.0, 0.5, 0.1, 0.05$. Otrzymane wyniki dla wybranego przykładu przedstawiono na rys. 8.13, 8.19. Wyniki otrzymane we wszystkich badanych przykładach były analogiczne i prowadziły do tych samych wniosków.



Rys. 8.18. Zależność wartości dolnego ograniczenia LB3 od wyboru γ^0 /przykład nr 11/

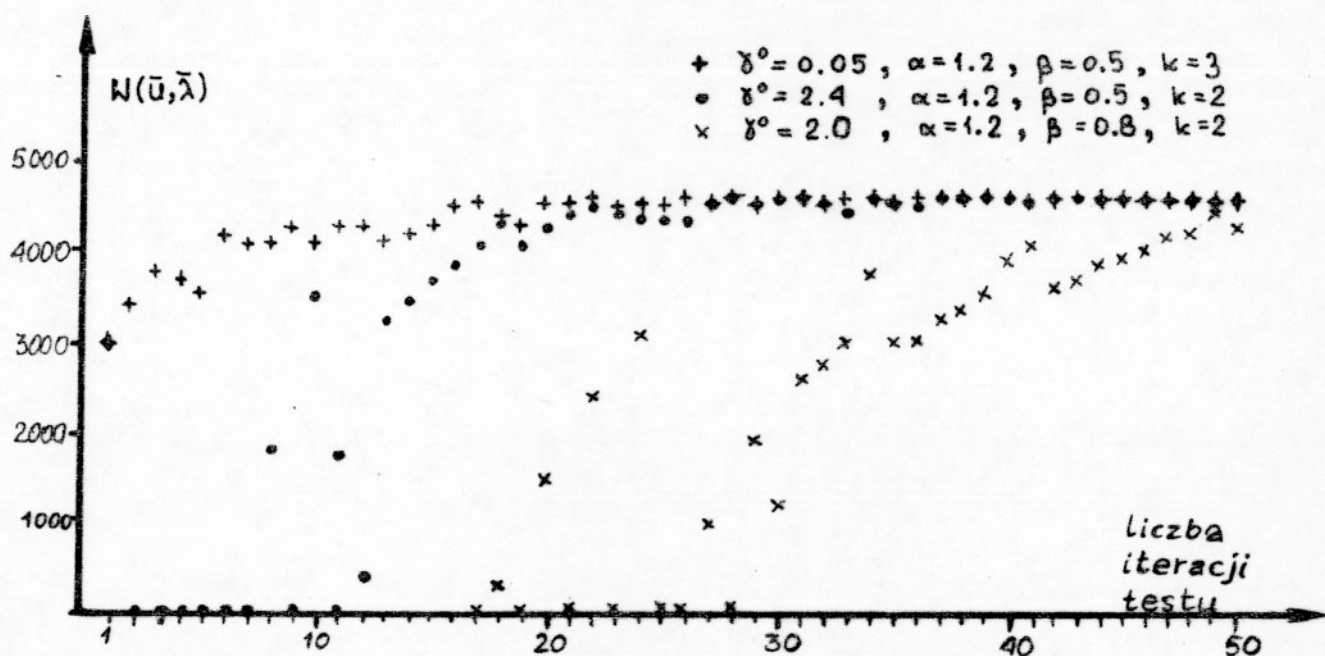


Rys. 8.19. Błąd oszacowania wartości funkcji celu przy użyciu LB3 w zależności od wyboru wartości γ^0 /przykład nr 11/.

B. Ocena wpływa postaci ciągu $\gamma^p = \theta(\gamma^{p-1}, p-1)$ na szybkość wzrostu wartości testu LB3 z relaksacją 8.5. Badanie przeprowadzono na przykładach z p. A.

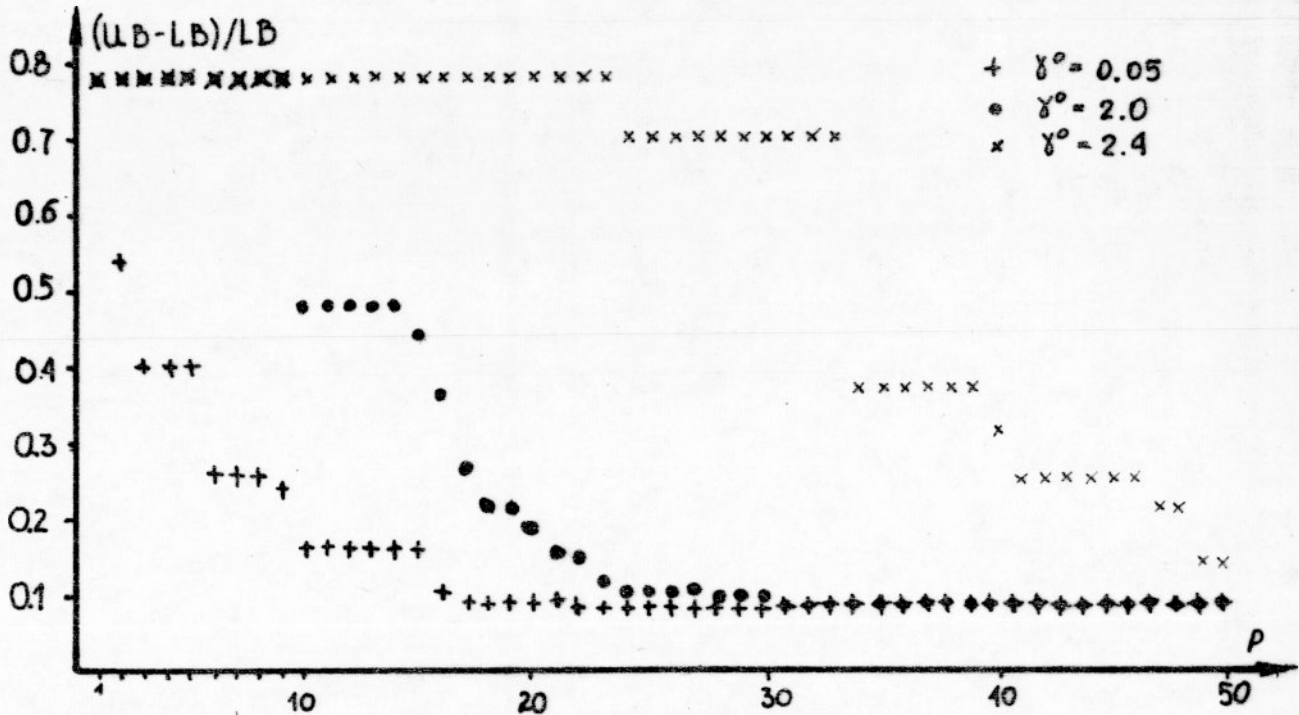
Przyjęto następującą postać ciągu: $\gamma^p = \alpha \gamma^{p-1}$ ($\alpha > 1$), gdy nastąpił wzrost LB^p w jednej iteracji oraz $\gamma^p = \beta \gamma^{p-1}$ ($0 < \beta < 1$) gdy w kolejnych k iteracjach nie nastąpił wzrost wartości LB /badane były również inne postacie ciągu γ^p /.

Otrzymane wyniki /dla wybranego przykładu/ przedstawiono na rys. 8.20, 8.21. Wyniki otrzymane we wszystkich badanych przykładach były analogiczne i prowadziły do tych samych wniosków.

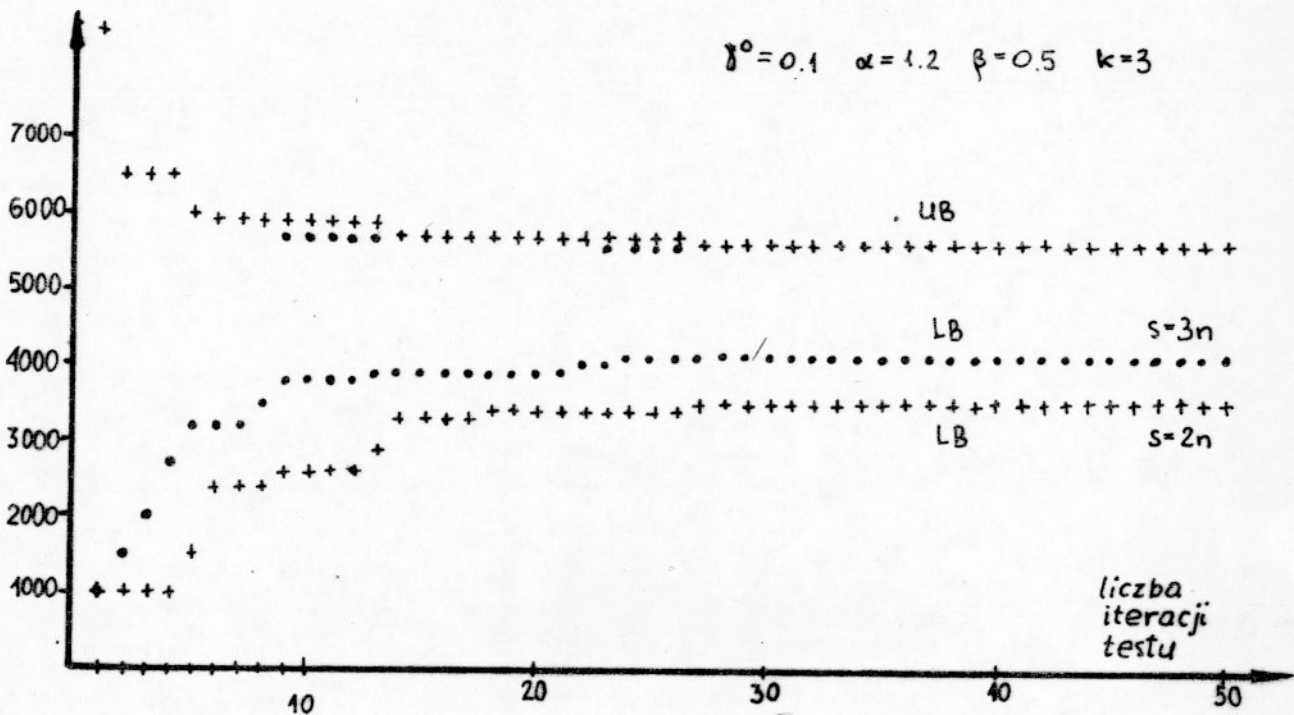


Rys. 8.20. Zależność wartości dolnego ograniczenia LB3 od wyboru postaci ciągu $\gamma^p = \theta(\gamma^{p-1}, p-1)$ /przykład nr 5/.

C. Ocena wpływu wyboru ciągu $\langle t_1, t_2, \dots, t_s \rangle$ w relaksacji 8.5 na wartość dolnego ograniczenia LB3. Badanie przeprowadzono na przykładach z p. A. Przyjęto postać ciągu $\gamma^p = \theta(\gamma^{p-1}, p-1)$, jak w p. B, tzn. $\gamma^p = \alpha \gamma^{p-1}$ ($\alpha = 1.2$), $\gamma^p = \beta \gamma^{p-1}$ ($\beta = 0.5$). Badano wpływ liczby elementów ciągu $\langle t_1, t_2, \dots, t_s \rangle$ na wartość LB3. Otrzymane wyniki /dla wybranego przykładu/ przedstawiono na rys. 8.22. Wyniki otrzymane we wszystkich badanych przykładach były analogiczne.



Rys. 8.24. Błąd oszacowania wartości funkcji celu przy użyciu LB3 w zależności od wyboru postaci ciągu $\gamma^p = \theta(\gamma^{p-1}, p-1)$ /przykład nr 5/



Rys. 8.22. Wpływ liczby elementów ciągu $\langle t_1, \dots, t_s \rangle$ na wartość dolnego ograniczenia LB3 /przykład nr 16/.

Badania wpływu wartości elementów ciągu $\langle t_1, t_2, \dots, t_s \rangle$ $s = 3n$ na wartość LB3 nie wykazało wyraźnej przewagi z żadnej z metod wyboru wartości $t_1, t_2, \dots, t_s$. Otrzymywane wyniki były zarówno lepsze, jak i gorsze od otrzymywanych w p. B.

D. Ocena podstawowych cech algorytmu problem $n|1||\sum h_i$, takich jak liczba iteracji, szybkość otrzymywania rozwiązania optymalnego i ϵ - optymalnego, praco- i czasochłonność obliczeniowa.

Uzyskane wyniki przedstawiono w tablicy 8.2.

Tablica 8.2.

nr przykł.	n	$F(S; \pi^0)$	F^*	LB_{π^0}	l. węzłów drzewa		liczba iteracji testu		czas obl. [s] **
					do rozw. najl.	całkowita	do rozw. najl.	całkowita	
1	10	4010	2588	2531	2	33	56	49	180
2	10	1036	861	801	5	28	65	134	180
3	10	6651	1757	1448	0	33	13	49	180
4	10	834	362*	362	6	6	68	63	40
5	10	5329	4756	4644	31	53	143	209	180
6	10	20491	16143	14897	0	30	35	140	180
7	10	7767	6104	3913	0	8	8	174	180
8	10	10989	8441	5328	0	24	11	122	180
9	10	25171	21043	14027	2	22	56	116	180
10	10	4284	2876	2283	0	65	50	245	180
11	10	5604	4887	1538	0	34	21	152	180
12	10	4301	509	459	26	33	128	149	180
13	10	3907	2022	1815	12	33	86	149	180
14	10	4731	1036*	1036	0	0	1	1	2
15	10	3077	1229	1075	28	51	134	103	180
16	10	8432	5258	3533	3	30	59	140	180
17	10	11975	8444	4976	0	30	50	140	180
18	10	8577	5943	4659	0	30	50	140	180
19	10	10083	5181	3532	4	27	62	131	180
20	10	7071	3533	2764	0	43	3	179	180
21	30	21738	13086	9325	0	5	20	65	180
22	30	146712	77363	33516	0	5	48	65	180
23	30	81089	20836	14526	0	5	50	65	180
24	30	120456	66780	29136	0	5	48	65	180

* rozwiązanie optymalne.

** ODRA 1305, limit czasu 180s

8.11. Uwagi i wnioski

Analiza czasów realizacji poszczególnych kroków algorytmu wykazała, że procedura obliczania dolnego ograniczenia zajmuje około 80 % czasu działania całego programu. Dlatego też zdecydowano się na

- (i) wprowadzenie relaksacji 8.4 w celu zmniejszenia złożoności obliczeniowej algorytmu rozwiązywania problemu Lagrange'a,
- (ii) maksymalne zmniejszenie liczby iteracji procedury subgradientowej w każdym węźle,

przy jednoczesnym maksymalnym zachowaniu "dobrych" własności testu LB3.

Analiza wykresów przedstawionych na rys. 8.18 - 8.22 prowadzi do wniosku, że już po kilkunastu iteracjach procedury subgradientowej wartość testu LB3 nie ulega większej zmianie i ustala się, w przybliżeniu, na poziomie wartości optymalnej problemu zrelaksowanego. Przeprowadzone eksperymenty doprowadziły do określenia "najkorzystniejszych" wartości oraz postaci ciągu $\delta^p = \theta(\delta^{p-1}, p-1)$, które zapewniają maksymalnie szybki wzrost wartości dolnego ograniczenia /rys. 8.18/. Dalszy wzrost wartości dolnego ograniczenia jest możliwy jedynie poprzez zmianę przyjętej relaksacji tj. zmianę liczby elementów ciągu $\langle t_1, t_2, \dots, t_s \rangle$ /powoduje to niekorzystny wzrost nakładów obliczeniowych/ i/lub zmianę wartości elementów ciągu $\langle t_1, t_2, \dots, t_s \rangle$. Przyjęte w rozdziale 8.10 wartości δ^0 , $\alpha, \beta, s, \langle t_1, t_2, \dots, t_s \rangle, IT_{\pi^0}, IT_{\pi}$ są więc kompromisem pozwalającym uzyskać maksymalnie pozytywne rezultaty przy względnie niedużych nakładach obliczeniowych. Określenie wpływu wartości podanych parametrów każdego z osobna oraz łączenie z pozostałymi, wymaga wielu czasochłonnych badań komputerowych.

Należy nadmienić, że we wszystkich rozwiązywanych przykładach wystąpił znaczny spadek wartości funkcji celu najlepszego rozwiązania względem rozwiązania początkowego. Wartość dolnego ograniczenia była w granicach 50 - 98 % wartości najlepszego rozwiązania wyznaczonego w trakcie działania algorytmu. Rozwiązanie to było znajdowane w kilku pierwszych iteracjach algorytmu i nie było dalej poprawiane. Należy się więc spodziewać, że otrzymane rozwiązanie jest bliskie rozwiązaniu optymalnemu. Eksperymenty obliczeniowe wskazują na możliwość zastosowania przedstawionego algorytmu

do rozwiązywania zagadnień średniego rozmiaru /rzędu $n = 50$ /.
 W przypadku zagadnień o większym rozmiarze należałoby zweryfikować wartości $\delta^0, \delta^p = \theta(\delta^{p-1}, p-1), t_1, t_2, \dots, t_s$, δ^0 pod kątem dalszego zmniejszenia nakładów obliczeniowych algorytmu.

9. MINIMALIZACJA KOSZTU CAŁKOWITEGO. ZAGADNIENIE $n|m|J, \tau \geq 1$, $m_w^k \geq 1, RT \mid \sum h_i$

9.1. Sformułowanie problemu

Dane są:

- (i) zbiór niepodzielnych operacji $N = \{1, 2, \dots, n\}$,
- (ii) zbiór maszyn różnych typów $M = \{1, 2, \dots, m\}$ przeznaczonych do wykonywania tych operacji,
- (iii) zbiór typów tych maszyn $Q = \{1, 2, \dots, q\}$,
- (iv) czasy wykonywania operacji p_{ik} ($i \in N, k \in M$), $p_{ik} > 0$;
 p_{ik} jest czasem wykonywania operacji i -tej na maszynie k -tej,
- (v) relacja poprzedzania RT reprezentująca wymogi porządku technologicznego wykonywania operacji $RT \subset N \times N$; $\langle i, j \rangle \in RT$ pociąga, że operacja i -ta musi być zakończona przed rozpoczęciem operacji j -tej,
- (vi) najwcześniejszy możliwy termin rozpoczęcia wszystkich operacji $t_0, t_0 \geq 0$ /zwykle $t_0 = 0$ /,
- (vii) najwcześniejszy pożądaný termin rozpoczęcia realizacji operacji i -tej na maszynie k -tej, $r_{ik} \geq 0$ / $i=1, 2, \dots, n, k=1, 2, \dots, m$ /,
- (viii) najpóźniejszy pożądaný termin zakończenia realizacji operacji i -tej na maszynie k -tej, $d_{ik} \geq 0$ / $i=1, 2, \dots, n, k=1, 2, \dots, m$ /.

Zakłada się, że

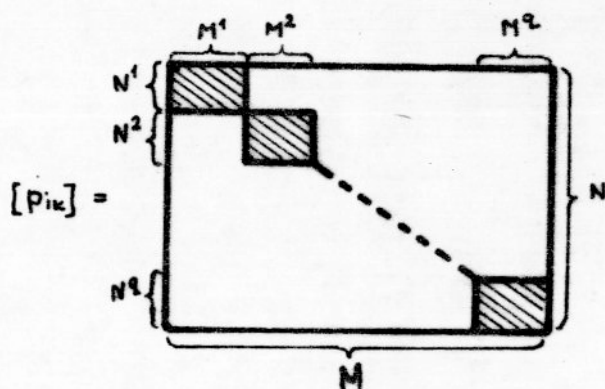
- (a) w zbiorze maszyn M istnieją rozłączne podzbiory maszyn jednego typu, tzn. istnieje rozbitcie zbioru M na podzbiory $M^k \subseteq M$ parami rozłączne i wyczerpujące

$$\bigcup_{k \in Q} M^k = M \quad M^k \cap M^j = \emptyset, \quad k \neq j, \quad k, j \in Q,$$

- (b) w zbiorze operacji N istnieją rozłączne podzbiory operacji, które mogą być wykonywane przy użyciu maszyn jednego typu, tzn. istnieje rozbięcie zbioru N na podzbiory $N^k \subseteq N$ parami rozłączne i wyczerpujące

$$\bigcup_{k \in Q} N^k = N, \quad N^k \cap N^j = \emptyset, \quad k \neq j, \quad k, j \in Q,$$

- (c) każda operacja ze zbioru N^k ($k \in Q$) może być wykonywana tylko przy użyciu dokładnie jednej maszyny ze zbioru M^k ($k \in Q$), co pociąga za sobą, że wielkości p_{ik} są określone tylko dla takich par (i, k) , że $i \in N^j, k \in M^j, j \in Q$ /rys. 9.1/,



Rys. 9.1. Struktura blokowa macierzy $[p_{ik}]$

- (d) wykonywanie operacji nie może być przerywane,
 (e) każda maszyna $k \in M$ może wykonywać, co najwyżej jedną operację w dowolnej chwili czasowej.

Wprowadzono dodatkowe oznaczenia:

x_{ik} - zmienna przydziału

$$x_{ik} = \begin{cases} 1 & \text{gdy operacja } i\text{-ta jest wykonywana na maszynie} \\ & \text{k-tej, } i \in N^j, k \in M^j, j \in Q, \\ 0 & \text{przeciwnie,} \end{cases}$$

zmienne x_{ik} są określone w przypadku dla takich par (i, k) dla których p_{ik} są określone,

S_i - termin rozpoczęcia operacji $i \in N$,

C_i - termin zakończenia realizacji operacji $i \in N$;

$$C_i = S_i + \sum_{k \in M^j} x_{ik} p_{ik}, \quad i \in N^j, j \in Q,$$

- $E_i \triangleq \max [0, \sum_{k \in M^j} r_{ik} x_{ik} - S_i]$ przyspieszenie wykonywania operacji $i \in N^j, j \in Q$,

- $T_i \triangleq \max [0, c_i \sum_{k \in M^j} d_{ik} x_{ik}]$ spóźnienie w wykonywaniu operacji $i \in N^j$, $j \in Q$.

Dodatkowo dane są:

(ix) funkcje $h_{ik}(S_i)$ określające koszt związany z rozpoczęciem realizacji operacji i -tej na maszynie k -tej w terminie S_i , $i \in N^j$, $k \in M^j$, $j \in Q$; zakłada się, że funkcje $h_{ik}(S_i)$ posiadają jedno minimum niekoniecznie właściwe /analogicznie jak na rys. 7.1/ oraz dają się przedstawić w formie

$$h_{ik}(S_i) = g_{ik}(E_i) + f_{ik}(T_i) + \text{const}, \quad (9.1)$$

gdzie $g_{ik}(E_i)$ niemalejąca funkcja argumentu E_i , $g_{ik}(0) = 0$, $f_{ik}(T_i)$ - niemalejąca funkcja argumentu T_i oraz $f_{ik}(0) = 0$, $i \in N^j$, $k \in M^j$, $j \in Q$.

Problem szeregowania $n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $RT | \sum h_i$ polega na wyznaczeniu wielkości S_i ($i \in N$), x_{ik} ($i \in N^j$, $k \in M^j$, $j \in Q$) minimalizujących funkcję celu w postaci

$$\sum_{j \in Q} \sum_{i \in N^j} \sum_{k \in M^j} h_{ik}(S_i) x_{ik}, \quad (9.2)$$

przy ograniczeniach

$$S_i + \sum_{k \in M^j} p_{ik} x_{ik} \leq S_j, \quad \langle i, j \rangle \in RT, \quad (9.3)$$

$$t_0 \leq S_i, \quad i \in N, \quad (9.4)$$

$$(x_{ik} = 1) \wedge (x_{jk} = 1) \Rightarrow (S_i + p_{ik} \leq S_j) \vee (S_j + p_{jk} \leq S_i), \\ i, j \in N^p, \quad k \in M^p, \quad p \in Q, \quad (9.5)$$

$$\sum_{k \in M^j} x_{ik} = 1, \quad i \in N^j, \quad j \in Q, \quad (9.6)$$

$$x_{ik} \in \{0, 1\}, \quad i \in N^j, \quad k \in M^j, \quad j \in Q. \quad (9.7)$$

Funkcja celu przedstawia sobą całkowity koszt wykonywania wszystkich operacji, przy założeniu, że każda operacja jest wykonywana przy użyciu dokładnie jednej maszyny /ograniczenie (9.6) -(9.7)/. Ograniczenie (9.3) reprezentuje wymagania kolejności realizacji operacji związane z istnieniem porządku technologicznego /relacja RT/. Ograniczenie (9.4) wyznacza najwcześniejszy moment rozpoczę-

cia wszystkich operacji. Ograniczenie kolejnościowe (9.5) określa, że jeśli dwie operacje (i, j) są realizowane na tej samej maszynie (k) to "realizacja operacji i poprzedza realizację operacji j " albo odwrotnie. Warunek (9.5) zapewnia, że operacje realizowane na tej samej maszynie muszą być uszeregowane i nie mogą być realizowane jednocześnie.

W zależności od wyboru postaci funkcji $h_{ik}(S_i)$ ($i \in N^j$, $k \in M^j$, $j \in Q$) możemy otrzymać zbiór różnych problemów szeregowania, które można zebrać w tabelkę analogiczną do tablicy 8.1. Co ciekawsze, wszystkie tak otrzymane problemy są NP-zupełne oraz brak jest w literaturze jakichkolwiek algorytmów rozwiązywania zagadnień tego typu.

Problem (9.2).-(9.7) jest zadaniem optymalizacji nieliniowej, ze zmiennymi częściowo całkowitoliczbowymi, z liniowymi ograniczeniami i dodatkowym logicznym warunkiem ograniczającym (9.5). Jest on również NP-zupełny więc wydaje się uzasadnione stosowanie np. metody podziału i ograniczeń do jego rozwiązania. Metoda grafu dysjunktywnego, tak bardzo pomocna przy rozwiązywaniu problemów szeregowania typu $n|m|\delta, \Gamma|f_{\max}$, w przypadku problemów $n|m|\delta, \Gamma|\sum h_i$ nie daje żadnych obiecujących rezultatów. Modele grafowe problemu mogą być jedynie ilustracją pewnych własności.

W poszukiwaniu jednolitego podejścia do problemów $n|m|\delta, \Gamma|\sum h_i$ zastosowano metodę podziału i ograniczeń z podejściem dualnym do wyznaczania dolnych ograniczeń. W niniejszej pracy dokonano szeregu uogólnień i przedstawiono możliwości zastosowania metody dualnej w różnych wariantach.

9.2. Pewne własności zagadnienia $n|m|J, \tau \geq 1, m_W^k \geq 1, RT|\sum h_i$

Z warunków (9.6) - (9.7) wynika, że w przypadku każdego $i \in N^j$ dokładnie jedna ze zmiennych x_{ik} , $k \in M^j$, $j \in Q$ przyjmuje wartość 1, tzn.

$$x_{i\mu(i)} = 1, x_{ik} = 0, k \neq \mu(i), k, \mu(i) \in M^j, i \in N^j, j \in Q, \quad (9.8)$$

gdzie $\mu(i)$ jest numerem maszyny, która została przydzielona do wykonywania i -tej operacji. Wówczas wektor $\bar{\mu} \triangleq \langle \mu(1), \mu(2), \dots, \mu(n) \rangle$ taki, że $i \in N^j$, $(i) \in M^j$, $j \in Q$ definiuje przydział wszystkich operacji do maszyn. Wielkość $\mu(i)$, $\mu(i) \in M^j$ będziemy nazywać

przydziałem operacji i-tej, $i \in N^j$, $j \in Q$. Zbiór przydziałów oznaczmy odpowiednio

$$\mathcal{M} = \{ \bar{\mu} \mid \bar{\mu} = \langle \mu(1), \mu(2), \dots, \mu(n) \rangle, \mu(i) \in M^j, \quad (9.9) \\ i \in N^j, j \in Q \}.$$

Korzystając z powyższego model (9.2) - (9.7) można przedstawić w prostszej postaci.

$$\min_{\bar{S}} \sum_{i \in N} h_i \mu(i) (S_i), \quad (9.10)$$

przy ograniczeniach

$$S_i + p_i \mu(i) \leq S_j, \quad \langle i, j \rangle \in RT, \quad (9.11)$$

$$t_0 \leq S_i, \quad i \in N, \quad (9.12)$$

$$(\mu(i) = \mu(j)) \Rightarrow (S_i + p_i \mu(i) \leq S_j) (S_j + p_j \mu(j) \leq S_i) \quad (9.13)$$

$$i, j \in N^p, \mu(i), \mu(j) \in M^p, p \in Q.$$

Modelem tym będziemy się posługiwać w dalszej części pracy.

Niech $\bar{\mu} \in \mathcal{M}$ będzie pewnym przydziałem. Wówczas

$$N_{k\bar{\mu}} \triangleq \{ i \in N \mid \mu(i) = k \}, \quad k \in M \quad (9.14)$$

definiuje zbiór operacji, które będą realizowane na k-tej maszynie przy danym przydziale $\bar{\mu}$. Oczywiście $N_{k\bar{\mu}} \leq N^j$, $k \in M^j$, $j \in Q$ oraz $n_{k\bar{\mu}} \triangleq |N_{k\bar{\mu}}|$, $0 \leq n_{k\bar{\mu}} \leq |N^j|$, $k \in M^j$, $j \in Q$. Kolejność realizacji operacji ze zbioru $N_{k\bar{\mu}}$ na maszynie k-tej jest określona permutacją

$$\pi_k = \langle \pi_k(1), \pi_k(2), \dots, \pi_k(n_{k\bar{\mu}}) \rangle, \pi_k(i) \in N_{k\bar{\mu}}, \quad (9.15) \\ i = 1, 2, \dots, n_{k\bar{\mu}}, k \in M,$$

zaś wektor permutacji $\bar{\pi} = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$ określa kolejność wykonywania operacji na wszystkich maszynach dla danego przydziału $\bar{\mu}$. Oczywiście zachodzi

$$S_{\pi_k(i)} + p_{\pi_k(i)} \mu(\pi_k(i)) \leq S_{\pi_k(i+1)}, \quad i = 1, 2, \dots, n_{k\bar{\mu}} - 1, \\ \pi_k(i), \pi_k(i+1) \in N_{k\bar{\mu}}, k \in M. \quad (9.16)$$

Wprowadzenie pojęcia permutacji umożliwiło spełnienie warunku logicznego (9.5) z modelu (9.2) - (9.7). Jeśli bowiem dla dwóch operacji $i, j \in N^p$ takich, że $\mu(i), \mu(j) \in M^p$ zachodzi $\mu(i) = \mu(j) = k$, to $i, j \in N_{k\mu}$ oraz "i poprzedza w permutacji j" albo "j poprzedza w permutacji i", co po uwzględnieniu (9.6) daje spełnienie tylko jednej z nierówności po prawej stronie warunku (9.5).

Niech dalej $\Pi_\mu = \{\bar{\pi} \mid \bar{\pi} = \langle \pi_1, \pi_2, \dots, \pi_m \rangle, \pi_k = \langle \pi_k(1), \pi_k(2), \dots, \pi_k(n_{k\mu}) \rangle, \pi_k(i) \in N_{k\mu}, k \in M\}$ będzie zbiorem wszystkich wektorów permutacji postaci (9.15) dla danego μ oraz niech $S_{\mu\pi}$ będzie zbiorem dopuszczalnych momentów rozpoczęcia operacji dla danego $\pi \in \Pi_\mu$, tzn.

$$S_{\mu\pi} = \{ \bar{s} \mid \bar{s} = \langle s_1, s_2, \dots, s_n \rangle, s_i + p_{i\mu(i)} \leq s_j, \langle i, j \rangle \in RT, \\ t_0 \leq s_i, i \in N, s_{\pi_k(i)} + p_{\pi_k(i)\mu(\pi_k(i))} \leq s_{\pi_k(i+1)}, \\ i = 1, 2, \dots, n_{k\mu} - 1, k = 1, 2, \dots, m \}, \quad (9.17)$$

Trójkę $\langle \bar{\mu}, \bar{\pi}, \bar{s} \rangle$, $\bar{s} \in S_{\mu\pi}$, $\bar{\pi} \in \Pi_\mu$, $\bar{\mu} \in M$ będziemy nazywać rozwiązaniem. Trójkę $\langle \bar{\mu}, \bar{\pi}, \bar{s}^* \rangle$, $\bar{s}^* \in S_{\mu\pi}$ taką, że

$$\sum_{i \in N} h_{i\mu(i)}(s_i^*) = \min_{\bar{s} \in S_{\mu\pi}} \sum_{i \in N} h_{i\mu(i)}(s_i) \quad (9.18)$$

będziemy nazywać rozwiązaniem optymalnym dla danego $\bar{\mu}$ i $\bar{\pi}$.

Problem (9.2) - (9.7) jest równoważny zagadnieniu wyznaczania trójki $\langle \bar{\mu}^*, \bar{\pi}^*, \bar{s}^* \rangle$, $\bar{s}^* \in S_{\mu^*\pi^*}$, $\bar{\pi}^* \in \Pi_{\mu^*}$, $\mu^* \in M$ takiej, że

$$\sum_{i \in N} h_{i\mu^*(i)}(s_i^*) = \min_{\bar{\mu} \in M} \min_{\bar{\pi} \in \Pi_\mu} \min_{\bar{s} \in S_{\mu\pi}} \sum_{i \in N} h_{i\mu(i)}(s_i) \quad (9.19)$$

Trójkę $\langle \bar{\mu}^*, \bar{\pi}^*, \bar{s} \rangle$ spełniającą warunek (9.19) będziemy nazywać rozwiązaniem globalnie optymalnym. Sposób wyznaczenia wartości $\bar{\mu}^* \in M$, $\bar{\pi}^* \in \Pi_{\mu^*}$ spełniających (9.19) opisano szczegółowo w rozdziale (9.3). W dalszym ciągu zajmiemy się tylko sposobem wyznaczenia wartości $\bar{s}^* \in S_{\mu\pi}$ tj. wyznaczeniem rozwiązania optymalnego (9.18). W ogólnym przypadku wyznaczenie rozwiązania optymalnego $\langle \bar{\mu}, \bar{\pi}, \bar{s} \rangle$, $\bar{s} \in S_{\mu\pi}$ dla danych wartości $\bar{\mu} \in M$, $\bar{\pi} \in \Pi_\mu$ wymaga rozwiązania następującego problemu optymalizacji

$$\min_{\bar{S}} \sum_{i=1}^n h_{i\mu(i)}(S_i), \quad (9.20)$$

$$S_i + p_{i\mu(i)} \leq S_j \quad \langle i, j \rangle \in RT, \quad (9.21)$$

$$t_0 \leq S_i \quad i = 1, 2, \dots, n, \quad (9.22)$$

$$S_{\pi_k(i) + p_{\pi_k(i)\mu(\pi_k(i))}} \leq S_{\pi_k(i+1)} \\ i = 1, 2, \dots, n_{k\mu} - 1, \quad k = 1, 2, \dots, m. \quad (9.23)$$

W przypadku, gdy funkcje $h_{ik}(S_i)$ mają postać (9.1) oraz funkcje $g_{ik}(E_i)$, $f_{ik}(T_i)$ są wypukłe, to funkcja celu (9.20) jest wypukła, nieliniowa i nieróżniczkowalna. Brak wypukłości funkcji $g_{ik}(E_i)$, $f_{ik}(T_i)$ może dodatkowo utrudniać rozwiązanie problemu (9.20) - (9.23) poprzez istnienie wieloekstremalności. Przy wprowadzeniu pewnych założeń o funkcjach $h_{ik}(S_i)$, $i \in N^j$, $k \in M^j$, $j \in Q$ można skonstruować efektywne algorytmy rozwiązania problemu (9.20) - (9.23). Niech $\mu \in \mathcal{M}$, $\bar{\pi} \in \prod_{\mu}$ będą odpowiednio przydziałem i wektorem permutacji zaś

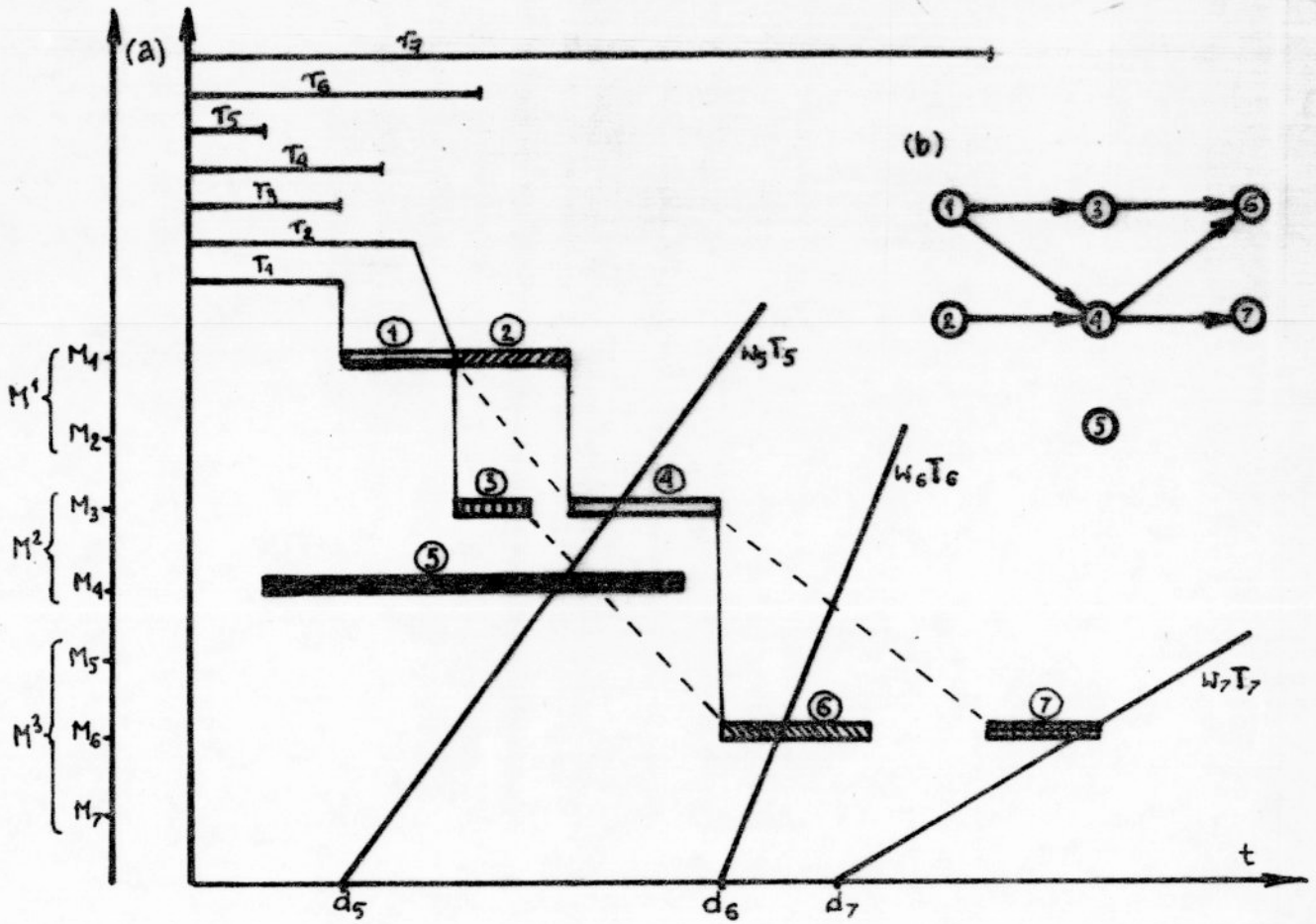
$$g_{ik}(E_i) = \begin{cases} 0 & E_i = 0 \\ \infty & E_i > 0 \end{cases} \quad i \in N^j, \quad k \in M^j, \quad j \in Q \quad (9.24)$$

oraz $f_{ik}(T_i)$ $i \in N^j$, $k \in M^j$, $j \in Q$ są funkcjami niemalejącymi, skończonymi. Wówczas zachodzi

$$r_{i\mu(i)} \leq S_i \quad i \in N^j, \quad i \in M^j, \quad j \in Q \quad (9.25)$$

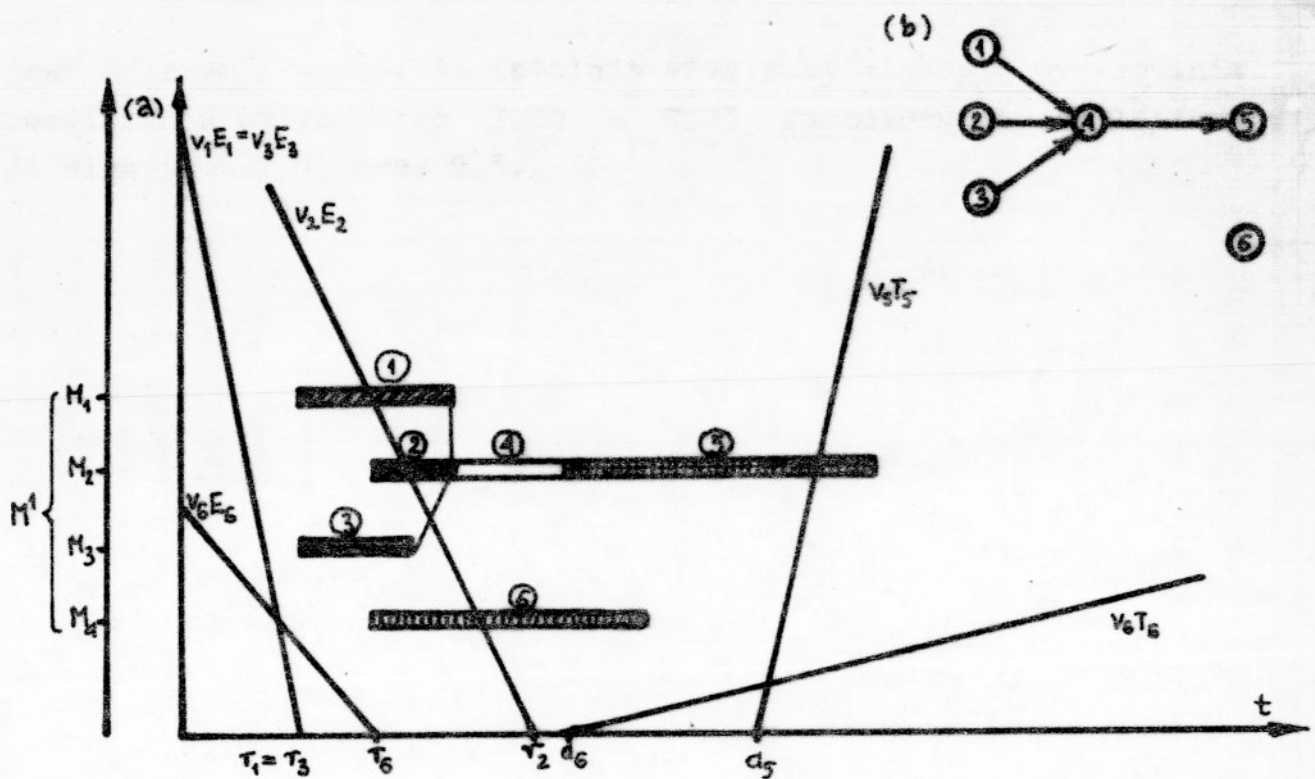
w rozwiązaniu optymalnym problemu (9.20) - (9.23) oraz optymalne momenty rozpoczęcia operacji można znaleźć przy zastosowaniu zasady maksymalnego wykorzystania maszyn /rys. 9.2/, tzn.

$$S_i = \max_{j \in B_i} \left[\max (S_j + p_{j\mu(j)}, r_{i\mu(i)}, t_0) \right]. \quad (9.26) \\ i = 1, 2, \dots, n$$



Rys. 9.2. Przykład problemu $n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $r_j \geq 0$, $RT[\sum$
 $N = \{1, 2, \dots, 7\}$, $M = \{1, \dots, 7\}$, $Q = \{1, 2, 3\}$,
 $N^1 = \{1, 2\}$, $N^2 = \{3, 4, 5\}$, $N^3 = \{6, 7\}$,
 $\bar{\mu} = \langle 1, 1, 3, 3, 4, 6, 6 \rangle$ $\pi_1 = \langle 1, 2 \rangle$, $\pi_2 = \emptyset$,
 $\pi_3 = \langle 4, 3 \rangle$, $\pi_4 = \langle 5 \rangle$, $\pi_5 = \emptyset$, $\pi_6 = \langle 6, 7 \rangle$,
 $\pi_7 = \emptyset$, (a) wykres Gantta, (b) graf relacji RT

W przypadku, gdy $h_{ik}(S_i)$, $i \in N^j$, $k \in M^j$, $j \in Q$ są funkcjami postaci (9.1) gdzie $g_{ik}(E_i)$, $f_{ik}(T_i)$, $i \in N^j$, $k \in M^j$, $j \in Q$ są funkcjami niemalejącymi, wypukłymi swoich argumentów oraz graf kolejności realizacji operacji wynikającej z warunków (9.21) i (9.23) jest drzewem /lasem/, to istnieje efektywny algorytm wyznaczania rozwiązania optymalnego (9.20) - (9.23) funkcjonujący analogicznie do algorytmu 7.1, rys. 9.3.



Rys. 9.3. Przykład problemu $|n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $RT | \sum h_i$
 $N = \{1, \dots, 6\}$, $M = \{1, \dots, 4\}$, $Q = \{1\}$, $N^1 = N$,
 $M^1 = M$, $\mu = \langle 1, 2, 3, 2, 2, 4 \rangle$, $\pi_1 = \langle 1 \rangle$,
 $\pi_2 = \langle 2, 4, 5 \rangle$, $\pi_3 = \langle 3 \rangle$, $\pi_4 = \langle 4 \rangle$;
 (a) wykres Gantta, (b) graf kolejności realizacji
 operacji

Niech $\langle \bar{\mu}, \bar{\pi}, \bar{S}^* \rangle$, $\bar{\mu} \in \mathcal{M}$, $\pi \in \Pi_{\mu}$, $\bar{S}^* \in \mathcal{S}_{\mu\pi}$ będzie rozwiązaniem optymalnym. Permutację częściową $\beta_k(x, y) = \langle \pi_k(x), \pi_k(x+1), \dots, \pi_k(y) \rangle$, $x \leq y$ nazywamy blokiem na maszynie k-tej, $k = 1, 2, \dots, m$ w rozwiązaniu $\langle \bar{\mu}, \bar{\pi}, \bar{S}^* \rangle$ jeśli

$$S_{\pi_k(i)}^* + p_{\pi_k(i)} \mu(\pi_k(i)) = S_{\pi_k(i+1)}^* \quad i = x, x+1, \dots, y-1 \quad (9.27)$$

Blok $\beta_k(x, y)$ nazywamy maksymalnym, jeśli $\beta_k(x-1, y)$ nie jest blokiem ani $\beta_k(x, y-1)$ nie jest blokiem. Blok maksymalny odpowiada maksymalnej sekwencji operacji wykonywanych na maszynie bez przerwy. Na rys. 9.2 blokami maksymalnymi są: na maszynie 1-szej - $\langle 1, 2 \rangle$, na maszynie 3-ciej - $\langle 3, 4 \rangle$, na maszynie 4-tej - $\langle 5 \rangle$, na maszynie 6-tej - $\langle 6, 7 \rangle$. Natomiast na rys. 9.3 blokami maksymalnymi są: na maszynie 1-szej - $\langle 1 \rangle$, na maszynie 2-giej - $\langle 2, 4, 5 \rangle$, na maszynie 3-ciej - $\langle 3 \rangle$, na maszynie

4-tej - $\langle 6 \rangle$.

Niech $\langle i_{0k}, i_{1k}, \dots, i_{l_k k} \rangle$, $k = 1, 2, \dots, m$ będą ciągami wskaźników takich, że $0 = i_{0k} < i_{1k} < i_{2k} < \dots < i_{l_k k} = n_{k\mu}$ oraz ograniczenia (9.23) w rozwiązaniu optymalnym $\langle \bar{\mu}, \bar{\pi}, \bar{S}^* \rangle$ problemu (9.20) - (9.23) są nieaktywne, tzn.

$$S_{\pi_k(i)}^* + p_{\pi_k(i)} \mu(\pi_k(i)) = S_{\pi_k(i+1)}^* \quad \begin{aligned} i_{j-1k} < i \leq i_{jk}, \\ j = 1, 2, \dots, l_k, \\ k = 1, 2, \dots, m, \end{aligned} \quad (9.28)$$

$$S_{\pi_k(i)}^* + p_{\pi_k(i)} \mu(\pi_k(i)) < S_{\pi_k(i+1)}^* \quad \begin{aligned} i = i_{1k}, i_{2k}, \dots, i_{l_k k} - 1, \\ k = 1, 2, \dots, m. \end{aligned} \quad (9.29)$$

Wówczas ciągi $\langle i_{0k}, i_{1k}, \dots, i_{l_k k} \rangle$, $k = 1, 2, \dots, m$ wyznaczają bloki maksymalne $\beta_k(i_{j-1k} + 1, i_{jk})$, $j = 1, 2, \dots, l_k$, $k = 1, 2, \dots, m$ w rozwiązaniu $\langle \bar{\mu}, \bar{\pi}, \bar{S}^* \rangle$, $\bar{S}^* \in \mathcal{S}_{\mu\pi}$, $\bar{\pi} \in \Pi_\mu$, $\bar{\mu} \in \mathcal{M}$.

Twierdzenie 9.1.

Niech $\bar{\mu} \in \mathcal{M}$, $\bar{\pi} \in \Pi_\mu$, RT będą odpowiednio przydziałem, wektorem permutacji, relacją porządku w zbiorze operacji. Jeśli dla każdego bloku maksymalnego w rozwiązaniu $\langle \bar{\mu}, \bar{\pi}, \bar{S}^* \rangle$ zachodzi

$$\langle \pi_k(i), \pi_k(i+1) \rangle \in RT \quad \begin{aligned} i_{j-1k} < i \leq i_{jk}, \quad j = 1, 2, \dots, l_k, \\ k = 1, 2, \dots, m, \end{aligned} \quad (9.30)$$

to

$$\sum_{i=1}^n h_i \mu(i)(S_i^*) \leq \min_{\bar{\beta} \in \Pi_\mu} \min_{\bar{S} \in \mathcal{S}_{\mu\beta}} \sum_{i=1}^n h_i \mu(i)(S_i) \quad (9.31)$$

Dowód twierdzenia 9.1 jest analogiczny do dowodu twierdzenia 8.9.

9.3. Schemat przebiegu algorytmu

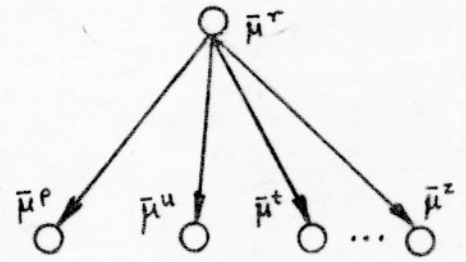
Ze wzoru (9.18) wynika, że rozwiązanie globalnie optymalne otrzymujemy generując ciąg przydziałów $\bar{\mu} \in \mathcal{M}$ i ciąg wektorów permutacji $\bar{\pi} \in \Pi_\mu$ / dla danego przydziału μ / oraz wyznaczając dla każdej wygenerowanej pary $\bar{\mu}, \bar{\pi}$ rozwiązanie optymalne (9.20) - (9.23).

W celu znalezienia rozwiązania globalnie optymalnego zastosowano metodę podziału i ograniczeń z mieszaną strategią podziału węzłów w drzewie rozwiązań.

Generacja przydziałów. Należy wygenerować wszystkie wektory $\bar{\mu} = \langle \mu(1), \mu(2), \dots, \mu(n) \rangle$ takie, że $\mu(i) \in M^k$, $i \in N^k$, $k \in Q$. Niech $\bar{\mu}^0 = \langle \mu^0(1), \mu^0(2), \dots, \mu^0(n) \rangle$ będzie pewnym przydziałem początkowym. Rozpoczynając od przydziału $\bar{\mu}^0$ będziemy generować ciąg przydziałów $\bar{\mu}^r$. Proces generowania przydziałów możemy przedstawić w postaci drzewa H rozwiązań. Każdy węzeł w H będzie odpowiadał pewnemu przydziałowi $\bar{\mu}^r$, zaś każdy łuk parze przydziałów $\langle \bar{\mu}^r, \bar{\mu}^p \rangle$ takich, że $\bar{\mu}^p$ został otrzymany z $\bar{\mu}^r$. Korzeń drzewa H będzie odpowiadał przydziałowi $\bar{\mu}^0$. Proces generowania przydziałów będzie się odbywał poprzez zamianę przydziału jednej z operacji, tzn. jeśli $\bar{\mu}^p$ został otrzymany z $\bar{\mu}^r$ to $\mu^p(i) = \mu^r(i)$ $i \in N^k$, $\mu^p(i) \in M^k$, $k \in Q$, $i \neq s$ oraz $\mu^p(s) \neq \mu^r(s)$ dla pewnego $s \in N^k$, $\mu^p(s), \mu^r(s) \in M^k$, $k \in Q$. Przydział operacji $\mu(s)$ będzie dokonywany kolejno do wszystkich maszyn ze zbioru M^k , $\mu(s) \in M^k$, $s \in N^k$, $k \in Q$. Podczas generowania nowego przydziału będziemy ustalać pewne przydziały operacji. Jeśli przydział $\bar{\mu}^p$ został otrzymany z $\bar{\mu}^r$ przez zamianę $k_p = \mu^p(s) \neq \mu^r(s)$, $\mu^p(s), \mu^r(s) \in M^k$, $s \in N^k$, $k \in Q$, to ustalamy chwilowo przydział $\mu^p(s) = k_p$ operacji s . Przydział tej operacji nie może zostać zmieniony w żadnym przydziale będącym następnikiem przydziału $\bar{\mu}^p$ w drzewie H .

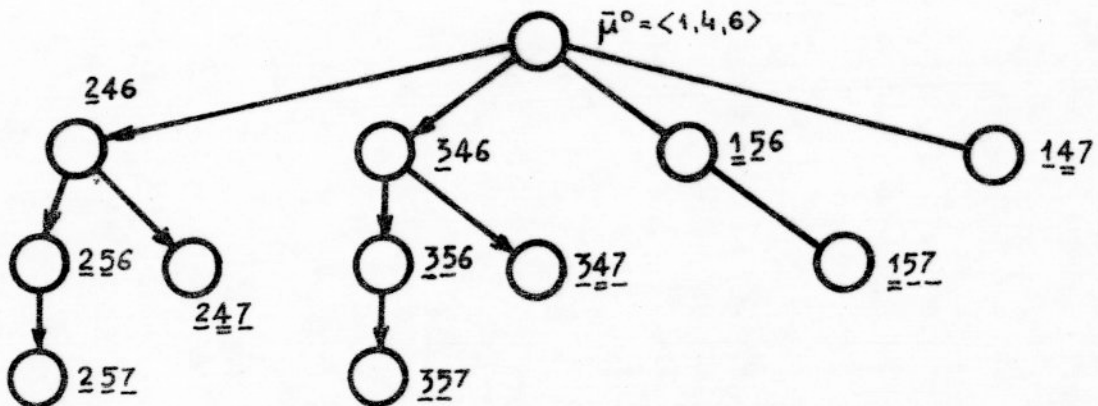
Jeżeli natomiast cofamy się z $\bar{\mu}^p$ do $\bar{\mu}^r$, to ustalamy tymczasowo przydział operacji $\mu^p(s)$. Jeżeli cofanie odbywa się po raz $|M^k| - 1$ - szy, gdzie $s \in N^k$, $\mu^p(s) \in M^k$, $k \in Q$, to przydział $\mu^r(s)$ zostaje ustalony trwale. Przydział operacji s dla której zachodzi $|M^k| = 1$, $s \in N^k$, $k \in Q$ będzie ustalony trwale od początku procesu generowania. Przydziały operacji ustalone tymczasowo nie biorą udziału w procesie generowania następników /w drzewie H / względem przydziału, dla którego zostały ustalone. Generowanie bezpośrednich następników $\bar{\mu}^p$ względem $\bar{\mu}^r$ powoduje, że zbiór przydziałów odpowiadających węzłowi $\bar{\mu}^p$ w H $Y(\bar{\mu}^p)$ zostaje rozbity na rozłączne podzbiory rys. 9.4. Jeżeli bezpośredni następnik $\bar{\mu}^p$ wygenerowano przez zamianę $k_p = \mu^p(s) \neq \mu^r(s)$, $\mu^p(s), \mu^r(s) \in M^k$, $s \in N^k$, $k \in Q$, to podzbiór $Y(\bar{\mu}^p)$ zawiera wszystkie przydziały takie, że $\mu^p(s) = k_p$. Jest to realizowane poprzez ustalenie

chwilowe przydziału operacji $\mu^p(s) = k$ nie zmieniającego się w żadnym następniku względem $\bar{\mu}^p$. Po dokonaniu cofania z $\bar{\mu}^p$ do $\bar{\mu}^r$ wygenerowanie kolejnego bezpośredniego następnika $\bar{\mu}$ względem $\bar{\mu}^r$ powinno odpowiadać podzbiorowi przydziałów takiemu, że $Y(\bar{\mu}^k) \cap Y(\bar{\mu}^p) = \emptyset$. Jest to możliwe tylko wówczas, gdy każdy przydział z podzbioru nie zawiera



Rys. 9.4. Fragment drzewa H przydziałów

przydziału operacji $\mu(s) = kp$. Realizuje się to poprzez ustalenie tymczasowe przydziału operacji $\mu(s) = kp$, co oznacza, że w żadnym następniku /innym niż omówiony $\bar{\mu}^p$ / względem $\bar{\mu}^r$ przydział ten nie może wystąpić. Po wykonaniu kolejnych $|M^k| - 1$ cofań ze względu na przydział operacji s-tej, $\mu(s) \in M^k$, $k \in Q$ otrzymujemy ustalenie tymczasowe wszystkich wartości $\mu(s) \in M^k$, $s \in N^k$, $k \in Q$ z wyjątkiem $\mu(s) = \mu^r(s)$. Stąd wniosek, że w celu zapewnienia rozłączności podzbiorów rozwiązań, wartość $\mu(s) = \mu^r(s)$ powinna być zachowana we wszystkich następnikach $\bar{\mu}^t$ generowanych w dalszym ciągu z $\bar{\mu}^r$. Realizuje się to poprzez ustalenie trwałe wartości $\mu^r(s)$.



Rys. 9.5. Drzewo przydziałów dla przykładu $N = \{1, 2, 3\}$, $Q = \{1, 2, 3\}$, $M = \{1, 2, \dots, 6\}$, $M^1 = \{1, 2, 3\}$, $M^2 = \{4, 5\}$, $M^3 = \{6, 7\}$, $N^1 = \{1\}$, $N^2 = \{2\}$, $N^3 = \{3\}$, $\bar{\mu}^0 = \langle 1, 4, 6 \rangle$. Podkreślono przydziały operacji ustalone chwilowo (-) i trwale (=)

Generowanie permutacji. Niech $\bar{\mu}$ będzie danym przydziałem oraz $\bar{\pi} \in \bar{\Pi} = \langle \pi_1, \pi_2, \dots, \pi_m \rangle$ wektorem permutacji gdzie

$$\pi_k = \langle \pi_k(1), \pi_k(2), \dots, \pi_k(n_{k\mu}) \rangle \quad \pi_k(i) \in N_{k\mu} \\ i = 1, 2, \dots, n_{k\mu}$$

jest permutacją operacji na maszynie k-tej. Należy wygenerować wszystkie permutacje $\bar{\pi} \in \bar{\Pi}$ takie, że $\bar{\pi}$ zawiera wszystkie permutacje operacji na wszystkich maszynach. Ponieważ będziemy dążyli do zastosowania eliminacyjnych własności bloków /twierdzenie 9.1/, więc w pierwszej kolejności będziemy generować permutacje w blokach na poszczególnych maszynach. Rozpoczynając od wektora permutacji $\bar{\pi}^0 \in \bar{\Pi}$ będziemy generować ciąg wektorów permutacji $\bar{\pi}^r \in \bar{\Pi}$. Proces generowania permutacji możemy przedstawić w postaci drzewa $\bar{H}$. Korzeniowi drzewa H odpowiada wektor permutacji $\bar{\pi}^0 \in \bar{\Pi}$. Każdemu węzłowi w H odpowiada pewien wektor $\bar{\pi}^r$, zaś łukowi w $\bar{H}$ - para $\langle \bar{\pi}^r, \bar{\pi}^p \rangle$ taka, że wektor $\bar{\pi}^p$ został otrzymany z $\bar{\pi}^r$ przez dokonanie zmiany kolejności wykonywania operacji na jednej z maszyn $k \in M$.

Niech $\bar{\pi}^p, \bar{\pi}^s, \dots, \bar{\pi}^u$ będzie ciągiem wektorów permutacji otrzymanych z $\bar{\pi}^r$ przez zmianę kolejności realizacji operacji na maszynie 1-szej, tzn. $\bar{\pi}^0 = \langle \pi_1^p, \pi_2^p, \dots, \pi_m^p \rangle$, $\pi_i^j = \pi_i^r$, $i = 1, 2, \dots, m$, $j = p, s, \dots, u$ oraz π_1^p jest pewną permutacją otrzymaną z π_1^r , zgodnie z zasadą opisaną w rozdziale /rys. 9.6/. Oczywiście otrzymanie nowej permutacji π_1^p powoduje chwilowe ustalenie relacji postaci (8.69) lub (8.70) w permutacji π_1 /analogicznie jak na rys 8.10/.

Wycofanie się z następników $\bar{\pi}^p, \bar{\pi}^s, \dots, \bar{\pi}^k$ spowoduje ustalenie tymczasowe, a po ostatnim wycofaniu ustalenie trwałe porządku w permutacji π_1^r . Porządek ten nie może być zmieniony w żadnym następniku /różnym od $\bar{\pi}^p, \bar{\pi}^s, \dots, \bar{\pi}^u$ / względem $\bar{\pi}^r$. Niech dalej $\bar{\pi}^t, \bar{\pi}^v, \dots, \bar{\pi}^z$ będzie ciągiem następników względem $\bar{\pi}^r$ takich, że $\pi_1^j = \pi_1^r$, $\pi_i^j = \pi_i^r$, $i = 3, 4, \dots, m$, $j = t, v, \dots, z$ oraz π_2^t jest pewną permutacją otrzymaną z π_1^r przez zmianę kolejności wykonywania operacji na maszynie 2-giej. Kolejne wycofywanie się z następników $\bar{\pi}^t, \bar{\pi}^u, \dots, \bar{\pi}^z$. Spowoduje ustalenie trwałe porządku w permutacji π_2^r . Połączenie otrzymanego ustalenia z poprzednim daje warunek, że porządek wykonywania operacji

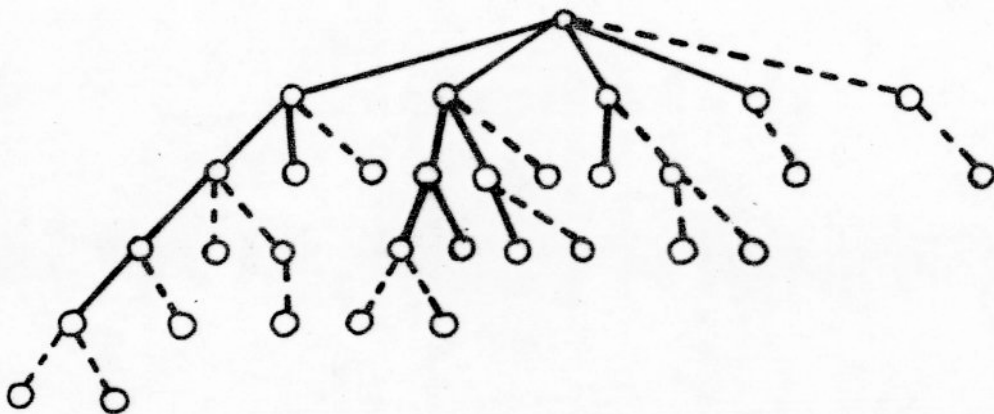
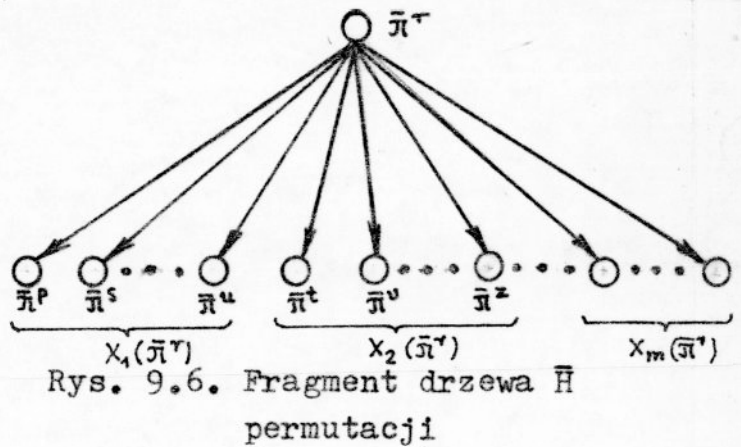
na maszynie 1-szej i 2-giej nie może ulec zmianie w żadnym następniku /różnym od $\bar{\pi}^p, \bar{\pi}^s, \dots, \bar{\pi}^u, \bar{\pi}^t, \bar{\pi}^v, \dots, \bar{\pi}^z$ / względem $\bar{\pi}^r$.

Kontynuując powyższe rozważania zauważamy, że zbiór następników względem $\bar{\pi}^r$ możemy przedstawić jako sumę

rozłącznych i wyczerpujących

podzbiorów $X_i(\bar{\pi}^r)$, $i = 1, 2, \dots, m$. Każdy podzbiór $X_i(\bar{\pi}^r)$ zawiera wektory permutacji otrzymane z $\bar{\pi}^r$ przez wygenerowanie wszystkich możliwych permutacji π_i na maszynie i-tej, przy nie zmienionych permutacjach operacji na maszynach pozostałych, tzn. $\pi_j = \pi_j^r$, $j = 1, 2, \dots, m$, $j \neq i$ oraz przy ustalonym trwale porządku w permutacjach π_j , $j = 1, 2, \dots, i-1$. Wobec powyższego proces generowania następników węzła odpowiadającego $\bar{\pi}^r$ możemy przedstawić jako superpozycję generacji permutacji π_i , $i = 1, 2, \dots, m$ na jednej maszynie i-tej przy założeniu, że w czasie generacji permutacji operacji na maszynie i-tej ustalamy arbitralnie porządek dla $i-1$ pierwszych maszyn. Proces generowania wszystkich permutacji na wybranej maszynie jest identyczny jak opisany w rozdziale

Drzewo rozwiązań. Pełne drzewo rozwiązań $\hat{H}$ jest superpozycją drzewa H odpowiadającego wszystkim możliwym przydziałom oraz drzew rozwiązań $\bar{H}_\mu$ odpowiadających wszystkim możliwym permutacjom /dla danego przydziału μ /, rys. 9.7.



Rys. 9.7. Drzewo $\hat{H}$ rozwiązań. Linia - zaznaczono drzewo H ,
linią - - - drzewa $\bar{H}_\mu$

Każdy węzeł "drzewa przydziałów" jest jednocześnie korzeniem "drzewa permutacji" w przypadku przydziału odpowiadającego temu węzłowi. Każdy węzeł "drzewa przydziału" odpowiada pewnemu rozwiązaniu, będącego rozwiązaniem początkowym "drzewa permutacji". Rozwiązanie to jest znajdowane na drodze heurystycznej. Każdy węzeł "drzewa permutacji" odpowiada pewnemu rozwiązaniu otrzymanemu z poprzedniego węzła poprzez zmianę kolejności operacji w pewnym bloku.

Węzeł w $\hat{H}$ możemy odrzucić, jeżeli:

- (a) dolne ograniczenie wartości funkcji celu (9.10) dla wszystkich następników wygenerowanych z danego węzła jest nie mniejsze niż aktualne górne ograniczenie,
- (b) żaden z bloków maksymalnych nie zawiera więcej, niż jedną operację,
- (c) żadna z operacji nie może być przesuwana w bloku bez naruszenia ustalonych wymagań częściowego porządku.

Dwa ostatnie przypadki wynikają z twierdzenia 9.1.

9.4. Dolne ograniczenia

Należy wyznaczyć dolne ograniczenie wartości funkcji celu problemu $n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $R_{\mu n} | \sum h_i$, gdzie $R_{\mu n}$ jest relacją ustaloną w węźle odpowiadającym wektorowi permutacji

W tym rozdziale przedstawione zostaną dwa sposoby wyznaczania dolnego ograniczenia.

- (i) poprzez rozwiązanie problemu dualnego programowania dyskretnego (LB1),
- (ii) rozwiązanie zrelaksowanej postaci LB1 (LB2).

Dolne ograniczenie LB1. Niech $[t_0, T]$ będzie przedziałem czasu o tej własności, że

$$S_i^* + p_i \mu^*(i) \leq T \quad (9.32)$$

w rozwiązaniu globalnie optymalnym

Podzielmy przedział $[t_0, T]$ na rozłączne podprzedziały o długości jednostkowej $[t-1, t]$, $t = t_0+1, t_0+2, \dots, T$. Zdefiniujmy zbiór operacji wykonywanych w przedziale $[t-1, t]$ na maszynie k -tej

$$G_{t,k}(\bar{S}, \bar{\mu}) = \{i \in N_{k\mu} \mid \mu(i) = k, t - p_i \mu(i) \leq S_i \leq t-1\}$$

$$t = t_0 + 1, \dots, T, \quad k = 1, 2, \dots, m, \quad (9.33)$$

oraz liczbę operacji wykonywanych w przedziale $[t-1, t]$ na maszynie k -tej

$$g_{t,k}(\bar{S}, \bar{\mu}) = |g_{t,k}(\bar{S}, \bar{\mu})|, \quad t = t_0 + 1, \dots, T, \quad k = 1, 2, \dots, m. \quad (9.34)$$

Wówczas ograniczenie na skończoną przepustowość wszystkich maszyn można zapisać w formie

$$g_{t,k}(\bar{S}, \bar{\mu}) \leq 1 \quad t = t_0 + 1, \dots, T, \quad k = 1, 2, \dots, m, \quad (9.35)$$

zaś problem (9.2) - (9.7) w formie

$$\min_{S_1, S_2, \dots, S_n} \sum_{i=1}^n h_i \mu(i)(S_i) \quad (9.36)$$

$$g_{t,k}(\bar{S}, \bar{\mu}) \leq 1, \quad t = t_0 + 1, \dots, T, \quad k \in M, \quad (9.37)$$

$$S_i + p_i \mu(i) \leq S_j, \quad \langle i, j \rangle \in \mathcal{R}_{\mu\pi} \quad (9.38)$$

$$t_0 \leq S_i, \quad i \in N, \quad (9.39)$$

$$S_i + p_i \mu(i) < T, \quad i \in N^j, \mu(i) \in M^j, \quad j \in Q. \quad (9.40)$$

W przypadku, gdy dane są najwcześniejsze dopuszczalne momenty rozpoczęcia operacji na maszynach a_{ik} / $i \in N^j$, $k \in M^j$, $j \in Q$ / oraz najpóźniejsze dopuszczalne momenty zakończenia operacji na maszynach b_{ik} ($i \in N^j$, $k \in M^j$, $j \in Q$) to możemy do problemu (9.36) - (9.40) wprowadzić ograniczenia w postaci

$$a_i \mu(i) \leq S_i, \quad i \in N^j, \mu(i) \in M^j, \quad j \in Q \quad (9.41)$$

$$S_i + p_i \mu(i) \leq b_i \mu(i), \quad i \in N^j, \mu(i) \in M^j, \quad j \in Q. \quad (9.42)$$

Wielkości a_{ik} , b_{ik} mogą być dane w sposób arbitralny / np. $a_{ik} = \max[\bar{a}_i, \hat{a}_k]$ gdzie $\bar{a}_i$ - termin dostępności operacji, $\hat{a}_k$ - termin dostępności maszyny / bądź wyznaczona na podstawie znajomości przydziału μ i relacji $\mathcal{R}_{\mu\pi}$. Tak na przykład

$$a_i \mu(i) = \min_{j \in \Gamma_i} \{a_j \mu(j) + p_i \mu(i)\}, \quad (9.43)$$

jeśli przydział $\mu(i)$ operacji i jest ustalony oraz

$$a_i \mu(i) = \min_{j \in \Gamma^{-1}(\mathcal{R}_{\mu\pi})} a_j \mu(j) + \min_{\mu(i) \in M} p_i \mu(i), \quad (9.44)$$

jeśli przydział $\mu(i)$ operacji i nie jest ustalony.

Wprowadzając zmienne dualne $\bar{u} \triangleq [u_{t,k}]^T, t = t_0+1, t_0+2, \dots, T$, $k = 1, 2, \dots, m$ skojarzone z każdym ograniczeniem (9.37) oraz zmienne dualne λ_{ij} , $\langle i, j \rangle \in \mathcal{R}_{\mu\pi}$ skojarzone z każdym ograniczeniem postaci (9.38) definiujemy funkcję Lagrange'a

$$\begin{aligned} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) = & \sum_{i=1}^n h_i \mu(i) (S_i) + \sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} g_{tk}(\bar{S}, \bar{\mu}) + \\ & + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} \lambda_{ij} (S_i + p_i \mu(i) - S_j) \end{aligned} \quad (9.45)$$

oraz problem Lagrange'a

$$W(\bar{u}, \bar{\lambda}) = \min_{\substack{\bar{S} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) - \sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk}, \quad (9.46)$$

gdzie zbiór $\mathcal{S}$ wyznaczony jest przez warunki (9.39) - (9.42) zaś $\mathcal{M}$ jest dany wzorem (9.9).

Własność 9.1.

Wartość $W(\bar{u}, \bar{\lambda})$ (9.46) jest dalszym ograniczeniem wartości kosztu problemu szeregowania $n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $RT = R_{\mu\pi} \sum h_i$ dla dowolnych wartości $\bar{u} \geq 0$, $\bar{\lambda} \geq 0$. ■

Dowód

Niech $\bar{\mu}^* \in \mathcal{M}$, $\bar{S}^* \in \mathcal{S}$ oznacza optymalne rozwiązanie problemu (9.46). Zauważmy, że

$$\begin{aligned} W(\bar{u}, \bar{\lambda}) = & \min_{\substack{\bar{S} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) - \sum_{k=1}^m \sum_{t=t_0+1}^T u_{t,k} \leq \\ \leq & L(\bar{S}^*, \bar{\mu}^*, \bar{u}, \bar{\lambda}) - \sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} = \sum_{i=1}^n h_i \mu^*(i) (S_i^*) + \\ & + \sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} (g_{t,k}(\bar{S}^*, \bar{\mu}^*) - 1) + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} \lambda_{ij} (S_i^* + p_i \mu^*(i) - S_j^*) \leq \end{aligned}$$

$$\leq \sum_{i=1}^n h_{i\mu^*}(i)(s_i^*) \quad , \quad (9.47)$$

bowiem

$$\sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} (g_{tk}(\bar{s}^*, \bar{\mu}^*) - 1) \leq 0 \quad \text{oraz} \quad (9.48)$$

$$\sum_{\langle i,j \rangle \in \mathcal{Q}_{\mu\pi}} \lambda_{ij} (s_i^* + p_{ij}\mu^*(i) - s_j^*) \leq 0, \quad (9.49)$$

co kończy dowód. ■

Najlepsza wartość dolnego ograniczenia jest określona przez zmienne dualne $\bar{u} \geq 0$, $\bar{\lambda} \geq 0$, które maksymalizują

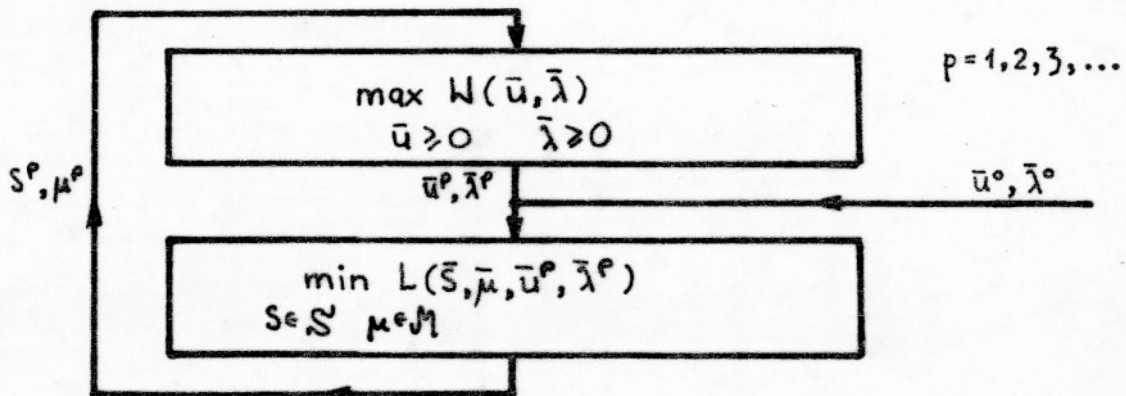
$$\max W(\bar{u}, \bar{\lambda}). \quad (9.50)$$

$$\bar{u} \geq 0, \quad \bar{\lambda} \geq 0 \quad (9.51)$$

Podobnie jak w przypadku problemu $n+1 \leq |\sum h_i|$ funkcja Lagrange'a przedstawiają sobą sumę:

- (i) kosztów realizacji operacji $h_{i\mu}(i)(s_i)$,
- (ii) kosztów użycia maszyn u_{tk} w przedziałach $[s_i, s_i + p_{ij}\mu(i)]$,
- (iii) kosztów wynikających z niespełnienia relacji $\mathcal{Q}_{\mu\pi}$.

Do rozwiązania problemu Lagrange'a zaproponowano dwupoziomowy algorytm z efektywną wersją PD na dolnym poziomie i metodą subgradientową na górnym poziomie, rys. 9.8.



Rys. 9.8. Dwupoziomowy algorytm problemu (9.36) - (9.42)

W celu określenia algorytmu dualnego poziomu wyrażenie (9.45) zostało przekształcone do odpowiedniej postaci

$$\begin{aligned}
 L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) &= \sum_{i=1}^n h_{i\mu(i)}(S_i) + \sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} g_{tk}(\bar{S}, \bar{\mu}) + \\
 &+ \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} \lambda_{ij} (S_i + p_{i\mu(i)} - S_j) = \sum_{i=1}^n h_{i\mu(i)}(S_i) + \sum_{i=1}^n \sum_{t=S_i+1}^{S_i+p_{i\mu(i)}} u_{t\mu(i)} + \\
 &+ \sum_{i=1}^n \sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} (S_i + p_{i\mu(i)}) - \sum_{i=1}^n \sum_{j \in \Gamma_i^{-1}(\mathcal{R}_{\mu\pi})} \lambda_{ji} S_i = \\
 &+ \sum_{i=1}^n \left[h_{i\mu(i)}(S_i) + \sum_{t=S_i+1}^{S_i+p_{i\mu(i)}} u_{t\mu(i)} + S_i \left(\sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} - \sum_{j \in \Gamma_i^{-1}(\mathcal{R}_{\mu\pi})} \lambda_{ji} \right) + \right. \\
 &\left. + \sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} p_{i\mu(i)} \right]. \tag{9.52}
 \end{aligned}$$

Korzystając z ostatecznej postaci wzoru (9.52) możemy napisać

$$\begin{aligned}
 \min_{\substack{\bar{S} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) &= \min_{\bar{\mu} \in \mathcal{M}} \min_{\bar{S} \in \mathcal{S}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) = \\
 &= \min_{\bar{\mu} \in \mathcal{M}} \min_{\bar{S} \in \mathcal{S}} \left[\sum_{i=1}^n h_{i\mu(i)}(S_i) + \sum_{t=S_i+1}^{S_i+p_{i\mu(i)}} u_{t\mu(i)} + \right. \\
 &+ \left. S_i \left(\sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} - \sum_{j \in \Gamma_i^{-1}(\mathcal{R}_{\mu\pi})} \lambda_{ji} \right) + \sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} p_{i\mu(i)} \right] = \\
 &= \sum_{i=1}^n \min_{\bar{\mu} \in \mathcal{M}} \min_{\bar{S} \in \mathcal{S}} \left[h_{i\mu(i)}(S_i) + \sum_{t=S_i+1}^{S_i+p_{i\mu(i)}} u_{t\mu(i)} + \right. \\
 &+ \left. S_i \left(\sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} - \sum_{j \in \Gamma_i^{-1}(\mathcal{R}_{\mu\pi})} \lambda_{ji} \right) + \sum_{j \in \Gamma_i(\mathcal{R}_{\mu\pi})} \lambda_{ij} p_{i\mu(i)} \right] \triangleq
 \end{aligned}$$

$$\triangleq \sum_{i=1}^n \min_{\bar{\mu} \in \mathcal{M}} \min_{\bar{S} \in \mathcal{S}} L_{i\mu(i)}(S_i, \mu(i), \bar{u}, \bar{\lambda}), \quad (9.53)$$

gdzie

$$L_{i\mu(i)}(S_i, \mu(i), \bar{u}, \bar{\lambda}) \triangleq h_{i\mu(i)}(S_i) + \sum_{t=S_i+1}^{S_i+p_{i\mu(i)}} u_t \mu(i) + \left(\sum_{j \in \Gamma_i(\mathcal{R}_{\mu n})} \lambda_{ij} - \sum_{j \in \Gamma_i^{-1}(\mathcal{R}_{\mu n})} \lambda_{ji} \right) S_i + \sum_{j \in \Gamma_i(\mathcal{R}_{\mu n})} \lambda_{ij} p_{i\mu(i)}. \quad (9.54)$$

Można zauważyć, że z wzorów (9.53) i (9.54) wynika, że jeśli przydział operacji i^* do maszyny $k^* = \mu(i^*)$ jest ustalony /chwilowo bądź trwale/ to

$$\min_{\substack{\bar{S} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda}) = \sum_{\substack{i=1 \\ i \neq i^*}}^n \min_{\bar{\mu} \in \mathcal{M}} \min_{\bar{S} \in \mathcal{S}} L_{i\mu(i)}(S_i, \mu(i), \bar{u}, \bar{\lambda}) + L_{i^*k^*}(S_{i^*}, k^*, \bar{u}, \bar{\lambda}). \quad (9.55)$$

Otrzymane rezultaty stały się podstawą konstrukcji algorytmu dolnego poziomu $\min_{\substack{\bar{S} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} L(\bar{S}, \bar{\mu}, \bar{u}, \bar{\lambda})$, rys. 9.9

Rozpatrzmy teraz problem dualny (9.50) - (9.51) do rozwiązania którego została zastosowana metoda subgradientowa [50].

Charakterystyczne cechy tej metody, w zastosowaniu do zagadnień szeregowania podano w rozdziale 8.6. Dla problemu (9.50) bez warunku (9.51) zaproponowano następujący wzór wyznaczania ciągu $(\bar{u}^p, \bar{\lambda}^p)$.

$$\bar{u}^0 = 0, \quad \bar{\lambda}^0 = 0 \quad (9.56)$$

$$u_{tk}^{p+1} = u_{tk}^p + E^p (g_{tk}(\bar{S}^p, \bar{\mu}^p) - 1) \quad t=t_0+1, t_0+2, \dots, T, \\ k=1, 2, \dots, m, \quad (9.57)$$

$$\lambda_{ij}^{p+1} = \lambda_{ij}^p + E^p (S_i^p + p_{i\mu^p(i)} - S_j^p) \langle i, j \rangle \quad (9.58)$$

$$E^p = \gamma^p \frac{UB - W(\bar{u}^p, \bar{\lambda}^p)}{\sqrt{\sum_{k=1}^m \sum_{t=t_0+1}^T (g_{tk}(\bar{S}^p, \bar{\mu}^p) - 1)^2 + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu n}} (S_i^p + p_{i\mu^p(i)} - S_j^p)^2}} \quad (9.59)$$

gdzie UB - górne ograniczenie, γ^P - skalar dobrany eksperymentalnie, $\bar{S}^P = \langle S_1^P, \dots, S^P \rangle$ oraz $\bar{\mu}^P = \langle \mu^P(1), \dots, \mu^P(n) \rangle$ są wartościami rozwiązania problemu dolnego poziomu (9.53) - (9.54) przy wartości zmiennych dualnych $\bar{u}^P, \bar{\lambda}^P$. Uwzględnienie warunku (9.51) wymaga zastosowania projekcji otrzymanych wartości $\bar{u}, \bar{\lambda}$ na zbiór $U = \{ \langle \bar{u}, \bar{\lambda} \rangle | \bar{u} \geq 0, \bar{\lambda} \geq 0 \}$, tzn. wyznaczania takich wartości $\hat{u}, \hat{\lambda}$, że

$$\sum_{k=1}^m \sum_{t=t_0+1}^T (u_{tk} - \hat{u}_{tk})^2 + \sum_{\langle i,j \rangle \in \mathcal{R}_{\mu\pi}} (\lambda_{ij} - \hat{\lambda}_{ij})^2 \rightarrow \min, \quad (9.60)$$

$$\hat{u}_{tk} \geq 0, \hat{\lambda}_{ij} \geq 0. \quad (9.61)$$

Rozwiązanie zagadnienia (9.60) - (9.61) można znaleźć z warunków Kuhna-Tackera i ma ono postać

$$\hat{u}_{tk} = \max [0, u_{tk}] \quad \begin{matrix} t = t_0+1, \dots, T \\ k = 1, \dots, m \end{matrix}, \quad (9.62)$$

$$\hat{\lambda}_{ij} = \max [0, \lambda_{ij}] \quad \langle i,j \rangle \in \mathcal{R}_{\mu\pi}. \quad (9.63)$$

Uwzględniając wzory (9.57) - (9.59) i (9.62) - (9.63) otrzymujemy końcową postać wzorów ciągu $(\bar{u}^P, \bar{\lambda}^P)$.

$$\begin{aligned} \hat{u}_{tk}^{p+1} &= \max [0, u_{tk}^p + E^P (g_{tk}(\bar{S}^P, \bar{\mu}^P) - 1)] , \\ &\quad t = t_0+1, \dots, T , \\ &\quad k = 1, 2, \dots, m , \end{aligned} \quad (9.64)$$

$$\begin{aligned} \lambda_{ij}^{p+1} &= \max [0, \lambda_{ij}^p + E^P (s_i^p + p_i \mu^p(i) - s_j^p)] \\ &\quad \langle i,j \rangle \in \mathcal{R}_{\mu\pi}. \end{aligned} \quad (9.65)$$

Szczegółowy schemat algorytmu wyznaczania dolnego ograniczenia przedstawiono na rys. 9.9.

Własność 9.2.

Algorytm dolnego poziomu posiada złożoność obliczeniową pseudowielomianową $O(n(T-t_0)\gamma)$, gdzie $\gamma = \max_j |M^j|$

Dowód

Z własności 8.10 wynika, że wyznaczenie wartości $L_{ik}(S_i, k, \bar{u}, \bar{\lambda})$ ze wzoru (9.54) dla ustalonego i oraz k wymaga $O(T-t_0)$ iteracji. Wyznaczenie wszystkich wartości $L_{ik}(S_i, k, \bar{u}, \bar{\lambda})$ wymaga co najwyżej γ razy więcej iteracji. ■

algorytm górnego poziomu

czy $\bar{\mu}^p, \bar{S}^p$ wyznaczają rozwiązanie dopuszczalne problemu (9.36) - (9.40), tzn.

$$g_{tk}(\bar{S}^p, \bar{\mu}^p) \leq 1 \quad t=t_0+1, \dots, T, \quad k=1, 2, \dots, m$$

$$\sum_{k=1}^m \sum_{t=t_0+1}^T u_{tk} (g_{tk}(\bar{S}^p, \bar{\mu}^p) - 1) = 0$$

$$s_{i+p_i p_i}^p - s_j^p \leq 0 \quad \langle i, j \rangle \in \mathcal{R}_{\mu\pi}$$

$$\sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} \lambda_{ij} (s_{i+p_i p_i}^p - s_j^p) = 0$$

T

N

$\bar{\mu}^p$ wyznaczają optymalny przydział
zaś $\bar{S}^p$ optymalne momenty
rozpoczęcia operacji

STOP

na podstawie $\bar{\mu}^p, \bar{S}^p$ wyznacz
algorytmem heurystycznym
wartość górnego ograniczenia UB^p .

UB^p UB

N

T

UB := UB^p

wyznacz
 $\gamma^p = \theta(\gamma^{p-1}, p-1)$

oblicz

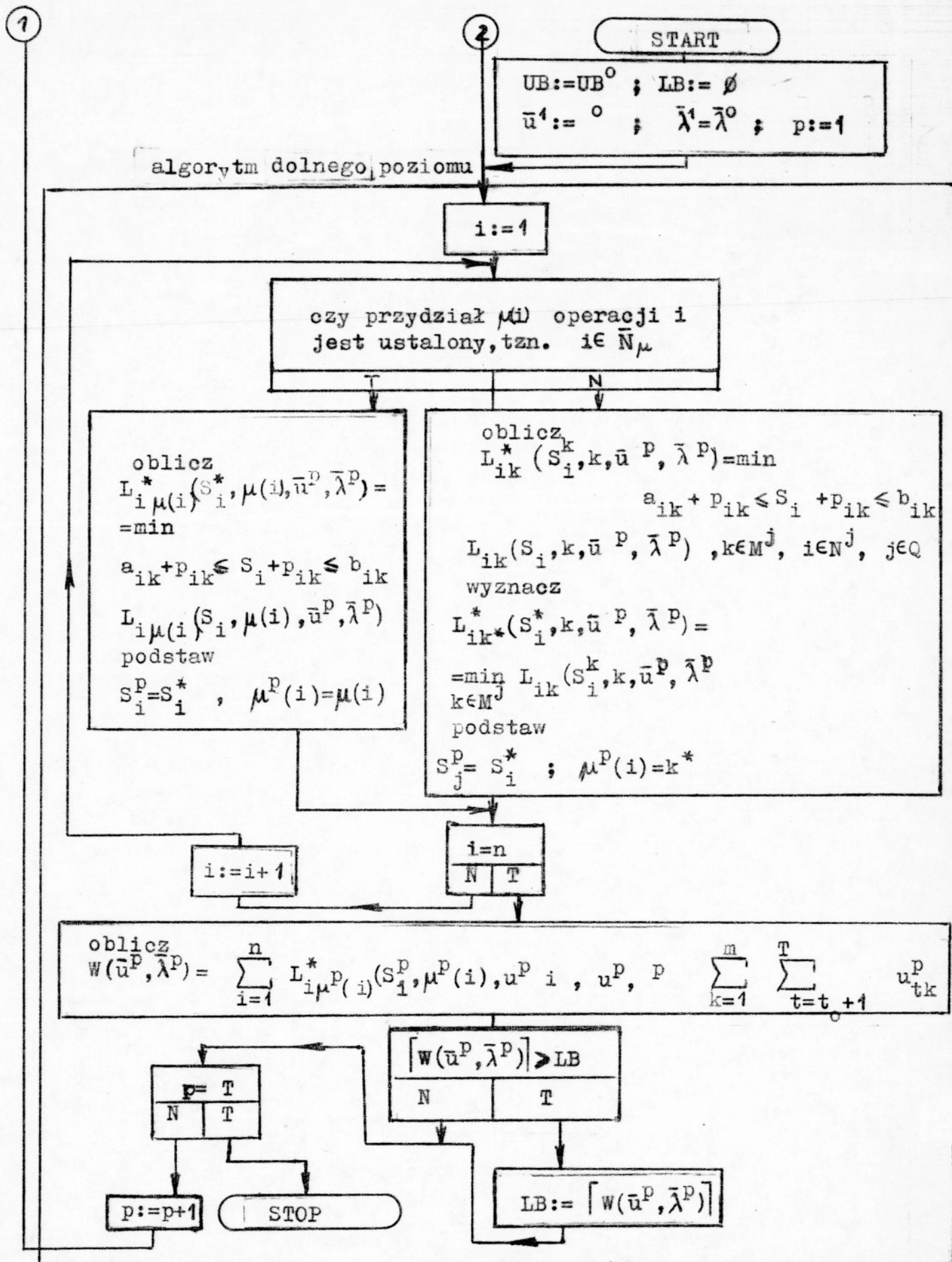
$$E^p = \gamma^p \sqrt{\frac{UB - W(u^p, \bar{\lambda}^p)}{\sum_{k=1}^m \sum_{t=t_0+1}^T (g_{tk}(s^p, \mu^p) - 1)^2 + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} (s_{i+p_i p_i}^p - s_j^p)^2}}$$

$$u_{tk}^{p+1} = \max [0, u_{tk}^p + E^p (g_{tk}(\bar{S}^p, \bar{\mu}^p) - 1)]$$

$$t=t_0+1, \dots, T, \quad k=1, 2, \dots, m$$

$$\lambda_{ij}^{p+1} = \max [0, \lambda_{ij}^p + E^p (s_{i+p_i p_i}^p - s_j^p)]$$

$$\langle i, j \rangle \in \mathcal{R}_{\mu\pi}$$



Rys. 9.9 Szczegółowy schemat blokowy dwupoziomowego algorytmu z rys. 9.8.

Dolne ograniczenie LB2. Niech będą dane ciągi liczb $\langle t_{0k}, t_{1k}, \dots, t_{s_k k} \rangle$ takie, że w problemie (9.54) - (9.55) wyprowadzono relaksację postaci

$$u_{tk} = u_{t_j k} \quad t_{j-1} < t \leq t_j, \quad j = 1, 2, \dots, s_k, \quad k = 1, 2, \dots, m. \quad (9.66)$$

Tak przyjęta relaksacja jest analogiczna do relaksacji 8.4 z rozdziału 8.6. W celu otrzymania odpowiednich wzorów dla procedury subgradientowej należy dokonać projekcji zmiennych $\bar{u}, \bar{\lambda}$ wyznaczonych ze zbiorów (9.57) - (9.58) na zbiór $U' = \{ \langle \bar{u}, \bar{\lambda} \rangle \mid u_{tk} = u_{t_j k} \geq 0, t_{j-1} < t \leq t_j, j = 1, 2, \dots, s_k, k = 1, 2, \dots, m \}$. Ze względu na warunek (9.66) można zmniejszyć liczbę zmiennych dualnych. Oznaczmy nowo wprowadzone zmienne $\hat{u}_{jk}$ $j = 1, 2, \dots, s_k$, $k = 1, 2, \dots, m$. Podana projekcja wymaga więc wyznaczenia wartości $\hat{u}_{jk}$ $j = 1, 2, \dots, s_k$, $k = 1, 2, \dots, m$, λ_{ij} $\langle i, j \rangle \in \mathcal{R}_{\mu\pi}$ spełniających

$$\sum_{k=1}^m \sum_{j=1}^{s_k} \sum_{t=t_{j-1}+1}^{t_j} (u_{tk} - \hat{u}_{jk})^2 + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} (\lambda_{ij} - \hat{\lambda}_{ij})^2 \rightarrow \min, \quad (9.67)$$

$$\hat{u}_{jk} \geq 0, \quad \hat{\lambda}_{ij} > 0. \quad (9.68)$$

Rozwiązanie możemy otrzymać z warunków Kuhna-Tackera i ma ono postać

$$\hat{u}_{jk} = \max \left[0, \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} u_{tk} \right] \quad (9.69)$$

$$j = 1, 2, \dots, s_k, \quad k = 1, 2, \dots, m,$$

$$\lambda_{ij} = \max [0, \lambda_{ij}]. \quad (9.70)$$

Podstawiając otrzymane wyniki do wzorów (9.57) - (9.58) otrzymujemy

$$\begin{aligned} \hat{u}_{jk}^{p+1} &= \max \left[0, \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} u_{tk}^{p+1} \right] = \\ &= \max \left[0, \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} \left[u_{tk}^p + E^p(g_{tk}(\bar{S}^p, \bar{\mu}^p) - 1) \right] \right] = \end{aligned}$$

$$\begin{aligned}
&= \max \left[0, \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} u_{tk}^p + E^p \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} g_{tk}(\bar{s}^p, \bar{\mu}^p) - \right. \\
&\quad \left. - \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} 1 \right] = \max \left[0, \hat{u}_{jk}^p + E^p(\hat{g}_{jk}(\bar{s}^p, \bar{\mu}^p) - 1) \right], \quad (9.71)
\end{aligned}$$

gdzie

$$\hat{g}_{jk}(\bar{s}^p, \bar{\mu}^p) = \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} g_{tk}(\bar{s}^p, \bar{\mu}^p). \quad (9.72)$$

Końcowy wzór procedury subgradientowej przyjmuje postać

$$\hat{u}_{jk}^{p+1} = \max \left[0, \hat{u}_{jk}^p + E^p(\hat{g}_{jk}(\bar{s}^p, \bar{\mu}^p) - 1) \right], \quad (9.73)$$

$$\hat{\lambda}_{ij}^{p+1} = \max \left[0, \hat{\lambda}_{ij}^p + E^p(s_i^p + p_i \mu_{p(i)} - s_j^p) \right], \quad (9.74)$$

gdzie $\hat{g}_{jk}(\bar{s}^p, \bar{\mu}^p)$ jest średnim obciążeniem maszyny k w przedziale $[t_{j-1,k}+1, t_{jk}]$.

Można zauważyć, że analogiczny rezultat otrzymany wprowadzając relaksację problemu pierwotnego (9.36) - (9.42) w postaci

$$\min_{\substack{\bar{s} \in \mathcal{S} \\ \bar{\mu} \in \mathcal{M}}} \sum_{i=1}^n h_i \mu(i)(s_i), \quad (9.75)$$

$$\hat{g}_{jk}(\bar{s}, \bar{\mu}) \leq 1, \quad (9.76)$$

$$s_i + p_i \mu(i) \leq s_j, \quad (9.77)$$

gdzie

$$\hat{g}_{jk}(\bar{s}, \bar{\mu}) = \frac{1}{t_{jk} - t_{j-1k}} \sum_{t=t_{j-1k}+1}^{t_{jk}} g_{tk}(\bar{s}, \bar{\mu}) \quad (9.78)$$

$$j = 1, 2, \dots, s_k, \quad k = 1, 2, \dots, m$$

są ograniczeniami otrzymanymi z ograniczeń przez wysumowanie części ograniczeń $g_{tk}(\bar{s}, \bar{\mu})$, $t \in [t_{j-1k}+1, t_{jk}]$.

Formułując odpowiednio problem dualny otrzymujemy następujące wzory procedury subgradientowej

$$\hat{u}_{jk}^{p+1} = \max [0, \hat{u}_{jk}^p + \hat{E}^p (\hat{g}_{jk}(\bar{S}^p, \bar{\mu}^p) - 1)] \quad (9.79)$$

$$j = 1, 2, \dots, l_k, k = 1, 2, \dots, m,$$

$$\hat{\lambda}_{ij}^{p+1} = \max [0, \hat{\lambda}_{ij}^p + \hat{E}^p (s_i^p + p_i \mu^{p(i)} - s_j^p)] , \quad (9.80)$$

gdzie

$$\hat{E}^p = \frac{UB - W(\bar{u}^p, \bar{\lambda}^p)}{\sqrt{\sum_{k=1}^m \sum_{j=1}^{l_k} (\hat{g}_{jk}(\bar{S}^p, \bar{\mu}^p) - 1)^2 + \sum_{\langle i, j \rangle \in \mathcal{R}_{\mu\pi}} (s_i^p + p_i \mu^{p(i)} - s_j^p)^2}} \quad (9.81)$$

Zauważmy, że w ogólnym przypadku $\hat{E}_p \neq E_p$ /wzory (9.59) i (9.81)/ oraz obie podane relaksacje, problemu dualnego i pierwotnego, prowadzą do otrzymania tej samej postaci wzorów /z dokładnością do współczynnika $\hat{E}_p$ /. Problem, która z podanych relaksacji pozwoli osiągnąć korzystniejsze rezultaty numeryczne może być rozwiązany na drodze eksperymentalnej.

Zauważmy, że stosując podejście analogiczne, jak w relaksacji 8.5 możemy spowodować odrzucenie ograniczenia na całkowitoliczowość zmiennych r_{ik} , p_{ik} , d_{ik} , t_{jk} bez straty ogólności rozważań. Ceną takiego postępowania będzie wzrost złożoności obliczeniowej odpowiednich algorytmów.

9.5. Zasady podziału

Podział zbioru rozwiązań związanego z węzłem w $\hat{H}$ jest dokonywany poprzez

- (i) wybór operacji oraz zmianę jej przydziału /podział zbioru rozwiązań w drzewie H /,
- (ii) wybór operacji oraz zmianę kolejności realizacji operacji na jedną z maszyn /podział zbioru rozwiązań w drzewie $\bar{H}$ /.

Niech $\bar{\mu} \in \mathcal{M}$ będzie przydziałem. Zdefiniujemy zbiór operacji, których przydział do maszyn jest ustalony w węźle $\bar{\mu}$

$$\bar{N}_{\bar{\mu}} \triangleq \{i \mid \mu(i) \text{ ustalony chwilowo lub trwale}\}.$$

Będziemy dokonywali zmiany przydziału operacji należących do zbioru $N - \bar{N}_{\bar{\mu}}$. Oznaczmy przez

$$E_{\bar{\mu}}^i \triangleq \{k \in M^j\}, \quad i \in N^j - \bar{N}_{\bar{\mu}}, \quad j \in Q \quad (9.82)$$

zbiór maszyn, do których możemy przydzielić operację i -tą. Wybór operacji oraz maszyny /do której ją przydzielimy/ będzie dokonywany na podstawie wyrażenia

$$\begin{aligned} LB''(i,k) &= W(\bar{u}^p, \bar{\lambda}^p) - L_{i\bar{\mu}}^*(S_i^p, \mu^p(i), \bar{u}^p, \bar{\lambda}^p) \\ &+ L_{ik}^*(S_i^*, k, \bar{u}^p, \bar{\lambda}^p) \geq W(\bar{u}^p, \bar{\lambda}^p), \quad k \in E_{\bar{\mu}}^i, \quad i \in N \end{aligned} \quad (9.83)$$

gdzie $W(\bar{u}^p, \bar{\lambda}^p)$ - wartość problemu Lagrange'a w węźle μ ,
 $L_{i\bar{\mu}}^*(S_i^p, \mu^p(i), \bar{u}^p, \bar{\lambda}^p)$ - minimalna wartość częściowego problemu Lagrange'a wyznaczona dla operacji i -tej / $i \in N - \bar{N}_{\bar{\mu}}$ /,
 $L_{ik}^*(S_i^*, k, \bar{u}^p, \bar{\lambda}^p)$ - wartość częściowego problemu Lagrange'a przy założeniu, że operacja i -ta została przydzielona do maszyny k -tej,
 p - numer iteracji procedury subgradientowej. Wartość $LB''(i,k)$ jest wyznaczana dla ostatniej iteracji procedury subgradientowej w węźle μ , a następnie zaokrąglana "w górę" do najbliższej liczby całkowitej.

Zmianę kolejności wykonywania operacji będziemy przeprowadzać w tych węzłach drzewa, w których przydziały wszystkich operacji zostały ustalone tzn. $\bar{N}_{\bar{\mu}} = N$, $\bigcup_{i \in N} E_{\bar{\mu}}^i = \emptyset$. Przyjęcie takiej zasady postępowania jest uzasadnione tym, że zmiana przydziału operacji daje zwykle lepsze rozwiązanie niż zmiana kolejności wykonywania

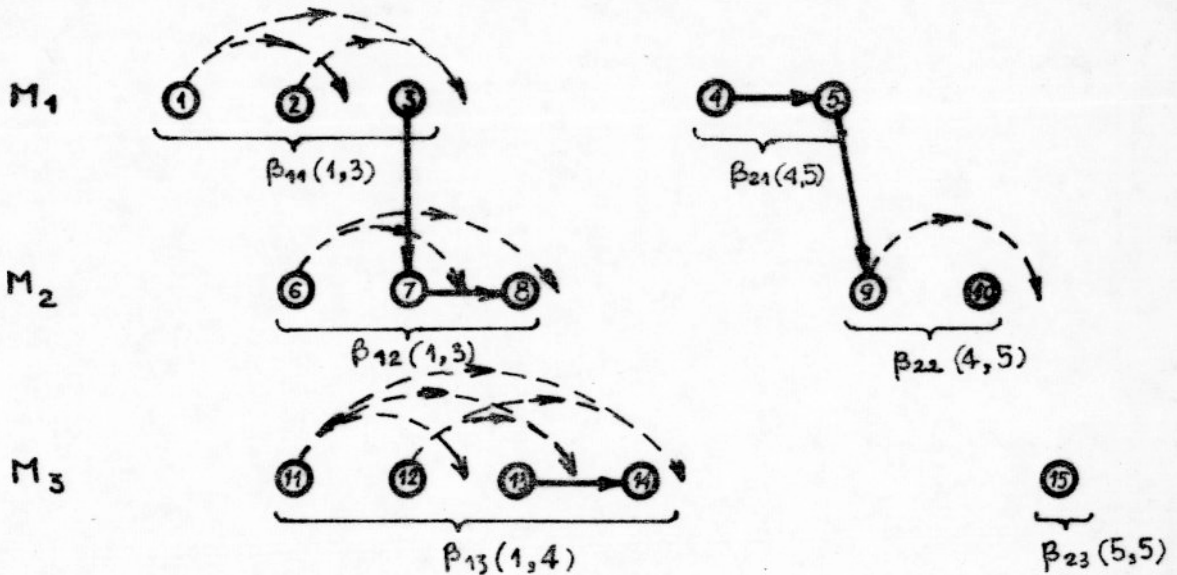
operacji. Będziemy zmieniać kolejność wykonywania operacji należących do zbiorów

$$E_{\pi}^{sk} = \{i \in N_{k\mu} \mid i_{j^*k} < i < s \leq i_{j^*+1,k},$$

$$\langle \pi_k(i), \pi_k(h) \rangle \in \mathcal{R}_{\mu\pi} \quad h = i+1, i+2, \dots, s \} \quad (9.84)$$

$$s = 2, \dots, n_k, \quad k = 1, 2, \dots, m$$

gdzie E_{π}^{sk} - zbiór kandydatów do przesunięcia na pozycję s -tą w permutacji π_k , $\mathcal{R}_{\mu\pi}$ - relacja poprzedzania w zbiorze operacji ustalona w węźle odpowiadającym przydziałowi $\bar{\mu}$ i permutacji $\bar{\pi}$. Dla przykładu przedstawionego na rys. 9.10



$$n_{1\mu} = 5 \quad \pi_1 = \langle 1, 2, 3, 4, 5 \rangle$$

$$n_{2\mu} = 5 \quad \pi_2 = \langle 6, 7, 8, 9, 10 \rangle$$

$$n_{3\mu} = 5 \quad \pi_3 = \langle 11, 12, 13, 14, 15 \rangle$$

$$\mathcal{R}_{\mu\pi} = \{ \langle 3, 7 \rangle, \langle 7, 8 \rangle, \langle 5, 9 \rangle, \langle 13, 14 \rangle \}$$

Rys. 9.10. Sposób generowania zmian kolejności wykonywania operacji dla ustalonego przydziału /zaznaczono możliwe przeniesienia operacji $\rightarrow$ oraz relację $\mathcal{R}_{\mu\pi} \rightarrow /$

zbiorami kandydatów są odpowiednio $E_{\pi}^{21} = \{1\}$, $E_{\pi}^{31} = \{1, 2\}$, $E_{\pi}^{41} = E_{\pi}^{51} = \emptyset$, $E_{\pi}^{22} = E_{\pi}^{32} = \{6\}$, $E_{\pi}^{42} = \emptyset$, $E_{\pi}^{52} = \{9\}$, $E_{\pi}^{23} = \{11\}$, $E_{\pi}^{33} = E_{\pi}^{43} = \{11, 12\}$, $E_{\pi}^{53} = \emptyset$. Będziemy dążyć do wyboru takiej operacji, której przesunięcie wygeneruje następnik najbardziej

perspektywiczny /posiadający najmniejsze dolne ograniczenie/.

W celu oceny kandydatów wprowadzona została wielkość $LB^i(i, s)$

$$LB^i(i, s) = W(\bar{u}^P, \bar{\lambda}^P) - L_{i\mu(i)}^*(S_i^P, \mu(i), \bar{u}^P, \bar{\lambda}^P) + \\ + L_{i\mu(i)}^*(S_i^*, \mu(i), \bar{u}^P, \bar{\lambda}^P) \geq W(\bar{u}^P, \bar{\lambda}^P) \quad (9.85)$$

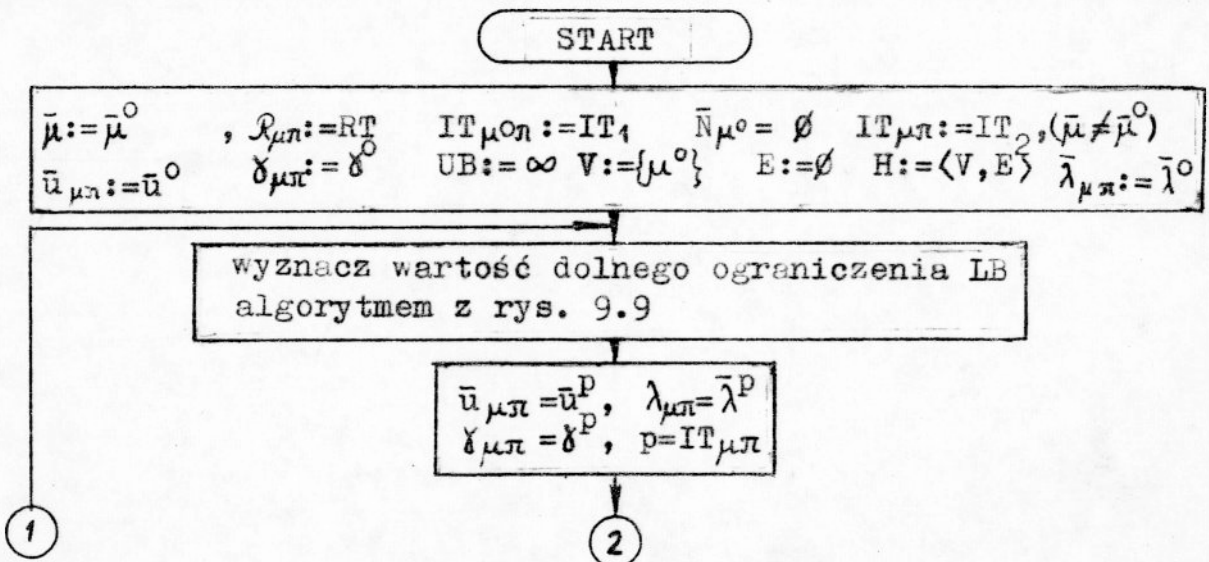
gdzie $W(\bar{u}^P, \bar{\lambda}^P)$ - wartość problemu Lagrange'a w węźle $\bar{\pi}$ /z ustalonym przydziałem $\bar{\mu}$ /, $L_{i\mu(i)}^*(S_i^*, \mu(i), \bar{u}^P, \bar{\lambda}^P)$ - wartość częściowego problemu Lagrange'a oraz

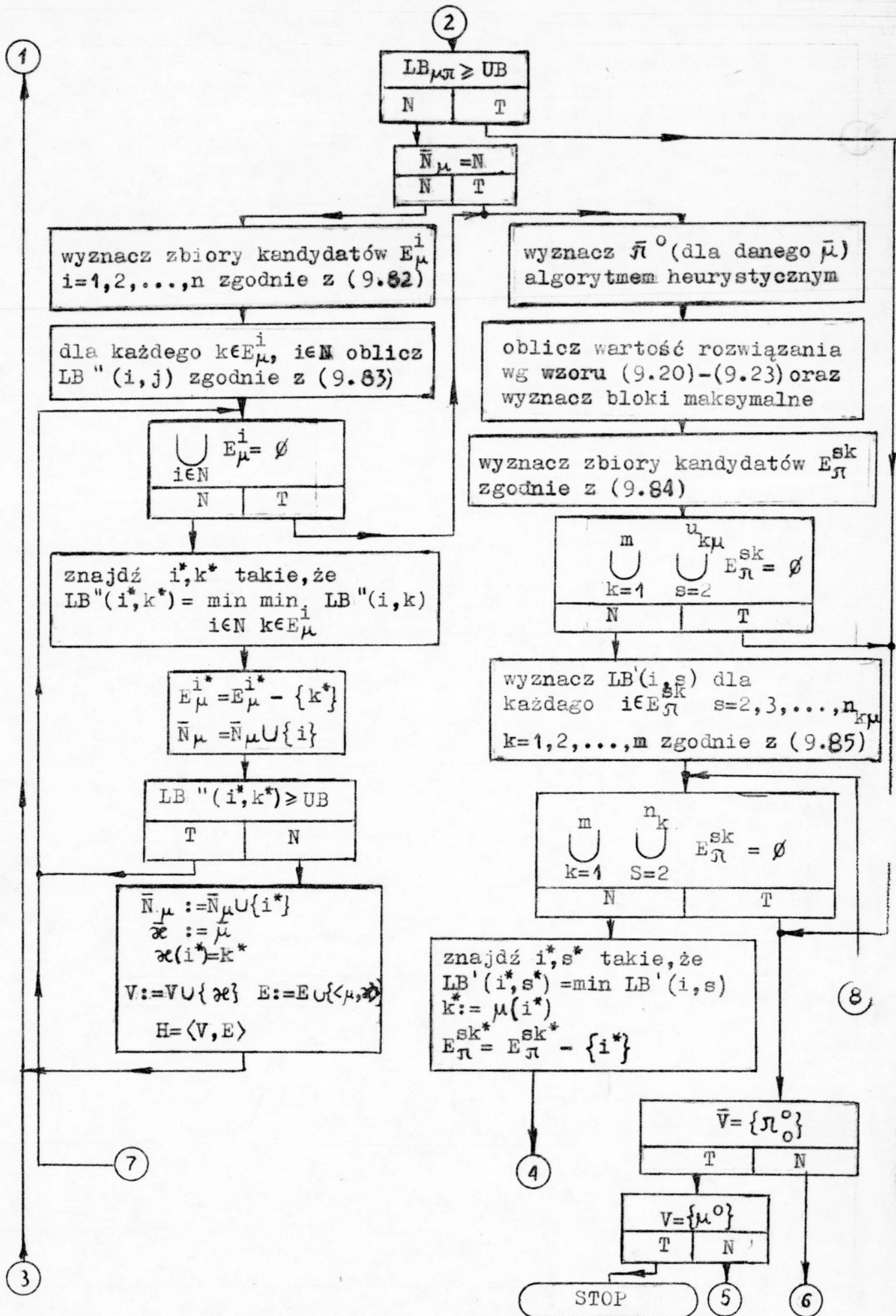
$$L_{i\mu(i)}^*(S_i^*, \mu(i), \bar{u}^P, \bar{\lambda}^P) = \min_{a_{i\mu(i)} + p_{i\mu(i)} \leq S_i^* + p_{i\mu(i)} \leq b_{i\mu(i)}} L_{i\mu(i)}(S_i, \mu(i), \bar{u}^P, \bar{\lambda}^P) \quad (9.86)$$

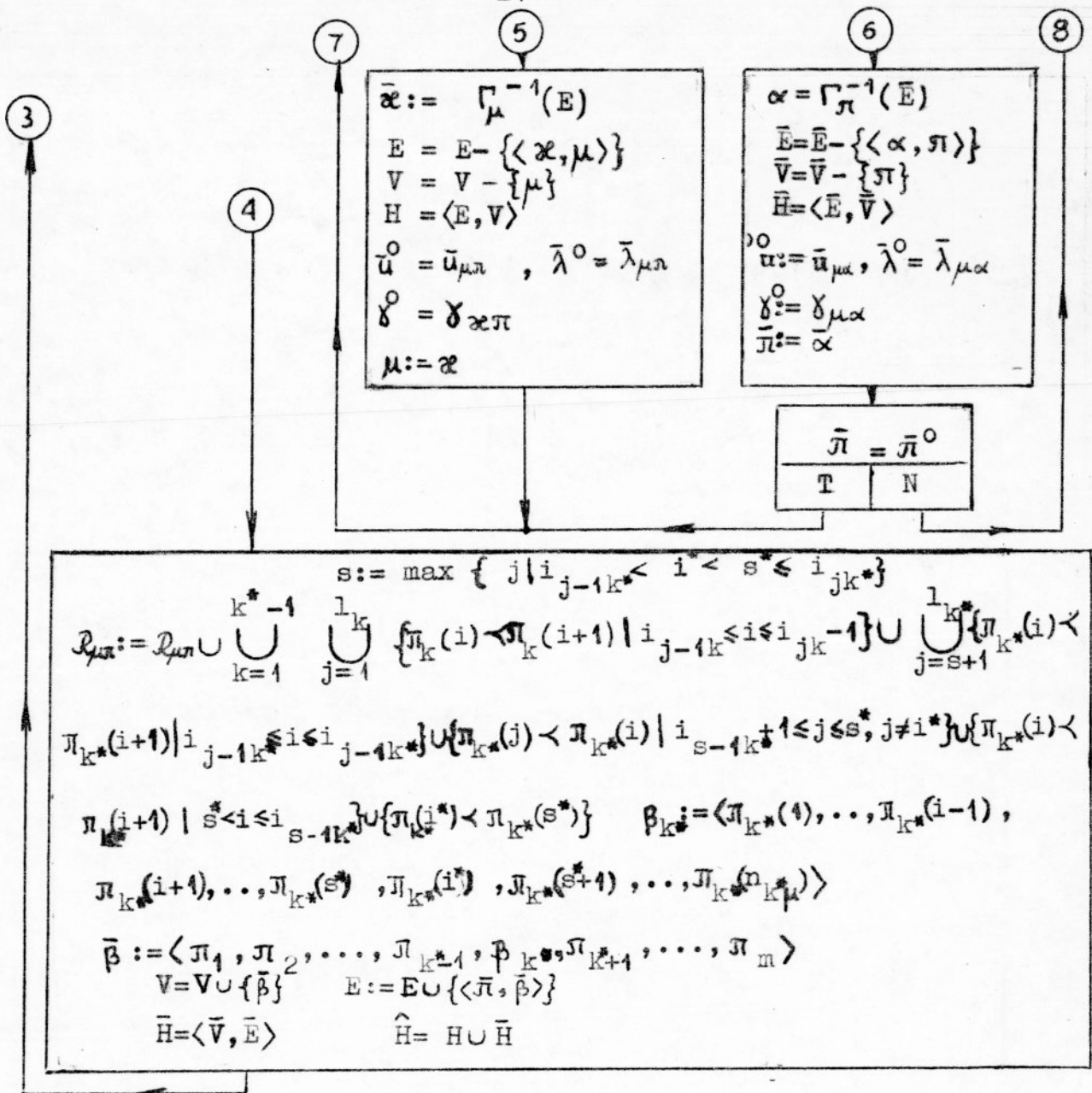
Wartości $a_{i\mu(i)}$, $b_{i\mu(i)}$ są odpowiednio najwcześniejszym momentem rozpoczęcia i najpóźniejszym momentem zakończenia operacji i w permutacji $\bar{p}$ otrzymanej z $\bar{\pi}$ przez przesunięcie operacji i -tej na pozycję s -tą w $\pi_{\mu(i)}$. Wartość $LB^i(i, s)$ jest wyznaczona dla ostatniej iteracji procedury subgradientowej w węźle $\bar{\pi}$, a następnie zaokrąglana "w górę" do najbliższej liczby całkowitej.

9.6. Algorytm

Niech IT_1 , IT_2 będą danymi liczbami iteracji procedury subgradientowej, zaś δ^0 danym skalar. Niech $R_{\mu\pi}$, $IT_{\mu\pi}$, $\bar{u}_{\mu\pi}$, $\bar{\lambda}_{\mu\pi}$, $\delta_{\mu\pi}$ będą odpowiednio ustaloną relacją, liczbą iteracji testu, wartościami zmiennych dualnych, skalar. w węźle drzewa rozwiązań odpowiadającym przydziałowi $\bar{\mu}$ i wektorowi permutacji $\bar{\pi}$.







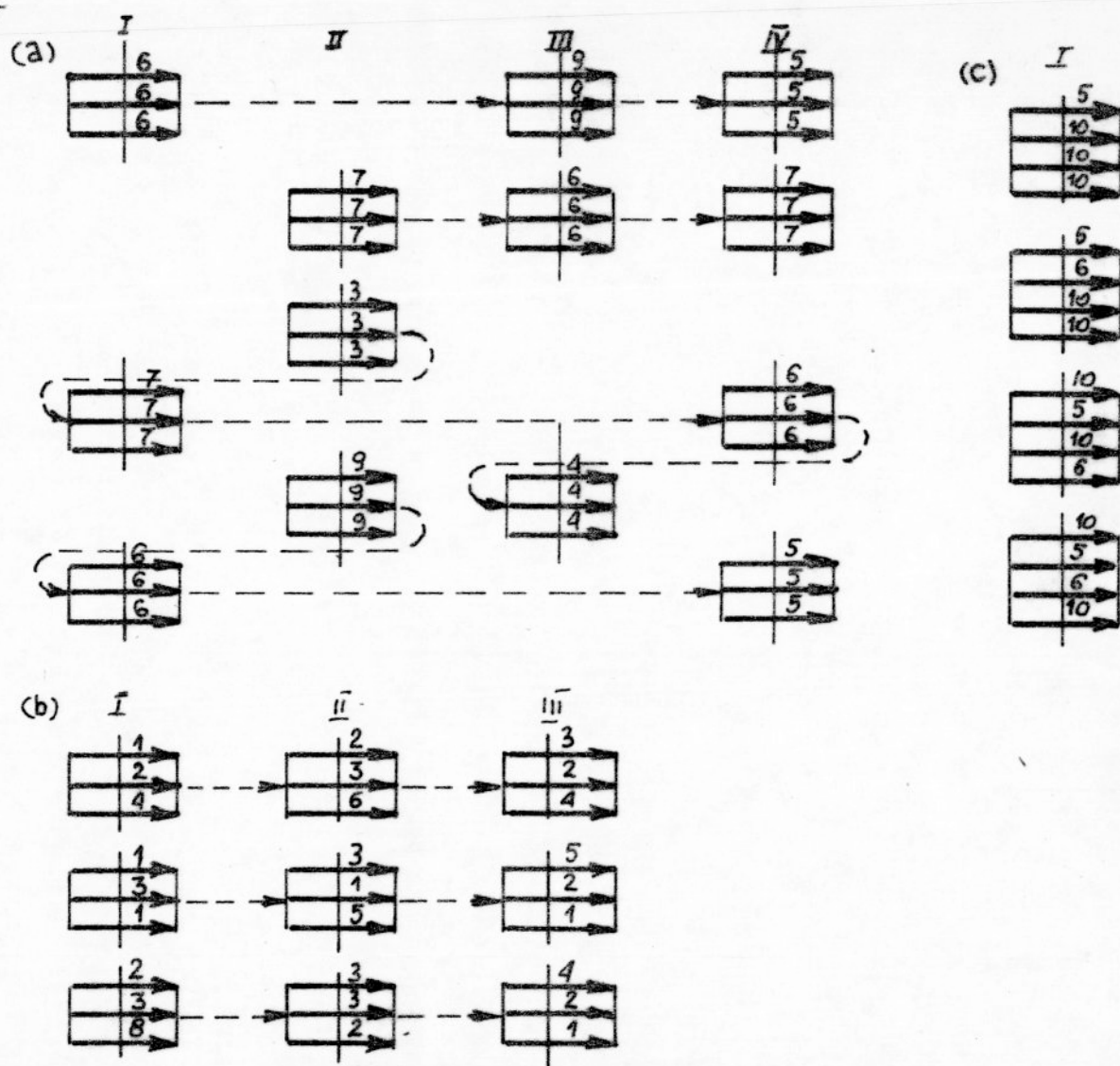
Rys. 9.11. Schemat blokowy algorytmu

9.7. Eksperymenty obliczeniowe

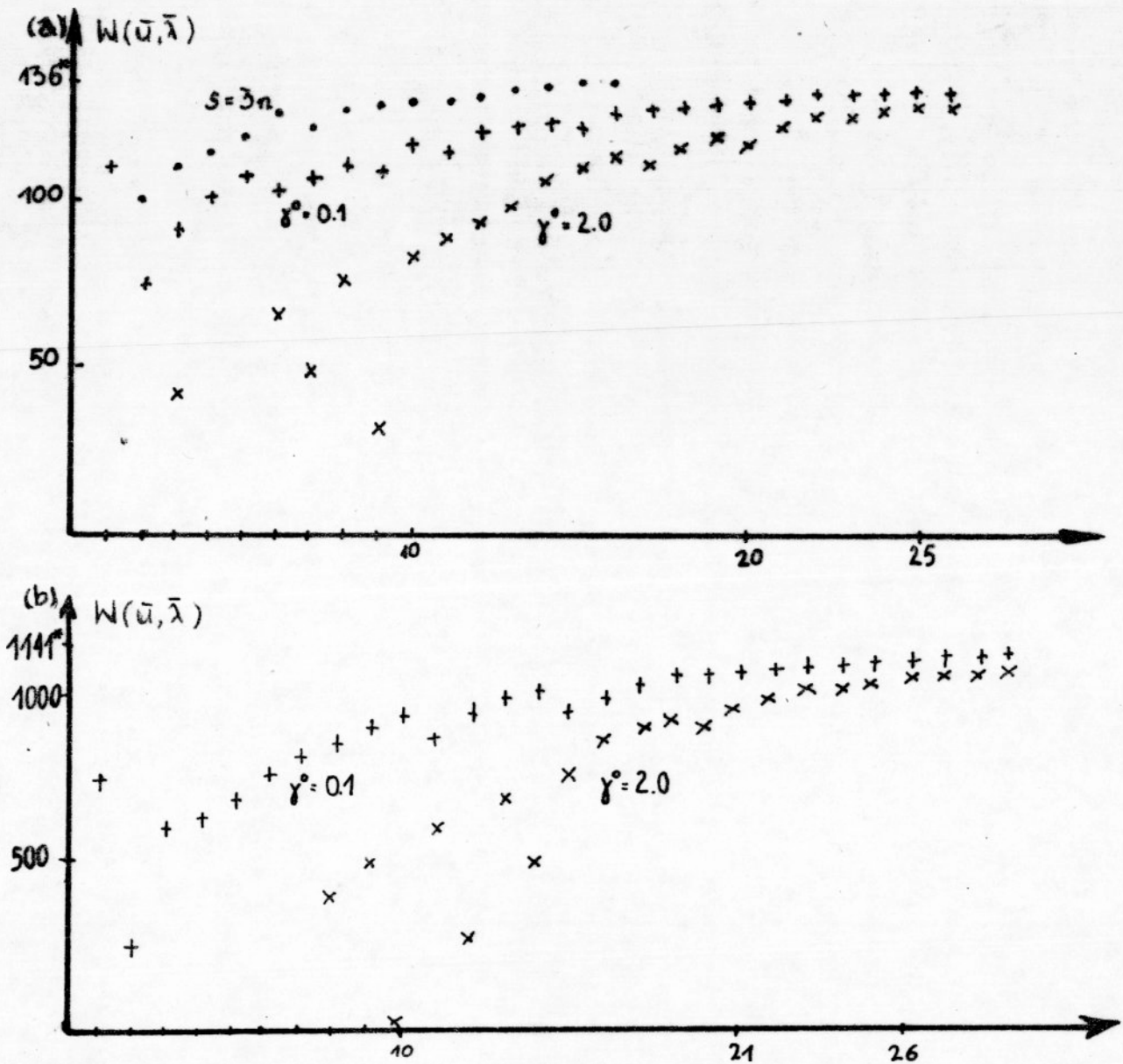
Przedstawiony algorytm został zaprogramowany w języku ALGOL 1900 i był testowany na maszynie ODRA 1305.

Implementacja programowa została zrealizowana przy założeniu funkcji kar w postaci $g_{ik}(E_i) = 0$, $E_i = 0$; $g_{ik}(E_i) = \infty$ $E_i > 0$, $f_{ik}(T_i)$, $w_i \geq 0$, $i \in N^j$, $k \in M^j$, $j \in Q$ oraz przy założeniu funkcji kar w postaci $g_{ik}(E_i) = v_i E_i$, $v_i \geq 0$, $f_{ik}(T_i) = w_i T_i$, $w_i \geq 0$, $i \in N^j$, $k \in M^j$, $j \in Q$. W drugim przypadku rozwiązania optymalne /dla danego $\bar{\mu}$, $\bar{\pi}$ / wyznaczono algorytmem dokładnym -

w przypadku, gdy graf relacji $R_{\mu n}$ był drzewem oraz algorytmem przybliżonym - w pozostałych przypadkach. W programie zastosowano zrelaksowaną postać testu (LB2) w celu zmniejszenia nakładów obliczeniowych algorytmu. Eksperymenty obliczeniowe przeprowadzono na przykładach, dla których struktura została zaczerpnięta z literatury, zaś dane liczbowe wylosowane. Uzyskane wyniki przedstawiono na rys. 9.12 - 9.13.



Rys.9.12. Przykłady obliczeniowe. (a) problem $n|m|J, \tau \geq 1, m_w^k \geq 1, r_j \geq 0, RT|\sum w_i C_i [71]$ (b) problem $n|m|J, \tau \geq 1, m_w^k \geq 1, r_j \geq 0, RT|\sum w_i T_i [3]$, (c) problem $n|m|J, \tau = 1, m_w^k \geq 1, RT|\sum w_i E_i + w_i T_i [3]$. Przykłady rozszerzono przyjmując losowe wartości r_i z przedziału $[0, \sum_{i \in N} p_{iv}]$ gdzie $p_{iv} = \min_{k \in M_j} p_{ik}$, $i \in N^j, j \in Q$, losowe wartości $d_i [0, \sum p_{iv}]$ oraz $w_i [1, 5]$.



(c)

przykład	l. węzłów	l. iter. testu	czas obl. [s]
a, $m_{\frac{w}{k}}^k=1$	2	31	109
a, $m_{\frac{w}{k}}^k=2$	1	28	142
a, $m_{\frac{w}{k}}^k=3$	1	28	193
b	1	28	208
c	8	49	600

ODRA 1305

Rys. 9.13. Wyniki obliczeń. (a) wartość LB3 w przykładzie a, (b) wartość LB3 w problemie b, (c) wyniki obliczeń.

9.8. Uwagi i wnioski

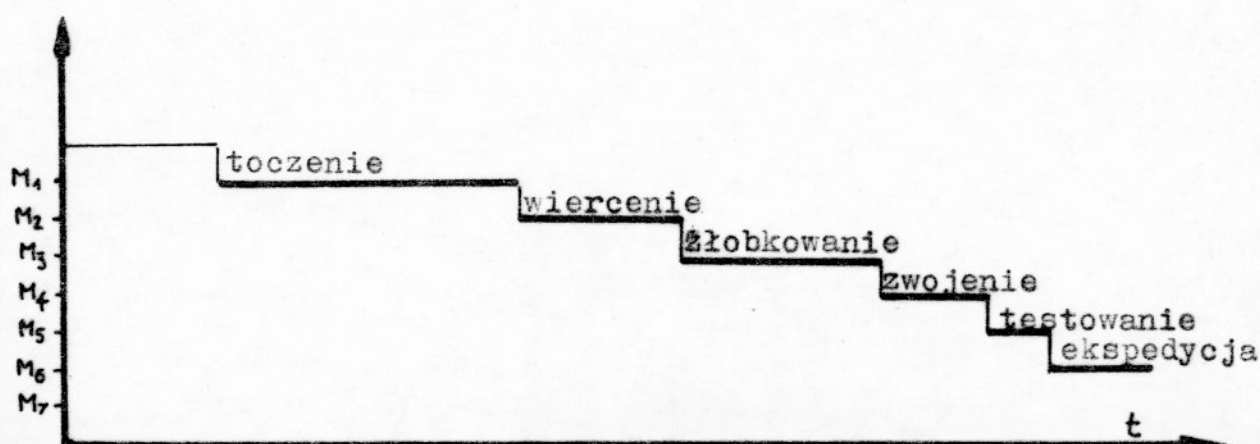
Z analizy pracy algorytmu wynika, że procedura wyznaczania wartości dolnego ograniczenia "preferuje" spełnienie ograniczeń związanych z relacją $R_{\mu n}$ w stosunku do ograniczeń przepustowości maszyn. Jest to korzystne, bowiem możliwe jest stosunkowo szybkie znalezienie rozwiązania dopuszczalnego. Należy zauważyć, że uwzględnienie możliwości alternatywnych przydziałów operacji do maszyn tylko nieznacznie zwiększyło złożoność algorytmu problemu Lagrange'a przy jednoczesnym uzyskaniu korzystniejszych wyników obliczeniowych dolnego ograniczenia/ rys.9.1/. We wszystkich badanych przykładach zaobserwowano znaczny spadek wartości funkcji celu w stosunku do wartości LB w korzeniu drzewa rozwiązań. Wartość dolnego ograniczenia wahała się w granicach 50 - 80 % najlepszego znalezionej rozwiązania. Dalszą poprawę wyników można by osiągnąć poprzez zwiększenie nakładów obliczeniowych algorytmu. Jest to bezpośrednio związane ze wzrostem nakładów obliczeniowych procedury dolnego ograniczenia, która zajmuje ponad 80 % czasu działania całego algorytmu. Eksperymenty obliczeniowe wskazują na możliwość praktycznego zastosowania przedstawionego algorytmu do rozwiązywania średniego rozmiaru /rzędu $n = 20$ /. Rozwiązywanie zagadnień o większym rozmiarze wymaga dalszych badań własności algorytmu pod kątem zmniejszenia nakładów obliczeniowych.

10. ZASTOSOWANIA

A. Planowanie operatywne w ZPRE*. Rozważany jest przykład problemu optymalizacji procesu produkcyjnego w Zakładach Remontowych Energetyki. Wyrobami i półwyrobami są w tym przypadku pojedyncze elementy /części składowe turbin, silników, prądnic, generatorów, regulatorów lub ich partie. Niektóre z tych elementów to wirniki i łopatkę turbin, wirniki generatorów, silników i prądnic, obudowy silników

* opracowano na podstawie kart technologicznych z ZPRE Wrocław[23]

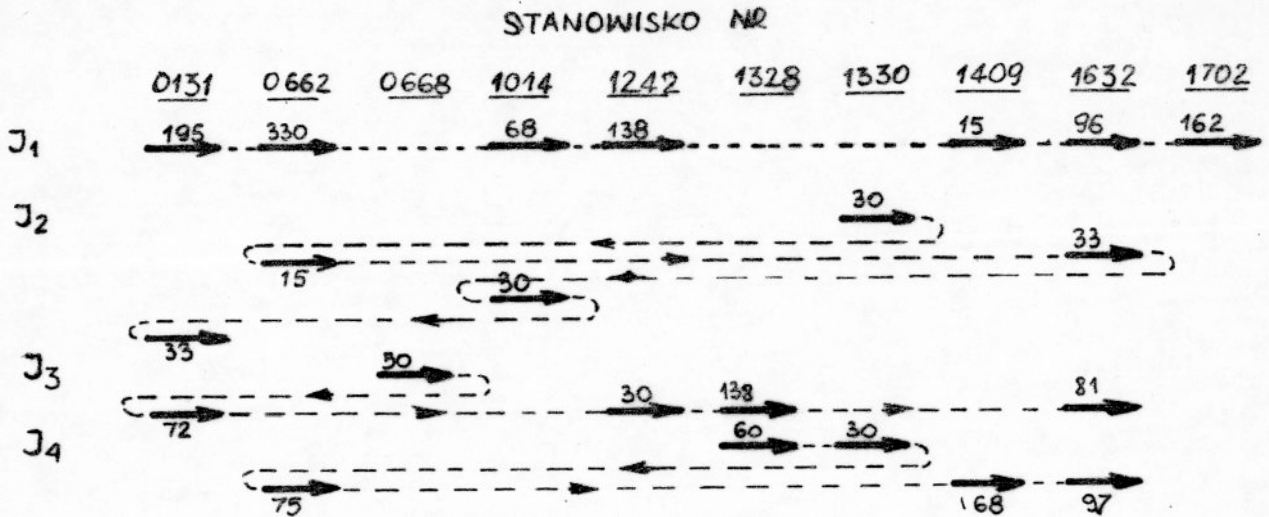
i prądnic wykonywane w ramach napraw podzespołów energetycznych lub produkcji roboczej. W ogólnym przypadku elementy te są różne. Wydział ZPRE ma gniazdową strukturę usytuowania agregatów technologicznych. Każde gniazdo składa się z jednej lub kilku maszyn o tym samym przeznaczeniu. Podstawowymi maszynami na wydziale są: frezarki, tokarki, heblarki, wiertarki, szlifierki. Niektóre spośród maszyn tego samego typu są przeznaczone do prac specjalizowanych /np. tokarki karuzelowe do toczenia dużych elementów turbin i generatorów, frezarki do frezowania łopatek turbin, heblarki do rowkowania wirników/ są więc faktycznie różne. Pozostałe maszyny tego samego typu są identyczne funkcjonalnie /zamienne/ lecz o różnej szybkości działania. Na wydziale oprócz agregatów występują stanowiska zwojowania silników i generatorów, montażu, kontroli i testowania oraz ekspedycji. Proces produkcyjny odznacza się "przepływem" poszczególnych elementów przez agregaty technologiczne. Elementy te poddaje się operacjom obróbki na kolejnych agregatach technologicznych. Czasy trwania poszczególnych operacji są znane i różne dla różnych operacji i agregatów. W ogólnym przypadku każdy z elementów przepływa przez agregaty w różnym porządku technologicznym. Porządek wykonywania operacji dla poszczególnych elementów jest zadany w postaci cyklogramów. Na rys. 10.1 przedstawiono cyklogram obróbki wirnika generatora małej mocy.



Rys. 10.1. Przykładowy cyklogram obróbki wirnika generatora

Każdy z elementów jest dostępny do obróbki w terminie r_i , który odpowiada terminowi przekazania odpowiedniego podzespołu do remontu w ZPRE.

Dodatkowo występuje wymaganie zwrotu wyremontowanego podzespołu w terminie d_i . Przekroczenie terminu d_i powoduje straty związane z wyłączeniem remontowanego podzespołu z ruchu i eksploatacji. Wielkość strat w ogólnym przypadku jest różna dla różnych podzespołów.



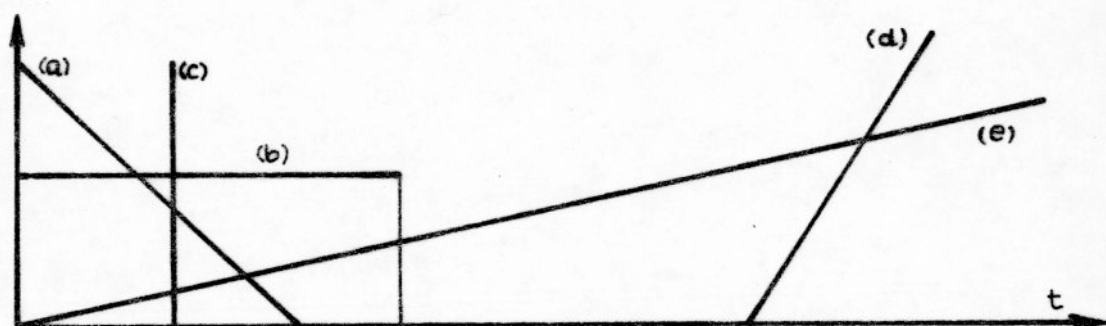
Rys. 10.2. Fragment zadania produkcyjnego w ZPRE.

Podano czas realizacji operacji w minutach.

Zwykle w praktyce przyjmuje się, że każdy z obrabianych elementów jest dostępny nie wcześniej niż w terminie r_i . W ogólnym przypadku możliwe jest jednak wcześniejsze niż zaplanowano wyłączenie podzespołu z eksploatacji i przekazanie go do remontu. Konsekwencją takiego sposobu postępowania będą dodatkowe straty, które należy ponieść. Jednakże w globalnym rachunku ekonomicznym może to być opłacalne, bowiem skrócony zostanie termin oddania innych podzespołów do eksploatacji po remoncie. Podejście tego rodzaju jest stosowane w energetyce, gdzie realizacja remontów i zmiany w instalacji są wykonywane w okresie największego spadku poboru mocy /maj-wrzesień/.

Problem minimalizacji całkowitych strat określonych powyżej, można sformułować jako problem $n|m|J, \tau \geq 1, m_w^k \geq 1, RT | \sum h_i$ i rozwiązać przy pomocy algorytmu z rozdz. 9.6.

B. Realizacja inwestycji. Realizacja inwestycji jest rozumiana jako planowanie czynności zadania inwestycyjnego składającego się z niepodzielnych podzadań realizowanych przez poszczególnych podwykonawców. Charakterystycznymi cechami zadania inwestycyjnego są: możliwość alternatywnego wyboru podwykonawców, różne terminy gotowości podwykonawców do rozpoczęcia realizacji podzadań, brak synchronizacji dostaw podzespołów i elementów montowanych o obiekcie inwestycyjnym, narzucone terminy zakończenia realizacji poszczególnych podzadań, możliwość przyspieszenia realizacji podzadań kosztem zwiększenia nakładów finansowych. Brak synchronizacji dostaw podzespołów i elementów powodują powstawanie kosztów związanych z żądaniem dostawy wcześniej niż to wynika z umownego terminu /koszt dodatkowej ekspedycji/. Całkowite koszty związane z realizacją pojedynczego podzadania można przedstawić następująco /rys. 10.3/:



Rys.10.3 . Funkcje kosztu realizacji podzadania inwestycyjnego; (a) - (c) - koszty przyspieszeń , (d) - (e) - koszty opóźnień

- (a) przyspieszenie realizacji i -tego podzadania inwestycyjnego o wartości E_i w stosunku do ustalonego terminu r_{ik} wymaga poniesienia kosztów $g_{ik}^1(E_i)$ na skrócenie czasów trwania czynności przygotowawczych oraz podzadań poprzedzających podzadanie i -te,
- (b) przyspieszenie ekspedycji /dodatkowa ekspedycja/ elementów montowanych w i -tym podzadaniu o wartości E_i w stosunku do ustalonego terminu dostawy r_{ik} wymaga poniesienia kosztów $g_{ik}^2(E_i)$,

- (c) niemożliwe jest przyspieszenie rozpoczęcia realizacji i -tego podzadania, bowiem k -ty podwykonawca nie zapewnia wejścia na obiekt inwestycyjny w terminie wcześniejszym niż r_{ik} , tzn. $g_{ik}^3(E_i) = 0$, gdy $E_i = 0$ oraz $g_{ik}^3(E_i) = M$, $E_i > 0$, w przypadku, gdy podwykonawca zgadza się przyspieszyć swoje wejście na obiekt za cenę $g_{ik}(E_i)$ to postać funkcji kosztu jest jak w (a),
- (d) spóźnienia realizacji i -tego podzadania o wartości T_i w stosunku do ustalonego terminu d_{ik} wymaga dodatkowych nakładów na utrzymanie na placu budowy sprzętu i elementów wykorzystywanych w innych podzadaniach,
- (e) użycie kapitałów, sprzętu i materiałów wymaga poniesienia kosztu proporcjonalnego do czasu trwania całego zadania inwestycyjnego.

Realizacja inwestycji polega na określeniu takiego przydziału podzadań inwestycyjnych podwykonawcom oraz określeniu takiej kolejności realizacji podzadań przez podwykonawców, aby całkowity koszt związany z realizacją inwestycji był minimalny. Problem ten można sformułować jako rozważany w pracy problem $n|m|J$, $\tau \geq 1$, $m_w^k \geq 1$, $RT \mid \sum h_i$ i rozwiązać przy pomocy algorytmu z rozdz. 9.6.

11. ZAKOŃCZENIE

Rozważania przedstawione w pracy koncentrowały się wokół zagadnień szeregowania o dużym stopniu ogólności. Sformułowano zbadano własności i rozwiązano 4 nowe problemy szeregowania. Rozważano i zanalizowano alternatywne metody rozwiązania postawionych zagadnień. Skonstruowano algorytmy rozwiązania będące podstawą oprogramowania użytkowego w systemach komputerowych optymalizujących przebieg procesu produkcyjnego. Zbadano podstawowe własności numeryczne zaproponowanych algorytmów na szeregu przykładach obliczeniowych. Oceniono złożoność obliczeniową poszczególnych kroków algorytmów. Wyniki przeprowadzonych badań wskazały na:

- (a) możliwość zastosowania podanych algorytmów w systemach uwarunkowanych czasowo; jest to realizowane poprzez przyjęcie, że na każdym etapie obliczeń istnieje rozwiązanie dopuszczalne /najlepsze dotychczas znalezione/, które w przypadku przerwania obliczeń można uznać za rozwiązanie suboptymalne,

- (b) możliwość połączenia korzystnych własności algorytmów przybliżonych /heurystycznych/ i dokładnych; rozwiązanie początkowe jest wyznaczone szybką metodą heurystyczną a następnie poprawiane algorytmem dokładnym,
- (c) przewagę zastosowanych metod rozwiązywania w stosunku do alternatywnych metod rozwiązywania zagadnień tego typu; w szczególności rozwiązanie optymalne lub bliskie optymalnemu znajduje się stosunkowo szybko, zaś liczba wszystkich węzłów w drzewie rozwiązań jest równa liczbie węzłów wiszących w drzewie rozwiązań algorytmów tego typu rozważanych w literaturze, co w ogólnym przypadku przyspiesza otrzymanie rozwiązania optymalnego.

Podane własności umożliwiają wykorzystanie w większości komputerowych systemów sterowania procesami produkcyjnymi.

Rozważane w pracy problemy są znacznie ogólniejsze niż problemy szeregowania uwzględniane do tej pory w literaturze, stąd też konstrukcja algorytmów możliwych do praktycznego wykorzystania jest problemem szczególnie trudnym. Wszystkie rozważone zagadnienia są silnie NP-zupełne i nie ma aktualnie możliwości skonstruowania dla nich algorytmów wielomianowych. Uwzględniając powyższe uzyskane wyniki obliczeniowe, dla najbardziej złożonych problemów, trudno uznać, że w pełni zadowalające. W tej sytuacji wydaje się uzasadnionym prowadzenie dalszych badań nad własnościami podanych problemów w celu poprawy efektywności i zmniejszenia nakładów obliczeniowych algorytmów.

Rozważania w pracy dotyczą zagadnień taśmowych z kryterium $f_{\max}$ i $h_{\max}$ oraz zagadnienia gniazdowego z równoległymi maszynami i kryterium $\sum h_i$. Analogiczne rozważania można przeprowadzić dla innych typów problemów szeregowania i przyjętych postaci funkcji kryterialnych. W pracy zawarto szereg sugestii dotyczących realizacji tych uogólnień w przypadku problemów minimaksowych. Początkiem zrealizowanych uogólnień jest zagadnienie przedstawione w pracy [45], w którego opracowaniu autor uczestniczył.

Ze względu na brak miejsca nie było możliwe umieszczenie w pracy szeregu wyników dotyczących przeprowadzonych przez autora badań oraz skonstruowanych algorytmów. W szczególności brak miejsca nie pozwolił na szersze omówienie alternatywnych strategii przeglądania drzewa rozwiązań i podziału węzłów oraz modyfikacji

metod wyznaczania dolnych ograniczeń. Nie zamieszczono również szeregu szczegółów związanych z implementacją programową podanych algorytmów, uważanych bardziej za stronę techniczną niż naukową problemu. Jednakże w przypadku problemów NP-zupełnych zrealizowanie "dobrej implementacji" jest problemem nietrywialnym, bowiem od przyjętego sposobu realizacji algorytmu zależy faktyczna końcowa ocena jego jakości i praktycznej przydatności /np. realizacja algorytmu z rozdz. 9 wymaga zorganizowania funkcjonowania 5 różnych algorytmów działających na obszarze pamięci o złożonej strukturze adresacji/. W konsekwencji rozwiązania zaprezentowane w pracy są jedynie fragmentem wszystkich rozważań prowadzonych przez autora i reprezentują aktualnie "najlepsze" rezultaty wyłonione w szeregu dyskusji i badań.

Dalszym istotnym kierunkiem badań jest uogólnienie wspomnianych zagadnień na przypadek uwzględnienia innych rodzajów kosztów [92], przypadek istnienia dodatkowych zasobów i/lub uwzględnienia probabilistycznego opisu zmiennych problemu.

Serdecznie dziękuję promotorowi, doc. dr hab. inż. Józefowi Grabowskiemu za pomoc i cenne uwagi.

mgr inż. Czesław Smutnicki

Raport wpłynął do Redakcji
styczeń 1981

LITERATURA

- [1] ABDEL-WAHAL H.M., KAMEDA T. - Scheduling to Minimize Maximum Cumulative Cost Subject to Series-Parallel Precedence Constraints, *Opns Res.*, vol. 26, pp. 141-158, 1978.
- [2] ACHUTHAN N.R., GRABOWSKI J., SIDNEY J.B. - Optimal Flow-Shop Scheduling with Earliness and Tardiness Penalties, *Opsearch* /to appear/.
- [3] ADRABINSKI A., GRABOWSKI J., WODECKI M. - Algorytm rozwiązywania ogólnego problemu kolejnościowego, Instytut Informatyki Uniwersytetu Wrocławskiego, Raport nr. N-28, Wrocław, 1977.
- [4] BAKER K.R., LAWLER E.L., LENSTRA J.K., RINNOOY KAN A.H.G. - Preemptive Scheduling of a Single Machine to Minimize Maximum Cost Subject to Release Dates and Precedence Constraints, *Mathematisch Centrum*, raport BW 128/80, Amsterdam, 1980.
- [5] BAKER K.R. - Introduction to Sequencing and Scheduling, John Wiley & Sons, New York 1974.
- [6] BAKER K.R., MARTIN J.B. - An Experimental Comparison of Solutions Algorithms for the Single - machine Tardiness Problem, *Naval Res. Log. Quart.*, vol. 21, pp. 187-199, 1974.
- [7] BAKER K.R., Schrage L.E. - Finding an Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks, *Opns Res.*, vol. 26, pp. 111-120, 1978.
- [8] BAKER K.R. - Computational Experience with a Sequential Algorithm Adapted to the Tardiness Problem, *AIIE Trans.*, no 9, pp. 32-55, 1977.
- [9] BAKER K.R., SU Z. - Sequencing with Due Dates and Early Start Times to Minimize Maximum Tardiness, *Navale Res. Log. Quart.*, vol. 21, pp. 171-175, 1974.
- [10] BAKER K.R. - A Comparative Study of Flow-Shop Algorithms, *Opns Res.*, vol. 23, pp. 62-73, 1975.
- [11] BALAS E. - Machine Sequencing via Disjunctive Graphs .An Implicit Enumeration Algorithm, *Opns, Res.*, vol. 17, pp. 941-957, 1969.

- [12] BANSAL S.P. - Global Lower Bound in the Flow-shop Problem, Opsearch, vol. 13, pp. 115-118, 1976.
- [13] BARGIELSKI M. - Algorytm zarządzania wydziałem wacłowni, Podstawy Sterowania, t.1, nr 4, 1971.
- [14] BARAN - JAROSZ B., GRABOWSKI W. - Wyznaczenie kolejności detali przechodzących w procesie produkcji ciąg operacji, Przegląd Statystyczny, 3-4, 1971.
- [15] BESTWICK P.F., HASTINGS N.A.J. - A New Bound for Machine Scheduling, Opł Res. Quart., vol. 27, pp. 479-487, 1976.
- [16] BŁAŻEWICZ J. - Złożoność obliczeniowa algorytmów i problemów szeregowania zadań, Wydawnictwa Polit. Poznańskiej, Ser. Rozprawy, nr 104, Poznań, 1979.
- [17] BROWN A.P.G., ŁOMNICKI Z.S. - Some Applications of the Branch-and-Bound Algorithm to the Machine Scheduling Problem, Opnal. Res. Quart., vol. 17, pp. 173-186, 1966.
- [18] BRATLEY P., FLORIAN M., ROBILLARD P. - On Sequencing with Earliest Starts and Due Dates with Application to Computing Bounds for /the $n|m|G|F_{\max}$ / Problem, Naval Res. Log. Quart., vol. 20, pp. 57-67, 1973.
- [19] BRUCHER P., GAREY M.R., JOHNSON D.S. - Scheduling Equal - Length Tasks Under Treelike Precedence Constraints to Minimize Maximum Lateness, Math. Ops. Res., vol. 2, pp. 275-384, 1977.
- [20] BRUNO J., COFFMAN E.G. jr., SETHI R. - Scheduling Independent Tasks to Reduce Mean Finishing Time, Comm. ACM, vol. 17, pp. 382-387, 1974.
- [21] BURNS R.N. - Scheduling to Minimize the Weighted Sum of Completion Times with Secondary Criteria, Naval Res. Log. Quart., vol. 23, pp. 125-129, 1976.
- [22] CAMBELL H .H ., DUDEK R.A., Smith M.L. - A Heuristic Algorithm for the n Job, m Machine Sequencing problem, Mgmt Sci, vol. 16, pp. 630-637B, 1970.
- [23] CHUDZINSKI B. - Planowanie produkcji i materiałów w dyskretnym procesie produkcyjnym. System PLANOR, Przegląd Organizacyjny /w druku/.
- [24] CIESLINSKI J., GOSCINSKI A. - Zastosowanie przybliżonego planowania kalendarzowego do szeregowania operacji na

- odcinku stalownia - walcownia, Archiwum Automatyki i Telemechaniki, z. 3, 1977.
- [25] COFFMAN E.G. jr. /red/ - Teoria szeregowania zadań, WNT, Warszawa, 1980.
 - [26] CONWAY R.W., MAXWELL W.L., MILLER L.W. - Theory of Scheduling, Addison-Wesley, 1967.
 - [27] ELMAGHRABY S.E. - The One - Machine Sequencing Problem with Delay Costs, J. Indust. Eng. vol. 19, pp. 105-108, 1968.
 - [28] EMMONS H. - One Machine Sequencing to Minimize Certain Functions of Job Tardiness, Opns Res., vol. 17, pp. 701-715, 1969.
 - [29] EMMONS H. - One Machine Sequencing to Minimize Mean Flow Time with Minimum Number Tardy, Naval Res. Log. Quart., vol. 22, pp. 585-592, 1975.
 - [30] ESKEW J.D., PARKER R.G. - Computational Experience with a Cost-based Algorithm for the Shop Scheduling Problem, Opl. Res. Quart., vol. 26, pp. 211-215, 1975.
 - [31] FISHER M.L. - A Dual Algorithm for the One - Machine Scheduling Problem, Math. Progr., vol. 11, pp. 229-251, 1976.
 - [32] GAREY M.R. - Optimal Tasks Sequencing with Precedence Constraints, Discr. Math., no 4, pp. 37-56, 1973.
 - [33] GAREY H.R., JOHNSON D.S. - Two processor scheduling with start times and deadlines, SIAM J. Comput., vol. 6, pp. 416-426, 1977.
 - [34] GARFINKEL R.S., NEMHAUSER G.L. - Programowanie całkowito-liczbowe, PWN, Warszawa 1978.
 - [35] GELDERS L., KLEINDORFER P.R. - Coordinating Agregate and Detailed Scheduling in the One - Machine Job Shop: I - Theory, Opns Res., vol. 22, pp. 46-60, 1974.
 - [36] GELDERS L., KLEINDORFER P.R. - Coordinating Agregate and Detailed Scheduling in the One - Machine Job Shop: II - Computation and Structure, Opns Res., vol. 23, pp. 312-324, 1975.
 - [37] GEOFFRION A.M. - Lagrangian Relaxation for Integer Programing, Math. Progr. Study, no 2, pp. 82-114, 1974.
 - [38] GERE W.S. Jr. - Heuristics in job Sheduling, Mgmt Sci, 13, pp. 167, 1966.

- [39] GILMORE P.C., GOMORY R.E. - Sequencing a one - state variable machine: a solvable case of the traveling salesman problem, *Opns. Res.*, vol. 12, pp. 655-679, 1964.
- [40] GRABOWSKI J. - Algorytmy optymalizacji i sterowania w dyskretnych systemach produkcyjnych, *Prace Naukowe ICT Politechniki Wrocławskiej*, Seria: Monografie, nr 42-7, Wrocław 1977.
- [41] GRABOWSKI J. - Modele matematyczne i optymalizacja zagadnień sekwencyjnych w dyskretnych procesach produkcyjnych, *Zeszyty Naukowe Politechniki Śląskiej*, Seria: Automatyka, Zeszyt 43, s. 29-37, 1978.
- [42] GRABOWSKI J. - Algorithmization of control processes in a discrete production System, *Systems Science*, vol. 1, no 2, pp. 23-34, 1975.
- [43] GRABOWSKI J. - Uogólniane zagadnienia optymalizacji kolejności operacji w dyskretnych systemach produkcyjnych. *Prace Naukowe ICT PWr.*, Ser. Monografie, nr 9, Wrocław, 1979.
- [44] GRABOWSKI J. - On Two - Machine Scheduling with Release and Due Dates to Minimize Maximum Lateness, *Opsearch*, /to appear/.
- [45] GRABOWSKI J., SKUBALSKA E., SMUTNICKI C. - On Flow - Shop with Release and Due Dates to Minimize Maximum Lateness, *Opns. Res. Quart* /to appear/
- [46] GRABOWSKI J., SMUTNICKI C. - Zagadnienia kolejnościowe. Klasyfikacja i złożoność obliczeniowa algorytmów. *Przegląd Statystyczny*/w druku/
- [47] GRABOWSKI J., SYSŁO M. - On a machine sequencing problems (I), *Zastosowanie Matematyki XIII*, 3, pp. 339-345, 1973.
- [48] GREENBERG H.H. - A Branch-bound Solution to the General Scheduling Problem, *Opns. Res.*, vol. 16, pp. 353-361, 1968.
- [49] GUPTA J.N.D. - Analysis of a Combinatorial Approach to Flowshop Scheduling Problems, *Opnl Res. Quart*, vol. 26, pp. 431-440, 1975.
- [50] HELD M., WOLFE P., CROWDER P. - Validation of Subgradient Optimization, *Math., Progr.*, 6, pp. 62-88, 1974.
- [51] HELD H., KARP R.M. - A Dynamic Programing Approach to Sequencing Problems, *Journal of SIAM*, vol. 10, pp. 196-210,

1962.

- [52] HORN W.A. - Single Machine Job Sequencing with Treelike Precedence Ordering and Linear Delay Penalties, SIAM Journal Appl. Math., vol. 23, pp. 189-202, 1972.
- [53] HORN W. - Some Simple Scheduling Algorithms, Naval Res. Log. Quart., vol. 21, pp. 177-186, 1974.
- [54] HORN W.A. - Minimizing Average Flow Time with Parallel Machines, Opns. Res., vol. 21, pp. 846-847, 1973.
- [55] JACKSON J.R. - An Extension of Johnson's Results on Job Lot Scheduling, Naval Res. Log. Quart., vol. 3, pp. 201-203, 1956.
- [56] JOHNSON S.M. - Optimal 2 and 3 Stage Production Schedules with Set - up Time Included, Naval Res. Log. Quart., vol. 1, pp. 61-68, 1954.
- [57] KOFMAN A., Anri - Labordier A. - Metody i modeli issledowania operacji, Izd. Mir, Moskwa 1977.
- [58] KOHLER W.H., STEIGLITZ K. - Exact approximate and quarantead accuracy algorithms for the flow-shop problem $n|2|F|F$, Journal of the ACM, vol. 22, pp. 106-114, 1975.
- [59] KORBUT A.A., FINKELSZTEJN J.J. - Programowanie dyskretne, PWN, Warszawa 1974.
- [60] KURISU T. - Single - Machine Sequencing with Precedence Constraints and Deferral Costs, J. of the Ops. Res. Soc. of Japan, vol. 22, no. 1, pp. 1-15, 1979.
- [61] KUSIAK A. - Zadania optymalnego planowania produkcji dla przypadku identycznych maszyn równoległych. Prace IBS PAN, Warszawa, 1978.
- [62] LAKSHMINARAYAN, LAKSHMANAN R., PAPINEAU R., ROCHETE R. - Optimal Single - machine Scheduling with Earliness and Tardiness Penalties, Opns Res., vol. 26, no. 6, pp. 1079-1082, 1978.
- [63] LAGEWEG B.J., LENSTRA J.K., RINNOOY KAN A.H.G. - A General Bounding Scheme for the Permutation Flow-Shop Problem, Opns. Res., vol. 26, pp. 53-67, 1978.
- [64] LAGEWEG B.J., LENSTRA J.K., RINNOOY KAN A.H.G. - Minimizing Maximum Lateness on One Machine: computational experience and some applications, Statistica Neerlandica, vol. 30, 1976, pp. 24-41.

- [65] LAWLER E.L. - Efficient Implementation of Dynamic Programming Algorithms for Sequencing Problems, Matematish Centrum, Raport BW 106/79, Amsterdam, 1979.
- [66] LAWLER E.L. - Sequencing Jobs to Minimize Total Weighted Completion Time Subject to Precedence Constraints, Ann. Discr. Math., no. 2, pp. 75-90, 1978.
- [67] LAWLER E.L., SIVAZLIAN B.D. - Minimization of Time Varying Costs in Single - Machine Scheduling, Opns Res., vol. 26, pp. 563-569, 1978.
- [68] LAWLER E.L. - A Pseudopolynomial Algorithm for Sequencing Jobs to Minimize Total Tardiness, Ann. Discr. Math., no. 1, pp. 331-342, 1977.
- [69] LAWLER E.L. - On Scheduling Problems with Deferral Cost, Mgmt Sci, vol. 11, pp. 280-288, 1964.
- [70] LAWLER E.L. - Optimal Sequencing of a Single Machine Subject to Precedence Constraints, Mgmt Sci, vol. 19, pp. 544-546, 1973.
- [71] LENSTRA J.K. - Sequencing by Enumerative Methodes, Mathematical centre Tract 69, Matematish Centrum, Amsterdam, 1977.
- [72] LENSTRA J.K., RINNOOY KAN A.H.G. - Complexity of Scheduling Under Precedence Constraints, Ops Res., vol. 26, pp. 22-35, 1978.
- [73] LOMNICKI Z.A. - A "Branch - and - Bound Algorithm" for the Exact Solution of the Three - machine Scheduling Problem, Opl. Res. Quart., vol. 16, pp. 89-100, 1965.
- [74] Mc MAHON G., FLORIAN M. - On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness, Op Res., vol. 23, pp. 475-482, 1975.
- [75] Mc MAHON G.B., BURTON D.G. - Flow-Shop Scheduling with Branch - and - Bound Method, Opns Res., vol. 15, pp. 473-481, 1967.
- [76] Mc NAUGHTON R. - Scheduling with Deadline and Loss Functions, Ngmt Sci, vol. 6, pp. 1-12, 1959.
- [77] NABESHIMA I. - On the Bound of Makespans and its Application in M machine Scheduling Problem, J. of Ops. Res. Japan, vol. 9, pp. 98-136, 1967.
- [78] NABESHIMA J. - Algorithms for Multiproject Scheduling with Resource Constraints and Related Parallel Scheduling, Rep.

- Univ. of Electro-Comm., nr 23-1, pp. 29-50, 1972.
- [79] PICARD J.C., QUEYRANNE M. - The TDTSP and its Application to the Tardiness Problem in One - Machine Scheduling, *Ops Res.*, vol. 26, pp. 86-110, 1978.
 - [80] REKLAITIS V., TAMULYNAS B. - Computer Aided Design of Job Sequencing Rules in Flow Shop Systems /niepublikowany rękopis/.
 - [81] RINNOOY KAN A.H.G., LENSTRA J.K., LAGEWEG C.Z. - Minimizing Total Costs in One - Machine Scheduling, *Opns. Res.*, vol. 23, pp. 908-927, 1975.
 - [82] RINNOOY KAN A.H.G. - Machine Scheduling Problems: Classification, Complexity and Computations, Nijhoff, The Hague, 1976.
 - [83] SCHILD A., FREDMAN J. - Scheduling Tasks with Deadlines and Nonlinear Loss Functions, *Mgmt Sci*, vol. 9, pp. 73-81, 1962.
 - [84] SCHRAGE L., BAKER K.R. - Dynamic Programing Solution of Sequencing Problems with Precedence Constraints, *Opns. Res.*, vol. 26, pp. 444-449, 1978.
 - [85] SHWIMER J. - On the N - Job, One - Machine Sequence - Independent Scheduling Problem with Tardiness Penalties: a Branch-and-Bound Solution, *Mgmt Sci*, vol. 18, pp. 301-313, 1972.
 - [86] SIDNEY J.B. - Optimal Single - machine Scheduling with Earliness and Tardiness Penalties, *Opns. Res.*, vol. 25, pp. 62-69, 1977.
 - [87] SIDNEY J.B. - Decomposition Algorithms for Single - Machine Sequencing with Precedence Relations and Deferral Costs, *Opns. Res.*, vol. 23, pp. 283-298, 1975.
 - [88] SŁOWINSKI R. - Algorytmy sterowania rozdziałem zasobów różnych kategorii w kompleksie operacji, Politechnika Poznańska, Seria Rozprawy nr 114, Poznań 1980.
 - [89] SMITH M.L., PANWALKER S.S. DUDEK R.A. - Flow Shop Sequencing Problem with Ordered Processing Time Matrices, *Mgmt Sci*, vol. 21, pp. 544-540, 1974/75.
 - [90] SMITH R.D., DUDEK R.A. - A General Algorithm for the Solution of the n-Job, m - Machine Sequencing Problem of the Flow-Shop, *Opns. Res.*, vol. 15, pp. 171-82, 1967.

- [91] SMITH W.E. - Various Optimizer for Single-Stage Production, Naval Res. Log. Quart., vol. 3, pp. 59-66, 1956.
- [92] SMUTNICKI C. - Minimalizacja kosztów użytych maszyn w dyskretnym systemie produkcyjnym, Materiały VIII KKA, Szczecin 1980.
- [93] SMUTNICKI C., GRABOWSKI J. - Metody i algorytmy optymalizacji kolejności operacji z kryterium minimalizacji kosztów, Zeszyty Naukowe Politechniki Śląskiej, Seria Automatyka, z. 55, 1980.
- [94] SRINIVASAN V. - A Hybrid Algorithm for the One - Machine Sequencing Problem to Minimize Total Tardiness, Naval Res. Log. Quart., vol. 18, pp. 317-327, 1971.
- [95] SZWARC W. - Mathematical Aspects of the $3 \times n$ Job - Shop Sequencing Problem, Naval Res. Log. Quart., vol. 21, pp. 145 - 154, 1974.
- [96] SZWARC W. - Optimal Elimination Methods in the $m \times n$ Sequencing Problem, Opns. Res., vol. 21, pp. 1250-1259, 1973.
- [97] TOWNSEND W. - Minimising the Maximum Penalty in the m -Machine Sequencing Problem, Opl. Res. Quart., vol. 28, pp. 727-734, 1977.
- [98] WILKERSON L.J., IRWIN J.D. - An Improved Method for Scheduling Independent Task, AIIE Trans., no. 3, pp. 239-245, 1971.

Niniejszy raport otrzymują

1. OINT	1 egz.
2. Biblioteka Główna Politechniki Wrocławskiej	1 egz.
3. Promotor	1 egz.
4. Recenzenci	2 egz.
5. Z-ca Dyr. d/s Kształcenia Kadry Naukowej	1 egz.
6. Autor	2 egz.

Razem 8 egz.

