ROCZNIKI POLSKIEGO TOWARZYSTWA MATEMATYCZNEGO
SERIA III: MATEMATYKA STOSOWANA III (1974)

Joanna MALICKA-WĄSOWSKA (Warszawa)

O pewnej metodzie rozwiązywania problemów sekwencyjnych z ciągłą pracą maszyn

1. Sformułowanie problemu. Zajmijmy się problemem wyznaczania kolejności obróbki n detali na m maszynach postawionym przez Tiutiukina [4]. Przyjmijmy, że muszą być spełnione następujące założenia:

1. Dana jest macierz $T = [t_{ji}]_{n \times m}$, gdzie $t_{ji} \geqslant 0$ jest czasem obróbki j-tego detalu ($j = 1, 2, \ldots, n$) na i-tej maszynie ($i = 1, 2, \ldots, m$).
2. Każdy detal musi przejść obróbkę kolejno na maszynach o numerach $1, 2, \ldots, m$.
3. Każda maszyna może obrabiać w danym momencie tylko jeden detal.
4. Jeden detal w tym samym czasie może podlegać obróbce tylko na jednej maszynie.
5. Kolejność obróbki detali na wszystkich maszynach jest taka sama.
6. Każda maszyna pracuje w sposób ciągły, tzn. po rozpoczęciu obróbki pierwszego w kolejności detalu pracuje nieprzerwanie aż do zakończenia ostatniego w kolejności detalu.

Poszukuje się takiej kolejności obróbki detali, przy której łączny czas obróbki n detali na m maszynach jest minimalny.

W pracy zaproponowano metodę rozwiązania tego problemu, która polega na przeglądzie wszystkich permutacji detali z jednoczesną minimalizacja działań arytmetycznych na każdym kroku.

Za pomocą tej metody można również rozwiązywać problemy sekwencyjne z ciągłą pracą maszyn spełniające nieco inne założenia jak 1—6.

2. Model zagadnienia. Wprowadźmy (jak w pracy [3]) macierz $X = [x_{jk}]_{n \times m}$ przyjmując

$$x_{jk} = \begin{cases} 1 & \text{jeśli } j\text{-ty detal przechodzi obróbkę jako } k\text{-ty w kolejności,} \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Ponadto wprowadźmy macierz $W = [w_{ki}]_{n \times m}$, gdzie w_{ki} oznacza moment, w którym zaczyna się obróbka k-tego w kolejności detalu na i-tej maszynie.

Minimalizujemy wartość funkcji

$$f(W, X) = w_{nm} + \sum_{j=1}^{n} x_{jn} t_{jm} \tag{1}$$

przy ograniczeniach

$$\sum_{k=1}^{n} x_{jk} = 1 \qquad \text{dla } j = 1, 2, \ldots, n, \tag{2}$$

$$\sum_{j=1}^{n} x_{jk} = 1 \qquad \text{dla } k = 1, 2, \ldots, n, \tag{3}$$

$$x_{jk} \geqslant 0, \text{ całkowite dla } j = 1, 2, \ldots, n; \qquad k = 1, 2, \ldots, n, \tag{4}$$

$$w_{l+1,i} = w_{li} + \sum_{j=1}^{n} x_{jl} t_{ji} \qquad \text{dla } l = 1, 2, \ldots, n-1; i = 1, 2, \ldots, m, \tag{5}$$

$$w_{l,i+1} \geqslant w_{li} + \sum_{j=1}^{n} x_{jl} t_{ji} \qquad \text{dla } l = 1, 2, \ldots, n; i = 1, 2, \ldots, m-1. \tag{6}$$

Równość (5) można zapisać w następującej postaci:

$$w_{l+1,i} = w_{1i} + \sum_{j=1}^{n} \sum_{k=1}^{l} x_{jk} t_{ji} \qquad \text{dla } l = 1, 2, \ldots, n-1; i = 1, 2, \ldots, m. \tag{7}$$

Stąd:

$$f(W, X) = w_{1m} + \sum_{j=1}^{n} \sum_{k=1}^{n} x_{jk} t_{jm}. \tag{8}$$

Widzimy, że minimalizacja funkcji (8) jest równoważna minimalizacji wartości w_{1m}. Podstawiając (7) do nierówności (6) otrzymamy:

$$\begin{cases} w_{1,i+1} \geqslant w_{1i} + \sum_{j=1}^{n} x_{j1} t_{ji} & \text{dla } i = 1, 2, \ldots, m-1, \\ w_{1,i+1} + \sum_{j=1}^{n} \sum_{k=1}^{r-1} x_{jk} t_{j,i+1} \geqslant w_{1i} + \sum_{j=1}^{n} \sum_{k=1}^{r} x_{jk} t_{ji} & \\ & \text{dla } i = 1, 2, \ldots, m-1; r = 2, 3, \ldots, n. \end{cases} \tag{9}$$

Otrzymujemy zagadnienie minimalizacji (8) przy warunkach (2), (3), (4), (7), (9).

Łatwo jest wyznaczyć najmniejszą wartość $w_{1,i+1}$ spełniającą warunki (9).

$$w_{1,i+1} \geqslant w_{1i} + \max \left[\max_{2 \leqslant r \leqslant n} \left(\sum_{j=1}^{n} \sum_{k=1}^{r} x_{jk} t_{ji} - \sum_{j=1}^{n} \sum_{k=1}^{r-1} x_{jk} t_{j,i+1} \right), \sum_{j=1}^{n} x_{j1} t_{ji} \right] \tag{10}$$

$$\text{dla } i = 1, \ldots, m-1.$$

Wprowadźmy macierz $C = [c_{ir}]_{(m-1) \times n}$, $i = 1, \ldots, m-1; r = 1, 2, \ldots, n$, gdzie:

(11)
$$\begin{cases} c_{i1} = \sum\limits_{j=1}^{n} x_{j1} t_{ji} & \text{dla } i = 1, 2, \ldots, m-1, \\ c_{ir} = \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{r} x_{jk} t_{ji} - \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{r-1} x_{jk} t_{j,i+1} & \end{cases}$$

$$\text{dla } r = 2, \ldots, n;\ i = 1, \ldots, m-1.$$

Nierówność (10) można zapisać następująco:

(12)
$$w_{1,i+1} \geqslant w_{1i} + \max_{1 \leqslant r \leqslant n} c_{ir} \qquad \text{dla } i = 1, \ldots, m-1.$$

Niech

(13)
$$c_{ir_i} = \max_{1 \leqslant r \leqslant n} c_{ir} \qquad \text{dla } i = 1, \ldots, m-1.$$

Wtedy najmniejszą wartością w_{1i} spełniającą (12) jest

(14)
$$w_{1,i+1} = w_{1i} + c_{ir_i} \qquad \text{dla } i = 1, \ldots, m-1.$$

Nie zmniejszając ogólności rozważań możemy przyjąć $w_{11} = 0$. Stąd

(15)
$$w_{1m} = \sum_{i=1}^{m-1} c_{ir_i}.$$

3. Metoda rozwiązania zagadnienia. Rozpatrzmy pewną permutację detali

$$V^{(n)} = \{j_1, j_2, \ldots, j_n\}.$$

Dla tej permutacji zapiszmy macierz C:

(16)
$$\begin{cases} c_{i1} = t_{j_1 i} & \text{dla } i = 1, \ldots, m-1, \\ c_{ir} = \sum\limits_{k=1}^{r} t_{j_k i} - \sum\limits_{k=1}^{r-1} t_{j_k, i+1} & \text{dla } r = 2, \ldots, n;\ i = 1, \ldots, m-1. \end{cases}$$

Mamy zminimalizować wartość funkcji:

(17)
$$f(V^{(n)}) = \sum_{i=1}^{m-1} \max_{1 \leqslant r \leqslant n} c_{ir}.$$

Znana metoda podziału i ograniczeń często prowadzi do przeglądu wszystkich permutacji detali.

Zaproponowana w pracy metoda jest metodą przeglądu wszystkich permutacji z jednoczesną minimalizacją działań arytmetycznych na każdym kroku.

Rozpatrzmy permutację detali:

$$V^{(n)} = \{j_1, \ldots, j_s, j_{s+1}, \ldots, j_n\},$$

$$\bar{V}^{(n)} = \{j_1, \ldots, j_{s+1}, j_s, \ldots, j_n\}.$$

Obie permutacje różnią się jedynie kolejnością detali j_s i j_{s+1}. Permutacje takie nazwijmy sąsiednimi. Niech c_{ir} i $\bar{c}_{ir}$ oznaczają wielkości określone wzorami (16) odpowiednie dla permutacji $V^{(n)}$ i $\bar{V}^{(n)}$. Widać, że $c_{ir} = \bar{c}_{ir}$ dla $r \neq s$, $s+1$, tzn. dla dwóch sąsiednich permutacji macierz C różni się tylko dwoma kolumnami.

Proponowana metoda polega na przeglądzie wszystkich permutacji, przy czym następna permutacja musi być sąsiednia dla permutacji poprzedzającej ją. Permutacje należy ustawić w ciąg w ten sposób, aby każda permutacja występowała w nim tylko jeden raz. Ponieważ w literaturze [5] podano takie ustawienie permutacji, więc korzystając z tego proponuje się postępowanie następujące:

Dwa elementy można ustawić w następujący sposób: {1,2}, {2,1}. Załóżmy, że ustawiliśmy w taki ciąg permutacje k-elementowe. Permutacje $(k+1)$-elementowe otrzymujemy ustawiając element $(k+1)$-szy w pierwszej k-elementowej permutacji kolejno na miejscach $k+1$, k, ..., 1, a dla następnej permutacji k-elementowej odwrotnie, tzn. na miejscach 1, 2, ..., $k+1$ itd.

Przykład:

{1, 2}	{1, 2, 3}	{1, 2, 3, 4}
		{1, 2, 4, 3}
		{1, 4, 2, 3}
		{4, 1, 2, 3}
	{1, 3, 2}	{4, 1, 3, 2}
		{1, 4, 3, 2}
		{1, 3, 4, 2}
		{1, 3, 2, 4}
	{3, 1, 2}	{3, 1, 2, 4}
		{3, 1, 4, 2}
		{3, 4, 1, 2}
		{4, 3, 1, 2}
{2, 1}	{3, 2, 1}	{4, 3, 2, 1}
		{3, 4, 2, 1}
		{3, 2, 4, 1}
		{3, 2, 1, 4}
	{2, 3, 1}	{2, 3, 1, 4}
		{2, 3, 4, 1}
		{2, 4, 3, 1}
		{4, 2, 3, 1}
	{2, 1, 3}	{4, 2, 1, 3}
		{2, 4, 1, 3}
		{2, 1, 4, 3}
		{2, 1, 3, 4}

Metoda ta łatwo daje się programować na EMC.

4. Algorytm obliczeniowy.

1. Obliczmy elementy macierzy $C = [c_{ir}]$ według wzorów (16) dla permutacji $V^{(n)}$ i wybierzmy w każdym wierszu element maksymalny – c_{ir_i}. Wyznaczamy wartość funkcji celu $f(V^{(n)})$ z wzoru (17).

2. Dla sąsiedniej permutacji $\bar{V}^{(n)}$ wyznaczamy macierz $\bar{C}$, która ma tylko 2 odpowiednie kolumny różne od kolumn macierzy C. Elementy z tych dwóch kolumn obliczamy z następujących wzorów:

$$(18)\qquad \begin{cases} \bar{c}_{is} = c_{is} - t_{j_s i} + t_{j_{s+1} i} & \text{dla } i = 1, \ldots, m-1, \\ \bar{c}_{i,s+1} = c_{i,s+1} + t_{j_s, i+1} - t_{j_{s+1}, i+1} & \text{dla } i = 1, \ldots, m-1. \end{cases}$$

3a. Jeżeli dla każdego i ($i = 1, \ldots, m-1$) zachodzi $r_i \neq s,\ s+1$, to $f(V^{(n)}) \leqslant f(\bar{V}^{(n)})$. Wyznaczamy nowe elementy maksymalne w wierszach macierzy $\bar{C}$.

b. Jeżeli dla jakiegoś i mamy $r_i = s$ lub $r_i = s+1$, to wyznaczamy nowe elementy maksymalne w wierszach macierzy $\bar{C}$ oraz obliczamy wartość funkcji celu $f(\bar{V}^{(n)})$, która może zmaleć.

4. Wracamy do punktu 2, jeśli nie zostały sprawdzone wszystkie permutacje.

5. Inne zagadnienia z ciągłą pracą maszyn. Opisaną wyżej metodę można wykorzystać nie tylko do minimalizacji łącznego czasu obróbki detali, ale i do rozwiązania problemów, w których bądź jest inne kryterium optymalizacyjne, bądź zagadnienie spełnia inne założenia niż 1–6 ze wstępu.

Rozpatrzmy problem minimalizacji strat oczekiwania detali na obróbkę przy ograniczeniach (2), (3), (4), (7), (9) (patrz [4]). Niech d_j oznacza straty spowodowane leżeniem j-tego detalu w jednostce czasu ($j = 1, 2, \ldots, n$). Minimalizujemy wartość funkcji:

$$(19)\qquad f(W, X) = \sum_{l=1}^{n} d_{j_l}\Big(w_{1m} + \sum_{j=1}^{n}\sum_{k=1}^{l} x_{jk} t_{jm} - \sum_{i=1}^{m} t_{j_l i}\Big).$$

Rozpatrzmy permutację detali $V^{(n)} = \{j_1, j_2, \ldots, j_n\}$. Minimalizacja funkcji (19) jest równoważna minimalizacji funkcji

$$(20)\qquad f(V^{(n)}) = \sum_{l=1}^{n} d_{j_l}\Big(\sum_{i=1}^{m-1} \max_{1\leqslant r\leqslant n} c_{ir} + \sum_{k=1}^{l} t_{j_k m}\Big).$$

Badając 2 sąsiednie permutacje w trakcie obliczeń w punkcie 3 algorytmu sprawdzamy dodatkowo warunek:

$$(21)\qquad d_{j_s} t_{j_{s+1} m} - d_{j_{s+1}} t_{j_s m} = \varphi.$$

Jeśli $\varphi \geqslant 0$ oraz $r_i \neq s,\ s+1$ dla $i = 1, 2, \ldots, m-1$, to

$$f(V^{(n)}) \leqslant f(\bar{V}^{(n)}).$$

Jeśli $\varphi < 0$, wyznaczamy nową wartość funkcji celu.

Drugim problemem jest problem z ustalonymi możliwie najwcześniejszymi momentami rozpoczęcia pracy maszyn (patrz [4]). Oprócz założeń 1–6 sformułowanych we wstępie za-

kładamy, że dla $i = 1, 2, \ldots, m$ dana jest liczba L_i oznaczająca najwcześniejszy moment, w którym i-ta maszyna może zacząć pracować. Minimalizujemy funkcję (8) przy ograniczeniach (2), (3), (4), (7), (9) oraz przy dodatkowym warunku:

$$w_{1i} \geqslant L_i \qquad \text{dla } i = 1, 2, \ldots, m. \tag{21}$$

Dla tego zagadnienia nie zachodzi oczywiście równość (14). Natomiast prawdziwą jest następująca:

$$w_{1,i+1} = \max\left(w_{1i} + \max_{1 \leqslant r \leqslant n} c_{ir},\ L_{i+1}\right) \qquad \text{dla } i = 1, 2, \ldots, m-1 \tag{22}$$

przy czym $w_{11} = L_1$.

Rozwiązując ten problem obliczenia wykonujemy według podanego wyżej algorytmu obliczając wartość funkcji celu ze wzoru (22).

Trzecim problemem jest zadanie wyznaczania kolejności obróbki n detali na m maszynach postawione przez Kuzina [2].

Zakładamy, że

1. Detale wykonuje się partiami, przy czym wielkość partii danego detalu jest taka sama na wszystkich maszynach.
2. Dana jest macierz $T = [t_{ji}]_{n \times m}$, gdzie $t_{ji} \geqslant 0$ jest czasem obróbki partii j-tego detalu ($j = 1, 2, \ldots, n$) na i-tej maszynie ($i = 1, 2, \ldots, m$), przy czym wielkość tej partii nie interesuje nas.
3. Każda partia detali musi przejść kolejno przez maszyny o numerach $1, 2, \ldots, m$.
4. W danym momencie każda maszyna może obrabiać tylko jedną partię detali.
5. Jedna partia detali może być jednocześnie obrabiana na kilku maszynach.
6. Kolejność obróbki partii detali jest taka sama na wszystkich maszynach.
7. Wszystkie maszyny pracują w sposób ciągły.

Minimalizujemy łączny czas obróbki wszystkich partii detali na wszystkich maszynach. Problem ten spotyka się w praktyce przy pracy na „taśmie produkcyjnej".

Problem sprowadza się do minimalizacji wartości funkcji (8) przy warunkach (2), (3), (4), (5) oraz

$$w_{l,i+1} \geqslant w_{li} \qquad \text{dla } l = 1, 2, \ldots, n;\ i = 1, 2, \ldots, m-1, \tag{23}$$

$$w_{l,i+1} + \sum_{j=1}^{n} x_{jl} t_{j,i+1} \geqslant w_{li} + \sum_{j=1}^{n} x_{jl} t_{ji} \tag{24}$$

$$\text{dla } l = 1, 2, \ldots, n;\ i = 1, 2, \ldots, m-1.$$

Podstawiając (7) do nierówności (24) otrzymamy:

$$w_{1,i+1} + \sum_{j=1}^{n} \sum_{k=1}^{l-1} x_{jk} t_{j,i+1} + \sum_{j=1}^{n} x_{jl} t_{j,i+1} \geqslant w_{1i} + \sum_{j=1}^{n} \sum_{k=1}^{l-1} x_{jk} t_{ji} + \sum_{j=1}^{n} x_{jl} t_{ji} \tag{25}$$

$$\text{dla } i = 1, 2, \ldots, m-1;\ l = 1, 2, \ldots, n.$$

Stąd:

(26)
$$w_{1,i+1} \geq w_{1i} + \sum_{j=1}^{n} \sum_{k=1}^{l} x_{jk} t_{ji} - \sum_{j=1}^{n} \sum_{k=1}^{l} x_{jk} t_{j,i+1}$$

dla $i = 1, 2, \ldots, m-1$; $l = 1, 2, \ldots, n$.

Z nierówności (23) i (26) otrzymujemy

(27)
$$w_{1,i+1} \geq w_{1i} + \max\left(0, \max_{1 \leq r \leq n} \sum_{j=1}^{n} \sum_{k=1}^{l} x_{jk} (t_{ji} - t_{j,i+1})\right)$$

dla $i = 1, 2, \ldots, m-1$.

Wprowadźmy macierz $D = [d_{ir}]_{(m-1)\times(n+1)}$, gdzie

(28)
$$\begin{cases} d_{i0} = 0 & \text{dla } i = 1, 2, \ldots, m-1, \\ d_{ir} = \sum_{j=1}^{n} \sum_{k=1}^{r} x_{jk} (t_{ji} - t_{j,i+1}) & \text{dla } i = 1, 2, \ldots, m-1;\ r = 1, 2, \ldots, n. \end{cases}$$

Nierówność (27) możemy teraz zapisać:

(29)
$$w_{1,i+1} \geq w_{1i} + \max_{0 \leq r \leq n} d_{ir} \qquad \text{dla } i = 1, 2, \ldots, m-1.$$

Przyjmując, że $w_{11} = 0$, otrzymamy:

(30)
$$w_{1m} = \sum_{i=1}^{m-1} \max_{0 \leq r \leq n} d_{ir}.$$

Rozpatrzmy teraz permutację $V^{(n)} = \{j_1, j_2, \ldots, j_n\}$. Dla tej permutacji zapiszmy macierz D:

(31)
$$\begin{cases} d_{i0} = 0 & \text{dla } i = 1, 2, \ldots, m-1, \\ d_{ir} = \sum_{k=1}^{r} (t_{j_k i} - t_{j_k, i+1}) & \text{dla } i = 1, 2, \ldots, m-1;\ r = 1, 2, \ldots, n. \end{cases}$$

Rozważmy dwie sąsiednie permutacje $V^{(n)}$ i $\bar{V}^{(n)}$ Widzimy, że $d_{ir} \neq \bar{d}_{ir}$ tylko dla $r = s$, tzn. przy przejściu od permutacji $V^{(n)}$ do sąsiedniej do niej $\bar{V}^{(n)}$ zmienia się tylko jedna kolumna macierzy D według wzoru

(32)
$$\bar{d}_{is} = d_{is} - t_{j_s i} + t_{j_{s+1} i} + t_{j_s, i+1} - t_{j_{s+1}, i+1} \qquad \text{dla } i = 1, 2, \ldots, m-1.$$

Algorytm obliczeniowy dla tego problemu jest analogiczny jak opisany wyżej dla poprzedniego zagadnienia.

6. Przypadki szczególne. Zaproponowanej wyżej metody nie opłaca się stosować dla małej liczby maszyn.

Rozpatrzmy problem minimalizacji (8) przy warunkach (2), (3), (4), (7), (9). Jak wyka-

zano w pracy [3] dla $m = 2$ (2 maszyny) rozwiązanie optymalne znajdujemy według algorytmu Johnsona [1], opracowanego dla zagadnienia uszeregowania detali na 2 maszynach przy założeniach 1–5 ze wstępu. Dla tegoż zagadnienia dla 3 maszyn w przypadku, gdy

$$\max_{1 \leqslant j \leqslant n} t_{j1} \leqslant \min_{1 \leqslant j \leqslant n} t_{j2}$$

rozwiązujemy podobnie n razy problem Johnsona dla maszyn 2 i 3 stawiając za każdym razem inny detal jako pierwszy i wybieramy rozwiązanie najlepsze lub, gdy

$$\max_{1 \leqslant j \leqslant n} t_{j2} \leqslant \min_{1 \leqslant j \leqslant n} t_{j3}$$

rozwiązujemy podobnie n razy problem Johnsona dla maszyny 1 i 2.

Minimalizując funkcję (8) przy ograniczeniach (2), (3), (4), (7), (9), (21) w przypadku 2 maszyn również rozwiązujemy zadanie Johnsona, przy czym wartość funkcji celu liczymy oczywiście według wzoru (22).

P r z y p a d k i s z c z e g ó l n e z a g a d n i e n i a p o s t a w i o n e g o p r z e z K u z i n a.

I. $m = 2$ (2 maszyny):

$$w_{12} = \max\Big(\max_{1 \leqslant r \leqslant n} \sum_{k=1}^{r} (t_{j_k 1} - t_{j_k 2}),\ 0\Big) + w_{11}.$$

1. Jeśli $t_{j_k 1} \leqslant t_{j_k 2}$ dla $= k = 1, 2, \ldots, n$, to $w_{12} = w_{11}$ przy dowolnej permutacji.

2. Jeśli $t_{j_k 1} \geqslant t_{j_k 2}$ dla $k = 1, 2, \ldots, n$, to $w_{12} = \sum_{k=1}^{n} (t_{j_k 1} - t_{j_k 2})$ przy dowolnej permutacji.

3. Jeśli $t_{j_k 1} < t_{j_k 2}$ dla $k = 1, 2, \ldots, s$ i $t_{j_k 1} \geqslant t_{j_k 2}$ dla $k = s + 1, s + 2, \ldots, n$, to wśród optymalnych kolejności jest następująca permutacja $\{j_1, j_2, \ldots, j_s, j_{s+2}, \ldots, j_n\}$

$$w_{12} = \max\Big(\sum_{k=s+1}^{n} (t_{j_k 1} - t_{j_k 2}) + \sum_{k=1}^{s} (t_{j_k 1} - t_{j_k 2}),\ 0\Big).$$

Widać, że jeśli pewien detal z grupy $j_{s+1}, \ldots, j_n$ wprowadzimy przed jakimś detalem z grupy $j_1, \ldots, j_s$, to w_{12} może wzrosnąć.

II. $m = 3$ (3 maszyny):

1. Jeśli $t_{j_k 1} \geqslant t_{j_k 2}$ dla $k = 1, 2, \ldots, n$ lub $t_{j_k 1} \leqslant t_{j_k 2}$ dla $k = 1, 2, \ldots, n$, to optymalna kolejność będzie taka, jak dla 2 maszyn wyliczona dla maszyny drugiej i trzeciej (na podstawie równości (30)).

2. Jeśli $t_{j_k 2} \geqslant t_{j_k 3}$ dla $k = 1, 2, \ldots, n$, $t_{j_k 2} \leqslant t_{j_k 3}$ dla $k = 1, 2, \ldots, n$, to optymalną kolejność znajdujemy dla maszyny pierwszej i drugiej.

3. Niech

(a) $t_{j_k 1} \leqslant t_{j_k 2} \leqslant t_{j_k 3}$ dla $k = 1, 2, \ldots, s$,

(b) $t_{j_k 1} \geqslant t_{j_k 2} \leqslant t_{j_k 3}$ dla $k = s + 1, \ldots, s + r$,

(c) $t_{j_k 1} \leqslant t_{j_k 2} \geqslant t_{j_k 3}$ dla $k = s + r + 1, \ldots, s + p$,

(d) $t_{j_k 1} \geqslant t_{j_k 2} \geqslant t_{j_k 3}$ dla $k = s + p + 1, \ldots, n$.

Ze wzorów (30) i (31) widać, że permutowanie między sobą detali w poszczególnych grupach (a), (b), (c), (d) nie wpływa na zmniejszenie funkcji celu. Detale z grupy (a) powinny zawsze stać na początku a z grupy (d) na końcu (patrz przypadek 2 maszyn).

7. Wnioski końcowe. Sformułowane w tej pracy zagadnienia dotychczas rozwiązywano metodą podziału i ograniczeń (patrz [2], [3], [4]) i metodą eliminacji [3]. Autorzy tych prac podają przykłady, w których za pomocą zaproponowanych przez nich metod można wyeliminować pewną ilość kolejności detali. Niemniej wiadomo, że metody te mogą prowadzić do przeglądu wszystkich kolejności detali [1].

Ze względu na to, że z góry nie można powiedzieć czy metody te prowadzą do przeglądu wszystkich permutacji, czy tylko pewnych, proponuje się stosowanie przedstawionej w tej pracy metody (przegląd wszystkich permutacji detali z jednoczesną minimalizacją działań arytmetycznych na każdym kroku).

Literatura

[1] *Industrial Scheduling*, Edited by F. Muth and G. L. Thompson with the collaboration of P. R. Winters, New Jersey 1963.

[2] B. I. Kuzin, *Algoritm postrojenija optimalnogo kalendarnogo grafika nieprierywnoj raboty oborudowanija w usłowijach paralelno-posledowatielnogo wida dwiżenija proizwodstwa*, Primienienije matematiki w ekonomikie, wypusk 6, Izdatielstwo Leningradskogo Uniwiersytieta, 1970.

[3] J. Malicka-Wąsowska, *Model sekwencyjny z ciągłą pracą maszyn*, Praca zgłoszona do druku w „Przeglądzie Statystycznym".

[4] W. K. Tiutiukin, *„Nachożdienije optimalnych płanow w mnogoopieracjonnych procesach obrabotki izdielij metodom „Wietwiej i granic"*, Primienienije matiematiki w ekonomikie, wypusk 6, Izdatielstwo Leningradskogo Uniwiersytieta, 1970.

[5] G. Ehrlich, *Loopless algorithms for generating permutations, combinations, and other combinatorial configurations*, Journal of the Association for Computing Machinery 20(3), July 1973.

[1] W pracy [3] wykazano, że dla problemów sekwencyjnych z ciągłą pracą maszyn przy $n \geqslant 2m$ można wyeliminować przynajmniej jedną permutację. Niemniej ta metoda wymaga pamiętania wszystkich permutacji, co uniemożliwia stosowanie jej przy dużych wartościach n.