SZACOWANIE WARTOŚCI FUNKCJI CELU W PROBLEMIE GNIAZDOWYM Z RÓWNOLEGŁYMI MASZYNAMI

Wojciech BOŻEJKO, Mariusz UCHROŃSKI, Mieczysław WODECKI

Streszczenie: W pracy rozpatrujemy elastyczny problem kolejnościowy z równoległymi maszynami (ang. *flexible job shop problem*). Na podstawie badań statystycznych pojawiła się w literaturze hipoteza, iż moduł wyznaczający wartość funkcji kryterialnej w algorytmach rozwiązywania tego problemu (głównie metaheurystycznych) stanowi nawet 97% całkowitego czasu pracy algorytmu. W niniejszej pracy proponujemy zastosowanie metod szacowania wartości funkcji celu, co pozwala bardzo znacząco skrócić czas pracy algorytmów, przy zachowaniu dobrej jakości otrzymywanych rozwiązań.*

Słowa kluczowe: elastyczne systemy wytwarzania, problem gniazdowy, równoległe maszyny.

1. Wstęp

Rozważany w pracy problem gniazdowy z równoległymi maszynami należy do klasy problemów silnie NP-trudnych, jest bowiem uogólnieniem klasycznego problemu gniazdowego (*job shop*, [4]). Algorytm dokładny rozwiązywania rozpatrywanego zagadnienia został przedstawiony w pracy [8], ale pozwala on rozwiązać zadania o nie więcej niż 20 zadaniach i 10 maszynach. W literaturze proponuje się także metaheurystyki. Hurink [5] oraz Mastrolilli i Gambardella [6] zaproponowali metodę poszukiwania z zabronieniami (*tabu search*) dla rozważanego problemu. Gao i in. [3] proponują metody algorytmu genetycznego oraz poszukiwania za zmiennym otoczeniem (VNS). Bożejko, Uchroński i Wodecki [1] opracowali dwupoziomowy algorytm oparty na otoczeniu golfowym. Bożejko [2] przedstawia metodę zrównoleglenia procesu wyznaczania otoczenia w rozważanym tu elastycznym problemie gniazdowym. Dla rozpatrywanego problemu w niniejszej pracy proponujemy metodę szacowania wartości funkcji celu, która znacznie przyspiesza działanie algorytmów heurystycznych.

2. Sformułowanie problemu

Ogólny problem kolejnościowy z równoległymi maszynami można sformułować następująco, opierając się na notacji wprowadzonej w [2]: dany jest zbiór zadań $J = \{1,2,\ldots,n\}$, które należy wykonać na maszynach ze zbioru $M = \{1, 2,\ldots, m\}$. Niech $O = \{1,2,\ldots,o\}$ będzie zbiorem wszystkich operacji. Zbiór ten można rozbić na ciągi odpowiadające zadaniom, przy czym zadanie $j \in J$ jest ciągiem o_j operacji, które będą kolejno wykonywane na odpowiednich maszynach (tj. w ciągu technologicznym). Operacje te *są* indeksowane liczbami $(l_{j-1}+1,\ldots, l_{j-1}+o_j)$, gdzie l_j jest liczbą operacji pierwszych j zadań, $j = 1, 2,\ldots, n$, przy czym $l_0 = 0$. Zbiór maszyn $M = \{1, 2,\ldots, m\}$ można rozbić na q

* Praca finansowana z projektów badawczych MNiSW nr N N514 232237 oraz N N514 470439.

podzbiorów maszyn tego samego typu (*gniazd*), przy czym i-ty (i = 1, 2,..., q) typ M_i zawiera m_i maszyn, które *są* indeksowane liczbami (t_{i-1}+1,... ,t_{i-1} + m_i), gdzie t_i jest liczbą maszyn w pierwszych i typach, $i = 1,2,\dots,q$, przy czym $t_0 = 0$.

Operację $v \in O$ należy wykonać w gnieździe $\mu(v)$, tj. na jednej z maszyn zbioru $M^{\mu\ (v)}$ w czasie $p_{v,j}$, gdzie $j \in M^{\mu\ (v)}$. Niech $O^k = \{ v \in O:\ \mu(v) = k \}$ będzie zbiorem operacji wykonywanych w k-tym (k = 1, 2,..., q) gnieździe. Ciąg zbiorów operacji $Q = [Q^1, Q^2, \dots, Q^m]$, takich, że dla każdego $k = 1,2,\dots,q$ zachodzi zależność $O^k = \bigcup_{i=t_{k-1}+1}^{t_{k-1}+m_k} Q^i$ oraz $Q^i \cap Q^j = \emptyset,\ i \neq j,\ i,j = 1,2,\dots,m$, nazywamy *przydziałem operacji zbioru O do maszyn ze zbioru M* (w skrócie *przydziałem operacji do maszyn*). Ciąg $[Q^{t_{k-1}+1}, Q^{t_{k-1}+2}, \dots, Q^{t_{k-1}+m_k}]$ jest *przydziałem* maszynom operacji w i-tym gnieździe (w skrócie *przydziałem* w i-tym gnieździe). W szczególnym przypadku maszyna może nie wykonywać żadnej operacji i wówczas w przydziale operacji w gnieździe zbiór operacji do wykonywania przez tę maszynę jest pusty.

3. Szacowanie wartości funkcji kryterialnej

Jeżeli dokonano przydziału operacji do maszyn, wówczas wyznaczenie optymalnego terminu wykonywania operacji (w tym i kolejności wykonywania operacji na maszynach) sprowadza się do rozwiązania klasycznego problemu szeregowania, tzw. problemu gniazdowego.

Niech $K = [K_1, K_2, ..., K_m]$, będzie ciągiem zbiorów, gdzie $K_i \in 2^{\mathrm{O}^i}$, $i = 1,2,...,m$. W szczególności elementy tego ciągu mogą być zbiorami pustymi. Przez $\mathcal{K}$ oznaczamy zbiór wszystkich takich ciągów. Moc zbioru $\mathcal{K}$ wynosi $2^{|\mathrm{O}^1|} \times 2^{|\mathrm{O}^2|} \times ... \times 2^{|\mathrm{O}^q|}$.

Jeżeli $\mathcal{Q}$ jest dowolnym przydziałem operacji do maszyn, to $\mathcal{Q} \in \mathcal{K}$ (oczywiście, zbiór $\mathcal{K}$ zawiera także ciągi, które nie są dopuszczalne, tj. nie są przydziałem operacji do maszyn).

Dla dowolnego ciągu zbiorów $K = [K_1, K_2, ..., K_m]$ ($K \in \mathcal{K}$) przez $\Pi_i(K)$ oznaczamy zbiór wszystkich permutacji elementów z K_i. Dalej, niech Dalej, niech

$$\pi(K) = (\pi_1(K), \pi_2(K), ..., \pi_m(K))$$

będzie konkatenacją (złożeniem) m ciągów (permutacji), gdzie $\pi_i(K) \in \Pi_i(K)$:

$$\pi(K) \in \Pi(K) = \Pi_1(K) \times \Pi_2(K) \times, ..., \Pi_m(K).$$

Łatwo zauważyć, że jeżeli $K = [K_1, K_2, ..., K_m]$ jest pewnym przydziałem operacji do maszyn, to zbiór $\pi_i(K)$ $(i = 1,2,...,m)$ zawiera wszystkie permutacje (możliwe kolejności wykonywania) operacji ze zbioru K_i na maszynie i. Dalej, niech

$$\Phi = \{(K, \pi(K)) : K \in \mathcal{K} \wedge \pi(K) \in \Pi(K)\},$$

będzie zbiorem par, których pierwszym elementem jest ciąg zbiorów, a drugim - konkatenacja permutacji elementów tych zbiorów. Dowolne rozwiązanie dopuszczalne problemu PJOBS jest parą $(\mathcal{Q}, \pi(\mathcal{Q})) \in \Phi$, gdzie $\mathcal{Q}$ jest przydziałem operacji do maszyn, a $\pi(\mathcal{Q})$ konkatenacją permutacji wyznaczających kolejność wykonywania operacji przydzielonych każdej z maszyn spełniającą ograniczenia (*i-iv*). Przez Φ° oznaczamy zbiór rozwiązań dopuszczalnych dla problemu PJOBS. Oczywiście, $\Phi^{\circ} \subset \Phi$.

4. Reprezentacja grafowa rozwiązania

Dowolne rozwiązanie dopuszczalne $\Theta = (\mathsf{Q}, \pi(\mathsf{Q})) \in \Phi^{\circ}$ (gdzie Q jest przydziałem operacji do maszyn, a $\pi(\mathsf{Q})$ jest kolejnością wykonywania operacji na każdej maszynie) problemu PJOBS można przedstawić w postaci grafu skierowanego z obciążonymi wierzchołkami (sieci) $G(\Theta) = (\mathsf{V}, \mathsf{R} \cup \mathsf{E}(\Theta))$, gdzie V jest zbiorem wierzchołków, a $\mathsf{R} \cup \mathsf{E}(\Theta)$ zbiorem łuków, przy czym:

1. $\mathsf{V} = \mathsf{O} \cup \{s, c\}$, gdzie s i c są dodatkowymi (fikcyjnymi) operacjami reprezentującymi odpowiednio ,,start" i ,,zakończenie". Wierzchołek $v \in \mathsf{V} \setminus \{s, c\}$ ma dwie cechy:

- $\lambda(v)$ -- numer maszyny na której należy wykonać operację $v \in \mathsf{O}$,
- $p_{v,\lambda(v)}$ -- wagę wierzchołka równą czasowi wykonywania operacji $v \in \mathsf{O}$ na maszynie $\lambda(v)$.

Wagi dodanych wierzchołków $p_s = p_c = 0$.

2. $$\mathsf{R} = \bigcup_{j=1}^{n} \bigcup_{i=1}^{o_j - 1} \{(l_{j-1} + i, l_{j-1} + i + 1)\} \cup \{(s, l_{j-1} + 1)\} \cup \{(l_{j-1} + o_j, c)\}.$$

Zbiór R zawiera łuki łączące kolejne operacje tego samego zadania oraz łuki z wierzchołka s do pierwszej operacji każdego zadania i łuki od ostatniej operacji każdego zadania do wierzchołka c.

3. $$\mathsf{E}(\Theta) = \bigcup_{k=1}^{m} \bigcup_{i=1}^{|\mathsf{O}^k| - 1} \{(\pi_k(i), \pi_k(i+1))\}.$$

Łatwo zauważyć, że łuki ze zbioru $\mathsf{E}(\Theta)$ łączą operacje wykonywane na tej samej maszynie (π_k jest permutacją operacji wykonywanych na maszynie M_k, tj. operacji ze zbioru O^k).

Łuki ze zbioru R wyznaczają kolejność wykonywania operacji w zadaniach (porządek technologiczny), a łuki ze zbioru $\mathsf{E}(\Theta)$ kolejność wykonywania operacji na każdej z maszyn.

Uwaga 1 *Para* $\Theta = (\mathsf{Q}, \pi(\mathsf{Q})) \in \Phi$ *jest rozwiązaniem dopuszczalnym dla problemu PJOBS wtedy i tylko wtedy, gdy graf* $G(\Theta)$ *nie zawiera cykli.*

Ciąg wierzchołków $(v_1, v_2, \ldots, v_k)$ grafu $G(\Theta)$ taki, że $(v_i, v_{i+1}) \in \mathsf{R} \cup \mathsf{E}(\Theta)$ dla $i = 1, 2, \ldots, k-1$, nazywamy *drogą* (lub *ścieżką*) z wierzchołka v_1 do v_k. Przez

$C(v,u)$ oznaczmy najdłuższą drogę (zwaną *drogę krytyczną*) w grafie $G(\Theta)$ z wierzchołka v do u ($v,u \in \mathsf{V}$), a przez $L(v,u)$ *długość* (sumę wag wierzchołków) tej drogi.

Łatwo zauważyć, że czas wykonywania wszystkich operacji $C_{\max}(\Theta)$ zgodnie z przydziałem operacji Q i kolejnością (uszeregowaniem) $\pi(\mathsf{Q})$ jest równy długości $L(s,c)$ drogi krytycznej $C(s,c)$ w grafie $G(\Theta)$. Rozwiązanie problemu gniazdowego z równoległymi maszynami sprowadza się więc do wyznaczenia takiego rozwiązania dopuszczalnego $\Theta = (\mathsf{Q}, \pi(\mathsf{Q})) \in \Phi^{\circ}$, dla którego odpowiadający mu graf $G(\Theta)$ ma najkrótszą drogę krytyczną, tj. minimalizuje $L(s,c)$.

Niech $C(s,c) = (s, v_1, v_2, \ldots, v_w, c)$, gdzie $v_i \in \mathsf{O}$ ($1 \le i \le w$) będzie drogą krytyczną w grafie $G(\Theta)$ z wierzchołka początkowego s do końcowego c. Drogę tę można podzielić na podciągi wierzchołków

$$\mathsf{B} = [B^1, B^2, \ldots, B^r],$$

zwane ***blokami*** w permutacji na drodze krytycznej $C(s,c)$, przy czym

1. blok jest podciągiem wierzchołków z drogi krytycznej zawierającym kolejne operacje wykonywane bezpośrednio jedna po drugiej,
2. blok zawiera operacje wykonywane na tej samej maszynie,
3. przekrój dwóch dowolnych bloków jest zbiorem pustym.
4. blok jest maksymalnym (ze względu na zawieranie) podzbiorem operacji z drogi krytycznej spełniającym ograniczenia 1.-3.

W dalszej części będą rozpatrywane tylko te bloki, dla których $|B^k| > 1$, czyli tzw. bloki niepuste.

Jeżeli B^k ($k = 1,2,...,r$) jest blokiem na maszynie M_i ($i = 1,2,...,m$) z gniazda t ($t = 1,2,...,q$), to będziemy go oznaczali następująco:

$$B^k = (\pi_i(a^k), \pi_i(a^k+1), \ldots, \pi_i(b^k-1), \pi_i(b^k)),$$

gdzie $1 \le a^k \le b^k \le |Q^i|$. Operacje $\pi(a^k)$ i $\pi(b^k)$ w bloku B^k są nazwane odpowiednio *pierwszą* i *ostatnią*. Z kolei blok bez pierwszej i ostatniej operacji nazywamy *blokiem wewnętrznym*.

Twierdzenie 1 *Niech* $\mathsf{B} = [B^1, B^2, \ldots, B^r]$ *będzie ciągiem bloków drogi krytycznej w acyklicznym grafie* $G(\Theta)$, $\Theta \in \Phi^{\circ}$. *Jeżeli graf* $G(\Omega)$ *jest dopuszczalny i został wygenerowany z* $G(\Theta)$ *przez zmianę kolejności wykonywania operacji na pewnej maszynie oraz* $C_{\max}(\Omega) < C_{\max}(\Theta)$, *to w* $G(\Omega)$

- przynajmniej jedna operacja z pewnego bloku B^k, $k \in \{1,2,\ldots,r\}$ poprzedza pierwszy element $\pi(a^k)$ tego bloku, lub
- przynajmniej jedna operacja z pewnego bloku B^k, $k \in \{1,2,\ldots,r\}$ występuje za ostatnim elementem $\pi(b^k)$ tego bloku.

Zmiana kolejności operacji w dowolnym bloku nie generuje rozwiązania o mniejszej wartości funkcji celu. Aby więc poprzez zmianę kolejności wykonywania operacji na maszynach wygenerować graf o mniejszej długości ścieżki krytycznej należy przynajmniej jedną operację z dowolnego bloku przesunąć przed pierwszą lub za ostatnią operację tego bloku. Fakt ten będzie wykorzystywany do eliminowania z otoczenia elementów zbędnych (nie dających poprawy wartości funkcji celu), tj. do wyznaczania subotoczenia.

5. Problem przydziału operacji do maszyn

W tym rozdziale jest rozpatrywany problemem wyznaczenia "dobrego" (suboptymalnego) przydziału operacji do maszyn. Każda operacja w sposób jednoznaczny jest przydzielona dokładnie do jednego gniazda. W każdym z gniazd należy więc dokonać rozbicia przydzielonych operacji na poszczególne maszyny. Sposób rozbicia ma wpływ na czas zakończenia wykonywania wszystkich zadań, tj. wartość rozwiązania problemu gniazdowego (job shop). Generalnie, należy dokonać takiego rozbicia, aby wartość funkcji celu dla tego rozbicia (tj. rozwiązania problemu job shop) była minimalna.

Niech $\Theta = (\mathsf{Q}, \pi(\mathsf{Q})) \in \Phi^{\circ}$ będzie rozwiązaniem dopuszczalnym problemu PJOBS, gdzie $\mathsf{Q} = [\mathsf{Q}^1, \mathsf{Q}^2, \ldots, \mathsf{Q}^m]$ jest przydziałem operacji do maszyn, ρ_i liczbą operacji wykonywanych na maszynie M_i (tj. $\rho_i = |\mathsf{Q}^i|$), a

$$\pi(\mathsf{Q}) = (\pi_1(\mathsf{Q}), \pi_2(\mathsf{Q}), ..., \pi_m(\mathsf{Q}))$$

konkatenacją *m* permutacji. Permutacja $\pi_i(\mathsf{Q})$ wyznacza kolejność operacji ze zbioru Q^i, które należy wykonać na maszynie M_i ($i = 1,2,...,m$).

W dalszej części tego rozdziału, tam gdzie nie prowadzi to do niejednoznaczności, będziemy pomijali przydział operacji Q występujący jako parametr permutacji. Wobec tego konkatenację $\pi(\mathsf{Q}) = (\pi_1(\mathsf{Q}), \pi_2(\mathsf{Q}), ..., \pi_m(\mathsf{Q}))$ będziemy w skrócie zapisywali $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$.

Przez $t_j^i(k,l)$ oznaczamy ruch typu *transfer* (w skrócie *t-ruch*) polegający na przeniesieniu operacji znajdującej się na pozycji k w permutacji π_i (tj. operacji $\pi_i(k)$) na pozycję l w permutacji $\pi_j(k)$ (przesuwając wcześniej operacje znajdujące się na pozycjach k, $k+1$, itd. o jedną pozycję w prawo). Wykonanie ruchu $t_j^i(k,l)$ generuje z $\Theta = (\mathsf{Q}, \pi) \in \Phi^{\circ}$ nowe rozwiązanie $\Theta' = (\mathsf{Q}', \pi')$ takie, że

$$\pi'_v = \pi_v, \quad v \neq i, j, \; v = 1,2,...,m \tag{1}$$

oraz

$$\pi'_i = (\pi_i(1), \pi_i(2), \ldots, \pi_i(\mathrm{k}-1), \pi_i(\mathrm{k}+1), \ldots, \pi_i(\rho_i - 1), \tag{2}$$

$$\pi'_j = (\pi_j(1), \pi_i(2), \ldots, \pi_i(\mathrm{l}-1), \underline{\pi_i(\mathrm{k})}, \pi_j(\mathrm{l}), \ldots, \pi_i(\rho_j + 1). \tag{3}$$

Wykonanie tego ruchu powoduje przeniesienie operacji $\pi_i(k)$ ze zbioru Q^i (tj.

z maszyny M_i) do zbioru $\mathcal{Q}^j$ (tj. na maszynę M_j). Wobec tego

$$\mathcal{Q}'^v = \mathcal{Q}^v, \quad v \neq i, j, \; v = 1,2,...,m$$

oraz

$$\mathcal{Q}'^i = \mathcal{Q}^i \setminus \{\pi_i(k)\}, \; \mathcal{Q}'^j = \mathcal{Q}^j \cup \{\pi_i(k)\}.$$

Graf $G(\Theta')$ wygenerowany przez wykonanie *t-ruchu* może zawierać cykl i wówczas rozwiązanie $\Theta' = (\mathcal{Q}', \pi')$ nie jest dopuszczalne.

Wykonanie t*-ruchu* powoduje przeniesienie operacji z pewnej maszyny na inną, tj. nowy przydział operacji do maszyn w pewnym gnieździe. Wobec tego, z dowolnego rozwiązania (przydziału operacji do maszyn), wykonując *t-ruchy* można otrzymać dowolny inny przydział operacji do maszyn, tj. rozbicia zbiorów operacji w poszczególnych gniazdach.

Jeżeli τ jest *t-ruchem*, to przez $\tau(\Theta)$ oznaczamy rozwiązanie wygenerowane z Θ przez wykonanie ruchu τ .

Dla ustalonego rozwiązania dopuszczalnego Θ, niech $\mathcal{T}(\Theta)$ będzie zbiorem wszystkich *t-ruchów*. *Otoczeniem* Θ jest zbiór

$$\mathcal{N}(\Theta) = \{\tau(\Theta) \in \Phi^\circ : \tau \in \mathcal{T}\}, \tag{4}$$

gdzie Φ° jest zbiorem rozwiązań dopuszczalnych, przy czym dopuszczalność $\tau(\Theta)$ jest równoważna acykliczności grafu $G(\tau(\Theta)$.

Na początku tego rozdziału napisaliśmy, że metoda rozwiązania problemu PJOBS składa się z dwóch kroków. Pierwszy - wyznaczenie pewnego przydziału operacji do maszyn, oraz drugi - wyznaczenie kolejności wykonywania operacji, tj. rozwiązanie problemu gniazdowego.

Niech $\Theta = (\mathcal{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym problemu PJOBS. Nowy przydział operacji do maszyn $\mathcal{Q}'$ będziemy generowali z przydziału $\mathcal{Q}$ następująco:

- wyznaczyć otoczenie $\mathcal{N}(\Theta)$,
- wybrać z otoczenia rozwiązanie $\Theta' = (\mathcal{Q}', \pi')$ o najmniejszej wartości funkcji celu - nowy przydział operacji do maszyn $\mathcal{Q}'$.

Ponieważ liczba *t-ruchów* może być bardzo duża, więc niektóre z nich pominiemy i będziemy rozpatrywali jedynie te, które mogą przynieść poprawę wartości funkcji celu. Ponadto, nie będziemy wyznaczali dokładnej wartości funkcji celu, rozwiązań generowanych przez *t-ruchy*, a jedynie je szacowali. Jak wykazały eksperymenty obliczeniowe powoduje to znaczne przyśpieszenie działania algorytmu przy niewielkim pogorszeniu wyników. W dalszej części dokładnie opiszemy metody eliminowania z otoczenia (4) zbędnych ruchów oraz szacowania wartości funkcji celu.

6. Wyznaczanie otoczenia

Wykonanie *t-ruchu* może generować rozwiązanie niedopuszczalne, tzn. odpowiadający temu rozwiązaniu graf zawiera cykl. Badanie dopuszczalności rozwiązania jest więc równoważne badaniu acykliczności grafu. Odpowiedni algorytm na złożoność obliczeniową $O(o)$, gdzie o jest liczbą operacji. W dalszej części udowodnimy twierdzenia umożliwiające w czasie stałym badanie dopuszczalności rozwiązań (tj.

acykliczności odpowiadających im grafów) generowanych przez *t-ruchy*.
Niech $\Theta = (\mathsf{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym, gdzie $\mathsf{Q} = [\mathsf{Q}^1, \mathsf{Q}^2, \ldots, \mathsf{Q}^m]$ jest przydziałem operacji do maszyn, a $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ konkatenacją *m* permutacji. Permutacja π_i wyznacza kolejność wykonywania operacji ze zbioru Q^i na maszynie M_i ($i = 1,2,...,m$).
Rozpatrujemy dwie maszyny z tego samego gniazda M_i i M_j. Permutacja π_i wyznacza kolejność wykonywania operacji zbioru Q^i na maszynie M_i, a π_j kolejność wykonywania operacji zbioru Q^j na maszynie M_j. Dla dowolnej operacji $\pi_i(k) \in \mathsf{Q}^i$ definiujemy dwa parametry związane z drogami w grafie $G(\Theta)$:

- pierwszy parametr

$$\eta_j(k) = \begin{cases} 1 & \textit{gdy } \forall v = 1,2,\ldots,\rho_j \textit{ nie istnieje droga } C(\pi_j(v), \pi_i(k)), \\ 1 + \max\limits_{1 \le v \le \rho_j} \{\textit{istnieje} \text{ d}\textit{roga } C(\pi_j(v), \pi_i(k))\} & \textit{w p. p.} \end{cases} \tag{5}$$

Wobec tego z żadnej operacji znajdujących się w permutacji π_j na pozycjach $\eta_j(k), \eta_j(k)+1, \ldots, \rho_j$ nie istnieje w grafie $G(\Theta)$ droga do operacji (wierzchołka) $\pi_i(k)$,

- drugi parametr

$$\rho_j(k) = \begin{cases} 1+\rho_j & \textit{gdy } \forall v = 1,2,\ldots,\rho_j \textit{ nie istnieje droga } C(\pi_j(v), \pi_i(k)), \\ 1 + \min\limits_{\eta_j(k) \le v \le \rho_j} \{\textit{istnieje} \text{ d}\textit{roga } C(\pi_i(k), \pi_j(v))\} & \textit{w p. p.} \end{cases} \tag{6}$$

Z powyższej definicji wynika, że w grafie $G(\Theta)$ nie istnieje droga z wierzchołka $\pi_i(k)$ do żadnej z operacji znajdujących się na pozycjach $\eta_j(k), \eta_j(k)+1, \ldots, \rho_j(k)$ w permutacji π_j

Twierdzenie 2
Niech $\Theta = (\mathsf{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym dla problemu PJOBS oraz π_i, π_j permutacjami operacji wykonywanych na maszynach M_i, M_j. Jeżeli maszyny M_i, M_j należą do tego samego gniazda, to wykonanie t-ruchu $t_j^i(k,l)$ $(l = 1,2,\ldots,\eta_j(k)-1)$ generuje rozwiązanie, które nie jest dopuszczalne.

Dowód. Niech $\Theta = (\mathsf{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym, a $G(\Theta)$ odpowiadającym mu grafem. Permutacja π_i wyznacza kolejność wykonywania operacji na

maszynie M_i, a π, kolejność na maszynie M_j. Rozpatrujemy *t-ruch* $t_j^i(k,l)$ polegający na przeniesieniu operacji $\pi_i(k)$ z maszyny M_i na pozycję l $(1 \le l \le \eta_j(k)-1)$ w permutacji $\pi_j(l)$, tj. na maszynę M_j. Ruch ten generuje nowe rozwiązanie $\Theta' = (Q', \pi')$. Udowodnimy, że graf ten zawiera cykl.
Z definicji (5) parametru $\eta_j(k)$ wynika, że w grafie $G(\Theta)$ istnieje droga z wierzchołka $\pi_j(\eta_j(k)-1)$ do $\pi_i(k)$, tj. droga $C(\pi_j(\eta_j(k)-1), \pi_i(k))$.

Ponadto, w grafie tym istnieje także droga $C(\pi_j(l) = (\pi_i(k), \pi_j(\eta_j(k)-1))$. Wykonanie ruchu $t_j^i(k,l)$ powoduje wstawienie operacji $\pi_i(k))$ na pozycję l w permutacji π_j. Wynikiem tego jest wstawienie do grafu $G(\Theta')$, między innymi, łuku $(\pi_i)(k), \pi_j(l))$. Wynikiem tego jest powstanie cyklu $((\pi_i(k), \pi_j(l)), C(\pi_i(k), \pi_j(\eta_j(k)-1)), C(\pi_j(\eta_j(k)-1), \pi_i(k)))$, co kończy dowód twierdzenia.

Podobne twierdzenie można udowodnić dla parametru $\rho_j(k)$ z definicji (6).

Twierdzenie 3
Niech $\Theta = (Q, \pi)$ będzie rozwiązaniem dopuszczalnym dla problemu PJOBS oraz π_i, π_j permutacjami operacji wykonywanych na maszynach M_i, M_j. Jeżeli maszyny M_i, M_j należą do tego samego gniazda, to wykonanie $\text{t}-$ruchu $t_j^i(k,l)$ $(l = \rho_j(k)+1, \rho_j(k)+2, \ldots, \rho_j)$ generuje rozwiązanie, które nie jest dopuszczalne.
Dowód. Podobnie jak w dowodzie Twierdzenia 2 można pokazać, że po wykonaniu ruchu $t_j^i(k,l)$ $(l = \rho_j(k)+1, \rho_j(k)+2, \ldots, \rho_j)$ w wygenerowanym grafie $G(\Theta')$ powstanie cykl

$$(C(\pi_i(k), \pi_j(\rho_j(k)-l)), C(\pi_j(\rho_j(k)-l), \pi_j(l-1)), (\pi_j(l-1), \pi_i(k)).$$

Przez T^{noacc} oznaczamy zbiór tych *t-ruchów* z $\mathsf{T}(\Theta)$, które spełniają założenia Twierdzenia 2 lub 3. Są to więc ruchy generujące z Θ niedopuszczalne rozwiązania.

Twierdzenie 4
Niech $\Theta = (Q, \pi)$ będzie rozwiązaniem dopuszczalnym dla problemu PJOBS oraz π_i, π_j permutacjami operacji wykonywanych na maszynach M_i, M_j. Jeżeli maszyny M_i, M_j należą do tego samego gniazda, to wykonanie $\text{t}-$ruchu $t_j^i(k,l)$ $(l = \eta_j(k), \eta_j(k)+1, \ldots, \rho_j(k))$ generuje rozwiązanie dopuszczalne.
Dowód. Dowód tego twierdzenia wynika bezpośrednio z definicji parametrów

$\eta_j(k), \rho_j(k)$ $(\pi(k) \in \mathrm{Q}^i)$ oraz Twierdzenia 2 i 3.

Własność 1 *Dla każdej operacji* $\pi_i(k)$ *wykonywanej na maszynie* M_i *istnieje pozycja* l *w permutacji* π_j *(tj. na maszynie* M_j *z tego samego gniazda) taka, że wykonanie ruchu* $t^i_j(k,l)$ *generuje z* Θ *rozwiązanie dopuszczalne.*

Dowód. Wystarczy zauważyć, że jeżeli Θ jest rozwiązaniem dopuszczalnym to $\rho_j(k) \geq \eta_j(k)$.

Obecnie udowodnimy twierdzenie będące podstawą do eliminowania, w procesie generowania otoczenia, pewnych *t-ruchów*. Pełni ono podobną funkcję jak twierdzenie 1.

Załóżmy, że dla pewnego przydziału Q konkatenacja π jest optymalnym uszeregowaniem operacji na maszynach, $G(\Theta)$ grafem odpowiadającym rozwiązaniu $\Theta = (\mathrm{Q}, \pi)$, a $C_{\max}(\Theta)$ wartością funkcji celu, tj. długością drogi krytycznej w grafie $G(\Theta)$.

Twierdzenie 5

Niech $\Theta = (\mathrm{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym dla problemu PJOBS, a $\mathrm{B} = [B^1, B^2, \ldots, B^r]$ ciągiem bloków drogi krytycznej w grafie $G(\Theta)$. Jeżeli $\Theta' = (\mathrm{Q}', \pi')$ jest rozwiązaniem dopuszczalnym, które zostało wygenerowane z Θ przez zmianę przydziałów operacji do maszyn w pewnym gnieździe oraz $C_{\max}(\Theta') < C_{\max}(\Theta)$ to w Θ' przynajmniej jedna operacja z pewnego bloku została przeniesiona na inną (w tym samym gnieździe) maszynę.

Dowód. Niech $\mathrm{B} = [B^1, B^2, \ldots, B^r]$ będzie ciągiem bloków drogi krytycznej w grafie $G(\Theta)$. Każdy blok jest ciągiem operacji

$$B^i = (\pi(a^i), \pi(a^i+1), \ldots, \pi(b^i)),$$

dla $i = 1,2,\ldots,r$, gdzie $1 \leq a^1 \leq b^1 < a^2 \leq b^2 <,\ldots,< a^k \leq b^k$.

Dla uproszczenia oznaczeń w dowodzie tego twierdzenia zakładamy, że każda operacja z drogi krytycznej należy do pewnego bloku. Tak więc blok może zawierać tylko jedną operację.

Przez

$$\mathrm{Y}^i(\pi) = \{\pi(a^i), \pi(a^i+1), \ldots, \pi(b^i)\},$$

oznaczamy zbiór zadań z bloku B^i.

Ścieżka krytyczna $C(s,c)$ w grafie $G(\Theta)$ zawiera wszystkie wierzchołki (operacje) zbioru $\bigcup_{i=1}^r \mathrm{Y}^i$, a jej długość $L(s,c) = C_{\max}(\Theta) = \sum_{i=1}^r \sum_{v \in \mathrm{Y}^i} p_v$.

Niech $\Theta' = (\mathrm{Q}', \pi')$ będzie rozwiązaniem dopuszczalnym takim, że

$C_{\max}(\Theta') < C_{\max}(\Theta)$. Przypuśćmy, że w przydziale operacji do maszyn $\mathcal{Q}'$ żadna operacja z żadnego bloku $B^1, B^2, \ldots, B^r$ nie zostało przeniesiona na inną maszynę z tego samego gniazda. Tak więc

$$\mathcal{Y}^i(\pi) = \mathcal{Y}^i(\pi'), \quad i = 1,2,\ldots,r.$$

Wobec tego ciąg zadań $(\pi(a^i), \pi(a^i+1), \ldots, \pi(b^i))$ w permutacji π oraz $(\pi'(a^i), \pi'(a^i+1), \ldots, \pi'(b^i))$ w β są permutacjami tego samego podzbioru zadań $\mathcal{Y}^i = \{\pi(a^i),\ \pi(a^i+1), \ldots, \pi(b^i)\}$. Rozpatrujemy drogę $C'(s,c)$ w grafie $G(\Theta')$. Wierzchołki tej drogi należą do zbioru $\mathcal{A} = \cup_i^r \mathcal{Y}^i(\pi')$. Długość tej drogi $L'(s,c) = \sum_{v \in \mathcal{A}} p_v$, a więc jest równa długości $L(s,c) = C_{\max}(\Theta)$ drogi krytycznej $C(s,c)$ w grafie $G(\Theta)$. Wobec tego $C_{\max}(\Theta') \geq C_{\max}(\Theta)$, co jest sprzeczne z założeniem.

Niech Θ będzie rozwiązaniem dopuszczalnym, $\mathcal{B}$ ciągiem bloków drogi krytycznej w grafie $G(\Theta)$ a $\mathcal{T}$ zbiorem *t-ruchów* określonych dla Θ.

Przez $\mathcal{T}^{out}(\Theta)$ oznaczamy zbiór tych ruchów z $\mathcal{T}(\Theta)$, które przenoszą operacje nie należące do żadnego bloku (tj. z poza bloku) na inną maszynę.
Bezpośrednio z Twierdzenia 5 wynika własność, która jest podstawą do eliminowania zbędnych ruchów.

Własność 2 *Jeżeli rozwiązanie dopuszczalne* Θ' *zostało wygenerowane z* Θ *przez wykonanie pewnego t-ruchu należącego do zbioru* $\mathcal{T}^{out}(\Theta)$*, to*

$$C_{\max}(\Theta') \geq C_{\max}(\Theta).$$

Dowód. Dowód wynika bezpośrednio z Twierdzenia 5.

Tak więc wykonanie t*-ruchu* polegającego na przeniesieniu na inną maszynę operacji nie leżącej na ścieżce krytycznej nie generuje rozwiązania o mniejszej wartości funkcji celu.

Twierdzenie 6 *Niech* $\Theta = (\mathcal{Q}, \pi)$ *będzie rozwiązaniem dopuszczalnym dla problemu* PJOBS. *Jeżeli* B^u *jest blokiem na maszynie* M_i*, a* B^v *blokiem na* M_j *oraz obie maszyny należą do tego samego gniazda, to ruch typu transfer polegający na przeniesienie operacji z bloku wewnętrznego* B^u *do bloku wewnętrznego* B^v *nie generuje rozwiązania o mniejszej wartości funkcji celu.*

Dowód. Niech $\Theta = (\mathcal{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym, $G(\Theta)$ adpowiadającym mu grafem, a $\mathcal{B} = [B^1, B^2, \ldots, B^r]$ ciągiem bloków z drogi krytycznej. Zakładamy, że

$$B^u = (\pi(a^u), \pi(a^u+1), \ldots, \pi(b^u)),\ \ B^v = (\pi(a^v), \pi(a^v+1), \ldots, \pi(b^v)),$$

są blokami $(1 \le u < v \le r)$ odpowiednio na maszynie M_i oraz M_j. Rozpatrujemy *t-ruch* $t_j^i(k,l)$, gdzie $a^u < k < b^u$ oraz $a^v < l < b^v$ przenoszący operację z bloku B^u do bloku B^v. Ruch ten generuje z Θ nowe rozwiązanie $\Theta' = (Q', \pi')$. Przydział operacji do maszyn Q' oraz permutacja π' są określone w (1)÷(3). Drogę krytyczną $C(s,c)$ w grafie $G(\Theta)$ można rozbić następująco:

$$C(s,c) = (C(s,\pi(a^u)), C(\pi(a^u),\pi(b^u)),$$

$$C(\pi(b^u),\pi(a^v)), C(\pi(a^v),\pi(b^v)), C(\pi(b^v),c)).$$

W grafie $G(\Theta')$ istnieje droga

$$C'(s,c) = (C'(s,\pi'(a^u)), C'(\pi'(a^u),\pi'(b^u)),$$

$$C'(\pi'(b^u),\pi'(a^v)), C'(\pi'(a^v),\pi'(b^v)), C'(\pi'(b^v),c)).$$

Łatwo zauważyć, że równe są następujące drogi:

$$C'(s,\pi'(a^u)) = C(s,\pi(a^u)),\ \ C'(\pi'(b^u),\pi'(a^v)) = C(\pi(b^u),\pi(a^v))$$

oraz

$$C'(\pi'(b^v),s) = C(\pi(b^v),s),$$

a więc równe są także ich długości.

Rozpatrujemy drogi $C(\pi(a^u),\pi(b^u))$ i $C(\pi(a^v),\pi(b^v))$ w grafie $G(\Theta)$ oraz $C'(\pi(a^u),\pi(b^u))$ i $C'(\pi(a^v),\pi(b^v))$ w grafie $G(\Theta')$. Ponieważ droga $C'(\pi'(a^u),\pi'(b^u))$ zawiera wszystkie wierzchołki drogi $C(\pi(a^u),\pi(b^u))$ z wyjątkiem wierzchołka $\pi(k)$, który został przeniesiony przez *t-ruch* bloku B^u, więc zachodzi

$$L'(\pi'(a^u),\pi'(b^u)) = L(\pi(a^u),\pi(b^u)) - p_{\pi(k)}.$$

Podobnie, rozpatrując drogi $C(\pi(a^v),\pi(b^v))$ i $C'(\pi(a^v),\pi'(b^v))$ można pokazać, że

$$L'(\pi'(a^v),\pi'(b^v)) = L(\pi(a^v),\pi(b^v)) + p_{\pi(k)}.$$

Z tego wynika, że $L'(c,s) = L(c,s)$.

Reasumując, długość pewnej drogi $C'(c,s)$ w grafie $G(\Theta')$ jest równa $C(s,t) = C_{\max}(\Theta)$. Wobec tego $C_{\max}(\Theta') \ge C_{\max}(\Theta)$, co kończy dowód twierdzenia.

Wobec tego aby przez wykonanie *t-ruchu* wygenerować ewentualnie lepsze rozwiązanie należy pierwszą lub ostatnią operację bloku przenieść przed pierwszą lub za ostatnią operację innego bloku.

Udowodnione w tym rozdziale Twierdzenia 2 i 3 dotyczą dopuszczalności rozwiązań generowanych przez *t-ruchy*. Jeżeli znane są drogi pomiędzy dowolnymi parami wierzchołków w grafie, wówczas to sprawdzenie jest wykonywane w czasie stałym. Z kolei

Twierdzenia 5 i 6 wyrażają tzw. własności eliminacyjne bloków. Umożliwiają bowiem pominięcie w procedurze generowania otoczenia ruchów, które nie generują lepszych, od bieżącego, rozwiązań. Ze zbioru ruchów $\mathrm{T}(\Theta)$ generujących otoczenie rozwiązania Θ będziemy pomijali wszystkie ruchy spełniające założenia twierdzeń 2,3,5 i 6. Wobec tego do generowania otoczenia rozwiązania Θ będą stosowane *t-ruchy* ze zbioru

$$\mathrm{T}^{acc}(\Theta) = \mathrm{T}(\Theta) \setminus (\mathrm{T}^{noacc} \cup \mathrm{T}^{out})$$

gdzie T^{noacc} jest zbiorem ruchów generujących rozwiązania niedopuszczalne (tw. 2 i 3), a T^{out} zbiorem ruchów generujących rozwiązania o wartości funkcji celu nie mniejszej niż $C_{\max}(\Theta)$ (tw. 5 i 6).

Jeżeli $\mathrm{B} = [B^1, B^2, \ldots, B^r]$ jest ciągiem bloków z drogi krytycznej w grafie $G(\Theta)$, wówczas zbiór $\mathrm{T}^{acc}(\Theta)$ zawiera ruchy przestawiające pierwszą (lub ostatnią) operację każdego bloku na inną (z tego samego gniazda) maszynę. Jeżeli $\pi(v)$ jest pierwszą (lub ostatnią) operacją pewnego bloku oraz M_j jest maszyną z tego samego gniazda, wówczas zbiór $\mathrm{T}^{acc}(\Theta)$ zawiera ruchy przestawiające $\pi(v)$ na następujące pozycje: $\eta_j(v), \eta_j(v+1), \ldots, \rho_j(v)$. Okazało się, że generowane przez te wszystkie ruchy otoczenie jest duże i zawiera wiele "złych" ruchów. Ograniczyliśmy się więc do ruchów przestawiających pierwszą (lub ostatnią) operację $\pi(v)$ bloku **jedynie** na pozycję $\eta_j(v)$ lub $\rho_j(v)$. Ostatecznie więc

$$\mathrm{T}^{subm}(\Theta) = \{t_j^i(v,w) \in \mathrm{T}^{acc} : w \in \{a^k, b^k\},\ w \in \{\eta_j(v), \rho_j(v)\},\ k = 1,2,\ldots,r\}. \quad (7)$$

Ostatecznie więc otoczeniem Θ jest zbiór rozwiązań dopuszczalnych

$$\mathrm{N}(\Theta) = \{\tau(\Theta) : \tau \in \mathrm{T}^{subm}(\Theta)\} \quad (8)$$

Otoczenie to zawiera $4 \times r$ elementów, gdzie r jest liczbą bloków drogi krytycznej.

7. Metody szacowania wartości funkcji celu

Każde rozwiązanie $\Theta = (\mathrm{Q}, \pi)$ jest parą, której pierwszym elementem jest ciąg zbiorów - przydziałów operacji do maszyn. Nowy przydział będziemy wyznaczali wybierając z otoczenia (8) element o najmniejszej wartości funkcji celu. Wymaga to wyznaczenia, dla każdego elementu otoczenia, drogi krytyczne. Aby przyśpieszyć tę procedurę, jako kryterium wyboru będziemy stosowali obliczane w czasie stałym dolne ograniczenia wartości funkcji celu. W tym rozdziale przedstawimy metody wyznaczania takich ograniczeń.

Niech $\Theta = (\mathrm{Q}, \pi)$ będzie rozwiązaniem dopuszczalnym, gdzie $\mathrm{Q} = [\mathrm{Q}^1, \mathrm{Q}^2, \ldots, \mathrm{Q}^m]$ jest przydziałem operacji do maszyn, a $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ konkatenacją *m* permutacji. Dalej, niech $\mathrm{B} = [B^1, B^2, ..., B^r]$ będzie ciągiem bloków drogi krytycznej w grafie $G(\Theta)$.

Rozpatrujemy dwie maszyny M_i oraz M_j należące do tego samego gniazda. Na maszynie M_i są wykonywane operacje ze zbioru Q^i w kolejności $\pi_i = (\pi_i(1), \pi_i(2), \ldots, \pi_i(\rho_i))$, a na maszynie M_j operacje ze zbioru Q^j w kolejności $\pi_j = (\pi_j(1), \pi_j(2), \ldots, \pi_j(\rho_j))$. Załóżmy, że blok

$$B^k = (\pi_i(a^k), \pi_i(a^k+1), \ldots, \pi_i(b^k-1), \pi_i(b^k)),$$

zawiera operacje wykonywane na maszynie M_i. Dla uproszczenia zapisu pomijamy indeks *k* oznaczający numer bloku. Wobec tego $\pi_i(a)$ jest pierwszą, a $\pi_i(b)$ ostatnią operacją bloku B^k.

Zgodnie ze strategią przeszukiwania otoczenia $\mathsf{N}(\Theta)$ szukamy takiego ruchu $\tau \in \mathsf{T}^{\mathsf{N}}$, który wygeneruje graf $G(\tau(\Theta)$ - rozwiązanie dopuszczalne o możliwie najmniejszej wartości oszacowania długości drogi krytycznej (tj. wartości funkcji celu).

Dla ruchów z T^{N} przestawiających pierwszą operację $\pi_i(a^k)$ bloku B^k na pozycję $\eta_j(a^k)$ lub $\rho_j(a^k)$ w permutacji π_j wprowadzamy oznaczenia

$$\Delta^{a^k}_{\mathrm{x}(a^k)} = \max\{L_1^{\mathrm{x}(a^k)}, L_2^{\mathrm{x}(a^k)}, L_3^{\mathrm{x}(a^k)}, L_4^{\mathrm{x}(a^k)}\},\ \ \mathrm{x}(a^k) \in \{\eta_j(a^k), \rho_j(a^k)\},$$

gdzie

$$\begin{aligned}
L_1^{\mathrm{x}(a^k)} &= L(s, \pi_i(a^k-1)) - L(s, \pi_i(a^k)),\\
L_2^{\mathrm{x}(a^k)} &= L(s, \pi_i(a^k+1)) - L(s, \pi_i(a^k)) - p_{\pi_i(a^k+1)},\\
L_3^{\mathrm{x}(a^k)} &= L(s, \pi_j(1)) + \sum_{h=2}^{w-1} p_{\pi_i(h)} + p_{\pi_i(a^k)} + L(\pi_j(w), c)\\
&- \ L(s, \pi_i(a^k)) - \sum_{h=a^k+1}^{b^k-1} p_{\pi_i(h)} - L(\pi_i(b^k), c),\\
L_4^{\mathrm{x}(a^k)} &= \sum_{h=\mathrm{x}(a^k)+1}^{w-1} p_{\pi_j(h)} + L(\pi_j(w), c) - \sum_{h=a^k+1}^{b^k-1} p_{\pi_i(h)} - L(\pi_i(b^k), c).
\end{aligned}$$

Podobnie, dla ruchów z T^{N} przestawiających ostatnią operację $\pi_i(b^k)$ bloku B^k na pozycję $\eta_j(a^k)$ lub $\rho_j(a^k)$ w permutacji π_j wprowadzamy oznaczenia

$$\Delta^{b^k}_{\mathrm{y}(b^k)} = \max\{L_1^{\mathrm{y}(b^k)}, L_2^{\mathrm{y}(b^k)}, L_3^{\mathrm{y}(b^k)}, L_4^{\mathrm{y}(b^k)}\},\ \ \mathrm{y}(b^k) \in \{\eta_j(b^k), \rho_j(b^k)\},$$

gdzie

$$\begin{aligned}
L_1^{\mathrm{y}(b^k)} &= L(\pi_i(b^k-1), c) - p_{\pi(b^k-1)} - L(\pi_i(b^k), c)\\
L_2^{\mathrm{y}(b^k)} &= L(\pi_i(b^k+1), c) - L(\pi_i(b^k), c),
\end{aligned}$$

$$L_3^{\mathrm{y}(b^k)} \;=\; L(s,\pi_j(1)) + \sum_{h=2}^{w-1} p_{\pi_j(h)} + p_{\pi_i(b^k)} + L(\pi_j(w),c)$$

$$-\;\; L(s,\pi_i(a^k)) - \sum_{h=a^k+1}^{b^k-1} p_{\pi_i(h)} - L(\pi_i(b^k),c),$$

$$L_4^{\mathrm{y}(b^k)} \;=\; L(s,\beta(1)) + \sum_{h=2}^{\mathrm{y}(b^k)-1} p_{\pi_j(h)} - L(s,\pi_i(a^k)) - \sum_{h=a^k+1}^{b^k-1} p_{\pi_i(h)}.$$

Obecnie udowodnimy twierdzenia pozwalające na szacowanie wartości funkcji celu dla rozwiązania wygenerowanego z Θ przez przestawienia pierwszej operacji $\pi_i(a)$ z bloku B^k (wykonanie *t-ruchu*) na pozycję $\eta_j(a)$ lub $\rho_j(a)$ na maszynie M_j.

Twierdzenie 7 *Jeżeli rozwiązanie* $\Theta' = (\mathrm{Q}',\pi')$ *zostało wygenerowane z* $\Theta = (\mathrm{Q},\pi)$ *przez wykonanie ruchu* $t_j^i(a^k, \mathrm{x}(a^k)) \in \mathrm{T}^{\mathrm{N}}$, $\mathrm{x}(a^k) \in \{\eta_j(a^k), \rho_j(a^k)\}$ *to*

$$L'(s,c) \geq L(s,c) + \Delta_{\mathrm{x}(a^k)}^{a^k}.$$

Dowód. Niech $\pi_i = (\pi_i(1),\pi_i(2),\ldots,\pi_i(\rho_i))$ oraz $\pi_j = (\pi_j(1),\pi_j(2),\ldots,\pi_j(\rho_j))$ będą permutacjami operacji wykonywanych odpowiednio na maszynie M_i oraz M_j. Ciąg operacji $B^k = [\pi_i(a^k),\pi_i(a^k+1),\ldots,\pi_i(b^k)]$ $(1 \leq a^k \leq b^k \leq \rho_i)$ jest blokiem na maszynie M_i, tj. B^k jest podciągiem π_i.

Dla uproszczenia zapisu przyjmujemy, że $a = a^k, b = b^k$ oraz $\alpha = \pi_i = (\alpha(1),\ldots,\alpha(a),\ldots,\alpha(b),\ldots,\alpha(u))$, $\beta = \pi_j = (\beta(1),\ldots,\beta(w))$, gdzie $u = \rho_i, w = \rho_j$.

Graf $G(\Theta)$ jest acykliczny, więc istnieje droga krytyczna $C(s,c)$ o długości $L(s,c)$. Dla każdego wierzchołka $v \in \mathrm{O}$, zachodzi

$$C(s,c) = (C(s,v),C(v,c))$$

oraz

$$L(s,c) = L(s,v) + C(v,c) - p_v.$$

Wierzchołki drogi krytycznej można rozbić na podciągi

$$C(s,c) = (C(s,\alpha(a)),C(\alpha(a),\alpha(b)),C(\alpha(b),c))) \tag{9}$$

Rozpatrujemy kolejno następujące drogi: $d_1(s,c)$, $d_2(s,c)$, $d_3(s,c)$ i $d_4(s,c)$ z wierzchołka s do c w grafie $G(\Theta')$. Symbolicznie są one przedstawione na rysunku 6.

$$d_1(s,c) \;=\; (C'(s,\alpha(a-1)),C'(\alpha(a-1),\alpha(b)),C'(\alpha(b),c)),$$

$$d_2(s,c) \;=\; ((C'(s,\alpha(a+1)),C'(\alpha(a+1),\alpha(b)),C'(\alpha(b),c)),$$

$$d_3(s,c) \;=\; (C'(s,\beta(1),C'(\beta(1),\beta(w)),C'(\beta(w),c)),$$
$$d_4(s,c) \;=\; (C'(s,\alpha(a)),C'(\alpha(a)=\beta(x(a)),\beta(w)),C'(\beta(w),c).$$

Korzystając z faktu, że
$C'(s,\alpha(a-1)) = C(s,\alpha(a-1))$ oraz $C'(\alpha(b),c)) = C(\alpha(b),c))$
otrzymujemy
$$d_1(s,c) \;=\; (C(s,\alpha(a-1)),C'(\alpha(a-1),\alpha(b)),C(\alpha(b),c))$$
oraz podobnie
$$d_2(s,c) \;=\; ((C(s,\alpha(a+1)),C'(\alpha(a+1),\alpha(b)),C(\alpha(b),c)),$$
$$d_3(s,c) \;=\; (C(s,\beta(1),C'(\beta(1),\beta(w)),C(\beta(w),c)),$$
$$d_4(s,c) \;=\; (C(s,\alpha(a)),C'(\alpha(a)=\beta(x(a)),\beta(w)),C(\beta(w),c).$$
Wobec tego długości tych dróg (w grafie $G(\Theta')$) można można wyrazić poprzez długości pewnych dróg w grafie $G(\Theta)$. Wynoszą one odpowiednio:
$$l^1(s,c) \;=\; L(s,\alpha(a-1)) + \sum_{h=a+1}^{b-1} p_{\alpha(h)} + L(\alpha(b),c),$$
$$l^2(s,c) \;=\; L(s,\alpha(a+1)) + \sum_{h=a+2}^{b-1} p_{\alpha(h)} + L(\alpha(b),c),$$
$$l^3(s,c) \;=\; L(s,\beta(1)) + \sum_{h=2}^{w-1} p_{\beta(h)} + p_{\alpha(a)} + L(\beta(w),c),$$
$$l^4(s,c) \;=\; L(s,\alpha(a) + \sum_{h=x(a)+1}^{w-1} p_{\beta(h)} + L(\beta(w),c).$$
Ponieważ graf $G(\Theta')$ jest acykliczny, więc istnieje droga krytyczna $C'(s,c)$, której długość nie może być krótsza od długości dowolnej innej drogi z wierzchołka s do c w $G(\Theta')$. Dlatego też

$$L'(s,c) \;\geq\; l^1(s,c),\; L'(s,c) \geq l^2(s,c),\; L'(s,c) \geq l^3(s,c),\; L'(s,c) \geq l^4(s,c).$$
Stąd korzystając z (9), otrzymujemy

$$L'(s,c) \;\geq\; \max\{l^1(s,c), l^2(s,c), l^3(s,c), l^4(s,c)\} = \max\{L(s,\alpha(a-1))$$

$$+ \sum_{h=a+1}^{b-1} p_{\alpha(h)} + L(\alpha(b),c),\; L(s,\alpha(a+1)) + \sum_{h=a+2}^{b-1} p_{\alpha(h)} + L(\alpha(b),c)),$$
$$L(s,\beta(1)) + \sum_{h=2}^{w-1} p_{\alpha(h)} + p_{\alpha(a)} + L(\beta(w),c),$$

$$L(s,\alpha(a)+\sum_{h=x(a)}^{w-1} p_{\alpha(h)} + L(\beta(w),c)\}$$

$$= \quad \max\{L(s,c)+L(s,\alpha(a-1))-L(s,\alpha(a)),$$

$$L(s,c)+L(s,\alpha(a+1))-L(s,\alpha(a))-p_{\alpha(a+1)},$$

$$L(s,c)+L(s,\beta(1))+\sum_{h=2}^{w-1} p_{\beta(h)} + p_{\alpha(a)} + L(\beta(w),c)$$

$$- \quad L(s,\pi_i(a^k)) - \sum_{h=a^k+1}^{b^k-1} p_{\pi_i(h)} - L(\pi_i(b^k),c),$$

$$L(s,c)+\sum_{h=x(a)+1}^{w-1} p_{\beta(h)} + L(\beta(w),c) - \sum_{h=a+1}^{b-1} p_{\alpha(h)} - L(\alpha(b),c)\}$$

$$= \quad L(s,c)+\max\{L_1^{x(a^k)}, L_2^{x(a^k)}, L_3^{x(a^k)}, L_4^{x(a^k)}\}$$

$$= \quad L(s,c)+\Delta_{x(a)}^{a},$$

co kończy dowód twierdzenia.

Kolejne twierdzenie jest związane z przestawianem ostatniej operacji $\pi(b^k)$ z bloku B^k na maszynę M_j.

Twierdzenie 8 *Jeżeli rozwiązanie* $\Theta' = (Q', \pi')$ *zostało wygenerowane z* $\Theta = (Q, \pi)$ *przez wykonanie ruchu* $t_j^i(b^k, y(b^k)) \in T^N$, $y(b^k) \in \{\eta_j(b^k), \rho_j(b^k)\}$ *to*

$$L'(s,c) \geq L(s,c) + \Delta_{y(b^k)}^{b^k}.$$

Dowód. Podobnie jak w dowodzie Twierdzenia 7 zakładamy, że $\pi_i = (\pi_i(1), \pi_i(2), \ldots, \pi_i(\rho_i))$ oraz $\pi_j = (\pi_j(1), \pi_j(2), \ldots, \pi_j(\rho_j))$ są permutacjami operacji wykonywanych na maszynie M_i oraz M_j, a $B^k = [\pi_i(a^k), \pi_i(a^k+1), \ldots, \pi_i(b^k)]$ $(1 \leq a^k \leq b^k \leq \rho_i)$ jest blokiem na maszynie M_i.

Dalej, dla uproszczenia zapisu przyjmujemy, że $\alpha = \pi_i = (\alpha(1), \alpha(2), \ldots, \alpha(u))$, $\beta = \pi_j = (\beta(1), \ldots, \beta(w))$, gdzie $u = \rho_i$, $w = \rho_j$, a blok

$$B^k = [\pi_i(a), \pi_i(a+1), \ldots, \pi_i(b)].$$

W wygenerowanym przez ruch $t_j^i(b^k, y(b^k))$ acyklicznym grafie $G(\Theta')$ rozpatrujemy następujące drogi:

$$d_1(s,c) \quad = \quad (C'(s,\alpha(a)), C'(\alpha(a), \alpha(b-)), C'(\alpha(b-1),c)),$$

$$
\begin{aligned}
d_2(s,c) &= ((C'(s,\alpha(a)),C'(\alpha(a),\alpha(b+1)),C'(\alpha(b+1),c)),\\
d_3(s,c) &= (C'(s,\beta(1),C'(\beta(1),\beta(w)),C'(\beta(w),c)),\\
d_4(s,c) &= (C'(s,\beta(1)),C'(\beta(1),\beta(y(b)) = \alpha(b)),C'(\beta(y(b)),c).
\end{aligned}
$$

Długości tych dróg wynoszą one odpowiednio:

$$
\begin{aligned}
l^1(s,c) &= L(s,\alpha(a)) + \sum_{h=a+1}^{b-2} p_{\alpha(h)} + L(\alpha(b-1),c),\\
l^2(s,c) &= L(s,\alpha(a)) + \sum_{h=a+1}^{b-1} p_{\alpha(h)} + L(\alpha(b+),c),\\
l^3(s,c) &= L(s,\beta(1)) + \sum_{h=2}^{w-1} p_{\beta(h)} + L(\beta(w),c),\\
l^4(s,c) &= L(s,\beta(1) + \sum_{h=2}^{\mathrm{y}(b)-1} p_{\beta(h)} + L(\beta(b),c).
\end{aligned}
$$

Ponieważ droga krytyczna $C'(s,c)$ w grafie $G(\Theta')$ jest najdłuższą drogą z wierzchołka s do c, więc jej długość

$$L'(s,c) \geq \max\{l^1(s,c),\, l^2(s,c),\, l^3(s,c),\, l^4(s,c)\}.$$

Po wykonaniu podobnych jak w dowodzie Twierdzenia 7 przekształceń, otrzymujemy:

$$L'(s,c) \geq L(s,c) + \Delta^b_{\mathrm{y}(b)},$$

co kończy dowód twierdzenia.

Przestawiając operację $\pi(a^k)$ na pozycje $\eta_j(a^k)$ lub $\rho_j(a^k)$ generuje się graf, w którym dolnym ograniczeniem długości drogi krytycznej wierzchołka s do c jest wartość wyrażenia $L(s,c)+\Delta^{a^k}_{\eta_j(a^k)}$ (lub $L(s,c)+\Delta^{a^k}_{\rho_j(a^k)}$). Tak więc wyrażenie $\Delta^{a^k}_{\mathrm{x}(a^k)}$, $\mathrm{x}(a^k)\in\{\eta_j(a^k),\, \rho_j(a^k)\}$ może być wykorzystane do wyboru operacji (tj. elementu z otoczenia), która będzie przestawiana.

Podobnie, $L(s,c)+\Delta^{b^k}_{\eta_j(b^k)}$ (lub $L(s,c)+\Delta^{b^k}_{\rho_j(b^k)}$) jest dolnym ograniczeniem długości drogi krytycznej w grafie wygenerowanym przez przestawienie operacji $\pi(b^k)$ na pozycje $\eta_j(b^k)$ lub $\rho_j(b^k)$ i wyrażenie $\Delta^{b^k}_{\mathrm{y}(b^k)}$, $\mathrm{y}(b^k)\in\{\eta_j(b^k),\, \rho_j(b^k)\}$ może być wykorzystane do elementu z otoczenia.

Wybieramy operację $\pi(v)\in O$ taką, że

$$\Delta^v_{\chi(v)} = \min_{1\leq k\leq r}\{\Delta^z_{\mu(z)} : z\in\{a^k,b^k\},\ \mu(z)\in\{\eta_j(z),\rho_j(z)\}\} \tag{10}$$

Minimalna wartość $\Delta^v_{\chi(v)}$ odpowiada wówczas „najlepszemu" *t-ruchowi* polegającemu na przaestawieniu pierwszą lub ostatnią operację z pewnego bloku na inną maszynę.

Z Twierdzenia 7 i 8 wynika, że jeśli $\Delta^{v}_{\chi(v)} > 0$, to w wygenerowanym grafie $G(\Theta')$ długość drogi krytycznej $L'(s,c) > L(s,c)$.

Reasumując, dla rozwiązania $\Theta = (\mathsf{Q},\pi)$ (ustalonego przydziału operacji do maszyn Q) proponowanym następującą metodę wyznaczenia nowego przydziału Q'. W grafie $G(\Theta)$ wyznaczamy drogę krytyczną $C(s,c)$ (jeżeli jest więcej niż jedna, to wybieramy dowolną z nich) oraz obliczamy jej długość $L(s,c) = C_{\max}(\Theta)$. Następnie, wyznaczamy podział drogi na bloki $\mathsf{B} = [B^1, B^2, \ldots, B^r]$ oraz zgodnie z (7) zbiór ruchów $T^{subm}(\Theta)$. Korzystając z (10) wyznaczamy $\Delta^{v}_{\chi(v)}$ i wybieramy "najlepszy" *t-ruch* $t^{i}_{j}(v,\chi(v))$. Ruch ten generuje rozwiązanie (nowy przydział operacji do maszyn) z otoczenia $\mathsf{N}(\Theta)$ o najmniejszej wartości dolnego ograniczenia funkcji celu.

8. Eksperymenty obliczeniowe

Zaproponowana metoda szacowania wartości funkcji celu została zastosowana w algorytmie poszukiwania z zabronieniami (*tabu search*) dla rozważanego problemu gniazdowego z równoległymi miaszynami. Algorytm był testowany na przykładach podanych przez Hurink'a [5] i uruchomiony na komputerze HP xw4600 z procesorem Intel Core 2 Duo 3.16GHz pracującym pod kontrolą systemu operacyjnego Linux Fedora 12. Uzyskane wyniki prezentuje Tabela 1.

Tab. 1. Wyniki eksperymentów obliczeniowych na przykładach Hurink'a [5]

problem	$n \times m$	Flex.	t [ms]		C_{max}	
			TS1	TS2	TS1	TS2
abz5	10×10	2	0.1125	0.0139	1146	1181
abz6	10×10	2	0.1581	0.0198	933	952
abz7	20×15	2	0.4001	0.0201	637	641
abz8	20×15	2	0.3221	0.0158	635	640
abz9	20×15	2	0.3743	0.0169	647	656

Poszczególne kolumny Tabeli 1. oznaczają:

- Flex. – średnia liczba maszyn równoległych na operację,
- TS1 – algorytm *tabu search* z dokładnie wyznaczaną wartością funkcji celu,
- TS2 – algorytm *tabu search* z **szacowaną** wyznaczaną wartością funkcji celu,
- t – czas działania funkcji wyznaczania (lub szacowania) wartości funkcji celu dla wszystkich rozwiązań generowanych przez t-ruchy,
- C_{max} – wartość otrzymanej funkcji kryterialnej.

Otrzymane rezultaty pokazują, że użycie modułu szacowania wartości funkcji celu zamiast wyznaczania jej dokładnej wartości skutkuje skróceniem czasu działania algorytmu. Praktycznie nie ma różnicy w czasie działania procedury szacującej dla różnych badanych rozmiarów problemów. Złożoność obliczeniowa procedury szacowania jest stała

$O(1)$. Algorytm TS2 jest szybszy nież algorytm T1 kosztem niewielkiej straty jakości otrzymanych rozwiązań. Eksperymenty obliczeniowe pokazują, że proponowane rozwiązanie jest przydatne w rozwiązywaniu przykładów o dużych rozmiarach, szczególnie przy wysokiej wartosci wskaźnika Flex.

9. Wnioski

W pracy zaproponowane zostały teoretyczne metody szacowania wartości funkcji celu przy generowaniu otoczenia w problemie gniazdowym z równoległymi maszynami. Podejście to pozwala znacznie skrócić czas działania algorytmów przybliżonych rozwiązywania rozważanego problemu, zachowując dobrą jakość otrzymywanych rozwiązań.

Literatura

1. Bożejko W., Uchroński M., Wodecki M.: The new golf neighborhood for the flexible job shop problem, Proceedings of the ICCS 2010, Procedia Computer Science 1 (2010), Elsevier, 289--296.
2. Bożejko W.: A new class of parallel scheduling algorithms, Oficyna Wydawnicza Politechniki Wrocławskiej, seria: monografie, 2010.
3. Gao J., Sun L., Gen M.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems, Computers & Operations Research 35, 2008, 2892-2907.
4. Garey M.R., Johnson D.S., Sethi R.: The complexity of the flowshop and jobshop scheduling, Math. Oper. Res. 1 (2), 1974, 117-128.
5. Hurink E., Jurisch B., Thole M.: Tabu search for the job shop scheduling problem with multi-purpose machine, Operations Research Spektrum 15, 1994, 205-215.
6. Mastrolilli M., Gambardella L.M.: Effective neighborhood functions for the flexible job shop problem, Journal of Scheduling 3(1), 2000, 3-20.
7. Nowicki E., Smutnicki C.: A fast tabu search algorithm for the permutation flow-shop problem, European Journal of Operational Research 91, 1996, 160-175.
8. Pinedo M.: Scheduling: theory, algorithms and systems, Englewood cliffs, NJ: Prentice-Hall, 2002.

Dr Wojciech BOŻEJKO
Mgr inż. Mariusz UCHROŃSKI
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska
e-mail: wojciech.bozejko@pwr.wroc.pl
mariusz.uchronski@pwr.wroc.pl

Dr Mieczysław WODECKI
Instytut Informatyki
Uniwersytet Wrocławski
e-mail: mwd@ii.uni.wroc.pl