ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1984
Seria: AUTOMATYKA z. 74 Nr kol. 810

Franciszek Marecki
Politechnika Śląska

ALOKACJA OPERACJI NA RÓWNOLEGŁEJ LINII MONTAŻOWEJ

Streszczenie. W pracy sformułowano i rozwiązano problem alokacji operacji na równoległej linii montażowej. Istota tego problemu polega na określeniu minimalnej liczby stanowisk pracy i wyznaczeniu ich lokalizacji na stacjach montażowych. Na stacji montażowej mogą się znajdować dwa równoległe stanowiska pracy, po obu stronach taśmy montażowej. Problem alokacji operacji rozwiązano metodą programowania wieloetapowego.

1. Wprowadzenie

Linia montażowa składa się z transportera /lub taśmy/, na którym w równych odstępach znajdują się montowane obiekty. Wzdłuż linii rozmieszczone są stacje montażowe tak, by na każdej stacji znajdował się tylko jeden obiekt. Na linii szeregowej na każdej stacji montażowej znajduje się jedno stanowisko pracy. Wówczas operacje montażu obiektu są wykonywane sekwencyjnie. Na linii równoległej na stacjach montażowych mogą się znajdować dwa stanowiska pracy, po obydwu stronach transportera. W tym przypadku w trakcie montażu obiektu, niektóre operacje są wykonywane równolegle.

Problem balansowania linii montażowej, sformułowany w [22] i [23], można przedstawić następująco. Dany jest zbiór operacji z relacją kolejności ich wykonywania oraz czasy operacji/*. Dla zadanego cyklu linii należy wyznaczyć minimalną liczbę podzbiorów operacji /tworzących stanowiska pracy/. Cykl jest odstępem czasu pomiędzy zejściem z linii dwóch kolejnych obiektów. Inaczej mówiąc cykl jest czasem jakim dysponuje monter dla wykonania operacji na swym stanowisku pracy, w trakcie montowania jednego obiektu.

Do rozwiązania problemu balansowania szeregowej linii montażowej zaproponowano wiele różnych algorytmów [2], [24]. Ponieważ jest to problem NP-zupełny w sensie złożoności obliczeniowej [4], [10], [25], dlatego prawdopodobnie nie istnieje wielomianowy algorytm wyznaczania rozwiązania optymalnego. Z tego też względu stosowane są algorytmy przeglądowe lub heurystyczne.

W licznych publikacjach dotyczących balansowania linii nie uwzględniano wielu istotnych ograniczeń procesu montażu [16]. Ograniczenia te prowadzą do wyróżnienia stacji montażowych i stanowisk pracy [14], [17], [20]. W tym przypadku pojawia się problem alokacji stanowisk pracy do stacji montażowych. Problem ten można rozwiązywać w dwóch fazach - najpierw

* w niektórych publikacjach operacja jest nazwana czynnością.

balansowanie, a następnie alokacja stanowisk pracy /podzbiorów operacji/ do stacji montażowych. Jednakże przy ograniczeniach lokalizacji operacji na stacjach montażowych można w ten sposób nie uzyskać rozwiązania dopuszczalnego /chociaż ono istnieje/. A zatem problem ten wymaga równoczesnego przydzielania operacji do stacji montażowych i tworzenia stanowisk pracy. Zagadnienie takie nazywane jest alokacją operacji na linii montażowej [18].

W problemach balansowania linii lub alokacji operacji zadany jest cykl linii. Jeżeli linia szeregowa ma ograniczoną liczbę stacji, to przy zbyt krótkim cyklu rozwiązanie problemu balansowania /lub alokacji/ nie jest możliwe. W takim przypadku powstaje problem alokacji operacji na linii równoległej [21]. W praktyce przemysłowej linie równoległe występują często [3]. Pomimo to, jak dotąd. nie opublikowano algorytmu rozwiązania problemu alokacji operacji na linii równoległej. W niniejszej pracy problem ten zostanie rozwiązany za pomocą algorytmu programowania wieloetapowego [19].

2. Sformułowanie problemu

Załóżmy, że dany jest zbiór operacji montażowych

$$\Omega = \{\omega_n\}_{n = 1,\ldots,N} \qquad /1/$$

gdzie: ω_n - n-ta operacja montażowa; N - liczba operacji.
Dla zmontowania obiektu muszą być wykonane wszystkie operacje ze zbioru /1/.

Niech linia montażowa składa się z K stacji. Na każdej stacji montażowej mogą być zlokalizowane co najwyżej dwa stanowiska pracy /po lewej oraz po prawej stronie transportera/.

Załóżmy, że dany jest cykl montażu c, który jest czasem pobytu obiektu na stacji. A zatem każdy obiekt przebywa na linii montażowej w ciągu czasu $K \cdot c$. Tym samym czas montażu obiektu na linii szeregowej nie może przekroczyć $K \cdot c$, natomiast na linii równoległej 2Kc.

Niech czasy wykonywania operacji będą dane wektorem

$$\Theta = [\vartheta_n]_{n = 1,\ldots,N} \qquad /2/$$

gdzie: ϑ_n - czas wykonywania operacji ω_n.
Przyjmiemy, że czasy operacji spełniają warunek

$$\max_{1 \leq n \leq N} \vartheta_n \leq c < \sum_{n=1}^{n=N} \vartheta_n \qquad /3/$$

A zatem każda operacja może być wykonana w całości na jednym stanowisku pracy. W dalszym ciągu przyjmujemy, że operacje są niepodzielne.

Załóżmy, że relacja kolejności wykonywania operacji dana jest macierzą

$$\Gamma = \left[\gamma_{\nu,n}\right] \qquad /4/$$

$$\nu = 1,\ldots,N$$
$$n = 1,\ldots,N$$

Elementy tej macierzy określamy następująco

$$\gamma_{\nu,n} = \begin{cases} 1 : \text{jeśli operacja } \omega_\nu \text{ jest bezpośrednim poprzednikiem operacji } \omega_n \\ 0 : \text{w przypadku przeciwnym} \end{cases} \qquad /4a/$$

Jeżeli jest spełniony warunek

$$K < \left[\frac{1}{c}\sum_{n=1}^{n=N} \vartheta_n\right]^+ \qquad /5/$$

gdzie: $[\cdot]^+$ - najmniejsza liczba naturalna nie mniejsza od wartości w nawiasie kwadratowym

to na linii montażowej muszą być zorganizowane równoległe stanowiska pracy. Gdy warunek /5/ nie jest spełniony można rozwiązywać problem balansowania /lub alokacji operacji/ dla linii szeregowej. Jednakże w przypadku braku rozwiązania dopuszczalnego - pozostaje do rozwiązania problem alokacji operacji na linii równoległej.

Jako kryterium optymalizacji alokacji operacji na linii równoległej przyjmiemy minimalizację sumarycznego nie wykorzystanego czasu pracy monterów

$$Q = \sum_{m=1}^{m=M} q_m \longrightarrow \min \qquad /6/$$

gdzie: q_m - nie wykorzystany czas pracy na m-tym stanowisku pracy,
M - liczba stanowisk pracy /monterów/

przy tym

$$q_m = c - \sum_{\omega_n \in \Omega_m} \vartheta_n \geqslant 0 \qquad /7/$$

gdzie: Ω_m - podzbiór operacji przydzielonych do m-tego stanowiska pracy.

Podzbiór Ω_m, $m = 1,\ldots,M$, musi być dopuszczalny, tzn. musi spełniać ograniczenia czasowe /7/ oraz ograniczenia kolejnościowe wynikające z relacji /4/. Jeżeli każdy podzbiór operacji Ω_m jest dopuszczalny, to z /6/ i /7/ otrzymamy

$$Q = M \cdot c - \sum_{n=1}^{n=N} \vartheta_n \longrightarrow \min \qquad /8/$$

A zatem minimalizacja sumarycznego nie wykorzystanego czasu pracy monterów jest równoważna z minimalizacją liczby M stanowisk pracy.

Dla ścisłego sprecyzowania warunków jakie musi spełniać dopuszczalny podzbiór operacji Ω_m, $m = 1,\dots,M$, oznaczymy przez t_n chwilę zakończenia wykonywania operacji ω_n na linii. Aby rozróżnić stanowiska pracy na stacji montażowej przyjmiemy, że chwila t_n będzie poprzedzona znakiem minus dla operacji wykonywanych po lewej stronie linii /transportera/. Chwila t_n jest czasem jaki upływa od wejścia montowanego obiektu na linię, do zakończenia operacji ω_n. Tak więc możemy napisać

$$k_n = \left[\frac{1}{c}\,|\,t_n\,|\,\right]^+ \text{sign}\; t_n \qquad /9/$$

gdzie: $|k_n|$ - numer stacji, do której została przydzielona operacja ω_n. A zatem, jeżeli $k_n < 0$, to operacja ω_n jest wykonywana po lewej stronie linii na stacji o numerze $|k_n|$. W przypadku przeciwnym, gdy $k_n > 0$, operacja ω_n jest wykonywana po prawej stronie linii na stacji o numerze $|k_n|$.

Podzbiór Ω_m jest dopuszczalny, jeżeli operacje do niego należące spełniają następujące warunki

- niepodzielności

$$\forall_{1 \le n \le N} \left\{(t_n < 0) \Rightarrow \left\{\left[\frac{1}{c}\,|\,t_n|\right]^+ = \left[\frac{1}{c}\,|t_n + \vartheta_n|\right]^+\right\}\right\} \vee$$

$$\vee \left\{(t_n > 0) \Rightarrow \left\{\left[\frac{1}{c}\,t_n\right]^+ = \left[\frac{1}{c}\left(t_n - \vartheta_n\right)\right]^+\right\}\right\} \qquad /10/$$

- sumy czasów

$$\sum_{n \in I_k} \vartheta_n \le c \qquad /11/$$

przy tym

$$\forall_n \left\{\left[\frac{1}{c}\,|t_n|\right]^+ \text{sign}\; t_n = k\right\} \Rightarrow \left(n \in I_k\right) \qquad /11a/$$

- kolejności

$$\forall_{\nu} \forall_n \left(\gamma_{\nu,n} = 1\right) \Rightarrow \left(|t_\nu| \le |t_n| - \vartheta_n\right) \qquad /12/$$

Z warunku /10/ wynika, że każda operacja ω_n musi być wykonana w całości na jednym stanowisku pracy. A zatem wykluczone jest nawet wykonanie operacji ω_n równocześnie na lewym i prawym stanowisku pracy tej samej stacji montażowej. Warunek /11/ jest identyczny z /7/, z tym, że zamiast Ω_m wprowadzono I_k. Zgodnie z warunkiem /12/ operacje ω_ν i ω_n, dla których $\gamma_{\nu,n} = 1$, muszą być wykonane w określonej sekwencji czasowej.

Warto zauważyć, że dla spełnienia /12/ pomiędzy operacjami na stanowisku pracy mogą występować luzy czasowe. Efekt ten nie występuje w balansowaniu linii szeregowej. A zatem dla linii równoległej należy określić przedziały czasu, w których mają być wykonane operacje. Na podstawie tych przedziałów, zgodnie z /9/, można wyznaczyć lokalizację operacji do stacji montażowych. Z powyższych względów, dla linii równoległej mówimy o problemie alokacji operacji, zamiast o balansowaniu.

Kryterium minimalizacji liczby stanowisk pracy na linii równoległej możemy zapisać w postaci

$$Q' = \sum_{i=1}^{i=2} Q_i \longrightarrow \min \qquad /13/$$

przy tym

$$Q_i = \sum_{k=1}^{k=K} \chi_{k,i} \qquad /14/$$

a ponadto

$$\chi_{k,1} = \begin{cases} 1 : \text{jeśli} & \exists_{\nu} \left[\frac{1}{c} \mid t_\nu \mid \right]^+ \cdot \operatorname{sign} t_\nu = k < 0 \\ 0 : \text{w przypadku przeciwnym} \end{cases} \qquad /15a/$$

$$\chi_{k,2} = \begin{cases} 1 : \text{jeśli} & \exists_{n} \left[\frac{1}{c} \mid t_n \mid \right]^+ \cdot \operatorname{sign} t_n = k > 0 \\ 0 : \text{w przypadku przeciwnym} \end{cases} \qquad /15b/$$

Funkcja /13/ wyróżnia liczbę stanowisk pracy po lewej i po prawej stronie linii. Pozwala to uwzględnić dodatkowe ograniczenia dla różnicy liczb stanowisk pracy po obydwu stronach linii. Jeżeli różnica ta jest mała, to również mała jest liczba wykorzystanych stacji montażowych. W przeciwnym przypadku liczba wykorzystanych stacji montażowych jest duża. Ponadto można ograniczyć liczbę stanowisk pracy po jednej z dwu stron linii.

W sformułowanym problemie alokacji operacji na linii równoległej występuje N niewiadomych. Są to chwile zakończenia wykonywania operacji. Jeżeli wyznaczone chwile t_n spełniają ograniczenia /10/, /11/ i /12/, to rozwiązanie jest dopuszczalne. Rozwiązanie optymalne otrzymujemy wówczas, gdy chwile t_n są wyznaczone tak, by spełnione były ograniczenia, a funkcja celu /13/ osiągnęła minimum. Stanowiska pracy, do których przydzielono operacje, otrzymujemy z /9/.

3. Algorytm

Sformułowany problem alokacji operacji na równoległej linii montażowej rozwiążemy metodą programowania wieloetapowego. Idea tej metody jest oparta na wieloetapowych procesach decyzyjnych [1] oraz metodzie podziału i ograniczeń [5], [13]. Zawiera ona pewne elementy programowania dynamicznego [6], [7], [8] oraz parametry analogiczne do przedstawionych w [12] dla metody podziału i ograniczeń. W algorytmach metody programowania wieloetapowego podstawowe znaczenie mają definicje: stanu procesu decyzyjnego, funkcji wartości stanu, procedury generowania stanów oraz reguły eliminowania stanów nieperspektywicznych [19]. Algorytmy te pozwalają wyznaczyć rozwiązanie optymalne, a w przypadku limitowanego czasu obliczeń - rozwiązanie przybliżone z oszacowaniem dokładności.

3.1. Koncepcja algorytmu

Proces przydzielania N operacji do stanowisk pracy może być rozważany jako N etapowy proces decyzyjny. Na każdym η-tym etapie, $\eta=1,...,N$, wychodzimy z pewnego stanu i podejmując decyzję o przydzielaniu operacji przechodzimy do kolejnego stanu /bezpośredniego następnika/. Stan wyraża sytuację jaka pozostaje po przydzieleniu pewnych operacji. Stan początkowy /startowy/ odpowiada sytuacji, w której żadna operacja nie została jeszcze przydzielona. Stany końcowe interpretują dopuszczalne przydziały wszystkich operacji. Ciąg stanów nazwiemy trajektorią, a ciąg decyzji strategią. Wszystkie trajektorie wychodzące ze stanu startowego interpretują drzewo decyzyjne.

Z uwagi na liczbę przydzielonych operacji /podjętych decyzji/ stany można pogrupować w N podzbiorach. Stany należące do η-tego etapu, $\eta < N$. nazwiemy aktywnymi, ponieważ pozwalają wygenerować dalsze stany. Podzbiór N-tego etapu zawiera stany końcowe /lub aktualnie najlepszy stan końcowy/. Najlepszy stan końcowy /w sensie pewnego kryterium/, jest stanem globalnie optymalnym. Najlepszy stan końcowy, który można wygenerować z pewnego stanu aktywnego nazwiemy stanem lokalnie optymalnym.

Generowanie stanów polega na wyborze pewnego stanu aktywnego i uzupełnieniu go o dopuszczalną operację. Jeżeli takich operacji jest wiele, to z jednego stanu η-1-go etapu można wygenerować wiele stanów η-tego etapu. Wyboru stanu dokonujemy na podstawie pewnych reguł, np. FIFO, LIFO, DF/LLB, [12]. Można przy tym stosować jedną regułę /strategia czysta/ lub wybierać w sposób losowy różne reguły dla kolejnych stanów aktywnych /strategia mieszana/. Strategia wyboru reguł ma wpływ na efektywność algorytmu [11], [9], [15]. Oprócz reguł wyboru stanu aktywnego stosowane są reguły podziału zbioru rozwiązań dopuszczalnych, które można uzyskać z danego stanu aktywnego. W dalszym ciągu przyjmiemy podział zupełny. W tym przypadku z każdego stanu aktywnego są generowane wszystkie jego bezpośrednie następniki. Stąd wybrany stan aktywny przestaje być aktywnym. Stany są generowane na podstawie pewnych procedur, które zależą od specyfiki rozwiązywanego problemu.

W trakcie obliczeń eliminowane są stany nieperspektywiczne /czyli wiązki trajektorii/, które nie dają rozwiązania optymalnego. Do eliminacji tych stanów wykorzystamy reguły sondowania oraz dominacji.

Reguła sondowania /dla problemu minimalizacyjnego/ jest oparta na wyznaczeniu ograniczenia dolnego d i górnego g wartości v stanu lokalnie optymalnego. Definicja funkcji wartości stanu koresponduje z przyjętym kryterium optymalizacji. Jeżeli znana jest wartość v^a aktualnie najlepszego stanu końcowego, to gdy $v^a \leqslant d$, wnioskujemy, że analizowany stan jest nieperspektywiczny. Ponadto, gdy $d = g$, analizowany stan jest nieperspektywiczny /dla $v^a \leqslant g$/ lub odpowiadający mu stan lokalnie optymalny staje się aktualnie najlepszym /dla $g < v^a$/.

Dominacja stanów polega na porównaniu stanu, który został wygenerowany z danymi stanami aktywnymi. Z dwóch porównywanych stanów ten jest perspektywiczny, którego stan lokalnie optymalny jest lepszy. Reguły sondowania i dominacji zostaną przedstawione w postaci stwierdzeń.

Efektywność algorytmu może być rozważana z punktu widzenia czasu obliczeń lub zajętości pamięci komputerowej [26]. Przy zachowaniu pewnej redundancji informacji można skrócić czas obliczeń. Czas ten jest zależny od reguł wyboru i podziału ale również od komputerowych struktur danych [26], [27].

3.2. Podstawowe definicje

Załóżmy, że stany ponumerujemy oddzielnie w ramach każdego etapu. Stan procesu decyzyjnego zdefiniujemy tak, by można było z niego odczytać przydział operacji do stanowisk pracy.

Def. 1. : Stan jest wektorem

$$P^{l,\eta} = \left[p_n^{l,\eta}\right] \qquad /10/$$

$$n = 1,\ldots,N$$
$$\eta = 0,\ldots,N$$
$$l = 1,\ldots,L$$

gdzie: η - numer etapu decyzyjnego, l - numer stanu w ramach η-tego etapu, L_η - liczba stanów η-tego etapu.

Elementy tego wektora określamy następująco

$$p_n^{l,\eta} = \begin{cases} t_n & : \text{jeśli operacja } \omega_n \text{ została przydzielona} \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \qquad /10a/$$

Sposób wyznaczania chwili t_n zostanie przedstawiony przy omawianiu generowania stanów.

A zatem stan początkowy $P^{1,0}$ ma wszystkie współrzędne równe zeru. Każdy stan końcowy $P^{L,N}$ ma wszystkie współrzędne różne od zera. Stanowisko pracy, do którego przydzielono operację ω_n otrzymujemy z formuły

$$k_n = \left[\frac{1}{c}\ p_n^{1,N}\right]^+ \qquad /17/$$

Dla oceny stanu $P^{1,\eta}$ zdefiniujemy funkcję wartości stanu, którą oznaczymy przez $V^{1,\eta}$

Def. 2. Funkcja wartości stanu ma postać

$$V^{1,\eta} = V\left(P^{1,\eta}\right) \qquad /18/$$

przy tym

$$V^{1,\eta} = \left[v_i^{1,\eta}\right] \qquad i = 1,2 \qquad /19/$$

a ponadto

$$v_i^{1,\eta} = \sum_{k=1}^{k=K_i^{1,\eta}} \chi_{k,i} \qquad /20/$$

gdzie: $K_1^{1,\eta}$ $\left(K_2^{1,\eta}\right)$ - numer ostatniej stacji, na której po lewej /po prawej/ stronie linii znajduje się ostatnie stanowisko pracy, w stanie $P^{1,\eta}$ Numery te wyznaczamy następująco

$$K_i^{1,\eta} = \left[\frac{1}{c}\ T_i^{1,\eta}\right]^+ \qquad i = 1,2 \qquad /21/$$

przy tym

$$T_1^{1,\eta} = \left|\min_{1 \leq j \leq N} p_j^{1,\eta}\right| \qquad /22a/$$

oraz

$$T_2^{1,\eta} = \max_{1 \leq j \leq N} p_j^{1,\eta} \qquad /22b/$$

Jak wynika z /22a/ i /22b/, $T_1^{1,\eta}$ i $T_2^{1,\eta}$ są odpowiednio chwilami zakończenia montażu operacji po lewej i po prawej stronie linii, w stanie $p^{1,\eta}$.

W stanie $P^{1,\eta}$ dla każdej operacji ω_n, można wyznaczyć chwilę $T_{\gamma,n}^{1,\eta}$ zakończenia jej bezpośrednich poprzedników. Zatem

$$T_{\gamma,n}^{1,\eta} = \max_{\nu \in \Gamma_n} \left|p_\nu^{1,\eta}\right| \qquad /23/$$

przy tym

$$\forall_{\nu} \; (\gamma_{\nu,n} = 1) \Rightarrow (\nu \in \Gamma_n) \qquad /23a/$$

Z przyjętego kryterium optymalizacji /13/ oraz definicji funkcji wartości stanu /18/ wynika, że globalnie optymalny stan końcowy P^o można wyznaczyć z warunku

$$\left[\min_{1 \le l \le N} \left(v_1^{l,N} + v_2^{l,N}\right) = v_1^{l^o,N} + v_2^{l^o,N}\right] \Rightarrow \left(P^{l^o} = P^o\right) \qquad /24/$$

Ze stanu P^o wyznaczamy optymalny przydział operacji do stanowisk pracy, zgodnie z /17/. W trakcie obliczeń stany końcowe $P^{l,N}$ są generowane sekwencyjnie. A zatem zapamiętywany jest jedynie aktualnie najlepszy stan końcowy P^n. Ostatni ze stanów aktualnie najlepszych jest stanem globalnie optymalnym.

3.3. Generowanie stanów

Oznaczmy przez $P^{\lambda,\eta-1}$ wybrany / za pomocą dowolnej reguły/ stan aktywny. Załóżmy, że reguła podziału jest zupełna. Procedura generowania stanów polega na uzupełnieniu stanu $P^{\lambda,\eta-1}$ o dopuszczalną operację ω_n. W rezultacie otrzymujemy dwa stany, P^1 - przydzielając operację ω_n na lewą stronę linii, oraz P^2 - przydzielając operację ω_n na prawą stronę linii. Stany P^1 i P^2 otrzymają odpowiednie indeksy stanów aktywnych jeśli okażą się perspektywiczne.

Ogólna procedura generowania stanów ma postać

$$\forall_n \forall_{\nu} \; \left(P_n^{\lambda,\eta-1} = 0\right) \wedge \left[\left(\gamma_{\nu,n} = 1\right) \Rightarrow \left(P_{\nu}^{\lambda,\eta-1} \neq 0\right)\right] \Rightarrow \left[\left(P^1 = P^{\lambda,\eta-1} + \Delta P^1\right) \wedge \left(P^2 = P^{\lambda,\eta-1} + \Delta P^2\right)\right] \qquad /25/$$

Wektory ΔP^1 i ΔP^2 są określone następująco

$$\Delta p_j^1 = \begin{cases} t_n^1 & : \text{ dla } j = n \\ 0 & : \text{ w przypadku przeciwnym} \end{cases} \qquad /26a/$$

oraz

$$\Delta p_j^2 = \begin{cases} t_n^2 & : \text{ dla } j = n \\ 0 & : \text{ w przypadku przeciwnym} \end{cases} \qquad /26b/$$

Chwilę t_n^1 wyznaczamy z następujących formuł:

Jeśli $\quad T_2^{\lambda,\eta-1} \le T_1^{\lambda,\eta-1}$

to wówczas

$$t_n^1 = \begin{cases} -\left(T_1^{\lambda,\eta-1} + \vartheta_n\right): \text{ jeżeli } K_1^{\lambda,\eta-1}.c - T_1^{\lambda,\eta-1} \geqslant \vartheta_n \\ -\left(K_1^{\lambda,\eta-1}.c + \vartheta_n\right) : \text{ w przypadku przeciwnym} \end{cases} \qquad /28/$$

W przypadku, gdy nierówność /27/ nie jest spełniona - otrzymujemy

$$t_n^1 = \begin{cases} -\left[\max\left(T_1^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right) + \vartheta_n\right] : \text{ jeżeli} \\ \left[\frac{1}{c}\max\left(T_1^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right)\right]^+ = \left[\frac{1}{c}\left\{\max\left(T_1^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right) + \vartheta_n\right\}\right]^+ \\ -\left\{\left[\frac{1}{c}\max\left(T_1^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right)\right]^+.c + \vartheta_n\right\} : \text{ w przypadku przeciwnym} \end{cases} \qquad /29/$$

Analogicznie dla chwili t_n^2 otrzymamy:

Jeśli
$$T_1^{\lambda,\eta-1} \leqslant T_2^{\lambda,\eta-1} \qquad /30/$$

to

$$t_n^2 = \begin{cases} T_2^{\lambda,\eta-1} + \vartheta_n : \text{ jeżeli } K_2^{\lambda,\eta-1}.c - T_2^{\lambda,\eta-1} \geqslant \vartheta_n \\ K_2^{\lambda,\eta-1}.c + \vartheta_n : \text{ w przypadku przeciwnym} \end{cases} \qquad /31/$$

Gdy nierówność /30/ nie jest spełniona, to otrzymamy

$$t_n^2 = \begin{cases} \max\left(T_2^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right) + \vartheta_n : \text{ jeżeli} \\ \left[\frac{1}{c}\max\left(T_2^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right)\right]^+ = \left[\frac{1}{c}\left\{\max\left(T_2^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right) + \vartheta_n\right\}\right]^+ \\ \left[\frac{1}{c}\max\left(T_2^{\lambda,\eta-1}, T_{\gamma,n}^{\lambda,\eta-1}\right)\right]^+.c + \vartheta_n : \text{ w przypadku przeciwnym} \end{cases} \qquad /32/$$

A zatem wychodząc ze stanu $P^{1,0}$ można, za pomocą przedstawionej procedury, wygenerować wszystkie stany. Stan P^1 jest dopuszczalny jeśli $t_n^1 \leq K \cdot c$, analogicznie dla stanu P^2 sprawdzamy warunek dopuszczalności $t_n^2 \leq K \cdot c$. A zatem ogólną procedurę generowania stanów /25/ można rozbić na dwie części /dla prawej i lewej strony linii/, w których należy uwzględnić odpowiednie warunki dopuszczalności.

W trakcie generowania stanów można obliczać współrzędne wektorów V^1 i V^2. A zatem, zamiast definicji /18/, stosujemy formułę rekurencyjną

$$V = V^{\lambda,\eta-1} + \Delta V \qquad /33/$$

Dla stanu P^1 otrzymujemy

$$\Delta v_1^1 = \begin{cases} 1 : \text{ jeśli } /t_n^1/ - T_1^{\lambda,\eta-1} < \vartheta_n \\ 0 : \text{ w przypadku przeciwnym} \end{cases} \qquad /34a/$$

$$\Delta v_2^{\,1} = 0 \qquad /34b/$$

Analogicznie dla stanu P^2 wyznaczymy

$$\Delta v_1^{\,2} = 0 \qquad /35a/$$

$$\Delta v_2^{\,2} = \begin{cases} 1 : \text{jeśli} \quad t_n^{\,2} - T_2^{\lambda,\eta-1} < \vartheta_n \\ 0 : \text{w przypadku przeciwnym} \end{cases} \qquad /35b/$$

Tak więc obliczenia wartości generowanych stanów upraszczają się.

3.4. Eliminowanie stanów

Niektóre z wygenerowanych stanów nie prowadzą do rozwiązania optymalnego. Są to stany nieperspektywiczne. W trakcie obliczeń stany nieperspektywiczne są eliminowane. Do eliminacji stanów wykorzystamy regułę sondowania oraz dominacji.

Reguła sondowania pozwala sprawdzić, czy z wygenerowanego stanu P można otrzymać lepszy stan końcowy od stanu aktualnie najlepszego P^a.
Tw. 1.: Stan P jest nieperspektywiczny jeżeli spełnia warunek

$$v_1^{\,a} + v_2^{\,a} \leq v_1 + v_2 - 2 + \left[\frac{1}{c}\left\{T_1 + T_2 - (K_1 + K_2 - 2).c + \sum_{n \in I_k} \vartheta_n\right\}\right]^+, \qquad /36/$$

gdzie: I_k - zbiór numerów przydzielonych operacji /patrz /11a//.

Parametry: v_1, v_2, T_1, T_2, K_1, K_2 są wyznaczone dla stanu P. Jeżeli stan końcowy nie został wygenerowany to w miejsce $v_1^{\,a} + v_2^{\,a}$ podstawiamy $\min(N, 2K)$.

Dowód tego twierdzenia polega na wykazaniu, że prawa strona nierówności /36/ jest dolnym ograniczeniem liczby stanowisk pracy stanu lokalnie optymalnego P', /otrzymanego z P/. Jeżeli w stanie P, po lewej stronie linii, montaż kończy się w chwili T_1 na stacji K_1, to wykorzystano pełnych v_1-1 stanowisk pracy. Czas wykonania operacji na ostatnim stanowisku wynosi $T_1 - (K_1 - 1)$. c. Analogiczna sytuacja występuje po prawej stronie linii. Suma czasów operacji wykonywanych w stanie P na ostatnich stanowiskach pracy po obu stronach linii wynosi $T_1 + T_2 - (K_1 + K_2 - 2).c$. Dodając do tej sumy czasy operacji nie przydzielonych $n \notin I_k$, otrzymamy łączny nakład czasu pracy na stacjach: K_1, $K_1 + 1, \ldots, K$, po lewej oraz K_2, $K_2 + 1, \ldots, K$, po prawej stronie linii. Na tej podstawie /stosując formułę $[\cdot]^+$, otrzymujemy minimalną liczbę wymaganych stanowisk pracy. A zatem suma stanowisk pracy po prawej stronie nierówności /36/ jest dolnym ograniczeniem d.

Ze stanu P można za pomocą heurystycznego algorytmu wyznaczyć pewien stan końcowy $P^{1,N}$; a liczbę stanowisk pracy tak otrzymanego rozwiązania dopuszczalnego przyjąć jako górne ograniczenie g. Jeżeli d = g, to stan końcowy $P^{1,N}$ jest stanem lokalnie optymalnym P'. W przypadku, gdy nierówność /36/ nie jest spełniona i zachodzi d = g, stan $P^{1,N}$ staje się stanem aktualnie najlepszym.

Reguła dominacji stanów pozwala wyeliminować jeden z dwóch stanów $P^{1,\eta}$ /aktywny/ lub P /wygenerowany/.

Tw. 2.: Stan $P^{1,\eta}$ dominuje nad stanem P jeżeli spełniony jest warunek

$$\forall_{n} \ \forall_{1 \leq i \leq 2} \left[\left(p_n^{1,\eta} = 0\right) \Leftrightarrow \left(p_n = 0\right)\right] \wedge \left(T_i^{1,\eta} \leq T_i\right) \wedge \left(T_{\gamma,n}^{1,\eta} \leq T_{\gamma,n}\right) \wedge$$
$$\wedge \left(v_1^{1,\eta} < v_1\right) \wedge \left(v_2^{1,\eta} < v_2\right) \qquad /37/$$

Dowód tego twierdzenia polega na wykazaniu, że trajektoria lokalnie optymalna wychodząca ze stanu P, może być również zrealizowana od stanu $P^{1,\eta}$. Oznacza to, że współrzędne p_n', /lokalnie optymalnego stanu P'/, dla operacji nie przydzielonych w stanie P, /tzn. $p_n = 0$/ - są takie same jak w stanie końcowym $P^{1,N}$. Stan końcowy $P^{1,N}$ otrzymujemy ze stanu $P^{1,\eta}$. Ponieważ trajektorie lokalnie optymalne: P,..., P_o' oraz $P^{1,\eta}$,..., $P^{1^o,N}$ są na ogół różne, zatem stany $P^{1,N}$ i $P^{1^o,N}$ są różne. A zatem

$$v_1^{1^o,N} + v_2^{1^o,N} \leq v_1^{1,N} + v_2^{1,N} \qquad /38/$$

Ponadto ponieważ $v_1^{1,\eta} < v_1$ i $v_2^{1,\eta} < v_2$ a ze stanów P i $P^{1,\eta}$ prowadzimy tę samą trajektorię do odpowiednio P' i $P^{1,N}$, zatem możemy napisać

$$v_1^{1,N} + v_2^{1,N} \leq v_1' + v_2' \qquad /39/$$

Z /38/ i /39/ wynika, że

$$v_1^{1^o,N} + v_2^{1^o,N} \leq v_1' + v_2' \qquad /40/$$

A zatem stan $P^{1,\eta}$ dominuje nad stanem P, lub jest mu równoważny. Jeżeli stany $P^{1,\eta}$ i P są równoważne, to do dalszych obliczeń pozostawiamy stan $P^{1,\eta}$, bo znajduje się już w zbiorze stanów aktywnych.

Możliwość realizacji optymalnej trajektorii ze stanu P również od stanu $P^{1,\eta}$ wynika bezpośrednio z /37/. W obydwu stanach $P^{1,\eta}$ i P pozostały do wykonania te same operacje. W stanie $P^{1,\eta}$ wcześniej niż w P zostały zakończone wszystkie bezpośrednie poprzedniki nie wykonanych operacji. Ponadto w stanie $P^{1,\eta}$ zakończono wcześniej niż w P montaż wszystkich operacji po lewej i po prawej stronie linii. W tych warunkach jakikolwiek przydział /a zatem i lokalnie optymalny/ operacji dotąd nie wykonanych ze stanu P, jest również dopuszczalny od stanu $P^{1,\eta}$.

4. Zakończenie

W przedstawionym algorytmie programowania wieloetapowego istotne znaczenie ma organizacja danych. Zachowanie pewnej redundancji informacji przyspiesza obliczenia, lecz obciąża pamięć komputerową. Wartości: v_1, v_2, T_1, T_2, K_1, K_2, $T_{\gamma,n}$ oraz d mogą być każdorazowo obliczone dla wybranego stanu aktywnego. Zapamiętywanie tych wartości skraca czas obliczeń. Wprowadzenie kodowanego zapisu stanu /ujemne współrzędne dla operacji wykonywanych z lewej strony linii/ zmniejsza o połowę zajętość pamięci komputerowej /w stosunku do macierzy dwukolumnowej - każda kolumna dla jednej strony linii/.

Czas obliczeń zależy od stosowanych reguł: wyboru stanu aktywnego, podziału /przy generowaniu/ oraz eliminacji stanów nieperspektywicznych. Efektywność reguł eliminacji stanów zależy od złożoności reguły /czasu obliczeń/ oraz prawdopodobieństwa wyeliminowania stanu. Na ogół większe prawdopodobieństwo dają reguły bardziej złożone. Czas testowania stanów w regule dominacji jest w dużym stopniu zależny od uporządkowania stanów aktywnych.

Przedstawiony problem alokacji operacji na liniach równoległych uwzględniał tylko relację kolejności. Pokazany algorytm można uogólnić dodając relacje wykluczania i synchronizacji operacji. Ponadto przy pewnych modyfikacjach za pomocą przedstawionego algorytmu można rozwiązać problem alokacji operacji na linii równoległej dla montażu wielowersyjnego.

LITERATURA

[1] Bellman R.: Adaptacyjne procesy sterowania, PWN, Warszawa 1965, ss. 80-92

[2] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Algorytmy sterowania rozdziałem zadań i zasobów w kompleksie operacji, WPP, Poznań 1978, ss. 114-136

[3] Buxey G.M.: Assembly Line Balancing with Multiple Stations, Management Science, V.20, No 6, 1974, pp.1010-1021

[4] Cook S.A.: The Complexity of Theorem Proving Procedures, Proceedings, 3 rd ACM Symposium on Theory of Computing, 1971, pp. 151-158

[5] Garfinkel R.S., Nemhauser G.L.: Programowanie całkowitoliczbowe, PWN, Warszawa 1978

[6] Held M., Karp R.M.: A dynamic programming approach to sequencing problems, J. Soc Indust. Appl. Math., V. 10, No 1, 1962, pp.196-210

[7] Held M., Karp R.M., Shareshian R.: Assembly Line Balancing - Dynamic Programming with Precedence Constraints, Operations Research, V.11, No 3, 1963, pp. 442-459

[8] Held M., Karp R.M.: The construction of Discrete Dynamic Programming Algorithms, IBM Systems Journal, V.4, No 2, 1965, pp. 136-147

[9] Ibaraki T.: Theoretical Comparisons of Search Strategies in Branch - and - Bound Algorithms, International Journal of Computer and Information Sciences, V.5, No 4, 1976, pp. 315-344

[10] Karp R.M.: Reducibility Among Combinatorial Problems, Complexity of Computer Computation, /R.E.Miller i J.W.Thatcher eds./, New York, Plenum Press, 1972, pp. 85-104

[11] Kohler W.H., Steiglitz K.: Charakterization and Theoretical Comparison of Branch - and - Bound Algorithms for Permutation Problems, JACM, V.21, No 1, 1974, pp. 140-156

[12] Kohler W.H., Steiglitz K.: Przeglądowe i iteracyjne metody obliczeniowe, Teoria szeregowania zadań /red. Coffman jr /, WNT, Warszawa 1980, ss. 241-301

[13] Korbut A.A., Finkelsztejn J.J.: Programowanie dyskretne, PWN, Warszawa 1974

[14] Kowalowski H., i inni: Opracowanie algorytmów i programów systemu programowania i sterowania montażami PLISTEM, Praca naukowo-badawcza Instytut Automatyki, Politechnika Śląska, Gliwice 1981 /nie publikowana/

[15] Laptin J.P.: O wierojatnostnom modielirowanii mietoda wietwiej i granic, Kibiernietika, No 3, 1980, ss.111-116

[16] Marecki F.: Modelowanie symulacyjne linii montażowej samochodu małolitrażowego, Informatyka, No 7-8, 1975, ss. 25-28

[17] Marecki F.: Balansowanie linii montażowej z ograniczeniami wykluczania operacji, ZN Pol.Śl., seria: Automatyka, z.63, Gliwice 1982, ss.81-89

[18] Marecki F.: Static Control of Tasks Allocation on Assembly Line, International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest, 1981, V.III, pp.17-23

[19] Marecki F.: Discrete Processes Control, International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest, 1983

[20] Przybysz T.: Balansowanie szeregowej linii montażowej. Praca dyplomowa magisterska /nie publikowana/, Instytut Automatyki, Politechnika Śląska, Gliwice, 1981

[21] Przybysz T., Jurczyk Z., Pawlik S., Marecki F.: System sterowania montażem podwozi ciągnika licencyjnego, ZN Pol.Śl., seria: Automatyka z.64, Gliwice 1982, ss.109-114

[22] Salveson M.E.: The Assembly Line Balancing Problem, Transactions of the American Society of Mechanical Engineers, V.77, No 16, 1955, pp. 939-947

[23] Salveson M.E.: The Assembly Line Balancing Problem, The Joural of Industrial Engineering, V.6, No 3, 1955, pp. 18-25

[24] Szkurba W.W., Bieleckij S.A.: Czislienny je mietody w rieszenii zadaczi bałansirowanija sboroczno j linii, Kibiernietika, No 1, 1977, ss. 96-108

[25] Ullman J.D.: Złożoność obliczeniowa problemów szeregowania, Teoria szeregowania zadań /red: Coffman jr/, WNT, Warszawa 1980, ss.150-177

[26] Węgrzyn S.: Podstawy informatyki, PWN, Warszawa 1982

[27] Wirth N.: Algorytmy + Struktury danych = programy

Recenzent: Prof.dr hab.inż. Stanisław Piasecki

Wpłynęło do Redakcji do 30.03.1984r.

АЛОКАЦИЯ ОПЕРАЦИЙ В ПАРАЛЛЕЛЬНОЙ СБОРОЧНОЙ ЛИНИИ

Р е з ю м е

В работе сформулирована и решена проблема алокации операций в параллельной сборочной линии. Сущность проблемы состоит в определении минимального числа рабочих мест и приписание их к сборочным станциям. Сборочная станция может состоять из двух параллельных рабочих мест. Проблема решена методом многоэтапного программирования.

OPERATIONS ALLOCATION FOR PARALLEL ASSEMBLY LINE

S u m m a r y

Operations allocation problem for parallel assembly line is considered and solved. The problem is to find a minimal number of stands and their localization among assembly stations. Two parallel stands may exist on both sides of assembly line. The allocation problem is solved using multistage programming method.