ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1986

Seria: AUTOMATYKA z. 81 Nr kol. 851

Franciszek MARECKI

Instytut Automatyki
Politechniki Śląskiej w Gliwicach

ALOKACJA OPERACJI I MONTERÓW NA LINII MONTAŻOWEJ

Streszczenie. W pracy przedstawiono uogólniony problem balansowania linii montażowej z uwzględnieniem kwalifikacji monterów. Monter przydzielony do stanowiska pracy musi mieć kwalifikacje dla wykonania wszystkich operacji na tym stanowisku. Z tego względu monterzy i operacje muszą być równocześnie przydzielane do stanowisk pracy. Sformułowany problem rozwiązano metodą programowania wieloetapowego.

1. WPROWADZENIE

Problem alokacji operacji na linii montażowej (zwany balansowaniem linii) został sformułowany przez Salvesona [19], [20]. Polega on na wyznaczeniu minimalnej liczby podzbiorów operacji tworzących stanowiska pracy. Zakłada się przy tym, że dane są: zbiór operacji, relacja kolejności i czasy operacji oraz cykl linii. Cykl jest czasem pobytu montowanego obiektu na stanowisku pracy linii. Przyjmuje się, że suma czasów operacji przydzielonych do stanowiska pracy nie może przekroczyć cyklu, by montaż był kompletny.

Dla rozwiązania problemu balansowania linii montażowej opublikowano wiele algorytmów [2], [21], opartych na programowaniu dyskretnym, sieciowym, dynamicznym oraz na metodzie podziału i ograniczeń. Ponieważ problem balansowania linii montażowej jest NP-zupełny w sensie złożoności obliczeniowej [3], do jego rozwiązania stosowane są metody przeglądowe lub heurystyczne [17], [18].

Na przemysłowych liniach montażowych istotne znaczenie ma nie tylko balansowanie, ale również alokacja monterów na stanowiska pracy. Monterzy na ogół nie są uniwersalni, tzn. nie mają kwalifikacji do wykonywania wszystkich operacji. Aby rozwiązać problem przydziału monterów do stanowisk pracy można najpierw rozwiązać problem balansu, a następnie alokacji monterów do utworzonych podzbiorów operacji. Zakłada się przy tym, że monter może pracować na stanowisku pracy, jeśli ma kwalifikacje do wykonania każdej operacji przydzielonej do tego stanowiska.

W ogólnym przypadku takie dwufazowe postępowanie może nie dać rozwiązania optymalnego, a nawet dopuszczalnego (chociaż takie istnieje). Z tego

względu w niniejszej pracy sformułowany został problem alokacji operacji i monterów na linii montażowej.

W problemie alokacji operacji i monterów na linii montażowej zakłada się, że monterzy mają różne (indywidualne) czasy wykonywania operacji. Jako kryterium optymalizacji przyjmuje się minimalizację liczby stanowisk pracy (monterów).

Problem alokacji operacji i monterów na linii montażowej był rozwiązywany metodą programowania sieciowego [4] oraz metodą podziału i ograniczeń w [15]. Rezultaty obliczeń komputerowych świadczą o dużej zajętości pamięci i długim czasie obliczeń. Dlatego w niniejszej pracy podano algorytm, który pozwala uzyskać rozwiązanie dokładne lub przybliżone (jeżeli przekroczone zostanie ograniczenie pamięci lub czasu obliczeń). Idea tego algorytmu została zaczerpnięta z [13] i [15].

Przedstawiony w pracy algorytm jest oparty na metodzie programowania wieloetapowego [14]. Jest to metoda wykorzystująca pewne elementy wieloetapowych procesów decyzyjnych [2], programowania dynamicznego [6], [7] oraz metody podziału i ograniczeń [5], [8], [9]. Rezultaty obliczeniowe algorytmów programowania wieloetapowego są w dużym stopniu zależne od konstrukcji programów komputerowych. W szczególności istotne znaczenie mają komputerowe struktury danych [2], [23].

2. MODEL MATEMATYCZNY

Załóżmy, że dany jest zbiór operacji montażowych

$$\Omega = \{\omega_n\}, \qquad n = 1,\dots,N, \tag{1}$$

gdzie:

ω_n - n-ta operacja,

N - liczba operacji.

Relację kolejności wykonywania operacji przedstawia macierz:

$$\Gamma = [\gamma_{\vartheta,n}] \qquad \begin{matrix} \vartheta = 1,\dots,N \\ n = 1,\dots,N \end{matrix} \tag{2}$$

Elementy tej macierzy przyjmują binarne wartości:

$$\gamma_{\vartheta,n} = \begin{cases} 1 : \text{jeśli } \omega_\vartheta \text{ jest bezpośrednim poprzednikiem } \omega_n \\ 0 : \text{w przypadku przeciwnym} \end{cases} \tag{2a}$$

Załóżmy, że dany jest zbiór monterów

$$W = \{w_m\} \qquad m = 1,\dots,M \tag{3}$$

gdzie:

w_m - m-ty monter,

M - liczba monterów,

Czasy wykonywania operacji przez monterów dane są macierzą:

$$\Theta = \left[\vartheta_{m,n}\right]. \tag{4}$$

gdzie:

$\vartheta_{m,n}$ - czas wykonania ω_n przez w_m.

W procesie montażu wymuszony jest cykl linii c.
Jeżeli

$$c < \vartheta_{m,n} \tag{5}$$

to w_m nie może wykonać ω_n.

Alokacja operacji i monterów polega na utworzeniu dopuszczalnych podzbiorów operacji wykonywanych przez monterów na kolejnych stanowiskach pracy. Na linii szeregowej na każdym stanowisku pracuje jeden monter. Podzbiór operacji przydzielonych do stanowiska pracy musi spełniać ograniczenia kolejnościowe i czasowe. Ponieważ suma czasów operacji należących do podzbioru zależy od montera, który te operacje wykonuje, zatem obok alokacji operacji trzeba alokować monterów.

Ograniczenia kolejnościowe można zapisać w postaci:

$$\forall_{\vartheta} \forall_{n} \exists_{\varkappa \leq k} (\vartheta_{\vartheta,n} = 1) \wedge (\omega_{\vartheta} \in A_{\varkappa}) \Longrightarrow (\omega_n \in A_k). \tag{6}$$

gdzie:

A_k - podzbiór operacji na k-tym stanowisku pracy.

Ograniczenia czasowe zapiszemy jako:

$$\forall_{m} (w_m \rightarrow A_k) \Longrightarrow (\sum_{\omega_n \in A_k} \vartheta_{m,n} \leq c), \tag{7}$$

gdzie:

$\rightarrow$ - symbol przydzielenia w_m do A_k.

Jako kryterium optymalizacji alokacji operacji i monterów przyjmiemy minimalizację liczby stanowisk pracy (czyli liczby monterów).

$$Q = K \longrightarrow \min, \tag{8}$$

gdzie:

K - liczba stanowisk pracy utworzonych na linii.

Warunkiem koniecznym rozwiązania problemu jest, by:

$$K \leqslant M \tag{9}$$

Nie jest to jednak warunek wystarczający, bowiem ze względu na (5), nawet przy spełnieniu (9), problem może nie być rozwiązalny.

Dla rozwiązania problemu alokacji operacji i monterów przedstawimy dwa odmienne algorytmy oparte na metodzie programowania wieloetapowego.

3. ALGORYTM JEDNOKROKOWY

Problem alokacji operacji i monterów może być rozważany jako N-etapowy proces decyzyjny. Decyzje dotyczą przydziału operacji do kolejnych stanowisk pracy. Monterzy są traktowani jako dodatkowy substytucyjny zasób, podwójnie ograniczony. Liczba monterów M jest ograniczona w procesie montażu obiektu na linii. Ponadto na każdym stanowisku pracy występuje jeden monter.

W procesie alokacji operacji wyróżnimy stany przedstawiające sytuację po przydzieleniu pewnych operacji do stanowisk pracy. Jeżeli wszystkie operacje zostaną przydzielone do stanowisk pracy, to otrzymamy stan końcowy. Stanów końcowych na ogół jest wiele. Stany pośrednie przedstawiają efekt rozdziału nie wszystkich operacji. W obliczeniach będziemy posługiwali się również stanem początkowym, przedstawiającym sytuację przed rozdziałem operacji. Ciąg stanów nazwiemy trajektorią. Każda trajektoria wychodzi z tego samego stanu początkowego. Trajektoria kompletna łączy stan początkowy ze stanem końcowym. Trajektoria częściowa łączy stan początkowy ze stanem pośrednim.

Stan zdefiniujemy tak, by zawierał informację o przydzieleniu operacji i montera do stanowiska pracy. A zatem każdy stan końcowy daje dopuszczalne rozwiązanie problemu. Z każdym stanem zwiążemy jego wartość. Wartość stanu zdefiniujemy tak, by korespondowała ona z przyjętym kryterium optymalizacji. A zatem wartość stanu pozwala wybrać lepszy stan końcowy.

Aby rozwiązać problem alokacji operacji i monterów należy wygenerować wszystkie trajektorie. W algorytmie jednokrokowym z danego stanu generowane są tylko jego pośrednie następniki, czyli stany otrzymane po uzupełnieniu danego stanu o jedną operację. Do generowania stanów wykorzystywane są tzw. stany aktywne, czyli stany, które pozwalają wygenerować dalsze stany. Z tego względu stany końcowe nie są aktywne. Stany aktywne są wybierane zgodnie z określoną regułą, np. FIFO, LIFO, LLB itp. (stosując strategię czystą lub mieszaną). Z wybranego stanu aktywnego generowane są wszystkie jego bezpośrednie następniki. W ten sposób wybrany stan przestaje być stanem aktywnym. Pierwszym stanem aktywnym jest stan początkowy. Obliczenia kończymy, gdy zbiór stanów aktywnych jest pusty.

W trakcie obliczeń niektóre stany mogąbyć wyeliminowane, jeżeli nie pozwalają uzyskać rozwiązania optymalnego. Do eliminacji stanów wykorzystamy reguły: wyczerpywania, dominacji i sondowania. Stan jest wyczerpany, jeżeli nie pozwala uzyskać dopuszczalnego rozwiązania problemu. Dominacji podlega jeden z dwóch stanów aktywnych, o gorszym stanie lokalnie optymalnym. Stan lokalnie optymalny jest najlepszym stanem końcowym uzyskanym z danego stanu. Eliminacja stanu przez sondowanie następuje, gdy jego dolne ograniczenie funkcji kryterialnej jest większe od wartości aktualnie najlepszego stanu końcowego. Dla reguły sondowania potrzebna jest znajomość stanu końcowego.

3.1. Stan i wartość stanu

Załóżmy, że w procesie decyzyjnym wyróżnimy etapy η, (= 0,...,N). Stany ponumerujemy oddzielnie w ramach każdego etapu.

Def. 1.: Stan jest macierzą

$$p^{l,\eta} = \left[p^{l,\eta}_{n,j}\right] \qquad \begin{array}{l} n = 1,\dots,N \\ j = 1,2 \\ l = 1,\dots,L_\eta \end{array} \tag{10}$$

gdzie:

l - numer stanu w ramach η-tego etapu,

L_η - liczba stanów η-tego etapu.

Elementy tej macierzy definiujemy następująco

$$p^{l,\eta}_{n,1} = \begin{cases} m & : \text{jeśli do wykonania } \omega_n \text{ przydzielono } w_m \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{10a}$$

$$p^{l,\eta}_{n,2} = \begin{cases} t_{m,n} & : \text{jeśli } p^{l,\eta}_{n,1} = m \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{10b}$$

Przy tym $t_{m,n}$ jest chwilą zakończenia operacji ω_n na linii montażowej. Przypomnijmy, że montowany obiekt porusza się jednostajnie wzdłuż linii. Załóżmy, że w chwili t=0 montowany obiekt wchodzi na pierwsze stanowisko pracy. A zatem $t_{m,n}$ jest również czasem, jaki upływa od wejścia montowanego obiektu na pierwsze stanowisko pracy do zakończenia operacji ω_n. A zatem stan początkowy $P^{1,0}$ jest macierzą zerową, natomiast każdy stan końcowy $p^{l,N}$ ma wszystkie współrzędne dodatnie.

Z każdym stanem $p^{l,\eta}$ związemy jego wartość, którą oznaczymy przez $V^{l,\eta}$.

Def. 2.: Wartość stanu jest skalarem

$$V^{l,\eta} = \left[\frac{1}{c}\, T^{l,\eta}\right]^{+}, \tag{11}$$

gdzie:

$[\cdot]^{+}$ - najmniejsza liczba całkowita nie mniejsza od wartości w nawiasie

Przy tym:

$$T^{l,\eta} = \max_{1 \leqslant n \leqslant N} p^{l,\eta}_{n,2} \tag{12}$$

Tym samym (12) określa chwilę zakończenia montażu operacji należących do stanu $P^{l,\eta}$, natomiast (11) określa odpowiednią liczbę stanowisk pracy.

Optymalne rozwiązanie problemu daje optymalny stan końcowy, wyznaczony z warunku:

$$\left(\min_{1 \leqslant l \leqslant L_N} V^{l,N} = V^{l^o,N}\right) \Rightarrow \left(P^{l^o,N} = P^o\right), \tag{13}$$

gdzie:

P^o - stan globalnie optymalny.

Ze stanu P^o wyznaczamy stanowiska pracy, do których przydzielono operacje

$$k = \left[\frac{1}{c}\, p^o_{n,2}\right]^{+} \tag{14}$$

Ponadto wyznaczamy monterów przydzielonych do stanowisk pracy:

$$\exists_n \; (p^o_{n,1} = m) \wedge \left(\left[\frac{1}{c}\, p^o_{n,2}\right]^{+} = k\right) \Rightarrow (w_m \rightarrow A_k) \tag{15}$$

W analogiczny sposób wyznaczamy alokację operacji i monterów z każdego stanu końcowego.

3.2. Generowanie stanów

Generowanie stanów polega na wybraniu pewnego stanu aktywnego $P^{\lambda,\eta-1}$ i uzupełnieniu go o dopuszczalną operację ω_n. Operacja ta może być wykonana przez montera w_n z ostatniego stanowiska pracy stanu $P^{\lambda,\eta-1}$, jeżeli

$$\exists_n \; (T^{\lambda,\eta-1} = p^{\lambda,\eta-1}_{k,2}) \wedge (p^{\lambda,\eta-1}_{k,1} = \mu) \wedge (V^{\lambda,\eta-1} \cdot c - T^{\lambda,\eta-1} \geqslant \vartheta_{\mu,n}). \tag{16}$$

Jeśli warunek (16) nie jest spełniony, to do wykonania operacji ω_n przydzielamy montera w_m, który spełnia warunek:

$$\forall_{1 \leq j \leq N} (p_{j,1}^{\lambda,\eta-1} \neq m) \wedge (\vartheta_{m,n} < c) \tag{17}$$

W przypadku, gdy warunek (16) jest spełniony, procedura generowania stanów ma postać:

$$\forall_{n} \forall_{\vartheta} \exists_{\mu} (p_{n,1}^{\lambda,\eta-1} = 0) \wedge \left[(\gamma_{\vartheta,n} = 1) \Rightarrow (p_{\vartheta,1}^{\lambda,\eta-1} > 0)\right] \wedge$$

$$\wedge (V^{\lambda,\eta-1} \cdot c - T^{\lambda,\eta-1} \geq \vartheta_{\mu,n}) \Rightarrow (P = P^{\lambda,\eta-1} + \Delta P) \tag{18}$$

Elementy macierzy ΔP mają postać:

$$\Delta p_{i,1} = \begin{cases} \mu & : \text{dla } i = n \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{19a}$$

$$\Delta p_{i,2} = \begin{cases} t_{\mu,n} & : \text{dla } i = n \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{19b}$$

Chwilę $t_{\mu,n}$, zgodnie z (16), obliczamy jako:

$$t_{\mu,n} = T^{\lambda,\eta-1} + \vartheta_{\mu,n} \tag{20}$$

A zatem wg procedury (18) generowanie stanów polega na przydzielaniu operacji ω_n temu samemu monterowi w_μ, na ostatnim stanowisku pracy w stanie $P^{\lambda,\eta-1}$.

W przypadku, gdy warunek (16) nie jest spełniony, procedura generowania stanów ma postać:

$$\forall_{m} \forall_{1 \leq j \leq N} \forall_{n} \forall_{\vartheta} (p_{j,1}^{\lambda,\eta-1} \neq m) \wedge (p_{n,1}^{\lambda,\eta-1} = 0) \wedge (\vartheta_{m,n} < c) \wedge$$

$$\wedge \left[(\gamma_{\vartheta,n} = 1) \Rightarrow (p_{\vartheta,1}^{\lambda,\eta-1} > 0)\right] \Rightarrow (P = P^{\lambda,\eta-1} + \Delta P) \tag{21}$$

Elementy macierzy ΔP mają postać:

$$\Delta p_{i,1} = \begin{cases} m & : \text{dla } i = n \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{22a}$$

$$\Delta p_{i,2} = \begin{cases} t_{m,n} & : \text{dla } i = n \\ 0 & : \text{w przypadku przeciwnym} \end{cases} \tag{22b}$$

Chwila $t_{m,n}$ jest wyznaczona ze wzoru:

$$t_{m,n} = V^{\lambda,\eta-1} \cdot c + \vartheta_{m,n} \tag{23}$$

A zatem wg procedury (21) generowanie stanów polega na przydzieleniu pierwszej operacji ω_n na kolejnym stanowisku pracy, dla montera w_m, który nie występował w stanie $P^{\lambda,\eta-1}$. Widzimy więc, że procedura (21) jest stosowana przy przejściu pomiędzy dwoma kolejnymi stanowiskami pracy. Z kolei procedura (18) służy do generowania stanów w ramach jednego stanowiska pracy.

Warto zauważyć, że w trakcie generowania stanu obliczamy

$$T = t_{m,n} \tag{24a}$$

oraz

$$V = \left[\frac{1}{c}\ t_{m,n}\right]^{+} \tag{24b}$$

3.3. Eliminowanie stanów nieperspektywicznych

Do eliminacji stanów nieperspektywicznych wykorzystamy reguły: wyczerpywania, dominacji i sondowania, które są oparte na następujących twierdzeniach:

Tw. 1.: Stan P jest nieperspektywiczny, jeżeli spełnia warunek:

$$\exists_{n}\ \forall_{m}\ \forall_{1 \leq i \leq N}\ (p_{n,1} = 0) \wedge (p_{i,1} \neq m) \wedge \left[(c < \vartheta_{m,n}) \vee (V\ c - T < \vartheta_{\mu,n})\right], \tag{25}$$

gdzie w_μ - monter, który wykonuje operacje na ostatnim stanowisku pracy w stanie P.

Dowód jest krótki. Operacja spełniająca warunek (25) nie zostanie wykorzystana do generowania stanu wg procedur (18) lub (21). A zatem ze stanu P nie można dojść do stanu końcowego, dającego dopuszczalne rozwiązanie problemu.

Reguła sondowania jest oparta na twierdzeniu.

Tw. 2.: Stan P jest nieperspektywiczny, jeżeli spełnia warunek:

$$V^{a} \leq V - 1 + \left[\frac{1}{c} \sum_{n \in J} \vartheta'_{n}\right]^{+} \tag{26}$$

Zbiór J definiujemy następująco:

$$\forall_{n} \left[(p_{n,1} = 0) \vee (p_{n,1} = \mu)\right] \Longrightarrow (n \in J) \tag{26a}$$

Ponadto wyznaczamy:

$$\vartheta'_n = \min_{m \in I} \vartheta_{m,n} \tag{26b}$$

Zbiór I definiujemy następująco:

$$\forall_m \ \forall_{1 \leq i \leq N} \ (p_{i,1} \neq m) \vee (\mu = m) \Rightarrow (m \in I), \tag{26c}$$

gdzie w_μ - monter, który wykonuje operacje na ostatnim stanowisku pracy w stanie P.

W dowodzie tego twierdzenia wystarczy pokazać, że prawa strona nierówności (26) jest dolnym ograniczeniem liczby stanowisk pracy dla rozwiązania, jakie można uzyskać ze stanu P. Wynika to z faktu, że dla operacji (nie wykonanych i wykonanych na ostatnim stanowisku pracy) przyjęto minimalne czasy. W regule sondowania przed wyznaczeniem pierwszego stanu końcowego przyjmujemy $V^0 = M$.

Reguły dominacji stanów są oparte na następujących twierdzeniach:

- A).: Dla stanów, w których na ostatnie stanowisko nie można już przydzielić żadnej operacji, tzn. nie jest spełniony warunek (16):

<u>Tw. 3a.</u>: Stan P^{l_1, η_1} dominuje nad stanem P^{l_2, η_2}, jeżeli jest spełniony warunek:

$$(\alpha^{l_1, \eta_1} \supseteq \alpha^{l_2, \eta_2}) \wedge (\beta^{l_1, \eta_1} = \beta^{l_2, \eta_2}) \tag{27}$$

gdzie:

$\alpha^{l,\eta}$ - zbiór przydzielonych operacji w stanie $p^{l,\eta}$,
$\beta^{l,\eta}$ - zbiór przydzielonych monterów w stanie $p^{l,\eta}$.

Zbiór $\alpha^{l,\eta}$ określamy jako:

$$\forall_n \ (p^{l,\eta}_{n,1} > 0) \Rightarrow (\omega_n \in \alpha^{l,\eta}), \tag{28}$$

natomiast zbiór $\beta^{l,\eta}$ jako:

$$\forall_m \ \exists_{1 \leq i \leq N} \ (p^{l,\eta}_{i,1} = m) \Rightarrow (w_m \in \beta^{l,\eta}) \tag{29}$$

<u>Dowód</u>: Z warunku (27) wynika, że

$$V^{l_1, \eta_1} = V^{l_2, \eta_2} \tag{30}$$

Dla trajektorii lokalnie optymalnej $P^{l_2,\eta_2},\ldots,P^{\lambda_2,N}$ możemy napisać:

$$V^{\lambda_2,N} = V^{l_2,\eta_2} + \Delta V^2, \tag{31}$$

gdzie ΔV^2 - minimalna liczba stanowisk pracy dla wykonania operacji ze zbioru $\Omega\setminus\alpha^{l_2,\eta_2}$, przy pomocy monterów ze zbioru $W\setminus\beta^{l_2,\eta_2}$.

Dla trajektorii $P^{l_1,\eta_1},\ldots,P^{\lambda_1,N}$ otrzymamy:

$$V^{\lambda_1,N} = V^{l_1,\eta_1} + \Delta V^1 \tag{32}$$

Jeżeli spełniony jest warunek (27), to

$$\Omega\setminus\alpha^{l_2,\eta_2} \supseteq \Omega\setminus\alpha^{l_1,\eta_1}, \tag{33}$$

a zatem

$$\Delta V^1 \leqslant \Delta V^2 \tag{34}$$

Uwzględniając (34) w (31) i (32) widzimy, że

$$V^{\lambda_1,N} \leqslant V^{\lambda_2,N} \tag{35}$$

czyli stan P^{l_2,η_2} nie daje lepszego rozwiązania niż stan P^{l_1,η_1}.

- B).: Dla stanów, w których na ostatnie stanowisko pracy można przydzielić operacje, tzn. spełniony jest warunek (16):

<u>Tw. 3b.</u>: Stan $P^{l_1,\eta}$ dominuje nad stanem $P^{l_2,\eta}$, jeżeli jest spełniony warunek

$$(\alpha^{l_1,\eta} = \alpha^{l_2,\eta}) \wedge (\beta^{l_1,\eta} = \beta^{l_2,\eta}) \wedge (w_k^{l_1,\eta} = w_k^{l_2,\eta}) \wedge (T^{l_1,\eta} \leqslant T^{l_2,\eta}) \tag{36}$$

gdzie $w_k^{l,\eta}$ - monter wykonujący operacje na ostatnim stanowisku pracy w stanie $p^{l,\eta}$.

<u>Dowód</u>: Dla trajektorii lokalnie optymalnej $P^{l_2,\eta},\ldots,P^{\lambda_2,N}$ napiszemy:

$$V^{\lambda_2,N} = V^{l_2,N} + \Delta V^2 \tag{37}$$

Jeżeli spełniony jest warunek (36), to trajektoria ta może być zrealizowana również od stanu $P^{l_1,\eta}$. Wówczas otrzymamy $P^{l_1,\eta},\ldots,P^{l,N}$, przy tym

$$V^{l,N} = V^{\lambda_2,N} \tag{38}$$

Ponieważ $T^{l_1,\eta} \leqslant T^{l_2,\eta}$, zatem stan lokalnie optymalny $P^{\lambda_1,N}$ spełnia warunek:

$$V^{\lambda_1,N} \leqslant V^{l,N} \tag{39}$$

Z (38) i (39) wynika, że

$$V^{\lambda_1,N} \leqslant V^{\lambda_2,N}, \tag{40}$$

czyli stan $P^{l_2,\eta}$ nie daje lepszego rozwiązania niż stan $P^{l_1,\eta}$.

Jak widać, reguła (27) jest mniej czasochłonna niż reguła (36). Jednakże reguła (36) może być stosowana na każdym etapie, a reguła (27) dla kompletnych stanowisk pracy. Z (36) wnioskujemy, że kolejność wykonywania operacji w ramach jednego stanowiska pracy nie jest istotna. Różne kolejności (strategie) przydzielania operacji prowadzą do identycznych stanów. A zatem do wybranego stanu $P^{\lambda,\eta-1}$ należy przydzielać dopuszczalne podzbiory operacji tworzące stanowisko pracy.

4. ALGORYTM WIELOKROKOWY

Koncepcja algorytmu wielokrokowego wynika z przeprowadzonej dotąd analizy. Istotne znaczenie mają przy tym dwa spostrzeżenia:

- Na stanowisku pracy zbiór operacji jest wykonywany przez jednego montera. A zatem czasy operacji zależą od wybranego montera.
- Kolejność operacji nie ma wpływu na czas ich wykonywania.

Generując stany za pomocą procedury jednokrokowej (w ramach jednego stanowiska pracy) dla każdej operacji dobiera się montera. W algorytmie wielokrokowym można raz wybrać montera i przydzielić mu dopuszczalny podzbiór operacji.

W algorytmie wielokrokowym wyróżniamy stany aktywne $P^{l,k}$, $k = 0,\ldots,K-1$, które opisują dopuszczalną alokację operacji i monterów na k stanowiskach pracy. Algorytm wielokrokowy polega na wyborze pewnego stanu $P^{\lambda,k-1}$ i wygenerowaniu jego następników $P^{l,k}$. A zatem zbiorów stanów aktywnych $P^{l,k}$ jest mniej niż $P^{l,\eta}$, gdyż $K < N$, (chociaż K nie jest z góry dane, lecz jedynie oszacowane).

Dla każdego stanu $P^{\lambda,k-1}$ można określić zbiory:

$$\bar{\alpha}^{\lambda,k-1} = \Omega \setminus \alpha^{\lambda,k-1} \tag{41}$$

oraz

$$\bar{\beta}^{\lambda,k-1} = W \setminus \beta^{\lambda,k-1} \tag{42}$$

wykorzystując określenia (28) i (29).

Generowanie stanu $P^{l,k}$ polega na wyborze montera w_m ze zbioru $\bar{\beta}^{\lambda,k-1}$ i przydzieleniu mu dopuszczalnego podzbioru operacji ze zbioru $\bar{\alpha}^{\lambda,k-1}$. Dopuszczalny podzbiór operacji musi spełniać ograniczenia kolejnościowe i czasowe (suma czasów przydzielonych operacji nie może przekroczyć cyklu).

Dla lokalnych obliczeń (w ramach jednego stanowiska pracy, wychodząc ze stanu $P^{\lambda,k-1}$) wykorzystujemy uproszczoną definicję stanu:

$$S^{l,e} = \left[s_i^{l,e}\right] \qquad i = 0,\dots,N-\eta \qquad e = 1,\dots,N-\eta \qquad l = 1,\dots,L_e \tag{43}$$

gdzie η - liczba operacji przydzielonych w stanie $P^{\lambda,k-1}$.
Elementy wektora (43) określamy następująco:

$$s_i^{l,j} = \begin{cases} m & : \text{ dla } i = 0 \\ t_{m,r} & : \text{ jeśli operacja } \omega_r \text{ została przydzielona monterowi } w_m \\ 0 & : \text{ w pozostałych przypadkach} \end{cases} \tag{43a}$$

Numeracja operacji w zbiorach: Ω, (od 1 do N) i $\bar{\alpha}^{\lambda,k-1}$, (od 1 do N-$\eta$) jest różna, dlatego przyjmujemy wektor incydencji

$$U^{\lambda,k-1} = \left[u_i^{\lambda,k-1}\right] \qquad i = 1,\dots,N-\eta \tag{44}$$

przy tym:
$u_i^{\lambda,k-1} = n$, jeśli operacji ω_n nadano w zbiorze $\bar{\alpha}^{\lambda,k-1}$ numer i.

W procedurze generowania stanów w ramach jednego stanowiska wyróżniamy pierwszy krok i pozostałe kroki. Dla pierwszego kroku otrzymujemy:

$$\forall_m \forall_i \forall_j \; (w_m \in \beta^{\lambda,k-1}) \wedge (\gamma_{i,j} = 0) \wedge (\vartheta_{m,j} \leqslant c) \Longrightarrow (S = S^{1,0} + \Delta S) \tag{45}$$

Elementy wektora ΔS określamy następująco:

$$\Delta s_i = \begin{cases} m & : \text{ dla } i = 0 \\ t_{m,j} & : \text{ dla } i = j \\ 0 & : \text{ w pozostałych przypadkach} \end{cases} \tag{46}$$

Chwilę $t_{m,j}$ wyznaczamy jako:

$$t_{m,i} = V^{\lambda,k-1} \cdot c + \vartheta_{m,i} \tag{47}$$

Dla pozostałych kroków procedura generowania stanów ma postać:

$$\forall_i \forall_j \; (s_j^{\lambda,e-1} = 0) \wedge \left\{ (\delta_{i,j} = 0) \vee \left[(\delta_{i,j} = 1) \Longrightarrow (s_i^{\lambda,e-1} > 0) \right] \right\} \wedge$$

$$\wedge \left[t_{m,j} \leqslant (V^{\lambda,K-1} + 1)\, c \right] \Longrightarrow (S = S^{\lambda,e-1} + \Delta S)$$

Elementy wektora ΔS są określone przez (46), a chwilę $t_{m,j}$ wyznaczamy z formuły:

$$t_{m,j} = T^{\lambda,e-1} + \vartheta_{m,j}, \tag{48}$$

przy tym

$$T^{1,0} = V^{\lambda,k-1} \cdot c \tag{49a}$$

oraz

$$T^{1,e} = T^{\lambda,e-1} + \vartheta_{m,j} = t_{m,j} \tag{49b}$$

Stan $S^{1,e}$ jest końcowy na rozpatrywanym stanowisku pracy, jeżeli nie można z niego wygenerować dalszych stanów. Ze stanu $S^{1,e}$ otrzymujemy odpowiedni stan $P^{1,k}$, bowiem

$$p_{n,1}^{1,k} = \begin{cases} m & : \text{ jeśli } (s_o^{1,e} = m) \wedge (s_i^{1,e} > 0) \wedge (u_i^{\lambda,k-1} = n) \\ p_{n,1}^{\lambda,k-1} & : \text{ w przypadku przeciwnym} \end{cases} \tag{50a}$$

oraz

$$p_{n,2}^{1,k} = \begin{cases} s_i^{1,e} & : \text{ jeśli } p_{n,1}^{1,k} = m \\ p_{n,2}^{\lambda,k-1} & : \text{ w przypadku przeciwnym} \end{cases} \tag{50b}$$

W trakcie generowania stanów $S^{1,e}$ eliminujemy stany identyczne na podstawie twierdzenia:

<u>Tw. 4.</u>: Stan $S^{1,e}$ dominuje nad stanem S, jeżeli jest spełniony warunek:

$$\forall_j \left[(s_j^{1,e} = 0) \Longleftrightarrow (s_j = 0) \right] \wedge (T^{1,e} = T), \tag{51}$$

gdzie S jest stanem wygenerowanym później niż $S^{1,e}$.

Dla eliminacji nieperspektywicznych stanów $P^{1,k}$ wykorzystujemy reguły: wyczerpywania (25), sondowania (26) i dominacji (27).

5. EFEKTYWNOŚĆ ALGORYTMÓW

Przedstawiony problem alokacji operacji i monterów na linii montażowej jest NP-zupełny w sensie złożoności obliczeniowej (bo NP-zupełny jest problem balansowania linii). Do rozwiązania takich problemów stosowane są algorytmy przeglądowe. Efektywność tych algorytmów jest określona przez zajętość pamięci komputerowej lub (częściej) czas obliczeń.

Algorytmy jednokrokowy i wielokrokowy są oparte na metodzie programowania wieloetapowego. Efektywność tej metody jest zależna od reguł wyboru (stanu aktywnego) i podziału oraz reguł eliminacji stanów nieperspektywicznych. Dla reguły FIFO zapamiętywane są stany dwóch sąsiednich etapów obliczeniowych, natomiast dla reguł LIFO lub LLB - [illegible] które stany wszystkich etapów decyzyjnych. Jeżeli stosowane są reguły podziału częściowego (generowanie określonej liczby następników wybranego stanu), to zajętość pamięci komputerowej można dostosować do ograniczeń.

Algorytm wielokrokowy wymaga mniej pamięci komputerowej niż algorytm jednokrokowy. W algorytmie wielokrokowym zapamiętywane są jedynie stany, które interpretują przydział kompletu operacji (i monterów) do pewnej liczby stanowisk pracy. W przypadku algorytmu jednokrokowego zapamiętywane są dodatkowo stany, które interpretują przydział nie wszystkich operacji na aktualnie ostatnie stanowisko pracy.

Efektywność algorytmów w sensie czasu obliczeń jest zależna od liczby wygenerowanych i liczby wyeliminowanych stanów. Reguły eliminacji winny mieć krótki czas testowania i duże prawdopodobieństwo wyeliminowania stanu. Jeżeli testowany stan nie zostanie wyeliminowany, to czas obliczeń rośnie. W przypadku przeciwnym nie jest generowana wiązka trajektorii wychodzących z wyeliminowanego stanu. A zatem czas obliczeń maleje.

Algorytmy jednokrokowy i wielokrokowy różnią się jedną z reguł dominacji stanów. Łatwo zauważyć, że reguła (36) jest bardziej czasochłonna niż reguła (51). Obie reguły służą do eliminacji tych samych stanów. Zatem w algorytmie wielokrokowym reguły dominacji wymagają mniej czasu niż w algorytmie jednokrokowym. Jednakże dla pełnej oceny efektywności tych algorytmów należy uwzględnić konstrukcję i rekonstrukcję stanu S.

Dla skrócenia czasu obliczeń zbiór stanów aktywnych jest odpowiednio uporządkowany. Przede wszystkich wyróżnia się w nim podzbiory stanów dla etapów decyzyjnych. Jeżeli liczba stanów aktywnych w podzbiorze jest duża, to podzbiór należy uporządkować. Ponieważ istotne znaczenie ma czas poszukiwania stanu dominującego (lub który ma być zdominowany), zatem porządek leksykograficzny konstruuje się odpowiednio dla reguły dominacji.

Rozważmy dla przykładu regułę dominacji (51). Powiemy, że stan $S^{1,e}$ poprzedza stan S, jeżeli jest spełniony warunek:

$$\exists_{1<j} \ \forall_{i<j} \ \left[(s_i^{1,e}>0)\wedge(s_i=0)\right] \vee \left\{\left[(s_i^{1,e}=0)\wedge(s_i=0)\right]\vee\left[(s_i^{1,e}>0)\wedge(s_i>0)\right]\right\} \tag{52}$$

Jeżeli stan $S^{l,e}$ nie poprzedza stanu S i odwrotnie, to mówimy, że stany te są alternatywne. Zgodnie z regułą (51) jeden z dwóch stanów alternatywnych zostaje wyeliminowany. A zatem po dominacji w zbiorze stanów aktywnych e-tego etapu nie ma stanów alternatywnych. Skoro zbiór stanów aktywnych jest uporządkowany, to stosując regułę dominacji stanów nie trzeba przeszukiwać całego zbioru. Stosując np. podział połówkowy sprawdzamy, czy stan $S^{l,e}$ dla $l = \left[\frac{1}{2} L_e\right]^+$ jest alternatywny ze stanem S. Jeżeli $S^{l,e}$ poprzedza S, to do dalszych poszukiwań bierzemy drugą część zbioru stanów aktywnych, dla $l + 1,\ldots,L_e$. W przypadku przeciwnym przeszukujemy pierwszą część zbioru stanów aktywnych dla $l = 1,\ldots,\left[\frac{1}{2} L_e\right]^+ - 1$. W stosunku do wybranego podzbioru stanów aktywnych stosujemy dalej podział połówkowy.

W bardziej złożonych regułach dominacji definicja porządku leksykograficznego w zbiorze stanów aktywnych jest również bardziej złożona.

6. ZAKOŃCZENIE

Rozwiązany w pracy problem alokacji operacji i monterów na linii stanowi uogólnienie znanego problemu balansowania. W przypadkach gdy monterzy nie są uniwersalni, problem alokacji operacji i monterów istotnie różni się od problemu balansowania.

Efektywność algorytmów programowania wieloetapowego, wykorzystanych do rozwiązania problemu alokacji operacji i monterów na linii, zależy od liczby wygenerowanych i liczby wyeliminowanych stanów. Ponieważ kolejność operacji na stanowisku pracy nie wpływa na czas montażu, zatem zamiast generowania ciągów operacji (algorytm jednokrokowy) należy generować dopuszczalne podzbiory operacji. Podzbiór dopuszczalny musi spełniać ograniczenia kolejnościowe i czasowe. Z tego względu w algorytmie wielokrokowym generowano ciągi operacji na stanowisku pracy i stosowano dominację stanów identycznych.

Przedstawiony algorytm wielokrokowy nie był oparty wprost na dopuszczalnych zbiorach, lecz na ciągach operacji. W algorytmie wielokrokowym można generować podzbiory operacji (jedno-dwuelementowe itp), a następnie sprawdzać, czy są one dopuszczalne. Podzbiór $\alpha \subset \Omega$ jest dopuszczalny, jeśli spełnia warunek:

$$\forall_{\vartheta} \forall_{n} (\delta_{\vartheta,n} = 1) \wedge (\omega_n \in \alpha) \Longrightarrow (\omega_{\vartheta} \in \alpha) \tag{53}$$

Jeżeli czas obliczeń jest ograniczony, wyznaczany jest stan aktualnie najlepszy P^a oraz błąd względny ε takiego rozwiązania

$$\varepsilon = \frac{V^a - V^o}{V^o} \leqslant \frac{V^a}{d} - 1, \tag{54}$$

gdzie d - najmniejsze dolne ograniczenie wartości stanu końcowego.

W analogiczny sposób można rozwiązać problem alokacji operacji i monterów na linii z kryterium minimalizacji kosztów. Koszty montażu wynikają z ceny operacji lub kosztów pracy monterów.

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] Coffman E. G., jr (red).: Teoria szeregowania zadań, WNT, Warszawa 1980.

[4] Duda J.: PRZEMONT - system optymalnego przydziału zadań dla monterów na linii montażowej, ZN Pol. Śl., seria: Automatyka z. 55, Gliwice, 1980, ss. 129-137.

[5] Gorfinkel R. S., Nemhauser G. L.: Programowanie całkowito-liczbowe. PWN, Warszawa 1978.

[6] Held M., Karp R. M.: A Dynamic Programming Approach to Sequencing Problems, J. Soc. Industr. Appl. Math., V. 10, No. 1, 1962, pp. 196-210.

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

[8] Kohler H. W., Steiglitz K.: Przeglądowe i iteracyjne metody obliczeniowe. Teoria szeregowania zadań (red. Coffman E. G.,). WNT, Warszawa 1980, ss. 241-301.

[9] Korbut A. A., Finkelsztejn J. J.: Programowanie dyskretne. PWN, Warszawa 1974, ss. 164-176.

[10] Kowalowski J. i inni: Balansowanie linii montażu silnika samochodu FIAT-126P w warunkach akordu indywidualnego, Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1977.

[11] Kowalowski H. i inni: System PRZEMONT, Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1978.

[12] Kowalowski H. i inni: Opracowanie algorytmów i programów dla systemu PLISTEM, Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1982.

[13] Marecki F. i inni: Studium sterowania procesem montażu ciągnika licencyjnego MG. Raport z pracy n-b, TNOiK, Bielsko-Biała, 1981.

[14] Marecki F.: Controle of Discrete Processes, t-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[15] Marecki F.: Wielokryterialne balansowanie linii montażowej, 23 Sympozjum "Modelowanie w mechanice", PTMTiS, Gliwice-Szczyrk, 1984, ss. 555-563.

[16] Pawlik S., Marecki F.: Algorytm rozdziału i obsługi zadań na linii montażowej, ZN Pol. Śl., seria Automatyka z. 44, Gliwice 1978, ss. 49-55.

[17] Piasecki S.: Badania operacyjne, VII KKA, Rzeszów 1977, t. 3, ss. 189-192.

[18] Piasecki S.: Operatywne kierowanie pracą linii montażowych. Prace Instytutu Badań Systemowych PAN, nr 64, Warszawa 1981, ss. 3-20.

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

[20] Salveson M. E.: The Assembly Line Balancing Problem, Transactions of the ASME, V. 77, No. 16, 1955, pp. 939-947.

[21] Szkurba W. W., Bieleckij S. A.: Czislienyje mietody w rieszenii zadaczi bałansirowanija sborocznoj linii. Kibiernierika, No. 1, Kijev 1977, ss. 96-108.

[22] Węgrzyn S.: Podstawy Informatyki, PWN, Warszawa 1982.

[23] Wirth N.: Algorytmy + Struktury Danych = Programy, WNT, Warszawa 1980.

Recenzent: prof. dr hab. inż. Stanisław Piasecki

Wpłynęło do Redakcji 9.01.1985 r.

ПРИПИСАНИЕ ОПЕРАЦИЙ И МОНТЁРОВ НА СБОРОЧНОЙ ЛИНИИ

Р е з ю м е

В работе представлена обобщённая проблема балансирования монтажной линии с учётом квалиффикаций монтёров. Монтёр приписан к рабочему месту должен иметь квалиффикации для выполнения всех операций на этом месте. В виду этого монтёры и операции должны быть одновременно приписываны к рабочим местам. Поставленную проблему решено методом многоэтапного программирования.

ALLOCATION OF OPERATIONS AND ASSEMBLERS AT AN ASSEMBLY LINE

S u m m a r y

In the paper generalized problem of assembly line balancing is presented. Operations are performed by assemblers with different qualifications. The assembler must have qualifications for performing each operation at his work station. The problem is solved with the aid of the multi-stage programming method.