ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1980
Seria: AUTOMATYKA z. 55 Nr kol. 651

Franciszek Marecki
Politechnika Śląska

HARMONOGRAMOWANIE DOSTAW DETALI NA LINIĘ MONTAŻOWĄ

Streszczenie. Dla zapewnienia rytmicznej pracy linii montażowej na stanowiska pracy winny być dostarczane sukcesywnie odpowiednie detale. Detale te mogą być dostarczane w kontenerach lub za pomocą specjalnych transporterów /typu taśmowego/. Referat dotyczy harmonogramowania dostaw detali w kontenerach. W referacie przedstawiono algorytm optymalnego harmonogramowania dostaw detali na pola odkładcze linii montażowych. Kryterium optymalizacji jest wykonanie wszystkich zadań /dostaw detali/ - w trakcie zmiany roboczej - przez minimalną liczbę środków transportowych. Proponowane podejście jest oparte na programowaniu dynamicznym i stanowi uogólnienie algorytmu Helda - Karpa - Sharesiana.

1. Wprowadzenie

Rozważymy problem, gdy detale są przekazywane z magazynów wewnętrznych na stanowiska pracy linii montażowej za pomocą kołowych środków transportowych. Dana jest lokalizacja magazynów i drogi transportowe pomiędzy nimi. Środki transportowe są opisane takimi parametrami, jak: średnia prędkość, ładowność, zdolność do przewożenia określonych kontenerów itp. Zadania transportowe określane są dla grupy środków transportowych na jedną zmianę roboczą. Wykonanie tych zadań zapewnia odpowiedni zapas detali na linii na następnej zmianie roboczej.

Celem harmonogramowania dostaw detali na stanowiska pracy linii jest określenie zadań i czasu ich wykonywania dla każdego środka transportowego. Założymy, że kryterium optymalizacji harmonogramu jest minimalizacja liczby wykorzystanych środków transportowych niezbędnych do wykonania zadań zmianowych.

Do rozwiązania tak sformułowanego zadania zostanie wykorzystane podejście programowania dynamicznego, przedstawione w pracach: [2], [3], [4], [5], i odpowiednio zmodyfikowane w [6] i [1].

2. Założenia

Załóżmy, że dany jest wektor zadań:

$$\beta = \left[\beta_n\right] \qquad /1/$$

/n = 1 ..., N/

gdzie: β_n - kod "n-tego" zadania;
$\mathcal{B}$ - wektor zadań.

Kolejność wykonania zadań określa wektor ich priorytetów:

$$B = \left[b_n \right] \qquad /2/$$

gdzie: B - wektor priorytetów;
b_n - priorytet zadania " β_n".

Priorytety zadań są ustalonymi liczbami całkowitymi. Przyjmiemy, że zadania o najwyższym priorytecie muszą być wykonane w pierwszej kolejności.

Niech normatywne czasy wykonywania zadań przedstawia wektor:

$$T = \left[t_n \right] \qquad /3/$$

gdzie: T - wektor normatywnych czasów wykonywania zadań;
t_n - normatywny czas wykonania zadania " β_n".

Rozpatrzmy wektor środków transportowych:

$$\mathcal{A} = \left[\alpha_m \right] \qquad /4/$$

$$/m = 1, \ldots, M/$$

gdzie: $\mathcal{A}$ - wektor środków transportowych;
α_m - kod "m-tego" środka transportowego.

Możliwość wykonywania "n-tego" zadania przez "m-ty" środek transportowy przedstawia macierz:

$$A = \left[a_{n,m} \right] \qquad /5/$$

gdzie: A - macierz przyporządkowania zadań do środków transportowych;

oraz:

$$a_{n,m} = \begin{cases} 1 & \text{- jeśli zadanie " } \beta_n \text{" może być wykonane przez środek transportowy " } \alpha_m \text{";} \\ 0 & \text{- jeśli zadanie " } \beta_n \text{" nie może być wykonane przez środek transportowy " } \alpha_m \text{";} \end{cases}$$

Załóżmy, że zadania /1/ są rozlokowane w przestrzeni. Ponadto położenie środków transportowych, przed rozpoczęciem wykonywania zadań i po ich wykonaniu, jest znane /punkt zerowy/. Załóżmy, że dane są czasy transportu pomiędzy wszystkimi punktami. Czasy te są zapisane w macierzy:

$$D = \left[d_{\mu,\nu} \right] \qquad /6/$$

$$/\ \nu = 0, 1, \ldots, N/$$
$$/\ \mu = 0, 1, \ldots, N/$$

gdzie: $d_{\mu,\nu}$ - czas transportu od punktu "μ-tego" /lokalizacja zadania " β_μ "/ do punktu " ν -tego" /lokalizacja zadania " β_ν "/.

Wprowadźmy ~~z kolei wektor współczynników korygujących czasy wykonywa-~~ nia zadań dla poszczególnych środków transportowych:

$$\Phi = [\varphi_m] \qquad /7/$$

gdzie: Φ - wektor współczynników korygujących czasy wykonywania zadań;

φ_m - współczynnik korygujący dla " α_m ".

A zatem można rozpatrywać wektory czasów wykonywania zadań:

$$T^m = [t_n^m] \qquad /8/$$

gdzie: T^m - wektor czasów wykonywania zadań dla " α_m ";

t_n^m - czas wykonania zadania " β_n " przez " α_m ".

przy czym założymy, że:

$$\exists_{1 \leqslant m \leqslant M} \ \exists_{1 \leqslant n \leqslant N} \quad (a_{n,m} = 1) \Rightarrow t_n^m = \frac{t_n}{\varphi_m} \qquad /9/$$

Elementy wektora /8/, które nie wynikają z warunku /9/, nie mają znaczenia /można dla nich przyjąć dowolne wartości/.

Analogicznie przyjmiemy wektor współczynników korygujących czasy transportu:

$$\Psi = [\Upsilon_m] \qquad /10/$$

gdzie: Ψ - wektor współczynników czasów transportu;

Υ_m - współczynnik korygujący dla " α_m ";

A zatem można rozpatrywać macierze czasów transportu:

$$D^m = [d^m_{\mu,\nu}] \qquad /11/$$

gdzie: D^m - macierz czasów transportu dla " α_m ";

$d^m_{\mu,\nu}$ - czas transportu od punktu " μ -tego" do punktu " ν -tego", dla " α_m "

a więc

$$\exists_{1 \leqslant m \leqslant M} \ \exists_{1 \leqslant \mu \leqslant N} \ \exists_{1 \leqslant \nu \leqslant N} \left[(a_{\mu,m} = 1) \wedge (a_{\nu,m} = 1) \right] \Rightarrow \left(d^m_{\mu,\nu} = \frac{d_{\mu,\nu}}{\Upsilon_m} \right) \qquad /12/$$

Elementy macierzy /11/, które nie wynikają z warunku /12/, nie mają znaczenia /można dla nich przyjąć dowolne wartości/.

Przy powyższych założeniach określimy optymalny harmonogram pracy środków transportowych. Kryterium optymalizacji sformułujemy jako minimalizację liczby środków transportowych, potrzebnych do wykonania zadań /1/. Założymy przy tym, że czas pracy środków transportowych jest ograniczony /zmiana robocza/.

3. Formalizacja problemu

Przedstawimy rozwiązanie rozpatrywanego problemu metodą programowania dynamicznego. Oznaczmy:

$P(l,\eta)$ - "l-ty" stan w " η-tej" warstwie
$/\eta = 0,1,\ldots, N/$
$/l = 1,\ldots, L_\eta/$

$V(l,\eta)$ - wartość stanu "$P(l,\eta)$ ".

gdzie: L_η - liczba stanów w " η-tej" warstwie.

Niech wartością stanu będzie minimalny czas potrzebny na wykonanie wszystkich zadań tego stanu. Załóżmy, że:

$$V_{1,0} = 0 \tag{13}$$

Rozwiązanie zadania uzyskujemy poprzez wyznaczenie stanu w ostatniej warstwie o minimalnym czasie, a więc:

$$V_{opt} = \min_{1 \leq l \leq L_N} V(l, N) \tag{14}$$

Jednakże jest to jedynie ostatni punkt trajektorii optymalnej. Wartości /czasy/ stanów kolejnych warstw można uzyskać z formuły:

$$V(\lambda,\eta+1) = \min_{P(l,\eta)\in U_\eta^\lambda}\left[V(l,\eta) + \Delta V(l,\eta ; \lambda, \eta+1)\right] \tag{15}$$

gdzie: U_η^λ - zbiór stanów " η -tej" warstwy, z których istnieje dopuszczalne przejście do stanu "$P(\lambda, \eta+1)$";

$\Delta V(l,\eta;\lambda,\eta+1)$ przyrost wartości stanu, przy przejściu od stanu "$P(l,\eta)$ " do stanu "$P(\lambda, \eta+1)$".

Przyrost wartości stanu jest czasem odpowiedniej operacji. Warto zwrócić uwagę na fakt, że czas ostatniej operacji musi uwzględniać dodatkowo powrót środka transportowego do punktu zerowego.

Ze stanu "$P(l,\eta)$ " istnieje dopuszczalne przejście do stanu "$P(\lambda, \eta+1)$", jeżeli zbiór zadań stanu "$P(l,\eta)$ " jest zawarty w zbiorze zadań stanu "$P(\lambda, \eta+1)$". Procedura generowania stanów i dopuszczalnych przejść zostanie pokazana w punkcie 6.

Jeżeli znane są wartości wszystkich stanów, to na podstawie /14/, a następnie /15/ można wyznaczyć trajektorię optymalną.

4. Stan procesu rozdziału zadań

Stan procesu rozdziału zadań należy zdefiniować tak, by zminimalizować zajętość pamięci i czas obliczeń maszyny cyfrowej. Dlatego też przedyskutujemy różne definicje stanu.

Definicja 1: Stan procesu rozdziału zadań jest macierzą o postaci:

$$P(1,\eta) = \left[p\,(1,\eta)_{n,m}\right] \qquad /16/$$

$$/n = 0,\ 1, \ldots,\ N/$$
$$/m = 1, \ldots,\ M/$$

gdzie: $p(1,\eta)_{o,m}$ - kod zadania, które "m-ty" środek transportowy wykonał jako ostatnie.

Natomiast dla: $1 \leq n \leq N$ otrzymamy:

$$p(1,\eta)_{n,m} = \begin{cases} 1 - \text{jeśli zadanie "}\beta_n\text{" zostało przydzielone "m-temu" środkowi transportowemu;} \\ 0 - \text{jeśli zadanie "}\beta_n\text{" nie zostało przydzielone "m-temu" środkowi transportowemu.} \end{cases}$$

Generowanie stanów zdefiniowanych wg /16/ jest proste. Jednakże stany te zajmują duży obszar pamięci. Jeżeli w pamięci maszyny cyfrowej przechowywana jest informacja o całej sieci /wszystkich stanach/, to trajektorię optymalną można wyznaczyć korzystając z formuł: /15/ i /14/. Jeżeli natomiast zapamiętanie całej sieci nie jest możliwe, to sieć można generować warstwami - pamiętając tylko stany sąsiednich warstw. W ten sposób, ma. ąc tylko końcowy fragment sieci, można wyznaczyć tylko końcowy fragment trajektorii optymalnej. Początkowy fragment sieci trzeba zatem wygenerować ponownie. Procedura taka wydłuża czas obliczeń.

Definicja 2: Stan procesu rozdziału zadań jest macierzą o postaci:

$$P(1,\eta) = \left[p\,(1,\eta)_{n,m}\right] \qquad /17/$$

$$/n = 1, \ldots,\ N/$$
$$/m = 1, \ldots,\ M/$$

gdzie

$$p(1,\eta)_{n,m} = \begin{cases} \eta - \text{jeżeli zadanie "}\beta_n\text{" zostało przydzielone "m-temu" wykonawcy na "}\eta\text{-tym" etapie decyzyjnym;} \\ 0 - \text{jeżeli zadanie "}\beta_n\text{" nie zostało przydzielone "m-temu" wykonawcy na "}\nu\text{-tym" etapie;} \\ \text{gdzie: } 1 \leq \nu \leq \eta \ . \end{cases}$$

Lokalizację "m-tego" środka transportowego, po wykonaniu ostatniego /przydzielonego mu/ zadania, można wyznaczyć z warunku:

$$\underset{1 \leq m \leq M}{\exists} \left[\max_{n} p(1,\eta)_{n,m} = p(1,\eta)_{n_o,m}\right] \Rightarrow \beta_{n_o} \qquad /18/$$

gdzie: β_{n_o} - ostatnie zadanie przydzielone "m-temu" środkowi transportowemu na "ν-tym" etapie.

Warto zwrócić uwagę na fakt, że w sensie definicji /17/ każdy stan zawiera informację o kolejności wykonywania zadań - oddzielnie przez każdy środek transportowy oraz kolejność rozdziału zadań dla grupy środków transportowych. Tak więc sieć stanów można generować warstwami - po dwie sąsiednie warstwy. Natomiast trajektorię optymalną można odczytać wprost ze stanu ostatniej warstwy, spełniającego warunek /14/. Zatem wykorzystując definicję /17/ można zapamiętywać stany dwóch kolejnych sąsiednich warstw - co daje oszczędność zajętości pamięci maszyny cyfrowej. Ponadto trajektorię optymalną odczytujemy z ostatniego stanu - co skraca czas obliczeń.

Definicja 3: Stan procesu rozdziału zadań jest macierzą o postaci:

$$P(l,\eta) = \left[p(l,\eta)_{n,i} \right] \qquad /19/$$

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

przy czym:

$$p(l,\eta)_{n,1} = \begin{cases} m - \text{jeśli zadanie } "\beta_n" \text{ zostało przydzielone "m-temu" środkowi transportowemu na etapie } \nu\text{-tym; gdzie: } (1 \leqslant \nu \leqslant \eta). \\ 0 - \text{jeśli zadanie } "\beta_n" \text{ nie zostało przydzielone żadnemu środkowi transportowemu na etapie } "\nu\text{-tym", gdzie: } (1 \leqslant \nu \leqslant \eta). \end{cases}$$

$$p(l,\eta)_{n,2} = \begin{cases} \mu/m/ & \text{dla: } p(l,\eta)_{n,1} = m \\ 0 & \text{dla: } p(l,\eta)_{n,1} = 0 \end{cases}$$

gdzie: $\mu/m/$ - numer kolejny zadania przydzielonego "m-temu" środkowi transportowemu.

Stan /19/ spełnia następujące warunki:

$$\forall_{0 \leqslant \eta \leqslant N}\ \forall_{1 \leqslant l \leqslant L_\eta}\ \exists_{1 \leqslant k \leqslant N} \left[p(l,\eta)_{k,1} > 0 \right] \Rightarrow \left[\sum_{k=1}^{k=N} H\left[p(l,\eta)_{k,1} \right] = \eta \right] \qquad /20/$$

↳ w „η" etapach przydzielonych jest η zadań

$$\text{gdzie: } H(x) = \begin{cases} 1 & \text{dla: } x > 0 \\ 0 & \text{dla: } x \leqslant 0 \end{cases}$$

$$\forall_{0 \leqslant \eta \leqslant N}\ \forall_{1 \leqslant l \leqslant L} \max_n \left[p(l,\eta)_{n,2} \right] \leqslant \eta \qquad /21/$$

Tak więc pierwsza kolumna macierzy /19/ zawiera informację o przydziale zadań do środków transportowych, a druga kolumna określa kolejność wykonywania zadań przez każdy środek transportowy. Lokalizację środka transportowego - po wykonaniu ostatniego z przydzielonych mu zadań - wyznaczamy z warunku:

$$\exists_{1 \leqslant n \leqslant N}\ \exists_{1 \leqslant m \leqslant M}\ \exists_{1 \leqslant \eta \leqslant N}\ \exists_{1 \leqslant l \leqslant L_\eta} \left\{ \left[p(l,\eta)_{n,1} = m \right] \wedge \right.$$
$$\left. \wedge \left[\max_n p(l,\eta)_{n,2} = p(l,\eta)_{n_0,2} \right] \right\} \Rightarrow \beta(m)_{n_0} \qquad /22/$$

gdzie: $\beta(m)_{n_0}$ - ostatnia operacja wykonana przez "m-ty" środek transportowy na " ν-tym" etapie; gdzie: $(1 \leqslant \nu \leqslant \eta)$.

Jeżeli liczba środków transportowych jest większa od dwóch, to stany /19/ zajmują mniej pamięci maszyny cyfrowej niż stany /17/. Jednocześnie stany

/19/ posiadają zalety stanów /17/, wyszczególnione wyżej. W dalszym ciągu będziemy rozpatrywać stany /19/.

5. Generowanie stanów

Stany występujące w procesie rozdziału zadań można pogrupować w "N+1" warstwach. W każdej warstwie należy wygenerować tylko stany nieidentyczne. Dwa stany "$P(l_1,\eta)$" oraz $P(l_2,\eta)$" nie są identyczne, jeżeli spełniony jest warunek:

$$\exists_{1\leq\eta\leq N}\ \exists_{\substack{1\leq l_1\leq L_\eta \\ l_1\neq l_2}}\ \exists_{1\leq n\leq N}\ \exists_{1\leq i\leq 2}\ p(l_1,\eta)_{n,i}\neq p(l_2,\eta)_{n,i} \qquad /23/$$

W procesie rozdziału zadań stany są generowane etapami. Zakładając, że dany jest stan "$P(l,\eta)$", można wyznaczyć stan "$P(\lambda,\eta+1)$" wg następującego algorytmu:

$$1^\circ\quad \exists_{1\leq n\leq N}\left[p(l,\eta)_{n,i}>0\right]\Rightarrow\left[p(\lambda,\eta+1)_{n,i}=p(l,\eta)_{n,i}\right] \qquad /24/$$

$$/i=1,2/$$

$$2^\circ\quad \exists_{1\leq n^*\leq N}\ \exists_{\substack{1\leq n\leq N \\ n\neq n^*}}\ \exists_{1\leq m\leq M}\left\{\left[p(l,\eta)_{n^*,1}=0\right]\wedge\left[p(l,\eta)_{n,1}=\right.\right.$$
$$\left.\left.=0\right]\wedge\left[b_{n^*}\gg b_n\right]\wedge\left[a_{n^*,m}=1\right]\right\}\Rightarrow\left\{\left[p(\lambda,\eta+1)_{n^*,1}=\right.\right.$$
$$\left.\left.=m\right]\wedge\left[p(\lambda,\eta+1)_{n,1}=0\right]\right\} \qquad /25/$$

$$3^\circ\quad \exists_{1\leq n^*\leq N}\ \exists_{\substack{1\leq n\leq N \\ n\neq n^*}}\ \exists_{1\leq m\leq M}\left\{\left[p(l,\eta)_{n,1}=m\right]\wedge\left[\max_n p(l,\eta)_{n,2}=\right.\right.$$
$$\left.\left.=p(l,\eta)_{n_0,2}\right]\wedge\left[p(\lambda,\eta+1)_{n^*,1}=m\right]\right\}\Rightarrow\left[p(\lambda,\eta+1)_{n^*,2}=\right.$$
$$\left.=p(l,\eta)_{n_0,2}+1\right] \qquad /26/$$

A zatem na podstawie: /24/, /25/ i /26/ można wyznaczyć współrzędne stanu $P(\lambda,\eta+1)$ na podstawie danego stanu "$P(l,\eta)$".

Wartość stanu "$P(\lambda,\eta+1)$" nie może przekroczyć ograniczenia czasu pracy środka transportowego, a więc:

$$\gamma^*(\lambda,\eta+1)\leq c \qquad /27/$$

gdzie: c - maksymalny czas pracy środka transportowego,

γ^* - wartość stanu, przy założeniu, że po wykonaniu zadań środki transportowe wracają do punktu zerowego.

Wyznaczenie wartości stanów zostanie przedstawione bardziej szczegółowo w następnym punkcie.

Stany spełniające warunek /27/ nazwiemy stanami dopuszczalnymi.

Wygenerowane - na podstawie: /24/, /25/, /26/ i /27/ stany dopuszczalne nie mogą być identyczne. Dlatego należy sprawdzić, czy w " η +1-szej" warstwie nie istnieje stan spełniający warunek /23/ i wyznaczony wcześniej. Sprawdzenia tego można dokonywać sukcesywnie, po wyznaczeniu każdej kolejnej współrzędnej nowego stanu. Każdą kolejną współrzędną należy porównać z odpowiednimi współrzędnymi poprzednio wygenerowanych stanów " η +1-szej" warstwy. Jeżeli dla pewnego stanu współrzędne są różne - to pomijamy go w dalszych sprawdzeniach /ponieważ nie może on być identyczny/. Jeżeli w trakcie kolejnego pomijania stanów usuniemy wszystkie z nich - to dalszego sprawdzania nie trzeba dokonywać, bo generowany stan jest nieidentyczny. Wygenerowany stan winien spełniać warunek /27/, by być dopuszczalny.

6. Wyznaczanie wartości stanów

Wartości stanów w każdej warstwie można wyznaczyć za pomocą ogólnych formuł: /13/, /14/ i /15/. Rozważmy obecnie wyznaczanie "U_η^λ" oraz "$\Delta V(1,\eta;\lambda,\eta+1)$" bardziej szczegółowo. Zbiór stanów "η-tej" warstwy, z których istnieje przejście do stanu "$P(\lambda,\eta+1)$", można wyznaczyć z warunku:

$$\forall_{\substack{1\leq n\leq N\\ n\neq n^*}} \left[p(1,\eta)_{n,i} = p(\lambda,\eta+1)_{n,i}\right] \wedge \left[p(\lambda,\eta+1)_{n^*,i} > 0\right] \wedge \left[p(1,\eta)_{n^*,i} = 0\right] \Rightarrow \left[P(1,\eta) \in U_\eta^\lambda\right] \qquad /28/$$

Jeżeli założymy, że ze stanu "$P(1,\eta)$" istnieje dopuszczalne przejście do stanu "$P(\lambda,\eta+1)$", to wartość "$V(\lambda,\eta+1)$" można wyznaczyć następująco:

1° Z warunku:

$$\exists_{1\leq n^*\leq N} \left\{\left[p(1,\eta)_{n^*,1} = 0\right] \wedge \left[p(\lambda,\eta+1)_{n^*,1} > 0\right]\right\} \Rightarrow \beta_{n^*} \qquad /29/$$

gdzie: β_{n^*} - zadanie wykonane przy przejściu od stanu "$P(1,\eta)$" do stanu "$P(\lambda,\eta+1)$".

$p(\lambda,\eta+1)_{n^*,1}$ - numer środka transportowego, który wykonał zadanie "β_{n^*}".

2° Z warunku:

$$\exists_{1\leq n\leq N} \left\{\left[p(1,\eta)_{n,1} = p(\lambda,\eta+1)_{n^*,1}\right] \wedge \left[\max_n p(1,\eta)_{n,2} = p(1,\eta)_{n_0,2}\right]\right\} \Rightarrow \beta_{n_0} \qquad /30/$$

gdzie: β_{n_0} - zadanie wykonane uprzednio przez środek transportowy o numerze: "$p(\lambda,\eta+1)_{n^*,1}$".

A zatem na podstawie /29/ i /30/ otrzymujemy:

$$\Delta\mathcal{V}^{m}(1,\eta\ ;\ \lambda,\ \eta+1) = d^{m}_{n_o,n^*} + t^{m}_{n^*} \qquad /31/$$

gdzie: $m = p(\lambda,\ \eta+1)_{n,1}$

$\Delta\mathcal{V}^{m}$ - przyrost wartości stanu liczony dla "m-tego" środka transportowego,

czyli:

$$\mathcal{V}^{m}(\lambda,\ \eta+1) = \mathcal{V}^{m}(1,\eta) + \Delta\mathcal{V}^{m}(1,\eta;\lambda,\eta+1) \qquad /32/$$

gdzie: $\mathcal{V}^{m}$ - wartość stanu liczona dla "m-tego" środka transportowego

a więc:

$$\mathcal{V}(\lambda,\ \eta+1) = \max_{m} \mathcal{V}^{m}(\lambda,\ \eta+1) \qquad /33/$$

Z uwagi na ograniczenie czasu pracy środka transportowego /27/ oraz /33/ możemy określić warunek dopuszczalności stanu "P $(\lambda,\ \eta+1)$":

$$\mathcal{V}^{*m}(\lambda,\ \eta+1) \leqslant c \qquad /34/$$

gdzie: $\mathcal{V}^{*m}$ - wartość stanu, liczona dla "m-tego" środka transportowego, przy założeniu, że wykonał on wszystkie zadania,

czyli:

$$\mathcal{V}^{m}(1,\eta) + d^{m}_{n_o,n^*} + t^{m}_{n^*} + d^{m}_{n^*,o} \leqslant c \qquad /35/$$

Jeżeli warunek /35/ nie jest spełniony, to stan P $(\lambda,\ \eta+1)$" pomijamy w dalszych obliczeniach.

W ten sposób, aby wyznaczyć wartość stanu "P $(\lambda,\ \eta+1)$", należy pamiętać wartości stanów "P$(1,\eta)$ " liczone dla poszczególnych środków transportowych. Jeżeli wartość stanu liczona dla środka transportowego, któremu przydzielono zadanie " β_{n^*}", spełnia warunek /35/, to stan "P $(\lambda,\ \eta+1)$" jest dopuszczalny. Wartość tego stanu określa formuła /33/.

Po dojściu do ostatniej warstwy /po rozdzieleniu wszystkich zadań/ należy wyznaczyć stan optymalny. Jeżeli dane są stany "P (λ,N)" oraz wartości tych stanów, liczone dla poszczególnych środków transportowych " $\mathcal{V}^{m}(\lambda,N)$" - to stan optymalny wyznaczamy w sposób następujący:

$$\exists_{1\leqslant n^*\leqslant N} \left[p(1,\ N-1)_{n,1} = 0\right] \wedge \left[p(\lambda,N)_{n^*,1} > 0\right] \Rightarrow \beta(m)_{n^*} \qquad /36/$$

gdzie: $m = p(\lambda,N)_{n^*,1}$

$\beta(m)_{n^*}$ - ostatnie zadanie wykonane przez "m-ty" środek transportowy.

Tak więc

$$\mathcal{V}^{*m}(\lambda,N) = \mathcal{V}^{m}(\lambda,N) + d^{m}_{n^*,o} \qquad /37/$$

oraz:

$$\cancel{\mathcal{V}^{*}(\lambda,N) = \max_{m} \mathcal{V}^{*m}(\lambda,N)} \qquad /38/$$

A zatem stan optymalny w ostatniej warstwie otrzymamy jako:

$$\gamma^*(\lambda_{opt}, N) = \min_{\lambda} \gamma^*(\lambda, N) \qquad /39/$$

Ponieważ wszystkie stany wygenerowane i pozostawione w ostatniej warstwie są dopuszczalne, zatem:

$$\gamma^*(\lambda_{opt}, N) \leqslant c \qquad /40/$$

jeżeli istnieje rozwiązanie zadania.

7. Podsumowanie

W pracy przedstawiono algorytm harmonogramowania dostaw detali na stanowiska pracy linii montażowej. Idea tego algorytmu - oparta na programowaniu dynamicznym - polega na generowaniu stanów sieci procesu decyzyjnego. Przyjęta definicja stanu /19/ pozwala zminimalizować zajętość pamięci i czas obliczeń maszyny cyfrowej. Stany " η +1-szej"warstwy sieci generowane są na podstawie stanów warstwy " η-tej". Ponadto stan określa jednoznacznie kolejność wykonywania zadań przez każdy środek transportowy.

Przedstawiony algorytm polega na:
- wygenerowaniu - na podstawie formuł /24/, /25/ i /26/ - stanu następnej warstwy;
- sprawdzeniu - na podstawie /23/ - czy wygenerowano wcześniej identyczny stan, w tej samej warstwie;
- wyznaczeniu - na podstawie /31/ i /32/ - wartości stanu, liczonego dla środka transportowego, któremu przydzielono ostatnie zadanie;
- sprawdzeniu - na podstawie /35/, czy wygenerowany stan jest dopuszczalny.

W identyczny sposób generujemy wszystkie stany. Stany identyczne lub niedopuszczalne eliminujemy z dalszych obliczeń. Obliczenia kończymy po przydzieleniu wszystkich zadań środkom transportowym.

Jeżeli nie można wygenerować dopuszczalnych stanów w pewnej warstwie, to zadanie nie ma rozwiązania. Jeżeli natomiast uzyskano rozwiązanie dla maksymalnej liczby środków transportowych, to należy powtórzyć obliczenia przy zmniejszeniu ich liczby. Należy przy tym rozpatrzyć wszystkie kombinacje pominiętych środków transportowych, gdyż środki te - z uwagi na /5/, /8/ i /11/ - nie są identyczne.

LITERATURA

[1] Dusza K., Kowalowski H., Marecki F.: Sterowanie dyspozytorskie obsługą robót spawalniczych na KWK, Materiały Sympozjum nt.:"Systemy zarządzania i sterowania kopalniami", Sekcja Cybernetyki w Górnictwie PAN, Szklarska Poręba, 1979, ss. 164-175.

[2] Held M., Karp R.M.: A dynamic programming approach to sequencing problem, SIAM, Journal Appl. Math. Vol. 10, No. 1, 1962, pp.196-210.

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

[4] Held M., Karp R.M.: Primienienije dinamiczeskogo programirowanija k zadaczam uporiadoczienija, Kiberneticzeskij zbornik, wyp. 9, Izd. "Mir"; Moskwa, 1964, ss. 202-218.

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

[6] Marecki F.: Harmonogramowanie procesu wykrawania blach karoseryjnych - metodą programowania dynamicznego, seminarium nt.: Harmonogramowanie dyskretnych procesów przemysłowych", I.A. Pol. Śl., Gliwice, 1979.

СОСТАВЛЕНИЕ ГРАФИКОВ ПОСТАВКИ ДЕТАЛЕЙ НА СБОРОЧНУЮ ЛИНИЮ

Р е з ю м е

В рботе представлено проблемму составления графиков поставки деталей на сборочную линию. Принято, что матерялы поставляются в контенерах. Решение проблеммы показано с помощю обобщенного алгоритма Нелда-Карпа-Шерезяна.

SCHEDULING OF MATERIALS'DELIVERY TO THE ASSEMBLY LINE

S u m m a r y

In the paper a problem of scheduling of materials'delivery to the assembly line is presented. It is assumed that delivery of materials is performed witch the use of containers. The solution of the problem with aid of Held - Karp - Sharesian's generalised algorithm is given.