ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1990
Seria: AUTOMATYKA z.101 Nr kol.1083

Franciszek Marecki

Politechnika Śląska

BALANSOWANIE ZROBOTYZOWANEJ LINII MONTAŻOWEJ

Streszczenie. W referacie przedstawiono dwa algorytmy rozwiązania problemu balansowania zrobotyzowanej linii montażowej. Algorytmy te są oparte na metodzie programowania wieloetapowego.

1. Wstęp

Współczesne linie montażowe są zrobotyzowane. Zapewnia to odpowiednią jakość oraz elastyczność montażu [1].

W referacie rozważymy linię montażową, na której operacje są wykonywane przez roboty przemysłowe. Wyróżnimy stacje wzdłuż linii. Na stacjach tych zainstalowane są roboty. Dana jest lokalizacja przestrzenna montowanego (na stacji) obiektu oraz odpowiedniego robota przemysłowego. W lokalizacji tej istotne znaczenie ma położenie bazowe chwytaka robota oraz punkty, w których mają być wykonane operacje w montowanym obiekcie. Można założyć, że dane są czasy transportu chwytaka pomiędzy wyróżnionymi punktami [4]. Ponadto dane są czasy wykonania każdej operacji technologicznej.

Załóżmy, że na linii montowane są obiekty tej samej wersji. W każdym z takich obiektów należy wykonać te same operacje. Obiekty takie są zbiorami tych samych części (detali).

Na zrobotyzowanej linii montażowej przyjmuje się, że jakość wykonania operacji jest gwarantowana odpowiednim programem pracy robota. Ponadto odpowiedni program pracy linii (tzn. wszystkich robotów) gwarantuje montaż kompletny (tzn. wykonane zostaną wszystkie operacje) [3]. Przy tych założeniach powstaje problem maksymalizacji wydajności linii, tzn. liczby obiektów zmontowanych na linii w trakcie zmiany roboczej. W tym przypadku zakłada się, że na linii znajduje się określona liczba robotów.

W praktyce produkcyjnej mogą wystąpić dwie sytuacje [2]:

A) dana jest liczba robotów, a należy wyznaczyć maksymalną wydajność linii,
B) dana jest wydajność linii, a należy wyznaczyć minimalną liczbę robotów dla realizacji montażu.

W obydwu przypadkach trzeba rozwiązać problem sterowania dla określonego

Praca wchodzi w zakres Programu RP.I.02. w temacie 4.1

przedziału czasu, np. zmiany roboczej. Liczba robotów dana na początku zmiany roboczej może być różna dla kolejnych zmian roboczych. Wynika to z awarii i konserwacji robotów. Równocześnie należy podkreślić, że wyłączanie całej linii przy awarii jednego robota nie jest dopuszczalne. Przyjmując, że awarie robotów są zakłóceniami, należy sterować montażem poprzez wyznaczanie programów pracy dostępnych (sprawnych) robotów. W analizowanym wariancie przyjmuje się, że nie ma ograniczeń dostaw detali. Wydajność linii jest liczbą obiektów zmontowanych w danym przedziale czasu (zmianie roboczej). Jeśli dana jest liczba obiektów, które mają być zmontowane w trakcie zmiany roboczej, to można wyznaczyć tzw. takt, czyli czas, po którym z linii schodzi kolejny obiekt. Wariant sterowania linii montażowej z zadaną wydajnością wynika z ograniczeń dostaw detali. Każdy montowany obiekt wymaga określonej liczby detali określonego typu. Dla każdej zmiany roboczej można wyznaczyć liczbę obiektów, które można zmontować z uwagi na liczby dostępnych detali. Jeżeli wydajność zmianowa linii przy wszystkich włączonych robotach jest większa od wydajności zadanej (wynikającej z dostaw detali), to mogą wystąpić dwie sytuacje:

- wszystkie roboty będą włączone przez część zmiany roboczej, a następnie linia będzie zatrzymana do końca zmiany roboczej,
- tylko niektóre roboty będą włączone przez całą zmianę roboczą.

Pierwszy przypadek wymaga określenia czasu pracy każdego robota i odpowiedniego programu jego pracy. W drugim przypadku należy określić liczbę robotów i ich programy pracy- dla zadanego taktu.

Z teoretycznego punktu widzenia łatwiej jest rozwiązać problem B. Ponadto rozwiązując wielokrotnie problem B dla różnych taktów otrzymać można rozwiązanie problemu A. Z powyższych względów problem balansowania linii montażowej (BLM) zostanie rozwiązany dla przypadku B.

2. Sformułowanie problemu

Rozważmy linię, na której znajduje się M robotów.

Załóżmy, że dany jest czas taktu linii, który oznaczymy przez c.

Dany jest zbiór operacji montażowych

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

gdzie: ω_n - n-ta operacja.

Dane są czasy wykonania operacji technologicznych, zapisane w wektorze

$$\Theta = [\vartheta_n] \quad n=1, \dots, N \tag{2}$$

gdzie: ϑ_n - czas wykonania operacji ω_n

Przyjmiemy, że operacje nie mogą być przerywane (są niepodzielne).

Załóżmy, że każda operacja jest zlokalizowana w innym punkc montowanego obiektu.

Dane są czasy transportu chwytaka robota pomiędzy punktami lokalizacji operacji

$$\tau = [\ \tau_{k,n}\ ]_{\substack{k=0,1,\dots,N \\ n=0,1,\dots,N}} \tag{3}$$

gdzie: $\tau_{k,n}$ – czas transportu chwytaka od punktu wykonania operacji ω_k do punktu wykonania operacji ω_n,

$\tau_{0,n}$ – czas transportu chwytaka z położenia bazowego do punktu wykonania operacji ω_n,

$\tau_{k,0}$ – czas transportu chwytaka z punktu wykonania operacji ω_k do położenia bazowego.

Robot wycofuje chwytak do swojego położenia bazowego po wykonaniu wszystkich swoich operacji, po czym następuje przesunięcie montowanych obiektów o jedną stację w przód.

Załóżmy, że relacja kolejności wykonywania operacji dana jest macierzą

$$\Gamma = [\ \gamma_{\nu,n}\ ]_{\substack{\nu=1,\dots,N \\ n=1,\dots,N}} \tag{4}$$

gdzie:

$$\gamma_{\nu,n} = \begin{cases} 1, \text{ jeśli } \omega_\nu \text{ musi być wykonana przed } \omega_n \text{ i może} \\ \quad\quad \text{być wykonana bezpośrednio przed } \omega_n \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{4a}$$

Każdy robot pracuje cyklicznie, tzn. w każdym takcie wykonuje tę samą sekwencję operacji. Robot startuje ze swojego położenia bazowego, przechodzi przez punkty, w których są zlokalizowane przydzielone mu operacje i powraca do swojego położenia bazowego.

Z punktu widzenia lokalnego kryterium dla każdego robota, należy wyznaczyć taką sekwencję, by suma czasów operacji technologicznych wykonanych w takcie była maksymalna.

Z punktu widzenia globalnego kryterium dla linii, operacje ze zbioru (1) należy tak rozdzielić pomiędzy roboty, by liczba potrzebnych robotów przy zadanym takcie była minimalna.

Dalej rozwiążemy obydwa problemy.

3. BLM - kryterium lokalne

Załóżmy, że dany jest zbiór operacji Ω, oraz parametry opisujące operacje zapisane w: (2),(3) i (4). Problem polega na wyznaczeniu podzbioru Ω_1 operacji, które mają być wykonane przez pierwszego robota.

$$\Omega_1 \subset \Omega \tag{5}$$

Dopuszczalny podzbiór Ω_1 musi spełniać następujące ograniczenia:

$$\forall_{n} \forall_{\nu} [(\gamma_{\nu,n} = 1) \wedge (\omega_n \in \Omega)] \Rightarrow (\omega_\nu \in \Omega_1) \tag{6}$$

$$\forall_{\nu} \forall_{n} [(\omega_\nu \in \Omega_1) \wedge (\omega_n \in \Omega_1) \wedge (\gamma_{\nu,n} = 1)] \Rightarrow (t_\nu < t_n) \tag{7}$$

$$\forall_{n} (\omega_n \in \Omega_1) \Rightarrow (t_n \le c) \tag{8}$$

gdzie: t_n - chwila zakończenia wykonywania operacji ω_n.

$$\forall_{n} (\omega_n \in \Omega_1) \Rightarrow (t_n - \vartheta_n - \tau_{n,0} \ge 0) \tag{9}$$

$$\exists_{n} \forall_{i \ne n} [(\omega_n \in \Omega_1) \wedge (\omega_i \in \Omega_1) \wedge (t_n < t_i)] \Rightarrow (t_n = \tau_{0,n} + \vartheta_n) \tag{10}$$

$$\exists_{n} \forall_{i \ne n} [(\omega_n \in \Omega_1) \wedge (\omega_i \in \Omega_1) \wedge (t_i < t_n)] \Rightarrow (t_n + \tau_{n,0} \le c) \tag{11}$$

$$\exists_{\nu} \exists_{n} \forall_{i \ne \nu} \{(\omega_\nu \in \Omega_1) \wedge (\omega_n \in \Omega_1) \wedge (\omega_i \in \Omega_1) \wedge (t_\nu < t_n) \wedge$$
$$\wedge [(t_i < t_\nu) \vee (t_n < t_i)]\} \Rightarrow (t_n = t_\nu + \tau_{\nu,n} + \vartheta_n) \tag{12}$$

Wśród rozwiązań dopuszczalnych należy wyznaczyć optymalne w sensie kryterium lokalnego

$$Q^l = \sum_{\omega_n \in \Omega_1} \vartheta_n \longrightarrow \max \tag{13}$$

Tak sformułowany problem rozwiążemy za pomocą algorytmu programowania wieloetapowego. W algorytmie tym zdefiniujemy: stan, wartość stanu, procedurę generowania stanów oraz reguły eliminacji stanów nieperspektywicznych. Aby wyznaczyć rozwiązanie dopuszczalne, należy wygenerować trajektorię stanów:

$$P^{0,1}, \ldots, P^{e-1,\lambda}, P^{e,l}, \ldots, P^{E,J}$$

Pierwszy indeks stanu oznacza numer etapu decyzyjnego, a drugi numer stanu w ramach etapu.

Na każdym etapie podejmowana jest decyzja o wyborze jednej operacji (dla Ω_1), zatem

$$E < N \tag{14}$$

Stan powinien zawierać informację o przedziale czasu, w którym ma być wykonana przydzielona operacja. Ponadto każdy stan powinien zawierać ocenę z punktu widzenia przyjętego kryterium optymalizacji. Zatem wartość stanu jest sumą czasów przydzielonych operacji.

Def.1. Stan jest wektorem

$$P^{e,l} = [p_n^{e,l}] \quad n = 1, \ldots, N \tag{15}$$

Elementy tego wektora są określone następująco:

$$p_i^{e,l} = \begin{cases} t_n \text{ , jeśli na etapie } \eta \leq e \text{ podjęto decyzję o} \\ \qquad \text{o przydzieleniu } \omega_n \text{ do } \Omega_1, \\ 0 \text{ , w przypadku przeciwnym} \end{cases} \tag{15a}$$

Zatem stan początkowy $P^{0,1}$ jest wektorem zerowym.

Stan $P^{e,l}$ jest końcowy, jeśli spełnia warunek:

$$\underset{n}{\forall} \; \underset{\nu}{\forall} (\omega_n \notin \Omega_1) \wedge [(\gamma_{\nu,n} = 1) \rightarrow (\omega_\nu \in \Omega_1)] \wedge$$
$$\wedge (T^{e,l} + \tau_{K^{e,l},n} + \theta_n + \tau_{n,0} > c) \tag{16}$$

przy czym

$$T^{e,l} = \max_{1 \leq i \leq N} p_i^{e,l} \tag{17}$$

oraz

$$(K^{e,l} = j) \leftrightarrow (p_j^{e,l} = T^{e,l}) \tag{18}$$

Z warunku (16) wynika, że w stanie końcowym nie istnieje możliwość dodania do Ω_1 kolejnej operacji.

Jeżeli w stanie $P^{0,1}$ spełniony jest warunek

$$\underset{1 \leq n \leq N}{\forall} \tau_{0,n} + \theta_n + \tau_{n,0} > c \tag{19}$$

to problem nie ma rozwiązania, dla zadanego c.

Na podstawie danego stanu $P^{e,l}$ można wyznaczyć zbiór już przydzielonych operacji $\Omega_1^{e,l}$.

$$\Omega^{e,l} = \{ \omega_i : p_i^{e,l} > 0 \} \tag{20}$$

Def. 2. Wartość stanu jest skalarem

$$V^{e,l} = \sum_{\omega_i \in \Omega^{e,l}} \theta \tag{21}$$

Procedura generowania stanów jest zdaniem logicznym, które mówi, przy jakich warunkach można wygenerować kolejny stan.

Załóżmy, że dany jest stan $P^{e-1,\lambda}$, który nie jest stanem końcowym. Generowanie stanów rozpoczynamy od stanu $P^{0,1}$, który jest znany.

Procedura generowania stanów ma postać

$$\underset{n}{\forall} \; \underset{\nu}{\forall} \{ (p_n^{e-1,\lambda} = 0) \wedge [(\gamma_{\nu,n} = 1) \rightarrow (p_\nu^{e-1,\lambda} > 0)] \wedge$$
$$\wedge (T^{e-1,\lambda} + \tau_{K^{e-1,\lambda},n} + \theta_n + \tau_{n,0} \leq c) \} \rightarrow (P^{e-1} = P^{e-1,\lambda} + \Delta P) \tag{22}$$

gdzie:

$$\Delta p_i = \begin{cases} t_n \text{ , jeśli } i=n \\ 0 \text{ , w przypadku przeciwnym} \end{cases} \tag{23}$$

Chwilę t_n wyznaczamy ze wzoru

$$t_n = T^{e-1,\lambda} + \tau_{K^{e-1,\lambda},n} + \theta_n \tag{24}$$

W (24) wykorzystujemy odpowiednio określenia (17) i (18).
Następnie sprawdzamy, za pomocą warunku (16), czy wygenerowany stan $P^{e,l}$ jest końcowy. Jeśli stan $P^{e,l}$ nie jest końcowy, to jest on zapamiętywany razem z parametrami:

$$V^{e,l}=V^{e-1,\lambda}+\vartheta_n \tag{25}$$

oraz

$$T^{e,l}=t_n \tag{26a}$$

$$K^{e,l}=n \tag{26b}$$

Wzory (25) i (26) przyśpieszają obliczenia.

Do eliminacji stanów nieperspektywicznych można wykorzystać regułę dominacji.

<u>Twierdzenie 1.:</u> Stan $P^{e,l1}$ dominuje nad stanem $P^{e,l2}$ jeśli spełniony jest warunek

$$(\Omega^{e,l1} = \Omega^{e,l2})\wedge(K^{e,l1}=K^{e,l2})\wedge(T^{e,l1}<T^{e,l2}) \tag{27}$$

<u>Dowód:</u>

Załóżmy, że wychodząc, ze stanu $P^{e,l2}$ wygenerowano trajektorię lokalnie optymalną (najlepszą z $P^{e,l2}$). Najlepszy lokalnie stan końcowy $P^{E,l2}$ ma wartość $V^{E,l2}$.

Zauważmy, że jeśli spełniony jest warunek (27), to ze stanu $P^{e,l1}$ można wygenerować trajektorię (niekoniecznie lokalnie optymalną),

$$P^{0,1},\dots,P^{e,l1},\dots,P^{E,\lambda}$$

która spełnia warunki

$$\forall_i \; (p_i^{e,l1} > 0) \rightarrow (p_i^{E,\lambda} = p_i^{e,l1}) \tag{28a}$$

oraz

$$\forall_i \; (p_i^{e,l1} = 0) \rightarrow (p_i^{E,\lambda} = p_i^{E,l2}) \tag{28b}$$

Dla trajektorii

$$P^{0,1},\dots,P^{e,l2},\dots,P^{E,l2}$$

można zapisać

$$V^{E,l2} = V^{e,l2} + \sum_{\omega_n \in \Omega'} \vartheta_n \tag{29}$$

przy czym

$$\Omega' = \Omega^{E,l2} \ominus \Omega^{e,l2} \tag{30}$$

Można zauważyć, że z uwagi na (27) i (28)

$$\Omega^{E,l2} = \Omega^{E,\lambda} \tag{31}$$

czyli

$$\Omega^{E,\lambda} - \Omega^{e,l1} = \Omega' \tag{32}$$

Stąd wnioskujemy, że

$$V^{E,\lambda} = V^{e,l1} + \sum_{\omega_n \in \Omega'} \vartheta_n \tag{33}$$

Ponieważ

$$V^{e,l1} = \sum_{\omega_i \in \Omega^{e,l1}} \theta_i = \sum_{\omega_i \in \Omega^{e,l2}} \theta_i = V^{e,l2} \qquad (34)$$

stąd ostatecznie otrzymujemy

$$V^{E,\lambda} = V^{E,l2} \qquad (35)$$

Wniosek wynikający z (35) jest następujący:
Ze stanu $P^{e,l2}$ nie można otrzymać lepszego rozwiązania niż ze stanu $P^{e,l1}$. Zatem stan $P^{e,l2}$ można w dalszych równaniach pominąć.

Stan można również wyeliminować za pomocą reguły sondowania.

Twierdzenie 2.: Stan $P^{e,l}$ jest nieperspektywiczny, jeśli spełnia warunek

$$V^{e,l} + \Delta V < V^{a} \qquad (36)$$

gdzie: V^{a} -wartość stanu aktualnie najlepszego P^{a}.

Dla ΔV można sformułować różne oszacowania, np.

$$\Delta V = 0 \qquad (37)$$

jeśli wygenerowany stan $P^{e,l}$ jest końcowy.

Przedstawiony algorytm dla pierwszego robota można powtórzyć dla pozostałych robotów. Dla każdego kolejnego robota należy określić zbiór operacji

$$\Omega^{m}, \quad m=1,\dots,M$$

przy czym

$$\left.\begin{aligned} \Omega^{1} &= \Omega \\ \Omega^{2} &= \Omega - \Omega_{1}^{opt} \\ \Omega^{m} &= \Omega - \Omega_{m-1}^{opt} \\ \Omega^{M} &= \Omega - \Omega_{M-1}^{opt} \end{aligned}\right\} \qquad (38a)$$

lecz

$$\Omega^{M} = \Omega_{M}^{opt} \qquad (38b)$$

gdzie: Ω_{m}^{opt} – optymalny zbiór operacji przydzielonych m-temu robotowi.

Stosując lokalne kryterium BLM otrzymujemy w rezultacie balans linii dla M robotów. Nie ma jednak gwarancji, że otrzymana liczba M robotów jest najmniejszą z możliwych dla zadanego czasu taktu c. Ponadto, jeżeli otrzymana za pomocą powyższego algorytmu liczba M robotów jest większa niż liczba robotów na linii - to problemu BLM nie można rozwiązać przy stosowaniu kryterium lokalnego.

4. BLM-kryterium globalne

Załóżmy, że dany jest zbiór operacji (1) oraz parametry: (2), (3) i (4). Zbiór Ω należy rozbić na uporządkowany ciąg podzbiorów

$$\langle \Omega_1, \ldots, \Omega_m, \ldots, \Omega_M \rangle \tag{39}$$

Podzbiory ciągu (39) muszą spełniać następujące warunki:

- wszystkie operacje muszą być przydzielone

$$\bigcup_{m=1}^{m=M} \Omega_m = \Omega \tag{40}$$

- każda operacja może być przydzielona tylko dla jednego robota

$$\Omega_\mu \cap \Omega_m = 0 \qquad \mu \neq m \tag{41}$$

- relacja kolejności

$$\forall_n \forall_\nu \quad [(\omega_n \in \Omega_m) \wedge (\gamma_{\nu,n} = 1)] \Rightarrow (\exists_{\mu \leq m} \omega_\nu \in \Omega_\mu) \tag{42}$$

- czas taktu

$$\forall_{1 \leq m \leq M} \quad \exists_{\omega_{n_i} \in \Omega_m} \quad (\tau_{n_{i-1}, n_i} + \theta_{n_i}) + \tau_{n_I, 0} \leq c \tag{43}$$

gdzie: n_i -numer kolejnej operacji w sekwencji m-tego robota.

Analogicznie jak w: (6), (7), (8), (9), (10), (11) i (12) ograniczenia można przedstawić za pomocą chwil t_n. Jednakże obecnie t_n jest chwilą zakończenia ω_n na dowolnej stacji linii, od terminu wejścia obiektu do pierwszej stacji.

Globalne kryterium optymalizacji można zapisać w postaci

$$Q^g = [\frac{1}{c} \max_{1 \leq n \leq N} (t_n + \tau_{n,0})]^+ \longrightarrow \min \tag{44}$$

gdzie: $[\ ]^+$ -najmniejsza liczba całkowita nie mniejsza niż wyrażenie w nawiasie [].

Do rozwiązania tak sformułowanego problemu można wykorzystać algorytm programowania wieloetapowego.

W algorytmie tym przyjmiemy definicję stanu (15). Wartość stanu wyznaczymy z formuły

$$V^{e,l} = \sum_{k=1}^{k=K^{e,l}} q_k^{e,l} \tag{45}$$

przy czym

$$q_k^{e,l} \begin{cases} 1, \text{ jeśli } \exists_n [\frac{1}{c} p_n^{e,l}]^+ = k \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{46}$$

Procedura generowania stanów ma postać

$$\forall_{n} \forall_{\nu} \{(p_n^{e-1,\lambda} = 0) \wedge [(\gamma_{\nu,n} = 1) \rightarrow (p_\nu^{e-1,\lambda} > 0)]\} \rightarrow$$
$$\rightarrow (P^{e,l} = P^{e-1,\lambda} + \Delta P) \qquad (47)$$

przy czym

$$\Delta p_i = \begin{cases} t_n \text{, jeśli } i=n \\ 0 \text{ , w przypadku przeciwnym} \end{cases}$$

Chwilę t_n wyznaczamy:

- dla $e < N$:

$$t_n = \begin{cases} T^{e-1,\lambda} + \tau_{\kappa^{e-1,\lambda},n} + \theta_n \text{, jeśli } [\frac{1}{c} T^{e-1,\lambda}]^{+} * c - T^{e-1,\lambda} \geq \theta_n + \tau_{\kappa^{e-1,\lambda},n} \\ [\frac{1}{c} T^{e-1,\lambda}]^{+} * c + \theta_n + \tau_{\kappa^{e-1,\lambda},n} \text{, w przypadku przeciwnym} \end{cases} \qquad (48)$$

- dla $e = N$:

$$t_n = \begin{cases} T^{e-1,\lambda} + \theta_n + \tau_{\kappa^{e-1,\lambda},n} + \tau_{n,o} \text{, jeśli } [\frac{1}{c} T^{e-1,\lambda}]^{+} * c - T^{e-1,\lambda} \geq \\ \qquad \geq \theta_n + \tau_{\kappa^{e-1,\lambda},n} + \tau_{n,o} \\ [\frac{1}{c} T^{e-1,\lambda}]^{+} * c + \tau_{\kappa^{e-1,\lambda},n} + \theta_n + \tau_n \text{, w przypadku przeciwnym} \end{cases} \qquad (49)$$

Do eliminacji stanów nieperspektywicznych można wykorzystać regułę dominacji opartą na twierdzeniu.

Twierdzenie 3.: Stan P^{e,l_1} dominuje nad stanem P^{e,l_2} jeśli spełniony jest warunek

$$(\Omega^{e,l_1} = \Omega^{e,l_2}) \wedge (t_\mu^{e,l_1} \leq t_\mu^{e,l_2}) \qquad (50)$$

gdzie: μ -numer ostatniej operacji wykonanej w stanie $P^{e,l}$.

Dowód tego twierdzenia można przeprowadzić analogicznie jak tw. 1.

5. Uwagi końcowe

Przedstawione w referacie algorytmy odnoszą się do dwóch różnych sytuacji na liniach montażowych. W przypadku pierwszym (deterministycznym) lepszy jest algorytm globalny, gdyż daje optymalne rozwiązanie dla całej zmiany roboczej (wielu taktów). W drugim przypadku (z zakłóceniami) lepszy jest algorytm lokalny. Aby montaż był kompletny, ostatni robot musi wykonać wszystkie operacje, które nie zostały wykonane wcześniej. Robot ten wyznacza czas taktu dla linii. W tym czasie inne roboty powinny być wykorzystane efektywnie.

Stosowanie algorytmu globalnego w przypadku zakłóceń montażu wymagało by rozwiązania problemu balansu dla każdego obiektu. Ponadto przy zakłóceniach montażu problem balansu musi być rozwiązany ponownie.

LITERATURA

[1] Assembly Automation: Procedings of the 7-th International Conference, Zurich, 1986.

[2] Marecki F.: Modele matematyczne i algorytmy alokacji operacji i zasobów na linii montażowej. ZN Pol. Śl., s. Automatyka, z.82, Gliwice 1986.

[3] Marecki F., Ptasznik K.: Model matematyczny jednowersyjnego procesu montażu na liniach. ZN Pol. Śl., s.Automatyka, z. , Gliwice 1990.

[4] Szkodny T.: Manipulatory robotów przemysłowych - modele matematyczne, ZN Pol. Śl., s. Automatyka nr 1530, Gliwice 1990.

Recenzent: Doc. dr hab. inż. K. Wala

Wysłano do redakcji do 1990-04-30.

BALANCING OF AN ASSEMBLY LINE WITH INDUSTRIAL ROBOTS

S u m m a r y

In the paper are presented two algorithms for balancing of an assembly line with industrial robots. The algorithms are based on a multi-stage programming method.

БАЛАНСИРОВАНИЕ СБОРОЧНОЙ ЛИНИИ С ПРОМЫШЛЕННЫМИ РАБОТАМИ

Р е з ю м е

В роботе представлены два алгоритма балансирования сборочной линии с промышленными роботами. Эти алгоритмы опираются на методе многоэтапного праграммирования.