ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1984

Seria: AUTOMATYKA z. 74 Nr kol. 810

Franciszek Marecki
Instytut Automatyki Politechniki Śląskiej
Gliwice

METODA PROGRAMOWANIA WIELOETAPOWEGO

Streszczenie. W pracy przedstawiono koncepcję metody programowania wieloetapowego dla harmonogramowania dyskretnych procesów przemysłowych. Podano przegląd pozycji literatury, w których były wykorzystywane algorytmy programowania wieloetapowego.

1. Wprowadzenie

Wśród problemów sterowania dyskretnymi procesami przemysłowymi istotne znaczenie ma harmonogramowanie [45 , 42]. Do rozwiązania problemu harmonogramowania stosowane są metody: programowania dynamicznego [3 , 12 , 13], programowania całkowitoliczbowego [28 , 23 , 8], podziału i ograniczeń [27 , 5 , 9 , 14 , 22] oraz grafów dysjunktywnych [2 , 1 , 10 , 15]. W niniejszej pracy zostanie przedstawiona koncepcja metody programowania wieloetapowego. Idea tej metody wywodzi się z wieloetapowych procesów decyzyjnych oraz parametryzacji metody podziału i ograniczeń. Podstawowe znaczenie w metodzie programowania wieloetapowego ma stan procesu decyzyjnego. Dla problemów harmonogramowania stan ten interpretuje harmonogram przebiegu procesu przemysłowego. Z tego względu metoda programowania wieloetapowego jest uogólnieniem metody symulacji /przepływu obiektów przez agregaty/.

Algorytmy programowania wieloetapowego były wykorzystywane do harmonogramowania różnorodnych dyskretnych procesów przemysłowych. Poniżej przedstawimy krótki przegląd niektórych zastosowań metody programowania wieloetapowego.

W pracach [39 , 51 , 52 i 58] przedstawiono problem harmonogramowania zadań w przemysłowych magazynach wysokiego składowania. Rolę agregatów odgrywają środki transportowe, natomiast obiektami są kontenery lokowane w regałach magazynu.

Harmonogramowanie procesu wykrawania blach karoseryjnych było przedmiotem kilku prac [11 , 59 , 60 , 68 i 71]. Agregatami w tym procesie są nożyce gilotynowe, natomiast obiektami - partie blach. Występują tu relacje: kolejności, wykluczania i synchronizacji obiektów, które wynikają z tzw. karty rozkroju technologicznego.

Proces wytłaczania blach karoseryjnych był analizowany w pracach [17 , 18 , 19 , 20 i 54]. Agregatami są prasy tworzące linie, natomiast obiektami - partie blach do wytłaczania. W procesie tym obiekty są podzielne w zależności od stanu zapasu wytłoczek w magazynie.

Harmonogramowanie procesu lakierowania karoserii przedstawiono w [40 i 66]. Agregatem jest tunel lakierni a obiektami - partie karoserii. Cechą charakterystyczną tego procesu jest powtórne lakierowanie niektórych karoserii.

Problemy harmonogramowania montażu były rozwiązywane w pracach [26 , 29 ,30 , 50 i 69]. Są to problemy związane z kolejnością montażu obiektów i dostarczaniem detali na linię.

W kilkunastu pracach przedstawiono harmonogramowanie kompleksów operacji w kopalniach węgla kamiennego. Harmonogramowanie pracy brygad utrzymania ruchu na jednym oddziale było analizowane w [33 , 53 , 57], natomiast problem dla wielu oddziałów przedstawiono w [4 , 7 , 24 , 34 , 37 , 56 i 61].

Innym zastosowaniem metody programowania wieloetapowego jest harmonogramowanie obsługi statków w porcie. Problem ten przedstawiono w [32 i 43]. Obiektami są statki a agregatami nabrzeża.

Harmonogramowanie procesu walcowania oraz regeneracji walców było przedmiotem prac [41 , 55 i 70]. W procesie walcowania agregatami są walcownie a obiektami partie materiału stanowiącego, tzw. wsad. Z kolei w procęsie regeneracji - obiektami są walce a agregatami napawarki i tokarki.

Metoda programowania wieloetapowego była również stosowana do harmonogramowania produkcji struktur półprzewodnikowych [21], transportu szynowego [62] oraz procesu kucia matrycowego [63].

Poza zastosowaniami przemysłowymi algorytmy programowania wieloetapowego wykorzystywano dla analizy systemów o różnych strukturach, np. pojedynczego agregatu [25], systemu o strukturze równoległej [47], szeregowej [64], drzewa lub antydrzewa [16] oraz systemu o strukturze szeregowo-równoległej [48]. W pracach tych porównano za pomocą testów komputerowych algorytmy o różnych parametrach.

Oprócz harmonogramowania metoda programowania wieloetapowego była wykorzystywana do rozwiązywania problemów balansowania linii montażowej [36 , 38 i 49], alokacji zadań i zasobów na liniach [31 , 35 i 44], a ponadto do problemów przydziału [46].

Algorytmy programowania wieloetapowego były stosowane do rozwiązywania problemów NP-zupełnych [6], z dużą liczbą ograniczeń czasowych, logicznych i przestrzennych. Efektywność tych algorytmów zależy w dużym stopniu od komputerowych struktur danych [65 , 69]. Ilustracją tego faktu są rezultaty analiz przedstawione w [11].

W oparciu o dotychczasowe zastosowania metody programowania wieloetapowego w dalszej części pracy przedstawiono jej podstawowe założenia i elementy. Celem tej pracy jest uogólnienie algorytmów i wskazanie nowych zastosowań metody programowania wieloetapowego.

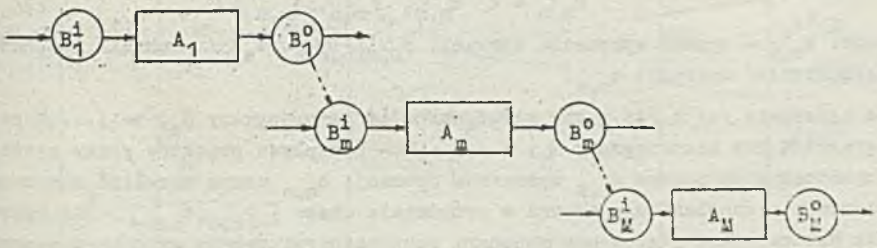
2. Problem harmonogramowania

Problem harmonogramowania jest formułowany następująco. Dany jest zbiór maszyn oraz zbiór obiektów. Obiekty przepływają przez maszyny zgodnie z określonymi marszrutami. Dane są czasy obsługi obiektów w agregatach /czasy operacji/. Na wykonanie operacji nałożone są ograniczenia czasowe i kolejnościowe. Harmonogram dopuszczalny winien określać dla każdego obiektu, maszyny i przedziały czasu wykonania operacji. Harmonogram optymalny minimalizuje przyjęte kryterium /np. czasu wykonania wszystkich operacji/.

Do problemu harmonogramowania sprowadza się również zagadnienie sterowania zasobami. W tym przypadku zakłada się, że do wykonania operacji potrzebne są dodatkowe zasoby, które są ograniczone. Dostępność zasobu może być ograniczona w każdej chwili /zasoby odnawialne/ lub globalnie w całym okresie harmonogramowania /zasoby nieodnawialne/. Istotne znaczenie ma przyjęcie modelu matematycznego operacji. W modelu tym czas wykonania operacji jest zależny od ilości /liczby/ zasobów. Wyróżnia się tym samym sposób wykonania operacji.

Sterowanie zasobami polega na określeniu harmonogramów wykonywania operacji oraz dysponowania zasobami. Harmonogram sterowania zasobami winien określać ilości /liczby/ zasobów każdego typu, które należy przydzielić do każdej maszyny w określonych przedziałach czasu. Harmonogram zasobów można wyznaczyć na podstawie harmonogramu operacji. W przypadku sterowania zasobami występuje problem polioptymalizacji. Oprócz kryteriów czasu wykonania operacji stosowane są kryteria minimalizacji zużycia zasobów.

Sformułowanie problemu harmonogramowania dyskretnych procesów przemysłowych wymaga pewnych istotnych uzupełnień. Dyskretny proces przemysłowy jest kompleksem operacji przepływu obiektów ω_n, n=1,...,N przez tzw. system wejściowo - wyjściowy, co pokazano na rys.1.



Rys.1. System o złożonej strukturze z magazynami buforowymi

Podstawowymi elementami tego systemu są agregaty A_m, m=1,...,M,/gdzie: M - liczba agregatów/ z lokalnymi magazynami buforowymi wejściowymi B_m^i oraz wyjściowymi B_m^o. Przez system ten przepływają obiekty ω_n, n=1,...,N /gdzie: N - liczba obiektów/.

Magazyny buforowe mają istotne znaczenie w dyskretnych procesach przemysłowych. Wnoszą one szereg ograniczeń z uwagi na: ograniczoną pojemność, reguły magazynowania /np. FIFO lub LIFO/ oraz ograniczony czas pobytu obiektów. W niektórych przypadkach zamiast magazynów lokalnych w systemie występują magazyny centralne. W pewnych przypadkach magazyny buforowe wnoszą skomplikowane ograniczenia logiczne przepływu obiektów.

W dyskretnych procesach przemysłowych istotne znaczenie mają nie tylko operacje technologiczne /obsługi obiektów w agregatach/ ale również operacje transportowe /przemieszczenia obiektów pomiędzy agregatami/. Z tego względu prócz zasobów skupionych /dostępnych bez zwłoki czasowej/ wyróżnia się również zasoby rozproszone /np. środki transportowe/. Rezultat wykorzystania jednostki zasobu rozproszonego jest zależny od jej położenia w systemie /zasób może być dostępny z opóźnieniem/.

Harmonogramowanie dyskretnych procesów przemysłowych jest oparte na danych normatywnych /czasu/, które są kwantylami odpowiednich rozkładów prawdopodobieństwa. Z tego względu istotne znaczenie ma niezawodność harmonogramu, określona prawdopodobieństwem jego"załamania się". Kryterium niezawodności harmonogramu jest również brane pod uwagę w polioptymalizacji.

W dalszym ciągu rozważań dla ilustracji metody programowania wieloetapowego uwzględnimy jedynie harmonogram operacji o postaci

$$H = \left\{ H_{m,n} \right\}_{\substack{m=1,\ldots,M \\ n=1,\ldots,N}}, \qquad /1/$$

gdzie: $H_{m,n}$ - harmonogram obsługi n-tego obiektu w m-tym agregacie /czyli wykonania operacji $o_{m,n}$/

przy tym

$$H_{m,n} = \langle s_{m,n}, \vartheta_{m,n}, t_{m,n} \rangle, \qquad /2/$$

gdzie: $s_{m,n}$ - sposób wykonania operacji $o_{m,n}$; $\vartheta_{m,n}/t_{m,n}$/ - chwila rozpoczęcia /zakończenia/ operacji $o_{m,n}$.

Na podstawie /1/ i /2/ można łatwo określić harmonogramy H_m, m=1,...,M pracy agregatów lub harmonogramy H_n, n=1,...,N - przepływu obiektów przez system. Ponadto znając sposób $s_{m,n}$ wykonania operacji $o_{m,n}$ można określić harmonogram sterowania zasobami skupionymi w przedziale czasu $[\vartheta_{m,n}, t_{m,n}]$. Dla wyznaczenia harmonogramu sterowania zasobami rozproszonymi sposób wykonania operacji transportowej musi uwzględnić numer i lokalizację jednostki zasobu, wykorzystanej do przemieszczenia obiektu.

3. Koncepcja programowania wieloetapowego

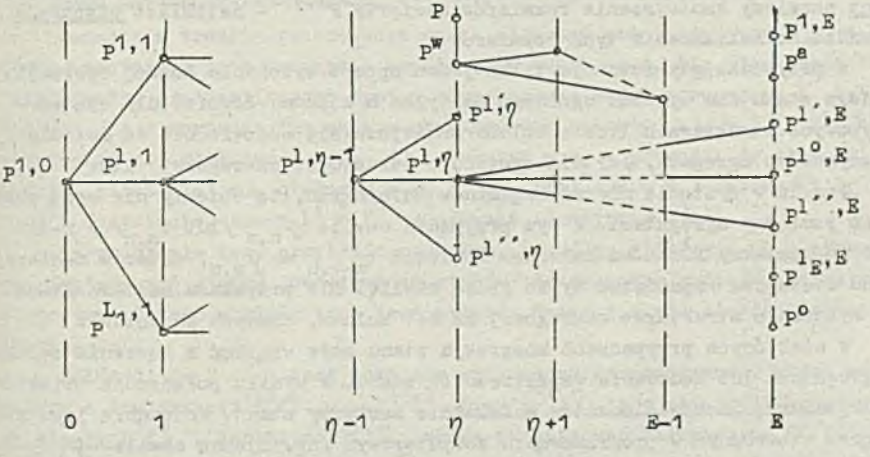
W wieloetapowych procesach decyzyjnych wyróżnia się stany, decyzje oraz funkcje transformacji stanów. W dyskretnym procesie przemysłowym stan przedstawia sytuację w systemie w pewnej chwili czasu, a więc: zaawansowanie wykonania operacji oraz wykorzystanie zasobów /i ich lokalizację/. Istotne znaczenie mają chwile czasu, w których podejmowane są decyzje o wykonaniu operacji. W procesach deterministycznych na podstawie danego stanu i decyzji można za pomocą funkcji transformacji, wyznaczyć kolejny stan. W wyniku podejmowania decyzji w kolejnych etapach, przechodzimy od stanu początkowego do końcowego. Ciąg stanów nazwiemy trajektorią a ciąg decyzji - strategią. Z każdego stanu można wygenerować wiązkę trajektorii.

W problemach kombinatorycznych /np. harmonogramowania/ wiązka trajektorii wychodząca ze stanu początkowego interpretuje drzewo decyzyjne. Węzłami tego drzewa są stany a łukami decyzje /transformacje stanów/. Dla wyznaczenia optymalnej trajektorii /optymalnego stanu końcowego/ generowane są wszystkie trajektorie. Niektóre trajektorie są nieperspektywiczne, tzn. nie prowadzą do rozwiązania optymalnego. Idea algorytmów opartych na programowaniu wieloetapowym polega na eliminacji nieperspektywicznych trajektorii.

Podstawowymi elementami konstrukcyjnymi algorytmów programowania wieloetapowego są: stan, wartość stanu, procedury generowania stanów oraz reguły eliminacji stanów nieperspektywicznych. Elementy te omówimy szczegółowo w kolejnych punktach.

3.1 Stan

Rozważmy wieloetapowy proces decyzyjny pokazany na rys.2.



Rys.2. Ilustracja ogólnego algorytmu metody programowania wieloetapowego

W procesie tym wyróżniamy etapy decyzyjne: 0, 1,..., η-1, η, η+1,...,E. Na każdym η-tym etapie występuje L_η stanów $P^{l,\eta}$. Stany numerujemy w ramach etapu /l=1,...,L_η/.

Def.1. Stan $P^{l,\eta}$ jest macierzą o wymiarach N x J.

A zatem każdemu obiektowi ω_n, n=1,...,N odpowiada jeden wiersz macierzy stanu. Liczba kolumn J jest zależna od struktury systemu. W η-tym wierszu macierzy $P^{l,\eta}$ zapisywana jest informacja o numerze m agregatu A_m /w którym jest obsługiwany obiekt ω_n/, sposobie $s_{m,n}$ /realizacji operacji $o_{m,n}$/ oraz chwili $\varsigma_{m,n}$ /lub $t_{m,n}$/ rozpoczęcia /lub zakończenia/ operacji $o_{m,n}$.

Dla pojedynczego agregatu stan $P^{l,\eta}$ jest macierzą o dwóch kolumnach /J=2/, ponieważ informacja o numerze agregatu nie jest potrzebna.

Dla systemu o strukturze równoległej stan $P^{l,\eta}$ jest macierzą o trzech kolumnach /J=3/, w których zapisujemy m, $s_{m,n}$ oraz $\varsigma_{m,n}$.

Systemy o strukturze drzewa oraz antydrzewa stanowią szeregowe połączenie pojedynczego agregatu z systemem o strukturze równoległej. Z tego względu macierz stanu dla tych systemów ma J=5 kolumn.

Dla systemu szeregowego o M agregatach macierz stanu $P^{l,\eta}$ ma J=2M kolumn. W tym przypadku do agregatu A_m są przyporządkowane kolumny o numerach 2m-1 oraz 2m. Stąd informacja o numerze agregatu wynika z numeru kolumny macierzy $P^{l,\eta}$.

System ogólny pokazany na rys.1, wykorzystuje również macierz stanu $P^{l,\eta}$ o 2M kolumnach, ponieważ najdłuższa marszruta przebiega przez M agregatów.

Dla systemu ogólnego macierz stanu ma wymiary N x 2M. Jak pokazano wyżej dla systemów o typowych strukturach liczba kolumn macierzy stanu J $\leq$ 2M. W niektórych przypadkach macierz stanu może ulec dalszej kompresji. Kompresja stanu nazwiemy zmniejszenie rozmiarów macierzy $P^{l,\eta}$ - natomiast ekspansja prowadzi do zwiększenia tych rozmiarów.

W przypadku, gdy istnieje tylko jeden sposób wykonania każdej operacji, macierz stanu dla systemu ogólnego ma tylko M kolumn. Również dla systemów o typowych strukturach liczba kolumn zmniejsza się odpowiednio do J=1 dla pojedynczego agregatu, J=2 dla systemu o strukturze szeregowej itd.

Jeżeli w systemie nie ma magazynów buforowych, to obiekty nie mogą oczekiwać pomiędzy agregatami. W tym przypadku chwile $\varsigma_{m,n}$ /lub $t_{m,n}$/ w n-tym wierszu macierzy $P^{l,\eta}$ są zależne. Ponieważ $t_{m-1,n} = \varsigma_{m,n}$, zatem w macierzy stanu wystarcza zapamiętać tylko jedną chwilę. Dla przykładu macierz stanu dla systemu o strukturze szeregowej ma M+1 kolumn, zamiast 2M kolumn.

W niektórych przypadkach kompresja stanu może wynikać z łączenia obiektów w agregatach lub kodowania współrzędnych stanu. W wyniku połączenia obiektów zmniejsza się liczba elementów w kolumnie macierzy stanu. Kodowanie jest zabiegiem stosowanym w programowaniu komputerowym /np. ujemne chwile $\varsigma_{m,n}$ dla jednego z dwóch równoległych agregatów/.

Ekspansja stanu następuje w przypadkach podziału obiektów w agregatach lub dla procesów acyklicznych. Dla procesów z repetycjami /dwukrotne przejście

obiektu przez ten sam agregat/, w stanie zapisywana jest informacja o dwóch operacjach $o^1_{m,n}$ i $o^2_{m,n}$.

3.2 Klasyfikacja stanów

W wieloetapowym procesie decyzyjnym pokazanym na rys.2., wyróżnia się następujące stany:

Stan początkowy $P^{1,0}$, który interpretuje warunki początkowe procesu, przed rozpoczęciem harmonogramowania. Jeżeli wszystkie obiekty znajdują się przed systemem, to stan $P^{1,0}$ jest macierzą zerową. W przypadku przeciwnym niektóre elementy macierzy $P^{1,0}$ są dodatnie.

Stan aktywny $P^{1,\eta}$ /perspektywiczny/, pozwala wygenerować dalsze stany.

Stan wybrany $P^{\lambda,\eta-1}$ jest stanem aktywnym, który został wybrany dla generowania dalszych stanów.

Stan wygenerowany P jest stanem otrzymanym z $P^{\lambda,\eta-1}$. Stan ten jest testowany z uwagi na perspektywiczność. Jeżeli P okaże się perspektywicznym, to zostaje stanem aktywnym. W przypadku przeciwnym jest eliminowany.

Stan wyczerpany P^W jest stanem wygenerowanym P, z którego nie można otrzymać żadnego stanu końcowego.

Stan końcowy $P^{1,E}$ jest stanem spełniającym warunki zakończenia harmonogramowania. Warunkiem tym może być wykonanie wszystkich operacji lub upłynięcie pewnego czasu w modelowanym procesie.

Stan aktualnie najlepszy P^a jest najlepszym stanem końcowym wyznaczonym w ograniczonym czasie obliczeń$_1$.

Stan lokalnie optymalny $P^{1o,E}$ jest najlepszym stanem końcowym otrzymanym ze stanu $P^{1,\eta}$.

Stan globalnie optymalny P^o jest najlepszym stanem końcowym.

Ponadto w trakcie generowania można otrzymać stany identyczne P^1 i P^2. Stany identyczne mają wszystkie współrzędne takie same. Dla ilustracji takiego przypadku przeanalizujemy system równoległy o dwóch agregatach A_1 i A_2. Załóżmy, że w systemie tym należy obsłużyć dwa obiekty ω_1 i ω_2. Niech strategia: ω_1 obsłużyć w A_1 i ω_2 obsłużyć w A_2 - prowadzi do stanu P^1. Z kolei strategia: ω_2 obsłużyć w A_2 i ω_1 obsłużyć w A_1 - prowadzi do stanu P^2. Oczywiście, stany P^1 i P^2 są wówczas identyczne.

Na podstawie tego przykładu zauważmy, że różne strategie mogą prowadzić do identycznych stanów. Generowanie stanów identycznych zmniejsza efektywność algorytmu. Z tego względu stan jest zbiorem decyzji a nie ciągiem decyzji /strategia/.

Załóżmy, że $H^{1o,E}$ jest lokalnie optymalnym harmonogramem, który otrzymujemy z trajektorii $P^{1,\eta},\ldots,P^{1o,E}$. Stan $P^{\lambda,\eta}$ jest stanem alternatywnym ze stanem $P^{1,\eta}$, jeżeli z $P^{\lambda,\eta}$ można zrealizować harmonogram $H^{1o,E}$. Zauważmy, że nie wymaga się by równocześnie z $P^{1,\eta}$ można było zrealizować harmonogram $H^{\lambda o,E}$.

Alternatywność stanów wykorzystuje się w eliminacji stanów nieperspektywicznych.

3.3 Wartość stanu

Każdy stan końcowy $P^{1,E}$ określa wprost dopuszczalny harmonogram $H^{1,E}$. Ponadto każdy stan $P^{1,\eta}$, $\eta<E$ reprezentuje harmonogram $H^{1,\eta}$. Do oceny harmonogramu H przyjmuje się kryteria optymalizacji Q_i, i=1,...,I. Analogicznie dla oceny stanu $P^{1,\eta}$ przyjmiemy wartość stanu $V^{1,\eta}$.

Def.2. Wartość stanu jest wektorem

$$V^{1,\eta} = \left[v_i^{1,\eta} \right]_{i=1,\ldots,I} \qquad /3/$$

Współrzędne tego wektora są określane na podstawie funkcji wartości $v_i\left(P^{1,\eta}\right)$. Funkcje wartości $v_i\left(P^{1,\eta}\right)$ korespondują z kryterium Q_i. A zatem znając stan $P^{1,\eta}$ można wyznaczyć jego wartość $V^{1,\eta}$. Posługiwanie się funkcjami wartości $v_i\left(P^{1,\eta}\right)$ wydłuża czas obliczeń. Dlatego w trakcie generowania stanu P /ze stanu $P^{\lambda,\eta-1}$/, równocześnie wyznaczamy jego wartość. Do tego celu służą formuły rekurencyjne, które dla kryteriów addytywnych mają postać

$$v_i = v_i^{\lambda,\eta-1} + \Delta v_i \qquad /4/$$

z warunkiem początkowym $v_i^{1,0}$.

Analogicznie można stosować formuły rekurencyjne dla kryteriów maksymalizujących lub minimalizujących.

Formuły rekurencyjne /4/ skracają czas obliczeń lecz wymagają zapamiętywania oprócz stanu $P^{1,\eta}$ jego wartości $V^{1,\eta}$.

Dla problemu jednokryterialnego wartość stanu jest skalarem. W tym przypadku stan globalnie optymalny wyznaczamy z warunku

$$\left(\min_{1\le l\le L_E} V^{l,E} = V^{l^o,E} \right) \Rightarrow \left(P^{l^o,E} = P^o \right) \qquad /5/$$

W problemach wielokryterialnych /np. z kryteriami minimalizacji/ na podstawie wartości stanu można wyznaczyć zbiór $\mathcal{B}$ stanów Pareto-optymalnych. Tak więc powiemy, że stan $P^{\lambda,E}$ dominuje nad stanem $P^{1,E}$ jeżeli spełniony jest warunek

$$\bigwedge_{1\le i\le I} \bigvee_{j\ne i} \left(v_i^{\lambda,E} \le v_i^{1,E} \right) \wedge \left(v_j^{\lambda,E} < v_j^{1,E} \right) \qquad /6/$$

Stany niezdominowane należą do zbioru stanów Pareto-optymalnych.

Dominacja stanów, określona warunkiem /6/, może być rozszerzona dla stanów alternatywnych $P^{\lambda,\eta}$ i $P^{1,\eta}$ dla etapów $\eta < E$. Twierdzenie o dominacji stanów zostanie podane w dalszej części pracy.

Dla wyznaczenia stanu polioptymalnego stosowana jest metoda dialogowa lub problem wielokryterialny sprowadza się do problemów jednokryterialnych, za pomocą funkcji użyteczności. W dyskretnych procesach przemysłowych często stosowana jest metoda hierarchizacji kryteriów.

Załóżmy, że kryteria są ustawione w hierarchii od Q_1 do Q_I. Tolerancje

wskaźników wynoszą q_i. Wówczas stan polioptymalny wyznaczamy z warunku

$$\exists_{\lambda}\ \forall_{1\le l\le L_E,\ l\neq\lambda}\ \exists_{1<j\le I}\ \forall_{i<j}\ \left(v_i^{\lambda,E} < v_i^{l,E} + q_i\right) \vee \left(\left|v_i^{\lambda,E} - v_i^{l,E}\right| \le q_i\right) \wedge \left(v_j^{\lambda,E} < v_j^{l,E} + q_j\right) \Rightarrow \left(P^{\lambda,E} = P^{o}\right) \qquad /7/$$

Jeżeli $q_I=0$, to z warunku /7/ można wyznaczyć jedyny stan polioptymalny. Ponadto, gdy stany końcowe są generowane kolejno, to warunek /7/ pozwala określić lepszy z dwóch stanów /a więc stan aktualnie najlepszy/.

W metodzie programowania wieloetapowego stany Pareto-optymalne /lub stan polioptymalny/ są wyznaczane po jednokrotnym wygenerowaniu trajektorii.

3.4 Generowanie stanów

Celem generowania stanów jest wyznaczenie kompletnej wiązki trajektorii, pokazanej na rys.2. Każda trajektoria wychodzi z danego stanu startowego $P^{1,0}$. Stany generowane są etapami, tzn. na podstawie stanu $P^{1,0}$ otrzymujemy stany $P^{l,1}$, $l=1,\ldots,L_1$, a ogólnie z wybranego stanu $P^{\lambda,\eta-1}$ generowane są stany η-tego etapu $P^{l,\eta}$, $l=1',\ldots,1''$. W ten sposób można otrzymać stany końcowe $P^{l,E}$, $l=1,\ldots,L_E$.

Z uwagi na zajętość pamięci komputerowej i czas obliczeń istotne znaczenie ma sposób generowania kompletnej wiązki trajektorii. Podstawowe znaczenie dla generowania stanów mają: lista stanów aktywnych, reguły wyboru, reguły podziału oraz procedury generowania stanów.

Do generowania stanów wykorzystywane są stany aktywne $P^{l,\eta}$, umieszczane na odpowiedniej liście. Lista stanów aktywnych $\mathcal{L}$ jest uporządkowanym zbiorem tych stanów. Uporządkowanie to polega na wyróżnieniu list $\mathcal{L}_\eta$, stanów η-tego etapu, $\eta=0,\ldots,E-1$, oraz odpowiednim ponumerowaniu stanów znajdujących się na tych listach. Sposób numeracji stanów aktywnych ma wpływ na efektywność algorytmu.

Generowanie stanów polega na wyborze pewnego stanu $P^{\lambda,\eta-1}$ i wyznaczeniu na jego podstawie stanów $P^{l,\eta}$. A zatem w trakcie generowania stanów zmienia się zawartość list $\mathcal{L}_\eta$. Po wygenerowaniu wszystkich stanów ze stanu $P^{\lambda,\eta-1}$ stan ten jest usuwany z listy $\mathcal{L}_{\eta-1}$. Wygenerowane stany są wprowadzane na listę $\mathcal{L}_\eta$. Ze stanów $P^{\lambda,E-1}$ generowane są stany końcowe, które nie są aktywne. Generowanie stanów kończy się, jeżeli lista stanów aktywnych jest pusta.

Reguły wyboru stanu aktywnego służą do wyznaczenia stanu $P^{\lambda,\eta-1}$, z którego będą generowane dalsze stany. Przykładem klasycznych reguł wyboru są: FIFO /najwcześniej wygenerowany - najwcześniej wybrany/, LIFO /najpóźniej wygenerowany - najwcześniej wybrany/, LLB /wybór stanu o najmniejszym dolnym ograniczeniu wartości stanu lokalnie optymalnego/ itp. Konsekwencją reguły FIFO jest lista $\mathcal{L}$ składająca się ze stanów aktywnych dwóch sąsiednich etapów. W rezultacie otrzymujemy algorytm "bez powrotów", w którym wszystkie

trajektorie są generowane równocześnie. Reguły LIFO i LLB dają listy stanów aktywnych z różnych etapów. Trajektorie są generowane fragmentami, a w algorytmie występują "powroty". Reguły wyboru dają różną zajętość pamięci komputerowej, wynikającą z liczby zapamiętywanych stanów aktywnych.

Reguły podziału określają podział wiązki trajektorii generowanej z wybranego stanu $P^{\lambda,\eta-1}$. Jeśli podział jest zupełny, to ze stanu $P^{\lambda,\eta-1}$ są generowane wszystkie jego bezpośrednie następniki $P^{l,\eta}$. Wówczas stan $P^{\lambda,\eta-1}$ przestaje być aktywnym. W przypadku podziału częściowego, ze stanu $P^{\lambda,\eta-1}$ generowanych jest tylko część jego bezpośrednich następników $P^{l,\eta}$. A zatem stan $P^{\lambda,\eta-1}$ pozostaje aktywny. Reguły częściowe pozwalają na respektowanie ograniczeń wielkości list stanów aktywnych. W szczególnym przypadku ze stanu $P^{\lambda,\eta-1}$ jest generowany tylko jeden stan $P^{l,\eta}$. Pozwala to na zapamiętywanie w trakcie obliczeń tylko jednego stanu aktywnego. Jednakże stosowanie reguł częściowego podziału wymaga zapamiętywania pewnych dodatkowych informacji o stanie.

Procedura generowania stanów jest zadaniem logicznym, z którego wynika, że jeśli jest spełniony określony warunek logiczny, to można wygenerować stan. Ponadto sprecyzowane są formuły pozwalające wyznaczyć elementy macierzy nowego stanu. Procedury generowania stanów mogą być jednokrokowe lub wielokrokowe. W przypadku procedury jednokrokowej ze stanu $P^{\lambda,\eta-1}$ otrzymujemy stan $P^{l,\eta}$, /przez przydzielenie jednego obiektu do jednego agregatu/. Procedura wielokrokowa pozwala wygenerować ze stanu $P^{\lambda,\eta-1}$, stan $P^{l,k}$, $k>\eta$. Procedury wielokrokowe występują w procesach, w których obiekty mogą być łączone.

Celem generowania stanów jest wyznaczenie wszystkich trajektorii. Jednocześnie generowanie stanów identycznych zmniejsza efektywność algorytmu. Dla eliminacji generowania stanów identycznych w procedurach stosuje się porządek leksykograficzny. Dla przykładu, w systemie o strukturze równoległej z niezależnymi operacjami obsługi - w stanie $P^{\lambda,\eta-1}$ dostępne są agregaty A_m spełniające warunek: $\mu \leq m$, gdzie μ - jest najwyższym numerem agregatu wykorzystanego w stanie $P^{\lambda,\eta-1}$.

W systemach /o strukturze drzewa, antydrzewa, szeregowo-równoległej/ z ograniczonymi magazynami buforowymi w procedurach generowania stanów stosuje się porządek chronologiczny. Podejmowane decyzje muszą być zgodne z chronologią dopływu i wypływu obiektów z magazynu.

3.5 Eliminowanie stanów

Wygenerowany stan P jest testowany z uwagi na perspektywiczność. Stan ten jest nieperspektywiczny jeżeli nie pozwala wyznaczyć rozwiązania optymalnego. A zatem w tym przypadku stan P i wiązka trajektorii, które z niego wychodzą, mogą być pominięte. W przypadku przeciwnym stan P zostaje umieszczony na liście stanów aktywnych. Do eliminacji stanów służą reguły wyczerpywania, dominacji i sondowania.

Reguła wyczerpywania jest warunkiem logicznym jaki musi spełniać stan P, by nie można było z niego wygenerować stanu końcowego $P^{1,E}$. Warunek taki może wynikać z ograniczeń procesu, np. wykluczenie przestojów agregatów, wykluczenie oczekiwania obiektów, wykluczenie opóźnienia operacji itp. Ogólne twierdzenie o wyczerpywaniu ma postać:

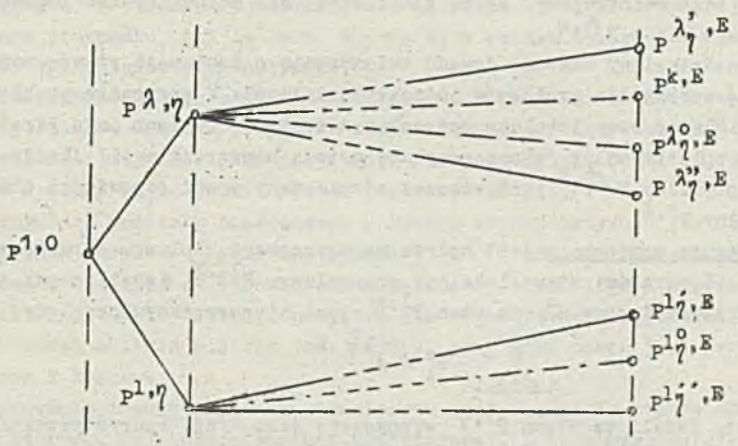
Tw.1. Jeżeli w stanie P pewnej operacji nie można wykonać bez naruszenia ograniczeń procesu, to wyczerpane zostały możliwości generowania rozwiązań dopuszczalnych /stanów końcowych/.

Istota tego twierdzenia polega na tym, że generowanie trajektorii ze stanu P jest nie celowe, ponieważ na pewnym etapie otrzymamy stan, który nie ma bezpośredniego następnika. Takie generowanie jest możliwe, ponieważ w procedurach generowania stanów uwzględniane są dopuszczalne decyzje. A zatem wykrycie, że stan P nie daje rozwiązania dopuszczalnego, eliminuje wiązkę trajektorii z dalszych obliczeń.

Relację dominacji określimy następująco. Załóżmy, że dane są dwa stany $P^{\lambda,\eta}$ i $P^{1,\eta}$. Niech $P^{\lambda^o,E}$ i $P^{1^o,E}$ będą odpowiednio stanami lokalnie optymalnymi. Jeżeli stany $P^{\lambda^o,E}$ i $P^{1^o,E}$ spełniają warunek /6/, to mówimy, że stan $P^{\lambda,\eta}$ dominuje nad stanem $P^{1,\eta}$. W trakcie obliczeń na η-tym etapie nie znane są stany $P^{\lambda^o,E}$ i $P^{1^o,E}$. O dominacji stanów $P^{\lambda,\eta}$ i $P^{1,\eta}$ rozstrzyga następujące ogólne twierdzenie.

Tw.2. Jeżeli stan $P^{\lambda,\eta}$ jest alternatywny ze stanem $P^{1,\eta}$, a ponadto $v_i^{\lambda,\eta} \leq v_i^{1,\eta}$, i=1,...,I, to stan $P^{\lambda,\eta}$ dominuje nad stanem $P^{1,\eta}$.

Idea dowodu jest zilustrowana na rys.3.



Rys.3. Ilustracja reguły dominacji

Załóżmy, że ze stanu $P^{\lambda,\eta}$ otrzymujemy wiązkę trajektorii $P^{\lambda,\eta},\ldots,P^{\lambda_\eta,E}$, $\lambda_\eta=\lambda'_\eta,\ldots,\lambda''_\eta$. Analogicznie ze stanu $P^{l,\eta}$ otrzymujemy wiązkę trajektorii $P^{l,\eta},\ldots,P^{l_\eta,E}$, $l_\eta = l'_\eta,\ldots,l''_\eta$. Realizując ze stanu $P^{\lambda,\eta}$ optymalny harmonogram $H^{\lambda^\circ_\eta,E}$ otrzymujemy stan lokalnie optymalny $P^{\lambda^\circ_\eta,E}$. Analogicznie ze stanu $P^{l,\eta}$ otrzymujemy dla harmonogramu optymalnego $H^{l^\circ_\eta,E}$, stan lokalnie optymalny $P^{l^\circ_\eta,E}$. Ponadto, ponieważ stan $P^{\lambda,\eta}$ jest alternatywny ze stanem $P^{l,\eta}$ zatem harmonogram $H^{l^\circ_\eta,E}$ jest realizowalny od stanu $P^{\lambda,\eta}$. W rezultacie realizacji tego harmonogramu od stanu $P^{\lambda,\eta}$ otrzymujemy stan końcowy $P^{k,E}$, który nie jest lokalnie optymalny. A zatem stan $P^{k,E}$ nie dominuje nad stanem $P^{\lambda^\circ_\eta,E}$. Dla trajektorii $P^{1,0},\ldots,P^{l,\eta},\ldots,P^{l_\eta,E}$ możemy napisać

$$v_i^{l^\circ_\eta,E} = f\left(v_i^{l,\eta},\ \Delta v_i\right)\quad i=1,\ldots,I, \qquad /8/$$

gdzie: f - funkcja kryterialna /w postaci sumy lub max /, Δv_i - składnik i-tej współrzędnej wektora wartości, wyznaczony z harmonogramu $H^{l^\circ_\eta,E}$.

Analogicznie dla trajektorii $P^{1,0},\ldots,P^{\lambda,\eta},\ldots,P^{k,E}$ otrzymamy

$$v_i^{k,E} = f\left(v_i^{\lambda,\eta},\ \Delta v_i\right)\quad i=1,\ldots,I \qquad /9/$$

Ponieważ $v_i^{\lambda,\eta} \leq v_i^{l,\eta}$, $i=1,\ldots,I$, więc z /8/ i /9/ otrzymujemy

$$v_i^{k,E} \leq v_i^{l^\circ_\eta,E} \qquad /10/$$

A zatem zgodnie z /6/ stan $P^{l^\circ_\eta,E}$ jest zdominowany przez stan $P^{k,E}$ lub co najwyżej mu równoważny. Ponieważ równocześnie stan $P^{k,E}$ nie dominuje nad stanem $P^{\lambda^\circ_\eta,E}$, stąd wnioskujemy, że ze stanu $P^{l,\eta}$ nie można uzyskać lepszego stanu końcowego niż z $P^{\lambda,\eta}$.

Przedstawiony schemat dowodu twierdzenia o dominacji stanów można wykorzystać wprost dla problemów jednokryterialnych. W przypadku problemu wielokryterialnego stan lokalnie optymalny może być rozumiany jako Pareto-optymalny lub polioptymalny /wyznaczony np. metodą hierarchizacji/. Każdy stan Pareto-optymalny $P^{l^\circ_\eta,E}$ jest wówczas zdominowany przez odpowiedni stan Pareto-optymalny $P^{k,E}$.

Reguła sondowania jest oparta na oszacowaniu dolnego ograniczenia $d_i^{l^\circ_\eta,E}$ $i=1,\ldots,I$ wartości stanu lokalnie optymalnego $P^{l^\circ_\eta,E}$. Jeżeli znany jest stan aktualnie najlepszy P^a, to stan $P^{l,\eta}$ jest nieperspektywiczny, gdy

$$\bigvee_{1\leq i\leq I} v_i^a \leq d_i^{l^\circ_\eta,E} \qquad /11/$$

Ponadto, jeżeli ze stanu $P^{l,\eta}$ wyznaczony jest /np. heurystycznie/ pewien stan końcowy $P^{k,E}$, spełniający warunek

$$\bigvee_{1\leq i\leq I} v_i^{k,E} = d_i^{l^\circ_\eta,E} \qquad /12/$$

to stan ten jest lokalnie optymalny. A zatem generowanie wiązki trajektorii

ze stanu $P^{l,\eta}$ jest niepotrzebne, gdyż stan $P^{l^{o}_{\eta},E}$ dominuje nad pozostałymi stanami końcowymi $P^{L_\eta,E}$.

Przedstawione koncepcje reguł eliminacji stanów nieperspektywicznych mogą być wykorzystane w wielu typowych problemach harmonogramowania dyskretnych procesów przemysłowych. W niektórych problemach reguły eliminacji stanów /szczególnie dominacji/ ulegają modyfikacjom.

3.6 Efektywność algorytmów

Metoda programowania wieloetapowego pozwala konstruować algorytmy przeglądowe, którymi można rozwiązywać problemy należące do klasy NP-zupełnych /lub również do klasy P/. Stosowanie algorytmów przeglądowych do rozwiązywania problemów NP-zupełnych jest powszechne. Efektywność tych algorytmów, w sensie czasu obliczeń i zajętości pamięci komputerowej, zależy od wielu parametrów oraz danych liczbowych.

Zajętość pamięci komputerowej jest określana liczbą stanów, które są zapamiętywane w trakcie obliczeń. W metodzie programowania wieloetapowego stan zawiera historię podejmowanych decyzji. W związku z tym wystarczy zapamiętywać tylko ostatni stan każdej trajektorii /stan aktywny/.

Liczba zapamiętywanych stanów aktywnych jest zależna od stosowanej reguły wyboru. Dla reguły FIFO zapamiętywane są wszystkie stany co najwyżej dwóch sąsiednich etapów. Reguły LIFO lub LLB powodują zapamiętywanie niektórych stanów z etapów η, $0 < \eta < E$.

Liczba zapamiętywanych stanów aktywnych może być ograniczona dopuszczalną pojemnością L_η zbioru stanów aktywnych $\mathcal{A}_\eta$. A zatem z wybranego stanu aktywnego $P^{\lambda,\eta-1}$ można wygenerować nie wszystkie stany lub tylko $L_\eta - |\mathcal{A}_\eta|$ W skrajnym przypadku, gdy $L_\eta = 1$, $0 \leq \eta \leq E$, w każdym zbiorze $\mathcal{A}_\eta$ zapamiętywany jest tylko jeden stan aktywny.

Ograniczenie liczby stanów aktywnych L_η wymaga stosowania reguły częściowego podziału. W takim przypadku z wybranego stanu aktywnego $P^{\lambda,\eta-1}$ są generowane nie wszystkie jego następniki /bezpośrednie/. Poza tym, stan $P^{\lambda,\eta-1}$ może być wybrany wielokrotnie /jeśli pozostaje aktywny/.

W regułach podziału częściowego z każdym stanem aktywnym jest związany jego wskaźnik /indeks/. Jeśli następniki $P^{\lambda,\eta-1}$ są generowane w uporządkowany sposób, to wskaźnik stanu określa parametry ostatniego z wygenerowanych następników /ostatniej decyzji podjętej dla wygenerowania tego następnika/. Wskaźnik stanu zmienia się tak jak licznik, przy czym znana jest jego wartość początkowa i końcowa.

Stosowanie reguły podziału częściowego oraz wskaźników stanów aktywnych pozwala sterować zajętością pamięci komputerowej. Z tego względu efektywność algorytmu w sensie zajętości pamięci przestaje być problemem. Zatem w algorytmach programowania wieloetapowego zapamiętywane są pewne dodatkowe parametry stanów /np. chwile zwalnienia agregatów, ostatni obiekt realizowany

w agregacie, wartość stanu, zużycie zasobów,itp/, które skracają czas obliczeń.

Efektywność algorytmu w sensie czasu obliczeń jest zależna od liczby wyeliminowanych stanów. Skuteczność reguły sondowania jest związana z aktualnie najlepszym stanem końcowym P^a. Jeżeli stosowane reguły wyboru i podziału są dobre, to stan P^a jest bliski stanowi optymalnemu P^o. Pozwala to na eliminację większej liczby stanów nieperspektywicznych.

Parametrami określającymi jakość reguły eliminacji stanów nieperspektywicznych są: czas testowania i prawdopodobieństwo wyeliminowania stanu. Jeżeli testowany stan nie zostaje wyeliminowany, to czas obliczeń algorytmu wydłuża się. W przypadku przeciwnym czas ten skraca się ponieważ nie jest generowana wiązka trajektorii wychodząca z wyeliminowanego stanu.

W ogólnym przypadku czas testowania jest większy dla reguł, które eliminują stan z większym prawdopodobieństwem. W niektórych przypadkach prawdopodobieństwo wyeliminowania stanu rośnie wraz ze wzrostem η /np. ograniczony przedział czasu pracy agregatu może być przekroczony z większym prawdopodobieństwem jeżeli przydzielono do niego więcej obiektów/.

Reguła dominacji stanów wymaga porównania dwóch stanów: wygenerowanego P i aktywnego $P^{l,\eta}$. Aby skrócić czas poszukiwania stanu $P^{l,\eta}$, który podlega dominacji z P, wprowadza się leksykograficzne uprządkowanie stanów aktywnych. Konstrukcja porządku leksykograficznego wynika z reguły dominacji stanów.

Jeżeli czas obliczeń komputerowych jest limitowany, to zamiast rozwiązania optymalnego P^o można uzyskać jedynie rozwiązanie aktualnie najlepsze P^a. Błąd względny ε_i tego rozwiązania dla i-tego kryterium wynosi

$$\varepsilon_i \leq \frac{v_i^a - d_i}{d_i}, \qquad /13/$$

gdzie: d_i - najmniejsze dolne ograniczenie wartości stanu lokalnie optymalnego.

A zatem algorytm programowania wieloetapowego pozwala wyznaczyć rozwiązanie optymalne lub przybliżone /gdy limitowany jest czas obliczeń komputerowych/.

4. Zakończenie

W referacie przedstawiono w sposób ogólny koncepcję metody programowania wieloetapowego. Wyróżniono podstawowe elementy konstrukcji algorytmów dających rozwiązanie dokładne lub przybliżone. Zastosowanie metody programowania wieloetapowego do harmonogramowania procesów dyskretnych jest dogodne z uwagi na interpretację stanu. Programowanie wieloetapowe może być stosowane do harmonogramowania procesów acyklicznych i cyklicznych.

Metoda programowania wieloetapowego jest przydatna do rozwiązywania problemów harmonogramowania dyskretnych procesów przemysłowych z wyróżnionymi:obiektami, agregatami i magazynami buforowymi. Efektywność algorytmów jest większa przy dużej liczbie ograniczeń procesu. Dla problemów o małej liczbie ograniczeń oraz obiektach i zasobach podzielnych w sposób ciągły, metoda ta nie

jest zalecana.

Dobra metoda harmonogramowania winna dać rozwiązanie optymalne lub przybliżone /z oszacowaniem dokładności/ gdy limitowany jest czas obliczeń. W praktyce ważna jest również ocena niezawodności wyznaczanego harmonogramu. Wynika to z faktu,że harmonogram jest wyznaczany na podstawie danych normatywnych, które są zakłócane w trakcie procesu rzeczywistego. W związku z tym harmonogramy "załamują się" po pewnym czasie. Prowadzi to do polioptymalizacji harmonogramowania. Harmonogram polioptymalny winien mieć dobrą niezawodność i zapewniać efektywną realizację procesu.

Wydaje się, że metoda programowania wieloetapowego znajdzie dalsze zastosowania do polioptymalnego harmonogramowania dyskretnych procesów przemysłowych.

LITERATURA

[1] Adrabiński A.,Wodecki M.,Grabowski J.: Algorytm optymalizacji zagadnień sekwencyjnych w dyskretnych procesach produkcyjnych. ZN Politechniki Śląskiej,s.Automatyka nr 46, Gliwice 1978, ss.90-99

[2] Balas E.: Machine sequencing via disjunctive graphs; an implicite enumeration algorithm. Operation Research, V.17,No.6, 1969, pp.941-957

[3] Bellman R.: Adaptacyjne procesy sterowania, PWN, Warszawa 1965

[4] Biolik L.: Opracowanie i zbadanie efektywności algorytmów harmonogramowania pracy wielu brygad utrzymania ruchu w wielu oddziałach wydobywczych KWK. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

[5] Błażewicz J.,Cellary W.,Słowiński R.,Węglarz J.: Algorytmy sterowania rozdziałem zadań i zasobów w kompleksie operacji. WPP, Poznań 1976

[6] Coffman E.G. /red./: Teoria szeregowania zadań. WNT, Warszawa 1980

[7] Dusza K.,Marecki F.: Automatisiertes Leitssystem für Schweissarbeiten im Schacht. ARS'81, Ostrava 1981

[8] Garfinkel R.S.,Nemhauser G.L.: Programowanie całkowitoliczbowe, PWN, Warszawa 1976

[9] Grabowski J.: Zagadnienia kolejnościowe optymalizacji ciągów krytycznych w dyskretnych systemach produkcyjnych. ZN Politechniki Śląskiej, s. Automatyka, nr 54, Gliwice 1980, ss.47-56

[10] Goetz J.: Modelowanie na grafie dysjunktywnym ogólnych zagadnień przydziału zasobów i szeregowania operacji. ZN Politechniki Śląskiej, s. Automatyka, nr 54, Gliwice 1980, ss.33-46

[11] Hajewski M.: Poszukiwanie efektywnego algorytmu harmonogramowania zadań w systemach o strukturze równoległej. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki , Gliwice 1983

[12] Held M.,Karp R.M.: A dynamic programming approach to sequencing problems. SIAM, Journal on Applied Mathematics, V.10,No.1, 1962, pp.196-210

[13] Held.M.,Karp R.M.: The construction of discrete dynamic programming algorithms. IBM Systems Journal, V.4,No.2, 1965, pp.136-147

[14] Jamroż L.: O pewnym zastosowaniu metody podziału i ograniczeń do rozwiązywania problemu szeregowania zadań. ZN Politechniki Śląskiej, s.Automatyka, nr 63, Gliwice 1982, ss.19-28

[15] Janiak A.: Problem kolejnościowy gniazdowy z rozdziałem zasobów. ZN Politechniki Śląskiej, s.Automatyka, nr 63, Gliwice 1982,ss.29-39

[16] Jarosz H.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemach o strukturze drzewa i antydrzewa. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

[17] Jurczyk Z.: Harmonogramowanie zadań na linii technologicznej. Mat. III Konferencji nt."Metody i środki projektowania automatycznego", Instytut Podstaw Budowy Maszyn Polit. Warszawskiej, Warszawa 1981

[18] Jurczyk Z.: Kaliendarnoje płanirowanije raboty sztampowki linii. ARS'81, Ostrawa 1981

[19] Jurczyk Z.: Kaliendarnoje płanirowanije raboty sztampowocznoj linii s uczietom fiksirowannych riesursnych ograniczienij. 4-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981, V.IV, pp. 266-272

[20] Jurczyk Z.,Marecki F.: Discrete process scheduling in a system with series structure. 5-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983

[21] Kempny B.: Algorytm harmonogramowania produkcji struktur półprzewodnikowych. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

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

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

[24] Kowalowski H.,Marecki F.,Starzyczny L.,Dusza K.: An algorithm of independent tasks scheduling in a system with dispersed parameters. 4-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981, V.III,pp.8-16

[25] Ksyta J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych realizowanych w jednym agregacie. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

[26] Lach H.: Modele harmonogramowania i sterowania montażem podwozi ciągników MF w ZM URSUS. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1981

[27] Land A.H.,Doig A.G.: An automatic method of solving discrete programming problems. Econometrica, V.28,No.3, 1960, pp.497-520

[28] Mann A.S.: On the job-shop scheduling problem. Operation Research, V.8,No.2, 1960, pp.219-223

[29] Marecki F.: Problem kolejnościowy w montażu wielowersyjnym. ZN Politechniki Śląskiej, s.Automatyka, nr 54, Gliwice 1980, ss.119-128

[30] Marecki F.: Harmonogramowanie dostaw detali na linię montażową. ZN Politechniki Śląskiej, s.Automatyka, nr 55, Gliwice 1980, ss.51-61

[31] Marecki F.: Static control of tasks allocation on assembly line. 4-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981, V.III,pp.17-23

[32] Marecki F.: Sterowanie obsługą statków w porcie. Konferencja nt."Cybernetyka w gospodarce morskiej", WSM, Gdynia 1981, t.2, ss.85-96

[33] Marecki F.,Starzyczny L.: Uprawlienije processami dobyczi na podzjemnoj szachtie. Konferencja nt.:"Awtomatizacja gornodobywajuszczijej promyszliennosti", Nauczno-Tiechniczieskije Sojuzy Bałgarii, Warna 1981

[34] Marecki F.,Dusza K.: Centralizowannoje dispietczerskoje uprawlienije remontnymi i konserwacjonnymi robotami. Konferencja nt.:"Awtomatizacja gornodobywajuszczej promyszliennosti", Nauczno-Tiechniczeskije Sojuzy Bałgarii, Warna 1981

[35] Marecki F.: Leitung der Quellenverteilung im System Technologischer Streken. ARS'81, Ostrava 1981

[36] Marecki F.: Assembly line balancing problem. Prace Naukowe Instytutu Cybernetyki Technicznej, nr 63, s. Konferencje, nr 22, "System Science VII", Wrocław 1981, ss.136-138

[37] Marecki F.: An algorithm of dependent tasks scheduling in a system with dispersed parameters. Prace Naukowe Instytutu Cybernetyki Technicznej, nr 63, s. Konferencje, nr 22, "System Science VII", Wrocław 1981, ss.132-135

[38] Marecki F.: Bałansowanie szeregowej linii montażowej z ograniczeniami wykluczania operacji. ZN Politechniki Śląskiej, s.Automatyka, nr 63, Gliwice 1982, ss.81-90

[39] Marecki F.: Harmonogramowanie załadunku kontenerów w przemysłowym magazynie wysokiego składowania. ZN Politechniki Śląskiej, s.Automatyka, nr 64, Gliwice 1982, ss.69-82

[40] Marecki F.,Ślesińska I.: Problem harmonogramowania procesu lakierowania karoserii. ZN Politechniki Śląskiej, s.Automatyka, nr 64, Gliwice 1982, ss.83-94

[41] Marecki F.,Zielińska-Król E.: Model matematyczny i algorytm harmonogramowania procesu regeneracji walców. ZN Politechniki Śląskiej, s.Automatyka, nr 64, Gliwice 1982, ss.95-108

[42] Marecki F.; Control of discrete processes. 5-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983,

[43] Marecki F.: Model sterowania przeładunkiem statków w porcie. Konferencja nt.:"Cybernetyka w gospodarce morskiej", WSM, t.II, Gdynia 1983,ss.104-112

[44] Marecki F.: Wielokryterialne bałansowanie linii montażowej. XXIII Sympozjon nt.:"Modelowanie w mechanice", PTMTiS, Gliwice 1984

[45] Niederliński A.: Harmonogramowanie produkcji a wielopoziomowe, wielowymiarowe dyskretne układy regulacji nadążnej. ZN Politechniki Śląskiej s.Automatyka, nr 55, Gliwice 1980, ss.63-70

[46] Niepokój J.: Analiza algorytmów problemu przydziału z ograniczeniami czasowymi. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1984

[47] Prejmar J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemach o strukturze równoległej. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

[48] Przegędza P.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemie o strukturze szeregowo-równoległej. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1984

[49] Przybysz T.: Bałansowanie szeregowej linii montażowej. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1981

[50] Przybysz T.,Jurczyk S.,Pawlik S.,Marecki F.: System sterowania montażem podwozi ciągnika licencyjnego. ZN Politechniki Śląskiej, s.Automatyka, nr 64, Gliwice 1982, ss.109-114

[51] Przybysz T.: Industrial high storage control. 5-th International Conference on: "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983

[52] Przybysz T.: Polioptymalne harmonogramowanie w magazynach wysokiego składowania. XXIII Sympozjon nt.:"Modelowanie w mechanice", PTMTiS, Gliwice 1984

[53] Pyszny H.: Sterowanie kompleksem operacji wykonywanych przez wiele brygad na jednym oddziale KWK. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[54] Serafin J.: Ocena efektywności wybranych algorytmów harmonogramowania na przykładzie tłoczni blach karoseryjnych. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[55] Staszewski A.: Harmonogramowanie procesu walcowania ciągłego z uwzględnieniem współpracy z systemem walcowni finalnych. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

[56] Starzyczny L.,Dusza K.,Marecki F.: Harmonogramowanie niezależnych zadań w systemie o rozproszonych parametrach. ZN Politechniki Śląskiej, s.Górnictwo, nr 112, Gliwice 1981

[57] Starzyczny L.,Marecki F.: **Operative Leitung der Instandhaltung im Kohlenbergwerk.** ARS'81, Ostrava 1981

[58] Stokłosa J.: Harmonogramowanie pracy przemysłowego magazynu wysokiego składowania. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[59] Szendzielorz K.: Kaliendarnoje płanirowanije riezanija karoseryjnych listow. ARS'81, Ostrava 1981

[60] Szendzielorz K.: Harmonogramowanie zadań na maszynach równoległych. III Konferencja nt.:"Metody i środki projektowania automatycznego", Instytut Podstaw Budowy Maszyn, Warszawa 1981

[61] Szwarc J.: Model matematyczny i algorytm harmonogramowania pracy wielu brygad utrzymania ruchu w wielu oddziałach wydobywczych KWK. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[62] Ślesińska I.,Marecki F.: Niezawodność funkcjonowania systemu transportu szynowego. Konferencja nt.:"X dni jakości i niezawodności", NOT, Gliwice 1981

[63] Torońska-Łuczyńska M.: Sistiema uprawlienija proizwodstwom w kuźniecznom processie. 5-th International Conference on:" Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983

[64] Turzański J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemie o strukturze szeregowej. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1983

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

[66] Wikierska I.: Uprawlienije processom okraski kuzow. 5-th International Conference on:"Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983

[67] Wirth N.: Algorytmy + struktury danych = programy. PWN, Warszawa 1981

[68] Wołany D.:Harmonogramowanie procesu wykrawania blach karoseryjnych. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[69] Woszczyk J.: Harmonogramowanie procesu montażu karoserii samochodowych. Praca dyplomowa magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1982

[70] Zielińska-Król E.: Probliemy iskliuczenija sostojanij pri kaliendarnom płanirowanii processa regeneracji wałkow. 5-th International Conference on:"Control Systems and Computer Science", Politechnical Institut of Bucharest, Bucharest 1983

[71] Zemelka P.:Aneliza sterowania procesem wykrawanie blach karoseryjnych. Prace dyplomowe magisterska /niepublikowana/, Instytut Automatyki, Gliwice 1980

Recenzent: Prof.dr hab.inż.Stanisław Piasecki
Wpłynęło do Redakcji do 30.03.1984r.

МЕТОД МНОГОЭТАПНОГО ПРОГРАММИРОВАНИЯ

Р е з ю м е

В работе представлена концепция метода многоэтапного программирования для календарного планирования дискретных производственных процессов.

Дан обзор литературы, где был использован алгоритм многоэтапного программирования.

MULTISTAGE PROGRAMMING METHOD

S u m m a r y

In the paper a conception of multistage programming method for discrete industrial processes scheduling is presented. A survey of bibliography on the application of multistage programming algorithm is given.