

Ewa DUDEK-DYDUCH, Lidia DUTKIEWICZ
Akademia Górniczo-Hutnicza

METODA ZADAŃ ZASTĘPCZYCH DO ROZWIĄZYWANIA NP-TRUDNYCH PROBLEMÓW SZEREGOWANIA

Streszczenie. W artykule przedstawiona jest heurystyczna metoda służąca do rozwiązywania skomplikowanych problemów szeregowania. Polega ona na tym, że w każdym stanie procesu decyzja podejmowana jest na podstawie specjalnie skonstruowanego zastępczego zadania optymalizacji. Metoda opisana jest na bazie modelu algebraiczno-logicznego. Opisany został też NP-trudny problem udostępniania pól eksploatacyjnych oraz algorytm jego rozwiązania oparty na proponowanej metodzie.

SUBSTITUTION TASK METHOD FOR NP-HARD SCHEDULING PROBLEMS

Summary. The paper deals with a heuristic method for complex scheduling problems. According to this method a substitution optimization task is created in each state of the decision process. The method is described with the use of an algebraic-logical model. An NP-hard problem of preparing access to exploitation fields and algorithm for this problem based on the method is also described.

1. Wprowadzenie

Obecnie intensywnie prowadzone są badania, dotyczące przybliżonych metod rozwiązywania skomplikowanych problemów, w szczególności należących do klasy problemów NP-trudnych. Niniejszy artykuł związany jest z tym kierunkiem badań. Jego celem jest przedstawienie heurystycznej metody, należącej do obszaru sztucznej inteligencji, nazwanej metodą zadań zastępczych (MZZ).

Podstawą do przedstawienia metody jest ogólny algebraiczno-logiczny model problemu optymalizacji sterowania dyskretnego procesu produkcyjnego. Istotą tej klasy modeli jest fakt, że wykorzystywane są w nich różnorodne formy opisu, a więc obok opisu analitycznego stosowany jest opis za pomocą formuł logicznych. Równocześnie model ten odpowiada pewnej formalnej postaci wieloetapowego procesu decyzyjnego połączonego z symulacją dyskretnego procesu produkcyjnego.

Model algebraiczno-logiczny został zaproponowany w [1], [8], a metody i algorytmy bazujące na nim przedstawione m.in. w [2]-[7], [11]. Model ten określa poniższa definicja.

Definicja 1. *Dyskretny proces* P jest to proces jednoznacznie określony przez czwórkę:

$$P = (s_0, f, S_N, S_G)$$

gdzie: $f: U \times S \rightarrow S$ jest funkcją częściową (a więc określoną tylko dla pewnych par $(u, s) \in U \times S$), zwaną funkcją przejścia, przy czym:

- zbiór U jest nazywany zbiorem decyzji sterujących lub sygnałów sterowania,
- zbiór $S = X \times T$ jest nazywany zbiorem stanów uogólnionych,
- zbiór X jest nazywany zbiorem stanów właściwych,
- $T \subset \mathbb{R}^+$ jest podzbiorem nieujemnych liczb rzeczywistych reprezentujących chwile czasowe,

$s_0 = (x, t)$ jest uogólnionym stanem początkowym, $s_0 \in S$,

$S_N \subset S$ jest zbiorem uogólnionych stanów niedopuszczalnych,

$S_G \subset S$ jest niepustym zbiorem uogólnionych stanów docelowych, a więc stanów, w których proces powinien się znaleźć w wyniku działania właściwych sterowań.

Zadanie poszukiwania rozwiązania dopuszczalnego polega na znalezieniu ciągu decyzji, wyznaczającego trajektorię dopuszczalną. Zadanie optymalizacji sterowania procesem polega na znalezieniu takiego ciągu sterowań dopuszczalnych, który ekstremalizuje pewien funkcjonal Q . Zadanie optymalizacji określone jest zatem przez model procesu oraz funkcjonal Q i zapisywane jest jako para (P, Q) . Symbol P reprezentuje tutaj wszystkie ograniczenia nałożone na proces.

2. Idea metody zadań zastępczych jako metody sztucznej inteligencji

Metoda zadań zastępczych polega na zdekomponowaniu (przybliżonym) zadania generowania rozwiązania na szereg dynamicznie tworzonych zadań zastępczych, przy czym każde zadanie zastępcze związane jest z odpowiednim dla niego zestawem tzw. celów pośrednich. Jako cel pośredni rozumie się osiągnięcie wyróżnionego podzbioru stanów przy pewnym kryterium, najczęściej czasowym lub kosztowym.

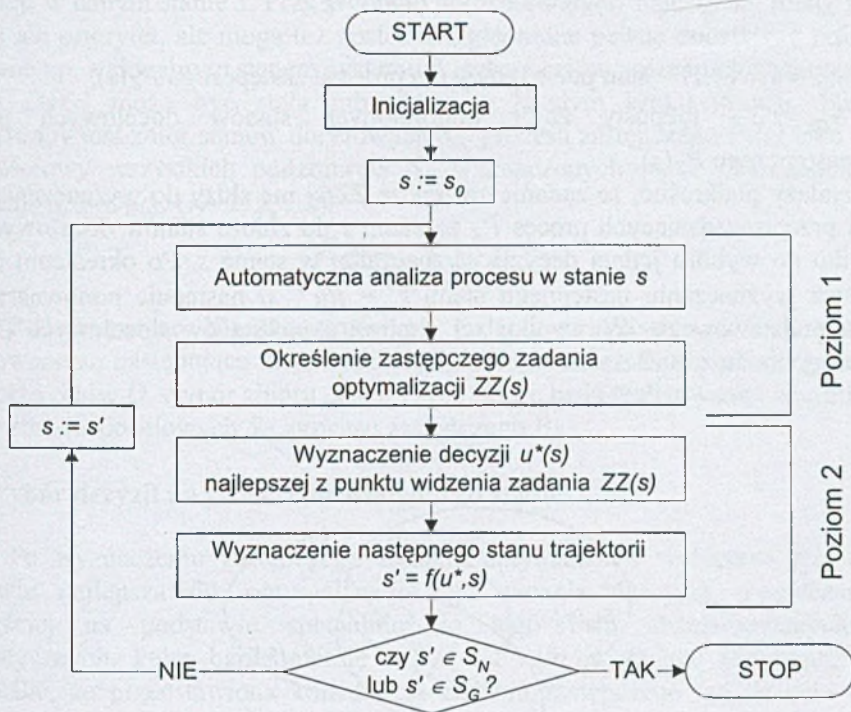
Metoda zadań zastępczych wzorowana jest na sposobie rozwiązywania skomplikowanego zadania przez człowieka. Aby wyznaczyć sekwencje realizowanych w dłuższym horyzoncie czasowym decyzji, w wyniku których ma być osiągnięty jakiś cel, w sytuacji, gdy trudno jest przeanalizować dalekosiężne skutki podjętych decyzji, człowiek radzi sobie najczęściej w następujący sposób. Zamiast rozważać całkowity problem, stara się go zastąpić jednym lub kilkoma relatywnie prostymi zadaniami cząstkowymi (celami pośrednimi), które pozwolą mu podjąć w miarę dobrą decyzję lub wyznaczyć sekwencję kilku decyzji realizowanych w znacznie krótszym horyzoncie czasu. Ponadto, rozwiązując sformułowane zadanie cząstkowe, zdobywa dodatkowe informacje, które stara się wykorzystać. Po zrealizowaniu przyjętej decyzji przeprowadza analizę nowej sytuacji. Na podstawie zdobytych informacji stara się wyznaczyć nowe cząstkowe zadanie lub zmodyfikować (uaktualnić) zadanie postawione wcześniej. W postępowaniu tym można zatem wyróżnić następujące etapy: analiza sytuacji i postawienie zadania cząstkowego oraz wybór i realizacja decyzji, w wyniku których osiągnięta zostaje następna sytuacja. Podczas tego procesu cały czas zdobywane są dodatkowe informacje, które wspomagają podejmowanie kolejnych

decyzji poprzez wyznaczanie następnych celów pośrednich. Dobrą ilustracją przedstawionej koncepcji jest sposób postępowania gracza w grach pozycyjnych, np. w szachach.

Ogólna idea proponowanej metody zadań zastępczych obejmuje wymienione etapy i jest następująca. Podczas generowania rozwiązania (trajektorii), w każdym rozważanym stanie s procesu decyzja wyznaczana jest na podstawie specjalnie skonstruowanego zastępczego zadania optymalizacji $ZZ(s)$. W różnych stanach procesu zadanie to może mieć inną postać. Celem tworzenia takich zadań jest ułatwienie wyznaczenia decyzji w danym stanie poprzez zastąpienie optymalizacji zadania globalnego prostszym zadaniem lokalnym.

Po wyznaczeniu decyzji $u^*(s)$, najlepszej z punktu widzenia zadania zastępczego, generowany jest kolejny stan trajektorii s' . Następnie przeprowadzana jest „automatyczna analiza” nowego stanu procesu, w wyniku której określone jest nowe, zmodyfikowane zadanie zastępcze. Tak więc w każdej iteracji obliczenia prowadzone są na dwóch poziomach:

1. Poziom automatycznej analizy procesu i konstruowania zadania zastępczego.
2. Poziom wyznaczania możliwie dobrej decyzji dla zrealizowania zadania zastępczego i obliczania następnego stanu.



Rys. 1. Schemat metody zadań zastępczych

Schemat metody zadań zastępczych prezentuje rysunek 1. Przedstawione na nim etapy wykonywane są dla każdego stanu systemu, który wystąpił w trakcie

generowania trajektorii. Generowanie trajektorii przerywane jest, gdy osiągnięty zostanie stan niedopuszczalny lub docelowy.

Jako zadanie zastępcze $ZZ(s)$ rozumiana jest para (P_Z, Q_Z) , gdzie P_Z oznacza pewien proces zastępczy, a Q_Z – kryterium zastępcze. Dla podkreślenia, że proces zastępczy konstruowany jest dla stanu s , używany jest zapis $P_Z(s)$.

Przedstawiona w niniejszym artykule konstrukcja zadania zastępczego opiera się na koncepcji celów pośrednich.

Definicja 2. *Celem pośrednim*, oznaczanym jako d , jest osiągnięcie przez proces pewnego wyróżnionego zbioru stanów S_d przy ustalonym kryterium.

W problemach szeregowania najczęściej podzbiór stanów związanych z celem pośrednim zawiera takie stany procesu, w których wykonane jest wyróżnione zadanie lub jedno z kilku wyróżnionych zadań.

Cele pośrednie wykorzystywane są następnie do zdefiniowania zbioru stanów docelowych procesu zastępczego. W procesie zastępczym określony jest zatem nowy zbiór stanów docelowych, stan początkowy jest aktualnym stanem s procesu podstawowego (wyjściowego) P , natomiast funkcja przejścia i zbiór stanów S_N są takie same jak dla procesu P . W związku z tym proces zastępczy definiowany jest następująco:

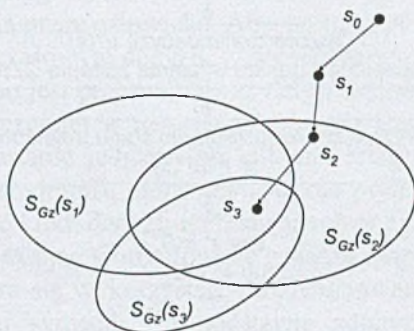
$$P_Z(s) = (s_{0z}, f, S_N, S_{Gz})$$

gdzie:

$s_{0z} = s = (x, t)$ – stan początkowy dla procesu zastępczego $P_Z(s)$,

$S_{Gz} \subset S$ – niepusty zbiór uogólnionych stanów docelowych procesu zastępczego $P_Z(s)$.

Należy podkreślić, że zadanie zastępcze $ZZ(s)$ nie służy do wyznaczenia ciągu decyzji przeprowadzających proces P_Z ze stanu s do zbioru stanów docelowych S_{Gz} , lecz tylko do wyboru jednej decyzji wyznaczanej w stanie s . Po określeniu decyzji $u^*(s)$ oraz wyznaczeniu następnego stanu $s' = f(u^*, s)$ następuje ponowna analiza procesu podstawowego. W wyniku tej analizy zbiór stanów docelowych procesu zastępczego może zostać zmieniony lub nie.



Rys. 2. Przykładowy schemat osiągnięcia przez trajektorię procesu zbiorów S_{Gz}

Poniżej przedstawione zostaną poszczególne etapy metody zadań zastępczych.

2.1. Automatyczna analiza i utworzenie zastępczego zadania optymalizacji

W pierwszym etapie metody analizowany jest aktualny stan systemu, ograniczenia oraz kryterium. Na tej podstawie określone są, w sposób heurystyczny, cele pośrednie d . Ich zbiór oznaczany jest jako D . Reguły wyboru celów oraz utworzone na ich podstawie procedury silnie zależą od konkretnego problemu optymalizacji. Cele te mogą być też określone przez eksperta dla konkretnej instancji problemu. Po ustaleniu celów pośrednich należy określić najbardziej pożądaną kolejność, w jakiej trajektoria powinna przechodzić przez podzbiory stanów związane z tymi celami. Jest to równoznaczne z wprowadzeniem pewnej hierarchii ważności tych celów, co ustalane jest za pomocą nadawania priorytetów poszczególnym celom pośrednim. Dzięki formalnemu przedstawieniu problemu w postaci modelu algebraiczno-logicznego proces analizy można zalgorytmizować.

Jak powiedziano, zadanie zastępcze ZZ określone jest poprzez proces zastępczy P_Z oraz kryterium zastępcze Q_Z . Dla zdefiniowania procesu zastępczego konieczne jest określenie nowego zbioru stanów docelowych S_{G_z} . Zbiór ten definiowany jest na podstawie omówionych wcześniej celów pośrednich. Pierwszym krokiem jest wyznaczenie ze zbioru D jego podzbioru $D_W(s)$, jako podzbioru celów wybranych do realizacji w danym stanie s . Przy wyborze podzbioru $D_W(s)$ najczęściej brany jest pod uwagę ich priorytet, ale mogą też zostać uwzględnione pewne dodatkowe informacje związane np. z aktualnym stanem systemu. Liczba k celów pośrednich wybieranych do zbioru $D_W(s)$ może być stała lub różna w każdym kroku iteracji. Następnie definiowany jest zbiór stanów docelowych S_{G_z} procesu zastępczego $P_Z(s)$ jako iloczyn mnogościowy wszystkich podzbiorów S_{d_i} wyznaczonych przez poszczególne cele pośrednie $d_i \in D_W(s)$ ($i = 1, 2, \dots, k$).

$$S_{G_z} = \bigcap_{d_i \in D_W(s)} S_{d_i}$$

Podsumowując, dla skonstruowania zadania zastępczego w danym stanie s realizowane są następujące kroki: określenie zbioru celów D , określenie priorytetów w zbiorze celów D , wybór zbioru celów $D_W(s)$, które będą realizowane i zdefiniowanie zbioru stanów docelowych S_{G_z} procesu zastępczego P_Z .

2.2. Wybór decyzji i wyznaczenie następnego stanu

Po wyznaczeniu zastępczego zadania optymalizacji wybierana jest decyzja możliwie najlepsza dla optymalizacji tego zadania. Decyzja wyznaczana jest najczęściej na podstawie specjalnie do tego celu skonstruowanych reguł heurystycznych, które bardzo silnie zależą od rodzaju danego problemu. Należy podkreślić, że przedstawiona konstrukcja zadania zastępczego ma na celu właśnie ułatwienie wyznaczenia takiej decyzji.

Po ustaleniu decyzji obliczany jest następny stan przy użyciu funkcji przejścia $f: U \times S \rightarrow S$. Następnie sprawdzane jest, czy stan ten należy do zbioru stanów docelowych S_G lub niedopuszczalnych S_N . Jeżeli tak, to obliczenia zostają zakończone i otrzymywany jest wynik, czyli wygenerowana trajektoria (odpowiednio dopuszczalna lub niedopuszczalna). Jeżeli stan nie jest ani niedopuszczalny, ani

docelowy, to wykonywana jest następna iteracja metody, która rozpoczyna się od etapu przeprowadzania analizy procesu w tym nowo wygenerowanym stanie.

3. Problem udostępniania pól eksploatacyjnych

Przedstawiona metoda została zastosowana do rozwiązania NP-trudnego problemu szeregowania związanego z udostępnianiem pól eksploatacyjnych, a nazwanego problemem UPE. Na podstawie metody stworzony został szereg algorytmów optymalizujących problem UPE, przedstawionych w [12]. Algorytmy te wyznaczają jedną trajektorię. Jeśli trajektoria otrzymana za pomocą wybranego algorytmu jest niedopuszczalna, to stosuje się inny algorytm z tej klasy. W artykule zaprezentowany zostanie jeden ze stworzonych algorytmów.

Problem UPE polega na realizacji prac przygotowawczych w kopalniach podziemnych. Prace te wykonywane są bezpośrednio przed przystąpieniem do wydobywania surowców, jako jeden z etapów działalności górniczej. Aby odpowiednio przygotować złoża do eksploatacji, drażona jest w nim sieć wyrobisk korytarzowych (chodników). Ich zadaniem jest udostępnienie złoża i jednocześnie podzielenie go na mniejsze części, zwane parcelami (polami). Ze względu na terminy oddawania pól do eksploatacji, dla niektórych chodników określone są pewne nieprzekraczalne terminy, do których muszą być one wydrążone. Drażenie wyrobisk korytarzowych może odbywać się za pomocą dwóch technologii o różnicowanej wydajności i kosztach eksploatacji. Technologie te można rozpatrywać jako maszyny należące do jednego z dwóch typów. Maszyny typu I mają większą prędkość drażenia, ale też wyższe koszty eksploatacji. Ponadto, po skończeniu drażenia w jednym chodniku, maszyny często wymagają przetransportowania w inne miejsce, skąd mogą rozpocząć drażenie następnego. Transport tych maszyn jest drogi oraz czasochłonny. Transport taki może odbywać się wyłącznie chodnikami już wydrążonymi. W związku z tym droga i czas transportu maszyny zależy od aktualnego stanu wydrążenia całej sieci. Natomiast maszyny typu II pracują wolniej, ale koszt ich eksploatacji jest mniejszy.

Fakt konieczności transportu maszyn istotnie zwiększa stopień skomplikowania problemu. Celem planistów jest opracowanie harmonogramu wykonywania prac przygotowawczych, czyli określenie kolejności drażenia poszczególnych wyrobisk korytarzowych oraz przydzielenie do nich odpowiednich technologii (maszyn). Harmonogram powinien być tak utworzony, aby nie przekroczyć określonych wcześniej terminów krytycznych dla chodników. Równocześnie harmonogram ten powinien być zrealizowany przy minimalnym koszcie całkowitym prac. Na koszt składają się koszty drażenia, transportu oraz postoju poszczególnych maszyn.

Dla problemu UPE stan procesu w danej chwili t opisany jest poprzez aktualny stan wszystkich maszyn oraz zbiór ukończonych zadań (chodników) [12], [13].

$$x = (x^0, x^1, x^2, \dots, x^{|M|})$$

gdzie: x^0 – zbiór chodników które zostały wykonane do chwili t , x^m – stan m -tej maszyny dla $m=1, 2, \dots, |M|$, M – zbiór maszyn.

Najważniejsze informacje, opisujące stan maszyny, to działanie jakie maszyna wykonuje (drażenie, transport, postój), miejsce, w którym maszyna się znajduje oraz ilość czasu, jaka pozostała maszynie do ukończenia aktualnego zadania.

Decyzje polegają na wyznaczaniu chodników, które mają być przydzielone poszczególnym maszynom do drażenia oraz na określeniu ewentualnych dróg transportowych dla tych maszyn. Struktura decyzji u definiowana jest jako $|M|$ -elementowy wektor:

$$u = (u^1, u^2, \dots, u^{|M|})$$

gdzie kolejne współrzędne oznaczają odrębne decyzje dla poszczególnych maszyn m . Model uwzględnia również możliwość oczekiwania transportowanej maszyny na ukończenie pewnych chodników, a tym samym na pojawienie się nowych, korzystniejszych dróg transportowych.

Na podstawie aktualnego stanu $s = (x, t)$ i decyzji u podjętej w tym stanie, generowany jest przy użyciu funkcji przejścia f następny stan $s' = (x', t')$.

4. Algorytm dla problemu UPE

W problemie UPE analiza procesu dokonywana przez algorytm w danej iteracji polega na sprawdzeniu informacji zawartych we współrzędnych stanu (w szczególności we współrzędnej x^0) oraz związanych ze strukturą sieci wyrobisk, po czym na tej podstawie wyznaczeniu celów pośrednich i ich priorytetów. Cele pośrednie określone zostały jako najszybsze ukończenie pewnych wyróżnionych chodników oznaczanych c_d . Są to te chodniki, dla których określony został termin krytyczny. Dla uzyskania rozwiązania dopuszczalnego konieczne jest bowiem zagwarantowanie, że nie zostaną przekroczone ograniczenia czasowe zdefiniowane dla problemu. Równocześnie, im wcześniej wydrażone zostaną wszystkie chodniki z terminami krytycznymi, tym wcześniej będzie można zrezygnować z szybszych, ale droższych maszyn. Pozostałe chodniki mogą być wtedy drażone za pomocą najtańszej maszyny, co zmniejsza koszt wydrażenia całej sieci. Kryterium zastępcze implikuje ponadto wybór optymalnych dróg transportowych.

Cel pośredni d zdefiniowany został zatem jako osiągnięcie jak najszybciej podzbioru stanów $S_d \subset S$, w którym ukończony jest chodnik c_d , określonego następująco:

$$S_d = \{s = (x, t) : c_d \in x^0(s)\}, \text{ gdzie } x^0(s) \text{ to wartość współrzędnej } x^0 \text{ w stanie } s.$$

Algorytm optymalizacji problemu UPE, oparty na prezentowanym podejściu, jest następujący (ze względu na ograniczoną ilość miejsca w opisie pominięto istotne szczegóły).

Krok 1. Ustalenie początkowego stanu procesu: $s := s_0$

Krok 2. Wyznaczenie celów pośrednich

W danej sieci chodników wyszukiwane są chodniki c_d , dla których został określony termin krytyczny. Z każdym z nich związany jest jeden cel pośredni d . Ilość celów pośrednich w zbiorze D jest zatem równa ilości chodników z terminami krytycznymi.

Krok 3. Wyznaczenie priorytetów celów pośrednich

Każdemu celowi d nadawany jest priorytet $p(d)$ zależny od oszacowanej rezerwy czasowej dla chodnika c_d , związanego z tym celem. Wykorzystuje się górne oszacowanie rezerwy czasowej, zakładając, że do wydrażenia tego chodnika i ewentualnej drogi prowadzącej do niego użyta zostanie maszyna z największą prędkością drażenia. Im rezerwa czasowa dla danego chodnika jest mniejsza, tym wyższy jest priorytet związanego z nim celu.

Krok 4. Wyznaczenie zbioru $D_W(s)$ celów pośrednich wybranych do realizacji. W zbiorze tym umieszczanych jest $|M|$ celów pośrednich o najwyższych priorytetach (gdzie $|M|$ to ilość maszyn w systemie). Jeżeli zbiór D jest mniej liczny niż zbiór M , to w zbiorze $D_W(s)$ znajdzie się $|D|$ celów pośrednich.

Krok 5. Wyznaczanie decyzji dla danego stanu

Aby wyznaczyć decyzję $u^*(s)$ w danym stanie s należy wyznaczyć wszystkie współrzędne tej decyzji $u(s) = (u^1, u^2, \dots, u^{|M|})$. Zadanie jest częściowo zdekomponowane poprzez określenie celów pośrednich $d \in D_W(s)$. Można zatem związać realizację poszczególnych celów z kolejnymi współrzędnymi decyzji $u(s) = (u^1, u^2, \dots, u^{|M|})$, czyli dedykować realizację tych celów poszczególnym maszynom m . Realizacja celu polega tu na wykonaniu wyróżnionego chodnika c_d . Aby to osiągnąć, należy najpierw wydrążyć – o ile jeszcze w danym stanie nie istnieje – drogę do niego prowadzącą. Decyzją dla rozpatrywanej maszyny będzie zatem albo drażenie chodnika c_d , albo pierwszego chodnika na tej drodze. Dodatkowo, jeżeli istnieje konieczność transportu maszyny do tego wyznaczonego chodnika, obliczana jest optymalna droga transportowa. Uwzględniane są przy tym chodniki wydrążone oraz te, których ukończenie przewidziane jest w najbliższym czasie.

Krok 6. Obliczenie następnego stanu

Mając wyznaczoną decyzję $u^*(s)$ obliczany jest następny stan procesu s' za pomocą funkcji przejścia: $s' = f(u^*, s)$

Krok 7. Sprawdzenie warunków stopu

Jeżeli stan s' należy do stanów docelowych lub niedopuszczalnych, to generowanie trajektorii zostaje zakończone i wypisywany jest odpowiedni wynik. W przeciwnym przypadku należy przejść do kolejnego kroku.

Krok 8. Aktualizacja zbioru D

Po wyznaczeniu następnego stanu aktualizowany jest zbiór D celów pośrednich. Usuwany jest z niego każdy taki cel d , dla którego odpowiadający mu chodnik c_d został wykonany.

Krok 9. Inicjowanie następnej iteracji algorytmu

Należy podstawić $s := s'$, a następnie należy przejść do kroku 3, czyli przeprowadzić aktualizację priorytetów celów pośrednich.

5. Eksperymenty

W ramach eksperymentów zaproponowany algorytm został przebadany na zestawie sieci testowych o liczbie krawędzi (reprezentujących chodniki) od 20 do 27. Obliczenia prowadzone były dla przypadku z dwoma maszynami. Prezentowany algorytm dla większości sieci wygenerował trajektorie dopuszczalne, przy czym czas obliczeń dla każdej nie przekraczał 1s (na komputerach klasy Pentium IV).

Aby ocenić efektywność algorytmu, dla sieci testowych, zastosowano przegląd zupełny. Wyniki przedstawione są w tabeli 1. Otrzymane rezultaty pozwalają stwierdzić, że zaproponowane algorytmy znajdują rozwiązanie z dość dobrym przybliżeniem przy bardzo krótkim czasie obliczeń.

Tabela 1

Porównanie rezultatów uzyskanych przez algorytm oparty na MZZ oraz algorytm przeglądu (*- najlepszy wynik otrzymany do momentu przerwania obliczeń)

Sieć	Koszt wydrążenia sieci		Błąd	Czas obliczeń algorytmu przeglądu
	Algorytm oparty na MZZ	Algorytm przeglądu		
A	14434,50	13880*	3,99%	30h
B	14419,20	13942,80	3,42%	20h 1min 49s
C	15202,30	14777,20	2,88%	29h 1min 55s
D	20030,90	18971,59*	5,58%	30h
E	26844,80	25880,88*	3,72%	30h

LITERATURA

1. Dudek-Dyduch E.: Formalizacja i analiza problematyki dyskretnych procesów produkcyjnych. Zeszyty Nauk. AGH, seria Automatyka, z. 54, Kraków 1990.
2. Dudek-Dyduch E.: Dolne ograniczenia algorytmów branch & bound w sterowaniu DPP. Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka, z. 109, 1992, s. 47–56.
3. Dudek-Dyduch E.: Optimization of discrete manufacturing processes - branch and bound method. Operations Research' 92, Springer-Verlag, Heidelberg 1992, p. 15–22.
4. Dudek-Dyduch E.: Control of discrete event processes - branch and bound method. Prepr. of IFAC/Ifors/Imacs Symposium Large Scale Systems: Theory and Applications, Vol. 2, Chinese Association of Automation, 1992, p. 573–578.
5. Dudek-Dyduch E., Fuchs-Seliger S.: Approximate algorithms for some tasks in management and economy. System, Modelling, Control, No. 7, Vol.1, 1993, p. 148–152.
6. Dudek-Dyduch E., Dyduch T.: Formal approach to optimization of discrete manufacturing processes. Proc. of the Twelfth IASTED Int. Conference Modelling, Identification and Control, Acta Press, Zurich 1993, p. 467–468.
7. Dudek-Dyduch E.: Learning based algorithm in scheduling. Journal of Intelligent Manufacturing (JIM), Vol. 11, No. 2, Kluwer Academic Publisher, 2000, p. 135–143.
8. Dudek-Dyduch E.: Systemy informacyjne zarządzania produkcją. Wydawnictwo Poldex, Kraków 2002.
9. Dudek-Dyduch E., Dutkiewicz L., Kucharska E.: Model algebraiczno-logiczny szeregowania zadań z uwzględnieniem transportu maszyn. Automatyka: półrocznik Akademii Górniczo-Hutniczej, t.8, z.3, Kraków 2004, s. 553–562.
10. Dudek-Dyduch E., Dutkiewicz L., Kucharska E.: Formalny model symulacji procesów decyzyjnych jako model algebraiczno-logiczny. XI Warsztaty Naukowe PTSK: „Symulacja w badaniach i rozwoju”, Polskie Towarzystwo Symulacji Komputerowej, Warszawa 2005, s. 394–402.

11. Dudek-Dyduch E., Dyduch T.: Learning algorithms for scheduling using knowledge based model. Lecture Notes in Artificial Intelligence, Vol. 4029, Springer-Verlag 2006 (przyjęte do druku).
12. Dutkiewicz L.: Dwupoziomowe algorytmy optymalizacji procesów wytwarzania z zasobami zależnymi od stanu systemu. Praca doktorska, Promotor: prof. dr hab. inż. Ewa Dudek-Dyduch, Kraków 2005.
13. Dutkiewicz L., Kucharska E.: Model dla problemu szeregowania zadań z zasobami zależnymi od stanu systemu. Automatyka: półrocznik Akademii Górniczo-Hutniczej, t. 9, z. 1-2, Kraków 2005, s. 67-77.
14. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999.

Recenzent: Prof. dr hab. inż. Jan Węglarz

Abstract

The aim of the paper is twofold. The first one is to present a conception of a new approximate optimization method, named a substitution task method (STM). It is an artificial intelligent technique for solving NP-hard scheduling problems. The second aim is to present an algorithm based on this method, applied for NP-hard problem of preparing access to exploitation fields. The main idea of the proposed method resembles a way in which human decision-makers deal with complex problems. When they have to make a sequence of decisions in a long time horizon, they usually try to decompose the problem and replace it by simpler partial tasks.

The substitution task method works as follows. First, a thorough analysis of the process in a present state is made. According to gained information a special optimization task is created, named a substitution task. It replaces the original task in order to simplify the choice of the decision. When the possibly best decision for the task is taken, the next process state is calculated. An analysis is made for this state and new or modified substitution task is defined. Substitution tasks are created on a basis of so-called intermediate goals.

Results of experiments for problem of preparing access to exploitation fields are also presented. The results confirm efficiency of the proposed method.