ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 134 Nr kol. 1554

Piotr FORMANOWICZ
Politechnika Poznańska
Instytut Chemii Bioorganicznej Polskiej Akademii Nauk

ALGORYTMY TABU SEARCH DLA PROBLEMÓW SZEREGOWANIA ZADAŃ NA JEDNEJ MASZYNIE Z OGRANICZONĄ DOSTĘPNOŚCIĄ

Streszczenie. W pracy zaprezentowano algorytmy oparte na metodzie tabu dla problemów szeregowania zadań niepodzielnych na jednej maszynie z okresami niedostępności i kryterium minimalizacji długości uszeregowania, sumy czasów zakończenia zadań oraz maksymalnego opóźnienia. Jakość zaprezentowanych algorytmów przebadano w obszernym eksperymencie obliczeniowym.

TABU SEARCH ALGORITHMS FOR SCHEDULING TASKS ON A SINGLE MACHINE WITH LIMITED AVAILABILITY

Summary. In the paper algorithms based on tabu search method for scheduling non-preemptable tasks on a single machine with limited availability have been presented. The criterion functions to be minimized are the makespan, the total completion time, and the maximum lateness. The quality of the algorithms has been examined in an extensive computational experiment.

1. Wprowadzenie

W deterministycznej teorii szeregowania zadań na ogół przyjmuje się, że maszyny dostępne są w sposób ciągły. Założenie to, mimo iż powszechnie akceptowane, nie zawsze jest dobrze uzasadnione z praktycznego punktu widzenia. W realnym świecie często się zdarza bowiem, że maszyny są w pewnych okresach niedostępne dla przetwarzania zadań. Okresy te mogą być bądź nie być znane z góry. W tym drugim przypadku mogą one wynikać np. z awarii maszyn. W pracy zajmujemy się modelami, w których okresy takie znane są z góry. Mogą być one rezultatem m. in. realizacji zaplanowanych wcześniej prac konserwacyjnych, stosowania algorytmów planowania z przesuwanym horyzontem

czasowym, mogą się też pojawiać w systemie komputerowym w przypadku wykonywania zadań o różnych priorytetach.

Algorytmy planowania z przesuwanym horyzontem czasowym stosowane są stosunkowo często ze względu na fakt, że większość problemów planowania produkcji ma naturę dynamiczną. Oznacza to, że dane wejściowe dla takich problemów często są aktualizowane w trakcie trwania procesu produkcyjnego. Naturalnym podejściem do rozwiązywania tego typu problemów jest ustawienie nowego horyzontu czasowego, gdy jest to uzasadnione zmianą danych wejściowych. Jednak ze względu na występowanie różnorodnych ograniczeń, m. in. technologicznych, nierzadko konieczne jest przyjęcie pewnej części dotychczas ustalonego planu za niezmienną, co oczywiście ogranicza dostępność maszyn dla szeregowania kolejnych zadań.

Z kolei w systemie komputerowym, w którym szeregowane są dwie grupy zadań - o niskim i wysokim priorytecie, takich że te ostatnie mają określone momenty gotowości oraz linie krytyczne, konieczne jest uszeregowanie w pierwszej kolejności zadań z drugiej grupy. Ponownie, ogranicza to dostępność procesorów dla szeregowania zadań o niskim priorytecie. Procedura szeregująca "postrzega" okresy, w których procesory zostały przydzielone do zadań o priorytecie wysokim jako okresy ich niedostępności.

Problemy szeregowania zadań z ograniczoną dostępnością maszyn od pewnego czasu coraz bardziej przyciągają uwagę badaczy. Literatura dotycząca tego nurtu teorii szeregowania nie jest jeszcze zbyt obszerna, tym niemniej istnieją już ciekawe prace poświęcone tego typu zagadnieniom. Przegląd takich prac można znaleźć m. in. w [6] oraz [3].

Niniejsza praca dotyczy jednomaszynowych problemów szeregowania zadań z ograniczoną dostępnością. Dotychczas opublikowanych zostało zaledwie kilka prac dotyczących tego typu zagadnień. W [1] zostało udowodnione, że problem szeregowania zadań niepodzielnych na maszynie z jednym okresem niedostępności w celu minimalizacji sumy czasów zakończenia jest NP-trudny. W pracy [5] podany został prostszy dowód tego faktu oraz pokazane zostało, że algorytm SPT zastosowany do rozwiązania tego problemu daje w najgorszym przypadku rozwiązanie z błędem względnym nie większym niż $\frac{9}{7}$. W pracy [4] pokazane zostało, że problemy szeregowania zadań niepodzielnych na maszynie z jednym okresem niedostępności są w ogólności NP-trudne i stają się silnie NP-trudne, gdy okresów takich jest dowolna liczba. W tym samym artykule podane zostały również osza-

cowania błędów względnych algorytmów LPT, EDD oraz algorytmu Moore'a-Hodgsona zastosowanych do rozwiązania, odpowiednio, problemów minimalizacji długości uszeregowania, maksymalnego opóźnienia oraz liczby spóźnionych zadań. Ponadto w pracy tej został wprowadzony nowy typ zadań, charakterystyczny dla problemów z ograniczoną dostępnością - zadań *wznawialnych*. Wykonywanie zadań tego rodzaju może zostać przerwane wyłącznie ze względu na rozpoczęcie okresu niedostępności maszyny, a po jego zakończeniu jest ono kontynuowane bez dodatkowego kosztu (na tej samej maszynie - zadania tego typu mogą być oczywiście wykonywane również w systemach wielomaszynowych, stąd uwaga ta jest konieczna). Autor cytowanej pracy pokazał, że niektóre problemy, trudne obliczeniowo w przypadku zadań niepodzielnych, stają się łatwe w przypadku zadań wznawialnych. Niektóre z wyników przedstawionych w [4] zostały rozszerzone na przypadki problemów z wieloma okresami niedostępności maszyny w pracach [2] oraz [3].

Problemy szeregowania zadań, w których uwzględniona jest okresowa niedostępność maszyn, są na ogól trudniejsze (w sensie złożoności obliczeniowej) od swoich klasycznych odpowiedników [3]. Jest tak zwłaszcza w przypadku szeregowania zadań niepodzielnych. Stąd też wynika konieczność opracowywania algorytmów dokładnych wyliczeniowych (wykładniczych), np. typu podziału i ograniczeń, bądź algorytmów heurystycznych (wielomianowych), w szczególności metaheurystycznych.

W pracy przedstawione są algorytmy oparte na metodzie tabu dla problemów szeregowania zadań niepodzielnych na jednej maszynie ze znanymi z góry okresami niedostępności. Rozważane są trzy kryteria optymalności uszeregowań: długość uszeregowania, suma czasów wykonania zadań oraz maksymalne opóźnienie. Rozdział 2. zawiera sformułowanie problemów. W rozdziale 3. opisane są zaproponowane algorytmy, natomiast w rozdziale 4. zaprezentowane są wyniki eksperymentów obliczeniowych. Rozdział 5. stanowi podsumowanie.

2. Sformułowanie problemów

Dana jest jedna maszyna oraz zbiór zadań $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$. Dla każdego zadania T_j, $j = 1, 2, \ldots, n$ określony jest czas przetwarzania oraz być może żądany termin zakończenia d_j. Maszyna może wykonywać co najwyżej jedno zadanie w danym momen-

cie. Ponadto jest ona niedostępna w pewnych przedziałach czasu. Oznaczmy przez s_k oraz h_k, odpowiednio, moment rozpoczęcia oraz długość k-tego okresu niedostępności. Niech ponadto K będzie liczbą takich okresów. Wykonywanie żadnego z zadań, jeżeli zostało rozpoczęte, nie może zostać przerwane przed zakończeniem.

Problem minimalizacji długości uszeregowania można sformułować w postaci następującego zagadnienia programowania matematycznego [2, 3]:

$$\text{zminimalizować } C_{max} \tag{1}$$

przy ograniczeniach:

$$\sum_{l=1}^{n} z_{jl} = 1, \quad j = 1, \ldots, n \tag{2}$$

$$\sum_{j=1}^{n} z_{jl} = 1, \quad l = 1, \ldots, n \tag{3}$$

$$z_{jl} \in \{0, 1\}, \quad j = 1, \ldots, n; \; l = 1, \ldots, n \tag{4}$$

$$t_1 \geq 0 \tag{5}$$

$$t_l + \sum_{j=1}^{n} p_j z_{jl} \leq s_k \vee t_l \geq s_k + h_k; \quad l = 1, \ldots, n; \; k = 1, \ldots, K \tag{6}$$

$$t_{l+1} \geq t_l + \sum_{j=1}^{n} p_j z_{jl}; \quad l = 1, \ldots, n-1 \tag{7}$$

$$C_{max} = \max_{l=1,\ldots,n} \left\{ t_l + \sum_{j=1}^{n} p_j z_{jl} \right\} \tag{8}$$

gdzie:

$$z_{jl} = \begin{cases} 1, & \text{jeżeli zadanie } T_j \text{ znajduje się na } l\text{-tej pozycji w uszeregowaniu} \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

t_l - moment rozpoczęcia wykonywania zadania na pozycji l.

Równości (2), (3) oraz (4) przyporządkowują zadania i pozycje w uszeregowaniu do siebie nawzajem. Dzięki nierówności (5) moment rozpoczęcia zadania znajdującego się na pierwszej pozycji w uszeregowaniu nie może być ujemny. Nierówności (6) gwarantują, że żadne z zadań nie będzie wykonywane w czasie niedostępności maszyny, z kolei dzięki nierównościom (7) zadania nie nakładają się na siebie w uszeregowaniu. Równość (8) stanowi określenie wartości C_{max}.

Jeżeli w zagadnieniu (1)-(8) zamienimy (8) na

$$L_{max} = \max_{l=1,\dots,n} \left\{ t_l + \sum_{j=1}^{n} p_j z_{jl} - \sum_{j=1}^{n} d_j z_{jl} \right\}, \tag{9}$$

otrzymamy sformułowanie problemu minimalizacji maksymalnego opóźnienia [2, 3].

Jeżeli natomiast (8) zamienimy na

$$C = \sum_{l=1}^{n} \left(t_l + \sum_{j=1}^{n} p_j z_{jl} \right), \tag{10}$$

otrzymamy zagadnienie programowania matematycznego dla problemu minimalizacji sumy czasów zakończenia zadań [2, 3].

3. Algorytmy

Jak już zostało powiedziane we Wprowadzeniu, problemy szeregowania zadań niepodzielnych na maszynie z okresami niedostępności są obliczeniowo trudne. W związku z tym, w niniejszej pracy zaproponowano algorytmy heurystyczne oparte na metodzie tabu search dla problemów minimalizacji długości uszeregowania, sumy czasów zakończenia zadań oraz maksymalnego opóźnienia.

Każdy z zaproponowanych algorytmów rozpoczyna przeszukiwanie przestrzeni rozwiązań od sekwencji zadań zgodnej z tą, w jakiej wystąpiły one w instancji problemu. Sąsiedztwo danego rozwiązania określone jest jako zbiór tych wszystkich rozwiązań, które osiągalne są z niego przez zamianę pozycji w uszeregowaniu dowolnej pary zadań.

W każdym kroku zaproponowanych algorytmów oceniana jest wartość odpowiedniej funkcji celu (C_{max}, $\sum C_j$ lub L_{max}) dla każdego z rozwiązań należących do sąsiedztwa bieżącego rozwiązania. Z sąsiedztwa jako kolejne bieżące rozwiązanie wybierane jest to, które posiada najmniejszą wartość funkcji celu, natomiast ruch, którego wykonanie doprowadziło do przejścia do wspomnianego nowego rozwiązania, wpisywany jest na listę tabu. Oczywiście, ruchy znajdujące się na tej liście są zabronione i można je wykonać tylko w sytuacji, gdy prowadzą do osiągnięcia rozwiązania o najlepszej z dotychczas osiągniętych wartości funkcji celu (kryterium aspiracji). Jeżeli algorytm w pewnym ciągu kroków, którego długość jest parametrem algorytmu, nie uzyskuje nowego rozwiązania o wartości

funkcji celu mniejszej niż wartość poprzednia, wykonywana jest seria ruchów losowych (której długość również jest parametrem algorytmu). Po wykonaniu określonej z góry liczby cykli ruchy deterministyczne – ruchy losowe, algorytmy zatrzymują się podając jako rozwiązanie najlepsze z uszeregowań znalezionych w trakcie działania metody.

4. Wyniki eksperymentu obliczeniowego

Zaproponowane algorytmy zostały przetestowane w obszernym eksperymencie obliczeniowym. Wygenerowane zostały dwa zbiory instancji. Zbiór $\mathcal{A}$ zawiera instancje przeznaczone do testowania algorytmów minimalizujących długość uszeregowania oraz sumę czasów zakończenia, natomiast zbiór $\mathcal{B}$ - instancje przeznaczone do testowania algorytmu minimalizującego maksymalne opóźnienie. Sposób generowania instancji był następujący. Liczba okresów niedostępności K przyjmowała wartości $2, 4, 6, 8, 10$, natomiast liczba zadań n wartości $10, 20, 30, 40, 50$. Dla każdej pary K i n wygenerowanych zostało 5 instancji w każdym ze zbiorów. Długości okresów niedostępności oraz zadań losowane były z przedziału $[1, 100]$. Oprócz czasów wykonywania zadań oraz długości okresów niedostępności instancje zawierały również momenty rozpoczęcia tych okresów, a instancje ze zbioru $\mathcal{B}$ również momenty gotowości zadań.

Jakość rozwiązań otrzymanych za pomocą zaproponowanych algorytmów oceniono porównując je z rozwiązaniami, w których zadania znajdują się w kolejności niemalejących czasów wykonania (SPT), nierosnących czasów wykonania (LPT) oraz w kolejności zgodnej z kolejnością żądanych terminów zakończenia (EDD). Zaprezentowane poniżej wyniki otrzymano za pomocą metod, w których lista tabu miała długość równą 10% liczby szeregowanych zadań, liczba ruchów bez poprawy wartości funkcji celu, po których następowała seria ruchów losowych, wynosiła 10, liczba ruchów losowych w jednej serii wynosiła 5, natomiast liczba serii ruchów losowych, po których algorytmy zatrzymywały się, wynosiła 4.

W tablicach 1 i 2 przedstawiono wyniki dotyczące jakości rozwiązań dostarczanych przez algorytm tabu dla problemu minimalizacji długości uszeregowania. Każda pozycja w tablicy 1 zawiera średnią wartość stosunku (wartość kryterium dla uszeregowania SPT)/(wartość kryterium dla uszeregowania wygenerowanego przez algorytm tabu). Dla każdej z instancji wykorzystanych w eksperymencie obliczeniowym algorytm tabu uru-

chomiony został 5 razy. Ponadto, jak zostało powiedziane wcześniej, dla każdej pary K i n wygenerowanych zostało 5 instancji. Średnia, o której mowa, obliczona jest dla 25 uruchomień algorytmu - 5 uruchomień dla każdej z pięciu instancji dla danej pary K, n. Tablica 2 zawiera analogiczne wyniki porównań z uszeregowaniami LPT.

Tablica 1

Porównanie uszeregowań SPT i tabu - kryterium C_{max}

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	1.10	1.84	1.13	1.35	1.09
20	1.06	1.97	1.87	1.23	1.03
30	1.06	1.75	16.34	1.14	1.07
40	1.04	1.08	1.35	1.87	1.02
50	1.02	1.07	1.38	1.10	2.54

Tablica 2

Porównanie uszeregowań LPT i tabu - kryterium C_{max}

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	1.12	1.13	1.02	1.01	1.06
20	1.07	1.18	1.30	1.30	1.04
30	1.06	1.37	1.50	1.54	1.09
40	1.04	1.10	1.09	2.04	1.05
50	1.03	1.10	1.70	1.15	1.14

Z uzyskanych rezultatów wynika, że uszeregowania SPT oraz LPT są dłuższe niż uszeregowania wygenerowane przez algorytm tabu, przy czym uszeregowania SPT i LPT mają zbliżoną jakość z punktu widzenia kryterium C_{max}.

Tablice 3 oraz 4 zawierają informacje dotyczące jakości uszeregowań znajdowanych przez algorytm tabu minimalizujący sumę czasów zakończenia zadań. Sposób przedstawienia tych informacji jest analogiczny jak w przypadku tablic 1 i 2.

Tablica 3

Porównanie uszeregowań SPT i tabu - kryterium $\sum C_j$

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	1.00	1.03	1.01	1.00	1.01
20	1.03	1.02	1.02	1.03	1.02
30	1.02	1.01	1.02	1.05	1.01
40	1.05	1.02	1.02	1.02	1.02
50	1.01	1.02	1.04	1.02	1.02

Tablica 4

Porównanie uszeregowań LPT i tabu - kryterium $\sum C_j$

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	1.84	1.44	1.51	1.81	1.56
20	1.90	1.51	1.78	1.69	1.94
30	1.65	1.71	1.90	1.76	1.75
40	1.84	1.67	1.60	1.62	1.75
50	1.92	1.91	1.60	1.76	1.76

Wyniki te zgodne są z intuicją. Uszeregowania SPT mają optymalną wartość kryterium $\sum C_j$ w przypadku klasycznym i są często również dobrymi rozwiązaniami suboptymalnymi w przypadku maszyny z ograniczoną dostępnością, co sugerują wyniki z tablicy 3. Jednak algorytm tabu wygenerował również w tym przypadku rozwiązania nieco lepsze.

W tablicach 5, 6 oraz 7 zaprezentowane zostały informacje na temat jakości trzeciego z zaproponowanych algorytmów, tj. algorytmu minimalizującego maksymalne opóźnienie. Tablice te zawierają średnie różnice maksymalnego opóźnienia w uszeregowaniu, odpowiednio, SPT, LPT i EDD i takiego opóźnienia w uszeregowaniu wygenerowanym przez algorytm tabu.

Tablica 5

Porównanie uszeregowań SPT i tabu - kryterium L_{max}

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	163	194	284	306	232
20	604	523	450	580	537
30	807	754	916	899	935
40	1087	1177	1437	1145	1419
50	1560	1359	1413	1366	1618

Tablica 6

Porównanie uszeregowań LPT i tabu - kryterium L_{max}

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	325	384	316	432	301
20	626	805	437	714	628
30	1072	828	1230	1154	1117
40	1325	1258	1491	1650	1667
50	1756	1838	1822	1483	1857

I w tym przypadku uszeregowania optymalne dla klasycznej wersji problemu (EDD) okazują się być dobrymi uszeregowaniami suboptymalnymi dla problemu z ograniczoną

Tablica 7

Porównanie uszeregowań EDD i tabu - kryterium L_{max}

Liczba zadań	Liczba okresów niedostępności				
	2	4	6	8	10
10	0	6	0	0	0
20	8	2	9	23	-4
30	18	-4	7	-26	0
40	-56	-34	-65	-8	-46
50	-178	-8	-51	-116	-21

dostępnością maszyny. Algorytm tabu dał uszeregowania dużo lepsze od uszeregowań SPT i LPT, ale w wielu przypadkach nieco gorsze od uszeregowań EDD.

5. Zakończenie

W pracy przedstawione zostały algorytmy oparte na metaheurystyce tabu search dla problemów szeregowania zadań niepodzielnych na jednej maszynie przy kryterium minimalizacji C_{max}, $\sum C_j$ oraz L_{max}. Wyniki przeprowadzonego eksperymentu obliczeniowego pokazują, ze algorytmy te dają na ogół rozwiązania lepsze niż uszeregowania SPT i LPT. Jednak w przypadku minimalizacji maksymalnego opóźnienia uszeregowania zgodne z kolejnością EDD były często nieco lepsze. Może to swiadczyć o słabości algorytmu tabu, lub o wyjątkowo dobrej jakości uszeregowań EDD, optymalnych w przypadku klasycznym, również w przypadku z ograniczoną dostępnością.

Nie było, niestety, możliwe porównanie jakości uszeregowań znajdowanych przez zaproponowane algorytmy z uszeregowaniami optymalnymi, ani z uszeregowaniami suboptymalnymi generowanymi przez algorytmy zaprojektowane do rozwiązywania rozważanych problemów, gdyż algorytmy takie nie były dotąd publikowane.

Przedstawione algorytmy tabu mogą być punktem wyjściowym do dalszych badań w tej dziedzinie. Jednym z możliwych ulepszeń opisanych metod mogłoby być zachowywanie w trakcie poszukiwania najlepszego rozwiązania tych części uszeregowania, które składają

się z zadań dokładnie wypełniających przestrzeń między dwoma sąsiednimi okresami niedostępności maszyny. W zaprezentowanych wersjach algorytmów takie dobrze dopasowane grupy zadań mogą zostać rozbite w trakcie poszukiwania lepszego uszeregowania.

LITERATURA

1. Adiri I., Bruno J., Frostig E., Rinnoy Kan A. H. G.: Single machine flow-time scheduling with a single breakdown. Acta Informatica, 26, 1989, 679-696.
2. Błażewicz J., Formanowicz P.: Scheduling tasks on a single machine with limited availability, Research report RA-005/2000, Institute of Computing Science, Poznań University of Technology, Poznań 2000.
3. Formanowicz P.: Szeregowanie zadań w systemach z ograniczoną dostępnością procesorów, Rozprawa doktorska, Politechnika Poznańska, Poznań 2000.
4. Lee C.-Y.: Machine scheduling with an availability constraint Journal of Global Optimization, 9, 1996, 395-416.
5. Lee C.-Y., Liman S. D.: Single machine flow-time scheduling with scheduled maintenance Acta Informatica, 29, 1992, 375-382.
6. Schmidt G.: Scheduling with limited machine availability. European Journal of Operational Research, 121, 2000, 1-15.

Recenzent: Dr hab. inż. Ewa Dudek-Dyduch, Prof. AGH

Abstract

In deterministic scheduling theory it is usually assumed that machines are continuously available for processing tasks. This assumption is not well justified from practical point of view. In many situations it may happen that machines are not available for processing in certain periods of time. This limited availability may result from preschedules, preventive maintenance activities or from an application of rolling time horizon planning algorithms. Also in computer systems processors may be not available.

Scheduling problems concerning machines with limited availability are most often computationally harder than their classic, i.e. with continuously available machines, counterparts. This especially takes place in case of non-preemptable tasks. On the other hand, such problems are of great practical importance, since the limited availability has

its origins in real life situations. Hence, development of good polynomial time heuristic algorithms is an important issue.

In the paper tabu search algorithms for scheduling non-preemptable tasks on a single machine with limited availability have been presented. The algorithms are dedicated for problems of minimization the makespan, the total completion time, and the maximum lateness. The quality of the algorithms has been examined in an extensive computational experiment.