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

Aleksander BACHMAN, Adam JANIAK, Andrzej KOZIK, Marcin WINCZASZEK
Politechnika Wrocławska

JEDNOMASZYNOWY PROBLEM SZEREGOWANIA Z POTĘGOWYMI FUNKCJAMI ZMIANY WARTOŚCI ZADAŃ

Streszczenie. W niniejszej pracy rozpatrzono jednomaszynowy problem szeregowania, w którym wartości zadań zależą od momentów zakończenia ich wykonywania i są opisane malejącymi funkcjami potęgowymi. Rozwiązanie problemu polega na znalezieniu takiego uszeregowania, dla którego suma wartości zadań jest maksymalna. Problem sformułowany powyżej znajduje praktyczne zastosowanie np. w procesie odzysku surowców (np. części ze starych komputerów, samochodów), planowaniu sprzedaży, czy też magazynowaniu i transporcie produktów psujących się. Złożoność obliczeniowa rozpatrywanego problemu jest wciąż zagadnieniem otwartym, jednakże w pracy wykazano szereg własności określających optymalne rozwiązanie badanego problemu. Własności te zostały wykorzystane przy konstrukcji algorytmu opartego na metodzie podziału i ograniczeń. Dla skonstruowanego algorytmu przeprowadzono eksperyment obliczeniowy w celu określenia jego efektywności.

SINGLE MACHINE SCHEDULING PROBLEM WITH EXPONENTIALLY DEPENDENT JOB VALUES

Summary. The paper deals with a single machine scheduling problem, where job value is characterized by an exponential function dependent on job completion time. The objective is to find a sequence of jobs for which the sum of job values calculated at their completion times is maximized. The computational complexity status of the considered problem is still an open question, however it seams to be NP-hard. We proved some properties characterizing the optimal solution of the problem stated above. Based on these properties we constructed a branch and bound algorithm which quality was experimentally analyzed.

1. Wprowadzenie

Bardzo szybki rozwój technologii powoduje, że klasyczna teoria szeregowania, w której parametry określające zadanie są z góry zadanymi wielkościami [1], staje się niewystarczająca do opisu wielu rzeczywistych procesów produkcyjnych. Wobec tego zachodzi potrzeba rozszerzenia istniejącej teorii na problemy, w których parametry zadań są pewnymi funkcjami zależnymi np. od ilości dostarczonego zasobu [2], bądź od momentu rozpoczęcia wykonywania zadania [2]. Zastosowanie nowych dokładniejszych modeli charakteryzujących parametry zadań powoduje z jednej strony uzyskiwanie dokładniejszych rozwiązań, z drugiej jednak strony - utrudnia analizę problemu.

Problem rozpatrywany w niniejszej pracy charakteryzuje proces, w którym wartość zadania jest opisana pewną funkcją zależną od momentu zakończenia jego wykonywania. W szczególności, modelowany proces może dotyczyć ustalania planu remontu dróg, zabytków itp. obiektów. Załóżmy, że dana jest pewna instytucja (np. Urząd Gminy), która musi ustalić, w jakiej kolejności remontować obiekty określonego typu (np. zabytki, drogi, mosty itp.), znajdujące się pod jej zarządem. Zadaniem będzie wyremontowanie określonego obiektu. Każdy z tych obiektów w momencie ustalania harmonogramu remontów jest w określonym stanie technicznym. Stan ten wpływa bezpośrednio na wartość rynkową obiektu, która może stanowić wartość początkową zadania. Wiadomo, że nie remontowany obiekt z czasem niszczeje, rosną więc koszty związane z późniejszym remontem takiego obiektu, natomiast wartość rynkowa obiektu spada. Problem polega na ustaleniu takiej kolejności (harmonogramu) remontowania obiektów, która zapewnia, że suma wartości obiektów (liczona po ich wyremontowaniu) jest maksymalna, natomiast koszty związane z remontami tych obiektów są jak najmniejsze. Inny przykład zastosowania rozpatrywanego problemu można znaleźć opisując proces demontażu produktów zaawansowanych technologicznie. Problem badany w niniejszej pracy został sformułowany po raz pierwszy w pracy [3]. Autorzy tej pracy wykazali kilka szczególnych przypadków problemu, które można rozwiązać optymalnie w wielomianowym czasie oraz zaproponowali algorytmy rozwiązujące badany problem w sposób przybliżony. Praca [3] jest, o ile nam wiadomo, jedyną pozycją poświęconą problemowi badanemu w niniejszym artykule.

Pozostała część pracy została zorganizowana następująco. Rozdział 2 zawiera dokładne sformułowanie problemu. W rozdziale 3 wykazano trzy własności eliminacyjne problemu, które zostały wykorzystane w algorytmach opartych na metodzie podziału i oszacowań skonstruowanych w celu rozwiązania rozpatrywanego problemu. W rozdziale 4 przedstawiono wyniki analizy eksperymentalnej rozwiązań dostarczanych przez zaproponowane algorytmy. Rozdział 5 zawiera krótkie podsumowanie.

2. Sformułowanie problemu

Dany jest zbiór $J = \{1,...,n\}$ zawierający n niezależnych i niepodzielnych zadań dostępnych do realizacji w chwili $t = 1$, które mają być wykonane na pojedynczej maszynie. Każde zadanie i jest scharakteryzowane przez czas jego wykonania p_i oraz funkcję zmiany jego wartości $v_i = \varpi_i C_i^{a_i}$, gdzie $\varpi_i > 0$ oznacza wartość początkową zadania w chwili $t = 1$, natomiast $a_i < 0$ jest współczynnikiem spadku wartości zadania. Rozwiązanie problemu polega na znalezieniu takiego uszeregowanie π, dla którego suma wartości zadań liczona w momentach zakończenia ich wykonywania jest maksymalna, tzn.

$$\sum v_{\pi(i)} = \sum \varpi_{\pi(i)} C_{\pi(i)}^{a_{\pi(i)}} \rightarrow \max \qquad (1)$$

gdzie $C_{\pi(i)}$ oznacza czas zakończenia wykonywania zadania umieszczonego na i-tej pozycji w uszeregowaniu π.

3. Własności problemu i algorytmy rozwiązania

Złożoność obliczeniowa problemu rozpatrywanego w niniejszej pracy nie została jeszcze określona, jednakże z dużym prawdopodobieństwem można stwierdzić, że jest on NP-trudny. Opierając się na tym przypuszczeniu, do rozwiązania problemu skonstruowano algorytmy wykorzystujące metodę podziału i oszacowań [2], która jest stosowana do znajdowania optymalnych rozwiązań problemów NP-trudnych. Ogólną zasadę działania tej metody dla problemów, w których funkcja celu jest maksymalizowana, można opisać następująco. Zbiór wszystkich rozwiązań problemu jest dzielony na mniejsze podzbiory. Dla każdego z tych podzbiorów wyznacza się wartość górnego ograniczenia funkcji celu. Jeżeli wyznaczone górne ograniczenie funkcji celu jest mniejsze od dolnego ograniczenia funkcji celu (najczęściej jest to najlepsza znana wartości tej funkcji), to dany podzbiór jest odrzucany, w przeciwnym przypadku podzbiór podlega dalszym podziałom. Proces ten jest powtarzany tak długo, dopóki nie zostanie znalezione optymalne rozwiązanie problemu. Zatem, aby skonstruować algorytm oparty na metodzie opisanej powyżej, należy określić: sposób generowania podzbiorów rozwiązań, strategię ich przeszukiwania oraz metodę wyznaczania górnego ograniczenia wartości funkcji celu. W algorytmach skonstruowanych w niniejszej pracy zastosowano strategię przeszukiwania „w głąb". Rozwiązania były dzielone na podzbiory zawierające wszystkie rozwiązania częściowe (permutacje częściowe), w których na pierwszych pozycjach kolejność wykonywania zadań była ustalona. Oznaczmy poprzez k liczbę początkowych pozycji w permutacji π, na których uszeregowano już konkretne zadania, natomiast poprzez A zbiór zadań jeszcze nie uszeregowanych. Postać funkcji wyznaczającej górne ograniczenie funkcji celu dla takiego rozwiązania częściowego ma następującą postać:

$$UB(\pi,k)=\sum_{i=1}^{k} v_{\pi(i)}+\sum_{j\in A}\varpi_j\cdot\left(C_{\pi(k)}+p_j\right)^{a_j}. \qquad (2)$$

Algorytm oparty na metodzie podziału i oszacowań opisany powyżej (oznaczany jako B&B') stanowił punkt odniesienia dla algorytmu B&B, w którym zostały zaimplementowane pewne własności problemu. Zastosowanie tych własności pozwoliło wyeliminować z procesu poszukiwań dużą liczbę podzbiorów rozwiązań nie zawierających rozwiązania optymalnego, dzięki czemu działanie algorytmu B&B zostało przyspieszone. Własności zastosowane w algorytmie B&B wykazano poniżej.

Własność 1. Niech A oznacza zbiór zawierający dokładnie k zadań. Niech π oraz π' oznaczają dwie permutacje, w których na pierwszych k pozycjach zostały uszeregowane zadania ze zbioru A. Jeżeli zachodzi $\sum_{i=1}^{k} v_{\pi(i)} < \sum_{i=1}^{k} v_{\pi'(i)}$, to permutacja π nie jest optymalna.

Dowód. Załóżmy, że dane są dwie permutacje π oraz π'. W obu permutacjach kolejność wykonywania zadań na pozycjach od $k+1$-szej do n-tej jest taka sama. Bez straty ogólności możemy założyć, że na pozycjach od $k+1$-szej do n-tej zadania zostały uszeregowane optymalnie. Załóżmy nie wprost, że permutacja π jest optymalna oraz że zachodzi $\sum_{i=1}^{k} v_{\pi(i)} < \sum_{i=1}^{k} v_{\pi'(i)}$. Wartość funkcji celu dla permutacji π wynosi $\sum_{i=1}^{n} v_{\pi(i)} = \sum_{i=1}^{k} v_{\pi(i)} + \sum_{i=k+1}^{n} v_{\pi(i)}$. Z przyjętego założenia $\sum_{i=1}^{k} v_{\pi(i)} < \sum_{i=1}^{k} v_{\pi'(i)}$ wynika, że $\sum_{i=1}^{n} v_{\pi(i)} = \sum_{i=1}^{k} v_{\pi(i)} + \sum_{i=k+1}^{n} v_{\pi(i)} < \sum_{i=1}^{k} v_{\pi'(i)} + \sum_{i=k+1}^{n} v_{\pi(i)} = \sum_{i=1}^{n} v_{\pi'(i)}$, co jest sprzeczne z założeniem o optymalności permutacji π.

Własność 2. Jeżeli zachodzi $v_{\pi(i)} + v_{\pi(j)} < \varpi_{\pi(j)}\cdot\left(C_{\pi(i-1)} + p_{\pi(j)}\right)^{a_{\pi(j)}} + \varpi_{\pi(i)}\cdot C_{\pi(j)}^{\;a_{\pi(i)}}$ oraz $p_{\pi(i)} \geq p_{\pi(j)}$ dla $i<j$ $(i,j=1,\ldots,n)$, wtedy zamiana zadań z pozycji i-tej oraz j-tej poprawia wartość funkcji celu.

Dowód. Załóżmy, że dane są dwie permutacje π oraz π'. Permutacja π' powstała z permutacji π poprzez zamianę zadań z pozycji i-tej oraz j-tej. Załóżmy nie wprost, że permutacja π jest optymalna i że zachodzi $v_{\pi(i)} + v_{\pi(j)} < \varpi_{\pi(j)} \cdot \left(C_{\pi(i-1)} + p_{\pi(j)}\right)^{a_{\pi(j)}} + \varpi_{\pi(i)} \cdot C_{\pi(j)}^{a_{\pi(i)}}$ oraz $p_{\pi(i)} \geq p_{\pi(j)}$. Łatwo można zauważyć, że w obu permutacjach tylko czasy zakończenia wykonywania zadań uszeregowanych na pozycjach od i-tej do j-1-szej są różne. Z założenia $p_{\pi(i)} \geq p_{\pi(j)}$ wynika, że zachodzi $C_{\pi(k)} \geq C_{\pi'(k)}$ oraz $v_{\pi(k)} = C_{\pi(k)}^{a_{\pi(k)}} \leq C_{\pi'(k)}^{a_{\pi(k)}} = v_{\pi'(k)}$ dla $k = i+1, \ldots, j-1$, czyli $\sum_{k=i+1}^{j-1} v_{\pi(k)} < \sum_{k=i+1}^{j-1} v_{\pi'(k)}$. Zatem, różnica wartości funkcji celu wyznaczonych dla permutacji π oraz π' jest dana następująco

$$\sum_{k=1}^{n} v_{\pi(k)} - \sum_{k=1}^{n} v_{\pi'(k)} = v_{\pi(i)} - v_{\pi'(i)} + \sum_{k=i+1}^{j-1} v_{\pi(k)} - \sum_{k=i+1}^{j-1} v_{\pi'(k)} + v_{\pi(j)} - v_{\pi'(j)}. \tag{3}$$

Jak wykazano powyżej, dla wyrażenia (3) zachodzi $\sum_{k=i+1}^{j-1} v_{\pi(k)} \leq \sum_{k=i+1}^{j-1} v_{\pi'(k)}$, natomiast z założenia $v_{\pi(i)} + v_{\pi(j)} < \varpi_{\pi(j)} \cdot \left(C_{\pi(i-1)} + p_{\pi(j)}\right)^{a_{\pi(j)}} + \varpi_{\pi(i)} \cdot C_{\pi(j)}^{a_{\pi(i)}}$ wynika, że nierówność $v_{\pi(i)} + v_{\pi(j)} < v_{\pi'(i)} + v_{\pi'(j)}$ jest prawdziwa. Zatem, zachodzi również $\sum_{i=1}^{n} v_{\pi(i)} < \sum_{i=1}^{n} v_{\pi'(i)}$, co oznacza sprzeczność w stosunku do założenia o optymalności permutacji π.

Własność 3. Niech t_0 oznacza czas rozpoczęcia wykonywania zadania zajmującego pozycję i-tą w permutacji π. Jeżeli zachodzi $p_{\pi(i)} > p_{\pi(j)}$ $(i < j)$ oraz dla każdego t takiego, że $t_0 + p_{\pi(j)} < t < \sum_{k=1}^{n} p_{\pi(k)}$ spełniona jest nierówność $\frac{d(\varpi_{\pi(i)} \cdot t^{a_{\pi(i)}})}{dt} > \frac{d(\varpi_{\pi(j)} \cdot t^{a_{\pi(j)}})}{dt}$, to permutacja π nie jest permutacją optymalną.

Dowód. Załóżmy, że dane są dwie permutacje π oraz π'. Permutacja π' powstała z permutacji π poprzez zamianę zadań z pozycji i-tej oraz j-tej. Załóżmy nie wprost, że permutacja π jest optymalna i że spełnione są dla niej następujące nierówności: (i) $p_{\pi(i)} > p_{\pi(j)}$ oraz (ii) $\frac{d(\varpi_{\pi(i)} \cdot t^{a_{\pi(i)}})}{dt} > \frac{d(\varpi_{\pi(j)} \cdot t^{a_{\pi(j)}})}{dt}$ dla każdego t takiego, że $t_0 + p_{\pi(j)} < t < \sum_{k=1}^{n} p_{\pi(k)}$. W celu uproszczenia notacji przyjęto następujące oznaczenie $f_{\pi(k)}^{'} = \frac{d(\varpi_{\pi(k)} \cdot t^{a_{\pi(k)}})}{dt}$, gdzie $k \in \{i, j\}$. Łatwo można pokazać, że $f_{\pi(k)}^{'} = a_{\pi(k)} \cdot \varpi_{\pi(k)} \cdot t^{(a_{\pi(k)}-1)} < 0$ dla $k \in \{i, j\}$, ponieważ $a_{\pi(k)} < 0$.

Wykorzystując twierdzenie Newtona-Leibniza, dla związku pomiędzy wartością zadania uszeregowanego na pozycji i-tej w permutacji π a wartością tego samego zadania, które w permutacji π' jest uszeregowane na pozycji j-tej, otrzymujemy:

$$v_{\pi'(j)} = v_{\pi(i)} + \int_{C_{\pi(i)}}^{C_{\pi'(j)}} \left(f_{\pi(i)}^{'}\right)dt\,. \tag{4}$$

Dla zadania uszeregowanego w permutacji π na pozycji j-tej, które w permutacji π' zajmuje pozycję i-tą, otrzymujemy analogicznie:

$$v_{\pi(j)} = v_{\pi'(i)} + \int_{C_{\pi'(i)}}^{C_{\pi(j)}} \left(f_{\pi(j)}^{'}\right)dt\,. \tag{5}$$

Suma wartości zadań zajmujących pozycję i-tą oraz j-tą w permutacji π' wyliczona na podstawie wyrażeń (4) oraz (5) wynosi:

$$v_{\pi'(i)} + v_{\pi'(j)} = v_{\pi(i)} + v_{\pi(j)} + \int_{C_{\pi(i)}}^{C_{\pi'(j)}} \left(f_{\pi(i)}^{'}\right)dt - \int_{C_{\pi'(i)}}^{C_{\pi(j)}} \left(f_{\pi(j)}^{'}\right)dt\,. \tag{6}$$

Z założenia $p_{\pi(i)} > p_{\pi(j)}$ wynika, że $C_{\pi(i)} = C_{\pi(i-1)} + p_{\pi(i)} \ge C_{\pi'(i-1)} + p_{\pi(j)} = C_{\pi'(i)}$. Zatem, wykorzystując twierdzenie o addytywności całkowania względem obszarów całkowania otrzymujemy:

$$\int_{C_{\pi(i)}}^{C_{\pi'(j)}} \left(f_{\pi(i)}^{'}\right)dt = \int_{C_{\pi(i)}}^{C_{\pi'(i)}} \left(f_{\pi(i)}^{'}\right)dt + \int_{C_{\pi'(i)}}^{C_{\pi'(j)}} \left(f_{\pi(i)}^{'}\right)dt\,. \tag{7}$$

Wykorzystując założenie $f_{\pi(i)}^{'} = \dfrac{d(\varpi_{\pi(i)} \cdot t^{a_{\pi(i)}})}{dt} > \dfrac{d(\varpi_{\pi(j)} \cdot t^{a_{\pi(j)}})}{dt} = f_{\pi(j)}^{'}$ oraz twierdzenie o zachowaniu nierówności przy całkowaniu otrzymujemy $\int_{C_{\pi(i)}}^{C_{\pi'(j)}} \left(f_{\pi(i)}^{'}\right)dt - \int_{C_{\pi'(i)}}^{C_{\pi(j)}} \left(f_{\pi(j)}^{'}\right)dt > 0$, a to oznacza, że zachodzi $v_{\pi'(i)} + v_{\pi'(j)} > v_{\pi(i)} + v_{\pi(i)}$. Wobec powyższego permutacja π nie może być optymalna, co kończy dowód.

Wykorzystując własności wykazane powyżej, stworzono trzy dodatkowe metody odrzucania podzbiorów rozwiązań nie zawierających rozwiązań optymalnych. Niech k oznacza liczbę uszeregowanych zadań, natomiast A będzie zbiorem zadań nie uszeregowanych. Dodatkowo przyjmijmy, że J^* oznacza zadanie znajdujące się na pozycji k-tej w permutacji π, tzn. J^* jest ostatnio uszeregowanym zadaniem.

Metoda 1. Wstaw zadanie J^* kolejno na pozycje $x = 1,\ldots,k$, wyliczając za każdym razem wartość $V_x = \sum_{i=1}^{k} v_{\pi(i)}$. Jeżeli dla dowolnego $x = 1,\ldots,k-1$ zachodzi $V_x < V_k$, to na podstawie Własności 1 możemy odrzucić podzbiór rozwiązań x.

Metoda 2. Jeżeli istnieje takie $x \in \{1,\ldots,k-1\}$, że zachodzi $p_{\pi(x)} > p_{\pi(k)}$ oraz $v_{\pi(x)} + v_{\pi(k)} < \varpi_{\pi(k)}\left(\sum_{j=1}^{x-1} p_{\pi(j)} + p_{\pi(k)}\right)^{a_{\pi(k)}} + \varpi_{\pi(x)}\left(\sum_{j=1}^{x} p_{\pi(j)}\right)^{a_{\pi(x)}}$, to na podstawie Własności 2 możemy odrzucić podzbiór rozwiązań x.

Metoda 3. Niech $t_0 = \sum_{i=1}^{k} p_{\pi(i)}$. Jeżeli dla dowolnego $j \in A$ spełniona jest nierówność $\frac{d(\varpi_{\pi(k)} \cdot t^{a_{\pi(k)}})}{dt} > \frac{d(\varpi_j \cdot t^{a_j})}{dt}$ dla każdego t takiego, że $t_0 < t < \sum_{i=1}^{n} p_i$ oraz zachodzi $p_{\pi(k)} > p_j$, to na podstawie Własności 3 możemy odrzucić rozważany podzbiór rozwiązań.

Dużą zaletą wszystkich powyższych metod eliminacji podzbiorów rozwiązań jest ich niska złożoność obliczeniowa, która wynosi $O(n)$. Ma to duży wpływ na czas działania całego algorytmu opartego na metodzie podziału i oszacowań.

4. Eksperyment obliczeniowy

Jakość skonstruowanego algorytmu B&B (tzn. czas działania oraz liczba przeszukanych rozwiązań) została zweryfikowana eksperymentalnie. W tym celu wygenerowano po 100 różnych instancji dla dwóch rozmiarów problemu ($n = 30$ oraz $n = 40$). Dla każdej instancji parametr ω_i losowano z przedziału (0,8]. Przedziały, z których generowano pozostałe parametry problemu, zostały opisane w tablicy 1. Wielkości $\bar{t}$, min(t), max(t) oznaczają odpowiednio średni, najmniejszy oraz największy czas działania algorytmu. Wielkości $\bar{c}$, min(c) oraz max(c) oznaczają odpowiednio średnią, minimalną oraz maksymalną liczbę sprawdzonych podzbiorów rozwiązań.

Przydatność metod eliminacji podzbiorów rozwiązań (Metody 1-3), które zostały skonstruowane na podstawie Własności 1-3, sprawdzono w dodatkowym eksperymencie. W tym celu przeprowadzono obliczenia za pomocą dwóch algorytmów opartych na metodzie podziału i oszacowań. W algorytmie B&B zaimplementowano wyżej wspomniane metody, natomiast w algorytmie B&B' nie zastosowano tych metod.

Tablica 1

Charakterystyka działania algorytmu B&B w zależności od parametrów zadań

n	a_i	p_i	$\bar{t}$ [s]	min(t) [s]	max(t) [s]	$\bar{c}$	min(c)	max(c)
30	[-1,0)	(0,1]	13.82	2.61	85.51	187206	38367	1061765
		[1,100]	8.20	0.97	54.03	103444	13375	686586
40	[-5,-1]	(0,1]	9.92	1.46	79.45	123964	21268	913994
		[1,100]	1.33	0.17	8.02	20281	2642	91065

Eksperyment obliczeniowy przeprowadzono dla dwóch rozmiarów problemu ($n = 16$ oraz $n = 22$). Dla każdego rozmiaru problemu wygenerowano 100 różnych instancji. Parametry ϖ_i oraz p_i generowano z rozkładem jednostajnym odpowiednio z przedziałów (0,8] oraz [-1,0). Wartości parametrów a_i dla $n = 16$ generowano z przedziału [-1,0), natomiast dla $n = 22$ parametry te generowano z przedziału [-5,-1]. Wyniki przeprowadzonych eksperymentów zamieszczono w tablicy 2.

Tablica 2

Porównanie działania algorytmu B&B względem algorytmu B&B'

Algorytm	n	a_i	$\bar{t}$ [s]	min(t) [s]	max(t) [s]	$\bar{c}$	min(c)	max(c)
B&B	16	[-1,0)	0.09	0.02	0.72	2661	550	16413
B&B'			16.58	0.11	552.92	1190191	7620	36204845
B&B	22	[-5,-1]	0.11	0.02	0.78	2867	651	15887
B&B'			1.82	0.05	57.18	85997	2614	2756288

Wyniki zawarte w tablicy 1 potwierdzają duży wpływ parametrów zadań na prędkość działania algorytmu B&B. Dla $a_i \in [-5,-1]$ algorytm działa znacznie szybciej niż dla $a_i \in [-1,0)$. Dzieje się tak dlatego, że przy małych wartościach parametru a_i wartości zadań bardzo szybko się zmniejszają. W związku z tym o jakości rozwiązania decyduje uszeregowanie zadań na kilku początkowych pozycjach. Dzięki temu bardzo dużą liczbę podzbiorów rozwiązań można odrzucić już w początkowej fazie działania algorytmu pokazując, że nie zawierają one rozwiązania optymalnego. Prędkość działania algorytmu jest również uzależniona od przedziału, z którego generowane są parametry p_i. Zależność tę można zaobserwować dla dużych różnic pomiędzy czasami wykonywania poszczególnych zadań. W szczególności, część podzbiorów rozwiązań można odrzucić w przypadku, gdy zadanie o krótkim czasie wykonywania znajduje się przed zadaniem o długim czasie wykonywania.

Wyniki zawarte w tablicy 2 potwierdzają duże uzależnienie czasu obliczeń od metod eliminacji podzbiorów rozwiązań wykorzystujących Własności 1-3. Jest to szczególnie widoczne przy porównaniu najdłuższego czasu wykonywania obu algorytmów oraz maksymalnej liczby sprawdzonych podzbiorów rozwiązań. Przeprowadzone eksperymenty potwierdzają dużą przydatność zaimplementowanych metod w rozwiązywaniu tzw. „trudnych" instancji problemu.

5. Podsumowanie

W pracy rozpatrzono jednomaszynowy problem szeregowania zadań o zmiennych wartościach przy kryterium maksymalizacji sumy wartości zadań. Wartość zadania jest opisana potęgową funkcją zależną od momentu zakończenia jego wykonywania. Opierając się na przypuszczeniu, że badany problem jest NP-trudny, do jego rozwiązania skonstruowano algorytm oparty na metodzie podziału i oszacowań. W skonstruowanym algorytmie zastosowano trzy własności eliminacyjne, dzięki którym znacząco przyspieszono działanie algorytmu. Przeprowadzony eksperyment obliczeniowy potwierdził użyteczność wykazanych własności.

LITERATURA

1. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.: Optimization and approximation in sequencing and scheduling: a survey, Annals of Discrete Mathematics, vol. 5, 1979, s. 287-326.
2. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów, Problemy Współczesnej Nauki, Teoria i Zastosowania – Informatyka, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999.
3. Voutsinas T. G., Pappis C. P.: Scheduling jobs with values exponentially deteriorating over time, Raport Techniczny Uniwersytetu w Pireusie, 2000.

Recenzent: Prof. zw. dr hab. inż. Jan Węglarz

Abstract

A single machine scheduling problem with job values exponentially dependent on their completion times was considered in this paper. The objective function was the maximization of the sum of job values calculated at their completion times. An application example of the considered problem describes a renovation process of some buildings or roads. Assume that there is given an institution which has to determine the order in which some objects (buildings, roads, bridges etc.) will be renovated. A job is a renovation of the object. Each object can be described by its value which characterizes its technical state at some time. The value of an object without renovation goes down and the costs required to renovate it increase proportionally to time from its last renovation. The costs represent the loss of the object's value. The problem is to find the order in which the objects will be renovated so that the sum of the costs is minimized. Basing on the assumption that the considered problem is NP-hard, a branch and bound algorithm was constructed to solve it. In this algorithm three properties were utilized. An experimental analysis showed that these properties were very useful.