ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2008
Seria: AUTOMATYKA z. 151 Nr kol. 1797

Grzegorz WALIGÓRA
Politechnika Poznańska

MAKSYMALIZACJA ZAKTUALIZOWANEJ WARTOŚCI NETTO W DYSKRETNO-CIĄGŁYCH PROBLEMACH ROZDZIAŁU ZASOBÓW

Streszczenie. W pracy rozważany jest dyskretno-ciągły problem rozdziału zasobów ze zdyskontowanymi dodatnimi przepływami gotówkowymi. Kryterium jest maksymalizacja zaktualizowanej wartości netto. Rozważane są wypukłe i wklęsłe funkcje szybkości wykonywania czynności w zależności od przydzielonej ilości zasobu ciągłego. Przedstawiona jest metodyka podejścia do rozwiązania problemu w przypadku obu klas funkcji.

MAXIMIZING THE NET PRESENT VALUE IN DISCRETE-CONTINUOUS PROJECT SCHEDULING

Summary. A discrete-continuous project scheduling problem with positive discounted cash flows is considered. The objective is the maximization of the net present value. Convex and concave processing rate functions of activities are analyzed. Methodology for solving the problem in the cases of both considered classes of functions is presented.

1. Wstęp

W pracy rozważany jest dyskretno-ciągły problem rozdziału zasobów, w którym czynności wymagają do swego wykonania jednocześnie zasobów dyskretnych i ciągłych. W rozważanym problemie liczba zasobów dyskretnych, odnawialnych jest dowolna, natomiast występuje jeden odnawialny, ograniczony zasób ciągły. Czynności są niepodzielne i ograniczone kolejnościowo. Szybkość wykonywania czynności zależy od ilości zasobu ciągłego przydzielonej tej czynności w danej chwili. Tak zdefiniowany problem jest uogólnieniem znanego z literatury, klasycznego dyskretnego problemu rozdziału zasobów z ograniczeniami zasobowymi (ang. *Resource-Constrained Project Scheduling Problem* – RCPSP). Ponadto, z każdą czynnością związany jest dodatni przepływ gotówkowy, reprezentujący zapłatę za wykonanie tej czynności. Kryterium jest maksymalizacja zaktualizowanej wartości netto (ang. *net present value* – NPV) wszystkich przepływów gotówkowych projektu. Rozważane kryterium jest analizowane dla dwóch, istotnych z praktycznego punktu widzenia, postaci funkcji szybkości wykonywania czynności – funkcji wypukłych i wklęsłych. Przedstawione są własności uszeregowań optymalnych oraz metodyka rozwiązywania problemu dla rozważanych klas funkcji.

2. Sformułowanie problemu

Dyskretno-ciągły ograniczony zasobowo problem rozdziału zasobów (ang. *Discrete-Continuous Resource-Constrained Project Scheduling Problem* – DCRCPSP) jest zdefiniowany następująco. Dany jest zbiór n ograniczonych kolejnościowo czynności niepodzielnych, które wymagają do swego wykonania zasobów odnawialnych dwóch kategorii: dyskretnych i ciągłych. Zakładamy, że dostępnych jest R zasobów dyskretnych i jeden zasób ciągły. Wektor $\mathbf{r}_i = [r_{il}]$, $i = 1,2,\ldots,n$; $l = 1,2,\ldots,R$, określa ustalone dyskretne żądania zasobowe czynności A_i. Całkowita liczba jednostek zasobu dyskretnego l, $l = 1,2,\ldots,R$ jest ograniczona przez R_l. Zasób ciągły może być przydzielony czynności w dowolnej ilości z przedziału [0,1], a jego całkowita dostępna ilość wynosi 1. Nieznana z góry ilość zasobu ciągłego $u_i(t)$ przydzielona czynności A_i w chwili t określa szybkość wykonania czynności A_i zgodnie z następującą zależnością [3]:

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = f_i[u_i(t)], \quad x_i(0) = 0, \quad x_i(C_i) = \tilde{x}_i \tag{1}$$

gdzie: $x_i(t)$ - stan czynności A_i w chwili t; $u_i(t)$ - ilość zasobu ciągłego przydzielona czynności A_i w chwili t; f_i - ciągła, rosnąca funkcja, $f_i(0) = 0$; C_i - nieznany z góry czas zakończenia czynności A_i; $\tilde{x}_i$ - rozmiar (stan końcowy) czynności A_i.

W problemie DCPCPSP kryterium jest minimalizacja długości uszeregowania. Jego rozszerzeniem jest problem ze zdyskontowanymi przepływami gotówkowymi (ang. *Discrete-Continuous Resource-Constrained Project Scheduling Problem with Discounted Cash Flows* – DCRCPSPDCF), w którym z każdą czynnością A_i związany jest przepływ gotówkowy CF_i. Kryterium optymalizacji jest wówczas maksymalizacja zaktualizowanej wartości netto (NPV) wszystkich przepływów gotówkowych projektu.

Zatem należy znaleźć uszeregowanie czynności dopuszczalne ze względu na ograniczenia kolejnościowe i zasobowe (dyskretne) oraz, równocześnie, przydział zasobu ciągłego, które maksymalizują NPV. Przydział zasobu ciągłego zdefiniowany jest jako odcinkami ciągła, nieujemna funkcja wektorowa $\mathbf{u}(t)=[u_1(t),u_2(t),\ldots,u_n(t)]$, której wartości $\mathbf{u}^*=[u_1^*,u_2^*,\ldots,u_n^*]$ są optymalnymi ilościami przydzielonego zasobu ciągłego odpowiadającymi NPV^* – maksymalnej wartości NPV.

W przypadku problemów z kryteriami finansowymi uwzględnia się wartość pieniądza w czasie; wartość ta zależy od przyjętej stopy dyskontowej α. Czynnik dyskontowy $\beta = (1 + \alpha)^{-T}$ oznacza obecną wartość złotówki, którą mamy otrzymać po okresie T przy zastosowaniu stopy dyskontowej α. Czynnik dyskontowy ma decydujące znaczenie przy obliczaniu zaktualizowanej wartości netto. Drugim istotnym czynnikiem jest przyjęty model płatności. W pracy rozważany jest model, w którym płatności dokonywane są w momentach zakończeń wykonywania poszczególnych czynności (ang. *payments at activity completion times* – PAC).

Metodyka rozwiązywania problemu DCRCPSPDCF zależy od postaci funkcji szybkości wykonywania czynności, tj. funkcji f_i. W kolejnym rozdziale przedstawione będą własności rozwiązań optymalnych problemu dla dwóch istotnych z praktycznego punktu widzenia klas funkcji – funkcji wypukłych i funkcji wklęsłych oraz dodatnich przepływów gotówkowych związanych z czynnościami.

3. Maksymalizacja NPV w problemie DCRCPSPDCF

3.1. Wypukłe funkcje szybkości wykonywania czynności

W przypadku wypukłych funkcji szybkości wykonywania czynności i dodatnich przepływów gotówkowych można udowodnić, że w rozwiązaniu NPV-optymalnym czynności są wykonywane sekwencyjnie bez przerw, a każda czynność wykorzystuje całą dostępną ilość zasobu ciągłego. W konsekwencji długość uszeregowania NPV-optymalnego jest równa minimalnej długości uszeregowania dla problemu DCRCPSP z kryterium $C_{\max}$. Wynika to z faktu, iż w pracy [2] pokazano, że takie wykonanie zbioru czynności minimalizuje długość uszeregowania. Zatem czynności należy uszeregować sekwencyjnie, zachowując ograniczenia kolejnościowe. W rezultacie, można obliczyć rzeczywisty czas wykonania każdej czynności, korzystając ze wzoru (1) przyjmując, że każdą czynność wykonuje się w całości przy użyciu ilości zasobu ciągłego równej 1. Niech d_i oznacza czas wykonywania czynności A_i w uszeregowaniu sekwencyjnym. Wówczas d_i dane jest ze wzoru (1) przez:

$$d_i = \frac{\tilde{x}_i}{f_i(1)}. \tag{2}$$

W konsekwencji długość każdego uszeregowania NPV-optymalnego może być obliczona z góry jako:

$$C^*_{\max} = \sum_{i=1}^{n} d_i . \tag{3}$$

Jednakże powyższy wynik znajduje tylko długość uszeregowania NPV-optymalnego, a nie znajduje optymalnej wartości NPV. Ponieważ czynności mają, w ogólności, różne przepływy gotówkowe, pozostaje problem znalezienia kolejności czynności w uszeregowaniu sekwencyjnym, która maksymalizuje wartość NPV.

Zauważmy, po pierwsze, że jeśli płatności następują w momentach zakończeń wykonywania czynności, to wkład czynności A_i do wartości NPV wyraża się wzorem:

$$NPV_i = CF_i \beta^{C_i} \tag{4}$$

co oznacza, że każda czynność ma przypisany "zysk" za wykonanie jej najwcześniej jak to tylko możliwe. Całkowita wartość NPV jest sumą wartości NPV_i wniesionych przez poszczególne czynności, tzn.:

$$NPV = \sum_{i=1}^{n} CF_i \beta^{C_i} . \tag{5}$$

Ponieważ szeregowanie sekwencyjne jest równoważne szeregowaniu na jednej maszynie, więc można zauważyć, że powstały problem szeregowania czynności zależnych, niepodzielnych jest równoważny znanemu z literatury problemowi szeregowania na jednym procesorze, w którym kryterium jest osiągnięcie maksymalnego zysku ze zbioru zadań. Korzystając ze znanej z literatury klasyfikacji problemów szeregowania maszynowego, rozważany problem szeregowania na jednej maszynie oznaczamy jako $1 \mid prec \mid \Sigma\,(w_i \exp(-cC_i))$, gdzie c jest stopą dyskontową, a w_i jest wagą zadania A_i. Łatwo udowodnić, że po zastosowaniu podstawienia:

$$\begin{cases} w_i = -CF_i \\ c = \ln(1+\alpha) \end{cases} \tag{6}$$

powyższy problem szeregowania na jednej maszynie staje się równoważny rozważanemu problemowi DCRCPSPDCF. Jakkolwiek problem ten jest w ogólności NP-trudny, to jednak w pracy [1] udowodniono, że dla pewnych klas ograniczeń kolejnościowych (np. ograniczenia typu "series-parallel") problem może być rozwiązany w czasie wielomianowym. Oczywiście, to samo zachodzi przy braku ograniczeń kolejnościowych, tj. dla zadań niezależnych.

Zatem dla wypukłych funkcji szybkości wykonywania czynności i dodatnich przepływów gotówkowych wartość NPV jest maksymalizowana przez sekwencyjne wykonanie wszystkich czynności w dopuszczalnym kolejnościowo porządku i bez czasów zwłoki. Ponadto powstały problem jest równoważny znanemu z literatury problemowi szeregowania na jednej maszynie, który dla pewnych klas ograniczeń kolejnościowych może być rozwiązany w czasie wielomianowym.

3.2. Wklęsłe funkcje szybkości wykonywania czynności

W pracy [2] pokazano, że w przypadku wklęsłych funkcji szybkości wykonywania czynności długość uszeregowania jest minimalizowana przez jak najbardziej równoległe wykonywanie zbioru czynności. Można pokazać, że własność ta jest zachowana także dla innego kryterium regularnego, jakim jest maksymalizacja zaktualizowanej wartości netto przy dodatnich przepływach gotówkowych. Do opracowania metodyki podejścia do problemu w tym przypadku wykorzystuje się również pojęcie *sekwencji dopuszczalnej*, wprowadzone w pracy [3].

Zauważmy, że dowolne uszeregowanie dopuszczalne, będące rozwiązaniem dyskretno-ciągłego problemu rozdziału zasobów, może być podzielone na $s \le n$ przedziałów o długości M_k, $k = 1,2,\ldots,s$, określone przez momenty zakończeń kolejnych czynności. Niech Z_k oznacza kombinację czynności wykonywanych równolegle w k-tym przedziale. Z każdym uszeregowaniem dopuszczalnym jest związana sekwencja dopuszczalna S kombinacji Z_k, $k = 1,2,\ldots,s$. Dopuszczalność takiej sekwencji jest zachowana, jeśli:

- każda czynność występuje co najmniej w jednej kombinacji;
- zagwarantowana jest niepodzielność każdej czynności, tzn. każda czynność występuje w dokładnie jednej bądź w kolejnych kombinacjach w S;
- spełnione są ograniczenia kolejnościowe między czynnościami;
- spełnione są ograniczenia zasobowe na zasoby dyskretne.

Dla danej sekwencji dopuszczalnej S można wyznaczyć optymalny podział rozmiarów czynności A_i, $i = 1,2,\ldots,n$, pomiędzy kombinacje w S, rozwiązując odpowiedni problem nieliniowego programowania matematycznego. W problemie tym maksymalizowana jest suma wartości NPV wynikająca z czasów zakończeń kolejnych czynności w taki sposób, że minimalizowane są długości przedziałów odpowiadających kolejnym kombinacjom w S. Jak już wspomniano, przedziały te wyznaczone są przez momenty zakończeń kolejnych czynności, co oznacza, że im mniejsza długość przedziału, tym wcześniejszy moment zakończenia odpowiedniej czynności i większa wartość NPV z niego wynikająca.

Niech M_k^* będzie minimalną długością przedziału wygenerowanego przez kombinację $Z_k \in S$, $k = 1,2,\ldots,s$, funkcją wektora $\tilde{x}_k = [\tilde{x}_{ik}]_{A_i \in Z_k}$, a K_i – zbiorem indeksów wszystkich kombinacji Z_k, takich że $A_i \in Z_k$. Poniższy problem programowania matematycznego znajduje dla danej sekwencji dopuszczalnej S podział rozmiarów czynności pomiędzy kombinacje w S, który maksymalizuje wartość NPV.

Problem P_{NPV}: zmaksymalizować

$$NPV = \sum_{i=1}^{n} CF_i \beta^{C_i} \tag{7}$$

przy ograniczeniach

$$\sum_{k \in K_i} \tilde{x}_{ik} = \tilde{x}_i, \quad i = 1,2,\ldots,n \tag{8}$$

$$\tilde{x}_{ik} \geq 0, \qquad i = 1,2,\ldots,n; \quad k \in K_i \tag{9}$$

gdzie

$$C_i = \sum_{k=1}^{\max\{K_i\}} M_k^* \tag{10}$$

a $M_k^*(\tilde{x}_k)$ jest jedynym dodatnim pierwiastkiem równania

$$\sum_{A_i \in Z_k} f_i^{-1}(\tilde{x}_{ik} / M_k) = 1. \tag{11}$$

Funkcja celu (7) maksymalizuje wartość NPV dla rozważanego modelu płatności PAC. Ograniczenia (8) odpowiadają warunkowi wykonania w całości każdej czynności, a ograniczenia (9) zapewniają, że części czynności wykonywane w poszczególnych przedziałach są nieujemne. Równanie (10) pozwala znaleźć czas zakończenia każdej czynności jako sumę długości wszystkich przedziałów od pierwszego aż do przedziału, w którym czynność jest zakończona. Równanie (11) z kolei pozwala obliczyć minimalną długość k-tego przedziału. Po znalezieniu NPV-optymalnego podziału rozmiarów czynności odpowiadający mu optymalny rozdział zasobu ciągłego dla kombinacji Z_k dany jest przez równanie [3]:

$$u_{ik}^* = f_i^{-1}(\tilde{x}_{ik} / M_k^*), \quad A_i \in Z_k. \tag{12}$$

Zatem problem P_{NPV} pozwala znaleźć optymalny rozdział zasobu ciągłego dla danej sekwencji dopuszczalnej i wklęsłych funkcji szybkości wykonywania czynności. W konsekwencji problem w tym przypadku można zdekomponować na dwa podproblemy: (i) konstrukcję kolejnościowo i zasobowo (ze względu na zasoby dyskretne) dopuszczalnej sekwencji czynności (czyli sekwencji dopuszczalnej zdefiniowanej powyżej) oraz (ii) optymalny rozdział zasobu ciągłego dla sekwencji wynikającej z podproblemu (i), który maksymalizuje zaktualizowaną wartość netto, przez rozwiązanie problemu P_{NPV} dla tej sekwencji. Dla problemów małych rozmiarów można więc znaleźć rozwiązanie globalnie optymalne przez pełen przegląd zbioru sekwencji dopuszczalnych i wybór najlepszej z nich. Niestety, w ogólności liczba wszystkich sekwencji dopuszczalnych rośnie wykładniczo wraz z liczbą czynności. Stąd sensowne staje się zastosowanie algorytmów lokalnego przeszukiwania, poszukujących dobrych rozwiązań w zbiorze wszystkich sekwencji. Takie podejście wykorzystano np. w pracy [4], gdzie dla rozważanej klasy problemów badano efektywność metaheurystyki przeszukiwania tabu (ang. *tabu search* – TS).

4. Zakończenie

W pracy rozważano problem maksymalizacji zaktualizowanej wartości netto w dyskretno-ciągłym problemie rozdziału zasobów ze zdyskontowanymi dodatnimi przepływami gotówkowymi. Zdefiniowano problem DCRCPSPDCF oraz pokazano metodykę podejścia do jego rozwiązywania dla wypukłych i wklęsłych funkcji szybkości wykonywania czynności.

BIBLIOGRAFIA

1. Monma C.L., Sidney J.B.: Optimal sequencing via modular decomposition: characterization of sequencing functions. Mathematics of Operations Research, vol.12(1), 1987, p.22-31.
2. Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Project scheduling under discrete and continuous resources. **in**: Project Scheduling – Recent Models, Algorithms and Applications, Węglarz J. (ed.), Kluwer, Dordrecht 1999, p.289-308.
3. Józefowska J., Węglarz J.: On a methodology for discrete-continuous scheduling. European Journal of Operational Research, vol.107(2), 1998, p.338-353.
4. Waligóra G.: Discrete-continuous project scheduling with discounted cash flows – a tabu search approach. Computers & Operations Research, vol.35(7), 2008, p.2141-2153.

Recenzent: Prof. dr hab. inż. Jerzy Klamka

Abstract

In this work a discrete-continuous resource-constrained project scheduling problem with positive discounted cash flows (DCRCPSPDCF) is considered. In this problem activities simultaneously require for their processing discrete renewable resources, and an amount of a single continuous renewable resource, whose total amount available at a time is limited. The processing rate of an activity depends on the amount of the continuous resource allotted to this activity at a time. Activities are non-preemptable and precedence-related. A positive discounted cash flow is associated with each activity. The PAC payment model is considered, where payments are made at completion times of successive activities. The problem is to find a precedence- and discrete resource-feasible schedule and, simultaneously, a continuous resource allocation that maximize the net present value. Convex and concave processing rate functions of activities are analyzed. It is shown that for convex functions, sequential schedule is optimal, and the problem is equivalent to a single machine scheduling problem known from the literature. For concave functions, parallel configuration is optimal, where an optimal continuous resource allocation can be found by solving a non-linear mathematical programming problem. As a result, the process of looking for optimal solution can be seen as searching for an optimal feasible sequence.