ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: AUTOMATYKA z. 109 Nr kol. 1175

Jan Węglarz
Politechnika Poznańska

O PEWNYCH DYSKRETNO-CIĄGŁYCH PROBLEMACH SZEREGOWANIA
ON SOME DISCRETE-CONTINUOUS SCHEDULING PROBLEMS
DE CERTAINS PROBLÈMES DISCRETS-CONTINUS D'ORDONNANCEMENT

Streszczenie: W pracy rozpatruje się problem szeregowania zadań niezależnych, niepodzielnych, na identycznych, równoległych maszynach, z jednoczesnym rozdziałem zasobu ciągłego, w celu minimalizacji długości uszeregowania. Pokazano, że dyskretne sformułowanie tego problemu jest szczególnym przypadkiem sformułowania dyskretno-ciągłego oraz że to ostatnie daje podstawę do pełnego uwzględnienia natury problemu w konstrukcji algorytmów szeregowania.

Summary: This paper deals with the problem of scheduling independent, nonpreemptable tasks on identical, parallel machines to minimize schedule length. It is proved that the discrete formulation of this problem is a special case of the discrete-continuous formulation, and that the last formulation allows for a deep analysis of the problem specificity and its utilization in the design of scheduling algorithms.

Résumé: Cet article traite le problème d'ordonnancement des tâches indépendantes, indivisibles sur les machines identiques parallèles pour minimaliser la longueur de l'ordonnancement. On a prouvé que la formulation discrète de ce problème était le cas spécial de la formulation discrète-continue. Alors, cela permet de prendre en considération la nature specifique de ce problème et l'utiliser en construction des algorithms d'ordonnancement.

1. Wstęp

O dyskretno-ciągłych problemach szeregowania mówimy wówczas, gdy jednoczesnemu rozdziałowi podlegają zasoby dyskretne (czyli podzielne w sposób dyskretny) oraz ciągłe, czyli podzielne w sposób ciągły. Problemy te stanowią ważną klasę tzw. mieszanych problemów szeregowania (lub rozdziału zasobów) [6]. Najczęstsze w praktyce są tu problemy, w których zasobami dyskretnymi są, ogólnie rozumiane, maszyny. Jeśli są to maszyny równoległe (w szczególności jedna maszyna), to mamy do czynienia z jednym zasobem dyskretnym, złożonym z ustalonej lub dowolnej liczby jednostek (maszyn). Oprócz tego zasobu mamy na ogół do czynienia z jednym zasobem ciągłym, np. pieniędzmi, mocą. Problemy tego typu były badane w licznych pracach i przy różnych modelach wykonywania zadań - przegląd głównych wyników znaleźć można w [2] i [3].

Ogólnie można stwierdzić, że dogłębna identyfikacja specyfiki problemów dyskretno-ciągłych wymaga uwzględnienia obu konstytutywnych elementów ich natury. Tymczasem w literaturze pojawiają się prace, których autorzy zdają się nie dostrzegać dyskretnej bądź ciągłej natury problemu dyskretno-ciągłego. Prowadzi to niekiedy do wykazywania wyników dawno znanych, a także do rozpatrywania modeli, które w konkretnym kontekście praktycznym nie mają sensu.

W tej pracy przedstawimy przykład powyższej sytuacji, w którym problem dyskretno-ciągły został sformułowany jako dyskretny. Sformułowanie to przytoczymy w rozdziale 2, a w rozdziale 3 pokażemy, że jest ono szczególnym przypadkiem sformułowania dyskretno-ciągłego. Rozdział 4 zawiera propozycje algorytmów szeregowania, wykorzystujące sformułowanie dyskretno-ciągłe, a rozdział 5 - uwagi końcowe.

2. Sformułowanie dyskretne

W [4] rozpatruje się problem praktyczny, polegający na tym, że m identycznych dystrybutorów paliwa zasilanych ze wspólnego źródła mocy ma napełnić zbiorniki n statków. Mamy zatem m identycznych maszyn $M_1, M_2, \ldots, M_m$ i n niepodzielnych, niezależnych zadań $Z_1, Z_2, \ldots, Z_n$ o stanach początkowych $x_i(0)=0$, $i=1,2,\ldots,n$ oraz końcowych $x_i(C_i)=\tilde{x}_i>0$, $i=1,2,\ldots,n$, gdzie C_i jest nie znanym a priori momentem zakończenia wykonywania zadania Z_i, a $\tilde{x}_i$ jest jego znanym stanem końcowym (tu: pojemnością zbiorników paliwa i-tego statku). Model wykonywania zadania Z_i (czyli napełniania zbiorników i-tego statku), $i=1,2,\ldots,n$, ma postać:

$$\dot{x}_i(t) = \phi[Y(t)] \tag{1}$$

gdzie ϕ jest funkcją dodatnią, nierosnącą, a $Y(t)$ jest liczbą zadań wykonywanych w chwili t; $Y(t)\in\{1,2,\ldots,m\}$. Prędkość wykonywania zadania (napełniania zbiorników statku) maleje zatem wraz ze wzrostem liczby zadań równocześnie wykonywanych.

Z różnych kryteriów optymalności szeregowania największe znaczenie praktyczne ma w tym wypadku długość uszeregowania $T = \max\{C_i\}$. Autorzy [4] udowadniają m.in. następujące wyniki.

Twierdzenie 1

Jeśli w (1) $\phi = c/Y(T)$, to wartość T jest stała, niezależna od przydziału zadań do maszyn, i wyraża się wzorem

$$T = \frac{1}{c}\sum_{i=1}^{n}\tilde{x}_i \tag{2}$$

Twierdzenie 2

Jeśli w (1) $\phi \le c/Y(t)$, $c=\phi(1)$, to wykonywanie zadań na jednej maszynie minimalizuje T.

Twierdzenie 3

Jeśli w (1) $\phi = c/Y(t)$ dla $Y(t) > n^*$ oraz $\phi = k$ dla $Y(t) \le n^*$, gdzie $n^* < n < m$, $c > k > c/(n^*+1)$, to równoległe wykonywanie zadań zawsze prowadzi do mniejszej wartości T niż szeregowe (czyli na jednej maszynie).

Powyższe wyniki posłużyły do podania w [4] raczej enigmatycznej i nie popartej wynikami obliczeniowymi propozycji algorytmu heurystycznego, rozwiązującego pewien ogólniejszy przypadek rozważanego problemu.

3. Sformułowanie dyskretno-ciągłe

Zauważmy, że w sformułowaniu dyskretnym nie rozpatruje się *explicite* rozdziału zasobu ciągłego, którym jest wydajność wspólnego źródła mocy (pompy). Zasób ten występuje jedynie *implicite*: w związku z uruchomieniem kolejnego dystrybutora (równoległe wykonywanie kolejnego zadania) zmniejsza się część wydajności pompy przypadająca na jeden dystrybutor, co wyraża model (1). Poniżej pokażemy, że takie podejście nie pozwala na pełną analizę problemu, a także prowadzi do wyważania dawno otwartych drzwi. Przedtem jednak przedstawmy sformułowanie dyskretno-ciągłe rozpatrywanego problemu, ukazując jednocześnie związek tego sformułowania ze sformułowaniem dyskretnym.

W tym celu oznaczmy przez $u_i(t)\in[0,1]$ ilość zasobu ciągłego przydzieloną do zadania Z_i, to znaczy część wydajności pompy przypadającą na dystrybutor napełniający zbiorniki paliwa i-tego statku.

Rozważmy model

$$\dot{x}_i(t) = f[u_i(t)], \; i=1,2,\dots,n \tag{3}$$

gdzie f jest funkcją ciągłą, niemalejącą, f(0)=0. Zauważmy, że podstawiając

$$u_i(t) = 1/Y(t) \text{ dla } Y(t)\in\{1,2,\dots,m\} \tag{4}$$

mamy $f[1/u_i(t)] = \phi[Y(t)]$ oraz dla każdego Y(t)

$$\sum_{i=1}^{Y(t)} u_i(t) = 1 \tag{5}$$

Widzimy zatem, że problem sformułowany w rozdziale 2 jest szczególnym przypadkiem problemu, w którym należy uszeregować n zadań niepodzielnych, niezależnych na m identycznych, równoległych maszynach, przy czym każde zadanie wymaga do swego wykonywania w każdej chwili t jednej (dowolnej) maszyny i zasobu ciągłego w (nie znanej *a priori*) ilości $0 < u_i(t) \le 1$, a model wykonywania zadania ma postać (3). Jest oczywiste, że wszystkie wyniki dotyczące własności optymalnych rozdziałów zasobu o łącznej ilości równej 1, uzyskane dla ciągłych $u_i(t)$, zgodnie z modelem (3), są w szczególności prawdziwe także dla wartości dyskretnych wynikających z (4) i (5). Zauważmy, że sformułowany problem w swym aspekcie ciągłym (tzn. dotyczącym rozdziału zasobu ciągłego pomiędzy zadania, które mają w danej chwili przyznane maszyny) jest klasycznym problemem czaso-optymalnego rozdziału zasobu odnawialnego między zadania (operacje) niezależne. Jego jedyna specyfika polega na tym, że funkcja f w (3) nie jest związana z zadaniem, które jest charakteryzowane wyłącznie przez stan końcowy $\tilde{x}_i$, a z maszyną. Jest to okoliczność upraszczająca (w ogólności w (3) mieliśmy różne funkcje f_i również dla maszyn identycznych - por. [8]), która jednak nie ma wpływu na jakościowe własności rozwiązań optymalnych. Własności te znane są od ponad dwudziestu lat i były podane wraz z dowodami w czasopismach o zasięgu międzynarodowym (np. [6]), w tym również (dla ogólniejszego przypadku zasobów podwójnie ograniczonych) w Management Science [9], a więc w czasopiśmie, w którym ukazała się praca [4]. Powyższe stwierdzenie absolutnie nie sugeruje plagiatu, ilustruje tylko zjawisko coraz węższej specjalizacji naukowców.

Osoby zajmujące się optymalizacją kombinatoryczną, nawet te z elity światowej, nie dostrzegają niekiedy wyników "zza miedzy", także wtedy, gdy tę miedzę przekraczają. W szczególności wyniki cytowane w Tw. 1-3 w rozdziale 2 są od dawna znane. W kategoriach modelu (3) dotyczą one odpowiednio: funkcji liniowych (Tw. 1), funkcji spełniających warunek $f \le cu_i(t)$, a ogólnie: $f_i \le c_i u_i(t)$, $i=1,2,\dots,n$ (Tw. 2) oraz (dowolnych) funkcji wklęsłych (Tw. 3). W drugim przypadku czaso-optymalne jest szeregowe wykonywanie zadań (a więc, jeśli uwzględnić także maszyny, wykonywanie zadań na jednej maszynie), w trzecim - maksymalnie równoległe wykonywanie zadań (a więc na wszystkich dostępnych maszynach). Przypadek pierwszy jest graniczny - wartość T jest stała, niezależna od rozdziału zasobu (a więc także od przydziału zadań do maszyn) i wyraża się wzorem (2).

4. Algorytmy szeregowania

W [4] wykazano, że rozpatrywany problem, sformułowany jako dyskretny (por. rozdział 2), jest NP-trudny już dla m=2 oraz dla dowolnych nierosnących funkcji ϕ takich, że $\phi(1) = c$, $\phi(2) > c/2$. Jest to oczywiste uogólnienie klasycznego wyniku dla problemu szeregowania zadań niepodzielnych o stałych (ale nie identycznych) czasach wykonywania (czyli, w tym modelu, dla $\phi(1) = \phi(2) = c$). Uzasadnione jest zatem poszukiwanie efektywnych i skutecznych algorytmów przybliżonych. Poniżej pokażemy, że sformułowanie dyskretno-ciągłe daje racjonalne podstawy do konstrukcji i badania takich algorytmów, a ponadto pozwala na określenie sytuacji, w których istnieją rozwiązania analityczne.

Rozpatrzmy problem ogólniejszy, gdy w (3) mamy różne funkcje f_i, $i=1,2,\dots,n$, przy czym załóżmy, że są one wklęsłe, co odpowiada sytuacjom praktycznym.

Zacznijmy od przypadku, gdy $n = m$, to znaczy liczba zadań jest równa liczbie maszyn. Wówczas problem przydziału zadań do maszyn jest trywialny, gdyż, jak wspomniano w rozdziale 3, dla funkcji wklęsłych optymalne jest równoległe wykonywanie zadań. Pozostaje zatem tylko problem rozdziału zasobu ciągłego pomiędzy m równolegle wykonywanych zadań, którym przydzielono po jednej maszynie. Jak wiadomo (por.np. [6]), dla f_i wklęsłych, $i=1,2,\dots,m$, minimalny czas T^* wykonania zadań o stanach końcowych $\tilde{x}_i$, $i=1,2,\dots,m$, jest jedynym dodatnim pierwiastkiem równania

$$\sum_{i=1}^{m} f_i^{-1}(\tilde{x}_i/T) = 1 \tag{6}$$

gdzie f_i^{-1} jest funkcją odwrotną do f_i, a optymalny rozdział zasobu jest stały dla każdego zadania i ma postać:

$$u_i^* = f_i^{-1}(\tilde{x}_i/T^*),\ i=1,2,\dots,m \tag{7}$$

Równanie (6) można niekiedy rozwiązać analitycznie, w szczególności dla $f_i = c_i u_i^{1/\alpha_i}$, $c_i > 0$, $\alpha_i \in \{1,2,3,4\}$, kiedy (6) jest równaniem algebraicznym rzędu ≤ 4.

Powyższe rozumowanie sugeruje dwa podejścia do konstrukcji algorytmów przybliżonych dla $n > m$.

Pierwsze podejście jest co do istoty identyczne z przedstawionym w [8] dla analogicznego problemu z zadaniami podzielnymi i składa się z następujących punktów:

1° Przydziel zadaniom, w sposób heurystyczny, ilości zasobu $\hat{u}_i$, $i=1,2,\dots,n$.

2° Dla czasów wykonywania zadań wynikających z powyższego rozdziału zasobu, czyli dla

$$\tau_i = \tilde{x}_i / f_i(\hat{u}_i), \; i=1,2,\dots,n \tag{8}$$

rozwiąż klasyczny problem szeregowania n zadań na m maszynach.

3° Rozdziel zasób ciągły pomiędzy części zadań wykonywane równolegle w uszeregowaniu otrzymanym w p. 2°. Jeśli długość uszeregowania uległa przy tym zmniejszeniu, to przyjmij ten rozdział zasobu, w przeciwnym razie przyjmij rozdział z p. 1°.

Skomentujmy krótko poszczególne punkty.

W p. 1° należy, ogólnie biorąc, wziąć pod uwagę modele i stany końcowe zadań w ten sposób, by punkty 2° i 3° doprowadziły do uszeregowania o jak najmniejszej długości. Można w tym celu skorzystać ze wskazówek, które dały dobre wyniki dla zadań podzielnych [8], nie ma jednak gwarancji, że również obecnie będzie to postępowanie właściwe.

Problem szeregowania rozwiązywany w p. 2° jest, jak wspomnieliśmy, NP-trudny już dla m=2. Do jego przybliżonego rozwiązania można zastosować algorytm lub jeden z algorytmów mających lepsze oszacowanie najgorszego przypadku, np. typu MULTIFIT [3]. W wielu sytuacjach praktycznych realne jest nawet zastosowanie algorytmu dokładnego, opartego na programowaniu dynamicznym, o złożoności $O(n^{2k}\log m)$, gdzie k jest liczbą klas zadań o jednakowych τ_i [2].

W p. 3° wykorzystujemy wyniki uzyskane dla rozdziału zasobu ciągłego, gdyż liczba zadań wykonywanych równocześnie jest $\leq m$. Najlepiej, rzecz jasna, wyznaczyć dla każdego zbioru części zadań optymalny rozdział zasobu, co w większości wypadków praktycznych jest w pełni realne obliczeniowo.

Przejdźmy teraz do podejścia drugiego, które jest ciekawsze teoretycznie i pozwala na zidentyfikowanie sytuacji, w których można efektywnie wyznaczyć uszeregowanie optymalne [10]. Główna idea tego podejścia polega na badaniu tzw. potencjalnie optymalnych przydziałów zadań do maszyn, to znaczy przydziałów mających tę własność, że każde uszeregowanie optymalne (czyli przyporządkowanie zadaniom odcinków czasu pracy maszyn i ilości zasobu ciągłego minimalizujące T) odpowiada jakiemuś przydziałowi potencjalnie optymalnemu. Łatwo wykazać następujące twierdzenie (dla prostoty utożsamiamy zadanie z jego indeksem).

Twierdzenie 4

Dla m maszyn i n zadań niezależnych, niepodzielnych, o funkcjach f_i wklęsłych, $i=1,2,\ldots,n$, potencjalnie optymalny przydział zadań do maszyn jest ciągiem m-elementowych kombinacji zbioru $\{1,2,\ldots,n\}$, zapewniającym niepodzielność każdego zadania i mającym maksymalną długość. Długość ta wynosi $n-m+1$.

Dla ilustracji rozpatrzmy następujący przykład.

Przykład

Niech $n=5$, $m=3$. Wówczas przykładowe potencjalnie optymalne przydziały zadań do maszyn mają postać:

(a) $\{1,2,3\}$, $\{2,3,4\}$, $\{3,4,5\}$

(b) $\{1,2,3\}$, $\{1,2,4\}$, $\{1,2,5\}$

Zbadajmy strukturę tych przydziałów, wyróżniając zadania jednoczęściowe (występujące w dokładnie jednej kombinacji) oraz zadania wieloczęściowe, występujące w większej liczbie kombinacji. Struktura ta jest następująca:

(a)	zadania jednoczęściowe:	1,5
	zadania wieloczęściowe:	2,3,4
	liczba części :	2 3 2
(b)	zadania jednoczęściowe:	3,4,5
	zadania wieloczęściowe:	1,2
	liczba części :	3 3

Podkreślmy, że podział zadania Z_i (to znaczy wartości $\tilde{x}_i$) na części realizowane w różnych kombinacjach nie oznacza, w wypadku potencjalnie optymalnego przydziału, przerywania wykonywania tego zadania, a więc nie narusza założenia niepodzielności.

Możemy teraz przedstawić podejście do konstrukcji algorytmów szeregowania, które składa się z dwóch punktów:

1° Wybierz potencjalnie optymalny przydział zadań do maszyn.

2° Dla wybranego przydziału rozdziel zasób ciągły.

Wykonując punkt 1° uwzględniamy strukturę przydziału (por. Przykład), starając się uzyskać jej jak najlepszą zgodność ze strukturą zbioru zadań. Dla ilustracji tej idei załóżmy, że $f_i = f$, $i=1,2,\ldots,n$, czyli struktura zbioru zadań jest określona wyłącznie przez ich stany końcowe $\tilde{x}_i$, $i=1,2,\ldots,n$. Niech w rozpatrywanym wyżej przykładzie wynoszą one: $\tilde{x}_1 = \tilde{x}_5 = 10$, $\tilde{x}_2 = \tilde{x}_4 = 20$, $\tilde{x}_3 = 30$. Wówczas łatwo zauważyć, że jeśli dla przydziału (a) rozdzielimy zasób ciągły w ten sposób, że każde zadanie w każdej kombinacji otrzyma $u_i = 1/m$, to uzyskamy uszeregowanie optymalne. Odpowiada to bowiem pełnej zgodności struktury przydziału (potencjalnie optymalnego) ze strukturą zbioru zadań (podział Z_2 i Z_4 na 2 równe części i Z_3 na 3 równe części zapewnia równomierność "obciążenia" każdej kombinacji). Ten sam wynik otrzymamy dla $\tilde{x}_1 = \tilde{x}_2 = 30$, $\tilde{x}_3 = \tilde{x}_4 = \tilde{x}_5 = 10$, jeśli zastosujemy przydział (b).

Wykorzystując powyższą ideę można wykazać szereg wyników szczegółowych, przykładowo następujący.

Twierdzenie 5

Jeśli: (i) $f_i = f$, $i=1,2,\dots,n$, f jest wklęsła oraz

(ii) zbiór zadań $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$, gdzie $\mathcal{Z}_1 \cap \mathcal{Z}_2 = \emptyset$,

$\tilde{x}_i = \tilde{x}^{(1)}$ dla $Z_i \in \mathcal{Z}_1$, $\tilde{x}_i = \tilde{x}^{(2)}$ dla $Z_i \in \mathcal{Z}_2$, $\tilde{x}^{(1)} > \tilde{x}^{(2)}$,

to uszeregowanie optymalne, w którym w każdej chwili każde zadanie jest wykonywane stałą ilością zasobu $u_i = 1/m$, istnieje wtedy i tylko wtedy, gdy $|\mathcal{Z}_1| = m-1$. Odpowiada ono potencjalnie optymalnemu przydziałowi utworzonemu według schematu (b).

Zauważmy, że w rozpatrywanym przykładzie mieliśmy już tę sytuację ($\tilde{x}_1 = \tilde{x}_2 = \tilde{x}^{(1)} = 30$, $\tilde{x}_3 = \tilde{x}_4 = \tilde{x}_5 = \tilde{x}^{(2)} = 10$).

Oczywiście, w ogólności potencjalnie optymalny przydział zadań do maszyn uzyskany w p. 1° prowadzi tylko do uszeregowania przybliżonego, nawet jeśli p. 2° wykonamy w sposób optymalny. Jednak znając schematy tworzenia potencjalnie optymalnych przydziałów i wynikającą stąd ich strukturę, można wygenerować przydział, który z dużym prawdopodobieństwem prowadzi do uszeregowania optymalnego. Prawdopodobieństwo to można zwiększyć rozpatrując w p. 2° więcej potencjalnie optymalnych przydziałów, w skrajnym przypadku wszystkie, co dla rozsądnych rozmiarów pozwala rozwiązać problem w sposób dokładny.

Chcąc znaleźć optymalny rozdział zasobu dla danego przydziału, w ogólności musimy rozwiązać problem programowania nieliniowego z liniowymi ograniczeniami. W problemie tym szukamy optymalnego podziału $\tilde{x}_i$, $i=1,2,\dots,n$, na części realizowane w poszczególnych kombinacjach. Jeśli części te dla kombinacji K_j oznaczymy przez $\tilde{x}_{ij}$, to minimalny czas trwania tej kombinacji, T_j^*, w funkcji $\tilde{x}_{ij}$ wyznaczamy rozwiązując równanie (6) napisane dla $i \in K_j$, $\tilde{x}_{ij}$ oraz T_j. Stosując twierdzenie Lagrange'a łatwo pokazać, że dla $f_i = c_i u_i^{1/\alpha}$, $c_i > 0$, $\alpha > 1$, $i=1,2,\dots,n$, w optymalnym uszeregowaniu każde zadanie wykonywane jest przez cały czas stałą ilością zasobu. Pozwala to zredukować liczbę zmiennych w odpowiednim problemie programowania nieliniowego.

Rozdział zasobów w p. 2° można także, rzecz jasna, wyznaczyć heurystycznie, dzieląc $\tilde{x}_i$, $i=1,2,\dots,n$, pomiędzy kombinacje według pewnej reguły uwzględniającej strukturę rozpatrywanego przydziału zadań do maszyn. Jak wspomnieliśmy, można wykazać, że w niektórych sytuacjach reguły takie prowadzą do uszeregowań optymalnych.

5. Uwagi końcowe

Przedstawione w pracy dyskretno-ciągłe problemy szeregowania mają istotne znaczenie praktyczne, m.in. w sytuacjach, gdy maszyny (w rozumieniu urządzeń wykonujących pewne zadania) są zasilane ze wspólnego źródła mocy (elektrycznej, hydraulicznej, pneumatycznej). Są one, jak się okazuje, ciekawe także z punktu widzenia teorii szeregowania. Dotyczy to zarówno identyfikacji przypadków, w których można w sposób efektywny obliczeniowo uzyskiwać uszeregowania optymalne, jak i badania algorytmów przybliżonych. Ciekawa jest np. analiza najgorszego przypadku dla algorytmów reprezentujących podejścia przedstawione w rozdziale 4. Ważne są także liczne możliwe uogólnienia przedstawionej metodyki. Mogą one dotyczyć np. różnych momentów gotowości zadań, innych kryteriów szeregowania czy podejścia wielokryterialnego.

LITERATURA

[1] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Scheduling Under Resource Constraints-Deterministic Models. J.C.Baltzer, Basel, 1986.

[2] Błażewicz J., Kubiak W., Szwarcfiter J.: Scheduling independent, fixed-type tasks. W: Słowiński R., Węglarz J. (Eds.): Advances in Project Scheduling. Elsevier, Amsterdam, 1989, 225-237.

[3] Coffman E.G. (Jr.), Garey M.R., Johnson D.S.: Approximation algorithms for bin-packing - an updated survey. W: Ausiello G., Lucertini M., Serafini P. (Eds.): Algorithms Design for Computer System Design. Springer Verlag, Wien, 1984, 49-106.

[4] Dror M., Stern H.J., Lenstra J.K.: Parallel machine scheduling: processing rates dependent on number of jobs in operation. *Management Sci.* 33, No.8, 1987, 1001-1009.

[5] Janiak A.: Dokładne i przybliżone algorytmy szeregowania zadań i rozdziału zasobów w dyskretnych procesach przemysłowych. Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej 87, Seria Monografie 20, Wrocław, 1991.

[6] Węglarz J.: Time-optimal control of resource allocation in a complex of operations framework. *IEEE Trans. Systems, Man and Cybernet.* SMC-6, No.11, 1976, 783-788.

[7] Węglarz J.: On certain models of resource allocation problems. *Kybernetes* 2, No.1, 1980, 61-66.

[8] Węglarz J.: Multiprocessor scheduling with memory allocation - a deterministic approach. *IEEE Trans. Computers* C-29, No.8, 703-710.

[9] Węglarz J.: Project scheduling with continuously divisible doubly-constrained resources. *Management Sci.* 27, No.9, 1981, 1040-1053.

[10] Węglarz J.: Scheduling nonpreemptable tasks with additional continuous resource (w przygotowaniu).

Recenzent: Prof.dr h.inż. Jerzy Klamka
Wpłynęło do Redakcji do 30.04.1992 r.

Abstract:

This paper deals with scheduling independent, nonpreemptable tasks on identical, parallel machines which share a common continuous, renewable resource (e.g. power). We start with the discrete formulation (task processing rates depend on a number of tasks processed at time t), which is a special case of the discrete-continuous formulation. Then, on the basis of the second formulation, we propose two general methods of constructing approximate scheduling algorithms. The first method is a modification of the method developed for preemptable tasks, whereas the second one uses the concept of so called virtually optimal assignments of tasks to machines. Studying the structure of these assignments one can even find simple rules producing optimal schedules in some cases.