ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: AUTOMATYKA z. 117 Nr kol. 1337

Stanisław ZDRZAŁKA
Politechnika Wrocławska

# GRUPOWANIE ZADAŃ W DWUMASZYNOWYM SYSTEMIE PRZEPŁYWOWYM

**Streszczenie.** W pracy zajmujemy się zagadnieniem grupowania zadań i szeregowania grup w dwumaszynowym systemie przepływowym, w którym występują rodziny zadań, czasy przezbrojeń związane z wykonywaniem grup zadań, a zadania dostępne są grupami. Przedstawiamy dwa algorytmy aproksymacyjne o wskaźniku dokładności dla najgorszego przypadku 3/2.

# BATCHING IN THE TWO-MACHINE FLOWSHOP

**Summary.** The paper deals with the problem of batching and sequencing in a two machine flowshop with job families, batch setup times and batch availability of jobs. We present two approximation algorithms with the worst-case performance ratio of 3/2.

## 1. Wstęp

W systemie przepływowym każde zadanie wykonywane jest kolejno na maszynach $1,2,\ldots,m$, w każdej chwili każda z maszyn może wykonywać co najwyżej jedno zadanie i każde zadanie może być wykonywane przez co najwyżej jedną maszynę. Najbardziej znany problem optymalizacyjny polega na znalezieniu kolejności wykonywania zadań (jednakowej dla każdej z maszyn) minimalizującej czas wykonania wszystkich zadań. To klasyczne sformułowanie problemu przepływowego nie uwzględnia wielu ważnych, z punktu widzenia zastosowań w zautomatyzowanych systemach produkcyjnych, ograniczeń oraz charakterystyk zadań i zbiorów zadań. W pracy rozważamy jego uogólnienie, nazywane dalej problemem grupowania zadań, w którym przyjmuje się następujące założenia:

- Zbiór zadań jest sumą rozłącznych podzbiorów (*rodzin*) zadań podobnych, wymagających jednakowego uzbrojenia maszyn.
- Zadania każdej rodziny wykonywane są i przekazywane do kolejnych maszyn w podzbiorach zwanych *grupami*. Zadania grupy wykonywane są na maszynie jedno po drugim, zaś zadanie staje się dostępne do dalszego przetwarzania na kolejnej maszynie dopiero po

wykonaniu ostatniego zadania swojej grupy na maszynie bieżącej (*grupowa dostępność zadań*).

- Z każdą grupą związane są czasy przezbrojenia maszyn. Wyróżniamy przezbrojenie *przywiązane do grupy*, które może się rozpocząć dopiero po zakończeniu wykonywania grupy na maszynie poprzedzającej, oraz *odłączone od grupy*, które może się odbyć w czasie, gdy grupa, której ono dotyczy, okupuje maszynę poprzedzającą.

Rozważania ograniczamy do przypadku dwumaszynowego. Klasyczny dwumaszynowy problem przepływowy jest rozwiązywalny w wielomianowym czasie w oparciu o regułę Johnsona [3], natomiast zagadnienia z liczbą maszyn większą od dwóch są NP-trudne. Powstaje więc pytanie, jak dalece korzystne własności klasycznego problemu z dwoma maszynami przenoszą się na jego uogólnienie. Jak dotąd, w literaturze badany był tylko przypadek szczególny, w którym wszystkie zadania wewnątrz rodziny są identyczne, wszystkie grupy danej rodziny muszą być wykonywane jedna po drugiej, brak jest przezbrojeń maszyn związanych z grupami zadań, a problem polega na znalezieniu rozbicia rodzin na grupy, porządku grup wewnątrz rodzin i porządku rodzin minimalizujących czas wykonania wszystkich zadań, *Potts, Baker* [5], *Centikaya* [1], *Vickson* [10]; ponieważ przy identycznych zadaniach wewnątrz rodziny ważny jest raczej rozmiar grupy aniżeli jej skład, omawiany przypadek szczególny nazywany jest problemem wyznaczania porcji transferowych (w literaturze angielskiej, *lot streaming*). W [1] i [10] pokazano, że przypadek ten jest rozwiązywalny w wielomianowym czasie, przy czym zagadnienie to dekomponuje się w naturalny sposób na problem znajdowania optymalnych grup w każdej rodzinie, niezależnie od kolejności rodzin, oraz zagadnienie szeregowania rodzin; to ostatnie jest rozwiązywalne za pomocą odpowiednio zaadaptowanej reguły Johnsona. Odpowiednik problemu grupowania zadań w dwumaszynowym systemie przepływowym, w którym przyjmuje się w miejsce grupowej *jednostkową dostępność zadań*, badany był w pracach: *Sekiguchi* [7], gdzie podano rozszerzenie algorytmu Johnsona na przypadek, gdy podział rodzin na grupy jest zabroniony (założenie *technologii grupowej*), *Kleinau* [4], *Zdrzałka* [13],[16], gdzie pokazano, że problem ogólny jest NP-trudny i zaproponowano [13],[16] algorytm aproksymacyjny o wskaźniku dokładności dla najgorszego przypadku 4/3. *Vickson i Alfredsson* [9] podali pewne wyniki dotyczące wariantu z jednostkową dostępnością zadań oraz identycznymi zadaniami wewnątrz rodzin. Pokrewne jednomaszynowe zagadnienia grupowania zadań dla różnych funkcji celu rozważano w pracach, np. [2],[6],[11],[14],[15].

## 2. Notacja i własności

Danych jest $f$ rodzin zadań $I_1, I_2, \ldots, I_f$, które mają być wykonane na dwóch maszynach systemu przepływowego. Niech para $(i, j)$ oznacza $j$-te zadanie rodziny $I_i$, a $n_i$ niech będzie liczbą zadań w tej rodzinie; oznaczmy $n = \sum_{i=1}^{f} n_i$. Zadanie $(i, j)$ wykonywane jest najpierw przez $a_{ij}$ jednostek czasu na maszynie 1 i następnie przez $b_{ij}$ jednostek czasu na maszynie 2, wszystkie zadania są gotowe do realizacji w chwili zerowej. Zadania każdej rodziny są wykonywane w grupach, przy czym obowiązuje założenie o grupowej dostępności zadań. Wykonanie dowolnej (niepustej) grupy zadań $B \subset I_i$ na maszynie $k$ wymaga $s_{ki}$ jednostek czasu na uprzednie przezbrojenie maszyny. Rozbicie rodzin na grupy oraz porządek wykonywania grup nazywamy *harmonogramem*.

Problem grupowania zadań (GZ) polega na znalezieniu harmonogramu, dla którego czas wykonania wszystkich zadań jest minimalny.

Problem GZ jest NP-trudny. Wynika to z dowodów NP-zupełności decyzyjnej wersji odpowiednika problemu GZ, z jednostkową dostępnością zadań, podanych w [4] (inna postać redukcji [16]), które pozostają ważne przy założeniu grupowej dostępności zadań. Kwestią otwartą pozostaje złożoność obliczeniowa przypadku szczególnego z identycznymi zadaniami wewnątrz rodzin.

Niech dla $i$ oraz $B \subset I_i$, $P_1(B) = \sum_{(i,j)\in B} a_{ij}$, $P_2(B) = \sum_{(i,j)\in B} b_{ij}$, $S_1(B) = s_{1i}$ oraz $S_2(B) = s_{2i}$.

Można pokazać, że dla zadanego rozbicia rodzin na grupy, $B_1, B_2, \ldots, B_v$, oraz kolejności wykonywania grup ,2,..., $v$, czas wykonania wszystkich zadań wynosi

$$C = \max_{1\le l\le v}\left(\sum_{k=1}^{l} r_{1k} + \sum_{k=l}^{v} r_{2k}\right) + \sum_{k=1}^{v} d_k = \sum_{k=1}^{w} r_{1k} + \sum_{k=w}^{v} r_{2k} + \sum_{k=1}^{v} d_k\,, \tag{1}$$

gdzie dla przezbrojeń odłączonych od grup

$r_{1i} = \max\{0, S_1(B_i) + P_1(B_i) - S_2(B_i)\}$,

$r_{2i} = \max\{P_2(B_i), S_2(B_i) + P_2(B_i) - S_1(B_i) - P_1(B_i)\}$,

$d_i = \min\{S_2(B_i), S_1(B_i) + P_1(B_i)\}$, $i = 1, 2, \ldots, v$,

natomiast dla przezbrojeń przywiązanych do grup

$r_{1i} = S_1(B_i) + P_1(B_i),\quad r_{2i} = S_2(B_i) + P_2(B_i),\quad d_i = 0,\ i = 1,2,\dots,\nu.$

Grupę $w$, dla której spełniona jest równość w (1), nazywamy dalej *grupą krytyczną*.

Dla zadanego rozbicia rodzin na grupy $B_1, B_2, \dots, B_\nu$, problem GZ sprowadza się do znalezienia uszeregowania *zadań złożonych* $1,2,\dots,\nu$ z czasami wykonywania $P_1(B_k)$, $P_2(B_k)$ oraz czasami przezbrojeń $S_1(B_k)$, $S_2(B_k)$, $k = 1,2,\dots,\nu$, na maszynach, odpowiednio, 1, 2, dla którego czas wykonania całości jest minimalny. Optymalną kolejność zadań złożonych (grup) można znaleźć w wielomianowym czasie stosując regułę Johnsona, która w tym przypadku przybiera następującą postać.

*Reguła Johnsona:* Istnieje optymalne uszeregowanie grup $1,2,\dots,\nu$, w którym grupa $k$ poprzedza grupę $l$, jeżeli $\min\{r_{1k}, r_{2l}\} \le \min\{r_{1l}, r_{2k}\}$.

Uszeregowanie spełniające warunek reguły Johnsona można łatwo uzyskać w następujący sposób: niech $A = \{i: r_{1i} \le r_{2i},\ 1 \le i \le \nu\}$ oraz $B = \{i: r_{1i} > r_{2i},\ 1 \le i \le \nu\}$, najpierw ustaw grupy ze zbioru $A$ według niemalejących $r_{1i}$, następnie grupy ze zbioru $B$ według nierosnących $r_{2i}$. Otrzymany ciąg grup nazywany *uszeregowaniem Johnsona* [3] *grup*.

W przypadku przezbrojeń odłączonych od grup *Yoshida, Hitomi* [12] oraz *Sule* [8] podają regułę Johnsona opartą na parametrach: $r_{1i} = S_1(B_i) - S_2(B_i) + P_1(B_i)$, $r_{2i} = P_2(B_i)$, $i = 1,2,\dots,\nu$. Obydwa sformułowania są równoważne ze względu na minimalną wartość funkcji celu; postać (1) wybrano ze względu na to, że wszystkie występujące tam wielkości są nieujemne, co ułatwia analizę algorytmów aproksymacyjnych.

W pracy *Potts, Van Wassenhove* [6] sygnalizowany jest algorytm programowania dynamicznego dla wariantu rozważanego problemu (oznaczanego dalej przez GZJ), w którym zakłada się jednostkową dostępność zadań, przezbrojenia odłączone od grup oraz jednakową kolejność zadań na obydwu maszynach, o złożoności obliczeniowej $O(f^2 n^{2f})$; algorytm jest zatem wielomianowy ze względu na liczbę zadań oraz wykładniczy ze względu na liczbę rodzin. Procedura ta oparta jest na następującej własności problemu.

**Twierdzenie 1.** Istnieje harmonogram optymalny dla problemu GZJ, w którym zadania każdej rodziny uporządkowane są względem siebie zgodnie z regułą Johnsona.

Dowód można przeprowadzić metodą „odpowiednich przesunięć zadań nie spełniających warunku reguły Johnsona". *Kleinau* [4] pokazał, że gdy przezbrojenia są przywiązane do grup, powyższa własność nie jest prawdziwa. W związku z tym złożoność obliczeniowa tego przypadku dla ustalonej liczby rodzin pozostaje problemem otwartym. Własność Twierdzenia 1 nie zachodzi również dla problemu GZ.

**Twierdzenie 2.** Istnieje problem konkretny zagadnienia GZ, dla przyzbrojeń odłączonych do grup oraz dla przezbrojeń przywiązanych do grup, dla którego nie istnieje harmonogram optymalny, taki że zadania każdej rodziny uporządkowane są względem siebie zgodnie z regułą Johnsona.

**Dowód.** Rozważmy problem konkretny o następujących danych: $f = 2$, $n_1 = 1$, $n_2 = 5$ oraz

$a_{11} = 4$, $b_{11} = 8$, $s_{11} = 2$, $s_{21} = 2$,

$a_{21} = 1$, $b_{21} = 2$, $a_{22} = a_{23} = a_{24} = a_{25} = 2$, $b_{22} = b_{23} = b_{24} = b_{25} = 4$, $s_{12} = 2$, $s_{22} = 2$.

Rozważmy przypadek, gdy przezbrojenia są odłączone od grup. Dla hamonogramu

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

wartość funkcji celu wynosi 34, jednakże zadania $(2,2)$, $(2,1)$ rodziny $I_2$ nie spełniają warunku reguły Johnsona; rzeczywiście $\min\{a_{22}, b_{21}\} = \min\{2,2\} > \min\{1,4\} = \min\{a_{21}, b_{22}\}$. Tę samą wartość funkcji celu, również z naruszeniem reguły Johnsona, uzyskamy kładąc w miejsce zadania $(2,2)$ dowolne z pozostałych zadań tej rodziny z wyjątkiem $(2,1)$. Łatwo można sprawdzić, że dla pozostałych możliwych harmonogramów wartość funkcji celu nie jest mniejsza niż 35. Rozważania dla przezbrojeń przywiązanych do grup są identyczne. □

Ponieważ dla problemu GZ brak jest, jak dotąd, własności harmonogramu optymalnego porządkującej liniowo zadania wewnątrz rodzin, podejście tego samego typu co w pracy *Potts, Van Wassenhove* [6], prowadzące do algorytmu programowania dynamicznego o wielomianowej złożoności obliczeniowej dla ustalonej liczby rodzin, nie jest możliwe. Nie jest to wielką stratą, ponieważ ewentualny algorytm o złożoności $O(n^{2f+1})$, jaki można by zbudować na podstawie takiej własności, nie ma praktycznego znaczenia. Niemniej jednak złożoność problemu GZ dla ustalonej liczby rodzin pozostaje problemem otwartym. Oczywiście, w przypadku GZ z identycznymi zadaniami wewnątrz rodzin własność Twierdzenia 1 zachodzi dla obydwu typów przezbrojeń, co umożliwia konstrukcję algorytmu programowania dynamicznego o wielomianowej złożoności ze względu na liczbę zadań ($O(n^{2f+1})$). Złożoność obliczeniowa przypadku GZ z identycznymi zadaniami wewnątrz rodzin dla dowolnej liczby rodzin i dowolnej liczby zadań pozostaje problemem otwartym.

### 3. Algorytmy aproksymacyjne

Wobec NP-trudności problemu GZ oraz braku własności umożliwiających konstrukcję algorytmu dokładnego, pozwalającego rozwiązywać zagadnienia o praktycznych rozmiarach, algorytmy przybliżone mogą stanowić jedyne praktyczne podejście. W rozdziale tym przedstawimy dwa algorytmy przybliżone dla zagadnienia z przezbrojeniani odłączonymi od grup, obydwa o wskaźniku dokładności dla najgorszego przypadku 3/2. W dalszym ciągu przez $C_X$ oznaczać będziemy wartość funkcji celu otrzymaną przez algorytm aproksymacyjny X dla pewnego problemu konkretnego, a przez $C^*$, minimalną wartość funkcji celu.

Rozważmy najpierw algorytm **UJR** (uszeregowanie Johnsona rodzin) polegający na uporządkowaniu rodzin zadań, traktowanych jako grupy, zgodnie z regułą Johnsona. Dla tego zgrubnego podejścia zachodzi nierówność

$$\frac{C_{UJR}}{C^*} \le 2$$

dla każdego problemu konkretnego, i ograniczenie to jest ścisłe. Rzeczywiście, niech $(1,\ldots,w,\ldots,f)$ będzie uszeregowaniem Johnsona rodzin, gdzie $w$ jest rodziną krytyczną. Na podstawie (1) otrzymujemy

$$C_{UJR} - C^* \le \min\{r_{1w}, r_{2w}\}, \tag{2}$$

co wobec nierówności $\min\{r_{1w}, r_{2w}\} \le C^*$ daje pożądaną nierówność. Dla problemu z pojedynczą rodziną i zerowymi czasami przezbrojeń, składającą się z dwóch zadań z czasami $a_{11} = 1, b_{11} = K, a_{12} = K, b_{21} = 1$, otrzymujemy $C_{UJR} = 2K + 2$ oraz $C^* = K + 2$, co daje $C_{UJR}/C^* = (2K+2)/(K+2) \to 2$ dla $K \to \infty$.

Nierówność (2) sugeruje jedną z możliwych metod poprawy harmonogramu otrzymanego przez UJR, poprzez podział rodziny krytycznej $I_w$ na grupy; w algorytmie poniżej przyjmuje się podział na co najwyżej dwie grupy.

**Algorytm PRK** *(podział rodziny krytycznej)*
*Inicjalizacja:*
Znajdź uszeregowanie Johnsona rodzin $(1,\ldots,w,\ldots,f)$ (po odpowiednim przenumerowaniu rodzin). Wylicz czas wykonania zadań $C$, wyznacz rodzinę krytyczną $I_w$, podstaw $C_{PRK} := C$. Zadania wewnątrz rodziny krytycznej przenumeruj zgodnie z porządkiem wyznaczonym przez regułę Johnsona.

(1) Podstaw $B_1 := I_w$, $B_2 := \varnothing$, *Grupa_Krytyczna*$:= B_1$, wstaw $B_2$ do bieżącego uszeregowania bezpośrednio przed $B_1$; jeżeli grupa jest pusta, to przyjmujemy, że związane z nią przezbrojenia są zerowe.

(2) Jeżeli *Grupa_Krytyczna* $\neq B_1$, to STOP.

(3) Zadanie z najmniejszym numerem przenieś z $B_1$ do $B_2$.

(4) Wyznacz *Grupa_Krytyczna* oraz $C$ w nowym harmonogramie.

(5) Podstaw $C_{PRK} := \min\{C_{PRK}, C\}$ i przejdź do kroku (2).

Uszeregowanie Johnsona rodzin może być również poprawione poprzez podział rodzin różnych od rodziny krytycznej. Algorytm przedstawiony dalej, obok podziału rodziny krytycznej na grupy zgodnie z metodą zastosowaną w PRK, sukcesywnie dzieli pewne rodziny na dwie grupy i szereguje otrzymane grupy zgodnie z regułą Johnsona; podobne podejście dla odpowiednika GZ z jednostkową dostępnością zadań można znaleźć w [13]. Załóżmy, bez straty ogólności, że $(1,...,w,...,f)$ jest uszeregowaniem Johnsona rodzin z rodziną krytyczną $w$. Dla rodziny $i > w$ niech

$$I_i^1 = \{(i,j) \in I_i : a_{ij} \le b_{ij}\},\ I_i^2 = I_i \setminus I_i^1,\ \Delta_i = \max\{0, P_2(I_i^1) - s_{1i} - P_1(I_i^1)\},$$

zaś dla rodziny $i < w$

$$I_i^1 = \{(i,j) \in I_i : a_{ij} > b_{ij}\},\ I_i^2 = I_i \setminus I_i^1,\ \Delta_i = \max\{0, P_1(I_i^1) - s_{2i} - P_2(I_i^1)\}.$$

Mówimy, że *rodzina i jest podzielna*, jeżeli $\Delta_i > 0$ i $I_i^1 \neq \varnothing$ i $I_i^2 \neq \varnothing$. Można zauważyć, że jeżeli $i > w$ oraz $I_i \in B$, to $I_i^1 \in A$ i $I_i^2 \in B$, gdzie $A$ i $B$ podane są w definicji uszeregowania Johnsona. Zatem po podziale $I_i$ na $I_i^1$ i $I_i^2$ uszeregowanie grup wg reguły Johnsona może przesunąć $I_i^1$ przed rodzinę krytyczną $w$. Ponadto, jeżeli $\Delta_i > 0$, to rozbicie $I_i$ na $I_i^1$ i $I_i^2$ oraz przeniesienie $I_i^1$ przed rodzinę krytyczną $w$ może przynieść zmniejszenie czasu wykonania zadań; operacja taka nie spowoduje zmniejszenia czasu, jeżeli $\Delta_i = 0$. To samo dotyczy rodzin umiejscowionych na lewo od rodziny krytycznej. Szczegóły metody podaje następujący algorytm.

**Algorytm MJR** *(modyfikacja uszeregowania Johnsona rodzin)*

*Inicjalizacja:*

Znajdź uszeregowanie Johnsona grup $(1,\dots,w,\dots,f)$ (po odpowiednim przenumerowaniu rodzin). Wylicz czas wykonania zadań $C$, wyznacz rodzinę krytyczną $I_w$, podstaw $C_{MJR} := C$. Znajdź zbiór rodzin podzielnych $Q$.

*Etap I: (Podział rodzin różnych od rodziny krytycznej)*

(1) Podstaw *Grupa_Krytyczna*$:=I_w$.

(2) Jeżeli *Grupa_Krytyczna* $\neq I_w$, to STOP.

(3) Jeżeli $Q = \varnothing$, to przejdź do Etapu II.

(4) Rodzinę $I_l \in Q$ podziel na grupy $I_l^1$ i $I_l^2$, znajdź nowe uszeregowanie Johnsona grup, wyznacz jego *Grupa_Krytyczna* oraz $C$.

(5) Podstaw $C_{MJR} := \min\{C, C_{MJR}\}$, $Q := Q \setminus \{I_l\}$ i przejdź do kroku (2).

*Etap II: (Podział rodziny krytycznej)*

Zadania wewnątrz rodziny krytycznej przenumeruj zgodnie z porządkiem wyznaczonym przez regułę Johnsona i zastosuj algorytm PRK.

Złożoność obliczeniowa algorytmów PRK i MJR wynosi $O(n^2)$, a oszacowanie dokładności dla najgorszego przypadku podaje następujące twierdzenie.

**Twierdzenie 3.** Dla X=PRK, MJR,

$$\frac{C_X}{C^*} \leq \frac{3}{2}$$

dla każdego problemu konkretnego, i ograniczenie to jest ścisłe.

**Dowód** można znaleźć [17]. Algorytm aproksymacyjny PRK wydaje się być „minimalnym" schematem obliczeniowym, który gwarantuje dokładność 3/2. Jego rozszerzenie o podział rodzin różnych od rodziny krytycznej, MJR, nie przynosi jednak polepszenia dokładności w najgorszym przypadku. Sukcesywne polepszanie początkowego harmonogramu obraca się w obydwu algorytmach wokół początkowej rodziny krytycznej i kończy się w chwili, gdy inna rodzina lub grupa stanie się krytyczną. Pomimo że już w tym miejscu obydwie procedury iteracyjne doprowadzają do rozwiązania o oszacowaniu 3/2, można by je dalej kontynuować, dokonując kolejnych ulepszeń wokół nowej grupy krytycznej. Proces taki jest jednak trudny do oszacowania w sensie analizy najgorszego przypadku.

## LITERATURA

1. Centikaya F.C.: Lot streaming in a two-stage flow shop with set-up, processing and removal times. J. Opl. Res. , 45, 1994, pp.1445-1455.
2. Dobson G., Karmarkar U. S., Rummel J. L.: Batching to minimize flow times on one machine. Management Sci., 33, 1987, pp. 784-799.
3. Johnson S.M.: Optimal two- and three-stage production schedules with setup times included. Naval Res. Logistics Quarterly 1, 1954, pp. 61-68.
4. Kleinau U.: Two-machine shop scheduling problems with batch processing. Mathl. Comput. Modelling, 17, 1993, pp. 55-66
5. Potts C.N., Baker K.R.: Flow shop scheduling with lot streaming. Oper. Res. Letters, 8, 1989, pp. 297-303.
6. Potts C.N., Van Wassenhove L.N.: Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. J. Opl. Res. Soc., 43, 1992, pp. 395-406.
7. Sekiguchi Y.: Optimal schedule in a GT-type flow-shop under series-parallel precedence constraints. J. Oper. Res. Society of Japan 26, 1983, pp. 226-251.
8. Sule D.R.: Sequencing n jobs on two machines with setup, processing and removal times separated, Naval Res. Logistics Quarterly 29, 1982, pp. 517-519.
9. Vickson R.G., Alfredsson B.E.: Two- and three-machine flow shop scheduling problems with equal sized transfer batches. Int. J. Prod. Res., 30, 1992, pp.1551-1574.
10. Vickson R.G.: Optimal lot streaming for multiple products in a two-machine flow shop. European J. Oper. Res. , 1996 (w druku).
11. Webster S., Baker K.R.: Scheduling groups of jobs on a single machine. Working paper No. 307, 1995, The Amos Tuck School of Business Administration, Dortmouth College, Hanover.
12. Yoshida T., Hitomi K.: Optimal two-stage production scheduling with setup times separated. AIIE Transections, 11, 1979, pp.261-263.
13. Zdrzałka S.: Dwumaszynowe przepływowe problemy harmonogramowania z przezbrojeniami maszyn. Elektrotechnika, 14, 1995, pp. 503-509.
14. Zdrzałka S.: Analysis of approximation algorithms for single-machine sequencing with delivery times and sequence independent batch setup times. European J. Oper. Res., 80, 1995, pp. 371-180
15. Zdrzałka S.: A sequencing problem with family setup times. Discrete Appl. Math., 1996 (w druku)
16. Zdrzałka S.: Two-machine flowshop problem with family setup times. Raport nr 16/95, Instytut Cybernetyki Technicznej, Politechnika Wrocławska, 1995.
17. Zdrzałka S.: Batching and sequencing in a two-machine flowshop. Instytut Cybernetyki Technicznej, Politechnika Wrocławska, 1996.

Recenzent: Dr hab. inż. Mirosław Zaborowski, prof. Pol.Śl.

Wpłynęło do Redakcji do 30.06.1996 r.

**Abstract**

We consider a two-machine flowshop in which jobs are processed and transferred to the second machine in batches; that is a job in a batch becomes available for processing on the next machine when all jobs of the same batch are completed on the preceding machine. All jobs are partitioned into disjoint subsets called families, and a batch is a subset of a family. Before processing a batch, setup times are incurred on both machines, specific for the family to which the jobs of the batch belong; no setup times are required between executing the jobs of the same batch. Setups are either detached, when they can be performed on a downstream machine as soon as this machine is available, or attached, if they can start after completion of a batch on the preceding machine. The problem is to find a partition of families into batches and a processing order of batches (a schedule) that minimize the makespan. We present arguments showing that this problem is NP-hard, and that the computational complexity of the problem with fixed number of families is an open problem; in a contrary to the case with item availability of jobs and detached setups which is polynomial in the number of jobs and exponential in the number of families. Then we present two approximation algorithms with the worst-case performance ratio of 3/2. Both algorithms iteratively improve an initial schedule obtained by sequencing families, treated as batches, according to Johnson rule. The first one iteratively partitions the critical family into two batches. The second algorithm consists of two stages. In the first one some divisible families are succesively partitioned into two batches and batches are sequenced in accordance with Johnson rule, and in the second stage, the algorithm partitioning the critical family is applied.