ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2008
Seria: AUTOMATYKA z. 150 Nr kol. 1796

Zbigniew BUCHALSKI
Politechnika Wrocławska

# HEURYSTYCZNA PROCEDURA SZEREGOWANIA ZADAŃ W DYSKRETNYM SYSTEMIE PRODUKCYJNYM Z UWZGLĘDNIENIEM OGRANICZEŃ ZASOBOWYCH

**Streszczenie.** Praca dotyczy zagadnienia czasowo-optymalnego przydziału $n$ zadań niezależnych i zasobu nieodnawialnego do $m$ maszyn równoległych w dyskretnym systemie produkcyjnym. Dla zadanej funkcji czasu realizacji zadań sformułowano model matematyczny zagadnienia oraz zaprezentowano algorytm heurystyczny dla rozwiązania postawionego problemu. Przedstawiono wyniki badań numerycznych wykonanych na bazie zaproponowanego algorytmu heurystycznego.

# AN HEURISTIC PROCEDURE FOR TASKS SCHEDULING IN DISCRETE PRODUCTION SYSTEM WITH LIMITED RESOURCES ALLOCATION

**Summary.** In the paper problem of time-optimal allocation of $n$ independent tasks and nonrenewable resource's units to $m$ parallel machines in discrete production system is considered. For some tasks execution time function the mathematical model of this problem is formulated and an heuristic algorithm for solution this problem is presented. Some results of numerical experiments for basis of proposed heuristic algorithm are presented.

## 1. Wstęp

Odczuwalny od wielu lat intensywny rozwój równoległych systemów przetwarzania informacji pociągnął za sobą wzrost zainteresowania problematyką szeregowania zadań w systemach maszyn pracujących równolegle (niezależnie). Szczególnego znaczenia nabiera problem minimalizacji długości uszeregowania zadań na maszynach [1, 2, 6, 7]. Prowadzone od wielu lat badania problematyki czasowo-optymalnego szeregowania zadań i rozdziału zasobów dowiodły, że zadania optymalizacji dyskretnej i ciągłej należą do klasy problemów bardzo trudnych zarówno z teoretycznego, jak i obliczeniowego punktu widzenia i najczęściej należą do klasy proble-mów *NP*-zupełnych [8, 9, 10].

Problem szeregowania zadań na maszynach równoległych z równoczesnym rozdziałem zasobów bardzo często spotykany jest w różnego rodzaju złożonych procesach produkcyjnych. Czasy wykonywania zadań mogą być z góry zadane lub też mogą się zmieniać w zależności od współdzielonych zasobów. Zadania mogą być pewnymi

procesami dynamicznymi opisywanymi za pomocą skomplikowanych modeli matematycznych.

Wyniki teorii złożoności obliczeniowej oraz rozmiar problemów praktycznych w sposób jednoznaczny eliminują z rozważań algorytmy dokładne, pozostawiając do zastosowania praktycznego jedynie algorytmy heurystyczne umożliwiające rozwiązanie postawionych problemów w krótkim czasie z zadowalającą dokładnością. Badania nad algorytmami heurystycznymi dostarczającymi rozwiązań zagadnień, w których zastosowanie metod dokładnych jest nieefektywne lub wręcz niemożliwe, stanowią jedną z najszybciej rozwijających się gałęzi nauki.

W niniejszej pracy przedstawiono pewien algorytm heurystyczny wyznaczający czasowo-optymalne szeregowanie $n$ zadań niezależnych niepodzielnych i jednostek zasobu nieodnawialnego podzielnego w sposób ciągły do $m$ maszyn pracujących równolegle. Przedstawiono wyniki badań numerycznych przeprowadzonych na tym algorytmie dla losowo generowanych danych.

## 2. Sformułowanie problemu

Rozpatrzmy dyskretny system produkcyjny zawierający maszyny połączone równolegle. Dany jest zbiór $n$ zadań niezależnych ponumerowanych liczbami 1, 2, ..., $n$ i oznaczony przez $J$. Zadania te należy wykonać bez przerwań przy użyciu $m$ maszyn równoległych ze zbioru $M = \{1, 2, \ldots, m\}$. Każda maszyna w dowolnej chwili może wykonywać co najwyżej jedno zadanie oraz wykonywanie zadania nie może być przerywane. Realizacja każdego z zadań na maszynach musi następować niezwłocznie po zakończeniu wykonywania poprzedniego zadania (lub nastąpić w chwili t = 0, gdy zadanie realizowane jest jako pierwsze na jednej z maszyn). Każde zadanie może być wykonywane na dowolnej maszynie, a liczba zadań do wykonania jest większa od liczby maszyn ($n > m$).

Zakładamy, że dysponujemy $N$ jednakowymi jednostkami zasobu nieodnawialnego; $N \geq m$. Oznaczmy przez $u_k$ tę liczbę jednostek zasobu, która zostanie przydzielona $k$-tej maszynie. Zbiór zasobów oznaczmy przez $U = \{1, 2, \ldots, N\}$.

Zachodzą następujące ograniczenia zasobowe:

$$\sum_{k=1}^{m} u_k \leq N, \quad k \in M, \quad u_k \in U, \quad u_k \geq 1.$$

Czas wykonywania $i$-tego zadania na $k$-tej maszynie, jeżeli przydzielono jej $u_k$ jednostek zasobu, określony jest następującą funkcją:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, u_k \in U, k \in M, i \in J. \tag{1}$$

Parametry $a_{ik} > 0$, $b_{ik} > 0$ charakteryzują $i$-te zadanie i $k$-tą maszynę.

Postawiony do rozwiązania problem polega na znalezieniu takiego uszeregowania wszystkich $n$ zadań na $m$ maszynach i takiego przydziału jednostek zasobu do maszyn, aby minimalizować czas $T_{opt}$ zakończenia wykonywania całego zbioru zadań $J$.

## 3. Model formalny zagadnienia

Oznaczmy przez $J_1, J_2, \ldots, J_k, \ldots, J_m$ podzbiory zadań realizowanych kolejno na 1, 2, ..., $k$, ..., $m$-tej maszynie. Należy rozwiązać następujący problem minimalizacji czasu zakończenia wykonywania zbioru zadań $T_{opt}$:

$$T_{opt} = \min_{\substack{J_1, J_2, \ldots, J_m \\ u_1, u_2, \ldots, u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in J_k} T_i(u_k, k) \right\} \tag{2}$$

przy następujących ograniczeniach:

(i) $J_w \cap J_z = \emptyset; \quad w, z = 1, 2, \ldots, m, \quad w \ne z, \quad \bigcup_{k=1}^{m} J_k = J,$

(ii) $\sum_{k=1}^{m} u_k \le N; \quad u_k \in U, \quad k = 1, 2, \ldots, m.$

Postawione zadanie należy do klasy problemów *NP*-zupełnych. Problem, który należy rozwiązać, jest dość skomplikowany. Aby go rozwiązać, dokonamy relaksacji warunku $u_k \in U$, zastępując go warunkiem $1 \le u_k \le N$, $k = 1, 2, \ldots, m$. Po rozwiązaniu odpowiedniego zadania minimalizacji dyskretno-ciągłej zaokrągla się otrzymane wartości jednostek zasobu $u_k$ do liczb całkowitych dodatnich (patrz **Krok 14** algorytmu heurystycznego). Uwzględniając powyższą relaksację, minimalny czas $T_{opt}$ znajdziemy rozwiązując następujące zadanie minimalizacji dyskretno-ciągłej:

$$T_{opt} = \min_{\substack{J_1, J_2, \ldots, J_m \\ u_1, u_2, \ldots, u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in J_k} \widetilde{T}_i(u_k, k) \right\} \tag{3}$$

przy następujących ograniczeniach:

(i) $J_w \cap J_z = \emptyset; \quad w, z = 1, 2, \ldots, m, \quad w \ne z, \quad \bigcup_{k=1}^{m} J_k = J,$

(ii) $\sum_{k=1}^{m} u_k \le N; \quad u_k \ge 1, \quad k = 1, 2, \ldots, m,$

gdzie $\widetilde{T}_i : [0, N] \times \{1, 2, \ldots, m\} \to R^+$ jest rozszerzeniem funkcji

$T_i : \{1, 2, \ldots, N\} \times \{1, 2, \ldots, m\} \to R^+$ i określone jest przez funkcję:

$$\widetilde{T}_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in [1, N], \quad a_{ik} > 0, \quad b_{ik} > 0, \quad k \in M, \quad i \in J. \tag{4}$$

Rozwiązania optymalne zadania (3) oznaczmy przez $u_k^*, J_k^*$, $k = 1, 2, \ldots, m$. W celu znalezienia tych rozwiązań konieczne będzie wykorzystanie poniższego **Lematu**:

**Lemat**

Jeżeli $u_k^*, J_k^*$, $k = 1, 2, \ldots, m$ są rozwiązaniami optymalnymi zadania (3), to:

(i) $\sum_{k=1}^{m} u_k^* = N; \quad u_k^* > 0, \quad k : J_k^* \neq \varnothing, \quad k = 1,2,\ldots,m,$

$$u_k^* = 0, \quad k : J_k^* = \varnothing, \quad k = 1,2,\ldots,m,$$

(ii) $\sum_{i \in J_k^*} \widetilde{T}_i(u_k^*, k) = const; \quad k : J_k^* \neq \varnothing, \quad k = 1, 2, \ldots, m.$

Warunek (i) w **Lemacie** oznacza, że w przydziale czasowo-optymalnym jednostek zasobu i zadań do maszyn wykorzystuje się wszystkie jednostki zasobu, a warunek (ii), że czasy pracy tych maszyn, na których wykonywane są jakieś zadania, są identyczne.

Zdefiniujmy funkcję $F(J_1, J_2, \ldots, J_m)$ określoną dla $m$ zbiorów $J_1, J_2, \ldots, J_m$, dla których zachodzi ograniczenie (i) zadania (3). Wartość tej funkcji jest rozwiązaniem następującego układu równań:

$$\begin{cases} \sum_{i \in J_k} a_{ik} + \dfrac{\sum_{i \in J_k} b_{ik}}{u_k} = F(J_1, J_2, \ldots, J_m); k : J_k \neq \varnothing, k = 1,2,\ldots,m \\ \sum_{k : J_k \neq \varnothing} u_k = N; \quad u_k > 0; \quad k : J_k \neq \varnothing, k = 1,2,\ldots,m. \end{cases} \tag{5}$$

Gdy wykorzystamy **Lemat** oraz (5), zadanie minimalizacji (3) przyjmie ostateczną postać:

$$T_{opt} = \min_{J_1, J_2, \ldots, J_m} F(J_1, J_2, \ldots, J_m) \tag{6}$$

przy następujących ograniczeniach:

(i) $J_w \cap J_z = \varnothing; \quad w, z = 1, 2, \ldots, m, \quad w \neq z,$

(ii) $\bigcup_{k=1}^{m} J_k = J; \quad k = 1, 2, \ldots, m.$

Jeżeli $J_1^*, J_2^*, \ldots, J_m^*$ są rozwiązaniami zadania (6), to $u_k^*, J_k^*$ są rozwiązaniami zadania (3) gdzie:

$$T_{opt} = \min_{J_1, J_2, \ldots, J_m} F(J_1, J_2, \ldots, J_m). \tag{7}$$

## 4. Algorytm heurystyczny

Zakładamy, że maszyną najszybszą jest maszyna pierwsza ze zbioru $M$, a maszyną najwolniejszą jest maszyna $m$. Miarą szybkości realizacji zadania $i$ przez maszynę $k$ jest tzw. współczynnik podziału zasobu $\gamma$; $\gamma > 1$. Ostatniej maszynie M przydzielamy $u_m$ jednostek zasobu wg następującej zależności:

$$u_m = \frac{N}{1 + \sum_{k=1}^{m-1} [(m-k) \cdot \gamma]}. \tag{8}$$

Pozostałym maszynom przydzielamy jednostki zasobu wg następującej zależności:

$$u_k = (m-k) \cdot \gamma \cdot u_m; \quad k = 1, 2, ..., m-1. \tag{9}$$

Kolejne kroki algorytmu heurystycznego są następujące:

**Krok 1.** Oblicz czasy wykonywania zadań $T_i(u_k, k)$ na kolejnych maszynach wg wzoru (1) dla zadanej wartości $u_k = \frac{N}{m}$ i losowo generowanych parametrów $a_{ik}$, $b_{ik}$.

**Krok 2.** Uszereguj malejąco czasy wykonywania poszczególnych zadań wg czasów ich trwania i utwórz listę $L$ tych zadań.

**Krok 3.** Oblicz średni czas $T_{śr}$ wykonywania zadań przez każdą z maszyn wg wzoru:

$$T_{śr} = \frac{\sum_{i=1}^{n} T_i(u_k, k)}{m}; \qquad i \in J, k \in M, u_k = \frac{N}{m}.$$

**Krok 4.** Przydzielaj kolejno najkrótsze i najdłuższe zadania z listy $L$ do pierwszej wolnej maszyny aż do momentu, gdy suma czasów wykonywania zadań przydzielonych tej maszynie nie przekroczy czasu $T_{śr}$. Przydzielone zadania usuń z listy $L$.

**Krok 5.** Jeżeli są jeszcze maszyny, na których nie uszeregowano żadnych zadań, to wróć do **Kroku 4**. W przeciwnym wypadku przejdź do **Kroku 6**.

**Krok 6.** Uszereguj kolejne zadania z listy $L$ na maszynach, poczynając od pierwszej maszyny, a kończąc na maszynie $m$, a następnie usuń te zadania z listy $L$. Jeżeli lista $L$ się nie wyczerpała, to wykonaj **Krok 7**. W przeciwnym wypadku przejdź do **Kroku 8**.

**Krok 7.** Uszereguj kolejne zadania z listy $L$ na maszynach, poczynając od maszyny $m$, a na pierwszej kończąc, a następnie usuń te zadania z listy $L$. Jeżeli lista $L$ się nie wyczerpała, to wróć do **Kroku 6**. W przeciwnym wypadku przejdź do **Kroku 8**.

**Krok 8.** Oblicz sumaryczne czasy wykonywania zadań na poszczególnych maszynach dla uszeregowania zadań $J_1$, $J_2$, ..., $J_m$ utworzonego w **Krokach 3÷7** i dla zadanej liczby jednostek zasobu $u_k = \frac{N}{m}$ oraz podaj czas zakończenia $T_{opt}$ wykonywania wszystkich zadań będący najdłuższym z tych czasów.

**Krok 9.** Usuń najkrótsze zadanie z maszyny o najdłuższym czasie wykonywania zadań i przydziel je do maszyny o najkrótszym czasie wykonywania zadań.

**Krok 10.** Oblicz czas zakończenia wykonywania wszystkich zadań $T_{opt}$ po zamianie zadań w **Kroku 9**. Jeżeli czas ten ulegnie skróceniu, to wróć do **Kroku 9**. W przeciwnym wypadku cofnij ostatnio wykonaną czynność w **Kroku 9** i zakończ szeregowanie zadań na maszynach.

**Krok 11.** Dla zadanego współczynnika $\gamma$ przydziel liczbę jednostek zasobu $u_k$, $k \in M$ poszczególnym maszynom wyliczoną z zależności (8) i (9).

**Krok 12.** Dla uszeregowania zadań utworzonego **w Krokach 3÷10** i dla liczby jednostek zasobu $u_k$, $k \in M$ przydzielonych maszynom w **Kroku 11** oblicz czas zakończenia wykonywania wszystkich zadań $T_{opt}$.

**Krok 13**. Powtórz **Krok 11** i **Krok 12** dla następnych sześciu zwiększających się kolejno wartości współczynnika $\gamma$. Porównaj wartości czasów zakończenia wykonywania zadań $T_{opt}$ z kolejnych prób i wybierz najkrótszy z tych czasów.

**Krok 14**. Wyznacz dyskretne liczby zasobów $\hat{u}_k$, $k \in M$ według zależności:

$$\hat{u}_{\alpha(k)} = \begin{cases} \lfloor u_{\alpha(k)} \rfloor + 1; & k = 1, 2, \ldots, \Delta, \\ \lfloor u_{\alpha(k)} \rfloor \quad ; & k = \Delta + 1, \Delta + 2, \ldots, m, \end{cases}$$

gdzie $\Delta = N - \sum_{j=1}^{m} \lfloor u_j \rfloor$ oraz $\alpha$ jest permutacją elementów zbioru $M = \{1, 2, \ldots, m\}$ taką, że $u_{\alpha(1)} - \lfloor u_{\alpha(1)} \rfloor \geq u_{\alpha(2)} - \lfloor u_{\alpha(2)} \rfloor \geq \ldots \geq u_{\alpha(m)} - \lfloor u_{\alpha(m)} \rfloor$.

## 5. Wyniki badań numerycznych

Na bazie przedstawionego w pracy algorytmu heurystycznego przeprowadzono badania numeryczne. Algorytm heurystyczny poddano ocenie dla siedmiu zwiększających się kolejno wartości współczynnika podziału zasobu $\gamma$ ze zbioru $\{3, 6, \ldots, 21\}$. Dla zadanej liczby zadań $n$ = 25, 50, 75, 100, liczby maszyn $m$ = 2, 4, 6, 8 i liczby jednostek zasobu $N$ = 10.000 wylosowano parametry $a_{ik}$, $b_{ik}$ ze zbioru $\{2, 4, \ldots, 30\}$ przez generator o jednostajnym rozkładzie prawdopodobieństwa. Dla każdej kombinacji $n$ i $m$ wygenerowano 40 instancji. Rezultaty analizy porównawczej podanego w pracy algorytmu heurystycznego i algorytmu *LPT* przedstawione zostały w tabeli 1.

Tabela 1

Wyniki analizy porównawczej algorytmu heurystycznego i algorytmu *LPT*

| $n/m$ | Liczba instancji, dla których: | | | $\Delta^H$ | $S^H$ | $S^{LPT}$ |
|---|---|---|---|---|---|---|
| | $T_{opt}^H < T_{opt}^{LPT}$ | $T_{opt}^H = T_{opt}^{LPT}$ | $T_{opt}^H > T_{opt}^{LPT}$ | % | sec | sec |
| 25/2 | 22 | 1 | 17 | 2,1 | 1,3 | 1,0 |
| 50/2 | 22 | 2 | 16 | 2,3 | 2,3 | 1,6 |
| 75/2 | 23 | 0 | 17 | 3,1 | 3,9 | 3,1 |
| 100/2 | 24 | 3 | 13 | 4,2 | 5,6 | 5,1 |
| 25/4 | 20 | 2 | 18 | 2,0 | 1,8 | 1,4 |
| 50/4 | 22 | 1 | 17 | 2,8 | 3,6 | 3,0 |
| 75/4 | 24 | 1 | 15 | 3,6 | 5,7 | 4,6 |
| 100/4 | 26 | 4 | 10 | 4,3 | 7,1 | 6,1 |
| 25/6 | 20 | 2 | 18 | 1,9 | 1,9 | 1,3 |
| 50/6 | 22 | 2 | 16 | 2,6 | 3,9 | 2,8 |
| 75/6 | 23 | 2 | 15 | 3,8 | 5,3 | 4,4 |
| 100/6 | 25 | 1 | 14 | 4,6 | 7,8 | 6,9 |
| 25/8 | 21 | 0 | 19 | 1,8 | 2,1 | 1,7 |
| 50/8 | 21 | 3 | 16 | 2,4 | 4,2 | 3,3 |
| 75/8 | 24 | 1 | 15 | 4,0 | 6,4 | 4,9 |
| 100/8 | 27 | 0 | 13 | 4,8 | 8,7 | 7,3 |

W tabeli 1 występują następujące wielkości:

$n$ - liczba zadań, $m$ - liczba maszyn,

$T_{opt}^{H}$ - czas zakończenia wykonywania wszystkich zadań ze zbioru $J$ przy wykorzystaniu algorytmu heurystycznego,

$T_{opt}^{LPT}$ - czas zakończenia wykonywania wszystkich zadań ze zbioru $J$ przy wykorzystaniu algorytmu *LPT*,

$\Delta^{H}$ - średnia procentowa poprawa czasu $T_{opt}^{H}$ w stosunku do czasu $T_{opt}^{LPT}$ wyrażona następującym wzorem: $\Delta^{H} = \frac{T_{opt}^{LPT} - T_{opt}^{H}}{T_{opt}^{H}} \cdot 100\%$,

$S^{H}$ - średni czas obliczeń dla algorytmu heurystycznego,

$S^{LPT}$ - średni czas obliczeń dla algorytmu *LPT*.

## 6. Uwagi końcowe

Przedstawione w poprzednim punkcie pracy wyniki badań numerycznych wykazały, że jakość szeregowania zadań na równoległych maszynach na bazie zaproponowanego w pracy algorytmu heurystycznego uległa poprawie w stosunku do szeregowania za pomocą znanego z literatury algorytmu *LPT*. Kilkuprocentowa poprawa czasu $T_{opt}^{H}$ w stosunku do $T_{opt}^{LPT}$ może być zachętą do dalszych prac nad efektywnymi algorytmami heurystycznymi. W przypadku gdyby dla algorytmu *LPT* czasy wykonywania zadań były zależne od ilości przydzielonego zasobu, wyniki analizy porównawczej przedstawione w tabeli 1 mogłyby być nieco mniej obiecujące.

Zaprezentowany w pracy algorytm heurystyczny może służyć zarówno do rozdziału operacji na stanowiska produkcyjne wyposażone w odpowiednie maszyny w dyskretnych systemach produkcyjnych, jak i do szeregowania programów w wieloprocesorowych systemach komputerowych [3, 4, 5].

BIBLIOGRAFIA

1. Błażewicz J., Dell'Olmo P., Drozdowski M., Speranza M. G.: Scheduling multiprocessor tasks on three dedicated processors. Information Processing Letters 41, 1992, p. 275-280.
2. Boctor F. F.: A new and efficient heuristic for scheduling projects with resources restrictions and multiple execution models. European Journal of Operational Research, Vol. 90, 1996, p. 349-361.
3. Buchalski Z.: An heuristic solution procedure to minimize the total processing time of programs in multiprocessing computer system. Information Systems Architecture and Technology ISAT 2005, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005, p. 20-26.
4. Buchalski Z.: A Program Scheduling Heuristic Algorithm in Multiprocessing Computer System with Limited Memory Pages. Polish Journal of Environmental Studies, Vol. 15, No. 4C, 2006, p. 26-29.

5. Buchalski Z.: An Heuristic Algorithm for Solving the Scheduling Problem in Multiprocessing Computer System. Polish Journal of Environmental Studies, Vol. 16, No. 4A, 2007, p. 44-48.
6. Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, Vol. 53, 1991, p. 317-325.
7. Janiak A., Kovalyov M.: Single machine scheduling subject to deadlines and resources dependent processing times. European Journal of Operational Research, 1996, Vol. 94, p. 284-291.
8. Józefowska J., Węglarz J.: On a methodology for discrete-continuous scheduling. European Journal of Operational Research, Vol. 107, 1998, p. 338-353.
9. Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Rozwiązywanie dyskretno-ciąglych problemów rozdziału zasobów przez dyskretyzację zasobu ciągłego. Zeszyty Naukowe Politechniki Śląskiej Nr 1474, seria Automatyka, Gliwice 2000, z. 129, s. 221-229.
10. Kubale M., Giaro K.: Złożoność zwartego szeregowania zadań jednostkowych w systemie otwartym, przepływowym i mieszanym. Uczelniane Wydawnictwo Naukowo-Dydaktyczne AGH, S. Automatyka, półrocznik, tom 5, zeszyt 1-2, Kraków 2001, s. 329-334.

Recenzent: Prof. dr hab. inż. Tadeusz Sawik

**Abstract**

In the paper problem of time-optimal allocation of $n$ independent tasks and nonrenewable resource's units to $m$ parallel machines in discrete production system is considered. We assume, that tasks are nonpreemtive and number of tasks is greater then number of machines ($n > m$). Each of machine may execute every of $n$ tasks and multimachines system has $N$ nonrenewable resource's units. During execution of all $n$ tasks the number $u_k$ resource's units is allocated to the $k$-th machine; $\sum_{k=1}^{m} u_k \le N$.

Results obtained in the paper are for some execution time function:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, m, \quad u_k \in U,$$

where $a_{ik} > 0$, $b_{ik} > 0$ are parameters characterized $i$-th task and $k$-th machine, $u_k$ is number of resource's units allocated to $k$-th machine. The purpose of optimization is to find such a schedule of $n$ tasks on $m$ parallel machines and such an allocation of limited resource's units among machines that schedule length criterion is minimized.

Because our problem belongs to the class of $NP$-complete problems we propose an heuristic algorithm, which employs some problem properties. Some results of executed numerical experiments for basis of proposed heuristic algorithm are presented.