AN HEURISTIC SOLUTION PROCEDURE TO MINIMIZE THE TOTAL PROCESSING TIME OF TASKS IN PARALLEL MACHINES SYSTEM

Zbigniew BUCHALSKI

Summary: This paper present results of research on the problem of time-optimal tasks scheduling and resources allocation in parallel machines system. We consider an parallel machines system consisting of m parallel machines. This system can execute n tasks. We assume that all n tasks are independent and number of tasks is greater than number of machines. We also assume that is constancy of resources allocation in execution time all tasks set. For some tasks processing time function the mathematical model of this problem is formulated. Because our problem belongs to the class of *NP*-complete problems we propose an heuristic algorithm for solution this problem. Some results of executed numerical experiments for basis of proposed heuristic algorithm are presented.

Keywords: parallel machines system, tasks scheduling, resources allocation.

1. Introduction

In many production processes there arise problems of scheduling a set of tasks on parallel machines with simultaneous resources allocation. By resources we understand arbitrary means tasks compete for. They can be of a very different nature, e.g. energy, tools, money, manpower. Tasks can have a variety of interpretation starting from machining parts in manufacturing systems up to processing information in computer systems. A structure of a set of tasks and different criteria which measure the quality of the performance of a set of tasks can be taken into account [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

The time-optimal problem of tasks scheduling and resources allocation are intensive developing, as in [11, 12, 13]. The further development of the research has been connected with applications, among other things, in many production processes and in multiprocessing computer systems, as in [14, 15, 16, 17, 18, 19, 20, 21].

These tasks scheduling and resources allocation problems in parallel machines systems are very complicated problems and belongs to the class of *NP*-complete problems. Therefore in this paper we propose an heuristic algorithm for solving of a optimization problem. The purpose of optimization is to find such a schedule of tasks on parallel machines and such an allocation of limited nonrenewable schedule length criterion is minimized.

2. Description of the problem

We consider an parallel machines system (as shown in Fig. 1) having m parallel machines and let $M = \{1, 2, ... , k, ... , m\}$ be the set of these machines..This parallel machines system can execute n independent tasks from the tasks set $J = \{1, 2, \dots , i, ... , n\}$.

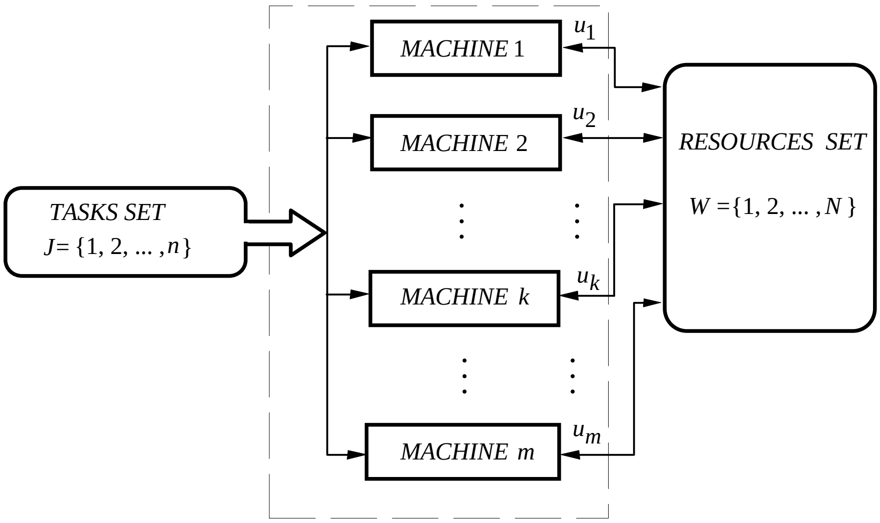

Fig. 1. Parallel machines system

We assume about this system that:

- each the machine may execute every one of n tasks,
- number of tasks is greater than number of machines ($n > m$),
- the system has N available nonrenewable resources, which are denoted as set $W = \{1, 2, \ldots , N\}$, $N \geq m$,
- during execution of all n tasks, the number of u_k resources is allocated to the k-th machine; $\sum_{k=1}^{m} u_k \leq N$. Each machine may use only allocated to his resources and is constancy of resources allocation in execution time all tasks set.

Results obtained in this paper are for some processing time function:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in W, \quad 1 \leq k \leq m, \quad i \in J, \tag{1}$$

where $a_{ik} > 0$, $b_{ik} > 0$ – parameters characterized i-th task and k-th machine, u_k is number of resources allocated to k-th machine.

This tasks scheduling and resources allocation problem in parallel machines system can be formulated as follows: find scheduling of n independent tasks on the m machines running parallel and partitioning of N resources among m machines, that schedule length criterion is minimized.

3. Formulation of optimization problem

Let $J_1, J_2, ..., J_k, ..., J_m$ be defined as subsets of tasks, which are processing on the machines 1, 2,..., *k*,..., *m*. The problem is to find such subsets $J_1, J_2, ..., J_k, ..., J_m$ and such resources numbers $u_1, u_2, ..., u_k, ..., u_m$, which minimize the T_{opt} of all set *J*:

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

under the following assumptions:

$$(i) \quad J_s \cap J_t = \varnothing, \quad s, t = 1,2,..., m, \quad s \neq t, \quad \bigcup_{k=1}^{m} J_k = J,$$

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

$$(iii) \quad u_1, u_2, ..., u_m - \text{positive integer.}$$

The assumption (iii) is causing, that the stated problem is very complicated therefore to simplify the solution our problem we assume in the sequel that the resources are continuous. The numbers of resources obtained by this approach are rounded to the integer numbers (look **Step 12** in the heuristic algorithm) and finally our problem can formulated as following minimizing problem:

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

under the following assumptions:

$$(i) \quad J_s \cap J_t = \varnothing, \quad s, t = 1,2,..., m, \quad s \neq t, \quad \bigcup_{k=1}^{m} J_k = J,$$

$$(ii) \quad \sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad k = 1,2,..., m,$$

where $\tilde{T}_i : [0, N] \times \{1,2,..., m\} \to R^+$ is the extension of function $T_i : \{1,2,..., N\} \times \times \{1,2,..., m\} \to R^+$ and formulated by following function:

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

Taking into account properties of the function $\tilde{T}_i(u_k, k)$, it is easy to show the truth of the following theorem:

Theorem 1.

If the sets $u_k^*,\ J_k^*,\ k = 1,2,\dots,m$ are a solutions of minimizing problem (3), then:

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

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

$$(ii) \quad \sum_{i \in J_k^*} \tilde{T}_i(u_k^*, k) = \text{const}, \quad k : J_k^* \ne \varnothing, \quad k = 1,2,\dots,m.$$

We define function $F(J_1 J_2, \dots, J_m)$, which value is solution following system of equations:

$$\begin{cases} \sum\limits_{i \in J_k} a_{ik} + \dfrac{\sum\limits_{i \in J_k} b_{ik}}{u_k} = F\left(J_1, J_2, \dots, J_m\right), & k : J_k \ne \varnothing, \quad k = 1, 2, \dots, m \\ \sum\limits_{k : J_{k \ne \varnothing}} u_k = N; \quad u_k > 0 & k : J_k \ne \varnothing, \quad k = 1, 2, \dots, m \end{cases} \tag{5}$$

On the basis of **Theorem 1** and (5), problem (3) will be following:

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

under the assumptions:

$$(i) \quad J_s \cap J_t = \varnothing; \quad s, t = 1, 2, \dots, m, \quad s \ne t$$

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

If $J_1^*, J_2^*, ..., J_m^*$ are solutions of problem (6), it u_k^*, J_k^* are solutions of problems (3), where:

$$u_k^* = \begin{cases} \dfrac{\sum_{i \in J_k^*} b_{ik}}{F(J_1^*, J_2^*, ..., J_m^*) - \sum_{i \in J_k^*} a_{ik}}; & k : J_k^* \neq \emptyset, \quad 1 \leq k \leq m, \\ 0 \qquad ; & k : J_k^* = \emptyset, \quad 1 \leq k \leq m. \end{cases} \tag{7}$$

4. The heuristic algorithm

Resources allocation formula to the machines is a follows:

- we assume that the first machine from the set M has highest speed and the last machine from the set M has least speed,
- we assume if be of assistance in resources allocation so-called partition of resources coefficient γ; γ>1,
- to the last m machine is allocated u_m resources according to the following formula:

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

- to the remaining machines are allocated resources according to the formula:

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

The proposed heuristic algorithm is as follows:

Step 1. For given $u_k = \dfrac{N}{m}$ and random generate parameters a_{ik}, b_{ik} calculate the processing times of tasks $T_i(u_k, k)$ according to the formula (1).

Step 2. Schedule tasks from longest till shortest processing times of tasks $T_i(u_k, k)$ and formulate the list L of these tasks.

Step 3. Calculate mean processing time T_{mean} every machines according to follows formula:

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

Step 4. Allot in turn longest and shortest tasks which the list L to the first empty machine for the moment, when the sum of processing times these tasks to keep within the bounds of time T_{mean} and eliminate these tasks from the list L.

Step 5. If there are also empty machines, where are not schedule tasks go to the **Step 4**. If is not empty machine go to the **Step 6.**

Step 6. Remainder of tasks from the list L allott to the machines according to the algorithm *LPT* (Longest Processing Time) to moment of finish the list L.

Step 7. Calculate total processing time T_{opt} of all tasks for scheduling $J_1, J_2, \ldots, J_m$, which was determined in the **Steps 3÷6** and for given numbers of resources $u_k = \frac{N}{m}$.

Step 8. For given partition of resources coefficient γ allot resources u_k, $k = 1, 2, \ldots, m$ to succeeding machines as calculated according formula (8) and (9).

Step 9. For tasks scheduling which was determined in **Steps 3÷6** and for numbers of resources u_k, $k = 1, 2, \ldots, m$ allotted to machines in the **Step 8** calculate total processing time T_{opt} of all tasks.

Step 10. Repeat the **Step 8** and **Step 9** for the next six augmentative succeeding another values of coefficient γ.

Step 11. Compare values of total processing times T_{opt} of all tasks calculated after all samples with different values of coefficient γ (**Steps 8÷10**). Take this coefficient γ when total processing time T_{opt} of all tasks is shortest.

Step 12. Find the discrete numbers $\hat{u}_k$ of resources, $k = 1, 2, \ldots, m$ according to follows dependence:

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

where $\Delta = N - \sum_{j=1}^{m} \lfloor u_j \rfloor$ and α is permutation of elements of set M ={1, 2, ... , m} such, that: $u_{\alpha(1)} - \lfloor u_{\alpha(1)} \rfloor \geq u_{\alpha(2)} - \lfloor u_{\alpha(2)} \rfloor \geq \ldots \geq u_{\alpha(m)} - \lfloor u_{\alpha(m)} \rfloor$.

5. Results of numerical experiments

On the base this heuristic algorithm were obtained results of numerical experiments for seven another values of coefficient γ = 4, 8, 12, 16, 20, 24, 28. For the definite number of tasks n = 50, 100, 150, 200, 250, number of machines m = 6, 12, 18, 24, 30, 36, 42 and number of resources N =10.000 were generated parameters a_{ik}, b_{ik} from the set {0.2, 0.4, ... , 9.8, 10.0}. For each combination of n and m were generated 40 instances. The results of comparative analysis of heuristic algorithm proposed in this paper and the algorithm *LPT* are showed in the Table 1.

Tab.1. The results of comparative analysis of heuristic algorithm and algorithm *LPT*

n/m	number of instances, when:			Δ^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
50/6	21	1	18	2,4	2,4	2,0
100/6	22	0	18	2,7	4,6	3,9
150/6	22	1	17	3,6	8,9	8,1
200/6	23	2	15	4,8	10,7	9,4
250/6	24	1	15	5,4	12,7	10,6
50/12	20	2	18	2,5	3,2	2,9
100/12	21	2	17	3,6	5,1	4,3
150/12	24	0	16	3,8	10,1	8,2
200/12	25	2	13	4,3	12,9	10,8
250/12	26	1	13	5,7	15,2	12,3
50/18	20	0	20	2,5	3,2	2,7
100/18	21	2	17	3,8	7,2	6,1
150/18	22	1	17	4,5	10,2	8,9
200/18	23	2	15	4,7	13,9	11,2
250/18	25	2	13	6,0	15,1	12,9
50/24	20	1	19	2,3	3,9	3,2
100/24	21	0	19	3,7	6,8	5,9
150/24	23	1	16	4,8	11,2	9,9
200/24	24	1	15	5,4	14,9	12,7
250/24	25	3	12	5,9	17,2	15,2
50/30	20	1	19	2,8	4,8	3,8
100/30	22	2	16	3,7	7,8	5,9
150/30	24	2	14	4,6	11,3	9,4
200/30	24	0	16	5,4	15,2	13,3
250/30	25	1	14	6,3	17,3	14,8
50/36	21	0	19	2,7	6,1	4,9
100/36	22	2	16	3,9	8,9	6,8
150/36	24	3	13	4,8	10,9	8,8
200/36	25	0	15	5,8	15,7	13,9
250/36	26	2	12	7,2	18,2	16,3
50/42	21	1	18	4,1	5,9	4,9
100/42	22	1	17	4,4	8,9	7,0
150/42	23	2	15	5,5	11,8	10,1
200/42	27	1	12	6,1	16,6	14,5
250/42	29	2	11	7,6	19,2	16,8

In the Table 1 there are the following designations:

n – number of tasks,

m – number of machines,

T_{opt}^{H} – total processing time of all set of tasks J for the heuristic algorithm,

T_{opt}^{LPT} – total processing time of all set of tasks J for the algorithm *LPT*,

Δ^{H} – the mean value of the relative improvement T_{opt}^{H} in relation to T_{opt}^{LPT} :

$$\Delta^{H} = \frac{T_{opt}^{LPT} - T_{opt}^{H}}{T_{opt}^{H}} \cdot 100\%,$$

S^{H} – the mean time of the numerical calculation for the heuristic algorithm,

S^{LPT} – the mean time of the numerical calculation for the algorithm *LPT*.

6. Final remarks

Numerical experiments presented above show, that quality of tasks scheduling in parallel machines system based on the proposed in this paper heuristic algorithm increased in compare with simple *LPT* algorithm. The few percentages improvement of time T^{H} in compare with T^{LPT} can be the reason why heuristic algorithms researches will be successfully taken in the future.

Application of presented in this paper heuristic algorithm is especially good for production systems with great number of tasks because in this case the Δ^{H} improvement is the highest. Proposed heuristic algorithm can be used not only to task scheduling in parallel machines but also to programs scheduling in multiprocessing computer systems.

References

1. Błażewicz J., Dell’Olmo P., Drozdowski M., Speranza M. G.: Scheduling multiprocessor tasks on three dedicated processors. Information Processing Letters 41, 1992, pp. 275-280.
2. Błażewicz J., Ecker K., Schmidt G., Węglarz J.: Scheduling in Computer and Manufacturing Systems. Springer-Verlag, Berlin-Heidelberg, 1993.
3. Boctor F. F.: A new and efficient heuristic for scheduling projects with resources restric-tions and multiple execution models. European Journal of Operational Research, Vol. 90, 1996, pp. 349-361.
4. Cheng J., Karuno Y., Kise H.: A shifting bottleneck approach for a parallel-machine flow-shop scheduling problem, Journal of the Operational Research Society of Japan, Vol. 44, No. 2, 2001, pp. 140-156.
5. Gupta J.N.D., Hariri A.M.A., Potts C.N.: Scheduling a two-stage hybrid flow shop with parallel machines at the first stage, Annals of Operations Research, Vol. 69, No. 0, 1997, pp. 171-191.
6. Hoogeveen J.A., Lenstra J.K., Veltman B.: Preemptive scheduling in a two-stage multiprocessor flow shop in NP-hard, European Journal of Operational Research, Vol. 89, No.1, 1996, pp. 172-175.

7. Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, Vol. 53, 1991, pp. 317-325.
8. Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Discrete-continuous scheduling to minimize maximum lateness, Proceedings of the Fourth International Symposium on Methods and Models in Automation and Robotics MMAR’97, Międzyzdroje, Poland, 1997, pp. 947-952.
9. Józefowska J., Węglarz J.: On a methodology for discrete-continuous scheduling. European Journal of Operational Research, Vol. 107, 1998, pp. 338-353.
10. Nowicki E., Smutnicki C.: The flow shop with parallel machines. A tabu search approach. European Journal of Operational Research 106, 1998, pp. 226-253.
11. Buchalski Z.: Minimising the Total Processing Time for the Tasks Scheduling on the Parallel Machines System. Proc. of the 12th IEEE International Conference on Methods and Models in Automation and Robotics, Domek S., Kaszyński R. (Eds.), Międzyzdroje, Poland, MMAR 2006, 28-31 August 2006, pp. 1081-1084.
12. Janiak A., Kovalyov M.: Single machine scheduling subject to deadlines and resources dependent processing times. European Journal of Operational Research, 1996, Vol. 94, pp. 284-291.
13. Ng C.T., Cheng E.T.C., Janiak A., Kovalyov M.Y.: Group scheduling with controllable setup and processing times: minimizing total weighted completion time, Ann. Oper. Res., 133, 2005, pp. 163-174.
14. Bianco L., Błażewicz J., Dell'Olmo P., Drozdowski M.: Linear algorithms for preemptive scheduling of multiprocessor tasks subject to minimal lateness. Discrete Applied Mathematics, 72, 1997, pp. 25-46.k
15. Błażewicz J., Drozdowski M., Werra D., Węglarz J.: Scheduling independent multiprocessor tasks before deadlines. Discrete Applied Mathematics 65 (1-3), 1996, pp. 81-96.
16. Błażewicz J., Liu Z.: Scheduling multiprocessor tasks with chain constraints. European Journal of Operational Research, 94, 1996, pp. 231-241.
17. Brah S.A., Loo L.L.: Heuristics for scheduling in a flow shop with multiple processors, European Journal of Operational Research, Vol. 113, No. 1, 1999, pp. 113-122.
18. 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, pp. 20-26.
19. 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, pp. 26-29.
20. Buchalski Z.: An Heuristic Algorithm for Solving the Scheduling Problem in Multiprocessing Computer System. Polish Journal of Environmental Studies, Vol. 16, No. 4A, 2007, pp. 44-48.
21. Węglarz J.: Multiprocessor scheduling with memory allocation - a deterministic approach. IEEE Trans. Comput., C-29, 1980, pp.703-710.

Ph.D. Zbigniew BUCHALSKI
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
50-372 Wrocław, Janiszewskiego 11/17
tel.: (0-71) 320 32 92
e-mail: zbigniew.buchalski@pwr.wroc.pl