MATHEMATICAL MODELS AND HEURISTIC ALGORITHMS FOR SCHEDULING IDENTICAL PARALLEL MACHINES WITH A SERVER: A SURVEY

Keramat Hasani[1], Svetlana A. Kravchenko[2], Frank Werner[3]
[1]Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran
hasani@iau-malayer.ac.ir
[2]United Institute of Informatics Problems, Surganova St. 6, 220012 Minsk, Belarus
kravch@newman.bas-net.by
[3]Fakultät fur Mathematik,Otto-von-Guericke-Universität Magdeburg, Postfach 4120, 39016 Magdeburg, Germany
frank.werner@ovgu.de

In this paper, the problem of scheduling jobs on a set of identical parallel machines with a single server (robot) is considered. Models having these features are common in network computing. We survey recent advances in the development of mathematical models and heuristic algorithms for such problems. The major focus is on own results by the authors obtained in the last years.

Introduction

The problem considered can be formulated as follows. A set of *n* independent jobs $\{J_1,\ldots,J_n\}$ has to be scheduled on two identical parallel machines M_1 and M_2 without preemptions. Each machine can process at most one job at a time. For each job J_j, there is given its processing time p_j. Before processing any job, say J_j, on a machine, say M_q, job J_j has to be loaded on M_q by a single server. During such a setup, the job J_j, the machine M_q and the server are involved into this process for s_j time units, i.e., no other job can be processed on this machine or can be loaded by the server during this setup.

A schedule specifies each job on one machine for one time interval. Throughout such an interval the prescribed job has to be loaded and processed on the prescribed machine. A schedule is feasible if no two intervals on the same machine overlap. Given a schedule s, one can compute for any job J_j, the completion time C_j. The optimality criteria we are focussing on in this study are the minimization of the maximum completion time or makespan, $C_{\max} = \max\{C_1,\ldots,C_n\}$, the total completion time $\sum_{j=1}^{n} C_j$ and the total

weighted completion time $\sum_{j=1}^{n} w_j C_j$. Using the common notation, the problems $P2,S1||C_{\max}$, $P2,S1||\sum C_j$ and $P2,S1||\sum w_j C_j$ are considered. Problem $P2,S1||C_{\max}$ is strongly NP-hard since already problem *P2,S1* | s_j = *s* | C_{max} is strongly NP-hard [4]. Problem $P2,S1||\sum C_j$ is strongly NP-hard since already problem $P2,S1|s_j = s|\sum C_j$ is strongly NP-hard [4].

Mathematical Models

For problem $P2,S1||C_{\max}$, two mixed integer linear programming (MILP) formulations and two variants of a branch-and-price scheme were developed in [3]. Computational experiments have shown that for small instances with $n \in \{8, 20\}$, one of the MILP formulations was the best whereas for the larger instances with $n \in \{50, 100\}$, the branch-and-price scheme worked better.

For the same problem, three MILP models were proposed in [5]. The first model M0 is based on the idea of using the linear order of loading in the optimal schedule. The other two models M1 and M2 are based on the idea of the block structure of an optimal schedule. We briefly describe the latter two models.

It is easy to see that any schedule for the problem $P2,S1||C_{max}$ can be considered as a unit of blocks $B_1,\ldots,B_z$, where $z \le n$. Each block B_k can be completely defined by the first level job J_a and a set of second level jobs $\{J_{a1},\ldots,J_{ak}\}$, where inequality $p_a \ge s_{a1}+\ldots+s_{ak}+p_{a1}+\ldots+p_{ak}$ holds. The binary variable $B_{k,f,j}$ is used for a block. We have $B_{k,f,j}=1$ if job J_j is scheduled in level f in the k-th block, otherwise $B_{k,f,j}=0$. The index $k=1,\ldots,n$ indicates the serial number of the block. The index $f \in \{1,2\}$ indicates the level, i.e., we have $f=1$ if the level is the first one, and $f=2$ if the level is the second one. The index $j=1,\ldots,n$ indicates the job. ST_k and PT_k indicate the loading and processing part of block B_k, respectively. $ch[j]$ denotes the maximal number of second level jobs for the same block. Model M1 can be described as follows.

$$F + \sum_{x=1}^{n}\sum_{j=1}^{n}(s_j + p_j)B_{x,2,j} \to \min$$

$$\sum_{k=1}^{n}\sum_{y=1}^{2}B_{k,y,j} = 1, \quad \sum_{j=1}^{n}B_{k,1,j} \le 1 \text{ for any } k = 1..n$$

$$ST_k \ge \sum_{j=1}^{n} s_j B_{k,1,j}, \; PT_k \ge \sum_{j=1}^{n} p_j B_{k,1,j} - \sum_{j=1}^{n}(s_j + p_j)B_{k,2,j}$$

$$st_j + ST_j \le st_{j+1},$$

$$st_j + ST_j + PT_j \le st_{j+2}$$
$$F \ge st_n + ST_n + PT_n,$$
$$F \ge st_{n-1} + ST_{n-1} + PT_{n-1}$$
$$B_{x,2,1} + \ldots + B_{x,2,n} \le ch[1]B_{x,1,1} + \ldots + ch[n]B_{x,1,n}.$$

Model M2 (in contrast to model M1) does not contain the last inequality. This allows to use model M2 also for problems with $n \in \{100,200,250\}$. The developed models clearly outperform all existing models in the literature.

For problem $P2,S1||\sum C_j$, two MILP models were developed in [9]. Here we describe the best model M2 in more detail. Suppose one has a staggered schedule s defined by a job list π. Let the variable $N_{i,j}$ define the position of the job j in the list π, i.e., $N_{i,j} = 1$ if job j is i-th scheduled and otherwise $N_{i,j} = 0$. Let the variable ft_i be the time of completing the i-th job j in the list π. Moreover, let st_i be the starting time of loading the i-th job j in the list π, and let mt_i be the completion time of loading the i-th job j in the list π. Then, the minimization of $\sum C_j$ is equivalent to the minimization of $\sum_i ft_i$. Since each position in the list π can be occupied by only one job, the equality $\sum_i N_{i,j} = 1$ holds. Since each job has to be placed at some position in the list π, the equality $\sum_j N_{i,j} = 1$ holds. The inequality $ft_i - st_i \ge \sum_j N_{i,j}(s_j + p_j)$ holds since between the starting time st_i and the finishing time ft_i of the i-th job j, one has to load and to process the job j. The inequality $mt_i - st_i \ge \sum_j N_{i,j} s_j$ holds since between the starting time st_i and the finishing time mt_i of the i-th job j, one has to load the job j. The inequality $st_i \ge mt_{i-1}$ holds since the list π defines the order for loading all the jobs. The inequality $st_i \ge ft_{i-2}$ holds since we consider schedules where all jobs are processed in a staggered order, i.e., jobs are processed alternately on the two machines. Thus, the model M2 can be described in the following way:

$$\sum_{i=1}^{n} ft_i \to \min,$$
$$\sum_{i=1}^{n} N_{i,j} = 1, \quad \sum_{j=1}^{n} N_{i,j} = 1,$$
$$ft_i - st_i \ge \sum_{j=1}^{n} N_{i,j}(s_j + p_j), \qquad mt_i - st_i \ge \sum_{j=1}^{n} N_{i,j} s_j,$$

$$st_i \geq mt_{i-1},\ i = 2,...,n; \qquad st_i \geq ft_{i-2},\ i = 3,...,n.$$

Here $N_{i,j}$ are binary variables; ft_i, st_i and mt_i are positive variables. Testing the models for the instances with up to 100 jobs showed that the obtained solutions are rather close to the optimal ones.

Heuristic Algorithms

Since the problems considered are strongly NP-hard, the development of heuristic algorithms is required. For problem $P2,S1||C_{\max}$, in [1] two simple backward/forward *O(n log n)* heuristics were given. Two versions of a greedy heuristic, a genetic algorithm and a version of the Gilmory-Gomory algorithm were proposed and tested in [2] for problem $P2,S1||C_{\max}$. In [6], a simulated annealing algorithm and a genetic algorithm were presented for the problem $P2,S1\|C_{max}$. The performance of these algorithms is evaluated for instances with up to 1000 jobs. The results are compared with existing algorithms from the literature. It is observed that both these heuristic algorithms show an excellent behavior and that the objective function values obtained are very close to a lower bound. The superiority over existing algorithms is obtained by using a composite neighborhood (mutation), generating several neighbors from sub-neighborhoods with different probabilities and taking the best solution as generated neighbor. Recently, two fast constructive algorithms with a complexity of $O(n^2)$ have been presented in [11], which have an excellent performance for instances with up to 10,000 jobs. This superiority is obtained by sequencing the jobs on the two machines such that the machine idle time and the server waiting time are mimimized.

We only note that for the closely related problem of minimizing forced idle time (interference problem) a heuristic was given in [7] which has been tested on instances with even up to 100,000 jobs. The computational results indicate an excellent performance even for the large instances, and often an optimal solution was obtained.

For problem $P2,S1||\sum C_j$, two heuristics (simulated annealing and a hybridization of simulated annealing and harmony search) were given in [8]. Computational experiments have been done for instances with up to 250 jobs.

Finally, we give some comments on approximation algorithms. For the problem $P,S1\|\sum w_j C_j$ a $(5-\frac{1}{m})$- approximation algorithm was given in [12]. In [10], a $(3-\frac{1}{m})$- approximation algorithm is proposed for the same problem. In fact, the same approach as known from the literature was used, i.e., the original problem is replaced by a relaxed problem, which can be easily solved and thus, an optimal order of the completion times can be found. Then, in the original problem, the jobs are scheduled in the order determined and the improved performance bound was derived.

References

1. Abdekhodaee, A. Scheduling parallel machines with a single server: some solvable cases and heuristics / A. Abdekhodaee, A. Wirth // Computers & Operations Research –2002. – Vol. 29. – P. 295–315.

2. Abdekhodaee, A. Scheduling two parallel machines with a single server: the general case / A. Abdekhodaee, A. Wirth, H.S. Gan // Computers & Operations Research –2006. – Vol. 33. – P. 994–1009.

3. Gan, H.S. A branch-and-price algorithm for the general case of scheduling parallel machines with a single server / H. S. Gan, A. Wirth,A. Abdekhodaee // Computers & Operations Research–2012. – Vol. 39. – P. 2242–2247.

4. Hall, N.G. Parallel machine scheduling with a Common server / N.G. Hall, C.N. Potts, C. Sriskandarajah // Discrete Applied Mathematics.– 2000. – Vol. 120. – P. 223–243.

5. Hasani, K. Block models for scheduling jobs on two parallel machines with a single server / K. Hasani, S.A. Kravchenko, F. Werner // Computers & Operations Research. – 2014. – Vol. 41. P. 94–97.

6. Hasani, K. Simulated Annealing and Genetic Algorithms for the Two–Machine Scheduling Problem with a Single Server / K. Hasani, S.A. Kravchenko, F. Werner // International Journal of Production Research.–2014, Vol. 52, № 13, P. 3778 - 3792.

7. Hasani, K. Minimizing Interference for Scheduling Two parallel Machines with a Single Server/ K. Hasani, S.A. Kravchenko, F. Werner // International Journal of Production Research.–2014, Vol. 52, № 24, P. 7148 - 7158.

8. Hasani, K. A Hybridization of Harmony Search and Simulated Annealing to Minimize Mean Flow Time for the Two-Machine Scheduling Problem with a Single Server / K. Hasani, S.A. Kravchenko, F. Werner // International Journal of Operational Research Nepal. – 2014. – Vol. 3. № 01. P. 9–26.

9. Вернер, Ф. Минимизация суммарного времени обслуживания для системы с двумя приборами и одним сервером / Ф. Вернер, С.А. Кравченко, К. Хасани // Информатика. – 2014. – № 1(41). – С. 15 – 24.

10. Hasani, K. Minimizing Total Weighted Completion Time Approximately for the Parallel Machine Problem with a Single Server / K. Hasani, S.A. Kravchenko, F. Werner. – Information Processing Letters, Vol. 114, № 02. P. 500 – 503.

11. Hasani, K. Minimizing the Makespan for the Two-Machine Scheduling Problem with a Single Server: Two Algorithms for Very Large Instances / K. Hasani, S.A. Kravchenko, F. Werner. – Engineering Optimization, 2016. –Vol. 48, № 01, 173 – 183.

12. Wang, G. An approximation algorithm for parallel machine scheduling with a common server / G. Wang, T. C. E. Cheng // Journal of Operation Research Society. – 2001. – Vol. 52. – P. 234–237.