

A Mixed Integer Linear Programming Solution for Insertion of Idle Time in the Jobs Scheduling

Chi-Yang Tsai

Department of Industrial Engineering and Management
Yuan-Ze University
Taoyuan, Taiwan
iecytsai@saturn.yzu.edu.tw

Yi-Chen Wang

Department of Industrial Engineering and Management
Yuan-Ze University
Taoyuan, Taiwan
s978906@mail.yzu.edu.tw

Abstract—This paper considers the use of insertion of idle time in the unrelated parallel machine scheduling problem with an objective of minimizing the sum of earliness and tardiness. An effective and efficient mixed integer linear programming formulation is proposed. For comparison, two models where the use of idle time is restricted and prohibited, respectively, are also developed. The results of the conducted numerical experiment showed ET values can be significantly reduced if idle time is properly inserted before jobs.

Keywords- Jobs scheduling; Unrelated parallel machine; Insertion of Idle Time; Earliness and Tardiness; Mixed-integer programming.

I. INTRODUCTION

When scheduling jobs to be processed on machines, it is common to avoid any machine idle when there are waiting jobs. The attempt is to pursue high equipment utilization and throughput. Thus, idle time is seen as waste and has to be prevented. However, deliberately delaying job processing and holding a machine idle can be beneficial in many situations. For example, when the predicted capacity utilization is less than ideal, which is quite common in global economic crisis, machine idle is bound to occur. It is then practical to focus on other performance measures. If on-time job completion is preferred, it is desired to schedule jobs such that they are completed right on their due dates. Starting times of job processing are arranged aiming to reduce earliness and tardiness. Machines may be idle between processing of jobs. The main issue becomes how to suitably insert idle time in job schedules.

This study considers the parallel machine scheduling problem with Insertion of Idle Time (IIT). Unrelated parallel machine scheduling is widely applied in manufacturing environments [1]. In addition, idle time insertion arises in JIT environments where costs associated with early completion time of jobs are relevant [2]. As a result, it has drawn attention of numerous researchers. However, most related research either does not allow idle time or does not address the use of IIT.

The goal of this paper is to investigate how inserted idle time can be utilized in the unrelated parallel machine scheduling problems (UPMSP) given the objective of

minimizing the total earliness and tardiness (ET). We propose an effective and efficient mixed integer linear programming formulation (MIP) for finding optimal solutions to the addressed problem. Results of our numerical experiment show that by allowing IIT, scheduling flexibility increased and performance is improved.

UPMSP have been drawn attention to many researchers [3] and [4]. However, researches on those problems with idle time are very limited. Among the few papers, [5] proposed the MIP formulations the objective of ET minimization, then the same problem is solved by formulations with dispatching rules [6]. A lot of research work has been done to date on the development of solutions for UPMSP tackling a variety of objectives. However, in many manufacturing environments, it is desired to finish jobs on time rather than being early or late. This paper which addresses this problem does consider objective of minimizing ET. Reference [7] studied this UPMSP with sequence dependent setup times minimize the sum of weighted ET and proposed a heuristic to solve it. For the same objective, [8] presented the heuristics to solve the job scheduling problem with sequence dependent setup times on UPMSP. Reference [9] employed simulated annealing to schedule jobs with sequence dependent setup times on UPMSP with objective of ET to obtain near optimal solutions. Authors considered the use of heuristic for problems of continuous [10] and discrete-variable [11]. Then, UPMSP is discrete-variable problem, and then [12] developed the heuristic with dispatching rules to solve the same problem with sequence-dependent setups time given the weighted tardiness minimization, and [13] developed and solved, too.

To the best of the authors' knowledge, there have been only three publications that proposed MIP solutions for the earliness and tardiness parallel machine scheduling problem with setup time. Reference [14] proposed a MIP for minimizing ET in an UPMSP with sequence dependent setup times considerations. More recently, [15] improved the MIP model [14].

The work presented in this paper differs from the above three publications in that an improved MIP is developed and analysis on the effects of allowing idle time on job schedule performance is conducted. Overall, the contribution of this study is twofold.

- The proposed mixed-integer model is improved in two ways. A potential problem of negative job starting time that may occur in some instances with the models in the above-mentioned publications. It has been fixed in the proposed model. In addition, the proposed model also finds optimal solutions more efficiently.
- How job scheduling performance is influenced by allowing idle time is investigated in this study. Three models with various degrees of idle time allowance are developed. Numerical experiments designed for that purpose is conducted.

The rest of the paper is organized as follows: the next section introduces the applied methodology and mathematical formulations. The conducted numerical experiments and results are displayed in Section 3. The conclusion and future work is presented in the last section.

II. MATHEMATICAL FORMULATIONS

This study considers three models for four jobs and two machines in Fig. 1. Model MTA of idle time is allowed to be inserted before the processing of any job; Then, Model MTR of idle time is allowed to be inserted between any two jobs on the same machine. However, the first job on each machine has to be processed without any delay. Third model is Model MTP, which idle time is not allowed at all. The formulation for MTA with allowing machine idle time is constructed [14] shown as follows.

$$\text{Min } \sum_{j=1}^n (E_j + T_j) \quad (1)$$

Subject to:

$$C_j + E_j - T_j = d_j, \quad \forall j = 1, 2, \dots, n. \quad (2)$$

$$\sum_{k=1}^m y_{jk} = 1, \quad \forall j = 1, 2, \dots, n. \quad (3)$$

$$\begin{aligned} C_j - C_i - Mx_{ij} - My_{ik} - My_{jk} &\geq p_{jk} - 3M \\ i &\neq j; \forall i = 1, 2, \dots, n; \forall j = i+1, i+2, \dots, n; \\ &\forall k = 1, 2, \dots, m. \end{aligned} \quad (4)$$

$$\begin{aligned} C_i - C_j + Mx_{ij} - My_{ik} - My_{jk} &\geq p_{ik} - 2M \\ i &\neq j; \forall i = 0, 1, 2, \dots, n; \forall j = i+1, i+2, \dots, n; \\ &\forall k = 1, 2, \dots, m. \end{aligned} \quad (5)$$

$$C_0 = 0 \quad (6)$$

$$C_j, E_j, T_j \geq 0, \quad \forall j = 1, 2, \dots, n. \quad (7)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i = 0, 1, 2, \dots, n; j = 1, 2, \dots, n. \quad (8)$$

$$y_{jk} \in \{0, 1\}, \quad \forall i = 0, 1, 2, \dots, n; j = 1, 2, \dots, n. \quad (9)$$

Equation (1) represents the objective function which aims to minimize the sum of earliness and tardiness. Equation (2) calculates completion time C_j , earliness E_j and tardiness T_j of each job j . Equation (3) ensures that each job is processed on one and only one machine. To be used calculate completion times and to ensure that no job can precede and succeed the same job by (4) and (5). Equation (6) states that the completion time for the dummy job 0 is zero. Equation (7) ensures that completion times, earliness and tardiness are non-negative. Equations (8) and (9) specify that the decision variables x and y are binary over all domains. In addition, $x_{ij} = 1$, if job j is processed after job i , 0 otherwise; $y_{jk} = 1$, if job j is processed on machine k , and is 0 otherwise.

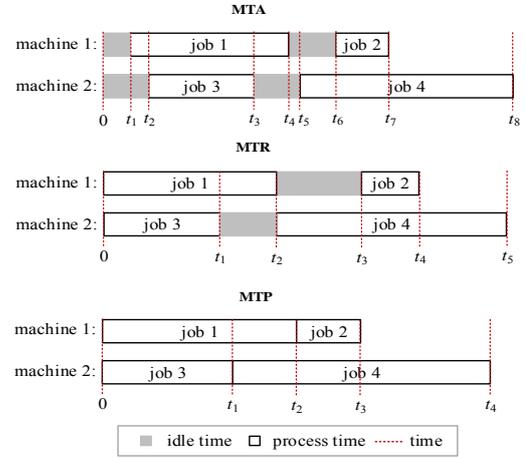


Figure 1. Job schedules of a case with two machines and four jobs in models

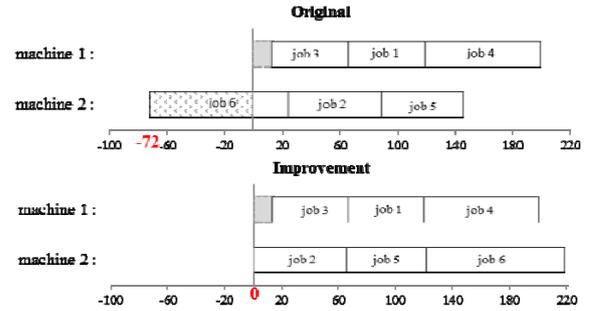


Figure 2. Job scheduling by original and improvement of MTA

There is a potential problem with this model. Since the completion time of a job is not restricted to be greater than or equal to the processing time of the job, the starting time of a job can be negative. This problem usually occurs when due dates are tight. Fig. 2 depicts the jobs schedules for a case with six jobs and two machines. Therefore, model state of negative can be obtained improvement by (10), where makes sure that the completion time of any job is not less than the processing time of that job.

$$C_j - My_{jk} \geq p_{jk} - M, \quad \forall j = 1, 2, \dots, n; k = 1, 2, \dots, m. \quad (10)$$

Equation (11) proposed by [15] is also added to the formulation. With this constraint, the search space is greatly reduced and run time can be significantly shortened. In our numerical experiment, the run time can be reduced over 100 times with this constraint in some test instances.

$$y_{jk} + 1 - x_{ij} \geq y_{ik}, \quad (11)$$

$$i \neq j; \forall i, j = 1, 2, \dots, n; \forall k = 1, 2, \dots, m.$$

In MTR, idle time is not allowed to be inserted before the first job on each machine. In addition to the eleven constraints in MTA, we introduce (12) in the formulation of MTR to ensure that the completion time of the first job on each machine is equal to its processing time.

$$M \left(\sum_{i=1; i \neq j}^n x_{ij} - y_{jk} \right) \geq C_j - p_{jk} - M, \quad (12)$$

$$j = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

With this constraint, however, (4) and (5) have to be replaced by (13) and (14), respectively. The difference between the two sets of constraints is the range of index j . In the formulation of MTA, a range of j from $i + 1$ to n is sufficient for (4) and (5). Since (12) is included in the formulation of MTR, the range of j has to be adjusted to be from 1 to n in order to match the range in (12).

$$C_j - C_i - Mx_{ij} - My_{ik} - My_{jk} \geq p_{jk} - 3M \quad (13)$$

$$i \neq j; i, j = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

$$C_i - C_j + Mx_{ij} - My_{ik} - My_{jk} \geq p_{ik} - 2M \quad (14)$$

$$i \neq j; i = 0, 1, 2, \dots, n;$$

$$j = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

MTP does not allow any inserted idle time. Its formulation is similar to the one of MTA with an extra constraint as follows. Equation (15) ensures that the total completion time of each machine is smaller than any job's completion time.

$$\sum_{i=1}^n (p_{ik} \times y_{ik}) \geq C_j - M(1 - y_{jk}), \quad (15)$$

$$j = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

III. COMPUTATIONAL EXPERIENCE

The constructed formulations of the three models presented in the previous section are solved using Gurobi v4.5.2 on a desktop computer with Intel Xeon W3520 2.67GHz CPU and memory capacity of 4GB. Processing

times are randomly generated from U [50,100]. There are three due date ranges (DDRs), loose, moderate and tight, adapted from [6]. They are generated from U [0.75 Δ , 1 Δ], U[0.5 Δ , 0.75 Δ] and U[0.25 Δ , 0.5 Δ], respectively, where $\Delta = P/m$, P is the total all job process times and m is the number of machines. The number of machines is set between 2 to 4 and the number of jobs ranges from 6 to 10. For each combination of job number, machine number and due date range, 15 instances are generated, resulting in a total of 675 test instances.

Table I lists the averages of optimal ET and the total insertion of idle time (IITs) under the three models. The average of the resulting makespan (C_{max}) is also provided. As can be seen, MTA has the smallest average ET (275.38). It clearly shows that by IITs (102.32), scheduling flexibility increases and better ET can be achieved. It can also be observed from that MTR has lower average ET over MTP. When comparing the average IITs, it is also interesting to see that the value drops dramatically from 102.32 in MTA to 36.08 in MTR. This indicates, when allowed, a large portion of idle time is preferred to be inserted before the first job on each machine.

TABLE I. AVERAGE ET, IITS, AND CMAX VALUES

Model	ET	IITs	C_{max}
MTA	275.38	102.32	247.63
MTR	334.36	36.08	235.96
MTP	348.03	0.00	221.97

The tradeoff of pursuing scheduling flexibility is increasing C_{max} . Without insertion of idle time, MTP has the highest average ET but the lowest average C_{max} when . To further investigate this tradeoff, *relative percentage difference (RPD)* of MTA's performance over those of the other two models is calculated using $(O_{sol} \square MTA_{sol}) / MTA_{sol} \times 100$, where O_{sol} and MTA_{sol} are the solutions (ET or C_{max}) obtained in MTA and the other model (MTR or MTP), respectively.

The average *RPD* are listed in Table II. The improvement in average ET values by allowing idle time is significant. The average ET obtained in MTA is 21.42% and 26.28% lower than those in MTR and MTP, respectively. Even though the average C_{max} increases, the percentage difference is a lot smaller. In addition, we again see that allowing inserted idle time before the first job to be processed on a machine has a lot more effect on the improvement of ET than allowing inserted idle time between jobs.

TABLE II. AVERAGE RELATIVE PERCENTAGE DIFFERENCE

Model	ET	C_{max}
MTR	21.42%	-4.71%
MTP	26.38%	-10.36%

Table III illustrates the average performance measure values with various DDRs, numbers of jobs and machines in MTA, including earliness (E) and tardiness (T). The ET values with loose and moderate DDRs are very close, even though there is an obvious difference in the amount of idle

time used. However, the gap in the C_{max} is more apparent. The ET increases significantly when the DDRs is tight. It is not surprising, as the room for inserting idle time is a lot less. Furthermore, it becomes less possible to find the balance between ET. When DDRs are tight, more jobs are tardy.

To further investigate the various numbers of jobs, all the values increase in the number of jobs. However, to various numbers of machines, all the values decrease as the number of machines increases. It is worth pointing out that the average total amount of inserted idle times has the same trend, but not very significantly.

TABLE III. AVERAGE PERFORMANCE MEASURE VALUES WITH VARIOUS DDRS, NUMBERS OF JOBS AND MACHINES

	Groups	ET	IITs	C_{max}	E	T
DDRs	Loose	247.40	225.84	295.38	117.63	129.76
	Moderate	249.53	77.64	239.48	108.28	141.25
	Tight	329.20	3.48	208.03	31.66	297.54
Jobs	6	158.01	62.63	185.87	36.26	121.75
	7	218.75	79.13	218.76	54.98	163.77
	8	269.21	101.51	245.63	82.33	186.88
	9	316.15	119.30	271.47	103.96	212.19
	10	403.24	139.72	308.04	145.19	258.05
Machines	2	387.74	104.68	336.11	135.69	252.05
	3	252.08	103.28	229.02	80.01	172.07
	4	186.32	99.00	177.76	41.88	144.44

IV. CONCLUSION

We studied the unrelated parallel machine scheduling problem with insertion of idle time. An improved MIP was developed for the model where idle time is allowed to be inserted before any job. The proposed formulation fixed a potential problem by [14]. In addition, it can be solved a lot more efficiently. For comparison, we also constructed MIP for the model where idle time is allowed only between jobs and the model where insertion of idle time is prohibited. The results of our numerical experiment showed ET values can be significantly reduced if idle time is properly inserted before jobs. It indicates that the use of inserted idle time is able to improve on-time job completion. It is especially beneficial if capacity utilization is low and idle time is inevitable. On-time completion should be considered as the primary objective instead of total completion time. Furthermore, allowing idle time to be inserted before the first job has greater contribution to the benefit compared to inserted idle time between jobs.

REFERENCES

- [1] L. Yu, H. M. Shih, M. Pfund, W. M. Carlyle, J.W. Fowler, "Scheduling of unrelated parallel machines: an application to PWB manufacturing," *IIE Trans.*, vol. 34, no. 11, pp. 921-931, 2002.
- [2] E. C. Colin, and R. C. Quinino, "An algorithm for insertion of idle time in the single-machine scheduling problem with convex cost functions," *Comput. Oper. Res.*, vol. 32, no. 9, pp. 2285-2296, 2005.
- [3] Y. C. Wang, and C. Y. Tsai, "Minimizing Maximum Completion Time on Parallel Machines Scheduling with Limited Resources," in *Proc. the 10th Asia Pacific Industrial Engineering & Management Systems Conf.*, Kitakyushu, Japan (2009).
- [4] Y. C. Wang, and C. Y. Tsai, "Makespan minimization on unrelated parallel machines with sequence-dependent setup times using a simulated annealing approach," in *Proc. the 7th International Conf. on Intelligent Manufacturing Logistics Systems*, Chung-Li, Taiwan (2011).
- [5] C. Y. Tsai, and Y. C. Wang, "The sum of earliness and tardiness minimization on unrelated parallel machines with inserted idle time," in *Proc. the 12th WSEAS International Conf. on Applied Computer Science*, Singapore City, Singapore (2012).
- [6] C. Y. Tsai, and Y. C. Wang, "Application of Mixed-Integer Programming and Dispatching Rules on Parallel Machine Scheduling with Inserted Idle Time," *A. M. M.*, vol. 437, pp. 748-751, 2013.
- [7] R. B. Heady, and Z. Zhu, "Minimizing the sum of job earliness and tardiness in a multimachine system," *Int. J. Prod. Res.*, vol.36, no. 6, pp. 1619-1632, 1998.
- [8] F. Sivrikaya-Şerifoğlu, and G. Ulusoy, "Parallel machine scheduling with earliness and tardiness penalties," *Comput. Oper. Res.*, vol. 26, no. 8, pp. 773-787, 1999.
- [9] S. Radhakrishnan, and J. A. Ventura, "Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times," *Int. J. Prod. Res.*, vol. 38, no. 10, pp. 2233-2252, 2000.
- [10] C. Y. Tsai, and I. W. Kao, "Particle swarm optimization with selective particle regeneration for data clustering," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6565-6576, 2011.
- [11] J. P. Armaout, G. Rabadi, and R. Musa, "A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times," *J. Intell. Manuf.*, vol. 21, no. 6, pp. 693-701, 2010.
- [12] R. Logendran, B. McDonell, and B. Smucker, "Scheduling unrelated parallel machines with sequence-dependent setups," *Comput. Oper. Res.*, vol. 34, no. 11, pp. 3420-3438, 2007.
- [13] Y. K. Lin, and C. W. Lin, "Dispatching rules for unrelated parallel machine scheduling with release dates," *Int. J. Adv. Manuf. Technol.*, vol. 67, pp. 1-11, 2013.
- [14] Z. Zhu, and R. B. Heady, "Minimizing the sum of earliness/tardiness in multi-machine scheduling: A mixed integer programming approach," *Comput. Ind. Eng.*, vol. 38, no. 2, pp. 297-305, 2000.
- [15] M. K. Omar, and S. C. Teo, "Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach," *Appl. Math. Comput.*, vol. 181, no. 2, pp. 1008-1017, 2006.