IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 18, 2019, accepted October 13, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947685

Comparative Analysis of Mixed Integer Programming Formulations for Single-Machine and Parallel-Machine Scheduling Problems

KUO-CHING YING[1], CHEN-YANG CHENG[1], SHIH-WEI LIN[2,3,4], AND CHIA-YANG HUNG[5]

[1]Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 106, Taiwan
[2]Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan
[3]Department of Neurology, Linkou Chang Gung Memorial Hospital, Taoyuan 333, Taiwan
[4]Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei 243, Taiwan
[5]United Microelectronics Corporation, Hsinchu 30078, Taiwan

Corresponding author: Shih-Wei Lin (swlin@mail.cgu.edu.tw)

The work of S.-W. Lin was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST107-2410-H-182-005-MY2, and in part by the Linkou Chang Gung Memorial Hospital under Grant BMRPA19.

ABSTRACT This study evaluates various Mixed Integer Programming (MIP) formulations for solving single-machine and parallel-machine scheduling problems, with the objective of minimizing the total completion time and the makespan of jobs. Through extensive numerical study, the MIP formulation, which is suitable for dealing with each specific single-machine or parallel-machine scheduling problem, is identified. Benchmarks are also provided for the development of other algorithms for future research.

INDEX TERMS Scheduling, total completion time, makespan, single-machine, parallel-machine.

I. INTRODUCTION

The single-machine scheduling problem (SMSP), which is one of the most studied issues relating to manufacturing systems, can be found in numerous real-world production systems that require the effective scheduling of jobs performed on a unique machine. In the past several decades, a wide variety of studies have focused on SMSPs, in order to consider different scheduling criteria and explore efficient and effective methods to find optimal and near-optimal schedules [1], [2]. As a generalization of the SMSP, the parallel-machine scheduling problem (PMSP) has also received considerable attention from researchers [3], [4]. According to the similarity of the machines used for processing, PMSP issues can be further classified as identical PMSPs (IPMSPs), non-identical PMSPs (NIPMSPs), and unrelated PMSPs (UPMSPs). In recent years, many PMSP-related studies have endeavored to develop efficient heuristics (e.g., [5] and [6]).

For many NP-hard problems, Mixed Integer Programming (MIP) is one of the exact methods commonly used to find optimal solutions for small- and medium-sized problems, as well as lower and/or upper bounds in larger problems, and to benchmark the quality of the solutions and efficiency of the compared methods [7]–[9]. The advantages of using MIP rather than other approaches (e.g. heuristics and meta-heuristics) for solving small- to medium-size NP-hard problems include but are not limited to the following. First, MIP is a common language that uniquely describes a problem in strictly mathematical terms. Second, there are many types of commercially available software that can be used to solve such problems out-of-the-box without further knowledge in scheduling or coding from the user. Third, most solvers allow the integration of external heuristics into their solution process (via starting solutions or callback functionalities) to speed up the solution process. Fourth, if real instances are too big to be solved to optimality by a solver, then MIP can be used to compute lower and/or upper bounds that certify the quality of other approaches. Moreover, unlike heuristics and meta-heuristics, the solution of MIP is the optimum with reasonable computational time for small- and medium-sized problems.

Consequently, the performance of MIP formulations indicates, to a certain degree, the ability to improve and apply the

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Gawanmeh.

solution methods to be developed. For SMSPs, the MIP formulations can be classified into the different types of decision variables, including completion time variables (CTV), time index variables (TIV), linear ordering variables (LOV), and assignment and positional date variables (APDV). A survey has been provided by Queyranne and Schulz [10].

The MIP formulation with CTVs, as initiated by Balas [11], addressed the SMSP to minimize the total weighted completion time. Queyranne [12] considered SMSPs with a view to minimizing the total weighted completion time and total weighted tardiness, respectively. That study defined feasible schedules using the vector of job completion times, and gave a complete description of the face lattice of the scheduling polyhedron. Regarding the MIP formulation with TIV, Sousa and Wolsey [13] investigated the SMSP and found that such an MIP formulation can yield better lower bounds than other types of MIP formulations. Van den Akker *et al*. [14] proposed a time-indexed formulation for the SMSP in order to minimize the total weighted completion time, and presented the complete characterization of all facet-inducing inequalities in the MIP formulation. Dyer and Wolsey [15] formulated an MIP model using LOV in an effort to solve the SMSP aiming at minimizing the total weighted job completion time. Lasserre and Queyranne [16] proposed an MIP formulation using APDV to determine the optimal SMSP solution, again aimed at minimizing the total weighted job completion time. Dauzère-Pérès [17] presented an MIP model formulated by APDV to minimize the number of late jobs on the SMSP, and showed that the MIP formulation not only allowed the lower bound to be determined quickly, but that it could also readily be extended to the weighted case.

For the purpose of the comparison of different MIP formulations, Khowala *et al*. [18] investigated the above-mentioned four types of MIP formulations for SMSPs, which aimed at minimizing total weighted tardiness. The experimental results showed that the most promising MIP formulation for this SMSP depended on the sum of the job processing times. However, the optimal solutions for most of the test instances were not found. After that, Keha *et al.* [19] conducted a comparative analysis of MIP formulations for SMSPs with different criteria, including total weighted completion time, total weighted tardiness, maximum lateness, and number of tardy jobs for all jobs, and the computational results revealed that the performances of these formulations were highly dependent on the objective function, number of jobs, and the sum of the processing times of all the jobs. For certain problems, the MIP model formulated with APDV was more efficient in computation than other commonly used MIP formulations. Subsequently, Baker and Keller [20] concentrated on different integer programming formulations, as based on binary precedence variables, for solving the SMSP with the total tardiness objective. They also found that one formulation, based on "sequence-position" variables, performed much more effectively than the others. Following these investigations of mathematical programming for SMSPs, some optimization and heuristic algorithms have been successfully developed in recent years.

Extending the research line to PMSPs, Rabadi *et al.* [21] presented an MIP formulation with LOV for solving UPMSPs to minimize the makespan. Rocha *et al.* [22] constructed an MIP formulation with CTV, which aimed at finding the optimal job sequence in UPMSPs, and the resulting solutions were used to contrast with those of their proposed algorithm. Kedad-Sidhoum *et al*. [23] formulated an MIP model using TIV, and obtained efficient lower bounds for PMSPs with the aim of minimizing earliness and tardiness costs. Li and Yang [24] further proposed an MIP formulation with APDV for solving the NIPMSP, in order to minimize the total and mean weighted completion times. Recently, Unlu and Mason [25] analyzed various MIP formulations for PMSPs, with the objective of minimizing the total weighted completion time and maximum completion time of jobs. The experimental results showed that the most promising MIP formulations depended on the small/large processing times of the jobs.

Some researchers, such as Khowala *et al*. [18], Keha *et al*. [19], Baker and Keller [20], and Unlu and Mason [25], have conducted comparative analyses of different types of MIP formulations on SMSPs and IPMSPs, respectively. However, to the best of our knowledge, there is no research in literature that conducted comparative analyses of different MIP formulations between SMSPs and IPMSPs. This work investigated four types of MIP formulations for simultaneously solving SMSPs and IPMSPs. The scheduling criteria are to minimize the total completion time and the maximum completion time for all jobs, which are adapted by the most traditional scheduling in the single-machine and parallel machine settings [26].

To the best of our knowledge, no new formulation has been proposed up until 2019 that is suitable for both the SMSPs and PMSPs considered in this study. The performances of these formulations usually depend on shop types, objective functions, number of jobs, and total processing times of jobs. Therefore, the contribution of this paper is to give a wider understanding of the computational efficiencies between different MIP formulations, as well as between SMSPs and IPMSPs. Based on the experimental results, this study offers a discussion and recommendations on which MIP formulation might perform best for the addressed SMSPs and IPMSPs, respectively.

The remainder of this paper is organized, as follows: Sections 2 and 3 present eight MIP formulations regarding SMSPs and IPMSPs, respectively; Section 4 details the computational experiments; Section 5 gives the conclusions of this study.

II. MIP FORMULATIONS FOR SMSPS

A. ASSUMPTIONS

The SMSP involves a set of n jobs/works/demands, $J = \{1, \ldots, n\}$, which must be processed independently on a single machine/facility/person (see Fig. 1), with p_j and C_j

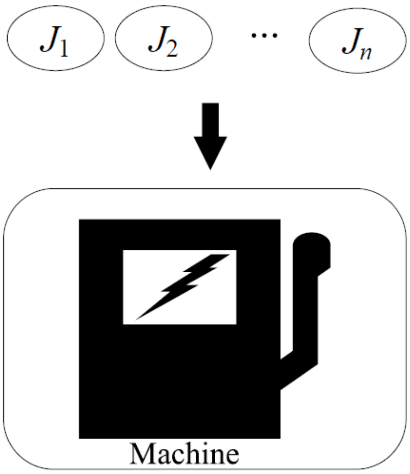


FIGURE 1. Single machine scheduling setup.

($j = 1, 2, \ldots, n$) as the processing time and completion time of the jth job, respectively, and $C_{\max} \geq C_j$ as the maximum completion time (makespan) for all jobs. The considered objectives are the minimization of the total completion time and the makespan, respectively, as well as the determination of the production sequence of all jobs. Using the three-field classification scheme [27], the two SMSPs under study can be expressed as the triplets $1||\sum C_j$ and $1||C_{\max}$, and defined with critical assumptions, as follows:

- All jobs are independent and assumed to be ready at time zero.
- All machines have autonomy, and there is no collaboration between machines.
- The machine can only process one job at a time, and it must process all jobs without interruption, from the start of processing the first job to the completion of processing the last job.
- The setup time of the machine is not taken into account.
- The breakdown of the machine is ignored.

For the purposes of comparison, the MIP formulations using the four different decision variables, named CTV-MIP, TIV-MIP, LOV-MIP, and APDV-MIP [19], are revised and described in the following subsections, as based on the above notations and assumptions.

B. CTV-MIP FORMULATION

The concept of the CTV-MIP formulation involves the completion times of jobs with different processing priorities. To this end, we introduce the binary decision variable $Y_{j\ell}$, $j \in J$, $\ell \in J, j \neq \ell$. If job j is processed before job ℓ, then $Y_{j\ell} = 1$; otherwise, $Y_{j\ell} = 0$. The necessary constraint sets of the CTV-MIP formulation for the $1||\sum C_j$ and $1||C_{\max}$ problems can thus be represented, as follows:

$$C_j + p_\ell \leq C_\ell + M(1 - Y_{j\ell}), \quad \forall j \in J,\ \ell \in J, j \neq \ell \quad (1)$$

$$C_\ell + p_j \leq C_j + MY_{j\ell}, \quad \forall j \in J,\ \ell \in J, j \neq \ell \quad (2)$$

$$C_j \geq p_j, \quad \forall j \in J \quad (3)$$

$$Y_{j\ell} \in \{0, 1\}, \quad \forall j \in J,\ \ell \in J,\ j \neq \ell \quad (4)$$

where M is the arbitrarily large value. Constraint sets (1) and (2) restrict the relations of the completion times of jobs with different processing priorities. Constraint set (3) ensures that the completion time of a job is greater than its processing time, and also implies non-negative CTV. Constraint set (4) defines the binary decision variable.

C. TIV-MIP FORMULATION

The concept of TIV is constituted on the planning horizon that is partitioned as a finite number of discrete periods, $t = 1, 2, \ldots, T$, where period t represents the time interval from $(t - 1)$ to t. Here, we define x_{jt} as the binary TIV, in which job j starts at time t, then $x_{jt} = 1$; otherwise, $x_{jt} = 0$. The necessary constraint sets of the TIV-MIP formulation can be represented as the following equations:

$$C_j = \sum_{t=1}^{T-p_j+1} (t - 1 + p_j)x_{jt}, \quad \forall j = 1, 2, \ldots, n \quad (5)$$

$$\sum_{t=1}^{T-p_j+1} x_{jt} = 1, \quad \forall j = 1, 2, \ldots, n \quad (6)$$

$$\sum_{j=1}^{n} \sum_{s=\max\{0,t-p_j+1\}}^{t} x_{js} \leq 1, \quad \forall t = 1, 2, \ldots, T \quad (7)$$

$$x_{jt} \in \{0, 1\}, \quad \forall j = 1, 2, \ldots, n,\ \forall t = 1, 2, \ldots, T \quad (8)$$

Constraint set (5) calculates the completion time of each job, where $(T - p_j + 1)$ represents the last starting period of the processing of job j. Constraint set (6) imposes the restriction that each job can be assigned for only one processing time, and constraint set (7) indicates that, at most, one job can be processed for each time period. Constraint set (8) defines the binary TIV.

D. LOV-MIP FORMULATION

The concept of LOV is based on the processing priorities of different jobs. Let $\delta_{j\ell}, j \in J, \ell \in J, j \neq \ell$ be the binary LOV. If job j has a higher processing priority than job ℓ, then $\delta_{j\ell} = 1$; otherwise, $\delta_{j\ell} = 0$. The necessary constraint sets of the LOV-MIP formulation can be represented as:

$$\delta_{j\ell} + \delta_{\ell j} = 1, \quad \forall j \in J,\ \ell \in J, j \neq \ell \quad (9)$$

$$\delta_{j\ell} + \delta_{\ell\theta} + \delta_{\theta j} \leq 2, \quad \forall j \in J,\ \ell \in J,\ \theta \in J,\ j \neq \ell \neq \theta \quad (10)$$

$$C_j = \sum_{\ell \in J, \ell \neq j}^{p_\ell} \delta_{\ell j} + p_j, \quad \forall j \in J \quad (11)$$

$$\delta_{jt} \in \{0, 1\}, \quad \forall j \in J,\ \ell \in J \quad (12)$$

Constraint set (9) allows only one ordering result between two different jobs. Constraint set (10) imposes the restriction that, at most, two ordering results among three different jobs are valid. Constraint set (11) computes the completion time of each job. Constraint set (12) defines the binary LOV.

E. APDV-MIP FORMULATION

The main concept of APDV is to assign n jobs to n positional dates. Let u_{jd}, $j = 1, 2, \ldots, n$, $d = 1, 2, \ldots, n$ be the binary APDV. If job j is assigned to positional date d, then $u_{jd} = 1$; otherwise, $u_{jd} = 0$. The necessary constraint sets of the APDV-MIP formulation can be constructed, as follows:

$$\sum_{j=1}^{n} u_{jd} = 1, \quad \forall d = 1, 2, \ldots, n \tag{13}$$

$$\sum_{d=1}^{n} u_{jd} = 1, \quad \forall j = 1, 2, \ldots, n \tag{14}$$

$$C_d = \sum_{j=1}^{n} p_j u_{jd}, \quad \forall d = 1 \tag{15}$$

$$C_d \geq C_{d-1} + \sum_{j=1}^{n} p_j u_{jd}, \quad \forall d = 2, 3, \ldots, n \tag{16}$$

$$u_{jd} \in \{0, 1\}, \quad \forall j = 1, 2, \ldots, n, \ \forall d = 1, 2, \ldots, n \tag{17}$$

Constraint sets (13) and (14) ensure that the job assignments to different positional dates are feasible. Constraint sets (15) and (16) calculate the completion time of each job on each position, where C_d is the completion time of the job in the dth position. Constraint set (17) defines the binary APDV.

III. MIP FORMULATIONS FOR IPMSPS

A. ASSUMPTIONS

This section extends the assumptions of SMSPs (as described in Section II-A) to IPMSPs. Some additional assumptions that must be considered are described, as follows:

- A set of identical machines is available for processing jobs.
- Each job should be processed on one machine until completed.
- The processing times of a job on different machines are equal.

Due to the fact that the m machines ($K = \{1, 2, \ldots, m\}$) are identical (see Fig. 2), we naturally continue the previous assumptions of processing time p_j, completion time C_j and makespan $C_{\max} \geq C_j$, $j = \{1, 2, \ldots, n\}$ for the n jobs. We might visualize these as being machines in parallel because they are identical to each other, and thereby, perform the same services. However, these machines do not have to be physically parallel.

For the IPMSPs, we also consider the scheduling criteria to minimize the total completion time and makespan, respectively. Using the three-field classification scheme [27], the two IPMSPs under study can be expressed as the triplets $P_m||\sum C_j$ and $P_m||C_{\max}$, respectively. For solving these two IPMSPs, we modify the MIP formulations with CTV of Rocha et al. [22], with TIV of Kedad-Sidhoum et al. [23], with LOV of Rabadi et al. [21], and with APDV of Li and Yang [24], and present four MIP formulations, namely

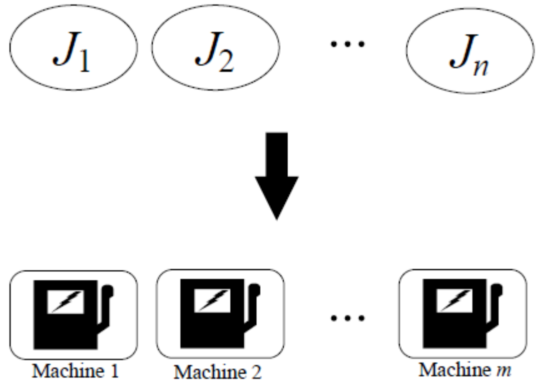


FIGURE 2. Parallel-machine scheduling setup.

CTV-MIP, TIV-MIP, LOV-MIP, and APDV-MIP, in the following subsections.

B. CTV-MIP FORMULATION

Consider that the two binary decision variables of α_{jk}, $j \in J$, $k \in K$ and $\beta_{jj'k}$, $(j, j') \in J$, $j < j'$, $k \in K$ need to be defined. If job j is processed on machine k, then $\alpha_{jk} = 1$; otherwise, $\alpha_{jk} = 0$. And, if job j is processed before job j' on machine k, then $\beta_{jj'k} = 1$; otherwise, $\beta_{jj'k} = 0$. Let S_j be the starting time of job j, and M be the arbitrarily large value; the CTV-MIP formulation can then be constructed with the following constraints:

$$\sum_{k \in K} \alpha_{jk} = 1, \quad \forall j \in J \tag{18}$$

$$C_j \geq S_j + p_j - (1 - \alpha_{jk})M, \quad \forall j \in J, \ k \in K \tag{19}$$

$$(1 - \alpha_{jk})M + (1 - \alpha_{j'k})M + (1 - \beta_{jj'k})M + S_{j'} \geq S_j + p_j, \quad \forall (j, j') \in J, j < j', k \in K \tag{20}$$

$$(1 - \alpha_{jk})M + (1 - \alpha_{j'k})M + \beta_{jj'k}M + S_j \geq S_{j'} + p_{j'}, \quad \forall (j, j') \in J, \ j < j', \ k \in K \tag{21}$$

$$\alpha_{jk} \in \{0, 1\}, \quad \forall j \in J, \ k \in K \tag{22}$$

$$\beta_{jj'k} \in \{0, 1\}, \quad \forall (j, j') \in J, \ j < j', \ k \in K \tag{23}$$

Constraint set (18) restricts each job to being processed on only one machine. Constraint set (19) calculates the completion time of each job on each machine. Constraint sets (20) and (21) state that, on each machine, the starting times of jobs with lower priorities must be posterior to the completion times of jobs with higher priorities. Constraint sets (22) and (23) define the binary decision variables.

C. TIV-MIP FORMULATION

Regarding the TIV-MIP formulation, we continue to use the binary decision variable x_{jt}, $j = 1, 2 \ldots, n$, $t = 1, 2, \ldots, T$, as defined in Section II-C. If job j starts at time t, then $x_{jt} = 1$; otherwise, $x_{jt} = 0$. The constraint sets of the

TIV-MIP formulation can be stated, as follows:

$$C_j = \sum_{t=1}^{T-p_j+1} (t-1+p_j)X_{jt} = 1, \quad \forall j = 1, 2, \ldots, n \quad (24)$$

$$\sum_{t=1}^{T-p_j+1} x_{jt} = 1, \quad \forall j = 1, 2, \ldots, n \quad (25)$$

$$\sum_{j=1}^{n} \sum_{s=\max\{0,t-p_j+1\}}^{t} x_{js} \leq m, \quad \forall t = 1, 2, \ldots, T \quad (26)$$

$$x_{jt} \in \{0, 1\}, \quad \forall j = 1, 2, \ldots, n, \ \forall t = 1, 2, \ldots, T \quad (27)$$

where Constraint sets (24), (25), and (27) are equivalent to Constraint sets (5), (6), and (8) of the TIV-MIP formulation in the case of SMSPs (see the definitions in Section II-C). Constraint set (26), which is different from Constraint set (7), states that, at most, m machines can be handled at any one time.

D. LOV-MIP FORMULATION

Here, we define the binary LOV, $\delta_{ijk}, i = 0, 1, \ldots, n, j = 1, 2, \ldots, n, k = 1, 2, \ldots, m$. If job i is processed directly before job j on machine k, then $\delta_{ijk} = 1$; otherwise, $\delta_{ijk} = 0$, in which, the LOV δ_{ijk} for $i = 0$ represents that job j is the first job to be processed on machine k. Based on this, the LOV-MIP formulation can be formulated using the constraints, as follows:

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{k=1}^{m} \delta_{ijk} = 1, \quad \forall j = 1, 2, \ldots, n \quad (28)$$

$$\sum_{\substack{i=0 \\ i \neq \ell}}^{n} \delta_{i\ell k} - \sum_{\substack{j=1 \\ j \neq \ell}}^{n} \delta_{\ell jk} = 0, \quad \forall \ell = 1, 2, \ldots, n, \ k = 1, 2, \ \ldots, m \quad (29)$$

$$C_j \geq C_i + \sum_{k=1}^{m} p_j \delta_{ijk} + M. \sum_{k=1}^{m} (\delta_{ijk} - 1), \quad \forall i = 0, 1, \ldots, n, \ j = 1, 2, \ldots, n \quad (30)$$

$$\delta_{ijk} \in \{0, 1\}, \quad \forall i, j = 1, 2, \ldots, n, \ k = 1, 2, \ldots, \ m \quad (31)$$

Constraint set (28) restricts that one job can be processed on only one machine. Constraint set (29) states the processing priorities of each job on each machine. Constraint set (30) bounds the completion time for each job, where M is an arbitrarily large value. Constraint set (31) defines the binary LOV.

E. APDV-MIP FORMULATION

Let $u_{jdk}, j = 1, 2, \ldots, n, d = 1, 2, \ldots, n, k = 1, 2, \ \ldots, m$ be the binary APDV. If job j is processed in position d regarding machine k, then $u_{jdk} = 1$; otherwise, $u_{jdk} = 0$. The APDV-MIP formulation can be constructed with the following constraints:

$$\sum_{d=1}^{n} \sum_{k=1}^{m} u_{jdk} = 1, \quad \forall j = 1, 2, \ldots, n \quad (32)$$

$$\sum_{j=1}^{n} u_{jdk} \leq 1, \quad \forall d = 1, 2, \ldots, n, \ k = 1, 2, \ \ldots, m \quad (33)$$

$$C_{dk} = \sum_{s=1}^{d-1} p_s u_{jsk} + p_d, \ \forall d = 1, 2, \ldots, n, \ k = 1, 2, \ldots, m \quad (34)$$

$$u_{jdk} \in \{0, 1\}, \quad \forall j, d = 1, 2, \ldots, n, \ k = 1, 2, \ldots, \ m \quad (35)$$

Constraint set (32) restricts that each job can only be processed by one machine on a positional date. Constraint set (33) ensures that, at most, one job is assigned to a positional date for each machine. Constraint set (34) calculates the completion time of the job, as based on the certain processing position of each machine. Constraint set (35) defines the binary APDV.

IV. COMPARATIVE ANALYSIS

To evaluate the presented MIP formulations in solving the $1||\sum C_j$, $1||C_{\max}$, $P_m||\sum C_j$, and $P_m||C_{\max}$ problems, computational experiments were conducted involving test instances with different problem sizes. All the MIP formulations were solved by the commercial software Lingo 12.0, and run on a PC with an Intel®Core$^{(TM)}$ i7-980X processor and 6 GB of RAM. The computational results of the MIP formulations were mutually contrasted, in order to identify the superior MIP formulation for dealing with a specific scheduling problem.

A. EXPERIMENT DESIGN

The computational experiments were conducted using a set of test instances, and classified with three problem sizes by considering the numbers of jobs, $n = \{15, \ 30, \ 50\}$. For each number of jobs, there were 5 test instances. For the SMSPs with different objectives, namely $1||\sum C_j$ and $1||C_{\max}$, there was a total of 30 test instances for the associated four MIP formulations. Based on these definitions of the problem's scale, this study further considered three numbers of identical machines, $m = \{2, \ 4, \ 8\}$. Thus, there were 90 test instances in total for the four MIP formulations of the $P_m||\sum C_j$ and $P_m||C_{\max}$ problems, respectively. For all the test instances, the job processing times were randomly yielded in uniform distribution, U(1,50).

To evaluate the performance of the MIP formulations, we considered the degree of deviation of the objective values from the best ones (DOVB), as follows:

$$\text{DOVB} = \frac{OBJ - OBJ^{best}}{OBJ^{best}} \times 100\% \quad (36)$$

where OBJ is the objective value obtained by applying a certain MIP formulation to solve an instance, while OBJ^{best} is the best (or optimal) objective value among the compared MIP formulations.

TABLE 1. Computational results of MIP formulations for SMSPs.

	CTV-MIP			TIV-MIP			LOV-MIP			APDV-MIP		
No.	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time
Instances for $1\|\|\sum C_j$ *with* $n=15$												
1	3413	0.03	3600	3412*	0.00	113	3412*	0.00	< 1	3412*	0.00	< 1
2	1934*	0.00	3600	1934*	0.00	58	1934*	0.00	< 1	1934*	0.00	< 1
3	1861*	0.00	3600	1861*	0.00	48	1861*	0.00	< 1	1861*	0.00	< 1
4	1554	0.52	3600	1546*	0.00	38	1546*	0.00	< 1	1546*	0.00	< 1
5	3563*	0.00	3600	3563*	0.00	112	3563*	0.00	< 1	3563*	0.00	< 1
Avg.		0.11	3600		0.00	73.8		0.00	< 1		0.00	< 1
Instances for $1\|\|C_{max}$ *with* $n=15$												
6	526*	0.00	3600	526*	0.00	120	526*	0.00	3600	526*	0.00	< 1
7	377*	0.00	3600	377*	0.00	67	377*	0.00	3600	377*	0.00	< 1
8	344*	0.00	3600	344*	0.00	55	344*	0.00	3600	344*	0.00	< 1
9	302*	0.00	3600	302*	0.00	40	302*	0.00	3600	302*	0.00	< 1
10	524*	0.00	3600	524*	0.00	128	524*	0.00	3600	524*	0.00	< 1
Avg.		0.00	3600		0.00	82		0.00	3600		0.00	< 1
Instances for $1\|\|\sum C_j$ *with* $n=30$												
11	10700	3.01	3600	10387*	0.00	1579	10387*	0.00	< 1	10387*	0.00	< 1
12	7507	3.96	3600	7221*	0.00	941	7221*	0.00	1	7221*	0.00	< 1
13	7813	3.33	3600	7561*	0.00	884	7561*	0.00	1	7561*	0.00	< 1
14	7962	2.84	3600	7742*	0.00	951	7742*	0.00	< 1	7742*	0.00	< 1
15	9304	3.17	3600	9018*	0.00	1403	9018*	0.00	1	9018*	0.00	< 1
Avg.		3.26	3600		0.00	1151.6		0.00	< 1		0.00	< 1
Instances for $1\|\|C_{max}$ *with* $n=30$												
16	874*	0.00	3600	874*	0.00	2026	874*	0.00	3600	874*	0.00	< 1
17	701*	0.00	3600	701*	0.00	1045	701*	0.00	3600	701*	0.00	< 1
18	691*	0.00	3600	691*	0.00	1096	691*	0.00	3600	691*	0.00	< 1
19	688*	0.00	3600	688*	0.00	962	688*	0.00	3600	688*	0.00	< 1
20	799*	0.00	3600	799*	0.00	1391	799*	0.00	3600	799*	0.00	< 1
Avg.		0.00	3600		0.00	1304		0.00	3600		0.00	< 1
Instances for $1\|\|\sum C_j$ *with* $n=50$												
21	24074	4.14	3600	N/A	-	-	23118*	0.00	6	23118*	0.00	< 1
22	24776	5.17	3600	N/A	-	-	23557*	0.00	9	23557*	0.00	1
23	23809	4.95	3600	N/A	-	-	22687*	0.00	9	22687*	0.00	< 1
24	22370	8.86	3600	N/A	-	-	20549*	0.00	6	20549*	0.00	1
25	28188	5.91	3600	N/A	-	-	26616*	0.00	8	26616*	0.00	< 1
Avg.		5.80	3600					0.00	7.6		0.00	< 1
Instances for $1\|\|C_{max}$ *with* $n=50$												
26	1264*	0.00	3600	N/A	-	-	1264*	0.00	3600	1264*	0.00	1
27	1319*	0.00	3600	N/A	-	-	1319*	0.00	3600	1319*	0.00	< 1
28	1284*	0.00	3600	N/A	-	-	1284*	0.00	3600	1284*	0.00	< 1
29	1177*	0.00	3600	N/A	-	-	1177*	0.00	3600	1177*	0.00	1
30	1420*	0.00	3600	N/A	-	-	1420*	0.00	3600	1420*	0.00	1
Avg.		0.00	3600		-	-		0.00	3600		0.00	< 1

*: optimal objective value.

TABLE 2. Computational results of MIP formulations for IPMSPs with two machines.

	CTV-MIP			TIV-MIP			LOV-MIP			APDV-MIP		
No.	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time
Instances for $P_2 \| \sum C_j$ *with* $n=15$												
1	1877	2.34	3600	1834*	0.00	29	1928	5.13	3600	1872	2.07	3600
2	1071	0.37	3600	1067*	0.00	21	1107	3.75	3600	1077	0.94	3600
3	1062	3.91	3600	1022*	0.00	14	1054	3.13	3600	1045	2.25	3600
4	874	2.10	3600	856*	0.00	11	958	11.91	3600	857	0.12	3600
5	1955	1.77	3600	1921*	0.00	28	1956	1.82	3600	1926	0.26	3600
Avg.		2.09	3600		0.00	20.6		5.14	3600		1.12	3600
Instances for $P_2 \| C_{max}$ *with* $n=15$												
6	263*	0.00	3600	263*	0.00	36	263*	0.00	3600	263*	0.00	3600
7	189*	0.00	3600	189*	0.00	16	189*	0.00	3600	189*	0.00	3600
8	172*	0.00	3600	172*	0.00	23	172*	0.00	3600	172*	0.00	3600
9	151*	0.00	3600	151*	0.00	11	151*	0.00	3600	151*	0.00	3600
10	262*	0.00	3600	262*	0.00	34	262*	0.00	3600	262*	0.00	3600
Avg.		0.00	3600		0.00	24		0.00	3600		0.00	3600
Instances for $P_2 \| \sum C_j$ *with* $n=30$												
11	6087	12.35	3600	5418*	0.00	334	6044	11.55	3600	5575	2.90	3600
12	4639	22.27	3600	3794*	0.00	268	4863	28.18	3600	3906	2.95	3600
13	4782	20.79	3600	3959*	0.00	255	4816	21.65	3600	4046	2.20	3600
14	4791	18.47	3600	4049*	0.00	251	4897	20.94	3600	4178	3.19	3600
15	5364	13.76	3600	4715*	0.00	323	5548	17.68	3600	4961	5.21	3600
Avg.		17.52	3600		0.00	286.2		20.00	3600		3.29	3600
Instances for $P_2 \| C_{max}$ *with* $n=30$												
16	437*	0.00	3600	437*	0.00	434	437*	0.00	3600	437*	0.00	3600
17	351*	0.00	3600	351*	0.00	238	351*	0.00	3600	351*	0.00	3600
18	346*	0.00	3600	346*	0.00	251	346*	0.00	3600	346*	0.00	3600
19	344*	0.00	3600	344*	0.00	327	344*	0.00	3600	344*	0.00	3600
20	400*	0.00	3600	400*	0.00	309	400*	0.00	3600	400*	0.00	3600
Avg.		0.00	3600		0.00	311.8		0.00	3600		0.00	3600
Instances for $P_2 \| \sum C_j$ *with* $n=50$												
21	15094	27.05	3600	11880*	0.00	1800	15660	31.81	3600	13929	17.24	3600
22	15731	29.85	3600	12115*	0.00	2181	15619	28.92	3600	13584	12.13	3600
23	15787	35.26	3600	11672*	0.00	2100	15672	34.27	3600	12567	7.67	3600
24	13321	50.54	3600	8849*	0.00	1422	12706	43.58	3600	9864	11.47	3600
25	17184	25.72	3600	13668*	0.00	2416	17754	29.89	3600	15103	10.52	3600
Avg.		33.68	3600		0.00	1983.8		33.69	3600		11.80	3600
Instances for $P_2 \| C_{max}$ *with* $n=50$												
26	632*	0.00	3600	632*	0.00	2207	632*	0.00	3600	N/A	-	-
27	665	0.76	3600	660*	0.00	2165	660*	0.00	3600	N/A	-	-
28	642*	0.00	3600	642*	0.00	2017	642*	0.00	3600	N/A	-	-
29	529	0.00	3600	528*	0.00	1459	528*	0.00	3600	N/A	-	-
30	710*	0.00	3600	710*	0.00	2546	710*	0.00	3600	N/A	-	-
Avg.		0.15	3600		0.00	2078.8		0.00	3600		-	-

TABLE 3. Computational results of MIP Formulations for IPMSPs with four machines.

	CTV-MIP			TIV-MIP			LOV-MIP			APDV-MIP		
No.	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time
Instances for $P_4 \| \sum C_j$ *with* $n = 15$												
1	1077	0.56	3600	1071*	0.00	10	1104	3.08	3600	1077	0.56	3600
2	649	1.72	3600	638*	0.00	3	716	12.23	3600	638	0.00	3600
3	639	3.57	3600	617*	0.00	3	641	3.89	3600	617	0.00	3600
4	515*	0.00	3600	515*	0.00	2	567	10.10	3600	515	0.00	3600
5	1123	1.35	3600	1108*	0.00	9	1136	2.53	3600	1110	0.18	3600
Avg.		1.44	3600		0.00	5.4		6.36	3600		0.14	3600
Instances for $P_4 \| C_{\max}$ *with* $n = 15$												
6	133	0.76	3600	132*	0.00	12	133	0.76	3600	134	1.52	3600
7	95*	0.00	3600	95*	0.00	5	95*	0.00	3600	97	2.11	3600
8	87	1.16	3600	86*	0.00	6	87	1.16	3600	87	1.16	3600
9	76*	0.00	3600	76*	0.00	4	76*	0.00	3600	77	1.32	3600
10	133	0.76	3600	132*	0.00	25	133	0.76	3600	133	0.76	3600
Avg.		0.53	3600		0.00	10.4		0.53	3600		1.37	3600
Instances for $P_4 \| \sum C_j$ *with* $n = 30$												
11	3509	19.31	3600	2941*	0.00	82	3412	16.01	3600	3032	2.79	3600
12	2493	19.57	3600	2085*	0.00	52	3021	44.89	3600	2262	8.49	3600
13	2833	30.73	3600	2167*	0.00	57	2433	12.28	3600	2337	7.84	3600
14	2732	23.68	3600	2209*	0.00	50	2656	20.24	3600	2287	3.53	3600
15	3343	30.18	3600	2568*	0.00	72	3499	36.25	3600	2660	3.58	3600
Avg.		24.69	3600		0.00	62.6		25.93	3600		5.24	3600
Instances for $P_4 \| C_{\max}$ *with* $n = 30$												
16	220	0.46	3600	219*	0.00	94	294	34.24	3600	219	0.00	3600
17	176	0.00	3600	176*	0.00	57	240	36.36	3600	183	3.98	3600
18	174	0.58	3600	173*	0.00	58	195	12.71	3600	174	0.58	3600
19	174	1.16	3600	172*	0.00	58	184	6.98	3600	189	9.88	3600
20	203	1.50	3600	200*	0.00	82	205	2.50	3600	201	0.50	3600
Avg.		0.74	3600		0.00	69.8		18.55	3600		2.98	3600
Instances for $P_4 \| \sum C_j$ *with* $n = 50$												
21	10179	62.37	3600	6269*	0.00	472	11895	89.74	3600	N/A	-	-
22	9537	49.01	3600	6400*	0.00	572	9606	49.02	3600	N/A	-	-
23	8454	37.06	3600	6168*	0.00	506	9650	56.45	3600	N/A	-	-
24	6814	44.95	3600	4701*	0.00	329	6845	45.61	3600	N/A	-	-
25	9701	34.68	3600	7203*	0.00	609	12974	80.12	3600	N/A	-	-
Avg.		45.61	3600		0.00	497.6		64.18	3600		-	-
Instances for $P_4 \| C_{\max}$ *with* $n = 50$												
26	349	10.44	3600	316*	0.00	496	N/A	-	-	N/A	-	-
27	333	0.90	3600	330*	0.00	562	N/A	-	-	N/A	-	-
28	393	22.43	3600	321*	0.00	593	N/A	-	-	N/A	-	-
29	295	11.74	3600	264*	0.00	367	N/A	-	-	N/A	-	-
30	371	4.51	3600	355*	0.00	705	N/A	-	-	N/A	-	-
Avg.		10.00	3600		0.00	544.6		-	-		-	-

TABLE 4. Computational Results of MIP Formulations for IPMSPs with eight machines.

	CTV-MIP			TIV-MIP			LOV-MIP			APDV-MIP		
No.	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time	OBJ	DOVB	Time
Instances for $P_8 \| \sum C_j$ *with* $n=15$												
1	689*	0.00	3600	689*	0.00	3	751	9.00	3600	689	0.00	3600
2	443*	0.00	3600	443*	0.00	3	544	22.79	3600	443	0.00	3600
3	417*	0.00	3600	417*	0.00	2	438	5.04	3600	417	0.00	3600
4	359*	0.00	3600	359*	0.00	2	384	6.96	3600	359	0.00	3600
5	710	0.56	3600	706*	0.00	3	722	1.69	3600	706*	0.00	3600
Avg.		0.11	3600		0.00	2.6		9.09	3600		0.00	3600
Instances for $P_8 \| C_{\max}$ *with* $n=15$												
6	75*	0.00	3600	75*	0.00	4	77	2.67	3600	80	6.67	3600
7	50*	0.00	26	50*	0.00	3	58	16.00	3600	50*	0.00	3600
8	50*	0.00	25	50*	0.00	5	50*	0.00	3600	50*	0.00	3600
9	49*	0.00	19	49*	0.00	4	49*	0.00	3600	49*	0.00	3600
10	70*	0.00	3600	70*	0.00	3	78	11.42	3600	78	11.42	3600
Avg.		0.00	1454		0.00	3.8		6.01	3600		3.61	3600
Instances for $P_8 \| \sum C_j$ *with* $n=30$												
11	2015	17.56	3600	1714*	0.00	38	2231	30.16	3600	1786	4.20	3600
12	1603	29.27	3600	1240*	0.00	39	3794	205.97	3600	1302	5.00	3600
13	1904	48.17	3600	1285*	0.00	43	1917	49.18	3600	1344	4.59	3600
14	1783	36.83	3600	1303*	0.00	37	1646	26.32	3600	1386	6.37	3600
15	1986	31.70	3600	1508*	0.00	38	2208	46.41	3600	1566	3.85	3600
Avg.		32.70	3600		0.00	39		71.60	3600		4.80	3600
Instances for $P_8 \| C_{\max}$ *with* $n=30$												
16	113	2.70	3600	110*	0.00	78	135	22.73	3600	122	10.91	3600
17	91	3.41	3600	88*	0.00	46	108	22.73	3600	121	37.50	3600
18	100	14.94	3600	87*	0.00	44	126	44.83	3600	122	40.23	3600
19	108	25.58	3600	86*	0.00	47	143	66.27	3600	136	58.14	3600
20	133	33.00	3600	100*	0.00	44	131	31.00	3600	179	79.00	3600
Avg.		15.92	3600		0.00	51.8		37.51	3600		45.15	3600
Instances for $P_8 \| \sum C_j$ *with* $n=50$												
21	4727	36.03	3600	3475*	0.00	120	8497	144.52	3600	N/A	-	-
22	5387	51.49	3600	3556*	0.00	134	7801	119.38	3600	N/A	-	-
23	5269	53.75	3600	3427*	0.00	118	5594	63.23	3600	N/A	-	-
24	4055	30.22	3600	3114*	0.00	151	4709	51.22	3600	N/A	-	-
25	5501	38.22	3600	3980*	0.00	151	8691	118.37	3600	N/A	-	-
Avg.		41.94	3600		0.00	134.8		99.34	3600		-	-
Instances for $P_8 \| C_{\max}$ *with* $n=50$												
26	200	26.58	3600	158*	0.00	152	413	161.39	3600	N/A	-	-
27	307	86.06	3600	165*	0.00	141	303	83.64	3600	N/A	-	-
28	232	44.10	3600	161*	0.00	132	307	90.68	3600	N/A	-	-
29	187	41.67	3600	132*	0.00	94	249	88.64	3600	N/A	-	-
30	212	19.10	3600	178*	0.00	155	285	60.11	3600	N/A	-	-
Avg.		43.50	3600		0.00	134.8		96.89	3600		-	-

For each test instance of SMSPs and IPMSPs, 1 hour (3600 s) was set as the maximum computational time. When the optimal solution cannot be found in an hour, the lower bound reached is substituted and used for performance measuring. Note that, in some cases, an MIP formulation may not finish the solution search within 3600 s; however, the currently found lower bound can be validated to be optimal if this lower bound is the same as the optimal objective value, as obtained by other formulations.

B. COMPUTATIONAL RESULTS OF MIP FORMULATIONS FOR SMSPS

Regarding the 30 test instances with respect to the $1||\sum C_j$ and $1||C_{\max}$ problems, the computational results of the four MIP formulations are shown in Table 1. For the instances with $n = 15$, with the exception of the CTV-MIP formulation, all other MIP formulations always found the optimal solutions with a zero DOVB, in which the TIV-MIP formulation required the average completion time of 77.9 s, while the LOV-MIP formulation was efficient enough to only address the $1||\sum C_j$ instances. In contrast, the APDV-MIP formulation is completely efficient in solving all the instances within 1 s on average. Regarding the instances with $n = 30$, with the exception of the CTV-MIP formulation which failed to reach the optimality of instances 11 to 15 within 3600 s, most of the MIP formulations found the optimal solutions, where the TIV-MIP formulation needed 1227.80 s on average, while the LOV-MIP formulation still presented a significant difference of computational efficiency under different objectives. In contrast, the APDV-MIP formulation exhibited its superior efficiency in finding the optimal solutions within 1 s. In solving the instances with $n = 50$, the CTV-MIP formulation could not finish the solution search in 3600 s, although the found lower bounds; for example, 26 to 30, can be validated to be optimal. The worst situation occurred with the TIV-MIP formulation, which failed to find a feasible solution within 3600 s, and reflects its known drawback regarding model size, as resulted from the planning horizon definition $T = \sum p_j$. On the other hand, while the LOV-MIP formulation showed its efficiency in minimizing the $1||\sum C_j$ instances; it still fell behind the APDV-MIP formulation with less than 1 s spent.

Clearly, with the increased number of jobs, the DOVB values of the CTV-MIP formulation increased significantly, which increased the computational times of the LOV-MIP and TIV-MIP formulations. In contrast, the APDV-MIP formulation outperformed the others in solving the $1||\sum C_j$ and $1||C_{\max}$ instances.

C. COMPUTATIONAL RESULTS OF MIP FORMULATIONS FOR IPMSPS

The computational results of the MIP formulations in solving 30 test instances with two parallel machines are shown in Table 2. For the test instances with $n = 15$, the TIV-MIP formulation obtains the optimal solutions with zero DOVB on average of 22.30 s, while all other MIP formulations either cannot reach the optimality by 3600 s or cannot finish the solution search even though the optimal lower bounds have been found via contrasts. This weakness is reflected in the average DOVB of 1.05% for the CTV-MIP, 2.57% for the LOV-MIP, and 0.56% for the APDV-MIP formulation. A similar circumstance arises in solving the instances with $n = 30$ and $n = 50$, in that only the TIV-MIP formulation consistently found the optimal solutions and yielded, on average, a zero DOVB, thus, its performance surpassed those of the other MIP formulations. This outstanding performance comes from the two parallel machines that reduced planning horizon $T = \lceil \sum p_j/2 \rceil$, as well as the computational effort.

Table 3 summarizes the computational results of the MIP formulations in solving 30 test instances with four parallel machines, and the solution results are similar to those of the previous case of two parallel machines. The TIV-MIP formulation exhibits excellent performance with an average zero DOVB, with the average computational times of 7.90, 66.20, and 521.10 s in solving the instances with $n = 15$, 30 and 50, respectively. Although the computational time of the TIV-MIP formulation increases with the increment of jobs, its effectiveness and efficiency give it a significant advantage over the other formulations, which could not find the optimal solutions in most instances or found no feasible solutions within 3600 s. Table 4 shows the computational results of the MIP formulations in solving 30 test instances with eight parallel machines. The TIV-MIP formulation still outperformed the others, which found the optimal solutions with zero DOVB on average 3.20, 45.40 and 134.80 s for the test instances with $n = 15$, 30, and 50, respectively. While the other MIP formulations reached optimality in a few instances with $n = 15$; they failed to find the optimal solutions in most instances. In addition, with the exception of TIV-MIP, the other formulations yielded significant DOVBs, as compared with the zero DOVB of the TIV-MIP formulation. In the cases considering 50 jobs, the APDV-MIP formulation could not find even one feasible solution.

Focusing on the top-performing TIV-MIP formulation, the average computational times (ACTs) computed over 15 test instances with different numbers of parallel machines and performance criteria, as summarized in Table 5

TABLE 5. Total average computational time of the TIV-MIP formulation.

Types of IPMSPs	Objective	Total average time (s)
Two parallel machines	$\sum C_j$	763.53
	$C_{\max}$	804.87
Four parallel machines	$\sum C_j$	188.53
	$C_{\max}$	208.27
Eight parallel machines	$\sum C_j$	58.80
	$C_{\max}$	70.13

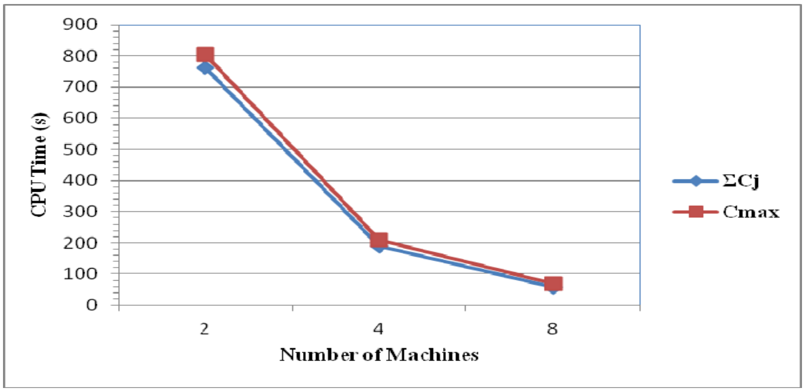
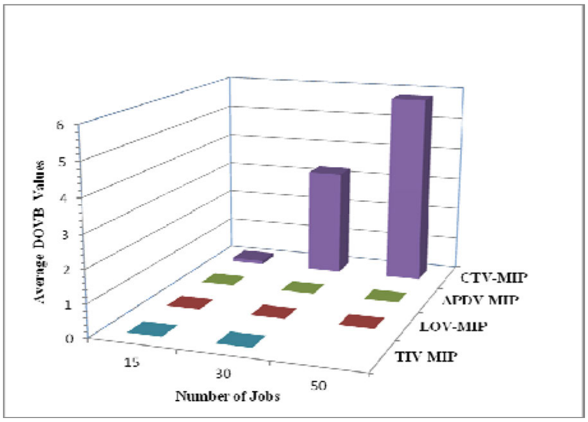
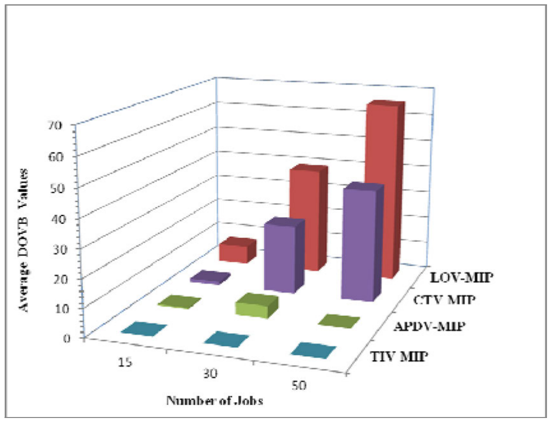


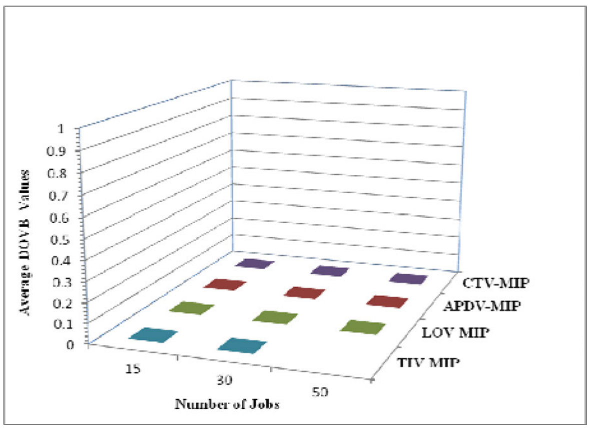
FIGURE 3. Average computational times of the TIV-MIP formulation.



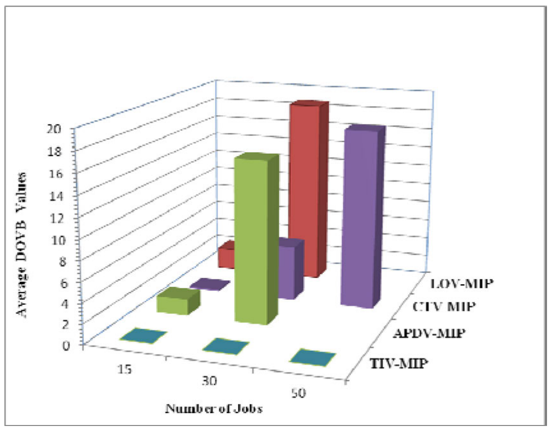
(a) Average DOVB values in solving $1\|\sum C_j$



(b) Average DOVB values in solving $P_m\|\sum C_j$



(c) Average DOVB values in solving $1\|C_{\max}$



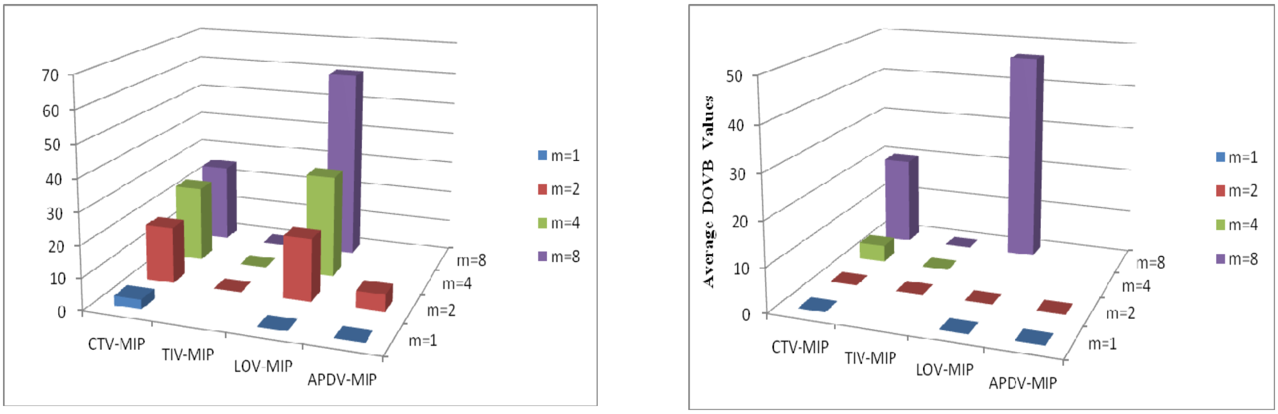
(d) Average DOVB values in solving $P_m\|C_{\max}$

FIGURE 4. Average DOVB values of different MIP formulations with varying numbers of jobs and performance measures.

(see Fig. 3), which shows that the time spent decreases with the increase in the number of parallel machines. Although only the objectives of total completion time and makespan were considered in this study, it is apparent that the number of parallel machines can effectively reduce planning horizon $T = \lceil \sum p_j/m \rceil$ and the computational time for the TIV-MIP formulation.

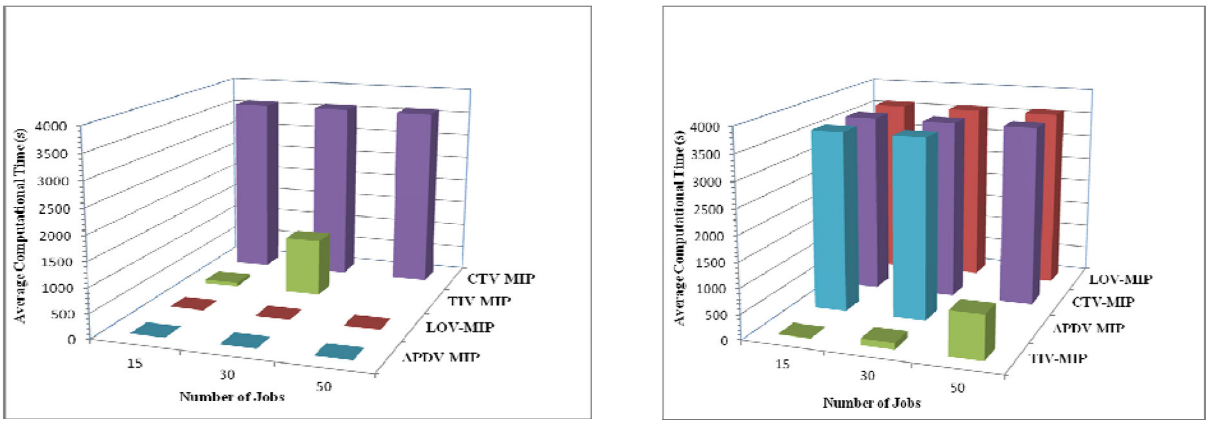
D. COMPARATIVE ANALYSIS OF DIFFERENT MIP FORMULATIONS BETWEEN SMSPS AND IPMSPS

In order to compare the effectiveness of different MIP formulations between SMSPs and IPMSPs, the average DOVB values computed over 5 test instances with varying numbers of jobs and performance criteria are depicted in Figure 4. As shown in Figure 4, with respect to the performance

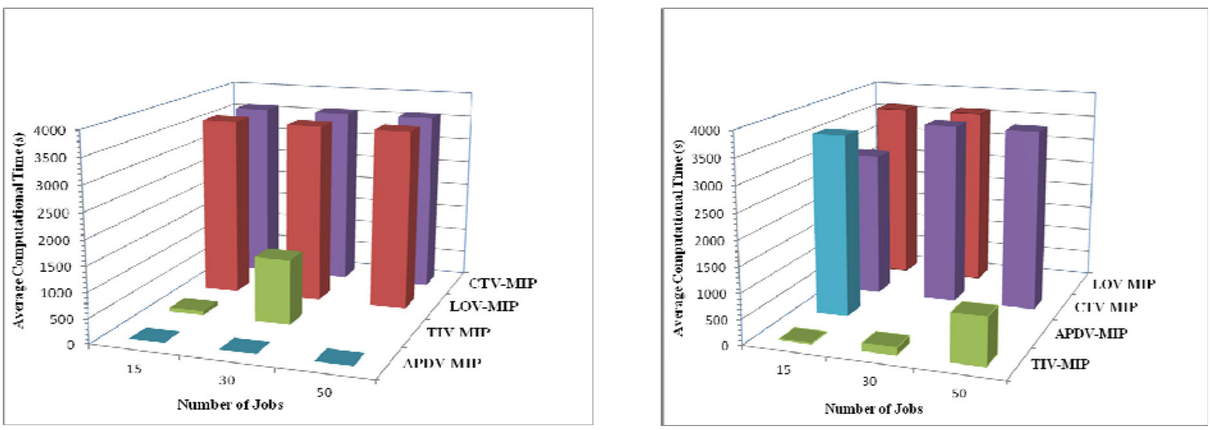


(a) Average DOVB values with respect to $\sum C_j$ (b) Average DOVB values with respect to $C_{\max}$

FIGURE 5. **Average DOVB values of different MIP formulations with varying numbers of machines and performance measures.**



(a) ACTs in solving $1\|\sum C_j$ (b) ACTs in solving $P_m\|\sum C_j$



(c) ACTs in solving $1\|C_{\max}$ (d) ACTs in solving $P_m\|C_{\max}$

FIGURE 6. **Average computational times (ACTs) of different MIP formulations.**

criterion of $\sum C_j$, the effectiveness of TIV-MIP, LOV-MIP, and APDV-MIP formulations is higher than that of the CTV-MIP formulation in solving SMSPs; while the effectiveness of the TIV-MIP formulation is higher than that of APDV-MIP, CTV-MIP, and LOV-MIP formulations in solving IPMSPs. However, regarding the performance criterion of $C_{\max}$, the effectiveness of the TIV-MIP formulation is lower than that of APDV-MIP, CTV-MIP, and LOV-MIP formulations in solving SMSPs with $n = 50$; while the effectiveness of the TIV-MIP formulation is higher than that of APDV-MIP, CTV-MIP, and LOV-MIP formulations in solving IPMSPs.

To further compare the effectiveness of different MIP formulations between SMSPs and IPMSPs, Figure 5 depicts the average DOVB values computed over 30 test instances with varying numbers of machines and performance criteria. The TIV-MIP formulation is unable to achieve optimal solutions for all test instances with $n = 30$ in solving SMSPs. As shown in Figure 5, the average DOVB values obtained by the TIV-MIP formulation are very robust in solving both SMSPs and IPMSPs with respect to different numbers of machines and performance measures. In regards to the performance criterion of $\sum C_j$, the greater the number of machines is, the larger are the DOVB values of the LOV-MIP, APDV-MIP, and CTV-MIP formulations in solving the addressed problems.

Khowala *et al.* [18], Keha *et al.* [19], and Baker and Keller [20] found that the performances of MIP formulations are depend on the sum of the processing times of all the jobs in solving SMSPs, when looking at the performance measures of total tardiness, total weighted tardiness, total weighted completion time, maximum lateness, and number of tardy jobs. However, in this study the performances of MIP formulations are independent of the sum of the processing times of all the jobs in solving SMSPs, when looking at the performance criterion of $C_{\max}$. On the other hand, Unlu and Mason [25] noted that the performances of MIP formulations depend on the sum of the processing times of the jobs in solving IPMSPs with respect to total weighted completion time and maximum completion time performance measures, respectively. In this present study, the performances of the TIV-MIP formulation are independent of the sum of the processing times of all the jobs in solving IPMSPs, in regards to the performance criteria of $C_{\max}$ and $\sum C_j$.

In order to compare the efficiency of different MIP formulations between SMSPs and IPMSPs, the ACTs of test instances with different numbers of jobs and performance criteria are depicted in Figure 6. The ACTs of the SMSPs and IPMSPs are computed over five and 15 test instances, respectively. As shown in Figure 6, with respect to the performance criterion of $\sum C_j$, the efficiency of APDV-MIP and LOV-MIP formulations is higher than that of TIV-MIP and CTV-MIP formulations in solving SMSPs; while the efficiency of the TIV-MIP formulation is higher than that of APDV-MIP, CTV-MIP, and LOV-MIP formulations in solving IPMSPs. However, regarding the performance criterion of $C_{\max}$, the efficiency of the APDV-MIP formulation is higher than that of TIV-MIP, LOV-MIP, and CTV-MIP formulations in solving SMSPs and IPMSPs.

V. CONCLUSION

This study analyzed and presented four types of MIP formulations on SMSPs and IPMSPs, simultaneously, where the aim is to minimize the total completion time and makespan of jobs. The MIP formulations of CTV-MIP, TIV-MIP, LOV-MIP, and APDV-MIP, as based on four different types of decision variables, are used to establish eight MIP mathematical formulations with respect to SMSPs or IPMSPs. Through a series of experiments regarding computational time and the ability to reach optimality, it is found that the APDV-MIP formulation outperforms all other MIP formulations in solving SMSPs. Regarding the instances of IPMSPs, the performance of the TIV-MIP formulation for all problem sizes surpass that of the other MIP formulations, with an increase in the numbers of parallel machines effectively reducing the key factor of computational effort, i.e., the planning horizon. Clearly, these experimental results will provide potential researchers with identified and top-performing MIP formulations for dealing with specific scheduling problems to be solved, as well as providing benchmarks for the development of potential solution algorithms. In future research, other scheduling criteria and various conditions, such as learning effect and sequence-dependent setup time, will be taken into consideration, in order to extend the evaluation ability of mathematical programming models.

ACKNOWLEDGMENT

(Kuo-Cing Ying and Chen-Yang Cheng are co-first authors.).

REFERENCES

[1] C.-C. Lu, S.-W. Lin, and K.-C. Ying, "Robust scheduling on a single machine to minimize total flow time," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1682–1691, 2012.

[2] Y. Yin, W. Wang, D. Wang, and T. C. E. Cheng, "Multi-agent single-machine scheduling and unrestricted due date assignment with a fixed machine unavailability interval," *Comput. Ind. Eng.*, vol. 111, pp. 202–215, Sep. 2017.

[3] D. Wang, Y. Yin, and T. C. E. Cheng, "Parallel-machine rescheduling with job unavailability and rejection," *Omega*, vol. 81, pp. 246–260, Dec. 2018.

[4] Y. Yin, Y. Chen, K. Qin, and D. Wang, "Two-agent scheduling on unrelated parallel machines with total completion time and weighted number of tardy jobs criteria," *J. Scheduling*, vol. 22, no. 3, pp. 315–333, Jun. 2019. doi: 10.1007/s10951-018-0583-z.

[5] S.-W. Lin, C.-C. Lu, and K.-C. Ying, "Minimization of total tardiness on unrelated parallel machines with sequence- and machine-dependent setup times under due date constraints," *Int. J. Adv. Manuf. Technol.*, vol. 53, nos. 1–4, pp. 353–361, Mar. 2011.

[6] K.-C. Ying, "Scheduling identical wafer sorting parallel machines with sequence-dependent setup times using an iterated greedy heuristic," *Int. J. Prod. Res.*, vol. 50, no. 10, pp. 2710–2719, Jul. 2012.

[7] A. E. Jahromi and R. I. Yazdeli, "A mixed integer-programming model for periodic routing of special vessels in offshore oil industry," *Int. J. Ind. Eng.*, vol. 22, no. 5, pp. 524–528, 2015.

[8] M. Mrad and H. K. Alfares, "Optimum multi-period, multi-plant, and multi-supplier production planning for multi-grade petrochemicals," *Int. J. Ind. Eng.*, vol. 23, no. 6, pp. 399–411, Nov. 2016.

[9] D. J. Noh, B. I. Kim, H. Cho, and J. S. Park, "A multi-leg load-planning algorithm for a high-speed freight train," *Int. J. Ind. Eng.*, vol. 23, no. 3, pp. 183–194, Mar. 2016.

[10] M. Queyranne and A. S. Schulz, "Polyhedral approaches to machine scheduling," Dept. Math., Tech. Univ. Berlin, Berlin, Germany, Tech. Rep. 408/1994, 1994.

[11] E. Balas, "On the facial structure of scheduling polyhedra," *Math. Program.*, vol. 24, pp. 179–218, 1985.

[12] M. Queyranne, "Structure of a simple scheduling polyhedron," *Math. Program.*, vol. 58, nos. 1–3, pp. 263–285, Jan. 1993.

[13] J. P. Sousa and L. A. Wolsey, "A time indexed formulation of non-preemptive single machine scheduling problems," *Math. Program.*, vol. 54, nos. 1–3, pp. 353–367, Feb. 1992.

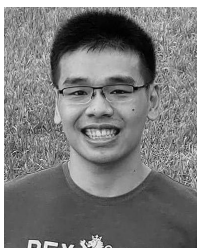
[14] J. M. van den Akker, C. P. M. van Hoesel, and M. W. P. Savelsbergh, "A polyhedral approach to single-machine scheduling problems," *Math. Program.*, vol. 85, no. 3, pp. 541–572, Aug. 1999.

[15] M. E. Dyer and L. A. Wolsey, "Formulating the single machine sequencing problem with release dates as a mixed integer program," *Discrete Appl. Math.*, vol. 26, nos. 2–3, pp. 255–270, Mar. 1990.
[16] J. B. Lasserre and M. Queyranne, "Generic scheduling polyhedral and a new mixed-integer formulation for single-machine scheduling," in *Proc. 2nd IPCO Conf.*, 1992, pp. 136–149.
[17] S. Dauzère-Pérès, "An efficient formulation for minimizing the number of late jobs in single-machine scheduling," in *Proc. IEEE 6th Int. Conf. Emerg. Technol. Factory Automat. (EFTA)*, Sep. 1997, pp. 442–445.
[18] K. Khowala, A. Keha, and J. Fowler, "A comparison of different formulations for the non-preemptive single machine total weighted tardiness scheduling problem," in *Proc. 2nd Multidisciplinary Int. Conf. Scheduling, Theory Appl. (MISTA)*, Jul. 2005, pp. 643–651.
[19] A. B. Keha, K. Khowala, and J. W. Fowler, "Mixed integer programming formulations for single machine scheduling problems," *Comput. Ind. Eng.*, vol. 56, no. 1, pp. 357–367, 2009.
[20] K. R. Baker and B. Keller, "Solving the single-machine sequencing problem using integer programming," *Comput. Ind. Eng.*, vol. 59, no. 4, pp. 730–735, Nov. 2010.
[21] G. Rabadi, R. J. Moraga, and A. Al-Salem, "Heuristics for the unrelated parallel machine scheduling problem with setup times," *J. Intell. Manuf.*, vol. 17, no. 1, pp. 85–97, Feb. 2006.
[22] P. L. Rocha, M. G. Ravetti, G. R. Mateus, and P. M. Pardalos, "Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times," *Comput. Oper. Res.*, vol. 35, no. 4, pp. 1250–1264, Apr. 2008.
[23] S. Kedad-Sidhoum, Y. R. Solis, and F. Sourd, "Lower bounds for the earliness–tardiness scheduling problem on parallel machines with distinct due dates," *Eur. J. Oper. Res.*, vol. 189, no. 3, pp. 1305–1316, Sep. 2008.
[24] K. Li and S.-L. Yang, "Non-identical parallel-machine scheduling research with minimizing total weighted completion times: Models, relaxations and algorithms," *Appl. Math. Model.*, vol. 33, no. 4, pp. 2145–2158, Apr. 2009.
[25] Y. Unlu and S. J. Mason, "Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems," *Comput. Ind. Eng.*, vol. 58, no. 4, pp. 785–800, May 2010.
[26] D. Wang, J. Zhu, X. Wei, T. C. E. Cheng, Y. Yin, and Y. Wang, "Integrated production and multiple trips vehicle routing with time windows and uncertain travel times," *Comput. Oper. Res.*, vol. 103, pp. 1–12, Mar. 2019.
[27] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Ann. Discrete Math.*, vol. 5, pp. 287–362, Jan. 1979.

KUO-CHING YING is currently a Distinguished Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology. He is an (Senior) Editor of ten international journals. His research interests include the operations scheduling and combinatorial optimization, in which he has published more than 100 academic research articles in refereed international journals, such as *Applied Intelligence*, *Applied Soft Computing*, *Computers and Operations Research*, *Computers and Industrial Engineering*, *European Journal of Industrial Engineering*, *European Journal of Operational Research*, IEEE Access, *International Journal of Production Economics*, *International Journal of Advanced Manufacturing Technology*, *International Journal of Innovational Computing*, *Information and Control*, *International Journal of Production Research*, *Journal of the Operational Research Society*, *OMEGA-The International Journal of Management Sciences*, *Production Planning & Control*, *Transportation Research Part E: Logistics and Transport Review*, among others.

CHEN-YANG CHENG received the Ph.D. degree in industrial and manufacturing engineering from Penn State University. He is currently a Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology. His research interests include computer integrated manufacturing, human computer interaction, distributed systems and control, and intelligent systems.

SHIH-WEI LIN received the bachelor's, master's, and Ph.D. degrees in industrial management from the National Taiwan University of Science and Technology, Taiwan, in 1996, 1998, and 2000, respectively. He is currently a Professor with the Department of Information Management, Chang Gung University, Taiwan. He is also with the Department of Neurology, Linkuo Chang Gung Memorial Hospital, Taoyuan, Taiwan, and also with the Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei, Taiwan. His articles have appeared in *Computers & Operations Research*, *European Journal of Operational Research*, *Journal of the Operational Research Society*, *European Journal of Industrial Engineering*, *International Journal of Production Research*, *International Journal of Advanced Manufacturing Technology*, *Knowledge and Information Systems*, *Applied Soft Computing*, *Applied Intelligence*, and *Expert Systems with Applications*. His current research interests include meta-heuristics and data mining.



CHIA-YANG HUNG received the master's degree in industrial engineering and management from the National Taipei University of Technology, Taiwan, in 2015. He is currently a Senior Engineer with United Microelectronics Corporation (UMC).