Computers & Industrial Engineering 56 (2009) 357–367

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Mixed integer programming formulations for single machine scheduling problems

Ahmet B. Keha *, Ketan Khowala, John W. Fowler

Department of Industrial Engineering, Arizona State University, P.O. Box 875906, Tempe, AZ 85287-5906, USA

ARTICLE INFO

Article history:
Received 16 November 2007
Received in revised form 23 March 2008
Accepted 11 June 2008
Available online 18 June 2008

Keywords:
Single machine scheduling
Mixed integer programming
Valid inequalities

ABSTRACT

In this paper, the computational performance of four different mixed integer programming (MIP) formulations for various single machine scheduling problems is studied. Based on the computational results, we discuss which MIP formulation might work best for these problems. The results also reveal that for certain problems a less frequently used MIP formulation is computationally more efficient in practice than commonly used MIP formulations. We further present two sets of inequalities that can be used to improve the formulation with assignment and positional date variables.

1. Introduction

Scheduling is motivated by questions that arise in production planning, in balancing processes sent to compute nodes, in telecommunication and generally in all situations in which scarce resources have to be allocated to activities over time. Although finding a feasible solution to a scheduling problem is often easy, it is usually nontrivial to find an optimal solution to a scheduling problem. In this paper, we approach scheduling problems from the point of view of a practitioner who does not have expertise in scheduling and integer programming. Formulating a problem as a mixed integer program (MIP) and using the default settings of one of the commercially available software to solve this model is the first thing that a practitioner without an expertise in scheduling would do instead of using problem specific algorithms. Therefore this paper is focused on solving scheduling problems using mixed integer programming formulations.

In this paper, we compare the computational performance of different mixed integer programming (MIP) formulations for different single machine scheduling problems. The MIP formulations for scheduling problems are often classified based on the choice of the decision variables. The different decision variables used to distinguish four different MIP formulations in this paper are: (i) completion time variables Balas (1985) (ii) time index variables Sousa and Wolsey (1992) (iii) linear ordering variables Dyer and Wolsey (1990) and (iv) assignment and positional date variables Lasserre and Queyranne (1992). Queyranne and Schulz (1994) give a comprehensive survey of these MIP formulations. We complement this paper by comparing the computational performances of these formulations.

We study various single machine problems, where n jobs are processed through one machine and there is no preemption allowed while processing the jobs. Let p_j, d_j, w_j, r_j, C_j and S_j be the processing time, due date, weight, release date, completion time and start time of job j, respectively. We define $N \in \{1, 2, \ldots, n\}$ as the set of the jobs. The lateness of job j, L_j, is defined as $L_j = C_j - d_j$ and the tardiness of job j, T_j, is defined as $T_j = \max\{C_j - d_j, 0\}$. A binary variable U_j is defined to count the number of tardy jobs such that U_j is equal to 1 if job j is tardy, i.e. $C_j > d_j$ and 0 otherwise. The objective functions that are studied in this paper are total weighted completion time $[\sum w_j C_j]$, maximum lateness $[L_{\max}]$, number of tardy jobs $[\sum U_j]$ and total weighted tardiness $[\sum w_j T_j]$. In the scheduling notation of Graham, Lawler, Lenstra, and Rinnooy Kan (1979), the problems studied in this paper are denoted as $1||\sum w_j c_j, ||\sum L_{\max}||, 1||\sum U_j, 1||\sum w_j T_j, 1|r_j|\sum w_j C_j, 1|r_j|\sum L_{\max}, 1|r_j|\sum U_j$ and $1|r_j|\sum w_j T_j$.

Three out of the eight single machine problems that we study are solvable in polynomial time by well known algorithms: WSPT (weighted shortest processing time first) for total weighted competition time Smith (1956), EDD (earliest due date) for maximum lateness Jackson (1955) and Moore's algorithm for number of tardy jobs Moore (1968). However with the release date constraints, these problems become NP-hard Lenstra, Rinnooy Kan, and Brucker (1977), Kise, Ibaraki, and Mine (1978). The total weighted tardiness problem is NP-hard with and without the release dates Lawler (1977). We wanted to get a wider understanding of the computational efficiencies and behavior of different MIP formulations and therefore we also considered some easy problems as well as the harder problems.

* Corresponding author.
E-mail address: Ahmet.Keha@asu.edu (A.B. Keha).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.06.008

Table 1
Previous research on formulations for single machine scheduling problems

Performance measure	Different MIP formulations			
	(1) Completion time variables	(2) Time index variables	(3) Linear ordering variables	(4) Assignment and positional date variables
$\sum w_jC_j$	Balas (1985), Queyranne and Wang (1991), Queyranne (1993), Queyranne and Schulz (1994)	Abdul-Razaq et al. (1990), Queyranne and Schulz (1994), Šorić (2000), Sousa and Wolsey (1992), van den Akker et al. (1999)	Blazewicz et al. (1991), Chudak and Hochbaum (1999), Dyer and Wolsey (1990), Potts and Van Wassenhove (1983), Queyranne and Schulz (1994)	Lasserre and Queyranne (1992), Queyranne and Schulz (1994)
$L_{\max}$			Blazewicz et al. (1991)	
$\sum U_j$		Queyranne and Schulz (1994)		Dauzère-Pérès (1997), Dauzère-Pérès and Sevaux (2003)
$\sum w_jT_j$	Khowala et al. (2005), Queyranne (1993)	Abdul-Razaq et al. (1990), Khowala et al. (2005), Queyranne and Schulz (1994)	Khowala et al. (2005), Blazewicz et al. (1991)	Khowala et al. (2005)

2. MIP formulations

This section lists the four different MIP formulations used to model single machine scheduling problems.

2.1. Completion time variables [F1]

In the first MIP formulation we use completion time variable, C_j, to model the problems. We also introduce a binary variable, y_{jk}, which is equal to 1 if job j is processed before job k and equal to 0 otherwise. The constraints of the MIP formulation with completion time variables are given below. These constraints could also be written in terms of the start time variables.

$$C_j \geqslant p_j \quad \forall j \in N, \tag{1.1}$$

$$C_j + p_k \leqslant C_k + M(1 - y_{jk}) \quad \text{for } j,k \in N \text{ and } j < k, \tag{1.2}$$

$$C_k + p_j \leqslant C_j + My_{jk} \quad \text{for } j,k \in N \text{ and } j < k, \tag{1.3}$$

$$C_j \geqslant 0 \quad \forall j \in N, \tag{1.4}$$

$$y_{jk} \in \{0,1\} \quad \forall j,k \in N \text{ and } j < k. \tag{1.5}$$

Constraint set (1.1) ensures that the completion time of each job is greater than or equal to its processing time. Constraint sets (1.2) and (1.3) are disjunctive constraints which enforce that either job j is processed before job k or job k is processed before job j for any pair of jobs. Further, constraint sets (1.4) and (1.5) are the non-negativity and integrality constraints. In this formulation, the value of big M is generally taken to be equal to the sum of the processing times of all jobs. For the problems with release dates, the value of M is taken to be greater than the sum of processing time of all the jobs and the maximum value of the release date for all the jobs.

The objective function for minimizing the total weighted completion time can be written as $\sum_{j=1}^{n} w_jC_j$. The problem of minimizing the maximum lateness can be modeled by minimizing *LMAX* as the objective function and adding the following constraints to (1.1)–(1.5):

$$LMAX \geqslant (C_j - d_j) \quad \forall j \in N. \tag{1.6}$$

The problem of minimizing the number of tardy jobs is formulated by minimizing $\sum_{j=1}^{n} U_j$ as the objective function and adding the following constraints to (1.1)–(1.5):

Table 2
Results for single machine total weighted completion time problem $1||\sum w_jC_j$ for P = U [1, 100]

Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
Completion time variables [F1]	20	0	1882569	0	0	3 [60.78%]	0
	40	0	395559	0	0	3 [88.23%]	0
	60	0	172588	0	0	3 [93.41%]	0
	100	0	45430	0	0	3 [96.79%]	0
Time index variables [F2]	20	0	0	3 [41.14]	0	0	3
	40	0	0	3 [478.9]	0	0	3
	60	0	0	3 [1926.48]	0	0	3
	100	3	0	0	3	0	0
Linear ordering variables [F3]	20	0	0	3 [0.1]	0	0	3
	40	0	0	3 [1.09]	0	0	3
	60	0	0	3 [4.25]	0	0	3
	100	0	0	3 [22.92]	0	0	3
Positional & assignment variables [F4]	20	0	2813399	0	0	3 [87.73%]	0
	40	0	475664	0	0	3 [98.20%]	0
	60	0	164150	0	0	3 [99.25%]	0
	100	0	1108	0	3	0	0

Table 3
Results for single machine maximum lateness problem $1||\sum L_{\max}$ for P = U [1, 100]

Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
Completion time variables [F1]	20	0	1938442	5 [2.93]	0	13 [344.84%]	16
	40	0	559034	4 [5.29]	0	14 [276.4%]	14
	60	0	213123	3 [62.77]	0	15 [341.40%]	10
	100	0	41459	0	1	17 [243.97%]	9
Time index variables [F2]	20	0	269	6 [767.6]	3	9 [147.54%]	6
	40	12	4	3 [413.6]	15	0	3
	60	18	0	0	18	0	0
	100	18	0	0	18	0	0
Linear ordering variables [F3]	20	0	192698	6 [0.18]	0	12 [156.16%]	18
	40	0	232	6 [557.4]	12	0	6
	60	0	2	3 [239.1]	15	0	3
	100	16	0	0	18	0	0
Positional & assignment variables [F4]	20	0	259094	17 [6.7]	0	1 [34.41%]	18
	40	0	623075	10 [40.52]	0	8 [149.73%]	12
	60	0	382129	6 [30.66]	0	12 [87.40%]	8
	100	0	86241	7 [261.93]	0	11 [102.20%]	12

$$C_j \leqslant d_j + MU_j \quad \forall j \in N, \tag{1.7}$$
$$U_j \in \{0,1\} \quad \forall j \in N. \tag{1.8}$$

The problem of minimizing the total weighted tardiness is formulated by minimizing the $\sum_{j=1}^{n} w_j T_j$ as the objective function and adding the following constraints to (1.1)–(1.5):

$$T_j \geqslant C_j - d_j \quad \forall j \in N, \tag{1.9}$$
$$T_j \geqslant 0 \quad \forall j \in N. \tag{1.10}$$

To complete the formulation using completion time variables for the problems with the release date constraints, the Eq. (1.1) is replaced by

$$C_j \geqslant p_j + r_j \quad \forall j \in N. \tag{1.11}$$

Balas (1985) presented the first work on formulating scheduling problems using disjunctive constraints. This MIP formulation is also studied by Queyranne and Wang (1991) and Queyranne (1993).

2.2. Time index variables [F2]

In the time index variables formulation, the planning horizon is discretized into the periods 1, 2, 3,...T, where period t starts at time $t-1$ and ends at time t. We introduce a binary time index variable, x_{jt}, which is equal to 1 if job j starts at time t and is equal to 0 otherwise. The constraints of the MIP formulation with time index variables are as follows:

$$\sum\nolimits_{t=1}^{T-p_j+1} x_{jt} = 1 \quad \forall j \in N, \tag{2.1}$$
$$\sum\nolimits_{j=1}^{n} \sum_{s=\max(0,t-p_j+1)}^{t} x_{js} \leqslant 1 \quad t = 1, \ldots, T, \tag{2.2}$$
$$x_{jt} \in \{0,1\} \quad \forall j \in N; \ t = 1, \ldots, T. \tag{2.3}$$

The first constraint set (2.1) enforces that each job can start only at exactly one particular time and the second constraint set (2.2) ensures that at any given time at most one job can be processed. Constraint set (2.3) states the integrality restriction. T assumes a value greater than the sum of processing times of all the jobs. For the problems with release dates, T assumes a value greater than the sum of processing time of all the jobs and the maximum value of the release date for all the jobs. Using the time index variables, the completion time of a job j can be written as

$$C_j = \sum_{t=1}^{T-p_j+1} (t-1+p_j) x_{jt} \quad \forall j \in N. \tag{2.4}$$

The objective function is

$$minimize \sum_{j=1}^{n} \sum_{t=1}^{T-p_j+1} \xi_{jt} x_{jt},$$

where

$$\xi_{jt} = w_j(t-1+p_j) \quad \forall j \in N, \ t = 1, \ldots, T, \tag{2.5}$$

if we are minimizing the total weighted completion time;

$$\xi_{jt} = \begin{cases} 1, & \text{if } t > (d_j - p_j + 1), \quad \forall j \in N, t = 1, \ldots, T, \\ 0, & \text{otherwise}, \end{cases} \tag{2.6}$$

if we are minimizing the number of tardy jobs; and

$$\xi_{jt} = w_j \max[0, t-1+p_j-d_j] \quad \forall j \in N, t = 1, \ldots, T, \tag{2.7}$$

if we are minimizing the total weighted tardiness. Note that for all these problems we do not need any additional variables or constraints.

The problem of minimizing the maximum lateness can be modeled by minimizing *LMAX* as the objective function and adding the constraint (1.6) by substituting C_j from (2.4).

To complete the formulation using time index variables for the problems with the release date constraints, we set $x_{jt} = 0$ for $t \leqslant r_j$, $\forall j \in N$.

Time index variables formulation was introduced by Sousa and Wolsey (1992) for non-preemptive single machine scheduling problems. van den Akker, van Hoesel, and Savelsbergh (1999) and Šorić (2000) later studied this formulation for different machine settings and objective functions.

2.3. Linear ordering variables [F3]

This formulation is based on binary linear ordering variables, δ_{jk}, which are equal to 1 when job j precedes job k and equal to 0,

Table 4
Results for single machine number of tardy jobs problem $1||\sum U_j$ for $P = U$ [1, 100]

Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
Completion time variables [F1]	20	0	1988581	7 [2.15]	0	11 [92.42%]	15
	40	0	701762	0	0	18 [91.23%]	0
	60	0	189275	0	0	18 [90.30%]	0
	100	0	24792	0	0	18 [93.82%]	7
Time index variables [F2]	20	0	2	18 [19.74]	0	0	18
	40	0	4	17 [414.8]	0	1 [38.03%]	18
	60	0	2	12 [995.2]	2	4 [38.43%]	16
	100	13	0	0	18	0	0
Linear ordering variables [F3]	20	0	46699	8 [0.39]	0	10 [72.45%]	17
	40	0	131	8 [21.3]	0	10 [93.49%]	8
	60	4	2	6 [362.6]	4	10 [98.32%]	6
	100	16	0	0	16	2 [10.00%]	0
Positional & assignment variables [F4]	20	0	620506	16 [165.13]	0	2 [100.00%]	18
	40	0	556003	8 [3]	1	9 [86.72%]	12
	60	0	308130	4 [62.6]	6	10 [79.83%]	8
	100	0	78154	2 [158]	7	9 [79.04%]	11

otherwise. The constraints of the MIP formulation with linear ordering variables are as follows:

$$\delta_{jk} + \delta_{kj} = 1 \quad 1 \leqslant j \leqslant k \leqslant n, \tag{3.1}$$

$$\delta_{jk} + \delta_{kl} + \delta_{lj} \leqslant 2 \quad j,k,l \in N \text{ and } j \neq k \neq l, \tag{3.2}$$

$$\delta_{jk} \in \{0,1\} \quad \forall j,k \in N. \tag{3.3}$$

Constraint set (3.1) is a set of conflict constraints, which ensure that either job j is processed before job k or job k is processed before job j. Constraint set (3.2) represents the transitivity constraints that ensure a linear order between three jobs. Constraint set (3.3) states the integrality restriction. Using linear ordering variables, the completion time of job j can be written as shown in constraint set (3.4) and holds only if all the release dates are equal to zero.

$$C_j = \sum_{\substack{k \in N \\ k \neq j}} p_k \delta_{kj} + p_j \quad \forall j \in N. \tag{3.4}$$

The objective function for minimizing the total weighted completion time can be written as $\sum_{\substack{j,k \in N \\ k \neq j}} w_j p_k \delta_{kj} + \sum_{j \in N} w_j p_j$. The MIP formulations for the other three objectives can be obtained by substituting C_j from (3.4) into the constraints (1.6), (1.7) and (1.9), and adding them to (3.1)–(3.3).

To formulate the problems with release date constraints using linear ordering variables, we arrange the jobs in non increasing order of the release dates and add the following constraints (3.5) and (3.6) (Nemhauser & Savelsbergh, 1992 can be referred for extra details).

$$S_j \geqslant r_i \delta_{ij} + \sum_{k<i,k\neq j} p_k(\delta_{ik} + \delta_{kj} - 1) + \sum_{k \geqslant i,k \neq j} p_k \delta_{kj} \quad \forall i,j \in N, \tag{3.5}$$

$$\delta_{jj} = 1, \quad \forall j \in N. \tag{3.6}$$

Further to formulate the various objective functions: $\sum w_j C_j, L_{\max}, \sum U_j$ and $\sum w_j T_j$ with release date constraint, the term

Table 5
Results for single machine total weighted tardiness problem $1||\sum w_j T_j$ for $P = U$ [1, 100]

Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
Completion time variables [F1]	20	0	1773223	6 [277.7]	0	12 [97.60%]	6
	40	0	646424	0	0	18 [92.77%]	0
	60	0	295818	0	9	9 [99.88%]	0
	100	0	47623	0	9	9 [100.00%]	3
Time index variables [F2]	20	0	2186	16 [584.2]	0	2 [4.89%]	16
	40	0	1118	7 [408]	0	11 [10.26%]	14
	60	1	30	5 [874.45]	5	8 [48.70%]	9
	100	16	0	0	18	0	0
Linear ordering variables [F3]	20	0	97658	10 [200]	0	8 [65.37%]	16
	40	0	645	7 [630.3]	11	0	7
	60	1	289	4 [1318]	12	2 [1.26%]	6
	100	13	32	0	17	1 [1.47%]	1
Positional & assignment variables [F4]	20	0	839441	15 [316.9]	0	3 [44.65%]	16
	40	0	525773	5 [195.2]	0	13 [83.42%]	6
	60	0	185670	2 [154.78]	0	16 [87.46%]	8
	100	0	9229	0	11	7 [100.00%]	6

Table 6
Results for $1||\sum w_jC_j$ and $1||\sum L_{\max}$ for P = U [1, 10]

Problem	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
						Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
$1\|\|\sum w_jC_j$ Does not matter as this is not a due date based objective	Time index variables [F2]	20	0	0	3 [0.21]	0	0	3
		40	0	0	3 [1.69]	0	0	3
		60	0	0	3 [7.67]	0	0	3
		100	0	0	3 [38.3]	0	0	3
	Positional & assignment variables [F4]	20	0	2484222	0	0	3 [88.48%]	0
		40	0	433311	0	0	3 [98.51%]	0
		60	0	119325	0	1	2 [98.84%]	0
		100	0	1150	0	3	0	0
$1\|\|\sum L_{\max}$ L = {0.5}, R = {0.4, 0.8, 1.4}	Time index variables [F2]	20	0	129961	7 [23.14]	0	2 [75.26%]	9
		40	0	5548	8 [698.48]	0	1 [2.5%]	9
		60	0	1919	3 [839.74]	5	1 [105.21%]	3
		100	0	107	1 [475.35]	6	2 [0.95%]	3
	Positional & assignment variables [F4]	20	0	895025	8[112.10]	0	1[2.70%]	9
		40	0	747097	4[5.96]	0	5[78.61%]	7
		60	0	163658	6[17.71]	0	3[110.20%]	6
		100	0	111947	4[332.215]	0	5[80.64%]	6

$\sum_{k\in N}^{k\neq j} p_k\delta_{kj}$ is replaced by S_j from the objective function and the inequalities.

The linear ordering formulation was first introduced by Dyer and Wolsey (1990). It was later studied by Blazewicz, Dror, and Weglarz (1991), Nemhauser and Savelsbergh (1992) and Chudak and Hochbaum (1999).

2.4. Assignment and positional date variables [F4]

In this formulation, we define binary assignment variables, u_{jk}, which are equal to 1 if job j is assigned to position k and are equal to 0 otherwise. Further, we introduce positional date variables, γ_k, which define the completion time of the job at position k. The constraints of the MIP formulation with assignment and positional date variables are as follows:

$$\sum_{k\in N} u_{jk} = 1 \quad \forall j \in N, \tag{4.1}$$
$$\sum_{j\in N} u_{jk} = 1 \quad \forall k \in N, \tag{4.2}$$
$$\gamma_1 \geqslant \sum_j p_j u_{j1} \tag{4.3}$$
$$\gamma_k \geqslant \gamma_{k-1} + \sum_j p_j u_{jk} \quad k = 2, \ldots, N, \tag{4.4}$$
$$\gamma_k \geqslant 0 \quad \forall k \in N, \tag{4.5}$$
$$u_{jk} \in \{0,1\} \quad \forall j,k \in N. \tag{4.6}$$

The constraint sets (4.1) and (4.2) ensure that a particular job is assigned exactly to one position and each position is assigned to exactly one job. Constraint sets (4.3) and (4.4) give the completion time of the job at position k. Constraint sets (4.5) and (4.6) are the non negativity and integrality constraints, respectively.

Table 7
Results for $1||\sum U_j$ and $1||\sum w_jT_j$ for P = U [1, 10]

Problem	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
						Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
$1\|\|\sum U_j$ L = {0.5}, R = {0.4, 0.8, 1.4}	Time index variables [F2]	20	0	0	9 [0.18]	0	0	9
		40	0	3	9 [2.05]	0	0	9
		60	0	10	9 [15.84]	0	0	9
		100	0	31	9 [119.1]	0	0	9
	Positional & assignment variables [F4]	20	0	693528	7[1.00]	0	2[38.29%]	9
		40	0	727446	2[22.43]	0	7[46.94%]	4
		60	0	145725	5[44.18]	0	4[17.05%]	5
		100	0	82411	1[92.81]	3	5[24.18%]	1
$1\|\|\sum w_jT_j$ L = {0.5}, R = {0.4, 0.8, 1.4}	Time index variables [F2]	20	0	65	9 [1.02]	0	0	9
		40	0	31848	9 [255.4]	0	0	9
		60	0	6568	9 [194.2]	0	0	9
		100	0	37130	2 [633.74]	0	7 [2.67%]	9
	Positional & assignment variables [F4]	20	0	1777692	4 [86.48]	0	5 [46.92%]	4
		40	0	799491	0	0	9 [85.70%]	0
		60	0	238139	0	0	9 [91.43%]	0
		100	0	8569	0	2	7 [98.65%]	0

In the MIP formulation of minimizing maximum lateness, we minimize *LMAX* as the objective function and add the following constraints to (4.1)–(4.6):

$$LMAX \geqslant \left(\gamma_k - \sum_j d_j u_{jk}\right) \quad \forall k \in N, \tag{4.7}$$

where $\sum_j d_j u_{jk}$ gives the due date of the job at position k. The problem of minimizing the number of tardy jobs is formulated by minimizing $\sum_{k=1}^{n} U_k$ as the objective function and adding the following constraint sets to (4.1)–(4.6):

$$\gamma_k \leqslant \sum_j d_j u_{jk} + MU_k \quad \forall k \in N, \tag{4.8}$$

$$U_k \in \{0,1\} \quad \forall k \in N. \tag{4.9}$$

To define the objectives of total weighted completion time and total weighted tardiness we need the completion time of the job j; therefore we need the following inequalities to find the completion time of job j. Constraint set (4.10) helps define the lower bounds on C_j

$$C_j \geqslant \gamma_k - M(1 - u_{jk}) \quad \forall j,k \in N, \tag{4.10}$$

$$C_j \geqslant 0 \quad \forall j \in N. \tag{4.11}$$

Hence the objective of minimizing the total weighted completion time can be formulated by minimizing $\sum w_j C_j$ as the objective function and adding the constraint sets (4.10) and (4.11) to (4.1)–(4.6). The tardiness constraint sets (1.9) and (1.10) are added along with the sets (4.10) and (4.11) to (4.1)–(4.6) to complete the formulation for the objective of minimizing the total weighted tardiness.

The formulation for the problems with release date constraints using assignment and positional date variables can be completed by adding the following inequality (4.12).

$$\gamma_k \geqslant \sum_j (p_j + r_j) u_{jk} \quad \forall k \in N. \tag{4.12}$$

The value of M is taken to be greater than the sum of the processing times of all the jobs. For the problems with release dates, the value of M is taken to be greater than the sum of processing time of all the jobs and the maximum value of the release date for all the jobs.

Lasserre and Queyranne (1992) introduced the formulation using assignment and positional date variables for the single machine scheduling problems. Dauzère-Pérès (1997) and Dauzère-Pérès and Sevaux (2003) later studied this problem for minimizing the number of tardy jobs. A summary of the prior research done in this area is presented in Table 1.

3. Computational comparison of the formulations

3.1. Data sets

We run our experiments for different set of parameters for all of these four formulations. We adopt the idea of parameter selection presented by Hariri and Potts (1983), Potts and Van Wassenhove (1983, 1982), Abdul-Razaq, Potts, and Van Wassenhove (1990).

For each job j, an integer processing time, p_j, is generated from uniform distribution [1, 100] and [1, 10] and an integer weight w_j is generated from the uniform distribution [1, 10]. The due date, d_j, of job j is an integer generated from the uniform distribution $[P(L - R/2), P(L + R/2)]$, where P is the sum of processing times of all jobs and the two parameters L and R are relative measures of the location and range of the distribution, respectively. The release date, r_j, of job j is an integer generated from the uniform distribution $[0, QP]$, where P is the sum of processing times of all the jobs and the parameter Q defines the range of the distribution. We choose L from the set $L \in \{0.5, 0.7\}$, R is chosen from the set $R \in \{0.4, 0.8, 1.4\}$, and Q is taken as 0.4, which makes the release date distribution range to be $[0, 0.4P]$. We run 3 problem instances for each of the six different combinations of L and R, generating a total of 18 runs for each of the four different formulations for the problems. Also we restrict our computation to $L \in \{0.5\}$ for the problems with release dates, thus reducing from 18 to 9 runs for different formulations. The number of jobs, n, is chosen from the set $n \in \{20, 40, 60, 100\}$.

3.2. Results

The MIP formulations are modeled using AMPL and CPLEX 8.1 with default settings is used to solve the generated problem instances. The experiments are run on a Linux distributed machine with a 2.4 GHz processor and 1GB memory. The runs are terminated after an hour of CPU time.

To compare the different formulations arising from the choice of different variables, we look at the objective function value of the LP relaxation, the number of nodes explored, the computational time for the problem instances for which the optimal solution is obtained within 1 h, the number of tests cases unsolved in 1 h, and the optimality gap for the instances that could not be solved within 1 h. The formulations are also compared based on the number of test instances in which either the optimal or the best feasible solution (in case the optimal is not found by any formulation) is obtained by each formulation.

The results obtained for total weighted completion time, maximum lateness, number of tardy jobs and total weighted tardiness objectives when there are no release dates are presented in Tables 2–5, respectively.

For the $\sum w_j C_j$ objective (see Table 2) without release dates we test only three instances because the parameters L and R are irrelevant as this objective is not due date based. For this objective, F3 (formulation with linear ordering variables) performed the best. It turns out that the LP relaxation of this formulation reduces to the Weighted Shortest Processing Time (*WSPT*) rule, which gives the optimal solution for this problem. Hence even the test cases with 100 jobs were solved within 23 s. Similarly, F2 (formulation with time index variables) was also able to provide the optimal solution at the root node, but the computation time increased exponentially as the number jobs increased and the test cases with 100 jobs were not solved within 1 h of computational time. F1 (formulation with completion time variables) and F4 (formulation with assignment and positional date variables) did not perform well for this problem. These formulations were able to solve the LP relaxation much faster but the bounds obtained were not tight enough to converge to the optimal solution within 1 h of computational time.

The results obtained for the other three objectives $(\sum L_{\max}, \sum U_j$ and $\sum w_j T_j)$ without the release dates were similar. The findings from the computations reported in Tables 3–5 are summarized below:

- All formulations have difficulty as the number of jobs increases (increase in the computational time).
- F1 and F2 do not produce optimal solutions as frequently as F3 and F4 when the number of jobs is increased. It is obvious that F2 does not perform well because as the number of jobs increases, we are not even able to solve the LP relaxation. F1 solves the LP relaxation faster, but that does not help us very much as the bound is not tight.

- F4 often finds a feasible solution, but the optimality gap is higher than F3 because the lower bound obtained by the LP relaxation of F4 is not as tight as the one for F3. Further, the number of test instances unsolved with F3 is much higher than with F4 when the number of jobs increases. This is because it gets harder to solve the LP relaxation of F3 as the number of jobs increases.
- Overall it is harder to solve the LP relaxations of F2 and F3 with an increased number of jobs, though the bounds are tighter. With F1 and F4, it is easier to solve the LP relaxations, but the bounds are not very tight. Comparatively, the number of test cases solved that have either the optimal solution or the best feasible solution is much higher for F4 but usually with a large optimality gap.
- F4 might be the choice of formulation for an expert in integer programming because the LP relaxation of this formulation can be solved faster and a larger number of nodes can be explored in a fixed amount of time. This creates a potential to use the recent advancements in integer programming literature (e.g. branch and cut).

Based on our personal communication with Queyranne (2004), we decided to make additional experiments, since the performance of F2 is known to be highly influenced by the sum of the processing times. An initial set of computational experiments was done on the total weighted tardiness problem. The results of these experiments are presented in Khowala, Keha, and Fowler (2005). In these experiments the processing times are created from the discrete uniform distribution [1, 10]. As expected, F2 was found to be much more efficient with a lower range of processing times. This is because the number of the variables of F2 is a function of the sum of the processing times of the jobs and the LP relaxation is much easier to solve when the sum of the processing times of the jobs is small. But as the number of jobs increases, F2 was not able to give even a

Table 8
Results for single machine total weighted completion time with release dates problem $1|r_j|\sum w_jC_j$ for P = U [1, 100]

Location & range parameters for due date	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulation
						Test cases with no solution	Test cases with some integer solution [Avg. optimality gap]	
Q = {0.4}, L = {0.5}, R = {0.4, 0.8, 1.4}	Start time & completion time variables [F1]	20	0	2538467	0	0	9 [22.5%]	0
		40	0	648450	0	0	9 [41.25%]	0
		60	0	275931	0	0	9 [42.71%]	0
		100	0	85736	0	0	9 [50.50%]	9
	Time index variables [F2]	20	0	455	9 [125.91]	0	0	9
		40	0	1305	1 [2421.05]	0	8 [1.78%]	5
		60	0	72	0	0	9[5.80%]	9
		100	9	0	0	9	0	0
	Linear ordering variables [F3]	20	0	696	9 [21.68]	0	0	9
		40	0	1335	0	0	9 [3.88%]	4
		60	0	27	0	9	0	0
		100	3	1	0	9	0	0
	Positional & assignment variables [F4]	20	0	1912557	0	0	9 [88.57%]	0
		40	0	273709	0	0	9 [99.62%]	0
		60	0	14563	0	9	0	0
		100	0	356	0	9	0	0

Table 9
Results for single machine maximum lateness with release dates problem $1|r_j|\sum L_{\max}$ for P = U [1, 100]

Location & range parameters for due date	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulation
						Test cases with no solution	Test cases with some integer solution [Avg. optimality gap]	
Q = {0.4}, L = {0.5}, R = {0.4, 0.8, 1.4}	Start time & completion time variables [F1]	20	0	1739733	5 [576.98]	0	4 [74.52%]	6
		40	0	479734	4 [644.59]	0	5 [62.84%]	4
		60	0	213914	1 [42.95]	0	8 [41.48%]	2
		100	0	50020	1 [160.41]	0	8 [46.59%]	5
	Time index variables [F2]	20	0	1739	3 [129.67]	2	4 [47.61%]	3
		40	2	3	4 [594.52]	5	0	4
		60	4	1	2 [2797.05]	7	0	2
		100	9	0	0	9	0	0
	Linear ordering variables [F3]	20	0	43131	5 [325.57]	0	4 [35.98%]	9
		40	0	40	1 [3180.66]	7	1 [5.30%]	1
		60	2	1	0	9	0	0
		100	8	0	0	9	0	0
	Positional & assignment variables [F4]	20	0	937270	7 [10.74]	0	2 [58.93%]	9
		40	0	363343	3 [6.03]	0	6 [74.58%]	7
		60	0	281885	1 [59.46]	0	8 [56.21%]	6
		100	0	62686	3 [1118.77]	2	4 [89.38%]	4

Table 10
Results for single machine number of tardy jobs with release dates problem $1|r_j|\sum U_j$ for P = U [1, 100]

Location & range parameters for due date	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulation
						Test cases with no solution	Test cases with some integer solution [Avg. optimality gap]	
Q = {0.4}, L = {0.5}, R = {0.4, 0.8, 1.4}	Start time & completion time variables [F1]	20	0	787690	7 [85.17]	0	2 [70.31%]	8
		40	0	680070	1 [6.78]	0	8 [42.61%]	1
		60	0	198345	0	0	9 [51.51%]	0
		100	0	30535	0	0	9 [64.70%]	9
	Time index variables [F2]	20	0	7	9 [16.45]	0	0	9
		40	0	7	9 [331.51]	0	0	9
		60	0	4	6 [995.10]	0	3 [14.85%]	9
		100	4	0	0	7	2 [79.82%]	0
	Linear ordering variables [F3]	20	0	14431	6 [16.74]	0	3 [52.99%]	8
		40	0	37	2 [19.43]	0	7 [78.21%]	2
		60	1	0	0	9	0	0
		100	9	0	0	9	0	0
	Positional & assignment variables [F4]	20	0	2002975	2 [0.87]	0	7 [38.97%]	8
		40	0	491760	0	2	7 [59.79%]	2
		60	0	245978	0	6	3 [58.80%]	0
		100	0	38335	0	6	3 [66.89%]	0

feasible solution. It was also observed that F1 and F4 also perform better with the lower range of processing times because the value of big M in F1 and F4 depends on the sum of the processing times of the jobs. However improvements for F4 were more than those for F1. The performance of F3 was not affected by changing the range of the processing times because the constraints of F3 do not contain processing times.

We tested F2 and F4 to see the effect of reducing the processing time range. We only tested the instances with L = 0.5 to reduce the number of experiments. These results are presented in Tables 6 and 7. F2 is more efficient with lower ranges of processing times, but as we increase the number of jobs (therefore the sum of the processing times), the performance of F2 is reduced.

Further we conducted a similar series of experiments for these four problems with release dates. We limit our experimentations to 9 runs for each formulation and each different number of jobs by selecting L = 0.5. As mentioned earlier, the value of Q is selected as 0.4 to determine the range for the release dates. These results are presented in Tables 8–11. The findings from the computational results can be summarized below for the problems with release dates:

- F3 can no longer solve the LP relaxation for the $\sum w_jC_j$ problem when there are release dates.
- F1, F3 and F4 do not produce an optimal solution as frequently as the F2 formulation for most of the problems (except for the $\sum L_{\max}$, which we feel could also behave the same if we allowed the computation to run longer than 1 h). F2 is either able to provide an optimal solution with an increasing larger number of jobs (up to 60 jobs) or is within an optimality gap of 30%.
- F1 and F4 often find a feasible solution, but the optimality gap is higher than F2, because the lower bound obtained by the LP relaxation of F1 and F4 are not as tight as the one obtained from F2.
- F3 also provides much tighter lower bounds compared to F1 and F4, but the LP relaxation is much harder to solve. For most of the test instances F3 terminated without any solution.
- The comparison of the quality of solution (comparing the optimal or the best feasible solution) obtained from the four formulations also indicates that F2 performs the best and also the performance of F2 is better with increasing number of jobs (except as noted above for $\sum L_{\max}$). The

Table 11
Results for single machine total weighted tardiness with release dates problem $1|r_j|\sum w_jT_j$ for P = U [1, 100]

Location & range parameters for due date	Formulation	Number of jobs	Number of runs where LP relaxation cannot be solved in 1 h	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulation
						Test cases with no solution	Test cases with some integer solution [Avg. optimality gap]	
Q = {0.4}, L = {0.5}, R = {0.4, 0.8, 1.4}	Start time & completion time variables [F1]	20	0	2464169	2 [1694.83]	0	7 [61.86%]	2
		40	0	836104	0	0	9 [68.39%]	0
		60	0	297155	0	0	9 [71.82%]	0
		100	0	72422	0	0	9 [77.62%]	9
	Time index variables [F2]	20	0	5285	7 [74.35]	0	2 [4.68%]	8
		40	0	1890	3 [1434.45]	0	6 [8.47%]	9
		60	0	296	0	0	9 [28.51%]	9
		100	9	0	0	9	0	0
	Linear ordering variables [F3]	20	0	41777	6 [444.13]	0	3 [12.38%]	9
		40	0	110	0	9	0	0
		60	5	2	0	9	0	0
		100	9	0	0	9	0	0
	Positional & assignment variables [F4]	20	0	2366900	1 [21.93]	0	8 [59.95%]	1
		40	0	406505	0	0	9 [96.17%]	0
		60	0	122699	0	1	8 [96.92%]	0
		100	0	936	0	9	0	0

increase in the efficiency of F2 after adding the release date constraints can be attributed to the reduction in the time index variables for these problems. The release data constraints forces the time index variables to be zero for the values of $t < r_j$.

- F2 should be the choice with the release date constraints for the test cases we investigated.

4. Improved assignment and positional date formulation

The computational results presented in the subsequent sections and also in Khowala et al. (2005), suggested that the bounds obtained from the formulations using time index variables and linear ordering variables were tighter than those from the other formulations but the LP relaxations were harder to solve, hence the branch and bound algorithm would not be able to explore a large number of nodes given a fixed computational time budget. On the other hand, the LP relaxation of the formulation with assignment and positional variables was easy to solve but the bounds were not tight enough to yield a better feasible solution within a given computation time. Hence we further studied this formulation and came up with two families of valid inequalities that help in improving the lower bounds obtained from this formulation. The first family of valid inequalities will also help us to remove the big-M constraints given by (4.10) and the second set of inequalities gives a better lower bound on completion time variables. The inequalities assume that the jobs are arranged in non decreasing order of the processing times, i.e. $0 \leqslant p_1 \leqslant p_2 \leqslant \ldots \leqslant p_n$.

For each $j, k \in N$, we define π_{jk} and ρ_{jk} as

$$\pi_{jk} = \begin{cases} \sum_{l=1}^{k-1} p_l, & \text{if } k \leqslant j, \\ \sum_{l=1}^{j-1} p_l + \sum\limits_{l=j+1}^{k} p_l, & \text{if } k > j. \end{cases}$$

$$\rho_{jk} = \begin{cases} \sum_{l=k+1}^{n} p_l, & \text{if } k \geqslant j, \\ \sum_{l=k}^{j-1} p_l + \sum\limits_{l=j+1}^{n} p_l, & \text{if } k < j. \end{cases}$$

Note that π_{jk} gives the minimum value the completion time of the job at position $k-1$ can take given that job j is at position k and ρ_{jk} gives the maximum value the sum of the processing times of the jobs that are positioned after the k^{th} job can take given that the job j is at position k.

Proposition 1. *The inequality*

$$C_j \geqslant \gamma_k - \sum_{l=1}^{k-1} a_{jl}u_{jl} + \sum_{l=k+1}^{n} a_{jl}u_{jl} \quad \forall j,k \in N, \tag{4.13}$$

where

$$a_{jl} = \begin{cases} \rho_{j,n-k+l}, & \text{if } l < k, \\ p_j + \pi_{j,l-k}, & \text{if } l > k, \end{cases}$$

is valid.

Proof. If $u_{jk} = 1$, then the inequality reduces to $C_j \geqslant \gamma_k$ which is valid. If $u_{jl} = 1$ for $l < k$ then the inequality reduces to $C_j \geqslant \gamma_k - \rho_{j,n-k+l}$. The inequality is valid for this case because $\gamma_l \geqslant \gamma_k - \rho_{j,n-k+l}$ from the definition of ρ's. If $u_{jl} = 1$ for $l > k$ then the inequality reduces to $C_j \geqslant \gamma_k + \pi_{j,l-k} + p_j$. The inequality is valid for this case because $\gamma_l \geqslant \gamma_k + \pi_{j,l-k} + p_j$ from the definition of π's. □

Also note that when $u_{jk} = 1$ the inequality (4.13) forces $C_j \geqslant \gamma_k$, therefore the inequalities given by (4.10) can be replaced by (4.13).

Proposition 2. *For a job $j \in N$, the inequality*

$$C_j \geqslant p_j + \sum_{k=2}^{n} \pi_{jk}u_{jk} \tag{4.14}$$

is valid.

Proof. If the job j is at position $k > 1$ then $u_{jk} = 1$ and (4.14) becomes $C_j \geqslant p_j + \pi_{jk}$ and is valid from the definition of $\pi'_{jk}s$. □

4.1. Computational performance of the improved formulation

Our findings so far, suggest F4 worked consistently well across most of the problems without release dates. F4 could be improved significantly for some problems by adding the new set of inequalities described in the previous section. We conducted an additional set of computational experiments by replacing the set of constraints in Eq. (4.10) by these two new set of inequalities (4.13) and (4.14) for the $1||\sum w_jC_j$ and $1||\sum w_jT_j$ and problems. The

Table 12
Results for $1||\sum w_jC_j$ and $1||\sum w_jT_j$ for P = U [1, 10] with improved inequalities

Problem	Formulation	Number of jobs	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
$1\|\|\sum w_jC_j$ Does not matter as this is not a due date based objective	Positional & assignment variables [F4]	20	2484222	0	0	3 [88.48%]	0
		40	433310	0	0	3 [98.51%]	3
		60	119324	0	1	2 [98.83%]	2
		100	1150	0	3	0	0
	Positional & assignment variables (improved formulation)	20	539638	0	0	3 [6.15%]	3
		40	40257	0	2	1 [27.25%]	0
		60	6373	0	3	0	0
		100	9	0	3	0	0
$1\|\|\sum w_jT_j$ L = {0.5}, R = {0.4, 0.8, 1.4}	Positional & assignment variables [F4]	20	1777692	4 [86.48]	0	5 [46.92%]	4
		40	799491	0	0	9 [85.70%]	5
		60	238139	0	0	9 [91.43%]	4
		100	8569	0	2	7 [98.65%]	7
	Positional & assignment variables (improved formulation)	20	436811	6 [439.7]	0	3 [35.65%]	8
		40	98731	0	0	9 [60.58%]	5
		60	25570	0	1	8 [69.29%]	5
		100	651	0	9	0	0

Table 13
Results for $1||\sum w_jC_j$ and $1||\sum w_jT_j$ for P = U [1, 100] with improved inequalities

Problem	Formulation	Number of jobs	Average number of nodes	Number of test cases solved for optimal solution [Avg. computation time, seconds]	Number of test cases unsolved in 1 h		Number of time optimal or best feasible solution obtained compared to other formulations
					Test cases with no integer solution	Test cases with some integer solution [Avg. optimality gap]	
$1\|\|\sum w_jC_j$ Does not matter as this is not a due date based objective	Positional & assignment variables [F4]	20	2813399	0	0	3 [87.73%]	1
		40	475664	0	0	3 [98.20%]	1
		60	164150	0	0	3 [99.25%]	0
		100	1108	0	3	0	0
	Positional & assignment variables (improved formulation)	20	528710	0	0	3 [9.11%]	2
		40	49645	0	0	3 [22.18%]	2
		60	7734	0	0	3 [22.87%]	3
		100	43	0	3	0	0
$1\|\|\sum w_jT_j$ L = {0.5, 0.7}, R = {0.4, 0.8, 1.4}	Positional & assignment variables [F4]	20	839441	15 [316.9]	0	3 [44.65%]	16
		40	525773	5 [195.2]	0	13 [83.42%]	7
		60	185670	2 [154.78]	0	16 [87.46%]	8
		100	9229	0	11	7 [100.00%]	7
	Positional & assignment variables (improved formulation)	20	778305	18 [216.25]	0	0	18
		40	105692	6 [473.3]	0	12 [64.34%]	14
		60	26871	2 [353.68]	3	13 [80.13%]	11
		100	732	0	15	3 [100.00%]	1

results are presented in Table 12 for the cases where the processing times are from the discrete uniform distribution [1, 10], L = {0.5} and R = {0.4, 0.8, 1.4} and in Table 13 for the cases where the processing times are from the discrete uniform distribution [1, 100], L = {0.5, 0.7} and R = {0.4, 0.8, 1.4}.

The results obtained for these two objectives had a similar pattern, both for the original formulation as well as for the improved formulation. The findings from the computational experiments reported in Tables 12 and 13 are summarized below:

- The original formulation using the assignment and positional variables for both of these problem solves the LP relaxation faster, but that does not help us very much as the bound is not tight and the test instances end up with the optimality gaps between 50% and 100% in 1 h of computational time. For most of the test cases the lower bounds generated by LP relaxations are equal to zero. Also, the original formulation often finds a feasible solution but the optimality gap is higher.
- After adding these two new classes of inequalities, the bounds obtained were much tighter (for almost all of the test instances the objective value of LP relaxation with improved formulation was better), which helped in obtaining the optimal solutions for larger number of test instances as well as reducing the optimality gap to 5–20% range for $\sum w_jC_j$ problem and to 35–85% range for $\sum w_jT_j$ problem.
- Table 14 shows the average percentage difference between the objective values of the LP relaxation obtained for various numbers of jobs for $\sum w_jC_j$ problem compared to the optimal solution. These averages for each particular number of jobs are for three instances. Note that the objective values of the LP relaxations from the original formulation were zero for all the instances for $\sum w_jC_j$.

Table 14
Results for $1||\sum w_jC_j$ with $P\sim$ U[1, 10] and $P\sim$ U[1, 100]

Number of jobs	Avg. optimality gap of the LP relaxation for $1\|\|\sum w_jC_j$	
	$P\sim$ U[1, 10] (%)	$P\sim$ U[1, 100] (%)
20	11.4	31.6
40	15.3	23.3
60	15.1	12.9
100	17.5	20.9

- It takes longer to solve the LP relaxation of the improved formulation, but it provides a better bound. For most of the test instances, the number of nodes explored is less with the improved formulation. Since it takes longer to solve the LP relaxation of the improved formulation, the number of test cases with no integer feasible solution is more for larger number of jobs. The integer feasible solution could be achieved by providing more computation time to the problem instances.

For some instances where a feasible solution can not be found easily, *primal heuristics* could help us to find one easily. The fractional optimal solution found at a node can be modified to satisfy the integrality conditions. Suppose that at a node the solution (u^*, γ^*, C^*) has at least one (j,k) pair such that u_{jk} that is fractional. We can sort the job indices in non-decreasing order of the completion time variables C^*. These job indices will give us a feasible schedule and an upper bound to the problem. Our preliminary experiments showed that this primal heuristic gives solutions that are either optimal or very close to the optimal after a few number of nodes are explored. These results are not given here as the main focus of this paper is to compare the formulations with default settings of a commercial solver.

5. Conclusions and future work

In this paper, we have compared the computational performance of four different formulations on single machine scheduling problems with varying complexity. The performances of these formulations very much depend on the objective function, number of jobs and the sum of the processing times of all the jobs. F2 and F3 appear to be the most widely used formulations in the Integer Programming and Scheduling literature and F4 appears to be the least widely used. F1 often appears in textbooks and other literature that simply formulate/describe the problem (not the solution methodology) and it clearly does not generally perform well in practice.

With F1 and F4, the LP relaxation is easy to solve and provides a feasible solution easily. F2 and F3 have been preferred due to the fact that they generally produce tighter bounds. However, we have found that the LP relaxation of these formulations tends to be much more difficult to solve. This is particularly true for F2 when P (sum of processing time of all the jobs) is large. Therefore fewer nodes of the branch and bound tree can be explored for a fixed computational budget. This limits one's ability to explore recent advancement in IP methodology (such as branch and cut). On the other hand, the LP relaxation of F4 can be solved relatively quickly, so this MIP formulation offers more promise for these new advanced techniques. In Section 4 we gave two simple families of inequalities that improved the bounds obtained from the LP- relaxation. A more detailed polyhedral study would make this formulation work better. We note that when P is small or with release date constraints, F2 becomes the preferred formulation.

There was noticeable improvement in terms of achieving a better bound (LP relaxation) and reduction in the optimality gap by adding the new set of inequalities to F4 and removing the big-M constraint. Further if we are able to trade off the solution quality (in terms of reducing the optimality gap and obtaining better integer feasible solution) versus the computational time, this new formulation will be preferred, as we can notice the improvement in the solution quality.

We are aware that for most of the problems studied in this paper problem specific algorithms have been proposed and they are shown to be more effective than solving MIP formulations. But it should be noted that these are problem specific algorithms and require expertise in coding and scheduling. Also these algorithms are hard to modify for other problems in the same domain. The MIP formulations studied in this paper on the other hand could be easily solved using commercial solvers and don't require expertise in scheduling or coding.

This paper is, as of our knowledge, the first paper that compares the computational performances of these four MIP formulations in the scheduling literature. As future research, the MIP formulations can be compared with other additional restrictions, such as precedence constraints, or for more complex machine environments. F4 might be the choice of formulation for an expert in integer programming because the LP relaxation of this formulation can be solved faster and a larger number of nodes can be explored in a fixed amount of computational time. This creates a potential to use recent advancements found in the integer programming literature. Studying the polyhedral structure of this formulation and using the valid inequalities at a branch-and-cut algorithm is the subject of a forthcoming paper.

References

Abdul-Razaq, T. S., Potts, C. N., & Van Wassenhove, L. N. (1990). A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Applied Mathematics, 26*, 235–253.

Balas, E. (1985). On the facial structure of scheduling polyhedra. *Mathematical Programming, 24*, 179–218.

Blazewicz, J., Dror, M., & Weglarz, J. (1991). Mathematical programming formulations for machine scheduling: A survey. *European Journal of Operational Research, 51*, 283–300.

Chudak, F. A., & Hochbaum, D. S. (1999). A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. *Operations Research Letters, 25*, 199–204.

Dauzère-Pérès, S. (1997). An efficient formulation for minimizing the number of late jobs in single-machine scheduling. In *IEEE Symposium on Emerging Technologies & Factory Automation (ETFA)* (pp. 442–445).

Dauzère-Pérès, S., & Sevaux, M. (2003). Using Lagrangean relaxation to minimize the weighted number of late jobs on a single machine. *Naval Research Logistics, 50*(3), 273–288.

Dyer, M. E., & Wolsey, L. A. (1990). Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discrete Applied Mathematics, 26*, 255–270.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics, 5*, 287–326.

Hariri, A. M. A., & Potts, C. N. (1983). An algorithm for single machine sequencing with release dates to minimize total weighted completion time. *Discrete Applied Mathematics, 5*, 99–109.

Jackson, J. R. (1955). Scheduling a production line to minimize maximum tardiness. *Management Science Research Project*, (Research Report 43). Los Angles: University of California.

Khowala, K., Keha, A. B., & Fowler, J. (2005). A comparison of different formulations for the non-preemptive single machine total weighted tardiness scheduling problem. *The Second Multidisciplinary International Conference on Scheduling: Theory & Application (MISTA)*.

Kise, H., Ibaraki, T., & Mine, H. (1978). A solvable case of the one-machine scheduling problem with ready and due times. *Operations Research, 26*, 121–126.

Lasserre, J.B., & Queyranne, M. (1992). Generic scheduling polyhedral and a new mixed-integer formulation for single-machine scheduling. In *Proceedings of the Second IPCO Conference* (pp. 136–149). Pittsburgh: Carnegie-Mellon University.

Lawler, E. L. (1977). A 'pseudopolynomial' time algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics, 1*, 331–342.

Lenstra, J. K., Rinnooy Kan, A. H. G., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete mathematics, 1*, 343–362.

Moore, J. M. (1968). An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science, 15*, 102–109.

Nemhauser, G. L., & Savelsbergh, M. W. P. (1992). A cutting plane algorithm for the single machine scheduling problem with release times. Combinatorial optimization: New frontiers in the theory and practice. In M. Akgül, H. Hamacher, & S. Tufecki (Eds.). *NATO ASI series F: Computer and systems sciences* (Vol. 82, pp. 63–84). Berlin: Springer.

Potts, C. N., & Van Wassenhove, L. N. (1983). An algorithm for single machine sequencing with deadlines to minimize total weighted completion time. *European Journal of Operational Research, 12*, 379–389.

Potts, C. N., & Van Wassenhove, L. N. (1982). A decomposition algorithm for the single machine, total tardiness problem. *Operations Research Letters, 1*, 177–181.

Queyranne, M., & Wang, Y. (1991). Single-machine scheduling polyhedra with precedence constraints. *Mathematics of Operations Research, 16*, 1–20.

Queyranne, M. (1993). Structure of a simple scheduling polyhedron. *Mathematical Programming, 58*, 263–285.

Queyranne, M. (2004). *Personal communication at INFORMS annual conference*. Denver.

Queyranne, M., & Schulz, A. S. (1994). *Polyhedral approaches to machine scheduling*. Technical Report 408/1994. Berlin, Germany: Technical University of Berlin, Department of Mathematics.

Smith, W. E. (1956). Various optimizers for single stage production. *Naval Research Logistics Quarterly, 3*, 59–66.

Šorić, K. (2000). A cutting plane algorithm for a single machine scheduling problem. *European Journal of Operational Research, 127*, 383–393.

Sousa, J. P., & Wolsey, L. A. (1992). A time-indexed formulation of non-preemptive single-machine scheduling problems. *Mathematical Programming, 54*, 353–367.

van den Akker, J. M., van Hoesel, C. P. M., & Savelsbergh, M. W. P. (1999). A polyhedral approach to single machine scheduling problems. *Mathematical Programming, 85*, 541–572.