

Makespan Minimization for a Two-Machine Scheduling Problem with a Single Server

Keramat Hasani^{*,**} Svetlana A. Kravchenko^{**}
Frank Werner^{***}

^{*} Islamic Azad University, Malayer Branch, Malayer, Iran, (e-mail: hasani_keramat@yahoo.com)

^{**} United Institute of Informatics Problems, Surganova St. 6, 220012 Minsk, Belarus, (e-mail: kravch@newman.bas-net.by)

^{***} Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, Postfach 4120, 39016 Magdeburg, Germany, (e-mail: frank.werner@ovgu.de)

Abstract: We consider the following problem. A set of non-preemptable jobs has to be scheduled on two identical parallel machines such that the makespan is minimized. Before processing, each job must be loaded on a machine, which takes a given setup time. All these setups have to be done by a single server which can handle at most one job at a time. For this problem, we propose three mixed integer linear programming formulations. We compare our results with known heuristics.

Keywords: Scheduling algorithms, parallel machines, single server.

1. INTRODUCTION

The problem considered can be described as follows. There are n independent jobs and two identical parallel machines. For each job J_j , $j = 1, \dots, n$, its processing time p_j is given. Before processing, a job must be loaded on the machine M_q , $q = 1, 2$, where it is processed which requires a setup time s_j . During such a setup, the machine M_q is also involved into this process for s_j time units, i.e., no other job can be processed on this machine during this setup. All setups have to be done by a single server which can handle at most one job at a time. The goal is to determine a feasible schedule which minimizes the makespan. So, using the common notation, we consider the problem $P2, S1 \parallel C_{max}$. This problem is strongly NP-hard since the problem $P2, S1 \mid s_j = s \mid C_{max}$ is strongly NP-hard, see Hall et al. [2000]. Note also that the problem $P2 \parallel C_{max}$ is NP-hard in the ordinary sense.

The problem $P2, S1 \parallel C_{max}$ was considered in Gan et al. [2012], where some exact and heuristic solutions were derived and tested. We propose three mixed integer linear programming formulations for the problem $P2, S1 \parallel C_{max}$, and we compare the performance of these models with the heuristics proposed in Gan et al. [2012]. Additional information on server scheduling can be found in Brucker et al. [2002] and Werner and Kravchenko [2010].

2. SETUP SEQUENCE MODEL

In the following model, the loading order of the jobs is used as in Gan et al. [2012].

Let

$$x_{ij} = \begin{cases} 1, & \text{if } J_j \text{ is the } i\text{th job to be setup,} \\ 0, & \text{otherwise.} \end{cases}$$

Then, for any feasible schedule, the equalities

$$\sum_{j=1}^n x_{i,j} = 1 \quad (1)$$

and

$$\sum_{i=1}^n x_{i,j} = 1 \quad (2)$$

must hold.

Let ss_i be the loading time of the i th loading job and pp_i be the processing time of the i th loading job, i.e., the equalities

$$ss_i = \sum_{j=1}^n s_j x_{i,j} \quad (3)$$

and

$$pp_i = \sum_{j=1}^n p_j x_{i,j}. \quad (4)$$

hold. Now for the first and the second loading jobs, we can introduce the inequality

$$F_{1,2} \geq ss_1 + ss_2, \quad (5)$$

i.e., the part which forms the makespan. If the processing part of the first loading job is large enough, then one can introduce the inequality

$$L_{1,2} \geq pp_1 - ss_2, \quad (6)$$

and to denote the time interval when only one machine is busy, one can introduce L_2 with the inequalities

$$L_2 \geq L_{1,2} - pp_2, \quad (7)$$

and

$$L_2 \geq pp_2 - L_{1,2}. \quad (8)$$

Let

$$x_j = \begin{cases} 1, & \text{if } J_j \text{ is finished last among the jobs } 1, \dots, j, \\ 0, & \text{otherwise.} \end{cases}$$

Now, to estimate the overlapping part for the first two jobs, we introduce the inequalities

$$OF_2 \geq L_{1,2} - M(1 - x_2), \quad (9)$$

and

$$OF_2 \geq pp_2 - Mx_2, \quad (10)$$

where $M = \max_j \{p_j\}$. To know the earliest time when one of the machines is available, we introduce the inequality

$$F_2 \geq F_{1,2} + OF_2. \quad (11)$$

In an analogous way, for $j = 2, \dots, n - 1$, we have the following inequalities:

$$F_{j,j+1} \geq F_j + ss_{j+1}, \quad (12)$$

$$L_{j,j+1} \geq L_j - ss_{j+1}, \quad (13)$$

$$L_{j+1} \geq L_{j,j+1} - pp_{j+1}, \quad (14)$$

$$L_{j+1} \geq pp_{j+1} - L_{j,j+1}, \quad (15)$$

$$OF_{j+1} \geq L_{j,j+1} - M(1 - x_{j+1}), \quad (16)$$

$$OF_{j+1} \geq pp_{j+1} - Mx_{j+1}, \quad (17)$$

$$F_{j+1} \geq F_{j,j+1} + OF_{j+1}. \quad (18)$$

Now, to minimize the makespan, one has to minimize

$$F_n + L_n. \quad (19)$$

Then one can show that the following theorem holds.

Theorem 1. Any schedule s can be described as a feasible solution of system (1) - (18). The equality

$$C_{max}(s) = F_n + L_n$$

holds.

Now, to prove the equivalence between the scheduling problem $P2, S1 || C_{max}$ and the model (1) - (19), one has to prove the following theorem.

Theorem 2. Any feasible solution of system (1) - (18) can be described as a feasible schedule for the problem $P2, S1 || C_{max}$. The equality

$$F_n + L_n = C_{max}(s)$$

holds.

Thus, we will consider the model M0:

Minimize (19) subject to the constraints (1) - (18).

3. BLOCK MODELS

It is easy to see that any schedule for the problem $P2, S1 || C_{max}$ can be considered as a unit of blocks B_1, \dots, B_z , where $z \leq n$. Each block B_k can be completely defined

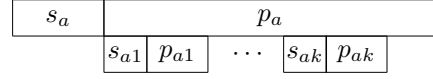


Fig. 1. One block, where J_a is the first level job and $\{J_{a1}, \dots, J_{ak}\}$ are the second level jobs

by the first level job J_a and a set of second level jobs $\{J_{a1}, \dots, J_{ak}\}$, where inequality

$$p_a \geq s_{a1} + \dots + s_{ak} + p_{a1} + \dots + p_{ak}$$

holds, see Fig. 1. Thus, the model that we suggest is based on the fact that any schedule can be decomposed into a set of blocks.

The variable $B_{k,f,j}$ is used for a block. We have $B_{k,f,j} = 1$ if job J_j is scheduled in level f in the k -th block, otherwise $B_{k,f,j} = 0$. The index $k = 1, \dots, n$ indicates the serial number of the block. The index $f \in \{1, 2\}$ indicates the level, i.e., we have $f = 1$ if the level is the first one, and $f = 2$ if the level is the second one. The index $j = 1, \dots, n$ indicates the job.

Each job belongs to some block, i.e., for any $j = 1, \dots, n$, the equality

$$\sum_{k=1}^n \sum_{y=1}^2 B_{k,y,j} = 1 \quad (20)$$

holds. There is only one job of the first level for each block, i.e., for each $y = 1$ and for any $k = 1, \dots, n$, the inequality

$$\sum_{j=1}^n B_{k,1,j} \leq 1 \quad (21)$$

holds.

Since all blocks are given, we define the following data for each block B_k , where $k = 1, \dots, n$:

- The loading part of the block B_k has the length $ST_k \geq 0$, formally inequality

$$ST_k \geq \sum_{j=1}^n s_j B_{k,1,j} \quad (22)$$

holds.

- The objective part of the block B_k has the length

$$\sum_{j=1}^n (s_j + p_j) B_{k,2,j}.$$

- The processing part of the block B_k has the length $PT_k \geq 0$, formally the inequality

$$PT_k \geq \sum_{j=1}^n p_j B_{k,1,j} - \sum_{j=1}^n (s_j + p_j) B_{k,2,j} \quad (23)$$

holds.

Thus, each block is composed into three parts: *loading*, *objective*, and *processing*.

We add the objective part to the objective function and delete it from the block. After deleting the objective part from each block, the schedule can be considered as a

set of modified jobs J'_k with the setup time ST_k and the processing time PT_k . The jobs J'_k , $k = 1, \dots, n$, are scheduled in staggered order, i.e., job J'_1 is scheduled on the first machine, job J'_2 is scheduled on the second machine, job J'_3 is scheduled on the first machine, job J'_4 is scheduled on the second machine, and so on.

Formally, if we denote by st_j the starting time of each modified job J'_j , then

$$st_1 + ST_1 \leq st_2, \quad st_2 + ST_2 \leq st_3,$$

and so on, i.e., the inequality

$$st_j + ST_j \leq st_{j+1} \quad (24)$$

holds for each $j = 1, \dots, n - 1$;

$$st_1 + ST_1 + PT_1 \leq st_3, \quad st_2 + ST_2 + PT_2 \leq st_4,$$

and so on, i.e., the inequality

$$st_j + ST_j + PT_j \leq st_{j+2} \quad (25)$$

holds for each $j = 1, \dots, n - 2$.

We denote by F the total length of the modified schedule, i.e., the inequality

$$F \geq st_n + ST_n + PT_n \quad (26)$$

holds,

and the inequality

$$F \geq st_{n-1} + ST_{n-1} + PT_{n-1} \quad (27)$$

holds.

For each job J_j , the integer number $ch[j]$ is introduced with the following meaning. If J_j is the first level job for some block B_x , then $ch[j]$ denotes the maximal number of second level jobs for the same block. Formally, one can write

$$B_{x,2,1} + \dots + B_{x,2,n} \leq ch[1]B_{x,1,1} + \dots + ch[n]B_{x,1,n} \quad (28)$$

The objective function is

$$F + \sum_{x=1}^n \sum_{j=1}^n (s_j + p_j) B_{x,2,j}. \quad (29)$$

Since any schedule can be decomposed into a set of blocks, the following theorem holds.

Theorem 3. Any schedule s can be described as a feasible solution of system (20) - (27) and as a feasible solution of system (20) - (28), respectively. In both cases, the equality

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

holds.

Now, to prove the equivalence between the scheduling problem $P2, S1 || C_{max}$ and the models (20) - (27) and (20) - (28), respectively, one has to prove the following theorem.

Theorem 4. Any feasible solution of system (20) - (27) and any feasible solution of system (20) - (28), respectively,

can be described as a feasible schedule for the problem $P2, S1 || C_{max}$. In both cases, the equality

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

holds.

Thus, we consider three models in the following.

Model M0: Minimize (19) subject to the constraints (1) - (18),

model M1: Minimize (29) subject to the constraints (20) - (28), and

model M2: Minimize (29) subject to the constraints (20) - (27).

To evaluate the results obtained, we use the known lower bound

$$LB = \max\{LB_1, LB_2, LB_3\},$$

where

$$LB_1 = \frac{1}{2} \left(\sum_{i \in J} (s_i + p_i) + \min_{i \in J} \{s_i\} \right),$$

$$LB_2 = \sum_{i \in J} s_i + \min_{i \in J} \{p_i\},$$

$$LB_3 = \max_{i \in J} \{s_i + p_i\}$$

(see Gan et al. [2012]).

Next, we compare the models M0, M1 and M2 with the model and heuristics developed in Gan et al. [2012].

4. COMPUTATIONAL RESULTS

The performance of the models M0, M1 and M2 was tested on the data generated in the same way as it is described in Abdekhodae and Wirth [2002] and Gan et al. [2012].

For the instances with $n \in \{8, 20, 50, 100\}$, we have chosen the same time limits of $(300/8)n$ seconds as in Gan et al. [2012]. For the instances with $n \in \{200, 250\}$, we have chosen a run time limit of 3600 seconds.

Two independent experiments were made.

In the first experiment, we compared the performance of the models M0 and M1 with the performance of the model proposed in Gan et al. [2012], which we denote as MP. For $n \in \{8, 10, 14, 16, 18, 20\}$, 10 instances were generated for each

$$L \in \{0.1, 0.5, 0.8, 1, 1.5, 1.8, 2.0\}$$

with

$$p_j \stackrel{d}{=} U(0, 100) \quad \text{and} \quad s_j \stackrel{d}{=} U(0, 100L).$$

In the second experiment, we compared the performances of the models M1 and M2 with the model MP and with the heuristics proposed in Gan et al. [2012]. For $n \in \{8, \dots, 20\}$, the data sets were generated for server load values ranging between 0.1 and 2 with increments of 0.1, i.e., for each

$$L = \{0.1, 0.2, \dots, 2\}$$

the value s_j is uniformly distributed in $(0, 100L)$. For each value L , 10 instances were randomly generated with

$$p_j \stackrel{d}{=} U(0, 100),$$

Table 1. The time limit is 300 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	$\frac{C_{max}}{LB}$	ave $\frac{C_{max}}{LB}$	max $\frac{C_{max}}{LB}$
8	0.1	M0	0.1	30.1	145.7	1	1	1
		M1	0.1	28.4	111.8	1	1	1
		MP	0.1	79.8	300.0	1	1	1
	0.5	M0	0.3	1.3	3.3	1	1	1
		M1	0.4	8.7	24.1	1	1	1
		MP	2.2	30.0	109.3	1	1	1
	0.8	M0	0.1	0.5	0.9	1	1	1
		M1	0.1	1.6	2.8	1	1	1
		MP	0.1	3.8	9.0	1	1	1
	1	M0	0.1	0.4	1.1	1	1	1
		M1	0.0	1.2	3.7	1	1	1
		MP	0.2	17.2	135.6	1	1	1
	1.5	M0	0.0	0.7	2.3	1	1	1
		M1	0.0	0.7	2.5	1	1	1
		MP	0.0	2.3	10.3	1	1	1
	1.8	M0	0.0	0.3	0.8	1	1	1
		M1	0.0	0.7	1.6	1	1	1
		MP	0.1	1.7	5.6	1	1	1
	2.0	M0	0.0	0.1	0.3	1	1	1
		M1	0.0	0.1	1.0	1	1	1
		MP	0.1	0.2	1.5	1	1	1

Table 2. The time limit is 375 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	$\frac{C_{max}}{LB}$	ave $\frac{C_{max}}{LB}$	max $\frac{C_{max}}{LB}$
10	0.1	M0	0.5	362.3	375.1	1	1	1
		M1	248.0	362.3	375.1	1	1	1
		MP	49.5	337.6	375.0	1	1	1
	0.5	M0	0.5	18.1	77.1	1	1	1
		M1	8.8	342.5	375.0	1	1	1
		MP	4.0	156.2	375.0	1	1	1
	0.8	M0	1.4	15.4	41.8	1	1	1
		M1	9.8	101.0	318.1	1	1	1
		MP	8.5	131.1	375.0	1	1	1
	1.0	M0	0.0	6.0	21.2	1	1	1
		M1	0.0	31.0	176.7	1	1	1
		MP	0.2	56.0	274.0	1	1	1
	1.5	M0	0.0	7.2	27.1	1	1	1
		M1	0.0	36.9	107.0	1	1	1
		MP	0.1	93.2	375.0	1	1	1
	1.8	M0	0.0	20.7	146.8	1	1	1
		M1	0.0	31.5	219.9	1	1	1
		MP	0.1	28.0	189.5	1	1	1
	2.0	M0	0.0	7.5	37.5	1	1	1
		M1	0.0	15.4	82.5	1	1	1
		MP	0.1	7.3	34.6	1	1	1

i.e., p_j is uniformly distributed in $(0,100)$. For $n \in \{50, 100, 200, 250\}$, 5 instances were generated for each

$$L \in \{0.1, 0.5, 0.8, 1, 1.5, 1.8, 2.0\}$$

with

$$p_j \stackrel{d}{=} U(0, 100) \quad \text{and} \quad s_j \stackrel{d}{=} U(0, 100L).$$

The test instances have been solved using CPLEX 10.1 with 2GB of memory available for working storage, running on a personal computer Intel(R) Core(TM)i5-2430M CPU @2.4GHz.

The results of the first experiment are presented in Tables 1– 6.

The models M0 and M1 worked very fast for $n = 8$, see Table 1. We were able to find optimal solutions for all generated instances twice faster than in the case of using the model MP. For most instances, the use of the model M0 appears to be the fastest variant.

The model M0 was the best for $n = 10$, see Table 2. However, all the models were able to find optimal solutions for all generated instances within the time limit of 375 seconds.

Starting with $n = 14$, the model MP was comparable with the models M1 and M0, see Table 3 and Figure 2. Here, the time limit was 525 seconds. For $L \in \{0.1, 0.5, 0.8, 1.5, 1.8, 2.0\}$, the models M0 and M1 were preferable to the model MP but for $L = 1$, the model MP appeared to be the best one in terms of average time.

For $n = 16$, the model M0 was the best for most instances. The models M1 and MP were comparable in terms of average time, see Table 4. Here, the time limit was 600 seconds.

For $n = 18$, the model M0 turned out to be the best with respect to the quality of the obtained solutions, see Table 5 and Figure 3. However, it is difficult to say which model

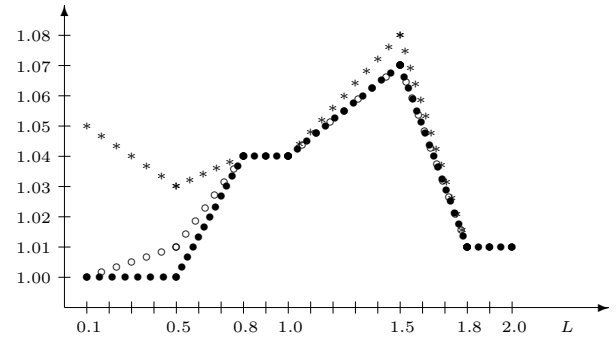


Fig. 2. Values $\max \frac{C_{max}}{LB}$ for $n = 14$. \bullet refers to the model M0, \circ refers to the model M1, and $*$ refers to the model MP.

turned out to be the fastest one. Here, the time limit was 675 seconds.

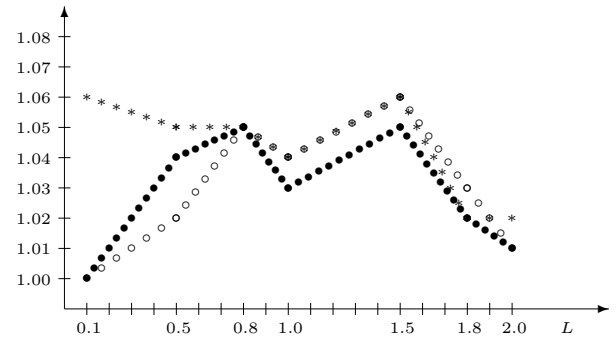


Fig. 3. Values $\max \frac{C_{max}}{LB}$ for $n = 18$. \bullet refers to the model M0, \circ refers to the model M1, and $*$ refers to the model MP.

Table 3. The time limit is 525 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	$\min \frac{C_{max}}{LB}$	$\text{ave} \frac{C_{max}}{LB}$	$\max \frac{C_{max}}{LB}$
14	0.1	M0	27.2	432.6	525.0	1.00	1.00	1.00
		M1	246.8	477.6	525.1	1.00	1.00	1.00
		MP	385.7	511.1	525.0	1.00	1.01	1.05
	0.5	M0	12.0	288.9	525.0	1.00	1.00	1.00
		M1	525.0	525.0	525.1	1.00	1.00	1.01
		MP	35.9	476.1	525.0	1.00	1.01	1.03
	0.8	M0	7.8	404.7	525.0	1.00	1.02	1.04
		M1	525.0	525.0	525.0	1.00	1.02	1.04
		MP	345.8	493.1	525.0	1.00	1.02	1.04
	1.0	M0	0.2	472.5	525.0	1.00	1.02	1.04
		M1	0.1	472.5	525.1	1.00	1.02	1.04
		MP	0.5	439.2	525.0	1.00	1.02	1.04
	1.5	M0	0.6	338.6	525.0	1.00	1.02	1.07
		M1	0.1	315.2	525.0	1.00	1.02	1.07
		MP	1.4	313.0	525.0	1.00	1.02	1.08
	1.8	M0	0.0	210.2	525.0	1.00	1.00	1.01
		M1	0.1	210.2	525.0	1.00	1.00	1.01
		MP	0.4	211.2	525.0	1.00	1.00	1.01
	2.0	M0	0.1	210.2	525.0	1.00	1.00	1.01
		M1	0.0	241.7	525.0	1.00	1.00	1.01
		MP	0.7	265.9	525.0	1.00	1.00	1.01

Table 5. The time limit is 675 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	$\min \frac{C_{max}}{LB}$	$\text{ave} \frac{C_{max}}{LB}$	$\max \frac{C_{max}}{LB}$
18	0.1	M0	25.1	326.5	675.0	1.00	1.00	1.00
		M1	675.0	675.0	675.1	1.00	1.00	1.00
		MP	6.3	477.1	675.0	1.00	1.02	1.06
	0.5	M0	114.9	583.5	675.0	1.00	1.01	1.04
		M1	675.0	675.0	675.1	1.00	1.00	1.02
		MP	152.2	622.7	675.0	1.00	1.02	1.05
	0.8	M0	675.0	675.0	675.1	1.00	1.02	1.05
		M1	675.0	675.0	675.1	1.00	1.02	1.05
		MP	675.0	675.0	675.0	1.00	1.02	1.05
	1.0	M0	675.0	675.0	675.1	1.00	1.01	1.03
		M1	12.2	542.7	675.0	1.00	1.01	1.04
		MP	11.7	517.0	675.0	1.00	1.01	1.04
	1.5	M0	0.5	413.2	675.1	1.00	1.02	1.05
		M1	1.2	406.0	675.0	1.00	1.02	1.06
		MP	3.1	346.3	675.0	1.00	1.02	1.06
	1.8	M0	0.2	508.2	675.0	1.00	1.01	1.02
		M1	0.2	540.1	675.0	1.00	1.01	1.03
		MP	2.9	475.1	675.0	1.00	1.01	1.02
	2.0	M0	0.4	204.3	675.0	1.00	1.00	1.01
		M1	0.1	276.8	675.1	1.00	1.00	1.01
		MP	1.3	205.8	675.0	1.00	1.00	1.02

Table 4. The time limit is 600 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	$\min \frac{C_{max}}{LB}$	$\text{ave} \frac{C_{max}}{LB}$	$\max \frac{C_{max}}{LB}$
16	0.1	M0	2.3	407.0	600.1	1.00	1.00	1.00
		M1	381.1	578.2	600.3	1.00	1.00	1.00
		MP	42.1	539.1	600.0	1.00	1.00	1.02
	0.5	M0	70.7	496.4	600.0	1.00	1.00	1.00
		M1	600.0	600.0	600.2	1.00	1.00	1.00
		MP	286.1	568.5	600.0	1.00	1.01	1.05
	0.8	M0	256.3	543.3	600.1	1.00	1.02	1.06
		M1	600.0	600.0	600.0	1.00	1.02	1.06
		MP	419.0	581.9	600.0	1.00	1.03	1.06
	1.0	M0	600.0	600.0	600.0	1.00	1.02	1.04
		M1	600.0	600.0	600.0	1.00	1.02	1.04
		MP	47.3	544.7	600.0	1.00	1.02	1.04
	1.5	M0	0.2	284.0	600.0	1.00	1.00	1.02
		M1	0.1	300.3	600.0	1.00	1.00	1.02
		MP	0.8	308.4	600.0	1.00	1.00	1.02
	1.8	M0	0.1	300.1	600.0	1.00	1.01	1.08
		M1	0.1	300.4	600.0	1.00	1.01	1.08
		MP	1.1	256.4	600.0	1.00	1.01	1.09
	2.0	M0	0.0	240.1	600.0	1.00	1.00	1.02
		M1	0.0	302.4	600.0	1.00	1.00	1.02
		MP	0.4	301.1	600.0	1.00	1.00	1.01

For $n = 20$, the models M0 and M1 were preferable in terms of the quality of the obtained solutions, see Table 6 and Figure 4. Here, the time limit was 750 seconds.

In the second experiment, the models M1 and M2 were compared with the results of Gan et al. [2012]. In Figure 5, we show the variations of the value $\frac{C_{max}}{LB}$ in dependence on the number of jobs.

The model M1 was used for $n \in \{8, 20, 50\}$, and the model M2 was used for $n \in \{100, 200, 250\}$.

For $n = 8$, we were able to find optimal solutions for all generated instances within 50 seconds. Note that in the first experiment, the maximal time used by the model M1 is 111.8 seconds, see Table 1. However, such a time was met

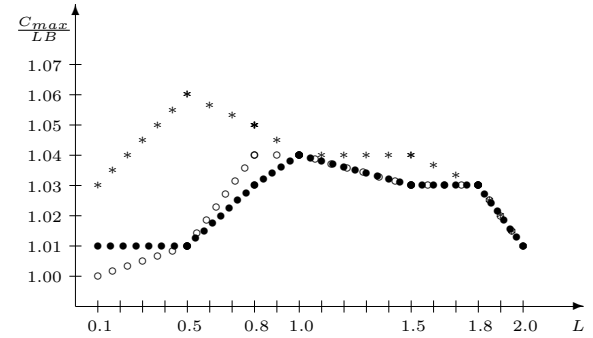


Fig. 4. Values $\max \frac{C_{max}}{LB}$ for $n = 20$. \bullet refers to the model M0, \square refers to the model M1, and $*$ refers to the model MP.

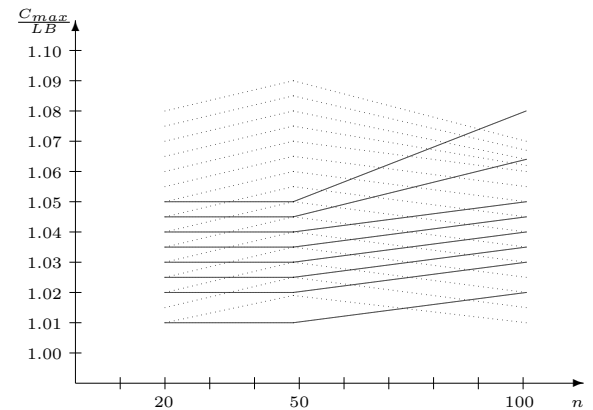


Fig. 5. Variations of $\frac{C_{max}}{LB}$. The thick lines refer to the models M1 and M2, the dotted lines refer to the results of Gan et al., where, however, for $n \in \{50, 100\}$ only the quotient "Worst makespan/Best makespan" is given.

only for one instance. In Gan et al. [2012], for the same instances with $n = 8$, optimal solutions for all generated instances were found only within 300 seconds.

Table 6. The time limit is 750 seconds

1	2	3	4	5	6	7	8	9
n	L	mdl	min time	ave time	max time	min $\frac{C_{max}}{LB}$	ave $\frac{C_{max}}{LB}$	max $\frac{C_{max}}{LB}$
20	0.1	M0	44.9	627.6	750.0	1.00	1.00	1.01
		M1	750.0	750.0	750.2	1.00	1.00	1.00
		MP	668.5	735.0	750.0	1.00	1.01	1.03
	0.5	M0	750.0	750.0	750.0	1.00	1.00	1.01
		M1	750.0	750.0	750.1	1.00	1.00	1.01
		MP	750.0	750.0	750.0	1.00	1.01	1.06
	0.8	M0	750.0	750.0	750.2	1.01	1.01	1.03
		M1	750.0	750.0	750.2	1.01	1.01	1.04
		MP	750.0	750.0	750.0	1.01	1.02	1.05
	1.0	M0	234.2	698.4	750.0	1.00	1.02	1.04
		M1	242.7	699.3	750.0	1.00	1.02	1.04
		MP	27.9	677.8	750.0	1.00	1.02	1.04
	1.5	M0	0.4	501.9	750.0	1.00	1.01	1.03
		M1	0.64	528.8	750.0	1.00	1.01	1.03
		MP	4.4	601.0	750.0	1.00	1.01	1.04
	1.8	M0	0.7	380.5	750.0	1.00	1.01	1.03
		M1	0.1	377.9	750.0	1.00	1.01	1.03
		MP	7.4	411.8	750.0	1.00	1.01	1.03
	2.0	M0	0.2	449.5	750.0	1.00	1.00	1.01
		M1	0.2	556.9	750.0	1.00	1.00	1.01
		MP	3.2	315.1	750.0	1.00	1.00	1.01

For $n = 20$, we used a time limit of 750 seconds.

- For M1, the maximal value for the relation $\frac{C_{max}}{LB}$ was 1.05, and the average value for the relation $\frac{C_{max}}{LB}$ was 1.01
- while in Gan et al. [2012], the maximal value for the relation $\frac{C_{max}}{LB}$ was 1.08 and the average value for the relation $\frac{C_{max}}{LB}$ was 1.01.

For $n = 50$, we used a time limit of 1875 seconds.

- For M1, the maximal value for the relation $\frac{C_{max}}{LB}$ was 1.05, and the average value for the relation $\frac{C_{max}}{LB}$ was 1.01
- while in Gan et al. [2012], the best makespan was compared not with LB but with the worst makespan among the heuristics developed. The relation "Worst makespan/Best makespan" was ranging from 1.02 to 1.09.

Table 7. The average and the maximal gaps for $n = 200$

Load	0.1	0.5	0.8	1.0	1.5	1.8	2.0
ave $\frac{C_{max}}{LB}$	1.01	1.04	1.07	1.09	1.01	1.00	1.00
max $\frac{C_{max}}{LB}$	1.01	1.08	1.10	1.12	1.02	1.01	1.01

For $n = 100$, we used a time limit of 3750 seconds.

- For M2, the maximal value for the relation $\frac{C_{max}}{LB}$ was 1.08, and the average value for the relation $\frac{C_{max}}{LB}$ was 1.02
- while in Gan et al. [2012], the relation "Worst makespan/Best makespan" was ranging from 1.01 to 1.07.

For $n = 200$ and for $n = 250$, we used a time limit of 3600 seconds.

Table 8. The average and the maximal gaps for $n = 250$

Load	0.1	0.5	0.8	1.0	1.5	1.8	2.0
ave $\frac{C_{max}}{LB}$	1.02	1.07	1.10	1.10	1.02	1.00	1.00
max $\frac{C_{max}}{LB}$	1.03	1.09	1.11	1.12	1.04	1.00	1.01

- For M2, the maximal values and the average values of C_{max}/LB for each load value are presented in Tables 7 and 8,
- while in Gan et al. [2012], tests have been made only for $n \leq 100$.

5. CONCLUSION

We developed three mixed integer linear programming formulations for the problem of scheduling a set of jobs on two parallel machines with a single server. Three models were tested and the performance was compared with that of the heuristics developed in Gan et al. [2012]. The computational results show that the new models outperform all heuristics proposed in Gan et al. [2012] for most types of instances.

REFERENCES

- A. Abdekhodae, A. Wirth. Scheduling parallel machines with a single server: some solvable cases and heuristics. *Computers & Operations Research*, 29:295–315, 2002.
- P. Brucker, C. Dhaenens-Flipo, S. Knust, S.A. Kravchenko, F. Werner. Complexity results for parallel machine problems with a single server. *Journal of Scheduling*, 5:429–457, 2002.
- H.S. Gan, A. Wirth, A. Abdekhodae. A branch-and-price algorithm for the general case of scheduling parallel machines with a single server. *Computers & Operations Research*, 39:2242–2247, 2012.
- N. Hall, C. Potts, C. Sriskandarajah. Parallel machine scheduling with a common server. *Discrete Applied Mathematics*, 102:223–243, 2000.
- F. Werner, S.A. Kravchenko. Scheduling with multiple servers. *Automation and Remote Control*, 71:2109–2121, 2010.