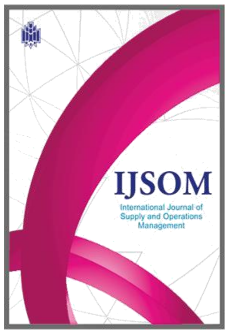
International Journal of Supply and Operations Management

IJSOM

May 2019, Volume 6, Issue 2, pp. 159-167
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



A New Formulation for the Single Machine Order Acceptance and Scheduling Problem with Sequence-Dependent Setup Times

Papatya Sevgin Bicakci *,[a] and İmdat Kara [b]

[a] *Department of Management, Faculty of Economic and Administrative Sciences, Başkent University, Ankara, Turkey*
[b] *Department of Industrial Engineering, Faculty of Engineering, Başkent University, Ankara, Turkey*

Abstract
Order acceptance and scheduling problem consist of simultaneously deciding which orders to be selected and how to schedule these selected orders. An extension of the sequence-dependent setup times and release dates was introduced in 2010 and a mathematical formulation was presented. Since then, a few mathematical formulations have appeared in the literature by addressing this problem on the basis of sequence-dependent setup times. However, some of the mathematical formulations are nonlinear or lack usability. Therefore, the model presented in 2010 is still being considered in recent studies. In this paper, we investigated the case in which there are sequence-dependent setup times with no release dates for all orders. We developed a new mathematical formulation with $O(n^2)$ binary variables and $O(n^2)$ constraints. In order to see the performance of our formulation, we conducted a computational analysis with CPLEX 12.4 by solving benchmark instances available in the literature. To manage the comparison, we reduced the existing formulation to the without release dates for all orders. As a result, we observed that the existing formulation can solve the test problems with up to 10 orders in a given time limit. On the other hand, our proposed formulation can solve all the available instances with up to 100 orders within the same time limit. Our proposed formulation is extremely faster than the existing one and can solve small and moderate sized real-life problems in a reasonable time. Thus, the researchers do not need any special heuristics for solving such problems. Instead, they can directly use our formulation with an optimizer.

Keywords: Order acceptance, Single-machine scheduling, Mathematical formulation, Order rejection, Sequence-dependent setup times

1. Introduction

Over the years, production scheduling has attracted considerable attention in industrial engineering and operation management fields. In classical scheduling problems, it is assumed that all orders must be processed. However, in real world applications, it may not be possible or profitable to accept all orders. There is a trade-off between the revenue brought in by the order and the amount of resources diverted to its processing (Silva, Y.L.T., et al. 2018). Specifically, in make-to-order production systems, accepting all orders may cause delays in completion time and tardiness cost accordingly (Wang, S. & Ye, B., 2019). More importantly, it may cause customer dissatisfaction or customer loss as well. To avoid these circumstances, the firms tend to reject some orders (Sarvestani, H.K., et al., 2019).

The order acceptance and scheduling (OAS) problem consists of simultaneously deciding which orders to be selected and how to schedule these selected orders (Slotnick, S.A., 2011). This problem occurs in make-to-order production systems when the capacity for meeting all demand on time is not enough. Ghosh, J.B. (1997) showed that the problem is NP-hard. Most of the studies have focused on heuristics in the related literature. Many papers in the literature do not even include mathematical models at the beginning of the study. However, today's technology enables that well-designed mathematical formulations to solve small and moderate sized real-life problems in a reasonable time.

Corresponding author email address: papatyas@baskent.edu.tr

In the literature, there are 27 papers related to single-machine order acceptance and scheduling problem. These papers are summarized in Table 1. Only 9 of them include mathematical formulations. Stern, H.I., & Avivi, Z. (1990)'s paper is considered as the first example. In their paper, an order is rejected if it is not finished before its due date. Charnsirisakskul, K., et al. (2004, 2006) considered varying prices and customer chosen due dates. They also took inventory costs into account and proposed two formulations. Oğuz, C., et al. (2010) introduced the original problem and proposed a formulation. They considered sequence-dependent setup times and release dates simultaneously. The problem they introduced has attracted a lot of researchers' attention. Nobibon, F.T., & Leus, R. (2011) considered the problem in which there are obligatory orders and proposed two mathematical formulations. In another study, Garcia, C. (2016) addressed the OAS problem considering resource limits and due windows and proposed a formulation. In his study, an order is rejected if it cannot be finished within its due window. Trigos, F., & Lopez, E.M. (2017) addressed the problem by using the sequence-dependent setup times depending on the lot size. They considered the available total time constraint and proposed a formulation. Zandieh, M., & Roumani, M. (2017) defined an upper limit for the number of accepted orders and considered sequence-dependent setup times. They proposed a nonlinear formulation. Silva, Y.L.T., et al. (2018) dealt with the same problem Oğuz, C., et al. (2010) addressed in their study. They proposed a time-indexed formulation with different revenue calculation method. Silva, Y.L.T., et al. (2018)'s time- indexed formulation has some structural difficulties with no flexibility for getting adapted to different circumstances.

Table 1. Single-Machine OAS Problem Research

#	Year	Authors	Problem structure	Release date	Setup time	Solution approach	Objectives
1	1990	Stern and Avivi*	$1\|rej, pmtn\|\sum P_i$	-	-	Heuristics algorithms	Max. profit
2	1996	Slotnick and Morton	$1\|rej\|\sum P_i-\sum w_jT_j$	-	-	One heuristic algorithm	Max. net profit
3	1997	Ghosh	$1\|rej\|\sum P_i-\sum w_jT_j$	-	-	Mathematical model and exact algorithm	Max. net profit
4	1997	Keskinocak et al.	$1\|rej, r_j\|\sum P_i$	✓	-	Exact algorithm	Max. profit
5	2001	Keskinocak et al.	$1\|rej, r_j\|\sum P_i$	✓	-	Exact algorithm	Max. profit
6	2002	Lewis and Slotnick	$1\|rej\|\sum P_i-\sum w_jT_j$	-	-	Exact and heuristic algorithms	Max. net profit
7	2003	Yang and Geunes	$1\|rej, r_j\|\sum P_j-\sum T_j$	✓	-	Heuristic algorithms	Max. net profit
8	2004	Charnsirisakskul et al.*	$1\|rej, pmtn\|\sum P_i-\sum w_jT_j$	-	-	Mathematical model	Max. net profit
9	2006	Charnsirisakskul et al.*	$1\|rej, pmtn,prc,r_j\|\sum P_i-\sum w_jT_j$	✓	-	Mathematical model	Max. net profit
10	2007	Slotnick and Morton	$1\|rej\|\sum P_i-\sum w_jT_j$	-	-	Exact and heuristic algorithms	Max. net profit
11	2007	Yang and Geunes	$1\|rej, r_j\|\sum P_j-\sum T_j$	✓	-	Numerical examples	Max. net profit
12	2009	Rom and Slotnick	$1\|rej\|\sum P_i-\sum w_jT_j$	-	-	Genetic algorithm	Max. net profit
13	2010	Oğuz et al.*	$1\|rej, s_{ij},\bar{d_i},r_j\|\sum R_j$	✓	✓	Mathematical model and heuristic algorithms	Max. revenue
14	2011	Nobibon and Leus *	$1\|rej\|\sum P_j-\sum w_jT_j$	-	-	Mathematical model and exact algorithms	Max. net profit
15	2012	Cesaret et al.	$1\|rej, s_{ij}, r_j\|\sum R_j$	✓	✓	Tabu search algorithm	Max. revenue
16	2014	Chen et al.	$1\|rej, s_{ij}, r_j\|\sum P_i-\sum w_jT_j$	✓	✓	Genetic algorithm	Max. net profit
17	2013	Lin and Ying	$1\|rej,s_{ij},r_j\|\sum P_i-\sum w_jT_j$	✓	✓	Artificial bee colony algortihm	Max. net profit
18	2014	Nguyen et al.	$1\|rej,s_{ij},r_j\|\sum P_i-\sum w_jT_j$	✓	✓	Genetic algorithm	Max. net profit

Table 1. Continued

#	Year	Authors	Problem structure	Release date	Setup time	Solution approach	Objectives
19	2015	Nafchi and Moslehi	$1\|rej, F^j\|\sum P_j$-RC	-	-	Hybrid genetic and linear programming algorithm	Max. net profit
20	2015	Wang et al.	$1\|rej\|\sum P_j-\sum c_j$-OC	-	-	Numerical examples	Max. net profit
21	2016	Hosteins et al.	$1\|rej\|\sum P_j-\sum T_j$	-	-	Exact algorithm	Max. net profit
22	2016	Garcia *	$1\|rej,r_j,D_j^-,D_j^+\|\sum P_j$-RC	✓	-	Mathematical model and heuristic algorithms	Max. net profit
23	2016	Nguyen	$1\|rej,s_{ij},r_j\|\sum P_i-\sum w_jT_j$	✓	✓	Learning and optimizing system	Max. net profit
24	2016	Trigos and Lopez *	$1\|rej,batch,s_{ij}\|\sum \Pi_j$	-	✓	Exact and heuristic algorithms	Max. profit
25	2016	Xie and Wang	$1\|rej, s_{ij}\|\sum P_i$	-	✓	ABC algorithm	Max. profit
26	2017	Zandieh ve Roumani *	$1\|rej,s_{ij}\|\sum P_i-\sum w_jT_j$	-	✓	Mathematical model and heuristic algorithms	Max. net profit
27	2018	Silva et al.*	$1\|rej, s_{ij},\bar{d_i},r_j\|\sum P_i-\sum w_jT_j$	✓	✓	Exact and heuristic algorithms	Max. net profit

Note: 1:single machine; rej: rejection; pmtn: preemption; s_{ij}: setup time; prc: pricing; r_i: release date; $\bar{d_i}$: deadline; p_i: processing time; R_i=max(0, e_i-w_iT_i); $\sum P_i$: total revenue; $\sum T_i$: total tardiness; RC: total rejection cost; $\sum c_i$: total cost; $\sum w_iTi$: total weighted tardiness; OC: total contract manufacturing cost; F^j: product families; $\sum \Pi_j$: total profit; D_j^-,D_j^+: due windows. Max.: maximizing; Min.: minimizing; net profit: (total profit – total tardiness penalties). ✓Included; - Not included; * Studies including mathematical formulation.

Order acceptance and scheduling problem are still drawing the researchers' attention. Studies on extensions of OAS problem with parallel machines have increased in recent years (Wu, G.H., et al., 2018; Wang, S., & Ye, B., 2019).

In recent years, the developments of computer technology and software enable us to solve the combinatorial problems by well-designed mathematical formulations. Mathematical formulations can be useful in real-life applications and allow post-optimality analysis. In most of the studies on OAS problem, solving methods generally focus on heuristic algorithms. However, as Della Croce, F. (2016) stressed, mathematical formulations are still useful for the scheduling problems.

In this study, we proposed a new formulation for single machine OAS problem with sequence-dependent setup times which can solve all the available instances with up to 100 orders in the literature. We also reduce the release date constraints according to Oğuz, C., et al. (2010) formulation and conduct a detailed computational analysis to compare Oğuz, C., et al.'s (2010) formulation with our proposed formulation.

The organization of this paper is as follows. Section 2 defines the OAS problem with sequence-dependent setup times. Section 3 introduces the new mathematical formulation for a single machine OAS problem with sequence-dependent setup times. Section 4 presents the results of the computational analysis by comparing the performance of the new formulation with Oğuz, C., et al.'s (2010) formulation. Finally, section 5 provides the concluding remarks.

2. The Order Acceptance and Scheduling Problem

The OAS problem is defined as follows. In a single-machine environment, we are given a set of orders {0,1,2,…,n,n+1}, where 0 is the beginning of the schedule and "n+1" shows the end of the schedule. The number of the orders is known, and the order of arrival time is assumed to be certain. Firm must reject some orders because of the limited production capacity. There is a tradeoff between the profit of accepted orders and the penalty weight of the delayed orders.

Set:
N={i|i=0,1,2,…,n,n+1} is the set of orders.

Parameters:
p_i: The processing time of order i
d_i: The due date of order i
$\bar{d_i}$: The deadline of order i

e_i: The maximum revenue of order i
w_i: The penalty weight of order i
s_{ij}: The setup time of order j, if it is processed immediately after order i, i,jϵN

Decision variables:
Z_{ij}: The completion time of order j immediately after order i,jϵN
T_i: The tardiness of order i
Y_i: It is equal to 1, if order i is accepted; 0 otherwise
X_{ij}: It is equal to 1, if order j is processed immediately after order i; 0 otherwise, i,jϵN

Let N' $\subset$N is the set of accepted orders. An order can be completed by the given deadline. As the tardiness of an order increases, its revenue linearly decreases; thus, its revenue would be equal to 0 after its deadline. Therefore, it is guaranteed that an order is rejected if it is completed after its deadline. To ensure this, w_i is set to $e_i/(\overline{d_i}-d_i)$ for each order iϵN. The revenue function for order i is showed in Figure 1. The revenue of order i would be equal to max (0, e_i-w_iT_i). The revenue varies according to the completion time of order i. The aim is to find the set N' and their sequence which maximizes the total revenue.

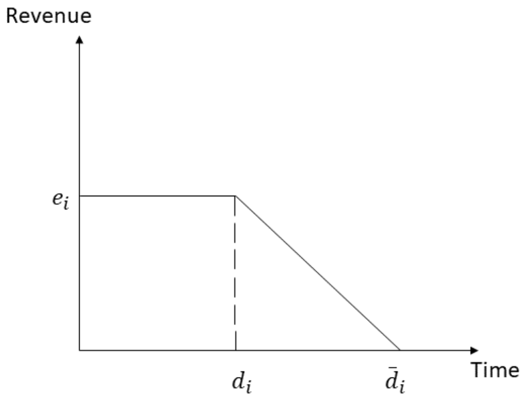


Figure 1. Revenue function for order i

3. The New Mathematical Formulation

In this section, we introduce the new formulation for single machine OAS problem with sequence-dependent setup times. The new formulation (BK) is given below.

$$\text{Max} \sum_{i=1}^{n} R_i \tag{1}$$

St.

$$\sum_{i=1}^{n} X_{0i} = 1 \tag{2}$$

$$\sum_{i=1}^{n} X_{i,n+1} = 1 \tag{3}$$

$$\sum_{j=1,i\neq j}^{n+1} X_{ij} = Y_i \qquad \forall_i = 1, \dots, n \tag{4}$$

$$\sum_{j=0,i\neq j}^{n} X_{ji} = Y_i \qquad \forall_i = 1, \dots, n \tag{5}$$

$$\sum_{j=1}^{n+1} Z_{ij} - \sum_{k=0}^{n} Z_{ki} = \sum_{j=1}^{n+1} (s_{ij} + p_j) X_{ij} \quad i \neq j, \quad \forall_i = 1, \dots, n, \quad \forall_j = 1, \dots, n+1 \tag{6}$$

$$Z_{0i} = (s_{0i} + p_i) X_{0i} \qquad \forall_i = 1, \dots, n \tag{7}$$

$$\sum_{k=0}^{n} Z_{ki} \leq \overline{d_i} Y_i \qquad \forall_i = 1, \dots, n \tag{8}$$

$$T_i \geq \sum_{k=0}^{n} Z_{ki} - d_i Y_i \qquad \forall_i = 1, \dots, n \tag{9}$$

$$Z_{ij} \leq \overline{d_{max}} X_{ij} \qquad i \neq j, \quad \forall_i = 0, \dots, n, \quad \forall_j = 1, \dots, n+1 \tag{10}$$

$$T_i \leq (\overline{d_i} - d_i) Y_i \qquad \forall_i = 1, \dots, n \tag{11}$$

$$R_i = e_i Y_i - T_i w_i \qquad \forall_i = 1, \dots, n \tag{12}$$

$$Y_i \epsilon \{0,1\} \qquad \forall_i = 1, \dots, n \tag{13}$$

$$X_{ij} \epsilon \{0,1\} \qquad i \neq j, \quad \forall_i = 0, \dots, n, \qquad \forall_j = 1, \dots, n+1 \tag{14}$$

In this formulation, 0 and “n+1” are the dummy orders. The objective function (1) maximizes the total revenue based on the accepted orders. Constraints (2) and (3) ensure that 0 is assigned to the first position in the sequence and “n+1” is assigned to the last position in the sequence. Constraint sets (4) and (5) guarantee that if an order is accepted, this order precedes only one order and is succeeded by only one order. Constraint set (6) ensures that if order j is processed immediately after order i, the completion time of order j must be equal to sum of the completion time of order i, the setup time from order i to order j and the processing time of order j. Constraint set (7) defines that the completion time of the first order in the sequence. Constraint set (8) enforces that if an order is not completed until its deadline, then it is not accepted. Constraint set (9) calculates the tardiness time of the orders. Constraint set (10) provides that the decision variables X_{ij} and Z_{ij} are connected and $\overline{d_{max}}$ denotes the $max_{i=1,\dots,n}\{\overline{d_i}\}$. Constraint set (11) provides an upper bound to the tardiness time of the orders. Constraint set (12) calculates the revenue gained when order i is accepted and incurs tardiness of T_i. Constraint sets (13) and (14) define the binary variables. The new formulation has $2n^2+8n+2$ constraints and n2+2n binary decision variables.

Constraints (2),(3),(4), and (5) are traditional assignment constraints, are the same as Oğuz, C., et al.’s (2010) formulation. Constraints (11),(12),(13) define the relations between tardiness, targets and revenue, so they are same as in Oğuz, C., et al.’s (2010) formulation. Constraints (14),(15),(16), and (17) are non-negativity and binary constraints. These constraints are not related to the structure of the formulation directly.
The main difference between our new formulation and Oğuz, C., et al.’s (2010) formulation depends upon the decision variables corresponding to the completion time of the orders. They define completion time by C_i for order i and assume the main constraints of their formulation as a function of C_i’s. We define totally different decision variables to calculate completion times. Z_{ij}’s are developed based on the successive orders, therefore, our main decision variables are arc-based. On the other hand, main decision variables of Oğuz, C., et al.’s (2010) formulation are node-based. Consequently, the main body of our formulation which includes constraints (6),(7),(8),(9), and (10) is completely different from Oğuz, C., et al.’s (2010) formulation.

4. Computational Analysis

In this section, we present the results of computational experiments in which we compare the performance of BK and reduceOğuz, C., et al.’s (2010) formulation (ROSB) based on run time and linear programming relaxations. Oğuz, C., et al.’s (2010) formulation is reduced by removing the release date constraints corresponding constraint set (4) to the original model. For detailed information, please check Oğuz, C., et al.’s (2010) paper.

The mathematical formulations were tested in the benchmark instances suggested by Cesaret, B., et al. (2012). They generated the benchmark instances as six groups according to the problem size. Problem size was determined from 10 to 100, more specifically, for n=10, 15, 20, 25, 50, and 100. They used two parameters; τ is tardiness factor and R is due date range. Both parameters can take the same values which are 0.1, 0.3, 0.5, 0.7, and 0,9. They generated 10 instances for each (τ,R) combination and there are totally 1500 instances. Processing times, revenues, and setup times were generated from uniform distribution with an interval of [1, 20], [1,20], and [1,10] respectively. Due dates are generated by using a factor which is selected randomly from a uniform distribution with an interval of [p_T(1-τ-R/2, 1-τ+R/2] in which p_T is the total processing time of all orders. Deadlines were generated by the equation $\overline{d_i}=d_i+Rp_i$. The tardiness penalties were determined by the equation $w_i=e_i/(\overline{d_i}-d_i)$.

BK and ROSB formulations were both executed in CPLEX 12.4 under the same computational environment which is Intel Xeon Phi 7290 1.5 GHz Processor 384 GB Ram. We set 7200 seconds time limit for n=10,15,20,25,50 and 14400 seconds time limit for n=100. Run times of BK and ROSB formulations for n=10 are presented in Table 2.

In Table 2, we clearly see that BK formulation outperforms ROSB in run times for each case. The average run times for BK are 0,86; 0,76, and 0,53, whereas ROSB formulation’s average run times are 446,27; 66,46, and 103,93 seconds. Maximum run times are 645,66; 147,83; 221,81 seconds for ROSB formulation whereas they are just 1,60; 1,22; 0,71 seconds for BK formulation.

Table 2. Run times of BK and ROSB formulations for n=10.

n=10

τ R		OPV	CPU (s)	
0.1 0.1			ROSB	BK
Instance 1		119	428,67	1,60
Instance 2		129	392,82	0,98
Instance 3		94	633,39	0,42
Instance 4		123	6,74	0,13
Instance 5		94	330,19	0,47
Instance 6		111	553,68	1,08
Instance 7		102	645,66	1,30
Instance 8		104	495,30	1,48
Instance 9		119	492,29	0,45
Instance 10		105	483,95	0,69
Avg.			446,27	0,86
0.1 0.5				
Instance 1		112	147,83	0,85
Instance 2		122	70,26	0,79
Instance 3		127	73,01	0,93
Instance 4		98	43,79	0,66
Instance 5		110	104,26	0,66
Instance 6		85	83,37	0,79
Instance 7		113	2,80	0,40
Instance 8		124	1,49	0,60
Instance 9		116,4	133,52	0,67
Instance 10		107	4,33	1,22
Avg.			66,46	0,76
0.3 0.1				
Instance 1		111	76,15	0,71
Instance 2		134	221,81	0,62
Instance 3		88	73,83	0,55
Instance 4		98	147,23	0,45
Instance 5		82	109,29	0,42
Instance 6		107	52,55	0,26
Instance 7		106	86,91	0,71
Instance 8		119	95,73	0,49
Instance 9		92	103,42	0,51
Instance 10		109	72,37	0,62
Avg.			103,93	0,53

Note: OPV: optimal value; CPU (s): run time in seconds.

The run time of ROSB formulation for the fourth instance for τ=0.1 and R=0.1 is short compared to others, because all orders can be processed in this case. Therefore, the LPR value is equal to the optimal value in the fourth instance.

ROSB formulation can be solved only for n=10. For larger n values, ROSB formulation cannot be solved in 7200 seconds for up to n=50 and for 14400 seconds for n=100. Therefore, we continue analysis with BK formulation for larger n values. Run times and LP relaxations of BK solutions for n=15, 20, 25, 50 are presented in Table 3.

In Table 3, we observe that BK formulation can solve all the instances for n=15, 20, 25 and 50 in average run times 1,10; 5,87; 16,38 and 1011,87 seconds respectively. The average percentages of deviations are 3% and 4%, which are very close to the optimal values. Run times and LP relaxations of BK formulation for n=100 are presented in Table 4.

Table 3. Performance of BK formulation for n=15, 20, 25, 50.

τ	R		
0.1	**0.1**	Avg. CPU (s)	Avg. %DEV.
n=15		1,10	0,04
n=20		5,87	0,04
n=25		16,38	0,03
n=50		1011,87	0,03

Note: CPU (s): run time in seconds; LPR: linear programming relaxation; %DEV.: percentage deviation between LPR and OPV which is (LPR-OPV)/OPV; Avg.: average.

For the instances with 100 orders, BK formulation can solve 4 of the 10 instances in given time limit. The optimal values are 1079, 1172, 1077, and 1062 for the instance 1, instance 2, instance 5, and instance 8, respectively. There are also six best values because the optimal values cannot be found for six instances which are instance 3, instance 4, instance 6, instance 7, instance 9, and instance 10.

According to LP relaxations, the average percentage of deviations is found to be % 3 which is very close to the optimal values. We believe that as the number of experiments increases, the more instances with 100 orders can be solved by BK formulation.

Table 4. Performance of BK formulation for n=100.

n=100

τ R	OPV	CPU (s)	LPR	%DEV.
0.1 0.1				
Instance 1	1079	13827,92	1106	0,03
Instance 2	1172	8377,3	1197	0,02
Instance 3	1040*	14400	1064	0,02
Instance 4	1128*	14400	1158	0,03
Instance 5	1077	12792,15	1101	0,02
Instance 6	1079*	14400	1106	0,03
Instance 7	1079*	14400	1106	0,03
Instance 8	1062	12856,63	1092	0,03
Instance 9	1104*	14400	1128	0,02
Instance 10	1009*	14400	1032	0,02
Avg.		-		0,03

Note: OPV: optimal value; CPU (s): run time in seconds; LPR: linear programming relaxation; %DEV.: percentage deviation between LPR and OPV which is (LPR-OPV)/OPV; Avg.: average. * Best values

5. Conclusion

In this paper, we studied order acceptance and scheduling problem with due dates, deadlines and sequence-dependent setup times which maximizes the total revenue from the accepted orders. The contribution of this paper is to propose a new mixed integer mathematical formulation which can solve all the instances available in the related literature. Our proposed formulation can solve all the instances with 10, 15, 20, 25, 50 orders and some of the instances with 100 orders in given time limit. Our proposed formulation also gives LP relaxations with the average of % 3 which is very close to the optimal values.

In the literature, many authors emphasize that the problem is strongly NP-hard. However, our proposed formulation can solve almost all the instances of OAS problem in a reasonable time. Most of the studies on single-machine OAS problem propose exact algorithms and heuristic algorithms. While there is a mathematical formulation capable of solving small to moderate real-life instances, heuristic algorithms are necessary to solve only larger instances. Therefore, our study is distinctive to show the capacity of a well-designed mathematical formulation.

In addition, our proposed formulation provides good upper bounds by LP relaxations. It is useful for the authors willing to develop heuristic algorithms for OAS problem with greater orders. As for future studies, one can try to propose a mathematical formulation which can solve the instances with more than 100 orders, or one can try to develop a heuristic algorithm for greater real-life problems.

References

Cesaret, B., Oğuz, C. and Salman, F. S. (2012). A tabu search algorithm for order acceptance and scheduling. *Computers & Operations Research,* Vol. 39, pp. 1197-1205.

Charnsirisakskul, K., Griffin, P. M. and Keskinocak, P. (2004). Order selection and scheduling with leadtime flexibility. *IIE transactions*, Vol. 36, pp. 697-707.

Charnsirisakskul, K., Griffin, P. M. and Keskinocak, P. (2006). Pricing and scheduling decisions with leadtime flexibility. *European Journal of Operational Research*, Vol. 171, pp. 153-169.

Chen, C., Yang, Z., Tan, Y. and He, R. (2014). Diversity controlling genetic algorithm for order acceptance and scheduling problem. *Mathematical Problems in Engineering*, Vol. 2014, pp. 1-11.

Della Croce, F. (2016). MP or not MP: that is the question. *Journal of Scheduling*, Vol. 19, pp. 33-42.

Garcia, C. (2016). Resource-constrained scheduling with hard due windows and rejection penalties. *Engineering Optimization*, Vol. 48, pp. 1515-1528.

Ghosh, J. B. (1997). Job selection in heavily loaded shop. *Computers and Operations Research*, Vol. 24, pp. 141-145.

Hosteins, P., Cordone, R. and Righini, G. (2016). The Prize-collecting Scheduling Problem with Deadlines. *Electronic Notes in Discrete Mathematics*, .Vol. 55, pp. 57-60.

Keskinocak, P., Ravi, R. and Tayur, S. (2001). Scheduling and reliable lead-time quotation for orders with availability intervals and lead-time sensitive revenues. *Management science*, Vol. 47, pp. 264-279.

Keskinocak, P., Ravi, R. and Tayur, S. (1997). Algorithms for reliable lead time quotation. *GSIA Working Paper*, Carnegie Mellon University, Pittsburgh, PA.

Lewis, H. F. and Slotnick, S. A. (2002). Multi-period job selection: planning workloads to maximize profit. *Computers & Operations Research,* Vol. 29, pp. 1081-1098.

Lin, S. W. and Ying, K. C. (2013). Increasing the total net revenue for single machine order acceptance and scheduling problems using an artificial bee colony algorithm. *Journal of the Operational Research Society*, Vol. 64, pp. 293-311.

Nguyen, S. (2016). A learning and optimizing system for order acceptance and scheduling. *The International Journal of Advanced Manufacturing Technology*, Vol. 86, pp. 2021-2036.

Nguyen, S., Zhang, M., Johnston, M. and Tan, K. C. (2013). Learning reusable initial solutions for multi-objective order acceptance and scheduling problems with genetic programming. *European Conference on Genetic Programming* pp. 157-168. Springer, Berlin, Heidelberg.

Nobibon, F. T. and Leus, R. (2011). Exact algorithms for a generalization of the order acceptance and scheduling problem in a single-machine environment. *Computers & Operations Research*, Vol. 38, pp. 367-378.

Oğuz, C., Salman, F. S. and Bilgintürk Yalçın, Z. (2010). Order acceptance and scheduling decisions in make-to-order systems. *International Journal of Production Economics*, Vol. 125, pp. 200-211.

Reisi-Nafchi, M. and Moslehi, G. (2015). A hybrid genetic and linear programming algorithm for two-agent order acceptance and scheduling problem. *Applied Soft Computing*, Vol. 33, pp. 37-47.

Rom, W. O. and Slotnick, S. A. (2009). Order acceptance using genetic algorithms. *Computers & Operations Research*, Vol. 36, pp. 1758-1767.

Sarvestani, H. K., Zadeh, A., Seyfi, M. and Rasti-Barzoki, M. (2019). Integrated order acceptance and supply chain scheduling problem with supplier selection and due date assignment. *Applied Soft Computing*, Vol. 75, pp. 72-83.

Silva, Y. L. T., Subramanian, A. and Pessoa, A. A. (2018). Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. *Computers & Operations Research*, Vol. 90, pp. 142-160.

Slotnick, S.A. (2011). Order acceptance and scheduling: A taxonomy and review. *European Journal of Operational Research*, Vol. 212, pp. 1-11.

Slotnick, S. A. and Morton, T. E. (2007). Order acceptance with weighted tardiness. *Computers & Operations Research*, Vol. 34, pp. 3029-3042.

Slotnick, S. A. and Morton, T. E. (1996). Selecting jobs for a heavily loaded shop with lateness penalties. *Computers & Operations Research*, Vol. 23, pp. 131-140.

Stern, H. I. and Avivi, Z. (1990). The selection and scheduling of textile orders. *European journal of operational research*, Vol. 44, pp. 11-16.

Trigos, F. and López, E. M. (2016). Maximising profit for multiple-product, single-period, single-machine manufacturing under sequential set-up constraints that depend on lot size. *International Journal of Production Research*, Vol. 54, pp. 1134-1151.

Wang, S. and Ye, B. (2019). Exact methods for order acceptance and scheduling on unrelated parallel machines. *Computers and Operations Research*, Vol. 104, pp. 159-173.

Wang, X., Zhu, Q. and Cheng, T. C. E. (2015). Subcontracting price schemes for order acceptance and scheduling. *Omega*, Vol. 54, pp. 1-10.

Wu, GH., Cheng, CY., Yang, HI. and Chena, CT. (2018). An improved water flow-like algorithm for order acceptance and scheduling with identical parallel machines. *Applied Soft Computing*, Vol. 71, pp. 1072-1084.

Xie, X. and Wang, X. (2016). An enhanced ABC algorithm for single machine order acceptance and scheduling with class setups. *Applied Soft Computing*, Vol. 44, pp. 255-266.

Yang, B. and Geunes, J. (2007). A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times. *Computers & Industrial Engineering*, Vol. 53, pp. 420-432.

Yang, B. and Geunes, J. (2003). Heuristic approaches for solving single resource scheduling problems with job-selection flexibility. *Department of Industrial & Systems Engineering*, University of Florida.

Zandieh, M. and Roumani, M. (2017). A biogeography-based optimization algorithm for order acceptance and scheduling. *Journal of Industrial and Production Engineering*, Vol. 34, pp. 312-321.