Order Acceptance and Scheduling Problem: A Proposed Formulation and the Comparison with the Literature

Papatya S. Bıçakcı[1](✉) and İmdat Kara[2]

[1] Faculty of Economics and Administrative Sciences, Department of Management, Başkent University Bağlıca Campus, 06790 Ankara, Turkey
papatyas@baskent.edu.tr

[2] Faculty of Engineering, Department of Industrial Engineering, Başkent University Bağlıca Campus, 06790 Ankara, Turkey

Abstract. In classical scheduling problem, it is assumed that all orders must be processed. In the order acceptance and scheduling (OAS) problem, some orders are rejected due to limited capacity. In make-to-order production environment, in which the OAS problem occurs, accepting all orders may cause overloads, delay in deliveries and unsatisfied customers. Oğuz et al. (2010) introduced the OAS problem with sequence-dependent setup times and release dates. In this paper, we propose a new mixed integer programming formulation with $O(n^2)$ decision variables and $O(n^2)$ constraints for the same problem. We conduct a computational analysis comparing the performance of our formulation with Oğuz et al. (2010) formulation. We use the benchmark instances, which are available in the literature. We observe that our formulation can solve all the instances up to 50 orders in a reasonable time, while Oğuz et al. (2010) formulation can solve only the instances with 10 orders in the same time limit.

Keywords: Scheduling · Order acceptance · Order rejection · Mathematical formulation · Single machine

1 Introduction

In scheduling problem, it is mostly assumed that all orders must be processed. However, in real-life applications this may not be the case. In make-to-order production environment, accepting all orders may cause overloads, delay in deliveries and unsatisfied customers. Therefore, firms tend to reject some orders. Order acceptance and scheduling (OAS) problem consists of deciding which orders to be accepted and determining the schedule of the accepted orders [1].

The OAS problem considered in this study was introduced by Oğuz et al. [2] in 2010 and a mathematical formulation was proposed. The problem is defined as follows. In a single machine environment, there is a set of orders shown as N = {1, 2, …, n}. For each order $i \epsilon N$, let r_i be the release date, p_i be the processing time, d_i be the due date, $\overline{d_i}$ be the deadline, e_i be the revenue, w_i be the unit penalty weight for the tardiness of the orders and s_{ji} be the sequence-dependent setup time occurring if the order i is processed immediately after order j. The setup operation of order $i \epsilon N$ can only be performed after its release date. The objective is to select and schedule a subset of orders that maximizes the total profit.

To the best of our knowledge, there are 27 studies related to single machine OAS problem in the literature. Only 9 of them include a mathematical formulation. These studies are summarized in Table 1. In [4]'s study, a job is rejected if it is not finished before its due date. This study is considered as the first example of OAS problem in the literature [1]. [5] considered varying prices and customer chosen due dates, while [6] considered inventory costs. They proposed two different formulations. [2] introduced the OAS problem with sequence-dependent setup times and release dates. They proposed a formulation and their study has taken remarkable attention by researches. In [7]'s study, there are obligatory jobs and they proposed two mathematical formulations. [8] addressed the OAS problem considering resource limits and due windows and proposed a formulation. In his study, a job is rejected if it cannot be finished within its due window. [9] studied the OAS problem with sequence-dependent setup times depending on lot sizes. They considered total available time constraint and proposed a formulation. [10] defined an upper limit for the number of accepted jobs and considered sequence-dependent setup times. They proposed a nonlinear formulation. [3] dealt with the problem as in [2]. They proposed a time-indexed formulation with different revenue calculation method. [3]'s time indexed formulation has some structural difficulties, so we did not consider this formulation in detail.

Table 1. Single machine OAS studies with mathematical formulation

Ref. #	Year	Authors	Problem structure	Release date	Setup time
[4]	1990	Stern and Avivi	$1\|rej,pmtn\|\sum P_i$	–	–
[5]	2004	Charnsirisakskul et al.	$1\|rej,pmtn\|\sum P_i-\sum w_jT_j$	–	–
[6]	2006	Charnsirisakskul et al.	$1\|rej,pmtn,prc,r_j\|\sum P_i-\sum w_jT_j$	✓	–
[2]	2010	Oğuz et al.	$1\|rej, s_{ij},\overline{d_i},r_j\|\sum R_j$	✓	✓
[7]	2011	Nobibon and Leus	$1\|rej\|\sum P_j-\sum w_jT_j$	–	–
[8]	2016	Garcia	$1\|rej,r_j,D_j^-,D_j^+\|\sum P_j-RC$	✓	–
[9]	2016	Trigos and Lopez	$1\|rej,batch,s_{ij}\|\sum \Pi_j$	–	✓
[10]	2017	Zandieh ve Roumani	$1\|rej,s_{ij}\|\sum P_i-\sum w_jT_j$	–	✓
[3]	2018	Silva et al.	$1\|rej, s_{ij},\overline{d_i},r_j\|\sum P_i-\sum w_jT_j$	✓	✓

Note 1: Single machine; rej: rejection; pmtn: preemption; s_{ij}: setup time; prc: pricing; r_i: release date; $\overline{d_i}$: 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. ✓ Included; - Not included

In recent years, the developments on computer technology and softwares enable us to solve the combinatorial problems to optimality 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 [12] express that mathematical formulations are still useful for the scheduling problems.

In this study, we propose a new formulation for single machine OAS problem with sequence-dependent setup times and release dates. We conduct a detailed computational analysis to compare Oğuz et al. [2] formulation and our proposed formulation.

The remainder of this paper is organized as follows. Section 2 introduces the new integer linear programming formulation for single machine OAS problem with sequence-dependent setup times and release dates. Section 3 presents the results of the computational experiments comparing the performance of our proposed and Oğuz et al. [2] formulation. Finally, Sect. 4 provides the concluding remarks of this work.

2 A New Formulation

In this section we introduce a new mixed integer linear programming formulation for OAS problem with sequence-dependent setup times and release dates. Let "0" and "n +1" be the dummy orders, which indicate the first order of the schedule and the last order of the schedule respectively. Decision variables are given as follows. Let T_i be the tardiness of order i, for iϵN. Let Z_{ij} be the completion time of order j, if order j is processed immediately after order i. Let Y_i be 1, if order i is accepted, 0 otherwise. Let X_{ij} be 1, if order j is processed immediately after order i, 0 otherwise. Our proposed formulation is given below:

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

s.t.

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

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

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

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

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

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

$$\sum\nolimits_{k=0}^{n} Z_{ki} \leq \overline{d_i} Y_i \quad \forall_i = 1, \ldots, n \tag{8}$$

$$Z_{ij} \leq \overline{d_{max}} X_{ij} \quad i \neq j, \quad \forall_i = 0, \ldots, n, \quad \forall_j = 1, \ldots, n+1 \tag{9}$$

$$T_i \geq \sum\nolimits_{k=0}^{n} Z_{ki} - d_i Y_i \quad \forall_i = 1, \ldots, n \tag{10}$$

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

$$T_i \geq 0 \quad \forall_i = 1, \ldots, n \tag{12}$$

$$R_i = e_i Y_i - T_i w_i \quad \forall_i = 1, \ldots, n \tag{13}$$

$$R_i \geq 0 \quad \forall_i = 1, \ldots, n \tag{14}$$

$$Y_i \in \{0, 1\} \quad \forall_i = 1, \ldots, n \tag{15}$$

$$X_{ij} \in \{0, 1\} \quad i \neq j, \quad \forall_i = 0, \ldots, n, \quad \forall_j = 1, \ldots, n+1 \tag{16}$$

$$Z_{ij} \geq 0 \quad i \neq j, \quad \forall_i = 0, \ldots, n, \quad \forall_j = 1, \ldots, n+1 \tag{17}$$

In the objective function (1), the difference between total revenue of accepted orders and total tardiness penalty of accepted orders is maximized. With constraint (2) and (3), it is guaranteed that "0" is assigned to the first position and "n+1" is assigned to the last position of the schedule. Constraint set (4) ensures that if an order is accepted, another order is processed immediately after this order; and constraint set (5) ensures that if an order is accepted another order is processed immediately before this order. Constraint set (6) calculates the completion time of the orders. Constraint set (7) calculates the completion time of the first processed order in the sequence. Constraint set (8) guarantees that if an order is not completed before its deadline, then the order is not accepted. Constraint set (9), where $\overline{d_{max}} = max_{i=1,\ldots,n}\{\overline{d_i}\}$, provides X_{ij} be zero, when Z_{ij} be zero, and gives an upper bound to Z_{ij}'s. Constraint set (10) calculates the tardiness of the orders. Constraint set (11) gives an upper bound to T_i's, while constraint set (12) provides a lower bound to T_i's. Constraint set (13) calculates the revenue of the orders, while constraint set (14) bounds it. Constraint sets (15) and (16) define the binary variables. Constraint set (17) gives a lower bound to Z_{ij}'s. Our proposed formulation has $2n^2+10n+2$ constraints and n^2+2n binary decision variables.

Constraints (2), (3), (4) and (5) are traditional assignment constraints, therefore they are same as in Oğuz et al. [2] formulation. Constraints (11), (12), (13) are due to the relations between tardiness, targets and revenue, so they are same as in Oğuz et al. [2] formulation. Constraints (14), (15), (16) and (17) are non-negativity and binary constraints. These constraints also are not related with the structure of the formulation directly.

Main difference between our formulation and Oğuz et al. [2] formulation depends upon the decision variables corresponding to completion times of the orders. Oğuz et al. [2] defines completion times by indexing with C_i for ith order and develops main constraints of their formulation as a function of C_i's and other decision variables. We define completely different decision variables for completion time in relation with preceding order i and j as Z_{ij}, thus our main decision variables are arc-based whereas Oğuz et al. [2] formulation's main decision variables are node-based.

In accordance with these above explanations, main body of our formulation which includes constraints (6), (7), (8), (9) and (10) is completely different from Oğuz et al. [2] formulation.

3 Computational Experiments

In this section we summarize the results of computational experiments comparing the performance of Oğuz et al. [2] formulation (OSB) and our proposed formulation (OPF). Mathematical formulations were coded in C++ and benchmark instances were solved by CPLEX 12.4 in an Intel Xeon Phi 7290 with 1.5 GHz and 384 GB of RAM.

Benchmark instances were generated by [11] and they are available on the web address http://home.ku.edu.tr/~coguz~/Research/Dataset_OAS.zip. There are six instance groups which consist of n = 10,15,20,25,50 and 100. Each instance group has 250 instances, and there are totally 1500 benchmark instances. For the instances with 10,15,20,25 and 50 orders the time limit is set 7200 seconds; for the instances with 100 orders the time limit is set 14400 seconds. Run times and LP relaxations of OPF and OSB for n = 10 are showed in Table 2.

Table 2. The results of OSB and OPF for n = 10

τ	**R**	OPV	CPU		LPR		% DEV.	
0.1	**0.1**		OSB	OPF	OSB	OPF	OSB	OPF
Instance 1		119	90,06	0,49	124	124	0,04	0,04
Instance 2		126	115,9	0,69	131	131	0,04	0,04
Instance 3		90	127,72	0,64	102	102	0,13	0,13
Instance 4		123	40,16	0,29	123	123	0,00	0,00
Instance 5		94	69,26	0,39	96	96	0,02	0,02
Instance 6		111	113,28	0,61	115	115	0,04	0,04
Instance 7		102	117,02	0,51	111	111	0,09	0,09
Instance 8		104	91,76	0,29	114	114	0,10	0,10
Instance 9		117	104,76	0,65	123	123	0,05	0,05
Instance 10		105	76,8	0,31	107	107	0,02	0,02
Avg.			94,67	0,48			0,05	0,05

Note: τ and R are the parameters for using to generate benchmark instances. OPV: optimal value; CPU: run time; LPR: linear programming relaxation; % DEV: percentage deviation between LPR and OPV which is (LPR-OPV)/(OPV). OSB: Oğuz et al. [2] formulation; OPF: our proposed formulation.

From Table 2, we observe that OPF is extremely faster than OSB with the average run times 0,48 and 94,67 respectively. LPR values are not different and the average percentage deviation is 0,05 which indicates the LPR values are very close to the optimal values.

We solved all the instances with n = 10 with each formulation. The average run time and the average percentage deviation are given in Table 3.

Table 3. The results of OPF for n = 10, 15, 20, 25, 50

	OSB		OPF	
	Ave.CPU	Ave.%DEV	Ave.CPU	Ave.%DEV
n = 10	68,70	0,07	0,56	0,07
n = 15	–	–	1,13	0,05
n = 20	–	–	4,18	0,04
n = 25	–	–	10,02	0,05
n = 50	–	–	1977,55	0,03

Ave.CPU: Average run time; Ave.%DEV: Average percentage deviation between LPR and OPV which is (LPR-OPV)/OPV.

OSB cannot solve the instances greater than 10 orders in given time limit. Therefore, we continue computational analysis with OPF. Average run times and average percentage deviations for n = 15, 20, 25 and 50 are given in Table 3. OPF can solve all the instances up to 50 orders in a reasonable time.

There are no benchmark instances between 50 and 100 orders. OPF cannot solve the instances with 100 orders in 14400 seconds time limit.

4 Concluding Remarks

In this study we proposed a new arc-based mathematical formulation for solving a variant of OAS problem that includes sequence-dependent setup times and release dates. Our proposed formulation and Oğuz et al. [2] formulation were tested on instances ranging from 10 to 50 orders. Oğuz et al. [2] formulation can solve only the instances with 10 orders in given time limit. Our proposed formulation can solve the instances up to 50 orders to optimality in the same time limit. Future studies may consider proposing new mathematical formulations capable of solving larger instances in a reasonable time.

References

1. Slotnick, S.A.: Order acceptance and scheduling: a taxonomy and review. Eur. J. Oper. Res. **212**, 1–11 (2011)
2. Oğuz, C., Salman, F.S., Bilgintürk Yalçın, Z.: Order acceptance and scheduling decisions in make-to-order systems. Int. J. Prod. Econ. **125**, 200–211 (2010)

3. Silva, Y.L.T., Subramanian, A., Pessoa, A.A.: Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. Comput. Oper. Res. **90**, 142–160 (2018)
4. Stern, H.I., Avivi, Z.: The selection and scheduling of textile orders. Eur. J. Oper. Res. **44**, 11–16 (1990)
5. Charnsirisakskul, K., Griffin, P.M., Keskinocak, P.: Order selection and scheduling with leadtime flexibility. IIE Trans. **36**, 697–707 (2004)
6. Charnsirisakskul, K., Griffin, P.M., Keskinocak, P.: Pricing and scheduling decisions with leadtime flexibility. Eur. J. Oper. Res. **171**, 153–169 (2006)
7. Nobibon, F.T., Leus, R.: Exact algorithms for a generalization of the order acceptance and scheduling problem in a single-machine environment. Comput. Oper. Res. **38**(1), 367–378 (2011)
8. Garcia, C.: Resource-constrained scheduling with hard due windows and rejectionpenalties. Eng. Optim. **48**, 1515–1528 (2016)
9. Trigos, F., López, E.M.: Maximising profit for multiple-product, single-period, single-machine manufacturing under sequential set-up constraints that depend on lot size. Int. J. Prod. Res. **54**, 1134–1151 (2016)
10. Zandieh, M., Roumani, M.: A biogeography-based optimization algorithm for order acceptance and scheduling. J. Ind. Prod. Eng. **34**, 312–321 (2017)
11. Cesaret, B., Oğuz, C., Salman, F.S.: A tabu search algorithm for order acceptance and scheduling. Comput. Oper. Res. **39**, 1197–1205 (2012)
12. Della Croce, F.: MP or not MP: That is the question. J. Sched. **19**(1), 33–42 (2016)