OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2018 – 11(3): 390-405 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
https://doi.org/10.3926/jiem.2541

Mathematical Models for a Batch Scheduling Problem to Minimize Earliness and Tardiness

Basar Ogun[1], Çigdem Alabas-Uslu[2]

[1]*Eczacibasi Yapi Gerecleri Furniture Plant (Turkey)*

[2]*Marmara University, Department of Industrial Engineering (Turkey)*

basarogun@marun.edu.tr, cigdem.uslu@marmara.edu.tr

Received: November 2017
Accepted: March 2018

Abstract:

Purpose: Today's manufacturing facilities are challenged by highly customized products and just in time manufacturing and delivery of these products. In this study, a batch scheduling problem has been addressed to enable on-time completion of customer orders in a lean manufacturing environment. The problem is optimizing the partitioning of product components into batches and scheduling of the resulting batches where each customer order is received as a set of products made of various components.

Design/methodology/approach: Three different mathematical models for minimization of total earliness and tardiness of customer orders are developed to provide on-time completion of customer orders and also, to avoid excess final product inventory. The first model is a non-linear integer programming model whereas the second is a linearized version of the first. Finally, to solve larger sized instances of the problem, an alternative linear integer model is presented.

Findings: Computational study using a suit set of test instances showed that the alternative linear integer model is able to solve all test instances in varying sizes within quite shorter computer times compared to the other two models. It has also been showed that the alternative model is able to solve moderate sized real-world problems.

Originality/value: The problem under study differentiates from existing batch scheduling problems in the literature owing to the inclusion of new circumstances that are present in real-world applications. Those are: customer orders consisting of multi-products made of multi-parts, processing of all parts of the same product from different orders in the same batch, and delivering the orders only when all related products are completed. This research also contributes to the literature of batch scheduling problem by presenting new optimization models.

Keywords: batch scheduling, earliness and tardiness, mixed integer programming formulation, on-time delivery

1. Introduction

A batch-processing machine (BPM) can simultaneously process several jobs in a batch, that is, different jobs in the same batch are processed at the same time. BPM scheduling problem is defined as a combination of sequencing and partitioning problems with an objective function to be optimized (Albers & Brucker, 1993). The aspect of partitioning is to find a partition of all jobs into batches whereas the sequencing aspect is to find a sequence of the batches formed by the partitioning.

BPM scheduling problem addressed in this study mainly stems from scheduling of a cutting machine in furniture manufacturing. The short-term production planning problem considers the following input data: customer orders which consist of a set of products, bill of materials which involves a set of components for each product, due dates of the customer orders, capacity of BPM (cutting machine) in terms of number of components and finally process time of BPM. Decision problem is to cluster the components of products into batches that have limited capacity and also to find a production sequence for these batches. The problem intends to minimize the sum of the weighted sum of early and late completion times of customer orders by determining optimum composition of each batch in terms of both type and number of components as well as optimum production sequence of these batches. To emphasize the importance of on-time completion of customer orders, we have called the problem BPM-On-time scheduling. One of the differentiating aspect of BPM-On-time scheduling problem arises from its policy of order of deliveries. According to this policy, a customer order cannot be delivered unless all products ordered by the same customer are completed. The other aspect is that products are made of several components and all components of a product belonging to the same customer order must be processed simultaneously because of technical requirements. In other words, products of a customer order can be partitioned into different batches but components of a product in the same customer order cannot be partitioned. This constraint is different from the constraint of incompatible job families considered in the literature since it does not allow processing of jobs from different job families simultaneously. Existing literature, which is closely related to the completion of customer orders just in time, focuses on either processing of jobs in batches to minimize due date related objectives or delivering of completed jobs in batches to minimize inventory and delivery costs. The latter is called batch delivery and was first introduced by Cheng and Kahlbacher (1993). Batch delivery problem mainly considers finding optimal partitions of jobs into batches for delivery. In BPM-On-time scheduling problem however, a customer order is a given set of products and it must be delivered once all the products ordered are completed. Therefore, BPM-On-time scheduling problem should not be treated in batch scheduling literature as batch delivery. In this study, we aim to contribute to the literature of batch scheduling by emphasizing the differentiating constraints of BPM-On-time scheduling problem. We also aim to contribute by suggesting optimization models to BPM-On-time scheduling problem since there is room for further research.

In this study, firstly, a nonlinear integer programming formulation to the problem, called Model-NL, is provided. Then, Model-NL is converted into an integer linear programming (ILP) model, named Model-PL, using the piecewise linearizing method. Finally, we suggest an alternative ILP model, Model-A, to deal with difficulties of both Model-NL and Model-PL. The proposed models, Model-A, Model-NL and Model-PL are compared using a suit set of test instances. The experimental studies confirm that Model-A strongly outperforms the other models in terms of both run time and problem size which can be solvable optimally.

Section 2 addresses the related works in the literature. In section 3, BPM-On-time scheduling problem is described and the three mathematical models are explained in detail. Section 4 contains experimental study in which generation of the test instances and results obtained by the models are given. A case study from furniture industry is presented in Section 5 to show the performance of suggested Model-A in real life. Finally, conclusions and future directions are discussed in the last section.

2. Related Works

BPM scheduling problem has received great attention since it was first proposed by Ikura and Gimple (1986). BPM scheduling may be seen in different types of manufacturing industries: Semiconductor industry, furniture manufacturing industry, iron and steel industry, and chemical processes incorporate several examples of BPM scheduling problem. Applications of BPM scheduling problem to varying real-world environments also resulted in variants of the problem. The literature involves a large quantity of these variants considering single or parallel machines, identical or non-identical jobs, compatible or incompatible job families, unbounded or bounded BPM capacity, processing a job at any time or processing it earliest at release time and so on. Different optimization objectives such as makespan, total completion time, maximum tardiness, total tardiness, and number of tardy jobs also contribute to the variants of BPM scheduling. There are several reviews of batch scheduling problems. Potts and van Wassenhove (1992) and Webster and Baker (1995) review the problem by focusing on the single machine environment. Potts and Kovalyov (2000) review the research related with different machine environments and several types of objectives. Mathirajan and Sivakumar (2003, 2006) and Mönch, Fowler, Dauzere-Peres, Mason and Rose (2011) especially focus on BPM scheduling problems which arise in semiconductor manufacturing. Xiao and Shao (2013) review BPM scheduling problem by emphasizing both job sizes and processing time. Yılmaz and Durmusoglu (2017) focus on dual resource constrain batch scheduling problem to minimize production cell response time. Genetic algorithm, simulated annealing and artificial bee colony metaheuristics are proposed ABC was found to deliver superior result compared to others.

In this study, in order to emphasize importance of on-time completion of customer orders, we restricted our survey to the research that aim to minimize early and tardy completion times of jobs in a BPM scheduling problem. There is an increasing trend in adoption of lean manufacturing tools and techniques to improve the efficiency of production systems and to decrease production costs. A lean manufacturing system requires completing customer orders just in time since early completion of orders results in increasing work-in-process costs while late completion causes both customer dissatisfaction and storage cost of finished products. Therefore minimization of both earliness and tardiness of orders is crucial in any manufacturing environment including BPM scheduling. Hall and Posner (1991) showed that the weighted sum of earliness and tardiness of jobs scheduled on a single processor considering a common due date is NP-complete. Brucker, Gladky, Hoogeveen, Kovalyov, Potts, Tautenhahn and van de Velde (1998) also proved that all batch scheduling problems with the optimization function related to due date are NP-hard. Qi and Tu (1999) proposed a dynamic programming algorithm for the scheduling of a single processing machine considering distinct due dates, a common process time and identical job size to minimize sum of earliness and tardiness. Mönch, Unbehaun and Choung (2006) suggested several two-phase heuristics to the same problem applied to scheduling of a burn-in oven in the semiconductor manufacturing industry with an additional content of maximum allowable tardiness. The proposed heuristics are based on genetic algorithms and dominance properties of optimal schedules. Mönch and Unbehaun (2007) also suggested decomposition heuristics in the case of parallel burn-in ovens. The suggested heuristics assign the jobs into early and tardy job sets for each of the parallel burn-in ovens. The early and tardy job sets are then used to form batches and finally, batches are sequenced optimally. Zhao, Hu and Li (2006) investigated properties of an optimal schedule for a batch scheduling problem with the assumptions of a common due window and identical jobs to minimize the total weighted earliness and tardiness. Li, Chen, Xu and Li (2015) proposed a new heuristic algorithm to assign jobs into batches for the scheduling problem under constraints of capacitated batch and non-identical jobs and assuming a common due date. They also hybridized the proposed algorithm with genetic algorithms to effectively solve the scheduling problem under study. Parsa, Karimi and Moattar-Husseini (2017) formulated the same research problem as a mixed integer linear programming model and developed a dynamic programming algorithm for a given set of batches. They developed several heuristics in which the dynamic programming algorithm is utilized. The authors also proposed a branch and bound algorithm which is able to solve instances with up to 20 jobs. Cheng, Leung and Li (2017) concluded that an algorithm exists to solve a minimum cost scheduling problem on a single batch processing machine with jobs having identical size in polynomial time. However, the typical problem with due date related objective same processing time is strongly NP-hard.

3. Problem Definition

BPM-On-time scheduling problem is mainly motivated by scheduling of a single cutting machine in furniture manufacturing. There exists a set of customer orders, $\{1, \ldots, m\}$. Each order i consists of a set of products, $\{1, \ldots, n\}$, and each product j is made of a set of components. The problem is bounded since number of batches, b, is less than the number of products in the system, n. It is given that the total number of components of product j for customer i is K_{ij} including the order amounts. A batch may contain any product j of any customer i, however, number of components, K_{ij}, must be cut in the same batch and cannot be partitioned into different batches because of technological requirements. Hence, K_{ij} is considered as size of product j of customer i and this situation results in non-identical job sizes. Capacity of each batch is limited to B in terms of number of components included and process time of the cutting machine, P, is fixed for all batches. The machine can cut out a set of components (K_{ij}) from a board of raw material simultaneously. Therefore, all components included in the same batch are completed at the same time. Customer order i, only can be completed when all the products in this order are completed. This condition causes the completion time of an order i, C_i, to be the latest completion time among the batches which contains at least one product of customer *i*. There is an agreed due date, D_i, for each customer. If the completion time, C_i, of customer order *i* excesses the due date, D_i, then tardiness, T_i, occurs for customer *i*. On the other hand, completion time C_i, which is less than the due date, results in earliness, E_i. Objective of the problem is to minimize the sum of the weighted early and tardy completion of customer orders to provide customer satisfaction by completing their orders just in time and to avoid both work-in-process inventory and product stocks. To achieve this objective, an optimal partitioning of products into batches and sequencing the resulting batches throughout the planning horizon is required. Assumptions for the problem are that all data is deterministic and known a priori and a batch cannot be interrupted once the machine starts the processing on it. As a summary, BPM-On-time scheduling problem can be represented as $1|p\text{-batch}, b < n, s_i < B, D_i, \mathrm{P}|\sum(\alpha_i E_i + \beta_i T_i)$ following the notation introduced by Graham, Lawler, Lenstra and Rinnooy-Kan (1979). Abbreviation p-batch introduced by Brucker (2007) defines a parallel batch which consists of a set of jobs that can be processed on the same machine simultaneously. Notation $b < n$ means that the problem is bounded, $s_i < B$ corresponds to non-identical job sizes and limited batch capacity, D_i stands for different due dates for each customer *i*, and *P* shows fixed process time. The parameters and variables which are related with time (e.i., P, D_i, w_b, T_{ijb}, C_i, l_i, e_i) are assumed in the same time unit.

Indices:

i index of customer order ($i = 1, ..., m$)

j index of products ordered by customer i ($j = 1, ..., n$)

b index of batches ($b = 1, ..., nb$)

s index of batches ($s = 1, ..., nb$)

Parameters:

α_i tardiness penalty for customer order *i* for each planning period

β_i earliness penalty for customer order *i* for each planning period

B fixed capacity for each batch

P process time of each batch

D_i due date of customer *i*

K_{ij} total number of components of product *j* in customer order *i*

m number of customer orders

n number of products

nb number of batches

M sufficiently large constant value

Variables:

w_b start time of batch *b*

T_{ijb} completion time of j^{th} product of i^{th} customer order in batch *b*

C_i completion time of i^{th} customer order

$$y_{ijb} = \begin{cases} 1 \text{ if } j^{\text{th}} \text{ product of } i^{\text{th}} \text{ customer order in batch } b \\ 0 \text{ otherwise} \end{cases}$$

$$z_{bs} = \begin{cases} 1 \text{ if batch } b \text{ precedes batch } s \\ 0 \text{ otherwise} \end{cases}$$

L_i = Duration tardiness of customer order *i*

E_i = Duration earliness of customer order *i*

3.1. The Nonlinear Integer Programming Model

BPM-On-time scheduling problem addressed in this study is, firstly, formulated as a nonlinear integer model, Model-NL, which is given by Equations (1-10). Objective function in Equation (1) minimizes the total of weighted tardiness and earliness of the customer orders. Equation (2) implies that product *j* of order *i* can be processed in only one batch. Total number of components included in a batch cannot exceed the batch capacity, *B*, as provided by Equation (3). Equations (4-5) stipulate sequencing of the batches considering the process time, *P*. Both Equations (6-7) are used to formulize completion time of *j*th product of *i*th order in *b*th batch. Equation (8) gives the completion time of order *i* which is the longest among the completion times of the batches in which at least one product of this order is included. Tardiness time and earliness time of an order *i* are calculated in Equations (9-10), respectively, in their relation with the due date, D_i, and the completion time of order *i*, C_i. Since both the tardiness, L_i, and earliness, E_i, are defined as positive variables (Equation 11), only one of them can take a positive value according to the Equations (9-10).

Model-NL:

$$Min\ \sum_i^m \propto_i E_i + \sum_i^m \beta_i L_i \tag{1}$$

Subject to:

$$\sum_{b=1}^{nb} y_{ijb} = 1 \qquad \forall i,j \tag{2}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} K_{ij}\, y_{ijb} \le B \qquad \forall b \tag{3}$$

$$(w_b - w_s) + Mz_{bs} \ge P \qquad \forall b,s \tag{4}$$

$$(w_s - w_b) + M(1 - z_{bs}) \ge P \qquad \forall b,s \tag{5}$$

$$T_{ijb} \le My_{ijb} \qquad \forall i,j,b \tag{6}$$

$$T_{ijb} \ge w_b + P + M(y_{ijb} - 1) \qquad \forall i,j,b \tag{7}$$

$$C_i = max\ (T_{ijb}) \qquad \forall i,j,b \tag{8}$$

$$C_i - D_i \le L_i \qquad \forall i \tag{9}$$

$$D_i - C_i \le E_i \qquad \forall i \tag{10}$$

$$\begin{array}{ll} y_{ijb} \in (0,1) & \forall i,j,b \\ z_{bs} \in (0,1) & \forall b,s \\ L_i, E_i, C_i \ge 0 & \forall i \\ w_b \ge 0 & \forall b \\ T_{ijb} \ge 0 & \forall i,j,b \end{array} \tag{11}$$

3.2. The Piecewise-Linear Integer Programming Model

To speed up Model-NL, it is linearized using the upper bound technique (Paschos & Quadri, 2010). For this purpose, the single nonlinear function of Model-NL which is given in Equation (8) is replaced with Equations (19-

20). Equation (19) stipulates that completion time of order *i*, C_i, can be at least equal to completion time of the latest batch (that is, the batch having the longest completion time, T_{ijb}) which involves a product or products of this order. On the other hand, Equation (20) guarantees that C_i cannot exceed the maximum T_{ijb}. By the way, optimization procedure sets C_i to the maximum T_{ijb}. The complete model of Model-PL, is given by Equations (12 to 23).

Model-PL:

$$Min\ \sum_i^m \propto_i E_i + \sum_i^m \beta_i L_i \tag{12}$$

Subject to

$$\sum_{b=1}^{nb} y_{ijb} = 1 \qquad \forall i,j \tag{13}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} K_{ij}\, y_{ijb} \le B \qquad \forall b \tag{14}$$

$$(w_b - w_s) + M z_{bs} \ge P \qquad \forall b,s \tag{15}$$

$$(w_s - w_b) + M(1 - z_{bs}) \ge P \qquad \forall b,s \tag{16}$$

$$T_{ijb} \le M y_{ijb} \qquad \forall i,j,b \tag{17}$$

$$T_{ijb} \ge w_b + P + M(y_{ijb} - 1) \qquad \forall i,j,b \tag{18}$$

$$C_i \ge T_{ijb} \qquad \forall i,j,b \tag{19}$$

$$C_i \le T_{ijb} + M(1 - y_{ijb}) \qquad \forall i,j,b \tag{20}$$

$$C_i - D_i \le L_i \qquad \forall i \tag{21}$$

$$D_i - C_i \le E_i \qquad \forall i \tag{22}$$

$$\begin{array}{ll} y_{ijb} \in (0,1) & \forall i,j,b \\ z_{bs} \in (0,1) & \forall b,s \\ L_i, E_i, C_i \ge 0 & \forall i \\ w_b \ge 0 & \forall b \\ T_{ijb} \ge 0 & \forall i,j,b \end{array} \tag{23}$$

3.3. The Alternative Integer Linear Model

Our pre-experiments showed that Model-PL fails to find optimum solutions within reasonable amount of time as the problem size gets larger. To be able to solve larger sized instances, BPM-On-time scheduling problem was re-formulated by fixing the sequence of batches in advance. Fixing of the batch sequence allows us to drop w_b (variable of start time for batch *b*), z_{bs} (binary variable of successive batches *b* and *s*) and T_{ijb} (completion time of batch b which includes product *j* of customer *i*) from the original nonlinear model. In the alternative formulation, the start time is taken as a parameter and computed by $w_{b+1} = w_b + P$ for $b = 1, ..., nb – 1$ by setting $w_1 = 0$ a priori and thereby, a new batch is taken into process after each P hours. By the way, any customer order *i* processed between time duration [$w_{b\text{-}1}$, w$_b$] is considered to be included in *b*th batch. Therefore, this alternative formulation of

BPM-On-time scheduling problem, called Model-A, involves the decisions of assigning products of customer orders to the batches (i.e., time intervals) while satisfying the batch capacity constraint and assuring that the total number of components of a product is included in the same batch (i.e., components of a product cannot be splitted).

Model-A is given by Equations (24-33). While Equations (24, 25, 26, 31, and 32) are the common constraints with Model-NL, in addition to dropping of variables w_b and z_{bs} from Model-NL, another distinction in Model-A is related with the formulation of completion time, C_i. Binary variable S_{ib} is introduced to find which batch is completed latest including any product of order *i*. Equation (27) guarantees that C_i corresponds to the completion time of exactly one batch. Equation (28) gives that the lower bound on C_i is the maximum one among the completion times of batches in which any product of order *i* is processed. Equation (29) is introduced to stipulate C_i equals its upper bound. Equation (30) means that S_{ib} must be zero unless batch *b* contains any product of order *i*.

Model-A:

$$Min\ \sum_i^m \propto_i E_i + \sum_i^m \beta_i L_i \tag{24}$$

Subject to

$$\sum_{b=1}^{nb} y_{ijb} = 1 \qquad \forall i,j \tag{25}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} K_{ij}\, y_{ijb} \le B \qquad \forall b \tag{26}$$

$$\sum_{b=1}^{nb} S_{ib} = 1 \qquad \forall i \tag{27}$$

$$C_i \ \ge\ w_b + P + M\big(y_{ijb} - 1\big) \qquad \forall i,j,b \tag{28}$$

$$C_i = \sum_{b=1}^{nb} (w_b + P) S_{ib} \qquad \forall i \tag{29}$$

$$S_{ib} \le \sum_{j=1}^{n} y_{ijb} \qquad \forall i,b \tag{30}$$

$$C_i - D_i \le L_i \qquad \forall i \tag{31}$$

$$D_i - C_i \le E_i \qquad \forall i \tag{32}$$

$$\begin{array}{ll} y_{ijb} \ \in (0,1) & \forall i,j,b \\ S_{ib} \ \in (0,1) & \forall i,b \\ L_i, E_i, C_i \ge 0 & \forall i \end{array} \tag{33}$$

4. Experimental Study

In this section performances of the models developed in the study are tested on test instances with varying sizes. This test data is generated by scaling down real life cases. The aim is to find out capability of Model-A in solving larger instances compared to Model-NL and Model-PL. The next subsection explains generation of the test instances and then computational results are given in subsection 4.2.

4.1. Generation of Test Instances

A real life furniture manufacturing system was examined in order to generate test instances. In the system, there are approximately 250 working days a year. Each day, a batch of components of one or more products must be cut on a single cutting machine. Each batch has a fixed capacity of 3000 components of the product. The number of customer orders changes between 70 and 250 daily. To generate test instances, this manufacturing environment was scaled down by 1/50. The number of batches should be at least 5 according to the scaled data. The run times of Model-NL, Model-PL, and Model-A were limited to 120 minutes in order to sustain effective planning time.

The test instances are generated by changing some parameters systematically and selecting other parameters randomly as given in Table 1. The only fixed parameters are process time, P = 10-hour and batch capacity, B = 60 components.

Systematically changed parameters	Randomly selected parameters
• The number of customer orders, m = (from 2 to 10)	• Number of components, K_{ij}, for product j of each order i = (the number of product component types from the bill of materials of product j) × (*Uniform*(0,5))
• The number of product types, n = (from 1 to 14)	• Earliness and tardiness penalties of each order i: α_i = *Uniform*(0, 10) β_i = *Uniform*(0, 10)
	• Due dates are selected from the set $\{P, 2P, \ldots, \gamma P\}$ where γ = 6 for small or moderate sized problems, γ = 10 for larger problems

Table 1. Parameter generation for the test instances

A total of 87 instances are generated. In Appendix A, the test instances are named using number of customers (m), number of products (n), number of batches (nb) and total number of parts ($\sum K$). For example, $m2$ $n3$ $nb2$ $K49$ stands for an instance with 2 customer orders, 3 products, 2 batches and 49 total parts.

4.2. Computational Results

In this section Model-NL, Model-PL, and Model-A are run to solve the 87 test instances. All the models are run using LINGO v15.0 and are terminated if the computer time exceeds 120 minutes. The optimum cost of earliness and tardiness and the computer time to solve the instances using three models are given in detail in Table A1 of Appendix. Table 2 is a summary of Table A1. As shown in Table 2, only 19 instances out of the 87 are solved optimally using Model-NL, the remaining 68 test instances could not be solved because of the run time limit of 120 min. As the problem size gets larger in terms of m, n, and K, Model-NL performs quite poorly. Model-PL starts to perform worse when $m \geq 9$ and $\sum K \geq 200$, however there are several smaller sized instances which Model-PL cannot solve within the allowed time limit. Total number of instances which are solved by Model-PL is 38. On the other hand, Model-A is able to find optimum solutions quite efficiently for all of the test instances. As seen from Table 2, the average run time of Model-A is quite shorter than the corresponding average run times over the same instances that are solved by Model-NL and Model-PL. Additionally, Table 3 gives the number instances which are solved by the models with respect to the total number of components. According to the results in Table 3, Model-NL is able to solve the instances up to only 200-component whereas Model-PL can solve the instances with at most 350-components. On the other hand, Model-A is capable of optimizing all the instances regardless of the component size. Figures 1-a, 1-b and 1-c show the number of instances solved by each model with respect to the average numbers of products, costumer orders, and components, respectively to figure out the performance of Model-A further. It is clear that, Model-A strongly outperforms the other two models considering the three problem parameters. These results confirm that Model-A is the best performing model among the three models. Consequently, we decided to apply Model-A to solve real-world problems.

Performance indicator	Model-NL	Model-PL	Model-A
Number of solved instances	24	38	87
Average run time in seconds over instances solved by all three models	49.6	0	0
Average run time in seconds over instances solved by Model-PL and Model-A	–	10.6	0
Average run time in seconds over instances solved by only Model-A	–	–	32.4

Table 2. Comparison of Model-NL, Model-PL, and Model-A

Range for total number of components	Total number instances	Number of solved intances		
		Model-NL	Model-PL	Model-A
0-50	1	1	1	1
50-100	14	11	11	14
100-150	19	9	12	19
150-200	17	3	10	17
200-300	10	0	3	10
300-500	7	0	1	7
500-700	11	0	0	11
700-1000	8	0	0	8

Table 3. Comparison of Model-NL, Model-PL, and Model-A respect to the number of components

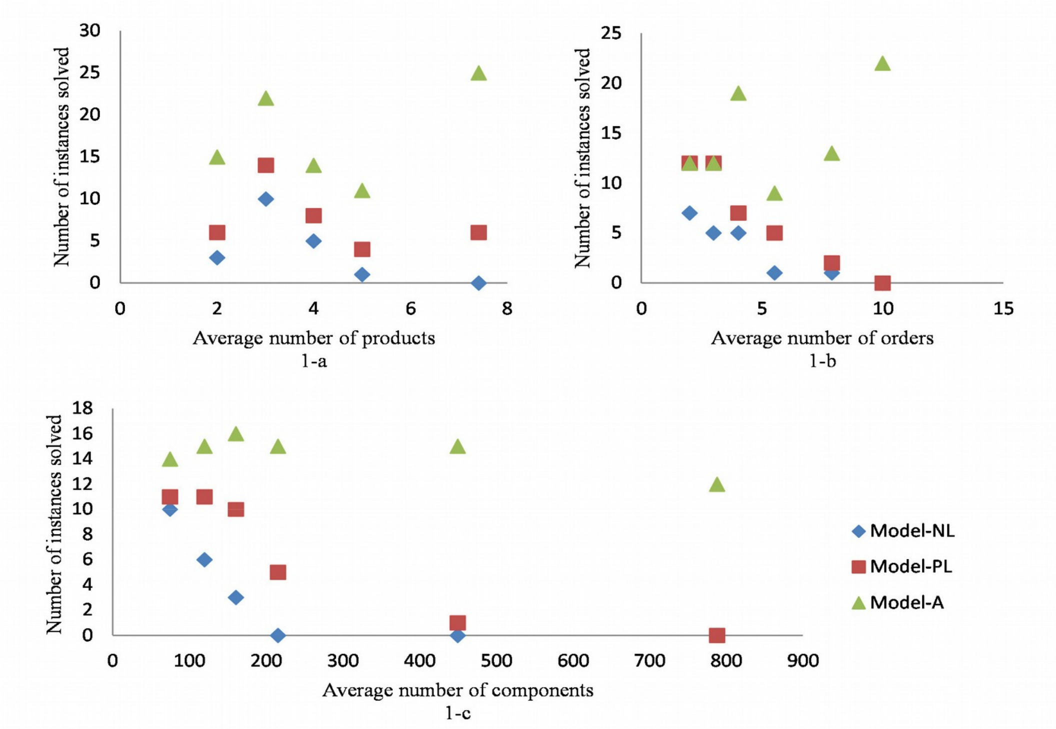

Figure 1. Comparison of Model-NL, Model-PL, and Model-A according to the varying ranges of number of products (1-a), number of orders (1-b) and number of components (1-c)

5. Case Study

The case study is from one of the biggest modular bathroom furniture production factories in Turkiye. The company produces for both local and export markets. Products are made up of thermofoil, melamine and lacquer coated medium density fiber board (MDF) and coloring is done according to customer choices. Because of high customized nature of the business, it is troublesome to keep products in the stock and therefore make-to-order production strategy is implemented. Production process starts with the arrival of customer orders to factory. On the other hand customers are very sensitive to service level of delivery and orders must be delivered just in time. Partial deliveries are not accepted by the customer since products are complementary to each other. The factory uses pull production system and production process is designed based on lean manufacturing philosophy.

Manufacturing process in the factory flows through in order of "panel cutting – edge bending – assembly line" and finally ends in warehouse. Raw material is a sheet of melamine coated MDF board. Cutting machine receives the raw material and cuts a batch of the components of one or more products. The number of components to be cut out depends on the assigned product types to the batch and order quantities of products. Capacity of a batch is constrained by the total capacity of one day process which is 3000 components. Once components are cut, in other words, a batch is processed, edge banding process is done via a roller conveyor. Through the edge banding process, parts take drilling operation and products are assembled in pre-determined times. After the assembly, products are stored in warehouse and are awaited until all products in the same order are produced. Under these circumstances, the management targets to minimize products awaited in stock and also to provide deliveries of the completed orders to customers on time. As a result, we focused on solving the scheduling of the panel cutting process since the main factors that affect the managerial targets arise from this process.

Customer orders arrive the manufacturing system continuously and BPM-On-time scheduling problem should be solved each day by taking into account all new orders received and the orders which have not been processed yet. The objective is to reduce total penalty cost of early and late completion of customer orders. To achieve this, all the products must be scheduled such that they will be completed as much as close to the promised due dates to customers.

In this section, only Model-A is used to solve BPM-On-time scheduling problem of the panel cutting machine, since the other models have already failed to solve test instances in Section 4.2. We obtain a set of real-case instances of the problem by extracting data from different five successive days of the system since pre-experiments showed that Model-A is able to solve the real-world problem for five successive days. The cases includes varying number of customer orders consisting of different number of products in different quantities. Size of the customer-product- component triple data gets huge as the planning horizon increases. To show the performance of Model-A in terms of total number of components, $\sum K$, the number of orders, m, the number of products, n, and the number of batches, nb, a problem size coefficient is proposed as given in Equation (34). Results obtained by Model-A for the eleven case instances drawn from the system are given in Table 4. Figure 2 also shows how the performance of Model-A changes with respect to P_S coefficient. Naturally, run time requirement of the model increases as the coefficient rises. However, the results in Table 4 implie that Model-A is able to solve the cases which involve up to 8342 total number of components within approximately 6 hours. It is clear that all the case instances but the three (cases 1, 3, and 4) have an optimum solution which makes the completion of orders just in due dates promised to customers. Although the problem under study is one of NP-hard problems, this study encourages the practitioners to solve this planning problem optimally up to 9000 components in total using Model-A.

$$\mathrm{Ps} = \frac{\mathrm{n}(\sum \mathrm{K})\mathrm{nb}}{\mathrm{m}} \tag{34}$$

Instances selected from the real life	Number of customer orders, *m*	Number of products, *n*	Number of batches, *nb*	Total number of components, $\sum K$	P_S	Run time (min.)	Optimum $\sum(\alpha_i E_i + \beta_i T_i)$
Case 1	65	97	50	2631	88152	2.17	620
Case 2	80	122	50	2950	96721	16.70	0
Case 3	50	120	53	4351	96085	7.30	29
Case 4	89	163	65	6180	219333	54.61	124
Case 5	61	96	65	2265	93549	15.30	0
Case 6	53	79	65	2055	89614	4.38	0
Case 7	51	99	65	2162	72394	11.59	0
Case 8	81	146	65	5306	191343	70.95	0
Case 9	55	136	65	4137	108748	37.24	0
Case 10	140	229	65	8342	331494	373.76	0
Case 11	108	183	53	6865	214728	60.67	0

Table 4. Perfomance of Model-A on the real-world cases

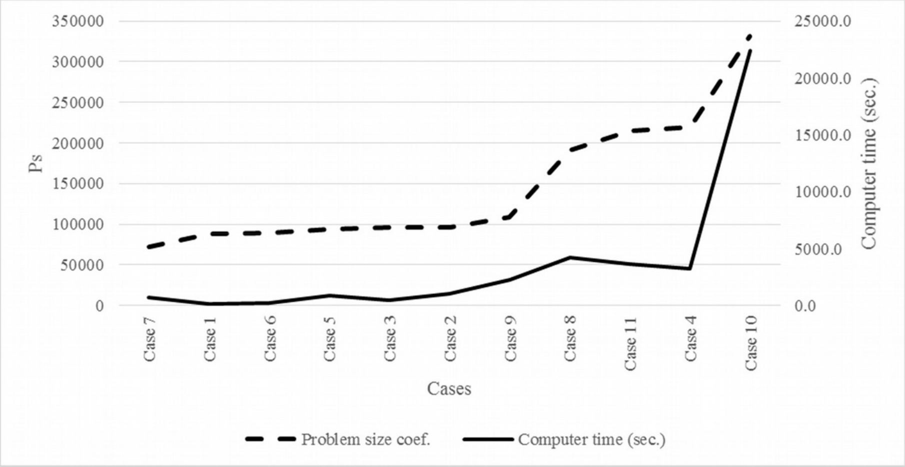

Figure 2. Run time performance of Model-A respect to P_S coefficient

6. Conclusion

In this study, BPM-On-time scheduling problem has been considered. The problem aims to minimize the weighted sum of early and tardy completion times of customer orders in order to provide both customer satisfaction and cost-effective production schedules in terms of reduced work-in-process and final product inventories. To achieve this objective, products from different customer orders are required to be clustered in batches and these batches must be scheduled on a single batch processing machine optimally. BPM-On-time scheduling problem differs from the literature with the addition of new constraints: Different products which belong to different customer orders can be processed simultaneously in the same batch. However, completion of the customer order is the completion of the latest batch which includes at least one product of this customer. Also, all components of the same product must be contained in the same batch. We examined the ways of solving this variant of batch scheduling problem developing three mathematical models. The two of the models, namely, Model-NL and Model-PL failed to solve even small sized instances of the problem. The last model, Model-A, however, has been found capable of solving all the test instances. The case study carried out also showed that Model-A can be used to solve real-world problems up to 9000 components.

For further research, we are planning to develop heuristic approaches to BPM-On-time problem to deal with very large sized instances considering longer planning horizons.

Declaration of Conflicting Interests

It is declared that there is no potential conflict of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research with the number FEN-C-DRP-110315-0062 is supported by Marmara University BAPKO foundation.

References

Albers, S., & Brucker, P. (1993). The complexity of one-machine batching problems. *Discrete Applied Mathematics*, 47(2), 87-107. https://doi.org/10.1016/0166-218X(93)90085-3

Brucker, P. (2007). *Scheduling algorithms*. (5th ed.). Berlin and New York: Springer.

Brucker, P., Gladky, A., Hoogeveen, H., Kovalyov, M.Y., Potts, C.N., Tautenhahn, T., & van de Velde, S.L. (1998). Scheduling a batching machine. *Journal of Scheduling*, 1(1), 31-54. https://doi.org/10.1002/(SICI)1099-1425(199806)1:1<31::AID-JOS4>3.0.CO;2-R

Cheng, T.C.E., & Kahlbacher, H.G. (1993). Scheduling with delivery and earliness penalties. *Asia-Pacific Journal of Operational Research*, 10, 145-152.

Cheng, B.Y., Leung, J.Y., & Li, K. (2017). Integrated scheduling on a batch machine to minimize production, inventory and distribution costs. *European Journal of Operational Research,* 258, 104-112. https://doi.org/10.1016/j.ejor.2016.09.009

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. In Hammer, P.L., Johnson, E.L., & Korte, B.H. (Eds.), *Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver* (287-326). Volume 5 of *Annals of Discrete Mathematics*, 5, 287-326. Elsevier Science. https://doi.org/10.1016/S0167-5060(08)70356-X

Hall, N.G., & Posner, M.E. (1991). Earliness-tardiness scheduling problems: weighted deviation of completion times about a common due date. *Operations Research*, 39(5), 836-846. https://doi.org/10.1287/opre.39.5.836

Ikura, Y., & Gimple, M. (1986). Efficient scheduling algorithms for a single batch processing machine. *Operations Research Letters*, 5(2), 61-65. https://doi.org/10.1016/0167-6377(86)90104-5

Li, Z., Chen, H., Xu, R., & Li, X. (2015). Earliness–tardiness minimization on scheduling a batch processing machine with non-identical job sizes. *Computers&Industrial Engineering*, 87, 590-599. https://doi.org/10.1016/j.cie.2015.06.008

Mathirajan, M., & Sivakumar, A.I. (2006). A literature review, classification and simple metaanalysis on scheduling of batch processors in semiconductor. *The International Journal of Advanced Manufacturing Technology*, 29(9), 990-1001. https://doi.org/10.1007/s00170-005-2585-1

Mathirajan, M., & Sivakumar, A.I. (2003). Scheduling of batch processors in semiconductor manufacturing - a review. In *Innovation in Manufacturing Systems and Technology (IMST).*

Mönch, L., Fowler, J.W., Dauzere-Peres, S., Mason, S.J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling,* 14(6), 583-599. https://doi.org/10.1007/s10951-010-0222-9

Mönch, L., & Unbehaun, R. (2007). Decomposition heuristics for minimizing earliness-tardiness on parallel burn-in ovens with a common due date. *Computers & Operations Research*, 34(11), 3380-3396.https://doi.org/10.1016/j.cor.2006.02.003

Mönch, L., Unbehaun, R., & Choung, Y.I. (2006). Minimizing earliness-tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint. *OR Spectrum*, 28(2), 77-198.https://doi.org/10.1007/s00291-005-0013-4

Parsa, N.R., Karimi, B., & Moattar-Husseini, S.M. (2017). Exact and heuristic algorithms for the just-in-time scheduling problem in a batch processing system. *Computers and Operations Research,* 80, 173-183. https://doi.org/10.1016/j.cor.2016.12.001

Paschos, V.Th., & Quadri, D. (2010). *Paradigms of Combinatorial Optimization: Problems and New Approaches* (pp. 246-256). https://doi.org/10.1002/9781118600207.fmatter

Potts, C.N., & Kovalyov, M.Y. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120(2), 228-249. https://doi.org/10.1016/S0377-2217(99)00153-8

Potts, C.N., & van Wassenhove, L.N. (1992). Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *Journal of the Operational Research Society*, 43(5), 395-406. https://doi.org/10.1057/jors.1992.66

Qi, X., & Tu, F. (1999). Earliness and tardiness scheduling problems on a batch processor. *Discrete Applied Mathematics,* 98, 131-145. https://doi.org/10.1016/S0166-218X(99)00113-4

Webster, S., & Baker, K.R. (1995). Scheduling groups of jobs on a single machine. *Operations Research*, 43(4), 692-703. https://doi.org/10.1287/opre.43.4.692

Yılmaz O.F., & Durmusoglu, M.B. (2017). A performance comparison and evaluation of metaheuristics for a batch scheduling problem in a multi-hybrid cell manufacturing system with skilled workforce assignment. *Journal of Industrial & Management Optimization.* https://doi.org/10.3934/jimo.2018007

Xiao, Z., & Shao, H. (2013). Two-Dimensional Scheduling: A Review. *Research Journal of Applied Sciences, Engineering and Technology,* 6(9), 1566-1572. https://doi.org/10.19026/rjaset.6.3870

Zhao, H., Hu, F. & Li, G. (2006). Batch Scheduling with a Common Due Window on a Single Machine. In Wang, L., Jiao, L., Shi, G., Li, X., & Liu, J. (Eds.), *Fuzzy Systems and Knowledge Discovery. Lecture Notes in Computer Science* (4223). Berlin, Heidelberg: Springer. https://doi.org/10.1007/11881599_76

Appendix

Test Instance	Optimum $\sum(\alpha_i E_i + \beta_i T_i)$	Model-NL	Model-PL	Model-A
		Run time (sec.)	Run time (sec.)	Run time (sec.)
m2 n3 nb2 K49	10	0	0	0
m2 n3 nb2 K61	20	9	0	0
m2 n3 nb2 K82	0	0	0	0
m2 n3 nb3 K69	0	0	0	0
m2 n4 nb3 K77	0	3	0	0
m2 n4 nb4 K100	0	0	0	0
m2 n5 nb4 K112	40	270	0	0
m2 n5 nb5 K190	80	NA*	0	0
m2 n6 nb4 K143	160	NA	1	0
m2 n7 nb4 K178	30	NA	0	0
m2 n7 nb4 K225	10	NA	2	0
m2 n8 nb4 K188	100	NA	156	0
m3 n2 nb2 K63	0	0	0	0
m3 n3 nb3 K106	0	0	0	0
m3 n3 nb3 K112	0	0	0	0
m3 n3 nb4 K134	0	NA	0	0
m3 n3 nb4 K165	50	NA	13	0
m3 n3 nb4 K90	0	NA	0	0
m3 n4 nb3 K127	100	NA	0	0
m3 n4 nb3 K145	0	0	0	0
m3 n4 nb3 K164	0	77	0	0
m3 n4 nb4 K79	0	NA	0	0
m3 n5 nb5 K148	0	NA	0	0
m3 n6 nb5 K236	0	NA	198	0
m4 n3 nb6 K169	0	323	0	0
m4 n4 nb4 K184	240	NA	12	0
m4 n4 nb5 K85	0	364	NA	0
m4 n4 nb6 K155	0	NA	0	0
m4 n5 nb4 K200	180	NA	NA	0
m4 n5 nb5 K230	0	NA	2	0
m4 n5 nb7 K144	20	NA	NA	0
m4 n5 nb7 K230	0	NA	NA	0
m4 n6 nb5 K199	0	NA	NA	0
m4 n6 nb6 K142	0	NA	NA	0
m4 n6 nb6 K187	20	NA	NA	0
m4 n6 nb6 K303	50	NA	626	0

Test Instance	Optimum $\sum(\alpha_i E_i + \beta_i T_i)$	Model-NL	Model-PL	Model-A
		Run time (sec.)	Run time (sec.)	Run time (sec.)
m4 n6 nb8 K433	190	NA	NA	0
m4 n7 nb5 K137	0	NA	NA	0
m4 n7 nb5 K261	60	NA	NA	0
m5 n3 nb4 K167	70	NA	1	0
m5 n3 nb5 K169	0	NA	NA	0
m5 n3 nb6 K101	0	NA	0	0
m5 n4 nb5 K146	0	NA	NA	0
m6 n2 nb2 K62	0	NA	0	0
m6 n2 nb4 K180	190	NA	9	0
m6 n3 nb2 K105	140	31	0	0
m6 n4 nb5 K192	40	NA	NA	0
m6 n5 nb5 K290	130	NA	NA	0
m4 n2 nb5 K75	0	4	0	0
m4 n3 nb4 K189	90	NA	NA	0
m4 n3 nb4 K56	0	10	NA	0
m4 n3 nb5 K95	0	53	0	0
m7 n2 nb6 K118	0	NA	NA	0
m7 n2 nb6 K99	0	NA	0	0
m7 n5 nb7 K320	410	NA	NA	0
m9 n2 nb8 K127	0	NA	NA	0
m9 n2 nb8 K153	0	NA	NA	0
m9 n2 nb9 K208	0	NA	NA	0
m9 n2 nb9 K262	0	NA	NA	0
m9 n2 nb9 K318	0	NA	NA	0
m9 n3 nb10 K146	0	NA	NA	0
m10 n10 nb14 K789	250	NA	NA	2
m10 n11 nb13 K776	1020	NA	NA	395
m10 n12 nb14 K795	200	NA	NA	2
m10 n13 nb16 K937	510	NA	NA	12
m10 n14 nb17 K996	550	NA	NA	66
m10 n5 nb11 K607	910	NA	NA	28
m10 n5 nb11 K623	220	NA	NA	1
m7 n2 nb3 K128	0	19	0	0
m7 n2 nb3 K91	0	NA	NA	0
m7 n2 nb4 K153	0	NA	NA	0
m7 n2 nb5 K232	0	NA	NA	0
m10 n3 nb10 K549	1320	NA	NA	4
m10 n3 nb10 K564	1040	NA	NA	10

Test Instance	Optimum $\sum(\alpha_i E_i + \beta_i T_i)$	Model-NL	Model-PL	Model-A
		Run time (sec.)	Run time (sec.)	Run time (sec.)
m10 n3 nb6 K335	110	NA	NA	0
m10 n3 nb9 K485	70	NA	NA	0
m10 n4 nb10 K556	300	NA	NA	1
m10 n4 nb12 K695	890	NA	NA	9
m10 n4 nb7 K382	130	NA	NA	0
m10 n6 nb11 K620	1070	NA	NA	61
m10 n6 nb9 K529	1170	NA	NA	9
m10 n7 nb10 K571	820	NA	NA	3
m10 n7 nb13 K726	300	NA	NA	1
m10 n7 nb16 K911	2030	NA	NA	1697
m10 n7 nb17 K974	1140	NA	NA	512
m10 n8 nb9 K524	30	NA	NA	0
m10 n9 nb10 K583	490	NA	NA	7

*NA: not available since the corresponding instance couldn't be solved within a given run time

Table A1. Results obtained from Model-NL, Model-PL, and Model-A on the test instances

Journal of Industrial Engineering and Management, 2018 (www.jiem.org)