An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.14, No.1, pp.32-40 (2024)
https://doi.org/10.11121/ijocta.1435

RESEARCH ARTICLE

Bin packing problem with restricted item fragmentation: Assignment of jobs in multi-product assembly environment with overtime

Mustafa Ustuncelik, Cagri Koc*, Huseyin Tunc

Department of Business Administration, Social Sciences University of Ankara, Ankara, Turkiye
mustafa.ustuncelik@gmail.com, cagri.koc@asbu.edu.tr, huseyin.tunc@asbu.edu.tr

ARTICLE INFO

Article history:
Received: 20 July 2023
Accepted: 10 October 2023
Available Online: 3 November 2023

Keywords:
Bin packing problem
Item fragmentation
Multi product assembly jobs
Mixed integer programming

AMS Classification 2010:
90B35, 90C11

ABSTRACT

This paper studies the assignment problem of multi product assembly jobs to days. The problem aims to minimize the amount of overtime while avoiding assembly delays for jobs that can be fragmented into smaller sub-tasks. When sequence-dependent setup times are negligible, the problem considered transforms into the bin packing problem with restricted item fragmentation where jobs represent items and days stand for bins. We present a mixed integer programming model of the problem by extending earlier formulations in the literature. Computational experiments show that the mathematical model obtained optimal solutions for majority of instances tested within reasonable computation times.

1. Introduction

This study is inspired from a practical production scheduling problem encountered in a construction machinery manufacturing company. The company deals with intricate product assemblies including multiple sub-assembly groups and multi-level product trees. The intricate nature of these product structures, characterized by a multitude of sub-assembly groups and their corresponding multi-level trees, presents a significant challenge in effectively planning and managing material resources. Furthermore, the characteristics of the work centers vary depending on whether job setup times are dependent on the job sequence.

In operational research literature, scheduling and assignment problems are extensively studied [1-3]. When the objective is to determine only the production day for each job, without considering sequence dependent setup times, the problem aligns with the bin packing problem (BPP). Using this insight, we tackled the complex real-world problem using a BPP-based solution approach.

The BPP is one of the most extensively studied combinatorial optimization problems in the literature. The BPP, which is known to be NP-hard, involves packing a set of items into the minimum number of bins of fixed size where each item has a known size and each bin has a known capacity. The BPP has many practical applications, such as packing items in a warehouse or shipping containers, assigning tasks on a set of machines and allocating resources in cloud computing. Furthermore, the BPP has theoretical implications, as it is related to other problems such as the knapsack problem, the cutting stock problem, and the vehicle routing problem [4].

The BPP has been subject to a detailed scrutiny for several decades, resulting in various solution approaches to solve the problem under different objectives and constraints. A variety of heuristics have been developed to find good-quality solutions in a reasonable amount of time. These methods include heuristics such as first-fit, next-fit, best-fit and worst-fit, and metaheuristics such as genetic algorithms, simulated annealing and tabu search. Despite extensive research on the issue, there are still many remaining challenges due to its practical importance, making it a currently active topic in optimization.

The BPP and its variants have been extensively studied in the literature. Table 1 provides a summary of various BPP studies. The table categorizes these studies according to their objectives, constraints, and solution methods employed. This overview offers a

*Corresponding author

Table 1. Summary of the literature on the BPP.

Study	Objective							Constraints					Solution Method
	B	F	C	N	L	M	O	C	F	B	K	I	
Eilon and Christofides [5]	+										+		Zero-one programming mode, heuristic
Jansen [6]	+							+					Asymptotic approximation
Loh et al. [7]					+					+			Weight annealing
Khanafer et al. [8]	+							+					Dual-feasible functions Data-dependent dual-feasible functions
Crainic et al. [9]	+		+								+		Heuristic, lower bounds
Elhedhli et al. [10]	+							+					Branch-and-price algorithm
Fleszar and Charalambous [11]	+										+		Heuristic
Khanafer et al. [21]				+						+			Column-generation methods, Heuristic, tabu search
Casazza and Ceselli [14]		+								+			Exact algorithms, heuristics
Dokeroglu and Cosar [17]			+							+			Genetic algorithm
LeCun et al. [15]		+								+			Approximation algorithms
Arbib and Marinelli [20]						+				+			MIP
Byholm and Porres [16]		+								+			Approximation and metaheuristic algorithms
Casazza [18]			+						+		+		Branch-and-price algorithm
Bertazzi et al. [12]	+								+				Worst-case analysis
Ekici [13]	+							+	+				Heuristic
Ekici [19]			+					+	+		+		Heuristic, lower bounds
This study							**+**			**+**	**+**	**+**	MIP

Objectives: B: number of bins, F: number of fragmentations, C: cost, N: number of conflicts, L: load of bin, M: makespan, O: over capacity usage.
Constraints: C: conflict items, F: fragmentable items, B: number of bins, K: variable bin capacity, I: item-bin conflict.

comprehensive, yet not exhaustive, view of the diversity in approaches and strategies utilized within the BPP.

One of the primary objectives of the BPP is to minimize the number of bins used. Minimizing the number of bins leads to better space utilization, further cost savings, improved operational efficiency and enhanced sustainability. Apart from this, the studies also consider minimizing the total cost of packing items, the makespan and over capacity usage as well.

The constraints have a significant impact on the combinatorial nature of the problem. In BPPs, items that should not be placed in the same bin are referred to as conflicted items. This constraint is particularly critical in real-life scenarios such as chemical material transportation or storage. Several papers focused on conflicted items [6, 8]. These studies developed several mathematical models and metaheuristics to provide solutions while ensuring that conflicted items are assigned to different bins [10, 13, 19]. In BPPs, bins may have a fixed or variable capacity each may have different costs associated with its usage. In this regard, several studies used heterogenous bins [5, 9, 11, 18, 19].

Loh et al. [7] studied the one-dimensional BPP and proposed a weight annealing heuristic. Khanafer et al. [8] developed lower bounds for the BPP with conflicts. Crainic et al. [9] considered the variable cost and size BPP and developed lower bounds and heuristics. Elhedhli et al. [10] proposed a branch-and-price algorithm for the BPP with conflicts. Fleszar and Charalambous [11] developed bin-oriented heuristics for one-dimensional BPP by packing one bin at a time. Khanafer et al. [21] studied the min-conflict packing problem as well as the bi-objective version of the problem. Dokeroglu and Cosar [17] proposed a set of robust and scalable hybrid parallel algorithms to solve the BPP. Arbib and Marinelli [20] addressed the one-dimensional BPP to minimize completion time and lateness.

The BPP with item fragmentation (BPPIF) involves efficiently packing items into a limited number of bins where items can be fragmented into multiple bins. The BPP objective is to minimize the total number of bins used. Unlike the classic BPP, where items are treated as indivisible units, this variation enables better fitting within the bins as it allows items to be divided into smaller fragments. Given a collection of items, each with a specific size and a known fragmentation capacity, and a set of bins with fixed capacities, the BPPIF entails determining the optimal packing configuration that minimizes the number of bins required while adhering to the item fragmentation constraints.

When all items are fragmentable with no additional consideration, the BPPIF transforms into a linear optimization problem [13]. However, this usually does not reflect the reality and there often exists special considerations or constraints regarding fragmentation. In real-life scenarios, item fragmentation may lead to additional costs for subsequent operations. In the literature, several studies focused on the BPPIF aiming to reduce the overall assignment cost [18, 19]. There are also studies that aim to minimize the number of fragmentations [12-16]. This objective is particularly important for the cases where item fragmentation incurs a cost [14]. Those studies often consider a given number of identical bins.

Bertazzi et al. [12] investigated a few special splitting policies and developed the worst-case performance bounds. Ekici [13] introduced mathematical models and a column generation-based heuristic for the BPPIF. Casazza and Ceselli [14] studied the BPPIF and developed mathematical formulation and algorithms and greedy heuristics. LeCun et al. [15] investigated the complexity of the problem considered and proposed a constant factor approximation algorithm. Byholm and Porres [16] suggested performance improving operators for the heuristic algorithms introduced earlier in the literature. Casazza [18] considered BPPIF with heterogenous bins and developed a branch-and-price algorithm. Furthermore, due to the characteristics of the items to be packed, some items may not be fragmentable during the packing process. In this respect, Ekici [13] and [19] investigated the BPPIF with conflicted items. Here, one cannot pack fragments of conflicted items into the same bin. The former assumes the identical bins, whereas the latter considers variable sized bins.

This study investigates the BPPIF with additional side constraints so as to analyze the production scheduling problem with sequence independent setup times considered in a multi-product assembly environment. Here, items correspond to jobs and bins represent days in the production scheduling setting. Our contribution is two-fold. First, we consider the additional side constraints, namely item-bin conflict and restricted item fragmentation. The former refers that items (jobs) cannot be packed into any given bin (day). The latter, on the other hand, represents an item can be fragmented at most once, and the fragmented parts must be packed into the bins representing two subsequent days. Second, we extended the mathematical formulation in the literature to consider the aforementioned side constraints and evaluated its performance in a real-life inspired extensive numerical study.

The rest of the paper is organized as follows. In Section 2, we define the problem. We introduce the mathematical model used to solve the BPPIF in Section 3. We present the results of the computational study in Section 4. Concluding remarks are provided in Section 5.

2. Problem defitinion

The problem is inspired from the production scheduling problem in a company that manufactures and assembles industrial machines. The main body of the machine, consisting of steel plates, is manufactured internally by the company, whereas the remaining materials are procured from multiple external suppliers. The machine parts manufactured by the company undergo a series of process including cutting and forming, machining, welding, painting, and assembly. Specifically, the items that require welding have a sub-assembly product tree structure that can reach up to level seven.

In the context of products comprised of multiple subcomponents, it is critical to ensure that all manufacturing processes for the subcomponents are completed prior to initiating the assembly process. Failure to do so, may result in disruptions in production. Given that each product has a distinct product structure and follows a unique routing sequence, it is challenging, yet crucial, to develop a planning strategy that guarantees the timely completion of all necessary operations for each product.

The complexity of this planning problem arises from the diverse product structures and routing sequences. Each product necessitates a distinct set of operations and follows a unique sequence of steps. Creating the optimal plan to ensure all required operations are completed for each product is a major challenge.

Figure 1 illustrates a multi-level product BOM and routings for sub-assembly items. Figure 1(b) presents the work centers responsible for processing all the parts required for Sub Assembly 3 body production, according to the product tree structure depicted in Figure 1(a). All parts must be processed at different work centers with different priorities. According to the product tree structure, Item 2 and Item 3 must be processed first at the same work center in order to produce Item 6 in sub-assembly group 3. Any delay in processing Item 2 and Item 3 directly affects the completion time of sub-assembly group 3.

In the manufacturing company where this study is inspired, MRP program is used to determine the requirement dates of materials. The MRP program determines the latest requirement days of all materials by performing backward date calculations by considering the demand date of the product on a daily basis. Because all materials have pre-determined routings set in the MRP program, all jobs for each work centers are also determined on a daily basis when materials latest requirement dates are calculated.

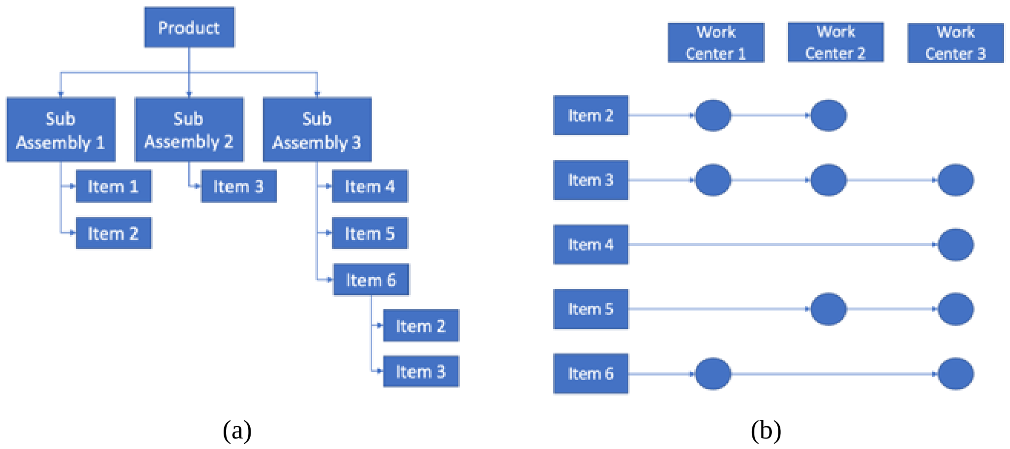

Figure 1. (a): Multi-level product BOM, (b): routings for sub-assembly items.

The MRP program does not consider the finite capacities of work centers and specific manufacturing conditions of jobs when determining the latest requirement dates. Backward calculation approach often results in unrealistic scenarios in production planning. For instance, a work center that operates only one shift per day may be loaded jobs with a total duration of two shifts.

Although knowing the deadlines of jobs is a necessity for an effective plan, it is not sufficient. In a work center, similar to the examples given earlier, there are two main challenges that can arise: exceeding the capacity of the work center and the unavailability of materials at the required time due to the involved nature of the sub-assembly products. These issues are typically addressed by resorting to overtime work.

To put it simply, when the workload surpasses the capacity of the work center or when the materials needed for production are not available, the solution often involves employing overtime work. Inevitably, this leads to additional hours of work beyond the regular schedule to compensate for the increased workload or to catch up on delayed tasks caused by material unavailability. By using overtime work strategically, these challenges can be effectively addressed and the smooth operation of the work center can be maintained.

In the company, when an overtime decision is made for a working day shift, three hours of overtime should be done as standard even if the total workload requirement is less than three hours. As such, this leads to additional costs for the company. In particular, the cost of overtime is fixed, i.e., when an overtime decision is made a fixed cost is incurred. This simply implies that it is cost-efficient to plan three hours of workload (full overtime capacity) if overtime is necessary.

In order to achieve a more balanced and cost-effective planning, it is necessary to consider the deadlines of tasks and plan them based on the capacity of the work center on a daily basis. By considering the capacity constraints, it is possible to allocate resources more efficiently, avoid excessive workloads, and ensure that materials are available when needed. This type of planning facilitates minimizing overtime work and reduces additional costs for the company.

We focus on a single work center and aim to determine the daily job assignments in there. The jobs do not have any sequence dependent setup costs, so the order in which the jobs are processed within a day does not have an impact on the total cost. Furthermore, it is allowed to leave at most one job unfinished during the day, which can be resumed in the following day. As such, this problem determines job-day assignments where the processing order of jobs within each day is not of interest. Since items can be considered as jobs, bins can be considered as days, and some jobs may not need to start and end at the same day, the resulting problem can be modelled as the BPPIF. The jobs that have to be completed within a day are referred to as non-fragmentable items. The remaining jobs are considered as fragmentable items whose processing operations must be completed at most one day after its starting day. As opposed to the common assumption adopted in the BPPIF literature, in our problem, each fragmentable job can be divided into at most two parts and fragmented parts can be packed into specific pair of bins, i.e., those referring to the subsequent days.

The objective of the problem is to minimize the total overtime while allocating jobs to variable-size bins (i.e., days at which only normal work hours are used and days at which the sum of normal work hours and overtime are used) with limited capacities. As the due dates of jobs are known, the number of days (bins) for planning is fixed. Each non-fragmentable job must be finished within one day, whereas each fragmentable job must be completed within either one day or two consecutive days. On any given day, only one job can be fragmented to the following day and the setup times between jobs are not sequence dependent. Also, we assume that processing time of a job cannot be longer then a working day capacity.

Figure 2 illustrates a feasible solution of the problem. The figure shows that since the total processing times of the jobs assigned to the first day exceed the capacity of that day, including overtime, the amount of work time that exceeds the capacity is transferred to the second day. As such, the amount of work transferred from the first day is also considered while determining the capacity need for the second day. If the remaining capacity of normal shift hours is insufficient to meet the daily scheduled requirement, overtime is considered as an additional capacity. As the setup times are not sequence-dependent, the fragmented item and hence the capacity transferred to the next day will be determined following the due dates. Put in other words, the solution does not impose a sequence within a given day.

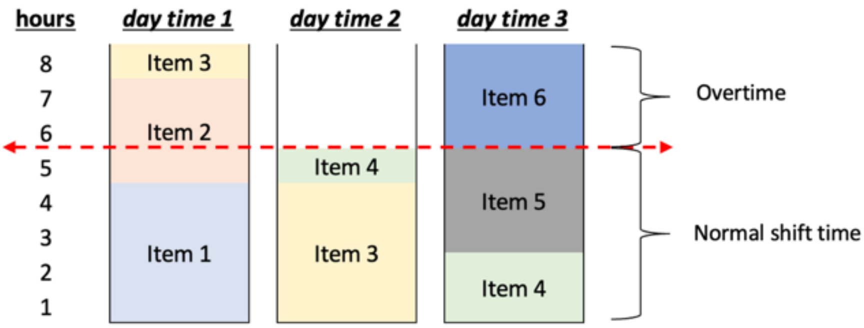

Figure 2. Illustration of a feasible solution.

3. Notation and mathematical model

We model the problem as MIP that balances the capacity utilization of the bins and minimizes the number of overtime hours required to complete all the jobs. We formulated the MIP with three binary decision variables indicating whether a job is assigned to a particular bin, whether a job is fragmented and whether an overtime decision is made or not. Furthermore, a continuous decision variable represents the fragmented part of a job that will be transferred to the following day.

The sets, parameters and decision variables are as follows.

Sets

$N = \{1, \dots, n\}$: $Job\ set$

$T = \{1, \dots, t\}$: $Day\ set$

F: $Fragmentable\ jobs\ set\ (F \in N)$

$\bar{F}$: $Non-fragmentable\ jobs\ set\ (\bar{F} \in N)$

$N = F \cup \bar{F}$

a_i: $Jobs\ that\ can\ be\ assigned\ to\ day\ i \in T$

$\bar{a}_i$: $Jobs\ that\ cannot\ be\ assigned\ to\ day\ i \in T$

A_j: $Days\ that\ job\ j \in N\ can\ be\ assigned\ to$

$\bar{A}_j$: $Days\ that\ job\ j \in N\ cannot\ be\ assigned\ to$

Parameters

P_j : $The\ processing\ time\ of\ job\ j \in N$

d: $The\ time\ for\ a\ normal\ day\ shift$

o: $Overtime\ for\ a\ working\ day$

Decision variables

$$x_{ji} = \begin{cases} 1, if\ job\ j \in N\ is\ assigned\ to\ day\ i \in T \\ 0, otherwise \end{cases}$$

$$\bar{x}_{ji} = \begin{cases} 1, if\ job\ j \in N\ is\ fragmented\ to\ day\ i \in T \\ 0, otherwise \end{cases}$$

$$y_i = \begin{cases} 1, if\ overtime\ decision\ is\ made\ in\ day\ i \in T \\ 0, otherwise \end{cases}$$

z_i: $fragmented\ time\ amount\ to\ day\ i \in T$

The mathematical model of the problem is as follows.

Objective

$$Min \sum_{i=1}^{t} y_i * o$$

Constraints

$$\sum_{j=1}^{n} \bar{x}_{j1} = 0 \quad (1)$$

$$z_1 = 0 \quad (2)$$

$$\sum_{i \in A_j} x_{ji} = 1 \qquad j \in N \; (3)$$

$$\sum_{i \in \bar{A}_j} x_{ji} = 0 \qquad j \in N \; (4)$$

$$\sum_{i \in A_j} \bar{x}_{ji} \leq 1 \qquad j \in F \; (5)$$

$$\sum_{i \in \bar{A}_j} \bar{x}_{ji} = 0 \qquad j \in F \; (6)$$

$$\sum_{i=1}^{t} \bar{x}_{ji} = 0 \qquad j \in \bar{F} \; (7)$$

$$\sum_{j \in a_i} \bar{x}_{ji} \leq 1 \qquad i \in T/\{1\} \; (8)$$

$$\sum_{j \in \bar{a}_i} \bar{x}_{ji} = 0 \qquad i \in T/\{1\} \; (9)$$

$$x_{ji} \geq \bar{x}_{ji+1} \qquad j \in a_i, i \in T/\{t\} \; (10)$$

$$z_i \leq \sum_{j=1}^{n} \bar{x}_{ji} * P_j \qquad i \in T/\{1\} \; (11)$$

$$z_2 + y_1 * o \geq \sum_{j=1}^{n} x_{j1} * P_j - d \qquad (12)$$

$$y_t * o \geq z_t + \sum_{j=1}^{n} x_{jt} * P_j - d \qquad (13)$$

$$z_{(i+1)} + y_i * o \geq z_i + \sum_{j=1}^{n} x_{ji} * P_j - d \quad i \in T/\{1, t\} \qquad (14)$$

$$x_{ji}, \bar{x}_{ji}, y_i \in \{0,1\} \qquad j \in N, i \in T \; (15)$$

$$z_i \geq 0 \qquad i \in T \; (16)$$

In the model, objective function minimizes the total overtime. Constraints (1) and (2) ensure that a job cannot be fragmented before the first day of the planning horizon. Constraints (3) and (4) guarantee that jobs have to be assigned to only one day and they cannot be assigned to days that are later than their delivery dates. Constraints (5) and (6) state that if a job is fragmentable, it can be fragmented for the days that are earlier than its delivery dates. Constraints (7) guarantee that non-fragmentable jobs cannot be fragmented. Constraints (8) ensure that at most one job can be fragmented among all jobs for the day. Constraints (9) ensure that a job can not be fragmented for days to which it cannot be assigned. Constraints (10) state that a job can only be fragmented on the day after its original assigned day. Constraints (11) guarantee that the fragmented time of the day cannot be more than the processing time of the fragmented job. Constraints (12-14) ensure that the capacity of each day is not exceeded. The planning horizon can be separated into three sections by considering their special conditions while setting up the model. The first day cannot have an earlier day's remaining capacity requirement. Constraints (12) control the capacity of the first day of the planning horizon. The last day cannot have over capacity to transfer the next day. Constraints (13) control the last day of the planning horizon. Constraints (14) control the capacity of the remaining days over the planning horizon. Constraints (15) and (16) define the domain for decision variables.

4. Numerical study

This section presents the numerical study conducted to evaluate the performance of the MIP formulation on the BPPIF. The MIP model was solved using Gurobi Optimizer version 9.5.0.0rc5. A PC with a 16 GB RAM and Apple M1 processor was used to carry out all the experiments. We imposed a run time limit of 60 seconds for the MIP model.

4.1. Data generation

We construct our experiment design based on average processing time (APT), fragmentable item ratio (FIR), job number (JN), and capacity ratio (CR) parameters. In our test instances, we aim to have a nearly equal number of jobs with the same due date. To do so, day numbers are calculated parametrically as follows:

$$DN = \frac{JN * APT * CR}{DT}$$

where *DN* represents the day number, *JN* represents the job number, *APT* represents the average processing time, *CR* represents the capacity ratio and *DT* represents the day time which is the total time of normal shift time and over time per day.

We generated the processing time of jobs randomly by using the uniform distribution with mean 40. The bounds of the uniform distribution is determined by the following equations.

$$upperLimit = APT * CR$$
$$lowLimit = APT * (2 - CR)$$

In our experimental setup, we used the following set of parameters; *JN*∈{176, 352, 704, 1408}, *FI*∈{20%, 40%, 60%} and *CR*∈{110%, 160%}.

We employed a daytime duration of 704 minutes, with 554 minutes designated as normal shift time and the remaining 150 minutes allocated for overtime. To ensure statistical robustness, we generated 10 samples for each instance class. This leads us to 240 test instances in total. The average values obtained from these samples were subsequently employed in analyzing results. Finally, *DN*∈{11, 16, 22, 32, 44, 64, 88, 128} are found implicitly based on parameters given above.

4.2. Results

In this section, we present the computational results of our experiments. The results on the test instances show that the model yields optimal solutions for 90% of the 240 test instances within a time limit of 60 seconds. In only 10% of the total instances, it fails to find the optimal solution within the given time limit. The maximum gap is found as %4 while the average gap found as %3 in those instances that were not solved optimally. This result highlights the effectiveness of the proposed mathematical model in solving a significant portion of the test instances efficiently.

Table 2 presents the average, minimum and maximum solution times for the instances based on each parameters. It can be observed from the table that as the number of jobs increases, the solution time of the model also increases. The relationship between the number of jobs and the solution time can be attributed to the computational complexity of the problem. As the number of jobs increases, the problem becomes more complex and requires additional computational resources to find the optimal solution. This increase in solution time is expected due to the combinatorial nature of the problem, where the number of possible solutions grow exponentially with the number of tasks.

The effect of FI ratio, which indicates the percentage of jobs that are fragmentable can be also observed in the Table 2. It is well-known that the problem becomes trivial if there is no restriction on the item fragmentation. This indicates that one would expect to have faster computation as the ratio between fragmentable and nonfragmentable jobs approaches to 1. The results show that, the average solution time decreases when the number of fragmentable jobs in the problem increases sufficiently. Yet, we cannot observe a similar trend in the maximum solution time statistics.

Table 2. Solution times for parameters.

Group	*Value*	AST	Min	Max
JN	176	0.020	0.002	0.100
	352	0.143	0.008	1.006
	704	2.501	0.033	18.073
	1408	6.633	0.142	56.255
FI	20%	0.762	0.002	12.710
	40%	2.788	0.002	42.831
	60%	1.877	0.002	56.255
CR	110%	4.060	0.020	56.255
	160%	0.052	0.002	0.190

JN: Job number, FI: Fragmentable item ratio, CR: Capacity ratio, AST: Average solution time (seconds), Min: Minimum solution time (seconds), Max: Maximum solution time (seconds)

The CR ratio, as mentioned before, has an effect on the randomly determined processing times for jobs. As the CR ratio increases, the average total processing time of jobs that must finish prior to any day decreases. This implies that, for lower CR ratio, the daily capacity becomes tighter as compared to that of higher CR ratio. As such, the problem inherently gets more difficult. In Table 2, it can be observed that solution times dramatically improve as CR ratio increases.

Finally, the average solution times for DN with same CR are given in Table 3. CR have a direct impact on the calculation of DN. CR, in a way, indicates the underutilization of the workcenter each day. Therefore, for the same CR, the rate of utilization, i.e., daily capacity tigthness, remains steady.

Table 3. Solution times of DN for same CR.

DN	CR	AST	Min	Max
16	160%	0.002	0.002	0.002
32	160%	0.008	0.008	0.008
64	160%	0.035	0.033	0.039
128	160%	0.165	0.142	0.190
11	110%	0.039	0.002	0.100
22	110%	0.278	0.089	1.006
44	110%	4.967	0.734	18.073
88	110%	45.436	39.386	56.255

DN: Day number, CR: Capacity ratio, AST: Avereage solution time (seconds), Min: Minimum solution time (seconds), Max: Maximum solution time (seconds)

In Table 3, we report the impact of day number on the average, minimum and maximum solution time for the instances with the same underutilization level. As a result, we can see that DN has a negative impact on the average solution time. More specifically, the average solution time increases with greater DN values. This is not surprising because the number of binary variables used within the MIP model depends on DN. Therefore, greater DN values lead us to larger MIP models and this eventually increases the solution times.

5. Conclusions and future research directions

We have studied the assignment problem of multi product assembly jobs to the days. Considering the similarity of the problem with the BPPIF, we have proposed a MIP model. We have considered the additional constraint of item-bin conflict. An item can be fragmented at most once, and only to the following bin. Computational experiments have shown that the MIP model obtained optimal solutions for majority of instances within reasonable computation times. Our results provide insights for researchers and practitioners in scheduling problems with due dates and overtime constraints.

The proposed model can be used as a decision support tool in planning production assignments where sequence-dependent setup times can be ignored, and delivery times are of interest. The MIP model can be considered as a simple yet effective model for systems involving processes with less than 1000 job numbers.

Several research directions could be considered for future work. The performance of the MIP model for the BPPIF could be examined further to minimize the number of fragmented jobs. Despite extensive research on the BPPIF, several open problems and challenging variants still exist, including the BPPIF with uncertain item sizes, multiple criteria, and precedence relations. Furthermore, developing efficient algorithms, such as metaheuristics, for handling large-scale instances of the problem remains an active research area.

References

[1] Gurkan, M. E., & Tunc, H. (2021). A fix-and-optimize heuristic for the capacitated multi-item stochastic lot-sizing problem. *An International Journal of Optimization and Control: Theories & Applications*, 11(1), 41-51.

[2] Cetin, K., Tuzkaya, G., & Vayvay, O. (2020). A mathematical model for personnel task assignment problem and an application for banking sector. *An International Journal of Optimization and Control: Theories & Applications*, 10(2), 147-158.

[3] Göçgün, Y., & Bakır, N. O. (2022). Optimal matchday schedule for Turkish professional soccer league using nonlinear binary integer programming. *An International Journal of Optimization and Control: Theories & Applications*, 12(2), 113-127.

[4] Koç, Ç., Erbaş, M., & Özceylan, E. (2018). A rich vehicle routing problem arising in the replenishment of automated teller machines. *An International Journal of Optimization and Control: Theories & Applications*, 8(2), 276-287.

[5] Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, 17(5), 259-268.

[6] Jansen, K. (1999). An approximation scheme for bin packing with conflicts. *Journal of Combinatorial Optimization*, 3(4), 363-377.

[7] Loh, K. H., Golden, B., & Wasil, E. (2008). Solving the one-dimensional bin packing problem with a weight annealing heuristic. *Computers & Operations Research*, 35(7), 2283-2291.

[8] Khanafer, A., Clautiaux, F., & Talbi, E. G. (2010). New lower bounds for bin packing problems with conflicts. *European Journal of Operational Research*, 206(2), 281-288.

[9] Crainic, T. G., Perboli, G., Rei, W., & Tadei, R. (2011). Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers & Operations Research*, 38(11), 1474-1482.

[10] Elhedhli, S., Li, L., Gzara, M., & Naoum-Sawaya, J. (2011). A branch-and-price algorithm for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 23(3), 404-415.

[11] Fleszar, K., & Charalambous, C. (2011). Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research*, 210(2), 176-184.

[12] Bertazzi, L., Golden, B., & Wang, X. (2019). The bin packing problem with item fragmentation: a worst-case analysis. *Discrete Applied Mathematics*, 261, 63-77.

[13] Ekici, A. (2021). Bin packing problem with conflicts and item fragmentation. *Computers & Operations Research*, 126, 105113.

[14] Casazza, M., & Ceselli, A. (2014). Mathematical programming algorithms for bin packing problems with item fragmentation. *Computers & Operations Research*, 46, 1-11.

[15] LeCun, B., Mautor, T., Quessette, F., & Weisser, M. A. (2015). Bin packing with fragmentable items: Presentation and approximations. *Theoretical Computer Science*, 602, 50-59.

[16] Byholm, B., & Porres, I. (2018). Fast algorithms for fragmentable items bin packing. *Journal of Heuristics*, 24, 697-723.

[17] Dokeroglu, T., & Cosar, A. (2014). Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. *Computers & Industrial Engineering*, 75, 176-186.

[18] Casazza, M. (2019). New formulations for variable cost and size bin packing problems with item fragmentation. *Optimization Letters*, 13(2), 379-398.

[19] Ekici, A. (2022). Variable-sized bin packing problem with conflicts and item fragmentation. *Computers & Industrial Engineering*, 163, 107844.

[20] Arbib, C., & Marinelli, F. (2017). Maximum lateness minimization in one-dimensional bin packing. *Omega*, 68, 76-84.

[21] Khanafer, A., Clautiaux, F., Hanafi, S., & Talbi, E. G. (2012). The min-conflict packing problem. *Computers & Operations Research*, 39(9), 2122-2132.

Mustafa Ustuncelik *is an Industrial Engineer and serves as a Planning and Control Assistant Manager at Hidromek Construction Equipment Company located in Thailand. He received his B.S. in Industrial Engineering from Eskisehir Osmangazi University in 2013 and his M.S. in Business Administration from Social Sciences University of Ankara in 2023.*

http://orcid.org/0009-0008-3385-7808

Cagri Koc *is an Associate Professor in the Department of Business Administration at Social Sciences University of Ankara. He received his Ph.D. degree (2015) in Operational Research/Management Science from the Southampton Business School of University of Southampton. He worked as a Postdoctoral Fellow at HEC Montreal and at CIRRELT. His research mainly focuses on the application of mathematical and metaheuristic optimization techniques to logistics and supply chain management problems.*

https://orcid.org/0000-0002-7377-204X

Huseyin Tunc *is an Associate Professor in the Department of Business Administration at Social Sciences University of Ankara. He received his Ph.D. in Industrial and Systems Engineering from Mississippi State University in 2012. He joined the Social Sciences University of Ankara in 2019. He worked at Hacettepe University, Swiss Federal Institute of Technology in Lausanne (EPFL), and Mississippi State University.*
https://orcid.org/0000-0001-5508-3702

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)