THE REPUBLIC OF TURKEY

SOCIAL SCIENCES UNIVERSITY OF ANKARA

THE GRADUATE SCHOOL OF SOCIAL SCIENCES

SINGLE MACHINE SCHEDULING WITH OVERTIME IN MULTI PRODUCT ASSEMBLY ENVIRONMENT

Master's Thesis

Mustafa ÜSTÜNÇELİK

Business Administration

May 2023

THE REPUBLIC OF TURKEY

SOCIAL SCIENCES UNIVERSITY OF ANKARA

THE GRADUATE OF SOCIAL SCIENCES

SINGLE MACHINE SCHEDULING WITH OVERTIME IN MULTI PRODUCT ASSEMBLY ENVIRONMENT

Master's Thesis

Mustafa ÜSTÜNÇELİK

Assoc. Prof. Dr. Çağrı KOÇ

Assoc. Prof. Dr. Hüseyin TUNÇ

Business Administration

May 2023

ACKNOWLEDGMENTS

In preparing my thesis a number of people have helped me directly and indirectly. I would like to take this opportunity to thank them.

Firstly, I am particularly grateful to my tutor Assoc. Prof. Dr. Çağrı KOÇ. He has guided me with great deal of patience and encouraged me throughout my studies in ASBU. He had great influence on my thesis and work in general.

Secondly, my sincere thanks go to Assoc. Prof. Dr. Hüseyin TUNÇ. He has provided valuable directions, on-the-point help and guidance on the topic in addition to reviewing my work patiently.

Finally, I would say a very special note of gratitude and indebtedness to my family, for their spiritual support, encouragement and inspiration on this and many other facets of my life.

Without these people, this study would never been possible.

TABLE OF CONTENTS

ACKNOWLEDGMENTS **i**

TABLE OF CONTENTS **ii**

ÖZET **iii**

ABSTRACT **iv**

LIST OF FIGURES **v**

LIST OF TABLES **vi**

1. INTRODUCTION **1**

1.1 Motivation and Problem Definition *1*

1.2 Contributions and Organization of This Dissertation *3*

2. LITERATURE REVIEW **5**

2.1 A Review on Machine Scheduling with Overtime *5*

2.2 A Review on the Bin Packing Problem *7*

3. MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME AND OVER TIME **10**

3.1 Problem Definition and Notation *10*

3.2 Mathematical Formulation *12*

4. SOLUTION METHODS **18**

4.1 Decomposition Based Heuristic 1 *18*

4.2 Decomposition Based Heuristic 2 *20*

5. BIN PACKING PROBLEM WITH ITEM FRAGMENTATION **23**

5.1 Problem Definition and Notation *23*

5.2 Mathematical Formulation *24*

6. COMPUTATIONAL EXPERIMENTS **28**

6.1 Data Sets and Experimental Settings *28*

6.2 Results *29*

7. CONCLUSIONS **35**

BIBLIOGRAPHY **36**

CURRICULUM VITAE **39**

ÖZET

Bu tez çalışmasında farklı montaj önceliklerine sahip, çoklu montaj gruplu ürünler için bir gerçek hayat uygulamasını temel alan iki tür problem ele alınmıştır. Birinci problem, sıra bağımlı hazırlık sürelerine sahip işler için istenilen planlama ufkunda ve sonlu kapasite kısıtı altında fazla mesai miktarını en küçükleyerek montaj gecikmesine sebep olmayacak günlük çizelgenin belirlenmesidir. Problem için ilk olarak karma tam sayılı matematiksel programlama modeli önerilmiştir. Model performansını ölçmek için oluşturulan rassal örnek veri setlerinde gün başına düşen iş sayısı arttıkça makul sürelerde matematiksel model çözüm bulamadığından iki ayrıştırma algoritması geliştirilmiştir. İkinci problem, sıra bağımlı hazırlık sürelerinin önemli olmadığı işler için fazla mesai miktarını en küçükleyerek montaj gecikmesine sebep olmayacak şekilde işlerin günlere atanmasıdır. Problemin bölünebilir ürünlü kutulama problemiyle olan benzerliği göz önüne alınarak literatürdeki karma tam sayılı matematiksel model genişletirelerek önerilmiş ve çözüm değerleri karşılaştırılmıştır.

Anahtar Kelimeler: *makina çizelgelme problemi, iş çizelgeleme, fazla mesai, kutulama problemi, bölünebilir ürün.*

ABSTRACT

In this thesis, two types of problems based on a real-life application for products with multiple assembly groups with different assembly priorities are discussed. The first problem is to determine the daily schedule that will not cause assembly delay by minimizing the amount of overtime under the desired planning horizon and finite capacity constraint for the jobs with sequence dependent setup times. A mixed integer mathematical programming model is proposed for the problem. In the random sample datasets created to measure the model performance, the solution could not be found in a reasonable time as the number of jobs per day increased. Therefore, two decomposition algorithms have been developed for the solution. The second problem is the assignment of jobs to days in a way that minimizes the amount of overtime and does not cause assembly delays for jobs where sequence dependent setup times are not important. Considering the similarity of the problem with the bin packing problem with item fragmentation, the mixed integer mathematical model in the literature was extended and proposed and the solution values were compared.

Keywords: *machine scheduling problem, job scheduling, overtime, bin packing problem, fragmentable item.*

LIST OF FIGURES

Figure 1 Production Order Based on Processes 11
Figure 2 Multiple Level Product BOM 11
Figure 3 Routings for Sub Assembly 3 Items 11
Figure 4 Binary Values of Task 2 16
Figure 5 An example of 5 Jobs 3 Day Problem for DBH 1 19
Figure 6 An example of 5 Jobs 3 Day Problem for DBH 2 21
Figure 7 A feasible solution example for 6 items and 3 days problem 27

LIST OF TABLES

Table 1 Summary of the literature on machine scheduling with overtime.....1
Table 2 Summary of the literature on bin packing problem.....1
Table 3 Results for 8,12 and 18 jobs - 2 days instances.....17
Table 4 Structure of instances 28
Table 5 Results for 8 jobs - 2 days instances.....30
Table 6 Results for 12 jobs - 2 days instances.....30
Table 7 Results for 18 jobs - 2 days instances.....31
Table 8 Results for 8 jobs - 4 days instances.....31
Table 9 Results for 12 jobs - 4 days instances.....32
Table 10 Results for 18 jobs - 4 days instances.....32
Table 11 Results for 8 jobs - 8 days instances.....33
Table 12 Results for 12 jobs - 8 days instances.....33
Table 13 Results for 18 jobs - 8 days instances.....34

1. INTRODUCTION

1.1 Motivation and Problem Definition

Production scheduling is an important operational problem as it directly affects the cost of production. Especially for productions that require assembly precedencies, an inappropriate production schedule may be the reason for overtime decisions for a finite planning period. An inappropriate production schedule may result in assembly jobs being delayed as a result of subassembly parts not being prepared on time. In order to meet product deadlines, overtime is often required to compensate for the delays. Overtime has a direct impact on the profitability of the company, as it means additional production costs.

This thesis studies the production scheduling problem in a company that specializes in the manufacturing of construction machinery, which involves multiple sub-assembly groups and multi-level product trees. The complicated nature of the product structures, characterized by the presence of numerous sub-assembly groups and their corresponding multi-level product trees, poses a significant challenge in the planning and management of material resources. In addition, work centers differ depending on whether the setup time spent for the job is sequence-based or not. For some work centers, the sequence in which the jobs are processed in the work center does not matter in terms of the set-up time spent; while for others, the sequence in which the jobs are processed in the work center has a direct effect on the setup times. That makes us face two different problems to develop solutions. The first problem is to find a sequence of jobs for only sequence-dependent set-up time work centers to schedule jobs. The second problem is to decide only the production day of each job without considering any sequence of the jobs because the sequence of the jobs is not important on behalf of set-up times.

The prioritization of tasks and the processing of different prioritized tasks on the same workstations directly affects the scheduling process. Failure to adhere to the priority order of tasks assigned to workstations leads to delays or unnecessary waiting times in the entire production plan. To minimize delays and unnecessary waiting times in the company's production schedule, overtime work is often necessary which increases production costs.

The production planning process in the company that served as the basis for this thesis is conducted on a monthly basis. Products are grouped, based on their operations, and transformed into weekly plans which are then used to generate job orders. The delivery times of products for each operation group are assumed to be the last day of the corresponding week. During the grouping process, the priority of work centers and the priority of product requisitions are not considered. In this method, products are generally introduced into the production process based on their arrival time at the work centers. If a product with a lower priority is started in the production process before a product with a higher priority, additional work hours are required to prevent delays in subsequent operations. If overtime work is decided for a work center in the company, even if the work is completed, the full overtime fee must be paid. Therefore, when scheduling is not managed carefully, idle waiting situations can also arise.

In real-life problem, there are a large number of products and work centers involved which significantly increase the complexity of the problem. To mitigate the complexity of the problem, this thesis studies the scheduling problem for a single machine where the processing and delivery times of jobs are predetermined. The objective is to minimize the total overtime required to complete all tasks.

For the first problem, initially, the problem was identified by examining the studies in the literature. An earlier mathematical model proposed to formulate a similar problem to the one considered in this thesis was chosen as a base model and extended to correspond the problem addressed. However, the resulting model could not provide an acceptable solution time for real-life situations with high workload and long planning periods. Therefore, two different heuristic solution methods were developed. In these methods, the main idea is to divide the tasks according to their delivery times and to use the mathematical model on a daily basis. The main goal here is to decompose the problem into smaller sub-problems to decrease the solution time. The proposed heuristics can be solved with off-the-shelf solvers as they are built on the introduced mixed-integer programming (MIP) formulation. As such, they can be easily and quickly adapted and utilized in practice. Furthermore, due to their MIP-based nature, parametrization is not an issue indeed.

In the first method, after each iteration, the assignment decisions are fixed for jobs considered, and the mathematical model is solved for the unassigned jobs in the next iteration. Therefore, a smaller solution set is searched in each iteration which reduces the solution time. The

second method is built on the first one, and hence, it strictly follows the former. The only difference with the second method is that it also fixes the sequencing decisions for the assigned tasks. Thus, the searched solution set is further narrowed down.

For the second problem, a different solution approach was developed to decide the processing days of jobs. In the second problem, since the sequence of the jobs to be processed in the work center on the same day has no effect on the total production time, the problem can be transformed into a variant of bin packing problem that aims to find optimal item placements into variable sized bins where the items can be fragmented. For the problem considered in this thesis, the items correspond to tasks, whereas the bins are replaced with days. Some jobs are allowed to fragment to the next day like item fragmentations to different bins. Considering this similarity, a mixed-integer mathematical programming model in the literature was extended to include specific conditions present in the problem considered.

1.2 Contributions and Organization of This Dissertation

This thesis presents a MIP model and solution approach that can be readily adapted in real-life problems with ease of parameterization. The MIP model offers a versatile and user-friendly framework to handle practical optimization problems. This flexibility allows for the incorporation of problem-specific constraints and objectives, making it suitable for a broad range of applications. The solution method combines heuristics and exact methods to balance solution quality and computation time efficiently.

A modified bin packing model is also presented, as a special case, where sequence-dependent setup times are not included. This modification eliminates the need to consider the sequence of jobs or tasks and results in a simpler and more flexible model that can be applied to a broader range of scenarios.

This thesis is structured as follows. Section 2 presents the literature review on machine scheduling with overtime and bin packing problems which collectively establish the foundation of the problem addressed in this thesis. Section 3 presents the first problem that machine scheduling problem for sequence-dependent setup time and over time, notations, and provides the mixed-

integer mathematical programming formulation. Section 4 describes heuristics and mathematical models. Section 5 presents the second problem which sequence-dependent setup times are not considered, notations mathematical model formulation. Section 6 presents numerical experiments which conducted to evaluate the performance of the solution methods. Finally, Section 7 provides conclusions along with recommendations for future research directions.

2. LITERATURE REVIEW

This thesis considered a real-life problem arises in a multi-product assembly environment that has overtime constraints in the context of machine scheduling. We conducted a comprehensive review of the literature on machine scheduling with overtime constraints (Section 2.1). Furthermore, we considered the case where sequence-dependent setup times are not significant and approached it as a bin packing problem with item fragmentation. To this end, we also reviewed the literature of the bin packing problems with item fragmentation to identify similarities and differences with our research problem (Section 2.2).

2.1 A Review on Machine Scheduling with Overtime

Machine scheduling is a well-known problem in the literature. The general class of single-machine (or single-resource) sequencing and scheduling problems is among the most studied problem classes in operations research (Pinedo, 2002). Single machine scheduling with overtime is a classic problem in the field of scheduling theory which involves scheduling a set of jobs on a single machine while allowing for overtime. In this problem, the objective is to minimize a given cost function, such as the sum of completion times or total overtime. The importance of this problem lies in its practical relevance to many real-world scheduling applications such as production planning, transportation, and service industries. In many cases, overtime is necessary to meet the demand or to avoid penalties for late delivery which makes it critical to find an optimal or near-optimal schedule that incorporates overtime efficiently. Scheduling problems with overtime are common in many industries, where the amount of work exceeds the available working hours. To solve such problems, researchers aim to find an optimal or near-optimal schedule that satisfies a set of constraints while minimizing a specific objective function. These objectives can include time, cost, profit, and earliness/tardiness.

When solving scheduling problems, researchers consider different types of constraints such as capacity constraints, due dates, and costs. Capacity constraints limit the amount of resources available to complete the tasks, while due dates impose deadlines which the tasks must be

completed. Furthermore, the cost of overtime and the cost of setup time are also considered as main constraints of scheduling problems with overtime.

To deal with these scheduling problems, several studies developed heuristics or exact algorithms. These algorithms aim to find the best schedule that satisfies constraints and minimizes the objective function. Table 1 summarizes the studies in the literature. Holloway et al. (1974) and Matsuo (1988) proposed algorithms that minimize overtime as the objective function while considering due dates and the cost of overtime as constraints. Similarly, Akkan (1996) and Freeman et al. (2014) developed algorithms that minimize cost as the objective function while considering due dates and the cost of setup time as constraints. On the other hand, Mestry et al. (2011) and Li et al. (2021) proposed algorithms that maximize profit as the objective function while considering capacity constraints. Zobolas et al. (2008), Jaramillon et al. (2017), Yang et al. (2004), and Ventura et al. (2013) developed algorithms that minimize earliness/tardiness as the objective function while considering capacity constraints and due dates as constraints.

In the literature on machine scheduling problems, the number of sources to be scheduled is an important criterion for evaluating problem-solving approaches. Studies in this area can be broadly classified into two groups based on the number of machines involved: (i) scheduling a single machine (Holloway et al., 1974; Yang et al., 2004; Zobolas et al., 2008; Jaramillo et al., 2017); and (ii) scheduling more than one machine (Matsuo, 1988; Akkan, 1996; Mestry et al., 2011; Ventura et al., 2013; Freeman et al., 2014; Li et al., 2021).

The aim of this thesis is to reduce the complexity of the problem and to demonstrate the practical applicability of the proposed method. To achieve this goal, this thesis focuses on the single-machine scheduling problem which is a real-world problem. In this problem, the parameters of all jobs such as processing times, delivery times and sequence dependent setup times etc. to be scheduled on a given machine are assumed to be pre-determined.

Table 1 Summary of the literature on machine scheduling with overtime

Study	Objective				Constraints			Resource	Solution Method
	T	*C*	*P*	*E / T*	*Cap*	*DD*	*C*		
Holloway et al. (1974)	+					+		Single	Heuristic
Matsuo (1988)	+						+	Multi	Heuristic
Akkan (1996)		+				+		Multi	Heuristic based on work-order insertion approach
Yang et al. (2004)		+		+			+	Single	Heuristic
Zobolas et al. (2008)		+		+	+			Mutli	Genetic algorithm
Mestry et al. (2011)			+		+			Multi	Mixed-Integer Linear Program Branch-and-Price (B&P) algorithm
Ventura et al. (2013)				+	+			Multi	Genetic algorithm
Freeman et al. (2014)		+					+	Multi	Mix Integer Programing Decomposition heuristic
Jaramillo et al. (2017)		+		+		+		Single	Heuristic based on an effective priority rule
Li et al. (2021)	+				+			Multi	Dynamic programming based exact algoritms Genetic Algorithm
This study	+				+	+		Single	Heuristic

T: overtime, C: cost, P: profit, E/T: earliness/tardiness, Cap: capacity, DD: due dates

2.2 A Review on the Bin Packing Problem

The bin packing problem (BPP) is a classic optimization problem that has been extensively studied in the literature. The problem involves packing a set of items into a minimum number of bins of fixed size where each item has a size and each bin has a known capacity. The objective is to minimize the number of bins used to pack all the items. The BPP is known to be NP-hard, which means that finding an optimal solution can be computationally infeasible for large problem instances. Therefore, various heuristics and metaheuristics have been proposed to find good-quality solutions in a reasonable amount of time. These methods include first-fit, next-fit, best-fit, worst-fit and various metaheuristic optimization techniques such as genetic algorithms, simulated annealing and tabu search. Despite the extensive research on the problem, there are still many open problems and challenges, making it an active research area in combinatorial optimization.

The BPP has many practical applications, including packing items in a warehouse or shipping containers, assigning tasks on a set of machines, and allocating resources in cloud computing. In addition, the BPP has theoretical implications, as it is related to other combinatorial optimization problems such as the knapsack problem, the cutting stock problem, and the vehicle routing problem.

The BPP has been a subject to intensive research for several decades, resulting in various solution approaches developed to solve the problem under different objectives and constraints. In Table 2, we summarize some of the studies that conducted on the BPP with item fragmentation classified based on the objectives, constraints, and solution methods employed. One of the primary objectives in the BPP is to minimize the number of bins used. This objective studied extensively in the literature, with numerous authors proposing various algorithms and heuristics to achieve this goal. (Eilon et al., 1971; Jansen, 1999; Loh et al., 2008; Khanafer et al., 2010; Crainic et al., 2011; Elhedhli et al., 2011; Fleszar et al., 2011; Bertazzi et al., 2019; Ekici, 2021). Another objective that received attention in the literature is the number of fragmentations or partial items in the bins (Casazza et al., 2014; LeCun et al., 2015; Byholm et al., 2018). Cost is another important objective in the BPP (Crainic et al., 2011; Dokeroglu et al., 2014; Casazza et al., 2018; Ekici, 2022). The load of bins, or the amount of space used in each bin, is an objective that also studied in the literature

(Loh et al., 2008). The makespan or the time required to pack all the items is studied (Arbib et al., 2017).

Constraints in the BPP include conflicting items (Jansen, 1999; Khanafer et al., 2010; Elhedhli et al., 2011; Ekici, 2021; 2022), and fragmentable items (Casazza et al., 2018; Bertazzi et al., 2019; Ekici, 2021; Ekici, 2022). The number of bins studied as a constraint (Loh et al., 2008; Khanafer et al., 2012; Casazza et al., 2014; Dokeroglu et al., 2014; LeCun et al., 2015; Arbib et al., 2017; Byholm et al., 2018). Bin capacity is another constraint that has been considered (Eilon et al., 1971; Crainic et al., 2011; Fleszar et al., 2011; Casazza et al., 2018; Ekici, 2022).

In the literature, constraints on the conflict of items and the total number of conflicts are considered. However, in this thesis, the additional constraint of conflict of bins is also considered. In the problem under investigation, an item can be fragmented at most once, and only to the following bin. In order to provide this constraint, the bins that fragmentable items cannot be fragmented are considered as conflict bins of items.

Table 2 Summary of the literature on bin packing problem

Study	Objective							Constraints					Solution Method
	NB	NF	C	NC	LB	M	OCU	CI	FI	NB	BC	CB	
Eilon et al. (1971)	+										+		Zero-one programming mode Heuristic
Jansen (1999)	+							+					Asymptotic approximation
Loh et al. (2008)					+					+			Weight annealing
Khanafer et al. (2010)	+							+					Dual-feasible functions Data-dependent dual-feasible fun.
Crainic et al. (2011)	+		+								+		Heuristic /Lower bounds
Elhedhli et al. (2011)	+							+					Branch-and-Price algorithm
Fleszar et al. (2011)	+										+		Heuristic
Khanafer et al. (2012)				+						+			Column-generation methods, Heuristic, Tabu search
Casazza et al. (2014)		+								+			Exact opt. algorithms, Heuristics
Dokeroglu et al. (2014)			+							+			Genetic algorithm
LeCun et al. (2015)		+								+			Approximation algorithms
Arbib et al. (2017)						+				+			MIL Programming
Byholm et al. (2018)		+								+			Approximation and metaheuristic algorithms
Casazza et al. (2018)			+						+		+		Branch-andprice algorithm
Bertazzi et al. (2019)	+								+				Worst-case analysis
Ekici (2021)	+							+	+				Heuristic
Ekici (2022)			+					+	+		+		Heuristic, Lower bounds
This Study							+		+	+	+	+	Mixed Integer Programming

NB: number of bins, NF: number of fragmentations, C: cost, NC: number of conflicts, LB: load of bin, M: makespan, OCU: over capacity usage, CI: conflict items, FI: fragmentable items, BC: bin capacity, CB: conflict bins

3. MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME AND OVER TIME

This chapter presents the problem definition and notation in Section 3.1, and mathematical formulation in Section 3.2.

3.1 Problem Definition and Notation

This thesis studies the production scheduling problem in a company that manufactures and assembles industrial machines. The main body of the machine, composed of steel plates, is produced in-house by the company while the remaining materials are sourced from various external suppliers. The machine parts produced by the company undergo cutting and forming, machining, welding, painting, and assembly processes. Particularly, the items that have the welding process have a sub-assembly product tree structure that can reach up to level 7. Some of these items have operation at the machining centers again after the welding process.

The complex multi-level product structure of the parts and the different processing sequences for each part make the planning process in the factory very challenging. Currently, a production order is opened for each machine, listing all the parts that need to be produced for that machine. All the parts in the production order are grouped based on the five basic operations mentioned above and shared with the production department. A weekly production time with an assumption of on-time delivery is assigned for each operation, and it is expected that all the parts for a product from the production department will be completed within a five-week period. An example is shown in Figure 1 for the basic planning approach of each operation group weekly.

Figure 1 Production Order Based on Processes

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
Cutting & Forming	INH14	INH15	INH16	INH17	INH18
Machining	INH13	INH14	INH15	INH16	INH17
Welding	INH12	INH13	INH14	INH15	INH16
Painting	INH11	INH12	INH13	INH14	INH15
Assembly	INH10	INH11	INH12	INH13	INH14

When grouping the parts within production orders, the work centers and processing priorities are not considered. Figure 3 presents the work centers for processing all the parts required for main body production, according to the product tree structure shown in Figure 2. All the parts need to be processed at different work centers with different priorities. According to the product tree structure, to produce Item 6 in sub-assembly group 3, Item 2 and Item 3 must be processed first at the same work center. Any delay in the processing of Item 2 and Item 3 directly affects the completion time of sub-assembly group 3.

Figure 2 Multiple Level Product BOM

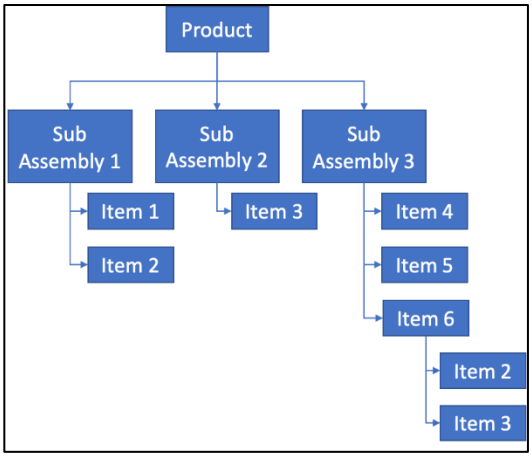
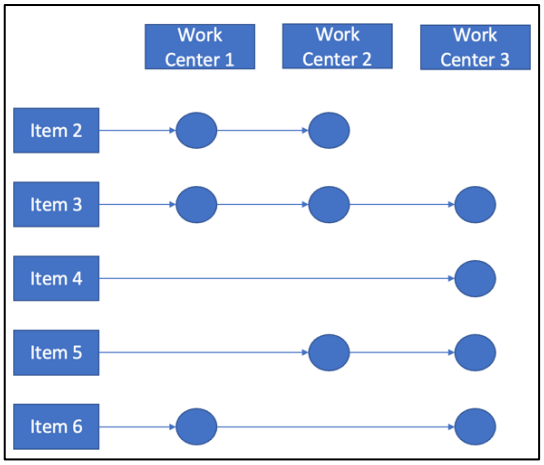


Figure 3 Routings for Sub Assembly 3 Items



Since there is no daily scheduling based on work centers and priorities in the production orders, these types of delays are handled by overtime, resulting in additional costs for the company. Furthermore, the inconsistencies in processing multiple-level products lead to some work centers to have idle time for work during the day, resulting in a direct decrease in the overall efficiency of the manufacturing department. An applicable daily production schedule based on work centers and production sequence can reduce the overtime costs that the company must bear and increase the efficiency of the manufacturing department, resulting in a decrease in overall manufacturing costs.

This thesis aims to simplify the complex problem of production scheduling, involving multiple products, sub-assemblies, order of operations, and work centers. The objective is to determine a daily production schedule for a limited-capacity work center within a specific planning horizon with the least amount of overtime required for completing the tasks assigned to a single work center.

The tasks to be performed in the work center are predetermined and their latest completion dates are predefined. The processing time and setup time for each task are also predetermined. The tasks can be divided into successive days, but the setup times cannot be divided. The setup time for a task must be completed on the same day it started. Otherwise, the task cannot be assigned to that day. Some tasks in the schedule can be divided into different days, while other tasks must start and finish on the same day. More than one task cannot be divided into the next day. The next task cannot be taken until the operation of a task is entirely finished in the work center. If a decision is made to do overtime for a day, the amount of overtime is used fully on that day.

3.2 Mathematical Formulation

In this thesis, a mathematical model is proposed based on the model developed by Bulbul (2017) to solve the problem. Furthermore, in our model, a solution is sought that will require the use of all overtime hours for the day on which the decision to work overtime is made, whereas in Bulbul (2007) overtime hours are allocated only up to the required amount and did not consider tasks that cannot be divided.

Sets

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

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

$JNF \quad \text{non} - fragmentable\ jobs\ set \qquad JNF \in N$

Parameters

$P_j \quad the\ processing\ time\ of\ job\ j \qquad j \in N$

$D_j \quad the\ delivery\ time\ of\ job\ j \qquad j \in N$

$A_{jk} \quad the\ setup\ time\ of\ job\ k\ after\ processing\ job\ j \qquad j, k \in N$

$ns \quad the\ time\ for\ a\ normal\ day\ shift$

$ds \quad the\ time\ for\ a\ working\ day$

$M \quad a\ big\ number$

Decision Variables

$$y_{jk} = \begin{cases} 1, if\ job\ j\ processed\ before\ job\ k \\ 0, otherwise \end{cases} \qquad j, k \in N$$

$$st_{ji} = \begin{cases} 1, if\ job\ j\ start\ to\ process\ in\ day\ i \\ 0, otherwise \end{cases} \qquad j \in N,\ i \in T$$

$$z_{ji} = \begin{cases} 1, if\ overtime\ made\ for\ job\ j\ in\ day\ i \\ 0, otherwise \end{cases} \qquad j \in N,\ i \in T$$

$$d_{ji} = \begin{cases} 1, if\ job\ j\ start\ to\ process\ in\ day\ i\ in\ normal\ shift \\ 0,\ if\ job\ j\ start\ to\ process\ in\ day\ i\ in\ overtime \end{cases} \qquad j \in N,\ i \in T$$

$$o_{ji} = \begin{cases} 1, if\ job\ j\ completed\ in\ or\ later\ day\ i \\ 0, otherwise \end{cases} \qquad j \in N,\ i \in T$$

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

$S_j \quad start\ time\ of\ job\ j \qquad j \in N$

$C_j \quad completion\ time\ of\ job\ j \qquad j \in N$

F_{ji}	$over\ time\ for\ job\ j\ in\ ith\ day$	$j \in N,\ i \in T$
B_{ji}	$over\ time\ that\ can\ be\ done\ for\ job\ j\ in\ day\ i$	$j \in N,\ i \in T$
θ_{ji}	$a\ variable\ to\ linearize\ S_j * z_{ji}$	$j \in N,\ i \in T$

Objective

$Min \sum_{i=1}^{t} \sum_{j=1}^{n} F_{ji}$

Constraints

$\sum_{j=0}^{n} y_{jk} = 1$	$k \in N$	(1)
$\sum_{k=1}^{n} y_{jk} \leq 1$	$j \in N\ \cup \{0\}$	(2)
$\sum_{i=1}^{t} (i-1) * ds * st_{ji}\ \leq S_j$	$j \in N$	(3)
$\sum_{i=1}^{t} st_{ji} = 1$	$j \in N$	(4)
$\sum_{i=1}^{t} i * ds * st_{ji} - S_j \geq\ \sum_{k=0}^{n} A_{jk} * y_{jk} + 1$	$j \in N$	(5)
$S_k \geq C_j - M * (1 - y_{jk})$	$j, k \in N\ \cup \{0\}$	(6)
$C_j \leq D_j$	$j \in N$	(7)
$C_k = S_k + P_k +\ \sum_{j=0}^{n} A_{jk} * y_{jk} + \sum_{i=1}^{t} (B_{ki} - F_{ki})$	$k \in N$	(8)
$i * (ds) - (ds - ns) - S_j \leq M * d_{ji}$	$j \in N,\ i \in T$	(9)
$S_j - (i * ds) + (ds - ns) \leq M * (1 - d_{ji})$	$j \in N,\ i \in T$	(10)
$(i * ds) - C_j \leq M * o_{ji}$	$j \in N,\ i \in T$	(11)
$C_j - (i * ds) \leq\ M * (1 - o_{ji})$	$j \in N,\ i \in T$	(12)
$B_{ji} \geq C_j - (i * ds) + (ds - ns) -\ M * (2 - d_{ji} - o_{ji})$	$j \in N,\ i \in T$	(13)
$B_{ki} \geq P_k + \sum_{j=0}^{n} A_{jk} * y_{jk} - M * (1 + d_{ki} - o_{ki})$	$k \in N,\ i \in T$	(14)

$$F_{ki} \geq P_k + \sum_{j=0}^{n} A_{jk} * y_{jk} - M * (1 + d_{ki} - o_{ki}) \qquad k \in N,\ i \in T \qquad (15)$$

$$B_{ji} \geq (i * ds) - S_j - M * (d_{ji} + o_{ji}) \qquad j \in N,\ i \in T \qquad (16)$$

$$B_{ji} \geq (ds - ns) - M * (1 - d_{ji} + o_{ji}) \qquad j \in N,\ i \in T \qquad (17)$$

$$F_{ji} \leq B_{ji} \qquad j \in N,\ i \in T \qquad (18)$$

$$\theta_{ji} \leq i * ds * z_{ji} \qquad j \in N,\ i \in T \qquad (19)$$

$$\theta_{ji} \leq S_j \qquad j \in N,\ i \in T \qquad (20)$$

$$\theta_{ji} \geq S_j - M * (1 - z_{ji}) \qquad j \in N,\ i \in T \qquad (21)$$

$$\left(i * ds * z_{ji}\right) - (ds - ns) \leq C_j \qquad j \in N,\ i \in T \qquad (22)$$

$$F_{ji} \leq M * z_{ji} \qquad j \in N,\ i \in T \qquad (23)$$

$$h_i \geq z_{ji} \qquad j \in N,\ i \in T \qquad (24)$$

$$\sum_{j=1}^{n} F_{ji} \geq (ds - ns) * h_i \qquad i \in T \qquad (25)$$

$$\sum_{i=1}^{t} st_{ji} = o_{jt} \qquad j \in JNF,\ t \in T \qquad (26)$$

$$y_{jk}, st_{ji}, z_{ji}, d_{ji}, o_{ji}, h_i \in \{0,1\} \qquad j,k \in N,\ i \in T \qquad (27)$$

$$S_j, C_j, B_{ji}, F_{ji}, \theta_{ji} \geq 0 \qquad j \in N,\ i \in T. \qquad (28)$$

In this model, objective function minimizes the total overtime for planning horizon. Constraints (1) and (2) are scheduling constraints that determine respectively premise and successor jobs. Constraint (1) ensures that there must be a premise job for each scheduled job. The set in which the successor jobs are defined contains only the set of jobs (*N*), while the set in which the premise jobs are defined also includes a dummy job that expresses the readiness state of the machine. Constraint (2) ensures that there must be at most one job scheduled as a successor of another job except the last job. Constraint (3) relates the variable S_j with the decision variable st_{ji} to shows st_{ji} as day unit. Constraint (4) ensures that each job must start one day over the planning horizon. Constraint (5) ensures that the job cannot be scheduled if setup cannot be completed or there is no time for processing after setup completion on that day. Constraint (6) ensures that a job cannot be started before its premise job is completed. Constraint (7) ensures that all jobs should be completed before their delivery time. Constraint (8) determines the completion time of each job considering overtime. Because the objective is to minimize total overtime, if overtime capacity will

not be fully used, the completion time will be postponed as ($Bj - Fj$) time. Starting time of each job is determined by Constraints (9) and (10). If a job starts in a normal shift Constraint (9) will be active, if a job starts in overtime Constraint (10) will be active.

Constraints (11) and (12) calculate the completion time of each job. Constraint (12) expresses the situation in which a job is completed in a day or later via binary variable o_{ij}. Constraint (13) expresses the situations in which a job is started on a normal shift and is completed normal shift or overtime on the same day. For both situations, the completion time will be forced to be equal to or less than the scheduled day due to B_j will be equal to zero (0). Constraints (14) and (15) express the situation in which a job is started on overtime and completed on same day. Constraints (16) and (17) express the situation in which a job is started respectively in a normal shift and in overtime and is not completed on the same day. Constraint (18) relates binary variables B_{ji} and F_{ji}.

Figure 4 Binary Values of Task 2

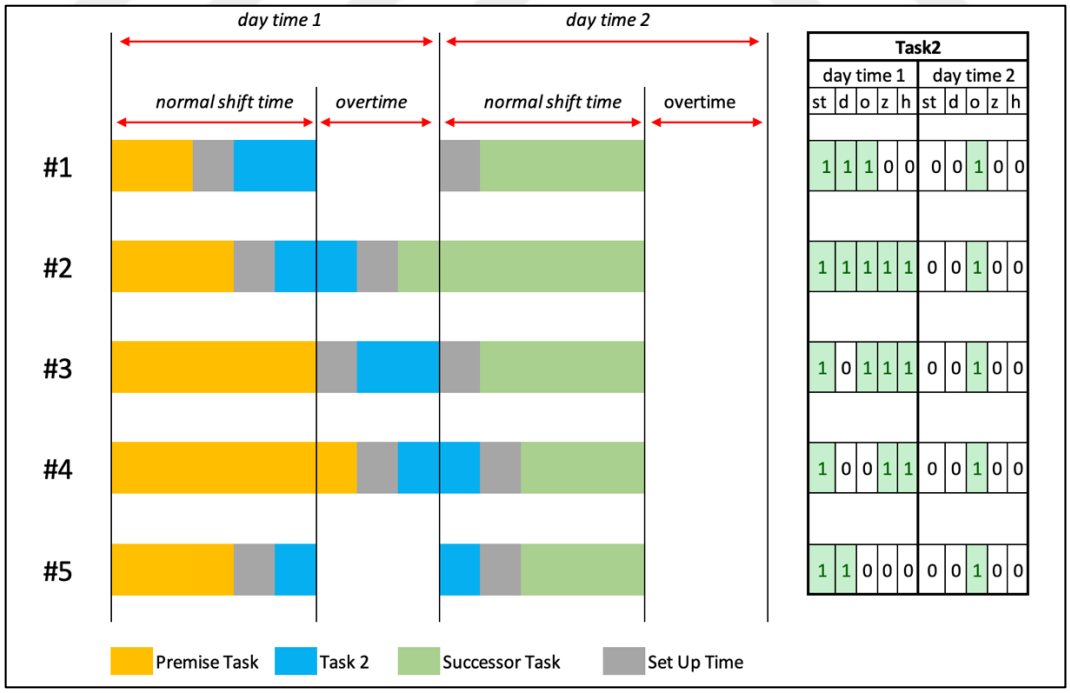


Figure 4 shows all possible values for Task 2 as an example. In the first scenario, Task 2 can start on the first day and finish on the same day within normal shift time hours. In the second

scenario, it may start in normal shift time on the first day and finishes in overtime. In the third scenario, it may start in overtime on the first day and finishes in overtime on the first day. In the fourth scenario, it may start in overtime on the first day and finishes in the normal shift time on the second day. Finally, in the fifth scenario, it may start on the first day and without doing over time finishes on the second day.

Constraints (19)–(23) relate binary variable z_{ji} with starting time variable (S_j), completion time variable (C_j) and overtime variable (F_{ji}). Constraints (24) and (25) ensure that if an overtime decision is made for the ith day, overtime should be used fully. Constraint (26) ensures that non-fragmentable jobs have to finish in the day they start. Constraints (27) and (28) are domain constraints for decision variables.

To evaluate the performance of the mathematical model random instances without considering sequence-dependent setup time for different job numbers for 2 days planning horizon was solved. As can be seen in below Table 3, although sequence-dependent setup times are not considered, after 12 jobs, the mathematical model cannot find optimal solution in time limit 250 seconds. To improve the solution approach for more job numbers, decomposition-based heuristics were developed.

Table 3 Results for 8,12 and 18 jobs - 2 days instances

T2	J8			J12			J18		
	R	Obj	St	R	Obj	St	R	Obj	St
1	OS	300	1.98	BS	300	250.02	BS	150	250.00
2	OS	300	2.01	BS	300	250.02	BS	300	250.01
3	OS	0	0.01	BS	300	250.02	BS	150	250.00
4	OS	300	2.43	BS	300	250.02	BS	150	250.02
5	OS	300	1.90	BS	300	250.03	NS		250.01

OS: optimal solution found, R: remark, Obj: objective, St: solution time,
BS: best solution found so far, NS: no solution in limit time

4. SOLUTION METHODS

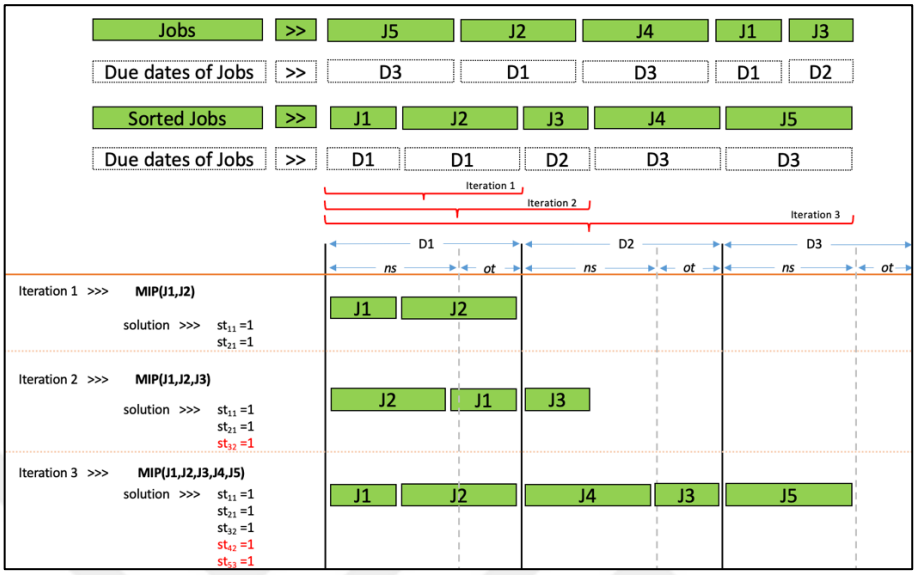
This chapter presents the decomposition-based heuristic 1 in Section 4.1, decomposition-based heuristic 2 in Section 4.2.

4.1 Decomposition-Based Heuristic 1

The proposed decomposition-based heuristic 1 (DBH 1) starts with grouping jobs based on their delivery times for each day. Then, an iterative daily scheduling approach is adopted starting at the outset. In each iteration, i.e. for each group is considered, a solution is generated by means of the MIP model introduced in the earlier section. Binary variables st_{ji}, which represent the assignment of each job, are fixed for the next iterations. The next iteration involves considering not only the current group of jobs but also previously scheduled jobs from earlier iterations whose day assignments are fixed. Although starting times are fixed, jobs within each group can be reordered in each iteration to optimize the overall schedule.

A simple example for 5 jobs -3 day is shown in Figure 5. After sorting jobs based on their due dates, DBH 1 find a solution for each iteration. In the first iteration, only J1 and J2 jobs are considered to be schedule because their due dates are first day of the planning horizon. For the second iteration, J1, J2 and J3 jobs are considered again while starting day decision binary st_{ji} for J1 and J2 are already fixed. As it can be seen on the Figure 5, jobs sequence can be changed due to sequence-dependent setup times while their start day cannot. For the third iteration, all jobs' sequence is considered again by fixing earlier starting day decisions and found a solution.

Figure 5 An example of 5 Jobs 3 Day Problem for DBH 1



DBH 1 aims to improve the scheduling efficiency in a rather myopic manner by considering the delivery times of jobs and optimizing the schedule through iterative grouping and reordering. By iteratively considering previously scheduled jobs and optimizing the current group, the DBH 1 can improve the overall schedule quality. Furthermore, by fixing the assigned delivery days, the DBH 1 can improve the computational efficiency of the scheduling process. The main idea of the DBH 1 solution is to improve solution time by decomposing the main problem into subproblems and fixing the values of binary variables representing day assignment of jobs.

DBH 1

Input: N,T, ns, ds, D_j, P_j, A_{jk}, JNF

groupJobs = {}
for i in T :
    for j in N :
        if $D_j \leq ds * i$:
            add job to groupJobs[i]
groupDays = {}

7: **for** i in T:
8: groupDays[i] = [1,i]
9: assignedJobsList = []
10: assignedJobsListUpdated = []
11: **for** i in T:
12: assignedJobsListUpdated = assignedJobsList
13: assignedJobsList = []
14: N = groupJobs[i]
15: T = groupDays[i]
16: MIP Model $(N, T, ns, ds, D_j, P_j, A_{jk}, JNF, assignedJobsListUpdated)$
17: **for** each job in assingedJobsListUpdated:
18: add constraints $st_{ji} = 1$
19: solution = Solve MIP Model
20: **for** i in T:
21: **for** j in N:
22: **if** $st_{ji} = 1$ *in solution*:
23: add st_{ji} to assignedJobsList

Output: Solution

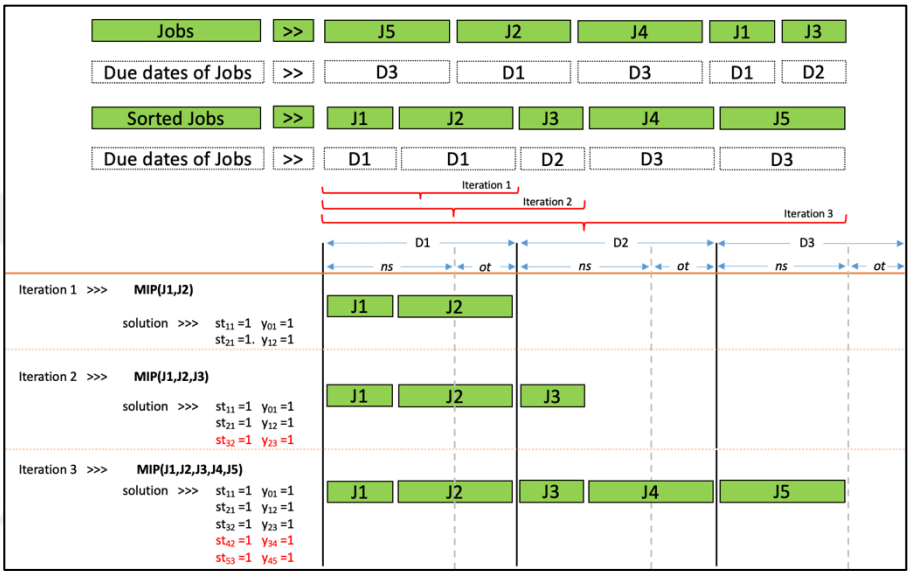
4.2 Decomposition-Based Heuristic 2

The decomposition-based heuristic 2 (DBH 2) approach presented below shares similar steps with the DBH 1 presented in the previous section. In the DBH 1, at the end of each iteration, binary variables showing job-day assignments are fixed. In the DBH 2, in addition to fixing the assignment variable (st_{ji}), decision variable values representing the prioritization of jobs (y_{jk}) are also fixed. The next iteration involves considering not only the current group of jobs but also the jobs previously scheduled in earlier iterations. The objective is to optimize the overall schedule while minimizing overtime by utilizing the flexibility of primer solutions.

The solution example of DBH 2 for the above example is shown in Figure 6. Until second iteration same steps are followed. In the second iteration, J1, J2 and J3 jobs are considered while not only starting day decision variables st_{ji} of J1 and J2 are fixed but also sequence variables y_{jk}

are considered as fixed. While J3 is scheduled after J1 in the second iteration in the above example for DBH 1, because we also fixed the sequence of the jobs at the end of the first iteration, J3 can be only scheduled after J2 in the second iteration. Remain processes are same as DBH 1.

Figure 6 An example of 5 Jobs 3 Day Problem for DBH 2



The main contribution of DBH 2 solution is its ability to improve solution time by decomposing the main problem into much smaller sub-problems as compared to those of decomposition DBH 1. By fixing the assigned day and order of jobs, the computational complexity of the scheduling problem is reduced, which may eventually improve the efficiency of the solution process.

DBH 2

Input: N,T, ns, ds, D_j, P_j, A_{jk}, JNF

1: groupJobs = {}
2: **for** i in T :
3: **for** j in N :

4: **if** $D_j \leq ds * i$:

5: add job to groupJobs[i]

6: groupDays = {}

7: **for** i in T:

8: groupDays[i] = [1,i]

9: assignedJobsList = []

10: assignedJobsListUpdated = []

11: sequenceList = []

12: sequenceListUpdated = []

13: **for** i in T:

14: assignedJobsListUpdated = assignedJobsList

15: assignedJobsList = []

16: sequenceListUpdated = sequenceList

17: sequenceList = []

18: N = groupJobs[i]

19: T = groupDays[i]

20: MIP Model ($N, T, ns, ds, D_j, P_j, A_{jk}, JNF, assignedJobsListUpdated$)

21: **for** each job in assingedJobsListUpdated:

22: add constraints $st_{ji} = 1$

23: **for** each job in sequenceListUpdated:

24: add constraints $y_{jk} = 1$

25: solution = Solve MIP Model

26: **for** i in T:

27: **for** j in N:

28: **if** $st_{ji} = 1\ in\ solution$:

29: add st_{ji} to assignedJobsList

30: **for** j in N:

31: **for** k in N:

32: **if** j not equal k AND $y_{jk} = 1$:

33: add $y_{jk}$to sequenceList

Output: Solution

5. BIN PACKING PROBLEM WITH ITEM FRAGMENTATION

This chapter presents the problem definition and notation in Section 5.1, and mathematical formulation in Section 5.2.

5.1 Problem Definition and Notation

In our second problem, there are situations where setup times are not dependent on the processing order of the jobs. When the sequence-dependent setup times are not considered in the problem, the sequence of the jobs is also not important as long as we keep complying with the due dates of each job. The problem transforms into a variant of the assignment problem where the order of assigned jobs in each day is insignificant. By removing the sequence-dependent setup from the scheduling problem, we can approach the problem as the bin packing problem, where the aim is to allocate fragmentable items with varying sizes to a fixed number of bins with varying capacities.

In the problem, the items correspond to tasks and bins refer to days. As such, this problem determines task-day assignments where the processing order of jobs within each day is not of interest. Furthermore, as some jobs may not need to start and end at the same day, the resulting problem is called the BPP with item fragmentation problem, which is a special case of the BPP. The proposed model assumes that a subset of the jobs is non-fragmentable referring that they have to be completed within a day. Thus, the remaining jobs are considered as fragmentable whose processing may start one day and finish in the next day. As opposed to the common assumption adopted in BPP with item fragmentation literature, hereby it is assumed that each fragmentable job can be divided into at most two parts. Finally, it is assumed that job operation times are deterministic and cannot exceed one day.

The objective of this research is to minimize capacity overruns (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. The problem assumes that jobs have predefined operation times and delivery deadlines, which are known in advance, and that the number of days (bins) for planning is fixed. The model assumes that each job can be

completed within one 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.

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

5.2 Mathematical Formulation

The mathematical model of the problem is given below.

Sets

N	$= \{1, \dots, n\}$ $Job\ set$	
T	$= \{1, \dots, t\}$ $Day\ set$	
JF	$fragmentable\ jobs\ set$	$JF \in N$
JNF	$non - fragmentable\ jobs\ set$	$JNF \in N$

$JF \cup JNF = N$

Parameters

P_j	$the\ processing\ time\ of\ job\ j$	$j \in N$
D_j	$the\ delivery\ time\ of\ job\ j$	$j \in N$
ns	$the\ time\ for\ a\ normal\ day\ shift$	
ot	$overtime\ for\ a\ working\ day$	
M	$a\ sufficiently\ large\ number$	

Ja_i	$jobs\ that\ can\ be\ assigned\ to\ day\ i$	$i \in T$
Jna_i	$jobs\ that\ cannot\ be\ assigned\ to\ day\ i$	$i \in T$
Da_j	$days\ that\ job\ j\ can\ be\ assigned\ to$	$j \in N$
Dna_j	$days\ that\ job\ j\ cannot\ be\ assigned\ to$	$j \in N$

Decision Variables

x_{ji}	$= \begin{cases} 1, if\ job\ j\ is\ assigned\ to\ day\ i \\ 0, otherwise \end{cases}$	$j \in N,\ i \in T$
xf_{ji}	$= \begin{cases} 1, if\ job\ j\ is\ fragmented\ to\ day\ i \\ 0, otherwise \end{cases}$	$j \in N,\ i \in T$
y_i	$= \begin{cases} 1, if\ overtime\ decision\ is\ made\ in\ day\ i \\ 0, otherwise \end{cases}$	$i \in T$
z_i	$fragmented\ time\ amount\ to\ day\ i$	$i \in T$

Objective

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

Constraints

$\sum_{j=1}^{n} xf_{j1} = 0$		(29)
$z_1 = 0$		(30)
$\sum_{i=1}^{Da_j} x_{ji} = 1$	$j \in N$	(31)
$\sum_{i=1}^{Dna_j} x_{ji} = 0$	$j \in N$	(32)
$\sum_{i=1}^{Da_j} xf_{ji} \leq 1$	$j \in JF$	(33)
$\sum_{i=1}^{Dna_j} xf_{ji} = 0$	$j \in JF$	(34)
$\sum_{i=1}^{t} xf_{ji} = 0$	$j \in JNF$	(35)
$\sum_{j=1}^{Ja_i} xf_{ji} \leq 1$	$i \in T/\{1\}$	(36)

$$\sum_{j=1}^{Jna_i} xf_{ji} = 0 \qquad i \in T/\{1\} \qquad (37)$$

$$x_{ji} \geq xf_{j(i+1)} \qquad j \in Ja_i,\ i \in T/\{t\} \qquad (38)$$

$$z_i \leq \sum_{j=1}^{n} xf_{ji} * P_j \qquad i \in T/\{1\} \qquad (39)$$

$$z_2 + y_1 * ot \geq \sum_{j=1}^{n} x_{j1} * P_j - nm \qquad (40)$$

$$y_t * ot \geq z_t + \sum_{j=1}^{n} x_{jt} * P_j - nm \qquad (41)$$

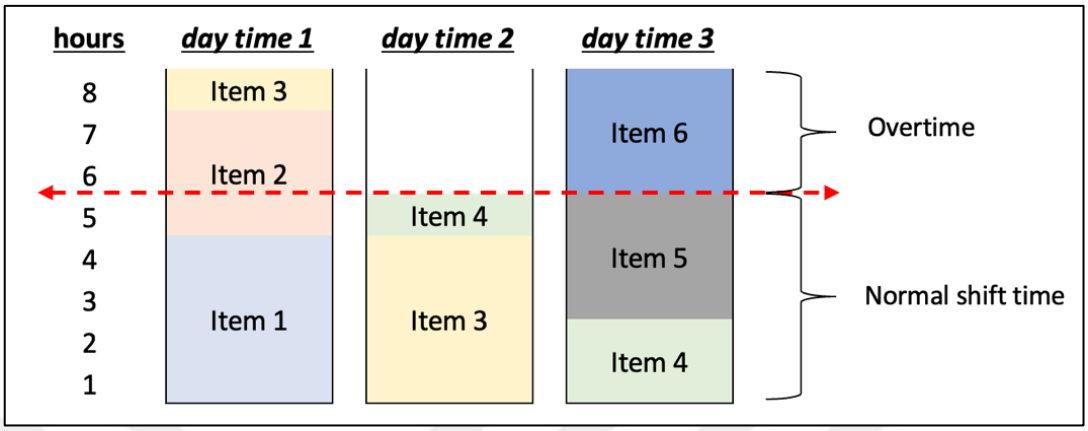
$$z_{(i+1)} + y_i \geq z_i + \sum_{j=1}^{n} x_{ji} * P_j - nm \qquad i \in T/\{1, t\} \qquad (42)$$

$$x_{ji}, xf_{ji}, y_i \in \{0,1\} \qquad j \in N, i \in T \qquad (43)$$

$$z_i \geq 0 \qquad i \in T. \qquad (44)$$

In this model, objective function minimizes the number of days where overtime decision is made over the planning horizon. Constraints (29) and (30) ensure that any job cannot be fragmented before the first day of the planning horizon. Constraints (31) and (32) 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 (33) and (34) state that if a job is fragmentable, it can be fragmented for the days that are earlier than its delivery dates. Constraints (35) guarantee not to fragment a job if it is a non-fragmentable job. Constraints (36) and (37) ensure that only one job can be fragmented among all jobs for the day. Constraints (38) state that a job has to be assigned to the day it is fragmented. Constraints (39) guarantee that the fragmented time of the day cannot be more than the processing time of the fragmented job. Constraints (40)-(42) ensure that each day cannot be loaded more than it is capacity. Constraints (40) control the capacity for the first day of planning horizon, while constraints (41) control the last day of the planning horizon. Constraints (42) control the capacity for remaining days of the planning horizon. Constraints (43) and (44) are domain constraints for decision variables

Figure 7 A feasible solution example for 6 items and 3 days problem



As seen in the example shared in Figure 7, since the total processing times of the jobs assigned to the 1st day exceed the capacity of that day, including overtime, the amount of work time that exceeds the capacity is transferred to the 2nd day. While determining the remaining capacity need for the 2nd day, the amount of work transferred from the 1st day is also considered. If the remaining capacity of normal shift time capacity is not enough for the scheduled day requirement, overtime is considered as additional capacity.

6. COMPUTATIONAL EXPERIMENTS

6.1 Data Sets and Experimental Settings

This chapter presents the results of computational experiments to assess the performance of the decomposition-based heuristics, the MIP formulation of the bin packing problem considered, and the MIP formulation of single machine scheduling with over time. Mathematical models were implemented 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. The decomposition heuristic algorithms were implemented in Python. We imposed a run time limit of 250 seconds for MIP models.

The performance of heuristic methods heavily depends on the problem instances and the specific problem settings. Therefore, it is crucial to evaluate the effectiveness of these methods on different problem instances with various characteristics. To do so, we conducted a study using 45 problem sets with different number of jobs and scheduling periods, each consisting of five randomly generated instances. Specifically, we considered problem instances with 8, 12, and 18 jobs for 2, 4, and 8-day scheduling periods. For each instances size, different number of fragmentable items are used. The test instances are summarized in Table 3 and number of fragmentable items for each instance are shown in column “FI”.

Table 4 Structure of instances

Jobs\Days	2	4	8	FI
8	*5 sets*	*5 sets*	*5 sets*	7
12	*5 sets*	*5 sets*	*5 sets*	9
18	*5 sets*	*5 sets*	*5 sets*	14

FI: number of fragmentable items total 45 instances

In our study, we generated random processing times for each job, and uniformly distributed the job due dates throughout the planning horizon. The fragmentable items are also randomly determined. We assumed negligible sequence-dependent setup times to be able to compare results

of MIP models. We used the decomposition-based heuristics and the proposed mathematical model to solve each problem set. We compared their performances in terms of the solution quality and computational efficiency.

The results of our study provide insights regarding the strengths and limitations of the four approaches and shed light on the potential trade-offs between solution quality and computational cost. By analyzing the experimental data, we can identify the factors that contribute to the success of each approach and provide recommendations for future research.

6.2 Results

As mentioned above, each problem set (number of jobs and number periods considered) consists of five examples. The results of each problem set are presented in the tables given below. Evaluating the results in two dimensions is essential to understand the performance of each solution approach. The first dimension is the length of the planning horizon, i.e., the number of days to be scheduled. The second dimension is the number of jobs to be scheduled.

The results for instances are presented in tables below. The "R" column in the tables indicates whether a solution was found within the specified time limit for MIP models. The "Obj" column presents the objective function value found for each solution approach, while the "St" column shows the solution time (in seconds) for each problem. The "OG" column presents the deviation of heuristic solution values from the optimal solution value. The solution times given for the heuristics are the sum of the solution times of each solved iteration. For the instances that give infeasible results before all iterations are completed, the total time until to that iteration is given as solution time.

The results obtained for 8 jobs - 2 days are shown in Table 5 For all instances, the single machine scheduling with overtime MIP (SMSOT) approach has found the optimal solution in an average of 1.676 seconds. Similarly, the optimal solutions were also obtained by decomposition-based heuristic 1 (DBH 1) in an average of 1.811 seconds, decomposition-based heuristic 2 (DBH 2) in 0.040 seconds, and the bin packing problem with item fragmentation MIP (BPPIT) in 0.0001

seconds. While DBH 2 and BPPIT solved the problem much faster than SMSOT, DBH 1 was able to solve the problem at a similar speed with SMSOT.

Table 5 Results for 8 jobs - 2 days instances

J8	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T2	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	OS	300	1.977	300	1.906	0%	300	0.027	0%	OS	300	0.000
2	OS	300	2.011	300	2.286	0%	300	0.046	0%	OS	300	0.000
3	OS	0	0.005	0	0.055	0%	0	0.003	0%	OS	0	0.000
4	OS	300	2.434	300	2.714	0%	300	0.072	0%	OS	300	0.001
5	OS	300	1.904	300	2.096	0%	300	0.050	0%	OS	300	0.000

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap

When the number of jobs is increased to 12 jobs with the same planning horizon, SMSOT could not find the optimal solution within the solution time limit. DBH 1 found solution in an average of 250.078 seconds, while the DBH 2 solution approach found the solution in an average of 0.780 seconds. On the other hand, BPPIT has reached the optimal solution for all instances in an average of 0.001 seconds as shown in Table 6.

Table 6 Results for 12 jobs - 2 days instances

J12	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T2	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	BS	300	250.019	300	250.199	0%	300	0.126	0%	OS	300	0.001
2	BS	300	250.017	300	250.075	0%	300	1.010	0%	OS	300	0.001
3	BS	300	250.016	300	250.092	0%	300	0.917	0%	OS	300	0.001
4	BS	300	250.017	300	250.013	0%	300	1.069	0%	OS	300	0.001
5	BS	300	250.028	300	250.011	0%	300	0.779	0%	OS	300	0.001

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, BS: best solution found so far

When we increase to the number of jobs to 18, similarly to the previous case, SMSOT could not reach the optimal solution within the time limit as shown in Table 7. However, DBH 1 found the solutions in average of 239.220 seconds when we consider only first three instances. For

remaining two instances, a solution couldn't find in time limit. DBH 2 has found the optimal solution in an average of 65 seconds, while BPPIT has reached the optimal solution in an average of 0.001 seconds.

Table 7 Results for 18 jobs - 2 days instances

J18	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T2	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	BS	150	250.001	150	252.930	0%	150	3.246	0%	OS	150	0.001
2	BS	300	250.008	300	207.976	0%	300	252.904	0%	OS	300	0.001
3	BS	150	250.004	150	256.755	0%	150	8.014	0%	OS	150	0.001
4	BS	150	250.017		255.480	N/A	150	5.386	0%	OS	150	0.001
5	NS		250.006		252.665	N/A	300	56.502	0%	OS	300	0.001

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, BS: best solution found so far, NS: no solution in limit time

The next set of problems consists of 4-day scheduling horizon. Table 8 presents the solutions obtained for 8 jobs. Out of 5 instances considered, two of them are found as infeasible. For the remaining jobs, SMSOT reached the optimal solution in 0.53 seconds, DBH 1 in 0.151 seconds, DBH 2 in 0.021 seconds, and BPPIT in 0.002 seconds.

Table 8 Results for 8 jobs - 4 days instances

J8	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T4	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	OS	600	0.522	600	0.220	0%	600	0.023	0%	OS	600	0.003
2	INF		0.740		0.229	N/A		0.025	N/A	INF		0.001
3	INF		0.295		0.008	N/A		0.006	N/A	INF		0.001
4	OS	300	0.623	300	0.081	0%	300	0.016	0%	OS	300	0.001
5	OS	600	0.446		0.057	N/A	600	0.023	0%	OS	600	0.002

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, INF: infeasible

When 12 jobs are considered, SMSOT failed to find the optimal solution within the time limit for all of the instances. However, as shown in Table 9, DBH 1(except instance 4), DBH 2, and BPPIT found the optimal solution in an average time of 4.702, 0.097, and 0.010 seconds,

respectively. One notable observation from the table is that DBH 1 had a deviation of 33% from the optimal solution for instance 4, although it took much less time compared to SMSOT. Despite the lower computation time, it achieved a higher value for the objective function.

Table 9 Results for 12 jobs - 4 days instances

J12	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T4	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	BS	600	250.007	600	3.897	0%	600	0.095	0%	OS	600	0.001
2	BS	600	250.006	600	4.132	0%	600	0.140	0%	OS	600	0.002
3	BS	600	249.962	600	3.891	0%	600	0.087	0%	OS	600	0.003
4	BS	450	250.004	600	9.410	33%	450	0.068	0%	OS	450	0.002
5	BS	450	250.010	450	6.888	0%	450	0.099	0%	OS	450	0.002

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, BS: best solution found so far

Finally, the results of instances with 18 jobs and 4-day planning horizon are presented in Table 10. SMSOT failed to find any feasible solution within the time limit for any of the instances. Except for instance 1, although the last iteration of the solution got stuck to the time limit DBH 1 found the solution the same as the optimum solution. On the other hand, DBH 2 found the optimal solution in an average time of 0.886 seconds, while BPPIT found it in an average time of 0.001 seconds.

Table 10 Results for 18 jobs - 4 days instances

J18	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T4	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	NS		250.031		459.050	N/A	600	0.803	0%	OS	600	0.001
2	NS		250.006	450	487.119	0%	450	0.326	0%	OS	450	0.002
3	NS		250.006	600	339.730	0%	600	1.170	0%	OS	600	0.001
4	NS		250.006	600	497.617	0%	600	1.892	0%	OS	600	0.001
5	NS		250.007	300	264.253	0%	300	0.238	0%	OS	300	0.002

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, NS: no solution in limit time

To evaluate the 8-day planning horizon, Table 11 presents the results obtained for 8 jobs. As the data were randomly generated, only two out of the five instances turn out to be feasible. SMSOT found the optimal solution in an average time of 1.043 seconds. For the same instances, DBH 1 and DBH 2 found the solutions in an average time of 0.116 and 0.072 seconds, respectively. However, both approaches yielded an objective function value that was twice the optimal solution for the first instance. BPPIT, on the other hand, found optimal solutions in an average time of 0.009 seconds.

Table 11 Results for 8 jobs - 8 days instances

J8	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T8	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	OS	300	0.756	600	0.093	100%	600	0.034	100%	OS	300	0.007
2	OS	750	1.330	750	0.139	0%	750	0.110	0%	OS	750	0.002
3	INF		0.030		0.054	N/A		0.037	N/A	INF		0.001
4	INF		0.029		0.031	N/A		0.019	N/A	INF		0.001
5	INF		0.029		0.034	N/A		0.018	N/A	INF		0.001

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, INF: infeasible

A similar trend can be observed for the instance size of 12 jobs as shown in Table 12. Only two instances are shown to be feasible. SMSOT failed to find the optimal solution within the time limit for the first instance. DBH 1 and DBH 2 provided feasible solutions within the time limit, while DBH 2 found a solution with a deviation of 17% from the optimal solution for the first instance. BPPIT found the optimal solution in an average time of 0.005 seconds.

Table 12 Results for 12 jobs - 8 days instances

J12	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T8	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	BS	900	250.017	900	2.194	0%	1050	0.096	17%	OS	900	0.004
2	OS	1050	57.065	1050	0.793	0%	1050	0.079	0%	OS	1050	0.005
3	INF		67.191		0.012	N/A		0.004	N/A	INF		0.007
4	INF		0.046		0.183	N/A		0.030	N/A	INF		0.001
5	INF		0.043		0.083	N/A		0.026	N/A	INF		0.002

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark

Obj: objective, St: solution time, OG: optimality gap, BS: best solution found so far , INF: infeasible

The solution results for the instance size of 18 jobs are shown in Table 13. Instances 1 and 3 are infeasible. SMSOT failed to find a solution within the time limit for all feasible problems. DBH 1 and DBH 2 found solutions for the feasible instances in an average time of 41.707 and 0.204 seconds, respectively. For similar instances, BPPIT found the optimal solution in an average time of 0.013 seconds.

Table 13 Results for 18 jobs - 8 days instances

J18	SMSOT MIP			DBH 1			DBH 2			BPPIF MIP		
T8	R	Obj	St	Obj	St	OG	Obj	St	OG	R	Obj	St
1	INF		250.010		42.611	N/A		0.229	N/A	INF		0.002
2	NS		249.990	900	91.584	0%	900	0.344	0%	OS	900	0.005
3	INF		250.007		0.018	N/A		0.009	N/A	INF		0.003
4	NS		250.008	600	23.689	0%		0.060	N/A	OS	600	0.023
5	NS		249.967	900	9.849	0%	900	0.207	0%	OS	900	0.011

SMSOT: single machine scheduling with overtime, DBH: decomposition-based heuristic,
BPPIF: bin packing problem with item fragmentation, OS: optimal solution found, R: remark
Obj: objective, St: solution time, OG: optimality gap, INF: infeasible, NS: no solution in limit time

7. CONCLUSIONS

This study aimed to minimize overtime required to complete jobs within specified due dates by addressing the Single Machine Scheduling Overtime problem. Problem was addressed as two sub-problem based on their sequence-dependent setup time requirement condition. For the first problem which considers sequence-dependent setup times, two decomposition-based heuristics were developed as solution approaches. In both heuristics, jobs were grouped based on their delivery deadlines and transformed into subproblems. Then, an iterative MIP-solving process was applied to each subproblem. After each iteration, the first heuristic fixed the assignment date of jobs, while the second heuristic fixed both the assignment date and the sequence of jobs. For the second problem which the sequence-dependent setup times were negligible, a MIP model was proposed to the problem as a bin packing problem with item fragmentation.

Our experiments revealed that the number of jobs scheduled per day had a significant impact on the performance of the MIP-based heuristics. The bin packing problem with item fragmentation model outperformed both MIP-based heuristics in terms of speed and optimality for cases where the sequence-dependent setup times were negligible. All experiments were conducted without considering sequence-dependent setup times. The proposed decomposition-based heuristics and an approach of the bin packing problem with item fragmentation model for scheduling proved to be effective solutions for the single machine scheduling with overtime problem. The results of this study provide insights for researchers and practitioners in scheduling problems with due dates and overtime constraints.

Future research endeavors could explore the application of existing methods to tackle scheduling problems that involve multiple machines. Furthermore, the performance of the MIP model for bin packing could be examined further to minimize the number of fragmented jobs. Despite extensive research on the BPP, several open problems and challenging variants still exist, including the BPP with uncertain item sizes, multiple criteria, time windows, and precedence relations. Furthermore, developing efficient algorithms for handling large-scale instances of the problem remains an active research area.

BIBLIOGRAPHY

Akkan, C. (1996). Overtime scheduling: an application in finite-capacity real-time scheduling. *Journal of the Operational Research Society*, *47*, 1137-1149.

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

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

Bülbül, Z. (2017). Sıra bağımlı ayar zamanı ve fazla mesai ile makine çizelgeleme. Master's thesis, TOBB University of Economics and Technology, Graduate School of Engineering and Science.

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

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

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

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.

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

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

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

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

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.

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.

Freeman, N. K., Mittenthal, J., & Melouk, S. H. (2014). Parallel-machine scheduling to minimize overtime and waste costs. *IIE Transactions*, *46*(6), 601-618.

Holloway, C. A., & Nelson, R. T. (1974). Job shop scheduling with due dates and overtime capability. *Management Science*, *21*(1), 68-78.

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

Jaramillo, F., & Erkoc, M. (2017). Minimizing total weighted tardiness and overtime costs for single machine preemptive scheduling. *Computers & Industrial Engineering*, *107*, 109-119.

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.

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

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

Li, X., Ventura, J. A., & Bunn, K. A. (2021). A joint order acceptance and scheduling problem with earliness and tardiness penalties considering overtime. *Journal of Scheduling*, *24*, 49-68.

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.

Matsuo, H. (1988). The weighted total tardiness problem with fixed shipping times and overtime utilization. *Operations Research*, *36*(2), 293-307.

Mestry, S., Damodaran, P., & Chen, C. S. (2011). A branch and price solution approach for order acceptance and capacity planning in make-to-order operations. *European Journal of Operational Research*, *211*(3), 480-495.

Pinedo, M. (2002). Scheduling: Theory, Algorithms and Systems, *Upper Saddle River, Prentice-Hall, NJ*

Ventura, J. A., & Yoon, S. H. (2013). A new genetic algorithm for lot-streaming flow shop scheduling with limited capacity buffers. *Journal of Intelligent Manufacturing*, *24*, 1185-1196.

Yang, B., Geunes, J., & O'Brien, W. J. (2004). A heuristic approach for minimizing weighted tardiness and overtime costs in single resource scheduling. *Computers & Operations Research*, *31*(8), 1273-1301.

Zobolas, G. I., Tarantilis, C. D., & Ioannou, G. (2008). Extending capacity planning by positive lead times and optional overtime, earliness and tardiness for effective master production scheduling. *International Journal of Production Research*, *46*(12), 3359-3386.

CURRICULUM VITAE

Mustafa Üstüncelik is a Planning & Control Assistant Manager with over 5 years of experience in the manufacturing industry. Currently working for Hidromek Construction Equipment (Thailand) Ltd. in Chon Buri, Thailand. He is responsible for managing the planning and control activities of the company, ensuring the timely delivery of products and services to customers, and optimizing production processes to achieve operational excellence. He has been in this role for nearly 2 years.

Before moving to Thailand, Mustafa spent over 3 years working for Hidromek in Ankara, Turkey, where he held the positions of Production Planning Team Leader and Production Planning Engineer. In these roles, he was responsible for managing the production planning and scheduling processes, ensuring that production plans were aligned with customer demand and capacity constraints, and leading a team of planners to achieve production targets. Prior to joining Hidromek, he had worked as a Production Planning Engineer for Formmetal Makina Sanayi A.Ş. in Ankara for 1.5 years.

His expertise also includes ERP project engineering, which he gained during his time at Aibis Bilişim A.Ş. in Eskişehir, Turkey, where he worked as an Enterprise Resources Planning Project Engineer for 1 year. In this role, he was responsible for managing the implementation of ERP systems for clients, ensuring that the systems met the clients' business requirements and were delivered on time and within budget.

He holds a bachelor's degree in industrial engineering from Eskişehir Osmangazi University. He also studied for an associate degree in business administration from Anadolu University. He is fluent in English and has a strong track record of achieving results through effective planning, communication, and collaboration with cross-functional teams.