AN APPLICATION OF UNRELATED PARALLEL MACHINE SCHEDULING WITH SEQUENCE-DEPENDENT SETUPS AT VESTEL ELECTRONICS

A Thesis

by

Merve Burcu Sarıkaya

Submitted to the

Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
August 2017

AN APPLICATION OF UNRELATED PARALLEL MACHINE SCHEDULING WITH SEQUENCE-DEPENDENT SETUPS AT VESTEL ELECTRONICS

Approved by:

Associate Professor Okan Örsan Özener,
Advisor,
Department of Industrial Engineering
Özyeğin University

Associate Professor Ali Ekici,
Department of Industrial Engineering
Özyeğin University

Associate Professor Serhan Duran,
Department of Industrial Engineering
METU

Date Approved: August 2017

To my family

ABSTRACT

Vestel Electronics produces LCD/LED televisions and has a significant market share in consumer electronics sector in Europe. TV manufacturing is planned based on a make-to order strategy, and Vestel uses 15 assembly lines to produce 110 different product groups and 3817 different models. Once the orders are received, production scheduling is performed at the beginning of each month, and the goal is to satisfy the demand on time as much as possible. Each order/job is processed on one of the compatible assembly lines, and preemption is not allowed.

In this thesis, we study the TV production scheduling operations at Vestel. The problem faced by Vestel is a variant of unrelated machine scheduling problem, and the objective is to minimize total tardiness. We propose a wide range of heuristics including a very simple sequential algorithm and a novel set partitioning-based approach. We test the heuristics on the real-life data and compare the solutions with the current practice. We observe up to 50% improvement in total tardiness.

Keywords: parallel machine scheduling; unrelated machines; sequence-dependent setups; tardiness

ÖZET

Vestel Elektronik Avrupa'da önemli bir pazar payına sahip olup ana ürün olarak LCD/LED televizyonlar üretmektedir. TV üretimleri siparişe dayalı olarak 110 farklı ürün grubu ve 3817 farklı model için 15 montaj hattında planlanmaktadır. Siparişler alındıktan sonra her ayın başında, talebi olabildiğince zamanında karşılayabilmek için üretim çizelgeleri oluşturulmaktadır. Her sipariş/iş uygun olan bir hatta üretilebilmektedir ve önceliğe izin verilmemektedir.

Bu tezde, Vestel'de üretim çizelgeleme operasyonlarını ele almaktayız. Vestel'in karşılaştığı problem, birbirinden bağımsız çeşitli makinalarda gecikmeyi minimum tutmayı amaçlayan çizelgeleme problemidir. Bu problem için çok basit ardışık algoritma ve küme ayrımı yaklaşımı da içeren çok çeşitli sezgisel yaklaşımlar önermekteyiz. Önerilen sezgisel yöntemler gerçek hayat verileri ile test edilmiş olup, mevcut uygulanan yöntem ile karşılaştırılmıştır. Toplam gecikmede %50 ye varan iyileştirmeler gözlenmiştir.

Anahtar kelimeler: paralel makine çizelgeleme; birbirinden bağımsız makinalar; sıra bağımlı ayar zamanları; gecikme

ACKNOWLEDGMENTS

I would like to thank my advisor Okan Örsan Özener for the continuous support, patience, guidance and motivation of my study and research. I have learnt so much from working with him.

I would like to thank also my family and especially my husband İlkay Sarıkaya for their unconditional patience in this journey.

TABLE OF CONTENTS

LIST OF TABLES

,

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Nowadays, production planning and scheduling are among the most important problems for manufacturing companies in order to respond rapid changing demands in a timely manner. Management of diversified demands has to be handled as a strategic problem, because meeting the requirements of the agreements made with the customers significantly affects the success of the business. Requirements of such an agreement may include certain quality standards, prices, due dates, etc. In order to meet the due dates, the companies have to effectively manage scarce resources including equipment and workforce. In this study, we analyze a real-life production planning and scheduling problem in Vestel Electronics company to develop solution approaches.

Vestel Electronics, located in Manisa, is one of the lead electronics manufacturing companies in Turkey. Vestel Electronics produces LCD/LED televisions and digital products that have 21% of the Europe market share in consumer electronics sector. Although Vestel Electronics produces a wide product range of products including TV, led lighting, set-top box, digital signage, smartphone, tablet and e-board, TVs account for the largest percentage of the production with 110 different product groups and 3817 different models. With around 4500 employees, the company produces 10 million TVs annually on average. The company delivers 90% of its final products to Europe, and 90% of the required semi-finished goods and raw materials are supplied from Far East. Vestel works with around 1500 different suppliers as illustrated on the map in Figure 1. The average time for order fulfillment is 20 days while average lead time of the materials incoming from Far East is 90 days.

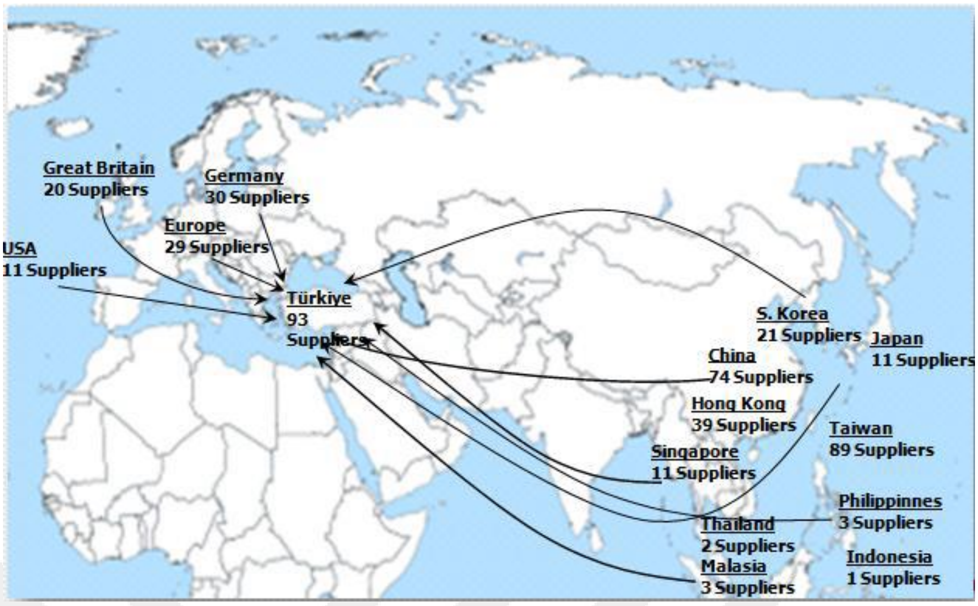

Figure 1: Locations of suppliers

Demand for TVs follows a seasonal pattern with peaks in autumn and winter. We provide the sales data of last six years at Vestel Electronics in Figure 2. In addition to fluctuations throughout the year, demand significantly increases when there are events such as war, crisis, sports organizations and promotions. Moreover, there is a rapid change in the technology, and hence, the product life cycles are getting shorter in consumer electronics sector. Such fluctuations in demand and short product life cycles force the companies to respond quickly to the changes in order to survive in the market.

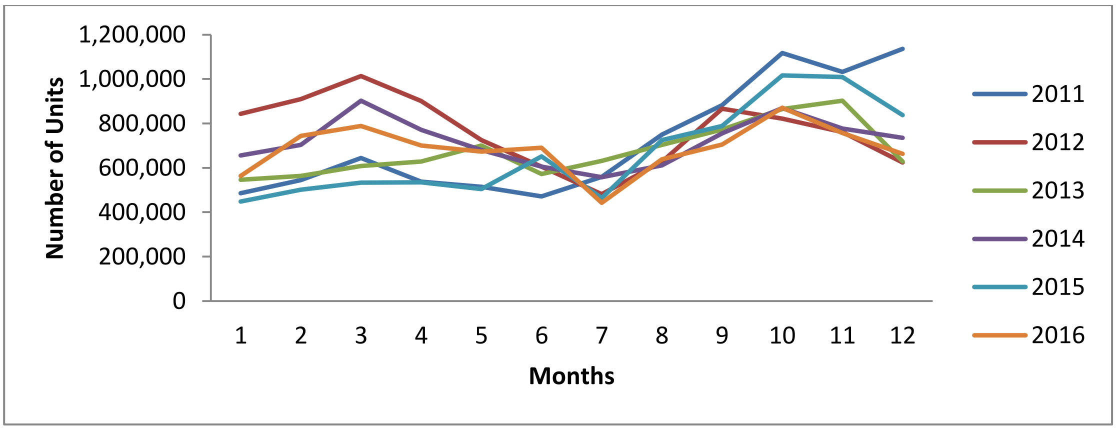

Figure 2: Seasonality of TV demand

At Vestel, TV manufacturing is planned based on a make-to-order (MTO) strategy and mass customization due to customers' demand diversification. Accordingly, the company utilizes multiple heterogeneous assembly/production lines that are specialized to produce TVs with different features including physical attributes such as size, color, design and variety of electronic options such as display frequency, USB/HDMI support, smart TV. Repeat order rate does not exceed 20%. Furthermore, 50% of the order sizes (lot sizes) are less than 500 units. This quantity corresponds to only three hours output of a production line, and it leads to frequent setups in the production process. Managing these setups between different runs is quite important for the efficiency of the production operations. Each setup results in downtime and affects the completion time of each job which in turn affects the final customer satisfaction. Satisfying the customer demand on time in such a dynamic environment and resulting improved customer satisfaction are very critical for the success of the business.

Since TVs are manufactured based on a make-to-order strategy at Vestel Electronics, production planning and scheduling system is triggered by customer requests. When customer requests are received, Demand Planning Department checks the available production capacity, material requirements, producibility of the products and requested delivery dates. After the negotiations (about delivery dates, prices, etc.) with the customer, a contract (including the product specifications, delivery date and price) is signed and the customer requests are converted to customer orders.

Although Vestel Electronics plan TV manufacturing using a make-to-order strategy, due to long lead times from the suppliers, company uses forecast for the next months in order to build a safety stock for the materials used in production. This helps the company deal with the uncertainty problem in a make to order setting especially in the presence of such long lead times. Since the forecasts always deviate from the actual values, managing the material procurement, with the additional wide range of products and model variety, becomes a challenging task.

It is the manufacturing facility's responsibility to complete the customer orders before the agreed delivery date while using the resources efficiently. It requires to consider several factors including product types ordered, order quantities, setup times and due dates. Two important factors that affect the TV manufacturing process are panel/cell availability and cardboard box supply. The most critical material supplied from Far East is the panel/cell whose specifications depend on the model to be produced. Due to long lead times, panel/cell procurement decisions significantly affect the TV manufacturing. Another critical material in TV manufacturing is the cardboard box used for customized packaging. These boxes are procured from local suppliers, and manufacturing cardboard boxes for customized products is a difficult and costly task.

At Vestel, production of TVs are planned in a monthly basis considering the customer orders and the availability of the two critical materials (cell and cardboard box). There are 15 assembly lines used for TV manufacturing. Each customer order is considered as a separate job and these jobs are completed on one of the compatible assembly lines. For a given job, only a subset of assembly lines (called *compatible* assembly lines) can be used to complete the job, and the total processing time of a job depends on the assembly line used for that job. A job can only be started after all the materials (especially cell and cardboard box) are available. Finally, before starting a new job on an assembly line, a setup time (depending on the previous job processed and the new job to be processed) is required to make the assembly line ready for production. The production plan is formed with the objective of completing the jobs on time as much as possible.

In this paper, we study the production planning problem for TVs at Vestel

Electronics. Our goal is to determine a production plan with minimum total tardiness while considering the job-assembly line compatibility, cell and cardboard box availability and the sequence-dependent setup times between jobs. We develop three solution approaches: (i) a sequential approach, (ii) a tabu search algorithm, and (iii) an iterative solution approach. In the sequential approach, we first assign the jobs to the assembly lines using a simple mathematical model, and solve the single assembly line ordering/sequencing problem for each assembly line. In the tabu search algorithm, we implement the classical tabu search algorithm using the assignment and ordering steps of the sequential approach as subroutines. In the iterative approach, we generate a large number of feasible solutions iteratively by implementing the sequential approach (after perturbing the objective coefficients) multiple times. Then, using the job-assembly line assignments in all those solutions we choose the best subset of assignments by solving a set partitioning problem. Finally, we implemented two variants of these approaches by combining the tabu search algorithm and the iterative approach. The hybrid approaching combining tabu search and the iterative approach in a way that the solutions found in the tabu search are used as candidate job-assembly line assignments in the set partitioning problem outperforms other approaches.

The rest of the paper is organized as follows. In Section 2, we review the related works in the literature. In Section 3, we provide a formal definition of the problem under consideration and present a mathematical model to find the optimal solution. In Section 4, we present our solution approach in order to solve more realistic large instances. We test the proposed model on real life instances provided by the company and compare the results against the real-life practice. We discuss the details of computational study in Section 5. Concluding remarks are provided in Section 6.

CHAPTER II

PREVIOUS WORK

Planning and scheduling problems are most commonly studied by researchers in the literature. In this section, we review the studies that consider minimizing the total (weighted) tardiness as the objective function.

The simplest version of the problem with minimum total (weighted) tardiness objective assumes that there is a single machine that can process all the jobs and each job is available at the beginning (Du and Leung, 1990; Holsenback and Russell, 1992; Lawler, 1977). The single machine total weighted tardiness problem (SMTWT) is consisted of each n jobs is to be processed without interruption on a single machine that can handle no more than one job at a time. Job i $(I=1,\ldots,n)$ has a positive weight $w_{i,}$ a due date d_i and a completion time c_i. Tardiness of job i (T_i) can be computed as $T_i=max\ |c_i\text{-}d_i;\ 0|$, while total weighted tardiness computation is $\sum_{i\in I} w_i T_i$.

In most of the studies, preemption is not allowed, and the machine can process only one job at a time. The variant with total tardiness minimization objective, zero setup times and zero/identical release times is shown to be an NP-hard problem in the ordinary sense (Du and Leung, 1990), whereas the variants with total weighted tardiness minimization objective (Lawler, 1977) or sequence-dependent setup times (Luo and Chu, 2006) or arbitrary release times (Rinnooy Kan, 1976) are shown to be strongly NP-hard.

Related studies in the literature focus on (i) dominance rules/properties (Akturk and Ozdemir, 2000, 2001; Su and Chen, 2006), (ii) exact branch-and-bound approaches which utilize lower, upper bound and dominance rules (Akturk and Ozdemir, 2000;

Baptiste et al., 2004; Bigras et al., 2008; Su and Chen, 2006), and (iii) heuristic approaches including greedy randomized adaptive search procedure (Gupta and Smith, 2006), simulated annealing (Tan and Narasimhan, 1997), genetic algorithm (Rubin and Ragatz, 1995).

We refer the reader to Koulamas (2010) for a recent review of single-machine total tardiness scheduling problem with zero setup times and identical release times. Besides, in many of papers about SMTWT problems, setup is assumed as negligible and committed as part of the processing time. In almost more than 90 percent of the scheduling surveys, setup has been ignored (Allahverdi, 2015). Allahverdi et al. (2008) and Allahverdi (2015) provide extensive review of literature on scheduling problems with setup costs including single-machine total tardiness problem. In addition, release date (r_i) can be admitted to be the available time of processing for job i. Akturk et al. (2000) developed a branch and bound algorithm to solve SMTWT problem with unequal release dates referring dominance properties and heuristics approach for large size problems. Chen et al. (2006) studied on same problem with same method. They showed that if release dates are equal, the problem can be solved with branch and bound algorithm for larger size problems. They also designed an algorithm based on dominance rule for SMTWT with unequal release dates. Baptiste et al. (2004) studied on same problem for 500 jobs instances with same solution method; branch and bound and dominance properties to reduce optimal search space. Regarding due dates, Gupta et al. (2006) observed that problems with narrow due date range are relatively hard to solve for their proposed greedy algorithm heuristics.

Multi-machine case of the problem, also called parallel machine scheduling, is extensively studied in the literature. Parallel machine scheduling problems can be classified into three main categories depending on the characteristics of the machines:

(i) identical machines, (ii) uniform machines, and (iii) unrelated machines. In identical parallel machine scheduling, the machines are identical, and hence, the processing time of a job and the setup time (if any) between two consecutive jobs is same for all machines. In uniform parallel machines case, machines have different speeds, and processing times of a job differs by speed factors. Finally, in the case of unrelated machines, the processing times are arbitrary and have no special characteristics.

Similar to single-machine case, most of the studies assume that preemption is not allowed, one machine can process a single job at a time and each job can be processed on one machine only. Alidaee and Rosa (1997) consider the identical parallel machines case where all the jobs are available at the beginning and there is no setup time. Their goal is to minimize total tardiness. They propose an algorithm that uses the modified due date algorithm for the single machine case (Baker and Bertrand, 1982) as a subroutine. Yalaoui and Chu (2002) study the same setting and develop a branch-and-bound algorithm. Similarly, Shim and Kim (2007) propose a branch-and bound algorithm using some new dominance properties, lower bounds and upper bounds found by a heuristic algorithm. Finally, Biskup et al. (2008) develop a heuristic algorithm to find near-optimal solution for instances with up to 200 jobs and 5 machines in less than a minute.

Armentano and Filho (2007) study uniform parallel machines case when there are sequence dependent setups. Each job has a certain release date, and the objective is to minimize the total tardiness. The authors propose a greedy random search adaptive procedure (GRASP) that incorporates adaptive memory-based approaches. Lin et al. (2011) and Chen (2009) consider the same setting. Lin et al. (2011) develop a simple iterated greedy heuristic, whereas Chen (2009) propose a heuristic based on the apparent-tardiness-cost-with-setup procedure, the simulated annealing method and designed improvement procedures. Balakrishnan et al. (1999) and Sivrikaya-Serifoglu and Ulusoy (1999) consider a different objective function. Their objective is to minimize the weighted earliness and/or tardiness. Balakrishnan et al. (1999) test the effectiveness of a compact mathematical model. Sivrikaya-Serifoglu and Ulusoy (1999) propose two genetic algorithms to find near-optimal solutions.

Unrelated parallel machine scheduling is the most related one to our study. Liaw et al. (2003) study the unrelated parallel machine scheduling problem assuming setup times between consecutive jobs are zero. They assume that all the jobs are available at the beginning, and they look for a minimum weighted tardiness solution. A branch-and-bound algorithm that can handle problems with up to 18 jobs and 4 machines is proposed. Logendran et al. (2007) study the same setting with sequence- and machine-dependent setups and unequal release times for the jobs. They further assume that each machine has an availability constraint which sets the earliest time a machine can be used for processing jobs. Six different search algorithms based on tabu search are developed to identify the best schedule. Lee et al. (2013) also study the unrelated parallel machine setting where jobs have sequence- and machine-dependent setups. Different from Logendran et al. (2007), they assume that all the jobs are available at the beginning, and the objective is to minimize total tardiness. The authors propose a tabu search algorithm that incorporates various neighborhood generation methods. Similarly, Zhu and Heady (2000) and Akyol and Bayhan (2008) consider unrelated parallel machine scheduling problem with sequence-dependent setups and equal release times. Different from studies above, their objective is to minimize the total weighted earliness and tardiness. Zhu and Heady (2000) propose a mixed integer programming formulation, and Akyol and Bayhan (2008) develop a neural network approach to address the problem. Finally, Zhang et al. (2007) consider the unrelated parallel machine setting with sequence-dependent setup times, unequal release times and machine-job compatibility restrictions. Their objective is to minimize the total weighted tardiness. They convert the problem into reinforcement learning problems by constructing a semi-Markov decision process and then apply the Q-Learning algorithm to find a solution. Similar to single-machine case, Allahverdi (2015) and Allahverdi et al. (2008) provide recent review of the literature on scheduling problems (including parallel machine scheduling) with setup times/costs.

CHAPTER III

PRODUCTION PLANNING AND SCHEDULING PROBLEM

3.1 Problem Definition

In this section, we provide a formal definition of the problem and present an integer programming formulation. Problem is indicated to complete customer orders at the requested due dates while considering setup times with multi assembly lines for TV production. It is aimed to develop a flexible production scheduling system that serves to customer orders with assigned assembly lines for a planning horizon in a make to order environment. Customers request many divergent models of products in short time period while there are many producers in the market; as a result of that, it is a necessity to meet customer demands on time. Therefore, from not only production and quality perspective, all the divisions of the company, especially production planning department should work on optimizing of the plans to meet customer orders at the promised time level and effectively. But scheduling problems are not easy to solve in reasonable time.

The assumptions and decisions of production planning and scheduling problem are presented below.

Assumptions

- Arrival dates of the materials, due dates and processing times, producibility of the orders in each line, setup times and order quantities are deterministic and known in advance.
- Processing times depend on assembly line assignment.

- Multiple models may contain different materials and require different processing times.
- Assembly lines are always available and never break down and have to be start at a same particular time.
- Material stock-outs is not considering except of two main materials; cells and carton boxes.
- Each assembly line can process at most one order at any time.
- Ready times of all orders are different from zero but deterministic; known in advance.
- No pre-emption is allowed, once a production started of an order, it is continued until completion.
- Setup times are known in advance and sequence dependent. Depends on cell and product group variance.
- Assembly line capacity is defined by the quantity of orders processed by a line at a particular time.

Decisions

- Assignment decisions are for determination which production order should be assigned to which assembly line and in which sequence.
- Sequencing decisions address precedence and successor of the orders.
- Timing decisions are for determining starting and finishing time of the orders.

In following sections, it is aimed to design a mathematical model for solving this complicated production scheduling problem.

3.2 *Mathematical Model Formulation*

We have n assembly lines and m jobs to be processed on one of these assembly lines. We use $L(:= \{1,2,...,n\})$ to denote the set of assembly lines and $I(:= \{1,2,...,m\})$ to denote the set of jobs. Job i can only be processed on a subset of the assembly lines. We use L_i to denote the set of assembly lines job i can be assigned to and I_l to denote the set of jobs that can be assigned to assembly line l. Processing time of a job depends on the assembly line it is assigned to. We denote the processing time of job i on assembly line l by p_{il}. For the assembly lines that cannot process job i, p_{il} is set to a very large number. When job i is processed immediately after job j on the same assembly line, then a sequence-dependent setup time t_{ij} is required to make the assembly line ready for processing job j. Each job has a certain due date d_i by which the job has to be completed. Job i can be started on an assembly line after its release date. Moreover, the two critical materials (cells and cardboard boxes) specific to each job have to be ready before a job can be started. Hence, the earliest time a job can be started is the maximum of the release time of the job, the available time of the cells and the available time of the cardboard boxes required for that job. We denote the earliest start time of job i by r_i. Finally, in order to maintain a balance between the workload of the assembly lines, Vestel imposes lower and upper limits on the number of jobs that can be assigned to an assembly line. We use C_1 and C_2 to denote these lower and upper limits, respectively. Our goal is to find an assignment of the jobs to the assembly lines and processing order of the jobs on each assembly line with the objective of minimizing the total tardiness of the jobs.

Next, we develop an integer programming model for the problem. The decision variables used in the model are as follows:

y_{ilk} : 1, if job i is processed on assembly line l at the k^{th} order;
0, otherwise $i,k \in I,\ l \in L$

x_{ij} : 1, if job i is immediate predecessor of job j;
0, otherwise $i,j \in I$

s_i : start time of job i $i \in I$

f_i : completion time of job i $i \in I$

u_i : amount of tardiness for job i $i \in I$

Using these decision variables, the mathematical model proposed for the problem under consideration is as follows:

$$\text{Min} \quad \sum_{i \in I} u_i \tag{1}$$

s.t.

$$\sum_{l \in L_i} \sum_{k \in I} y_{ilk} = 1 \qquad \forall i \in I \tag{2}$$

$$\sum_{i \in I_l} y_{ilk} \leq 1 \qquad \forall l \in L, k \in I \tag{3}$$

$$\sum_{i \in I_l} y_{ilk} - \sum_{i \in I_l} y_{il,k-1} \leq 0 \qquad \forall l \in L, k \in I \backslash \{1\} \tag{4}$$

$$y_{ilk} + y_{jlk-1} - x_{ij} \leq 1 \qquad \forall i,j \in I, k \in I \backslash \{1\} \tag{5}$$

$$\sum_{i \in I_l} \sum_{k \in I} y_{ilk} \geq C_1 \qquad \forall l \in L \tag{6}$$

$$\sum_{i \in I_l} \sum_{k \in I} y_{ilk} \leq C_2 \qquad \forall l \in L \tag{7}$$

$$s_i \geq r_i \qquad \forall i \in I \tag{8}$$

$$s_j - f_i + M(1 - x_{ij}) - t_{ij} x_{ij} \geq 0 \qquad \forall i,j \in I \tag{9}$$

$$f_i - s_i - \sum_{l \in L_i} \sum_{k \in I} p_{il} y_{ilk} \geq 0 \qquad \forall i \in I \tag{10}$$

$$f_i - u_i \leq d_i \qquad \forall i \in I \tag{11}$$

$$y_{ilk} \in \{0,1\} \qquad \forall i,k \in I, l \in L \tag{12}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in I \tag{13}$$

$$s_i, f_i, u_i \geq 0 \qquad \forall i \in I \tag{14}$$

In this model, the objective is to minimize the total tardiness of the jobs. Constraints (2) make sure that each job is assigned to one of the assembly lines. No two jobs can be assigned to the same order/sequence of an assembly line. This is enforced by Constraints (3). A job can be processed on assembly line at a certain order if all the prior orders are assigned a job. This is guaranteed by Constraints (4). When two jobs are processed at consecutive orders on an assembly line, then the corresponding decision variable x_{ij} should be equal to 1. This is enforced by Constraints (5). The upper and lower limits on the number of jobs that can be assigned to an assembly line is imposed by Constraints (6) and (7). Constraints (8) impose the earliest start time restriction. Start time of a job cannot be earlier than the end time of the preceding job. Constraints (9) enforce this restriction. The completion time of a job is determined by Constraints (10). Constraints (11) determine the tardiness of each job. Finally, Constraints (12-14) impose the nonnegativity and binary restrictions. This mathematical model can only handle small instances. To find solutions for real-life instances in a reasonable amount of time, we develop heuristic algorithms.

CHAPTER IV

SOLUTION APPROACH

In order to address the TV production scheduling problem of Vestel, we propose four different solution approaches. These approaches are named following;

- Sequential Approach (SA)
- Tabu Search Algorithm (TSA)
- Random Set Partitioning Approach (RSPA)
- Tabu-Based Set Partitioning Approach (TSPA)

In Sequential Approach (SA), we first assign the jobs to the assembly lines using a simple assignment model, and then determine the processing sequence for each assembly line using a mathematical model. In Tabu Search Algorithm (TSA), we implement the tabu search idea that uses the Sequential Approach (SA) as a subroutine. In Random Set Partitioning Approach (RSPA), we generate multiple feasible solutions for the problem using a randomized procedure. RSPA also uses the steps of SA as subroutines. Then, we solve a Set Partitioning Problem to select the best schedule for each assembly line among the schedules generated randomly. In Tabu-Based Set Partitioning Approach, we integrate the TSA with RSPA so that multiple feasible solutions that are fed into the Set Partitioning Problem are generated using TSA. Next, we explain the details of the proposed approaches.

4.1 Sequential Approach

In the sequential approach, we decompose set of decisions to be made into two, and make one set of decisions at each stage. More specifically, in the first phase we assign the jobs to the assembly lines. Then, for each assembly line we determine processing order of the jobs assigned to it. In each phase, we make the decisions by solving mathematical models.

In the first phase, we define a single type of decision variable that determines which job is assigned to which assembly line and show the mathematical model named as $\mathbf{MIP-A}$. Our objective in this phase is to minimize the total processing of the jobs. We also impose the lower and upper limits on the number of jobs that can be assigned to an assembly line. We use the following decision variable:

y_{il} : 1, if job i is assigned to assembly line l
0, otherwise $\quad i \in I, l \in L$

The mathematical model solved at the first phase is as follows:

$$\text{MIP} - \text{A:} \quad \text{Min} \quad \sum_{l \in L} \sum_{i \in I_l} p_{il} y_{il} \tag{15}$$

s.t.

$$\sum_{l \in L_i} y_{il} = 1 \qquad \forall i \in I \tag{16}$$

$$\sum_{i \in I_l} y_{il} \geq C_1 \qquad \forall l \in L \tag{17}$$

$$\sum_{i \in I_l} y_{il} \leq C_2 \qquad \forall l \in L \tag{18}$$

$$y_{il} \in \{0,1\} \qquad \forall i \in I, l \in L \tag{19}$$

In this model, the objective function minimizes the total processing times of the jobs. Constraints (16) make sure that each job is assigned to an assembly line. Constraints (17) and (18) are the lower and upper limits on the number of jobs that can be assigned to an assembly line. Constraints (19) are the sign restrictions. By solving this model, we determine a feasible assignment of jobs to the assembly lines.

Then, in the second phase we decide on the order/sequence of the jobs are processed on each assembly line and show the mathematical model named as $\mathbf{MIP-S}$. Let A_l be the set of jobs assigned to assembly line l. Then, for assembly line l, we define the following decision variables:

y_{ik}	: 1, if job i is processed at the k^{th} order 0, otherwise	$i \in A_l, k \in \{1,2,\ldots,\lvert A_l\rvert\}$
x_{ij}	: 1, if job i is immediate predecessor of job j 0, otherwise	$i,j \in A_l$
s_i	: start time of job i	$i \in A_l$
f_i	: completion time of job i	$i \in A_l$
u_i	: amount of tardiness for job i	$i \in A_l$

Then, we solve the following model to determine processing order with minimum tardiness:

$$\mathrm{MIP-S:}\quad \mathrm{Min} \qquad \sum_{i \in A_l} u_i \tag{20}$$

s.t.

$$\sum_{k \in \{1,2,\ldots,|A_l|\}} y_{ik} = 1 \qquad \forall i \in I \tag{21}$$

$$\sum_{i \in A_l} y_{ik} \leq 1 \qquad \forall k \in \{1,2,\dots,|A_l|\} \tag{22}$$

$$y_{jk} + y_{i,k-1} - x_{ij} \leq 1 \qquad \forall i,j \in Al, k \in \{2,\dots,|A_l|\} \tag{23}$$

$$s_i \geq r_i \qquad \forall i \in A_l \tag{24}$$

$$s_j - f_i + M(1 - x_{ij}) - t_{ij}x_{ij} \geq 0 \qquad \forall i,j \in A_l \tag{25}$$

$$f_i - s_i - p_{il} \geq 0 \qquad \forall i \in I \tag{26}$$

$$f_i - u_i \leq d_i \qquad \forall i \in A_l \tag{27}$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in A_l, k \in \{1,2,\dots,|A_l|\} \tag{28}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in A_l \tag{29}$$

$$s_i, f_i, u_i \geq 0 \qquad \forall i \in A_l \tag{30}$$

We solve this model for each assembly line separately. The assignment phase is very important for becoming first phase of production scheduling model and affects the results of line sequencing phase. Therefore we have to ensure that we made a true assignment at the first time. Since this problem is hard to solve in a reasonable period of time with large size of instances and ensure the first assignment done correctly, we propose to use heuristic algorithms.

4.2 Tabu Search Algorithm

Tabu search is a metaheuristic method for solving combinatorial optimization problems to find a near optimal solution in a timely manner. It is not guaranteed that to find global optimum solution with tabu search but it is widely and successfully used to solve huge scheduling problems. Tabu search starts searching with an arbitrary initial solution and examine each of the solutions in the neighbourhood and select best of them to move current solution to another point in the solution space. It continues searching while absence of improving moves without falling back into a local optimum found before. In tabu search new solution may be accepted if it is worse than the previous one. So it uses a tabu list (forbidden list) to prevent cycling around recent solutions similar to human memory. When a tabu move would result better solution than any other moves, its tabu classification can be overridden. It is called as aspiration criterion which allows such a condition to allow overridden.

In our proposed approach for TSA, we start with the two steps of SA. Then, at every iteration, we modify the current solution by avoiding some of the job-assembly line assignments in the first phase of SA. By doing this multiple times, we hope to obtain a better solution. Several different stopping criteria used to stop this approach such as a certain number of iterations, predetermined amount of CPU time or reference objective function values found by another method. Since we solve the two models proposed for SA multiple times in TSA, we limit the run time of each model (especially, the processing order determination model in the second phase) in order to keep the total run time low. However, after we obtain the best solution, we run the second model for a longer time to improve the final solution. The pseudocode for TSA is provided in Algorithm 1.

Algorithm 1 $TabuSearchAlgorithm(IterLim, DivIterLim, TabuListSize, ShortTabuListSize)$

Set $BestTardiness = +\infty$, $DivergeIterationCount = 0$
for $j = 1$ to $IterLims$ **do**
  Implement Sequential Approach
  Calculate the total tardiness: $CurrentTardiness$
  if $CurrentTardiness < BestTardiness$ **then**
    Set $DivergeIterationCount = 0$ and update best solution: $BestTardiness = CurrentTardiness$
    Order the jobs in the current solution according to the tardiness starting from the highest
    Take the first m_1 ($m_1 = \min\{ShortTabuListSize, 5\}$) jobs from the list and prohibit the current job-assembly line assignment for them with probability p_1 if they have positive tardiness
    Take the next m_2 ($m_2 = \min\{ShortTabuListSize - m_1, 5\}$) jobs from the list and prohibit the current job-assembly line assignment for them with probability p_2 if they have positive tardiness
    Implement Sequential Approach
    Calculate the total tardiness: $CurrentTardiness$
    if $CurrentTardiness < BestTardiness$ **then**
      Update best solution: $BestTardiness = CurrentTardiness$
    end if
  else
    $DivergeIterationCount++$
    if $DivergeIterationCount > DivIterLim$ **then**
      Order the jobs in the current solution according to the tardiness starting from the highest
      Take the first $TabuListSize$ jobs from the list and prohibit the current job-assembly line assignment for them if they have positive tardiness
    else
      Take the first m_1 ($m_1 = \min\{TabuListSize, 5\}$) jobs from the list and prohibit the current job-assembly line assignment for them with probability q_1 if they have positive tardiness
      Take the next m_2 ($m_2 = \min\{TabuListSize - m_1, 5\}$) jobs from the list and prohibit the current job-assembly line assignment for them with probability q_2 if they have positive tardiness
      Take the next m_3 ($m_3 = TabuListSize - m_1 - m_2$) jobs from the list and prohibit the current job-assembly line assignment for them with probability q_2 if they have positive tardiness
    end if
    Implement Sequential Approach
    Calculate the total tardiness: $CurrentTardiness$
    if $CurrentTardiness < BestTardiness$ **then**
      Update best solution: $BestTardiness = CurrentTardiness$
    end if
  end if
end for

4.3 Random Set Partitioning Approach

In Random Set Partitioning Approach (RSPA), we generate multiple production schedules for each assembly line, and try to choose the best combination of schedules overall by solving a Set Partitioning Problem. We generate the schedules to choose by perturbing the original problem. We use MIP-A model in the first phase of SA to generate these schedules. More specifically, we modify the processing times of the jobs, and resolve the MIP-A model at each iteration. We form the schedule for each assembly line by solving a time-restricted version of MIP-S. Then, due to high number of solutions considered, we solve a series of Set Partitioning Problems to find the best solution.

We use Γ_l to denote the schedules formed for assembly line l. For each $\gamma in \Gamma_l$, we use $I\gamma$ to denote the jobs in schedule γ and u_γ to denote the total tardiness of the jobs in schedule gamma. We choose the best schedule combination for the assembly lines while guaranteeing that each job is assigned to an assembly line. We do this by solving a Set Partitioning Problem. The decision variables are as follows:

z_γ : 1, if schedule γ is chosen
0, otherwise $\quad \gamma \in \Gamma_l,\ l \in L$

The mathematical model solved is as follows:

$$\text{SPP:} \quad \text{Min} \qquad \sum_{l \in L} \sum_{\gamma \in \Gamma_l} u_\gamma z_\gamma \tag{31}$$

s.t.

$$\sum_{\gamma \in \Gamma_l} z_\gamma = 1 \qquad \forall l \in L \tag{32}$$

$$\sum_{\gamma \in \{\gamma | i \in I\gamma\}} z_\gamma = 1 \qquad \forall i \in I \tag{33}$$

$$z_\gamma \in \{0,1\} \qquad \forall \gamma \in \Gamma_l, l \in L \tag{34}$$

In SPP, the objective is to minimize total tardiness of the chosen schedules. Constraints (32) make sure that a schedule is chosen for each assembly line. Constraints (32) guarantee that each job is included one of the chosen schedules. Constraints (34) are the sign restrictions.

The steps of RSPA are provided in Algorithm 2.

Algorithm 2 $RandomSetPartitioningApproach(IterLim, SolLimit, MaxPerturb)$

Set $BestTardiness = +\infty$
for $j = 1$ to $IterLim$ **do**
  Choose a random integer between 0 and $MaxPerturb$: $NumberPerturbed$
  Choose $NumberPerturbed$ number of jobs randomly and modify their processing times (for each assembly line) by a percentage randomly generated between -50% and +50%
  Implement Sequential Approach
  Store the solution found and calculate the total tardiness
end for
$NumberRemaining = IterLim$
while $NumberRemaning > 0$ **do**
  Take $SolLimit$ number of solutions from the solutions generated (including all the schedules for each assembly line)
  $NumberRemaning = NumberRemaning - SolLiimit$
  Solve SPP with these solutions and the best solution found so far
  Store the best solution
end while

Similar to TSA, we run the MIP-S model at the end of RSPA by increasing the time limit in order to improve the final solution.

4.4 Tabu-Based Set Partitioning Approach

In Tabu-Based Set Partitioning Approach, we integrate the Tabu Search Algorithm and Random Set Partitioning Approach. Instead of randomly perturbing the problem in order obtain multiple solution in RSPA, we use the solutions constructed at each iteration of TSPA and solve the series of Set Partitioning Problems with these solutions. The parameters used in TSPA are (i) IterLimit (the number of iterations in TSA, (ii) DivIterLim (the number of iterations performed to obtain divergent solutions), (iii) TabuListSize (the number of the prohibited assignments in case of a nonimproving solution, (iv) ShortTabuListSize (the number of prohibited assignments in case of an improving solution), and (v) SolLimit (the number of solutions/schedules fed into SPP model at each iteration).

CHAPTER V

COMPUTATIONAL STUDY

In this section, we present computational results for proposed algorithms to solve our production scheduling problem. Firstly, we describe problem instances tested and then we evaluate the effectiveness of the proposed algorithms on real-life instances.

5.1 Real-World Problem Instances

We collect data for 10 different months. Each different instance show different orders to be produced per month. At Vestel, we have 15 assembly lines dedicated for TV manufacturing. For each of the 10 instances, we have 150 jobs to be processed on one of these assembly lines. Processing times of the jobs vary between half of an hour and 1 day. In terms of assembly line-job compatibility, depending on the type of the job, it can be processed on 1 to 13 assembly lines. On average, a job can be processed on around 7 assembly lines. Release dates are around 15-20 days before the due dates. Due dates are provided by Demand Planning Department. We analyze average production completion times of TVs that produced in one year and it is approximately 2,26 days for each job. Figure 3 illustrates average monthly distribution of production completion times for a job.

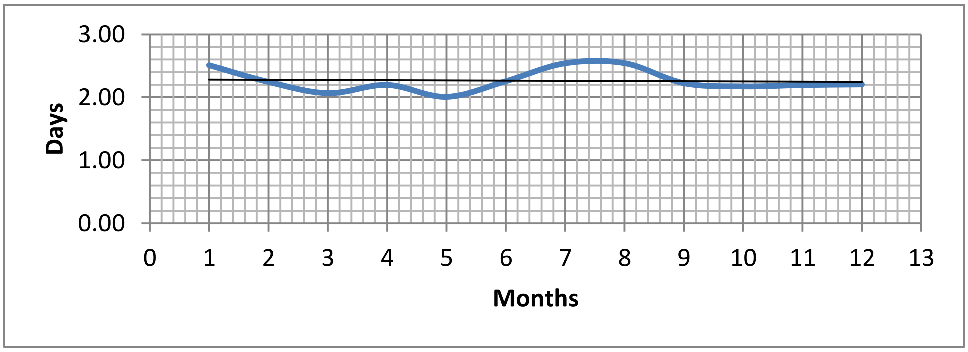

Figure 3: Distribution of production completion times

Setup times are known in advance but changes depending on sequence of jobs. It vary around 15 minutes to 30 minutes according to variance of cell and product groups. Our monthly instances have jobs with around 20 different cells and 14 different product groups.

Currently, Vestel uses a customized software which works on a network-based approach to schedule jobs in practice. This software takes the orders created in the Enterprise Resource Planning (ERP) software used at Vestel. Using the user-interface of this customized software, the user assigns the jobs to the lines manually considering the setup times and tardiness. Experience of the user is significantly important in the current practice.

5.2 Numerical Analysis of the Methods

All the computational experiments are carried out on a 64-bit Windows Server with two 2.4 Ghz Intel Xeon CPU's and 24 GB RAM. The algorithms are implemented using C++ and CPLEX Concert Technology.

We compare the performances of the proposed approaches against the current practice at Vestel in Table 1. In this table, we present the percentage improvements in the total tardiness taking the solution found in the current practice as the reference point. The first column specifies the instance number used in the computational study. Under "SA", we present the improvements obtained by Sequential Algorithm (SA). We implement TSA, RSPA and TSPA with 1000 overall iterations. Under "TSA", we present for three different implementations of TSA. In all those three implementations, the size of the short tabu list is set to 10. In "(5, 20)", the number of divergence iterations is set to 5, and the size of the tabu list is set to 20. Similarly, for the other two implementations the values taken for these two parameters are provided at the top of the next two columns. In RSPA, number of solutions fed into set partitioning problem is set to 250, and the maximum number of modified processing times to change the assignment solution is set to 10. Finally, we test TSPA with three different implementations. In all those implementations, we set the size of the short tabu list to 10, and the number of solutions in the set partitioning problem is set to 250. Under "TSPA" column, we provide values for the number of divergence iterations and the tabu list size for the three implementations.

Table 1: Comparison between the proposed algorithms and the current practice

Instance	SA	TS (1000 iterations)			RSPA	TSPA (1000 iterations)		
		(5, 20)	(5, 30)	(10, 40)	(1000, 250)	(5, 20)	(5, 30)	(10, 40)
1	30.92%	72.00%	74.22%	69.37%	57.91%	74.17%	77.27%	79.21%
2	38.60%	68.51%	68.51%	68.51%	71.44%	71.54%	71.60%	71.55%
3	69.60%	73.74%	72.54%	72.16%	72.74%	77.67%	75.51%	76.98%
4	-1.30%	39.24%	43.13%	45.29%	58.97%	67.23%	59.77%	61.13%
5	3.69%	15.93%	10.19%	14.10%	13.98%	22.31%	22.26%	21.63%
6	12.73%	20.35%	18.74%	20.10%	25.48%	27.77%	26.15%	28.07%
7	20.81%	39.88%	39.53%	31.60%	28.73%	42.43%	42.14%	43.44%
8	19.85%	25.30%	22.62%	25.50%	25.23%	28.32%	28.16%	30.09%
9	14.47%	38.76%	34.67%	34.21%	33.25%	42.82%	45.00%	42.80%
10	5.52%	41.66%	47.84%	47.65%	47.44%	44.85%	47.59%	46.31%
Average	**21.49%**	**43.54%**	**43.20%**	**42.85%**	**43.52%**	**49.91%**	**49.55%**	**50.12%**

We observe that all proposed algorithms provide significantly better results compared to the current practice. Even the Sequential Approach (SA) provides an improvement of around 21% on average in terms of total tardiness. Although Tabu Search Algorithm (TSA) and Random Set Partitioning Approach (RSPA) provide higher improvements of around 43% when compared to the current practice, the novel idea of integrating tabu search and set partitioning approach (TSPA) provides the best results on average. TSPA provides results with 50% less total tardiness compared to the actual schedules constructed in practice. Except one instance (Instance #10), TSPA finds the best solution. Among implementations of TSPA with different parameters, the one with 1000 overall iterations, 5 divergence iterations, tabu list size of 40, tabu small list of size 10 and 250 solutions for the set partitioning problem returns better results on average. Finally, run time for TS and TSPA is around 10-11 hours, whereas it takes around 12 hours for RSPA and 25 hours for SA to find a solution.

CHAPTER VI

CONCLUSION

In this thesis, we study the real-life production scheduling problem faced by Vestel Electronics in TV manufacturing. Vestel is currently using 15 assembly lines to manufacture TVs, and each assembly line produce certain type of TVs. Vestel operates based on a make-to-order strategy and plans the production after the orders are received from the customers. Each order has a certain release date, due date and order size. Each order is processed on one of the compatible assembly lines. Once an order is assigned to an assembly line, it is completed without preemption. Size of the order determines (depending on the assembly line it is assigned to) the processing time of the order/job. At Vestel, TV production planning is performed monthly, and at the beginning of each month for each job/order the start time, processing sequence and the assembly line on which the job is going to be processed are determined. Due to high priority of customer satisfaction, the goal is to minimize the total tardiness of the jobs.

The problem under consideration is a variant of the unrelated machine scheduling problem with sequence-dependent setups, unequal release times and machine-job compatibility restrictions. We propose several algorithms to address this problem including a quite intuitive sequential approach, tabu search algorithm, a set partitioning approach and a novel idea of integrating the tabu search and set partitioning approach. In the set partitioning approach, the main idea is to generate several feasible solutions by perturbing the problem parameters (more specifically processing times), and then choosing the best combination of job-assembly line assignments by solving a set partitioning approach. In the integrated tabu search and set partitioning approach, instead of randomly constructing the feasible solutions we use the solutions found in tabu search and feed them to the set partitioning problem. We observe that the all the proposed approaches provide significant improvements over the current practice. Among the proposed algorithms, the integrated tabu search and set partitioning approach perform best and provides 50% improvement in terms of total tardiness over the real-life solutions.

As future work, because real life problems are large and fast decision making is required, planning is done with heuristic methods. Decision support systems that can be integrated into the heuristic methods can provide faster and near-optimal results. We only focus on end products in this study. The scope of this study can be extended by considering the production planning and scheduling of semi-finished materials.

BIBLIOGRAPHY

[1] M.S. Akturk and D. Ozdemir "An exact approach to minimizing total weighted tardiness with release dates," *IIE Transactions,* **32**(11), pp.1091-1101, 2000.

[2] M.S. Akturk and D. Ozdemir "A new dominance rule to minimize total weighted tardiness with unequal release dates," *European Journal of Operational Research,* **135**(2), pp. 394–412, 2001.

[3] D.E. Akyol and G.M. Bayhan "Multi-machine earliness and tardiness scheduling problem: an interconnected neural network approach," *International Journal of Advanced Manufacturing Technology,* **37**(5-6), pp. 576–588, 2008.

[4] B. Alidaee, and D. Rosa "Scheduling parallel machines to minimize total weighted and unweighted tardiness," *Computers & Operations Research* **24**(8), pp. 775–788, 1997.

[5] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs," *European Journal of Operational Research* **246**(2), pp. 345–378, 2015.

[6] A. Allahverdi and C.T. Ng, T.C.E. Cheng and M.Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Journal of Operational Research* **187**(3), pp. 985–1032, 2008.

[7] V.A. Armentano, M.F.F. Filho, "Minimizing total tardiness in parallel machine scheduling with setup times: an adaptive memory-based GRASP approach," *European Journal of Operational Research* **183**(1), pp. 100–114, 2007.

[8] K.R. Baker and J.W.M. Bertrand, "A dynamic priority rule for sequencing against due-dates," *Journal of Operations Management* **3,** pp. 37–42, 1982.

[9] N. Balakrishnan , J.J. Kanet, S.V. Sridharan, "Early/tardy scheduling with sequence dependent setups on uniform parallel machines," *Computers & Operations Research* **26**(2), pp. 127–141, 1999.

[10] P. Baptiste, P, J. Carlier, A. Jouglet, "A branch-and-bound procedure to minimize total tardiness on one machine with arbitrary release dates," *European Journal of Operational Research* **158**(1) pp. 595–608, 2004.

[11] L.P. Bigras, M. Gamache, G. Savard, "The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times," *Discrete Optimization* **5**(4), pp. 685–699, 2008.

[12] D. Biskup, J. Herrmann, J.N.D. Gupta, "Scheduling identical parallel machines to minimize total tardiness," *International Journal of Production Economics* **115**(1), pp. 134–142, 2008.

[13] J.-F. Chen, "Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints," *International Journal of Advanced Manufacturing Technology* **44**(11-12), pp. 1204–1212, 2009.

[14] H. Du, J.Y.-T. Leung, "Minimizing total tardiness on one processor is NP-hard," *Mathematics of Operations Research* **15**(3), pp. 483–495, 1990.

[15] S.R. Gupta, J.S. Smith, "Algorithms for single machine total tardiness scheduling with sequence dependent setups," *European Journal of Operational Research* **175**(2), pp. 722–739, 2006.

[16] J.E. Holsenback, R.M. Russell, "A heuristic algorithm for sequencing on one machine to minimize total tardiness," *Journal of the Operational Research Society* **43**(1), pp. 53–62, 1992.

[17] C. Koulamas, "The single-machine total tardiness scheduling problem: review and extensions," *European Journal of Operational Research* **202**(1), pp. 1–7, 2010.

[18] E.L. Lawler, "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness," *Annals of Discrete Mathematics* **1,** pp**.** 311–342, 1977.

[19] J.-H. Lee, J.-M. Yu, D.-H. Lee, "A tabu search algorithm for unrelated parallel machine scheduling with sequence- and machine-dependent setups: minimizing total tardiness," *International Journal of Advanced Manufacturing Technology* **69**(9-12), pp. 2081–2089, 2013.

[20] C.-F. Liaw, Y.-K. Lin, C.-Y. Cheng, M. Chen, "Scheduling unrelated parallel machines to minimize total weighted tardiness," *Computers & Operations Research* **30**(12), pp. 1777–1789, 2003.

[21] S.-W. Lin, C.-C. Lu, K.-C. Ying, "Minimization of total tardiness on unrelated parallel machines with sequence- and machine-dependent setup times under due date constraints," *International Journal of Advanced Manufacturing Technology* **53**(1-4), pp. 353–361, 2011.

[22] R. Logendran, B. McDonell, B. Smucker, "Scheduling unrelated parallel machines with sequence-dependent setups," *Computers & Operations Research* **34**(11), pp. 3420–3438, 2007.

[23] X. Luo, F. Chu, "A branch and bound algorithm of the single machine schedule with sequence dependent setup times for minimizing total tardiness," *Applied Mathematics and Computation* **183**(1), pp. 575–588, 2006.

[24] A.H.G Rinnooy Kan, "Machine Scheduling Problems: Classification, Complexity and Computations," *Springer US*, 1976.

[25] P.A. Rubin, G.L. Ragatz, "Scheduling in a sequence dependent setup environment with genetic search," *Computers & Operations Research* **22**(1), pp. 85–99, 1995.

[26] S.-O. Shim, Y.-D. Kim, "Scheduling on parallel identical machines to minimize total tardiness," *European Journal of Operational Research* **177**(1), pp. 135–146, 2007.

[27] F. Sivrikaya-Serifoglu, G. Ulusoy, "Parallel machine scheduling with earliness and tardiness," *Computers & Operations Research* **26**(8), pp. 773–787, 1999.

[28] L.-H. Su, C.-J. Chen, "Minimizing total tardiness on a single machine with unequal release dates," *European Journal of Operational Research* 186(2), pp. 496–503, 2006.

[29] K.-C. Tan, R. Narasimhan, "Minimizing tardiness on a single processor with sequence dependent setup times: a simulated annealing approach," *Omega* **25**(6), pp. 619–634, 1997.

[30] F. Yalaoui, C. Chu, "Parallel machine scheduling to minimize total tardiness," *International Journal of Production Economics* **76**(3), pp. 265–279, 2002.

[31] Z. Zhang, L. Zheng, M.X. Weng, "Dynamic parallel machine scheduling with mean weighted tardiness objective by Q-Learning," *International Journal of Advanced Manufacturing Technology* **34**(9-10), pp. 968–980, 2007.

[32] Z. Zhu, R.B. Heady, "Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach," *Computers & Industrial Engineering* **38**(2), pp. 297–305, 2000.