ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

IDENTICAL PARALLEL MACHINES SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME IN THE AVIATION INDUSTRY

M.Sc. Thesis by

Ümmühan Zeynep Beyza ARIKAN

Department of Industrial Engineering

September, 2021

ANKARA

IDENTICAL PARALLEL MACHINES SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME IN THE AVIATION INDUSTRY

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering, Department of Industrial Engineering

by

Ümmühan Zeynep Beyza ARIKAN

September, 2021

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"IDENTICAL PARALLEL MACHINES SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME IN THE AVIATION INDUSTRY"** completed by **ÜMMÜHAN ZEYNEP BEYZA ARIKAN** under the supervision of Assoc. Prof. Dr BABEK ERDEBİLLİ and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr BABEK ERDEBİLLİ

Supervisor

Prof. Dr. Ergün ERASLAN

Jury Member

Assist. Prof. Dr. Yavuz Selim ÖZDEMİR

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information, and documents are obtained in the framework of academic and ethical rules,

- All information, documents, and assessments are presented in accordance with scientific ethics and morals,

- All the materials that have been utilized are fully cited and referenced,

- No change has been made on the utilized materials,

- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 2021, 24 September **Signature:**

Name & Surname: Ümmühan Zeynep Beyza ARIKAN

ACKNOWLEDGMENTS

I would first like to thank Ankara Yıldırım Beyazıt University for providing me all technical opportunities.

I would like to thank my family for supporting me at every stage of my life. This thesis would not have been without them. I would like to express my special thanks to my master friends, Vildan ÜNLÜ, Pelin ÜSTÜNYER and Şeyma Zeynep ÖZDOĞAN for being a great teammate as well as an excellent friend. I would also like to thank my thesis advisor, Assoc. Prof. Dr. BABEK ERDEBİLLİ.

2021, 24 September **Ümmühan Zeynep Beyza ARIKAN**

IDENTICAL PARALLEL MACHINES SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME IN THE AVIATION INDUSTRY

ABSTRACT

In this study, the machine scheduling process of a company working in the aviation industry was examined. There are identical parallel machines that process profile raw materials specific to the aviation industry. The sequential processing of parts in the form of different profile raw materials reveals the preparation processes as the tools and sequences to be used in production need to be changed. Setup time depends on job and sequencing. The mixed integer mathematical model, which considers sequencing setups has been developed to minimize the overall tardiness. Since scheduling problems are known to be difficult, the optimal solution has been found for the small size data set. In addition, sensitivity analysis was performed for 3 different scenarios.

Keywords: Scheduling, identical parallel machines, mixed integer programming, sequence-dependent setup time.

HAVACILIK SEKTÖRÜNDE SIRA BAĞIMLI HAZIRLIK ZAMANLI BENZER PARALEL MAKİNELERDE ÇİZELGELEME

ÖZ

Bu calışmada, havacılık sektöründe çalışan bir firmanın makine çizelgeleme süreci incelenmiştir. Havacılık sektörüne özgü profil ham malzeme işleyen özdeş paralel makinalar bulunmaktadır. Farklı profil ham malzeme formundaki parçaların ardışık işlenmesi, üretimde kullanılacak takım ve sıralarının değiştirilmesi gerektiğinden hazırlık işlemlerini ortaya çıkarmaktadır. Kurulum zamanı iş ve sıralama bağlıdır. Sıralamaya bağlı kurulumları göz önüne alan karışık tamsayılı matematiksel model, toplam gecikmeyi en aza indirmek amacıyla geliştirilmiştir. Çizelgeleme problemlerinin zor olduğu bilindiğinden küçük boyutlu veri seti için optimal çözüm bulunmuştur. Ayrıca 3 farklı senorya için duyarlılık analizi yapılmıştır.

Anahtar Kelimeler: Çizelgeleme, özdeş paralel makineler, karışık tamsayılı programlama, sıra bağımlı hazırlık zamanı.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Increasing competition, developing technology, and changes in production environments in recent years force businesses to adapt to these conditions. Businesses that want to meet customer demands quickly, ensure efficiency in production processes, and be superior to their competitors have to make better planning. Because businesses that cannot keep up with these market conditions are forced to withdraw from the market over time.

Since today's businesses operate with very low-profit margins in an environment of intense competition, there is no room for unplanned situation, idle capacity, and in short, any waste of resources. In order to react to the expanding needs in today's trendy markets, the importance of production planning has increased as production has become easily uncontrollable in growing enterprises. Therefore, business managers should handle production planning studies professionally and carefully. With these effective production plans, businesses maintain their existence in today's competitive environment by using their existing resources efficiently and allocating them in the proper time at the right location.

One of the predominant focuses of a company is production planning. It is the process of planning the necessary operations for the availability of resources (machine, human, raw materials ...) at the accurate time, place, and amount and to complete the finished product at the desired time with the least cost [1] .

There are 5 main methods in production planning. These are respectively job planning, flow planning, batch planning, mass production planning, process planning.

Job planning is for situations where all the work for a product is done by one worker or group of workers.

In ***batch planning***, production is done in groups. The production of each sub-group must be completed before the final product.

In ***flow planning***, the uninterrupted movement of raw material from one stage to another is provided to perform a series of operations.

Process planning is for systems where raw materials are continuously processed along with the final product along the production line, where special and advanced machines are used to process materials at every stage.

Mass Production Planning is done in the production of many same products in a short time that requires a high level of automation [2] .

Each company should choose the most suitable method for itself, taking into account factors such as product type, raw material, labor, business needs, production method, and cost.

Businesses need effective work scheduling in order to successfully make sure the efficient use of the existing property and in order to endure in today's competitive atmosphere. Scheduling is a decision process that deals with the allocation of limited resources to required activities and tries to optimize one or more purposes, taking into account time constraints. Scheduling determines when workers, equipment, and facilities will be needed to generate a product or contribute a service. It is the final stage of planning before production begins.

A scheduling function must interact with other functions in a service or production system or business system. These interactions are dependent on the system and vary from one situation to another. These are generally included in the information system which is covering the entire enterprise. The location of the scheduling process in a production system and its interaction with other processes are shown in Figure 1.1 [3].

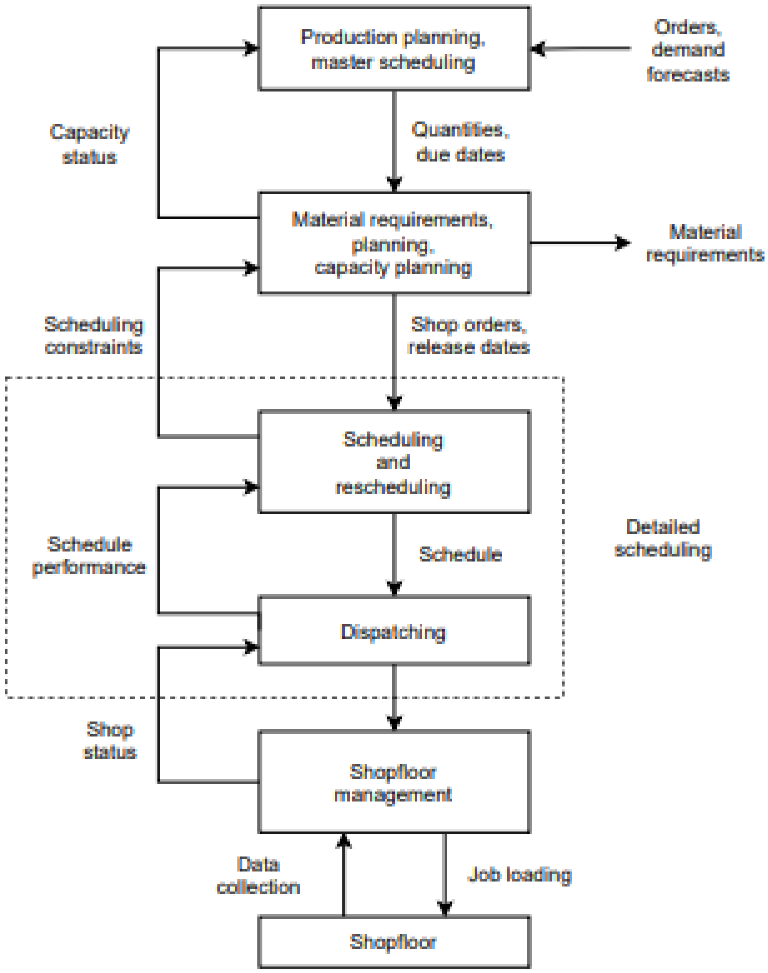

Figure 1.1: The scheduling process in a production system

Two different types of fitness constraints are encountered in scheduling problems. The first base of these is the constraints on the machine capacities, and the second one is the technological limitations on the processing order of some jobs. The solution of the scheduling problem refers to any suitable solution of these constraints, which can be grouped under two headings. Therefore, the process of solving the scheduling problem involves searching for responses to the following base concerns. In other words, the scheduling problem consists of assignment and sorting;

i. Which resource will be assigned to which task?

ii. When will each task be done? [4]

In solving the scheduling problem, many solution techniques have emerged since the first solution algorithm was developed by Johnson (1954). The interest in this problem, which has been on the order of business for just about 50 years, continues to increase today. Solution approaches to scheduling problems of business generally can be separated into two as trying to find an optimum result and trying to find an approximate result. Since scheduling problems are considered as difficult problems, especially in large-sized problems, methods that give up the optimum solution and find non-optimal solutions in a shorter time are used. These are the methods that guarantee the optimum result, but they become insufficient as the problem size grows in scheduling problems. Approximate algorithms can be approximately categorized into metaheuristics and heuristics. The root dissimilarity between them is those heuristic methods usually have simpler, single-solution rules, while metaheuristic methods try to improve the initial solution by set in motion from an initial solution and producing new solutions at every step with its complex and intelligent structure. Since metaheuristic algorithms have an intelligent structure, they are also called artificial intelligence search manner.

The fundamental objective of this inquiry is to examine production scheduling problems that are vital for businesses and to acquire the most favorable schedules with the help of a mathematical model that takes into account the sequence-dependent setup time in uniform parallel machines systems. In this structure, the scheduling problem of a machining company operating in the aviation sector is discussed. The sequential processing of parts in the form of different profile raw materials reveals the preparation processes as the tools and sequences to be used in production need to be changed. In this study, it is aimed to minimize the total lateness by reducing the time spent on a setup by creating an effective schedule where any job will be assigned to which machine after which job.

In the second interest, the literature assessment about this problem has been made. A summary table has been created.

In the third chapter of the thesis, not only widespread information about scheduling and definitions are given but also, classification and representation of scheduling problems are mentioned and solution approaches are understandably mentioned.

In the 4th chapter, the business that is the subject of the application problem is introduced, after the information such as the jobs, processing times, routes of the works, delivery dates are given, and the mathematical model is established for the defined problem. Mathematical model; Written in GAMS and solved with MILP for the objective function aiming to minimize the total tardiness. In the fifth and the sixth chapters, sensitive analysis and results and discussion are given respectively.

CHAPTER 2

LITERATURE REVIEW

Dang et al., they dealt with the scheduling problem on identical parallel machines in their study. They also took into account the tool change time and the sequence-independent setup times. They have adopted multiple objectives that minimize tool setup time and tardiness. They suggested a meta-heuristic algorithm combining genetic algorithm and linear programming for the solution of problem due to the make an assay of the problem and stated that they achieved 20-40% improvement [5].

Yepes-Borrero et al., In this article, sequence-dependent setup time and scheduling problem that requires additional resources during setup are examined in an independent parallel machine environment. The scarcity of resources reduces the cost of production and increases the completion time. The reverse is also true. therefore, a method that maximizes both objectives is presented. Mixed-integer linear programming combined with The Truncated Restarted Iterated Pareto Greedy algorithm is suggested to solving this problem. Associated to former methods in the literature, better results were obtained, especially as the problem size increased. they plan to apply the method they proposed in their later work to the other problem of conflict [6].

Kim et al., In their study, they discussed the scheduling problem where identical parallel machine job division is possible and takes into account sequence-dependent setup times. They aimed to minimize overall tardiness. For this, they proposed genetic algorithms and annealing algorithms, which are simulated with 6 combinations of three decoding methods and two encoding schemes. With the proposed coding schemes, they reduced the solution space, and the decoding methods divided the jobs into sub-jobs and assigned them to the machines, creating a schedule. They argue that the method they propose not only gives better results but also offers a solution in less runtime. In their future work, they plan to apply their method in different machine environments, with a different metaheuristic method, and for a different objective function [7].

Fanjul-Peyro et al., In their piece of work, they studied the scheduling problem in unrelated parallel machines, which takes into account the setup times associated with sequence and tries to minimize makespan. The proposed mixed-integer linear programming and a mathematical algorithm based on adaptations of sub-round elimination constraints to solve the problem. They are able to get results for very large samples of up to 1000 jobs and 8 machines. The results of the model and algorithm also compared the performance of solvers Gurobi and CPLEX. They concluded that Gurobi is more convenient for MILP models than CPLEX. For future studies they suggested adding limited production resources such as personnel and constraints such as overlaps, waiting times and release dates, etc. [8].

Gedik et al., The main target of their studies are to minimize the final production time of the non-preemptive jobs by producing them on unrelated parallel machines with setup time depends on both sequence and machine. They have presented a constraint programming (CP) model to solve this NP-hard problem. The model has two customized branching strategies that use interval decision variables, domain filtering algorithms, and global constraints of CP. They observed that the proposed model has slightly better solutions for small problems than other algorithms in the literature. 283 has proven the optimalness of the solution and offers effective solutions for major problems. In their future work, they plan to combine linear relaxation, which is the classic feature of MIP, with the ability of CP to obtain a feasible result in a short computation time. They will also consider machine breakdowns and job availability intervals [9].

Wang et al., In their work, they focused on rescheduling where job denials were allowed on similar parallel machines, which is an environment of unexpected disruption that caused some jobs to be unavailability. While trying to minimize accepted jobs' total completion time in the adjusted schedule, it is aimed that total rejection cost and the maximum finishing time deduction of any job accepted between the adjusted and original schedule does not exceed the specified limits. To figure out this trouble they put forwarded two ways: dynamic programming and the method of branch-and-price, which is a heuristic procedure that solves the pricing problem while looking for a feasible solution. The second was authorized to be more functional. They intend to develop their model for different machine environments and for other classic scheduling purposes [10].

Yin et al., In this paper some machines unusable due to possible interference were taken into account in the planning of works in the parallel machine environment. When the interruption occurred, they examined whether the maintenance on the defective machine was done immediately or not. Considering all, the fundamental principle goal is to determine the most suited schedule to miniaturize the expected comprehensive integration time of the works. They have shown that when an outage occurs, maintaining faulty machinery increases the cost of maintenance and decreases the total expected completion time of the jobs. They determined the computational complexity of the various states of the problem and presented pseudo-polynomial-time solution algorithms and purely polynomial-time approximation schemes for them. In their future work, they plan to expand their models to determine the most appropriate maintenance start time and the most appropriate schedule and to apply this for different machine environments [11].

Afzalirad & Rezaeian, In their studies, they offered a solution to the scheduling problem that minimizes the makespan in unrelated parallel machines. They commonly take into account sequence-dependent setup time and machine availability. They dealt with the priority relationship between jobs and different deadlines. At the same time, they stated that resources such as tools, labor, fixtures were limited and they took into account. They developed two meta-heuristic algorithms, the Artificial Immune System and Genetic Algorithm. Parameters were tuned using the Taguchi method to ensure that the algorithms they proposed were robust. While both models were effective in small problems, AIS worked better in large problems. In their future work, they will focus on machine breakdowns and reprocessing, and they plan to address the issue with different meta-heuristic methods [12].

González-Neira et al.,They set two decision-making goals, one quantitative and the other qualitative, to solve the stochastic process time scheduling problem. The goals are the total weighted delay and the customer's importance to the company, respectively. It applies both a simulation optimization approach and a Mixed Integral Linear Programming (MILP) model for the total weighted validation result. Each alternative is evaluated by Stochastic Multi-criteria Acceptability Analysis (SMAA) according to as on the prestige of the client. They argued that the Integral analysis method they proposed made it possible to select alternatives that best meet both types of criteria. They are considering applying their proposed method in different machine environments and for diverse aspire such as time to completion or flow time [13].

Yoo & Lee, Their study addresses the problem of scheduling, including maintenance activities on parallel machines over an interval of time. They make an effort to create a schedule that would minimize the cost of planning while coordinating works and maintenance activities. They have created algorithms for 5 different purposes: the sum of the finishing times and the weighted case, makespan, maximum delay and delay sum. Maintenance activities are planned for dependent situations where only one machine is allowed to be in maintenance activity at any one time, with an independent state that allows more than one machine to engage in maintenance activity at the same time [14].

Alvarez-Valdes et al.,They addressed the problem of just-in-time scheduling on similar parallel machines for a series of jobs with a common deadline. They examined how the jobs are delegated to the existing machines and how the jobs are sorted on each machine. They integrated urgency rules for assigning jobs to machines and a local search to solve single-machine sub-problems. They proposed the Path Relinking (PR) method, which constructs a path between good solutions to solve the problem, and the A Scatter Search algorithm, where the elements of certain subsets of the first convenient remedy are combined to reach the best known solution. They reported that the algorithms they proposed gave better results in large problems at reasonable computing times. They plan to expand their work for sequence dependent setup time and jobs with different deadlines problems [15].

Fattahi et al., They spotlight the problem of the hybrid flow shop scheduling, which takes assembly and preparation time into account. Each product consists of combining several parts produced in the hybrid flow shop at the assembly stage, and several products of the same type are produced. They proposed a limit algorithm and hierarchical branch to minimize production time, valid for all parts. To upgrade the proficiency of the algorithm, they developed three upper limits set by Greedy Randomized Adaptive Search Procedure (GRASP) and three lower limits. The algorithm performance was evaluated in scenarios where the hybrid flow shop is a bottleneck, the assembly phase is a bottleneck, and the two phases are in balance. The best result was obtained in the second scenario and the worst result was in the case of balance. They argued that their proposed algorithm has acceptable performance and that using GRASP is effective in reducing the computing time and reducing branches. In their next work, they consider using line balancing to reduce process time and minimize total tardiness and earliness [16].

Joo et al., In their work, they focalized the problem of scheduling in a three-step dynamic flexible flow workshop. They took into account that there is a quality feedback mechanism and that multiple types of work reach dynamically. They discussed the process defect rate and installation timing as interrelated. They aimed to maximize the mean tardiness and the quality rate. For the clarification of the problem, they proposed two distribution rule-based algorithms that take into account the quality feedback and the dynamic structure of the store, respectively, the apparent tardiness cost with sequence-dependent setup (ATCSQ) and slack per remaining work with the setup ratio (S / RW-SR) [17].

Xi & Jang, In their work, they aimed to minimize the total weighted delay in the identical parallel machine with sequence dependent setup and unequal ready-to-run time. Examines the performance of Apparent Tardiness Cost dispatching rules for two conditions: a continuous sequence-dependent setup that needs parts readiness, and a separable sequence dependent setup where the machine can remain idle in the time between machining of the part and setup. They argued that the models (MATCSR, ATCSSR) they proposed in their experiments performed better than existing ATC-based rules. In their future work, they plan to apply their models in different environments such as a job shop, a flow shop [18].

Lee et al.,In their work, they dealt with the planning of deteriorating jobs, which are known as the completion of delayed jobs in the literature. They tried to solve the problem of single machine scheduling, taking into account customer satisfaction, setup times and meeting promised delivery dates. A branch and bound algorithm were improved to diminish the number of late jobs. They accelerated the search process by placing an initial lower and upper limit with a heuristic algorithm. They demonstrate that the algorithm they proposed could solve 1000 jobs in a reasonable time [19].

Khalouli et al., They focused on solving the scheduling problem in the hybrid flow shop using the ant colony optimization method. In order to come through with increasing challenge, the importance of reducing inventories and timely delivery was mentioned. It was aimed at minimizing the total of earliness and tardiness for the problem addressed. They focused on solving the scheduling problem in the hybrid flow shop using the ant colony optimization method. They showed in their experiments that the algorithm they proposed gave better results than other heuristic methods. They plan to combine their algorithms with fuzzy logic techniques in future studies [20].

Table 2.1 below, literature review of scheduling problem is summarized.

Table 2.1: Summary of the literature review

	Deterministic/Stochastic Dynamic /Static	*Machine Structure*	*Setup Time*	*Objective Function*	*Methodology*
(Khalouli Et Al., 2010)	Deterministic Static	Flow Shop	-	Minimize The Total Of Tardiness And Earliness	Ant Colony Optimization
(Lee Et Al., 2011)	Stochastic Dynamic	Single Machine	Family Sequence-Dependent Setup Time	Minimize The Number Of Late Jobs	Branch-And-Bound Algorithm
(Xi & Jang, 2012)	Deterministic Static	İdentical Parallel Machines	Sequence-Dependent Setup Time	Minimize The Total Weighted Delay	Apparent Tardiness Cost Distpatching Rules
(Joo Et Al., 2013)	Deterministic Dynamic	Flexible Flow Shop	Sequence-Dependent Setup Time	Maximize The Mean Tardiness And The Quality Rate	Apparent Tardiness Cost Distpatching Rules With Sequence-Dependent Setups And Quality Ratio
(Fattahi Et Al., 2014)	Stochastic Static	Hybrid Flow Shop	Sequence-Dependent Setup Time	Minimize Production Time	Hierarchical Branch And Limit Algorithm
(Alvarez-Valdes Et Al., 2015)	Deterministic Static	Similar Parallel Machines	-	Total Weighted Earliness And Tardiness	Path Relinking And Scatter Search
(Yoo & Lee, 2016)	Dynamic Stochastic	Parallel Machines	-	Minimize The Scheduling Cost	Dynamic Programming
(González-Neira Et Al., 2016)	Stochastic Static	Flexible Flow Shop	-	Minimizing Tardiness Penalty Costs And Timely Fulfillment Of Due Dates According To Customer İmportance	İntegral Analysis Method And Mixed Integral Linear Programming (MILP) Model And A Simulation-Optimization Approach
(Afzalirad & Rezaeian, 2016)	Deterministic Static	Unrelated Parallel Machines	Sequence Dependent Setup Time	Minimizes The Makespan	Artificial Immune System And Genetic Algorithm

Table 2.1 (contined): Summary of the literature review

	Deterministic/Stochastic Dynamic /Static	***Machine Structure***	***Setup Time***	***Objective Function***	***Methodology***
(Yin Et Al., 2017)	Dynamic Deterministic	Parallel Machine	-	Minimize Total Completion Time	Polynomial-Time Solution Algorithms
(Wang Et Al., 2018)	Dynamic Deterministic	Similar Parallel Machines	-	Minimize Total Completion Time	Dynamic Programming And The Method Of Branch-And-Price
(Gedik Et Al., 2018)	Deterministic Static	Unrelated Parallel Machines	Sequence Dependent Setup Time And Machine Dependent Setup Times	Minimize The Final Completion Time	Constraint Programming
(Fanjul-Peyro Et Al., 2019)	Dynamic Stochastic	Unrelated Parallel Machines	Sequence Dependent Setup Time	Minimize Makespan	Mixed İnteger Linear Program
(Kim Et Al., 2020)	Deterministic Static	İdentical Parallel Machine	Sequence-Dependent Setup Times	Minimize Total Tardiness	Simulated Annealing Algorithms And Genetic Algorithms
(Dang Et Al., 2021)	Deterministic Dynamic	İdentical Parallel Machines	Sequence-İndependent Setup Time	Minimize Both Tardiness Of Operations And Tool Setup Times.	Combines A Genetic Algorithm And An İnteger Linear Programming Algorhtm
(Yepes-Borrero Et Al., 2021)	Deterministic Static	Unrelated Parallel Machine	Sequence Dependent Setup Times	Minimize Makespan And The Number Of Resources.	The Truncated Restarted Iterated Pareto Greedy Algorithm And Mixed İnteger Linear Programming

CHAPTER 3

SCHEDULING

Scheduling problems according to Pinedo are divided into 5 different categories according to the machine structure.

Single machine: all jobs are assigned to a single machine.

The ***parallel machine*** is divided into 3 within itself. The situation where the finalization duration of a job is the same on all machines is called an identical machine. Uniform machine, the time to achieve a task varies with the speed of the machine. Unrelated machine, the time it grabs to complete a job varies by machine.

The ***flow shop*** consists of several stages where one or more machines exist in each stage where all works are processed in the same routine.

Jop shops are systems that consist of different machines and each job follows its own order.

Open shop: Each job is processed on one of m machines, with no routine restrictions [3].

3.1. Classification of Production Planning

If we want to classify according to the readiness of the works;

If there is randomness in parameters such as processing time, setup times, arrival times, and delivery dates, it is considered as *stochastic*, if it is fixed and if it is known beforehand, it is considered as a *deterministic* problem [13] .

If all jobs are available and ready before scheduling, it is called *static scheduling* and no changes are made to the schedule. If jobs are ready dynamically over time, scheduling based on real-time events is *dynamic scheduling*. Jobs are processed continuously, completed are removed from the system [21].

Three approaches are used when performing dynamic scheduling. In *reactive scheduling*, there is no exact schedule considering uncertainty. Decisions are made locally during implementation. Dispatching rule is the most common reactive method. In *proactive scheduling*, an initial schedule is created by predicting the outbreak of unexpected events. Robustness and stability have been evaluated as a proactive methods in the literature. The most unremarkably utilized method, *proactive-reactive scheduling*, has two stages. The reactive method is applied to the first schedule that is created proactively, when the interruption occurs [22] .

Real time events fall into two groups. Job cancellation, urgent job, change in priority of jobs, change in delivery date, change in processing time, late or early arrival of job are handled as *work-related events*. For *resource-related events* the operator's illness, machine failure, unavailability or tool malfunction, defective material, lack of material or late arrival, loading limit can be given as examples [23].

When and how rescheduling is done to deal with real-time events is important. Schedule repair and complete rescheduling are the two main strategies of the first event. *Schedule repair* is a local modification of the existing scheduling as the stability of the system is maintained. *Complete rescheduling* is the creation of a schedule from scratch. Although it maintains optimality in principle, it is not preferred in practice due to cost and computation time. In the literature, three methods have been adopted regarding when to reschedule. In the *periodic policy*, scheduling is made at certain intervals in the light of the knowledge coming from the production. In the most preferred *event driven policy*, an unexpected event triggers a rescheduling. In the *hybrid policy*, periodic rescheduling is performed in the system, and when an emergency unexpected event occurs, a complete rescheduling is performed [23].

If we want to classify by delivery date, it can be divided into the case where all major works have a common due date or have different due dates. The delivery date may be determined by the manufacturer or agreed upon with the customer [24].

The classification regarding setup times is made as family and non-family. Usually, part families are created according to the required machine, tool, and process similarity. Then, each class is divided into two as sequence independent and sequence dependent. The setup time for a specific task is termed sequence-dependent when it depends on which job was produced on the machine before running that job. Also, in 2008, Koulamas and Kyparisis introduced the concept of past-sequence dependent setup time. Accordingly, the setup time of a job is proportional to the sum of the processing times of the previously planned jobs [25].

Setup time is the preparation time of resources such as machines and people required to perform a job. Setup times can be parcel out into three main groups. Minor Setup Time is required for the first job to be taken at the starting time of the scheduling or for the repair of the unfinished job immediately after the machine failure. Medium Setup Time is applied when a subsequent job similar type with after the completed job is on the same machine. Major Setup Time is required when the previous job and the subsequent job to be produced on the same machine are different [26].

Mokotoff divided the objective functions of scheduling problems into 3 groups. The first group is based on completion time. Makespan (Cmax), which ensures efficient use of resources, and objectives that focus on responding quickly to demands, average completion time ($\bar{C}$), average flow time ($\bar{F}$), etc can be counted as instances. The second is due date based objectives. They are average lateness ($\bar{L}$), number of tardy jobs (n_t), tardiness (T). The last objective is based on inventory and utilization costs such as average earliness ($\bar{E}$), idle time (I) [27].

3.2. Solution Approaches

Solution approaches to scheduling problems can be divided into two as trying to find out an optimum solution and trying to discover an estimated solution. Since scheduling problems are considered difficult problems, especially in large-sized problems, methods that give up the optimum solution and find non-optimal solutions in a shorter time are used.

3.2.1. Exact Approaches

Optimization is the process of choosing the best alternative. In an optimization problem, the correct set of input variables is systematically selected and the value of the objective function is maximized or minimized so that the output reaches the goal [28].

Optimization algorithms are examined in several groups. Figure 3.1 shows these types of organization algorithms. However, it cannot be alleged that this is a certain classification.

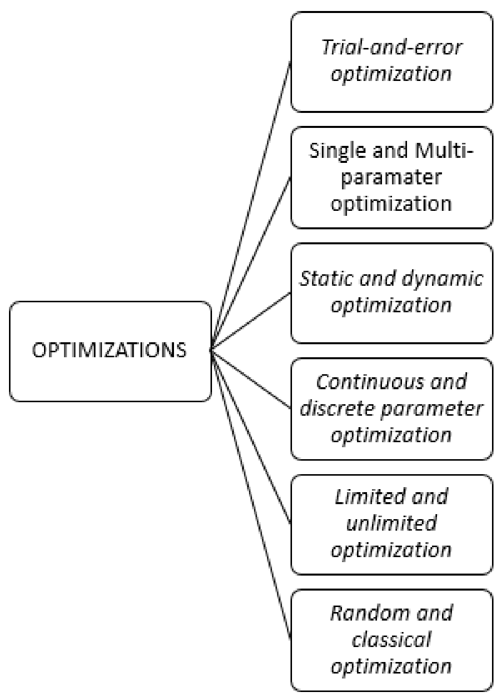

Figure 3.1: Types of optimization algorithms

Trial-and-error optimization: Even if not much is known about the process, it is simply adjusting the parameters that affect the output.

Single and Multi-parameter optimization; If there is only one parameter, the optimization is one parameter. It is multi-parameter optimization if it has more than one parameter. The abundant the numeral of parameters, the more difficult the optimization becomes.

Static and dynamic optimization: Static optimization is time-independent, while dynamic optimization is time-dependent.

Continuous and discrete parameter optimization: Continuous parameters take infinite values while discrete parameters take finite values. Discrete parameter optimization can also be called combinatorial optimization and is the problem of arranging, grouping, ordering or selecting discrete quantities optimally.

Limited and unlimited optimization: Limited optimization, its parameters are defined in a range of values. In unlimited optimization, the parameters can be of any value. By removing the boundaries of the variables, the limited parameters are converted to unlimited parameters. Most numerical optimization routines work with unlimited parameters. Bounded optimization, linear equations and parameters with linear limits When it optimizes, the program is called a linear program. If the boundaries and cost equations are nonlinear, the program is also called a nonlinear programming problem.

Random and classical optimization: Some algorithms try to minimize the fitness values by adjusting the initial values of the parameters. This search technique is fast but can reach local minimums. These are classical optimization algorithms based on numerical methods. Determining the other parameter from one parameter is carried out with some deterministic steps. On the other hand, random methods; they use probability calculations to find the optimum solution of the parameters. Although these methods are slow, they are more successful in finding the global minimum [29].

The Branch and Boundary Algorithm is a frequently used analytical method. A dynamically constructed tree is searched in the solution space consisting of all available charts. This technique formulates procedures and rules for removing large parts of the tree from research [30] .The branch is used to divide a large-scale problem into two or more sub-problems, and the boundary is used to calculate the lower bound for the optimal result of a sub-problem [31]. With this method, only small-sized problems can be solved in a reasonable time. The main reason for this is that strong elimination rules cannot be come up with to accept most of the problem from the solution in the early solution stages [32].

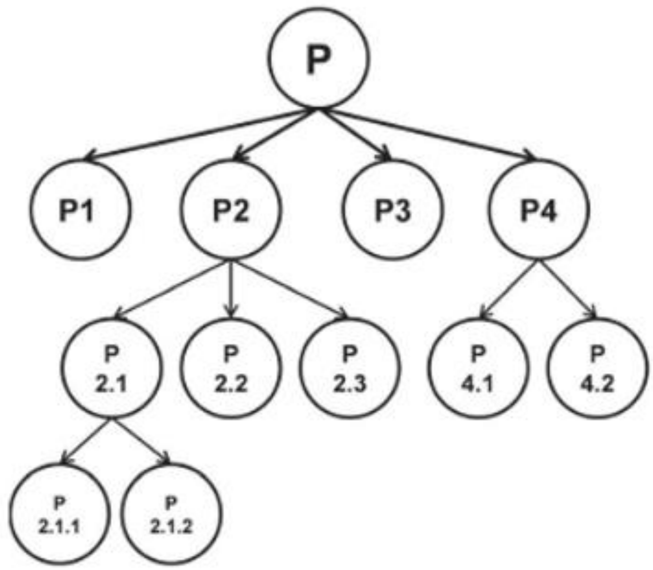

Figure 3.2: Branch and Bound tree

Mathematical methods aim to reach the full global optimum of the problems. To achieve this, it evaluates every possible solution to the problem against a target function, taking into account specified boundary conditions. This indicates that the required computational effort increases with problem complexity. As a result, mathematical methods for complex problems may become unsolvable or require too many constraints to produce meaningful results [33] .

3.2.2. Approximate Approaches

Approximate algorithms can be roughly divided into heuristics and metaheuristics. The main difference between them is those heuristic methods usually have simpler, single solution rules, while metaheuristic methods try to improve the initial solution by starting from an initial solution and producing new solutions at every step with its composite and rational structure. Since metaheuristic algorithms have an intelligent structure, they are also called artificial intelligence search techniques.

The first and simplest ***heuristics*** are priority allocation rules. These structural techniques give a priority to all available operations for sequencing, then select the operation with the highest priority. They are very easy to apply and have a low computational load. The frequently used ones of these rules are briefly mentioned below.

SPT (Shortest Processing Time First): The job which has the shortest processing time is chosen.

LPT (Longest Processing Time First The job which has the longest processing time is chosen.

WSPT (Weighted Shortest Processing Time First): The job with the shortest weighted processing time is chosen.

EDD (Earliest Due Date): The operation of the job with the earliest due date is chosen.

MS (Minimum Slack First): The operation of the job with the least time left to the deadline is selected.

SRPT (Shortest Remaining Processing Time First): The job with the shortest remaining processing time is selected.

SST (Shortest Setup Time First): When a machine is empty, the job requiring the shortest setup time is selected.

MRWT (Most Remaining Work Time): The operation with the highest sum of the processing times itself and after it is selected.

LRWT (Least Remaining Work Time): The operation with the least sum of the processing times itself and after it is selected.

FCFS (First Come First Served): The job that comes first to the machine is selected [3].

Metaheuristic methods that are frequently used in the literature are introduced respectively.

Artificial intelligence search techniques are relatively new approaches to solving optimization problems. AI search techniques are different from other heuristics, although they do not undertake an optimum solution. It can be said that heuristics, who generally seek a good solution by using or learning a general knowledge of the problem, belong to the category of artificial intelligence search techniques [34].

Tabu Search Algorithm

Tabu search algorithm is an iterative search algorithm developed by F. Glover. The algorithm proceeds through a single solution. It is derived from the assumption of generating and evaluating solutions adjacent to the current solution and moving to a neighboring solution as an outcome of the assessment. In some cases, the chosen neighbor solution can return the solution to the previous state in the next iteration. The flexibility of taboo search to accept bad solutions also provides this. Tabu search has elements called initial solution, movement mechanism, candidate list strategies, memory tabu breaking criteria, stopping conditions. The basic approach is to prohibit or penalize its repetition in the next cycle to prevent the step leading to the final solution from creating circular movements. Tabu search is a process that can search for any solution, unless it's one that has been explored before. Thus, by examining a new solution space, the desired solution can be reached by avoiding the local minimum [35]. Figure 3.3 shows the flow chart of tabu search algorithm.

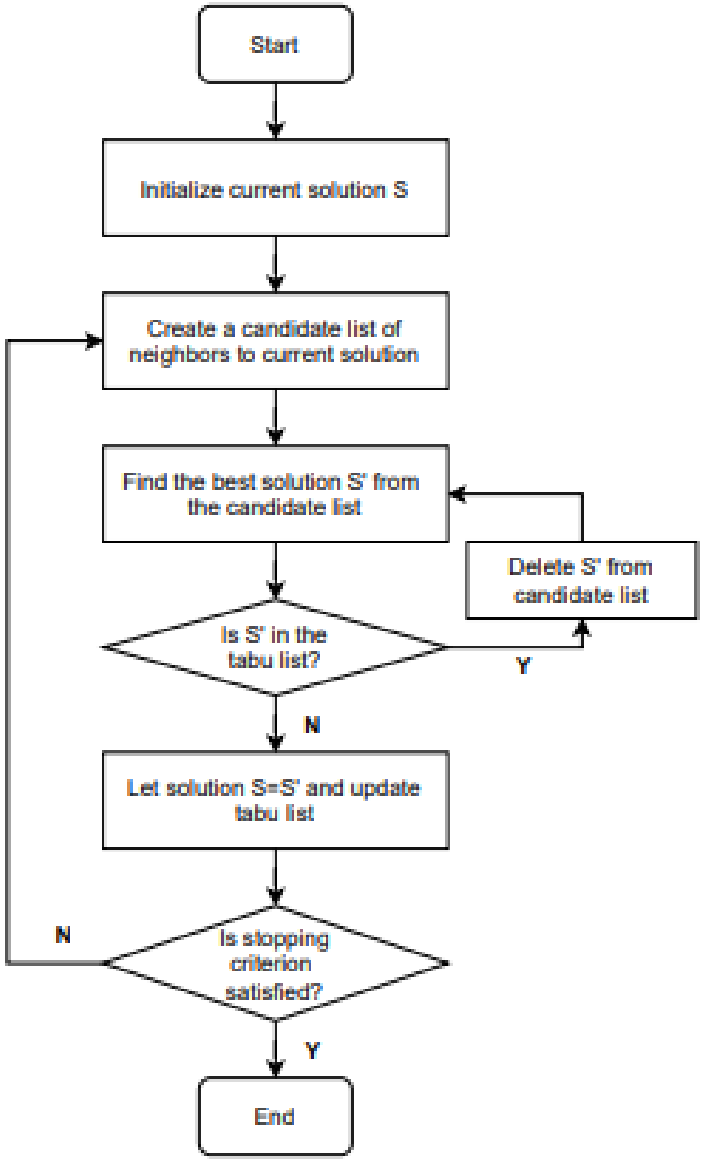

Figure 3.3: Flow chart of tabu search algorithm

Annealing Algorithm

The annealing algorithm is inspired by the analogy between the heat treatment of metals and the solution research events to an optimization problem. Annealing describes the process of reducing the temperature of a solid by increasing it to a certain maximum degree. In the similarity established between the optimization problem and annealing, the states of the solid correspond to the possible solutions of the optimization problem, the energies of the states correspond to the objective function values of the solutions, and the minimum energy, that is, the steady state (freezing), to the final solution. The simulated annealing procedure consists of a series of iterations. Starting from the initial solution, a new solution is found in each iteration based on neighboring solutions. The newly found solution is compared both with the previous one and with the most excellent solution found so far. If the new solution is finest than the previous one, it is accepted, if it is better than the best solution found so far, the new best solution becomes the new best solution. Compared to other methods, the most important advantage of simulated annealing is the ability to get rid of the local minimum [35] .The flow chart of the annealing algorithm is shown in Figure 3.4.

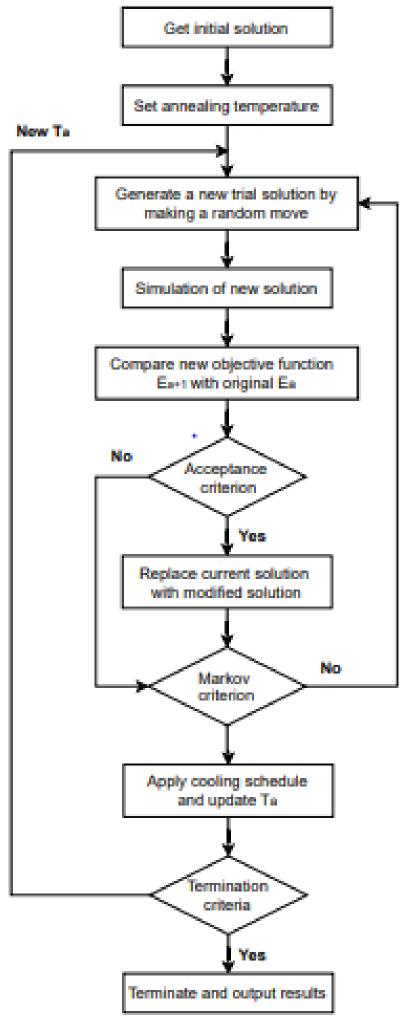

Figure 3.4: Flow chart of annealing algorithm

Genetic Algorithm

The keystone principles of the genetic algorithm first emerged when Holland brought together his work in 1975 in his book "Adaptation in Natural and Artificial Systems". Flow chart of genetic algorithm is given in Figure 3.5. Genetic algorithm is a type of evolutionary algorithm. An evolutionary algorithm is an analogy of natural evolution. Evolution in nature takes place by the survival of the matches. Weak individuals either live long enough to produce new individuals or die without producing any new individuals. However, since the stronger ones will survive longer, they will produce more new individuals and have offspring with similar characteristics. The genetic algorithm uses an initial population of some of the solutions found in the search space. The initial population is successively established in each generation through natural selection and reproduction processes. The highest quality individual of the latest generation becomes the ultimate solution to the problem. This solution may not always be the optimum solution, but it is a solution close to the optimum. [36]

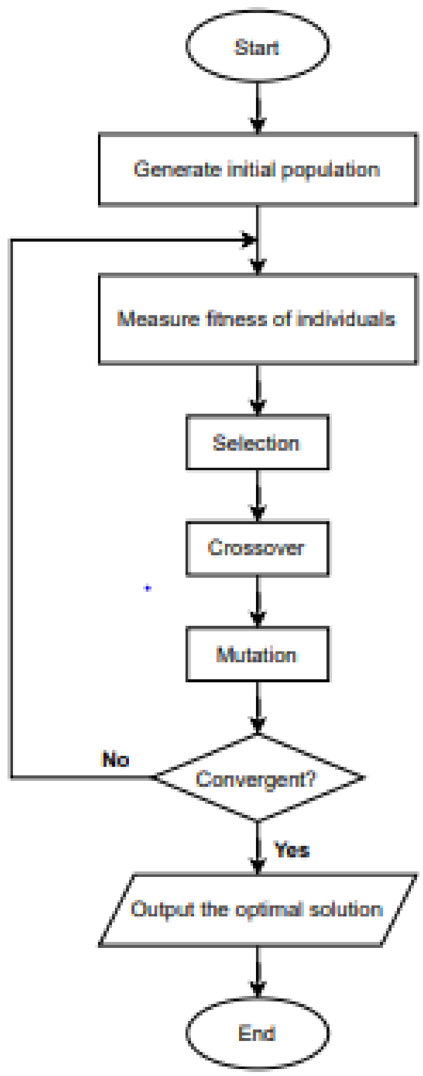

Figure 3.5: Flow chart of genetic algorithm

Artificial Bee Colony Algorithm

The artificial bee colony algorithm is another swarm algorithm developed by being inspired by the behavior of bees in nature. In nature, bees search for food by division of labor on their own. When a food source is found, they make an assessment and determine the value of the source. In search of new resources, scout bees engage in random behavior. When scout bees find a new source, the scout bee dances according to factors such as the amount of nectar found in the source, and in this way informs the other bees in the hive. Onlooker bees, seeing the dance of the scout bee, may begin to act as a worker bee and go to the source or choose to remain as onlooker bees. After collecting the food, the worker bees return to the hive and can decide whether to go to the same source or not. Worker bees also share information about the source they are assigned with other bees in the hive

In the artificial bee colony algorithm, the positions of the bees represent food sources. A scout bee informs the onlooker bees in the hive about the fitness value of its location and tries to attract the worker bees to the source where it is located and to ensure that the source and the source area are adequately investigated. Worker bees leave their source areas and create a new solution as scout bees when the criteria for adequate research are met [35]. Flow chart of bee colony algorithm shown in Figure 3.6.

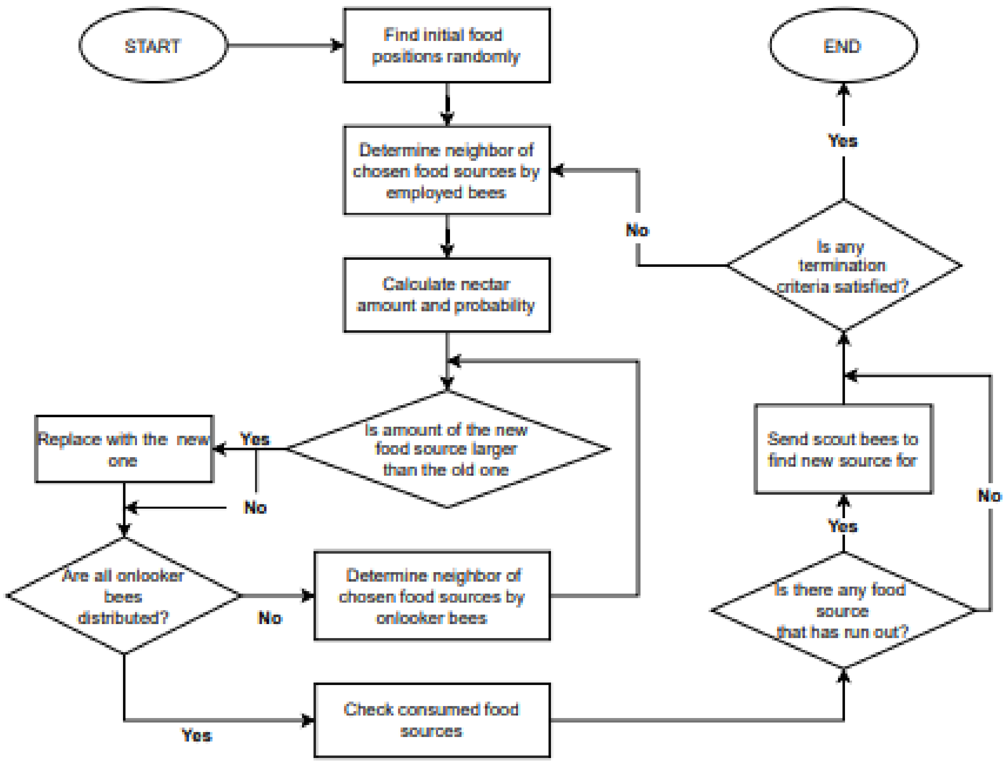

Figure 3.6: Flow chart of artificial bee colony algorithm

Ant Colony Optimization

The first emerging ant algorithm was known as the ant system. It is an algorithm and was proposed by Dorigo et al. in 1991. The ant algorithm aims to model the behavior used by blind ants to search for the shortest network-based path between their bee colony area and the finding food resource. Ants using pheromone tracks exhibit a random behavior on paths that initially contain no pheromones. A large number of ants cause different amounts of pheromones to accumulate on each road. Ants are very likely to pick the path with high pheromone levels. After a while, most of the ants start to use the shortest path between their colony centers and the nutrient resources. It is seen that the ant algorithm is mainly used in traveling salesman problems, and it is also used in job sequencing problems and assignment problems [36]. Figure 3.7 shows the ant algorithm flow chart.

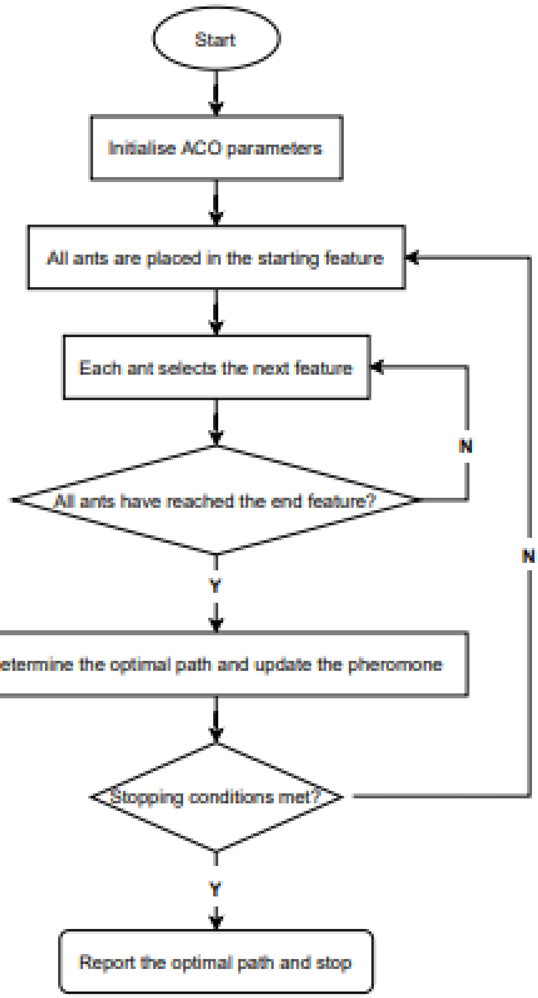

Figure 3.7: Flow chart of ant colony optimization

CHAPTER 4

APPLICATION

In this part of the study, a scheduling problem in an even parallel machine production system is discussed by considering the production system in an enterprise working as a sub-supplier in the aerospace industry. The machining company has an extrusion machine which is shown in Figure 4.1 specific to the aviation industry. These machines provide the desired part to be produced with high quality, fewer chips and in a curt interval.

At the final step of the production of profile raw material in the form of H-I-L-T-U-Z, a piece with the desired profile, length and number of holes are produced. The sequential processing of parts in the form of different profile raw materials reveals the preparation processes as the tools and sequences to be used in production need to be changed. Therefore, the problem includes sequencing setup times.

Figure 4.1: The extrusion machine

MATHEMATICAL MODEL

Parameters

I: Set of orders, $i \in I$, $i \in \{1..I_{max}\}$

J: Set of orders, $i \in J$, $j \in \{1..J_{max}\}$

M: Set of machines, $m \in M, m \in \{1..M_{max}\}$

K: Set of sequences, $k \in K$, $k \in \{1..K_{max}\}$

P_i: Processing time of order, $i \in I$

D_i: Due date of order, $i \in I$

$S_{i,j}$: Setup time between order i and order j, $i \in I$, $j \in J$

B: A big positive number

Decision Variables

$X_{i,m,k} \in \{0, 1\}$: 1 If order i is assigned to $k-$th in sequence in machine m,

$i \in I, m \in M$, $k \in K$

$Y_{i,m,j,k} \in \{0, 1\}$: 1 If there is a setup between order i and order j in k−th sequence

in machine, $i \in I$, $j \in I, m \in M$, $k \in K$

$W_{i,j} \in \{0, 1\}$: 1 If there a setup between order i and order j, $i \in I$, $j \in I$

T_i: Tardiness of order i, $i \in I$

F_i: Finishing time of order i, $i \in I$

Formulation

$$\text{Min Z} = \sum_i \text{T}i \tag{4.1}$$

Subject to

$$\sum_{m \in M, k \in K} \text{X}i, m, k = 1 \qquad \forall \text{i} \in \text{I} \tag{4.2}$$

$$\sum_{i \in I} \text{X}i, m, k \leq 1 \qquad \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K} \tag{4.3}$$

$$\sum_{i \in I, k \in K} \text{X}i, m, k \geq 1 \qquad \forall \text{m} \in \text{M} \tag{4.4}$$

$$\sum_{i \in I} \text{X}i, m, k \geq \sum_{j \in J} \text{X}j, m, k + 1 \qquad \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K}, \forall \text{k} \leq \text{Kmax - 1} \tag{4.5}$$

$$\text{Y}_{i,m,j,k} + 1 \geq \text{X}_{i,m,k} + \text{X}_{j,m,k+1} \qquad \forall \text{i} \in \text{I}, \forall \text{j} \in \text{J}, \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K}, \text{i} \neq \text{j}, \text{k} \leq \text{Kmax - 1} \tag{4.6}$$

$$2{*}\, \text{Y}_{i,m,j,k} \leq \text{X}_{i,m,k} + \text{X}_{j,m,k+1} \qquad \forall \text{i} \in \text{I}, \forall \text{j} \in \text{J}, \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K}, \text{i} \neq \text{j}, \text{k} \leq \text{Kmax - 1} \tag{4.7}$$

$$\text{W}_{i,j} = \sum_{m,k} \text{Y}i, m, j, k \qquad \forall \text{i, j} \tag{4.8}$$

$$\text{F}_j = \sum_{i \neq j} (\text{F}i * \text{W}i, j + \text{W}i, j * \text{S}i, j) + \text{P}_i \qquad \forall \text{j} \in \text{J} \tag{4.9}$$

$$\text{T}_i \geq \text{F}_i - \text{D}_i \qquad \forall \text{i} \in \text{I} \tag{4.10}$$

$$\text{F}_i \geq 0 \qquad \forall \text{i} \in \text{I} \tag{4.11}$$

$$\text{X}_{i,m,k} \in \{0, 1\} \qquad \forall \text{i} \in \text{I}, \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K} \tag{4.12}$$

$$\text{Y}_{i,,m,j,k} \in \{0, 1\} \qquad \forall \text{i} \in \text{I}, \forall \text{j} \in \text{J}, \forall \text{m} \in \text{M}, \forall \text{k} \in \text{K} \tag{4.13}$$

$$\text{W}_{i,j} \in \{0, 1\} \qquad \forall \text{i} \in \text{I}, \forall \text{j} \in \text{J} \tag{4.14}$$

The objective function (4.1) minimizes the total tardiness of orders. The 2nd constraint is to ensure that each order is assigned to one sequence of only one machine. Constraint (4.3) limits any sequence of a machine to at most one order. Constraint (4.4) prevents all orders from being assigned to the same machine. Constraint (4.5) allows an order to be assigned after the order assigned to any sequence of a machine. Constraint (4.6), (4.7) and (4.8) indicate whether there is a setup among sequential orders assigned to a machine. Constraint (4.9) gives the completion time of each order. Constraint (4.10) calculates the tardiness of each order. Constraint (4.11) indicates that the completion time cannot be negative. Constraint (4.12), (4.13) and (4.14) indicates that the variables X, Y and W are binary variables. In constraint (4.9), the multiplication of decision variables causes nonlinearity. To prevent this, a new variable has been defined and the constraints have been added as follows.

$A_{i,j}$ additional decision variable for nonlinear term, $i \in I$, $j \in J$

$$A_{i,j} \leq B * W_{i,j} \qquad \forall i \in I, j \in J \quad (4.16)$$

$$A_{i,j} \leq F_i \qquad \forall i \in I, j \in J \quad (4.17)$$

$$A_{i,j} \geq F_i - B*(1- W_{i,j}) \qquad \forall i \in I, j \in J \quad (4.18)$$

$$A_{i,j} \geq 0 \qquad \forall i \in I, j \in J \quad (4.19)$$

So the proposed model is converted to Mixed Integer Linear Programming (MILP).

The problem is stated as: there are 7 orders assigning to 2 identical parallel machines. Each order j has its, due date (dj), processing time (pj), and the setup time (sij) of each pair of orders i and j is sequence dependent and the continuous type, In general, sij=sji. The objective is to minimize the overall tardiness of orders,

$\sum_i Ti$ where Ti is the tardiness of order j, max{0,Fj–dj}, and Fj is the competition time of order j.

- All orders and machines are ready for production at the start time.
- Each machine can process at most one order at a time.
- Each order can be processed on a maximum of one machine at the same time.
- Preparation time depends on the job sequence.

- The preparation process starts when the work comes to the counter.
- There is no order of priority among jobs.
- Every order can be produced on any machine.
- The order attributes (dj, pj, sij) are known in advance.
- Since the machines are uniform, the process of each order on each machine is the same.
- The order quantity is specific for each job.
- It is presumed that there will be no situations such as order cancellation or machine failure.

The unit processing time and quantity of order the parts to be produced are given in Table 4.1.

Table 4.1: Unit processing time and quantity of order

Parts	Unit processing time (minute)	Quantity of the order
Part 1	3,6	150
Part 2	12	100
Part 3	20	45
Part 4	3	120
Part 5	1	180
Part 6	11	60
Part 7	10	30

The order processing time, which is found by multiplying the unit process time and quantity, is given in Table 4.2 in hours.

Table 4.2 : Processing time

Order	Processing time of order (hour)
Order 1	9
Order 2	20
Order 3	15
Order 4	6
Order 5	10
Order 6	11
Order 7	5

The due date for each orders are given in Table 4.3.

Table 2.3: Due date of orders

Order	Due date of order (hour)
Order 1	10
Order 2	25
Order 3	22
Order 4	12
Order 5	20
Order 6	24
Order 7	16

The required setup times in hours between order i and order j are given in Table 4.4.

Table 4.4: *Required setup times*

	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6	Order 7
Order 1	0	3	2	1	1	3	2
Order 2	3	0	2	3	2	3	1
Order 3	2	2	0	2	1	3	3
Order 4	1	3	2	0	2	1	2
Order 5	1	2	1	2	0	1	3
Order 6	3	3	3	1	1	0	2
Order 7	2	1	3	2	3	2	0

The Gantt diagram of the initial scheduling, which was created with the heuristic method considering the delivery dates, is given in Figure 4.2. Orders 1, 7, 3 are assigned to the first machine, while orders 4,5,6 and 2 are assigned to the second machine, respectively. Order 4 (6), order 1 (9), order 7 (16), order 5 (18), order 6 (30), order 3 (34) and the last completed order 2 were completed in 53 hours. Order 4 was completed 6 hours early, while ordering 2 was completed 28 hours late. The objective function was calculated as 37.

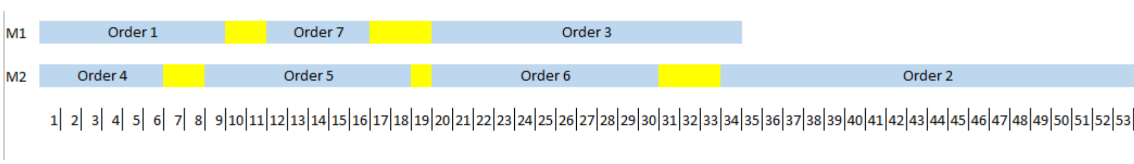

Figure 4.2: Gant diagram of initial schedule

The software of the mathematical model of the scheduling problem in the identical parallel machine environment, which takes into account the sequence-dependent setup time, in the GAMS program within the scope of the defined data. Optimal solution values provided by GAMS are given in detail in Appendix A. According to the solution obtained, the Gantt diagram showing the assignment of jobs to parallel machines is given in Figure 4.3. The time intervals indicated in yellow indicate the setup times. While orders 2,4,6 and 7 are assigned to the first machine, orders 1, 3 and 5 are assigned to the second machine. Order 7 (5), order 1 (9), order 4 (13), order 5 (20), order 6 (25), order (36) and the last completed order 2 were completed in 48 hours, respectively. Order 7 was completed 11 hours early, while ordering 2 was delayed 23 hours. The objective function that calculates the total delay was determined as 27.

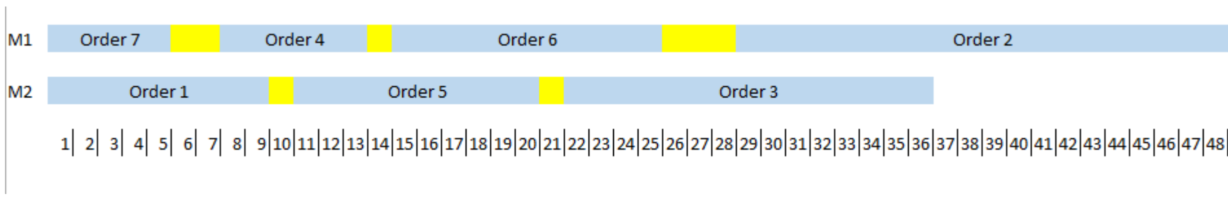

Figure 4.3: Gant diagram of model

CHAPTER 5

SENSITIVE ANALYSIS

Sensitivity analysis is important and necessary in real life scheduling problems, as parameters are variable and difficult to predict. In the sensitivity analysis, the value changes in the objective function and constraint coefficients and source values and the change in the optimal solution when a new variable and a new constraint are added are examined. In this part of the study, the changes in the outputs were examined by making changes in the inputs for the scenarios created.

5.1. First Scenario

In the first scenario, the change of the order quantities and thus the process times in the objective function was examined. The modified order quantity and process times are given in Tables 5.1 and 5.2. The results obtained are shown in the Figure 5.1 Gantt diagram of the first scenario. According to this Orders 5, 6, 4 and 2 are assigned to the first machine, while orders 7, 1, 3 are assigned to the second machine, respectively. Order 5 (2), order 7 (5), order 6 (14), order 1 (19), order 4 (24), order 3 (41) and the last completed order 2 were completed in 45 hours. Order 5 was completed 18 hours early, while order 2 was completed 20 hours late. The objective function was calculated as 21.

Table 5.1: Modified quantity of order

Parts	Unit processing time (minute)	Quantity of the order
Part 1	3,6	200
Part 2	12	90
Part 3	20	60
Part 4	3	180
Part 5	1	120
Part 6	11	60
Part 7	10	30

Table 5.2: Modified processing time

Order	Processing time of order (hour)
Order 1	12
Order 2	18
Order 3	20
Order 4	9
Order 5	2
Order 6	11
Order 7	5

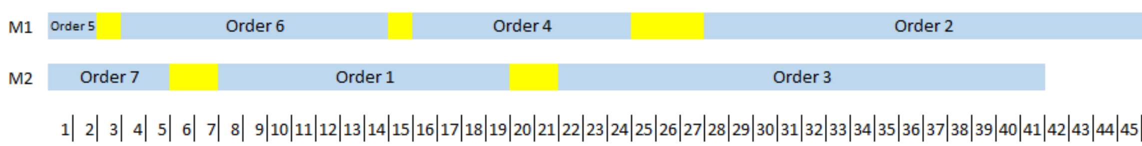

Figure 5.1: Gant diagram of first scenario

5.2. Second Scenario

In the second scenario, the effect of changes in due dates on the objective function is examined. The modified due dates are given in Table 5.3. The results obtained are shown in the Figure 5.2 Gantt diagram of the second scenario. According to this Orders 7, 4, 6 and 2 are assigned to the first machine, while orders 1, 5 and 3 are assigned to the second machine, respectively. Order 7 (5), order 1 (9), order 4 (13), order 5 (20), order 6 (25), order 3 (36) and the last completed order 2 were completed in 48 hours. Order 7 was completed 13 hours early, while order 2 was completed 18 hours late. The objective function was calculated as 18.

Table 5.3: Modified due dates of order

Order	Due date of order (hour)
Order 1	15
Order 2	30
Order 3	20
Order 4	10
Order 5	25
Order 6	20
Order 7	18

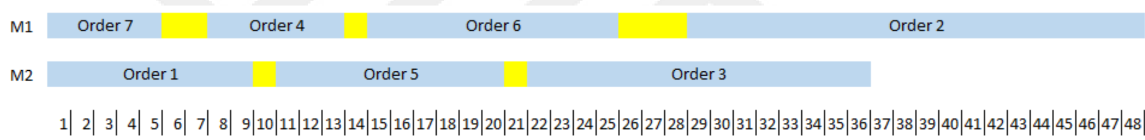

Figure 5.2: Gant diagram of second scenario

5.3. Third Scenario

Finally, the effect of the change in setup times on the objective function was investigated. The modified due dates are given in Table 5.4. The results obtained are shown in the Figure 5.3 Gantt diagram of third scenario. According to this Orders 4, 5 and 3 are assigned to the first machine, while orders 7,6,1 and 2 are assigned to the second machine, respectively. Order 7 (5), order 4 (6), order 5 (17), order 6 (17), order 1 (27), order 3 (35) and the last completed order 2 were completed in 48 hours. Order 7 was completed 11 hours early, while order 2 was completed 23 hours late. The objective function was calculated as 26.

Table 5.4: Modified required setup times

	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6	Order 7
Order 1	0	1	3	2	2	1	3
Order 2	1	0	2	2	3	1	2
Order 3	3	2	0	3	3	2	1
Order 4	2	2	3	0	1	2	3
Order 5	2	3	3	1	0	2	2
Order 6	1	1	2	2	2	0	1
Order 7	3	2	1	3	2	1	0

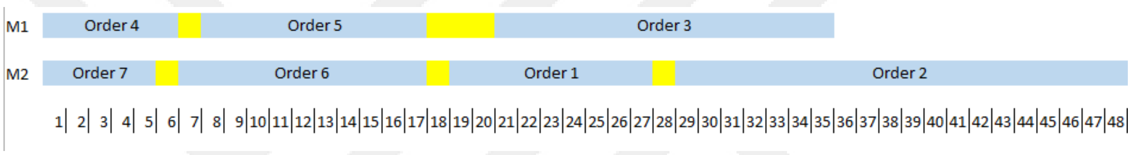

Figure 5.3: Gant diagram of third scenario

CHAPTER 6

RESULTS AND DISCUSSION

In many industrially developed countries, great efforts are made to increase production efficiency. One of the mainly prominent operations in manufacturing systems is production planning. Production Planning is the process of allocating a narrow amount of resources to a set of jobs in order to optimize certain performance measures over time. The real-life planning environment is extremely complex due to the dynamic character of production systems, such as the dynamic arrival of customer orders, the occurrence of unexpected perturbations, transforming priorities of jobs, and so on. To ensure sustainability in this environment, producing an effective production plan plays a key role.

Better quality, lower cost and shorter lead times are aimed with production planning. Each job should be finalized as soon as feasible to the deadline. A job completed before the due date will result in inventory carrying, storage and insurance costs, while a job completed after the due date will result in undesirable situations such as customer dissatisfaction and loss of reputation. In production planning, the main decision is to assign the jobs to the machines and determine the order in which they will be processed. While creating the plan, many criteria such as machine setup times and frequency should be taken into consideration as well as the priority of the work or delivery times.

In this study, an identical parallel machine scheduling problem with n parts and m machines is discussed. There is a sequence dependent setup time between sequential jobs. The purpose of the problem is to decide in which order each piece will be made on which of these two machines so that the total delay is minimized. Since scheduling problems are known to be difficult, the optimal solution has been found for the small size data set.

The proposed model is solved in the GAMS program, and the output is given in the Gantt diagram. Of the seven orders, four were assigned to machine 1, while three were assigned to machine 2. order 7 and order 1 were completed before the delivery date, and order 2 was the latest completed according to the delivery date. While the total tardiness was calculated as 37 according to the schedule created by using the heuristic method, considering the delivery date, the objective function decreased to 27 with the proposed model. In addition, sensitivity analysis was performed for 3 scenarios. The effects of changes in process time, due dates and setup times on the objective function are analyzed and the results are shared.

Various studies can be done as a continuation of this study. Models that take into account uncertain machine failures or tool changes can be produced to deal with real-life problems. Another situation that can be considered is that for larger problems in different machine environments, solutions can be proposed for different objective functions with metaheuristic methods.

REFERENCES

[1] Kiran D. R.. In: Production Planning and Control

[2] https://www.planettogether.com/blog/five-types-of-production-planning

[3] Pinedo, Michael L. (2008). Scheduling: Theory, Algorithms and Systems (Third Edition). New York: Springer Science+Business Media, LLC.

[4] K. Baker, Elements of Sequencing and Scheduling, Tuck School of Business Admimistration College, Hanover, NH, 1997.

[5] Q. V. Dang, T. van Diessen, T. Martagan, and I. Adan, “A matheuristic for parallel machine scheduling with tool replacements,” Eur. J. Oper. Res., vol. 291, no. 2, pp. 640–660, 2021

[6] J. C. Yepes-Borrero, F. Perea, R. Ruiz, and F. Villa, “Bi-objective parallel machine scheduling with additional resources during setups,” Eur. J. Oper. Res., vol. 292, no. 2, pp. 443–455, 2021,

[7] J. G. Kim, S. Song, and B. J. Jeong, “Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times,” Int. J. Prod. Res., vol. 58, no. 6, pp. 1628–1643, 2020

[8] L. Fanjul-Peyro, R. Ruiz, and F. Perea, “Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times,” Comput. Oper. Res., vol. 101, pp. 173–182, 2019

[9] R. Gedik, D. Kalathia, G. Egilmez, and E. Kirac, “A constraint programming approach for solving unrelated parallel machine scheduling problem,” Comput. Ind. Eng., vol. 121, no. May, pp. 139–149, 2018

[10] D. Wang, Y. Yin, and T. C. E. Cheng, "Parallel-machine rescheduling with job unavailability and rejection," Omega (United Kingdom), vol. 81, pp. 246–260, 2018,

[11] Y. Yin, Y. Wang, T. C. E. Cheng, W. Liu, and J. Li, "Parallel-machine scheduling of deteriorating jobs with potential machine disruptions," Omega (United Kingdom), vol. 69, pp. 17–28, 2017

[12] M. Afzalirad and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," Comput. Ind. Eng., vol. 98, pp. 40–52, 2016,

[13] E. M. González-Neira, R. G. García-Cáceres, J. P. Caballero-Villalobos, L. P. Molina-Sánchez, and J. R. Montoya-Torres, "Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria," Comput. Ind. Eng., vol. 101, pp. 128–144, 2016

[14] J. Yoo and I. S. Lee, "Parallel machine scheduling with maintenance activities," Comput. Ind. Eng., vol. 101, pp. 361–371, 2016,

[15] R. Alvarez-Valdes, J. M. Tamarit, and F. Villa, "Minimizing weighted earliness-tardiness on parallel machines using hybrid metaheuristics," Comput. Oper. Res., vol. 54, pp. 1–11, 2015,

[16] P. Fattahi, S. M. H. Hosseini, F. Jolai, and R. Tavakkoli-Moghaddam, "A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations," Appl. Math. Model., vol. 38, no. 1, pp. 119–134, 2014,

[17] B. J. Joo, Y. C. Choi, and P. Xirouchakis, "Dispatching rule-based algorithms for a dynamic flexible flow shop scheduling problem with time-dependent process defect rate and quality feedback," Procedia CIRP, vol. 7, pp. 163–168, 2013

[18] Y. Xi and J. Jang, “Scheduling jobs on identical parallel machines with unequal future ready time and sequence dependent setup: An experimental study,” Int. J. Prod. Econ., vol. 137, no. 1, pp. 1–10, 2012

[19] W. C. Lee, J. B. Lin, and Y. R. Shiau, “Deteriorating job scheduling to minimize the number of late jobs with setup times,” Comput. Ind. Eng., vol. 61, no. 3, pp. 782–787, 2011,

[20] S. Khalouli, F. Ghedjati, and A. Hamzaoui, “A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop,” Eng. Appl. Artif. Intell., vol. 23, no. 5, pp. 765–771, 2010,

[21] Fang, J., Xi, Y. (1997). "A Rolling Horizon Job Shop Rescheduling Strategy in the Dynamic Environment", International Journal of Advanced Manufacturing Technology, 13, s. 227–232.

[22] D. Rahmani and R. Ramezanian, “A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study,” *Comput. Ind. Eng.*, vol. 98, pp. 360–372, 2016,

[23] D. Ouelhadj and S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” *J. Sched.*, vol. 12, no. 4, pp. 417–431, 2009,

[24] X. Xiong, D. Wang, T. C. Edwin Cheng, C. C. Wu, and Y. Yin, “Single-machine scheduling and common due date assignment with potential machine disruption,” *Int. J. Prod. Res.*, vol. 56, no. 3, pp. 1345–1360, 2018,

[25] A. Allahverdi, “The third comprehensive survey on scheduling problems with setup times/costs,” *Eur. J. Oper. Res.*, vol. 246, no. 2, pp. 345–378, 2015,

[26] A. Hamzadayi and G. Yildiz, “Event driven strategy based complete rescheduling approaches for dynamic m identical parallel machines scheduling problem with a common server,” Comput. Ind. Eng., vol. 91, pp. 66–84, 2016

[27] Mokotoff E. Multi-objective simulated annealing for permutation flow shop problems. In: Chakraborty UK, editor. Computational intelligence in flow shop and job shop scheduling. Berlin, Heidelberg: Springer-Verlag;. p. 101–150, 2009

[28] L. Lassig, F. Mazzer, M. Nicolich, and C. Poloni, “Hybrid Flow Shop Management: Multi Objective Optimisation,” Procedia CIRP, vol. 62, pp. 147–152, 2017.

[29] Broyden, C.G., A Class of Methods for Solving Nonlinear Simultaneous Equations, American Mathematical Society, Vol. 19, No. 92, 577-593,1995

[30] Jain, A.S. and S. Meeran. “Deteministic job-shop scheduling: Past, present and future”, European Journal of Operational Research, 113, 390-434, 1999

[31] Baker, K.R. and D. Trietsch. Principles of Sequencing and Scheduling. New Jersey: Wiley, 47, 2009

[32] Lageweg, B.J., J.K. Lenstra and A.H.G. Rinnooy Kan. (1977). “Job-shop Scheduling by Implicit Enumeration”, Management Science, 24(4), 441-450.

[33] C. Hueber, G. Fischer, N. Schwingshandl, and R. Schledjewski, “Production planning optimisation for composite aerospace manufacturing,” Int. J. Prod. Res., vol. 57, no. 18, pp. 5857–5873, 2019

[34] Wang, H. “Flexible flowshop scheduling: optimum, heuristics and artificial intelligence solutions”, Expert Systems, 22(2), 78-85, 2005.

[35] Karaboğa, D.. Yapay Zeka Optimizasyon Algoritmaları, İstanbul: Atlas Yayın Dağıtım. 2004

[36] Cura, T. Modern Sezgisel Teknikler ve Uygulamaları, İstanbul: Papatya Yayıncılık. 2008

APPENDICES

Appendix A: Output of GAMS

G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m E x e c u t i o n

```
----    127 VARIABLE X.L

            1          2          3          4

1.2       1.000
2.1                                        1.000
3.2                             1.000
4.1                  1.000
5.2                  1.000
6.1                             1.000
7.1       1.000

----    127 VARIABLE Y.L

INDEX 1 = 1

            1

2.5       1.000

INDEX 1 = 4

            2
```

1.6 1.000

INDEX 1 = 5

2

2.3 1.000

INDEX 1 = 6

3

1.2 1.000

INDEX 1 = 7

1

1.4 1.000

---- 127 VARIABLE W.L

	2	3	4	5	6
1				1.000	
4					1.000
5		1.000			
6	1.000				
7			1.000		

---- 127 VARIABLE A.L

	2	3	4	5	6
1				9.000	
4					13.000
5		20.000			
6	25.000				
7			5.000		

---- 127 VARIABLE F.L

1 9.000, 2 48.000, 3 36.000, 4 13.000, 5 20.000, 6 25.000 7 5.000

---- 127 VARIABLE T.L

1 -1.000, 2 23.000, 3 14.000, 4 1.000, 6 1.000, 7 -11.000

---- 127 VARIABLE Z.L = 27.000

EXECUTION TIME = 25.454 SECONDS 5 MB 34.3.0 rac355f3 WEX-WEI

USER: GAMS Demo license for ummuhan arikan G210420|0002CO-GEN

gazi university, Turkey DL039025

****** FILE SUMMARY**

Input C:\Users\Uzba\Documents\gamsdir\projdir\Untitled_10.gms

Output C:\Users\Uzba\Documents\gamsdir\projdir\Untitled_10.lst

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Ümmühan Zeynep Beyza ARIKAN

EDUCATION

High School	:	Fethiye Kemal Mumcu High School
Bachelor	:	Gazi University / ANKARA
Master Degree	:	Ankara Yıldırım Beyazıt University

TOPICS OF INTEREST

Systems Engineering

Multi Criteria Decision Making

Lean Manufacturing

Production Planning

PUBLICATIONS/PRESENTATIONS

ARIKAN Ümmühan Z. B. , YILDIZBAŞI Abdullah, ERDEBİLLİ Babek, "Using Intuitionistic Fuzzy TOPSIS in Site Selection of Wind Energy Plants",Advances in Fuzzy Systems, 2018

PUBLICATIONS/PRESENTATIONS DERIVED FROM THESIS

ARIKAN Ümmühan Z. B. , ERDEBİLLİ Babek, "Havacilik Sektöründe Faaliyet Gösteren Bir Fabrikada Optimizasyon, 39. Yöneylem Araştırması ve Endüstri Mühendisliği Ulusal Kongresi,Ankara, 2019