ACADEMIC PLATFORM

Academic Platform Journal of Engineering and Science

journal homepage: https://dergipark.org.tr/tr/pub/apjes

Scheduling Optimization in Automotive Supplier Industry under Sequence Dependent Constraints

[1]Muhammed Mohamedsalih, *[2]Fuat Şimşir,
[1]Department of Industrial Engineering, Sakarya University,
msalih.ac21@gmail.com
[2]Department of Industrial Engineering, Karabuk University,
fuatsimsir@karabuk.edu.tr

Research Paper Arrival Date: 21.04.2021 Accepted Date: 25.05.2021

Abstract

In today's production conditions, it is very important for companies to produce products that meet the determined quality standards. In addition to the production of semi-finished products produced in plastic injection lines with high quality and efficiency, it is aimed to reduce the lost time and waste materials in production. Since automobile headlights can be in many different colors, it is necessary to keep the product quality at a high level in the transition between colors in the production made by plastic injection and to reduce the number of defective products caused by color errors. In this study, the problem of a single row-dependent machine to reduce the amount of waste that occurs between color transitions in the plastic injection line of an automotive side-industry company producing headlights is discussed. Tabu Search method, which is one of the heuristic optimization algorithms was used to solve the problem. In the solution, production sequences that will reduce the amount of waste of the products in the production program were created by using the annual data kept by the production department of the factory. The algorithm was run using data from two machines out of 18, saving 134 kg of waste materials. It has been clarified that if the developed algorithm is applied to all machines in the department, productivity and product quality will be increased by saving waste materials.

Keywords: Sequence dependent setup time, Single machine scheduling, Tabu Search, Plastic Injection

1 INTRODUCTION

It has become very important for each enterprise to produce products that meet the quality standards determined in the living and production conditions of recent times. For companies to maintain their continuity, their products must meet customer expectations, but they must often offer more. Because there are many competitors that put the same product on the market and the increase in quality standards is a factor that increases customer loyalty. To ensure the standing of companies, the importance given to quality in competition is increasing. While aiming to keep the quality high, it is important to minimize the cost.

The plastic industry, like other branches of industry, has developed rapidly and still continues this development [1]. Plastic injection molding is one of the methods used in the production of plastic materials. The most important advantage of injection is that parts with complex shapes are produced in a controlled manner without the need for a final process. Plastic injection molding is one of the most widely used industrial technologies for producing high productivity plastic products.

The point that distinguishes plastic parts production from other productions is that it includes raw material and color parameters at the same time. The reason of this, to produce different products sequentially on a single machine, the colors or raw materials used in the production of the next product are required to comply with the required quality standards at the time of change [1]. To ensure this compliance, it is a must to clean the color and raw material residues of the previous produced product in the injection part of the production machines. The process of cleaning the residues in the machine is known as the ejection stage. The waste raw material that occurs in the ejection process is called a residue. The ejection process continues until the parts in the queue to be produced reach the determined quality standards. Since the products produced during this process do not comply with the determined quality standards, it also causes serious losses in time and raw materials. Therefore, it is of great importance to complete the ejection process as soon as possible to minimize the loss of time. At this point, we encounter the sorting and scheduling problem in production systems. Determining production sequences to keep the quality at the highest level and reduce the amount of waste in the plastic injection molding process is one of the interesting research problems in this area.

*Corresponding Author: Karabük University Department of Industrial Engineering, fuatsimsir@karabuk.edu.tr, +90 533 249 42 32

Doi: 10.21541/apjes.923838

Sequencing and scheduling problems in production systems are classified from different perspectives. According to the classification made by considering the number of works and machines, the problems are examined under two main categories as "single-machine" and "multi-machine" scheduling. In the classification made in terms of the complexity of scheduling problems, it is accepted that there are four titles:

- Single stage, one machine problem is the simplest type of problem. The processes here require a single operation to be processed on a single machine.
- Single stage, parallel machine problem. The parts still require a single process step to be machined on each of the parallel machines. However, in such problems, there are more than one machine processing the similar part.
- Multistage problems are those in which there is a clear order of priority between operations. The functions involved in the process are optimally arranged according to the priority relationship in the machines. These types of problems are dealt with in terms of flow type and workshop type. In flow type problems, existing tasks are handled in the same set of devices with a similar process sequence. The operation order and priority relationship for each function on the devices are the same.
- Job shop type problems are the most general, comprehensive, and complex type of problem in creating classifications. There are no restrictions on process steps in this process. In other words, in the problems of workshop type, the person in charge of planning ensures that the works are processed in different machines in their specified sequence.

The simplest and special case of these problems is single-machine scheduling and it is one of the scheduling problems where jobs are processed sequentially. [2]. The problem of optimal scheduling of semi-finished products in injection molding machines is a good example of single machine scheduling problems [3]. Finding the best solutions for scheduling problems, which are included in combinatorial optimization problems, becomes more difficult as the size of the problem increases. That is why instead of methods that provide exact solutions in large-scale scheduling problems in the NP-Hard class, heuristic methods which may not reach the best solution but with short processing time are used.

Plastic injection department is the first stage of production in the company where the developed algorithm is applied. In this section, the required semi-finished products (lens and body) are produced. In order to produce a product in injection machines, the mold of this product must be clamped to the relevant machine [4]. Injection molding machines are identical in terms of production time. Since each mold cannot be clamped to all machines with different tonnages and dimensions, these machines are not completely alternative to each other. There is an intense setup time in this process and it is important to minimize this setup time. A second situation is that more than one product is produced with different raw materials on a machine. A certain amount of residue and waste arises due to the difference in raw materials in product transitions. The residue is the residual semi-finished product that is produced when the machine is cleaned. The semi-finished products produced in the production trials that take place after the cleaning process create the waste.

The contribution of this study is to determine the production sequence that will provide high quality and efficiency, low lost time and waste amount. The company works with a 100% quality policy and does not include any errors in the production line in line with this policy. The Plastic Injection department, which is in the first phase of the production line, is the process that forms the basis of the production. The loss of time and error in plastic injection affects other production lines. The decrease in the time and error rate gained in Plastic Injection will greatly affect the reduction of production waste costs and increase of production capacity. The gains in the work made strengthen the Just-In-Time Delivery Policy offered to the customer and ensure that orders are produced and shipped on time with minimum time and error.

The variables (time, cost, distance, wastage, etc.) between the elements of most problems whose solutions in the literature are based on scheduling and series creation were accepted as symmetrical. In our study, there is no symmetry in the transition between the elements / products due to the color effect. For example, in case of transition from product A to product B, the same amount of residue as in transition from B to A does not occur. This means more data and therefore longer solution time. In our study, it is aimed to create a production order that will create the least waste amount in the transition between products (during production) by using the transition data in the database of the company. Minimization of wastage is also directly related to the above-mentioned ejection process. The short duration of the ejection process depends on switching from light colored raw material to dark colored raw material or continuing production with the same raw material but switching from dark-colored raw materials to light-colored raw materials or continuing to produce with different raw materials may prolong the duration of the ejection process by a large amount. The purpose of this study is to create an optimal schedule to determine the preparation time of product transitions, it aims to minimize the amount of waste arising in the production trials, thus the production trial period and the amount of residue generated during the passes, accordingly, the machine cleaning time.

In the next parts of the study first, the literature was reviewed and examples were given, and as a result of the search, the appropriate methodology was determined and explained. Afterwards, the application of the methodology to the problem was explained and the results were analysed. In the last part, there is a conclusion and discussion part.

2 LITERATURE REVIEW

When the literature is examined, it is seen that there are studies conducted in various fields and with different solution methods for the sequence dependent scheduling problem. Viagas and Costa (2021) developed two different algorithms, one is constructive heuristic based on a beam search strategy and the other is metaheuristic population-based iterated greedy algorithm to solve the single machine scheduling problem. They aimed to minimise the makespan with release dates and sequence-dependent setup times. The population-based iterated greedy algorithm has been outperformed better results than the other algorithm and the most promising examples in the literature. [5]. Zhao et al. (2020) studied queue-linked single machine scheduling problem. In this problem, job sequences and processing times are not previously determined. The aim is to minimize the maximum delay. Although this problem is one of the most important production scheduling problems, it is underrepresented in the literature. In this study, a dynamic differential evolution algorithm was proposed to solve this problem. The local search method has been used to create a starting solution. Results were tested on 1000 samples from the literature. The algorithm proposed in the calculation results has been shown to be quite effective compared to known methods [6]. Mustu and Eren (2018) addressed the single machine scheduling problem to minimize total delay with a sequence-dependent learning effect and sequence-dependent setup times. They developed a branch-bound algorithm that includes some dominance properties and a lower bound to get the best solution. In addition, four well-known heuristics, two types of genetic algorithms and two types of variable neighborhood search algorithms, have been proposed to obtain the best solution. The results showed the efficiency of the Branch and Pound algorithm, as it could solve problems of smaller sizes or equal to 30 in reasonable time. [7]. Perez-Gonzalez and Framinan (2018) discussed the single-machine scheduling problem that aims to minimize the time to complete the final job (Cmax) in their study. Mixed integer programming was used in modeling the problem, and heuristic methods were used in the solution approach. [8]. Rosa et al. (2017) addressed a single machine scheduling problem with distinct time windows and sequence-based setup times. The goal is to reduce the overall early and the likely delay. In order to determine start times, a general variable neighborhood search algorithm (GVNS) and an implicit enumeration algorithm (IE) has been proposed. The (IE) algorithm found a result able to be suitable for problems of less than or equal to 15 jobs. A general variable neighborhood search algorithm produces good quality solutions for problems of larger sizes. [9]. Ben-Yehoshua and Mosheiov (2016) addressed the single-machine scheduling problem and aimed at minimizing the total number of jobs completed early. Dynamic programming algorithm is used to solve the problem. [10]. Herr and Goel (2015) discussed the problem of product transients in sequential single machine problems in their studies. In their study, the goal is to find a program that minimizes the total delay based on the given deadlines of the jobs. A mathematical formulation and heuristic solution approach are presented for two types of the problem. The computational results showed the superiority of the proposed inference in terms of time and solution quality over other commercial programming solutions.[11].

Prabhakar (2014) tackled one type of production scheduling problem with sequencing type present in various production facilities in the chemical industry. The problem is solved by using a mixed integer programming problem and branch and boundary algorithm [12]. Cheng et al. (2011) showed that single machine scheduling problems can be solved polynomially to minimize the production time, total completion time, and square sum of completion times. They also showed that if the working times and due dates are appropriate, the problems of minimizing the total delay and the maximum delay can be solved polynomially [13]. Wang and Li (2011) discussed the effects of general situational and time-dependent learning and the problems of scheduling sequential setup times for a single machine [14]. Türker and Sel (2011) presented an intuitive approach to minimize the completion time of the job scheduling problem, where the preparation time in parallel machines is sequence dependent, in other words, the setup time of the job differs depending on the previous job and the setup operations are carried out by a team. An approach that uses genetic algorithm and tabu search approaches together has been proposed for problem solving [15].

Zhao and Tang (2010), Wang et al. (2009) and Savard et al. (2008), addressed single machine scheduling problems related to lead times and time to complete jobs in their works [16,17,18]. Polynomial time algorithms have been proposed to solve these types of problems. Karabati and Akkan (2006) A branch bound algorithm (B & B) is presented to minimize the sum of preparation times in the sequence-dependent setup time in single machine scheduling problem. As a result, it has been shown that the B & B algorithm used can solve problems of up to 60 jobs and 12 groups, where setup and processing times are evenly distributed in various combinations of [1.50] and [1,100] ranges [19]. Rabadi et. al. (2004) proposed a branch and boundary algorithm for the solution of the problem with a main function consisting of a linear combination of total early termination and total delay. They have shown that they can solve problems up to 25 jobs in a reasonable time with the proposed method. [20]. Gupta and Smith (2006) is discussed the single machine scheduling problem to minimize the sequence-dependent setup times and total delay. Two heuristic algorithms have been proposed for this problem, namely the space-based local search and the Greedy Random Adaptive Search Procedure (GRASP). Comparisons were made with Simulated Annealing, Genetic Algorithm, Branch and Bound and Ant Colony to show that their algorithms perform very competitive [21]. Armentano and Mazzini (2000), proposed a genetic algorithm whose parameters are determined by statistical method for a single machine scheduling problem to minimize the total delay. It has been shown that the solutions obtained with the proposed algorithm for small-sized problems give better results by comparing them with the heuristic methods in the literature [22]. Lee et al. (1997) aimed to minimize the total weighted delay in a single machine with sequence dependent setup times. It has been proposed a three-phase heuristic method for this problem. The computational experiments emphasized that the algorithm can be successfully applied in an industrial planning system. [23]. Tan and Narasimhan (1997) addressed the problem of minimizing delay in sequence dependent scheduling. Where the delay time was reduced by using the simulated annealing algorithm, and the experimental results showed that the algorithm is able to find a good quality solution in record time compared to other algorithms. [24]. Shin et al. (2002), proposed a tabu search algorithm to reduce the completion time for several functions executed on the same machine. Each function has a start time, end time and turnaround time dependent on the sequence. The first solution was formed by MATCS (apparent delay cost modified with settings) rule, then the Tabu Search algorithm was used to find the optimal solution. Experimental results have shown that the taboo search algorithm obtains much better solutions than the heuristic method. [25]. Rubin and Ragatz (1995) discussed the application of a sequential scheduling problem. They aimed to determine the order of a series of n jobs on a single machine that will minimize the total delay of the jobs. In this study, the method of genetic algorithm was used to solve the problem and the performance of genetic search was compared with a branch and boundary algorithm with a pure random search on a set of 32 test problems [26].

Automotive manufacturers reacts zero tolerance for major problems, such as color tone defects in any part of automobiles. In terms of spare part suppliers, production plans that will not cause such errors and increase waste raw material costs are needed. It is aimed to contribute to the literature with this study made for the optimization of the use of expensive raw materials in the automobile sub-industry, where part costs are high and quality is given a high level of importance. The method can also be considered as a model that can be used in different sectors with similar production styles. Tabu Search is one of the important optimization algorithms used in solving combinatorial optimization problems. Tabu Search Algorithm avoids falling into local solution points in the search area and aims to reach the global best. In order to achieve this goal, it provides speed and efficiency in reaching the optimum solution since it ensures that the solutions visited in the previous iterations are banned.

In this study, it is aimed to reduce the lost time and wastage in the Plastic Injection section, which is at the first stage of the production line of an automotive supplier company and forms the basis of its production. To achieve this goal, a solution to a real scheduling problem is offered using the Tabu Search algorithm, which can create production sequences with less waste by using data that has been realized for a year.

3 TABU SEARCH

The Tabu Search (TA) technique, named after the word Tabu, which is used to express the sacred prohibitions in primitive tribes, is a heuristic algorithm proposed by Glover (1989,1990) to be used in the solution of combinatorial optimization problems [27,28]. Algorithm is a heuristic technique that guides local search methods to get rid of local best solutions and reach global best [29]. In order to prevent the steps leading to the final solution from making circular movements, it is ensured that repetition is prohibited or punished in the next cycle. Tabu is a successful algorithm that progresses in an iterative way (sequential processes) used in solving search optimization problems. The first step in TA is to create the initial solution. After that, it continues with the stages of creating the candidate list of the movements, selecting the most suitable candidate from this list, and repeating or terminating the search according to the stop criteria. The flow chart showing the stages of the Tabu Search algorithm is given in Figure 2.1. The basic components of the algorithm and the operation of these components are explained in the following section under subheadings.

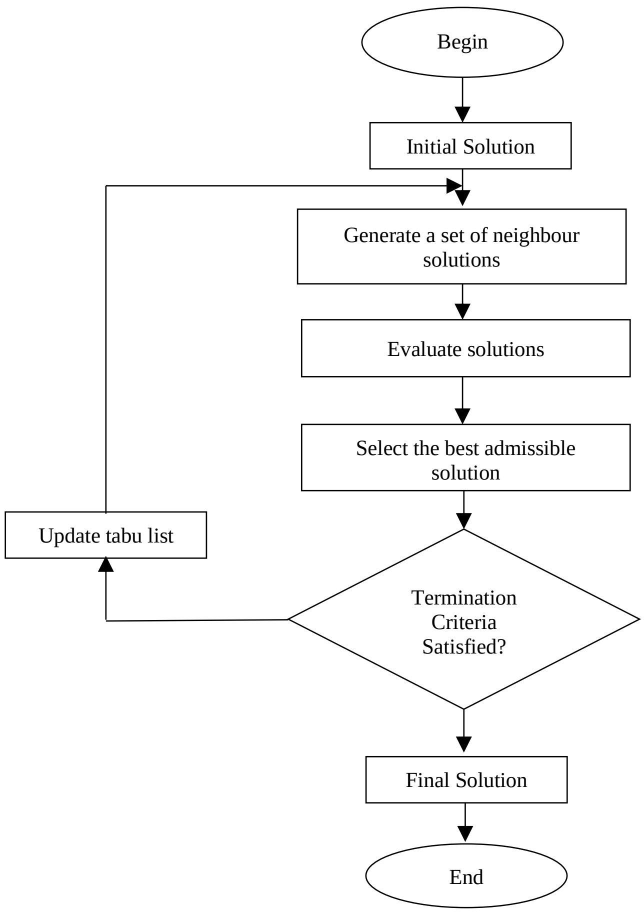

Figure 3.1 Tabu Search algorithm's Flow chart

3.1 Tabu List

The nature of this list is that the current tabu list can update itself through a ranking and comparison process after determining possible candidate solutions. For this purpose, this algorithm developed a tabu list in which the list of solutions found in previous loops is recorded. If a potential solution candidate is the same or has similar characteristics as a tabu-listed solution, that solution will not be considered. While creating the tabu list, the best potential solution is added to the list in line with the purpose function in each cycle, in case the capacity of the tabu list is exceeded, the initial solutions in the list are deleted from the list, and the new solutions obtained in the last loops are added to the list [FIFO]. When the tabu list is full and there is a new banned solution, as shown in Figure 3.1, the oldest available solution is removes from the list and the new solution is registers as the newest solution.

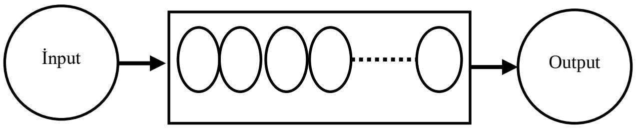

Figure 3.2 The structure of the Tabu List

3.2 Tabu Tenure

Proximity-based memory works with a tabu list that holds the tabu active elements. Each tabu active element is kept in the tabu list for a specific iteration period. When this period is completed, the tabu active element is will be removed from the list and the solutions containing these elements will be not tabu. The length of time (the number of iterations) that an element included in the tabu list will be kept in this list is generally determined by the size of the problem. The point to be considered in determining the tabu period is that this period will neither be small nor large. When the tabu duration is small, it can cause the algorithm to hang likewise this time may negatively affect the quality of the solution. A balance can be struck between the variable selection of the tabu retention period, meticulous examination of one region of the solution space, and then moving to another solution area [30].

3.3 Short Term Memory Strategy

Since the tabu search method is a general method, there are no definite principles regarding the application method [32]. In order to reach a satisfactory solution with the tabu search method, additions are often required according to the problem structure. Therefore, when designing the method, a suitable memory should be designed for the problem that is sought. In problems where short-term memory is used, an element marked as active is kept in its proximity-based memory, and in the next iterations, while searching for a solution, it is prevented to go to the area of that element. Although short-term memory does not obtain extensive information such as long-term memory, it is a highly preferred method in many problem structures. The working steps of short-term memory are given in Figure 3.2.

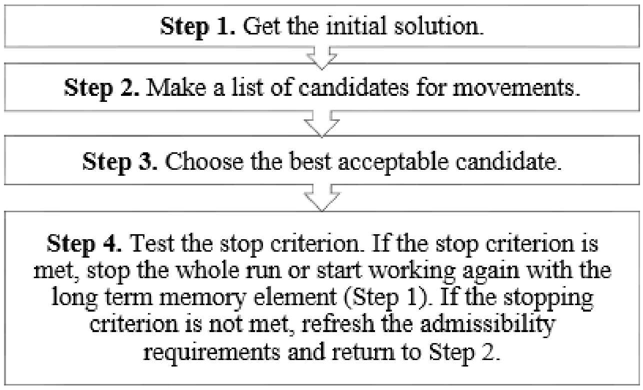

Figure 3.3 Short-term memory work steps

3.4 Long-Term Memory Strategy

For many types of problems, short term memory is quite sufficient to produce good quality solutions. By adding long-term memory, the method is getting stronger in finding the best solution [31]. Long-term memory saves information from all possible solution areas without being limited to specific field information during the operation of the algorithm, and the progress of the algorithm is monitored, and solutions that approach the global area can be accepted and solutions that lead to local areas can be prevented through certain strategies. For example, when registering elite solutions, only solutions that are significantly different from previous solutions can be registered. And when each elite solution is selected for review, the short-term memory of the previous solution can be freed.

4 PROBLEM DEFINITION

The company, which manufactures headlights in the automotive supplier industry, has 16 injection machines in the plastic injection department. The study focuses on machines 06-19 and 06-28, using one-year products production sequence information recorded by the company's production department. The explanation of the problem and the explanation of the problem data will be made over the information of the 06-19 coded machine. The algorithm solution results of the other machine will be discussed in the application part.

In Table 4-1, production transition data of 11 products in the 06-19 coded machine are shown. In the table, the raw materials of the previous and next product, the amount of the wasted residues formed, the number of produced product and duration of production are shown. Production transition number 1 in the table, it shows that the machine with the code 06-19 switched from producing the Aa87 / 88 coded product to the Bf32 product. The raw materials of these products are Aa81 and Ab80, respectively. In this transition, the production of the Bf32 coded product takes 240 minutes and it is shown that 145 pieces are produced, while 2.5 Kg of wasted residues is produced in this operation.

Table 4-1 Data samples of 06-19 coded machine

#	Previous Product	Prev. Raw Mater.	Next Product	Next Raw Material	Resid. amount (kg)	Prod. Time (min)	Prod. Unit (Piece)
1	Aa87/88	Aa81	Bf32	Ab80	2,50	240	145
2	Bf32	Ab80	Bf32	Ab80	3,50	525	417
3	Bf32	Ab80	Ae50/51	Ab44	14,8	365	548
4	Ae50/51	Ab44	Ab70/71	Bb44	8,80	525	1168
5	Ab70/71	Bb44	Ai70/71	Aa53	4,54	545	2048
6	Ai70/71	Aa53	Bh50/51	Aa14	2,41	550	854
7	Bh50/51	Aa14	Aa87/88	Aa81	5,00	525	74
8	Aa87/88	Aa81	Ar24/25	Ab400	0,00	300	704
9	Bh51/55	Aa14	Bh51/55	Aa14	7,00	180	252
10	Bh51/55	Aa14	Bh56/57	Aa14	0,00	350	496
11	Bh56/57	Aa14	Ah56/57	Aa14	5,00	120	84
12	Ah56/57	Aa14	Ar24/25	Ab400	4,70	340	780
13	Ar24/25	Ab400	Ar24/25	Ab400	4,91	590	1180
14	Ar24/25	Ab400	Aa87/88	Aa81	6,80	450	296
15	Ad96/97	Bb80	Ad96/97	Bb80	12,3	525	525
16	Ad96/97	Bb80	Ad96/97	Bb80	4,70	430	708
17	Ad96/97	Bb80	Aa87/88	Aa81	13,70	525	600
18	Ar24/25	Ab400	Ar24/25	Ab400	1,815	310	626
19	Ar24/25	Ab400	Ab70/71	Ab44	4,50	405	810
20	Ab70/71	Ab44	Ab70/71	Ab44	3,10	250	518
21	Bf32	Ab80	Bh51/55	Aa14	4,64	255	336
22	Bh51/55	Aa14	Bh51/55	Aa14	5,34	360	508
23	Bh51/55	Aa14	Bh51/55	Aa14	3,75	400	568
24	Bh51/55	Aa14	Bh56/57	Aa14	3,88	310	512
25	Bh56/57	Aa14	Bh56/57	Aa14	3,20	330	420

5 APPLICATION

Tabu search method was used for the solution of the problem presented in Chapter 4. The tabu algorithm was written using the Python programming language using the Spyder operator, where the Panda and Numpy libraries specialized in data analysis and mathematical operations were used to write the algorithm. The program was run on a computer with a Core i5 processor and 4GB RAM.

Table 5-1 Parameters of the tabu algorithm

	problem size (n)	Tabu Tenure	stop criterion
06-19	36	$\sqrt{36} = 6$	30000 iteration
06-28	21	$\sqrt{21} \cong 5$	30000 iteration

Table 5-1 shows that the algorithm parameters have been defined in line with the problem presented in our study. Where Tabu Tenure represents the number of steps in which the chosen solution in the current step will be blocked, and the stop criterion was defined as the arrival of the algorithm at 30,000 steps. Tabu search algorithm was first run with the monthly order information of the 06-19 coded machine, given in Table 5-2.

Table 5-2 One month of product order data

C	Week 1	Week 2	Week 3	Week 4
1	Bf32	Bh51/55	Bf32	Bh50/51
2	Aa87/88	Bh56/57	Ac72/73	Bh51/55
3	Ab70/71	Ac72/73	Aa87/88	Bh56/57
4	Ad96/97	Ab11/12	Ab70/71	Ab11/12
5	Ae50/51	Ad96/97	Ad96/97	Ad96/97
6	Ah56/57	Ai70/71	Ae50/51	Ai70/71
7	Ar24/25	Aa87/88	Ah56/57	Aa87/88
8	Ai70/71	Bh56/57	Ar24/25	Bh56/57
9	Bh50/51	Ad96/97	Ai70/71	Ad96/97

5.1 Problem Size and Initial Solution

Choosing a good initial solution saves time and allows better exploration of the solution space [33]. A procedure or any heuristic method that guarantees the feasibility of the solution can be used to create the initial solution. In our study, a product series consisting of n products whose optimum order is requested by the production engineer represents the initial solution of problem. For example, after the product series to be produced in the next month is determined, this product series is entered into the program and the initial solution is shown as in Figure 5.1.

1	2	3	4	………………..	n

Figure 5.1 Problem size

5.2 Objective Function

The objective function to be used in the tabu search method is a function that minimizes the residual amount in products production transition. The initial solution used in the tabu search method and the solutions obtained in the next steps are evaluated in terms of compliance with the objective function. The objective function is in the structure of a function that minimizes the sum of the cost (Billet quantities) that resulting from the transfer of production from product i to produc $(i + 1)$. in the evaluated product series. The number one equation represents the object function where $C_{i,(i+1)}$ represents the amount of residues resulting from the transfer of production from product i to product $(i + 1)$.

$$Min \sum_{i=1}^{n-1} C_{i,(i+1)} \tag{1}$$

5.3 Mechanism of Movements

In the transition from an existing solution to a neighboring solution, the movements of displacement and intervention were used. Within the limits of the size of the problem $(0 < r \leq n)$, two unequal $(r1, r2)$ random numbers are generated. Then, the neighbor/candidate solution is generated by replacing the r1 index product with the r2 index product in current best solution series.

5.4 Candidate Movement List and Choosing the Best Movement

The candidate movement list is determined according to the strategy of choosing the best move in our problem. This strategy chooses the best action produced from the current solution to achieve improvement in the value of the objective function, provided it meets the tabu constraints and aspiration criteria. This strategy is based on the assumption that a movement chosen as the best candidate movement will lead to the best solution or optimum solution faster after a few steps [34]. The first step is to create a list of candidate solutions. Then, the tabu status and aspiration criteria are checked for each solution in this list, respectively. Objective function values of non-tabu solutions are calculated and the action with the best objective function value is considered the best solution and used to generate a new solution in the next iteration. To better understand the working mechanism of this strategy, the flowchart is shown in Figure 5.2 below.

5.5 Tabu List and Tabu Periods

A $(n \times n)$ dimensional matrix is used to check whether a move is tabu or not. This matrix records the length of time a banned solution candidate will remain on the tabu lists. An action taken to the next action, that is, an accepted action, is kept in the tabu list for a period of time. This time is recorded as t in the cell where $r1$ and $r2$ intersect. At each subsequent iteration, it is decremented by 1 until it becomes zero or until the aspiration criterion is met. Tabu times were taken statically and calculated by rounding the square root of the number of products in the list (n) to the nearest integer.

$$t = round(\sqrt{n}) \quad (1)$$

The working mechanism of the tabu list is shown in Table 5-3. Here, if the displacement between the 2nd product and the 1st product is included in the candidate movement list and is considered the best move in the candidate action list, this action is prohibited until the next round $(\sqrt{n})$ iteration.

5.6 Aspiration Criterion

In this study, tabu breaking is used according to the purpose. Accordingly, if a tabu move provides a better solution than the best solution found up to the current iteration, it can be implemented despite being tabu. However, if all the actions that can be taken from the current solution without breaking the feasibility are tabu, the current solution is updated with the move closest to the end of the tabu situation.

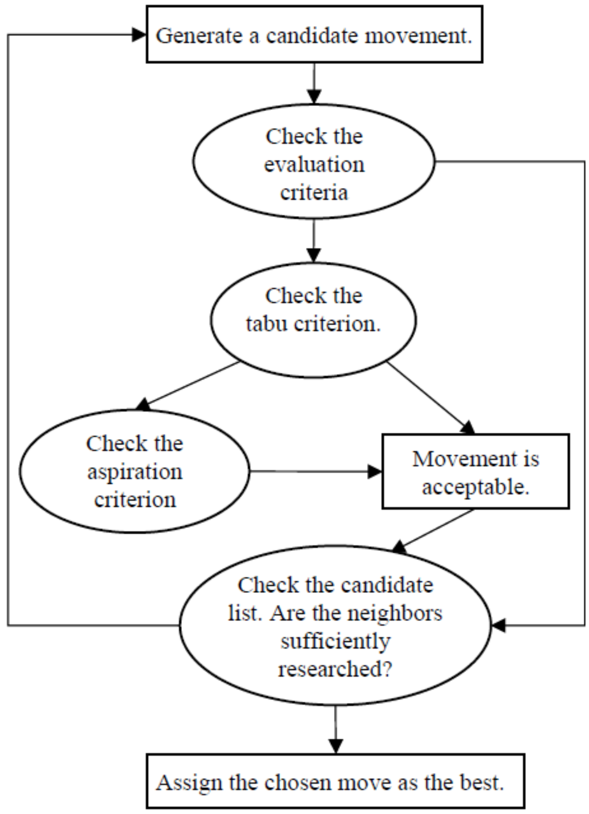

Figure 5.2 Best move selection strategy

Table 5-3 The working mechanism of the tabu list

	1	**2**	**…….**	***n***
1	0	0	……	0
2	$round(\sqrt{n})$	0	……	0
.	.	.	……	.
n	0	0	……	0

6 RESULTS

In order to achieve optimal sequencing, the tabu search algorithm coded in the Python programming language was run using the parameters defined in the previous titles. The results obtained from the runs were examined separately based on machines. Table 6-1 shows the starting sequence of the 06-19 coded machine.

Table 6-1 Starting order of products for 06-19 coded machine.

Item no	Week 1	Item no	Week 2	Item no	Week 3	Item no	Week 4
1	Bf32	10	Bh51/55	19	Bf32	28	Bh50/51
2	Aa87/88	11	Bh56/57	20	Ac72/73	29	Bh51/55
3	Ab70/71	12	Ac72/73	21	Aa87/88	30	Bh56/57
4	Ad96/97	13	Ab11/12	22	Ab70/71	31	Ab11/12
5	Ae50/51	14	Ad96/97	23	Ad96/97	32	Ad96/97
6	Ah56/57	15	Ai70/71	24	Ae50/51	33	Ai70/71
7	Ar24/25	16	Aa87/88	25	Ah56/57	34	Aa87/88
8	Ai70/71	17	Bh56/57	26	Ar24/25	35	Bh56/57
9	Bh50/51	18	Ad96/97	27	Ai70/71	36	Ad96/97

Table 6-2 Results by number of iterations for 06-19 coded machine.

Iteration no	Time	Object Function Value (Kg)
0	00:00:00	224,348
5000	00:02:28	177,000
15000	00:16:00	170,376
30000	00:28:25	164,394

As seen in Table 6-2, the algorithm's working time took approximately 29 minutes, the time varies according to the size of the problem. Even if the working time is short in a small size problem, it can take hours for large size ones. The Table 6-3 shows the optimal result given by the.

Table 6-3 the optimum solution for the 06-19 coded machine After 30 thousand iterations.

Item no	Week 1	Item no	Week 2	Item no	Week 3	Item no	Week 4
1	Bh56/57	**10**	Ah56/57	**19**	Bh50/51	**28**	Ai70/71
2	Bh56/57	**11**	Ad96/97	**20**	Ah56/57	**29**	Ab70/71
3	Bf32	**12**	Bh56/57	**21**	Bf32	**30**	Ad96/97
4	Ar24/25	**13**	Ad96/97	**22**	Ai70/71	**31**	Ai70/71
5	Ai70/71	**14**	Bf32	**23**	Bh56/57	**32**	Bh51/55
6	Aa87/88	**15**	Ab70/71	**24**	Ad96/97	**33**	Ae50/51
7	Bh51/55	**16**	Aa87/88	**25**	Ad96/97	**34**	Aa87/88
8	Bh56/57	**17**	Ar24/25	**26**	Ai70/71	**35**	Ar24/25
9	Ae50/51	**18**	Ad96/97	**27**	Aa87/88	**36**	Bh50/51

Figure 6.1, shows the solution value reached according to the number of iterations. In the light of the results obtained, the value of the objective function decreased from 224.3 Kg to 164.3 Kg in the best solution found. Thus, an improvement / saving of 60 Kg has been achieved in the use of raw materials. When the algorithm is run and the determined number of iterations is completed, the graph of the best solutions obtained with optimum ranking is shown.

In Table 6-4, shows the starting order of the product belonging to the 06-28 coded machine. The working time of the algorithm for the 06-28 machine shown in Table 6-5 is 20 minutes. At the end of this working period, it has been observed that the amount of wasted row materials has decreased by 54%.

Table 6-4 Starting order of products for 06-28 coded machine.

Item no	Week 1	Item no	Week 2	Item no	Week 3	Item no	Week 4
1	Ab02/03	**7**	Ab02/03	**13**	Ba76/77	**19**	Ba76/77
2	Bb18/19	**8**	Bb18/19	**14**	Ab02/03	**20**	Bc95/96
3	Bc97/98	**9**	Bc97/98	**15**	Bb18/19	**21**	Ab02/03
4	Ab02/03	**10**	Ba76/77	**16**	Ba78/79		
5	Bc97/98	**11**	Bc97/98	**17**	Ba76/77		
6	Ba76/77	**12**	Ab02/03	**18**	Bb18/19		

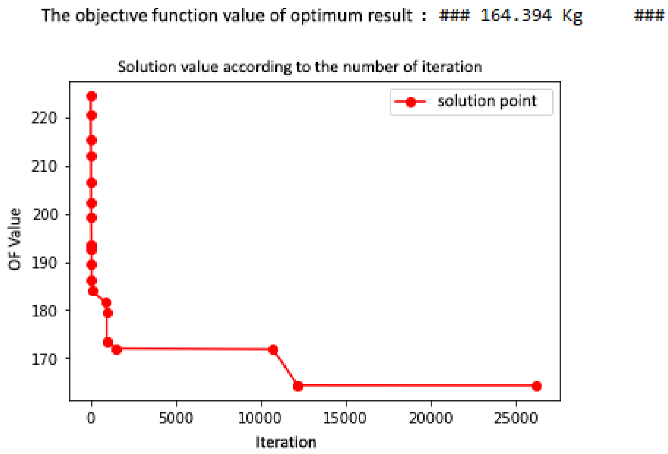

Figure 6.1 Solution graph according to iteration for 06-19 machine after 30 thousand iterations.

Table 6-5 Results by number of iterations for 06-28 coded machine.

Iteration no	Time	Object Function Value (Kg)
0	00:00:00	136,330
5000	00:02:12	93,200
15000	00:09:38	67,536
30000	00:19:44	62,024

Table 6-6, shows the algorithm's optimal result for the 06-28 coded machine. In addition, the best solutions provided by the algorithm in the working process are shown in Figure 6.2. Since the Tabu search method evaluates many potential solutions at each stage, the time to find the optimal solution can be long. However, as the duration increases, it may provide a small improvement in the target function value. Tabu search algorithm is a very general search method and the lack of precise principles about the operation of this method can provide great flexibility to the developer of the algorithm [34]. However, the success of the tabu search method depends on how well and appropriately the algorithm is designed for the relevant problem. For this reason, the advantage of the tabu search method to provide flexibility should be used well and it is extremely important to make a suitable design for the problem to be solved.

Table 6-6 the optimum solution for the machine 06-28 After 30 thousand iterations.

Item no	Week 1	Item no	Week 2	Item no	Week 3	Item no	Week 4
1	Ba78/79	**7**	Ba76/77	**13**	Ab02/03	**19**	Bb18/19
2	Ab02/03	**8**	Bb18/19	**14**	Ba76/77	**20**	Bc97/98
3	Ba76/77	**9**	Bc97/98	**15**	Bb18/19	**21**	Ba76/77
4	Bc97/98	**10**	Bb18/19	**16**	Ab02/03		
5	Ab02/03	**11**	Ba76/77	**17**	Ab02/03		
6	Ab02/03	**12**	Bc97/98	**18**	Bc95/96		

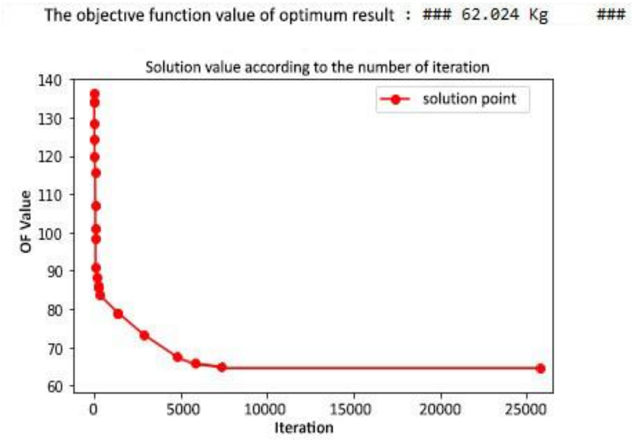

Figure 6.2 Solution graph according to iteration for 06-19 machine after 30 thousand iterations.

7 CONCLUSION

The plastic injection line of the company where this study was carried out was examined and it was observed that raw material and product wastage and lost time emerged due to the color change in semi-finished products. Accordingly, it has been determined that the problem should be handled as a sequence dependent single machine scheduling problem. With the aim of reducing lost time and wastage in the plastic injection line, one-year product sequence transition data were collected and analysed. When the structure of the problem and the data obtained were examined, it was decided to use the tabu search method for the solution with its flexible structure. In this study, it is aimed to find the order of products that will minimize the residues amounts to be produced by the monthly product orders to be produced in two machines in the injection department.

As a result of the solution, it was seen that the total amount of residues was reduced by 134 kg as a result of one month production on two machines. Although the focus of the study is two machines out of 18 machines, a large amount of raw material can be saved. If the algorithm is applied in all injection machines of the company, significant savings can be achieved in resources such as labor, material and time spent. On the other hand, in the study, orders were found to reduce waste by considering the finalized orders. Hybrid use of an artificial intelligence model, which can create more sorting options by predicting the customer orders in the future, can be carried out in future studies.

Author contributions:
Muhammed MOHAMEDSALİH: Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing - Original Draft, Visualization
Fuat ŞİMŞİR: Methodology, Investigation, Writing - Review & Editing, Supervision, Project administration

Conflict of Interest:
We do not have any conflict of interest or common interest with any institution or person that we know that could affect our work.

Financial Disclosure: We declared that this study has received no financial support.

REFERENCES

[1] T. Saraç and S. Kaya, "Plastik enjeksiyon makinelerinin vardiya bazinda çizelgelenmesi problemi için bir hedef programlama modeli," *Journal of Industrial Engineering*, vol. 24, pp. 12–26.

[2] T. Eren and E. Güner, "Sıra-Bağımlı Hazırlık Zamanlı İki Ölçütlü Çizelgeleme Problemi: Toplam Tamamlanma Zamanı Ve Maksimum Erken Bitirme," *Erciyes University Journal of Science and Technology*, vol. 23, no. 1-2, pp. 95–105, 2007.

[3] T. Eren, "Learning and Deteriorating Effects on the Single Machine Scheduling Problems," *International Journal of Engineering Research and Development*, vol. 6, no. 1, pp. 15–20, Jan. 2014.

[4] "Digital Conversion on the Way to Industry 4.0", Springer Science and Business Media LLC, 2021

[5] V.F. Viagas, A. Costa, Two novel population based algorithms for the single machine scheduling problem with sequence dependent setup times and release times, Swarm and Evolutionary Computation, Volume 63, 2021, 100869, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2021.10086.

[6] Y. Zhao and G. Wang, "A dynamic differential evolution algorithm for the dynamic single-machine scheduling problem with sequence-dependent setup times Yue," *Journal of the Operational Research Society*, vol. 71, no. 2, pp. 225–236, 2020.

[7] S. Muştu and T. Eren, "The Single Machine Scheduling Problem with Sequence-Dependent Setup Times and a Learning Effect on Processing Times.," *Applied Soft Computing*, vol. 71, pp. 291–306, 2018.

[8] P. Perez-Gonzalez and J. M. Framinan, "Single machine scheduling with periodic machine availability," *Computers & Industrial Engineering*, vol. 123, pp. 180–188, 2018.

[9] R. M. Souza, S. Ricardo, and M. Felizardo, "Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties," *Computers and Operations Research*, vol. 81, pp. 203–215, 2017.

[10] Z. Ceylan, E. Karan, Ç. Bakırcı, and S. Sabuncu "Single Machine Scheduling Problem with Sequence Dependent Setup Times : An Application in White Goods I," International Journal of Multidisciplinary Studies and Innovative Technologies, vol. 3, no. 1, pp. 14–21, 2019.

[11] O. Herr and A. Goel, "Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints," European Journal of Operational Research, vol. 248, no. 1, pp. 123–135, 2016.

[12] T. Prabhakar, "A Production Scheduling Problem with Sequencing," Management Science , vol. 21, no. 1, pp. 34–42, 2014.

[13] T. C. E. Cheng, W. C. Lee, and C. C. Wu. "Single-Machine Scheduling with Deteriorating Jobs and Past-Sequence-Dependent Setup Times." *Applied Mathematical*

Modelling 35 (4):1861–67,2011. https://doi.org/10.1016/j.apm.2010.10.015.
[14] J.B. Wang and J. X. Li., Single Machine Past-Sequence-Dependent Setup Times Scheduling with General Position-Dependent and Time-Dependent Learning Effects. *Applied Mathematical Modelling* 35 (3): 1388–95,2011. https://doi.org/10.1016/j.apm.2010.09.017.
[15] K. Türker and Ç. Sel, "Sıra Bağımlı Hazırlık Operasyonları İçin Tek Ekipli Paralel Makinalarda Çizelgeleme Problemine Karma Yaklaşım," J. Fac. Eng. Arch. Gazi Univ., vol. 26, no. 4, pp. 731–740, 2011.
[16] C. Zhao and H. Tang, "Single machine scheduling with past-sequence-dependent setup times and deteriorating jobs," Computers & Industrial Engineering, vol. 59, no. 4, pp. 663–666, 2010.
[17] J.-B. Wang, D. Wang, L.-Y. Wang, L. Lin, N. Yin, and W.-W. Wang, "Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times," Computers & Mathematics with Applications, vol. 57, no. 1, pp. 9–16, 2009.
[18] D. Biskup and J. Herrmann. "Single-Machine Scheduling against Due Dates with Past-Sequence-Dependent Setup Times", *European Journal of Operational Research* 191 (2): 587–92,2008. https://doi.org/10.1016/j.ejor.2007.08.028.
[19] S. Karabati and C. Akkan., "Minimizing Sum of Completion Times on a Single Machine with Sequence-Dependent Family Setup Times." *Journal of the Operational Research Society* 57 (3): 271–80,2006. https://doi.org/10.1057/palgrave.jors.2601989.
[20] G. Rabadi, M. Mollaghasemi, G.C. Anagnostopoulos, "A Branch-And-Bound Algorithm For The Early/Tardy Machine Scheduling Problem With A Common Due-Date And Sequence Dependent Setup Time," Computers and Operations Research, 31, 1727–1751, 2004.
[21] S.R. Gupta and J. S. Smith. "Algorithms for Single Machine Total Tardiness Scheduling with Sequence Dependent Setups." *European Journal of Operational Research* 175 (2): 722–39,2006. https://doi.org/10.1016/j.ejor.2005.05.018.
[22] V. A. Armentano and R. Mazzini. "A Genetic Algorithm for Scheduling on a Single Machine with Set-up Times and Due Dates." *Production Planning and Control* 11 (7): 713–20, 2000. https://doi.org/10.1080/095372800432188.
[23] Y.H. Lee, K. Bhaskaran and M. Pinedo. "A Heuristic to Minimize the Total Weighted Tardiness with Sequence-Dependent Setups." *IIE Transactions (Institute of Industrial Engineers)* 29 (1): 45–52,1997. https://doi.org/10.1080/07408179708966311.
[24] K.C. Tan and R. Narasimhan. "Minimizing Tardiness on a Single Processor with Sequence-Dependent Setup Times: A Simulated Annealing Approach." *Omega* 25 (6): 619–34, 1997.https://doi.org/10.1016/S0305-0483(97)00024-8.
[25] H. J. Shin, C.-O. Kim, and S. S. Kim, "A Tabu Search Algorithm for Single Machine Scheduling with Release Times, Due Dates, and Sequence-Dependent Set-up Times," The International Journal of Advanced Manufacturing Technology, vol. 19, no. 11, pp. 859–866, 2002.
[26] P.A. Rubinand and G. L. Ragatz. "Scheduling in a Sequence Dependent Setup Environment with Genetic Search." *Computers and Operations Research* 22 (1): 85–99,1995. https://doi.org/10.1016/0305-0548(93)E0021-K.
[27] F. Glover, "Tabu Search-Part I", ORSA Journal on Computing, Cilt 1, No 3, 190-206, 1989.
[28] F. Glover, "Tabu Search-Part II", ORSA Journal on Computing, Cilt 2, No 1, 4-32 1990
[29] S. Özyön , C. Yaşar ve H. Temurtaş , "Ham Enerji Kaynağı Kısıtlı Termik Birim İçeren Sistemlerde Çevresel Ekonomik Güç Dağıtımı Probleminin Çözümü", Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, c. 24, sayı. 1, ss. 47-72, Haz. 2011
[30] F. Glover and M. Laguna, "Tabu Search," *Tabu Search*, no. July, 1997, doi: 10.1007/978-1-4615-6089-0.
[31] J. Xu, M. Sohoni, M. McCleery, and T. G. Bailey, "A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 978–993, 2006, doi: 10.1016/j.ejor.2004.08.023.
[32] A. M. Mohammed and S. O. Duffuaa, "A tabu search based algorithm for the optimal design of multi-objective multi-product supply chain networks," *Expert Syst. Appl.*, vol. 140, p. 112808, 2020, doi: 10.1016/j.eswa.2019.07.025.
[33] A. Y. Abdelaziz, F. M. Mohamed, S. F. Mekhamer, and M. A. L. Badr, "Distribution system reconfiguration using a modified Tabu Search algorithm," *Electr. Power Syst. Res.*, vol. 80, no. 8, pp. 943–953, 2010, doi: 10.1016/j.epsr.2010.01.001.
[34] Wan, Guohua, and Benjamin P.-C. Yen, "Tabu Search for Single Machine Scheduling with Distinct Due Windows and Weighted Earliness/Tardiness Penalties.," European Journal ofOperational Research, vol. 142, no. 2, pp. 271–281, 2002.