APPLYING SIMULATED ANNEALING ALGORITHM FOR PARALLEL MACHINE TARDINESS PROBLEM SUBJECT TO JOB SPLITTING

Amir Arsalan Rajaee[1], Atefeh Amindoust[1*] and Milad Asadpour[1]
[1] Department of industrial engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

amirarsalanrajaei@yahoo.com
Atefeh_amindoust@yahoo.com
Milad.asadpour@sin.iaun.ac.ir

ABSTRACT

The problem of scheduling parallel machines due to its application in various industries has attracted many researchers. In a workshop with parallel machines, each production order is considered as a job so that in each case a certain number of a specific product must be delivered at the appointed time. In these workshops, each job can be split into several sub-jobs and each of these sub-jobs can be processed independently at different times or simultaneously on one or more machines. Minimizing the number of tardy jobs is a very important issue in the operations sequencing and scheduling. In many industries and service centers, what matters is not the amount of tardiness, but the presence or absence of tardiness. In this paper, the problem of scheduling parallel machines with job splitting property with the aim of minimizing the number of tardy jobs has been investigated and a mathematical mixed integer model is presented. A meta-heuristic simulated annealing algorithm has been applied to solve the model. The results show that the proposed algorithm is desirable for solving problems at a very short time.

Keywords: parallel machines, simulated annealing, scheduling, job splitting, mixed integer programming

* Corresponding Author

1 INTRODUCTION AND LITERATURE REVIEW

The classical parallel machine total tardiness problem can be stated as follows: There are n jobs to be processed on m continuously available identical parallel machines. Each machine can process only one job at a time, and each job can only be processed on only one machine. Each job is ready at the beginning of the scheduling horizon and has a distinct processing time and a distinct due date. The objective is to determine a schedule such that total tardiness is minimized, where tardiness of a job is the amount of time its completion time exceeds its due date [1].

Therefore, it is vital for many industries or even service centres to meet delivery times. Consider that an industry, faces customers who do not accept the slightest tardiness, and as soon as they are faced with a delay, they will dismiss their order. It has great importance in just-in-time production. As another example, consider a hospital that needs to minimize the amount of tardiness to schedule an operating room. In such cases, even the smallest amount of lateness can cause the death of a patient. So what is important, is not the amount of delay, but also is the existence or absence of the delay [2].

The problems of parallel machines scheduling are discussed depending on the application of different objective functions and assumptions. Considering the delivery time for each of the jobs is one of these assumptions that are expressed in problems with objective functions such as the sum of time-delayed or the number of delays. Another of these assumptions is job splitting property. In this case, each job may be broken up into several sub-jobs, and each of these sub-jobs can be processed independently at different times or simultaneously on one or more machines. That is, each job can be divided into the number of sub-jobs and each of them is processed separately on different machines. The problem of scheduling parallel machines can be considered as a two-step process. In the first step, the tasks will be allocated to the machines and in the second step, sequences are determined on each machine. But considering simultaneous assignment and sequencing will better results. It is worth mentioned that parallel machines are divided into three general categories:1- identical parallel machines with identical processing time 2- identical parallel machines with different processing time 3- non identical parallel machines.

Lee [3] developed an effective dispatching rule to fit the requirements of the acrylonitrile-butadiene-styrene (ABS) plate plant. A mixed integer programming (MIP) model is used to evaluate the effectiveness of the proposed dispatching rule. The objective is to to minimise the total tardiness. Moreover, an iterated greedy-based metaheuristic is developed to further improve the obtained solutions. The results indicated the appropriate efficiency of proposed metaheuristic. Ta, Billaut & Bouquard [4] considered the m-machine flow-shop scheduling problem. The objective is minimizing the total tardiness. To solve the problem, several metaheuristic algorithms are proposed that are compared to a genetic algorithm. The results illustrate that the metaheuristic algorithms are competitive with the genetic algorithm and except for a specific size of problems, in other sizes, the best metaheuristic algorithm outperforms the genetic algorithm.

Moslehi & Jafari [5] examined the problem of scheduling a single machine with the assumption of linear degeneracy with the goal function to minimize the number of tardy jobs, at which time the processing of tasks for each job depends on the start of that task. They provided a B&B algorithm to solve the problem. Briand & Ourari [6] considered the problem of scheduling tasks on a single machine that the time of work processing is deterministic and the objective function is to minimize the number of late tasks. They provided an integer programming method that created the upper and lower limits for the problem. Liu, Wang, Chu, C. & Chu, F. [7] considered a single-machine scheduling problem with periodic maintenance with objective of minimizing the number of tardy jobs. Because of the strong NP-hardness of the problem, they proposed a new branch-and-bound algorithm based on an efficient lower bounding procedure and several new dominance properties to solve the problem. Elyasy & Salmasi [8] considered two scheduling problems: first, single machine scheduling problem with minimizing the number of tardy jobs and second one, two machine flow shop scheduling problem with a common due date and minimizing the number of tardy jobs. Processing times were assumed as independent random variables. They proposed a solution based on chance constrained programming. Herr & Goel [9] considered a single machine scheduling problem in which each job to be scheduled belongs to a family and setups are required between jobs belonging to different families. Schedules must be generated regarding the total resource demand constraints. The objective is minimising total tardiness with respect to the given due dates of the jobs. Two variants of the problem are considered and the results show that the proposed heuristic outperforms a state-of-the-art commercial mixed integer programming solver both in terms of solution quality and computation time. Li & Chen [10] examined consider the problem of scheduling n-jobs in batches on a single parallel-batching machine, where the jobs are partitioned into jobs families and the jobs in each family have the same due date to minimize the weighted number of tardy jobs.

Bilge, Kıraç, Kurtulan & Pekgün [1] presented a Tabu search algorithm for the problem of parallel machines with the objective of minimizing total tardiness of the jobs. Ho & Chang [2] introduced several heuristic methods based on two basic approaches, pay attention to the jobs and the machine in order to minimize tardy jobs in parallel machines, and then evaluate these methods. In the first approach, taking into account the m-machine simultaneously, jobs are assigned one by one to the machines, and in the second approach, the problem of m-machine is broken up into the m single-machine problem to solve one by one. Gupta & Ruiz-Torres [11] considered the problem of parallel machines scheduling with two objective including average flow time and number of tardy jobs. Ruiz-Torres, López & Ho [12] examined the problem of scheduling tasks on a set of parallel machines, so that the speed of the machines depends on the allocation of the secondary source. The secondary source is the steady amount assigned to machines at the beginning of the schedule. The purpose of this scheduling is to minimize the number of delays. Choi & Lee [13] presented a scheduling program for the objective function minimizes the number of delays in a two-stage streaming workshop. Each work must pass two stages, in which there can be several parallel machines at each stage. In order to solve the problem, a B&B algorithm was used in which the upper and lower limits and the overcoming properties were used to limit the problem-solving space. In addition, a heuristic algorithm for large-size problems was also offered. Lin & Jeng [14], considered parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. They provided three dynamic programming algorithms to find optimal solutions for these two objectives. To solve parallel-batch-number of tardy jobs problem, they used three heuristic methods. The first method, first uses a machine-oriented approach, and then uses a dynamic programming method to allocate jobs on each machine. The second method is similar to the first one, except that it uses a work-oriented approach to send jobs to machines and finally, to reduce the time for all machines, the work-oriented approach has been modified so that the problem of sending and grouping is solved simultaneously in the third method. Chaudhry & Elbadawi [15] considered the problem of total tardiness minimizing for scheduling of n jobs on a set of m parallel machines. To solve the problem, they used a spread-sheet-based genetic algorithm (GA) approach. Also they measured the performance of GA in the compare with branch and bound and particle swarm optimisation approaches to illustrate the efficiency of proposed approach. They also demonstrated that the proposed approach can also be used to optimize all objective function without changing the basic GA routine. Lin & Yi [16] have studied the problem of scheduling multiple jobs on parallel machines with objective of minimizing the total tardiness time. To do that, a new computation model for ant colony optimization (ACO) is developed and a novel ACO algorithm is proposed, in which extra space that stores each job's processing machine is used to reduce the number of vertices, and selection space is compressed to improve the efficiency. To evaluate the validity of the algorithm, Ho and Chang's benchmark is applied and the MDD rule is proved to be the most suitable heuristic rule for the algorithm. The stated that their proposed algorithm is effective in resolving

P//T and can be applied in various practical fields, such as production scheduling, workflow controlling and logistics management. Su, Pinedo & Wan [17] proposed a two-phase heuristic for parallel machine scheduling problem, which is basically a generalization of the ATC rule. The objective is minimizing the total weighted tardiness when there are machine eligibility constraints. For this problem they constructed a new composite dispatching rule, namely apparent tardiness cost with flexibility considerations (ATCF) rule, and conducted experiments to empirically evaluate its performance. Furthermore, they have shown the performance of the dispatching rule can be improved significantly using some simple properties without requiring much additional computation time. They performed tests on the performance of the ATCF rule using a real data set from a large hospital in china. They further compare its performance with that of the classical ATC rule and eventually compare the schedules improved by the ATCF rule with near optimal schedules generated by a general search procedure. The computational results show that especially with a low due date tightness, the ATCF rule performs significantly better than the well-known ATC rule generating much improved schedules that are close to the near optimal schedules.

Rabiee, Jolai, Asefi, Fattahi, & Lim [18] addressed a no-wait hybrid flow shop scheduling problem (NWHFSSP) under assumptions such as unrelated parallel machines at each stage, machine eligibility, sequence-dependent set-up times and different ready times. The objective is minimizing the mean tardiness. The largest position value rule is proposed to transmute continuous vectors of each solution into job permutations and a novel biogeography-based optimisation (BBO) algorithm is developed to solve the problem. Response surface methodology (RSM) is employed to evaluate the impact of various parameters on the performance of the proposed BBO algorithm. To validate the proposed approach, different production scenarios for various scales of the problems are created and tested. Results indicate that the proposed BBO outperforms all of the tested algorithms and produces the best solutions in a reasonable time. Diana, de Souza & Moacir Filho [19] investigated the total weighted tardiness minimization problem on unrelated parallel machines with sequence dependent setup times and job ready times. The problem consists in scheduling a set of jobs reducing the penalty costs caused by the delays in the job due dates. an ILS-VND hybrid metaheuristic is proposed to solve the problem and compared with two state-of-art metaheuristics proposed in the literature. The results indicates that for the most scenarios the proposed method outperforms the references metaheuristics.

Schaller & Valente [20] considered the identical parallel machines problem and the objective is minimizing total weighted squared tardiness. Two efficient approaches that can generate solutions for the problem very quickly were proposed. Also an improvement procedure presented so that this improvement procedure can be used with either of the heuristic procedures to create two additional procedures. These heuristics and other heuristics are tested using problem sets that represent a variety of conditions. The results show that the QBP and QBP_I generated the best solutions and the QBP_I procedure is recommended for large scale problems. It is also shown how QBP_I and the improvement procedure heuristics can be incorporated into other procedures such as the existing lagrangian relaxation procedure or meta-heuristics to obtain improved solutions for medium sized problems. Yu, Wen & Yi [21] studied an agent-based scheduling problem of two identical parallel machines. The machines and tasks are regarded as agents and the objectives are minimizing the total tardiness time and minimizing the make span. The proposed procedure can be applied to dynamic and static environments flexibly and is suitable for a discrete and small-sized production problem.

Park & Kim [22] compared two methods, tabu search and simulated annealing, for the problem of scheduling identical parallel machines with the assumption of equal due dates and ready times. The objective of the scheduling problem is to minimize the holding costs of orders including work-in-process as well as finished job inventories. Hamzadayi & Yildiz [23] addressed the problem of static m identical parallel machines scheduling considering a common server and sequence dependent setup times. The used mixed integer linear programming (MILP) for formulating the problem. The objective function is to minimize the make span. To solve the problem, simulated annealing (SA) and genetic algorithm (GA) based solution approaches are developed. Also, the performance of the proposed MILP model, SA and GA based solution approaches are compared with the performance of basic dispatching rules such as, shortest processing time first (SPT) and longest processing time first (LPT) over a set of randomly generated problem instances. The results indicated that the proposed GA is generally more effective and efficient in compared to the proposed MILP model, SA, SPT and LPT. Hübscher & Glover [24] used tabu search approach and candidate list strategy and introduced an influential diversification strategy in parallel machine problems with the goal of minimizing make span. Wang, Liu, Zhang & Zheng [25] considered the problem of parallel machine scheduling considering job splitting and learning. The Objective is minimizing total completion time. Their study is motivated by real situations in labour-intensive industry and formulated as a nonlinear integer programming problem. They proposed a B&B algorithm which is efficient at solving small sized problems. For the large-sized problems, several constructive heuristics and meta-heuristics are presented. Among them, the greedy search, which can take both the current profit and future cost after splitting a job into consideration, obtains a near optimal solution for the small sized problems and performs best in all proposed heuristics for the large sized problems. Yin, Wang, Cheng, Liu & Li [26] considered parallel-machine scheduling of deteriorating jobs with potential machine disruptions. The considered two approaches, performing maintenance immediately on the disrupted machine when a disruption occurs and not performing machine maintenance. In each approach, the objective is to determine the optimal schedule to minimize the expected total completion time of the jobs in both non-resumable and resumable cases.

Chen [27] considered the problem of scheduling a single machine with regard to periodic repairs in which the objective function of the number of tardy jobs is minimized. Based on the Moore's algorithm, he developed a heuristic methodology that offers near-optimal answers. Lee & Kim [28] focused on the problem of scheduling tasks on a machine, taking into account periodic repairs and the objective function to minimize the number of late tasks. They presented a two-phase algorithm that at first phase, presents an initial answer based on Moore's algorithm, regardless of periodic repairs. Then, in the second phase, the initial response is improved.

Laguna, Barnes & Glover [29] studied the problem of a single machine scheduling with the objective of minimizing the sum of the set-up costs and linear delay penalties, and introduced a tabu search algorithm using hybrid neighbourhood. Wan, Yuan & Wei [30] studied minimization of the total number of tardy jobs and maximum cost per single machine and used the Pareto optimization method to solve it. Rasti-Barzoki & Hejazi [31] studied the Integrated problem of minimizing the weighted number of tardy jobs and a common due date is assigned to all the jobs of each customer in the supply chain. In such problems, the due date is defined as a variable in the supply chain. In fact, the objective function is to minimize the sum of the total weighted number of tardy jobs, the total due date assignment costs and the total batch delivery costs. They formulated the problem as an integer Programming model. Also, a heuristic algorithm and a B&B method for solving this problem are presented. Zhang & Chiong [32] have investigated the job shop scheduling problem with the total weighted tardiness criterion as well as the total energy consumption criterion. To solve the problem they used GA metaheuristic. Also two local improvement strategies have been proposed with the aim of reducing the total weighted tardiness and total energy consumption respectively. Mazdeh, Zaerpour, Zareei & Hajinezhad [33] studied the parallel machines scheduling problem with the effects of machine and job deterioration. The objectives of the problem are minimizing total tardiness and machine deteriorating cost. In their study deterioration is defined regarding to cost which depends on the production rate, the machine’s operating characteristics and the kind of work done by each machine. They proposed the LP-metric method to show the importance of multi-objective problem and used tabu search metaheuristic to get solution. The results show the efficiency of proposed model. Saidi-Mehrabad & Bairamzadeh [34] have studied the parallel machine scheduling problem considering machine and job deterioration in a batched delivery system. They formulated their problem as a mixed-integer programming (MILP) model, so that the objective functions are minimizing total tardiness, delivery, holding and machine deteriorating costs. An efficient hybrid genetic algorithm (HGA) is proposed to solve the problem, however, a new crossover and mutation operator and a heuristic algorithm have also been proposed depending on the type of problem. The results illustrate the effectiveness of the proposed model and GA for test problems. Wang & Ye [35] addressed the problem of unrelated parallel machine scheduling with the objective of minimizing the total cost including the machine opening cost and the cost related to maximum earliness and tardiness. The scheduling progress takes both machine utilization and customer satisfaction into consideration. The problem formulated as mixed-integer linear programming. Small-scale problems solved by CPLEX12.6, however, benders decomposition-based heuristic algorithm used to solve largescale problems. Ding, Song, Zhang, Chiong & Wu [36] have studied unrelated parallel machine scheduling problem considering time-of-use (TOU) pricing scheme. The objective is minimizing cost of total electricity by scheduling the jobs subject to total time of completion does not exceed a predetermined production deadline. Two solutions steps applied to solve the problem. In the first step, they formulated the problem using time-interval-based mixed integer linear programming and in the second step, they reformulated the problem using dantzig–wolfe decomposition. Finally they proposed a column generation heuristic to solve the model formulated in step2. Li, Yang, Zhang & Liu [37] considered the unrelated parallel machine scheduling problem considering energy and tardiness cost. The objective is minimizing the total tardiness and energy consumption where the energy consumption on each machine is also unrelated parallel. Because of the complexity of that problem, they proposed ten heuristic algorithms. In order to test the performance of these ten algorithms, computational experiments are designed and the computational results indicate that the algorithms based on the combinational rules outperform the ones based on the priority rules and energy consumption.

Cheng & Huang [38] investigated the problem of unrelated parallel machine scheduling for jobs with distinct due dates and dedicated machines. The objective function is allocating jobs to unrelated parallel machines, so that minimize the total earliness and tardiness time. They used mixed integer linear programming for formulating the problem and a modified genetic algorithm (GA) with a distributed release time control (GARTC) mechanism to obtain the near-optimal solution. The results indicated that the proposed approach can solve the problem with a reasonable time and provide good quality solutions. Wan & Yen [39] studied the scheduling of a single machine to minimize the total weight early, as well as minimize the number of late tasks. They provided many dominance properties for this problem. Result of their work was a heuristic method, as well as a B&B algorithm. Yazdani, Aleti, Khalili & Jolai [40] studied the problem of minimizing the sum of maximum earliness and tardiness of the job shop scheduling problem. A mixed integer linear programming (MIP) formulation of the job shop scheduling problem with the new objective function is developed. The model was tested for different problem sizes, with results showing that the MIP solvers cannot find solutions to instances larger than 7 jobs and 7 machines. So, for problems with more than 7 jobs and 7 machines, they developed a new hybrid imperialist competitive algorithm (HICA), which uses a neighbourhood search to intensify the exploitation of high quality solutions. The experimental evaluation showed that the HICA outperforms the state of the art in the jobs shop scheduling problem. They have also analysed the behaviour of the proposed method on different number of jobs and machines, with the results showing that HICA keeps its robust performance in the different levels of the problem sizes.

According to the literature review, we did not observe a specific research that used simulated annealing for solving the parallel machine tardiness problem subject to job splitting. Hence, in this article, the problem of scheduling parallel machines with the objective of minimizing the number of tardy jobs with job splitting property has been discussed. Initially, a mathematical model for this problem is presented. In order to solve large-scale problems at logical time, simulated annealing algorithm is presented. The quality of the proposed algorithm has been investigated in terms of the quality of the solution and in terms of solving time. Also In order to evaluate the parameters of proposed algorithm, Annona test has been used.

The rest of the paper is organized as follows: in section2, the problem is described, the mathematical model and simulated annealing algorithm are presented and briefly explain the Annona test. In section 3, the solution approach is presented and finally, in section 4, we discuss about the results and provide directions for future research.

2 PROBLEM DEFINITION

The problem is the scheduling of n-independent works on m-parallel machines with the goal of minimizing the number of tardy jobs with the job splitting property. In this case, each work has a specific due date. Every work can be divided into a number of sub-jobs and they can be processed independently on one or more parallel machines. Each sub-job consists of a number of units of job, of which, for every job, at least, the number of machines is considered. A job is completed when all of its sub-jobs are completed.

Based on three-part marking, this problem is indicated by $P|S_{s}, s \quad |\sum N_T$, so that $\boldsymbol{P}$ represents a system with parallel machines, $\boldsymbol{S_{s}}$ represents the setup time that is independent of the sequence of jobs, $\boldsymbol{s\ li}$ denotes the job splitting property, and $\sum \boldsymbol{N_T}$ is the objective function of minimizing the number of tardy jobs. For the description of the scheduling problem, it is necessary to define the assumptions of the problem precisely:

Each sub-job has a setup time, this setup time varies from job to job, but is the same for all subt-jobs of a job. This setup time is also independent of the sequence of jobs. Other assumptions are as follows:

1. All jobs are available at zero time
2. The processing time of a job unit is the same on all machines because the machines are identical.
3. Each machine can process only one sub-job at any time
4. Any sub-job can only be processed on a machine

2.1 Mathematical model

The parameters and decision variables in the proposed formulation are listed as follows:

Parameters:

m	number of parallel machines	
n	number of jobs	
i	machines index	$i = 1,2, \dots, m$
j	jobs index	$j = 1,2, \dots, n$
k	status index for sub-job of job j on machine i	$k = 1,2, \dots, n$
$\boldsymbol{u_j}$	Number of units of job	$u_j = u \quad \forall j$
$\boldsymbol{p_j}$	process time of job	$j = 1,2, \dots, n, j$

s_j	setup time independent of the sequence to work	$j = 1,2,\dots,n,j$
d_j	due date of job j	$j = 1,2,\dots,n,j$
C_j	completion time of job j	$j = 1,2,\dots,n,j$

M A large fixed number that is at least as large as the sum of the process times and setup times

Decision Variables:

$$x_i = \begin{cases} 1 & i\ the\ s - j\ j\ is\ p \quad o\ m\ hi\ i\ i\ the\ k\ h\ s \\ 0 & o\ he \end{cases}$$

y_i =Number unit of jth job on ith machines for kth status

$$Z_j = \begin{cases} 1 & I\ w\ \ j\ is\ d \\ 0 & o\ he \end{cases}$$

The mathematical model of the problem is as follows:

$$M \quad \sum_{j=1}^{n} Z_j \tag{1}$$

s.t.

$$\sum_{j=1}^{n} x_i \le 1 \qquad k = 1,\dots,n; \qquad i = 1,\dots,m. \tag{2}$$

$$\sum_{k=1}^{n} x_i \le 1 \qquad j = 1,\dots,n; \qquad i = 1,\dots,m. \tag{3}$$

$$x_i \le y_i \le u_j x_i \qquad i = 1,\dots,m; \qquad j = 1,\dots,n; \qquad k = 1,\dots,n. \tag{4}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{n} y_i = u_j \qquad j = 1,\dots,n. \tag{5}$$

$$\left(\sum_{s=1}^{k}\sum_{l=1}^{n}(p_l y_i + s_l x_i) - d_j - M(1 - x_i)\right)/ M \le Z_j \qquad i = 1,\dots,m; \qquad j = 1,\dots,n; \qquad k = 1,\dots,n. \tag{6}$$

$$Z_j \in \{0,1\} \qquad j = 1,\dots,n; \tag{7}$$

$x_i \quad \in \{0,1\}$ $\qquad i = 1, \dots, m; \qquad j = 1, \dots, n; \qquad k = 1, \dots, n.$ ***(8)***

$y_i \quad \geq 0$, integer $\qquad i = 1, \dots, m; \qquad j = 1, \dots, n; \qquad k = 1, \dots, n.$ ***(9)***

The lateness of a job is defined as $\boldsymbol{T_j} = \mathbf{m} \quad \{\mathbf{0}, \boldsymbol{C_j} - \boldsymbol{d_j}\}$(Ho & Chang, 1995), and for a tardy job $\boldsymbol{T_j} > 0$. The objective function is to minimize the number of tardy jobs. A job is considered as tardy job when its make span is not less than the due date. Therefore, the objective function of this model will be as Equation ***(1)***. Equation ***(2)*** Ensures that at any time, each machine can be processed at most one job. ***(3)*** States that a job on a car at most will be placed in one position. ***(4)*** establishes a relationship between decision variables, so that, if a sub-job of any job in any situation from any of the machines is placed, it ensures that the number of units of job is at least one and at most equal to the total number of units of job. ***(5)*** States that the total number of units of a job on different machines and in different positions is equal to the number of units of the job. ***(6)*** Determines whether the work j is tardy job or not. In this equation, if $\boldsymbol{x_{i}}$ **=1**, that is, the sub-job **j** is on the machine **i** and in the position **k**, then the coefficient **M** in the left fraction of the equation will be zero. Now, if this job is a tardy job, the numerator will be positive, and due to the presence of M in the denominator, the left-hand side of the equation becomes a very small number between 0 and 1. So $\boldsymbol{Z_j}$on the right hand side, has to take the value of one, since it is a binary variable. But if the job is not tardy job, that is, the numerator is negative, then the left side of the equation will be smaller than zero and $\boldsymbol{Z_j}$ can be zero or one, but since the objective function is minimizing, it should be zero. In the case $\boldsymbol{x_{i}}$ **=0**, that is, the sub-job **j** is not on the machine **i** and in the position **k**, Due to the presence of **M** in the numerator and denominator of the fraction, the left side gets a negative value, which again, since the objective function is minimizing, $\boldsymbol{Z_j}$ will surely take zero value. Constraints ***(7)***, ***(8)*** and ***(9)*** refers to the values of the decision variables.

2.2 Simulated annealing algorithm

SA is an appropriate metaheuristic algorithm to solve combined optimization problems. This algorithm, simulating the gradual annealing process to solve an optimization problem. This algorithm has the ability to exit local optimization with the possible acceptance of worse solutions. Neighbourhood generation, acceptance function, initial temperature, equilibrium state, annealing function and termination condition are the main components of this method [41]. The process of this algorithm is shown in Fig. 1

The following is an algorithm developed for our desired problem, which is inspired only by the original algorithm. The general framework of this algorithm is that in the first step the initialization, the initial and final temperature, the generation of the initial solution, and the value of the objective function are performed. The method for generating the initial solution r will be explained in next section. In the second step there is a loop that produces a certain number of neighbours from the initial solution.

In each iteration, if the neighbour's solution is better than the initial solution, it is accepted, and if worse, it is accepted with a certain probability determined by the acceptance function. After the completion of this loop, the average of the objective function of the solutions accepted in this loop is compared to the average of the total solutions given to this temperature. If the equilibrium criterion is satisfied, the temperature decreases; otherwise, the loop of the epochs will be repeated until equilibrium is achieved. This process continues until it reaches the final temperature.

In the following sections, we describe each of the steps in this algorithm.

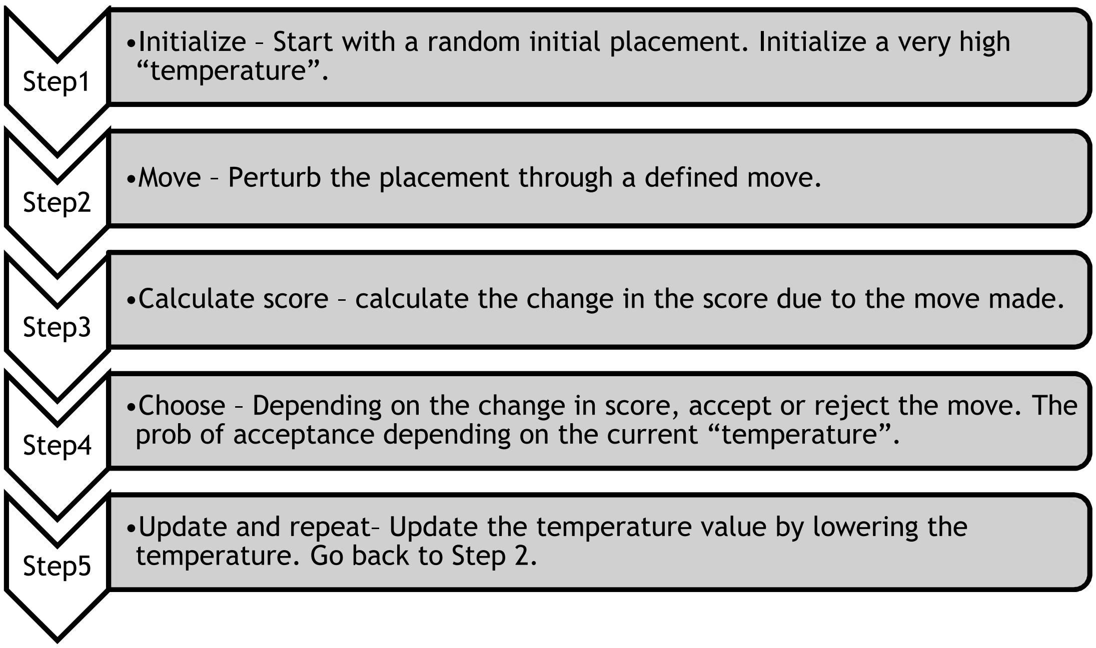

Fig1: Process of SA algorithm

2.2.1 Initial solution generation

An innovative algorithm was used to generate the initial solution. First, jobs are arranged according to the earliest due date and allocated to machines, respectively, until the last machine and again in the presence of job are assigned to the machines from the beginning. This process continues until the end of the jobs. With this method, we can get a good solution to start the algorithm. The reason for choosing this arrangement is that we know that in the objective function associated with the due date, sequences that are similar to the earliest due dates, are relatively appropriate.

2.2.2 Neighbourhood generation

We used two methods of neighbourhood generation in order to create a neighbourhood of one solution and reach other points of space. In the neighbourhood of the first type, the contents of the two positions are displaced on the machines. These two places are selected by the candidate list. This can occur between two machines or on a machine. Figure 2 shows how to create this neighbourhood schematically. In order to consider the job splitting property into several sections, the second method of neighbourhood generation has been developed. In this method, according to the candidate list, a job is selected and randomly a number of its sub-jobs puts an empty position in another machine. This method of neighbourhood generation in a schematic way is shown in Fig. 3.

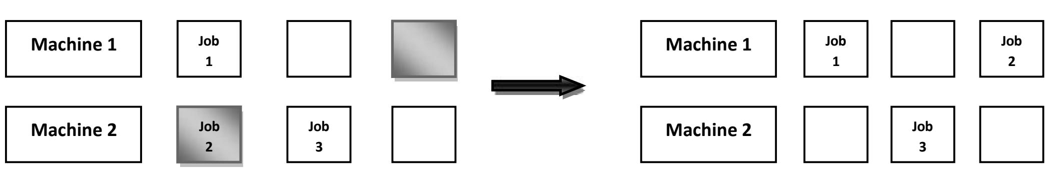

Fig 2: Create a neighbourhood type one

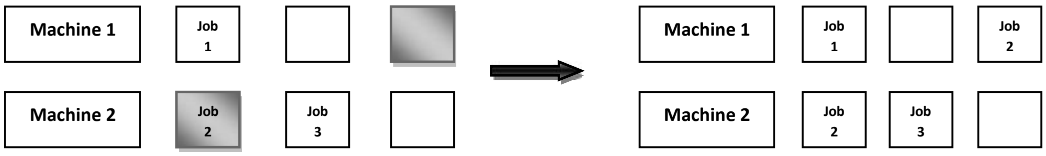

Fig 3: Create a neighbourhood type two

The mode of operation of these two neighbourhoods in the algorithm process is that in the process of the algorithm, each of them is randomly used from one of them. As stated in the first type neighbourhood, only job stations are replaced, and in the second type, a job is broken up randomly into two sub-jobs.

2.2.3 Candidate list

Since the SA algorithm does not examine all the neighbourhoods of an answer, and only a certain number of them are searched for the next move, a mechanism for determining how these neighbourhoods are to be selected should be designed. Accordingly, in the algorithm presented, two candidate lists with a specified length, which are the parameters of the algorithm, are considered.

In the first list, which is used in the first type neighbourhood, in each row from this list, randomly, displace is selected in the current solution. For example, displacing a second position from the first machine with the first position of the first machine, provided that at least one of these situations has a job. For example, if the length of the list is 10, the 10 displacements in the solution are randomly selected and placed in the list to be checked by the first type neighbourhood method and select the best displacement. In the list that is created for the second type neighbourhood, there are two positions in each row, which in the first position is job that needs to be splitter, and in the second position, the place where the splitter piece of job should be placed. For instance, in the first row of this list can be placed that the first place is splitter from the first machine and placed in the third place of the second machine.

For example, if the length of this list is ten, that means, the number of job splitting in the list is ten, so that it can be examined by the second method of neighbouring production, and the best one is chosen.

2.2.4 Acceptance function

In this algorithm, in order to escape the local optimality, bad solutions are accepted with a certain probability calculated by the following function:

$$P = \exp\left(\frac{F_a - F_n}{0.1 * T_i}\right) \qquad (10)$$

F_a is the value of previously accepted objective function and F_n is the value of new objective function. Also, T_i is the temperature of this repetition of the algorithm. Since the objective function in this problem is integer and there is no significant difference in the generation of the solution every time, a modification coefficient with a value of 0.1 in the denominator has been used to modify the acceptance function.

2.2.5 Initial and final temperature

In the generated algorithm, the initial temperature is equal to the number of jobs in the problem. The final temperature of the algorithm was also determined at 10% of the initial temperature.

Glover & Kouchenberger [41] have presented different temperatures for starting and ending, which we used to develop the algorithm at the above mentioned temperature.

2.2.6 Equilibrium Criterion

In each temperature, if the generated solutions are in equilibrium, temperature allowed to be reduced. Also in each temperature, a certain number of epochs, which is the algorithm parameters, are generated in the neighbourhood that some of them are accepted. The equilibrium criterion is considered to be the average of the objective function of the accepted solutions at any temperature close to the average of the accepted solutions at the past temperatures. This closeness is calculated by the following equation and the value of the closeness criterion is also the algorithm parameters:

$$\frac{\left|\bar{f}_T - \bar{f}_e\right|}{\bar{f}_T} < \xi \qquad \textbf{(11)}$$

In which $\bar{f}_T$ is the average of the objective function at the past temperatures, $\bar{f}_e$ is the average of the objective function at this temperature and ξ is the parameter of the equilibrium criterion.

2.2.7 Annealing function

In this algorithm, the geometric annealing function which is the most common annealing function, has been used:

$$T_{i+1} = \alpha \cdot T_i \qquad \textbf{(12)}$$

In which T_i is the temperature of this stage, T_{i+1} is the next step temperature and α is the temperature reduction rate that is the algorithm parameters. Experience has shown that this coefficient should be between [0.5.0.99] [41].

2.3 Analysis of variance

Analysis of variance is closely related to the design of experiments. Obviously, whenever different parts of the change are linked to certain effects, the experiments must be designed to allow such an action to be provided in a rational and precise manner. In this study, multivariate analysis of variance has been used. Multivariate analysis of variance is a type of variance analysis that is used when it has more than one dependent variable. These dependent variables must have some kind of relationship, or there must be a conceptual reason for their coexistence. In fact, it compares the multivariate analysis of the variance of the groups and tells you whether the mean difference between the groups was in the combination of the dependent variables due to the chance or not. For this purpose, the multivariate analysis of variance, creates a new summary dependent variable. It is the linear combination of each of the dependent variables. Then, analysis of variance using this compound dependent variable [42].

3 SOLUTION APPROACH

It is shown that the problem of a m-machine with N_T criterion is NP-complete, even when m is 2 [43]. In order to clarify the complexity of the problem, by using GAMS23.7 software, CPLEX solver, we solved a number of examples of this problem using a precise algorithm for solving complex integer programming problems, and the time and the solutions are given in Table 1.

Table 1 shows that with the increase in the dimensions of the problem, the solution time will increase strongly. For example, if we have 4 machines, by adding only two jobs, the time to reach the optimal solution increases by more than 10 times. Or, doubling the number of machines and jobss, the time tosolve more than 800 times increases. in the mode of 6 machines and 12 jobs, the algorithm was given 2,000 seconds, but did not achieve the optimal solution and obtained the value of the objective function 2. The point here is that this answer is not necessarily optimal, and is based on time constraints. In fact, the optimal answer can be somewhat better than this.

3.1 Design of experiments

Design of experiments was used to evaluate the parameters of the SA algorithm. At first, a number of problems were generated randomly. Computational experiments on problems of combinations with three levels of machines (4, 8 and 12), four levels of the number of jobs (10, 15, 20 and 25) and four levels of the number of units of work (4 and 6, 8 and 10) were performed. the problems of experiment were ranked in two categories, small and medium, in terms of the number of machines, jobs, and job units, each having three problems of the following dimensions ([number of units - number of jobs - number of machines):

Table 1: Results of model solving by GAMS software

Number of machines	Number of jobs	Objective function value	Running time(second)
2	4	2	0.234
3	6	2	1.313
4	6	2	11.169
4	8	2	193.877
5	10	3	1000.305
6	12	2	2000

1. Small problems category: [4-10-4], [6-15-4], [6-15-8].
2. Medium problems category: [8-20-8], [8-20-12], [10-25-12].

For each problem, the process time for each random number unit is selected in the interval [5, 60]. The preparation time is also generated randomly by an integer from the interval [5, 60]. Finally, the due date t is also randomly selected from the interval$[\alpha \sum(s_j + u_j p_j)/m, \beta \sum(s_j + u_j p_j)]$. s_j is setup time, u_{jj} Number of job units, p_j Process time, m number of machines and (α, β) Determine the severity of the due date, which is equal to (1,2).

In the small category of cases, for the parameters of the number of replicates in each temperature, three levels (5, 10, 20), the annealing rate of three levels (0.5, 0.75, 0.9) and the two-level equilibrium (0.05, 0.1) were considered. However, for the middle class issues, the three levels (20, 30, 50) for the number of repetitions (0.7, 0.8, 0.9) for annealing and two levels (0.05, 0.1) were considered for the equilibrium criterion.

A total of 540 tests were performed and multivariate analysis of variance was performed on SPSS software. The results of these tests are presented in table2 for a small category of issues. ***a*** Indicates the parameter of the number of repetitions at each temperature, ***b*** the annealing rate, ***c*** the equilibrium criterion and ***d*** block of the problem. As shown in table 2, changes in the parameters of the number of repetitions and the annealing rate affect the result of the algorithm. By examining the results, it was found that the number of repetitions 20 and the annealing rate of 0.9 would result in better results.

Table 2: The analysis of variance results for Small problems category

Source	Type III sum of squares	df	Mean square	F	Sig.
Corrected model	244.081a	19	12.846	45.994	0
Intercept	3549.423	1	3549.423	12708.152	0
a	6.157	2	3.079	11.022	0
b	3.575	2	1.787	6.4	0.004
c	0.167	1	0.167	0.597	0.445
d	230.05	2	115.025	411.83	0
a * b	1.745	4	0.436	1.562	0.207
a * c	0.871	2	0.436	1.559	0.225
b * c	0.618	2	0.309	1.106	0.343
a * b * c	0.898	4	0.224	0.804	0.531
Error	9.496	34	0.279		
Total	3803	54			
Corrected total	253.577	53			

a. R Squared = 0.963 (Adjusted R Squared = 0.942)

In table 2, the first column are the source of the effective design of the experiments (indicates the parameter of the number of repetitions at each temperature, b the annealing rate, c the equilibrium criterion and the d block of the problem). The second column is the sum of the squares of the mathematical relations used in the analysis of variance, and the third column represents the degree of freedom for resource. The fourth column shows the mean square of the mathematical relationships used in the analysis of variance. The fifth column shows the fisher statistic calculated from the mathematical relationships of analysis of variance, and finally, the last column shows the effect of each variable. Because we considered a significant level of 5%, if this value is lowered by 5%, it means that the source is effective, and as shown in the table, changes in the number of repetitions and annealing parameters affect the results of the algorithm.

These experiments were carried out for a medium-sized problem group whose results of the analysis of variance are presented in Table 3. The parameters ***a*** and ***d*** are significant at 5% level and parameter ***b*** is significant at 10% level. These results show that the number of repetitions in each temperature and annealing rate affects the results of the algorithm, but the equilibrium criterion has no effect. By examining the results, it was found that the number of repetitions 50 and the annealing rate 0.8 had better results.

Table 3: The analysis of variance results for medium problems category

Source	Type III sum of squares	df	Mean square	F	Sig.
Corrected model	245.755a	25	9.83	30.77	0
Intercept	7224.54	1	7224.54	22614.079	0
a	5.951	2	2.976	9.314	0.001
b	1.72	2	0.86	2.692	0.085
c	0.214	1	0.214	0.67	0.42
d	233.453	2	116.727	365.375	0
a * b	1.502	4	0.376	1.176	0.343
a * c	0.401	2	0.201	0.628	0.541
b * c	0.268	2	0.134	0.42	0.661
b * c * d	2.244	10	0.224	0.703	0.714
Error	8.945	28	0.319		
Total	7479.24	54			
Corrected total	254.7	53			

a.R Squared = 0.965 (Adjusted R Squared = 0.934)

In Table 3, the first column are the source of the effective design of the experiments. The second column is the sum of the squares of the mathematical relations used in the analysis of variance, and the third column represents the degree of freedom for resource. The fourth column shows the mean square of the mathematical relationships used in the analysis of variance. The fifth column shows the Fisher statistic calculated from the mathematical relationships of analysis of variance, and finally, the last column shows the effect of each variable. Because we considered a significant 5% level, if this value is lowered by 5%, it means that the source is effective. Therefore, it can be said that parameters ***a*** and ***d*** are significant at 5% level and the parameter ***b*** is significant at 10%. The results of multivariate analysis of 540 experiments have been shown in Table 4.

Table 4: Results of multivariate analysis of variance for SA algorithm parameters

Parameters	Small problems		Medium problems	
	Fisher statistic	Significance level	Fisher statistic	Significance level
Number of replication in each temperature	11.022	0	9.314	0.001
Annealing rate	6.4	0.004	2.692	0.085
Equilibrium criterion	0.597	0.455	0.67	0.42
(Number of replication in each temperature).(Annealing rate)	1.562	0.207	1.176	0.343
(Number of replication in each temperature).(Equilibrium Criterion)	1.559	0.225	0.928	0.541
(Equilibrium criterion).(Annealing rate)	1.106	0.343	0.42	0.661
(Number of replication in each temperature).(Annealing rate).(Equilibrium criterion)	0.804	0.531	0.703	0.714

As shown in table 4, for both categories, the change in the parameters of the number of repetitions and the annealing rate affect the result of the algorithm. Because for a significant level of 5%, in both categories, the value of the column has a significant level of less than 5% or a slight difference. By examining the results, it was found that for small problems category, the number of repetitions 20 and the annealing rate 0.9, and for the medium problems category, the number of repetitions 50 and the annealing rate 0.9, are better. Because the number of tardy jobs in them is smaller than other problems. Because in total 5 repetitions, have the minimum value of the objective function compared with other dimensions of the problem in other parameters. The above problem category is highlighted in the table. As a result, for small problems category, the values of the algorithm parameters are equal to (0.05, 0.9, 20) and for medium problems category are equal to (0.05, 0.9, 50). For the medium problems category, the above problem is highlighted in the table. Because in total 5 repetitions, have the minimum value of the objective function compared with other dimensions of the problem in other parameters.

3.2 Data analysis

In order to determine the efficiency of the proposed algorithm for very small problems, the solution obtained from this algorithm is compared with the optimal answer. To solve the problem, we have used the GAMS software, CPLEX solver. Table 5 presents the results of SA algorithm and an optimal answer for ten small-dimensional problems.

Table 5: Values of time and objective function for very small problems

SA				GAMS		Number of replication	Number of Units	Number of machines	Number of Jobs
Objective function		Time		Objective function	Time				
Standard deviation	Mean	Standard deviation	Mean						
0	0	0	0.02	0	1.61	5	5	2	2
0	0	0	0.02	0	0.12	5	5	2	3
0	0	0	0.02	0	0.23	5	5	3	3
0	2	0	0	2	0.89	5	5	2	4
0.49	1.4	0.01	0.01	1	0.71	5	5	3	4
0	1	0.01	0.02	1	0.62	5	5	2	5
0	3	0	0.01	3	47.3	5	5	2	6
0	2	0.01	0.06	3*	3600	5	5	4	8
0.49	3.6	0.02	0.03	3*	3600	5	5	4	10

*Run with one hour's time limit

As shown in Table 5, the algorithm's solutions are very close to optimal solutions. The middle column shows the answers given by the GAMS software and the right column shows the results of the SA algorithm.

To evaluate the performance of the algorithm, computational experiments were performed on a series of problems mentioned in the design of experiments section. The algorithm was designed according to the results obtained from the design of experiments section and was performed for both groups. The results of these experiments are presented for a small group of problems in Table 6 and for medium problems in table 7.

In table 6, the number of jobs, the number of machines, and the number of units of work are divided into three small-size groups, which are expressed in columns one to three. The fourth column shows the number of repetitions for each category.

Table 6: Time and objective function from solving small problems by SA

SA				Number of replication	Number of units	Number of machines	Number of jobs
Objective function		Time					
Standard deviation	Mean	Standard deviation	Mean				
0.49	4.4	0.07	0.05	5	4	4	10
0.4	8.8	0.02	0.06	5	6	4	15
0.4	9.2	0.01	0.06	5	6	8	15

In the four columns to the right, the mean and standard deviation obtained from the objective function and the time of solving the SA are presented. The low value of standard deviation indicates the efficiency of the algorithm, since it delivers near-matched answers each time.

Table 7: Time and objective function from solving medium problems by SA

SA				Number of replication	Number of units	Number of machines	Number of jobs
Objective function		Time					
Standard deviation	Mean	Standard deviation	Mean				
0.75	10.20	0.08	0.25	5	8	8	20
0.49	9.40	0.06	0.35	5	8	12	20
0.98	13.80	0.08	0.36	5	10	12	25

Also in table 7, the number of jobs, the number of machines, and the number of units of work are divided into three small-size groups, which are expressed in columns one to three. The fourth column shows the number of repetitions for each category. In the four columns to the right, the mean and standard deviation obtained from the objective function and the time of solving the SA are presented. The low amount of standard deviation in this category also confirms the efficiency of the algorithm.

In fig. 4 and fig. 5, the mean time and value of the objective function derived from the implementation of the algorithm are presented for small category of problems. In fig. 4, the vertical column represents the problem solving time in seconds, and the horizontal column represents different issues. As you can see, there is not much difference between the various categories of issues.

In fig. 5, the vertical column represents the value of the objective function and the horizontal column represents different issues. As you can see, the value of the objective function increases with increasing dimensions of the problem.

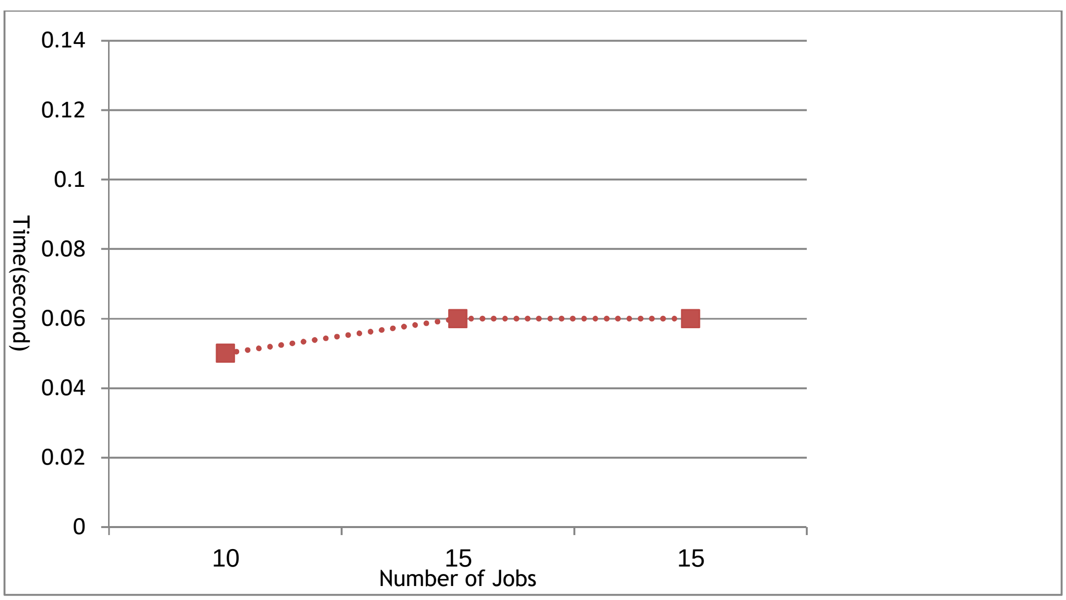

Figure 4: Algorithm solving time for small problems

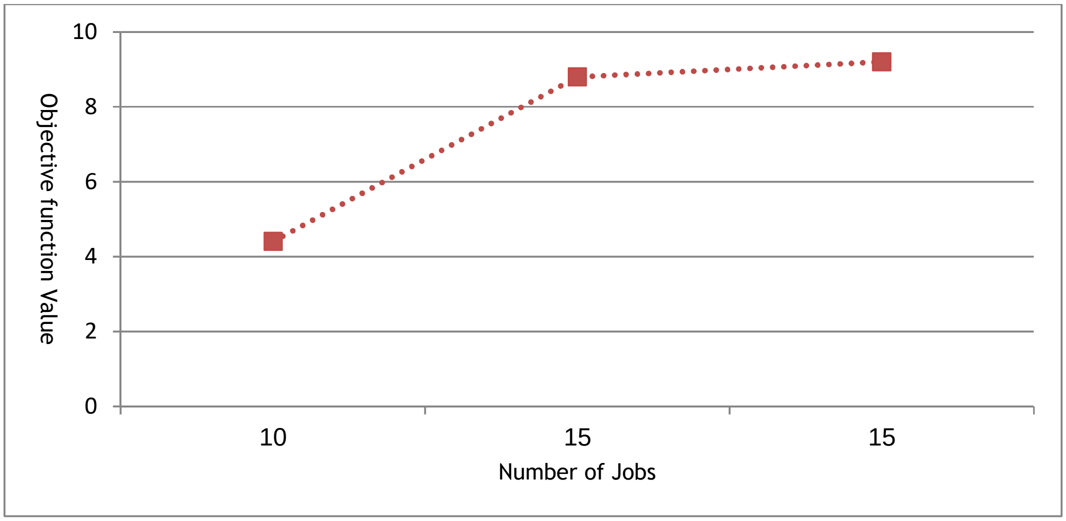

Figure 5: Value of the objective function algorithm for small problems

In fig. 6 and fig. 7, the mean time and value of the objective function derived from the implementation of the algorithm are presented for medium category of problems.

In fig. 6, the vertical column represents the problem solving time in seconds, and the horizontal column represents different issues. As it is shown, increasing the dimensions of the problem increases the solving time.

In fig. 7, the vertical column represents the value of the objective function and the horizontal column represents different issues. As you can see, the value of the objective function increases with increasing dimensions of the problem.

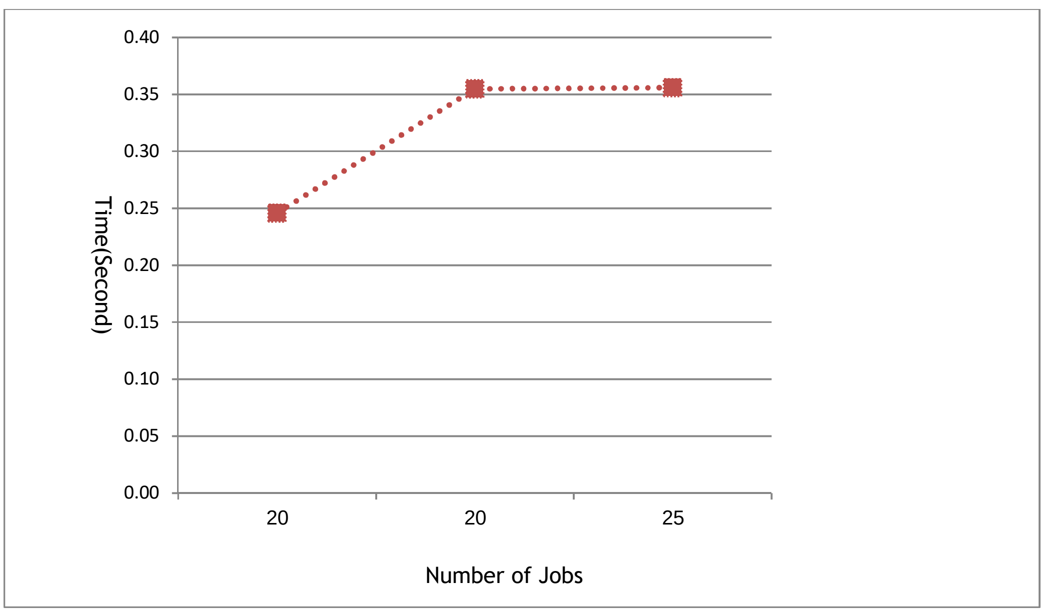

Figure 6: Algorithm solving time for medium problems

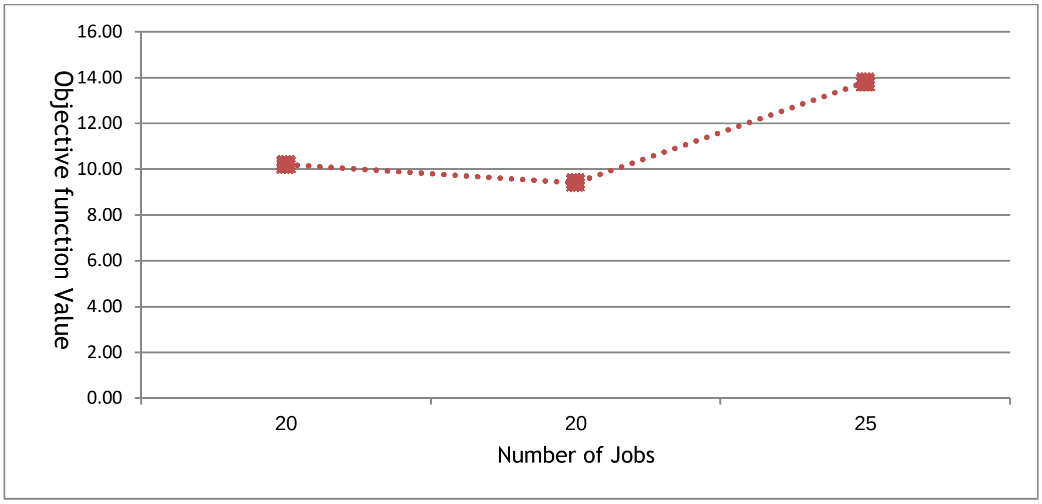

Figure 7: Value of the objective function algorithm for medium problems

Experiments have also been performed to investigate the algorithm's performance on large-scale issues, the results of which are shown in table 8.

Table 8: Time and objective function from solving large-scale problems by SA

SA				Number of replication	Number of units	Number of machines	Number of jobs
Objective function		Time					
Standard deviation	Mean	Standard deviation	Mean				
1.36	16.60	0.26	0.75	5	10	15	30
1.20	26.40	0.22	0.77	5	10	20	40
0.98	36.20	0.18	0.91	5	10	20	50
2.42	67.40	0.03	0.94	5	10	20	80
2.06	88.60	0.02	0.94	5	10	20	100

The results for this category of issues indicate the proper SA performance, especially in terms of problem solving time. Problems in this dimension are by no means capable of solving optimally at logical time. But we observe that the proposed algorithm offers very good quality answers at very low time.

In fig. 8 and fig. 9, the mean time and value of the objective function derived from the implementation of the algorithm are presented for large-scale category of problems.

In fig. 8, the vertical column represents the problem solving time in seconds, and the horizontal column represents different issues. As it is shown, even for large-scale issues, solving time is very little, which is the most important advantage of using this algorithm.

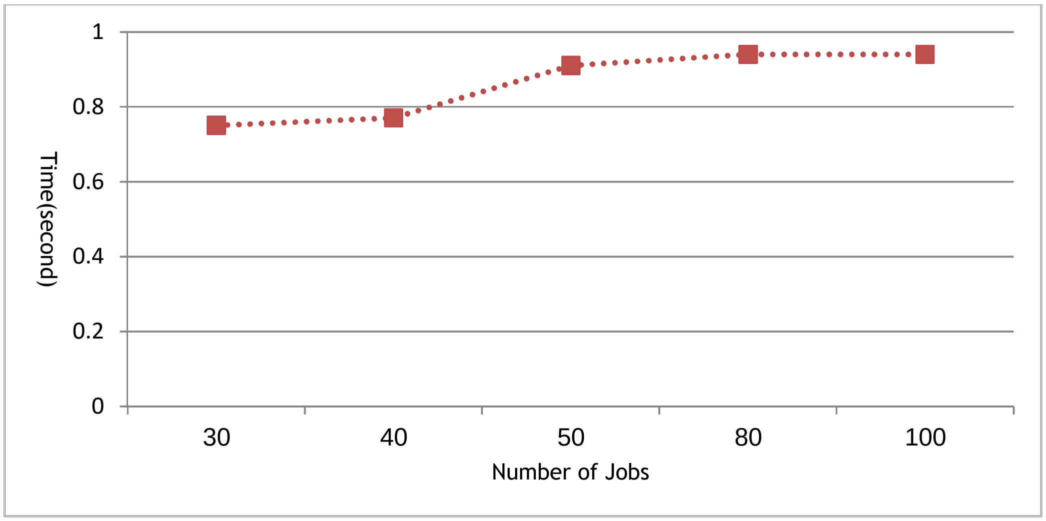

Figure 8: Algorithm solving time for large-scale problems

In fig. 9, the vertical column represents the value of the objective function and the horizontal column represents different issues. As you can see, the value of the objective function increases with increasing dimensions of the problem.

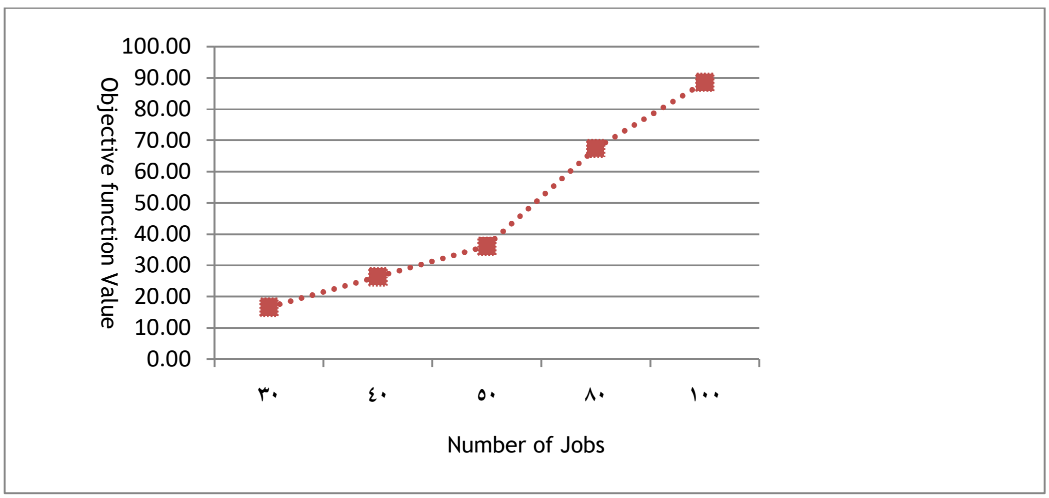

Figure 9: Value of the objective function algorithm for large-scale problems

4 DISCUSSION AND CONCLUSION

In this article, the problem of scheduling parallel machines with the objective of minimizing the number of tardy jobs in terms of job splitting property has been discussed. According to the research carried out in the literature review, so far, no model has been presented for the problem of scheduling parallel machines with the objective of minimizing the number of tardy jobs in terms of job splitting property.

In this study, a mathematical model was first proposed for this problem and the complexity of the problem was considered. Because this problem is in the NP-Complete series of problems, solving it at a time is not possible with polynomials with increasing dimensions of the problem. Therefore, due to the fact that in industrial and operational environments, it is very important to achieve a good solution at a very short time, and the metaheuristic algorithms provide such an opportunity for problem solving, these algorithms were used to solve this problem.

In the following, a metaheuristic SA method was proposed to solve large-scale problems at logical time. This method is very effective for issues of high complexity, and our computational results have shown that for a large-scale problem, a solution with very good quality can be obtained at a very small time.

The proposed SA solutions were compared with the optimal results obtained by GAMS software. Results show the high quality of the proposed method. Also, to investigate the quality of the algorithm's performance in terms of solving time, various issues were addressed in the form of medium and large dimensional problems in order to evaluate the performance of the proposed algorithm. The results show the high efficiency of the proposed SA in terms of solving time.

In this research, our focus was on the identical parallel machines system, but many manufacturing industries have more complex systems, such as a flexible flow workshop or workflow workshop, which can be studied in future researches. Also, in the present study, transportation between parallel machines is not considered, which can be considered as another research topic. On the other hand, given that today's different industries pursue different policies and goals, the use of combined objective functions can also be a fascinating field for future researches. Another approach to future researches may be to further develop other heuristic and metaheuristic methods to solve a problem, as well as adding other assumptions such as setup times dependent to the sequencing, of course can lead to complexity of the problem.

5 REFERENCES

[1] **Bilge, Ü., Kıraç, F., Kurtulan, M. and Pekgün, P.** 2004. A tabu search algorithm for parallel machine total tardiness problem. *Computers and Operations Research*, *31*(3), pp.397-414.

[2] **Ho, J.C., Chang, Y.L.** 1995. Minimizing the number of tardy jobs for m parallel machines. *European Journal of Operational Research*, *84*(2), pp.343-355.

[3] **Lee, C.H.** 2017. A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness. *International Journal of Production Research*, pp.1-17.

[4] **Ta, Q.C., Billaut, J.C., Bouquard, J.L.** 2018. Matheuristic algorithms for minimizing total tardiness in the m-machine flow-shop scheduling problem. *Journal of Intelligent Manufacturing*, *29*(3), pp.617-628.

[5] **Moslehi, G., Jafari, A.** 2010. Minimizing the number of tardy jobs under piecewise-linear deterioration. *Computers and Industrial Engineering*, *59*(4), pp.573-584.

[6] **Briand, C., Ourari, S.** 2013. Minimizing the number of tardy jobs for the single machine scheduling problem: MIP-based lower and upper bounds. *RAIRO-Operations Research*,*47*(1), pp.33-46.

[7] **Liu, M., Wang, S., Chu, C., Chu, F.** 2016. An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research*, *54*(12), pp.3591-3602.

[8] **Elyasi, A., Salmasi, N.** 2013. Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming. *Mathematical and Computer Modelling*, *57*(5-6), pp.1154-1164.

[9] **Herr, O., Goel, A.** 2016. Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints. *European Journal of Operational Research*, *248*(1), pp.123-135.

[10] **Li, S.S., Chen, R.X.** 2014. Single-machine parallel-batching scheduling with family jobs to minimize weighted number of tardy jobs. *Computers and Industrial Engineering*, *73*, pp.5-10.

[11] **Gupta, J.N., Ruiz-Torres, A.J.** 2005. Generating efficient schedules for identical parallel machines involving flow-time and tardy jobs. *European Journal of Operational Research*,*167*(3), pp.679-695.

[12] **Ruiz-Torres, A.J., López, F.J., Ho, J.C**. 2007. Scheduling uniform parallel machines subject to a secondary resource to minimize the number of tardy jobs. *European Journal of Operational Research*, *179*(2), pp.302-315.

[13] **Choi, H.S., Lee, D.H.** 2009. Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops. *Computers and Industrial Engineering*, *56*(1), pp.113-120.

[14] **Lin, B.M.T., Jeng, A.A.K.** 2004. Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. *International Journal of Production Economics*, *91*(2), pp.121-134.

[15] **Chaudhry, I.A., Elbadawi, I.A.** 2017. Minimisation of total tardiness for identical parallel machine scheduling using genetic algorithm. *Sādhanā*, *42*(1), pp.11-21.

[16] **Lin, C., Yi, Y.** 2017, June. A new ant colony optimization for minimizing total tardiness on parallel machines scheduling. In Electronics Instrumentation and Information Systems (EIIS), 2017 First International Conference on (pp. 1-4). IEEE.

[17] **Su, H., Pinedo, M., Wan, G.** 2017. Parallel machine scheduling with eligibility constraints: A composite dispatching rule to minimize total weighted tardiness. *Naval Research Logistics (NRL)*, *64*(3), pp.249-26.

[18] **Rabiee, M., Jolai, F., Asefi, H., Fattahi, P., Lim, S.** 2016. A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness. *International Journal of Computer Integrated Manufacturing*, *29*(9), pp.1007-1024.

[19] **Diana, R.O., de Souza, S.R., Moacir Filho, F.F.** 2018. A Variable Neighborhood Descent as ILS local search to the minimization of the total weighted tardiness on unrelated parallel machines and sequence dependent setup times. *Electronic Notes in Discrete Mathematics*, *66*, pp.191-198.

[20] **Schaller, J., Valente, J.M.** 2018. Efficient heuristics for minimizing weighted sum of squared tardiness on identical parallel machines. *Computers and Industrial Engineering*, *119*, pp.146-156.

[21] **Yu, F., Wen, P., Yi, S.** 2018. A multi-agent scheduling problem for two identical parallel machines to minimize total tardiness time and makespan. *Advances in Mechanical Engineering*, *10*(2), p.1687814018756103.

[22] **Park, M.W., Kim, Y.D.** 1997. Search heuristics for a parallel machine scheduling problem with ready times and due dates. *Computers and industrial engineering*, *33*(3-4), pp.793-796.

[23] **Hamzadayi, A., Yildiz, G.** 2017. Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times. *Computers and Industrial Engineering*, *106*, pp.287-298.

[24] **Hübscher, R., Glover, F.** 1994. Applying tabu search with influential diversification to multiprocessor scheduling.*Computers and operations research*, *21*(8), pp.877-884.

[25] **Wang, C., Liu, C., Zhang, Z.H., Zheng, L.** 2016. Minimizing the total completion time for parallel machine scheduling with job splitting and learning. *Computers and Industrial Engineering*, *97*, pp.170-182.

[26] **Yin, Y., Wang, Y., Cheng, T.C.E., Liu, W., Li, J.** 2017. Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. *Omega*, *69*, pp.17-28.

[27] **Chen, W.J.** 2009. Minimizing number of tardy jobs on a single machine subject to periodic maintenance. *Omega*, *37*(3), pp.591-599.

[28] **Lee, J.Y., Kim, Y.D.** 2012. Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance. *Computers and Operations Research*, *39*(9), pp.2196-2205.

[29] **Laguna, M., Barnes, J.W., Glover, F.W.** 1991. Tabu search methods for a single machine scheduling problem.*Journal of Intelligent Manufacturing*, *2*(2), pp.63-73.

[30] **Wan, L., Yuan, J., Wei, L.** 2016. Pareto optimization scheduling with two competing agents to minimize the number of tardy jobs and the maximum cost. *Applied Mathematics and Computation*, *273*, pp.912-923.

[31] **Rasti-Barzoki, M., Hejazi, S.R.** 2013. Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. *European Journal of Operational Research*,*228*(2), pp.345-357.

[32] **Zhang, R., Chiong, R.** 2016. Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, *112*, pp.3361-3375.

[33] **Mazdeh, M.M., Zaerpour, F., Zareei, A., Hajinezhad, A.** 2010. Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs. *Applied Mathematical Modelling*, *34*(6), pp.1498-1510.

[34] **Saidi-Mehrabad, M., Bairamzadeh, S.** 2018. Design of a hybrid genetic algorithm for parallel machines scheduling to minimize job tardiness and machine deteriorating costs with deteriorating jobs in a batched delivery system. *Journal of Optimization in Industrial Engineering*, *11*(1), pp.35-50.

[35] **Wang, S., Ye, B.** 2017, December. A benders decomposition-based heuristic algorithm framework for unrelated parallel machine scheduling problem with weighted maximum earliness and tardiness. In Industrial Engineering and Engineering Management (IEEM), 2017 IEEE International Conference on (pp. 614-618). IEEE.

[36] **Ding, J.Y., Song, S., Zhang, R., Chiong, R., Wu, C.** 2016. Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Transactions on Automation Science and Engineering*, *13*(2), pp.1138-1154.

[37] **Li, Z., Yang, H., Zhang, S., Liu, G.** 2016. Unrelated parallel machine scheduling problem with energy and tardiness cost. *The International Journal of Advanced Manufacturing Technology*, *84*(1-4), pp.213-226.

[38] **Cheng, C.Y., Huang, L.W.** 2017. Minimizing total earliness and tardiness through unrelated parallel machine scheduling using distributed release time control. *Journal of manufacturing systems*, *42*, pp.1-10.

[39] **Wan, G., Yen, B.P.C.** 2009. Single machine scheduling to minimize total weighted earliness subject to minimal number of tardy jobs. *European Journal of Operational Research*,*195*(1), pp.89-97.

[40] **Yazdani, M., Aleti, A., Khalili, S.M., Jolai, F.** 2017. Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem. *Computers and Industrial Engineering, 107*, pp.12-24.

[41] **Glover, F. W., Kochenberger, G. A.** 2006. *Handbook of metaheuristics* (Vol. 57): Springer Science and Business Media.

[42] **Montgomery, D.C.** 2017. *Design and analysis of experiments*. John wiley and sons.

[43] **Moore, J. M.** 1968 . An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15(1), pp.102-109.