Journal of Industrial and Systems Engineering
Vol. 13, Special issue: 16th International Industrial Engineering Conference
Winter (January) 2020, pp. 112-120

(IIEC 2020)
TEHRAN, IRAN
22-23 January 2020

Single-machine scheduling considering carryover sequence-dependent setup time, and earliness and tardiness penalties of production

Saeed Mozaffariyan[1], Rashed Sahraeian[1*]

[1]*Department of Industrial Engineering, Faculty of Engineering, Shahed University, Tehran, Iran*

saeed.mozafarian@shahed.ac.ir, sahraeian@shahed.ac.ir

Abstract

Production scheduling is one of the very important problems that industry and production are confronted with it. Production scheduling is often planned in the industrial environments while productivity in production can improve significantly the expansion of simultaneous optimization of the scheduling plan. Production scheduling and production are two areas that have attracted much attention in the industry literature and production and research in the operation systems. In this study, the problem of single-machine scheduling with linear earliness and tardiness costs considering the work failure, energy consumption restriction, and the allowed idleness have been investigated and a new nonlinear mathematical model has been presented for the single-machine scheduling problem. Considering complexity in solution, this problem has been regarded as NP-hard problem. However, using methods that produce optimized results, it is just suitable for small size problems. Based on this, a genetic algorithm has been presented for solving this problem in average and large sizes. Numerical samples show that the presented algorithm is effective and efficient.

Keywords: Single-machine scheduling, energy consumption restriction, earliness and tardiness penalties

1-Introduction

Scientific approach to the operation scheduling and sequencing problem is rooted in the industrial revolution and Gantt's artistic efforts. Today, one of the important problems in the operation research science is related to the scheduling problem that deals with the allocation of resources for implementation of a set of different tasks during time.

Scheduling is one of the most useful concepts in the production industrial and planning systems (Pei et al., 2019). In the scheduling studies, when a machine begins to process a task after processing another task, it needs new regulations (da Silva et al., 2019). These regulations that need time are called setup. Sometimes, the necessary time for setting up a machine to process a task is dependent on the previous task and these kinds of setup times are called sequence-dependent setup time (Sun and Geng , 2019). In some problems of the real world, the time to set up a machine for a task depends on all the previous tasks. This means that if the sequence of tasks changes, the setup time of the new task will also change. These kinds of setup times are called carryover sequence-dependent setup time.

---

*Corresponding author

ISSN: 1735-8272, Copyright c 2020 JISE. All rights reserved

To the extent that similarity of the sequential processed tasks is more, the setup time is less. Therefore, to reduce setup time, they are placed in similar groups and idiomatically, group scheduling is used. Using group scheduling results in the reduction of implementation times, half-finished tasks, transition costs, and also as a result of reduction of the setup times, less energy is consumed for production and the imposed costs of production reduces (Zhou and Peng , 2019).

The environment studied in this research is a single-machine environment. It is assumed that tasks have been available in the zero time and been classified in some groups. Processing all the tasks is performed by using a machine. Setup times are dependent on carryover sequence and machine is not placed in a certain time interval. Tasks that are placed in a group should be set up completely before processing. This act is called the external setup. This act is performed using the same machine operator when another group is automatically processing on machine. By this act, the role of the external setup operator is deleted completely. The setup dependent on the carryover sequence is called internal setup i.e. machine setup.

The main focus of the problem is on uniting external and internal setup times. This means that when machine is processing a group, machine operator is setting up the group or next groups. If this act is finished before the completion of the group processing, operator will be idle for a while but, if it is finished after that, machine will be idle for a while and as a result, these two conditions will influence the task completion time. Also, as it was said, it is assumed that machine cannot be accessed in a certain time interval (A, B), for example for keeping or repairing and in this period, processing is not performed by machine. If we assume that the act of keeping and repairing is also performed by machine operator; therefore, we will not have the external setup. In the given problem, the preemption is not allowed. In the production systems that follow the philosophy of just-in-time production (JIT), if a task sets up premature or late, they will be fined (Niu et al., 2019).

Therefore, in this research, minimizing the sum of the premature and late penalties of the tasks' delivery (not groups) has been considered so that these penalties are assumed to be related to the time. This research includes 5 parts that in the first part, the problem was investigated and in the second part, previous studies were evaluated. In the third part, the research method was presented; in the fourth part, results were presented, and in the last part, conclusion and future suggestions were provided.

2-Literature review

Determination of the tasks' sequence and their scheduling is a kind of decision making process that is used in many production and service industries. This decision making process has a significant role in reduction of the costs, increase of productivity, increase of customer satisfaction and generally, increase of the companies' competitive advantage. As a result, in today's competitive world, determination of the tasks' sequence and their scheduling has changed into a need for survival of companies in the market, effectively. Moreover, regarding the complexity and high variety of products in today's world which is derived from the difference and variety of the customer's needs, the importance of using effective methods for scheduling is raised more than the past. Some of the scheduling models include: single-machine scheduling, parallel-machine scheduling, flow working-place scheduling, working-place scheduling, etc.

Therefore, the problem of scheduling to minimize the completion time of the last task and maximum delay of tasks on a machine which cannot process more than one task in each time, is solved, especially when tasks have equal time access and each task has different size of the order and tasks belong to different families and only collateral tasks have the ability to be processed simultaneously by the machine. Sabouni and Jolai (2010) studied the carryover sequence-dependent setup times for the problem of group scheduling in a single-machine environment. These researchers seek to minimize the maximum time for task completion. Also, they considered two internal setup time (setting up the machine for processing a task) and external setup (setting up a task to begin processing on a machine) jointly. One of the important data of these researchers can be addressed as such that during processing a group by machine, operator can set up externally just one group (the next group) (Sabouni and Jolai, 2010). da Rocha et al., (2016) have studied the problem of scheduling unrelated parallel machines with the assumption of existence of sequence-dependent installation times with the aim of minimizing maximum time for completion of the tasks and weighted sum of the delay times.

They have used the branch and bound method which is a method for obtaining solution along with GRASP meta-heuristic method. Using two planning models of the mixed integer programming, they have found that the branch and bound method has a better performance in relation to the two mentioned methods (Alemão et al.,2018). Joo and Kim, (2013) have studied with the objective function of minimizing weighted sum of the tasks' delay in which tasks are processed in the form of groups. In this problem, they assumed that tasks have equal processing and delivery times and using methods of the earliest weighted delivery deadline, the shortest weighted processing time, innovative two-level group scheduling method, and annealed simulation method acted to solve this problem (Joo, and Kim, 2013). Aouam et al., (2018), in his research, analysed the presentation value of decision makers with flexibility to accept or reject the orders when the amount of orders is not clear. This flexibility identifies the requirement in two problems of production planning and integrates the first problem of accepting the order with the problem of capacity building production amount and present options to reject the orders, if high initiation cost is needed and to use the economics advantage, the scale is not collected with excess orders.

The second problem integrates the order acceptance with the planning problem of releasing the order with times of performing the order based on the load. Liu et al., (2019) in their research "the integrated decision model" stated that preemptive keeping decisions are coordinated based on the forewarning information with machine planning decisions so that the whole considered cost is minimized. In the integrated model, condition of simulated health and age is considered according to the system damage. Finally, the case study is used to show the value of the proposed methods and function of the integrated solution is compared to the achieved results of the preemptive keeping planning solution and problems of the production planning, independently. Results prove its efficiency.

3-Research method

The problem of production machine scheduling is one of the most important problems in the production and service systems. In this research, according to the above data in scheduling of this machine, we deal with the green scheduling that results in the reduction of the production costs and energy consumption. Also, due to the importance of the sequence-dependent setup time in many industries, the assumption of machine setup time dependency to the task's sequence is considered in this research. Researchers conducted about machinery are still theoretical and has distance with reality and by considering the limitation of time access to the tasks, sequence-dependence installation time, and limit access to machines, it can become closer to reality. Many studies are performed to improve production planning but these studies assumed the tasks' sequence as insignificant scheduling and ignored them. In this research, we are seeking to design an integrated mathematical model of production planning on a machine to reduce the costs of pausing and minimizing the earliness and tardiness penalties in production.

Indexes and sets

I: set of all the parts' class
i and i': index of the parts' class
J: set of all the task stages on a system
j and j': index of the task's stages
K: set of all the processes of each stage
k and k': the process index
J_i: set of all the task stages that is performed on a system for every class of parts
A_i: the first stage of the task's stages
Z_i: the last stage of the task's stages

Parameters

PT_{ijk}: the necessary time to process the i part group in the task stage j from the process k
RW_{ijk}: the necessary time to keep the system while producing the i part group in the task stage j from

the process k
ST_{ijk}: time of initiation of the system (machine) *k* in the stage *j* to process the group part *i*
LT_{ijk}: time interval of concurring human forces with the system initiation in the process *k* in the task stage *j* to process the group part *i*(time preemption to simultaneous setup of the human forces with system)

Variables
C_{ijk}: the time of finishing processing the group part *i* in the task stage *j* in the task process *k*

Y_{ijk}: binary variables if: group part *i* in the task stage *j* in the task process *k* is processed, it equals 1 otherwise it is 0

X_{iijk}: binary variables if: the group part *i* in the task stage *j* and in the task process *k* is processed, it equals 1 otherwise it equals 0.

Constraints

$$\sum_{i\neq i_1}^{i\in I}\sum_{i'\in I} X_{ii'jk} \leq 1 \qquad \forall j\in J, \forall k\in K \tag{1}$$

$$\sum_{k\in K_{ij}} Y_{ijk} = 1 \qquad \forall i\in I, j\in J_i \tag{2}$$

$$C_{ijk} \leq MY_{ijk} \qquad \forall i\in I, j\in J_i, k\in K_{ij} \tag{3}$$

$$C_{ijk} \geq \sum_{k'} C_{ij'k'} + ST_{ijk} + PT_{ijk} + RW_{ijk} + LT_{ijk} - M(1-Y_{ijk}) \tag{4}$$

$$\forall i\in I, j, j'\in J_i, j\neq A_i, j'=j-1, k\in K_{ij}$$

$$C_{ijk} \geq Y_{ijk} \times (ST_{ijk} + PT_{ijk} + RW_{ijk} + LT_{ijk}) \qquad \forall i\in I, \forall j\in A_i \forall k\in K_{ij} \tag{5}$$

$$X_{ii'jk} + X_{i'ijk} \leq 1 \qquad \forall i, i'\in I, i<i', j\in J_i \cap J_{i'}, k\in K_{ij} \cap K_{i'j} \tag{6}$$

$$2X_{ii'jk} \leq Y_{ijk} + Y_{i'jk} \qquad \forall i, i'\in I, i\neq i', j\in J_i \cap J_{i'}, \mathrm{k}\in \mathrm{K}_{ij} \cap Ki'j \tag{7}$$

$$Y_{ijk} + Y_{i'jk} \leq X_{i'ijk} + X_{ii'jk} + 1 \quad \forall i, i'\in \mathrm{I}, \mathrm{i}<i', \mathrm{j}\in J_i \cap J_{i'}, \mathrm{k}\in \mathrm{K}_{ij} \cap K_{i'j} \tag{8}$$

$$C_{ijk} \geq C_{i'jk} + ST_{ijk} + PT_{ijk} + RW_{ijk} + LT_{ijk} + -M \times X_{i'ijk} - 2M + MY_{ijk} + MY_{i'jk} \tag{9}$$

$$\forall k\in K: N_k > 1, j = J_k, i, i'\in I_k : i<i'$$

$$C_{i'jk} \geq C_{ijk} + ST_{i'jk} + PT_{i'jk} + RW_{i'jk} + LT_{i'jk} + -M \times X_{ii'jk} - 2M + MY_{ijk} + MY_{i'jk} \tag{10}$$

$$\forall k\in K: N_k > 1, j = J_k, \mathrm{i}, \mathrm{i'}\in \mathrm{I}_k : i<i'$$

$$Min\left\{Max \sum_{k\in Kiz_i} C_{iz_i} k\right\} \tag{11}$$

The required variable change:

$\underset{i}{Max} \sum_{k\in iz_i} C_{iz_i k} = T$ And instead of that it turns to T in the objective function that it is a constraint: $t \geq Max \sum_{k\in K_{iz_i}} C_{ikz_i}$ (12)

In every task stage and every process, only one part can be processed (1). Only one time the task process is done on every group of the parts in each task stage (2). The relation between the binary variable of production and the processing time of each part in its task stage and implementation process (3). Limitation that processing time of i part in the j stage in the task process k is done after the other part (4). The time to finish every part i task in the j task stage from the task process k includes task implementation time, setting up the part to implement the task and the time of lack of access to the machine during the process and time interval of concurring of the operator with the system (5). Constraint of the operation sequence of each group part on the system in every task stage and implementation process (6, 7, 8). Constraint related to the non-interference of the production of parts during production (9). Constraint related to the non-interference of the tasks based on the production operation sequence (10). The objective function based on minimizing the earliness and tardiness time of the group part i production after the task stage j and production processes k (11).

4-Research findings

Complex systems have many problems that contain combinatorial nature. For example, the production path of the parts should be determined and setup process should be done for all the parts. One of the most important combinatorial optimization problems is scheduling problem. Complexity theory tells us that combinatorial problems are not often polynomials. These problems in their operational and practicable sizes are such great that their optimal solution cannot be achieved during the acceptable time. However, these problems should be solved; therefore, there is no solution for suboptimal solutions to be sufficed in a way to be acceptable and achieve in the acceptable time. Several approaches have been suggested for designing acceptable and qualified solutions under acceptable time limit. These approaches are algorithms that can guarantee finding good solutions in a certain distance of the optimal solution that are called approximate methods. There is also another algorithm that guarantees to produce a solution close to optimal with high possibility which is called probabilistic algorithms. Apart from these two groups, the algorithms can be accepted that have no guarantee to present the solution but, based on the evidence and records of their results, on the average, they have the best quality contrast and time to solve the studied problem. According to the modelling performed to prove the mathematical model, an example was studied to examine the accuracy of the model. Therefore, the main parameters of the model are as below:

Table 1. Mathematical model inputs

Uniform (10-20)	PT_{ijk}
Uniform(5-15)	RW_{ij}
Uniform(5-10)	ST_{ijk}
Uniform(10-20)	LT_{ijk}

According to the changes in the objective function, the given problem is a linear planning model of the integer that was implemented using GAMS software. This implementation by the system with properties CPU core I3, RAM 4 in the time limit 0.08 seconds has achieved the optimal solution with zero gaps.

4-1-The problem complexity

GAMS software can calculate the optimal solution of the problems in the small size but, real problems or problems with large size cannot be solved by GAMS in a rational time due to high number of the variables and limitations. For example, a problem with 5 tasks in the GAMS and based on the listed parameters in the table1 is solved after 8 seconds and 665, 372, and 595 repetitions. Therefore, such problems are large problems. Based on the results, it was observed that solutions of these two methods (precise and meta-heuristic) for the problems with small sizes are similar. To prevent an unlimited increase of solution, a superior set is presented.

4-2-Genetic algorithm approach

The main idea of evolutionary algorithms was proposed by Richenberg in 1960. Genetic algorithm originates from these algorithms. Actually, a computer search method is based on optimization algorithm and the gene and chromosome structure that was proposed by professor Holland at Michigan University and after him, it was developed by some of the students like Goldberg in 1989 (Goldberg and Holland, 1988). Genetic algorithm is inspired by Darwin theory of evolution and based on the principle of the survival of the best. It can be said that one of the main advantages of genetic algorithm in relation to the old methods of optimization is that GA works with a population or set of points in a certain time while old methods of optimization just work with a certain point. This means that GA processes many designs at a time. Some studies investigate genetic algorithm in detail. The main steps in implementation of the genetic algorithm are as follow:

1. Chromosome form
2. The initial population
3. The function competence to calculate population proportion
4. Selection strategies
5. Genetic operators

4-3- The proposed genetic algorithm

To solve the scheduling problem of a machine by considering preemption and the idle time of machine, the genetic algorithm has been presented. The main components of the GA implementation include:

The chromosome form

To implement genetic algorithm, the first step is to write the specific properties of the solution in the form of a chromosome string. Each chromosome is made of a combination of genes and certain alphabet. This alphabet can be a set of binary numbers, real numbers, natural numbers, symbols, or matrices. The presenting design not only determines the effective amount of the problem design but also it determines that to what extent genetic operators can be used effectively. To show this, two tasks have been considered and the processing time, the finishing time, earliness and tardiness penalties is shown in table (2). Also, chromosome codification is shown graphically in the figure (1).

Table 2. Two tasks parameters

	PT_{ijk}	RW_{ij}	ST_{ijk}	LT_{ijk}
First task	4	5	2	4
Second task	2	7	1	3

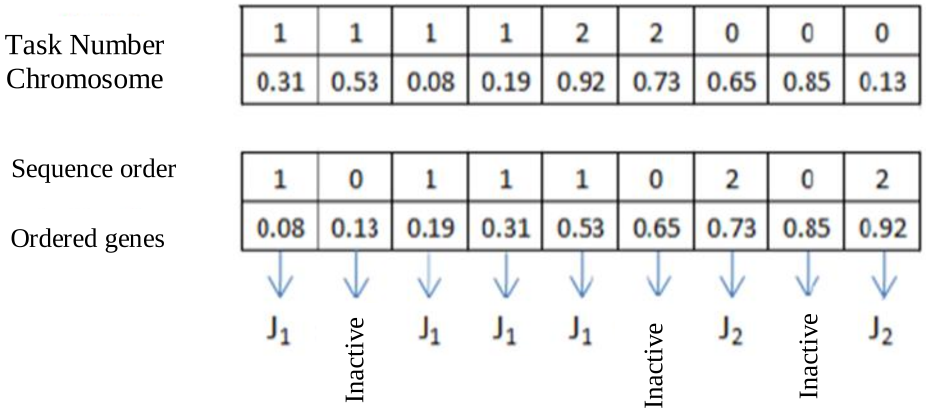

Fig 1. System codification

Initial population

The second stage of the genetic algorithm implementation is producing a series of initial solutions which are called population. Number of the initial solutions that should be placed in this population is called the size of the population. The initial population produced just one time at the beginning of the first genetic algorithm's generation. Determination of the suitable size of the population is an important selection in the implementation of the genetic algorithm. If the selected amount is small, we may not be able to reach a good solution. On the other hand, if this number is too big, it may allocate itself a long time from CPU to reach a better response.

Competence function

In the GA implementation, a fitness function is used to evaluate and reproduce new chromosomes named "the next generations' offspring". The purpose of this function is measuring the proposed candidates in the population by considering objective functions and constraints in the available model. According to the objective function, the chromosomes' fitness should be performed by calculating the two parameters of the start time and the completion time.

To produce the next generations, we select the intersection and parent's mutation operators in pairs from the population, i.e. a mechanism by which a new population of the available individuals in the size of the present population is formed. The size of the population is fixed during the process which is explained as follow.

Intersection: offspring or new solutions are generated by contact of two other order or parents through an operator named intersection. The intersection operators should prevent from producing impossible solutions. The purpose is to produce better offspring that is making better orders after combining the parents. According to the expansive variety of the suggested intersection operators, the considered intersection in this article is as below. To show the intersection operator consider two orders of one of the parents below:

The first parent: A B C |A A B| A C B A
The second parent: A A B | B B C | C A A A

Cutting points are determined randomly. In the following after replacing characters after right cutting to the beginning of the order, an order is formed that its results is as below:

New sequence 1: A C B|A A B| C A A B (A C A A A A B)
New sequence 2: C A A|A A A |B B B C (C A A A B B C)

From the new order, characters that conform to the characters between cutting points of the other parent are picked up (as is shown in the parentheses). Then, the order between the parent's parentheses and the remained characters from the other new order are used to produce offspring. Using this process, below offspring is produced:

Offspring 1: B B C |A A B | C A A A
Offspring 2: A A B | B B C |A C A A

5-Designing the experiments and regulating the data

In this section, the efficiency of the proposed GA is presented for this problem. The purpose is to experiment the efficiency of the proposed GA for the scheduling problem of the single-machine with allowed idleness, earliness and tardiness costs and recognition of the complex samples. To show the efficiency of the proposed algorithm, problems were produced randomly as below.

Problems with small size with 3,4, and 10 tasks, average size with 20 and 40 tasks, and large size with 80 and 100 tasks have been considered. Parameters of the model from table 1 are formed randomly. To classify the algorithm, there are several methods to design the experiments but, the approach that has the most use is a complete factorial experiment (Montgomery and Reitz, 2000). This approach is not always useful because implementation of the inspections when the number of factors considerably increases, solving the problem will be harder. To reduce the necessary number of the experiments, the factorial fraction experiment (FFE) was developed (Cochran and Cox, 1992). Taguchi (1986) developed a family of FFE matrices that eventually reduces the number of the experiments but it still provides necessary information. In the Taguchi method, to study many numbers of deciding variables with different experiments, the original arrays have been used. Here, below values are considered for the necessary parameters in the genetic algorithm. Possible intersection (pc): three levels (80.0, 85.0, and 90.0), possible mutation: three levels (09.0, 06.0, and 03.0), number of the initial population (np): three levels (50, 100, and 200), number of generations (ng): one level (100). The best value of the calculative experiments for the scheduling problem of the single-machine with sum of the used earliness and tardiness through results of the experiments with parameters changes that were considered as pre-assumption values is as below:

Ng=100, np=100, pm=0.06, pc=0.80. Table 3 shows that the designed genetic algorithm in contrast to GAMS, has solved average and large number of tasks in a proper time. Moreover, GAMS has solved small number of tasks in an acceptable time. The last column of the table 3 shows the percentage of difference between the optimal solution and the GA response. In all the solutions in the levels of small tasks, the value of Gap was zero and shows that the genetic algorithm in the level of small tasks has reached the GAMS optimal solution in the best response.

Table 3. Comparison of the GAMS and genetic solving

Row	**GAMS**		**Algorithm GA**			**GAP**
	T(s)	f_{opt}	f_{best}	f_{avr}	t(s)	
1	44	151	151	151	2, 5	0
2	98	152	152	152	2, 6	0
3	120	189	189	189	3, 2	0
4	125	340	340	340	3, 8	0
5	140	524	524	524	4, 3	0
6	-	-	935	980	5, 6	-
7	-	-	1050	1120	5,4	-
8	-	-	1135	1242	6,9	-
9	-	-	1530	1733	9,3	-
10	-	-	1745	2130	14,3	-

6-Conclusion

In this article, the development of the single-machine problem with earliness and tardiness costs, task failure and the idle time of machine was studied. For this problem, a mathematical model was presented that finds the configuration of the tasks' order with the aim of minimizing scheduling costs. An efficient algorithm was designed to solve this mathematical model. To confirm efficiency of the proposed GA, 10 experiment problems were solved. Calculation results were compared in relation to the problem’s solutions and since GAMS has not the ability to implement acts with average and large number of tasks, it shows superiority of the proposed algorithm in ordering the tasks in relation to the precise methods. Moreover, the genetic algorithm was compared in the light of time. By increasing the number of tasks, the ability to use algorithm to have proper calculative time has been confirmed.

References

Alemão, D., Parreira-Rocha, M., & Barata, J. (2018, January). Production and Maintenance Scheduling Supported by Genetic Algorithms. In *International Precision Assembly Seminar* (pp. 49-59). Springer, Cham.

Aouam, T., Geryl, K., Kumar, K., & Brahimi, N. (2018). Production planning with order acceptance and demand uncertainty. *Computers & Operations Research*, *91*, 145-159.

Cochran, W. G., & Cox, G. M. (1992). Notes on the statistical analysis of the results.

da Rocha, J. S. A., Sala, A. D., de Almeida, E. B., Mancusi, F. C. M., Gerolin, F. S. F., Bucione, F. T. S., ... & Moraes, S. (2016). Relato de experiência: construção do modelo assistencial Hospital Alemão Oswaldo Cruz. *Revista Acreditação: ACRED*, *6*(11), 72-85.

da Silva, N. C. O., Scarpin, C. T., Pécora Jr, J. E., & Ruiz, A. (2019). Online single machine scheduling with setup times depending on the jobs sequence. *Computers & Industrial Engineering*, *129*, 251-258.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning.

Joo, C. M., & Kim, B. S. (2013). Genetic algorithms for single machine scheduling with time-dependent deterioration and rate-modifying activities. *Expert Systems with Applications*, *40*(8), 3036-3043.. Expert Systems with Applications, 41 (2), 3136-3143.

Liu, Q., Dong, M., Chen, F. F., Lv, W., & Ye, C. (2019). Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and Computer-Integrated Manufacturing*, *55*, 173-182.

Montgomery, D. T., & Reitz, R. D. (2000). Optimization of heavy-duty diesel engine operating parameters using a response surface method. *SAE transactions*, 1753-1765.

Niu, S., Song, S., Ding, J. Y., Zhang, Y., & Chiong, R. (2019). Distributionally robust single machine scheduling with the total tardiness criterion. *Computers & Operations Research*, *101*, 13-28.

Pei, J., Cheng, B., Liu, X., Pardalos, P. M., & Kong, M. (2019). Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. *Annals of Operations Research*, *272*(1-2), 217-241.

Sabouni, M. Y., & Jolai, F. (2010). Optimal methods for batch processing problem with makespan and maximum lateness objectives. *Applied Mathematical Modelling*, *34*(2), 314-324.

Sun, X., & Geng, X. N. (2019). Single-machine scheduling with deteriorating effects and machine maintenance. *International Journal of Production Research*, *57*(10), 3186-3199.

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes* (No. 658.562 T3).

Zhou, B., & Peng, T. (2019). New single machine scheduling with nonnegative inventory constraints and discretely controllable processing times. *Optimization Letters*, *13*(5), 1111-1142.