



Paper Type: Research Paper



Production Scheduling of Parallel Identical Lines in a Multi-Product Manufacturing System with Genetic Algorithm

Mohammad Hossein Sattarkhan* 

Department of Industrial Engineering, Faculty of Industrial Engineering, Yazd University, Yazd, Iran; mh.sattarkhan@stu.yazd.ac.ir.

Citation:



Sattarkhan, M. H. (2023). Production scheduling of parallel identical lines in a multi-product manufacturing system with genetic algorithm. *International journal of research in industrial engineering*, 12(1), 7–20.

Received: 30/03/2022

Reviewed: 02/05/2022

Revised: 08/06/2022

Accepted: 15/07/2022

Abstract

A multi-product system is one of the different manufacturing systems in which many products have been produced that complement each other and have interdependence. These types of systems have recently been widely used in various industries. In some types of multi-product manufacturing industries that offer their products as a package, the scheduling of the production of components of each package affects the time it takes to complete the package. Therefore, a new problem has been defined that the primary purpose of its production scheduling, in addition to reducing the completion time of the products, is to make various items forming a package, get ready over a short interval of time, and be supplied to the sales unit so that the package can be delivered to the final consumer. This paper aims to express the problem of production scheduling of multi-product production systems in the form of linear programming. For this purpose, two mathematical models are presented, and their functions are compared. Besides, an efficient genetic algorithm is proposed to solve the problem, which is able to solve the problem in a reasonable time with acceptable accuracy.

Keywords: Parallel lines production scheduling, Operations sequence, Mixed-integer linear programming, Genetic algorithm.

1 | Introduction

Production planning is a branch of science that focuses on planning and scheduling productions at different levels of decision-making. Production planning has long been the subject of many studies to increase productivity in manufacturing systems. The problem of scheduling and sequencing of the operations is a branch of production planning that allocates any activity to the machine or the production line at the right time and determines the best possible sequencing of activities on machines or production lines. Various objectives have already been defined in these types of problems, each of which has been identified and resolved according to a specific need.

The production scheduling problem basically consists of the selection of a set of tasks to be performed and the construction of a schedule complying with the technological requirements and satisfying as much as possible the given demands for a final production [1]. With the global economy's rapid development, the manufacturing industry's production model has changed [2]. The



International Journal of Research in Industrial Engineering. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).



Corresponding Author: mh.sattarkhan@stu.yazd.ac.ir



<https://doi.org/10.22105/riej.2022.335596.1306>

real-life production scheduling problems may encompass many specific technological and business requirements such as due dates, sequence-dependent changeovers, unit blockages, etc. [3].

A multi-product system is one of the different manufacturing systems in which many products have been produced that complement each other and have interdependence. These types of systems have recently been widely used in various industries.

In recent years, some types of multi-product manufacturing industries, including the ceramic tile industry, have offered their products as packages. The components of each package have their own production schedule. The completion times of these components may directly affect the appropriate management of the supply chain, customer satisfaction, etc. Therefore, a new problem has been defined that the primary purpose of its production scheduling, in addition to reducing the completion time of the products, is to make various items forming a package, get ready over a short interval of time, and to be supplied to the sales unit so that the package can be delivered to the final consumer.

In this paper, we attempt to express such a problem in the form of a linear programming problem and solve it. Therefore, two mathematical models are built that, unlike the usual procedure in the literature, the objective function of both models is to reduce the weighted sum of the time intervals between the various products of each package. Both models take into account the general state of n production lines and m products (jobs), and the relevant constraints on reality are developed in the models. To our knowledge, this is the first study that considers this objective function for a real-world problem and would be the paper's most important contribution. Besides, since it is impossible to solve either of the two proposed models for large-scale problems precisely in a meaningful time, an efficient genetic algorithm is proposed to solve the problem logically.

2 | Literature Review

Parallel machines' scheduling has been the subject of several pieces of research over the years, and each has added a new dimension to the problem, depending on its intended application. Considering setup times/costs [4]-[8], [17], [24], cost of earliness/tardiness [4]-[6], [12], [15], [17]-[19], [24], and machines availability [9], [10] are the examples of various issues that gradually have been applied in the formulation of the problem.

Kim et al. [4] focused on the scheduling problem of identical parallel machines to minimize total tardiness. At the same time, there are Sequence-Dependent Setup Times (SDST) between the jobs with different part types. They presented a mathematical model with two encoding schemes for meta-heuristic solutions and three decoding methods for obtaining a schedule from the meta-heuristic solutions. They developed six different Simulated Annealings (SA) and genetic algorithms, with six combinations of two encoding schemes and three decoding methods. They then performed computational experiments to find the best combination. Their suggested algorithm provided better solution quality with less computation time than commercial optimization solvers.

With the same objective, Zhu and Heady [5] developed a mixed-integer programming formulation to minimize job Earliness and Tardiness (ET) in a multi-machine scheduling problem, which considers setup times for the jobs, due dates, and also cost penalties. At the same time, the characteristics of the machines are not uniform. Also, Omar and Teo [6] studied the problem of identical parallel machine scheduling with specific due dates and early due date restrictions. They developed a mixed-integer programming model to tackle such problems. The objective is to minimize the sum of earliness/tardiness in the presence of setups.

Lee et al. [7] considered the scheduling problem of two identical parallel machines with multi-attribute setup times, while each job has some attributes, and each attribute has several different levels. The objective is to minimize the makespan. They presented a heuristic and a Variable Neighborhood Search

(VNS) metaheuristic. Also, Heydari and Aazami [8] developed a two-objective model to solve job shop scheduling problems with SDST when the aim is to optimize both objectives (makespan and maximum tardiness) simultaneously. They utilized the ϵ -constraint method to solve the model. A set of generated numerical data validates the model's efficiency and flexibility.

Wang and Cheng [9] investigated a problem of two identical parallel machine scheduling, in which one machine is available for processing jobs in a limited time interval. In contrast, the other machine is always available over the scheduling horizon. The objective is to maximize the number of on-time jobs. They also developed a heuristic to deal with the problem. Liao and Sheen [10] studied the scheduling problem of identical machines considering machine availability and eligibility constraints to minimize the makespan. In contrast, the machines are not continuously available at all times, and each job can only be processed on specified machines. A network flow approach is utilized to formulate the problem into a series of maximum flow problems. They proposed a polynomial-time binary search algorithm to verify the infeasibility of the problem or find the optimal solution if a feasible schedule exists.

Jia et al. [11] studied the scheduling problem of parallel batch machines with arbitrary capacities, where the non-identical-size jobs have identical processing times and unequal weights. After being processed, the jobs are delivered to the customers by some vehicles. The objective is to minimize the total weighted delivery time of all jobs. To solve the problem, they presented two heuristic algorithms, developed an algorithm based on ant colony optimization, and compared their performance.

Tavakkoli-Moghaddam et al. [12] presented a new mathematical model for a multi-criteria parallel machine scheduling problem to minimize the total earliness, tardiness penalties, and machine costs. They proposed a metaheuristic method based on the genetic algorithm and represented computational results.

Cheng et al. [13] considered parallel batch processing machines scheduling problems, where the job sizes are non-identical and are processed in batches, and the machines' capacities are the same. Using a mixed-integer programming method, they presented models for minimizing makespan and total completion time and then provided a polynomial time algorithm for minimizing the two objectives. Different-scale random instances were used to test the effectiveness of the proposed algorithm. In a similar study, Muter [14] investigated the scheduling problem of single and parallel batch processing machines to minimize makespan and presented a reformulation for the scheduling of parallel batch processing machines, which is based on decomposition in two levels, and proposed an exact algorithm to solve it. Mirmohseni et al. [15] developed a dynamic programming framework for minimizing total tardiness for sequencing weighted jobs on a single machine. Fuzzy numbers were utilized to cope with the uncertainty.

Shabtay et al. [16] studied a single-machine scheduling problem, where there exists an expected due date for all jobs to minimize the objective function, including job-dependent penalties due to early and late work. Then they provided a pseudo-polynomial time algorithm to solve the problem and studied two particular cases that are solvable in polynomial time. Rafiei et al. [17] presented a mathematical model for optimizing multi-product single-machine scheduling problems when a considerable percentage of available production times is allocated to machine setup times. The model considers sequence-dependent setup costs, costs of delays in deliveries, holding costs, and costs of idle times. The objective is to minimize the total production time, earliness, and tardiness times. Random small-size test cases are defined and solved.

In recent years, some new topics have been considered. Considering energy consumption is one of these new topics [18]-[21]. In addition, several pieces of research have studied the integration of production scheduling and other subjects such as maintenance planning, distribution planning, etc. [22]-[25].

Antoniadis et al. [18] considered the problem of scheduling jobs on parallel machines with release dates, deadlines, and processing times, which aims to minimize the total energy consumed. Machines may be in one of the two states: 'sleep' or 'active'. By entering into the 'sleep' state, they consume no energy. Each machine requires L units of energy to awaken from the 'sleep' state, and by entering into its 'active' state,

the machine can process jobs and consumes a unit of energy per unit of time. They provided a constant approximation algorithm for this problem.

Módos et al. [19] considered a production scheduling problem in companies with large electricity consumption, where there is one machine and release times for the operations. The objective function is to minimize total tardiness. They presented robust production schedules that guarantee that the energy consumption constraints are not violated for each given set of uncertainty scenarios. A pseudo-polynomial algorithm was proposed to find the optimal robust schedule of the given sequence of operations. Then, they utilized this algorithm in three different (two exact and one heuristic) algorithms for finding the optimal sequence.

Aghelinejad et al. [20] investigated a single-machine manufacturing system to minimize the production system's total energy costs. They presented two mathematical models to formulate such a problem and developed a heuristic and a genetic algorithm to solve the model and provide solutions in reasonable computational time. Different numerical experiments were utilized to test the effectiveness of the proposed optimization methods. The results approved the accuracy and efficiency of both algorithms.

Anghinolfi et al. [21] investigated the multi-objective combinatorial optimization problem of scheduling jobs on multiple parallel machines, while the objective is minimizing both the makespan and the total energy consumption. An ad-hoc heuristic method was developed to solve the problem.

Bhosale and Pawar [22] considered integrating production planning and scheduling. They selected a case study based on the parallel-line continuous process plant and optimized its performance by a real coded genetic algorithm. Results represent that the algorithm outperforms the solutions obtained by previous researchers.

Cui et al. [23] investigated the integration of production scheduling and maintenance planning to optimize two objectives of quality robustness and solution robustness for flow shops when the occurrence of failure is uncertain. They proposed a mathematical model to formulate the problem. They also presented a two-loop algorithm that optimizes the sequence of jobs, positions of preventive maintenance, and idle times. Computational results approved the performance of the proposed algorithm. Also, Chansombat et al. [24] also utilized mixed-integer linear programming and presented a model that simultaneously solves the integrated production and preventive maintenance scheduling problem in the capital goods industry. The objective was to minimize total costs, including earliness/tardiness penalty costs, component and assembly holding costs, preventive maintenance costs, and the costs of setup, production, transfer, and production idle time. They tested the model using real data. The results show that the total cost may be reduced to 63.5%.

Furthermore, Devapriya et al. [25] focused on the integrated production and distribution scheduling problem of a perishable product that its production and distribution must be done before it becomes unusable. Minimizing the costs is considered when the product has a limited lifetime, and the total demand must be satisfied within the planning horizon. They presented a mixed-integer programming model to solve the problem, then provided heuristics based on evolutionary algorithms to resolve the model. Sifaleras et al. [26] proposed a mathematical production-planning model for a real-world production optimization problem of a non-alcoholic soft drinks company in Northern Greece. The model's objective is to minimize the company's idle human-hours subject to fulfilling customers' demands. Then they solve the model using Python and Gurobi solver.

Various objectives have been considered in the formulation of production schedules all over the world. Many of them studied minimizing the makespan. Others applied to minimize (total) costs, total completion time, total earliness and/or tardiness, total energy consumed (costs), delivery times, and maximizing the number of on-time jobs. To the best of our effort, minimizing the time intervals between

the completion times of different items of each package has not been studied before. Therefore, since several industries use multi-product systems, it requires a comprehensive study.

3 | Problem Statement

The research problem is determining the sequence and scheduling of producing the products offered in a product package. Each product package consists of some products; each has its own processing time and limitations; e.g., in a ceramic tile factory, a product package may include various components, such as floor tiles, wall tiles, decors, borders, etc. The goal is to schedule the production such that reducing product completion times minimizes the difference between the completion times of the first product and the final product in each package. In the literature, frequently minimizing the makespan has been considered the objective, which does not apply to this research.

3.1 | Definitions and Assumptions

The general state of n production lines and m jobs is considered. We define the first product of the first package as job 1, the second product of the first package as job 2,... and the final product of the last package as job m .

The following concepts are assumed in modeling the general state of the problem:

- I. Production lines are identical, and each one is able to do all the jobs.
- II. Preemption is not permitted; i.e., assigning one job to one line requires the whole job to be done on the same line, and part of the job is not entitled to transfer to another line. It is also not allowed to interrupt and perform part of it at another time.
- III. Setup times for all products on all lines are negligible.
- IV. Each line is capable of doing one job at a time.
 - V. Each job may be replaced with another job on each line (no precedence is considered).
- VI. All jobs may start from time zero.
- VII. All lines from time zero and during the planning horizon are continuously available and capable of operating (there is no unavailability to the lines).
- VIII. The processing time of each job on each line is known.
 - IX. All lines have continuous production and no idle time.

4 | Mathematical Modeling

4.1 | Model 1

Some indices are used to model this problem, which we describe below.

4.1.1 | Indices used in the first model

i : line (machine) index, $i = 1, 2, \dots, n$.

j : job index, $j = 1, 2, \dots, m$.

k : package index, $k = 1, 2, \dots, o$.

T_{ij} : the processing time of job j on line i .

M : a large positive number.

W_k : the importance (weight) of package k.

F_k : the set of jobs related to package k.

4.1.2 | Decision variables

In this linear integer model, six types of decision variables are used:

U_k : completion time of the last product of package k which gets prepared, $U_k = \max \{C_j / j \in F_k\}$.

L_k : completion time of the first product of package k which gets prepared, $L_k = \min \{C_j / j \in F_k\}$.

Y_{ij} : 1, if job j gets done on line i, otherwise 0.

$X_{ijj'}$: 1, if job j gets done immediately after job j' on line i, otherwise, 0.

X^1_{ij} : 1 if job j gets done as the first job on line i, otherwise 0.

C_j : completion time of job j.

Y_{ij} , $X_{ijj'}$ and X^1_{ij} are zero-one variables; U_k , L_k , and C_j are non-negative ones.

4.1.3 | First model

The proposed linear model is as follows:

$$\begin{aligned} \text{Min } Z &= \sum_{k=1}^O W_k \cdot (U_k - L_k), \\ \text{s. t.} \end{aligned} \quad (1)$$

$$\sum_{i=1}^n Y_{ij} = 1 \quad \text{for all } j, \quad (2)$$

$$C_j \leq X^1_{ij} \cdot T_{ij} + M(1 - X^1_{ij}) \quad \text{for all } i, j, \quad (3)$$

$$C_j \geq X^1_{ij} \cdot T_{ij} - M(1 - X^1_{ij}) \quad \text{for all } i, j, \quad (4)$$

$$C_j \leq C_{j'} + X_{ijj'} \cdot T_{ij} + M(1 - X_{ijj'}) \quad \text{for all } i, j, j', j \neq j', \quad (5)$$

$$C_j \geq C_{j'} + X_{ijj'} \cdot T_{ij} - M(1 - X_{ijj'}) \quad \text{for all } i, j, j', j \neq j', \quad (6)$$

$$\sum_{i=1}^n \sum_{\substack{j'=1 \\ j' \neq j}}^m X_{ijj'} + \sum_{i=1}^n X^1_{ij} = 1 \quad \text{for all } j, \quad (7)$$

$$\sum_{\substack{j'=1 \\ j' \neq j}}^m X_{ijj'} + X^1_{ij} = Y_{ij} \quad \text{for all } i, j, \quad (8)$$

$$\sum_{\substack{j'=1 \\ j' \neq j}}^m X_{ijj'} \leq Y_{ij'} \quad \text{for all } i, j', \quad (9)$$

$$\sum_{j=1}^m X^1_{ij} = 1 \quad \text{for all } i, \quad (10)$$

$$L_K \leq C_j \quad \text{for all } K, j \in F_K, \quad (11)$$

$$U_K \geq C_j \quad \text{for all } K, j \in F_K, \quad (12)$$

$$U_K, L_K, C_j \geq 0 \quad \text{for all } K, j, \quad (13)$$

$$Y_{ij}, X_{ijj'}, X^1_{ij} = 0, 1 \quad \text{for all } i, j, j'. \quad (14)$$

Eq. (1) represents the objective function of the problem. As stated, the purpose is to determine the production sequence of different products on different lines so that the least possible time interval would exist between the completion times of various items of each package.

Eq. (2) ensures that each job is only performed on a single line. *Eqs. (3) to (6)* indicate the completion times of different items and guarantee that idle time does not occur on the lines. *Eqs. (3) and (4)* guarantees that each line's first job begins at zero. *Eqs. (5) and (6)* guarantees no idle time between each job and its previous one. *Eq. (7)* ensures that none of the lines' jobs will have more than one position and will not be placed on more than one line (only has one position on all lines). *Eq. (8)* ensures that none of the lines' jobs will be placed after more than one job. *Eq. (9)* ensures that none of the lines' jobs will be placed before more than one job. *Eq. (10)* ensures that no more than one job will be identified as the first on any of the lines. *Eqs. (11) and (12)* define the decision variables U_k and L_k that to include these variables in the linear programming problem, *Eqs. (11) and (12)* have been added to the model. *Eqs. (13) and (14)* indicate the types of decision variables.

The number of decision variables in this model equals to: $\{(i,j) (j+1) + 2k + j\}$. The number of constraints in this model equals:

$$\{(i,j) (2j+1) + i(j-1) + 4j + i\} \text{ or } \{i \times (j-1) (2j+1) + 3(i,j) + 4j + i\}.$$

4.2 | Model 2

While both models give the same solutions ultimately, their mechanisms are different in determining the optimal solution. The first model specifies that each task should be performed after which task and on which line (without specifying the position number of that task), while the second model only specifies the position number of each task on each of the lines (without specifying the tasks before and after it).

4.2.1 | Indices used in the second model

i : Line (machine) index, $i = 1, 2, \dots, n$.

j : Job index, $j = 1, 2, \dots, m$.

f : Position number index, $f = 1, 2, \dots, m$.

k : Package index, $k = 1, 2, \dots, o$.

T_{ij} : The processing time of job j on line i .

M : A large positive number.

W_k : The importance (weight) of package k .

F_k : The set of jobs, related to package k .

4.2.2 | Decision variables

In this linear integer model, five types of decision variables are used:

U_k : Completion time of the last product of package k which gets prepared, $U_k = \max \{C_j / j \in F_k\}$.

L_k : Completion time of the first product of package k which gets prepared, $L_k = \min \{C_j / j \in F_k\}$.

Y_{ij} : 1, if job j gets done on line i , otherwise 0.

X_{ij}^f : 1 if job j gets done as the f^{th} job on line i , otherwise 0.

C_j : Completion time of job j .

Y_{ij} and X_{ij}^f are zero-one variables; U_k , L_k , and C_j are non-negative ones.

4.2.3 | Second model

The proposed linear model is as follows:

$$\text{Min } Z = \sum_{k=1}^O W_k (U_k - L_k), \quad (15)$$

s. t.

$$\sum_{i=1}^n Y_{ij} = 1 \quad \text{for all } j, \quad (16)$$

$$C_j \leq T_{ij} + M(1 - X_{ij}^1) \quad \text{for all } i, j, \quad (17)$$

$$C_j \geq T_{ij} - M(1 - X_{ij}^1) \quad \text{for all } i, j, \quad (18)$$

$$C_j \leq C_{j'} + T_{ij} + M(1 - X_{ij}^f) + M(1 - X_{ij'}^{f-1}),$$

for all $i, j, j', f, j \neq j', f \neq 1,$ (19)

$$C_j \geq C_{j'} + T_{ij} - M(1 - X_{ij}^f) - M(1 - X_{ij'}^{f-1}),$$

for all $i, j, j', f, j \neq j', f \neq 1,$ (20)

$$\sum_{i=1}^n \sum_{f=1}^m X_{ij}^f = 1 \quad \text{for all } j, \quad (21)$$

$$\sum_{j=1}^m X_{ij}^f \leq 1 \quad \text{for all } i, f, \quad (22)$$

$$\sum_{f=1}^m X_{ij}^f = Y_{ij} \quad \text{for all } i, j, \quad (23)$$

$$\sum_{j=1}^m X_{ij}^f \leq \sum_{j'=1}^m X_{ij'}^{f-1} \quad \text{for all } i, f \neq 1, \quad (24)$$

$$L_k \leq C_j \quad \text{for all } k, \quad j \in F_k, \quad (25)$$

$$U_k \geq C_j \quad \text{for all } k, \quad j \in F_k, \quad (26)$$

$$U_k, L_k, C_j \geq 0 \quad \text{for all } k, j, \quad (27)$$

$$Y_{ij}, X_{ij}^f = 0, 1 \quad \text{for all } i, j, f. \quad (28)$$

Eq. (15) represents the objective function of the problem, which is the same as the objective function of the first model. Eq. (16) ensures that each job is only performed on a single line. Eqs. (17)-(20) indicate the completion times of different items and guarantee that idle time does not occur on the lines. Eqs. (17) and (18) guarantees that each line's first job begins at zero. Eqs. (19) and (20) guarantees no idle time between each job and its previous one. Eq. (21) ensures that none of the lines' jobs will have more than one position and will not be placed on more than one line (only has one position on all lines). Eq. (22) ensures that none of the lines' positions (f indices) will be assigned to more than one job. Eq. (23) ensures that if one job is done on one line, it has precisely one position on that line. Eq. (24) ensures that no position is assigned on any lines until its preceding position is assigned. Eqs. (25) and (26) define the decision variables U_k and L_k that to include these variables in the linear programming problem, Eqs. (25) and (26) have been added to the model. Eqs. (27) and (28) indicate the types of decision variables.

The number of decision variables in this model equals to: $(i, j) (f + 1) + 2k + j$. The number of constraints in this model equals to: $(i, j) (2 (j-1). (f-1) + 3) + i (2f-1) + 4j$.

5 | Genetic Algorithm

Genetic algorithm is a global search technique based on natural genetic concepts and is one of the most widely used meta-heuristic approaches. We try to solve the stated problem utilizing the GA.

15

5.1 | Answer Representation

To represent the chromosomes, we use the structure presented in *Fig. 1*. Each chromosome contains $i + j - 1$ genes that the numbers 1, 2, ..., j are the jobs indicators, and $i - 1$ numbers from $j + 1$ onward, placed in the genes, each one indicates a change in producing line. An example of a sequence of genes in a chromosome on two lines is shown in the following figure:

Machine 1: 1, 2, 3, 4								
Machine 2: 5, 6, 7, 8								
8	7	6	5	<u>9</u>	4	3	2	1

Fig. 1. Chromosomes representation.

5.2 | Generation of an initial population

To get faster access to the optimal answer, a condition was considered for selecting the initial population:

First, one package is selected randomly, then the jobs related to that package are chosen randomly and randomly assigned to one of the lines. After completing the jobs of that package, the following package is selected randomly, and this process is repeated until all jobs are assigned to the lines. Each chromosome is compared to all the previous chromosomes and will not be selected if it is duplicated.

For example, if there are 12 jobs in 4 packages and 2 production lines, some of the permitted combinations in the initial population are shown in *Fig. 2*.

i_1				j_{11}	j_{10}	j_3		j_6	j_5
i_2	j_9	j_8	j_{12}			j_2	j_1	j_7	j_4

i_1	j_1	j_2	j_4	j_5	j_8	j_{10}	j_{11}		
i_2	j_3		j_6	j_7	j_9	j_{12}			

i_1	j_5	j_6	j_8	j_{10}	j_{11}				
i_2	j_4	j_7	j_9	j_{12}		j_3	j_1	j_2	

i_1	j_{10}		j_2		j_8	j_9	j_4	j_7	j_6
i_2	j_{12}	j_{11}	j_3	j_1			j_5		

Fig. 2. Some permitted combinations in the initial population.

5.3 | GA Operators

Selection operator

For selecting parents, we use the roulette wheel method so that by applying a coefficient (selection pressure) for each chromosome, a modified fitness function will be defined, and based on that, chromosomes will be ranked. Then, each chromosome will be assigned a probability (a fraction of the accumulation probability of the chromosomes). Finally, two non-same chromosomes will be selected randomly as the parents.

Crossover operator

The generation of offspring is performed by a particular single-point crossover operator designed below.

First, a random number is generated over the interval (the length of the chromosome), then as many as the generated random number, the genes of the first parent are repeated in the first offspring. Its remaining genes are arranged in the following order of placement in the second parent chromosome. Contrary to the above procedure will be done for generating the second offspring.

Mutation operator

Two genes of the parent's chromosome are randomly selected and replaced, called mutants population.

5.4 | Formation of the Next Generation

In order to resume the algorithm to the next generation, 30% of the elites of the previous generation will be transferred to the next generation. Then, 40% of the offspring produced by the intersection operator, 5% of the mutant population, and the rest of the population of the previous generation will be considered, and up to 70% of the size of the population will be chosen randomly among them and transferred to the next generation.

5.5 | Stopping Rule

Two conditions for stopping the algorithm are considered. If the algorithm meets each one, then it will stop, and the results will be announced. These two conditions are:

- I. Producing 500 generations.
- II. No change in the best value obtained from the fitness function for 200 consecutive generations.

6 | Solving Numerical Examples

To determine the efficiency of the proposed solution method, we will compare three small and modest size numerical examples, and the obtained objective function value, as well as the solution time, will be compared for the two proposed models and the proposed algorithm. To solve the models, the software GAMS 24.8.5 (r61358) and to solve the genetic algorithm, MATLAB R2015 (8.5.0.197613), was used on the laptop with Intel® Core™ i3 processor and 1066 MHz DDR3 memory SDRAM.

$i = 1, 2; j = 1, 2, 3, 4, 5, 6; k = 1, 2, 3;$

$f(k)$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$k=1$	*	*				
$k=2$			*	*		
$k=3$					*	*

$W(1) = W(2) = W(3) = 1$

T_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$i=1$	3	5	5	4	4	5
$i=2$	3	4	6	2	5	6

Comparison of answers of example 1				
	Objective function	Lower bound	Gap (%)	Solution time (s)
First model	2	2	0	0.31
Second model	2	2	0	1.76
The proposed GA (average of 50 trials)	2	2	0	2.65

Fig. 3. Example 1.

$i = 1, 2; j = 1, 2, 3, 4, 5, 6, 7, 8; k = 1, 2, 3, 4;$

$f(k)$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$k=1$	*	*						
$k=2$			*	*				
$k=3$					*	*		
$k=4$							*	*

$W(1) = W(2) = W(3) = W(4) = 1$

T_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$i=1$	3	5	5	4	4	5	3	3
$i=2$	3	4	6	2	5	6	2	4

Comparison of answers of example 2				
	Objective function	Lower bound	Gap (%)	Solution time (s)
First model	3	3	0	17.91
Second model	3	3	0	90.54
The proposed GA (average of 10 trials)	3	3	0	3.07

Fig. 4. Example 2.

$i = 1, 2; j = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12; k = 1, 2, 3, 4;$

f(k)	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11	j=12
k=1	*	*	*									
k=2				*	*	*	*					
k=3								*	*			
k=4										*	*	*

$W(1) = W(2) = W(3) = W(4) = 1$

T_{ij}	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11	j=12
i=1	3	5	5	4	4	5	3	3	7	2	5	3
i=2	3	4	6	2	5	6	2	4	6	2	4	3

Comparison of answers of example 3				
	Objective function	Lower bound	Gap (%)	Solution time (s)
First model	8	8	0	14499.75
Second model	8	7	12.5	500000.79
The proposed GA (average of 10 trials)	8	8	0	7.37

Fig. 5. Example 3.

7 | Conclusion

In this paper, the production scheduling of multi-product manufacturing systems as one of a variety of production systems was investigated, and we tried to express such a problem in the form of a linear programming problem. For this purpose, two mathematical models were presented that, unlike the usual procedure in the literature, the objective function of both models is to reduce the sum of the time intervals between the various products of each package. Both models take into account the general state of n production lines and m products (jobs) and the relevant constraints on reality developed in the models. Besides, since it is impossible to solve either of the two proposed models for large-scale problems, precisely in a meaningful time, an efficient genetic algorithm was proposed to solve the problem logically.

Furthermore, a few simple examples were defined. The result is neither of the two models presented for large-scale problems can be solved precisely in a justifiable time. Then, the genetic algorithm was utilized to solve the problem in a reasonable time with acceptable accuracy. Comparison with the exact solutions shows the efficiency of the proposed algorithm.

References

- [1] Borisovsky, P., Ereemeev, A., & Kallrath, J. (2020). Multi-product continuous plant scheduling: combination of decomposition, genetic algorithm, and constructive heuristic. *International journal of production research*, 58(9), 2677-2695. <https://doi.org/10.1080/00207543.2019.1630764>

- [2] Li, X. (2021). Research on optimization of setup time for product in the multi-product production system. *Microprocessors and microsystems*, 80, 103618. <https://doi.org/10.1016/j.micpro.2020.103618>
- [3] Kallrath, J. (2002). Planning and scheduling in the process industry. *OR spectrum*, 24, 219-250.
- [4] Kim, J. G., Song, S., & Jeong, B. (2020). Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times. *International journal of production research*, 58(6), 1628-1643. <https://doi.org/10.1080/00207543.2019.1672900>
- [5] Zhu, Z., & Heady, R. B. (2000). Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers & industrial engineering*, 38(2), 297-305. [https://doi.org/10.1016/S0360-8352\(00\)00048-6](https://doi.org/10.1016/S0360-8352(00)00048-6)
- [6] Omar, M. K., & Teo, S. C. (2006). Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: an improved MIP approach. *Applied mathematics and computation*, 181(2), 1008-1017. <https://doi.org/10.1016/j.amc.2006.01.068>
- [7] Lee, C. H., Liao, C. J., & Chung, T. P. (2014). Scheduling with multi-attribute setup times on two identical parallel machines. *International journal of production economics*, 153, 130-138. <https://doi.org/10.1016/j.ijpe.2014.02.013>
- [8] Heydari, M., & Aazami, A. (2018). Minimizing the maximum tardiness and makespan criteria in a job shop scheduling problem with sequence dependent setup times. *Journal of industrial and systems engineering*, 11(2), 134-150. http://www.jise.ir/article_57040_59ef11a3c39c282089bba7bd9aa316bd.pdf
- [9] Wang, X., & Cheng, T. C. E. (2015). A heuristic for scheduling jobs on two identical parallel machines with a machine availability constraint. *International journal of production economics*, 161, 74-82. <https://doi.org/10.1016/j.ijpe.2014.12.001>
- [10] Liao, L. W., & Sheen, G. J. (2008). Parallel machine scheduling with machine availability and eligibility constraints. *European journal of operational research*, 184(2), 458-467. <https://doi.org/10.1016/j.ejor.2006.11.027>
- [11] Jia, Z. H., Huo, S. Y., Li, K., & Chen, H. P. (2020). Integrated scheduling on parallel batch processing machines with non-identical capacities. *Engineering optimization*, 52(4), 715-730. <https://doi.org/10.1080/0305215X.2019.1613388>
- [12] Tavakkoli-Moghaddam, R., Jolai, F., Khodadadeghan, Y., & Haghnevis, M. (2006). A mathematical model of a multi-criteria parallel machine scheduling problem: a genetic algorithm (research note). *International journal of engineering-transactions A: basics*, 19(1), 79-86. https://www.ije.ir/article_71620.html
- [13] Cheng, B., Yang, S., Hu, X., & Chen, B. (2012). Minimizing makespan and total completion time for parallel batch processing machines with non-identical job sizes. *Applied mathematical modelling*, 36(7), 3161-3167. <https://doi.org/10.1016/j.apm.2011.09.061>
- [14] Muter, İ. (2020). Exact algorithms to minimize makespan on single and parallel batch processing machines. *European journal of operational research*, 285(2), 470-483. <https://doi.org/10.1016/j.ejor.2020.01.065>
- [15] Mirmohseni, S. M., Nasserli, S. H., & Khaviari, M. H. (2017). A new fuzzy hybrid dynamic programming for scheduling weighted jobs on single machine. *Journal of applied research on industrial engineering*, 4(2), 97-115. DOI: 10.22105/jarie.2017.49214
- [16] Shabtay, D., Mosheiov, G., & Oron, D. (2022). Single machine scheduling with common assignable due date/due window to minimize total weighted early and late work. *European journal of operational research*, 303(1), 66-77. <https://doi.org/10.1016/j.ejor.2022.02.017>
- [17] Rafiei, A., Homayouni, S. M., & Shafiei Alavijeh, A. (2015). A mathematical model for the single machine scheduling considering sequence dependent setup costs and idle times. *Journal of applied research on industrial engineering*, 2(2), 77-85. http://www.journal-aprie.com/article_42683.html
- [18] Antoniadis, A., Garg, N., Kumar, G., & Kumar, N. (2020). Parallel machine scheduling to minimize energy consumption. *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 2758-2769). Society for Industrial and Applied Mathematics.
- [19] Módos, I., Šúcha, P., & Hanzálek, Z. (2017). Algorithms for robust production scheduling with energy consumption limits. *Computers & industrial engineering*, 112, 391-408. <https://doi.org/10.1016/j.cie.2017.08.011>
- [20] Aghelinejad, M., Ouazene, Y., & Yalaoui, A. (2018). Production scheduling optimisation with machine state and time-dependent energy costs. *International journal of production research*, 56(16), 5558-5575. <https://doi.org/10.1080/00207543.2017.1414969>
- [21] Anghinolfi, D., Paolucci, M., & Ronco, R. (2021). A bi-objective heuristic approach for green identical parallel machine scheduling. *European journal of operational research*, 289(2), 416-434. <https://doi.org/10.1016/j.ejor.2020.07.020>

- [22] Bhosale, K. C., & Pawar, P. J. (2020). Integrated production planning and scheduling for parallel production lines. *Advanced engineering optimization through intelligent techniques: select proceedings of AEOTIT 2018* (pp. 679-687). Springer Singapore. https://doi.org/10.1007/978-981-13-8196-6_59
- [23] Cui, W., Lu, Z., Li, C., & Han, X. (2018). A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops. *Computers & industrial engineering*, 115, 342-353. <https://doi.org/10.1016/j.cie.2017.11.020>
- [24] Chansombat, S., Pongcharoen, P., & Hicks, C. (2019). A mixed-integer linear programming model for integrated production and preventive maintenance scheduling in the capital goods industry. *International journal of production research*, 57(1), 61-82. <https://doi.org/10.1080/00207543.2018.1459923>
- [25] Devapriya, P., Ferrell, W., & Geismar, N. (2017). Integrated production and distribution scheduling with a perishable product. *European journal of operational research*, 259(3), 906-916. <https://doi.org/10.1016/j.ejor.2016.09.019>
- [26] Sifaleras, A., Karakalidis, A., & Nikolaidis, Y. (2022). Shift scheduling in multi-item production lines: a case study of a mineral water bottling company. *International journal of systems science: operations & logistics*, 9(1), 75-86. <https://doi.org/10.1080/23302674.2020.1818144>