

Research Article

Simulation Optimization for the Multihoist Scheduling Problem

Ricardo Pérez-Rodríguez 

CONACYT-UAQ, Autonomous University of Queretaro, Faculty of Engineering, Centro Universitario, Cerro de las Campanas s/n, 76010, Santiago de Queretaro, Mexico

Correspondence should be addressed to Ricardo Pérez-Rodríguez; dr.ricardo.perez.rodriguez@gmail.com

Received 6 October 2020; Revised 12 April 2021; Accepted 25 May 2021; Published 7 July 2021

Academic Editor: Mehdi Keshavarz-Ghorabae

Copyright © 2021 Ricardo Pérez-Rodríguez. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Although the Multihoist Scheduling Problem (MHSP) can be detailed as a job-shop configuration, the MHSP has additional constraints. Such constraints increase the difficulty and complexity of the schedule. Operation conditions in chemical processes are certainly different from other types of processes. Therefore, in order to model the real-world environment on a chemical production process, a simulation model is built and it emulates the feasibility requirements of such a production system. The results of the model, i.e., the makespan and the workload of the most loaded tank, are necessary for providing insights about which schedule on the shop floor should be implemented. A new biobjective optimization method is proposed, and it uses the results mentioned above in order to build new scenarios for the MHSP and to solve the aforementioned conflicting objectives. Various numerical experiments are shown to illustrate the performance of this new experimental technique, i.e., the simulation optimization approach. Based on the results, the proposed scheme tackles the inconvenience of the metaheuristics, i.e., lack of diversity of the solutions and poor ability of exploitation. In addition, the optimization approach is able to identify the best solutions by a distance-based ranking model and the solutions located in the first Pareto-front layer contributes to improve the search process of the aforementioned scheme, against other algorithms used in the comparison.

1. Introduction

Some chemical-production systems use tanks containing treatment baths in order to generate finished products. Those treatment baths can contain rinsing, acid, or electroplating solutions. This kind of production system is necessary when a product needs to be treated with a specific treatment bath to enhance its mechanical, electrical, or esthetic properties. The products are soaked in each tank according to a given sequence [1]. Commonly, these kinds of products are loaded on carriers or rack baskets.

As other manufacturing facilities, the handling material of the chemical-production systems is executed by track-mounted hoists, i.e., the rack baskets are moved from facility to facility by one or more hoists. The hoists can be (or not) on the same track. Therefore, the hoists are used as means of transport between tanks [2].

The hoists move rack baskets from facility to facility according to the production schedule. It means that the

hoists should be available, as an assumption, in order to not compromise the already defined production schedule. However, the hoists frequently operate on shared tracks. Therefore, the hoists also should be scheduled to avoid colliding with each other. Normally, the production schedule is considered as input data for the hoists' schedule. It means that these schedules are usually planned separately. It can be considered a risk to enhance the performance of the chemical-production system. In this case, any production schedule should consider the hoists' availability.

The simultaneous schedule of jobs and hoists assumes a central relevance for surface treatments. The treatment processes can be modeled as a job-shop production.

Consequently, a strong need for enhanced hoist scheduling systems emerged, representing a key enabler to maximize productivity and product quality [3].

A classic job-shop configuration can be detailed based on the aforementioned chemical processes [4]. Let a set of n jobs (products) $J = \{J_1, J_2, \dots, J_n\}$ on a set of m machines (tanks)

$M = \{M_1, M_2, \dots, M_m\}$. Each job J_i requires a series of p_i operations (treatments) $O = \{O_{i1}, O_{i2}, \dots, O_{ip_i}\}$ with precedence constraints. As any job-shop configuration, each machine can process only one job at one time without preemption, and each job can be processed on one and only one machine at one time. However, compared with the classical job-shop configuration, the multihoist scheduling problem (MHSP) includes additional constraints. Such constraints increase the difficulty and complexity of the schedule. Constraints are outlined below:

- (i) The jobs require a hoist in order to move them between facilities.
- (ii) Bounded soaking times, i.e., each operation is bounded within a time window $a \leq O_{ip_i} \leq b$.
- (iii) Multifunction facilities, i.e., if a machine can perform several operations of a same job, it is a multifunction facility.
- (iv) Duplicated facilities, i.e., each operation O_{ip_i} can be executed on one machine chosen from a set of possible machines $M_{ij} \subseteq M$.
- (v) Although each operation is bounded within a time window, the processing time can vary between the set of possible machines.
- (vi) Hoists can move one rack basket at a time.
- (vii) Each rack basket contains only one job.
- (viii) Hoists can share the same track or crane. It could be a physical constraint between movements.
- (ix) No-storage and no-wait are allowed. Therefore, buffers are prohibited in the MHSP.

The limitations mentioned above make a difference between the classical job-shop configurations and the MHSP. It is a strongly constrained problem, known as an NP-hard one [5]. The problem is indeed NP-hard, and each extension involves an increase of the complexity of both the model and its resolution. This is the reason why we have decided to work on a new approach.

In addition, some industries such as pharmaceutical, chemical and plastic have similar operational conditions, i.e., no-wait in the processes. Therefore, building schedules with the no-wait component should be more suitable for modeling in those industries, and also other problems such as train scheduling, aircraft landing scheduling, surgery scheduling, among others [6].

Based on the previous constraints, the assumption on identical jobs, in the simultaneous schedule of jobs and hoists, should be omitted. In addition, this research does not have the objective to find a repetitive hoist scheduling. There exist multiple job types and diverse hoists on the chemical-treatment processes. Moreover, this research does not have as objective to preassign to each hoist a precise set of tanks. Although this simplification tackles the collision constraints, it restricts the sequence of treatment of the jobs to follow the layout of the tanks. If the requirements are low-demand, the utilization rate of the tanks would decrease.

In order to identify the difficulty and complexity of the MSHP, an example is provided below.

Assume a schedule with four jobs $J = \{J_1, J_2, J_3, J_4\}$. Every job J_i is formed by a sequence of three operations $\{O_{i,1}, O_{i,2}, O_{i,3}\}$ performed one after another. Table 1 details the alternative machines for every operation and additional data.

Let a sequence be $J_1 < J_2 < J_3$ with O_{11} in M_1 , O_{21} in M_4 , and O_{31} in M_2 . Assume the minimum processing time window for all the operations. Figure 1 details the sequence step by step by the hoist. Let the next set of operations be O_{12} in M_2 , O_{22} in M_4 , and O_{32} in M_1 . According to the previous sequence, the hoist has to break free M_2 with O_{31} , and break free M_1 with O_{11} . However, unlike a normal job-shop configuration, a blocking occurs in the MHSP. The simultaneous schedule of jobs and hoists is a necessity to avoid blocking through the treatment processes. To find a suitable sequence and appropriate assignment of the operations to the machines in order to avoid potential blockings and no-waits in the processes is the problem statement.

In order to simultaneously schedule jobs and hoists, this study proposes a solution for the MHSP. This paper details how a solution can be built by three decisions, i.e., operation scheduling, tank assignment decision, and hoist assignment. In particular, let us discuss vectors, such as permutations, and these vectors can represent the processing sequence of operations.

In this research, the processing sequence of operations is considered as a permutation. Table 2 shows a vector (ranking) of five operations as an example.

In this sense, the members of the population of the proposed algorithm are permutations of elements. Table 3 depicts some permutations, where each element represents an operation, and the processing sequence of the operations is executed according to each permutation.

As previous research, this research also tries to use the space of permutations as solutions. The proposal of this research is to use a probability distribution based on permutations.

For the aforementioned proposal, an Estimation of Distribution Algorithm (EDA) is considered for the MHSP. According to the literature review, the EDA has been scarcely studied for solving the MHSP. EDA is an experimental technique that belongs to the evolutionary computation field. Instead of using traditional evolutionary operators such as crossing and mutation, the EDA produces solutions through a probability model based on the previous solutions. The probability model mentioned above is built with statistical information, i.e., based on the search experience. The EDA makes use of the aforementioned probability model to describe the distribution of the solution space. In this research, the distribution of the permutations, for the MHSP, is the solution space. However, there is no probability model that can produce a feasible solution on a permutation-based representation [7]. Any EDA needs to be reconfigured to tackle permutation-based problems. Then, as hypothesis, using a specific probability model for this issue should be effective against other models.

TABLE 1: An MHSP example.

Operations job, precedence	Processing time window (min) machines			
	M_1	M_2	M_3	M_4
O_{11}	20–30	20–40	10–30	10–50
O_{12}	30–40	10–40	—	20–30
O_{13}	30–40	—	—	—
O_{21}	20–30	—	10–30	10–30
O_{22}	—	30–50	50–60	10–20
O_{23}	10–20	—	—	—
O_{31}	—	10–20	10–30	20–30
O_{32}	10–20	—	—	—
O_{33}	—	20–30	20–30	—
Tracks		1		
Hoists		1		
Transfer time		1 (between locations)		
Cross time		1 (for any machine)		

In order to produce feasible solutions from the proposed EDA, this study uses the Mallows model. Mallows [8] establishes the Mallows exponential model. The Mallows model incorporates to each permutation σ , i.e., each solution, a probability. In order to compute the probability mentioned above, the model uses the distance concept between σ and a central permutation σ_0 . A distance metric is computed between permutations, by algebra of permutations. The aforementioned probability will be smaller if the distance between σ and σ_0 is bigger. Figure 2 details an example of such exponential distribution, with five vectors.

Then, the Mallows model is considered as the specific probability model for the distribution of the permutations. The search process is based on the exponential model, and any traditional evolutionary operator is omitted. Based on Figure 2, the exponential shape of the model is built using the distance between all the permutations σ_n of the solution space and the central permutation σ_0 . Traditionally, the central permutation is estimated or selected between all the members of the population. The best solution, during the evolutionary progress, can be considered as the central permutation. However, the best solution is not a guarantee to be the best estimation. Therefore, finding the best estimation should be reached in order to get the best scheduling solution for the MHSP.

Once the central permutation is estimated or selected, generating new offspring, using the exponential model, is the main objective. Then, the main idea is to use the probability between permutations to generate a new offspring. However, the Mallows model is not able to generate offspring by itself. The reason is because there can be many permutations with the same distance to the central permutation. It creates confusion when choosing the offspring.

It is solved through the decomposition of the distance between each permutation and the central permutation. The Generalized Mallows Distribution (GMD) process, detailed in Fligner and Verducci [9], and Fligner and Verducci [10], explains the procedure to get the decomposition of the distance between permutations and thereby generate a new offspring. With this strategy, the inconvenience of probability models in permutation-based

problems is solved. The GMD process permits to produce feasible solutions.

Based on the concept of properly modeling the main variables that intervene in the performance of the process has been a priority in the solution of real-world scheduling problems [11]. Such variables or characteristics could be inside or outside of the shop floor and these should be incorporated to efficiently solve the scheduling problem. Therefore, this research appends the real environment on the chemical production process mentioned above. Moreover, if the real environment is considered, theoretical assumptions are not necessary to incorporate in the MHSP. Then, a simulation model is built, and it emulates the aforementioned chemical production system. Different resources, in the chemical production process, are considered in the simulation model mentioned above. Those resources perform all the operations required by any job that is scheduled. The aforementioned simulation model, built on Delmia-Quest® platform, included vast details that the chemical production process presents, i.e., the process itself.

The results of the simulation model, i.e., the makespan and the workload of the most loaded machine, are necessary for providing insights about which schedule on the shop floor should be implemented. The proposed optimization method uses the results mentioned above in order to build new scenarios for the MHSP and to solve the aforementioned conflicting objectives.

The research motivation is to show how the exponential distribution between permutations helps to reduce the deficiencies of the EDA. In addition, it is preferable that the methods used should contain some well-defined math expressions. It helps to understand how the solutions are generated. Therefore, the research motivation is to use new methods to explicitly establish the search process of new solutions. The main reason, for doing this research, was to determine what is most useful, i.e., the recent algorithms or the use of exponential distributions to obtain the same or better solutions.

Although diverse methods and strategies have been used to solve the MHSP, this paper contributes to the state of the art as follows:

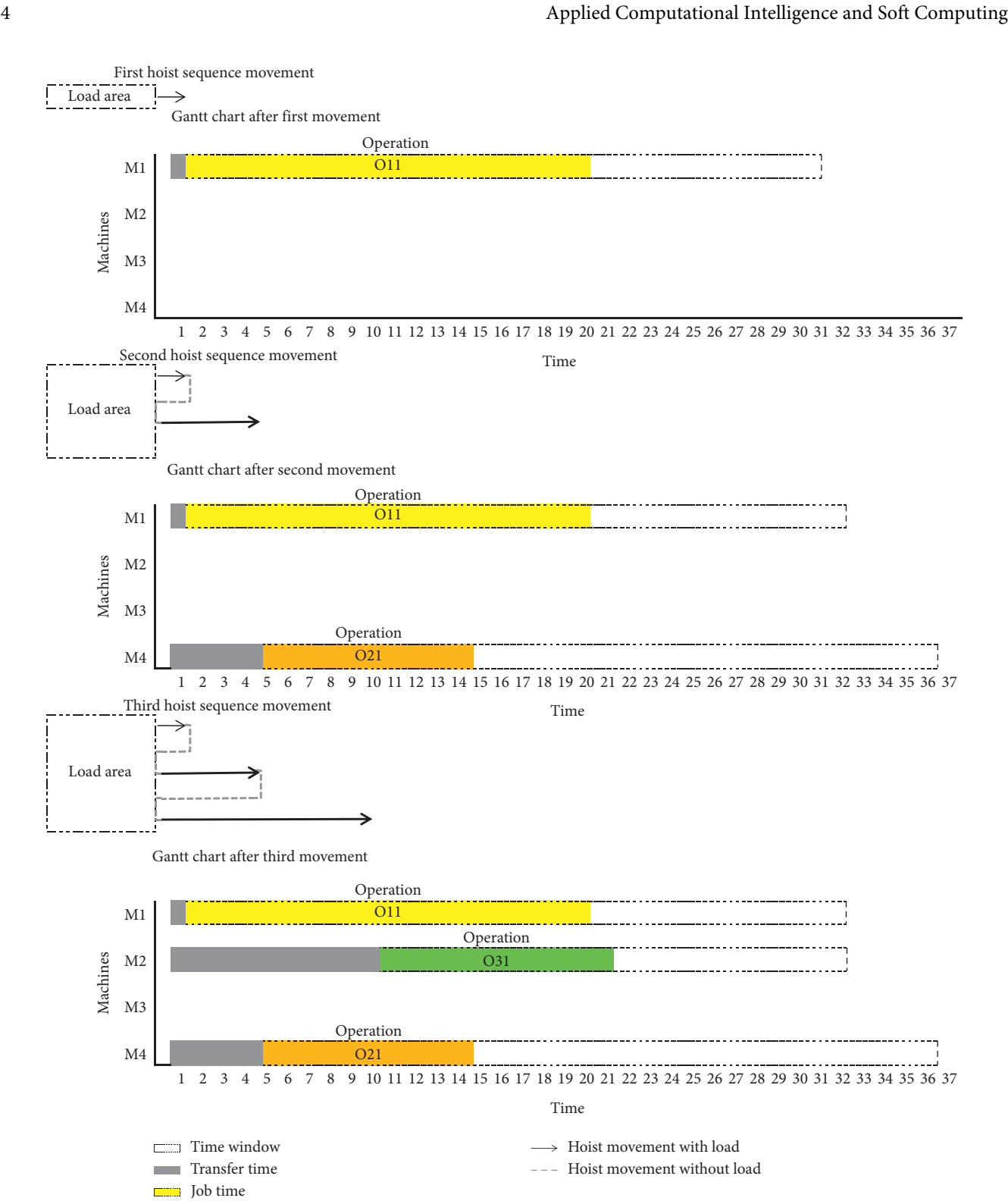


TABLE 3: Members of the population as permutations.

Processing sequence	1st	2nd	3rd	4th	5th
Vector 1	O_5	O_3	O_1	O_4	O_2
Vector 2	O_3	O_5	O_4	O_1	O_2
Vector 3	O_4	O_2	O_1	O_3	O_5

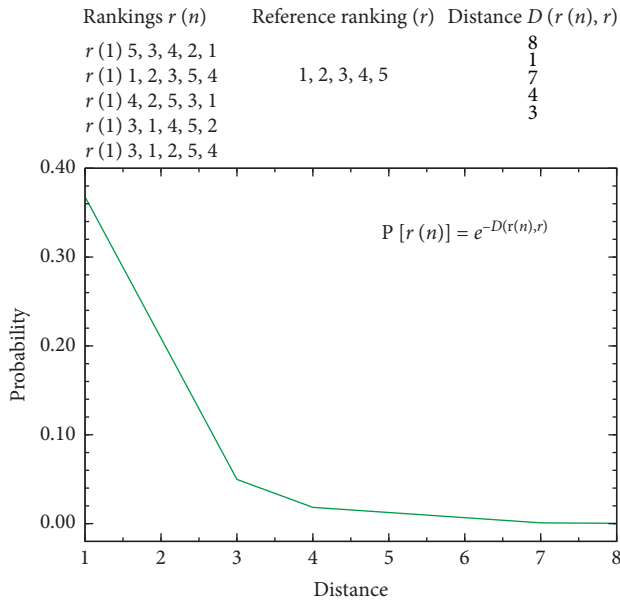


FIGURE 2: Exponential distribution using a distance between permutations.

- (i) To introduce the Mallows model, for the MHSP, to reduce the deficiencies of the EDA, i.e., lack of diversity of the solutions and poor ability of exploitation.
- (ii) To provide all details to help understand how the new solutions are generated using well-defined math expressions in the proposed algorithm.
- (iii) To utilize the GMD for the MHSP as an explicit probability distribution to enhance the performance of the EDA.
- (iv) To couple the GMD with the EDA, called MHEDA (Multihoist Estimation of Distribution Algorithm), on a bi-objective approach.
- (v) To construct scenarios over N jobs and M workstations in a flexible chemical production process through simulation and the MHEDA. Simulation model emulates the facility being studied, while MHEDA is used to get the best solution.

2. Related Work

2.1. Mathematical Programming. Scheduling the movements of a hoist in electroplating facilities is tackled by Varnier et al. [1]. Based on their research, in most works, for the mono-product case, the layout of the shop is generally considered as a fixed data. However, the results of their

research show that the layout should be considered in order to improve the productivity of the shop. The authors propose a combination of scheduling and layout design.

Other works, such as Grunder et al. [12], try to identify the interaction between the layout of the shop and the productivity of a treatment line. The authors show that a particular type of configuration of the shop can be maximizing the performance of the line. However, it is not functional for all the studied cases. In addition, the authors propose a branch and bound algorithm to process the optimal layout of a saturated single-hoist production line.

Other types of interaction can be found in Subai et al. [13], where environmental constraints are incorporated in the schedule of a treatment surface process. The authors focus to maximize the shop throughput by means of mathematical models. Such models include a nonlinear global cost function where the environmental cost plays a key role.

A mixed-integer linear programming model for a multiproduct batch plant with nonidentical parallel processing machines is used in Berber et al.'s [14] research. The authors show how to minimize the total production time without using heuristic rules. Diverse numerical instances and one industrial problem are used to test their proposed model. The aforementioned model is able to produce better solutions for the industrial example considered.

An exact solving method is proposed by El Amraoui et al. [15]. The mentioned method is a linear optimization approach. This work is implemented to optimize the cycle length and the throughput rate in a cyclic hoist scheduling problem where there are r different part-jobs in an electroplating line.

Other mixed-integer linear programming model for multirecipe and multistage material handling processes is developed in Zhao et al. [16]. In this research, the authors simultaneously consider the production line arrangement, and the customized production ratio. Various case studies are used to demonstrate the efficacy of this methodology.

For the two-hoist cyclic scheduling problem, Chtourou and Manier [17] propose a mixed integer linear programming model. The authors consider avoiding collision between the hoists which share a common track whilst the cycle time is minimizing.

For the dynamic hoist scheduling problem with multi-capacity machines, Feng et al. [18] detail a mixed-integer programming model. The mentioned model considers jobs randomly arriving throughout the horizon as a dynamic issue.

In a copper production plant, Suominen et al. [19] present a nonlinear optimization and scheduling approach to maximize a smelting furnace production. The proposed approach is presented by simulating the evolution of the process over the optimization horizon.

For the ethylene cracking process with feedstocks and energy constraints, Su et al. [20] address a scheduling problem by a hybrid mixed-integer nonlinear programming formulation. The problem mentioned above has a special situation, i.e., the facilities require periodic cleanup to restore the performance. The authors make useful suggestions for real cracking process production by means of numerous examples. Therefore, the examples mentioned help illustrate the utility of the model.

In an ice cream shop, Wari and Zhu [21] present a mixed-integer linear programming model. The key point of this contribution is that a multiweek production scheduling is considered. Furthermore, the model mentioned above incorporates operational aspects to enhance the solution quality. The aforementioned model is tested and compared with heuristics methods. The results report that the model of this research is able to efficiently and effectively handle the multiweek aspect.

Qu et al. [22] study the effect of integrating scheduling with the optimal design of a two-dimensional production line to maximize production efficiency in the multirecipe and multistage material handling processes. Various case studies are used to demonstrate the efficacy of the integration mentioned above.

Finally, Jianguang et al. [23] consider the cyclic jobshop hoist scheduling with multicapacity reentrant tanks and time-window constraints. As the authors explain, jobs are processed in a series of tanks with a defined processing sequence of operations for all of the jobs. A mixed-integer linear programming model is developed by addressing the time-window constraints and tank capacity constraints.

2.2. Heuristics. For the cyclic hoist scheduling problem considering material as well as resource handling constraints, El Amraoui et al. [24] present a new heuristic where the time windows are maintained for all soaking operations and also overlapping cycles are allowed. As in almost any research, the authors compare the proposed heuristic mentioned above with other existing algorithms to prove its efficiency.

Another example of heuristics for the hoist scheduling problem can be found in Kujawski and Świątek [2], where the sequence of products is analyzed by changing the ordered items in electroplating production lines in order to reduce the makespan. The items are located in queues where the sequence is monitored.

For the online hoist scheduling problem, Kujawski and Świątek [25] prepare a set of production scenarios before the real-time system starts. A real-life shop located at Wrocław, Poland, is used to explain carefully the algorithm. In this research, the utilization ratio is considered in order to choose the best schedule.

For the flexible and nonlinear electrochemical processes, Beerbühl et al. [26] propose a heuristic by combining scheduling and capacity planning. The heuristic mentioned above is able to tackle nonconvex and mixed-integer problems, such as the electrolysis of water to produce hydrogen, reformulating these kind of problems into convex and continuous nonlinear problems.

Recently, Laajili et al. [27] detail an adapted variable neighborhood search-based algorithm for the cyclic multihoist design and scheduling problem. The Laajili et al. study considers identical jobs, and therefore identical processing sequence of operations for all the jobs. When identical jobs occur, it permits to create a cyclic multihoist schedule.

2.3. Metaheuristics. For the printed-circuit-board electroplating line, Lim [28] consider to determine the best throughput rate. The authors propose a genetic algorithm-based approach for the cyclic hoist scheduling problem. Through an experiment, the proposed algorithm is more efficient than the previous mathematical programming-based algorithms.

In an aluminum casting center, Gravel et al. [29] present an ant colony optimization metaheuristic. The representation of the solution in this research considers different objectives. In addition, the authors implement the proposed method, by introducing software, in the shop casting center.

For the mixed batch and continuous processes, Wang et al. [30] develop a differential evolution algorithm. The solution representation considers capacity constraints. A key characteristic, in the proposed algorithm, is how to compute the crossover probability by using the logistic chaotic map method.

For the single hoist cyclic scheduling problem, El Amraoui et al. [31] consider hard resources and time-window constraints in their proposed genetic algorithm.

For the multihoist scheduling problem with transportation constraints, Zhang et al. [4] study how to reduce the makespan. The global limitation is that there are no buffers to storage work in process. Then, a right assignment is critical in the proposed solution. The authors detail a modified genetic algorithm to tackle the problem mentioned above. The aforementioned algorithm also uses a modified shifting bottleneck procedure, in order to build a feasible schedule. The results of this research show that the proposed algorithm is able to handle several transport resources.

For other electroplating shops, El Amraoui et al. [32] examine the processing time which is confined within a time window. A genetic algorithm approach is presented to solve the multijobs cyclic hoist scheduling problem with a single transportation resource.

For a single robot and flexible processing times in a robotic flow shop, Lei et al. [33] aim to increase the throughput rate. A hybrid algorithm based on the quantum-inspired evolutionary algorithm and genetic operators is presented for solving the cyclic scheduling problem. The algorithm integrates three different decoding strategies to convert quantum individuals into robot move sequences. Besides, crossover and mutation operators with adaptive probabilities are used to increase the population diversity. A repairing procedure is proposed to deal with infeasible individuals. Comparison results on both benchmark and randomly generated instances demonstrate that the proposed algorithm is more effective in solving the studied problem in terms of solution quality and computational time.

2.4. Hybrid Approaches. In a surface treatment system with time window constraints, Chové et al. [34] investigate how to improve throughput rate without loss of treatment quality. The authors propose a new combined approach based on both predictive scheduling and reactive scheduling in an industrial case where only one hoist is used.

Another example of hybrid approach can be found in El Amraoui and Nait-Sidi-Moh [35]. The authors use P-Temporal Petri Net models to describe the behavior of different scenarios of the shop in a specific cyclic hoist scheduling problem. The authors propose a linear programming model to determine the optimal plan, where the exact beginning and ending instants of each task is the output of the model. In addition, a simulation tool validates the hybrid method mentioned above.

For a multicrane scheduling problem, where a set of coils should be transported from a storage location to another side, Xie et al. [36] formulate it as a mixed-integer linear programming model and propose a heuristic algorithm to solve the problem.

Most publications belong to this category, i.e., hybrid approaches, such as the Yan et al.'s [37] research. The authors focus on a bi-objective, i.e., the cycle time and the material handling cost over a cyclic hoist scheduling problem where the parts have different flow patterns. A bi-objective linear programming model is detailed. In addition, a Pareto-front is formulated in this research with respect to the bi-criteria mentioned above. A hybrid discrete differential evolution algorithm computes the Pareto-front. Moreover, the work-in-process level is used to adjust the exploration and exploitation of the search of the best solution.

Another bi-objective case is found in El Amraoui and Elhafi [38]; where improving the productivity and quality are studied. Firstly, the authors formulate the problem as a mixed integer linear programming model. In addition, the authors detail an efficient heuristic procedure to obtain the movements of the hoist. The results of the aforementioned heuristic are compared to a lower bound obtained from the model mentioned above and to the best available heuristic in the literature.

For the aerospace and electroplating industries, Basán and Méndez [39] present a hybrid approach using mixed-integer linear programming, heuristics, and a simulation model. The mentioned simulation model contains multiple particularities in a multiproduct multistage production system where a single hoist is analyzed. Real data, of an aircraft manufacturing industry, are used to minimize the operating cost and maximize the productivity of the system.

Another real case is found in Mori and Mahalec [40]. The authors deal with scheduling of the continuous casting of steelmaking. As the authors explain, a mixed-integer linear programming model computationally is intractable. Then, the authors produce a production planning by solving a relaxed mixed-integer linear model at the first stage. After that, the authors build schedules by simulated annealing and a shuffled frog-leaping algorithm. As other previous studies, real data are utilized to test the proposed method.

For the scheduling of multicrane operations in an iron and steel enterprise, Xie et al. [41] study how to reduce the

makespan, by a mixed-integer linear programming model and a heuristic. The authors identify properties to avoid crane conflicts.

As another study, in a case example of the chemical industry, Hahn and Brandenburg [42] present a linear programming model and an aggregate stochastic queuing network model in order to obtain the best solution. The proposed hybrid approach considers the production-related carbon emission and overtime working hours in the solution.

For a steelmaking-continuous casting manufacturing system, Jiang et al. [43] develop a multiobjective soft scheduling to address the uncertain scheduling problem. Three objectives are tackled, i.e., waiting time, cast-break, and over-waiting. The authors detail a preference-inspired chemical reaction optimization algorithm, and a simulation-based *t*-test method is used to provide feedback of the solutions. The convergence is handled by a knowledge-based local search embedded to the aforementioned optimization algorithm. Real-world steelmaking-continuous casting instances served as the input parameter to work with the mentioned approach.

Finally, in flexible flow shops, such as the paper mill industry, Zeng et al. [44] construct a multiobject optimization model, where three objectives are analyzed, i.e., makespan, electricity consumption, and material wastage are included in the mentioned model. At the beginning, only two objectives are considered in the solution, i.e., electricity consumption and material wastage. After that, a hybrid nondominated sorting genetic algorithm II method is employed to solve all the objectives. Real-world case study is used in this research.

Other hybrid approaches can found in An et al. [45]; where the authors used simulation and cloud computing, to prevent collision detection, and hoisting path planning, in a three-dimensional hoisting system.

Li et al. [46] present a simulation-based solution for a multicrane-scheduling problem derived from a steelmaking shop. The problem is modeled considering different objectives for the jobs and workload objective for the cranes. The hybrid approach solves the problem by a heuristic.

Tamaki et al. [47] propose a simulation-based solution by adopting the metaheuristic methods to solve the crane-scheduling problem in manufacturing systems of the job-shop type where semi-products are picked up and delivered by using cranes between the facilities.

Zhang and Oliver [48] include the crane-scheduling problem into the production scheduling environment and combine them together to obtain an integrated schedule. A simulation-based optimization solves this integrated scheduling problem. A genetic algorithm is introduced to determine the allocation of machines and cranes. A simulation model referring to a queuing network is used to evaluate the crane and machine allocation results and provides the fitness value for the genetic algorithm.

Diverse gaps in the current state of the art can be noticed based on the review mentioned above. The main steps in the proposed methods contain greedy procedures to get promising solutions. Therefore, the performance of the

aforementioned approaches is related to those mentioned procedures. As an example, the genetic algorithms build new solutions by evolutionary operators; however, those operators do not permit to have control in characterizing the solution space explicitly. Then, in this research, an explicit probability distribution over the MHSP is offered to characterize the solution space explicitly. In addition, there are currently no publications that use an EDA for the MHSP.

It is clear, from the exposed review, that the EDAs have still a gap to improve their performance, and the exponential distributions approach might be a useful way to improve the EDAs. In addition, from the exposed review, it is of interest to determine how much better EDAs can be of the recent algorithms.

Finally, Table 4 depicts the state of the art on the MHSP with other related problems.

3. Problem Statement

The multihoist scheduling problem shares features and characteristics of a flexible jobshop configuration. Wang et al. [49] and Yan and Wang [50] explain the problem formulation for this configuration. The main constraints are detailed below:

- (i) For each job, the corresponding operations have to be processed in the given order, that is, the starting time for an operation must not be earlier than the point at which the preceding operation in the sequence of operations of the respective job is completed
- (ii) Moreover, each operation has to be assigned to exactly one tank
- (iii) Preemption is not allowed, i.e., each operation must be completed without interruption once it starts
- (iv) The operations assigned for each tank have to be subsequently established, that is, an operation is only allowed to be assigned to the sequence of a tank if the preceding position on the sequence is already established
- (v) If the operations i and j are assigned to the same tank k for consecutive positions $p-1$ and p , then the starting time of operation j must not be earlier than the completion time of operation i in order to prevent overlapping

Additional constraints involved by the MHSP are the following:

- (i) The processing time in each tank must respect a minimum and maximum limit, i.e., each processing time is bounded, and those limits must be strictly respected to ensure the quality of the products. Furthermore, the processing time window is different, for each i operation, in each tank of a set of M_i given tanks $M_i \subseteq M$. Let $L(O_{i,k})$: minimum processing time for the operation i in tank k . $U(O_{i,k})$: maximum processing time for the operation i in tank

k . $t_{i,k}$: the processing time of the operation i on tank M_i . Then, $L(O_{i,k}) \leq t_{i,k} \leq U(O_{i,k})$ for all $i \in O$, $M_i \in M$.

- (ii) A hoist can perform only one transport operation at a time, and a hoist must have enough time to move between two transport operations.

The mathematical model is based on Pérez-Rodríguez et al. [11] detailed below.

Let $J(i)$ denote the job to which operation i belongs, and let $P(i)$ be the position of operation i in the sequence of operations belonging to job $J(i)$ starting with one, i.e., $P(i) = 1$ if the operation i is the first operation of a job. Furthermore, the index set I_k defined by $I_k = \{i \in O | k \in M_i\}$ denotes the indices of operations $i \in O$ that can be processed on tank k . Consequently, there are $|I_k|$ positions on tank k .

In order to model the assignment of operations to tanks, assignment binary variables $x_{i,k,p}$ for all $p_k = 1, \dots, |I_k|$, $k = 1, \dots, m$, $i \in O$ are introduced if $x_{i,k,p} = 1$ means that the operation i is scheduled for position p on tank k . The processing time of the operation i on tank k is denoted by $t_{i,k}$. Furthermore, S_i is defined as the starting time for operation i . For each job, the corresponding operations have to be processed in the given order, that is, the starting time for an operation must not be earlier than the point at which the preceding operation in the sequence of operations of the respective job is completed. This constraint is imposed simultaneously on all appropriate pairs of operations, aggregated in the set of conjunctions C given by $C = \{(i, j) | i, j \in O: J(i) = J(j) \wedge P(j) = P(i) + 1\}$. Consequently, the precedence constraints are given by

$$S_i + \sum_{k \in M_i} \sum_{p=1}^{|I_k|} x_{i,k,p} t_{i,k} \leq S_j, \quad \text{for all } (i, j) \in C. \quad (1)$$

Moreover, each operation has to be assigned to exactly one position, which is ensured by

$$\sum_{k=1}^m \sum_{p=1}^{|I_k|} x_{i,k,p} = 1, \quad \text{for all } i \in O. \quad (2)$$

In addition, only one operation can be assigned to each position, due to constraints

$$\sum_{i \in O} x_{i,k,p} \leq 1, \quad \text{for all } p = 1, \dots, |I_k|, k = 1, \dots, m. \quad (3)$$

The positions on each tank have to be subsequently filled, that is, an operation is only allowed to be assigned to a position on a tank if the preceding position is already filled. This condition is ensured by

$$\sum_{i \in O} x_{i,k,p} \leq \sum_{i \in O} x_{i,k,p-1}, \quad \text{for all } p = 2, \dots, |I_k|, k = 1, \dots, m. \quad (4)$$

TABLE 4: Literature review of the MHSP.

Research	Process(es)	Objective(s)	Variable(s)	Approach
MHEDA	General treatment processes	Makespan, and the workload of the most loaded tank	Sequence	Simulation optimization approach, through an estimation of distribution algorithm
Zeng et al. [44]	Paper mill factory	Makespan, electricity consumption, and material wastage	Sequence	Hybrid-mathematical model with genetic algorithm
Jiang et al. [43]	Steelmaking-continuous casting	Waiting time, cast-break, and over-waiting	Soft-form schedule	Hybrid-preference-inspired chemical reaction optimization algorithm with knowledge-based local search
Hahn and Brandenburg [42]	Chemical industry	Cost minimization	Carbon emission and overtime	Hybrid-linear programming with aggregate stochastic queuing network
Xie et al. [41]	Iron and steel processes	Makespan	Unloading, no delays and sequence	Hybrid-mixed-integer linear programming with heuristic
Lei et al. [33]	Robotic manufacturing system	Throughput rate	Sequence	Metaheuristics-Quantum algorithm
Mori and Mahalec [40]	Continuous casting of steelmaking	Higher productivity	Sequence	Hybrid-Relaxed mixed-integer linear model with simulation annealing and a shuffled frog-leaping algorithm
Qu et al. [22]	Electrochemical processes	Higher productivity	Spatial locations and optimal design	Mathematical programming-mixed-integer linear programming
Basán and Méndez [39]	Aerospace and electroplating	Higher productivity	Time-window, zero-wait storage, and sequence-dependent transferring time	Hybrid-mixed-integer linear programming with heuristics and simulation
El Amraoui and Elhafsi [38]	Electrochemical processes	Higher productivity and better product quality	Sequence	Hybrid-mixed-integer programming model with heuristic
Yan et al. [37]	Electrochemical processes	Cycle time and material handling cost	Working process level	Hybrid-Bi-objective linear programming with hybrid discrete differential algorithm
Wari and Zhu [21]	Ice cream process	Performance	Multi week production scheduling	Mathematical programming-mixed-integer linear programming
Su et al. [20]	Ethylene cracking process	Performance	Periodic clean up, feed stocks, and energy constraints	Mathematical programming-mixed-integer nonlinear programming-based continuous-time representation
Suominen et al. [19]	Cooper production	Production maximization	Sequence	Mathematical programming-nonlinear optimization
Feng et al. [18]	Electrochemical processes	Makespan	Online arrives	Mathematical programming-mixed-integer linear programming
Beerbühl et al. [26]	Electrochemical processes	Costs	Nonlinear characteristics, energy storage, and capacity planning	Heuristic

In order to interconnect the tank position variables with the starting time variables and to enforce a feasible schedule, nonoverlapping constraints are defined by

$$S_i + t_{i,k} - M(2 - x_{i,k,p-1} - x_{j,k,p}) \leq S_j, \quad (5)$$

for all $p = 2, \dots, |I_k|$, $i \neq j \in I_k$, $k = 1, \dots, m$.

If the operations i and j are assigned to the same tank k for consecutive positions $p-1$ and p , then the starting time S_j of operation j must not be earlier than the completion time $S_i + t_{i,k}$ of operation i . M is a big constant taken sufficiently large in order to guarantee constraints (5) to be valid if at least one of the tank position variables $x_{i,k,p}$ and $x_{i,k,p-1}$ is zero; in other words, operations i and j are not assigned to consecutive positions on the same tank and

consequently, a nonoverlapping constraint does not have to be taken into account.

Each operation is bounded within a time window. This condition is ensured by

$$a_{i,k} \leq t_{i,k} \leq b_{i,k}, \quad \text{for all } k = 1, \dots, m, i \in O. \quad (6)$$

The total time required to conclude all the operations scheduled, C_{\max} , is defined by the constraints

$$S_i + \sum_{k \in M_i} x_{i,k,p} \leq C_{\max}, \quad \text{for all } i \in O, p = 1, \dots, |I_k|. \quad (7)$$

To minimize, the C_{\max} is given by

$$\text{Min } C_{\max}. \quad (8)$$

To ensure compliance with these constraints for each hoist and avoid any collision between hoists that share the same track, the Delmia-Quest[®] simulation language is preferred in this research. Delmia-Quest[®] avoids any collision between hoists in each simulation run. In addition, the simulation language is able to order each couple of hoist operations. A controller, in the simulation model, establishes the order of movements, in each simulation run, to satisfy the aforementioned constraints. The approach taken in this study combines the key advantages of both MHEDA and event-discrete simulation. Using the Delmia-Quest[®] simulation language, the main constraints related to the hoists are satisfied. Furthermore, considering Delmia-Quest[®], the makespan and the workload of the most loaded machine are obtained directly from the simulation model, and the MHEDA is in charge of modeling the solution space distribution.

4. MHEDA for the MHSP

To understand the MHEDA methodology, a multihost scenario is detailed below. Table 5 shows the jobs, operations, machines, and additional information about the multihost layout configuration. Figure 3 depicts the layout.

MHEDA contains differences and similarities with respect to a recent published algorithm, i.e., the MEDA (Malloves Estimation of Distribution Algorithm). The MEDA algorithm mentioned above can be consulted in Pérez-Rodríguez and Hernández-Aguirre [51]. The details of the MHEDA and also the differences and similarities between the algorithms, MHEDA and MEDA, are detailed below.

4.1. Solution Representation. In this research, three vectors are used to represent a solution, i.e., a task sequence vector, a machine assignment vector, and a hoist assignment vector.

The first vector, the task sequence vector, is a permutation-based representation. The task sequence vector models the processing sequence of operations. An example is depicted below; based on the Table 3, there are three jobs and ten operations, indexed from 1 to 10.

A task sequence vector example is

4 1 7 5 2 8 6 3 9 10,

where the operation number four should be executed at the beginning; after that, the operation number one, the operation number seven, and so on. Although each element in the task sequence vector shown above can be located in any position along the solution vector, as any permutation-based representation, the precedence between operations of each job must be kept in the vector. For example, the operations 1 and 2, from job number one, must appear in the same order in the vector, from left to right. If the vector solution of the operation sequence satisfies the precedence mentioned above, it satisfies the corresponding original operation sequence from the job mentioned. This representation is based on Gen et al. [52]. As a fixed parameter, 1000 solution vectors are defined for the population.

The second vector, the machine assignment vector: its length equals the total number of operations, where each element represents the corresponding selected machine for each operation. To explain the representation, an example is provided by considering the task sequence vector shown below:

1 2 3 4 5 6 7 8 9 10.

Then, a feasible machine assignment vector can be

4 3 2 1 8 6 3 2 1 2.

Based on Table 3, it means that the machine number four must be used for the operation number one, the machine number three for the operation number two, and so on. As a fixed parameter, 1000 solution vectors are defined for the population.

The third vector, the hoist assignment vector: its length equals the total number of operations, where each element represents the corresponding selected hoist for each operation. Based on Table 3, an example is depicted, by considering the task sequence vector shown below:

1 2 3 4 5 6 7 8 9 10,

with ten operations, and two hoists, indexed from 1 to 2, a feasible hoist assignment vector can be

2 1 2 1 2 1 2 2 1 1,

where the first operation should be executed with the hoist number two, after that, the second operation, with the hoist number one, and so on. As a fixed parameter, 1000 solution vectors are defined for the population.

By using this representation, i.e., through three vectors, infeasible individuals are not generated.

4.2. Fitness Computing. Two objectives to optimize are considered in this research, i.e., the makespan and the maximum workload. For each solution, the corresponding values are obtained from the simulation model, built on Delmia-Quest[®]. The main details of the simulation model are outlined below:

- (i) A conveyor system is used to move rack baskets through different hoists
- (ii) The conveyor system is only unidirectional
- (iii) When a rack has finished its sequence production, it goes through the conveyor system to exist the system
- (iv) Different hoists give service to any rack according to predefined sequence
- (v) Racks can receive service by different hoists based on the predefined sequence
- (vi) Putting racks in tanks is only possible by hoists
- (vii) Hoists are also used to put racks on the conveyor

With all these features, the simulation model is able to integrate operation times and workflows. Finally, the fitness is used by the MHEDA to build the Pareto-front and to obtain the best solution at the end of the execution.

TABLE 5: Scenario and data for the MHEDA.

Operations job, precedence	Index	Processing time window machines (tanks)									
		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
O_{11}	1	X	X	X	✓	X	X	X	X	X	X
O_{12}	2	X	X	✓	X	✓	X	X	✓	X	X
O_{13}	3	X	✓	X	X	X	✓	✓	X	✓	X
O_{21}	4	✓	X	X	X	✓	✓	✓	X	X	✓
O_{22}	5	X	X	✓	X	✓	X	X	✓	X	X
O_{23}	6	X	X	X	✓	X	✓	X	X	✓	X
O_{31}	7	X	X	✓	X	X	X	✓	X	✓	✓
O_{32}	8	X	✓	X	X	X	X	X	✓	X	X
O_{33}	9	✓	X	X	X	X	X	X	X	✓	X
O_{34}	10	X	✓	X	X	X	X	X	X	X	✓
Tracks						2					
Hoists						2					
Transfer time						1 (between locations)					
Cross time						1 (for any machine/tank)					

✓ means the machine can process the operation
 X means the machine does not perform the operation

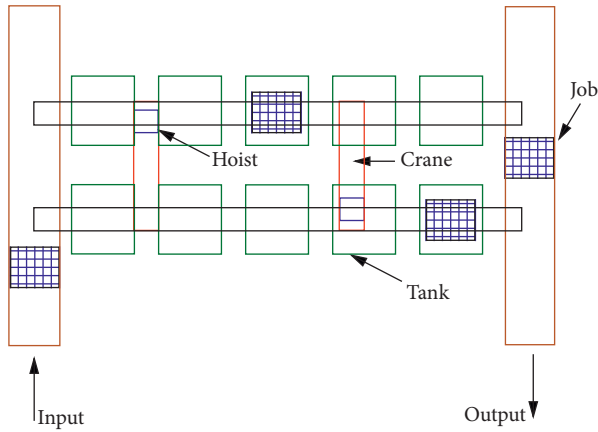


FIGURE 3: Multihoist layout configuration.

4.3. Pareto-Front. All members of the population are used in order to build a Pareto-front based on Kacem et al.'s [53] research. Once a Pareto-front is built, the selection process of the best-candidate solutions is based on where the candidates are located on the Pareto-front. Only the members located in the first Pareto-front layer are preferable. Although the MEDA utilizes the Pareto-front for the selection process as the MHEDA, the MEDA requires a tournament process to select the corresponding candidates. In the MHEDA, the candidates are selected without tournament, i.e., the selected candidates must be located in the first Pareto-front layer. Figure 4 depicts a Pareto-optimality approach example. A nondominated set of solutions in each generation is found and used for building the probability model.

4.4. Probability Model

4.4.1. Probability Model for Task Sequence Vectors. As in the MEDA, the MHEDA builds a probability model by the

Mallows model for the selected task sequence vectors. Fligner and Verducci [9] formally establish the Mallows model as

$$P(\sigma) = \psi(\theta)^{-1} \exp(-\theta D(\sigma, \sigma_0)), \quad (9)$$

where θ is a spread parameter, $D(\sigma, \sigma_0)$ is the distance from σ to the central permutation σ_0 , and $\psi(\theta)$ is a normalization constant. In the present work, the Kendall's τ is the distance metric with which the Mallows model is coupled. The Mallows model was initially proposed by Mallows [8] and later improved by Fligner and Verducci [9] through the generalized Mallows distribution (GMD). The GMD is given as follows:

$$P(\sigma) = \psi(\theta)^{-1} \exp\left(\sum_{j=1}^{n-1} -\theta_j V_j(\sigma, \sigma_0)\right), \quad (10)$$

where θ_j are dispersion parameters, $\psi(\theta)$ is a normalization constant, and $V_j(\sigma, \sigma_0)$ can be defined as an auxiliary vector, i.e., it represents the number of positions on the right side of j with values smaller than the current position in the permutation (σ, σ_0) .

Consider a task sequence vector with four tasks, n equals four, as an example.

Let a central permutation be given by $\sigma_0 = \{1, 2, 3, 4\}$, i.e., task 1 is located in the first position, task 2 is located in the second position, and so on.

Let a task sequence vector be given by $\sigma = \{4, 2, 3, 1\}$. Therefore, $V_j(\sigma, \sigma_0)$ for the j th position is calculated as shown below:

$$j V_j(\sigma, \sigma_0)$$

- (1) 3 because there are three numbers (positions), less than the number four (on its right), i.e., 2, 3, and 1
- (2) 1 because there is one number (position) less than the number two (on its right), i.e., 1
- (3) 1 because there is one number (position) less than the number three (on its right), i.e., 1

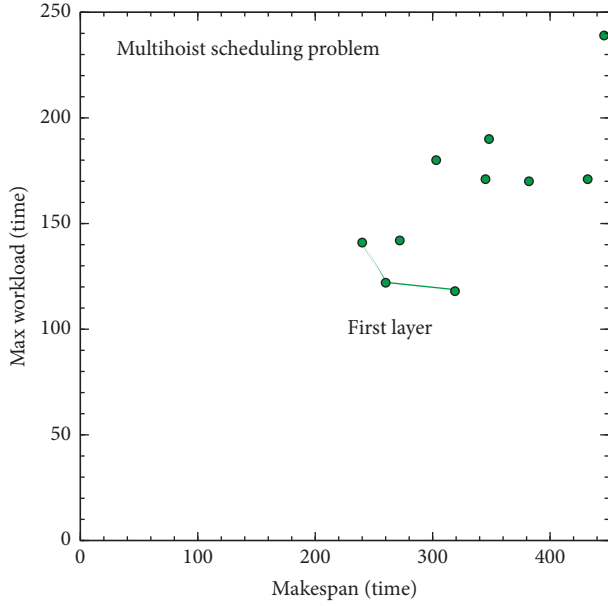


FIGURE 4: A Pareto-front example.

The corresponding Kendall's τ distance is five, i.e., $D_t = \sum_{j=1}^{n-1} V_j(\sigma, \sigma_0) = 5$, i.e., $3 + 1 + 1 = 5$. The procedure is carried out for each task sequence vector, i.e., $V_j(\sigma, \sigma_0)$ should be calculated for all the M vectors in the selected population.

The next step consists of computing the dispersion parameters θ_j , which is given by

$$\bar{V}_j = \frac{n-1}{\exp(\theta_j) - 1} - \frac{n-j+1}{\exp(\theta_j(n-j+1)) - 1}, \quad (11)$$

where $\bar{V}_j = \sum_{i=1}^N V_j(\sigma, \sigma_0)$. Equation (11) can be solved by the Newton-Raphson method.

The probability distribution of the random variables in the auxiliary vector $V_j(\sigma, \sigma_0)$ can be written as

$$P(V_j(\sigma, \sigma_0) = r_j) = \frac{\exp(-\theta_j r_j)}{\psi_j(\theta_j)}, \quad r_j \in \{0, \dots, n-j\}. \quad (12)$$

This means that the possible values for $V_j(\sigma, \sigma_0)$ in the j th position are located between 0 and $n-j$, where n is the length of the selected task sequence vectors.

For the operation scheduling decision, the offspring are obtained as follows. As an example, let $V_s(\sigma, \sigma_0)$ a sample vector obtained from equation (12). Specifically,

- (1) j $V_s(\sigma, \sigma_0)$
- (2) 2
- (3) 1

Generation of new offspring, i.e., the corresponding auxiliary vectors $V_s(\sigma, \sigma_0)$, is done by applying the algorithm proposed by Meilä et al. [54]; then, permutation $V_s(\sigma, \sigma_0)^{-1}$ is obtained. According to the sample vector shown above, the corresponding permutation $V_s(\sigma, \sigma_0)^{-1} = \{2, 4, 1, 3\}$.

Finally, each permutation σ_s is obtained by inverting and composing with the consensus ranking σ_0 , i.e., the last result is $\sigma_s = \{3, 1, 4, 2\}$.

4.4.2. Probability Model for the Machine Assignment Vectors. As in the MEDA, the MHEDA builds a probability model to determine an estimate of a distribution model to generate new offspring (machine assignment vectors) using the selected members. Again, as in the MEDA, the MHEDA obtains the estimation by the Univariate Marginal Distribution Algorithm (UMDA).

The probability model for the selected machine assignment vectors selected previously can be represented by a probability matrix p , i.e., each p_i value represents the amount of times where the machine i is elected for a specific position. Each element of the probability matrix p represents the probability that a position is processed on a machine. The value of each element indicates the rationality of a position processed on a certain machine.

Based on Table 3, and as a short example, the vectors shown below are used to build the corresponding probability matrix p for the first position, i.e., p^1 (Table 6).

The offspring are obtained as follows: for each position, generate a $U[0, 1]$ value. Then, the value mentioned above is interpolated in the cumulative probability matrix p to identify which machine should be selected.

4.4.3. Probability Model for the Hoist Assignment Vectors. The MHEDA uses the UMDA algorithm to determine an estimate of a distribution model to generate new offspring (hoist assignment vectors) using the selected members.

The probability model for the hoist assignment vectors selected previously can be represented by a probability matrix, q , i.e., each q_i value represents the amount of times where the hoist i is observed for a specific task. Each element of the probability matrix q represents the probability that a task is processed on a hoist. The value of each element indicates the rationality of a task processed on a certain hoist.

Again, as a short example, the vectors shown below are used to build the corresponding probability matrix q for the first task, i.e., q^1 (Table 7).

Therefore, the offspring are obtained with the same procedure as that of machine assignment vectors, i.e., for each task, generate a $U[0, 1]$ value. Then, the value mentioned above is interpolated in the cumulative probability matrix, q , to identify which hoist should be selected.

In each generation, new candidate solutions are used for building the Pareto-front, and the probability model is updated with the information from the new members located in the first Pareto-front layer.

4.5. Multihoist Simulation Model. The simulation model is built on Delmia-Quest®. This model includes several types of details that the multihoist process presents: set elements, set hoists and tracks, set jobs, set processes, load and unload processing, and transferring of jobs between tanks. These situations are present in the given process. The model is able to handle any operation of each job that

TABLE 6: Selected machine assignment vectors.

4	3	2	1	8	6	3	2	1	2
1	3	4	8	5	5	9	4	7	10
10	4	9	5	3	1	4	2	9	2
4	3	1	3	2	3	2	1	4	2
4	10	9	8	8	8	9	9	9	2
4	3	2	1	8	6	3	2	1	2
Hoist	$p(X_1 = x)$								
1	1/6								
4	4/6								
10	1/6								

TABLE 7: Selected hoist assignment vectors.

2	2	1	1	2	1	2	2	1
1	2	1	1	1	2	1	2	2
1	2	2	2	1	2	2	1	2
2	1	1	2	2	1	2	2	2
2	2	1	2	2	3	1	2	1
2	2	2	2	1	1	1	2	2
Hoist	$p(X_1 = x)$							
1	2/6							
2	4/6							

is scheduled. The model is able to identify the sequences of operations for each job. Figure 5 shows a global 3D layout of the production process. The reason to utilize a discrete-event simulation platform is due to the stochasticity of the underlying process.

The main procedure to build the simulation model, with the features mentioned above, is detailed below, and the model is elaborated using batch commands provided by Delmia-Quest®.

4.5.1. Conveyor Structure. The aforementioned process contains a conveyor system. It is used to transport any basket (job) through the shop floor. The conveyor system is built using 3D linear and arc segments provided by the platform. Each conveyor segment is positioned and connected according to the flow and the real distribution of the shop floor.

4.5.2. Hoists. Each hoist is used to execute load and unload operations between tanks (machines). Each hoist attends to a specific group of tanks. If any basket requires a specific operation in a specific machine, then the assigned hoist (to that machine) executes the transport task. The hoist's movements, such as forward, return, park, load, and unload, are executed or controlled by logic commands provided by the platform. The aforementioned logic commands are also able to avoid collision between hoists.

4.5.3. Source and Sink. A source is built in the model. It enables the baskets to enter the model. Also, a sink is built in the model to destroy all the baskets after the production process.

4.5.4. Buffers. The buffers are used as load and unload locations. Once a basket enters the shop floor, it goes through the conveyor system and waits on the corresponding buffer until the hoist executes the movement.

4.5.5. Tanks. The tanks are actually machines. These are built and positioned on the shop floor according to the layout of the process.

4.5.6. Baskets. The baskets are actually jobs. These are created by the source. These baskets go by different routes in the production process according to the requirements.

4.5.7. Setting Processes. A process is an operation in the MHSP. A process is executed by a previously established tank. However, the process must be detailed, i.e., indicating which basket should be processed, and the processing time window of the process. Once a process is established, it should be linked to the corresponding tank.

4.5.8. Setting Process Sequence. A process sequence must be established for each basket. This allows the model to verify if the basket has finished all its operations in the production process. If that is not the case, the basket continues in the shop floor by the conveyor system until all its operations have been concluded.

4.5.9. Verification and Validation. The simulation model is run under different conditions to determine if its computer programming and implementation are correct as an application in verification technique, which is known as the fixed values test [55]. The throughput, as model result, is verified against data provided by the managers of the process. Figure 6 depicts previous descriptions of three months of real production.

Furthermore, the validation of the simulation model is realized statistically. A comparison of the results derived by the simulation model with real production is done under the same initial conditions, satisfying statistical assumptions in the validation. Figure 7 depicts the information below.

4.5.10. Communication between MHEDA and the Simulation Model. The main steps in the communication of the MHEDA algorithm with the simulation model are given below.

In C++

- (i) For each member of the population do
 - (1) Build a batch file. It will be opened by Delmia Quest® to execute the instructions
 - (a) For each job, set the set of operations
 - (b) For each job, set the sequence between the operations
 - (c) For each operation, set the processing time
 - (d) For each tank, set the number of operations able to process

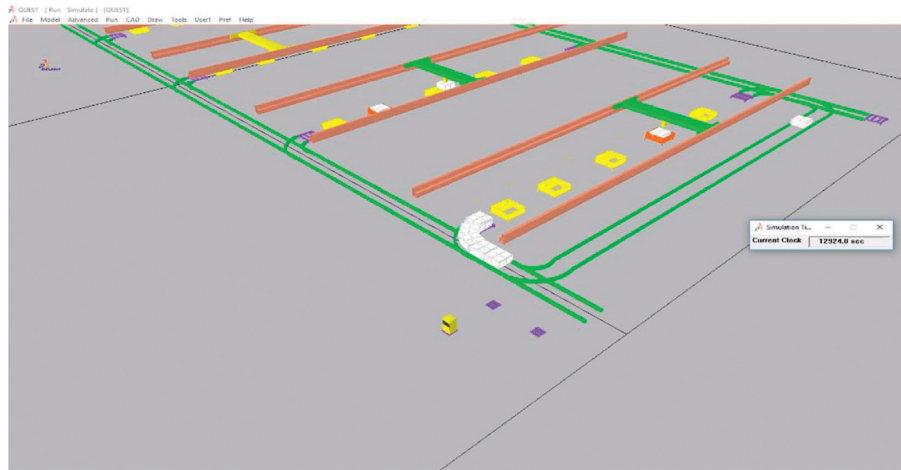


FIGURE 5: Simulation model's production process.

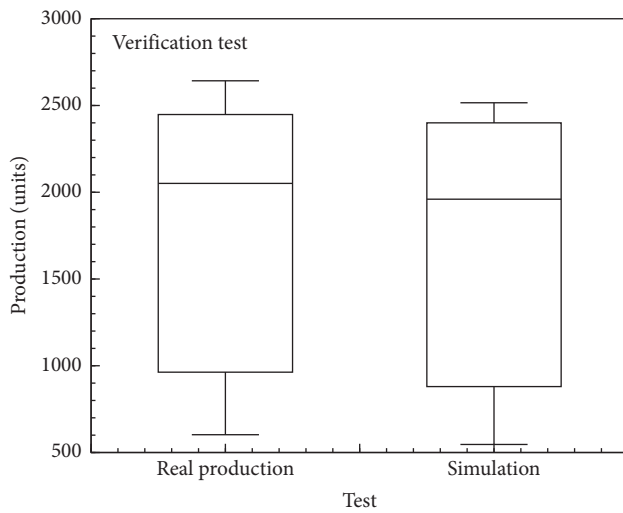


FIGURE 6: Fixed values test.

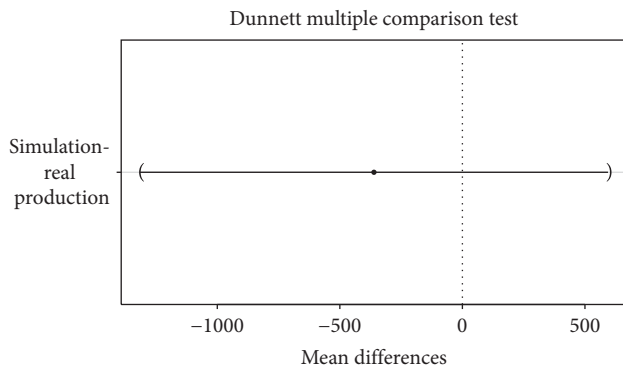


FIGURE 7: Operational validity.

- (e) For each tank, set the operations able to process
- (f) Save and close the batch file
- (2) Transfer program execution to the simulation language by command prompt

In Delmia-Quest®

- (a) Read the batch file
- (b) Build the multihoist simulation model
- (c) Run simulation until all the jobs have completed their operations
- (d) Save the results (fitness) in a text file
- (e) Clear the batch file
- (f) Close the multihoist simulation model
- (g) Return program execution to C++

In C++

- (a) Open the text file and collect the fitness
- (b) Clear the text file

End for

4.6. Replacement. All old members are replaced with the offspring. The MHEDA framework is shown in Algorithm 1.

Finally, Table 8 depicts the main differences and similarities between the MHEDA and the MEDA for clarify.

5. Results and Comparison

In order to validate the relevance of this paper, a comparison of the MHEDA results with others is done. A set of standard benchmarking datasets is used for comparison. The Adams et al. [56] instances; the Fisher and Thompson [57] instances; the Lawrence [58] instances; the Applegate and Cook [59] instances; the Storer et al. [60] instances; and the Yamada and Nakano [61] instances.

For each instance, 30 trials are executed to account for the stochastic nature of the MHEDA.

Three metrics are used to compare the performance of the algorithms. First, the mean absolute error (MAE) is

$$MAE(c_i) = |c_i - c^+|, \quad (13)$$

where c_i is the best hyper volume, from the Pareto-front, obtained after running each trial, and c^+ is the best hyper


```

 $D_0 \leftarrow$  Generate  $M$  individuals
 $t := 1$ 
Do
   $\text{FitD}_{t-1} \leftarrow$  Evaluate individuals (fitness) through Delmia Quest®
   $\text{Pareto}_{t-1} \leftarrow$  Select best individuals from  $\text{FitD}_{t-1}$ 
   $\sigma_0 \leftarrow$  Central permutation computing from  $\text{Pareto}_{t-1}$ 
   $V_{t-1} \leftarrow$  Distance computing from  $D_{t-1}$  and  $\sigma_0$ 
   $\emptyset \leftarrow$  Spread parameter computing from  $V_{t-1}$ 
   $Ds_t \leftarrow$  Sampling from  $V_{t-1}$ 
   $p_{t-1} \leftarrow$   $p$  matrix computing from  $\text{Pareto}_{t-1}$ 
   $Dp_t \leftarrow$  Sampling from  $p_{t-1}$ 
   $q_{t-1} \leftarrow$   $q$  matrix computing from  $\text{Pareto}_{t-1}$ 
   $Dq_t \leftarrow$  Sampling from  $q_{t-1}$ 
   $D_t \leftarrow$  Replacement all old members with new offspring
   $t := t + 1$ 
Until stopping criterion is met

```

ALGORITHM 1: Pseudocode MHEDA framework.

TABLE 8: Main differences and similarities between MHEDA and MEDA.

Characteristics	MEDA by Pérez-Rodríguez and Hernández-Aguirre [51]	MHEDA proposed
Selection process	Tournament process with Pareto-front	Only Pareto-front
Probability model for sequence vectors	Mallows model	Mallows model
Probability model for operations vectors	Marginal distribution	Marginal distribution
Simulation	Not available	Included

volume obtained in all the trials. It is computed for each instance.

The MAE is used to quantify the precision of the algorithm with respect to the values it should obtain.

Second, the mean square error (MSE) is

$$\text{MSE}(c_i) = (c_i - c^+)^2. \quad (14)$$

The MSE measures the amount of error between two datasets, that is, between the values that the algorithm returns and the values that it should obtain.

Third, the relative percentage increase (RPI) is

$$\text{RPI}(c_i) = \frac{(c_i - c^+)}{c^+}. \quad (15)$$

The RPI is used to compare two quantities while taking into account the “sizes” of the things being compared. The comparison is expressed as a ratio.

5.1. Comparison with Other Estimations of Distribution Algorithms. As other previous works, some EDAs have been included as a benchmark for comparison with the MHEDA scheme; the MIMIC by De Bonet et al. [62]; the COMIT by Baluja and Davies [63], and the BOA by Pelikan et al. [64].

Figure 8 indicates the output by the algorithms using equation (13). Through box and whisker charts, the dispersion of the values obtained, using the MAE metric, is appreciated. The dispersion of the values, using the MAE, shows the results over the instances and over different runs. As we can see, the MHEDA obtains better results than the

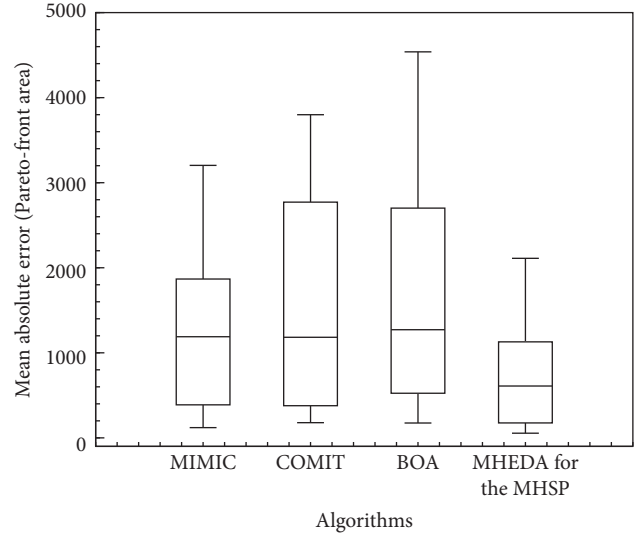


FIGURE 8: Comparison results for the MHSP using the mean absolute error.

other algorithms. Based on the MAE, the MHEDA is more accurate with respect to the other algorithms.

Figure 9 shows the output by the algorithms using equation (14). The dispersion of the results is similar to Figure 8. Based on the MSE metric, the MHEDA obtains the interval with the smallest error with respect to the other algorithms for the MHSP.

Figure 10 depicts the results obtained by the algorithms using the equation (15). The MHEDA obtains better results

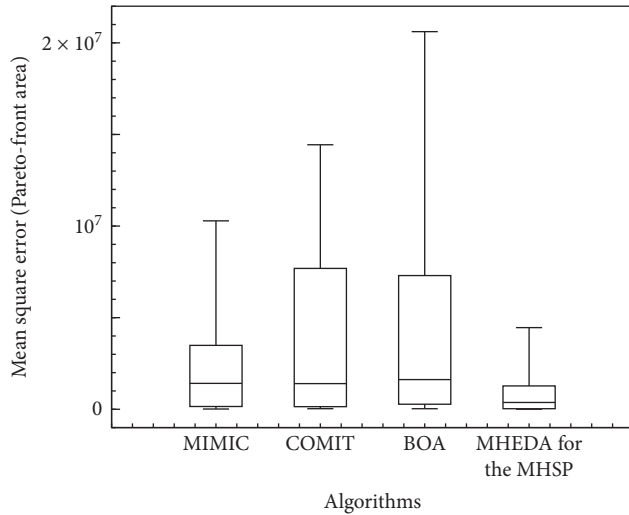


FIGURE 9: Comparison of results for the MHSP using the mean square error.

than the other algorithms. The MHEDA scheme outperforms all the previous results.

Based on the results, the members located in the first Pareto-front layer contribute to improve the search process of the MHEDA scheme, against other algorithms.

Although the performance of all the algorithms used in the comparison is outstanding, the MHEDA scheme can find the best value in all the trials. In addition, the MHEDA scheme is able to find the best hyper volume for all the instances used in the comparative. It always helps to find the best value in all the trails. The dispersion of MHEDA is less than other algorithms; it means that the solutions found by the MHEDA are more concentrated around the best value, than other algorithms, i.e., the average of solutions of the MHEDA converges better than other approaches to the best found value.

5.2. Comparison with Other Multiobjective Algorithms. Furthermore, as other previous works, two multiobjective algorithms have been considered to evaluate the MHEDA performance, such as the NSGA by Srinivas and Deb [65] and the NSGA-II by Deb et al. [66]. The experiments are executed in the same computer and language specification.

Figure 11 details the output for the algorithms using equation (13). The dispersion of the values, using the MAE metric, shows the results over the instances and over different runs. Through box and whisper charts, it is possible to identify that the behavior of the algorithms is similar than the output detailed above. Based on the MAE metric, the MHEDA is more accurate with respect to the other algorithms.

Figure 12 indicates the output by the algorithms using equation (14). The dispersion of the results is similar to Figure 11. Based on the MSE metric, the MHEDA obtains the smallest median error with respect to the other algorithms for the MHSP.

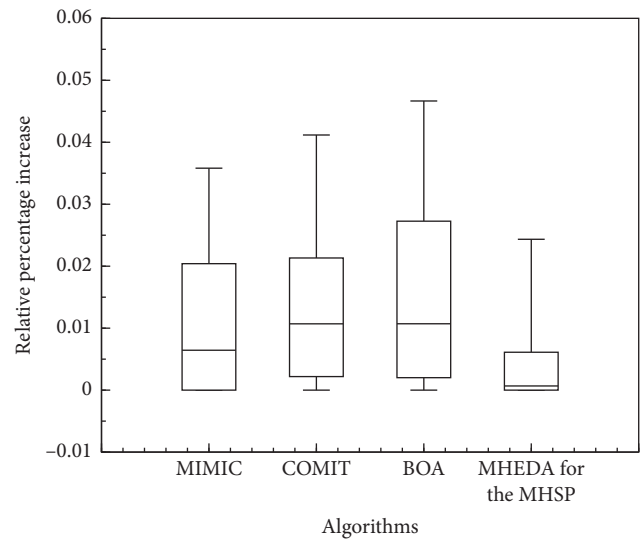


FIGURE 10: Comparative results for the MHSP using the relative percentage increase.

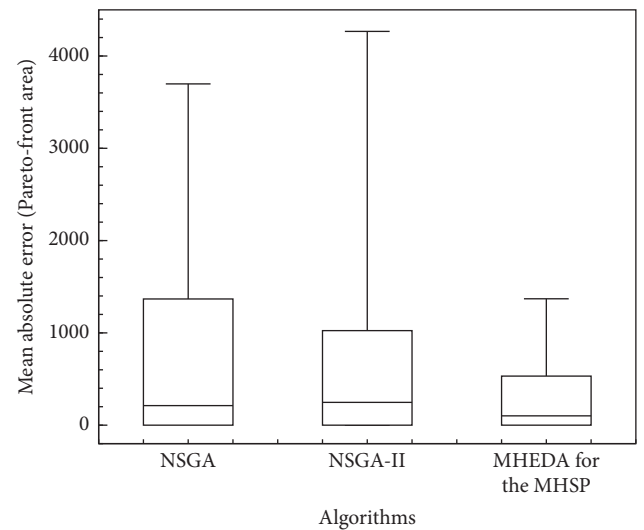


FIGURE 11: Comparison results for the MHSP using the mean absolute error.

Figure 13 shows the results obtained for the algorithms using equation (15). The behavior is practically the same between the algorithms, i.e., the MHEDA scheme again outperforms all the previous results.

Figure 13 includes the performance of the three algorithms: the NSGA, the NSGA-II, and the MHEDA after running all the instances. Based on equation (15), the MHEDA scheme outperforms all the algorithms used in the comparative. As we can see, the MHEDA is competitive in order to identify the best-candidate solutions. The performances of the algorithms used in the comparative are very similar than the previous comparison. The medians are 0.25% above the best found value. The MHEDA scheme again found the best value in all the trials. The MHEDA scheme is consistently able to find the best hyper volume for all the instances used in the comparative. Again, it

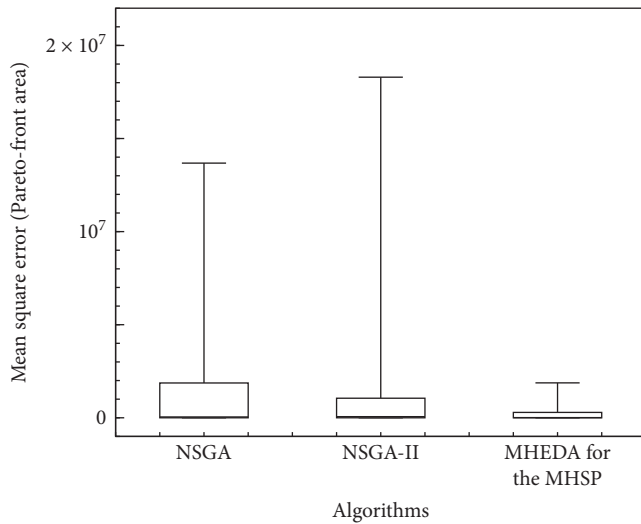


FIGURE 12: Comparison results for the MHSP using the mean square error.

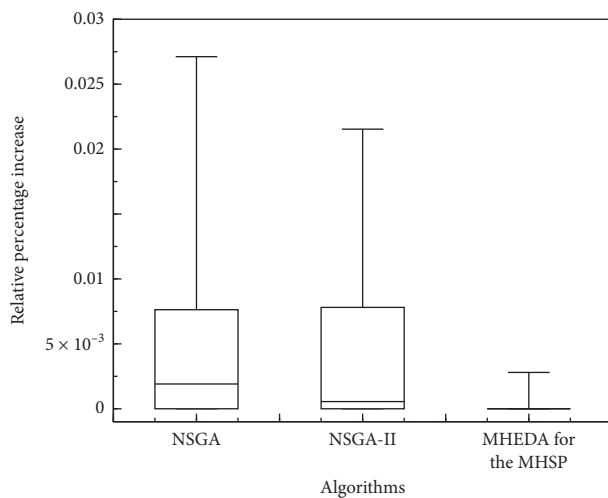


FIGURE 13: Comparative results for the MHSP using the relative percentage increase.

always helps to find the best value in all the trails. The dispersion of MHEDA is much less than other algorithms; it means that the solutions found by the MHEDA are more concentrated around the best value, than other algorithms, i.e., the average of solutions of the MHEDA converges better than other approaches to the best found value. Based on the results, the MHEDA scheme is able to identify the best solutions by a distance-based ranking model, i.e., the Mallows model.

5.3. Comparison with Recent Algorithms for the MHSP. Based on the previous results, recent algorithms are proposed as a benchmark for comparison with the MHEDA scheme. The recent algorithms mentioned above are the mixed-integer programming model with heuristic presented by El Amraoui and Elhafi [38]; the algorithm proposed by

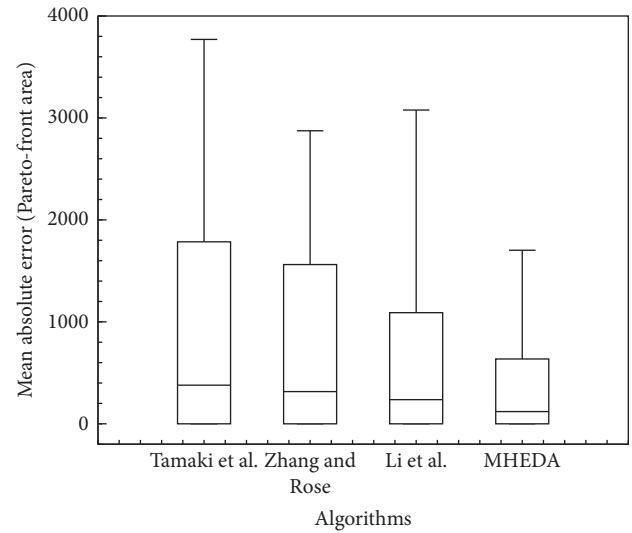


FIGURE 14: Comparison results for the MHSP using the mean absolute error.

Xie et al. [41]; and the mathematical model with genetic algorithm detailed by Zeng et al. [44]. All the algorithms mentioned are considered recent algorithms for the MHSP. These algorithms have been implemented by the authors. The experiments are executed with the same parameters and specifications detailed above.

Figure 14 shows the output for the algorithms using equation (13). The dispersion of the values, using the MAE metric, depicts the results over the instances and over different runs. Through box and whisker charts, it is possible to identify that the behavior of all the algorithms are similar. Based on the MAE metric, the MHEDA is more accurate with respect to the other algorithms and outperforms all the algorithms used in the comparison.

Figure 15 indicates the output by the algorithms using equation (14). The dispersion of the results is similar to Figure 14. Based on the MSE metric, the MHEDA obtains the smallest median error with respect to the other algorithms for the MHSP.

Figure 16 presents the results obtained for the algorithms using equation (15). In this case, the MHEDA outperforms all the other recent algorithms. Practically, there exists significant difference based on Figure 17. The performance of recent algorithms and the MHEDA scheme is different.

Based on Figure 16, the medians are 0.10% above the best found value. The MHEDA scheme again can find the best value in all the trials. The MHEDA scheme is consistently able to find the best hyper volume for all the instances used in the comparative. Consistently, it always helps to find the best value in all the trails. The dispersion of MHEDA is much less than other algorithms; it means that the solutions found by the MHEDA are more concentrated around the best value, than other algorithms, i.e., the average of solutions of the MHEDA converges better than other approaches to the best found value.

Based on the results, the MHEDA scheme tackles the inconvenience of the EDAs, i.e., lack of diversity of the solutions and poor ability of exploitation. The proposed

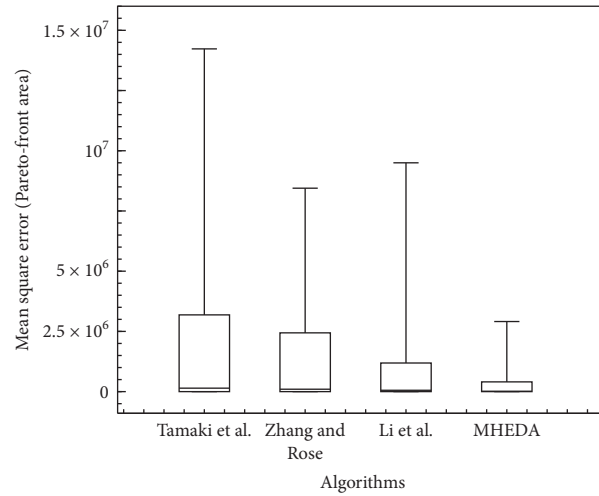


FIGURE 15: Comparison results for the MHSP using the mean square error.

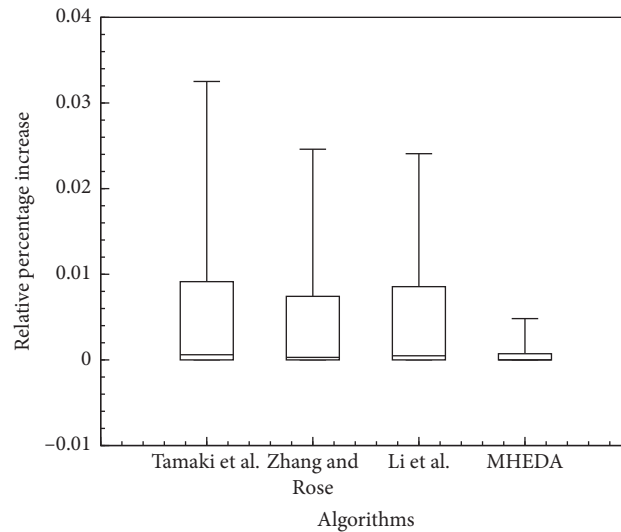


FIGURE 16: Comparative results for the MHSP using the relative percentage increase.

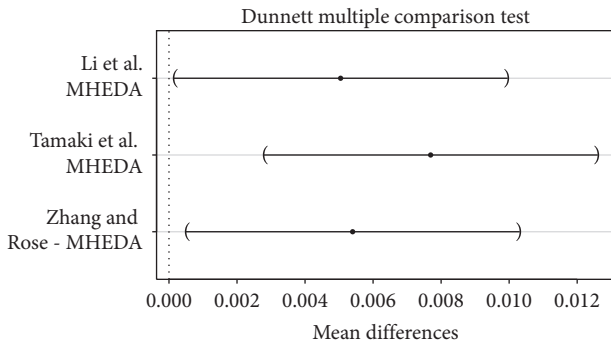


FIGURE 17: Dunnett test for the MHSP.

algorithm does not require evolutionary operators to get offspring, such as cross and mutation. The MHEDA makes use of the GMD process to establish a search direction.

The EDA scheme considers population size, replacement (also known as generation gap), and selection strategy as key parameters. It is consistently with Grefenstette [67].

- (i) The population size; in the current experiments, the population size ranged from 500 to 1000 solutions in increments of 500.
- (ii) The replacement; the current experiments allowed to vary the percentage of the population to be replaced during each generation between 50% and 100%, in increments of 50%.
- (iii) The selection; the experiments compared two ways to bubble, i.e., sorting by the makespan and sorting by the maximum workload.

A design of experiment is built to identify the best parameter of each parameter. Parameter tuning is detailed in Table 9.

TABLE 9: Parameter tuning data.

Case	Stopping criteria (generations)	Population size	Replacement (%)
1	50	500	50
2	25	500	100
3	25	1000	50
4	50	1000	100

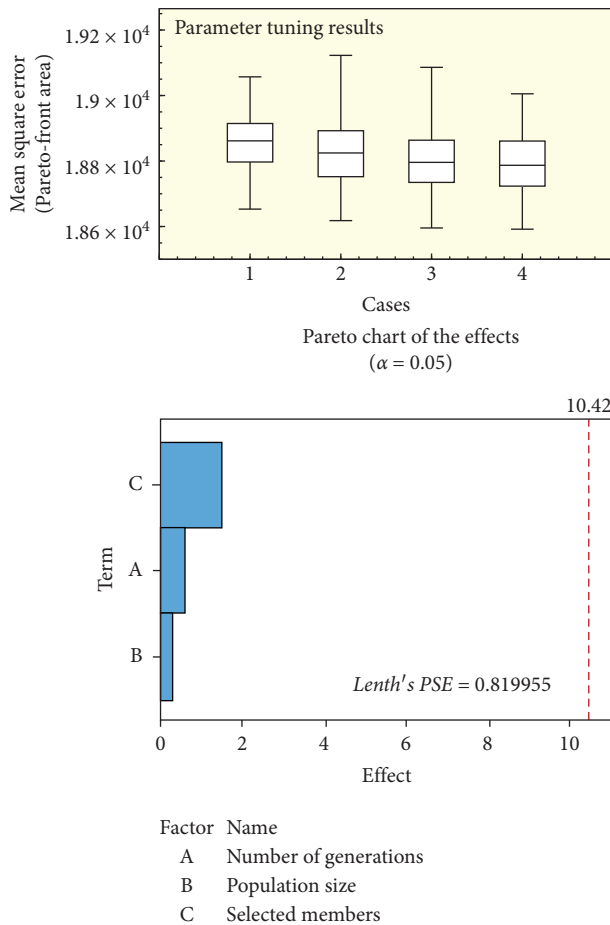


FIGURE 18: Parameter tuning.

Finally, the results of the parameter tuning are shown in Figure 18. There is no statistically significant difference of any of the three controlled parameters (number of generations, initial population size, and selected population size). Therefore, the parameters used are the same for all the algorithms.

6. Conclusions and Future Research

This paper considers the MHSP. It tries to determine the sequence of operations for each hoist to perform so that some performance metric is optimized. The MHEDA scheme is proposed for tackling the problem and simulating a solution. The aforementioned instances were used as input and test parameters in order to validate the Mallows model as a probability model for the MHSP. The hybridization between the Mallows model and the EDA proposed helps to identify an

explicit distribution over a set of permutations. Traditional operators are not considered for building suitable sequences, i.e., operation sequence vectors. These are obtained from the Mallows model. The proposed exponential model, i.e., the Mallows model, is able to tackle the inconvenience any EDA has with permutation-based problems. The MHEDA scheme does not require to be reconfigured for solving the MHSP. The Mallows model is considered as a specific probability model for this issue. Based on the results, the exponential model is more effective against other algorithms. In addition, the MHEDA scheme considers three probabilistic models instead of only one as almost all the EDAs reported in the literature.

The results of the MHEDA are more concentrated around the best Pareto-front obtained in all the trails. Meanwhile the rest of the algorithms have more disperse results. It is consistent in all the experiments detailed above.

Considering only the best solutions, in the selection process through a Pareto-front approach, is suitable to get a better estimation of the central permutation. A better estimate of the central permutation helps to improve the performance of the MHEDA scheme.

Based on the results detailed above, a simulation model is useful to model the critical variables that influence the performance of the process. The simulation model should be considered in almost any proposed solution. Moreover, simulation optimization is an enabling tool to handle diverse manufacturing limitations such as the MHSP. The simulation approach helps to model the difference between the classical job-shop configurations and the MHSP. The simulation language is able to order each couple of hoist operations, and avoid any collision between hoists, to satisfy the aforementioned constraints related to operations of the hoists.

In addition, the replacement step, in the MHEDA scheme, utilizes only the offspring to continue the evolutionary progress. It helps to reduce the dispersion of the results around the best solution.

Finally, as other previous works, this research contributes using the MHEDA as an optimization method for working with any simulation language.

Future research work could consider dynamic aspects such as failures of the hoists, shutdowns, jobs with priority, type of jobs, and type of hoists. Therefore, the aforementioned dynamic issues should be integrated for any proposed algorithm.

Other comparisons should be investigated, as future research, in order to know in which other permutation-based problems, the MHEDA has a competitive performance.

Data Availability

All data are included in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Varnier, O. Grunder, and P. Baptiste, "Improving the productivity of electroplating lines by changing the layout of the tanks," in *Proceedings of the Emerging Technologies and Factory Automation, 1995. ETFA'95, IEEE, Paris, France, October 1995*.
- [2] K. Kujawski and J. Świaołek, "Queue preparation in the dynamic hoist scheduling problem," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8189–8194, 2011.
- [3] D. Ramin, E. Leo, L. Nicolosi, S. Spinelli, and A. Brusaferrri, "A Unified PSO-based method for multi-hoist scheduling in advanced Galvanic plants," *CoDIT*, vol. 2020, pp. 656–661, 2020.
- [4] Q. Zhang, H. Manier, and M. A. Manier, "A disjunctive graph and shifting bottleneck heuristics for multi hoists scheduling problem," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6963–6968, 2011.
- [5] P. Baptiste, B. Legeard, M. A. Manier, and C. Varnier, "Optimization with constraint logic programming: the hoist scheduling problem solved with various solvers," in *Transactions on Information and Communications Technologies*, pp. 1473–3517, WIT Press, Southampton, UK, 1993.
- [6] A. Allahverdi, "A survey of scheduling problems with no-wait in process," *European Journal of Operational Research*, vol. 255, no. 3, pp. 665–686, 2016.
- [7] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer Science and Business Media, Berlin, Germany, 2001.
- [8] C. L. Mallows, "Non-null ranking models," *I. Biometrika*, vol. 44, no. 2, pp. 114–130, 1957.
- [9] M. A. Fligner and J. S. Verducci, "Distance based ranking models," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 48, no. 3, pp. 359–369, 1986.
- [10] M. A. Fligner and J. S. Verducci, "Multistage ranking models," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 892–901, 1988.
- [11] R. Pérez-Rodríguez, S. Jöns, A. Hernández-Aguirre, and C. Alberto-Ochoa, "Simulation optimization for a flexible jobshop scheduling problem using an estimation of distribution algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 1–4, pp. 3–21, 2014.
- [12] O. Grunder, P. Baptiste, and D. Chappe, "The relationship between the physical layout of the work stations and the productivity of a saturated single-hoist production line," *International Journal of Production Research*, vol. 35, no. 8, pp. 2189–2212, 1997.
- [13] C. Subai, P. Baptiste, and E. Niel, "Scheduling issues for environmentally responsible manufacturing: the case of hoist scheduling in an electroplating line," *International Journal of Production Economics*, vol. 99, no. 1–2, pp. 74–87, 2006.
- [14] R. Berber, M. Yuceer, and Z. Ozdemir, "Automatic generation of production scheduling models in single stage multi-product batch plants: some examples," *Mathematical and Computer Modelling*, vol. 46, no. 1–2, pp. 69–79, 2007.
- [15] A. El Amraoui, M. A. Manier, A. El Moudni, and M. Benrejeb, "A linear optimization approach to the heterogeneous r-cyclic hoist scheduling problem," *Computers & Industrial Engineering*, vol. 65, no. 3, pp. 360–369, 2013a.
- [16] C. Zhao, J. Fu, and Q. Xu, "Production-ratio oriented optimization for multi-recipe material handling via simultaneous hoist scheduling and production line arrangement," *Computers & Chemical Engineering*, vol. 50, pp. 28–38, 2013.
- [17] S. Chtourou and M.-A. Manier, "A hybrid algorithm for the cyclic hoist scheduling problem with two transportation resources," *Computers & Industrial Engineering*, vol. 65, no. 3, pp. 426–437, 2013.
- [18] J. Feng, A. Che, and C. Chu, "Dynamic hoist scheduling problem with multi-capacity reentrant machines: a mixed integer programming approach," *Computers & Industrial Engineering*, vol. 87, pp. 611–620, 2015.
- [19] O. Suominen, V. Mörsky, R. Ritala, and M. Villkko, "Framework for optimization and scheduling of a copper production plant," *Computer Aided Chemical Engineering*, vol. 38, pp. 1243–1248, 2016.
- [20] L. Su, L. Tang, and I. E. Grossmann, "Scheduling of cracking production process with feedstocks and energy constraints," *Computers & Chemical Engineering*, vol. 94, pp. 92–103, 2016.
- [21] E. Wari and W. Zhu, "Multi-week MILP scheduling for an ice cream processing facility," *Computers & Chemical Engineering*, vol. 94, pp. 141–156, 2016.
- [22] H. Qu, S. Wang, and Q. Xu, "Simultaneous 2D hoist scheduling and production line design for multi-recipe and multi-stage material handling processes," *Chemical Engineering Science*, vol. 167, pp. 251–264, 2017.
- [23] F. Jianguang, C. Chengbin, and C. Ada, "Cyclic jobshop hoist scheduling with multi-capacity reentrant tanks and time-window constraints," *Computers & Industrial Engineering*, vol. 120, pp. 382–391, 2018.
- [24] A. El Amraoui, M. A. Manier, A. El Moudni, and M. Benrejeb, "Heuristic for the resolution of the cyclic hoist scheduling problem with multi-items," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8195–8200, 2011.
- [25] K. Kujawski and J. Świątek, "Dynamic hoist scheduling problem on real-life electroplating production line," in *Proceedings of the Electrical Engineering and Applied Computing*, pp. 581–594, Springer, Dordrecht, The Netherlands, July 2011.
- [26] S. Schulte Beerbühl, M. Fröhling, and F. Schultmann, "Combined scheduling and capacity planning of electricity-based ammonia production to integrate renewable energies," *European Journal of Operational Research*, vol. 241, no. 3, pp. 851–862, 2015.
- [27] E. Laajili, S. Lamrous, M.-A. Manier, and J.-M. Nicod, "An adapted variable neighborhood search based algorithm for the cyclic multi-hoist design and scheduling problem," *Computers & Industrial Engineering*, vol. 157, Article ID 107255, 2021.
- [28] J. M. Lim, "A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line," *Computers & Industrial Engineering*, vol. 33, no. 3–4, pp. 789–792, 1997.
- [29] M. Gravel, W. L. Price, and C. Gagné, "Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic," *European Journal of Operational Research*, vol. 143, no. 1, pp. 218–229, 2002.
- [30] H. Y. Wang, Y. W. Zhao, X. L. Xu, and W. L. Wang, "Scheduling batch and continuous process production based on an improved differential evolution algorithm," *Systems Engineering-Theory & Practice*, vol. 29, no. 11, pp. 157–167, 2009.
- [31] A. El Amraoui, M. A. Manier, A. El Moudni, and M. Benrejeb, "Genetic algorithm for a cyclic Hoist Scheduling Problem with time-window constraints and heterogeneous part jobs," in *Proceedings of the 2010 18th Mediterranean Conference on*

- Control & Automation (MED)*, pp. 351–356, IEEE, Marrakech, Morocco, June 2010.
- [32] A. El Amraoui, M. A. Manier, A. El Moudni, and M. Benrejeb, “A genetic algorithm approach for a single hoist scheduling problem with time windows constraints,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1761–1771, 2013b.
 - [33] W. Lei, H. Manier, M. A. Manier, and X. Wang, “A hybrid quantum evolutionary algorithm with improved decoding scheme for a robotic flow shop scheduling problem,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 3064724, 2017.
 - [34] E. Chové, P. Castagna, and R. Abbou, “Hoist scheduling problem: coupling reactive and predictive approaches,” *IFAC Proceedings Volumes*, vol. 42, no. 4, pp. 2077–2082, 2009.
 - [35] A. El Amraoui and A. Nait-Sidi-Moh, “P-temporal petri nets for hoist scheduling problem,” *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 297–302, 2012.
 - [36] X. Xie, Y. Zheng, and Y. Li, “Multi-crane scheduling in steel coil warehouse,” *Expert Systems with Applications*, vol. 41, no. 6, pp. 2874–2885, 2014.
 - [37] P. Yan, G. Wang, A. Che, and Y. Li, “Hybrid discrete differential evolution algorithm for biobjective cyclic hoist scheduling with reentrance,” *Computers & Operations Research*, vol. 76, pp. 155–166, 2016.
 - [38] A. El Amraoui and M. Elhafs, “An efficient new heuristic for the hoist scheduling problem,” *Computers & Operations Research*, vol. 67, pp. 184–192, 2016.
 - [39] N. P. Basán and C. A. Méndez, “Hybrid MILP/Simulation/heuristic algorithms to complex hoist scheduling problems,” *Computer Aided Chemical Engineering*, vol. 38, pp. 1929–1934, 2016.
 - [40] J. Mori and V. Mahalec, “Planning and scheduling of steel plates production. Part II: scheduling of continuous casting,” *Computers & Chemical Engineering*, vol. 101, pp. 312–325, 2017.
 - [41] X. Xie, Y. Zheng, L. Tang, and Y. Li, “Multiple crane scheduling in a batch annealing process with no-delay constraints for machine unloading,” *Applied Mathematical Modelling*, vol. 49, pp. 470–486, 2017.
 - [42] G. J. Hahn and M. Brandenburg, “A sustainable aggregate production planning model for the chemical process industry,” *Computers & Operations Research*, vol. 94, pp. 154–168, 2017.
 - [43] S. L. Jiang, Z. Zheng, and M. Liu, “A preference-inspired multi-objective soft scheduling algorithm for the practical steelmaking-continuous casting production,” *Computers & Industrial Engineering*, vol. 115, pp. 582–594, 2018.
 - [44] Z. Zeng, M. Hong, Y. Man, J. Li, Y. Zhang, and H. Liu, “Multi-object optimization of flexible flow shop scheduling with batch process - consideration total electricity consumption and material wastage,” *Journal of Cleaner Production*, vol. 183, pp. 925–939, 2018.
 - [45] J. An, S. Shu, M. Wu, and Y. Xiong, “A three-dimensional simulation system for truck crane hoisting based on cloud computing,” *Intelligent Automation & Soft Computing*, vol. 19, no. 2, pp. 133–142, 2013.
 - [46] J. Li, X. Anjun, and Z. Xuesong, “Simulation-based solution for a dynamic multi-crane scheduling problem in a steel-making shop,” *International Journal of Production Research*, vol. 58, no. 22, pp. 6970–6984, 2020.
 - [47] H. Tamaki, K. Shinzo, and M. Hajime, “Simulation-based optimization model and metaheuristic solution of multiple crane scheduling problems,” vol. 2, pp. 1469–1474, in *Proceedings of the 2004 IEEE International Conference on Systems, Man & Cybernetics*, vol. 2, pp. 1469–1474, IEEE, The Hague, The Netherlands, October 2004.
 - [48] T. Zhang and R. Oliver, “Simulation-based overhead-crane scheduling for a manufacturing plant,” in *Proceedings of the 2003 Winter Simulation Conference*, pp. 2633–2642, WSC), Washington, DC, USA, December 2013.
 - [49] L. Wang, S. Wang, Y. Xu, G. Zhou, and M. Liu, “A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem,” *Computers & Industrial Engineering*, vol. 62, pp. 917–926, 2012.
 - [50] Y. Yan and G. Wang, “A job shop scheduling approach based on simulation optimization,” in *Proceedings of the 2007 I.E. IEEM*, Singapore, December 2007.
 - [51] R. Pérez-Rodríguez and A. Hernández-Aguirre, “A hybrid estimation of distribution algorithm for flexible job-shop scheduling problems with process plan flexibility,” *Applied Intelligence*, vol. 48, pp. 3707–3734, 2018.
 - [52] M. Gen, Y. Tsujimura, and E. Kubota, “Solving job-shop scheduling problems by genetic algorithm,” *International Journal of Computational Science and Mathematics*, vol. 2, pp. 1577–1582.
 - [53] I. Kacem, S. Hammadi, and P. Borne, “Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic,” *Mathematics and Computers in Simulation*, vol. 60, no. 3, pp. 245–276, 2002.
 - [54] M. Meilä, K. Phadnis, A. Patterson, and J. Bilmes, “Consensus ranking under the exponential model,” in *Proceedings of the 22nd Conference Uncertainty and Artificial Intelligence*, pp. 285–294, Vancouver, Canada, July 2007.
 - [55] R. Sargent, “Verification & validation of simulation models,” in *Proceedings of the 2007 Winter Simulation Conference*, Washington, DC, USA, December 2007.
 - [56] J. Adams, E. Balas, and D. Zawack, “The shifting bottleneck procedure for job shop scheduling,” *Management Science*, vol. 34, no. 3, pp. 391–401, 1988.
 - [57] H. Fisher and G. L. Thompson, “Probabilistic learning combinations of local job-shop scheduling rules,” in *Industrial Scheduling*, J. F. Muth and G. L. Thompson, Eds., pp. 225–251, Prentice-Hall, Englewood Cliffs, NJ, USA, 1963.
 - [58] S. Lawrence, *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement)*, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, USA, 1984.
 - [59] D. Applegate and W. Cook, “A computational study of the job-shop scheduling problem,” *ORSA Journal on Computing*, vol. 3, no. 2, pp. 149–156, 1991.
 - [60] R. H. Storer, S. D. Wu, and R. Vaccari, “New search spaces for sequencing problems with application to job shop scheduling,” *Management Science*, vol. 38, no. 10, pp. 1495–1509, 1992.
 - [61] T. Yamada and R. Nakano, “A genetic algorithm applicable to large-scale job-shop problems,” in *PPSN*, vol. 2, pp. 281–290, Berlin, Germany, September 1992.
 - [62] J. De Bonet, C. Isbell, and P. Viola, “MIMIC: finding optima by estimation probability densities,” *Advances in Neural Information Processing Systems*, vol. 9, 1997.
 - [63] S. Baluja and S. Davies, “Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space,” in *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 30–38, San Francisco, CA, USA, April 1997.

- [64] M. Pelikan, D. Goldberg, and E. Cantú-Paz, *Linkage Problem, Distribution Estimation, and Bayesian Networks*, The university of Illinois, Illinois Genetic Algorithm Laboratory, Urbana-Champaign, IL, USA, 1998.
- [65] N. Srinivas and K. Deb, "Muultiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [66] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [67] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.