Research Article

Schedule-Allocate and Robust Sequencing in Three-Machine Robotic Cell under Breakdowns

Bahareh Vaisi,1 Hiwa Farughi,2 and Sadigh Raissi3

1Young Researchers and Elite Club, Islamic Azad University, South Tehran Branch, Tehran, Iran
2Department of Engineering, University of Kurdistan, Sanandaj, Iran
3School of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

Correspondence should be addressed to Bahareh Vaisi; st_b_vaisi@azad.ac.ir

Received 22 April 2020; Revised 15 July 2020; Accepted 16 September 2020; Published 31 October 2020

Academic Editor: Georgios Dounias

The purpose of this paper is to model two problems comprising schedule-allocate (in case of producing identical parts) and sequencing of parts (in case of producing different parts). The first model is used for minimizing the cycle time and operational cost, and the second one for minimizing both the mean and standard deviation of the total production cost as well the cycle time, in an unreliable three-machine robotic cell which confronted with many uncertainty factors. In the current article, mathematical modelling and simulation-based optimization method have been presented to schedule-allocate similar parts and trace the optimal sequence of different parts. Several solution procedures, including epsilon-constraint method and multiobjective particle swarm optimization algorithm, for identical parts case and response surface methodology for different parts case are applied. The results derived from solving numerical examples revealed some advantages in terms of time to attain the optimal solution.

1. Introduction

Applying robots upturn the degree of automation and promote system productivity. In a general definition, a robotic cell consists of robot system, computer numerical control (CNC) machines, and other relevant systems. Robots are responsible for loading, unloading, and displacement of the parts within the cell.

Because of the arising complexity on mathematical modelling, machine breakdowns so far have relaxed on scheduling and sequencing optimization process while this may result in changing the optimal solutions. Thus, there is noticeably a need for further study in the areas of optimizing scheduling-allocation of similar parts and determining a sequence of parts in multiple-part-type case that would address the real management objectives.

Multicriteria scheduling was surveyed in a study and the most important results appeared in [1]. In a study, Gultekin et al. [2] relaxed the flow shop assumption in a three-machine robotic cell, and by considering the processing times of the parts on the machines as decision variables, they proposed new lower bounds for the one-unit and two-unit robot move cycles under the new problem domain for the flow shop-type robot move cycles. In a flow shop system, for producing the parts, each part meets all machines in the same sequence. In a flow shop robotic cell, one-unit cycles were proved to be optimal for three-machine cells. A one-unit robot move cycle is a robot move cycle that by starting the system from an initial state (input buffer), the robot performs each activity exactly once, and when the robot returns to the initial state, the cycle ends. In the one-unit robot move cycle, exactly one part is produced [2]. A bicriteria scheduling model that jointly minimizes the cycle time and total manufacturing cost in a robotic cell of two and three identical CNC machines producing identical parts was presented by Gultekin et al. [3]. They declared that manufacturing costs for the first time is being studied as an objective function in robotic cell scheduling problems. Alagöz and Azizoğlu [4] presented an optimization algorithm to minimize the stability measure considering the efficiency measure in its minimum level for rescheduling in a parallel machine environment under machine eligibility.
constraints. Minimization of total manufacturing cost and total weighted completion time were two objectives considered in [5, 6]. In some studies, such as [7], a list of qualitative and quantitative objectives was targeted. Again in [8], manufacturing cost and total weighted tardiness as objectives were minimized for a flexible manufacturing system. Restrepo and Balakrishnan [9] presented a fuzzy logic-based methodology. Shortest processing time and weighted earliest due date were the objectives to schedule a robot-centred cell. In [10], cycle time and total robot travel distance in a robotic cell with processing time windows and non-Euclidean travel times were minimized to schedule the cell. In studies [11, 12], the authors addressed a dynamic scheduling with multiobjective optimization in robotic cells. While Mansouri et al. [13] analysed the trade-off between minimizing make span and total energy consumption in a two-machine flow shop, Ghardiri Nejad et al. [14] considered the trade-off between scheduling of cell processes and the production cost. In another study, a fuzzy multiobjective flexible cell scheduling with two objectives including total cost of production system and total delivery tardiness of jobs under green and energy-efficient strategy was addressed by Far et al. [15]. Simultaneous analysis of throughput and cost of quality for inspection scenarios in a two-machine robotic cell was done by Foumani et al. [16]. The authors claimed that studying robotic cells with controllable inspection times is for the first time.

Scheduling in most of the researches in the literature was done based on a single criterion. Basically the criterion is cycle time minimization or output rate maximization. In papers [17–19], the criterion was maximizing throughput rate. For robotic cell scheduling problems, a classification was done and discussed based on machine environment, processing restrictions, and objective function and provided in [20]. In [21], robotic cells with stochastic processing times which are common in semiconductor manufacturing were considered. Online scheduling for improving output was offered in this study. Multidegree cyclic robotic flow shop cell scheduling problem in a cell with multiple single gripper robots to maximize the throughput rate was considered in [22]. An ant colony algorithm was proposed to optimize the problem. In [23], the criterion was maximizing total profit in a fully automated robot spot-welding line which has prescheduled breaks (lunch and tea breaks). In this study, the authors designed the robotic line so that tool replacements can be performed during the abovementioned breaks. Makespan minimization for the robotic cell with controllable processing times was investigated in [24], through a novel genetic algorithm. Foumani et al. [25] minimized the partial cycle time for the robotic cell with an inspection process under different inspection scenarios. Cyclic scheduling in an m-machine robotic cell was studied in [26, 27]. A common approach was minimizing cycle time objective which is equal to maximizing the throughput rate. In [28], an improved discrete bacterial foraging optimization algorithm was proposed to minimize the cycle time in a two-machine robotic cell with sequence-dependent setup times. Cyclic scheduling is defined as the production of parts by repeating a specific sequence of robot movements. The purpose of scheduling in [29] was to determine a cyclic scheduling of jobs performed by machines and an operator, in an automated cell which minimizes the cycle time.

From the part-type standpoint, there are three different groups of studies on the robotic cell’s part sequencing reported in the literature. In the first group, the problem is solved based on the well-known Gilmore and Gomory algorithm, such as in [30, 31]. In the second group, the problem is formulated based on TSP problem, such as in [17, 32–34]. Studies [35–40] belong to the third group. In this group, the authors employ heuristics or some special algorithms to solve the problem. In a new study of “Batur and Erol [41],” a metaheuristic approach with simulated annealing tool was applied to schedule a two-machine robotic cell with inconstant processing time of part.

Simulation-based optimization has not been applied to solve the problem of determining part input sequence to the machines heretofore.

One of the basic assumptions in the previous studies in the field of a robotic cell is that robots and machines never experience breakdown and never require maintenance. But due to the impossibility of this issue in the real world, in order to make more practical the robotic cell scheduling topic, the assumption of no failure in the machines was relaxed. On the contrary, breakdown and maintenance make the machines’ availability subject to random processes of lifetime. Some previous studies focused on maintenance and reliability appraisement in flexible manufacturing systems, such as [42–44]. Moreover, in none of the previous studies, availability of the robotic cell is advised as a constraint because in all previous studies, machine/robot is supposed to be available. Additionally, the review of the related literature reveals that applying design of experiment (DOE), response surface methodology (RSM), and multicriteria decision making (MCDM) methods together as tools to establish a robust sequence for parts’ entrance to robotic manufacturing cell has not been done before. Therefore, to bridge this gap, scenario concept is being developed. Another novelty of the present research partly lies in consideration of scenario concept for parts input sequencing to the robotic cell where $S_k$ cycle is planned as robot’s move cycle in the 3-machine robotic manufacturing cell. $S_k$ represents a specific robot move cycle in a three-machine robotic cell. The complete process of this move cycle will be described in the Problem Statement section.

In this study, there is uncertainty in robot’s waiting and cycle time considering machine’s stochastic breakdown. Furthermore, in case of producing dissimilar parts, different sequencing patterns for the parts entrance to the robotic cell make the robotic cell uncertain. The rest of the paper is structured as follows: first, the general information about the problem, definitions, and notations are stated. Then, a mathematical model for solving the problem in cases of producing identical parts, assumptions, and solution procedures for the problem are developed. With the purpose of overcoming different uncertainty, a computer simulation (CS) model in the case of producing different parts is developed. Afterwards, numerical examples for testing the applicability of the mathematical model (identical parts) and
simulation-based optimization approach (different parts) are introduced. The results for the two cases of producing similar and dissimilar parts through solution procedures are demonstrated, and in the case of producing different parts, the simulation model is validated, and also RSM and MCDM methods are detailed to optimize robustly the interactions between robotic cell inputs and outputs. Besides, statistical analysis for comparing the solution methods through optimal schedules and discussion based on optimal part input sequences are revealed. Finally, the conclusion and suggestions for upcoming studies are prepared.

2. Problem Statement

Assume a common robotic cell towards organizing rapidly fluctuating customer appeals [42]. A group of machines, such as CNCs, which use robot or robots (to transport materials or parts) and do useful work, form a robotic manufacturing cell. Applying a robotic cell could reduce production cost and promote quality features. Such cells, as described before, consist of one or more machines and are supported by robots. A typical in-line three-machine robotic cell is shown in Figure 1.

In the production system, there are three identical CNC machines that each of which has no priority in operation. This means that all machines are capable of performing the required operations at the same time. A single gripper robot is responsible for loading and unloading the parts and also doing the movement of fresh material basket to the selected machine and from the machine to the products basket. Single gripper robots can hold one part at a time. Generally, in a 3-machine cell, six cycles, nominated as $S_1$, $S_2$, $S_3$, $S_4$, $S_5$, and $S_6$ cycles, may be applied for part movements [2, 17]. The present study concentrated on the $S_6$ cycle since it is a one-unit robot move cycle, widely known and commonly used [2, 19], but less attention has been paid in the modelling due to more complexity than the other cycles.

The overall process of $S_6$ cycle describes as follows: in the $S_6$ cycle, primarily, the robot takes its place in front of the input buffer. Then, the following operations are followed by the robot successively: (1) picking up part $i$, (2) moving to the first machine, (3) loading the first machine, (4) moving to the third machine, (5) waiting for completing the part $(i - 2)$ process (if necessary), (6) unloading the part from the third machine, (7) moving the product to the output buffer, (8) loading the output buffer, (9) moving to the second machine, (10) if necessary, waiting until the completion of the process on part $(i - 1)$, (11) unloading from the second machine, (12) moving the part to the third machine, (13) loading the part on the third machine, (14) moving to the first machine, (15) if necessary, waiting until the completion of the process on part $i$, (16) then unloading the part from the third machine, (17) moving to the second machine, (18) loading part $i$ on the second machine, and (19) finally, turning back to the input buffer. $A_{i0}$, $A_{34}$, $A_{33}$, $A_{13}$ implies the activity sequencing of $S_6$ cycle as abovementioned; $A_{pq}$ is the robot’s activity sequence from station $p$ to station $q$ for $p = 0, 1, 2, 3$ and $q = 1, 2, 3, 4$ [2, 17].

This robotic cell is studied in two situations including the production of identical parts or different parts.

2.1. Definitions. In real manufacturing process, scheduling is one of the key factors at the operational level that affects production efficiency, quality, and cost. Optimal scheduling in robotic cells and performing maintenance activities to balance various objectives, including cost and time, taking into account system availability, increase productivity in these manufacturing systems.

Basic definitions for scheduling-allocation in the three-machine robotic cell under breakdown are very close to those in two-machine cell. By introducing a control limit to trigger PM actions for each machine ($C$), maintenance points and maintenance actions such as PM or CM are determined as explained in [45]. In the case of producing different parts as well, the $S_6$ move cycle is measured.

2.2. Notations. Notations for scheduling-allocation and sequencing in the three-machine robotic cell are introduced. First, the following parameters are presented besides/instead of their equivalent in [45]:

- $c$: the processing time for a part on the $3^{rd}$ machine
- $P$: the total processing time for a part (in 3-machine problem $P = a + b + c$)
- $T_{S6}$: cycle time based on the $S_6$ robot move cycle
- $t_{ij}$: the processing times of part $i$ based on the percentage of operations done by each machine $j$; $i = 1, \ldots, n$

To define the sequence of parts in the proposed unreliable robotic cell by minimizing robustly the cycle time ($S_6$ cycle) and total production cost, the total production cost is formulated based on "Selim Akturk and Gurel [46]" description. In many cases such as the current one, tooling cost considered to be a constant cost and has no effects on optimization process. Cycle time is the time a manufacturing system spends on producing one part. There are succeeding parameters and variables in the mathematical model for computing the total production cost:

- $C_o$: machining cost ($$/min)
- $C_{PM}$: cost of each preventive maintenance visit ($$/each visit) ignoring setup cost
\(C_{\text{TOOL}}\): cost of tool ($/tool)—tool replacement is forbidden in an operational cycle

\(p_j\): expected down rate of machine \(j\)

\(H\): observation period

\(n\): the number of produced parts by type based on the percentage of operations done by each machine during the simulation period

\(N\): the number of throughout products in the simulation period

\(F\): total cost ($/times unit)

\(T_{\text{TTF}}\): time to failure (times unit)

\(T_{\text{TTR}}\): time to repair (times unit)

\(\lambda\): failure rate

\(\mu\): repair rate

3. Schedule-Allocate of Identical Parts in Three-Machine Robotic Cell

3.1. Problem Formulation. According to the notations in Problem Statement, the cycle time formula based on robot’s \(S_6\) move pattern comes here. It should be noted that \(\varepsilon\) is the loading/unloading time and \(\delta\) is the travel time of the robot between two successive machines:

\[
T_{S6} = 8\varepsilon + 12\delta + \text{Max}\{0, W_1, W_2, W_3\},
\]

\[
W_1 = a - (4\varepsilon + 8\delta),
\]

\[
W_2 = b - (4\varepsilon + 8\delta),
\]

\[
W_3 = c - (4\varepsilon + 8\delta).
\]

The following model is proposed for scheduling-allocation of parts:

\[
\begin{align*}
\text{Min } F & = \left( \sum_{j=1}^{3} C_{o} \sum_{i=1}^{n} t_{i} O_{ij} + C_{PM} P_{ij} \right) + C_{\text{TOOL}}, \\
\text{Min } T_{S6} & = 8\varepsilon + 12\delta + W, \\
\text{s.t.,} & \\
& a = \left( \sum_{i=1}^{n} t_{i} O_{i1} \right) + W_{R1}, \\
& b = \left( \sum_{i=1}^{n} t_{i} O_{i2} \right) + W_{R2}, \\
& \bar{c} = \left( \sum_{i=1}^{n} t_{i} O_{i3} \right) + W_{R3}, \\
& W = \text{Max}\{0, a - [4\varepsilon + 8\delta + W_{R1}], b - [4\varepsilon + 8\delta + W_{R2}], \bar{c} - [4\varepsilon + 8\delta + W_{R3}]\}, \\
& W_{Rj} = (t_{j,PM} R_{j,PM}) + (t_{j,CM} R_{j,CM}) - (t_{j,PM} R_{j,PM}) (t_{j,CM} R_{j,CM}), \\
& P_{ij} = \left( A t_{i} U_{ij} + A W_{Rj} + \frac{B U_{ij}}{t_{i}^{k-1}} + \frac{BW_{Rj}}{t_{i}^{k}} \right) \frac{O_{ij}}{T_{S6}} C_{PM}, \\
& U_{ij} = \frac{\left( t_{i} * O_{ij} \right)}{\left( T_{S6} \right) \text{AV} (\infty)}, \\
& \text{AV} (\infty) = \frac{4\mu^3 + 12\lambda \mu^2 + 12\mu^3}{4\mu^3 + 12\lambda \mu^2 + 12\mu^3 + 6\lambda^3}, \\
& O_{i1} + O_{i2} + O_{i3} = 1, \\
& 0 < C < H,
\end{align*}
\]
The reported literature reveals that many approaches such as [46] focused on minimizing total production cost consisted of machining, tooling, and preventive maintenance costs. In many cases as this study, tooling cost is considered to be constant value and has no effects on optimization process. Minimizing the total cases as this study, tooling cost is considered to be constant value and has no effects on optimization process. Minimizing the total machining, tooling, and preventive maintenance costs. In many cases, this study is focused on minimizing total production cost consisted of machining, tooling, and preventive maintenance costs.

The constraints related to the maintenance are equations (8)–(11), (13), and (15). The explanations of equations (8)–(10), (13), and (15) are based on the descriptions of equations (7), (8), (11)–(13) one-to-one in [45], and redefined in accordance with $S_6$ robot move cycle.

As we may know, availability ($AV(t)$) is the probability that a component is functioning at time $t$ by ignoring failure and repair history during [0, t]. Accordingly, steady-state availability ($AV(\infty)$) is calculated based on equation (16):

$$AV(\infty) = \lim_{t \to \infty} AV(t). \quad (16)$$

Equation (11) expressed the steady-state availability for a system entailed three similar components; each one has exponential distribution lifetime, and two maintenance crews have responsibility to repair the failed machine based on exponential distribution repair time without any preemptions. Proof is accessible in [47]. The decision for allocating the operations to the machines is made through equation (12).

### 3.2. Assumptions 1

In addition to the assumptions in [45], by considering three machines in an in-line robotic cell, the basic assumptions for the current study are summed up as follows:

1. The machines in the robotic cell are CNC machines. They are capable of performing all the operations.
2. Every part to be processed has a number of operations to be completed on the machines.
3. There is no buffer storage between the machines, and each part is either on a machine or being handled by the robot.
4. Produced parts are identical.
5. Production time is continuous.
6. No preemption is allowed in the processing of any operation.
7. The focus is on the $S_6$ cycle in the 3-machine robotic cell.

### 3.3. Solution Procedures

Since in the process of solving multiobjective problems, rather than optimal points, non-dominant set points are reasoned, so as to attain the set, particular procedures should be done. Until now, many methods have been presented to deal with them [1]. One of the prevalent methods is epsilon-constraint, which changes the problem into a one dimensional problem [48]. In this section, epsilon-constraint approach and a multiobjective particle swarm optimization (MOPSO) algorithm are applied to make different sets of nondominated solutions for the proposed model in Problem Formulation.

#### 3.3.1. Epsilon-Constraint Approach

The epsilon-constraint approach represented by $\epsilon (F/T)^S$ is applied to generate a representation of the Pareto optimal solutions. Accordingly, minimizing $F$ as an objective gave an upper bound on $T_{S_6}$. The cycle time is acquired as a constraint, due to the independency of $S_6$ cycle time to the allocation of operations on the machines (equation (3)). $\epsilon (F/T)^S$ is considered to determine the appropriate settings for the processing time on the machines by minimizing total cost for $S_6$ cycle time ($T_{S_6}$). According to these settings, it is possible to mention the total cost as a function of the $S_6$ cycle time, which means that $F$-compose-$T_{S_6}$ of processing time is made. Consequently, for any given total cost/$S_6$ cycle time, determining the corresponding $S_6$ cycle time/total cost and the processing times of the parts on the machines is practicable.

In this study, to produce the Pareto optimal solutions, commercial software (GAMS) is utilized and the exact Pareto front of the bicriteria scheduling-allocation problem is addressed. The formulation of epsilon-constraint problem for the $S_6$ cycle is a mixed-integer nonlinear programming (MINLP) model which simultaneously allocates the operations to the machines and establishes processing time values for them by considering breakdowns caused by machine failures. The formulation is presented as follows:

$$\epsilon (F/T)^S: \min \left( \sum_{j=1}^{3} \sum_{i=1}^{n} C_{ij} t_{ij} O_{ij} + C_{PM} P_{ij} \right) + C_{\text{tool}},$$

$$t_{ij} \leq \text{EPSM},$$

s.t.

- Equation (4),
- Equation (15).

(17)
Lower bound’s quality for a minimization model is a crucial part of finding a best solution. Mostly, relaxing some constraints and solving an easier problem can compute the lower bound [17]. Here, the lower bound commences finding the best attainable value of \( f_1 \) in the absence of \( f_2 \left( f_{1}^{\text{min}} \right) \) and the best attainable value of \( f_2 \) in the absence of \( f_1 \left( f_{2}^{\text{min}} \right) \). These values are the lower bounds for the objective functions (cycle time and total cost) of the problem.

Due to the inefficiency of exact methods in terms of computational time, applying metaheuristic algorithms especially to solve big-size problems will be desirable. In the next step, we put MOPSO heuristics based entirely on Pareto dominance concepts to solve the model [49].

3.3.2. MOPSO. Before describing the MOPSO algorithm, the particle swarm optimization (PSO) needs to be presented. Short PSO is a computational method to optimize a problem by iteratively improving a candidate solution, concerning a certain measure of quality. The similarity of PSO with evolutionary algorithms makes clear the notion of using a Pareto ranking scheme as a way to extend the approach to handle multiobjective optimization problems, [50].

The fitness function value for algorithm is given identically as the determined score for the problem. After generating some predefined particles in each iteration, each particle should be moved based on other particles situations. Two types of solutions must be updated in each iteration of this algorithm. The best permutation experienced by each particle is denoted by \( PBest \), while the best experienced arrangement by all particles is denoted by \( GBest \). Particles movement direction and their final position in each iteration are calculated according to the following equations [51]:

\[
V_i(t+1) = wV_i(t) + c_1 r_1 \left( PBest_i(t) - X_i(t) \right) + c_2 r_2 \left( GBest(t) - X_i(t) \right), \tag{18}
\]

\[
X_i(t+1) = X_i(t) + V_i(t+1), \tag{19}
\]

where \( w \) is the inertia weight and is employed to control the impact of the previous history of velocities on the current velocity of a given particle. \( c_1 \) and \( c_2 \) are the learning factors and are usually constants. To deal with MOPSO, few changes have to be made to PSO. The purpose of PSO is to find one best solution, while objective of MOPSO is finding a set of different solutions (Pareto optimal set). To achieve this, nondominated solutions obtained in each iteration are stored in an archive. The evaluation of archival nondominated solutions is done to guarantee diversity and spacing measures on the Pareto front [52]. Diversity and spacing metrics as comparison metrics are defined in the Results section. The pseudocode of the MOPSO algorithm is divulged in Figure 2.

4. Sequence of Different Parts in Three-Machine Robotic Cell

4.1. Proposed Simulation Model. By considering the features of main parameters in Problem Statement, the cycle time formula in equation (1) could also be presented in the following equation precisely:

\[
T_{SE} = 8\varepsilon + 12\delta + W. \tag{20}
\]

Parameters and variables were introduced in the Problem Statement section for computing the cost; hence, the total cost per operational cycle could be formulated by the following equation:

\[
F = C_o \sum_{j=1}^{n} n t_j + HC_{PM} \sum_{j=1}^{3} P_j + N C_{\text{TOOL}}. \tag{21}
\]

Currently, the robotic cells require complex choices regarding sequencing topics caused by uncertainty which rises in different parameters such as time to failure, time to repair, and cycle time that can be supported efficiently by CS models. CS has been established since the early 1960s, and it may be one of the most commonly used analytical tools of management science [53]. In order to investigate the robotic cell, we integrate CS with linear programming (LP) optimization approach to control different sources of uncertainties. Uncertainty in sequencing of arriving the parts in addition to randomly distributed cycle time because of considering stochastic machine breakdowns face this robotic cell with a lot of complexity. It is too difficult to optimize such systems only by using the mathematical programming methods. In these cases, there is definitely no approach to help us efficiently without applying CS models. “As discussed by Fu [54],” CS has been proved to be a successful interface between operations research and computer science.

For simulation purposes, the enterprise dynamics (ED) tool is utilized to model the sequence of parts in this robotic manufacturing cell. Although using simulation tool in scheduling for the robotic cell has not been done so far, a comprehensive survey of scheduling on flexible manufacturing systems and analyzing future trend that employed simulation techniques as the analysing tool has been done in [55]. Literature review reveals some latest studies on the application of simulation in manufacturing systems such as simulating mass and small batch production [56], incremental sheet forming [57], and product processing in a mixed flow assembly line [58]. Kikolski [59] investigated some production scenarios using simulation models, built in a Tecnomatix Plant Simulation system. Simulation-based metaheuristic optimization was done by Diaz et al. [60]. A press shop was simulated using Arena [61]. To describe the operation of a job shop, a simulation model was developed [62]. Simulation-based optimization framework was the contribution for supporting strategic maintenance in the production system [63]. A combination of simulation and bow-tie modelling was proposed to study the disruption risks in oil and gas supply chains [64]. Additionally, a comprehensive review of papers between 2002
and 2013 concerning simulation in manufacturing has done in [65]. “Mourtzis et al. [66]” through a significant paper surveyed the major milestones in the development of simulation tools. They studied recent industrial practices of simulation, evolution, advances, and its future trends in the context of manufacturing system.

Figure 3 displays the layout of the CS model for the robotic cell. Here, three different products flow inside the system by the single gripper robot based on the S6 cycle in turn, by means of ED elements.

In order to reach an unbiased estimation, the simulation model is run for a long time (more than 10000 hours) with 25 replications, and data are cleared after a warm-up period taking 50 hours. In a simulation software such as ED, the system has to warm-up for some time before it reaches a balanced situation. This is described as the warm-up period. Warm-up period is usually equalled to 10% of observation period in Experiment wizard settings. The results are gathered during the observation period. An observation period starts at moment zero (the default value) or when the warm-up period has ended.

Hence, due to attaining more consistency in decision-making affected from different sources of uncertainties, four performance measures (PFMs) are defined as the CS model outputs. These set of objective functions look for solutions which minimize fluctuations of the two simulation outputs in terms of their standard deviations.

\[ Y_1 : \text{expected } S_6 \text{ cycle time in terms of mean value} \]
\[ Y_2 : \text{expected operating cost per part in terms of mean value} \]
\[ Y_{SD}^1 : \text{standard deviation of } S_6 \text{ cycle time} \]
\[ Y_{SD}^2 : \text{standard deviation of part operating cost} \]

Numerous researches advised on robust optimization especially when significant uncertainties arise from different sources. For the first time, “it was introduced by Soyster [67],” the category of robustness to the operations research. Actually, robustness leads to optimal solutions in uncertain circumstances to estimate a moderately consistent solution.

4.2. Assumptions 2. Since in robotic cells, scheduling and sequencing were usually done in deterministic atmosphere, few studies have been done on this issue under stochastic environments. Accepted that if machine’s failure/repair rates are constant over time, one may consider that time to failure and repair time deploy from an exponential distribution, otherwise, an alternate distribution such as Weibull, Gamma, or other density functions might be addressed for their modelling. In addition to the first seven assumptions in [68], the basic assumptions for the current study are summarized as follows:

1. An in-line robotic cell is available with three machines and a single gripper robot
2. Every part to be processed has a number of operations to be completed on the machines
3. Sequence of operations followed through S6 cycle for a 3-machine robotic manufacturing cell
4. Machines meet random failure and require maintenance.
5. Time to failure and Time to repair follow any specified statistical density function with recognized reliable parameters.
6. The processing time of each part per machine, without being influenced by stopovers, has been determined ultimately.
7. Manufacturing each part needs a diversity of operations to be done on the machines, by percentage, some of which are done on first machine, the rest are done on the subsequent machines, correspondingly.
5. Numerical Examples

5.1. Identical Parts Case. In this section, three Test Problems comprising different groups of operations with their processing times in a 3-machine robotic cell for producing identical parts are included. The epsilon-constraint and MOPSO methods are applied for the designated Test Problems. The Test Problems are randomly generated and given in Table 1. The parameters and user-defined values for the deliberated robotic cell are presented in Table 2. As described in [45], and in addition to the Problem Statement section’s notations, $A$, $B$, and $K$ are the coefficients related to PM index function; and $t_{j,PM}$ and $t_{j,CM}$ are sequentially the duration of a PM and CM visit for machine $j$. The parameters were set after a number of trials, based on the CNC machines’ physical characteristics and their average failure/repair rates, as discussed by respective authors in [46]. It should be noted that there is no tool replacement for the operations while performing $S_6$ cycle.

5.2. Different Parts Case. Two well-defined examples derived from previous works [33, 68] are applied to examine the relevancy to simulation-based optimization approach.

Example 1. Three dissimilar products are manufactured with consistent $t_{j,s}$. For the blue part, $t_{B}$ equals 57, the red part’s $t_{R}$ is 84, and the Purple part has 87 minutes processing time ($t_{P}$). $\epsilon$ and $\delta$, respectively, are 1 and 2 minutes.

Example 2. A robotic cell is contemplated producing a single product with 57 minutes processing time for a blue coloured product. Here, $\epsilon$ and $\delta$ are, respectively, 1 and 2 minutes.

| Test problem | Group size | Example no. | Processing times |
|--------------|------------|-------------|------------------|
| 1 (small size) | (5–10) operations | 1 | 10, 8, 7, 4, 3. |
|               |            | 2 | 10, 7, 13, 8, 5, 2, 5. |
|               |            | 3 | 7, 4, 3, 7, 4, 8, 10, 3, 7. |
| 2 (large size) | (10–18) operations | 4 | 10, 8, 7, 4, 3, 7, 4, 8, 10, 3. |
|               |            | 5 | 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 7, 4, 8. |
|               |            | 6 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 7, 4, 8, 10, 3, 7. |
|               |            | 7 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7. |
|               |            | 8 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 5, 2. |
| 3 (big size) | (18–50) operations | 9 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 5, 2, 3. |
|               |            | 10 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 10, 8, 7, 4, 3. |
|               |            | 11 | 10, 7, 13, 8, 5, 2, 5, 4, 10, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 10, 8, 7, 4, 3. |
|               |            | 12 | 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 2, 10, 8, 7, 4, 3, 7, 4, 8, 10, 3, 1, 2, 3, 7, 9. |
Table 2: Features of necessary parameters.

| Parameter | Value |
|-----------|-------|
| $C_{PM}$  | 35 $\text{in } $ |
| $C_{TOOL}$ | 45 $\text{in } $ |
| $B$       | 80 |
| $C$       | 7 |
| $\mu$     | 2 |
| $\epsilon$| 1 |
| $t_{PM}$  | 7 |
| $C_0$     | 50 $\text{in } $ |
| $A$       | 40 |
| $K$       | 4 |
| $H$       | 15 |
| $\lambda$ | 3 |
| $\delta$  | 2 |
| $t_{CM}$  | 10 |

Table 3: Tabulated main parameters estimation.

| Parameter                      | Value                          |
|--------------------------------|--------------------------------|
| Preventive cost; $C_{PM}$ in $|$ per visit | 35 $\text{in } $ |
| Machining cost; $C_0$ in $|$ per minutes | 50 $\text{in } $ |
| Tools cost; $C_{TOOL}$ in $|$ | 45 $\text{in } $ |
| Time to failure density function | Negative exponential with mean of 10 hours |
| Time to repair density function  | Negative exponential with mean of 2 hours |

Figure 4: Continued.
Figure 4: Epsilon-constraint results for the first twenty iterations in Test Problems. (a) Example 1. (b) Example 2. (c) Example 3. (d) Example 4. (e) Example 5. (f) Example 6. (g) Example 7. (h) Example 8. (i) Example 9. (j) Example 10. (k) Example 11. (l) Example 12.
Table 4: Upper and lower bounds of the generated Pareto front in the Test Problems.

| Example no. | Bounds | Cost | $S_6$ cycle time |
|-------------|--------|------|-----------------|
| 1           | L      | 1676.517 | 32              |
|             | U      | 1721.089 | 70              |
| 2           | L      | 2550.039 | 32              |
|             | U      | 2599.119 | 70              |
| 3           | L      | 2735.448 | 32              |
|             | U      | 2813.112 | 70              |
| 4           | L      | 3255.757 | 34              |
|             | U      | 3327.174 | 72              |
| 5           | L      | 4200.631 | 40              |
|             | U      | 4237.42  | 78              |
| 6           | L      | 4866.29  | 44              |
|             | U      | 4940.935 | 82              |
| 7           | L      | 5201.575 | 47              |
|             | U      | 5242.042 | 85              |
| 8           | L      | 5631.056 | 49              |
|             | U      | 5706.741 | 87              |
| 9           | L      | 5705.439 | 50              |
|             | U      | 5746.889 | 69              |
| 10          | L      | 8147.492 | 66              |
|             | U      | 8186.135 | 85              |
| 11          | L      | 11450.86 | 88              |
|             | U      | 11481.44 | 107             |
| 12          | L      | 15781.14 | 116             |
|             | U      | 15809    | 135             |

6. Results

6.1. Identical Parts Case

6.1.1. Epsilon-Constraint Results. ($\text{Min } F-\text{Min } T_{S6}$) is acquired by solving the coded epsilon-constraint with solver GAMS-BARON. The codes were run on a portable PC with MS-Windows Vista, 3.0 GB of RAM, and 2.0 GHz Core 2 Duo CPU. Figure 4 distinctively plots twenty nondominated solutions of feasible region for objective functions in Test Problems (Table 1). Then, in Table 4, the upper (U) and lower (L) bounds of the generated solutions are presented. It should be noted that the solution method was run 10 times per example, and the small-size and large-size results were achieved by 1800 seconds average computational time. Because of the effect of the group size on the model’s computation time, producing the parts with more than 18 operations will be more than 3600 seconds computation’s time. So, the computational time was adjusted to 3600 seconds in the third test problem, and then the model was run 10 times per example.

The allocation of operations to the machines based on the proposed model is achievable as well. As an instance, regarding the machines’ generated degradation level, the allocation of operations to the machines for Example 11 (Table 1) is demonstrated in Table 5.

Example 11 includes 35 operations to manufacture a product. The best allocation of operations to the machines according to Table 5 was revealed on the basis of the machines’ generated degradation level. Consequently, the operations numbers 9, 10, and 18 with processing times 10, 10, and 10, respectively, are allocated to the third machine; the operations numbers 1, 2, 4, 5, 8, 11, 12, 13, 14, 15, 17, 24, 27, 33, and 34 with their processing times 10, 7, 8, 5, 4, 8, 7, 4, 5, 7, 8, 3, 7, 10, and 3, respectively, are allocated to the second machine; and the rest of the operations are allocated to the first machine. This allocation leads to an instantaneous ($\text{Min } F-\text{Min } T_{S6}$). This result determines the maximum output rate subsequently.

The reliability condition in comparison with the proposed model was also tested. Considering no breakdown for machines in the cell (i.e., an available robotic cell), the model was run again using commercial software (GAMS). The results for the first five Examples of Test Problems (Table 1) are shown in Table 6.

It is clear from the results that in case of considering a reliable cell (i.e., a robotic cell with reliable machines where no failures occur on them), an optimal point for the bicriteria model comes as a specific value, whereas in case of considering machine failure and repair in the robotic cell, a Pareto optimal (nondominated) solution is attainable.

6.1.2. MOPSO Results. To express the performance of the proposed model for solving the problem, the proposed MOPSO is examined on the designated Test problems (Table 1) by means of MATLAB software on a portable PC with MS-Windows Vista, 3.0 GB of RAM, and 2.0 GHz Core 2 Duo CPU.

(1) Solution Representation. The MOPSO solution chosen for the problem is a string of decimal numbers. The string as a particle consists of processing times of operations to produce part $i (t_i)$. The structure of the string is shown in Figure 5.

All solutions are randomly made in harmony with the above construct. This solution representation is divided into three sections. The first section is related to the allocation of operations to the machines which is illustrated in Figure 6. Dimensions of this particle in the form of a matrix are equal to (number of machines * number of operations). In each row, the maximum value of that row’s processing time is determined. For example, in the first row of the matrix, 0.4 is selected, which means that the cell corresponding to that value takes 1, and the rest of the cells in that row takes the value zero. This means that machine 3 is assigned to process operation 1.

The second and third sections of solution representation are associated with maintenance decisions for the machines. In these particles, to convert the decimal solution to the zero-one solution, if the value is above 0.5, then the solution is equal to 1, otherwise, it is equal to zero. Figures 7 and 8 show maintenance decisions. Based on the example in Figure 7, second and third machines need PM actions. The example of Figure 8 shows that first and third machines require CM operations.

(2) Results. Parameter setting for MOPSO is done by hand in a trial-and-error procedure considering convergence analysis. In this model, $\omega$ is set as 0.5; $r_1$ and $r_2$ are the random variables between 0 and 1; $c_1$ and $c_2$ are 1 and 2, respectively.
The repository size is usually set equal to the swarm (population) size. Here, the algorithm is run for a population size of 50 and a repository size of 50 particles. Number of grids per dimension is 5. The values of other parameters are tabulated in Table 7. It should be noted that grid inflation rate ($\alpha$) and leader selection pressure ($\beta$) are the two predefined parameters. If $\beta > \alpha$, there is a cumulative tendency to converge towards $PBest$ for the particle. Higher amount of deletion selection pressure ($\gamma$) leads to an increase in probability of member deletion. Deletion selection pressure is a predefined parameter as well.

Accordingly, the model is analysed using the MOPSO algorithm, and its results are provided in Table 8. In Figure 9, the best results of feasible region for objective functions in
Table 8: Summary of the objective functions results and elapsed time for solution methods.

| Example no. | Minimum cost (Epsilon-constraint) | Minimum cost (MOPSO) | Minimum $S_b$ cycle time (Epsilon-constraint) | Minimum $S_b$ cycle time (MOPSO) | Elapsed time (in seconds) |
|-------------|-----------------------------------|-----------------------|-----------------------------------------------|----------------------------------|--------------------------|
| 1           | 1676.517                          | 2281.962              | 32                                            | 32                              | 10.87                    |
| 2           | 2550.039                          | 3451.423              | 32                                            | 32                              | 12.34                    |
| 3           | 2735.448                          | 3871.081              | 32                                            | 32                              | 13.97                    |
| 4           | 3255.757                          | 4556.381              | 34                                            | 34                              | 14.04                    |
| 5           | 4200.631                          | 5909.742              | 40                                            | 41                              | 598.95                   |
| 6           | 4866.29                           | 6695.402              | 44                                            | 44                              | 620.67                   |
| 7           | 5201.575                          | 7181.277              | 47                                            | 47                              | 900                      |
| 8           | 5631.056                          | 7802.139              | 49                                            | 49                              | 1800                     |
| 9           | 5705.439                          | 8085.507              | 50                                            | 50                              | 3600                     |
| 10          | 8147.492                          | 11245.819             | 66                                            | 67                              | 3600                     |
| 11          | 11450.86                          | 16030.427             | 88                                            | 88                              | 3600                     |
| 12          | 15781.14                          | 22353.457             | 116                                           | 116                             | 3600                     |

Figure 9: Continued.
Figure 9: Continued.
Figure 9: MOPSO results of feasible region for objective functions in Test problems. (a) Example 1. (b) Example 2. (c) Example 3. (d) Example 4. (e) Example 5. (f) Example 6. (g) Example 7. (h) Example 8. (i) Example 9. (j) Example 10. (k) Example 11. (l) Example 12.

Table 9: Allocation of operations to the machines, for Example 11, based on the MOPSO algorithm.

| Machine no. | 1 | 2 | 3 |
|-------------|---|---|---|
| Allocated operations no. | 6, 10, 13, 16, 17, 18, 20, 21, 24, 25, 28, 29, 35 | 3, 7, 8, 9, 11, 22, 23, 26, 30, 32 | 1, 2, 4, 5, 12, 14, 15, 19, 27, 31, 33, 34 |

Table 10: Comparison metrics.

| Spacing metric (SM) | \( \bar{d} = \frac{\sum_{i=1}^{n} d_{i}}{n} \) |
|---------------------|----------------------------------|
| SM                  | \( \sqrt{\sum_{i=1}^{n} (\bar{d} - d_{i})^2/n - 1} \) |

| Diversity metric (DM) | \( \bar{D} = \sqrt{\sum_{i=1}^{n} \max(||x_{i} - y_{i}||)} \) |

Table 11: Computational results for comparison metrics in Test Problems.

| Test Problem | Example no. | DM  | SM  |
|-------------|-------------|-----|-----|
| Small size  | 1           | 177.215 | 0   |
|             | 2           | 157.999 | 0   |
|             | 3           | 213.085 | 0   |
|             | 4           | 273.805 | 0   |
|             | 5           | 283.449 | 0   |
|             | 6           | 320.881 | 0   |
|             | 7           | 286.043 | 0   |
|             | 8           | 320.69  | 0   |
|             | 9           | 326.375 | 0   |
|             | 10          | 386.960 | 0   |
|             | 11          | 475.863 | 0   |
|             | 12          | 568.589 | 0   |
Test Problems (Table 1) are represented. Close to the epsilon-constraint, the allocation of operations to the machines for Example 11 (Table 1) is established in Table 9.

In Example 11, the best allocation of operations to the machines according to Table 9 and the reason of the machines generated degradation level was disclosed. With the declared allocation, minimum cost and cycle time simultaneously will be practicable, and also maximum output rate is reasonable.

A comparison between the solution methods for Example 11, based on Table 8 shows that, however, minimum cycle time is 88 based on both solution methods, and minimum cost was achieved regarding epsilon-constraint method with the amount of 11450.86 $/min. In other Examples of this Test Problem also, minimum cost is realisable considering epsilon-constraint approach. However, MOPSO’s elapsed time is less than that of epsilon-constraint according to this Example.

(3) Comparison Metrics. Diversity and spacing metrics are provided for assessing the relative performance of MOPSO. The metrics definition is summarized in Table 10, where \( \|x_i - y_j\| \) is the Euclidean distance between two nondominated solutions \( x_i \) and \( y_j \) [69]. Comparison metrics were calculated for Test problems (Table 1) over 10 runs of the MOPSO algorithm. Table 11 presents the metrics obtained for small-, large-, and big-sized instances.

6.1.3. Statistical Analysis. Objective functions and elapsed time required by each method are defined as criteria in Table 8 so that the outcome of solution methods will be compared. To evaluate the performance of solution methods, epsilon-constraint method’s minimum cost was measured against that of MOPSO algorithm. \( F \) (i.e., cost), in exact solution, is consistently less than metaheuristic solution, regarding Figure 10. In the considered Test Problems (Table 1), difference between the objective function values is at least 605 and at most 6572 currencies.

A statistical significant difference between the solution methods’ behaviour in terms of cycle time.

From the standpoint of elapsed time, although the MOPSO algorithm has larger amounts on the first four Examples of Table 1 thoroughly, on other Examples, this behaviour has changed significantly and the epsilon-constraint method has bigger values.

6.2. Different Parts Case

6.2.1. Simulation Results. In order to compare the results from a common circumstance, the simulation model is run for the 2nd example, after relaxing machine failure. This accomplishment enables to equate simulation the simulated results with the analytic method. After running the model more than 10000 hours and replicating the model 25 times, the average \( S_6 \) cycle time and operating cost damped on 86.719 and 187.5, respectively, at 95% confidence level.

6.2.2. Model Validation Testing. In order to examine validation of the proposed simulation model, the Mann–Whitney hypothesis testing was applied to compare statistically significant difference between the mean of daily production on the simulation model after replicating the model 25 times and the analytic results obtained from equation (20). Hypothesis testing via applying the Minitab statistical package reported high amount of \( p \) value (greater than 0.84) which revealed that there is no significant difference to reject the simulation model; it will be concluded that there is no significant difference between results derived from the simulation model and the analytic method.

Accordingly, some scenarios through the DOE method are designed to find the best sequence amongst experiments. This method could be conducted to quantitatively evaluate the desired objective functions through estimating fitness function using RSM on the presence of different sources of uncertainty arising from failure/repair process, stochastic setup or process times, and so on.

The proposed CS model could easily be customized to any real cases, and there is no restriction on applying a special density function. Experts could straightforwardly extend the model to any other sources of uncertainty stand up from tools, operators, environment/climate, preemption, etc.
6.2.3. RSM. RSM is an assemblage of statistical design and numerical optimization techniques and is used for optimizing processes and product designs [70]. RSM is applied in situations where some input variables influence performance measures of the system. This methodology could make a relationship between the variables. For the proposed simulated robotic cell, first, scenarios are designed and variables are defined.

To design the scenarios, DOE is used, and the methodology of fractional factorial DOE is applied. DOE is a method to recognize the main factors in a process and, then, ascertain and fix the problem in a process. In real engineering situations, there are typically multiple factors involved, and it is usually significant to consider them together in the case they interact. A fractional factorial experiment is a disparity of the basic fractional design in a way that only a division of the runs is used [71]. This method reduces the number of experiments one has to perform, by studying multiple factors instantaneously. Factorial design works well when interactions between the variables are strong and important and where every variable contributes significantly.

Therefore, input variables are defined including \(x_1\), \(x_2\), \(x_3\), and \(x_4\) correspondingly as the percentage of part’s performed operations on the second machine, the percentage of part’s performed operations on the third machine, the colour of the first allocated Part to the second/third machines, and the sequence type. Table 12 presents input settings on Minitab and their three levels; the low, centre, and high; for designing random scenarios of Example 1 (Different Parts Case).

In Minitab, full factorial design conducted \(2^4 = 16\) experiments. By replicating the experiments two times and considering one experiment based on the centre points, total runs are 33. In order to estimate the four mentioned objective functions, the following first-order regression equations are developed considering only the main effects of various process parameters:

\[
\begin{align*}
\bar{Y}_1 &= 61.68304 - 7.0973x_1 - 2.861x_2 - 0.0766x_3 + 1.65x_4 + e_1, \\
\bar{Y}_2 &= 21344.487 - 2304.6875x_1 - 1138.4625x_2 + 7.58775x_3 + 314.425x_4 + e_2, \\
\bar{Y}_1^D &= 3352.909 + 655.84375x_1 + 249.09375x_2 + 153.96875 - 577.59375 + e_3, \\
\bar{Y}_2^D &= 11.30727 + 0.10625x_1 - 0.125x_2 - 0.084375x_3 - 0.060625 + e_4.
\end{align*}
\] (22)

Table 13 exhibits the detailed results of the simulated responses as obtained from the 33 experimental trials. Moreover, the regression residuals are calculated via difference between the real and estimated output amounts for the objective functions and reported in Table 13.

Statistical hypothesis testing for the residuals based on applying normal goodness of fit test revealed that there is no trace to reject the null hypothesis (all \(p\) values are greater than 0.76). Consequently, the validation of the estimated fitness functions could be accepted statistically.

In the present study, an effort has been made to determine the optimal values of objective functions to get the best possible surface quality within the specific test range. Effective optimization methods including goal programming (GP), multichoice goal programming (MCGP), and augmented epsilon-constraint method (AUGMECON) are developed to solve the optimization problem; for this study, it is called as model (23):

\[
\begin{align*}
\text{Min} &\bar{Y}_1, \\
\text{Min} &\bar{Y}_2, \\
\text{Min} &\bar{Y}_1^D, \\
\text{Min} &\bar{Y}_2^D, \\
\text{s.t.} & \quad -1 \leq x_i \leq 1, \quad i = 1, 2, 3, 4.
\end{align*}
\] (23)

6.2.4. Solution Methods

(1) GP Method. The GP method as a MCDM technique is used for solving the optimization problem; model (23). Thus, modelling the GP technique comes in equations (24)–(29) which is the minimization of deviations from the set of goals. It should be noted that \(d_i^+\) and \(d_i^-\) are over achievement and under achievement of the goal \(i\) one-to-one, in accordance with GP technique.

### Table 12: Process parameters with their levels.

| Inputs          | Code levels          |
|-----------------|----------------------|
| Input variables | (−1)                 |
| \(x_1\)         | 0                    |
| \(x_2\)         | 30                   |
| \(x_3\)         | 30                   |
| \(x_4\)         | Red                  |

| \(x_1\)         | B-R-P                |
| \(x_2\)         | B-R-P                |
| \(x_3\)         | B-R-P                |
| \(x_4\)         | B-R-P                |
Table 13: Experimental details, predictions, and residuals of fitness functions for Example 1.

| Run | Factors | PFM | Residuals |
|-----|---------|-----|-----------|
| | | | Y | Y'S | e | e | e | e |
| S. No. | M_x (x_1) | M_y (x_2) | Col. (x_3) | Seq. (x_4) | Y_1 | Y_2 | Y'S_1 | Y'S_2 | e_1 | e_2 | e_3 | e_4 |
| 1 | -1 | -1 | -1 | 1 | 69.9895 | 25094.724 | 2763 | 10.5 | -3.378 | 0.250 | 1046.591 | -0.85 |
| 2 | -1 | 1 | -1 | 1 | 50.4583 | 18121.958 | 3582 | 10.57 | -17.188 | -4695.59 | 1367.404 | -0.53 |
| 3 | -1 | -1 | 1 | -1 | 70.0469 | 25115.143 | 3193 | 11.98 | 0.132 | 634.334 | 13.466 | 0.678 |
| 4 | -1 | 1 | -1 | -1 | 68.2703 | 24480.632 | 1567 | 10.64 | 3.924 | 2291.933 | -1802.78 | -0.581 |
| 5 | -1 | 1 | 1 | 1 | 75.0906 | 24478.783 | 1108 | 10.54 | 7.598 | 1646.058 | -1414.53 | -0.391 |
| 6 | 1 | 1 | 1 | 1 | 64.08 | 20905.797 | 5501 | 11.91 | 10.782 | 2682.447 | 1666.779 | 0.766 |
| 7 | -1 | -1 | -1 | 1 | 66.2355 | 23754.198 | 1545 | 11.21 | 7.0622 | 3269.099 | -1483.1 | -0.352 |
| 8 | 1 | -1 | 1 | -1 | 49.155 | 17657.079 | 5332 | 11.65 | -0.996 | 77.755 | 650.528 | 0.216 |
| 9 | -1 | 1 | 1 | -1 | 55.5073 | 18123.471 | 4670 | 10.37 | -8.685 | -4080.4 | 992.278 | -0.682 |
| 10 | 1 | 1 | 1 | 1 | 49.1544 | 17657.076 | 8272 | 10.8 | -0.844 | 62.576 | 3282.591 | -0.465 |
| 11 | -1 | -1 | -1 | 1 | 77.0841 | 25125.851 | 2136 | 11.7 | 3.716 | 31.377 | 419.591 | 0.350 |
| 12 | 1 | 1 | 1 | 1 | 55.9351 | 18262.415 | 2740 | 10.46 | 2.637 | 39.065 | 1094.22 | -0.683 |
| 13 | 1 | -1 | 1 | -1 | 49.1643 | 17659.954 | 4756 | 11.79 | -6.556 | -2211.47 | 264.778 | 0.275 |
| 14 | -1 | 1 | -1 | 1 | 75.1465 | 24497.28 | 1112 | 11.68 | 7.501 | 1679.731 | -1102.6 | 0.580 |
| 15 | -1 | 1 | 1 | 1 | 75.0933 | 24479.709 | 1919 | 10.62 | 7.601 | 1646.984 | -603.534 | -0.311 |
| 16 | 1 | -1 | 1 | 1 | 56.1236 | 18323.137 | 4867 | 10.5 | -2.897 | -2177.14 | 1530.966 | -0.893 |
| 17 | -1 | 1 | 1 | -1 | 50.4628 | 18123.472 | 2487 | 11.75 | -13.730 | -4080.4 | -1190.72 | 0.698 |
| 18 | 1 | 1 | -1 | 1 | 49.221 | 17680.594 | 5831 | 11.87 | -4.420 | -527.58 | 2304.716 | 0.558 |
| 19 | 1 | 1 | -1 | 1 | 55.9623 | 18271.136 | 2740 | 11.62 | 2.511 | 62.962 | -786.284 | 0.307 |
| 20 | -1 | -1 | 1 | -1 | 70.1489 | 25151.202 | 2250 | 10.63 | 0.234 | 670.402 | -929.534 | -0.672 |
| 21 | -1 | 1 | -1 | 1 | 70.0402 | 25113.195 | 4961 | 11.62 | -3.174 | 3.545 | 2936.654 | 0.439 |
| 22 | -1 | -1 | -1 | 1 | 68.3057 | 24493.579 | 1572 | 11.12 | 3.960 | 2304.88 | -1797.78 | -0.101 |
| 23 | 1 | -1 | -1 | 1 | 54.0548 | 17651.809 | 3723 | 11.58 | -1.818 | -2204.44 | -460.284 | -0.103 |
| 24 | 1 | 1 | 1 | 1 | 49.1897 | 17670.027 | 4813 | 11.62 | -0.808 | 75.527 | -176.409 | 0.355 |
| 25 | -1 | -1 | -1 | 1 | 76.9171 | 25071.426 | 6444 | 11.82 | 6.849 | 605.802 | 3572.404 | 0.349 |
| 26 | -1 | 1 | -1 | 1 | 69.9741 | 25088.895 | 1816 | 11.97 | -3.241 | -20.755 | -208.347 | 0.789 |
| 27 | -1 | 1 | -1 | 1 | 77.0645 | 25119.035 | 2525 | 11.73 | 6.997 | 653.411 | 346.597 | 0.259 |
| 28 | 1 | 1 | -1 | 1 | 49.2333 | 17684.92 | 5338 | 11.72 | -0.918 | 105.596 | 656.528 | 0.576 |
| 29 | 0 | 0 | 0 | 0 | 63.4534 | 22761.417 | 1449 | 11.26 | 1.770 | 1416.93 | -1903.91 | -0.047 |
| 30 | 1 | -1 | 1 | -1 | 73.7894 | 26450.876 | 2499 | 11.81 | 18.069 | 6579.451 | -1992.22 | 0.295 |
| 31 | 1 | -1 | -1 | 1 | 50.9987 | 18314.877 | 1526 | 11.98 | -8.175 | -2170.22 | -1502.1 | 0.418 |
| 32 | 1 | -1 | -1 | 1 | 49.1725 | 17662.833 | 4399 | 10.9 | -6.701 | -2193.42 | 215.716 | -0.783 |
| 33 | 1 | -1 | 1 | 1 | 51.0173 | 18321.587 | 1210 | 11.22 | -8.003 | -2178.69 | -2126.03 | -0.173 |

Figure 11: The linear and left triangular utility functions for the objectives.
The present work is an effort to effectively perform the Goal programming method for producing the optimal solution in a multicriteria sequencing problem. The MCGP method, following the idea of MCGP with utility functions “proposed by Chang [72],” first the utility function is depicted in Figure 11; then, model (23) is formulated as comes in the following equations:

\[
\begin{align*}
\text{Min} & \quad \sum_{i=1}^{4} \left( d_i^+ + d_i^- \right), \\
\text{s.t.,} & \quad 61.68304 - 7.0973x_1 - 2.861x_2 - 0.0766x_3 + 1.65x_4 - d_1^+ + d_1^- = 49.1544, \quad (25) \\
& \quad 21344.487 - 2304.6875x_1 - 1138.4625x_2 + 7.58775x_3 + 314.425x_4 - d_2^+ + d_2^- = 17651.809, \quad (26) \\
& \quad 3352.909 + 655.84375x_1 + 249.09375x_2 + 153.96875x_3 - 577.59375x_4 - d_3^+ + d_3^- = 1108, \quad (27) \\
& \quad 11.30727 + 0.10625x_1 - 0.125x_2 - 0.084375x_3 - 0.060625x_4 - d_4^+ + d_4^- = 10.37, \quad (28)
\end{align*}
\]

\[-1 \leq x_i \leq 1, \quad i = 1, 2, \]
\[x_3, x_4 = -1 \text{ or } 0 \text{ or } 1, \]
\[d_i^+, d_i^- \geq 0. \quad (29)\]

\[
\begin{align*}
\text{Min} & \quad \sum_{i=1}^{4} w_i(d_i^+ + d_i^-) + \beta f_i, \\
\text{s.t.,} & \quad 61.68304 - 7.0973x_1 - 2.861x_2 - 0.0766x_3 + 1.65x_4 - d_1^+ + d_1^- = y_1, \quad (31) \\
& \quad 21344.487 - 2304.6875x_1 - 1138.4625x_2 + 7.58775x_3 + 314.425x_4 - d_2^+ + d_2^- = y_2, \quad (32) \\
& \quad 3352.909 + 655.84375x_1 + 249.09375x_2 + 153.96875x_3 - 577.59375x_4 - d_3^+ + d_3^- = y_3, \quad (33) \\
& \quad 11.30727 + 0.10625x_1 - 0.125x_2 - 0.084375x_3 - 0.060625x_4 - d_4^+ + d_4^- = y_4, \quad (34) \\
& \quad 49.1544 \leq y_1 \leq 77.0841, \quad (35) \\
& \quad 17651.809 \leq y_2 \leq 26450.876, \quad (36) \\
& \quad 1108 \leq y_3 \leq 8272, \quad (37) \\
& \quad 10.37 \leq y_4 \leq 11.98, \quad (38)
\end{align*}
\]

\[
\begin{align*}
\lambda_1 & \leq \frac{77.0841 - y_1}{27.9297}, \quad (39) \\
\lambda_2 & \leq \frac{26450.876 - y_2}{8799.067}, \quad (40) \\
\lambda_3 & \leq \frac{8272 - y_3}{7164}, \quad (41)
\end{align*}
\]
(3) AUGMECON Method. In this study, to produce a representation of the Pareto optimal solutions for the problem, the epsilon-constraint approach is used. Following the idea of AUGMECON "proposed by Mavrotas [73]," firstly, payoff table is created as shown in Table 14. Then, the ranges are calculated and shown in Table 15. Finally, model (23) is formulated in equations (45)–(49).

Min $61.68304 - 7.0973x_1 - 2.861x_2 - 0.0766x_3 + 1.65x_4 + \text{epsilon}\left(\frac{s_2}{3757.600} + \frac{s_3}{1636.500} + \frac{s_4}{0.251}\right)$,  

s.t.,

$Y_2 = \frac{(21344.487 - 2304.6875x_1 - 1138.4625x_2 + 7.58775x_3 + 314.425x_4)}{1000}$,  

$Y_1^{SD} = \frac{(3352.909 + 655.84375x_1 + 249.09375x_2 + 153.96875x_3 - 577.59375x_4)}{1000}$,  

$Y_2^{SD} = 11.30727 + 0.10625x_1 - 0.125x_2 - 0.084375x_3 - 0.060625x_4$,  

$-1 \leq x_i \leq 1, \quad i = 1, 2$,  

$x_3, x_4 = -1 \text{ or } 0 \text{ or } 1$.
7. Discussion

The problem is solved based on the solution methods via LINGO software on a PC with MS-Windows 10 Pro, 4.0 GB of RAM, and 4 COMPUTE CORES 2C+2G to achieve the optimal solutions. The results based on GP and MCGP are presented in Table 16. The subsequent results are obtained after solving model (23) with AUGMECON. Regarding this solution method, Pareto optimal solutions are gained. The Pareto optimal solutions are shown in Table 17. In the next step, the ED simulation tool is being attuned to the above results and the above optimal scenarios are being run. Then, according to the defined PFM (Proposed Simulation Model), the results are shown in Table 18.

There is an acceptable closeness between the solutions in Tables 16–18, by comparing the results of CS and RSM. This indicates the accuracy of prediction made by RSM.

Consistent with the above results, optimal scenarios have the following characteristics.

Supposing that the percentage of part’s performed operations on second and third machines as well is at least 30%:

If the sequence type is R-B-P, place the first part in red on second and third machines.

If the sequence type is B-R-P, the initial part can also be in purple on the second and third machines.

A set of these conditions can provide robust optimization for the examined robotic cell to meet the defined goals.

8. Conclusions and Future Research Directions

This study purely tends to investigate the problem of three-machine robotic cell from two view points: when the system produces similar parts which leads to a scheduling-allocation problem and when the robotic cell produces dissimilar parts which leads to a sequencing problem.

For the first problem, a mathematical model was proposed. The proposed model was focused basically on minimizing jointly operational cost and cycle time in the robotic manufacturing cell under breakdowns including failures and repairs. In this case, the operating conditions decisions in robotic cells link up with maintenance decisions which will improve time and operational costs simultaneously. Consequently, this enhancement increases the production rate. This work is an effort to implement the epsilon-constraint approach for producing the Pareto optimal (nondominated) solutions in a mixed integer nonlinear programming problem in comparison with MOPSO algorithm. Numerical experiments were carried out on several designated instances, and the results were discussed.

The solution method for the second problem purely has a tendency to demonstrate robustness by simulation-based optimization to determine the sequencing in a defective three-machine robotic cell under $S_6$ cycle. By finding the best scenarios for the entrance of parts into the robotic cell, a robust simultaneous minimization of the time and operating costs is conceivable.

The contribution can be extended to the robotic cell considering robot failures or to the dual-gripper robot instead of single-gripper one. Eliminating flexibility in the robotic cell is also considerable for future research. On the contrary, developing new models with other probability distributions of the time to failure and time to repair for the machines besides the exponential probability distribution is an interest. Moreover, the present study can allow modelling different scenarios to be examined in forthcoming simulations with less expense, time, and resources than experimentation with the real-world system. Thus, enhancement of the manufacturing system earlier or through manufacture will be possible.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.
References

[1] H. Hoogeveen, “Multicriteria scheduling,” European Journal of Operational Research, vol. 167, no. 3, pp. 592–623, 2005.
[2] H. Gultekin, M. S. Akturk, and O. Ekin Karasan, “Scheduling in a three-machine robotic flexible manufacturing cell,” Computers & Operations Research, vol. 34, no. 8, pp. 2463–2477, 2007.
[3] H. Gultekin, M. S. Akturk, and O. E. Karasan, “Bicriteria robotic operation allocation in a flexible manufacturing cell,” Computers & Operations Research, vol. 37, no. 4, pp. 779–789, 2010.
[4] O. Alagöz and M. Azizoğlu, “Rescheduling of identical parallel machines under machine eligibility constraints,” European Journal of Operational Research, vol. 149, no. 3, pp. 523–532, 2003.
[5] R. K. Kayan and M. S. Akturk, “A new bounding mechanism for the CNC machine scheduling problems with controllable processing times,” European Journal of Operational Research, vol. 167, no. 3, pp. 624–643, 2005.
[6] S. Gür dol and M. S. Akturk, “Considering manufacturing cost and scheduling performance on a CNC turning machine,” European Journal of Operational Research, vol. 177, no. 1, pp. 325–343, 2007.
[7] A. M. Brintrup, J. Ramsden, and A. Tiwari, “An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization,” Computers in Industry, vol. 58, no. 3, pp. 279–291, 2007.
[8] A. Turkcan, M. S. Akturk, and R. H. Storer, “Due date and cost-based FMS loading, scheduling and tool management,” International Journal of Production Research, vol. 45, no. 5, pp. 1183–1213, 2007.
[9] I. M. Restrepo and S. Balakrishnan, “Fuzzy-based methodology for multi-objective scheduling in a robot-centered flexible manufacturing cell,” Journal of Intelligent Manufacturing, vol. 19, no. 4, pp. 421–432, 2008.
[10] J. Feng, A. Che, and N. Wang, “Bi-objective cyclic scheduling in a robotic cell with processing time windows and non-Euclidean travel times,” International Journal of Production Research, vol. 52, no. 9, pp. 2505–2518, 2014.
[11] K. Ma, P. Yan, and W. Dai, “A hybrid discrete differential evolution algorithm for dynamic scheduling in robotic cells,” in Proceedings of the 2016 13th International Conference on IEEE Service Systems and Service Management (ICSSSM), pp. 1–6, New York, NY, USA, 2016.
[12] K. Abd, K. Abhary, and R. Marian, “Multi-objective optimisation of dynamic scheduling in robotic flexible assembly cells via fuzzy-based Taguchi approach,” Computers & Industrial Engineering, vol. 99, pp. 250–259, 2016.
[13] S. A. Mamsouri, E. Aktaş, and U. Besikci, “Green scheduling of a two-machine flowshop: trade-off between makespan and energy consumption,” European Journal of Operational Research, vol. 248, no. 3, pp. 772–788, 2016.
[14] M. Ghadiri Nejad, S. M. Shavarani, B. Vizvári et al., “Trade-off between process scheduling and production cost in cyclic flexible robotic cells,” International Journal of Advanced Manufacturing Technology, vol. 96, 2018.
[15] M. H. Far, H. Haleh, and A. Saghaei, “A fuzzy bi-objective flexible cell scheduling optimization model under green and energy-efficient strategy using Pareto-based algorithms: SATPSPGA, SANRGA, and NSGA-II,” The International Journal of Advanced Manufacturing Technology, vol. 105, no. 9, pp. 3853–3879, 2019.
[16] M. Founami, A. Razeghi, and K. Smith-Miles, “Stochastic optimization of two-machine flow shop robotic cells with controllable inspection times: from theory toward practice,” Robotics and Computer-Integrated Manufacturing, vol. 61, p. 101822, 2020.
[17] S. P. Sethi, C. Sriskandarajah, G. Sorger et al., “Sequencing of parts and robot moves in a robotic cell,” International Journal of Flexible Manufacturing Systems, vol. 4, no. 3–4, pp. 331–358, 1992.
[18] S. P. Sethi, J. B. Sidney, and C. Sriskandarajah, “Scheduling in dual gripper robotic cells for productivity gains,” IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 324–341, 2001.
[19] M. Dawande, C. Sriskandarajah, and S. Sethi, “On throughput maximization in constant travel-time robotic cells,” Manufacturing & Service Operations Management, vol. 4, no. 4, pp. 296–312, 2002.
[20] M. Dawande, H. N. Geismar, S. P. Sethi, and C. Sriskandarajah, “Sequencing and scheduling in robotic cells: recent developments,” Journal of Scheduling, vol. 8, no. 5, pp. 387–426, 2005.
[21] H. N. Geismar and M. Pinedo, “Robotic cells with stochastic processing times,” IIE Transactions, vol. 42, no. 12, pp. 897–914, 2010.
[22] A. Elmi and S. Topaloglu, “Multi-degree cyclic flow shop robotic cell scheduling problem: ant colony optimization,” Computers & Operations Research, vol. 73, pp. 67–83, 2016.
[23] H. Gultekin, A. Tula, and M. S. Akturk, “Automated robotic assembly line design with unavailability periods and tool changes,” European Journal of Industrial Engineering, vol. 10, no. 4, pp. 499–526, 2016.
[24] M. Al-Salem, M. Haouari, M. Kharbeche et al., “A free-slack-based genetic algorithm for the robotic cell problem with controllable processing times,” in Proceedings of the Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling, Springer, Cham, pp. 77–93, 2016.
[25] M. Founami, K. Smith-Miles, and I. Gunawan, “Scheduling of two-machine robotic rework cells: in-process, post-process and in-line inspection scenarios,” Robotics and Autonomous Systems, vol. 91, pp. 210–225, 2017.
[26] H. Gultekin, B. Coban, and V. E. Akhlaghi, “Cyclic scheduling of parts and robot moves in m-machine robotic cells,” Computers & Operations Research, vol. 90, pp. 161–172, 2018.
[27] S. Gür dol, H. Gultekin, and V. E. Akhlaghi, “Energy conscious scheduling of a material handling robot in a manufacturing cell,” Robotics and Computer-Integrated Manufacturing, vol. 58, pp. 97–108, 2019.
[28] A. Majumder, D. Laba, and P. N. Suganthan, “Bacterial foraging optimization algorithm in robotic cells with sequence-dependent setup times,” Knowledge-Based Systems, vol. 172, pp. 104–122, 2019.
[29] W. Bożejko, J. Pempera, C. Smutnicki et al., “Cyclic scheduling in the manufacturing cell,” in Proceedings of the Modelling and Performance Analysis of Cyclic Systems, Springer, Cham, pp. 49–62, 2020.
[30] H. Kise, T. Shiroyama, and T. Ibaraki, “Automated two-machine flowshop scheduling: a solvable case,” IIE Transactions, vol. 23, no. 1, pp. 10–16, 1991.
[31] R. Logendran and C. Sriskandarajah, “Sequencing of robot activities and parts in two-machine robotic cells,”
International Journal of Production Research, vol. 34, no. 12, pp. 3447–3463, 1996.

32. Y. P. Aneja and H. Kamoun, "Scheduling of parts and robot activities in a two machine robotic cell," Computers & Operations Research, vol. 26, no. 4, pp. 297–312, 1999.

33. G. D. Batur, O. E. Karasan, and M. S. Akturk, "Multiple part-type scheduling in flexible robotic cells," International Journal of Production Economics, vol. 135, no. 2, pp. 726–740, 2012.

34. G. D. Batur, S. Erol, and O. E. Karasan, "Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells," Computers & Industrial Engineering, vol. 100, pp. 72–87, 2016.

35. N. G. Hall, H. Kamoun, and C. Sriskandarajah, "Scheduling in robotic cells: classification, two and three machines," Operations Research, vol. 45, no. 3, pp. 421–439, 1997.

36. A. H. Abdekhodaee, A. Wirth, and H.-S. Gan, "Scheduling two parallel machines with a single server: the general case," Computers & Operations Research, vol. 33, no. 4, pp. 994–1009, 2006.

37. H. J. Paul, C. Bierwirth, and H. Kopfer, "A heuristic scheduling procedure for multi-item hoist production lines," International Journal of Production Economics, vol. 105, no. 1, pp. 54–69, 2007.

38. C. Wen, S. D. Ekşioğlu, A. Greenwood, and S. Zhang, "Crane scheduling in a shipbuilding environment," International Journal of Production Economics, vol. 124, no. 1, pp. 40–50, 2010.

39. M. M. S. Abdulkader, M. M. ElBeheiry, N. H. Afia, and F. Aqlan, "Study of production scenarios with the use of simulation models," European Journal of Industrial Engineering, vol. 2, no. 3, pp. 321–367, 2007.

40. W. Lei, A. Che, and C. Chu, "Optimal cyclic scheduling of a robotic flowshop with multiple part types and flexible processing times," European Journal of Industrial Engineering, vol. 8, no. 2, pp. 143–167, 2014.

41. G. D. Batur and S. Erol, "Using simulated annealing for flexible robotic cell scheduling," Gazi University Journal of Science, vol. 29, no. 3, pp. 573–582, 2016.

42. M. Savsar and M. Aldaihani, "Modeling of machine failures in a flexible manufacturing cell with two machines served by a robot," Reliability Engineering & System Safety, vol. 93, no. 10, pp. 1551–1562, 2008.

43. M. Savsar, "Reliability modelling of a manufacturing cell operated under degraded mode," in Proceedings of the International Conference on Industrial Engineering and Operations Management (IEOM 2010), New York, NY, USA, 2010.

44. M. M. Hamasha, A. Alazzam, S. Hamasha, F. Aqlan, O. Almeanazel, and M. T. Khasawneh, "Multimachine flexible manufacturing cell analysis using a Markov chain-based approach," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 5, no. 3, pp. 439–446, 2015.

45. B. Vaisi, H. Farughi, and S. Raissi, "Bi-criteria robotic cell scheduling and operation allocation in the presence of breakdowns," International Journal of Industrial Engineering & Production Research, vol. 29, no. 3, pp. 343–357, 2018.

46. M. Selim and A. S. Gurel, "Machining conditions-based preventive maintenance," International Journal of Production Research, vol. 45, no. 8, pp. 1725–1743, 2007.

47. D. L. Grosh, A Primer of Reliability Theory, John Wiley & Sons, Berlin, Germany, 1989.

48. A. Mukerjee, R. Biswas, K. Deb, and A. P. Mathur, "Multi-objective evolutionary algorithms for the risk-return trade-off in bank loan management," International Transactions in Operational Research, vol. 9, no. 5, pp. 583–597, 2002.

49. S. Lalwani, S. Singhal, R. Kumar et al., "A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm," Transactions on combinatorics, vol. 2, no. 1, pp. 39–101, 2013.

50. C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 256–279, 2004.

51. H. Toroudi, M. Madani, F. Sarlak et al., "A multi-objective method for solving assembly line balancing problem," Decision Science Letters, vol. 6, no. 1, pp. 1–10, 2017.

52. M. Reyes-Sierra and C. C. Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," International Journal of Computational Intelligence Research, vol. 2, no. 3, pp. 287–308, 2006.

53. M. Pidd, Computer Simulation in Management Science, Vol. 4, Wiley, Chichester, UK, 1998.

54. M. C. Fu, "Feature article: optimization for simulation: theory vs. Practice," INFORMS Journal on Computing, vol. 14, no. 3, pp. 192–215, 2002.

55. F. T. S. Chan and H. K. Chan, "A comprehensive survey and future trend of simulation study on FMS scheduling," Journal of Intelligent Manufacturing, vol. 15, no. 1, pp. 87–102, 2004.

56. A. Caggiano and R. Teti, "Modelling, analysis and improvement of mass and small batch production through advanced simulation tools," Procedia CIRP, vol. 12, pp. 426–431, 2013.

57. F. A. Lora, U. Boff, C. C. Yurgel et al., "Validation of the computer simulation process applied to the incremental forming process for the evaluation of strain paths," in Key Engineering Materials, vol. 554, pp. 2453–2461, Trans Tech Publications, London, UK, 2013.

58. B. Yang, W. Chen, and C. Lin, "The algorithm and simulation of multi-objective sequence and balancing problem for mixed mode assembly line," International Journal of Simulation Modelling, vol. 16, no. 2, pp. 357–367, 2017.

59. M. Kikolski, "Study of production scenarios with the use of simulation models," Procedia Engineering, vol. 182, pp. 321–328, 2017.

60. J. E. Diaz, J. Handl, and D.-L. Xu, "Integrating meta-heuristics, simulation and exact techniques for production planning of a failure-prone manufacturing system," European Journal of Operational Research, vol. 266, no. 3, pp. 976–989, 2018.

61. M. Thenarasu, K. Rameshkumar, and S. P. Anbuudayansar, "Simulation modelling and innovative application of multi-criteria decision making rule for minimizing makespan in a large scale press shop," International Journal of Industrial Engineering: Theory, Applications and Practice, vol. 26, no. 6, 2019.

62. K. T. Vinod, S. Prabagaran, and O. A. Joseph, "Dynamic due date assignment method A simulation study in a job shop with sequence-dependent setups," Journal of Manufacturing Technology Management, vol. 30, no. 2, pp. 421–437, 2019.

63. G. Linneusson, A. H. C. Ng, and T. Aalås, "A hybrid simulation-based optimization framework supporting strategic maintenance development to improve production performance," European Journal of Operational Research, vol. 281, no. 2, pp. 402–414, 2020.

64. Z. Kbah, N. O. Erdil, and F. Aqlan, "Risk assessment in oil and gas industry using simulation and BOW-tie analysis," International Journal of Industrial Engineering: Theory, Applications and Practice, vol. 27, no. 1, p. 2020.
[65] A. Negahban and J. S. Smith, “Simulation for manufacturing system design and operation: literature review and analysis,” *Journal of Manufacturing Systems*, vol. 33, no. 2, pp. 241–261, 2014.

[66] D. Mourtzis, M. Doukas, and D. Bernidaki, “Simulation in manufacturing: review and challenges,” *Procedia CIRP*, vol. 25, pp. 213–229, 2014.

[67] A. L. Soyster, “Technical note-convex programming with set-inclusive constraints and applications to inexact linear programming,” *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973.

[68] B. Vaisi, H. Farughi, and S. Raissi, “TWO-MACHINE robotic cell sequencing under different uncertainties,” *International Journal of Simulation Modelling (IJSIMM)*, vol. 17, no. 2, 2018.

[69] K. Khalili-Damghani, M. Tavana, A.-R. Abtahi, and F. J. Santos Arteaga, “Solving multi-mode time-cost-quality trade-off problems under generalized precedence relations,” *Optimization Methods and Software*, vol. 30, no. 5, pp. 965–1001, 2015.

[70] R. H. Myer and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiment*, John Wiley & Sons, New York, NY, USA, 2002.

[71] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, New York, NY, USA, 2017.

[72] C.-T. Chang, “Multi-choice goal programming with utility functions,” *European Journal of Operational Research*, vol. 215, no. 2, pp. 439–445, 2011.

[73] G. Mavrotas, “Effective implementation of the ε-constraint method in multi-objective mathematical programming problems,” *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009.