

Research Article

Multiobjective Fuzzy Mixed Assembly Line Sequencing Optimization Model

Farzad Tahriri,¹ Siti Zawiah Md Dawal,¹ and Zahari Taha²

¹ Department of Mechanical Engineering, Centre for Product Design and Manufacturing, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

² Faculty of Mechanical Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang Darul Makmur, Malaysia

Correspondence should be addressed to Farzad Tahriri; farzad_tahriri@hotmail.com

Received 8 November 2013; Revised 20 February 2014; Accepted 11 March 2014; Published 8 May 2014

Academic Editor: Olabisi Falowo

Copyright © 2014 Farzad Tahriri et al. 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.

It can be deduced from previous studies that there exists a research gap in assembly line sequencing optimization model for mixed-model production lines. In particular, there is a lack of studies which focus on the integration between job shop and assembly lines using fuzzy techniques. Hence, this paper is aimed at addressing the multiobjective mixed-model assembly line sequencing problem by integrating job shop and assembly production lines for factories with modular layouts. The primary goal is to minimize the make-span, setup time, and cost simultaneously in mixed-model assembly lines. Such conflicting goals arise when switching between different products. A genetic algorithm (GA) approach is used to solve this problem, in which trapezoidal fuzzy numbers are implemented for variables such as operation and travelling time in order to generate results with higher accuracy and representative of real-case data.

1. Introduction

It is known that an efficient and effective production line sequencing and scheduling are required for manufacturing industries due to the fluctuations in market demand and increasing competition on the global scale. Hence, manufacturers are gearing towards optimizing the design of manufacturing systems in the shortest time possible in order to enhance competitiveness. The production lines of a multistage process can be generally classified as job-shop (JS) and flow-shop (FS) line processing, in which each type complies with a set of sequential operations. Production sequential operation is established in the factories in which the materials undergo a refining process in order to produce a product that is suitable for onward consumption and the components are assembled to make finished articles. Raw materials require a sequence of treatments to render them useful. Development of optimization methods for solving JS and FS problems effectively is of utmost importance to the apparel industry and other manufacturing industries which require similar assembly operations at large. JS and FS involve the assignment of a set of tasks to workstations

(machines) in a predefined sequence while optimizing one or more objectives without violating the restrictions imposed by the production line [1]. Most of the problems involving line balancing and sequencing problem in JS and FS fall within the NP-hard class of combinatorial optimization problems, which results in a critical need to develop efficient algorithms to attain optimum solutions [1–6]. It can be extremely time-consuming for optimum-seeking methods to obtain optimum solutions within a vast search space [2].

Efficient and effective JS and FS sequencing and scheduling is required for recent manufacturing processes in order to achieve higher customer satisfaction and shorter production lead times and enhance competitiveness in the escalating volatile market demand. This emphasizes the significance of just-in-time (JIT) and agile manufacturing in the modern era. To cope with such circumstances, JS and FS assembly lines are shifting towards mixed-model assembly lines, in which multiple models are produced on a constant basis. Therefore, sequencing with multiobjective goals is a critical operation scheduling that directly affects the efficiency of mixed-model production assembly lines for factories with modular layouts. It will be noted that modular layouts are

layouts which integrate JS and FS. Owing to the importance of this issue, this study is aimed at developing a multiobjective mixed-production line sequencing model by implementing genetic algorithms (GAs) and fuzzy logic.

2. Mixed-Model Assembly Line Sequencing

Mixed-model assembly lines (MMAL) have been widely used by manufacturers and they play a key role in producing a variety of products. Products with similar characteristics are assembled with different processing times on the same assembly line at very low cost [7–9]. This, in turn, requires the implementation of cost-efficient and flexible production systems. MMAL reduce setup operations to an extent that various models from a common base product can be manufactured in intermixed sequences. However, the observed diversity of MMAL makes thorough sequence planning essential in order to exploit the benefits of assembly line production. These benefits include diversified small lot production and prompt response to sudden changes in demand for models without large inventories [10]. Mixed-model sequencing (MMS) aims at avoiding or minimizing sequence-dependent work overload based on detailed scheduling which explicitly accounts for operation time, worker movement, station borders, and other operational characteristics of the line [10].

Solving the MMS problem involves determining a production sequence for multiple products along a modular layout. In order to reach this goal, manufacturers aim to achieve optimum scheduling times with minimum cost. Furthermore, it is impractical for manufacturers to constantly alter their machine layouts when producing a new product. For companies with fixed factory machine layouts, optimization of production lines is solely based on altering the sequence of production lines. However, this approach poses the following challenges that need to be addressed:

- (1) production of various products using existing facilities such as the number and location of machines, travelling time between the machines, and limited number of workers for different products;
- (2) assignment of parts from different products to the machines, in which the optimum sequence needs to be considered in order to minimize scheduling time. Problems also arise when changing the setup number of the machines;
- (3) assignment of parts to other parts during product assembly, in which the parts that are ready for assembly need to be considered in order to minimize the scheduling time required to assemble the product;
- (4) identification of the best minimum setup number and setup cost sequencing, in which the above mentioned issues need to be considered simultaneously.

It can be seen that identifying the best scheduling and sequencing production line is a challenging task, particularly for the production of a mixed-model from a large number of providers with multiple objectives. MMS is a NP-hard

problem which requires an effective model. A unique and stable model is needed to facilitate MMS in the production of mixed products in a manufacturing environment with multiple parts, machines, products, and assemblies in order to minimize scheduling time, idle time of the machines, setup number, and setup cost as well as to maximize the number of products and assembly of various products. An effective model provides companies with a means to reschedule their production lines based on the changing demands of a flexible market. The effective utilization of MMAL requires solving two problems in a sequential manner, as follows.

- (1) Levelling workloads for stations on the line: this is known as line design and balancing.
- (2) Maintaining a constant rate of usage for each part used on the production line: this involves determining the production sequence for various models.

The first goal involves sequencing the mixed-models in order to achieve a balanced workload at each assembly station, whereas the second goal involves sequencing the mixed-models in order to minimize variations in the usage of various parts and components over time [8, 9, 11]. A number of MMS studies attempted at solving mixed-model sequencing problems using sequencing procedures that will optimize various system measures such as throughput, scheduling time, number of stations, idle time, flow time, line length, work-in-process, and raw material demand deviations [12–16].

Zhu et al. [17] studied the complexity of product variety of MMAL defined by the operators such as selection of parts, tools, and fixtures as well as assembly procedures in sequence planning. The aim was to reduce manufacturing complexity in a multiproduct, multistage, manual assembly line environment. The methodology was developed to determine optimum assembly sequences which will minimize system complexity. Rekiek et al. [18] and De Lit et al. [19] developed an approach which integrates product family design (including assembly sequences) with the assembly system so that multiple products can be assembled on the same line. A new alternative for mixed-model assembly for low-volume manufacturing environments was investigated by Heike et al. [7]. The efficient utilization of labour and/or machinery presents significant challenges. The models developed focus on evaluating and understanding different alternatives for mixed-model assembly in the aerospace industry. These models include one linear and two nonlinear programs which are formulated to evaluate constant and variable scheduling time policies. The models were successfully implemented on an airplane assembly process characterized by low-volume manufacturing. Bukchin et al. [20] proposed a mathematical formulation for MMAL designs based on the make-to-order production policy. Hence, the arrival sequence is randomly distributed according to demand proportions, rendering the balancing procedure an important aspect of the design process. An integrated machine tool selection and sequencing model was proposed by Moona et al. [21]. The model determines machine visiting sequences for all part types, such that the total production time for

the production order is minimized and the workloads between machine tools are balanced. Numerical experiments were carried out using problems of various sizes in order to demonstrate the efficiency of the proposed GA on the integrated machine tool selection and sequencing problem.

Khan and Day [22] introduced a knowledge-based design methodology (KBDM) for automated and manual assembly lines, which can be applied equally well to single, multi-, and mixed-product assembly lines with either deterministic or stochastic operation times. Mendes et al. [23] developed simulation models for varying the levels of demand and line configurations and utilized a mixed-model PC camera assembly line. The simulation models were used to compare flow time and resource utilization, which provide operational support and help fine-tune line configurations. Joly and Frein [24] investigated a set of vehicles within an industrial environment in order to minimize manufacturing costs by considering assembly and paint shop objectives. The approach was implemented to solve a problem with a monoobjective function. One heuristic (a progressive, construction-sequence algorithm) and three metaheuristics (simulated annealing, variable neighbourhood search, and an evolutionary algorithm) were described and compared.

The use of GA for mixed-model sequencing line problems is discussed as follows. Ghosh and Gagnon [25] introduced a mathematical programming model and an iterative GA-based procedure for MALBP with parallel work stations, in which the goal was to maximize the production rate of the line for a predetermined number of operators. An intelligence-based GA was developed by Norozi et al. [26] in order to tackle the complexity of sequencing in parallel MMAL. A spreading and cutting sequencing (SCS) model using GA was implemented by Wong et al. [27] to solve the sequencing problem of a computerized fabric cutting system used in the garment industry. The job sequence of spreading and cutting operation can be optimized by reducing the completion time for daily operation of fabric spreading and cutting as well as improving the utilization of machines. Ponnambalam et al. [28] studied the performance of GAs for a multilevel MMAL sequencing problem. Both single objective and multiple objectives were considered. The main objective of the multiobjective GA was to minimize the total utility work levelling, parts usage, and total setup cost. Development of optimization methods for solving JS and FS that are involving assembly line balancing/sequencing in mixed-model problems falls under the NP-hard class problem. Optimization involves exploiting the capabilities of mathematical programming in order to minimize the overall make-span and setup objectives for sets of single lines [26]. Evolutionary computing is a research area within computer science for solving combinatorial optimization and complex problems based on the principles of generic population-based heuristic techniques. In the advent of metaheuristic algorithms in recent years, numerous complex scheduling problems have been studied and solved using metaheuristic search techniques such as ant colony optimization (ACO), tabu search (TS), genetic algorithm (GA), and simulated annealing (SA). Metaheuristic algorithms are used

to overcome the complexity of sequencing in assembly lines [26].

GA was introduced by David Goldberg based on the procedure of natural mechanism and natural genetics. The population is composed of a collection of chromosomes in which each string is encoded and the problem is solved as a finite length of genes [26]. Good gene characteristics produce new generations. After several evolutions, the genes become highly adaptive to a given environment [29]. GAs have been proven to be highly effective for achieving optimum or near-optimum solutions in complex real-world optimization problems. However, GAs are limited by the fact that their performance is very sensitive to parameter settings [29]. GA design consists of two key steps, namely, genetic operations and parameter settings [29]. The genetic operations involve choosing a suitable selection method. The typical methods for selection are roulette, tournament, stochastic uniform, remainder, and uniform selection. Selection is the only operation of evolution and thus choosing a suitable selection method is important. Parameter settings involve setting the required parameters and variables for controlling the algorithms such as population size, number of generations, the number of selected candidates, crossover rate, and mutation rate [30].

2.1. Multiobjective Mixed-Production Line Optimization.

Multiobjective formulations are realistic models of numerous complex engineering optimization problems. In most real-life problems, the objectives often contradict one another and optimizing a particular solution with respect to a single objective may give undesirable results for other objectives. Being a population-based approach, GA is well suited to solve multiobjective optimization problems [31]. A generic single-objective GA can be modified to search for a set of multiple nondominated solutions in a single run. The ability of GAs to simultaneously search different regions of a solution space makes it possible to search a diverse set of solutions for difficult problems with nonconvex, discontinuous, and multimodal solution spaces. Multiobjective optimization problems can be found in various fields, such as product and process design, finance, aircraft design, oil and gas industry, automobile design, or wherever optimum decisions are needed, by considering the trade-offs between multiple conflicting objectives. If more than one criterion is to be treated simultaneously, then it is a multiobjective optimization problem. Maximizing the profit and minimizing the cost of a product, maximizing the performance and minimizing fuel consumption of a vehicle, and minimizing the weight while maximizing the strength of a particular component are all examples of multiobjective optimization problems. If the multiobjective problem is well formed, there will not be a single solution that simultaneously minimizes each objective to its fullest. In each case, the aim is to search for a solution in which each objective has been optimized to an extent that if the objectives are optimized further, other objectives will suffer as a result. The search for an optimum solution and quantifying the extent to which this solution is superior compared to other such solutions

(note that there will generally be many solutions) are the goal when setting up and solving a multiobjective optimization problem. Most of the real-world decision problems involve multiple conflicting objectives that need to be tackled while adhering to the various constraints. In multiobjective problems, there may not exist a solution which is the best with respect to all objectives. It will be highlighted that there will be a set of solutions which are superior to other solutions in the search space when all objectives are considered but are inferior to the solutions for one or more objectives. These solutions are called nondominated solutions [32].

The job-shop scheduling problem with sequence-dependent setup times (SDST-JSP) is a generalization of the classical JSS, in which the setup operation of a machine is required when the machine switches between two jobs. The common goal of SDST-JSP is to minimize the completion time of the last job (make-span). SDST-JSP has been investigated by a number of researchers. A new multiobjective GA (MOGA) was introduced by Zacharia and Nearchou [33] for solving fuzzy assembly line balancing of SALBP-2 comprising multiple objectives, that is, minimizing the fuzzy scheduling time, balancing the fuzzy delay time, and smoothing the fuzzy index of the line. McMullen [34] proposed a technique to generate production sequences, whereby the number of setups and scheduling flexibility are considered. These objectives are inversely correlated, which poses a challenge for optimizing both objectives simultaneously. An efficient frontier approach was exploited for problems where simultaneous maximization of flexibility and minimization of setups are desired. McMullen [35] presented a relatively new approach using ACO to address a two-objective production-sequencing problem by stimulating the artificial intelligence agents of virtual ants to obtain desirable solutions for manufacturing logistics. The first objective involves minimizing the number of setups, while the second objective involves optimizing the stability of material usage rates. McMullen [36] developed a technique which addresses a JIT production-scheduling problem consisting of two objectives, namely, minimization of setups between product variants and optimization of schedule flexibility. These objectives are inversely correlated, which poses a challenge in attaining desirable results for both objectives. An efficient frontier approach was employed to address the problem, which yields desirable sequences for both objectives. The artificial neural network approach using Kohonen self-organizing map (SOM) was implemented to determine sequences which are desirable in terms of the number of setups and flexibility. McMullen and Tarasewich [37] introduced an efficient frontier approach to support sequencing decisions. A beam-search heuristic approach was used to effectively generate efficient frontiers. Rahimi-Vahed et al. [8] addressed a multiobjective sequencing problem using a mathematical formulation which considers three objectives simultaneously. These objectives are minimizing the total utility work, minimizing variations in the total production rate, and minimizing the total setup cost. The scatter search methodology was used to generate a set of solutions that approximates the nondominated frontiers where simultaneous minimization

of the above mentioned objectives is desired. Javadi et al. [9] proposed a fuzzy goal programming approach to solve a multiobjective mixed-model assembly line sequencing problem for a JIT production system. Three objectives are solved simultaneously as mentioned in [8]. These objectives comprise minimization of the total utility work and variations in the total production rate and total setup cost. The fuzzy goal programming based approach was proposed to solve the problem because of the conflicting objectives. Bard et al. [38] investigated the sequencing of MMAL, which was perceived as a multiobjective problem. They developed a method to minimize the overall line length and maintain a constant rate for part usage. The problem was solved using the weighted sum and tabu search methods. McMullen [39] focused on minimizing the number of setups and maintaining a constant rate for part usage, in which the multiobjective problem was solved using the TS method. McMullen and Frazier [40] developed an SA method for the model used by McMullen [39] as well as TS method. Ponnambalam et al. [28] compared the performance of selection mechanisms (i.e., Pareto stratum-niche cubicle and selection based on scalar fitness function values). The objectives of their study were to minimize variations in part usage, total utility work, and setup cost. The results indicated that the GA which incorporates Pareto stratum-niche cubicle exhibits superior performance compared to GAs with other selection mechanisms. Giard and Jeunet [41] presented an integer programming formulation for sequencing in mixed-model assembly lines, whereby the number of temporarily hired utility workers and the numbers of sequence-dependent setups are optimized simultaneously via a cost function. The objective function is a cost function involving two elements, namely, the cost associated with additional utility workers and setup cost.

2.2. Application of Fuzzy Numbers in Assembly Line Models.

Fuzzy set theory accounts for the ambiguities involved when assigning processing and scheduling times as well as the uncertainty contained within such time variables [42]. The data of real-world problems are imprecise, vague, or uncertain, in which the input data can only be estimated within this uncertainty. The uncertainty may be represented by a fuzzy number. The membership function of fuzzy data represents the grade of satisfaction of the decision-maker. The concept of fuzzy numbers was introduced to treat imprecise data such as the processing and travelling time of each job [43, 44]. The time variables were represented by triangular fuzzy numbers, which allow practitioners to account for ambiguities when assigning processing and scheduling times, while maintaining the variability of the stochastic environment [42].

Gen et al. [43] and Tsujimura et al. [44] proposed a new method for treating the data of real-world problems by introducing a fuzzy assembly line balancing based on GA. Fuzzy variables were implemented by Fonseca et al. [42] in order to solve assembly line balancing. Fuzzy set theory is a viable alternative method for modelling and solving stochastic assembly line balancing problems. The variability and uncertainty in assembly line balancing problems have

been traditionally modelled by statistical distribution. A novel fuzzy extension of the simple assembly line balancing problem (SALBP) was proposed by Zacharia and Nearchou [33]. The fuzzy job processing times reflect the uncertainty, variability, and imprecision with which real-world production systems are afflicted. The job processing times are formulated by triangular fuzzy membership functions. A new MOGA was introduced for solving the fuzzy SALBP in order to minimize the fuzzy scheduling time, fuzzy balance delay time, and fuzzy smoothness index of the line.

2.3. Summary of Previous Studies and Findings. It can be observed from the literature review that the majority of works are focused on an assembly line balancing and only a few papers address mixed-assembly line sequencing. Hence, mixed-assembly line sequencing needs to be investigated in detail. Most papers focused on the number and order of mixed-models, neglecting the assignment of a product's sub-parts to their respective machines (job sequence). In addition, most papers overlook the designation and adaptation of parts with other parts as well as their final assembly. It is found that the production of mixed-models is influenced by a number of criteria, which limits the applicability of the research results in real production line conditions. It will be highlighted that only one criterion was considered in previous works such as operation line, and therefore other factors such as the travelling time of the conveyor were neglected. Deterministic time has also been of interest in most studies. It will be highlighted that only assembly line balancing was carried out in previous works in order to minimize the prediction time for input data, which include the travelling and processing times for each available job. Although the results from mixed-assembly line sequencing studies are applicable in real manufacturing environments, little is known about fuzzy mixed-model assembly line sequencing. Hence, it is evident that there is a lack of studies which implements the concept of fuzzy time in order to minimize the prediction time of input data.

In general, the research papers can be classified into two groups, whereby the first group focuses solely on objective criteria, while the second group focuses on multiobjective investigations. A critical evaluation of previous works clarifies that addressing these objectives involves the development of various methods, in which MMAL sequencing is the ideal method for a single objective. Comparison of various GA methods reveals that multiobjective studies have not been investigated extensively, unlike single-objective studies. Much effort has been made to intensify and accelerate the running of GA methods to achieve optimum results. Although mixed-model assembly line sequencing is of prime importance, there is a lack of studies which focus on this topic. In this paper, a hybrid method is proposed, in which mixed-model assembly line sequencing is integrated with the operating and travelling time in the form of fuzzy numbers of a multiobjective optimization problem. The hybrid method will provide manufacturers with an ideal and feasible alternative since mixed-model assembly line sequencing is integrated into a software application. This has been ignored

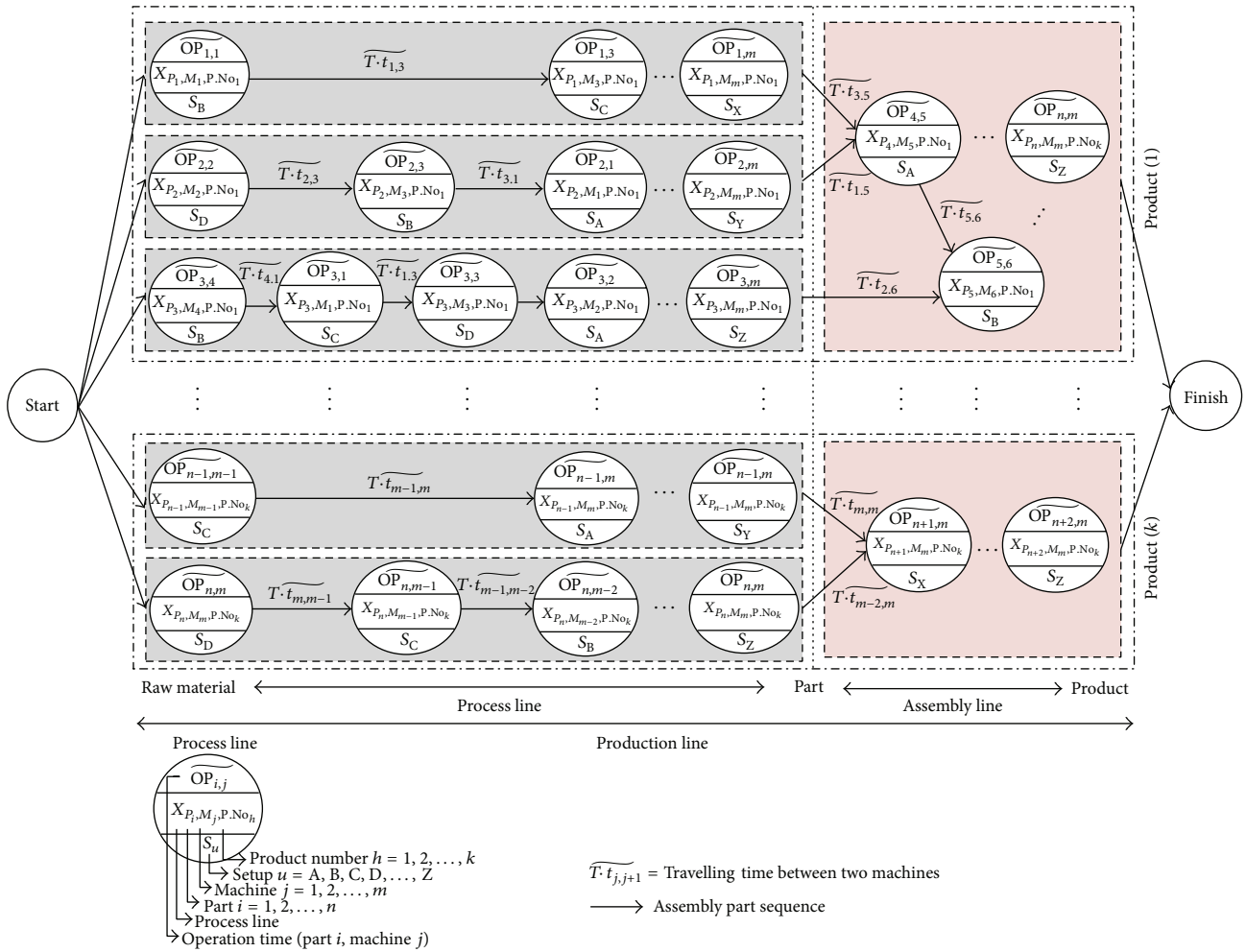
in previous works and much effort is needed to address the above issue.

3. Problem Definition and Assumptions

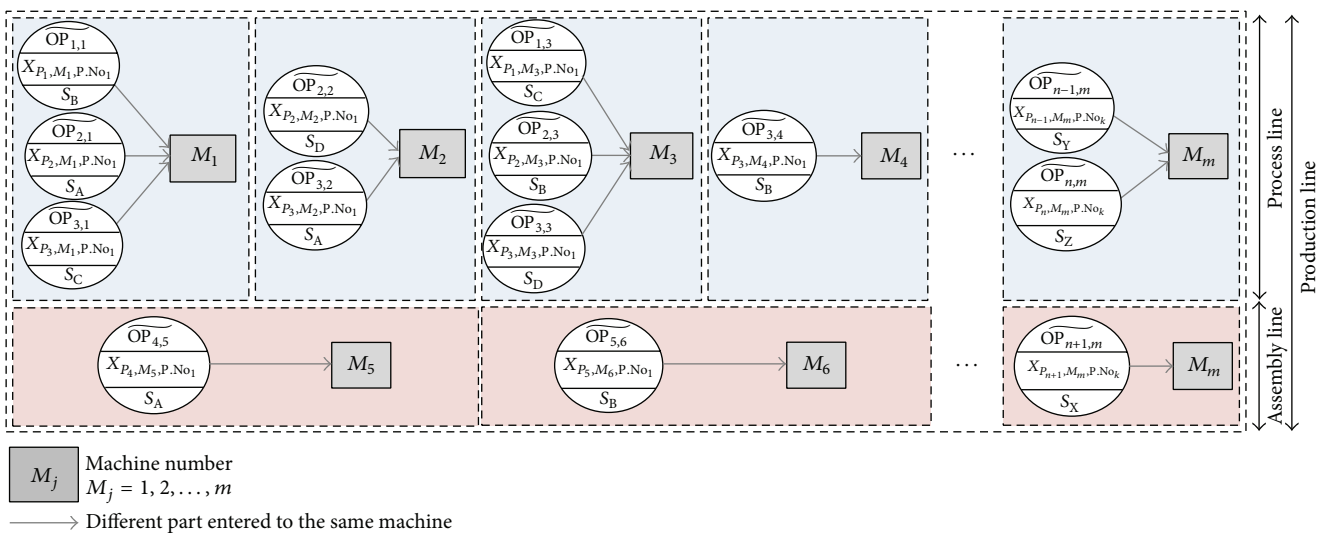
In a highly competitive market, companies try to produce various kinds of products to meet market demand. One manufacturing problem that is often associated with scheduling practices is mixed-model production line sequencing as shown in Figure 1. Solving this problem involves determining the optimal sequence for jobs (J_q) which includes the assigned part (P_i) to machine (M_j), considering each of the jobs with the availability of the machines (M_j) in job-shop. Moreover, with the need to solve the problem of determining the jobs (J_q) which includes assembly of the parts (P_i) in the assembly line for multiple products along a modular layout. The main solid arrows in Figure 1(a) represent the sequence to produce the parts, followed by a discrete part manufacturing assembly, leading to the final product. Likewise, the elapsed traveling time between machines for manufacturing a part is given by $(\overline{T \cdot t_{i,j}})$. To satisfy the no-wait restriction, the completion time of the operation ($\overline{OP_{i,j}}$) must be equal to the earliest time to start the operation. There must not be any waiting time in the processing of any consecutive operations of each of the jobs. Figure 1(b) represented by gray lines provides insight into problems involving the determining of the optimal number of setup tool changes sequencing (S_u) of jobs (J_q) assigned to "m" machines, throughout the job-shop and assembly production line.

The following assumptions are taken in the multiobjective evaluation: (i) all jobs are available at zero time; (ii) the conveyor (operator movement) moves at a constant speed. If job overlapping occurs, the remaining work will be accomplished by temporary operators; (iii) the job operation begins when the part enters the machine. Once the job is completed, the operator will move the part to the next machine; (iv) at a given time, only one job can be processed on one machine; (v) the operator is assigned to each selected part that is assigned to each machine; (vi) the position of the machines on the assembly line varies from one another depending on the user input, which is based on the travelling time; (vii) once operation has started on a machine, it cannot be interrupted before completion, either on or between the machines; (viii) the assembly line can process jobs for a product family, which is described by a joint priority matrix; (ix) the processing time varies for different jobs and these jobs are allocated to the same machine, no matter what the optimum processing time for each job might be; (x) the demand for all products as well as the sequence of the products entering the assembly line is predetermined; (xi) the completion time for all jobs is represented by a fuzzy number.

The goal is to minimize the make-span or total completion time and minimize the number of change tools setups for each machine in mixed-model production line sequencing of all the jobs without waiting between successive operations of a job by using the fuzzy operation and travelling time which makes the solution more realistic.



(a) The precedence problem in parts of the raw material process for producing a product



(b) The precedence problem of parts assigned to each machine based on changing the machine setup

FIGURE 1: The precedence of a multiobjective mixed-model production line sequencing problem.

4. Model Derivation

The objectives of the function value, for multiobjective mixed-production assembly line sequencing, are categorized into two main objectives: (1) minimizing the total make-span and (2) minimizing the total setup number as discussed in the following.

4.1. Minimizing the Total Make-Span. The step involves evaluating each chromosome based on the make-span by considering the product number and assembly sequence for a gene code (X_q). The Y_1, Y_2, Y_3 , and $\widetilde{S \cdot t}(X_q)$ are computed in which (X_q) are calculated as follows. Y_1 is determined by checking the start time ($\widetilde{S \cdot t}$) of the parts (P_i) entering the machines (M_j), based on the sequence assigned to the machines. This is expressed by

$$Y_1 = \begin{cases} 0, & P(X_q) \neq P(X_{q'}) \\ \widetilde{S \cdot t}(X_{q'}) + \widetilde{OP}(X_{q'}) + \widetilde{T \cdot t}(X_{q'}), & P(X_q) = P(X_{q'}) \end{cases} \quad (1)$$

$$q' = q - 1, q - 2, \dots, q - (q - 1), \quad q = 1, \dots, e.$$

Y_2 is determined by checking the start time ($\widetilde{S \cdot t}$) machine's availability (M_j) assigned to produce the parts (P_i), as given by

$$Y_2 = \begin{cases} 0, & M(X_q) \neq M(X_{q'}) \\ \widetilde{S \cdot t}(X_{q'}) + \widetilde{OP}(X_{q'}), & M(X_q) = M(X_{q'}) \end{cases} \quad (2)$$

$$q' = q - 1, q - 2, \dots, q - (q - 1), \quad q = 1, \dots, e.$$

Y_3 is then determined by checking the start time ($\widetilde{S \cdot t}$) of the parts (P_i) entering the machines (M_j) based on the sequence of the part's assembly (A.S._p) for each product (P.No._h) as given by (5) and (6). For each X_q , where $q = 1, 2, \dots, e$, note that the product number (P.No(X_q)) and assembly sequence (A.S(X_q)) are important. Equations (3) through (6) are given below.

(i) Consider

$$\text{If } A.S(X_q) = 0 \implies Y_3 = 0, \quad q = 1, 2, \dots, e \quad (3)$$

$$\text{If } A.S(X_q) \neq 0 \implies \text{compares} = P.No(X_q), \quad (4)$$

$$P.No(X_{q-1}).$$

(ii) If $P.No(X_q) = P.No(X_{q-1})$, then

$$Y_3 = \begin{cases} \widetilde{S \cdot t}(X_{q-1}) & A.S(X_q) = A.S(X_{q-1}) \\ \widetilde{S \cdot t}(X_{q-1}) + \widetilde{OP}(X_{q-1}) \\ \quad + \widetilde{T \cdot t}(X_{q-1}), & A.S(X_q) \neq A.S(X_{q-1}) \end{cases} \quad (5)$$

$$q = 1, 2, 3, \dots, e.$$

(iii) If $P.No(X_q) \neq P.No(X_{q-1})$ when searching gene code $X_{q'}$, where $q' = q - 1, q - 2, \dots, q - (q - 1)$, go backwards one by one until $P.No(X_q) = P.No(X_{q'})$ is met as follows:

$$Y_3 = \begin{cases} \widetilde{S \cdot t}(X_{q'}) & A.S(X_q) = A.S(X_{q'}) \\ \widetilde{S \cdot t}(X_{q'}) + \widetilde{OP}(X_{q'}) \\ \quad + \widetilde{T \cdot t}(X_{q'}), & A.S(X_q) \neq A.S(X_{q'}) \end{cases} \quad (6)$$

$$q' = q - 1, q - 2, \dots, q - (q - 1), \quad q = 1, 2, 3, \dots, e.$$

The start time is obtained using (7), in which Y_1, Y_2, Y_3 are determined from the previous steps. Consider

$$\widetilde{S \cdot t}(X_q) = \text{Max}(Y_1, Y_2, Y_3); \quad X_q, q = 1, 2, \dots, e. \quad (7)$$

Note. In the first run, if $Y_1 = Y_2 = Y_3 = 0$, then $\widetilde{S \cdot t} = \text{Max}(Y_1, Y_2, Y_3) + \widetilde{T \cdot t}$.

The make-span fitness function is then calculated for all chromosomes using

$$T.C.T = \text{Max} \widetilde{S \cdot t}(X_q) + \widetilde{OP}(X_q); \quad q = 1, 2, \dots, e. \quad (8)$$

4.2. Minimizing the Setup Number. In many industries, sequence-dependent setups are considered as an important item in assembly operations. The model considering sequence-dependent setups is considered in this paper. The machine number $M(X_q)$ and the setup number $S(X_q)$ are defined for the element (X_q) in the chromosome, where $q = 1, 2, \dots, e$; the total setup number at the beginning is T.S.N = 0. All the genes X_q , where $q = 1, 2, \dots, e$, in the chromosomes one by one will be checked and total setup number (T.S.N) based on the $M(X_q)$ and $S(X_q)$ has been determined for all X_q as shown in

(i)

$$\text{If } M(X_q) \neq M(X_{q'}), \quad \text{where } q = 1, 2, 3, \dots, e, \quad (9)$$

$$\text{and } q' = q - 1, q - 2, \dots, q - (q - 1)$$

then T.S.N = T.S.N + 1;

(ii)

$$\text{If } M(X_q) = M(X_{q'}), \quad \text{where } q = 1, 2, 3, \dots, e, \quad (10)$$

$$\text{and } q' = q - 1, q - 2, \dots, q - (q - 1) \text{ then}$$

$$T.S.N = \begin{cases} T.S.N + 1 & S(M(X_q)) = S(M(X_{q'})) \\ T.S.N + 0 & S(M(X_q)) \neq S(M(X_{q'})) \end{cases}$$

$$q = 1, 2, 3, \dots, e, \quad q' = q - 1, q - 2, \dots, q - (q - 1).$$

5. Fuzzification of the Mixed-Model Line Sequencing

In order to develop a fuzzy version of the mixed-model line sequencing, operation and travelling time had to represent a fuzzy number, which is applied for the processing

and scheduling time of each robot and machine. The fuzzy sets that are most appropriate to represent these values are triangular fuzzy numbers (TFN) due to the nature of the processing and scheduling times. Rather than assigning a specific value for each time variable as in the case of deterministic time methods, TFN establish extreme points to represent the most and least likely values of the individual variables. The use of TFN in-time modelling enables users to account for variability and ambiguity, similar to statistical distributions. However, TFN differ from statistical distributions such that they do not require prior knowledge or historical data when establishing their values. This is a major advantage of using TFN compared to statistics. TFN are achieved by defining the existing deterministic time as the most likely values for the new TFN. This implies that the deterministic value represents the fuzzy element in the set, with a membership value of 1. A value of 1 is added or subtracted from the most likely values in order to get the least likely values for the new TFN, with a membership value of 0. This same procedure is applied to adapt the deterministic scheduling times in the fuzzy domain [42]. No major changes are required when adapting these heuristics to handle fuzzy processing and scheduling times.

6. Solution Algorithm

In this study, a multiobjective optimization model is developed for a mixed-model process, comprising the manufacture of parts, assignment of parts to their respective machines, and assembly of parts to form the final products. The multi objective mixed model involves minimizing the make-span and setup cost. The operation time and travelling time are represented by fuzzy numbers. It will be highlighted that the steps for GA modelling include model, constraints, fitness function, and satisfying multiple objectives within the domain of model restrictions, fitness function, and multiobjective cases. These steps are carried out for various types of applications and include input data, fuzzy variables, and initialization of parameters, evaluation, crossover rate, mutation rate, and termination. The input data are coded by considering the sequence of part production stages, sequence of product assembly, and sequence of part allocation to each machine during the initialization of parameters. The proposed model consists of input data and fuzzy variables. Care is taken to ensure that the setup and operation time for each machine as well as the travelling time of the parts between various machines are accounted for. The evaluation stage involves developing the multiobjective fitness functions. Nine rules are used to control the crossover rate during the crossover stage, whereas eight rules were used to control the mutation rate during the mutation stage. Two rules are designed for the termination stage in order to control termination and gain optimum results. Sorting, selection of tournament candidates, elitism, and new population size were performed based on the existing conditions.

The model is developed based on the works of [1, 5, 8, 28, 34, 45, 46]. A contribution of this paper is the development of a multiobjective fuzzy mixed production line sequencing optimization model using the genetic algorithm approach

by integrating job-shop and assembly production lines for factories with modular layouts. The significance of this model is for those factories who want to produce the various kinds of products with fixed machine just by changing the sequencing of the products. The model helps the manager to sequence and schedule the production line easily and accurately by taking the market demand into consideration. The line sequencing is optimized based on the make-span and also the setup number of the production line. The flow chart of the proposed multiobjective fuzzy mixed-production assembly line sequencing model is shown in Figure 2.

7. Development of Multiobjective Genetic Algorithm

The procedure used in the development of the model is described in detail as follows.

Step 1 (input data). The first step involves identifying the machine number (M_j) such as CNC, NC, and robot, as well as assigning the parts (P_i) to their respective machines and robots based on the production and assembly line sequence. Also, it needs to identify the fuzzy completion time of the operation ($(\overline{OP})_{i,j}$) and fuzzy traveling time between machines in order to manufacture a part by $((\overline{T \cdot t})_{i,j})$.

Step 2 (initialization of parameters). Initialization involves setting the parameters of the GA, creating the scores for the simulation, and creating the first generation of chromosomes based on the Notations section. The results of the model will depend on the difficulties encountered during production planning and sequencing. The general schematic for reading data in mixed-production assembly line problems is presented in Figure 3. Figure 3 shows the number of finished products, the number of tasks involved in the total production line (J_q), the order of parts in the assembly ($A.S._p$), and the setup of each machine (S_u). Figure 3 also shows the number of parts (P_i) and the order for manufacturing each part. For example, the user proceeds to the next item (part) once the order of the manufacturing process is completed for each part. M_j represents the number of machines used for manufacturing parts and various products. The dashed arrows illustrate the coherence between these machines (M_j) during various phases of the production line. The fuzzy operation time ($\overline{OP}_{i,j}$) and fuzzy traveling time ($\overline{T \cdot t}_{i,j}$) are presented in Figure 3. The first column of Figure 3 (i.e., gene code (X_q)) shows the coding of each row, which will be discussed later.

From the data shown in Figure 3, the number of existing gene codes in the first column represents empty chromosomes (X_i), which are formed based on the job numbers. Once the chromosomes have been formed, the chromosomes will be filled with numbers via stochastic repeating. The numbers vary between 0 and the maximum number of genes in the chromosomes. The gene codes are filled up by random numbers, neglecting the encoding sequence of the produced chromosomes. Once the chromosomes have

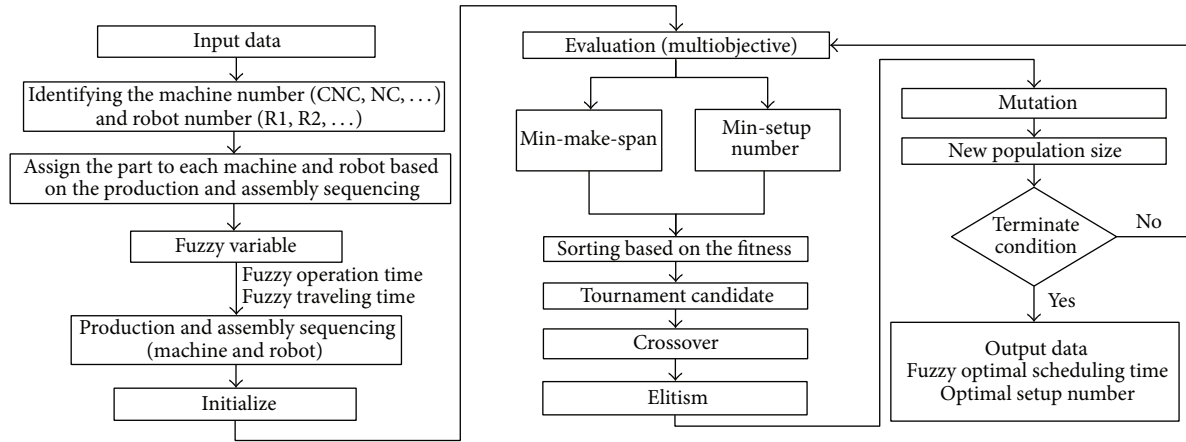


FIGURE 2: Flow chart of the proposed multiobjective fuzzy mixed-production assembly line sequencing model.

Gene code (X_q)	Jobs (J_q)	Product number (P.No. _{<i>h</i>})	Assembly sequence (A.S. _{<i>p</i>})	Setup (S_u)	Part (P_i)	Machine (M_j)	Fuzzy operation time (\overline{OP})	Fuzzy travelling time ($\overline{T \cdot t}$)
1	J_1	P.No ₁	A.S. ₀	S_A	P_1	M_1	$\overline{OP}(a_0, b_0, c_0)$	$\overline{T \cdot t}(a_0, b_0, c_0)$
2	J_2	P.No ₁	A.S. ₀	S_A	P_1	M_3	$\overline{OP}(a_1, b_1, c_1)$	$\overline{T \cdot t}(a_1, b_1, c_1)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_1	M_m	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_2	M_2	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_B	P_2	M_3	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_2	M_1	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_B	P_2	M_m	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_C	P_3	M_4	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_3	M_1	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_3	M_3	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_B	P_3	M_7	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₀	S_A	P_3	M_m	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₁	S_B	P_4	M_6	⋮	⋮
⋮	⋮	P.No ₁	A.S. ₂	S_C	P_5	M_7	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₁	A.S. _{<i>v</i>}	S_A	P_n	M_m	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_B	P_4	M_5	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_A	P_4	M_2	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_A	P_4	M_4	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_A	P_4	M_m	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_B	P_5	M_4	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_A	P_5	M_5	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_A	P_5	M_6	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₀	S_C	P_5	M_m	⋮	⋮
⋮	⋮	P.No ₂	A.S. ₁	S_B	P_6	M_5	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	P.No ₂	A.S. _{<i>v</i>}	S_A	P_n	M_m	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X_e	J_e	P.No _{<i>k</i>}	A.S. _{<i>v</i>}	S_z	P_n	M_m	$\overline{OP}(a_n, b_n, c_n)$	$\overline{T \cdot t}(a_n, b_n, c_n)$

FIGURE 3: General schematic of reading data for the mixed-model assembly line problem.

been filled with random numbers, the classified gene code numbers are ranked as a possible sequence which can be used by the GA. The steps are described briefly as follows.

Identification of the First of Each Part Type's Gene Code. The first of each part type's gene code is identified using (11), whereby each part has a sequence of gene codes with f -elements, where W_z , $z = 1, 2, \dots, f$, have the same part number. Consider

$$X_q - W_1 \leq (f - 1). \quad (11)$$

Classification Based on the Product's Part Sequence. Each gene code number in the chromosome line is compared to the previous one. Three conditions are used when comparing the gene code numbers and are described below.

- (1) The first or existing gene code number fills up the new chromosome without any changes in the gene code number.
- (2) If the gene code number is the same as other gene code numbers in the chromosome, the chromosome is filled up with the addition of one number out of the existing gene code numbers.
- (3) If the gene code number is different from other gene code numbers within the chromosome, the existing gene code number fills up the chromosome without any changes in the gene code number.

Classification Based on the Assembly Sequence. Once the product's part sequence has been classified, the chromosome is filled up using the assembly sequence number relating to a gene code (X_q) of the specific chromosome. Finally, the final gene code numbers are ranked by two sequence filters, which accounts for the assembly sequence code and product's part sequence.

Step 3 (multiobjective evaluation). After initializing the population size, each chromosome must be evaluated during each generation of the selection process. The fitness function value involves evaluating each chromosome by minimizing the make-span and considering the product number and assembly sequence for a gene code (X_q) which are defined as (1) through (8). Moreover, the evaluation of each chromosome for the second objective function to minimize the setup number in mixed-model line sequencing is defined as (9) through (10). Finally the total fitness values of the efficient frontiers are calculated based on these two objectives [35]. This is accomplished by looking up the score of each gene in the chromosome, by means of adding and averaging the scores for the chromosome. This step is repeated for each possible chromosome (X_q) in the population size. The elite chromosome of the generation is determined as part of the evaluation process.

Step 4 (sorting). The total fitness value is calculated for the population size based on Step 3. Calculations are carried out

in ascending order from "Min" to "Max" for the total fitness values of all chromosomes.

Step 5 (tournament candidate selection). The chromosomes are selected for the next generation using the tournament selection scheme, which is a proportionate random selection method. Two parent chromosomes are selected from a population according to their fitness function values.

The main characteristics of tournament selection are summarized as follows.

- (i) Tournament selection is quite useful in certain situations, such as multiobjective optimization.
- (ii) Tournament selection uses only local information.
- (iii) Tournament selection is easily implemented with low time complexity.
- (iv) Tournament selection can be easily implemented in a parallel environment.

However, tournament selection also suffers from selection bias, which means that the best one will not be selected if it is very unlucky.

Step 6 (crossover selection). Crossover probability crosses over parents to form new offspring (children). In the crossover phase, all chromosomes (except for the elite chromosome) are paired up and crossed over with a probability crossover rate. Crossover is accomplished by choosing a site randomly along the length of the chromosome and exchanging the genes of two chromosomes (parents) for each gene past this crossover site. The steps for crossover are as follows.

- (1) Identify the number of different requirements for manufactured products ($P.No._h$).
- (2) Create the one random number between product type numbers ($P.No._h$) as follows:

$$\begin{aligned} &\text{for (int } h = 1; i < P.No._h; h++) \\ &\{ \\ &\quad \text{Product Random} = \text{rand}() \% \text{Product}; \\ &\}. \end{aligned} \quad (12)$$

- (3) Identify the gene code number (X_q) for the selected product obtained from the previous step.
- (4) Search and identify the gene code from parent A based on random product selection and transfer these gene codes to child B, which is exactly at the same gene location.
- (5) Search and identify the gene code from parent B based on random product selection and transfer the gene code to child A, which is exactly at the same gene location.
- (6) Transfer the remaining gene code from parent A to the gene blank of child A.

- (7) Transfer the remaining gene code from parent B to the gene blank of child B.
- (8) Calculate the number of crossovers based on the crossover rate ($CR(t)$) from Step 9 and population size (PS) using

$$\text{Number of crossovers} = \frac{CR(t) \times PS}{2}. \quad (13)$$

- (9) Once new offspring have been created, the new offspring will have previous chromosomes in the current generation.

Step 7 (elitism). In crossover operation, the worst or weakest chromosomes will fade away, whereas the characteristics of the chromosomes will change continuously during mutation operation. The elite chromosome will not be subjected to mutation in the next generation. Consequently, GA does not lead to annihilation since several chromosomes (one, two, or three) from each generation are transferred directly to the next generation. Mutation is not applied on chromosomes which are immune. It is possible to maintain a fixed fitness value in some generations, but they will never deteriorate. The first three best chromosomes are selected in the elitism step.

Step 8 (mutation). Following the crossover operation, the genes will mutate to any of the codes with a mutation rate for each gene in the chromosomes, with the exception of the elite chromosome. When the crossover and mutation operations are complete, the chromosomes will be evaluated for another round of selection and reproduction. Considering elitism and after identifying the parts in which mutation will be applied, the number of mutations in each generation is calculated using (14) based on the mutation rate ($Pm(t)$), population size (PS), and maximum gene code ($Max .X_q$) as follows:

$$\text{Number of mutations} \cong [(PS \times Max .X_i) \times Pm]. \quad (14)$$

After identifying the number of gene mutations, a set of rules needs to be devised for the mutation of genes from point A to point B and vice versa, while focusing on the stability of the chromosome sequence. It is important that the sequence of the chromosomes is not displaced. There are eight sets of rules for this step which are classified into two groups as follows.

- (1) Four rules are used to check the mutation based on the part sequence (Rules 1 and 2) and product assembly (Rules 3 and 4) of genes from A to B.
- (2) Four rules are used to check the mutation based on the part sequence (Rules 5 and 6) and product assembly (Rules 7 and 8) of genes from B to A.

The first group of gene mutation from A to B is illustrated in Figure 4. Likewise, the second group of gene mutation from B to A is illustrated in Figure 5.

Note.

- (i) Care should be taken to ensure that the position of A is before B.

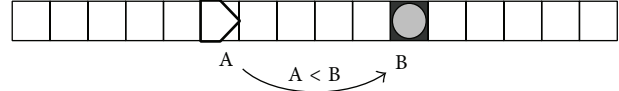


FIGURE 4: General overview of the first group of gene mutation.

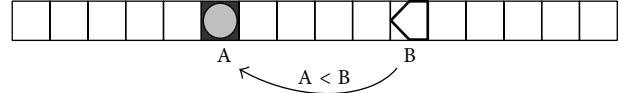


FIGURE 5: General overview of the second group of gene mutation.

- (ii) The above eight mutation rules should be checked thoroughly to ensure their ideal applications. In the adverse case, the above eight rules ought to be done over.

Step 9 (new population size). Having performed crossover, elitism, and mutation operations, the most ideal chromosomes of the current generation are compared and evaluated to identify its total value, after checking its termination in the following step.

Step 10 (termination). The loop of chromosome generations is terminated when certain conditions are met. The elite chromosome is returned as the best solution once the termination criteria are met. The termination criteria are listed below.

- (1) If the number of generations reaches its maximum, the loop of chromosome generations is terminated.
- (2) If there are no changes in the elite solution (i.e., no changes in fitness function value), the loop of chromosome generations is terminated using

$$\text{Fitness Value}(X_q) - \text{Fitness Value}(X_{q+1}) \leq 0.0001. \quad (15)$$

8. Computational Results

The results of the multiobjective fuzzy mixed-model assembly line model are presented in this section. Each metaheuristic model is verified, validated, and tested using a test-bed. It seems natural that one of the crucial elements for comparing heuristics is the test-bed onto which the heuristics are tested. Hence, test-beds are essential when comparing two metaheuristics. Silberholz and Golden [47] studied how to create and classify instances in a new test-bed and how to ensure that other researchers have access to the problems for future metaheuristic comparisons. Silberholz and Golden [47] proposed that two types of test-beds may be used for comparing metaheuristics, namely, (1) existing test-beds and (2) new test-beds. Existing test-beds are used when comparing a new metaheuristic to an existing one. It is advantageous to test problem instances that have been tested in previous studies. The results will be compared on a by-instance basis, allowing relative gap calculations between the two heuristics. New test-beds are developed when an existing test-bed is insufficient. For instance, there are no test-beds available for

TABLE 1: Input data for numerical example.

Number of jobs (J_q)	Number of products (P.No. $_h$)	Number of parts (P_i)	Number of lathe machines, CNC, and robots (M_j)	Number of machine setup tools (S_u)
50	4	20	5	5

cases where metaheuristics are written specifically for a new problem and thus a new test-bed needs to be developed for this purpose. A new test-bed also needs to be developed if the existing test-beds are insufficient, which are often due to the fact that the test-beds are too small to effectively test a heuristic. There are two points which need to be addressed when developing new test-beds, that is, the purpose of developing the test-beds and the accessibility of new test instances [47]. The purpose of a problem suite is to emulate real-world problem instances with a variety of test cases and difficulty levels. When creating a new test-bed, the focus is to provide others with accessibility to problem instances. This enables other researchers to make comparisons easily, while ensuring that the problem instances are widely used. One way to ensure this is to create a simple generating function for the problem instances. Capturing the real aspects of a problem is particularly significant when developing a new test-bed. In this section, the results are presented based on the development of new test-beds due to the fact that the existing test-beds are inadequate. This is followed by a discussion on the identification of the optimum scheduling time for a fuzzy mixed-model assembly line sequencing problem.

8.1. Input Data. A hypothetical numerical example is designed to test the fuzzy mixed-model assembly line sequencing problem. The input data of the hypothetical numerical example is given in Table 1, consisting of 50 jobs and 20 parts in order to produce four products. There are five machine tools (one lathe, two CNC, and two robots) assigned to assemble four products.

8.2. Initialization of Parameters and Fuzzy Variables. The initialization of parameters for the mixed-model assembly line sequencing example is shown in Table 3. It can be seen that 50 jobs (J_q ; $q = 1, 2, \dots, 50$) are required to produce 20 parts (P_i ; $i = 0, 1, \dots, 19$) and these parts are assembled to produce four types of product (P.No. $_h$; $h = 1, 2, 3, 4$). The number of tool changes is 5 (S_u ; $u = A, B, C, D, \text{ and } E$) and these tools are assigned to five machines ($M(X_q)$; $q = 0, 1, 2, 3, 4$). The job sequence is dependent upon the part and product assembly and is described as follows. First, the job number is assigned to produce the first product, ranging from 1 to 10 (P.No. $_1$, $h = 1$; J_q , $q = 1, 2, \dots, 10$). It will be highlighted that there are 10 jobs in this case and they are sequenced to produce three parts according to the following order. Job numbers 1, 2, and 3 are assigned to produce Part (0) ((P_i) , $i = 0 \rightarrow J_1, J_2, J_3$), while job numbers 4 and 5 are assigned to produce Part (1) ((P_i) , $i = 1 \rightarrow J_4, J_5$). Job numbers 6 and 7 are assigned to produce Part (2) ((P_i) , $i = 2 \rightarrow J_6, J_7$), whereas job numbers 8, 9, and 10 are assigned to produce Part (3), which is a subpart of the

product assembly ((P_i) , $i = 3 \rightarrow J_8, J_9, J_{10}$). Production of the second, third, and fourth products is based on the sequence described for Product (1), as shown in Table 2. The fuzzy processing time of each job (J_q , $q = 1, 2, \dots, 10$) is defined as a triplet (a_1, a_2, a_3) . The total operating time is based on the fuzzy triangular time and is required to complete the jobs sequentially when producing each part, in which each part is assigned to a machine (M_j , $j = 0, 1, 2, 3, 4$). The total operating time is defined as $(\widetilde{OP}_{i,j} = (a_1, a_2, a_3); i = 0, 1, 2, \dots, 19; j = 0, 1, \dots, 5; a = \text{time})$. The total travelling time based on the above information is defined as $(\widetilde{T} \cdot t_{i,j} = (a_1, a_2, a_3); i = 0, 1, 2, \dots, 19; j = 0, 1, \dots, 5; a = \text{time})$. The operation and travelling time are fuzzy numbers, which are indicated by “ a_1 ,” “ a_2 ,” and “ a_3 .” The parameters “ a_1 ,” “ a_2 ,” and “ a_3 ” represent the optimistic time, normal time, and pessimistic time, respectively.

8.3. Model Development. A numerical example for mixed-model assembly line sequencing is presented in Figure 6, based on the parameters listed in Table 2. The solid arrows represent the order of the product line (sequence of part production), discrete part manufacturing, and assembly, leading to the finished products, as indicated by the job numbers. Suppose that the production process involves manufacturing four products using the same assembly line. In other words, four different products are manufactured simultaneously on the assembly line, and hence the problem is a mixed-model assembly line problem. In this example, 20 parts need to be manufactured using five machines. From Figure 6, the order of the production of parts is represented by the dashed rectangles and is termed as the process line. The assembly lines are represented by the dotted rectangles. The assignment of parts to their respective machines based on job number is illustrated in Figure 7.

8.4. Multiobjective Evaluation. The optimized optimistic, medium, and pessimistic starting times ($\widetilde{S} \cdot t$) are shown in the sixth column of Table 3, based on the GA encoding chromosome sequence (X_q) and job sequence (J_q). The final optimized optimistic start time is 59.5, whereas the final medium and pessimistic start times are 69 and 85.5, respectively.

The scheduling of existing optimistic time for mixed-model assembly line sequencing (machine versus optimistic time) is shown in Figure 8(a), whereas the scheduling for optimized optimistic scheduling time is shown in Figure 8(b). This chart is created to demonstrate the allocated time and setup number based on the part assembly sequence for each mixed-model. The existing optimistic total scheduling time is obtained to be 166 min, whereas the optimized optimistic total scheduled time is 62.5 min, which indicates significant

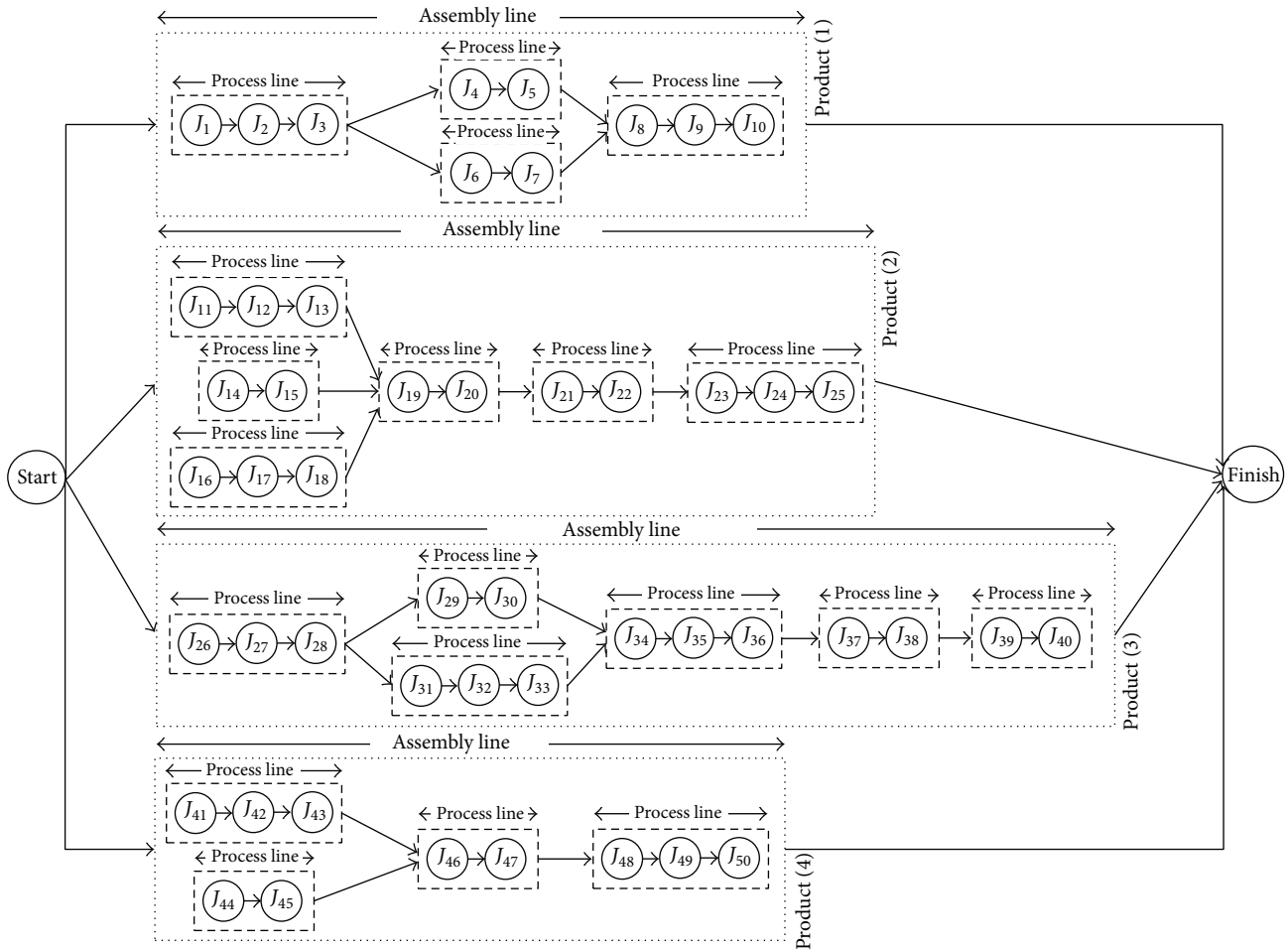


FIGURE 6: Example of mixed-model assembly line sequencing (50 jobs, 20 parts, 4 products).

time-savings. Moreover, the total fuzzy setup numbers for the existing and optimized optimistic data are 44 and 36, respectively.

The percentage of the existing and optimized fuzzy numbers which represent the efficiency and idle time of the machines is presented in Table 4. The efficiency and idle time are categorized as optimistic, medium, and pessimistic. The lowest efficiency is obtained for Machine (4), in which the optimistic, medium, and pessimistic times which are 3.3%, 6.4%, and 3.9% increase to 8.7%, 21.9%, and 11.7%. The highest idle time is also obtained for Machine (4), whereby the optimistic, medium, and pessimistic times which are 96.7%, 93.6%, and 96.1% decrease to 91.35%, 78.1%, and 88.3%.

The final results based on the existing and optimized data are shown in Table 5. The overall results show that the existing fuzzy data is improved by optimization. The total fuzzy existing scheduling time is optimized from 166, 250, and 266 to 62.5, 73, and 88.5. The total fuzzy setup numbers for the existing and optimized data are 44, 44, and 44 and 36, 36, and 39, respectively. The total fuzzy existing efficient frontier is 105, 147, and 155, while the optimized ones are 49.25, 54.5, and 63.75. The total fuzzy existing cycle and

setup time are optimized from 254, 338, and 354 to 135, 145, and 168. The total fuzzy operation setup times for the existing and optimized data are 88, 88, and 88 and 72, 72, and 78, respectively. The total fuzzy existing changing setup cost is optimized from \$ 3520, \$ 3520, and \$ 3520 to \$ 2880, \$ 2880, and \$ 3120. The total fuzzy units produced per day for the existing and optimized data are approximately 1.80, 1.92, and 2.89 and 5.33, 6.58, and 7.62, respectively. The total existing fuzzy percentage efficiency is optimized from 10.32%, 10.90%, and 11.45% to 30.16%, 32.22%, and 35.34%.

9. Conclusion

It is known that mixed-model assembly line sequencing is a problem which involves job-shop and assembly line with multiple conflicting objectives, namely, minimizing the setup time and make-span. These objectives have been achieved successfully and tested using a hypothetical numerical example. An optimization model is developed using GA for the mixed-model assembly line sequencing problem in order to address conflicting objectives which involve minimizing

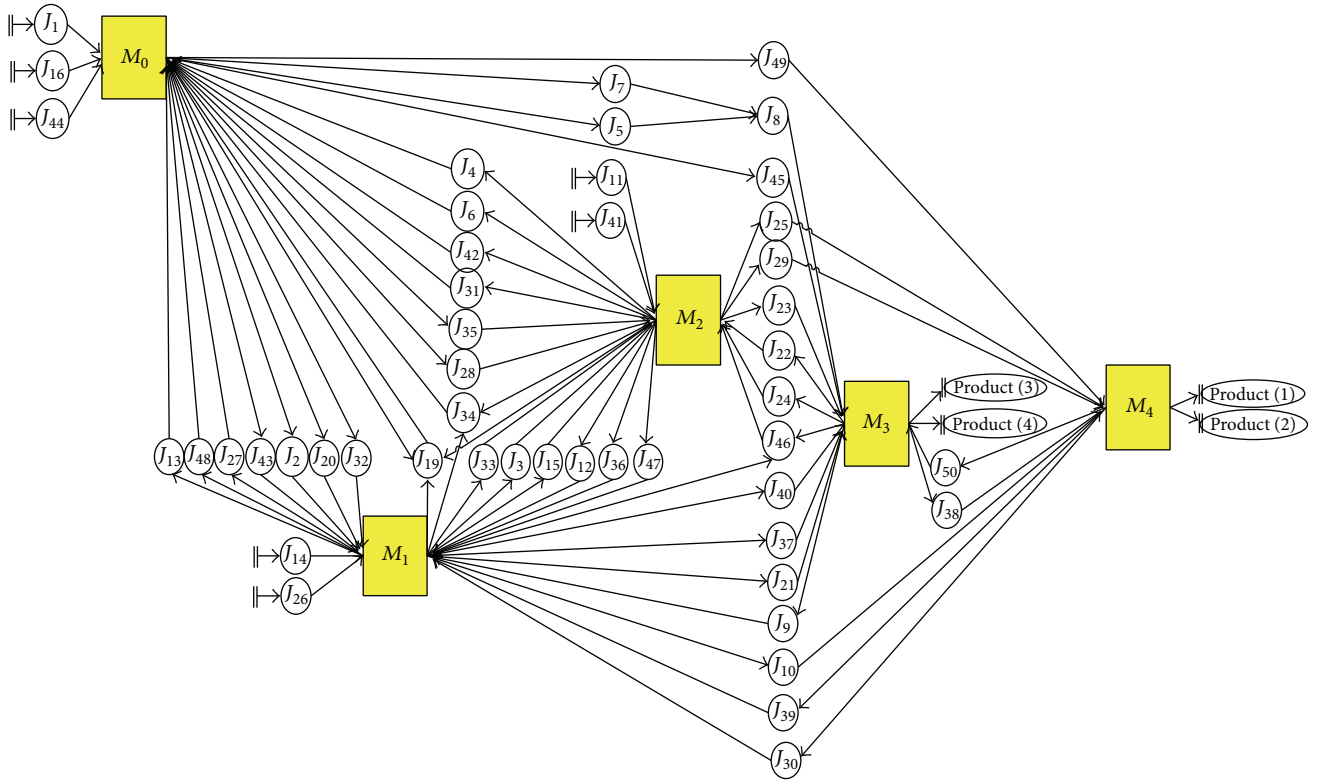
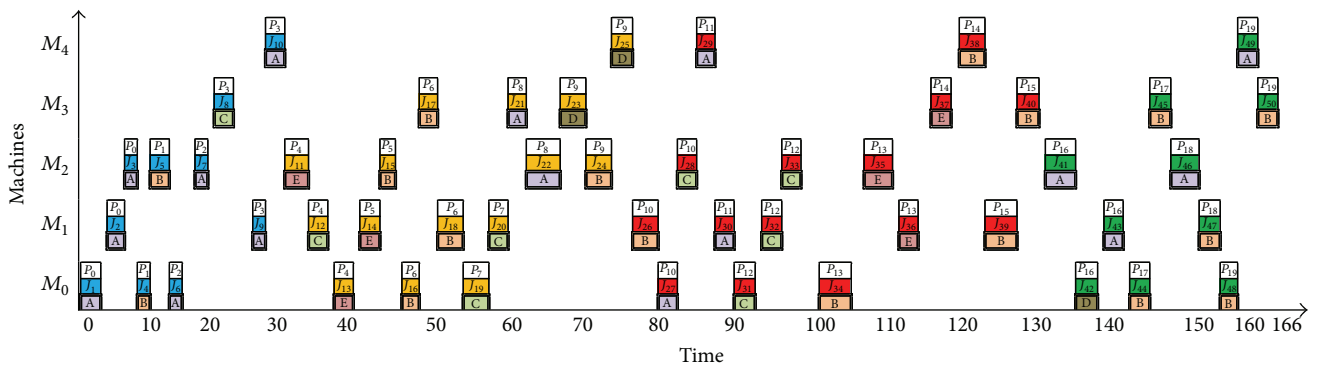
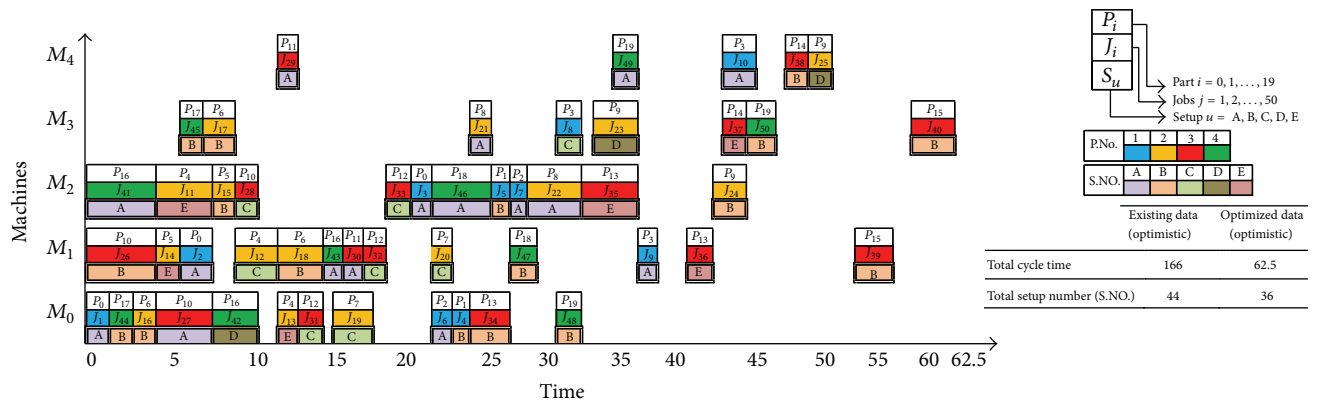


FIGURE 7: Assignment of parts to their respective machines based on the example for mixed-model assembly line sequencing.



(a) Existing optimistic time scheduling



(b) Optimized optimistic time scheduling

FIGURE 8: Optimistic time scheduling for mixed-model assembly line sequencing.

TABLE 2: Fuzzy variables and initialization of parameters.

X_q	J_q	P.No. _{<i>h</i>}	A.S. _{<i>p</i>}	S_u	P_i	M_j	$\overline{OP}_{i,j}$			$\overline{T} \cdot \overline{t}_{i,j}$		
1	J_1	P.No. ₁	A.S. ₀	S_A	P_0	M_0	1.50	2	2.50	0.50	1	1.5
2	J_2	P.No. ₁	A.S. ₀	S_A	P_0	M_1	2.50	3	3.50	4.50	5	5.5
3	J_3	P.No. ₁	A.S. ₀	S_A	P_0	M_2	0.50	1	1.50	0.50	1	1.5
4	J_4	P.No. ₁	A.S. ₁	S_B	P_1	M_0	0.50	1	1.50	1.50	2	2.5
5	J_5	P.No. ₁	A.S. ₁	S_B	P_1	M_2	1.50	2	2.50	2.50	3	3.5
6	J_6	P.No. ₁	A.S. ₁	S_A	P_2	M_0	1.50	2	2.50	2.50	3	3.5
7	J_7	P.No. ₁	A.S. ₁	S_A	P_2	M_2	0.50	1	1.50	2.50	3	3.5
8	J_8	P.No. ₁	A.S. ₂	S_C	P_3	M_3	1.50	2	2.50	3.50	4	4.5
9	J_9	P.No. ₁	A.S. ₂	S_A	P_3	M_1	2.50	3	3.50	4.50	5	5.5
10	J_{10}	P.No. ₁	A.S. ₂	S_A	P_3	M_4	1.50	2	2.50	0.50	1	1.5
11	J_{11}	P.No. ₂	A.S. ₀	S_E	P_4	M_2	3.50	4	4.50	1.50	2	2.5
12	J_{12}	P.No. ₂	A.S. ₀	S_C	P_4	M_1	2.50	3	3.50	0.50	1	1.5
13	J_{13}	P.No. ₂	A.S. ₀	S_E	P_4	M_0	1.50	2	2.50	2.50	3	3.5
14	J_{14}	P.No. ₂	A.S. ₀	S_E	P_5	M_1	0.50	1	1.50	1.50	2	2.5
15	J_{15}	P.No. ₂	A.S. ₀	S_B	P_5	M_2	0.50	1	1.50	0.50	1	1.5
16	J_{16}	P.No. ₂	A.S. ₀	S_B	P_6	M_0	1.50	2	2.50	3.50	4	4.5
17	J_{17}	P.No. ₂	A.S. ₀	S_B	P_6	M_3	1.50	2	2.50	0.50	1	1.5
18	J_{18}	P.No. ₂	A.S. ₀	S_B	P_6	M_1	2.50	3	3.50	1.50	2	2.5
19	J_{19}	P.No. ₂	A.S. ₁	S_C	P_7	M_0	2.50	3	3.50	2.50	3	3.5
20	J_{20}	P.No. ₂	A.S. ₁	S_C	P_7	M_1	0.50	1	1.50	1.50	2	2.5
21	J_{21}	P.No. ₂	A.S. ₂	S_A	P_8	M_3	0.50	1	1.50	0.50	1	1.5
22	J_{22}	P.No. ₂	A.S. ₂	S_A	P_8	M_2	4.50	5	5.50	1.50	2	2.5
23	J_{23}	P.No. ₂	A.S. ₃	S_D	P_9	M_3	2.50	3	3.50	2.50	3	3.5
24	J_{24}	P.No. ₂	A.S. ₃	S_B	P_9	M_2	1.50	2	2.50	1.50	2	2.5
25	J_{25}	P.No. ₂	A.S. ₃	S_D	P_9	M_4	0.50	1	1.50	4.50	5	5.5
26	J_{26}	P.No. ₃	A.S. ₀	S_B	P_{10}	M_1	3.50	4	4.50	0.50	1	1.5
27	J_{27}	P.No. ₃	A.S. ₀	S_A	P_{10}	M_0	2.50	3	3.50	0.50	1	1.5
28	J_{28}	P.No. ₃	A.S. ₀	S_C	P_{10}	M_2	1.50	2	2.50	2.50	3	3.5
29	J_{29}	P.No. ₃	A.S. ₁	S_A	P_{11}	M_4	0.50	1	1.50	1.50	2	2.5
30	J_{30}	P.No. ₃	A.S. ₁	S_A	P_{11}	M_1	1.50	2	2.50	1.50	2	2.5
31	J_{31}	P.No. ₃	A.S. ₁	S_C	P_{12}	M_0	0.50	1	1.50	1.50	2	2.5
32	J_{32}	P.No. ₃	A.S. ₁	S_C	P_{12}	M_1	1.50	2	2.50	0.50	1	1.5
33	J_{33}	P.No. ₃	A.S. ₁	S_C	P_{12}	M_2	2.50	3	3.50	2.50	3	3.5
34	J_{34}	P.No. ₃	A.S. ₂	S_B	P_{13}	M_0	3.50	4	4.50	3.50	4	4.5
35	J_{35}	P.No. ₃	A.S. ₂	S_E	P_{13}	M_2	4.50	5	5.50	4.50	5	5.5
36	J_{36}	P.No. ₃	A.S. ₂	S_E	P_{13}	M_1	1.50	2	2.50	0.50	1	1.5
37	J_{37}	P.No. ₃	A.S. ₃	S_E	P_{14}	M_3	0.50	1	1.50	4.50	5	5.5
38	J_{38}	P.No. ₃	A.S. ₃	S_B	P_{14}	M_4	2.50	3	3.50	2.50	3	3.5
39	J_{39}	P.No. ₃	A.S. ₄	S_B	P_{15}	M_1	4.50	5	5.50	1.50	2	2.5
40	J_{40}	P.No. ₃	A.S. ₄	S_B	P_{15}	M_3	3.50	4	4.50	0.50	1	1.5
41	J_{41}	P.No. ₄	A.S. ₀	S_A	P_{16}	M_2	3.50	4	4.50	0.50	1	1.5
42	J_{42}	P.No. ₄	A.S. ₀	S_D	P_{16}	M_0	2.50	3	3.50	2.50	3	3.5
43	J_{43}	P.No. ₄	A.S. ₀	S_A	P_{16}	M_1	1.50	2	2.50	1.50	2	2.5
44	J_{44}	P.No. ₄	A.S. ₀	S_B	P_{17}	M_0	0.50	1	1.50	3.50	4	4.5
45	J_{45}	P.No. ₄	A.S. ₀	S_B	P_{17}	M_3	0.50	1	1.50	0.50	1	1.5
46	J_{46}	P.No. ₄	A.S. ₁	S_A	P_{18}	M_2	3.50	4	4.50	1.50	2	2.5
47	J_{47}	P.No. ₄	A.S. ₁	S_B	P_{18}	M_1	2.50	3	3.50	1.50	2	2.5
48	J_{48}	P.No. ₄	A.S. ₂	S_B	P_{19}	M_0	1.50	2	2.50	2.50	3	3.5
49	J_{49}	P.No. ₄	A.S. ₂	S_A	P_{19}	M_4	0.50	1	1.50	1.50	2	2.5
50	J_{50}	P.No. ₄	A.S. ₂	S_B	P_{19}	M_3	3.50	4	4.50	2.50	3	3.5

TABLE 3: Optimized scheduling time based on optimistic, medium, and pessimistic data.

Optimize optimistic results							Optimize medium results							Optimize pessimistic results						
X_q	J_q	P.No. _h	S_u	P_i	M_j	S.T	X_q	J_q	P.No. _h	S_u	P_i	M_j	S.T	X_q	J_q	P.No. _h	S_u	P_i	M_j	S.T
41	J_{41}	P.No. ₄	S_A	P_{16}	M_2	0.5	41	J_{41}	P.No. ₄	S_A	P_{16}	M_2	1	26	J_{26}	P.No. ₃	S_B	P_{10}	M_1	1.5
1	J_1	P.No. ₁	S_A	P_0	M_0	0.5	1	J_1	P.No. ₁	S_A	P_0	M_0	1	11	J_{11}	P.No. ₂	S_E	P_4	M_2	2.5
26	J_{26}	P.No. ₃	S_B	P_{10}	M_1	0.5	16	J_{16}	P.No. ₂	S_B	P_6	M_0	3	1	J_1	P.No. ₁	S_A	P_0	M_0	1.5
44	J_{44}	P.No. ₄	S_B	P_{17}	M_0	2	26	J_{26}	P.No. ₃	S_B	P_{10}	M_1	1	12	J_{12}	P.No. ₂	S_C	P_4	M_1	9.5
16	J_{16}	P.No. ₂	S_B	P_6	M_0	2.5	44	J_{44}	P.No. ₄	S_B	P_{17}	M_0	5	41	J_{41}	P.No. ₄	S_A	P_{16}	M_2	7
14	J_{14}	P.No. ₂	S_B	P_5	M_1	4	2	J_2	P.No. ₁	S_A	P_0	M_1	5	14	J_{14}	P.No. ₂	S_E	P_5	M_1	13
11	J_{11}	P.No. ₂	S_E	P_4	M_2	4	11	J_{11}	P.No. ₂	S_E	P_4	M_2	5	27	J_{27}	P.No. ₃	S_A	P_{10}	M_0	7.5
45	J_{45}	P.No. ₄	S_B	P_{17}	M_3	6	42	J_{42}	P.No. ₄	S_D	P_{16}	M_0	6	2	J_2	P.No. ₁	S_A	P_0	M_1	14.5
2	J_2	P.No. ₁	S_A	P_0	M_1	4.5	45	J_{45}	P.No. ₄	S_B	P_{17}	M_3	10	16	J_{16}	P.No. ₂	S_B	P_6	M_0	11
17	J_{17}	P.No. ₂	S_B	P_6	M_3	7.5	17	J_{17}	P.No. ₂	S_B	P_6	M_3	12	28	J_{28}	P.No. ₃	S_C	P_{10}	M_2	12.5
12	J_{12}	P.No. ₂	S_C	P_4	M_1	9	14	J_{14}	P.No. ₂	S_E	P_5	M_1	8	42	J_{42}	P.No. ₄	S_D	P_{16}	M_0	13.5
27	J_{27}	P.No. ₃	S_A	P_{10}	M_0	4.5	27	J_{27}	P.No. ₃	S_A	P_{10}	M_0	9	15	J_{15}	P.No. ₂	S_B	P_5	M_2	17
15	J_{15}	P.No. ₂	S_B	P_5	M_2	7.5	12	J_{12}	P.No. ₂	S_C	P_4	M_1	11	44	J_{44}	P.No. ₄	S_B	P_{17}	M_0	17
42	J_{42}	P.No. ₄	S_D	P_{16}	M_0	7	15	J_{15}	P.No. ₂	S_B	P_5	M_2	11	43	J_{43}	P.No. ₄	S_A	P_{16}	M_1	20.5
13	J_{13}	P.No. ₂	S_E	P_4	M_0	12	46	J_{46}	P.No. ₄	S_A	P_{18}	M_2	13	17	J_{17}	P.No. ₂	S_B	P_6	M_3	18
18	J_{18}	P.No. ₂	S_B	P_6	M_1	11.5	43	J_{43}	P.No. ₄	S_A	P_{16}	M_1	14	13	J_{13}	P.No. ₂	S_E	P_4	M_0	18.5
28	J_{28}	P.No. ₃	S_C	P_{10}	M_2	8	13	J_{13}	P.No. ₂	S_E	P_4	M_0	15	18	J_{18}	P.No. ₂	S_B	P_6	M_1	23
43	J_{43}	P.No. ₄	S_A	P_{16}	M_1	14	3	J_3	P.No. ₁	S_A	P_0	M_2	17	29	J_{29}	P.No. ₃	S_A	P_{11}	M_4	18.5
29	J_{29}	P.No. ₃	S_A	P_{11}	M_4	12	18	J_{18}	P.No. ₂	S_B	P_6	M_1	16	45	J_{45}	P.No. ₄	S_B	P_{17}	M_3	23
31	J_{31}	P.No. ₃	S_C	P_{12}	M_0	13.5	31	J_{31}	P.No. ₃	S_C	P_{12}	M_0	17	3	J_3	P.No. ₁	S_A	P_0	M_2	23.5
30	J_{30}	P.No. ₃	S_A	P_{11}	M_1	15.5	32	J_{32}	P.No. ₃	S_C	P_{12}	M_1	20	31	J_{31}	P.No. ₃	S_C	P_{12}	M_0	21
32	J_{32}	P.No. ₃	S_C	P_{12}	M_1	17	28	J_{28}	P.No. ₃	S_C	P_{10}	M_2	18	32	J_{32}	P.No. ₃	S_C	P_{12}	M_1	26.5
33	J_{33}	P.No. ₃	S_C	P_{12}	M_2	19	19	J_{19}	P.No. ₂	S_C	P_7	M_0	21	46	J_{46}	P.No. ₄	S_A	P_{18}	M_2	26
19	J_{19}	P.No. ₂	S_C	P_7	M_0	16	29	J_{29}	P.No. ₃	S_A	P_{11}	M_4	23	30	J_{30}	P.No. ₃	S_A	P_{11}	M_1	29
3	J_3	P.No. ₁	S_A	P_0	M_2	21.5	6	J_6	P.No. ₁	S_A	P_2	M_0	24	19	J_{19}	P.No. ₂	S_C	P_7	M_0	29
6	J_6	P.No. ₁	S_A	P_2	M_0	22.5	20	J_{20}	P.No. ₂	S_C	P_7	M_1	27	6	J_6	P.No. ₁	S_A	P_2	M_0	32.5
4	J_4	P.No. ₁	S_B	P_1	M_0	24	33	J_{33}	P.No. ₃	S_C	P_{12}	M_2	23	33	J_{33}	P.No. ₃	S_C	P_{12}	M_2	30.5
20	J_{20}	P.No. ₂	S_C	P_7	M_1	22.5	4	J_4	P.No. ₁	S_B	P_1	M_0	26	7	J_7	P.No. ₁	S_A	P_2	M_2	38.5
46	J_{46}	P.No. ₄	S_A	P_{18}	M_2	22.5	5	J_5	P.No. ₁	S_B	P_1	M_2	29	47	J_{47}	P.No. ₄	S_B	P_{18}	M_1	33
5	J_5	P.No. ₁	S_B	P_1	M_2	26	34	J_{34}	P.No. ₃	S_B	P_{13}	M_0	29	4	J_4	P.No. ₁	S_B	P_1	M_0	35
21	J_{21}	P.No. ₂	S_A	P_8	M_3	24.5	47	J_{47}	P.No. ₄	S_B	P_{18}	M_1	29	5	J_5	P.No. ₁	S_B	P_1	M_2	40
7	J_7	P.No. ₁	S_A	P_2	M_2	27.5	21	J_{21}	P.No. ₂	S_A	P_8	M_3	30	20	J_{20}	P.No. ₂	S_C	P_7	M_1	36.5
47	J_{47}	P.No. ₄	S_B	P_{18}	M_1	27.5	7	J_7	P.No. ₁	S_A	P_2	M_2	31	34	J_{34}	P.No. ₃	S_B	P_{13}	M_0	37.5
22	J_{22}	P.No. ₂	S_A	P_8	M_2	28	30	J_{30}	P.No. ₃	S_A	P_{11}	M_1	32	21	J_{21}	P.No. ₂	S_A	P_8	M_3	40.5
34	J_{34}	P.No. ₃	S_B	P_{13}	M_0	24.5	22	J_{22}	P.No. ₂	S_A	P_8	M_2	32	35	J_{35}	P.No. ₃	S_E	P_{13}	M_2	46.5
48	J_{48}	P.No. ₄	S_B	P_{19}	M_0	31.5	48	J_{48}	P.No. ₄	S_B	P_{19}	M_0	34	48	J_{38}	P.No. ₄	S_B	P_{19}	M_0	42
49	J_{49}	P.No. ₄	S_A	P_{19}	M_4	35.5	35	J_{35}	P.No. ₃	S_E	P_{13}	M_2	37	49	J_{49}	P.No. ₄	S_A	P_{19}	M_4	48
8	J_8	P.No. ₁	S_C	P_3	M_3	31.5	8	J_8	P.No. ₁	S_C	P_3	M_3	36	8	J_8	P.No. ₁	S_C	P_3	M_3	46
35	J_{35}	P.No. ₃	S_E	P_{13}	M_2	32.5	49	J_{49}	P.No. ₄	S_A	P_{19}	M_4	39	9	J_9	P.No. ₁	S_A	P_3	M_1	53
9	J_9	P.No. ₁	S_A	P_3	M_1	36.5	9	J_9	P.No. ₁	S_A	P_3	M_1	42	22	J_{22}	P.No. ₂	S_A	P_8	M_2	52
10	J_{10}	P.No. ₁	S_A	P_3	M_4	43.5	23	J_{23}	P.No. ₂	S_D	P_9	M_3	39	36	J_{36}	P.No. ₃	S_E	P_{13}	M_1	57.5
23	J_{23}	P.No. ₂	S_D	P_9	M_3	34	36	J_{36}	P.No. ₃	S_E	P_{13}	M_1	47	10	J_{10}	P.No. ₁	S_A	P_3	M_4	62
36	J_{36}	P.No. ₃	S_E	P_{13}	M_1	41.5	10	J_{10}	P.No. ₁	S_A	P_3	M_4	50	23	J_{23}	P.No. ₂	S_D	P_9	M_3	60
37	J_{37}	P.No. ₃	S_E	P_{14}	M_3	43.5	37	J_{37}	P.No. ₃	S_E	P_{14}	M_3	50	37	J_{37}	P.No. ₃	S_E	P_{14}	M_3	63.5
50	J_{50}	P.No. ₄	S_B	P_{19}	M_3	44	50	J_{50}	P.No. ₄	S_B	P_{19}	M_3	51	50	J_{50}	P.No. ₄	S_B	P_{19}	M_3	65
24	J_{24}	P.No. ₂	S_B	P_9	M_2	43.5	24	J_{24}	P.No. ₂	S_B	P_9	M_2	50	24	J_{24}	P.No. ₂	S_B	P_9	M_2	67
38	J_{38}	P.No. ₃	S_B	P_{14}	M_4	48.5	25	J_{25}	P.No. ₂	S_D	P_9	M_4	54	38	J_{38}	P.No. ₃	S_B	P_{14}	M_4	70.5
25	J_{25}	P.No. ₂	S_D	P_9	M_4	51	38	J_{38}	P.No. ₃	S_B	P_{14}	M_4	56	39	J_{39}	P.No. ₃	S_B	P_{15}	M_1	77.5
39	J_{39}	P.No. ₃	S_B	P_{15}	M_1	53.5	39	J_{39}	P.No. ₃	S_B	P_{15}	M_1	62	25	J_{25}	P.No. ₂	S_D	P_9	M_4	77.5
40	J_{40}	P.No. ₃	S_B	P_{15}	M_3	59.5	40	J_{40}	P.No. ₃	S_B	P_{15}	M_3	69	40	J_{40}	P.No. ₃	S_B	P_{15}	M_3	85.5

TABLE 4: Percentage of existing and optimized fuzzy numbers representing the efficiency and idle time of machines.

	Existing data		Optimized data	
	Efficiency	Idle time	Efficiency	Idle time
	(Opt, Med, Pes)	(Opt, Med, Pes)	(Opt, Med, Pes)	(Opt, Med, Pes)
Machine (0)	(12%, 10.4%, 12%)	(88%, 89.6%, 88%)	(31.7%, 35.6%, 35.6%)	(68.3%, 64.4%, 64.4%)
Machine (1)	(16.6%, 13.6%, 15.2%)	(83.4%, 86.4%, 84.8%)	(43.7%, 46.6%, 45%)	(56.3%, 53.4%, 55%)
Machine (2)	(16.9%, 13.6%, 15%)	(83.1%, 86.4%, 85%)	(44.4%, 46.6%, 44.4%)	(55.6%, 53.4%, 55.6%)
Machine (3)	(8.4%, 7.6%, 8.3%)	(91.6%, 92.4%, 91.7%)	(22.2%, 26.0%, 24.4%)	(77.8%, 74%, 75.6%)
Machine (4)	(3.3%, 6.4%, 3.9%)	(96.7%, 93.6%, 96.1%)	(8.7%, 21.9%, 11.7%)	(91.35, 78.1%, 88.3%)

TABLE 5: Summary of the final results based on existing and optimized data.

	Existing data (Opt, Med, Pes)	Optimized data (Opt, Med, Pes)
Total scheduling time	(166, 250, 266)	(62.5, 73, 88.5)
Total setup number (No.)	(44, 44, 44)	(36, 36, 39)
Total efficient frontier	(105, 147, 155)	(49.25, 54.5, 63.75)
Total scheduling time with setup time	(254, 338, 354)	(135, 145, 168)
Total operation setup time	(88, 88, 88)	(72, 72, 78)
Total changing setup cost (\$)	(3520, 3520, 3520)	(2880, 2880, 3120)
Total units produced per day	(2.89, 1.92, 1.80)	(7.62, 6.58, 5.33)
Total efficiency (%)	(11.45%, 10.32%, 10.90%)	(30.16%, 35.34%, 32.22%)
Total idle time (%)	(88.55%, 89.68%, 89.10%)	(69.84%, 64.66%, 67.78%)

the make-span (i.e., minimizing scheduling time, travelling time, and machine idle time and maximizing production) and minimizing the setup time (i.e., minimizing the number of machine setup tool changes and minimizing the machine setup cost) simultaneously that occur when switching between different products. The triangular fuzzy numbers are applied for variables such as operation and travelling time. The hypothetical numerical example comprises 50 jobs to produce 20 parts using five machines in order to assemble four products. The fuzzy numbers are categorized as optimistic, medium, and pessimistic fuzzy total scheduling time. The results show that the fuzzy total scheduling time which is 166, 250, and 266 decreases to 62.5, 73, and 88.5 after optimization. Comparison is made between the existing and optimized results representing the efficiency and idle time of each machine. The existing and optimized results of the total scheduling time, total setup number, total efficient frontier, total scheduling time with setup time, total operation setup time, total changing setup cost (\$), and total units produced per day are also compared.

Notations

- i : An index used for parts; $i = 1, 2, \dots, n$
- j : An index used for machines;
 $j = 1, 2, \dots, m$
- n : Number of total parts
- m : Number of total machines

- u : An index used for a setup change for the machine; $u = A, B, \dots, Z$
- p : An index used for an assembly sequence;
 $p = 1, 2, \dots, v$
- v : Maximum number of assembly sequence orders in each product
- h : An index used for a product number;
 $h = 1, 2, \dots, k$
- k : Total product
- q : An index used for gene code number in each chromosome; $q = 1, 2, \dots, e$
- e : Total gene on each chromosome
- z : An index used for a same part's gene code
- f : Number of same part's gene code
- a : An index used for showing an optimistic time
- b : An index used for showing an average time
- c : An index used for showing a pessimistic time
- X_q : The gene code number in the chromosome;
 $q = 1, 2, \dots, e$
- $X_{q'}$: The alleviated gene code number in the chromosome; $q' = q - 1, q - 2, \dots, q - (q - 1)$
- J_q : The number of tasks involved in the total production line; $q = 1, 2, \dots, e$
- P_i : Number of parts; $i = 1, 2, \dots, n$
- M_j : Machine use for manufacturing parts and various products; $j = 1, 2, \dots, m$
- S_u : Setup number based on the jobs, assigned to each machine; $u = A, B, \dots, Z$

P.No._{*h*}: Product number; $h = 1, 2, \dots, k$
 A.S._{*p*}: The order of parts in the assembly;
 $p = 1, 2, \dots, v$
 $\widetilde{OP}_{i,j}$: Fuzzy operation time of parts P_i ,
 $i = 1, 2, \dots, n$, in machine M_j , $i = 1, 2, \dots, m$,
 defined as a triplet (a, b, c)
 $\widetilde{T \cdot t}_{i,j}$: Fuzzy traveling time of parts P_i ,
 $i = 1, 2, \dots, n$, travel between each machine
 M_j , $j = 1, 2, \dots, m$, defined as a triplet
 (a, b, c)
 W_z : A sequence of the same part's gene code with
 f -elements, $W_z, z = 1, 2, \dots, f$
 $\widetilde{S \cdot t}$: Fuzzy start time
 Y_1 : Checking the start time ($\widetilde{S \cdot t}$), based on the
 part's sequencing (P_i) assigned to the
 machine (M_j)
 Y_2 : Checking the start time ($\widetilde{S \cdot t}$) of machine's
 availability (M_j) assigned to produce the
 parts (P_i)
 Y_3 : Checking the start time ($\widetilde{S \cdot t}$) of the parts
 entering the machines based on the sequence
 of the part's assembly ($A.S._p$) for each
 product ($P.No._h$)
 T.S.N: Total setup number
 T.C.T: Total completion time
 PS: Population size
 CR(t): Crossover rate
 Pm(t): Mutation rate
 Max . X_q : Maximum gene code.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge the Ministry of Higher Education for the financial support under High Impact research Grant UM.C/HIR/MOHE/ENG/35 (D000035-16001).

References

- [1] Z. X. Guo, W. K. Wong, S. Y. S. Leung, J. T. Fan, and S. F. Chan, "Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: a case study based on the apparel industry," *Computers & Industrial Engineering*, vol. 50, no. 3, pp. 202–219, 2006.
- [2] S. O. Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *Journal of Intelligent Manufacturing*, vol. 19, no. 1, pp. 49–69, 2008.
- [3] N. Kriengkorkot and N. Pianthong, "The assembly line balancing problem: review articles," *KKU Engineering Journal*, vol. 34, no. 2, pp. 133–140, 2012.
- [4] I. Baybars, "A survey of exact algorithms for the simple assembly line balancing problem," *Management Science*, vol. 32, no. 8, pp. 909–932, 1986.
- [5] Z. Taha, F. Tahriri, and A. Zuhdi, "Job sequencing and layout optimization in virtual production line," *Journal of Quality*, vol. 18, no. 4, pp. 351–374, 2011.
- [6] P. Brucker and T. Kampmeyer, "Cyclic job shop scheduling problems with blocking," *Annals of Operations Research*, vol. 159, no. 1, pp. 161–181, 2008.
- [7] G. Heike, M. Ramulu, E. Sorenson, P. Shanahan, and K. Moinzadeh, "Mixed model assembly alternatives for low-volume manufacturing: the case of the aerospace industry," *International Journal of Production Economics*, vol. 72, no. 2, pp. 103–120, 2001.
- [8] A. R. Rahimi-Vahed, M. Rabbani, R. Tavakkoli-Moghaddam, S. A. Torabi, and F. Jolai, "A multi-objective scatter search for a mixed-model assembly line sequencing problem," *Advanced Engineering Informatics*, vol. 21, no. 1, pp. 85–99, 2007.
- [9] B. Javadi, A. Rahimi-Vahed, M. Rabbani, and M. Dangchi, "Solving a multi-objective mixed-model assembly line sequencing problem by a fuzzy goal programming approach," *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 975–982, 2008.
- [10] N. Boysen, M. Flidner, and A. Scholl, "Sequencing mixed-model assembly lines: survey, classification and model critique," *European Journal of Operational Research*, vol. 192, no. 2, pp. 349–373, 2009.
- [11] Y. Monden, *Toyota Production System: Practical Approach to Production Management*, Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, Ga, USA, 1997.
- [12] J. F. Bard, E. Dar-El, and A. Shtub, "Analytic framework for sequencing mixed model assembly lines," *The International Journal of Production Research*, vol. 30, no. 1, pp. 35–48, 1992.
- [13] E. M. Dar-El, "Mixed-model assembly line sequencing problems," *Omega*, vol. 6, no. 4, pp. 313–323, 1978.
- [14] E. M. Dar-El and R. F. Cothier, "Assembly line sequencing for model mix," *The International Journal of Production Research*, vol. 13, no. 5, pp. 463–477, 1975.
- [15] E. M. Dar-El and A. Nadivi, "A mixed-model sequencing application," *The International Journal of Production Research*, vol. 19, no. 1, pp. 69–84, 1981.
- [16] C. Merengo, F. Nava, and A. Pozzetti, "Balancing and sequencing manual mixed-model assembly lines," *The International Journal of Production Research*, vol. 37, no. 12, pp. 2835–2860, 1999.
- [17] X. Zhu, S. J. Hu, Y. Koren, S. P. Marin, and N. Huang, "Sequence planning to minimize complexity in mixed-model assembly lines," in *Proceedings of the IEEE International Symposium on Assembly and Manufacturing (ISAM '07)*, pp. 251–258, July 2007.
- [18] B. Rekiek, P. De Lit, and A. Delchambre, "Designing mixed-product assembly lines," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 3, pp. 268–280, 2000.
- [19] P. De Lit, A. Delchambre, and J.-M. Henrioud, "An integrated approach for product family and assembly system design," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 324–334, 2003.
- [20] J. Bukchin, E. M. Dar-El, and J. Rubinovitz, "Mixed model assembly line design in a make-to-order environment," *Computers & Industrial Engineering*, vol. 41, no. 4, pp. 405–421, 2001.
- [21] C. Moon, M. Lee, Y. Seo, and Y. H. Lee, "Integrated machine tool selection and operation sequencing with capacity and precedence constraints using genetic algorithm," *Computers & Industrial Engineering*, vol. 43, no. 3, pp. 605–621, 2002.

- [22] A. Khan and A. J. Day, "A knowledge based design methodology for manufacturing assembly lines," *Computers & Industrial Engineering*, vol. 41, no. 4, pp. 441–467, 2001.
- [23] A. R. Mendes, A. L. Ramos, A. S. Simaria, and P. M. Vilarinho, "Combining heuristic procedures and simulation models for balancing a PC camera assembly line," *Computers & Industrial Engineering*, vol. 49, no. 3, pp. 413–431, 2005.
- [24] A. Joly and Y. Frein, "Heuristics for an industrial car sequencing problem considering paint and assembly shop objectives," *Computers & Industrial Engineering*, vol. 55, no. 2, pp. 295–310, 2008.
- [25] S. Ghosh and R. J. Gagnon, "Comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems," *The International Journal of Production Research*, vol. 27, no. 4, pp. 637–670, 1989.
- [26] A. Norozi, M. K. Ariffin, and N. Ismail, "Application of intelligence based genetic algorithm for job sequencing problem on parallel mixed-model assembly line," *American Journal of Engineering and Applied Sciences*, vol. 3, no. 1, pp. 831–840, 2010.
- [27] W. K. Wong, C. K. Chan, and W. H. Ip, "Optimization of spreading and cutting sequencing model in garment manufacturing," *Computers in Industry*, vol. 43, no. 1, pp. 1–10, 2000.
- [28] S. G. Ponnambalam, P. Aravindan, and M. Subba Rao, "Genetic algorithms for sequencing problems in mixed model assembly lines," *Computers & Industrial Engineering*, vol. 45, no. 4, pp. 669–690, 2003.
- [29] S.-M. Im and J.-J. Lee, "Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms," *Artificial Life and Robotics*, vol. 13, no. 1, pp. 129–133, 2008.
- [30] R. Subbu, A. C. Sanderson, and P. P. Bonissone, "Fuzzy logic controlled genetic algorithms versus tuned genetic algorithms: an agile manufacturing application," in *Proceedings of the IEEE International Symposium on Intelligent Control (ISIC '98)*, pp. 434–440, September 1998.
- [31] M. Gen, R. Chen, and L. Lin, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*, Springer, New York, NY, USA, 2008.
- [32] B. S. P. Reddy and C. S. P. Rao, "A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS," *International Journal of Advanced Manufacturing Technology*, vol. 31, no. 5-6, pp. 602–613, 2006.
- [33] P. T. Zacharia and A. C. Nearchou, "Multi-objective fuzzy assembly line balancing using genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 23, no. 3, pp. 615–627, 2012.
- [34] P. R. McMullen, "An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics," *Computers & Industrial Engineering*, vol. 41, no. 3, pp. 335–353, 2001.
- [35] P. R. McMullen, "An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 309–317, 2001.
- [36] P. R. McMullen, "A Kohonen self-organizing map approach to addressing a multiple objective, mixed-model JIT sequencing problem," *International Journal of Production Economics*, vol. 72, no. 1, pp. 59–71, 2001.
- [37] P. R. McMullen and P. Tarasewich, "A beam search heuristic method for mixed-model scheduling with setups," *International Journal of Production Economics*, vol. 96, no. 2, pp. 273–283, 2005.
- [38] J. F. Bard, A. Shtub, and S. B. Joshi, "Sequencing mixed-model assembly lines to level parts usage and minimize line length," *The International Journal of Production Research*, vol. 32, no. 10, pp. 2431–2454, 1994.
- [39] P. R. McMullen, "JIT sequencing for mixed-model assembly lines with setups using Tabu Search," *Production Planning & Control*, vol. 9, no. 5, pp. 504–510, 1998.
- [40] P. R. McMullen and G. V. Frazier, "A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line," *IIE Transactions*, vol. 32, no. 8, pp. 679–686, 2000.
- [41] V. Giard and J. Jeunet, "Optimal sequencing of mixed models with sequence-dependent setups and utility workers on an assembly line," *International Journal of Production Economics*, vol. 123, no. 2, pp. 290–300, 2010.
- [42] D. J. Fonseca, C. L. Guest, M. Elam, and C. L. Karr, "A fuzzy logic approach to assembly line balancing," *Mathware & Soft Computing*, vol. 12, no. 1, pp. 57–74, 2005.
- [43] M. Gen, Y. Tsujimura, and Y. Li, "Fuzzy assembly line balancing using genetic algorithms," *Computers & Industrial Engineering*, vol. 31, no. 3-4, pp. 631–634, 1996.
- [44] Y. Tsujimura, M. Gen, and E. Kubota, "Solving fuzzy assembly-line balancing problem with genetic algorithms," *Computers & Industrial Engineering*, vol. 29, no. 1-4, pp. 543–547, 1995.
- [45] W. Cheung and H. Zhou, "Using genetic algorithms and heuristics for job shop scheduling with sequence-dependent setup times," *Annals of Operations Research*, vol. 107, no. 1–4, pp. 65–81, 2001.
- [46] G. Mosheiov, A. Sarig, and J. Sidney, "The Browne-Yechiali single-machine sequence is optimal for flow-shops," *Computers and Operations Research*, vol. 37, no. 11, pp. 1965–1967, 2010.
- [47] J. Silberholz and B. Golden, "Comparison of metaheuristics," in *Handbook of Metaheuristics*, Springer, New York, NY, USA, 2010.