IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 2, 2019, accepted January 30, 2019, date of publication February 5, 2019, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897730

Optimizing the Reliability and Efficiency for an Assembly Line That Considers Uncertain Task Time Attributes

YUCHEN LI[1], YAO FU[1], XIAOWEN TANG[1], AND XIAOFENG HU[2]

[1]Research Base of Beijing Modern Manufacturing Development, School of Economics and Management, Beijing University of Technology, Beijing 100124, China

[2]School of Mechanical Engineering, Institute of Intelligent Manufacturing and Information Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Xiaowen Tang (txw@bjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51475303 and Grant 71772009, in part by the Base Project of the Beijing Social Science Foundation under Grant 17JDGLB019, and in part by the International Research Cooperation Seed Fund of the Beijing University of Technology under Grant 2018B24.

ABSTRACT An assembly line is an industrial arrangement of machines, equipment, and operators for continuous flow of workpieces in mass-production operations. In an assembly line balancing problem, tasks are allocated to workstations according to their processing times and precedence relationships amongst tasks. Nowadays, some research investigated the reliability of assembly production by taking account of task time uncertainties. Our research utilizes uncertainty theory to model task time uncertainties and introduces the belief reliability measure to the assembly line production for the first time. We proposed a multi-objective optimization model that aimed at maximizing the belief reliability and minimizing the cycle time. The problem is solved using a newly developed restart neighborhood search method. The numerical experiments are conducted to verify its efficiency. The methodology proposed in this paper is applicable to any industry (including the automotive industry) when the historical data on task processing times are very scarce.

INDEX TERMS Assembly line balancing, belief reliability, neighborhood search, uncertain task times.

I. INTRODUCTION

An assembly line (AL) is an important manufacturing tool that utilizes machines to move material or parts from one place to another. Assembly lines are widely employed to produce various types of products, including automobiles, electronic products, and jewelry. The main components of a standard assembly line are a conveyor belt, workers, workstations, interchangeable parts and tasks. The layout for a standard assembly line is shown in Figure 1. The workstations are located on one side of the line. Workers perform the tasks at workstations simultaneously. Task time is the time that a task takes to execute by a worker. Cycle time is the interval of a product being finished or offered to a customer. The sum of task times of a station cannot be greater than the cycle time, otherwise the assembly line is not paced.

The assembly line balancing problem (ALBP) is a classical combinatorial optimization problem that falls into the NP-hard category. ALBP is to optimally allocate tasks to workstations to achieve certain objectives with respect to some constraints. Bryton [5] first proposed the ALBP and the first scientific research was done by Salveson [26]. For a comprehensive review for ALBP, readers can refer to [1] and [14].

The associate editor coordinating the review of this manuscript and approving it for publication was Jiajie Fan.

A. ASSEMBLY LINE BALANCING WITH UNCERTAIN TASK TIMES

Nowadays, ALBP with uncertain task times (ALBP-UT) receives a lot of attention in academia. Task time uncertainty may result from instability of the operator's work rates, the varied skills and motivations of workers, and the failure sensitivity of complex processes' uncertainty [3]. Suresh and Sahu [29] first introduced the task time uncertainty into ALBP and proposed a simulated annealing algorithm. Baykasoğlu and Özbakir [2] optimized a U-shaped ALBP-UT by genetic algorithm. Cakir *et al.* [6] proposed a hybrid simulated annealing algorithm to solve the multi-objective ALBP-UT. More recently, Delice *et al.* [7] developed a genetic algorithm to solve the two-sided U-shaped ALBP-UT. Li [12] solved the type-II ALBP-UT by a branch-and-bound

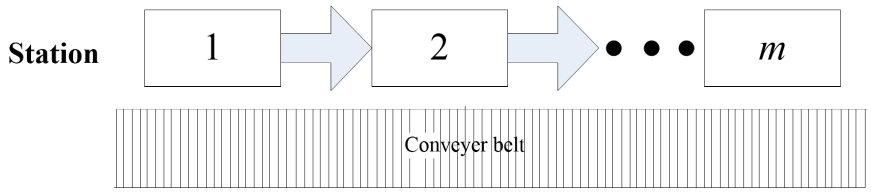


FIGURE 1. The basic structure of a standard assembly line.

method. Tang *et al.* [30] balanced the two-sided ALBP-UT by a teaching-learning-based algorithm. Zhang *et al.* [36] minimized the cycle time and processing cost for ALBP-UT by a hybrid evolutionary algorithm.

Most research focuses on optimizing the line efficiency, such as minimizing cycle time or the number of workstations for ALBP-UT. There are a few research investigating the production delay due to task time uncertainty. Actually, task time uncertainty may cause the line unbalanced (i.e.,some workstations may get overloaded because the increase of the task times), which is sometimes a more important issue than line efficiency optimization in practice. As such, maximizing the reliability of the assembly line becomes another objective for ALBP-UT. Reliability of the line is defined as the *completion probability* that the product can be successfully assembled within the predetermined cycle time. However, the line efficiency and reliability are conflicting goals. For the same task assignment scheme, the line efficiency decreases in cycle time whereas the reliability increases in cycle time (as the cycle time is larger, the ability to handle the task time variation is stronger). Saif *et al.* [24] minimized cycle time and maximized the system reliability for ALBP-UT by developing an artificial bee colony algorithm. Later, Saif *et al.* [25] further investigated the average reliability of the workstations and system reliability.

In the literature, the task times were modeled as random variables, and reliability was characterized as a probability. As is well known,in order to obtain the probability distributions for task times, enough historical data and information are required, i.e.,we need sufficient data to derive the probability distribution. However, sufficient data may be unavailable under low-volume production for mass-customized items and one-off production for large-scale projects [4]. For instance, one may not have any prior data before the production process when manufacturing a novel aircraft for a special purpose (as there is no time to run the pilot production and the cost to do that would be extremely high). To this end, we should invite some domain experts to obtain the belief degrees for these task times. Li *et al.* [13] utilized a new mathematical framework: "uncertainty theory" to handle the uncertain task times as there is a lack of past data. They proposed a branch and bound remember algorithm to minimize the number of stations for ALBP-UT. In this paper, we utilized the uncertainty theory to model uncertain task time attribute.

B. UNCERTAINTY THEORY

Uncertainty theory and probability theory can both be utilized to model indeterminate event. However, the scopes of applications of the two theories are totally different. The probability theory is useful when the distribution of past data or information is very close to the objective *frequency*. For example, we flip a coin 100 times, the percentages that the head will occur is approximately 50%. In this case, we can estimate the likelihood of the next head by probability theory as the coin is flipped again. However, past information may not be available in some situations. In those situations, we cannot use the probability theory since the frequencies cannot be obtained. For example, when assembling a new car, we cannot estimate the total operation times by analyzing the past samples. Instead, we can only predict it from our subjective mind (*belief*). For each possible operation times, there is a belief degree in our mind which describes the likelihood the specific times will be the real one. The belief degree depends heavily on the personal knowledge concerning the event. Therefore, different people have different belief degrees for the same event. Uncertainty theory is a branch of axiomatic mathematics that explores human uncertainty, which can be used to model the belief degree.

Uncertainty theory was invented by Liu [16] to model the belief degrees of experts. It has become a new branch in mathematics for gauging the indeterminate phenomena. Later, Liu [18] developed an uncertain programming model pertaining to the uncertain variables. Since then, uncertainty theory has been widely employed to model the uncertain inputs of various optimization problems, such as facility location allocation [31], project scheduling [10], decentralized planning [21], optimal control [32], vehicle routing [23], machine scheduling [11] and minimum spanning trees [28]. The basic properties and theorems of the uncertainty theory are introduced in the Appendix.

C. DISCREPANCIES

Our work distinguishes from the related work in the following aspects. The proposed problem is novel and has some merits in practice. The problem occurs when a new product is going to be assembled or a new production process is going to be launched, but there is very little information regarding the task times. The concept of the belief reliability is first introduced into the ALBP. Unlike the traditional reliability measure which is based on the probability theory, the belief reliability is derived from uncertainty theory. Therefore, the mathematical formulation is different from other research. Further, a new algorithm is devised by utilizing the specific characteristics of the problem.

The rest of the paper is organized as follows: Section 2 introduces the belief reliability of ALBP-UT, and Section 3 provides the mathematical model for the proposed problem. In Section 4, a restart neighborhood search method is developed to solve the problem. Numerical test for the algorithm is conducted, and the results are reported in Section 5. Finally, the paper is concluded in Section 6. The notations used throughout the paper are presented in Table 1. We put a "~" above the task-time-related parameters, which are uncertain variables.

TABLE 1. Notations.

Variable names	Descriptions
c	The set of cycle times under one task assignment scheme, $c = \{c_1, c_2, \cdots\}$
m	The number of workstations
n	The number of tasks
$\tilde{t}_i$	The task processing time for task $i, i = 1, 2, \cdots, n$
x_{ij}	$x_{ij} = 1$ if task i is assigned to station j; otherwise, $x_{ij} = 0$
Br	The belief reliability of the assembly line
Br_j	The belief reliability of station j
$\widetilde{Sl}_j$	The load of station j
UB, LB	The upper bound and lower bound for the belief reliability of the assembly line
Ta	Task assignment
Q_j	The task assignment at station j
P	The direct precedence relationships matrix, where $P(v, b) = 1$ if and only if task v is a direct predecessor of task b
$\Phi_i(x)$	The uncertainty distribution for the processing time of task i, which is normal distribution $\mathcal{N}(e_i, \sigma_i)$
$\Psi_j(x)$	The uncertainty distribution for the load of station j

II. BELIEF RELIABILITY OF THE ASSEMBLY LINE

Reliability measure is an important characteristic of products or processes, defined as the ability that a component or system can function normally for a given period of time [8]. It is always used to evaluate the product's life cycle. Traditional reliability metrics are defined on the basis of probability theory, which relies on historical data [22]. The reliability of the assembly line can be considered as the probability a product can be finished within the cycle time. Without task time variation, the reliability is always equal to 1. When task times are uncertain, the reliability is less than 1 because it is possible that the sum of task times of a station is greater than the cycle time, which is a predetermined value. In this case, the station is overloaded and the production process is delayed.

To evaluate the reliability of the production process, it always requires past data and information to obtain the probability distribution. However, in some situation, data is unavailable (c.f. section I-A) and probability theory is not applicable to obtain the reliability. To address that, Zeng *et al.* [34] provided the belief reliability metrics based on the uncertainty theory. The definition is as follows.

Definition 1 (Belief Reliability): Assume a system contains uncertain variables $\xi_1, \xi_2, \cdots, \xi_n$, and there is a function R such that the system is working if and only if $R(\xi_1, \xi_2, \cdots, \xi_n) \geq 0$. Then the belief reliability index is

$$R_B = \mathcal{M}\{R(\xi_1, \xi_2, \cdots, \xi_n) \geq 0\}, \tag{1}$$

where $\mathcal{M}$ is the uncertain measure (c.f.Appendix). Then, belief reliability is often utilized as a non-probabilistic reliability metrics in manufacturing [9]. Zeng *et al.* [33] integrated the belief reliability into the existing model-based reliability framework. Zhang *et al.* [35] utilized the belief reliability to describe the states in the uncertain random system.

In this paper, we adopt the concept of the belief reliability and apply it in ALBP-UT. Based on Definition 1, we proposed the belief reliability metrics for ALBP-UT.

Definition 2 (Belief Reliability of a Station): The belief reliability of a station Br_j is defined as the uncertain measure for a situation that station jth's load is not greater than the cycle time. Mathematically,

$$Br_j = \mathcal{M}\{c - \widetilde{Sl}_j \geq 0\}, \tag{2}$$

where $\widetilde{Sl}_j = \sum_{i \in Q_j} \tilde{t}_i$.

Definition 3 (Belief Reliability of an Assembly Line): The belief reliability of an assembly line Br is defined as the uncertain measure for a situation that **every** station load is not greater than the cycle time. Mathematically,

$$Br = \mathcal{M}\{c - \widetilde{Sl}_1 \geq 0, \cdots, c - \widetilde{Sl}_m \geq 0\}, \tag{3}$$

According to the Theorem 3 (c.f.Appendix), the station load $\widetilde{Sl}_j$ follows a normal uncertainty distribution $\Psi_j(c)$: $\mathcal{N}(\sum_{i \in Q_j} e_i, \sum_{i \in Q_j} \sigma_i)$ since tasks are independent of each other. Therefore,

$$Br_j = \mathcal{M}\{\widetilde{Sl}_j \leq c\} = \Psi_j(c)$$

Because the station loads are independent of each other, Axiom 4 (c.f.Appendix) can be used to calculate the belief reliability of the line as

$$Br = \text{Min}\{Br_1, Br_2, \cdots, Br_m\} \tag{4}$$

In the next section, we incorporate the belief reliability in the model of ALBP-UT.

III. PROBLEM FORMULATION

First, we present the following necessary assumptions and conditions for the problem formulation.

- The travel times and set-up times are ignored.
- A task can be assigned to only one workstation.
- The task assignment must conform to the precedence relationships, which are known and deterministic.
- The belief reliability must lie in the range $[LB, UB]$.
- The task times are uncertain variables, and their uncertainty distributions are known and given.
- The goal is to obtain the Pareto-optimal set which includes the cycle time and belief reliability of the assembly line.

We delineate the mathematical model of the proposed problem. The task time distributions, precedence relationships, the number of workstations and tasks are given as input into our model. The objective function of our problem is to minimize cycle time c and maximize the belief reliability of the assembly line Br (or equivalently, minimize $1 - Br$),

$$\begin{aligned} &Min \quad c \\ &Min \quad 1 - Br \end{aligned}$$

The decision variables are x_{ij}. x_{ij} determines the task assignment at workstation j. Now,we establish the constraints. The first constraint is task indivisible constraint, as a task can only be assigned to one station.

$$\sum_{j=1}^{m} x_{ij} = 1, \quad \forall i = 1, 2, \cdots, n. \tag{5}$$

The second constraint ensures the precedence relationship when the two tasks are assigned to different mated stations:

$$\sum_{j=1}^{m} jx_{vj} \leq \sum_{j=1}^{m} jx_{bj}, \text{ if } P(v, b) = 1, \quad \forall v, \ b = 1, 2, \cdots, n. \tag{6}$$

The third constraint guarantees that the belief reliability of the line Br is not greater than UB and not less than LB. Therefore, the third constraint is

$$LB \leq Br \leq UB \tag{7}$$

where $Br = \min_{j=1,2,\cdots,m} \Psi_j(c)$. The Pareto-optimal set of the model can provide various choices of cycle time and belief reliability combinations. Production managers can select the appropriate pair according to the specific practical requirement. In the next section, we propose an algorithm to find the Pareto-optimal set.

IV. SOLUTION PROCEDURE

In this section, we develop a restart neighborhood search (RNS) algorithm to solve the proposed problem. The general framework is depicted in Figure 2. The proposed algorithm features a restart mechanism to escape the local optimum. The algorithm begins with generating an initial random feasible solution using the encoding and decoding procedures. Then, the steps in the main modules are iterated until the termination criteria are met. The main modules are neighborhood search, Pareto-optimal set update, and restart mechanism. The details of the RNS are presented in the following sections.

A. ENCODING AND DECODING

The solution contains the information of cycle time c and task assignment Ta in each workstation, $S = \{c, Ta\}$. The length of Ta is equal to the number of tasks. In Ta, if the ith element is j, the ith task should be assigned to station j. The Pareto-optimal set consists of pairs of cycle time and belief reliability, $Ps = \{c, Br\}$

Step 0: Initialization: generate a random array Ta^0. If every value from 1 to m appears at least once, go to step 1; otherwise, repeat this step.

Step 1: Set $k = 1$, $c^0 = \sum_{i=1}^{n} E[\tilde{t}_i]/m$.

Step 2: Extract the elements whose corresponding values in Ta^0 are equal to k and form a vector A. If A is empty, set $k = k + 1$ and reimplement step 2;otherwise, go to step 3.

Step 3: Check whether the predecessors of the first task in A has been already assigned. If yes, go to step 5; otherwise, reset the values of Ta^0 that refer to the unassigned predecessors to k and go to step 4.

Step 4: Add the indices corresponding to the reset values of Ta^0 to the beginning of A and go back to step 3.

Step 5: Assign the task to station k and remove the task from A. If A is empty, set $k = k+1$ and go to step 6;otherwise, go back to step 3.

Step 6: Check whether all tasks have been assigned. If yes, go to step 7;otherwise, go back to step 2.

Step 7: Vary c^0 so as to find several belief reliabilities (the belief reliability can be calculated by Eq.(4)) that lie in the range $[LB, UB]$, then $c^0 = \{c_1^0, c_2^0, \cdots\}$ and $Br^0 = \{Br_1^0, Br_2^0, \cdots\}$.

With the above decoding scheme, the initial solution set $S_0 = \{c^0, Ta^0\}$ and Pareto-optimal set $Ps_0 = \{c^0, Br^0\}$ can be constructed. The advantage of the decoding scheme is that several solutions can be generated in one iteration.

B. NEIGHBORHOOD GENERATION

Suppose the $iter$th iteration is finished, we employ the following procedures to find its neighborhood solution,S_{iter+1}.

Step 1: Locate the station with the smallest belief reliability (station j), and go to step 2.

Step 2: Randomly choose a task i from station j and go to step 3.

Step 3: Find the set B that consists of all possible targeted stations to which task i can be assigned while the precedence relationships remain satisfied. Go to step 4.

Step 4: If there is no possible targeted station for task i, go back to step 2;otherwise, go to step 5.

Step 5: Randomly choose a value r from B and change the ith value of Ta^{iter} to r, and a new task assignment Ta^{iter+1} is found.

Step 6: Vary the cycle time so as to find several belief reliabilities that lie in the range $[LB, UB]$, then $c^{iter} = \{c_1^{iter}, c_2^{iter}, \cdots\}$ and $Br^{iter} = \{Br_1^{iter}, Br_2^{iter}, \cdots\}$.

Step 7: Update $S_{iter+1} = \{c^{iter+1}, Ta^{iter+1}\}$

The advantage of the generation procedure is that optimum may be improved since stations are likely to be balanced in terms of the belief reliability. Further, the feasibility of the new solution is guaranteed.

C. PARETO-OPTIMAL SET UPDATE

After a new set of solutions is generated, we check each solution to determine whether it is a non-dominated solution. If the solution is not dominated by any of the solutions in the Pareto-optimal set, it will be added into the current Pareto-optimal set, and the original solution will be eliminated from the set if it is dominated by the new solution. This procedure

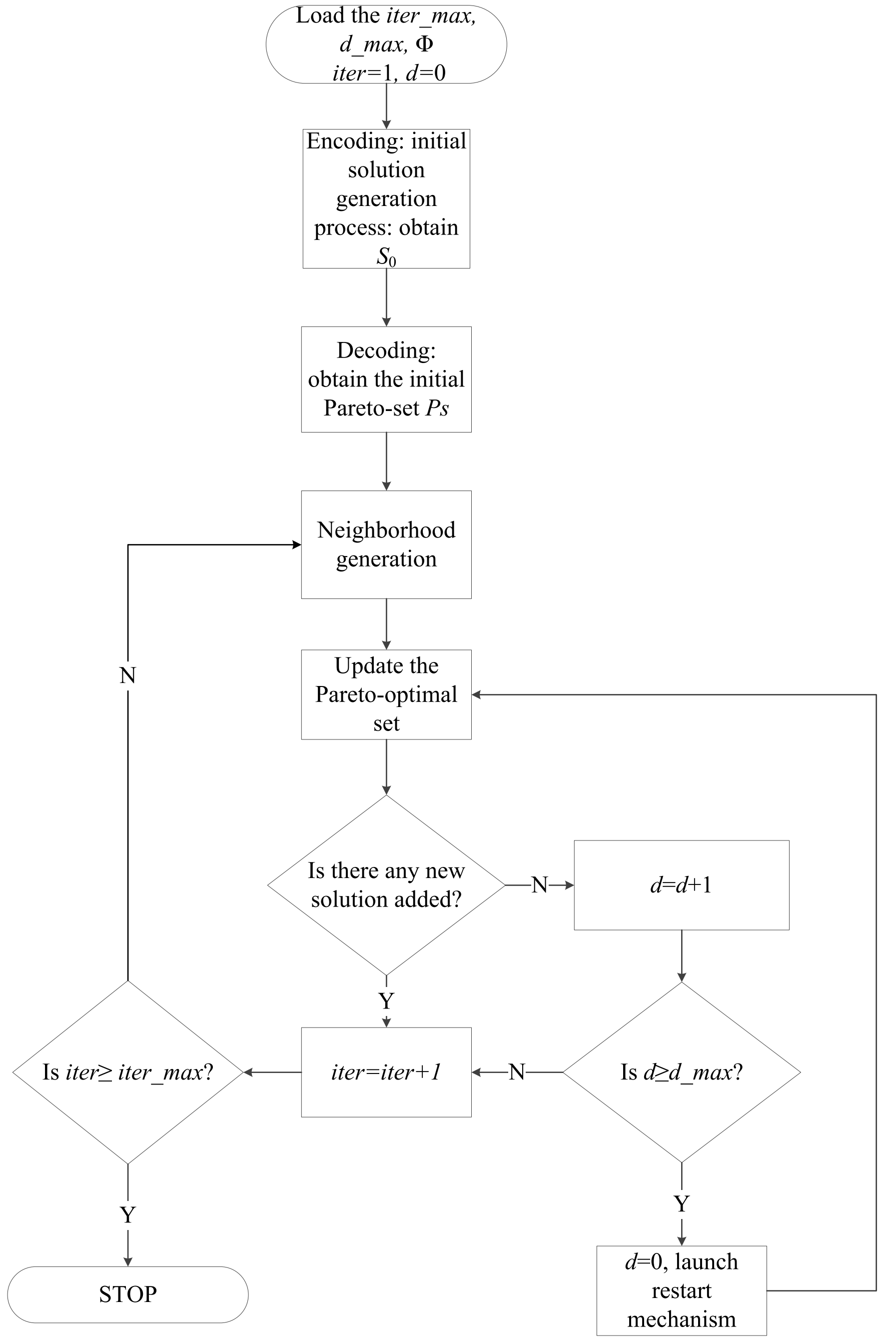


FIGURE 2. The general framework of RNS.

can be executed by locating the cycle time of new objective pair $\{c, Br\}$ in the current Pareto-optimal set Ps. Let c' and c'' are the two adjacent cycle time of c in Ps, $c' \leq c \leq c''$. Br' and Br'' are the corresponding belief reliabilities of c' and c'', respectively. Now we discuss the following scenarios.

Scenario 1: $Br \leq Br'$. This scenario indicates that sacrificing the line efficiency does not lead to reliability improvement. Therefore, $\{c, Br\}$ is dominated by $\{c', Br'\}$, and Ps remains unchanged.

Scenario 2: $Br' < Br < Br''$. This scenario indicates that sacrificing the line efficiency from c' to c leads to reliability improvement, and sacrificing the reliability from Br'' to Br lead to efficiency improvement. Therefore, $\{c, Br\}$ is a non-dominated solution and should be added to Ps.

Scenario 3: $Br'' < Br$. This scenario indicates that improving the line efficiency from c'' to c leads to reliability improvement. Therefore, $\{c, Br\}$ is a non-dominated solution.

TABLE 2. Restart mechanism.

$\%Cd(i)$ is the crowding distance for the ith solution in Pareto-optimal set Ps
$Sc(i)$ indicates the number of times the ith solution being selected
$Obj^{(1)}(i)$ and $Obj^{(2)}(i)$ are the normalized cycle time and belief reliability corresponding to the ith solution
$Size = \|Cd\|$
Set $Cd(i) = 0$ for all $i = 1, 2, \cdots, Size$
For objectives $l = 1, 2$
Sort Ps using the objective l
f_l^{max}=maximum value for objective l;f_l^{min}=minimum value for objective l
For $i = 2, \cdots, Size - 1$
$Cd(i) = Cd(i) + (Obj^{(l)}(i+1) - Obj^{(l)}(i-1))/(f_l^{max} - f_l^{min})$
Endfor
$Cd(1) = Cd(1) + 0.5; Cd(Size) = Cd(Size) + 0.5$
EndFor
For $i = 1, 2, \cdots, Size$
$Cd(i) = Cd(i)/Sc(i)$
Endfor
Select the solution i with the largest distance, and $Sc(i) = Sc(i) + 1$

However, $\{c'', Br''\}$ is a dominated solution and should be eliminated from Ps.

D. RESTART MECHANISM

As is well known, neighborhood search methods may lead to local optima. To address this issue, we develop a restart mechanism proposed to escape the local optima. It is a diversification strategy that will enlarge the capacity of the Pareto-optimal set. If a non-dominated solution is not found after a number of d_{max} moves, a solution from the Pareto-set is selected. Instead of choosing the solution randomly, the restart mechanism will choose the solution with the largest crowding distance. The crowding distance describes the degree of isolation of a solution at the Pareto front. If the most-isolated solution is chosen, the gaps in the objective solution space can be reduced and the spread of the Pareto-optimal set can be enhanced. To prevent a solution from being selected several times, we utilize a counter (Sc), which is set to 1 at the beginning, to reduce the crowding distance. The restart mechanism is presented as follows.

V. NUMERIC EXPERIMENTATION

A. EXPERIMENTAL SETTINGS

In this section, the performance of RNS is evaluated. Three benchmark problems, Arcus111 (111 tasks), Bartholdi148 (148 tasks) and Scholl297 (297 tasks), are solved [27]. All the algorithms are programmed in Matlab. The experiments are conducted on a computer equipped with Intel(R) I7 with CPU 2.8 GHz.

The uncertainties are introduced to the dataset as follows: each task time follows a normal uncertainty distribution $\mathcal{N}(\mu, \sigma)$; μ is the deterministic task time in the original dataset; σ is randomly generated between 0 and $(\frac{\mu}{2})^2$.

B. EVALUATION METHODS

To evaluate the algorithms, three metrics are applied:the ratio of non-dominated solutions(Rp), the convergence of the non-dominated solution (Cp) and the spread metric (Sp). These metrics are widely applied to multi-objective assembly line balancing problem to evaluate the performance of algorithms.

Suppose we have several Pareto-optimal sets ($Ps_1, Ps_2, \cdots, Ps_w$) generated by different algorithms. The higher Rp is, the better the Pareto-optimal set is. Let $P = Ps_1 \cup Ps_2 \cup \cdots$, then,

$$Rp(Ps) = \frac{|Ps - \{x \in Ps | \forall y \in P : y \prec x\}|}{|Ps|}, \tag{8}$$

where $y \prec x$ shows the solution x is dominated by solution y. $Rp(Ps)$ measures that the solution in Ps is not dominated by other solutions in P.

The convergence of the Pareto-optimal set ($Cp(Ps)$) describes the difference between one Pareto-optimal set and the approximated true Pareto-optimal set Tp. In the experiment, Tp is approximated by running the algorithms twenty times. Let dt_e refer to the minimum distance between solution e and the other solutions in the Pareto-optimal set. f_l^{max} = maximum value for objective l and f_l^{min} = minimum value for objective l. A low value of $Cp(Ps)$ indicates a good convergence to the true Pareto-optimal set. $Cp(Ps)$ can be calculated as follows,

$$dt_e = \min_{u=1,\cdots,|Tp|} \sqrt{\sum_{l=1}^{2} \left(\frac{f_l(e) - f_l(u)}{f_l^{max} - f_l^{min}} \right)^2}, \tag{9}$$

$$Cp(Ps) = \sum_{e=1}^{|Ps|} \frac{dt_e}{|Ps|}, \tag{10}$$

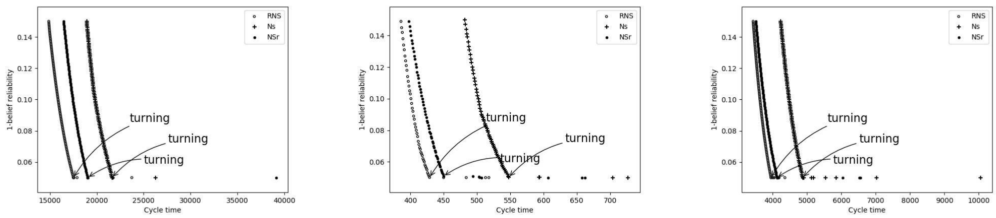
The spread metric captures the distributions of the solutions at the Pareto front. sd_1 and sd_2 are the Euclidean distance between the extreme solutions in Ps and the boundary solutions in Tp. sd_e ($e = 1, 2, \cdots, |Ps| - 1$) is the Euclidean distance between two consecutive solutions in Ps. $\bar{sd}$ is the average distance. The smaller value of $Sp(Ps)$ means a better spread of the Pareto-optimal set. $Sp(Ps)$ can be calculated as follows.

$$sd_e = \sqrt{\sum_{l=1}^{2} \left(\frac{f_l(e) - f_l(e+1)}{f_l^{max} - f_l^{min}} \right)^2}, \tag{11}$$

$$Sp(Ps) = \frac{sd_1 + sd_2 + \sum_{e=1}^{|Ps|-1} |sd_e - \bar{sd}|}{sd_1 + sd_2 + (|Ps| - 1)\bar{sd}}, \tag{12}$$

For the parameters of RNS, d_{max} is to be determined by users. d_{max} indicates the maximal number of moves for the neighborhood generation process to update the Pareto-optimal set without invoking the restart mechanism. we conduct some experiments to calibrate d_{max}. Three levels of d_{max} are tested:10, 30, 50. Therefore, there are three algorithm configurations. Each configuration is tested on Scholl297 5 times. m is set to 12. In each run, the algorithm is terminated when the CPU runtime reaches $t_{limit} = n \times n \times p$ milliseconds. In this experiment, $n = 297$ and $p = 10$. We select $Cp(Ps)$ as the response variable. ANOVA is employed to analyze the results. It shows that there is no significant difference among the results. Therefore, we arbitrarily choose $d_{max} = 30$.

To test the performance of RNS, two other related algorithms are utilized for comparison, namely, NS and NSr. NS



(a) Pareto fronts of Arcus111 with $m = 16$ (b) Pareto fronts of Bartholdi148 with $m = 14$ (c) Pareto fronts of SCHOLL297 with $m = 297$

FIGURE 3. Pareto fronts of different examples generated by the three algorithms. (a) Pareto fronts of Arcus111 with $m = 16$. (b) Pareto fronts of Bartholdi148 with $m = 14$. (c) Pareto fronts of SCHOLL297 with $m = 297$.

is the pure neighborhood search procedure (c.f.section 4.2) without the restart mechanism. NSr includes the restart algorithm, but the algorithm restarts with a new random feasible solution. The performance of the three algorithms is tested on 18 problem instances which include Arcus111, Bartholdi148 and Scholl297 with different number of stations. Each algorithm is tested on each instance 5 times. The termination criterion is the time limit of the CPU runtime, $t_{limit} = n \times n \times 10$ milliseconds. There are a total of $18 \times 5 \times 3 = 270$ samples. The average results for the three evaluation metrics are shown in Table 3. It can be observed that RNS outperforms the other two algorithms for all cases under the ratio of non-dominated solutions(*Rp*). As a result, most points on the true Pareto front belong to the output of RNS. As for the convergence of the non-dominated solution (*Cp*), three algorithms do not distinguish each other very much although RNS achieves the best results for all cases. That means solutions generated by the three algorithms, if not on the the true Pareto front, are very close regarding their distances from the true Pareto front. As for the spread metric (*Sp*), RNS outperforms the other two algorithms in 12 out of 18 cases. Again, the differences of *Sp* among algorithms are very small. To further examine two criteria (*Cp* and *Sp*), non-parameter Wilcoxon signed-rank test is conducted. For each criterion, the test is conducted twice (RNS versus NS, RNS versus NSr). For each sample, the best result receives a score 1, the worst result receives a score 2. The results show that the differences of two criteria, although small, are significant under a 1% significance level. The p values are 1.4×10^{-12} and 0.24 for RNS versus NS, 0.0004 and 0.009 for RNS versus NSr with criteria of *Cp* and *Sp*, respectively. As a result, RNS is better than NS in all criteria and better than NSr in *Cp*. In conclusion, the restart mechanism increases the efficiency of the neighborhood search.

To exhibit the performances of the three algorithms, Figure 3 depicts the Pareto fronts of some examples from the tested instances. Each case is solved 5 times, and the final Pareto front, which is obtained by considering all Pareto-optimal sets from five runs, is shown in the figures. The three figures show similar patterns: 1) on the Pareto front, there are many solutions before a specific point (the turning point) and few solutions after it; 2) the (1-belief reliability) decreases faster before the point than after it. Because the belief reliability cannot improve to a large degree after the cycle time goes beyond the cycle time of the turning point, one can make a good tradeoff between the cycle time and belief reliability by selecting the solutions with the cycle time smaller than or equal to the cycle time of the turning points. In contrary, the other solutions on the front correspond to a managerial decision of increasing a small amount of reliability but sacrificing a significant amount of production efficiency. These patterns can be explained by two reasons. First, by investigating Equation (4), the reliability deems to be a value controlled by one of the uncertainty distributions of station load. Therefore, As c increases, $\Psi_j(c)$ approaches 1 at a decreasing speed. Further, there are some limitations of our algorithm that the solutions are sparse near the extreme point (one with the largest cycle time) on the Pareto front. The turning point can be found by the following formula.

$$\max_i \frac{(1 - Br_i) - (1 - Br_{i-1})}{c_i - c_{i-1}} - \frac{(1 - Br_{i+1}) - (1 - Br_i)}{c_{i+1} - c_i}, \quad (13)$$

where $i \in Ps \backslash \{i = 1\}$.

TABLE 3. Results of the comparisons of algorithms.

		NS			NSr			RNS		
		Rp	Cp	Sp	Rp	Cp	Sp	Rp	Cp	Sp
Arcus	10	0.00	7.86	26.48	0.20	7.61	26.68	1.00	7.59	31.17
	12	0.00	7.67	24.49	0.20	7.59	28.38	1.00	7.60	29.19
	14	0.00	7.63	19.01	0.20	7.59	18.62	0.80	7.59	20.59
	16	0.00	7.60	20.56	0.07	7.59	19.34	0.98	7.58	20.09
	18	0.00	7.60	24.79	0.00	7.59	25.95	1.00	7.57	21.13
	20	0.00	7.62	17.54	0.20	7.60	18.47	0.80	7.59	21.96
Bartholdi	8	0.00	7.70	0.76	0.01	7.67	0.60	1.00	7.64	0.36
	10	0.00	7.73	0.67	0.01	7.70	0.63	1.00	7.69	0.46
	12	0.01	7.73	0.50	0.00	7.72	0.53	1.00	7.71	0.47
	14	0.00	7.79	0.59	0.01	7.76	0.59	1.00	7.74	0.56
	16	0.01	7.83	0.60	0.00	7.78	0.57	1.00	7.75	0.59
	18	0.00	7.87	0.58	0.00	7.79	0.62	1.00	7.80	0.58
Scholl	25	0.00	7.76	1.67	0.00	7.64	1.89	1.00	7.62	1.30
	27	0.00	7.68	1.74	0.20	7.64	1.68	1.00	7.60	1.51
	29	0.00	7.62	1.71	0.00	7.60	1.88	1.00	7.59	1.50
	31	0.00	7.71	1.54	0.00	7.61	1.67	1.00	7.61	1.34
	33	0.00	7.68	1.51	0.00	7.61	1.70	1.00	7.61	1.47
	35	0.00	7.61	1.50	0.00	7.60	1.57	1.00	7.60	1.42

Point i is the turning point where the maximal difference between the slopes of the turning point and its two adjacent points ($i-1$ and $i+1$) is achieved.

VI. CONCLUSIONS

Analyzing reliability is a very new topic for assembly line balancing problems. Almost all of the existing research focuses on measuring the reliability by probability theory, which requires data to obtain the task time distribution. However, there is a lack of data in some practical situations, especially when mass-producing customized items. The main objective of our paper is to characterize the reliability for assembly line balancing problem with uncertainty theory, which is based on the experts' information. The most recently established concept in production—belief reliability, is adopted herein as the reliability measure for assembly line production for the first time. A multi-objective optimization model is developed that considers the reliability and cycle time as objectives. Furthermore, a neighborhood search method with restart mechanism is proposed to solve the stated problem. Computational studies show that the restart mechanism increases the efficiency of the algorithm. Our research is the first attempt to optimize the reliability of the assembly line with uncertainty theory.

There are still some limitations to this study. The mathematical model only considers the regular constraints of ALBP, which can be complemented by incorporating more advanced constraints, e.g.,positional constraints and distance constraints. Moreover, the efficiency of the proposed algorithm can be further increased because the solutions near the extreme points of the Pareto front are sparse (Figure 3). In the future, there are several research directions worth to be investigated. Firstly, the objective can be the minimization of the number of workstations and belief reliability given the cycle time. Secondly, more real-world constraints can be integrated into the optimization models and other intelligent algorithms (particle swarm, genetic algorithm, etc.) are constructed to solve the models. Moreover, the ideas in this paper can be applied to other layout structures, including U-shaped lines and two-sided lines.

APPENDIX

In the appendix, we introduce some fundamental concepts and properties in uncertainty theory including uncertain measure, uncertain variable and uncertain programming.

Definition 4 [16]: Let $\mathcal{L}$ be a σ-algebra on a nonempty set Γ. A set function $\mathcal{M}:\mathcal{L}\to[0,1]$ is called an uncertain measure if it satisfies the following axioms,

Axiom 1: $\mathcal{M}\{\Gamma\}=1$ for the universal set Γ;

Axiom 2: $\mathcal{M}\{\Lambda\}+\mathcal{M}\{\Lambda^c\}=1$ for any event Λ;

Axiom 3: For every countable sequence of events $\Lambda_1,\Lambda_2,\cdots$, we have

$$\mathcal{M}\left\{\bigcup_{i=1}^{\infty}\Lambda_i\right\}\le\sum_{i=1}^{\infty}\mathcal{M}\{\Lambda_i\}.$$

In order to provide the operational law, the product uncertain measure on the product σ-algebre $\mathcal{L}$ is defined, which is called product axiom [17] .

Axiom 4: Let $(\Gamma_k,\mathcal{L}_k,\mathcal{M}_k)$ be uncertainty spaces for $k=1,2,\cdots$. The product uncertain measure $\mathcal{M}$ is an uncertain measure satisfying

$$\mathcal{M}\left\{\prod_{k=1}^{\infty}\Lambda_k\right\}=\bigwedge_{k=1}^{\infty}\mathcal{M}_k\{\Lambda_k\}$$

where Λ_k are arbitrarily chosen events from $\mathcal{L}_k$ for $k=1,2,\cdots$, respectively.

Definition 5 [16]: An uncertain variable ξ is a function from an uncertainty space $(\Gamma,L,\mathcal{M})$ to the set of real numbers such that for any Borel set B of real numbers, the set

$$\{\xi\in B\}=\{\gamma\in\Gamma|\xi(\gamma)\in B\}$$

is an event.

In order to describe uncertain variable in practice, uncertainty distribution $\Phi:\Re\to[0,1]$ of an uncertain variable ξ is defined as $\Phi(x)=\mathcal{M}\{\xi\le x\}$. An uncertainty distribution $\Phi(x)$ is said to be regular if it is a continuous and strictly increasing function with respect to x at which $0<\Phi(x)<1$, and

$$\lim_{x\to-\infty}\Phi(x)=0,\ \lim_{x\to+\infty}\Phi(x)=1.$$

If ξ is an uncertain variable with regular uncertainty distribution Φ, then we call the inverse function $\Phi^{-1}(\alpha)$ as the inverse uncertainty distribution of ξ.

An uncertain variable ξ is called normal if it has a normal uncertainty distribution

$$\Phi(x)=\left(1+exp\left(\frac{\pi(e-x)}{\sqrt{3}\sigma}\right)\right)^{-1},$$

denoted by $\mathcal{N}(e,\sigma)$, where e and σ are real numbers with $\sigma>0$. The inverse uncertainty distribution of normal uncertain variable $\mathcal{N}(e,\sigma)$ is

$$\Phi^{-1}(\alpha)=e+\frac{\sigma\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}.$$

Definition 6 [17]: The uncertain variables $\xi_1,\xi_2,\cdots,\xi_m$ are said to be independent if

$$\mathcal{M}\left\{\bigcap_{i=1}^{m}\{\xi_i\in B_i\}\right\}=\bigwedge_{i=1}^{m}\mathcal{M}\{\xi_i\in B_i\}$$

for any Borel sets $B_1,B_2,\cdots,B_m$ of real numbers.

The expected value operator of uncertain variable, proposed by Liu [16], is the average value of uncertain variable in the sense of uncertain measure, and represents the size of uncertain variable.

Definition 7 [16]: Let ξ be an uncertain variable. Then the expected value of ξ is defined as

$$E[\xi]=\int_0^{+\infty}\mathcal{M}\{\xi\ge x\}dx-\int_{-\infty}^{0}\mathcal{M}\{\xi\le x\}dx$$

provided that at least one of the two integrals is finite.

Further, for an uncertain variable ξ with uncertainty distribution $\Phi(x)$, Liu [19] showed that its expected value can be obtained by

$$E[\xi]=\int_{-\infty}^{+\infty}xd\Phi(x). \tag{14}$$

Furthermore, if $\Phi(x)$ is regular, then

$$E[\xi] = \int_0^1 \Phi^{-1}(\alpha)d\alpha. \tag{15}$$

Theorem 1 [17]: Let ξ and η be independent uncertain variables with finite expected values. Then for any real numbers a and b, we have

$$E[a\xi + b\eta] = aE[\xi] + bE[\eta].$$

Theorem 2 [19]: Assume $\xi_1, \xi_2, \cdots, \xi_n$ are independent uncertain variables with regular uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively. If $f(\xi_1, \xi_2, \cdots, \xi_n)$ is strictly increasing with respect to $\xi_1, \xi_2, \cdots, \xi_n$, then the uncertain variable $\xi = f(\xi_1, \xi_2, \cdots, \xi_n)$ has an inverse distribution

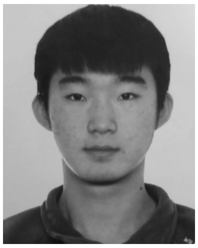
$$\Psi^{-1}(\alpha) = f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \cdots, \Phi_n^{-1}(\alpha)).$$

Theorem 3 [16]: Let $\xi_1, \xi_2, \cdots, \xi_n$ be independent normal uncertain variables with normal uncertainty distributions $\mathcal{N}(e_1, \sigma_1), \mathcal{N}(e_2, \sigma_2), \cdots,$ $\mathcal{N}(e_n, \sigma_n)$, respectively. Then, $\eta = \xi_1 + \xi_2 + \cdots + \xi_n$ follows normal uncertainty distributions $\mathcal{N}(\sum_{i=1}^n e_1, \sum_{i=1}^n \sigma_1)$.

REFERENCES

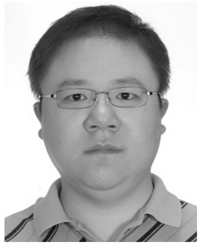
[1] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solution approaches," *Int. J. Prod. Econ.*, vol. 142, no. 2, pp. 259–277, 2013.
[2] A. Baykasoğlu and L. Özbakır, "Stochastic U-line balancing using genetic algorithms," *Int. J. Adv. Manuf. Technol.*, vol. 32, nos. 1–2, 139–147, 2007.
[3] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 694–715, 2006.
[4] N. Boysen, M. Fliedner, and A. Scholl, "Assembly line balancing: Which model to use when?" *Int. J. Prod. Econ.*, vol. 111, no. 2, pp. 509–528, 2008.
[5] B. Bryton, "Balancing of a continuous production line," M.S. thesis, Northwestern Univ., Evanston, IL, USA, 1954.
[6] B. Cakir, F. Altiparmak, and B. Dengiz, "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm," *Comput. Ind. Eng.*, vol. 60, no. 3, pp. 376–384, 2011.
[7] Y. Delice, E. K. Aydoğan, and U. Özcan, "Stochastic two-sided U-type assembly line balancing: A genetic algorithm approach," *Int. J. Prod. Res.*, vol. 54, no. 11, pp. 3429–3451, Feb. 2016.
[8] C. E. Ebeling, *An Introduction To Reliability And Maintainability Engineering*, Long Grove, IL, USA: Waveland Press, 2010.
[9] R. Kang, Q. Zhang, Z. Zeng, E. Zio, and X. Li, "Measuring reliability under epistemic uncertainty: Review on non-probabilistic reliability metrics," *Chin. J. Aeronaut.*, vol. 29, no. 3, pp. 571–579, 2016.
[10] H. Ke, H. Liu, and G. Tian, "An uncertain random programming model for project scheduling problem," *Int. J. Intell. Syst.*, vol. 30, no. 1, pp. 66–79, 2015.
[11] R. Li and G. Liu, "An uncertain goal programming model for machine scheduling problem," *J. Intell. Manuf.*, vol. 28, no. 3, pp. 689–694, 2017.
[12] Y. Li, "The type-II assembly line rebalancing problem considering stochastic task learning," *Int. J. Prod. Res.*, vol. 55, no. 24, pp. 7334–7355, 2017.
[13] Y. Li, M. Wen, R. Kang, and Z. Yang, "Type-1 assembly line balancing with uncertain task time," *J. Intell. Fuzzy Syst.*, vol. 35, no. 2, pp. 2619–2631, 2018.
[14] Z. Li, I. Kucukkoc, and J. M. Nilakantan, "Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem," *Comput. Oper. Res.*, vol. 84, pp. 146–161, Aug. 2017.
[15] Z. Li, Q. Tang, and L. Zhang, "Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm," *J. Cleaner Prod.*, vol. 135, pp. 508–522, Nov. 2016.
[16] B. Liu, *Uncertainty Theory*. Berlin, Germany: Springer, 2007.
[17] B. Liu, "Some research problems in uncertainty theory," *J. Uncertain Syst.*, vol. 3, no. 1, pp. 3–10, 2009.
[18] B. Liu, *Theory and Practice of Uncertain Programming*. Berlin, Germany: Springer, 2009.
[19] B. Liu, *Uncertainty Theory: A Branch of Mathematics for Modeling Human Uncertainty*. Berlin, Germany: Springer, 2010.
[20] B. Liu, "Why is there a need for uncertainty theory?" *J. Uncertain Syst.*, vol. 6, no. 1, pp. 3–10, 2012.
[21] B. Liu and K. Yao, "Uncertain multilevel programming: Algorithm and applications," *Comput. Ind. Eng.*, vol. 89, pp. 235–240, Nov. 2015.
[22] W. Q. Meeker and L. A. Escobar, *Statistical Methods for Reliability Data*. New York, NY, USA: Wiley, 1998.
[23] Y. Ning and T. Su, "A multilevel approach for modelling vehicle routing problem with uncertain travelling time," *J. Intell. Manuf.*, vol. 28, no. 3, pp. 683–688, 2017.
[24] U. Saif, Z. Guan, W. Liu, C. Zhang, and B. Wang, "Pareto based artificial bee colony algorithm for multi objective single model assembly line balancing with uncertain task times," *Comput. Ind. Eng.*, vol. 76, no. 1, pp. 1–15, 2014.
[25] U. Saif, Z. Guan, L. Zhang, J. Mirza, and Y. Lei, "Hybrid Pareto artificial bee colony algorithm for assembly line balancing with task time variations," *Int. J. Comput. Integr. Manuf.*, vol. 30, nos. 2–3, pp. 255–270, 2017.
[26] M. E. Salveson, "The assembly line balancing problem," *J. Ind. Eng.*, vol. 6, no. 3, pp. 18–25, 1955.
[27] A. Scholl, "Data of assembly line balancing problems," in *Schriften zur Quantitativen Betriebswirtschaftslehre*. Darmstadt, Germany: TU Darmstadt, 1993.
[28] Y. Sheng, Z. Qin, and G. Shi, "Minimum spanning tree problem of uncertain random network," *J. Intell. Manuf.*, vol. 28, no. 3, pp. 565–574, 2017.
[29] G. Suresh and S. Sahu, "Stochastic assembly line balancing using simulated annealing," *Int. J. Prod. Res.*, vol. 32, no. 8, pp. 1801–1810, 1994.
[30] Q. Tang, Z. Li, L. Zhang, and C. Zhang, "Balancing stochastic two-sided assembly line with multiple constraints using hybrid teaching-learning-based optimization algorithm," *Comput. Oper. Res.*, vol. 82, pp. 102–113, Jun. 2017.
[31] M. Wen, Z. Qin, and R. Kang, "The α-cost minimization model for capacitated facility location-allocation problem with uncertain demands," *Fuzzy Optim. Decis. Making*, vol. 13, no. 3, pp. 345–356, 2014.
[32] H. Yan and Y. Zhu, "Bang–bang control model with optimistic value criterion for uncertain switched systems," *J. Intell. Manuf.*, vol. 28, no. 3, pp. 527–534, 2017.
[33] Z. Zeng, R. Kang, M. Wen, and E. Zio, "A model-based reliability metric considering aleatory and epistemic uncertainty," *IEEE Access*, vol. 5, pp. 15505–15515, 2017.
[34] Z. Zeng, M. Wen, and R. Kang, "Belief reliability: A new metrics for products' reliability," *Fuzzy Optim. Decis. Making*, vol. 12, no. 1, pp. 15–27, 2013.
[35] Q. Zhang, R. Kang, and M. Wen, "Belief reliability for uncertain random systems," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3605–3614, Dec. 2018.
[36] W. Zhang, W. Xu, G. Liu, and M. Gen, "An effective hybrid evolutionary algorithm for stochastic multiobjective assembly line balancing problem," *J. Intell. Manuf.*, vol. 28, no. 3, pp. 783–790, 2017.

YUCHEN LI received the B.S. degree in reliability and systems engineering from Beihang University, in 2010, the M.S. degree in operations research from Columbia University, in 2012, and the Ph.D. degree in industrial and systems engineering from Rutgers University, in 2016. He is currently a Lecturer with the College of Economics and Management, Beijing University of Technology. His research has been published in some academic journals, including the *International Journal of Production Research* and *The International Journal of Advanced Manufacturing Technology*. His research interests include production optimization, game theory, and engineering economics.



YAO FU is currently pursuing the degree with the College of Economics and Management, Beijing University of Technology. His research interests include machine learning, financial engineering, and big data analysis.

XIAOWEN TANG received the Ph.D. degree from Sichuan University. He is currently an Associate Professor with the College of Economics and Management, Beijing University of Technology. His research interests include industry and business administration and strategic management.



XIAOFENG HU received the bachelor's and master's degrees from the Xi'an University of Architecture and Technology and the Ph.D. degree from Shanghai Jiao Tong University, where he is currently an Associate Professor with the School of Mechanical Engineering, Institute of Intelligent Manufacturing and Information Engineering. His research interests include the design and control of intelligent manufacturing systems and big data of manufacturing process.