Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

MIXED MODEL ASSEMBLY LINE BALANCING PROBLEM WITH FUZZY OPERATION TIMES AND DRIFTING OPERATIONS

Weida Xu
Tianyuan Xiao

Dept. of Automation
Tsinghua University
100084, P.R. CHINA

ABSTRACT

Assembly line balancing problem (ALBP) means assigning a series of task elements to uniform sequential stations under certain restrictions. This paper considers a special type of assembly line balancing problem with mixed models, fuzzy operation times and drifting operations, which has the objective of minimizing the total work overload time. According to chance constrained programming, a fuzzy α total work overload time minimization model is built. Moreover, fuzzy simulation and genetic algorithm are integrated to design a hybrid intelligent algorithm to solve the above model. Finally, Extensive computational results are reported to demonstrate the efficiency and effectiveness of the algorithm.

1 INTRODUCTION

1.1 Overview

An assembly line is a flow-oriented production system where the productive units performing the operations, referred to as stations, are aligned in a serial manner. The workpieces visit stations successively as they are moved along the line usually by some kind of transportation system, e.g. a conveyor belt. The current market is intensively competitive and consumer-centric. For example, in the automobile industry, most of the models have a number of features, and the customer can choose a model based on their desires and financial capability. Different features mean that different, additional parts must be added on the basic model. Due to high cost to build and maintain an assembly line, the manufacturers produce one model with different features or several models on a single assembly line. Under these circumstances, the mixed model assembly line balancing problem arises to smooth the production and decrease the cost.

Formally, a mixed model assembly line balancing problem can be stated as follows: given M models, the set of operations associated with each model, the processing time of each operation (operation time), and the set of precedence relations which specify the permissible orderings of the operations for each model, the problem is to assign the operations to an ordered sequence of stations such that precedence relations of each model are satisfied and some performance measures are optimized. Unlike the case of a single model line, different models of a product are assembled on a mixed model assembly line. The models are launched to the line one after another. Essentially, this problem is a sequencing problem with constraints: different sequences of operations being processed correspond to different allocation plans.

In mathematical complexity, the standard ALBP is NP-complete in strong sense because the NP-complete bin-packing problem can be easily transformed to it (Hop 2006). It becomes even more complicated when additional issues are addressed, such as mixed-model, uncertain operation times and so on. In this paper, we will consider a special type of assembly line balancing problem, which has mixed models, fuzzy operation times, and drifting operations.

The paper is organized as follows. The major difference between our research and other related works is explained in the remainder of this section. Section 2 provides a formal statement of our balancing problem, and Section 3 builds a fuzzy α total work overload time minimization model. In Section 4 a hybrid intelligent algorithm integrating fuzzy simulation and GA is designed. Then in order to reveal the effectiveness of the hybrid intelligent algorithm, Section 5 gives some numerical examples. Finally some conclusions are drawn in Section 6.

1.2 Major Differences between Our Research and Earlier Works

Assembly line balancing has been a topic of research for several decades. Most commonly, previous line balancing approaches have exploited one of two objectives. One objective (generally referred to as Type I problem) is to minimize the amount of stations given the required production rate (i.e. cycle time), operation times, and prece-

dence requirements, see Mendes et al. (2005) and Haq et al. (2006); the other objective (generally referred to as Type II problem) is to minimize the cycle time and maximize the production rate, with fixed number of stations, see Nearchou (2007). With either type, it is always assumed that the station time, which is the sum of times of all operations assigned to that station, must not exceed the cycle time. However, it is unnecessary or even impossible (e.g. when operation times are uncertain) to set a cycle time large enough to accommodate all the operations assigned to every station for each model. Whenever the operator cannot complete the preassigned operations on a workpiece, work overload occurs. Two different mechanisms can be adopted to cope with the case of an overload: (1) in the U.S., the operator returns back towards the next workpiece, and the uncompleted operation is left to utility workers (floaters) who have to be available whether or not work overload situations actually occur and must be highly qualified in order to perform a large set of different operations in a very short time (Tsai 1995); (2) in Japan, the operator stops the conveyor until the remaining work is completed (Zhao and Ohno 2000). Under the U.S. mechanism the operation diagram in a station is not influenced when overloads happen in other stations. Because of this advantage, many actual assembly lines in the world adopt the U.S. mechanism. Under the Japanese mechanism, however, an overload influences operation diagrams in all other stations due to the stoppage of the conveyor. Most research work focuses on the U.S. mechanism, see, for example, Tsai (1995), Matanachat and Yano (2001), Bautista and Cano (2008), and references cited in the survey Boysen et al. (2007) and a few focuses on the Japanese mechanism, see, for example, Zhao and Ohno (2000) and Celano et al. (2004). For mixed model assembly line balancing problem, the "work overload time" is an important performance measure, for the utility work is rather expensive. In this paper, we will use the U.S. mechanism and the objective is to minimize the total work overload time within a decision horizon.

To efficiently tune uneven operation times due to different models, in many actual mixed model assembly lines, in particular in automotive industry, the operator drifts with the operation, i.e. he or she visits the next consecutive station for a certain distance (which is called drifting distance in this paper). When the operation is finished, the operator leaves the workpiece and walks back to the station where the operation begun. If the operator is still unable to complete the operations after drifting to the next station for a drifting distance, work overload occurs. Very few researchers have considered this type of assembly line which is very common in reality. Merengo et al. (1999) and Zhao et al. (2004) have considered similar balancing problems, but they both suppose the operation times to be deterministic, Zhao et al. (2007) have considered the situation of stochastic operation times.

Yet another factor that distinguishes our research and others is the uncertainty of operation times. The large majority of literature addresses line balancing with deterministic operation times; some others concentrate on problems with stochastic operation times. Although, this assumption has been adopted and accorded with the facts in widespread cases, it is not reasonable in a vast range of situations. When we need to design a new, yet-to-be-built assembly line, the estimations of probability distributions of operation times are very difficult because there are not enough data to analysis. Instead, fuzzy theory can be employed to handle this case. For instance, based on experience, the time to finish a certain operation might be "between 60 and 100 seconds", or "around 2 minutes". However, very little attention has been paid to ALBP with fuzzy operation times. Only three paper can be found so far: Tsujimura et al. (1995) and Gen et al. (1996) have attempted to use genetic algorithm for solving fuzzy single model ALBP; Hop (2006) consider the fuzzy mixed model ALBP for the first time and formulated a fuzzy binary linear programming model which was then solved by a fuzzy heuristic. The purpose of fuzzy data approach is to represent more realistic situations whose data of problems are imprecise, vague, or almost unavailable. Also, the membership function of a fuzzy data represents the grade of satisfaction of a decision maker for the completion time of the operation. To the best of the authors' knowledge, no papers on balancing of the above-mentioned special type of assembly line with considerations of fuzzy operation times have been published so far. Nevertheless, fuzzy operation times are essential, especially when assembling tasks are performed by manual operations. Here, we use Triangular Fuzzy Numbers (TFNs) to represent fuzziness of operation time, and other forms of fuzzy number can be done in the same way.

2 PROBLEM DESCRIPTION

We first make some assumptions complied with most practical mixed model assembly lines:

1. The line is connected by a conveyor belt which moves at a constant speed. Consecutive workpieces are equispaced on the line by launching each after a cycle time.
2. Every workpiece is available at each station for a fixed time interval. During this interval, the work load (of the respective model) has to be performed by an operator while the workpiece ride downstream on the conveyor belt. If the work load is not finished within the cycle time, the operator can drift to the next consecutive station for a certain distance. If the drifting distance is reached without finishing the operations, work overload occurs. In this case, a utility worker is additionally employed to perform the remainder

work so fast that the work can be completed as soon as possible.

3. The operators of different stations do not interfere with each other while simultaneously servicing a workpiece (i.e. during drifting operations).
4. The operator returns to the upstream boundary of the station or the next workpiece, whatever is reached first, in zero time after finishing the work load on the current unit, because the conveyor speed is much smaller than the walking speed of the operators.
5. Precedence graphs of all models can be accumulated into a single combined precedence graph, similar operations of different models may have different operation time; zero operation time indicate that an operation is not required for a model.
6. Operation times of every model are independent variables and are descript by TFN.
7. Cycle time, number of stations, drifting distance, conveyor speed and the sequence of models to be assembled within the decision horizon are known.

And the notations used in the paper are as follows:

K number of stations; index: $k = 1, 2, \cdots, K$.

M number of models; index: $m = 1, 2, \cdots, M$.

D total number of workpieces to be assembled during the planning period.

$\boldsymbol{\pi}$ $\boldsymbol{\pi} = [\pi(1), \pi(2), \cdots, \pi(D)]$, assembly sequence vector, and $\pi(i)$ is the model identification number of the ith workpiece to be assembled during the planning period.

J number of operations; index: $j = 1, 2, \cdots, J$.

t_{mj} fuzzy processing time of operation j for one workpiece of model m.

T_{mk} total operation time (i.e. station time) of one workpiece of model m in station k.

ξ_{ij} processing time of operation j for the ith workpiece to be assembled during the planning period.

$\boldsymbol{\xi}$ $\boldsymbol{\xi} = [\xi_{ij}]_{D \times J}$, matrix representing all operation times of the workpieces to be assembled.

P combined precedence relation matrices corresponding to combined precedence graph.

l drifting distance, i.e. the distance an operator can get into the next consecutive station during drifting operations.

C cycle time.

V speed of the conveyor.

$\boldsymbol{A}$ $\boldsymbol{A} = (a_1, a_2, \cdots a_J)$, an allocation plan of operations, a_j means operation j is assigned to the station with the index of a_j .

$x_{ik}(\boldsymbol{A}, \boldsymbol{\xi})$ with allocation plan $\boldsymbol{A}$, the time spent by the operator of station k to process the i-th workpiece before a drifting operation happens.

$y_{ik}(\boldsymbol{A}, \boldsymbol{\xi})$ with allocation plan $\boldsymbol{A}$, the time spent by the operator of station k to process the i-th workpiece during the drifting operation.

$WOT_{ik}(\boldsymbol{A}, \boldsymbol{\xi})$ with allocation plan $\boldsymbol{A}$, work overload time caused by the ith workpiece in station k.

$WOT(\boldsymbol{A}, \boldsymbol{\xi})$ with allocation plan $\boldsymbol{A}$, total work overload time during the planning period.

With the above assumptions and notations, our special assembly line balancing problem can be stated as: given the sets of models and their operation times, the operations' precedence relations, the cycle time C, number of stations K, conveyor speed V, drifting distance l, and the sequence of workpieces to be assembled in the decision horizon $\boldsymbol{\pi}$, the question is to find an allocation plan $\boldsymbol{A}$ assigning operations of each model to K stations, and minimize the total work overload time. In addition, the processing times of all the operations are descript with TFN.

Theorem 1 *The following iterative relationships hold:*

$$x_{ik}(A, \xi) = \min\left(\frac{CV + l}{V} - y_{(i-1),k}(A, \xi), T_{\pi(i)k}\right) \quad (1)$$

$$y_{ik}(A, \xi) = \max\left[\min\left(T_{\pi(i)k} + y_{(i-1),k}(A, \xi), \frac{CV + l}{V}\right) - C, 0\right] \quad (2)$$

$$WOT_{ik}(A, \xi) = \max\left(T_{\pi(i)k} + y_{(i-1),k}(A, \xi) - \frac{CV + l}{V}, 0\right) . \quad (3)$$

Proof: Figure 1 depicts the movements of the operator of station k and four workpieces (i.e. $\pi(1), \pi(2), \pi(3), \pi(4)$) in two dimensions: the horizontal axis represents distance and the vertical axis, time. The distance between the two vertical full lines depicts the distance a workpiece travels within a cycle time, the distance between the right vertical full line and the vertical dash-dotted line depicts the drifting distance. The oblique lines (both full and dashdotted) trace the workpiece movements: a full line represents the workpiece's downstream movement while being worked by the operator and a dashdotted line shows its downstream movement before and after the assembly process. A horizontal dashed line traces the operator's walk upstream to meet the next workpiece.

From Figure 1, we can see that while working on the (*i-1*)-th workpiece, the time spent on drifting operation is $y_{(i-1)k}(\boldsymbol{A}, \boldsymbol{\xi})$. That means, before starting working on the i-th workpiece in sequence $\boldsymbol{\pi}$, the time left to the operator is only $(CV + l)/V - y_{(i-1),k}(A, \xi)$. If the operations allocated to this station can be successfully finished within this period, $x_{ik}(\boldsymbol{A}, \boldsymbol{\xi})$ is equal to the workpiece's station

time $T_{\pi(i)k}$; otherwise, drifting operation occurs, and $x_{ik}(\boldsymbol{A},\boldsymbol{\xi})$ is equal to $(CV+l)/V - y_{(i-1),k}(A,\xi)$, which leads to (1).

If the *i*-th workpiece can be finished in the cycle time, apparently, $y_{ik}(\boldsymbol{A},\boldsymbol{\xi})=0$; if it cannot, but can be finished through drifting operation (i.e. work overload does not happens), $y_{ik}(A,\xi)=T_{\pi(i)k}+y_{(i-1),k}(A,\xi)-C$; otherwise (i.e. work overload happens), $y_{ik}(A,\xi)=(CV+l)/V-C$, leading to (2).

If the *i*th workpiece can be finished before drifting operation ends, work overload does not occur, $WOT_{ik}(\boldsymbol{A},\boldsymbol{\xi})=0$; otherwise, $WOT_{ik}(\boldsymbol{A},\boldsymbol{\xi})$ is equal to the total time of unfinished operations when drifting operation ends $T_{\pi(i)k}+y_{(i-1),k}(A,\xi)-(CV+l)/V$, which gives (3). □

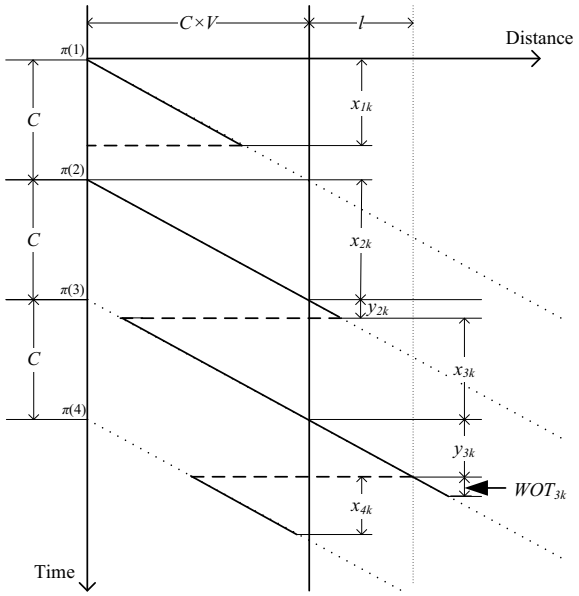

Figure 1: The movement paths of an operator and four workpieces in one station.

Therefore, with **Theorem 1**, we can get the total work overload time during the planning period $WOT(\boldsymbol{A},\boldsymbol{\xi})$. Considering the model is very complicated and involves fuzzy uncertainties, we will use computer simulation method to calculate $WOT(\boldsymbol{A},\boldsymbol{\xi})$ (details about the computer simulation method will be descript in Section 4).

3 FUZZY α TOTAL WORK OVERLOAD TIME MINIZATION MODEL

In some practical optimization problems, the decision maker may be disgusted with risk, and prefer the system to be more reliable rather than just having a very high expected performance. A conservation design strategy is therefore selected to optimize the system with some given confidence levels provided as an appropriate safety margin. This is the idea of chance-constrained programming (CCP). CCP was first developed by Charnes and Cooper (1959), and offered a powerful means of modeling stochastic decision systems with assumption that the stochastic constraints will hold at least α of time, where α is referred to as the confidence level. After that, Liu (1999) generalized CCP to the case with not only stochastic constraints but also stochastic objectives. A framework of fuzzy CCP has been present by Liu and Iwamura (1998a, 1998b) and Liu (1998). In our problem, we tend to minimize the total work overload time with a confidence level under the precedence constraints. In this case, an α total work overload time minimization model will be established base on a new concept of α total work overload time as follows.

Definition 1 *Given a predetermined confidence level α , the fuzzy α total work overload time during the decision horizon is defined as:*

$$\min\{WOT^0 \mid \mathrm{Cr}\{WOT(A,\xi)\le WOT^0\}\ge\alpha\}.$$

In order to minimize the α total work overload time of an assembly line, the following so-called α total work overload time minimization model is obtained:

$$\begin{cases} \min WOT^0 \\ \text{s.t.} \\ \quad \mathrm{Cr}\{WOT(\boldsymbol{A},\boldsymbol{\xi})\le WOT^0\}\ge\alpha \\ \quad \forall p_{ij}=1,\ a_i\le a_j \\ \quad 1\le a_i,a_j\le K,\ a_i,a_j \text{ are integers} \\ \quad 1\le i,j\le J,\ \ i,j \text{ are integers} \end{cases}, \tag{4}$$

where α a confidence level determined by the decision maker, $WOT(\boldsymbol{A},\boldsymbol{\xi})=\sum_{k=1}^{K}\sum_{i=1}^{D}WOT_{ik}(\boldsymbol{A},\boldsymbol{\xi})$, and $WOT_{ik}(\boldsymbol{A},\boldsymbol{\xi})$ is defined by equation (3) in **Theorem 1**. In addition, Cr is the credibility measure defined by Liu (2007).

4 HYBRID INTELLIGENT ALGORITHM

The development of algorithms to solve ALBP has a long history. A famous method is Jackson's algorithm, which enumerates possible combinations of operations based on precedence relationships between them (Jackson 1956). Another method is the Branch and Bound method which cuts combinations having performance index values lower than some bound value (Bukchin and Rabinowitch 2006). Some other methods are integer programming (Talbot and Patterson 1984), dynamic programming (Henig 1986, Carraway 1989) etc. It becomes hard to solve ALBP by these traditional methods for problems involving large numbers of operations and stations, and these methods are mainly

applied to problems of getting the minimum number of station for a given cycle time (Type I problem). Heuristic algorithms have also been applied to ALBP, for example, Helgeson and Birnie (1961), Moodie and Young (1965), Arcus (1966), McMullen and Frazier (1997), Gaither and Frazier (1999) etc. These heuristic algorithms are very practical; however, it is not guaranteed that they will introduce the optimum solution. In recent years, numerous research efforts have been directed towards the development of intelligent algorithms, such as neural networks, simulated annealing and genetic algorithm to provide an alternative to traditional optimization techniques. Hashimoto et al. (1993, 1994) have applied the Hopfield neural network to ALBP, and showed that middle-scale ALBP can be solved efficiently by it. Suresh and Sahu (1994), McMullen and Frazier (1998) have developed simulated annealing algorithms to solve ALBP, and get optimum solutions, but numerical examples have shown that when used to solve the above-mentioned special type of ALBP, especially for rather large-scale problems, the simulated annealing algorithm is inefficient in finding optimum solutions (see for Section 5).

Genetic algorithm (GA) is a stochastic search method for optimization problems based on the mechanics of natural selection and natural genetics (i.e. survival of the fittest). When the objective functions to be optimized are multimodal or the search spaces are particularly irregular, algorithms need to be highly robust in order to avoid get stuck at a local optimal solution. The advantage of GA is just able to obtain the global optimal solution fairly. During the past three decades, GA has demonstrated considerable success in providing good solutions to many combinational optimization problems, especially for sequencing problems such as travelling salesman problems, flow-shop and job-shop scheduling problems, and so on.

The application of GA for assembly line balancing has been widely studied so far: Anderson and Ferris (1994) proposed a GA for Type II problem, and Leu et al. (1994) presented a GA-based approach to solve Type I problems with multiple objectives. Tsujimura et al. (1995) and Gen et al. (1996) have developed similar GAs to solve fuzzy ALBP. Haq et al. (2006) studied mixed model ALBP using GA. All these studies are different from this work not only in terms of the problem considered, but also in terms of the hybrid approach. In this approach, we integrated fuzzy simulation and genetic algorithm, where fuzzy simulation is used to calculate the system performance (i.e. α total work overload time), while genetic algorithm is employed to search for the optimal solution.

4.1 Fuzzy Simulation

In model (4), there exists a uncertain function with fuzzy variables. Due to the complexity, we design a fuzzy simulation to calculate these uncertain functions according to the concepts of credibility measure of fuzzy variables. The interested readers can learn details pertaining to fuzzy simulation by referring to the book of Liu (2002).

Specifically speaking, the uncertain function need to be simulated is:

$$U : \boldsymbol{A} \to \min\{WOT^0 \mid \mathrm{Cr}\{WOT(\boldsymbol{A},\boldsymbol{\xi}) \le WOT^0\} \ge \alpha\}.$$

In order to estimate it, we need to find the minimal value WOT^0 such that $\mathrm{Cr}\{WOT(\boldsymbol{A},\boldsymbol{\xi}) \le WOT^0\} \ge \alpha$. The fuzzy simulation for calculating U can be summarized as follows:

Step 1. Randomly generate $\xi_{ij}^{(l)}$ from the ε-level sets of fuzzy variables ξ_{ij}, $i = 1,2,\cdots,D$, $j = 1,2,\cdots,J$, respectively, $l = 1,2,\cdots,N$, where N is a sufficiently big number that denotes the maximum cycle number of the fuzzy simulation and ε is a sufficiently small number.

Step 2. Set $\boldsymbol{\xi}^{(l)} = [\xi_{ij}^{(l)}]_{D\times J}$ and $v^{(l)} = \min_{i,j}[\mu_{ij}(\xi_{ij}^{(l)})]$, where $\mu_{ij}(\cdot)$ is the membership function of ξ_{ij}.

Step 3. Calculate $WOT(\boldsymbol{A},\boldsymbol{\xi}^{(l)})$ through computer simulation.

Step 4. Employ binary search to find the maximal value r such that $L(r) \ge \alpha$ holds, where

$$L(r) = \frac{1}{2}\left(\max_{1\le l\le N}\left\{v^{(l)} \mid WOT(\boldsymbol{A},\boldsymbol{\xi}^{(l)}) \le r\right\} + \min_{1\le l\le N}\left\{1 - v^{(l)} \mid WOT(\boldsymbol{A},\boldsymbol{\xi}^{(l)}) > r\right\}\right)$$

Step 5. Return r.

4.2 Genetic Algorithm

In this subsection, we design a genetic algorithm that embeds the fuzzy simulation proposed above. Now let us introduce the algorithm detailedly in the following way:

4.2.1 Representation Structure

We represent an allocation plan by the chromosome $\boldsymbol{A} = (a_1, a_2, \cdots a_J)$, where a_j means operation j is assigned to the station with the index of a_j.

4.2.2 Initial Process

The initial population of our GA is composed of two parts: some chromosomes of the population are generated through heuristics (details of these heuristics will be explicated in Section 5), and the others are randomly derived. We denote the chromosomes in the population by $\boldsymbol{A}^{(n)}$, $n = 1,2,\cdots, pop\_size$.

4.2.3 Evaluation Function

We first calculate the objective values for all chromosomes $\boldsymbol{A}^{(n)}$, $n=1,2,\cdots,pop\_size$ by the designed fuzzy simulation. According to the objective values, an order relationship among these chromosomes is presented to rearrange them from good to bad. Then, the fitness of each chromosome can be computed by some evaluation function. Here, the rank-based evaluation function is defined as:

$$Eval(\boldsymbol{A}^{(n)}) = p(1-p)^{n-1},\ n=1,2,\cdots,pop\_size,$$

where the chromosomes $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(pop\_size)}$ are assumed to have been rearranged from good to bad according to their objective values and $p\in(0,1)$ is a parameter in the genetic system.

4.2.4 Selection Process

Generally, we select the chromosomes for a new population based on spinning the roulette wheel characterized by the fitness of all chromosomes for *pop_size* times to generate a new generation to update the chromosomes. Firstly, we compute the cumulative probability q_i for each chromosome $\boldsymbol{A}^{(n)}$, where $q_0=0$ and

$$q_n=\sum_{j=1}^{n} Eval(\boldsymbol{A}^{(j)}),\ n=1,2,\cdots,pop\_size.$$

Secondly, we randomly generate a number *rand* in $[0,q_{pop\_size}]$. Then we select the chromosome $\boldsymbol{A}^{(n)}$ such that $q_{n-1}\le rand\le q_n$. Repeating the above steps *pop_size* times, we can obtain *pop_size* copies of chromosome to be a new generation of chromosomes.

4.2.5 Crossover Operation

The crossover operation is used to renew the chromosomes $\boldsymbol{A}^{(n)}$, $n=1,2,\cdots,pop\_size$ with the probability P_c. In order to determine two parents for crossover operation, we repeat the following process *pop_size* times: generating a random real number *rand* from the interval $[0,1]$, if $rand<P_c$, we randomly select two chromosomes (i.e. two parents) from the population, denoted as $\boldsymbol{A}^{(n_1)}$ and $\boldsymbol{A}^{(n_2)}$. Then we cross $\boldsymbol{A}^{(n_1)}$ and $\boldsymbol{A}^{(n_2)}$ in the following way: initially, randomly select a crossover point. Then, one child inherits the elements between the starting point and crossover point from the first parent, in the order and position in which this parent appeared; the remaining elements are inherited from the second parent, beginning with the crossover point and ending up with the last point. The other child is generated in the same way, except that firstly inherits from the second parent, and then from the first parent. If both children are feasible solutions, we replace the parents with them. If not, we keep the feasible one if it exists, and then redo the crossover operation by regenerating another crossover index until totally two feasible children are obtained. In this way, we only replace the parents with the feasible children. Finally, *pop_size* new chromosomes $\boldsymbol{A}^{(n)}$, $n=1,2,\cdots,pop\_size$ are obtained.

4.2.6 Mutation Operation

Now we update the chromosomes $\boldsymbol{A}^{(n)}$, $n=1,2,\cdots,pop\_size$ by mutation operation with the probability P_m. Similar to the process of selecting parents for crossover operation, we repeat the following steps *pop_size* times: generating random real number from the interval $[0,1]$, if $rand<P_m$, we randomly select a chromosome (i.e. a parent) from the population, denoted as $\boldsymbol{A}^{(n)}$. Then we mutate $\boldsymbol{A}^{(n)}$ through trade and transfer method (Reeve and Thomas 1973): (1) interchanging the positions of two operations (trade); (2) shifting an operation from one station to another (transfer). Like crossover operation, if the child is feasible for the precedence constraints, we replace the parent with the child; otherwise, we redo the mutation operation until a feasible child is obtained.

4.3 Hybrid Intelligent Algorithm

The hybrid intelligent algorithm designed above to solve the fuzzy model (4) are summarized as follows:

Step 1. Initialize *pop_size* chromosomes.

Step 2. Calculate the objective values for all chromosomes through fuzzy simulation.

Step 3. Compute the fitness of each chromosome by rank-based evaluation function based on their objective values.

Step 4. Select the chromosomes by spinning the roulette wheel.

Step 5. Update the chromosomes by crossover and mutation operations.

Step 6. Repeat the second to fifth steps a given number of times.

Step 7. Report the best chromosome as the optimal solution.

5 COMPUTATIONAL EXPERIMENTS

To assess the desirability of the proposed algorithms, we give some numerical examples that are performed on a personal computer. We choose five combined precedence relation matrices from the stand ALBP lib (Scholl 2007); operation times are assumed to be descript by TFNs, whose parameters are randomly generated. All problems have 3 models; total numbers of workpieces to be assembled are all 50 and every sequence is randomly generated. The conveyor speed is set as V=0.01m/s, cycle time is

C=380s, drifting distance is l=1m for all problems, and station quantities are determined by the number of operations.

Suppose the decision-maker want to reasonably assign operations, and the objective is to minimize the 0.95 total work overload time, the following 0.95 total work overload time minimization model is obtained:

$$\begin{cases} \min WOT^0 \\ \text{s.t.} \\ \quad \mathrm{Cr}\{WOT(\boldsymbol{A},\boldsymbol{\xi}) \le WOT^0\} \ge 0.95 \\ \quad \forall p_{ij} = 1,\ a_i \le a_j \\ \quad 1 \le a_i, a_j \le K,\ a_i, a_j \text{ are integers} \\ \quad 1 \le i, j \le J,\ \ i, j \text{ are integers} \end{cases}$$

For each problem, experimentation is performed to compare the hybrid intelligent algorithm with several others, including 13 heuristics extended from the literature and simulated annealing approach provided by McMullen and Frazier (1998). Details of the 13 heuristics are contained in Table 1. The interested reader can learn more about them by referring to Arcus (1966) for H1, Moodie and Young (1965) for H2 and H3, Gaither and Frazier (1999) for H7 and H8, Helgeson and Birnie (1961) for H13, and McMullen and Frazier (1997) for the remaining heuristics.

Each of the five problems (21, 35, 45, 58, and 83 tasks) is solved by the 15 approaches 10 times, i.e. totally 250 experiments are performed. Table 2 shows means, standard deviations of total work overload time, along with the ranking of each approach.

Table 1: Description of the heuristics used in the comparison and initialization of GA.

Label	Description
H1	The probability of selection for each operation on the appropriate list is directly proportional to its expected processing time.
H2	The operation with the longest expected processing time is selected to enter the current station.
H3	The operation with the shortest expected processing time is selected to enter the current station.
H4	The operation with the shortest 0.95 work overload time of the current station is selected, if several operations yield the same work overload time, the first one available is selected.
H5	The operation with the shortest 0.95 work overload time of the current station is selected, if several operations yield the same work overload time, the last one available is selected.
H6	Selects the operation on the appropriate list with the largest expected station time and smallest 0.95 work overload time. This rule is an attempt to "have the best of both worlds" in terms of station time and work overload time.
H7	The first available operation is selected.
H8	The last available operation is selected.
H9	The operation having the most followers is selected.
H10	The operation having the fewest followers is selected.
H11	The operation having the most immediate followers is selected.
H12	The operation having the fewest immediate followers is selected.
H13	The operation having the highest ranked positional weight is selected, and ranked positional weight means the sum of the processing time of the current operation and all its followers.

Table 2: Means (rankings) and standard deviations of investigated approaches. The first row of the table is the name of problems, and the number in the parentheses behind each name indicates the number of operations; the first row column lists the name of approaches.

	MITCHELL (21)		GUNTHER (35)		KILBRID (45)		WARNECKE (58)		ARC (83)	
	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
H1	2493.76 (11)	340.88	3557.60 (13)	926.50	5170.43 (7)	1663.07	6555.77 (7)	1648.24	11836.45 (9)	715.16
H2	2365.72 (9)	166.74	2805.37 (5)	322.83	6311.28 (11)	338.29	8853.30 (13)	318.20	16085.35 (13)	317.69
H3	2391.38 (10)	170.39	2806.06 (6)	318.72	6667.70 (12)	374.97	3780.78 (3)	227.44	9752.02 (6)	360.95
H4	2081.51 (6)	176.73	4187.85 (14)	574.09	5415.75 (8)	435.60	5894.32 (5)	1533.08	7270.18 (4)	1307.23
H5	2681.40 (12)	202.03	3500.12 (11)	286.04	4157.67 (4)	529.47	7793.57 (10)	1102.72	11112.79 (8)	1523.92
H6	2114.42 (7)	167.93	1560.63 (3)	855.69	5914.73 (9)	1336.82	6174.53 (6)	779.34	12996.29 (11)	2083.36
H7	3730.73 (15)	274.88	2581.79 (4)	192.90	6186.92 (10)	279.32	9890.55 (15)	389.79	16321.35 (14)	620.57
H8	2361.78 (8)	129.79	3517.88 (12)	135.86	2818.08 (3)	209.78	8365.24 (11)	434.00	7375.24 (5)	650.16
H9	1978.17 (5)	239.74	3445.48 (10)	208.39	8037.55 (14)	369.47	8968.59 (14)	386.81	10478.31 (7)	366.54
H10	1877.23 (3)	170.93	3301.63 (8)	284.81	5134.42 (6)	286.94	7654.68 (9)	465.08	12775.17 (10)	381.21
H11	3232.17 (14)	128.98	3014.38 (7)	237.40	8277.98 (15)	353.34	6963.46 (8)	322.20	16599.36 (15)	452.42
H12	1956.23 (4)	215.54	3406.55 (9)	284.46	7139.46 (13)	373.80	8787.19 (12)	522.15	7205.15 (3)	439.25
H13	2686.19 (13)	255.79	4236.95 (15)	136.72	4522.12 (5)	112.05	4652.90 (4)	254.94	13068.40 (12)	429.73
SA	0.00 (1)	0.00	528.50 (2)	193.47	1158.75 (2)	274.13	2045.08 (2)	630.01	2698.58 (2)	314.10
HIA	0.00 (1)	0.00	1.06 (1)	2.10	121.78 (1)	188.57	252.61 (1)	168.49	372.70 (1)	175.87

From inspection of Table 2, it is clear that heuristics are far worse than intelligent algorithms (i.e., simulated annealing and genetic algorithm). There does not exist a single heuristic that always guarantee better result than others for all the problems. For small- and middle-scale problems, both intelligent algorithms can get desirable allocation plans. However, when it comes to large-scale problems, the hybrid intelligent algorithm provides performances far better than what simulating annealing does. As a result, we can say that the hybrid intelligent algorithm is a relatively robust approach for the special type of ALBP outlined in this paper.

6 CONCLUSION

In this paper, a special type of assembly line balancing problem with station lengths longer than the distance conveyor moved within one cycle time is investigated in fuzzy environments, where operation times are assumed to be fuzzy variables, and the objective is to minimize the total work overload time during the decision horizon. Based on explicit formulation of this problem, we proposed a fuzzy α total work overload time minimization model. In order to solve this model efficiently, a fuzzy simulation is designed and embedded into genetic algorithm to produce a hybrid intelligent algorithm. Finally some computational experiments are given to show the effectiveness of the proposed algorithm.

REFERENCES

Anderson, E. J., and M. C. Ferris. 1994. Genetic algorithms for combinatorial optimization: The assemble line balancing problem. *ORSA Journal on Computing* 6 (2):161-183.

Arcus, A. L. 1966. COMSOAL: A computer method of sequencing operations for assembly lines. *International Journal of Production Research* 4 (4):259-277.

Bautista, J., and J. Cano. 2008. Minimizing work overload in mixed-model assembly lines. *International Journal of Production Economics* 112 (1):177-191.

Boysen, N., M. Fliedner, and A. Scholl. 2007. A classification of assembly line balancing problems. *European Journal of Operational Research* 183 (2):674-693.

Bukchin, Y., and I. Rabinowitch. 2006. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. *European Journal of Operational Research* 174 (1):492-508.

Carraway, R. L. 1989. A dynamic programming approach to stochastic assembly line balancing. *Management Science* 35 (4):459-471.

Celano, G., A. Costa, S. Fichera, and G. Perrone. 2004. Human factor policy testing in the sequencing of manual mixed model assembly lines. *Computers & Operations Research* 31 (1):39-59.

Charnes, A., and W. W. Cooper. 1959. Chance-constrained programming. *Management Science* 6 (1):73-79.

Gaither, N., and G. Frazier. 1999. *Production and operations management* 8th ed. Cincinnati, Ohio: South-Western College Pub.

Gen, M., Y. Tsujimura, and Y. Li. 1996. Fuzzy assembly line balancing using genetic algorithms. *Computers & Industrial Engineering* 31 (3-4):631-634.

Haq, A. N., K. Rengarajan, and J. Jayaprakash. 2006. A hybrid genetic algorithm approach to mixed-model assembly line balancing. *The International Journal of Advanced Manufacturing Technology* 28 (3):337-341.

Hashimoto, Y., I. Nishikawa, and H. Tokumaru. 1993. Line balancing using a hopfield network. In *Proceeding of the Korea Automatic Control Conference*, 391-394. Seoul, Korea.

Hashimoto, Y., I. Nishikawa, and H. Tokumaru. 1994. Line balancing problems using a hopfield network. In *Proceeding of the 1994 Japan-US Symposium on Flexible Automation*, 1369-1375. Kobe, Japan.

Helgeson, W. B., and D. P. Birnie. 1961. Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering* 12 (6):394-398.

Henig, M. I. 1986. Extensions of the dynamic programming method in the deterministic and stochastic assembly-line balancing problems. *Computers & Operations Research* 13 (4):443-449.

Hop, N. V. 2006. A heuristic solution for fuzzy mixed-model line balancing problem. *European Journal of Operational Research* 168 (3):798-810.

Jackson, J. R. 1956. A computing procedure for a line balancing problem. *Management Science* 2 (3):261-271.

Leu, Y. Y., L. A. Matheson, and L. P. Rees. 1994. Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria. *Decision Sciences* 25 (4):581-606.

Liu, B. 1998. Minimax chance constrained programming models for fuzzy decision systems. *Information Sciences* 112 (1-4):25-38.

Liu, B. 1999. *Uncertain programming*. New York: Wiley.

Liu, B. 2002. *Theory and practice of uncertain programming*. Heidelberg: Physica-Verlag.

Liu, B. 2007. *Uncertain theory*. 2nd ed. Berlin: Springer-Verlag.

Liu, B., and K. Iwamura. 1998a. Chance constrained programming with fuzzy parameters. *Fuzzy Sets and Systems* 94 (2):227-237.

Liu, B., and K. Iwamura. 1998b. A note on chance constrained programming with fuzzy coefficients. *Fuzzy Sets and Systems* 100 (1-3):229-233.

Matanachat, S., and C. A. Yano. 2001. Balancing mixed-model assembly lines to reduce work overload. *IIE Transactions* 33 (1):29-42.

McMullen, P. R., and G. V. Frazier. 1997. A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics* 51 (3):177-190.

McMullen, P. R., and G. V. Frazier. 1998. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *International Journal of Production Research* 36 (10):2717-2741.

Mendes, A. R., A. L. Ramos, A. S. Simaria, and P. M. Vilarinho. 2005. Combining heuristic procedures and simulation models for balancing a pc camera assembly line. *Computers & Industrial Engineering* 49 (3):413-431.

Merengo, C., F. Nava, and A. Pozzetti. 1999. Balancing and sequencing manual mixed-model assembly lines. *International Journal of Production Research* 37 (12):2835-2860.

Moodie, C. L., and H. H. Young. 1965. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering* 16 (4):23-28.

Nearchou, A. 2007. Balancing large assembly lines by a new heuristic based on differential evolution method. *The International Journal of Advanced Manufacturing Technology* 34 (9):1016-1029.

Reeve, N. R., and W. H. Thomas. 1973. Balancing stochastic assembly lines. *IIE Transactions* 5 (3):223-229.

Scholl, A. 2007. Homepage for assembly line optimization research [online]. Available via <http://www.assembly-line-balancing.de/> [accessed July 5, 2007].

Suresh, G., and S. Sahu. 1994. Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research* 32 (8):1801-1810.

Talbot, F. B., and J. H. Patterson. 1984. An integer programming algorithm with network cuts for solving the assembly line balancing problem. *Management Science* 30 (1):85-99.

Tsai, L.-H. 1995. Mixed-model sequencing to minimize utility work and the risk of conveyor stoppage. *Management Science* 41 (3):485-495.

Tsujimura, Y., M. Gen, and E. Kubota. 1995. Solving fuzzy assembly-line balancing problem with genetic algorithms. *Computers & Industrial Engineering* 29 (1-4):543-547.

Zhao, X., J. Liu, K. Ohn, and S. Kotani. 2007. Modeling and analysis of a mixed-model assembly line with stochastic operation times. *Naval Research Logistics* 54 (6):681-691.

Zhao, X., and K. Ohno. 2000. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. *European Journal of Operational Research* 124 (3):560-570.

Zhao, X., K. Ohno, and H.-S. Lau. 2004. A balancing problem for mixed model assembly lines with a paced moving conveyor. *Naval Research Logistics* 51 (3):446-464.

AUTHOR BIOGRAPHIES

WEIDA XU received his B.S. degree from the Department of Automatic Control, Huazhong University of Science and Technology, P.R. China in 2004, and was recommended to Tsinghua University with entrance-exam exemption in the same year. Mr. Xu is now a PhD candidate in the Department of Automation, Tsinghua University, P.R. China. His current research interests are in the fields of manufacturing system modeling, simulation and optimization, etc. His email is <xwd04@mails.tsinghua.edu.cn>.

TIANYUAN XIAO is the Professor in the Department of Automation, Tsinghua University, P.R. China. He also is the executive director of the National CIMS Engineering Research Center of China. His research fields include CIMS, Virtual Manufacturing, Modeling and simulation, Networked Manufacturing. His email is <xty-dau@mail.tsinghua.edu.cn>.