Earliness/Tardiness Flow-shop Scheduling under uncertainty

Li sufen¹, Zhu yunlong¹, Li xiaoying²

1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; 2. Hebei Polytechnic University, Tangshan 063009, China
lisufen@sia.cn

Abstract

For unconventional scheduling problems, both job sequence and starting time of every operation must be optimized, which increases the difficulties of solving this kind of problem. In this paper, we advanced a jointed algorithm for solving flow-shop earliness/tardiness problem based on genetic algorithm and heuristic algorithm. Hierarchy scheduling paradigm was introduced. First, the genetic algorithm is used to determine preferably scheduling sequence. Second, a kind of new heuristic algorithm was put forward to adjust the starting times for the present schedule. Where, the heuristic algorithm determined what time and how long every idle time should be. The object of this paper is to minimize the total earliness and tardiness penalties of all the jobs. The numerical results obtained prove the correctness and efficiencies of the jointed algorithm.

1. Introduction

Scheduling problems involving both earliness and tardiness costs have received significant attention in recent years. This type of problem became important with the advent of the just-in-time (JIT) concept, where early or tardy deliveries are highly discouraged.

In the JIT scheduling environment, the product should be finished as close to the due-date as possible. An early job completion results in inventory carrying costs, such as storage and insurance costs. On the other hand, a tardy job completion results in penalties, such as loss of customer goodwill and damaged reputation. But, in production, there is always set of jobs finished before the due-date, while another set will be completed after the desired due-date. By quantifying the inventory and other related costs into earliness penalties and customer dissatisfaction into tardiness penalties, one can evaluate the effect of different schedules and even try to reach the optimal sequence of jobs to be processed.

With increased interest in production scheduling in recent years, a significant amount of the work in this area has focused on the development of deterministic models, where the problem data is assumed to be known in advance. In reality, though, there can be uncertainty in a number of factors such as processing times and costs. As a result, a number of papers in the recent years have addressed scheduling in the face of uncertainties in different parameters, for example demands⁷⁻¹² and processing times¹⁻⁶⁻¹⁷.

The prevalent approach to the treatment of these uncertainties is through the use of probabilistic models that describe the uncertain parameters in terms of probability distributions. However, the evaluation and optimization of these models is computationally expensive, either because of the large number of scenarios resulting from a discrete representation of the uncertainty¹⁹, or the need to use complicated multiple integration techniques when the uncertainty is represented by continuous distributions¹⁵.

Furthermore, the use of probabilistic models is realistic only when these descriptions of the uncertain parameters are available. When such data is not available, we do not have enough information for inferring or deriving the probabilistic models. In such situations, we have to resort to an alternative treatment of uncertainty. For example, the modeler may be able to approximate the duration of the tasks and specify the longest and shortest durations or the interval in which the duration belongs at different levels of confidence.

In this paper, we draw upon concepts from fuzzy set theory to describe the imprecision and uncertainties in the durations of processing tasks. Indeed, this approach has been receiving increasing attention recently. McCahon and Lee (1992)¹¹ were the first to illustrate the application of fuzzy set theory as a means of analyzing performance characteristics for a flow-shop system. They modified the Campbell, Dudek and Smith (1970)⁴ sequencing heuristic for the case of fuzzy processing times. A mathematical programming approach to a single-machine scheduling problem with fuzzy precedence relation⁹ and job shop scheduling
with fuzzy processing time using GAs\cite{17} have been proposed. Balasubramanian and Gtosnann (2003)\cite{21} reviewed some researches about flow shop scheduling optimization under uncertainty.

While the present studies about earliness and tardiness (E/T) scheduling problem under uncertainty were few and deficient. As a kind of unconventional scheduling problems, the earliness and tardiness performance function is not a monotony increasing relation on completion time of every job. Both job sequence and starting time of every operation must be optimized, which increases the difficulties of solving this kind of problem. In order to achieve better schedulers for E/T problems based on the present corresponding results on fuzzy optimization and scheduling methods, this paper investigated the m-machine fuzzy flow-shop scheduling problem (FFSSP) with different due-dates and fuzzy processing times. Performance was measured by the minimization of the weighed sum of E/T penalties of jobs.

It has been testified that earliness and tardiness scheduling problem with different due-date is an NP-complete problem\cite{23}. In most of recent E/T literature, the single-machine E/T scheduling problems are mainly investigated\cite{5}\cite{10}\cite{16}. Baker and Sudder (1990)\cite{16} published a comprehensive state-of-the-art review for different versions of the E/T scheduling problems. Unfortunately, resolving results and conclusions on multi-machine E/T problems are few, and which mostly focus on parallel-machine E/T scheduling problems. For currently m-machine E/T scheduling problems, only searching algorithms can be adopted. Wu and Wang (2000)\cite{20} investigated the fuzzy flow-shop scheduling problem with E/T performance index by introducing GA, yet the result was not very good, because only a better job sequencing was found other than optimizing both job sequencing and starting time of every job. While for unconventional scheduling problems, both job sequence and starting time of every operation must be optimized, which increases the solving difficulties.

In this paper, we advanced a jointed algorithm for solving flow-shop earliness / tardiness problem based on genetic algorithm and heuristic algorithm. Hierarchy scheduling paradigm was introduced. First, the genetic algorithm is used to determine preferably scheduling sequence. Second, a kind of heuristic algorithm was put forward to adjust the starting time for the present scheduling sequence. Where, the heuristic algorithm determined what time and how long every idle time should be. The object of this paper is to minimize the total earliness and tardiness costs of all the jobs. As illustrative numerical examples, we consider 6 × 6 fuzzy flow shop scheduling problems with the object of minimizing total earliness and tardiness penalties costs, and demonstrate the feasibility and efficiencies of the proposed method by comparing with the GA method.

The paper is organized as follows. Section 2 defines the problems of interest. Section 3 presents a brief overview of fuzzy set theory and compares mathematical operations in the fuzzy framework. After an overview of our approach to the problem in Section 4, we present results from optimization of flow-shops in Section 5. Finally, the conclusions on this paper are drawn.

2. Problem statement

We consider in this paper the following general problem. Given are a set of tasks to be performed using certain resources (processing equipment), for e.g. these tasks could be those associated with the processing of a set of orders in a flow-shop. These tasks have their processing times specified as fuzzy numbers. The objective is to determine an allocation of the resources to the tasks that is optimal with respect to minimize the total earliness and tardiness costs.

Flow-shop plants are multi-product batch plants where the jobs associated with the manufacturing of product orders use the same set of processing units in the same order\cite{3}. A solution to the flow-shop scheduling problem specifies the order in which jobs are processed in each unit.

Given are n products (J_1, J_2, ..., J_n), i=1,2,...,n, that are to be manufactured in a flow-shop plant with m processing machines (M_1, M_2, ..., M_m), j=1,2,...,m. Each product requires processing in all of the m machines and flows the same sequence of operation, i.e. a permutation flow-shop plant.

The m-machine flow-shop scheduling problem with fuzzy processing times is considered in this paper. The n jobs to be processed are ready at time zero, for which due dates are known but processing times are fuzzy. The notations used in the paper are defined below.

n: number of jobs
m: number of machines
i: job number
k: position pointer
j: machine pointer

- the processing time of job i on machine j, \( P_{i,j} \) denoted as (\( P_{i,j}^1 \), \( P_{i,j}^2 \), \( P_{i,j}^3 \))

- the starting time of the job in position k \( B_{k,j} \) on machine j, i=1,2,...,n, j=1,2,...,m, fuzzy numbers

- the completion time of job i on machine j, \( C_{i,j} \) i=1,2,...,n, j=1,2,...,m, fuzzy numbers
\( \alpha_i \): earliness penalty power of job \( i \), \( i=1,2,\ldots,n \)

\( \beta_i \): tardiness penalty power of job \( i \), \( i=1,2,\ldots,n \)

\( D_i \): due-date of job \( i \), \( i=1,2,\ldots,n \)

In the case of deterministic flow-shop scheduling problem, the computation of the object (total earliness and tardiness costs) involves a series of recurrence relations for the earliness and tardiness cost \( ET_i \) (\( i=1,2,\ldots,n \)), as well as for the completion times \( C_{ij} \) (\( i=1,2,\ldots,n \), \( j=1,2,\ldots,m \)). The relations for calculating earliness and tardiness costs for the deterministic case are given below:

\[
B_{1,j} = (0, 0, 0) \quad j = 2,3,\ldots,m
\]

\[
B_{k,j} = B_{k-1,j+1} + P_{k,j-1} \quad k = 2,3,\ldots,n \quad j = 2,3,\ldots,m
\]

\[
B_{k,j} = \max\{B_{k-1,j} + P_{k,j} - P_{k-1,j} \} \quad k = 2,3,\ldots,n \quad j = 2,3,\ldots,m
\]

\[
C_{k,j} = B_{k,j} + P_{k,j} \quad k = 2,3,\ldots,n \quad j = 2,3,\ldots,m
\]

\[
ET_i = \alpha_i \max\{0, D_i - C_{i,m}\} + \beta_i \max\{C_{i,m} - D_i, 0\}
\]

\[
ET = \sum_{i=1}^{n} \{ET_i\}
\]

Where, \( B_{1,1} \) is a deterministic number in fact, for describing convenience, we denoted it as a fuzzy number \((0, 0, 0)\); the subscript \( i \) denote the job which is in position \( k \) in the schedule sequence; \( ET \) is the objective function needed to be optimized.

3. Operations on fuzzy numbers

In this section, we review key concepts from the theory of fuzzy sets that will be used for the scheduling models\superscript{21,17}.

3.1. Definitions

A classical set \( S \) of a universe \( X \) is a collection of elements of objects that are well defined and possess some common properties. An element \( x \) of the universe \( X \) may either belong to (true or 1) \( S \) or not (false or 0).

Zadeh (1965)\superscript{21} introduced the concept of a fuzzy set in which the membership of an element to a set need not be just binary-valued (i.e. 0-1), but could be any value over the interval \([0, 1]\) depending on the degree to which the element belongs to the set. The higher the degree of belonging, higher the membership value.

A fuzzy set \( \tilde{A} \) of the universe \( X \) is specified by a membership function \( \mu_{\tilde{A}} (x) \), which takes its value in the interval \([0, 1]\). For each element \( x \) of \( X \), the quantity \( \mu_{\tilde{A}} (x) \) specifies the degree to which \( x \) belongs in \( \tilde{A} \).

Thus, \( \tilde{A} \) is completely characterized by the set of ordered pairs:

\[
\tilde{A} = \{(x, \mu_{\tilde{A}} (x)) \mid x \in X\}
\]

The support of a fuzzy set \( \tilde{A} \) is a (crisp/classical) subset of \( X \) given by

\[
\text{sup}(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}} (x) > 0\}
\]

The \( \alpha \)-level set (\( \alpha \)-cut) of a fuzzy set \( \tilde{A} \) is crisp subset of \( X \) given by

\[
\tilde{A}_\alpha = \{x \in X \mid \mu_{\tilde{A}} (x) \geq \alpha\} \quad \forall \alpha \in (0, 1]
\]

A fuzzy set is said to be convex if all its \( \alpha \)-level set are convex. A fuzzy number is a convex normalized fuzzy set with piecewise-continuous membership function\superscript{23}. Different functional forms can be used for modeling the membership functions of fuzzy numbers. For convenience in our subsequent discussion, consider the operations on the fuzzy processing times represented as triangular fuzzy numbers (TFN).

3.2. Arithmetic operations

Arithmetic operations on fuzzy numbers that will be used for the objective function and the constraints, such as addition, subtraction, taking the maximum of two fuzzy numbers are defined through the extension principle\superscript{21}, which allows the extension of operations on real numbers to fuzzy numbers. In the scheduling problems of interest to us, the principal arithmetic operations that are involved are addition (computing the fuzzy completion time of a task given the fuzzy start time and duration) and maximization (computing the start time of a task as the maximum of the completion times of preceding tasks, and the earliness and tardiness costs). Hence, we restrict our description of arithmetic operations on fuzzy numbers to these two operations.

Denote a triangular fuzzy number \( \tilde{A} \) by a triplet \((\alpha^1, \alpha^2, \alpha^3)\). Then, as is well known, the addition of two
triangular fuzzy numbers \( \tilde{A} = (a^1, a^2, a^3) \) and \( \tilde{B} = (b^1, b^2, b^3) \) is shown by the flowing formula(Sakawa and Kubota, 2000):

\[
\tilde{A} + \tilde{B} = (a^1 + b^1, a^2 + b^2, a^3 + b^3)
\]

It can be shown that adding two TFN results in another TFN. Furthermore, in order to add two TFN, it is sufficient to perform the computations in Eq. (1) at \( \alpha = \varepsilon \rightarrow 0 \) and \( \alpha = 1 \) levels, when dealing with TFN. The addition operation is used when computing the fuzzy completion time of a task given the fuzzy start time and duration.

Furthermore, denote the membership functions of two triangular fuzzy numbers \( \tilde{A} = (a^1, a^2, a^3) \) and \( \tilde{B} = (b^1, b^2, b^3) \) by \( \mu_A \) and \( \mu_B \), respectively, then according to the extension principle of Zadeh, the membership function \( \mu_{\tilde{A} \vee \tilde{B}}(z) \) of \( \tilde{A} \vee \tilde{B} \) through the \( \vee \) (max) operation becomes as follows:

\[
\mu_{\tilde{A} \vee \tilde{B}}(z) = \sup_{x,y} \min(\mu_A(x), \mu_B(y))
\]

Unfortunately however, since the fuzzy number obtained as a result of the \( \vee \) (max) operation through the extension principle does not always become a triangular fuzzy number, in this paper, for simplicity, we approximate the \( \vee \) (max) operation with the following formula[17]:

\[
\tilde{A} \vee \tilde{B} = (a^1 \vee b^1, a^2 \vee b^2, a^3 \vee b^3)
\]

### 3.3. Objective function comparisons

Since we are interested in selecting a schedule with the optimum (minimum) E/T cost, we have to compare the fuzzy E/T costs of potential schedules.

There exists a large body of literature that deals with the comparison of fuzzy numbers[14] and a number of indices have been proposed. In this paper, we adopt the following comparison method for TFN[9][17][18]. In this comparison method, the criterion for dominance is one of the following three in the order given below.

**Criterion 1.** The greatest associate ordinary number

\[
ET_1(A) = \frac{a^1 + 2a^2 + a^3}{4}
\]

is used as a first criterion for comparison the two TFNs.

**Criterion 2.** If \( ET_1 \) does not comparison the two TFNs, those which have the best maximal presumption

\[
ET_2(A) = a^3 - a^1
\]

will be chosen as a second criterion.

**Criterion 2.** If \( ET_1 \) and \( ET_2 \) do not comparison the TFNs, the difference of the spreads

\[
ET_3(A) = a^3 - a^1
\]

will be used as a third criterion.

According to these three criteria, it becomes possible to rank all TFNs. Among TFNs \( \tilde{A}_i \) \((i=1,2,\ldots,n)\), the maximum and minimum TFNs are respectively denoted by \( \tilde{A}_{\text{max}} \) and \( \tilde{A}_{\text{min}} \).

### 4. Jointed algorithm based on GA and heuristic algorithm

#### 4.1. Genetic algorithm for sequencing

It has been testified that earliness and tardiness scheduling problem with different due-date is an NP-complete problem. In most of recent E/T literature, the single-machine E/T scheduling problems are mainly investigated. Unfortunately, resolving results and conclusions on multi-machine E/T problems are few, and which mostly focus on parallel-machine E/T scheduling problems. In this paper, we advanced a jointed algorithm for solving flow-shop E/T problem based on genetic algorithm and heuristic algorithm proposed in this paper. Where the genetic algorithm is used to determine preferably scheduling sequence and can be described as following:

1. **Chromosome representation**
   - Natural number coding is adopted. For example: 3-1-2, where, figures denote job number and the first job in ranking will be processed first.

2. **Compute fitness**
   - The objective function E/T costs, the function to be optimized, provide the mechanism for evaluating each chromosome at each iteration of the GA search. And the value of the objective function is the fitness.

3. **Genetic operators**
   - Tournament selection, order crossover and SWAP mutation methods are adopted respectively.

4. **Generate next population**
   - Let the best chromosome of all parents and offspring be in next population directly, and others are random selected from parents and offspring. This method can improve convergence speed as well as not lost population’s diversity.

5. **Terminating condition**
   - Maximum generation is adopted as terminating condition and the maximum iterative generation number is \( 3 \times n \times m \).
4.2. The proposed heuristic algorithm for starting time optimization

4.2.1. The proposed heuristic algorithm (HA). It is known that only a better job sequence can be found by using GA other than optimizing both the sequence and the starting time of every operation, so the results are not good for E/T scheduling problems. Therefore, a heuristic algorithm (HA) was proposed in this paper to optimize the starting times. Furthermore, based on the proposed HA and GA, a jointed algorithm was constructed.

In the HA, for the moment, we only optimized the earliness jobs. Based on the results (better job sequencing) of GA, The proposed HA is to adjust the starting time of every operation in order to minimizing the earliness penalties costs or the E/T performance index. For describing the HA particularly, some denoting are defined below.

$O_{i,j}$: the operation of job $i$ on machine $j$

$IT_{i,j}$: the allowed maximum idle time before operation $O_{i,j}$

$IT_{i,max}$: the allowed maximum idle time inserted into the operations of job $i$

$i_l$: The job number at position $l$ in earliness jobs set

$k$: The position number of each job in the whole job sequence

$X_{i,k}$: 1, if $i=k$; 0, elsewhere

The proposed HA can be described as following:

step1 to examine and note all the earliness jobs, and rank them at increasing sequence according to completing times. The earliness job sequence set can be marked as $\varphi = \{Job_{i_1}, Job_{i_2}, ..., Job_{i_l}, ... Job_{i_n} \}$, where, $Job_{i_l}$ denotes a job at the position of $l$ in earliness jobs set. Let $L$ memorize the number of the elements in set $\varphi$. If $\varphi = \Phi$, turn to step5.

step2 computing the allowed maximum inserting idle time of every job.

$\widetilde{IT}_{i_{l},max} = D_{i_{l}} - \widetilde{C}_{i_{l},m}$

For $l=1,2,...,L-1,$

$\widetilde{IT}_{i_{l-1},max} = \min\{\sum_{k=1}^{n} \tilde{B}_{k,l,i,k}X_{i_{l-1},k}, D_{i_{l-1}}\} - \widetilde{C}_{i_{l-1},m}$

step3 inserting idle times.

If $\exists j, i=1,2,...,n,$ make $\tilde{I}_{i_{l},max}=(0,0,0)$, then there is no need to insert any idle time in the operations of job $i$ through whole processes.

For the jobs of $\tilde{I}_{i_{l},max}$ $(0,0,0)$, $i=1,2,...,n$, we adjusted respectively its completion times and starting times of every operation as flowing:

The updating function of completion times is:

$\widetilde{C}_{i_{l},m} = \widetilde{C}_{i_{l},m} + \tilde{I}_{i_{l},max}, i=1,2,...,n$

The updating function of starting times of jobs on machine $m$ is

$\sum_{k=1}^{n} \tilde{B}_{k,m}X_{i_{l},k} = \sum_{k=1}^{n} \tilde{B}_{k,m}X_{i_{l},k} + \sum_{k=1}^{n} \tilde{I}_{i_{l},max} \tilde{X}_{i_{l},k}$

Where, $k$ denotes a position in the whole job sequence; $l$ denotes a position in the earliness jobs set

$\varphi : \sum_{k=1}^{n} \tilde{B}_{k,m}X_{i_{l},k}$ denotes the starting time of job $i_l$ on machine $m$.

The updating functions of starting times of jobs on machine $j$ $(j=1,2,...,m)$ are

For $j=m-1, m-2,...,1$

$\tilde{B}_{n,j}X_{i_{l},n} = \tilde{B}_{n,j+1}X_{i_{l},n} - \tilde{P}_{b,j}X_{i_{l},n}$

$\sum_{k=1}^{n} \tilde{B}_{k,j}X_{i_{l},k} = \min\{\sum_{k=1}^{n} \tilde{B}_{k,j+1}X_{i_{l},k}\}$

$\sum_{k=1}^{n} \tilde{B}_{k+1,j}X_{i_{l},k} - \sum_{k=1}^{n-1} \tilde{P}_{b,j}X_{i_{l},k}$

End For

step4 calculating the value of the E/T performance index.

$f = \sum_{i=1}^{n} \left\{ \alpha \max\{0, \tilde{D}_i - \tilde{\widetilde{C}_{i,m}}\} \right\} + \beta \max\{\tilde{C}_{i,m} - \tilde{D}_i, 0\}$

step5 end.

4.3. The jointed algorithm (JA)

In this section, we constructed the jointed algorithm for solving flow-shop earliness / tardiness problem based on genetic algorithm and heuristic algorithm. Hierarchy scheduling paradigm was introduced. First, the genetic algorithm is used to determine preferably scheduling sequence. Second, a kind of new heuristic algorithm was put forward to adjust the starting time of the present schedule. Where, the heuristic algorithm determined what time and how long every idle time should be. The jointed algorithms are described in the following paper.

step1 to generate the initial generation, and let $g=0$;

step2 to check the terminating condition, if it is satisfied, the algorithm should output the optimized results and the program is end.
step 3 to implement the genetic operations including tournament selection, order crossover and SWAP mutation strategies.  

step 4 to evaluate the generation and select the best individual, then which will be optimized furthermore by adopting the proposed HA in this paper.  

step 5 to generate the next generation, and let \( g = g + 1 \), turn to step 2.

5. Numerical examples and simulation results

Now we are ready to apply both the JA proposed in this paper and the GA presented in the previous section to FFSSP. As illustrative numerical examples, consider \( 6 \times 6 \) FFSSP problem. The datum of \( 6 \times 6 \) FFSSP are shown in Table 1 (from reference [17]). The due-dates involved in these numerical examples are generated in the following way. Taking the value \( P_{i,j}^{2} \) (\( i=1,2,\ldots,n; \ j=1,2,\ldots,m \)) of the fuzzy processing time for each operation as standard time, calculate the sum of the standard times for each job, and the resulting sum is multiplied by some appropriate value \( K \) for determining each \( D_{i} \) (\( i=1,2,\ldots,n \)). The earliness/tardiness punished powers are generated randomly on the interval \([1, 3]\) and \([3, 5]\) respectively.

The proposed algorithms have been coded with VC++ and run on Pentium 4 PC with a 1.8 Ghz processor under Windows 2000.

Each of the parameters values of GA is shown in Table 2, where \( N, P_{c}, P_{m}, M \), respectively, denote the population size, the crossover rate and mutation rate and the number of maximum generation.

The results obtained through the JA and GA are shown in Table 3-5 (\( 6 \times 6 \) FFSSP) with different \( K \).

| Table 1. Fuzzy processing times and due-dates of \( 6 \times 6 \) FFSSP |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Job             | \( M_{1} \)     | \( M_{2} \)     | \( M_{3} \)     | \( M_{4} \)     | \( M_{5} \)     | \( M_{6} \)     | Due-date        |
| \( K=1.0 \)     |                 |                 |                 |                 |                 |                 | \( K=1.5 \)     |
| \( K=1.8 \)     |                 |                 |                 |                 |                 |                 | \( K=1.8 \)     |
| \( J_{1} \)     | (9,13,17)       | (6,9,12)        | (10,11,13)      | (5,8,11)        | (10,14,17)      | (9,11,15)       | 66              |
|                 |                 |                 |                 |                 |                 |                 | 99              |
|                 |                 |                 |                 |                 |                 |                 | 118.8           |
| \( J_{2} \)     | (5,8,9)         | (7,8,10)        | (3,4,5)         | (3,5,6)         | (10,14,17)      | (4,7,10)        | 46              |
|                 |                 |                 |                 |                 |                 |                 | 69              |
|                 |                 |                 |                 |                 |                 |                 | 82.8            |
| \( J_{3} \)     | (3,5,6)         | (3,4,5)         | (2,4,6)         | (5,8,11)        | (3,5,6)         | (1,3,4)         | 29              |
|                 |                 |                 |                 |                 |                 |                 | 43.5            |
|                 |                 |                 |                 |                 |                 |                 | 52.2            |
| \( J_{4} \)     | (8,11,14)       | (5,8,10)        | (9,13,17)       | (8,12,13)       | (10,12,13)      | (3,5,7)         | 61              |
|                 |                 |                 |                 |                 |                 |                 | 91.5            |
|                 |                 |                 |                 |                 |                 |                 | 109.8           |
| \( J_{5} \)     | (8,12,13)       | (6,9,11)        | (10,13,17)      | (4,6,8)         | (3,5,7)         | (4,7,9)         | 52              |
|                 |                 |                 |                 |                 |                 |                 | 78              |
|                 |                 |                 |                 |                 |                 |                 | 93.6            |
| \( J_{6} \)     | (8,10,13)       | (8,9,10)        | (6,9,12)        | (1,3,4)         | (3,4,5)         | (2,4,6)         | 39              |
|                 |                 |                 |                 |                 |                 |                 | 58.5            |
|                 |                 |                 |                 |                 |                 |                 | 70.2            |

| Table 2. Parameters values of GA |
|-----------------|-----------------|-----------------|-----------------|
| Problem         | \( N \)         | \( P_{c} \)     | \( P_{m} \)     |
| \( 6 \times 6 \) | 50              | 0.6             | 0.1             |
| \( 10 \times 10 \) | 80              | 0.6             | 0.1             |

| Table 3. Simulation results for \( 6 \times 6 \) FFSSP (\( K=1.0 \)) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Job             | \( C \)         | 112.2           | 50              | 28.2            | 82.5            | 89.2            | 54              |
|                 | \( f_{1} \)     | 190.88          | 18.47           | 1.62            | 103.02          | 167.37          | 70.76           |
|                 |                 |                 |                 |                 |                 |                 | 552.1           |
| \( J_{2} \)     | \( C \)         | 112.2           | 50              | 29              | 82.5            | 89.2            | 54              |
|                 | \( f_{2} \)     | 190.88          | 18.47           | 0               | 103.02          | 167.37          | 70.76           |
|                 |                 |                 |                 |                 |                 |                 | 550.5           |

Where, GA: genetic algorithm; JA: the proposed jointed algorithm based on GA and HA;  
\( C \): the completion time of every job. Based on the first criterion in section 3.3, let \( C=(C^{1}+2 \times C^{2}+C^{3})/4 \), which is used to compare the results more convenience;  
\( f_{1} \) (\( f_{2} \)): the E/T penalties costs by using GA (JA) of each job;  
Object: The total E/T penalty costs of all jobs;  
\( f(\%)=(f_{1}−f_{2})/f_{1} \);  
The optimal job sequence achieved is: 3, 2, 6, 4, 5, 1;
The earliness job is only job 3.

Table 4. Simulation results for $6 \times 6$ FFSSP ($K=1.5$)

<table>
<thead>
<tr>
<th>Job</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>C</td>
<td>101</td>
<td>65.75</td>
<td>47.75</td>
<td>108.75</td>
<td>74.25</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>f1</td>
<td>8.25</td>
<td>4.50</td>
<td>16.82</td>
<td>82.66</td>
<td>9.92</td>
<td>26.29</td>
</tr>
<tr>
<td>JA</td>
<td>C</td>
<td>101</td>
<td>67.5</td>
<td>47.75</td>
<td>108.75</td>
<td>78</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>f2</td>
<td>8.25</td>
<td>2.07</td>
<td>16.82</td>
<td>82.66</td>
<td>0</td>
<td>18.20</td>
</tr>
</tbody>
</table>

Where, the optimal job sequence achieved is: 6, 3, 2, 5, 1, 4;
The earliness jobs are: 6, 2, 5.

Table 5. Simulation results for $6 \times 6$ FFSSP ($K=1.8$)

<table>
<thead>
<tr>
<th>Job</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>C</td>
<td>89.25</td>
<td>65.75</td>
<td>47.75</td>
<td>110.75</td>
<td>96</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>f1</td>
<td>29.62</td>
<td>23.64</td>
<td>9.65</td>
<td>4.55</td>
<td>10.78</td>
<td>42.06</td>
</tr>
<tr>
<td>JA</td>
<td>C</td>
<td>118.8</td>
<td>77.75</td>
<td>52.2</td>
<td>110.75</td>
<td>96</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>f2</td>
<td>0</td>
<td>7.00</td>
<td>0</td>
<td>4.55</td>
<td>10.78</td>
<td>33.97</td>
</tr>
</tbody>
</table>

Where, the optimal job sequence achieved is: 6, 3, 2, 1, 5, 4;
The earliness jobs are: 6, 2, 1.

From Table 3-5, some conclusions are gained, they are as following:

1) It can be seen from Table 3-5 that the advanced jointed algorithm in this paper has decreased the E/T penalties costs in a great degree. This indicated that better schedule could be found form JA than GA. And we can find that the efficiencies of the JA is better along with bigger value of $K$, such as, in the simulation of $6 \times 6$ FFSSP, the efficiencies is 0.29% ($K=1.0$), 13.77% ($K=1.5$), 53.19% ($K=1.8$).

2) The JA proposed in this paper will have good performance in this kind of production with excessive earliness jobs. And for the scheduling problem with higher earliness penalties costs, the JA should have better efficiencies, which can be obtained by compare the results in Table 3-5.

3) For the tardiness jobs, it can be seen from Table 3-5 that tardiness penalties costs after adopting JA were not less than the results of GA. This is because the proposed JA only optimized the operations’ starting times of earliness jobs. Although the JA have not optimized the tardiness jobs more, the efficiencies here is still better than GA obviously. So, there are great development spaces to improve the proposed JA and arrive more surprising effects.

6. Conclusions

As a kind of unconventional scheduling problem, fuzzy flow shop scheduling problem with the object of earliness/tardiness penalties costs has been investigated in this paper. In E/T scheduling problems, considering that both job sequence and starting time of every operation must be optimized, which increases the difficulties of solving this kind of problem, we proposed a heuristic algorithm to optimize the starting time of every operation. Furthermore, a jointed algorithm for solving flow-shop E/T problem has been constructed by jointing the genetic algorithm and the proposed heuristic algorithm. The jointed algorithm could find a better job sequence as well as optimize the starting time of every operation, so as to minimize the E/T penalties costs in a great degree. Through a lot of simulations for $6 \times 6$ FFSSP, it was demonstrated that, compared with GA, the proposed jointed algorithm gave better results in all trials. Extensions of the proposed method to more general FFSSP with E/T performance index as well as to interactive methods are now under investigation and will be reported elsewhere in the near future.

7. Acknowledgements

The authors wish to thank the anonymous refers for their useful suggestions and comments that improved this paper. This work is supported by the National Natural Science Foundation, China (No. 70431003),
the National High-Tech. R&D Program for CIMS, China (No. 2003AA412210) and the National Basic Research Program of China, China (No. 2002CB312204—03).

References


Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)