APEM journal

Advances in Production Engineering & Management
Volume 14 | Number 3 | September 2019 | pp 367–378
https://doi.org/10.14743/apem2019.3.334

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

An integrated system for scheduling of processing and assembly operations with fuzzy operation time and fuzzy delivery time

Yang, M.S.[a,*], Ba, L.[a,*], Zheng, H.Y.[a], Liu, Y.[a], Wang, X.F.[a], He, J.Z.[a], Li, Y.[a]

[a]School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, P.R. China

A B S T R A C T

This paper integrates the processing scheduling with assembly scheduling, aiming to satisfy the requirements for just-in-time (JIT) production. Considering the uncertainty of time factors in actual production, the operation time of the jobs were represented as triangular fuzzy numbers and the delivery time of the final product as trapezoidal fuzzy numbers. An extended job-shop scheduling problem (JSP) considering above factors was proposed in this paper. A mathematical model was established for processing and assembly scheduling, aiming to achieve the mean satisfaction degree on delivery time. In light of the complexity of the problem, a genetic algorithm (GA) was designed to realize the fuzzy integrated optimization of processing and assembly under uncertainty. The proposed algorithm includes selection, crossover, mutation operations, and reflects the spirits of two-section real number encoding and elite protection strategy. Each part of the GA was designed in detail. Finally, the proposed model and algorithm were verified through a case study on processing and assembly scheduling. The model enjoys high practical value by taking the customer satisfaction of the delivery period as the main goal. The results show that our scheduling strategy mirrors the actual production situation and provides a good reference for JSP scheduling under multiple uncertainties. The best solution obtained by our model is more feasible than basic JSP in real production environment.

A R T I C L E I N F O

Keywords:
Integrated scheduling;
Uncertainty;
Fuzzy operation time;
Fuzzy delivery time;
Genetic algorithm (GA)

**Corresponding author:*
yangmingshun@xaut.edu.cn
(Yang, M.S.)
xautbali@163.com
(Ba, L.)

Article history:
Received 18 April 2019
Revised 8 September 2019
Accepted 12 September 2019

1. Introduction

Job-shop scheduling problem (JSP) is a much-concerned NP-hard problem in the field of combinatorial optimization [1]. The basic JSP aims to minimize the makespan of several jobs under the following constraints: within each job there is a set of operations which need to be processed in a specific order; Each operation has a specific machine that it needs to be processed on [2, 3]. Compared with single-machine scheduling and flow job scheduling, the JSP can satisfy the growing demand for customized products and the increasingly diversified, small-batch production mode.

Nevertheless, the basic JSP, taking the minimal makespan as the goal, still deviates from the actual production situation, in that it fails to consider the close correlation between the production plan and assembly plan or take account of the delivery time. The production plan focuses on machine control and job-shop scheduling, while the assembly plan optimizes the production line resources. The delivery time is the key to just-in-time production and an important influencing factor of scheduling effect. A proper delivery time helps to rationalize production planning and scheduling, enabling the manufacturer to respond to market changes quickly at a low production cost.

Considering the effect of uncertainty factors on the scheduling results, this paper introduces several fuzzy parameters, namely fuzzy operation time, fuzzy assembly time and fuzzy delivery time, into the basic JSP, and attempts to realize integrated scheduling of processing and assembly with uncertain factors.

2. Literature review

Over the years, much research has been done on the JSP, yielding fruitful results. Below is a brief review of the related studies in recent years.

On algorithm improvement, Nirmala Sharma *et al.* [2] designed an improved bee colony algorithm (BCA), and verified its superiority by a number of standard examples. Piroozfard *et al.* [4] proposed an improved biogeographic optimization algorithm, and tested that it outperformed traditional algorithms like greedy randomized adaptive search procedure (GRASP), phase gradient autofocus (PGA) and hierarchical genetic algorithm (HGA). Akram *et al.* [5] developed a fast simulated annealing (SA) algorithm for the basic JSP, which can avoid the local optimum trap. An improved cuckoo search algorithm (CSA) was proposed for solving JSP by Hu *et al.* [6]. Modrák *et al.* [7] proposed a novel genetic algorithm within heuristics for solving flow shop scheduling problem.

Considering the diverse job types and processing routes in actual production, some scholars have incorporated multiple processing routes into the basic JSP, forming the flexible JSP (FJSP). For instance, Lin [8] derived a super heuristic algorithm based on backtracking algorithm, and used it to simulate the FJSP with fuzzy operation time. Shen *et al.* [9] described an improved tabu search algorithm that adjusts the operation time in FJSP according to the operation sequence. Kato *et al.* [10] combined hill climbing and particle swarm optimization (PSO) into a hybrid algorithm, and verified the effectiveness of this algorithm using standard examples. Wu [11] generated a new mathematical model through a non-dominated sorting genetic algorithm (NSGA), aiming to minimize the makespan and energy consumption of the FJSP by heuristic rules X. P. Yang *et al.* [12]proposed an dynamic multi-objective FJSP and design a parallel hybrid algorithm. A model for FJSP involving low carbon factor was established by Seng *et al.* [13]. An improved NSGA-II was proposed to solve the problem. Nidhiry *et al.* [14] proposed a modified NSGA-II for solving a multi-objective FJSP. Xu *et al.* [15] proposed a bat algorithm for solving a dual FJSP.

In addition, many scholars have introduced practical factors into the JSP to improve the applicability of the problem in actual production. For example, Nagata *et al.* [1] established a mathematical model with the goal of minimizing the makespan, and propounded a dynamic programming algorithm that adds the bottleneck process to the basic JSP. Bierwirth *et al.* [3] tendered an extended GRASP algorithm to minimize the total delay. Yazdani *et al.* [16] de-signed an improved competition algorithm and relied on it to construct a JSP model that minimizes the advance/tardiness penalty of delivery time. Chaouch *et al.* [17] solved cross-plant transportation JSP by an improved ant colony algorithm (ACA). Kurdi *et al.* [18] proposed an improved cultural genetic algorithm to minimize the makespan, total tardiness penalty and total advance penalty of the JSP. Shahrabi *et al.* [19] delineated the JSP with job insertion and machine failure. Kundakci [20] explored the JSP with random insertion, ma-chine failure and variable working hours. Kuhpfahl *et al.* [21] illustrated an improved local search algorithm and utilized its model to minimize the total tardiness penalty in the JSP problem with a specified delivery date. Zhong *et al.* [22] considered manpower and machine into basic JSP. A model for two-resource JSP was established. A branch population genetic algorithm was designed for solving the problem. A lean scheduling problem in job shop environment were studied by Haider *et al.* [23]. Chaudhry *et al.* [24] proposed an integrated scheduling problem within process planning and designed a genetic algorithm. Besides, there exist many uncertain factors in manufacturing [6], economic [25, 26], environment [27] and so on.

To sum up, more and more production factors have been introduced to the JSP, including multiple processing routes, delivery time, bottleneck process and machine failure, with the aim to make the JSP more in line with the actual production situation. The actual production is affected by various random factors, making it hard to obtain the exact operation and assembly time of each job. Due to the fuzziness in the operation and assembly time, the makespan becomes uncertain and the delivery time may fluctuate.

In light of the above, this paper takes account of uncertain time factors in the integrated scheduling of processing and assembly. These factors were expressed as fuzzy numbers. Next, the JSP was modelled with the goal to satisfy the delivery time, and solved by a genetic algorithm (GA). Finally, the model and the algorithm were verified through a case study.

3. Problem description

Our research problem is about the integrated scheduling of processing and assembly, considering the uncertainties in the operation time of each job on each machine in the production line, the assembly time in the assembly machines and the delivery time of the final product. The mathematical description of the problem is as follows.

Let n be the number of jobs in the final product, each of which has a unique processing route, p be the number of processing machines, and m be the number of assembly machines. It is assumed that each processing or assembly operation corresponds to one machine only. The start and finish time of the processing machines and assembly operation were expressed as triangular fuzzy numbers, while the delivery time was depicted by a semi-trapezoidal fuzzy number. Then, the processing and assembly operations of job were sorted to determine the sequence that best suits the delivery time. The following symbols were defined to establish a fuzzy scheduling model for the integrated scheduling.

n	The number of jobs in the final product
ℓ_i	The serial number of job i
m	The number of assembly machines
p	The number of processing machines
q_i	The number of processing operations for type i jobs
M	A random large positive integer
$\tilde{E}_{M(i,k)}$	Makespan of job i on processing machine k
$\tilde{E}_{A(j)}$	Makespan of assembly machine j
O_{ijk}	A Boolean variable indicating whether the j-th operation of job i is processed on machine k; if yes, the value of the variable is one; otherwise, the value is zero.
C_i	The assembly time of job i on an assembly machine
$\tilde{T}_{M(i,k)}$	The operation time of job i on processing machine k
$\tilde{T}_{A(j)}$	The assembly time of assembly machine j
$\tilde{S}_{M(i,1)}$	The start time of the processing task of job i on the first machine
$\tilde{S}_{A(j)}$	The start time of assembly task on assembly machine j
x_{ihk}	A Boolean variable indicating the operation sequence of job i on machine h and machine k
y_{ijk}	A Boolean variable indicating the operation sequence of jobs i and j on machine k
$R_{A(j)}$	A Boolean variable indicating whether job i is required for assembly machine j
$\tilde{E}^{i}_{M(j+1)}$	The makespan of job i required for assembly machine j+1
$\tilde{E}^{max}_{M(j+1)}$	The maximum makespan of all jobs required for assembly machine j+1

4. Model establishment

The fuzzy optimization model for the minimal total makespan of product assembly can be established as:

$$\tilde{Z} = min\{\widetilde{max}\, E_{A(j)}\} = min\{max(E^L_{A(j)}, E^M_{A(j)}, E^U_{A(j)})\} \tag{1}$$

The model is subjected to the following constraints:

$$\tilde{E}_{A(j)} = \tilde{E}_{M(j)}^{max_{A(j)}}, \quad (j = 1) \tag{2}$$

$$\tilde{Z} = min\{\widetilde{max}\, E_{A(j)}\} = min\{max(E_{A(j)}^{L}, E_{A(j)}^{M}, E_{A(j)}^{U})\} \tag{3}$$

$$\tilde{E}_{M(i,k)} = \sum_{j=1}^{q_i} O_{ijk} \times \tilde{T}_{M(i,k)} \times \ell_i, \quad (i = 1;\ k = 1) \tag{4}$$

$$\tilde{E}_{M(i,k)} = \tilde{E}_{M(i-1,k)} + \sum_{j=1}^{q_i} O_{ijk} \times \tilde{T}_{M(i,k)} \times \ell_i, (i = 2,3, \dots, n;\ k = 1) \tag{5}$$

$$\tilde{E}_{M(i,k)} = \max[\tilde{E}_{M(i-1,k)}, \tilde{E}_{M(i,k-1)}] + \sum_{j=1}^{q_i} O_{ijk} \times \tilde{T}_{M(i,k)} \times \ell_i\,, (i = 2,3, \dots, n;\ k = 2,3, \dots, p) \tag{6}$$

$$\tilde{E}_{M(i,k)} - \sum_{j=1}^{q_i} O_{ijk} \times \tilde{T}_{M(i,k)} \times \ell_i + M(1 - x_{ihk}) \geq \tilde{E}_{M(i,h)} \tag{7}$$

$$\sum_{k=1}^{p} O_{ijk} = 1, \quad \forall i, j \tag{8}$$

$$\sum_{j=1}^{q_i} O_{ijk} = 1, \quad \forall i, k \tag{9}$$

$$\tilde{E}_{M(j,k)} - \tilde{E}_{M(i,k)} + M(1 - y_{ijk}) \geq \tilde{T}_{M(j,k)} \tag{10}$$

$$\tilde{C}_i \leq (1 - R_{A(j)}) * \tilde{S}_{A(j)} + c, \text{ (c is a constant)} \tag{11}$$

$$\tilde{S}_{A(j+1)} = max[\tilde{E}_{A(j)}, \tilde{E}_{M(j+1)}^{max}] \tag{12}$$

$$\tilde{E}_{M(j+1)}^{max} = \max\{\tilde{E}_{M(j+1)}^{1}, \tilde{E}_{M(j+1)}^{2}\}, \dots, \tilde{E}_{M(j+1)}^{n} \tag{13}$$

$$R_{A(i,j)} = \begin{cases} 1 & \text{Job } i \text{ is not required at the } j\text{-th assembly position} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$x_{ihk} = \begin{cases} 1 & \text{Job } i \text{ is processed on machine } h \text{ before machine } k \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$y_{ijk} = \begin{cases} 1 & \text{Job } i \text{ is processed on machine } k \text{ before machine } j. \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$\tilde{T}_{M(i,k)} \geq 0 \tag{17}$$

$$\tilde{T}_{A(j)} \geq 0 \tag{18}$$

$$\tilde{S}_{A(j)} \geq 0 \tag{19}$$

Eq. 1 is the objective function of the model: minimizing the final fuzzy assembly time. Eq. 2 shows the relationship between assembly makespan and processing makespan. Eq. 3 specifies that the current assembly operation cannot start before the previous assembly operation has been completed. Eqs. 4 and 5 describe the makespan of a job on a processing machine. Eq. 6 specifies that the current job cannot be processed on the machine before the previous job has processed and removed. Eq. 7 describes the processing sequence of job i on machines k and h. Eq. 8 specifies that machine k cannot process two processes at the same time. Eq. 9 specifies that job i cannot be processed on two machines at the same time. Eq. 10 describes the operation sequence of jobs i and j on machine k. Eq. 11 specifies that the expected makespan of job i that ensures smooth assembly and minimizes the waiting time before assembly. Eq. 12 specifies that that assembly cannot begin unless all jobs of the final product in the previous or the next assembly task has been completed. Eq. 13 describes the makespan of the jobs required in assembly task *j*. Eq. 14 presents a Boolean variable that indicates whether job *i* is required for assembly task *j*. Eq. 15 presents a Boolean variable that specifies the operation sequence of job *i* on machines *h* and *k*. Eq. 16 presents a Boolean variable that specifies the operation sequence of jobs *i* and *j* on machine *k*.

5. Fuzzy transformation and fuzzy operation

5.1 Fuzzy transformation

Eq. 1 was transformed to convert the objective into the mean delivery satisfaction. Let triangular fuzzy numbers $\tilde{T}_{M(i,k)}(T^1_{M(i,k)}, T^2_{M(i,k)}, T^3_{M(i,k)})$ and $\tilde{T}_{A(j)}(T^1_{A(j)}, T^2_{A(j)}, T^3_{A(j)})$ be the operation time of each processing machine and that of each assembly machine, respectively (Fig. 1). Then, the delivery time of the final product was described as trapezoidal fuzzy numbers (Fig. 2). The fuzzified variables of operation time and delivery time can accurately demonstrate the effects of various uncertain factors on the production process, laying the basis for a flexible, adaptive and practical scheduling plan.

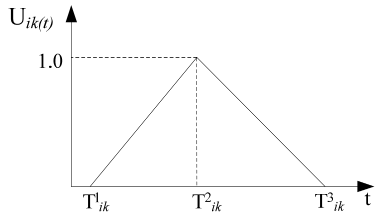

Fig. 1 Fuzzy operation time of each processing machine and assembly machine

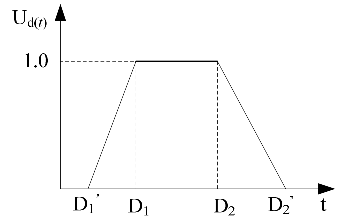

Fig. 2 Fuzzy delivery time of the final product

In Fig. 1, $T^1_{M(i,k)}$, $T^2_{M(i,k)}$ and $T^3_{M(i,k)}$ are the lower bound, mean and upper bound of the fuzzy operation time of each processing machine; $T^1_{A(j)}$, $T^2_{A(j)}$ and $T^3_{A(j)}$ are the lower bound, mean and upper bound of the fuzzy operation of each assembly machine. Fig. 2 explains if the final product is delivery within the window [D_1, D_2]. If yes, the degree of satisfaction is expressed as 1; otherwise, it is expressed as a linear membership function.

Under the premise of satisfying the delivery time, it is necessary to compare the product makespan with the fuzzy makespan in our model to see if it is optimal. For this purpose, the fuzzy objective function can be transformed as:

$$Max F_A(i) = \sum \omega_i \, f_A(i) \tag{20}$$

where ω_i is the weight of the delivery time of the assembled product; $f_{A(i)}$ can be calculated as:

$$f_A(i) = \begin{cases} 1 & E^m_{A(i)} \le E^1_i, E^u_{A(i)} \le E^2_i \\ \frac{(E^2_i - E^l_{A(i)})^2}{(E^2_i - E^1_i - E^l_{A(i)} + E^m_{A(i)})(E^u_{A(i)} - E^l_{A(i)})} & E^l_{A(i)} < E^2_i, E^u_{A(i)} \ge E^2_i, E^m_{A(i)} \ge E^1_i \\ 1 + \frac{(E^2_i - E^u_{A(i)})^2}{(E^2_i - E^1_i - E^u_{A(i)} + E^m_{A(i)})(E^u_{A(i)} - E^l_{A(i)})} & E^m_{A(i)} \le E^1_i, E^u_{A(i)} > E^2_i \\ \frac{(E^2_i - E^l_{A(i)})^2}{(E^2_i - E^1_i - E^l_{A(i)} + E^m_{A(i)})(E^u_{A(i)} - E^l_{A(i)})} - \frac{(E^2_i - E^u_{A(i)})^2}{\left(E^2_i - E^1_i - E^u_{A(i)} + E^m_{A(i)}\right)\left(E^u_{A(i)} - E^l_{A(i)}\right)} & E^l_{A(i)} < E^2_i, E^u_{A(i)} \le E^2_i, E^m_{A(i)} > E^1_i \end{cases} \tag{21}$$

5.2 Fuzzy operation

Our scheduling model involves the following fuzzy number operations: addition, subtraction, maximization and minimization. Let $\tilde{r} = (r^l, r^m, r^u)$ and $\tilde{t} = (t^l, t^m, t^u)$ be two fuzzy numbers. Then, the addition and maximization operations can be defined as:

$$\tilde{r} + \tilde{t} = (r^l + t^l, r^m + t^m, r^u + t^u) \tag{22}$$

$$\tilde{r} \vee \tilde{t} = (r^l \vee t^l, r^m \vee t^m, r^u \vee t^u) \tag{23}$$

The addition of fuzzy numbers determines the sequence and makespan of job processing, while the maximization determines the start time of the operation on a job. Under the above constraints, the fuzzy numbers in the inequalities can be compared in the following three steps:

Step 1: Compare the two fuzzy numbers by their c_1 values:

$$c_1(\tilde{r}) = \frac{r^l + 2r^m + r^u}{4} \tag{24}$$

Step 2: If the c_1 values are equal, compare the two fuzzy numbers by their c_2 values:

$$c_2(\tilde{r}) = r^m \tag{25}$$

Step 3: If the c_2 values are equal, compare the two fuzzy numbers by their c_3 values:

$$c_3(\tilde{r}) = r^u - r^l \tag{26}$$

5.3 Genetic algorithms

Based on the description and mathematical model of this extended JSP, the problem obviously belongs to NP-hard. Genetic algorithm (GA) is proposed by Professor Holland of the University of Michigan in the 1960s. So far, GA has been used to solve many combinatorial optimization problems. Due to the complexity of the problem, a GA is proposed to solve the problem. Each module will be introduced as follows.

6. Design of the integrated scheduling system based on genetic algorithms (GA)

This section designs a GA to solve our model under the time constraints on processing and assembly, owing to the computing complexity and constraint diversity. With the aim to satisfy the delivery time, the scheduling plan that best satisfies the delivery time under fuzzy operation time was considered as the optimal plan.

6.1 Encoding operation

The encoding was carried out in the real-coded mode, in which each gene in the chromosome indicates a processing operation. The genes were represented by the serial number of jobs. For example, each chromosome has $n \times m$ genes. If a job appears m times in the chromosome, it means the job needs to undergo m assembly operations. The processing sequence of a job was described by its order in the chromosome. Taking the five-operation chromosome [2 3 2 2 1 4 2 3 4 1 4 5 5 5 1 3 2 5 1 5 1 4 4 3 3] for instance, the processing should start with the first operation of the second job, followed by the first operation of the third job. The rest can be deduced by analogy. The process-based encoding approach differs from the traditional strategies in that it considers the order between processing and assembly operations and thus ensures the validity of the chromosomes.

6.2 Fitness function

The fitness $f_{(i)}$ of chromosome i can be calculated as:

$$f(i) = Z = min\{ max\, E_{M(i,k)} + \alpha \times \sum_{i=1}^{n} max[\,(C_i - E_{M(i)}),0] + max\, E_{A(j)+\beta\times\sum_{j=1}^{m} max[(S_{A(j+1)}-E_{A(j)}),0]\}} \tag{27}$$

The advance/tardiness penalty coefficients for jobs were set as α=1.2 and β=1.25, respectively.

6.3 Selection operation

Through steady-state replication, the most adaptive individuals were directly selected for the next generation, reflecting the spirit of the elite protection strategy, while the remaining individuals were selected for crossover and mutation at different probabilities. This selection approach ensures the robustness of the next iteration. The selection probability $P_{S(i)}$ of chromosome i can be expressed as:

$$P_{S(i)} = f(i)/\sum_{k=1}^{u} f(k) \tag{28}$$

where $f_{(i)}$ is the fitness of chromosome i.

6.4 Crossover operation

The general position intersection-based crossover was improved to ensure the feasibility of machines in the child chromosome. Based on the position intersections of the jobs, the serial number of a job was randomly determined and retained in a different parent chromosome from hat of the job position. Then, the vacant positions were filled by the remaining genes of another parent chromosome, forming a new child chromosome. For the two parent chromosomes P1={2 3 2 2 1 4 2 3 4 1 4 5 5 5 1 3 2 5 1 5 1 4 4 3 3} and P2={5 2 1 4 3 2 4 3 2 1 5 5 4 3 1 5 1 1 4 5 3 2 4 2 3}, the crossover was performed based on the intersection of job positions, producing a child chromosome shown in Fig. 3 below.

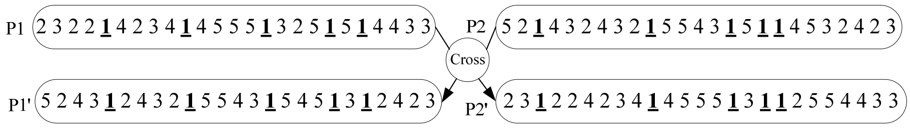

Fig. 3 Single-point crossover

6.5 Mutation operation

Because of the sequence constraints on jobs and machines, the assembly after the crossover and mutation may take place before the processing, resulting in an unfeasible solution. Thus, it is necessary to determine a solution is feasible through the following steps.

Step 1: Traverse all positions of a chromosome and find the job number of the previous assembly task.
Step 2: Determine if the job number is at the last position of the chromosome; if not, swap the job with the last job; otherwise, go to Step 3.
Step 3: Find the job number in the second to last position and record the current position P of this job number.
Step 4: Determine whether the job required for the assembly position has been completed before position P; if yes, terminate the determination process; otherwise, go to Step 5.
Step 5: Change the positions of all jobs that do not satisfy the sequence constraints until all the jobs are processed before the assembly operations.

According to the main parameters designed above, the GA to solve our integrated scheduling model for processing and assembly was developed based on single-point crossover and swap mutation. The flow chart of the algorithm is shown in Fig. 4.

As shown in Fig. 4, the proposed GA is implemented in the following steps.

Step 1: Initialize parameters like the population size (an even number), the number of iterations gen, the crossover probability P_c and the mutation probability P_m.
Step 2: Produce the initial population.
Step 3: Initialize the counter as $n = 0$.
Step 4: Calculate fitness $f(i)$ of each individual.
Step 5: Find the individuals $best\_flag$ with the best fitness $f(i)$ and directly import them into the next generation.
Step 6: Determine whether the $best\_flag$ in the next generation is N; if yes, go to Step 22; otherwise, go to Step 7.
Step 7: Calculate the selection probability of each remaining individual.
Step 8: Generate a random selection probability $P \in [0, 1]$, and perform roulette selection $(N - best\_flag)$ of the individuals in the mating pool.

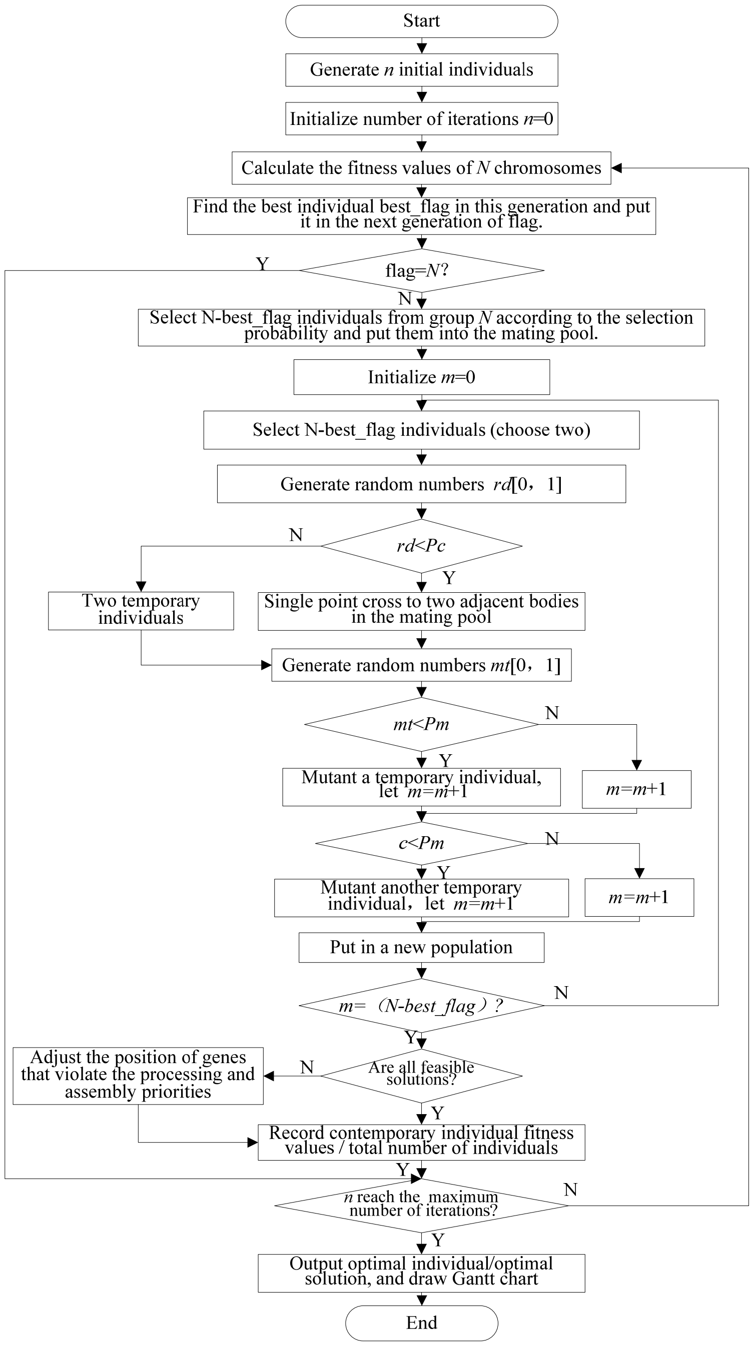

Fig. 4 Flow chart of the GA for solving our model

Step 9: Initialize the counter as $m = 0$.

Step 10: Determine the number of intersections k according to the number of optimal individuals, and randomly generate the crossover probability $rd \in [0, 1]$. If $rd < Pc$, go to Step 11; otherwise, regard the individuals as temporary ones and go to Step 13.

Step 11: Perform crossover of two parent chromosomes.

Step 12: Record the two generations after the crossover.

Step 13: Generate a random mutation probability as $mt \in [0, 1]$.

Step 14: If $mt < Pm$, go to Step 15; otherwise, go to Step 16.

Step 15: Perform mutation of one of the temporary individuals.

Step 16: Let $m = m + 1$.

Step 17: Generate another mutation probability as $mt \in [0, 1]$.

Step 18: If $mt < Pm$, go to Step 15; otherwise, go to Step 19.

Step 19: Perform mutation of another temporary individual.

Step 20: Let $m = m + 1$.
Step 21: Judge whether $m = N - best\_flag$. If yes, determine if the new population is a feasible solution, and adjust the infeasible solution into a feasible one.
Step 22: If $n = gen$, go to Step 23; otherwise, go to Step 4.
Step 23: Terminate the iteration and generate the optimal solution.

7. Results and discussion: A case study

This section aims to verify the effectiveness of the proposed model and algorithm through a case study. In our case, the final product A (Fig. 5) needs to be assembled from five jobs. Each job must go through operations on five processing machines (M_1, M_2, M_3, M_4, M_5). The processing machines can work in parallel. Thus, the time of each processing operation of each job can be denoted as T_{ij}, with i being the number of jobs (i=1, 2, 3, 4, 5) and j being the number of processing machines (j=1, 2, 3, 4, 5). In addition, each processed job must go through operations on four assembly machines (A_1, A_2, A_3, A_4). Note that the assembly operation of a job should not start before the completion of the job processing.

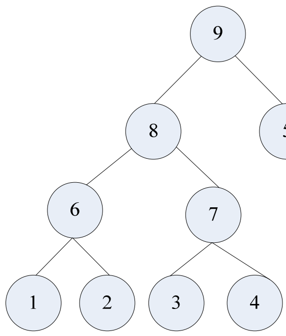

Fig. 5 The structure of the final product

Known parameters

The fuzzy operation time of each job is shown in Table 1, while the serial numbers of the machines that process and assemble each job are listed in Table 2, where the delivery time is expressed as a fuzzy number $\widetilde{D}$ (130, 135, 140, 145).

Table 1 The fuzzy operation time of each job

Parts	Processing and assembly machine tools								
	M1	M2	M3	M4	M5	A1	A2	A3	A4
job1	(4.5,5,5.5)	(6.5,7,7.5)	(4.5,5,5.5)	(6.5,7,7.5)	(5.5,6,6.5)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
job2	(9.5,10,10.5)	(8.5,9,9.5)	(5.5,6,6.5)	(4.5,5,5.5)	(9.5,10,10.5)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
job3	(8.5,9,9.5)	(4.5,5,5.5)	(5.5,6,6.5)	(4.5,5,5.5)	(6.5,7,7.5)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
job4	(5.5,6,6.5)	(5.5,6,6.5)	(9.5,10,10.5)	(3.5,4,4.5)	(4.5,5,5.5)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
job5	(9.5,10,10.5)	(5.5,6,6.5)	(5.5,6,6.5)	(5.5,6,6.5)	(3.5,4,4.5)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
part6	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(14.5,15,15.5)	(0,0,0)	(0,0,0)	(0,0,0)
part7	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(19.5,20,20.5)	(0,0,0)	(0,0,0)
part8	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(19.5,20,20.5)	(0,0,0)
part9	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(24.5,25,25.5)

Table 2 The serial numbers of the machines that process and assemble each job

Parts/Assembly process	Processing/assembly process				
	1	2	3	4	5
job1	M3	M1	M2	M4	M5
job2	M2	M3	M5	M1	M4
job3	M3	M4	M1	M2	M5
job4	M2	M1	M3	M4	M5
job5	M3	M2	M5	M1	M4
part6	A1	A1	A1	A1	A1
part7	A2	A2	A2	A2	A2
part8	A3	A3	A3	A3	A3
part9	A4	A4	A4	A4	A4

Assumptions

- The processed job can be transported directly to the assembly machines and the transport time should be taken into consideration.
- Regardless of the storage after all jobs are processed, the buffer storage between processing and assembly is finite, and the premature completion of jobs is subjected to a penalty on the buffer storage.
- Processing machines and assembly machine are readily available.
- The jobs are processed strictly according to the operation sequence, and the operations should not be interchanged.
- The jobs are allowed to wait between different operations, and the machines are allowed to remain idle before the jobs arrive.
- The machines do not malfunction.

Solution and analysis

The proposed GA was programmed on Matlab with the following parameters: the population size of 20, the number of iterations of 100, the crossover probability *Pc* of 0.9, and the mutation probability *Pm* of 0.1. Through the simulation, the optimal solution was determined as x= [4 2 2 1 4 1 1 2 3 3 2 1 5 3 4 5 4 5 5 2 1 3 5 4 3 6 7 8 9]; the degree of satisfaction of delivery time was 1. Then, the mean fuzzy time of each operation was plotted as a Gantt chart (Fig. 6).

The case study shows that the proposed model can successfully schedule the processing and assembly machines. Taking the assembly stage into consideration, this research improves the practical value of the scheduling problem and helps to prepare a feasible scheduling plan. After all, most processed jobs in real world need to be treated on the assembly line before forming the final product.

Besides, this paper fully considers the delivery time, a key determinant of the production schedule. Therefore, the scheduling of our model is closer to the actual production situation than the previous models. The proposed model enjoys high practical value by taking the customer satisfaction of the delivery period as the goal.

In addition, the processing and assembly time were expressed as triangular fuzzy numbers, while the delivery time was described as trapezoidal fuzzy numbers. These fuzzy numbers accurately reflect the uncertainty in operation time, assembly time and delivery time of actual processing and assembly lines.

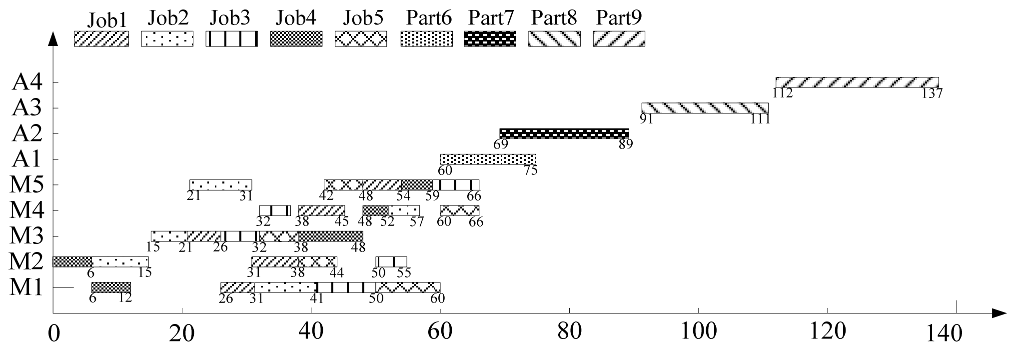

Fig. 6 Gantt chart of mean fuzzy time of each operation

Further discussion

Processing and assembly are considered into JSP simultaneously in this paper. The best solution obtained by scheduling processing machines and assembly machines simultaneously is more feasible for determining the real scheduling solution than the best solution obtained by scheduling processing machines only.

In the actual production environment, there are many factors affecting production scheduling, such as batch, multi-objective, multi-process routes and so on. In the future works, the extended JSP considering different factors will be studied. Besides, for improving performance of algorithms, algorithm improvment is another important aspect in future works.

8. Conclusion

The traditional JSPs often takes the minimal makespan as the only optimization goal, and only considers the scheduling of the machines related to processing. Hence, it is difficult for the traditional models to demonstrate the actual production situation. Considering the uncertainty of time factors in actual production, this paper represents operation time of the jobs as triangular fuzzy numbers and the delivery time of the final product as trapezoidal fuzzy numbers. Then, a mathematical model was established for processing and assembly scheduling, aiming to achieve the mean satisfaction degree on delivery time. In light of the complexity of the problem, a GA was designed to solve our model under the time constraints on processing and assembly, and verified through a case study on processing and assembly scheduling. The results show that our scheduling model mirrors the actual production situation and provides a good reference for JSP scheduling under multiple uncertainties.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant No: 61402361, 60903124); Project supported by the scientific research project of Shaanxi Provincial Department of Education (Grant No: 14JK1521); Shaanxi province science and technology research and development project (Grant No: 2012KJXX-34); Xi'an University of Technology Initial Foundation for the PhDs (Grant No: 102-451117013).

References

[1] Nagata, Y., Ono, I. (2018). A guided local search with iterative ejections of bottleneck operations for the job shop scheduling problem, *Computers & Operations Research*, Vol. 90, 60-71, doi: 10.1016/j.cor.2017.09.017.

[2] Sharma, N., Sharma, H., Sharma, A. (2018). Beer froth artificial bee colony algorithm for job-shop scheduling problem, *Applied Soft Computing*, Vol. 68, 507-524, doi: 10.1016/j.asoc.2018.04.001.

[3] Bierwirth, C., Kuhpfahl, J. (2017). Extended GRASP for the job shop scheduling problem with total weighted tardiness objective, *European Journal of Operational Research*, Vol. 261, No. 3, 835-848, doi: 10.1016/j.ejor.2017.03.030.

[4] Piroozfard, H., Wong, K.Y., Asl, A.D. (2017). An improved biogeography-based optimization for achieving optimal job shop scheduling solutions, *Procedia Computer Science,* Vol. 115, 30-38, doi: 10.1016/j.procs.2017.09.073.

[5] Akram, K., Kamal, K., Zeb, A. (2016). Fast simulated annealing hybridized with quenching for solving job shop scheduling problem, *Applied Soft Computing*, Vol. 49, 510-523, doi: 10.1016/j.asoc.2016.08.037.

[6] Hu, H.X., Lei, W.X., Gao, X., Zhang, Y. (2018). Job-shop scheduling problem based on improved cuckoo search algorithm, *International Journal of Simulation Modelling,* Vol. 17, No. 2, 337-346, doi: 10.2507/IJSIMM17(2)CO8.

[7] Modrák, V., Pandian, R.S. (2010). Flow shop scheduling algorithm to minimize completion time for n-jobs m-machines problem, *Tehnički Vjesnik – Technical Gazette*, Vol. 17, No. 3, 273-278.

[8] Lin, J. (2019). Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time, *Engineering Applications of Artificial Intelligence*, Vol. 77, 186-196, doi: 10.1016/j.engappai.2018.10.008.

[9] Shen, L., Dauzère-Pérès, S., Neufeld, J.S. (2018). Solving the flexible job shop scheduling problem with sequence-dependent setup times, *European Journal of Operational Research*, Vol. 265, No. 2, 503-516, doi: 10.1016/j.ejor.2017.08.021.

[10] Kato, E.R.R., Aranha, G.D.A., Tsunaki, R.H. (2018). A new approach to solve the flexible job shop problem based on a hybrid particle swarm optimization and random-restart hill climbing, *Computers & Industrial Engineering*, Vol. 125, 178-189, doi: 10.1016/j.cie.2018.08.022.

[11] Wu, X., Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures, *Journal of Cleaner Production*, Vol. 172, 3249-3264, doi: 10.1016/j.jclepro.2017.10.342.

[12] Yang, X.P., Gao, X.L. (2018). Optimization of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm, *International Journal of Simulation Modelling*, Vol. 17, No. 4, 724-733, doi: 10.2507/IJSIMM17(4)CO19.

[13] Seng, D.W., Li, J.W., Fang, X.J., Zhang, X.F., Chen, J. (2018). Low-carbon flexible job-shop scheduling based on improved nondominated sorting genetic algorithm-II, *International Journal of Simulation Modelling*, Vol. 17, No. 4, 712-723, doi: 10.2507/IJSIMM17(4)CO18.

[14] Nidhiry, N.M., Saravanan, R. (2014). Scheduling optimization of a flexible manufacturing system using a modified NSGA-II algorithm, *Advances in Production Engineering & Management*, Vol. 9, No. 3, 139-151, doi: 10.14743/apem2014.3.183.

[15] Xu, H., Bao, Z.R., Zhang, T. (2017). Solving dual flexible job-shop scheduling problem using a Bat Algorithm, *Advances in Production Engineering & Management*, Vol. 12, No. 1, 5-16, doi: 10.14743/apem2017.1.235.

[16] Yazdani, M., Aleti, A., Khalili, S.M., Jolai, F. (2017). Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 107, 12-24, doi: 10.1016/j.cie.2017.02.019.
[17] Chaouch, I., Driss, O.B., Ghedira, K. (2017). A modified ant colony optimization algorithm for the distributed job shop scheduling problem, *Procedia Computer Science,* Vol. 112, 296-305, doi: 10.1016/j.procs.2017. 08.267.
[18] Kurdi, M. (2017). An improved island model memetic algorithm with a new cooperation phase for multi-objective job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 111, 183-201, doi: 10.1016/j.cie.2017.07.021.
[19] Shahrabi, J., Adibi, M.A., Mahootchi, M. (2017). A reinforcement learning approach to parameter estimation in dynamic job shop scheduling, *Computers & Industrial Engineering*, Vol. 110, 75-82, doi: 10.1016/j.cie.2017.05.026.
[20] Kundakcı, N., Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 96, 31-51, doi: 10.1016/j.cie.2016.03.011.
[21] Kuhpfahl, J., Bierwirth, C. (2016). A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective, *Computers & Operations Research*, Vol. 66, 44-57, doi: 10.1016/j.cor.2015.07.011.
[22] Zhong, Q., Yang, H., Tang, T. (2018). Optimization algorithm simulation for dual-resource constrained job-shop scheduling, *International Journal of Simulation Modelling*, Vol. 17, No. 1, 147-158, doi: 10.2507/IJSIMM17(1)CO2.
[23] Haider, A., Mirza, J. (2015). An implementation of lean scheduling in a job shop environment, *Advances in Production Engineering & Management*, Vol. 10, No. 1, 5-17, doi: 10.14743/apem2015.1.188.
[24] Chaudhry, I.A., Usman, M. (2017). Integrated process planning and scheduling using genetic algorithms, *Tehnički Vjesnik – Technical Gazette,* Vol. 24, No. 5, 1401-1409, doi: 10.17559/TV-20151121212910.
[25] Yu, H. (2018). Numerical simulation of European option payoff based on stochastic differential delay equations, *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 2, 102-107, doi: 10.18280/mmep.050207.
[26] Song, S.L. (2018). Application of gray prediction and linear programming model in economic management, *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 1, 46-50, doi: 10.18280/mmep.050107.
[27] Kueh, S.M., Kuok, K.K. (2018). Forecasting long term precipitation using cuckoo search optimization neural network models, *Environmental Engineering and Management Journal*, Vol. 17, No. 6, 1283-1291, doi: 10.30638/eemj.2018.127.