

Chapter 42

Solving Fuzzy Job Shop Scheduling Problem Based on Interval Number Theory

Chuan He, Dishan Qiu and Hao Guo

Abstract This article discusses the job shop scheduling problem with fuzzy processing time and fuzzy deadline by using interval number theory, which is an efficient method to denote imprecise parameter. Firstly, we convert the original problem to constraint satisfaction problem (CSP) with the assumption that agreement index (AI) is a main optimization objective. Then, the particle swarm optimization (PSO) is merged with genetic algorithm (GA), i.e., an improved particle swarm optimization (IPSO) being used to solve the problem. Finally, the effectiveness of this algorithm is verified by large number of experiments.

Keywords Fuzzy job shop · Scheduling · IPSO algorithm · Interval number

42.1 Introduction

Over the past decades, intensive researches have been conducted aiming at various applications of job shop scheduling problem (JSSP), for instance production planning and control. The typical assumption of JSSP is: there are a set of N jobs and a set of M machines or work nodes. Each job consists of a specific operation set which expresses a distinct processing route that has already been fixed and known in advance. All machines are available at time zero, and preemption is forbidden in work process. Most of all, the processing time of each operation has been given clearly. However, owing to the time-based-competition, the impact of

C. He (✉) · D. Qiu · H. Guo
Science and Technology on Information Systems Engineering Laboratory, Nation
University of defense technology, Changsha, China
e-mail: chuanhe@nudt.edu.cn

uncertain factors cannot be ignored [1]. The conventional JSSP is not suitable in many cases due to various kinds of vagueness existing in real-world scheduling problems [2]. To meet the requirements of applications, many theses have applied the JSSP in uncertain environment, i.e., fuzzy job shop scheduling problem (FJSSP) [3].

The conventional optimization objectives of JSSP is to reduce tardiness (or Make span) or maximize the utilization of each machine [4, 5]. Generally speaking, many objectives are conflicting, which means it is impossible to optimize all the objectives. Bilkay and others [6] had ever proposed a delay cost concept during the decision-making to minimize the total penalty for the completion delay of the operations, which is one of the most important objectives. Sakawa and Kubota [7] provided a multi-objective FJSSP with fuzzy processing time, which is to maximize the agreement index (AI) and optimize the maximum fuzzy completion time.

In current research, stochastic method and fuzzy set theory are proposed to handle imprecise variables in FJSSP. Sakawa and Mori [8] have developed a novel concept of similarity among individuals, which is an effective solution to measure fuzzy parameters. Meanwhile, Lei [3], Canbolat and Gundogar [9] have established kinds of membership functions (e.g. triangle, trapezoid, and rectangle membership functions) to represent uncertain processing time. Furthermore, Wu and others [10] had applied the fuzzy ranking method and fuzzy procedure into FJSSP to compare the priority of fuzzy variables, which is an effective method to yield scheduling planning.

In this research, interval numbers are adopted to represent fuzzy parameters and a novel ranking method towards interval numbers is proposed. After that, we analyze the constraint conditions of FJSSP and convert it into constraint satisfaction problem with the maximum AI as an optimization objective. The IPSO algorithm is used to yield solutions and the experiment results have proved a good performance of the proposed algorithm.

42.2 Problems and Formulation

42.2.1 Interval Number Theory

Definition 1 [11] We use R to denote a real number set, and for any $a_i^-, a_i^+ \in R$ we refer to the close interval $a_i = [a_i^-, a_i^+]$ as an interval number when $a_i^- \leq a_i^+$. Especially in case of $a_i^- = a_i^+$, a_i is degenerated as a real number.

By Definition 1, for a discrete interval number a_i , the point values of a_i^- and a_i^+ must be known exactly. Generally speaking, the actual value of a_i is a real number in the close interval $[a_i^-, a_i^+]$, and subject to a special probability distribution (or a membership function).

Definition 2 Suppose X is the actual value of a_i , and x is a discrete variable in $[a_i^-, a_i^+]$. Then the probability of $X \leq x$ is $F_i(x) = \int_{a_i^-}^x f_i(X)dX$, where $F_i(x)$ and $f_i(x)$ are referred to as the probability distribution function and probability density function respectively.

At present, conventional ranking methods only suitable for interval numbers of special types, for instance, triangle interval number, uniform interval number, and trapezoid interval number [12]. Therefore, the definition of fuzzy binary ordering relation is proposed to determine the priority of any fuzzy variables.

Definition 3 Suppose ‘ \prec ’, ‘ \succ ’ and ‘ $=$ ’ are the fuzzy binary ordering relations for a nonempty interval number set I , and they are referred to as partial order in case the following conditions are satisfied:

- (1) Reflexivity: $a_i = a_i, \forall a_i \in I$;
- (2) Antisymmetry: If $a_j \succ a_i$, then $a_i \prec a_j, \forall a_i, a_j \in I$;
- (3) Transitivity: If $a_i \prec a_j$ and $a_j \prec a_k$, then $a_i \prec a_k, \forall a_i, a_j, a_k \in I$.

Definition 4 If a_i and a_j have been related to probability density functions $f_i(x)$ and $f_j(x)$ respectively, the probability reliability of the partial order ‘ \succ ’, ‘ \prec ’ between a_i and a_j are:

$$\begin{aligned}
 P(a_i \succ a_j) &= \iint_{x \geq y} f_i(x)f_j(y)dxdy, \\
 P(a_i \prec a_j) &= \iint_{x \leq y} f_i(x)f_j(y)dxdy.
 \end{aligned}
 \tag{42.1}$$

Definition 5 For any two interval numbers a_i, a_j , if $P(a_i \succ a_j) > 0.5$, we set $a_i \succ a_j$; if $P(a_i \succ a_j) < 0.5$ we set $a_i \prec a_j$; if $P(a_i \succ a_j) = 0.5$, we set $a_i = a_j$.

The summation and maximum operations are used to solve the FJSSP, therefore, we define aggregate and maximization below:

Definition 6 For discrete interval numbers a_i and a_j , aggregate operation \oplus is given as follows:

$$\tilde{a}_i \oplus \tilde{a}_j = [a_i^- + a_j^-, a_i^+ + a_j^+].
 \tag{42.2}$$

42.2.2 Problem Discussion

The $n \times m$ FJSSP can be described as follows: The scheduling system consists of a job set $\{J_1, J_2, \dots, J_n\}$ and a machine set $\{M_1, M_2, \dots, M_m\}$. There are exist

precedence constraints, therefore for each job J_i is composed of several operations $\{O_{i1}, O_{i2}, \dots, O_{im}\}$. In the meantime, owing to the constraints of executable, each operation O_{ij} requires uninterrupted and exclusive special machine in its whole processing time. Additionally, deadline of each job has to be considered in the thesis. Other constraints in JSSP are also suitable for FJSSP. We summarize main notations being used in this thesis for your reference as follows:

n	The amount of jobs in scheduling system;
m	The amount of operations in scheduling system;
$T = [t_{ij}]_{n \times m}$	The processing time matrix, where t_{ij} denotes the processing time of O_{ij} ;
$W = [w_{ij}]_{n \times m}$	The execution machine matrix, where w_{ij} denotes processing machine of O_{ij} ;
$D = \{d_1, d_2, \dots, d_n\}$	The deadline vector, where d_i denotes the deadline of J_i ;
$X = [x_{ij}]_{n \times m}$	The priority execution matrix, where x_{ij} is an integer in $[1, mn]$ to denote the priority execution ranking of O_{ij} ;
$TB = [tb_{ij}]_{n \times m}$	The beginning time matrix, where tb_{ij} denotes the beginning time of O_{ij} ;
$TE = [te_{ij}]_{n \times m}$	The end time matrix, where te_{ij} denotes the completion time of O_{ij} ;

The solution of FJSSP is the process of matching job tasks and machine resources being connected with time and space. We have confronted with two issues, i.e., assigning each operation to an appropriate machine and also ranking the sequence for all operations. There are several strict constraints must be followed:

- (1) Each machine can only execute one job once at the same time, and the jobs assigned to the same machines should be executed in order;
- (2) Each job has a unique priority of execution, i.e., any two jobs have different execution order;
- (3) Only one operation of a job can be executed at one time and the times of executions should no more than one;
- (4) All operations of a job should be executed in order, and the completion time is determined by the last operation;
- (5) Each operation is assigned to the fixed machine in advance, and the working time is no less than the required time;
- (6) Preemptive is not allowed; therefore the operation will be retained until completion.

Deadline is one of the most important factors and the scheduling planning must take it in consideration, namely, the operations must be completed in their due time. Therefore, the degree of agreement has been set up as an optimization objective in our thesis.

Definition 7 Agreement index (AI) is the expected value of those jobs which have been completed prior to deadline. It is approximated as follows:

$$\begin{aligned}
 AI &= \sum_{i=1}^n \mu(te_{im}, d_i), \\
 \mu(a_i, a_j) &= \int_{a_i^-}^{a_i^+} f_i(x)F_j(x) dx.
 \end{aligned}
 \tag{42.3}$$

where $f_i(x)$ is the probability density function of a_i ; $F_i(x)$ is the probability distribution function of a_j .

Based on the constraints described previously, FJSSP can be converted into constraint satisfaction problem (CSP), and the CSP model is given as:

$$\begin{aligned}
 & \text{Max } z = AI \\
 & \text{s.t. } \left\{ \begin{array}{l}
 te_{ij} \prec tb_{ij}, \text{ if } (x_{ij} < x_{kj}) \wedge (w_{ij} = w_{kj}), \\
 x_{ij} \neq x_{kl}, \text{ if } (i \neq k) \vee (j \neq l), \\
 (te_{ij} \prec tb_{il}) \wedge (tb_{ij} \prec te_{il}), \text{ if } j < l, \\
 x_{ij} < x_{il}, \text{ if } j < l, \\
 te_{ij} \succ tb_{ij} \oplus t_{ij}, \\
 (te_{ij} \prec tb_{il}) \vee (tb_{ij} \succ te_{il}), \text{ if } w_{ij} = w_{il}, \\
 i, k \in [1, n], j, l \in [1, m].
 \end{array} \right.
 \end{aligned}
 \tag{42.4}$$

where AI is the optimization objective of CSP model, and Eq. (42.4) corresponds with the mentioned six constraints.

42.3 Algorithm Designing

42.3.1 The Architecture of Algorithm

The IPSO algorithm has been employed to find the solution of the present combinatorial optimization problem. The efficiency of IPSO algorithm is reflected in its fast search ability. However, the convergent speed depends substantially on initial swarm and tends to integrate into local optimization. Consequently, we hereby combine PSO with genetic algorithm (GA) and heuristic algorithms, and then a novel improved PSO (IPSO) is generated as Algorithm 1.

Algorithm 1: The pseudocode of IPSO

```

1:  $PSOn \leftarrow PSOPopulation$ ;  $GlobalOpt \leftarrow NULL$ ;  $PartOpt \leftarrow NULL$ ;
2: HeuristicOperation(); /* initialize swarm by heuristic algorithm */
3: CalculateSwarmAI(); /* evaluate the AI of each particle in swarm */
4: While termination condition is not satisfied do
5:   Update( $GlobalOpt, PartOpt$ ); /* save the best individual of swarm */
6:   MoveOperation();
7:   CalculateSwarmAI();
8:   If particle swarm is evolve then /*the swarm is evolution */
9:     ChooseOperation();
10:    CrossoverOperation();
11:   Else
12:     MutationOperation();
13:   End if
14: End while

```

42.3.2 Heuristic Algorithm

To improve the search efficiency of scheduling method, three heuristic algorithms are proposed and applied to produce the initialized elitism particles.

42.3.2.1 EDF Algorithm

The earliest deadline first (EDF) algorithm mainly picks up each job according to corresponding deadline in non-descending order. In particular, since deadline is a fuzzy variable, we sort the jobs with the partial order of due time and arrange the scheduling prior sequence based on the ranking result.

42.3.2.2 SPTF/LPTF Algorithm

The shortest/longest processing time first (SPTF/LPTF) algorithm priority is used to calculate the process time of each job and select the job which consumes the minimum/maximum process time for executing object.

42.3.2.3 LNDF Algorithm

The largest nearby degree first (LNDF) algorithm will select the job which has the nearest processing time to its deadline. The degree of closeness for job_i can be defined as:

$$Var_i = \mu \left(\sum_{j=1}^m t_{ij}, d_i \right) \quad (42.5)$$

42.3.3 Algorithm Representation

In the thesis, we encode particles with job numbers, and the chromosome denotes the executing priority sequence of operations. For an $n \times m$ FJSSP, the $n \times m$ length chromosome is established. Each gene represents a special operation, which is expressed in a gene no more than m . The moving vector is designed to implement moving operation, and a chromosome is realized by ranking its moving vector in non-descending order. The chromosome is transferred into a decision matrix in encoding process, and then we can calculate the beginning time and completion time of each operation according to the processing route. The decoding method of particle is designed as Algorithm 2.

Algorithm 1: The pseudocode of heuristic algorithm

```

1:  $\forall i \in [1, n], Idle_i \leftarrow 0$ ; /* initialize idle times of each machine */
2: For  $k \leftarrow 1$  to  $n \times m$ , find  $x_{kj} = k$  do
3:    $g \leftarrow w_{kj}$ ; /* acquire the node of machine */
4:   IF  $j = 1$  then /* it is the first operation */
5:      $tb_{kj} \leftarrow Idle_g$ ; /* the beginning time */
6:   Else
7:      $tb_{kj} \leftarrow \max\{Idle_g, te_{i,(j-1)}\}$ ;
8:   End if
9:    $te_{kj} \leftarrow tb_{kj} \oplus t_{ij}$ ; /* the end time */
10:   $Idle_g \leftarrow te_{kj}$ ;
11: End for

```

42.4 Computation Experiments

In this section, a simulation program is manipulated in Matlab2007 on a personal computer with Pentium IV 3.06GHZ CPU. To test the efficiency of IPSO, we compare it with GA and PSO in twelve benchmark problems (e.g. LA01, LA07,

Table 42.1 Summary results of the different algorithms

Problems	GA		PSO		IPSO	
	AI	CPU time (s)	AI	CPU time (s)	AI	CPU time (s)
LA01(10 × 5)	9.84	1.04	9.09	2.17	9.84	2.19
LA07(15 × 5)	14.10	2.59	13.39	4.10	14.10	4.21
LA11(20 × 5)	16.01	4.09	14.56	6.43	15.45	7.08
LA19(10 × 10)	9.72	1.05	9.24	2.01	9.72	2.12
LA21(15 × 10)	14.74	3.86	14.90	7.47	14.90	7.65
LA27(20 × 10)	11.79	3.57	11.22	5.26	12.08	5.33
YN1(20 × 20)	19.69	16.61	18.57	6.21	19.83	6.56
YN2(20 × 20)	16.21	16.55	15.94	6.12	16.28	6.52
YN3(20 × 20)	19.28	14.83	19.63	3.81	19.63	4.00
SWV06(20 × 10)	13.78	17.93	14.24	23.37	14.24	25.19
SWV08(20 × 15)	17.88	6.51	16.65	7.23	17.88	7.44

LA11, LA19, LA21, LA27, SWV06, SWV08, SWV10, YN1, YN2 and YN3) and the GA and PSO also adopt the same parameters in IPSO. Given the variables in twelve benchmark problems are certain values, we use triangle interval numbers and trapezoid interval numbers to realize the fuzzy disposition of processing time and deadline respectively.

From Table 42.1, we can find out obviously that the IPSO algorithm has eight solutions better than GA. There are four equations, and only one of them is worse than GA. At the same time, only three solutions of IPSO have the same values as PSO, but the CPU time is 5.25 % longer than PSO since the mutation operation are involved in the IPSO. This experiment shows that IPSO can avert early convergence and also search for more optimal solutions, which means high efficiency could be achieved.

42.5 Conclusion

In the thesis, FJSSP with fuzzy processing time and fuzzy deadline has been analyzed. We convert the FJSSP to CSP by defining fuzzy priority rules, in addition, IPSO algorithm is proposed to solve the problem. Heuristic algorithms are combined with the PSO algorithms to improve the search ability of the latter issue. The experiment results have proved the effectiveness of the methods being employed.

References

1. Chen D, Liu F, Luh PB (1997) Scheduling job shops with uncertainties. In: Proceedings of the 36th conference on decision and control PCDCDZ, San Diego pp 3556–3561
2. Mohammad SM, Pahlavani A (2009) A Fuzzy multi-objective programming for scheduling of weighted jobs on a single machine. *Int J Adv Manuf Technol* 45(1–2):122–139

3. Lei DM (2010) Solving fuzzy job shop scheduling problems using random key genetic algorithm. *Int J Adv Manuf Technol* 49(1–4):253–262
4. Mohammad SM, Parviz F (2007) Flexible job shop scheduling with tabu search algorithms. *Int J Adv Manuf Technol* 32(5–6):563–570
5. Chen JS (2006) Single-machine scheduling with flexible and periodic maintenance. *J Oper Res Soc* 57(6):703–710
6. Bilkay O, Anlagan O, Kilic SE (2004) Job shop scheduling using fuzzy logic. *Int J Adv Manuf Technol* 23(7–8):606–619
7. Sakawa M, Kubota R (2000) Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *Eur J Oper Res* 120(2):393–407
8. Sakawa M, Mori T (1999) An efficient genetic algorithm for job shop scheduling problems with fuzzy processing time and fuzzy due date. *Comput Ind Eng* 36(2):325–341
9. Canbolat YB, Gundogar E (2004) Fuzzy priority rule for job shop scheduling. *J Intell Manuf* 15(4):527–533
10. Wu CS, Li DC, Tsai TI (2006) Applying the fuzzy ranking method to the shifting bottleneck procedure to solve scheduling problems of uncertainty. *Int J Adv Manuf Technol* 31(1–2):98–106
11. Sengupta PTK (2000) On comparing interval number. *Eur J Oper Res* 127(1):28–43
12. Feng TL (2002) Fuzzy job-shop scheduling based on ranking level interval-valued fuzzy numbers. *IEEE Trans Fuzzy Syst* 10(4):510–522