

An uncertain multi-objective programming model for machine scheduling problem

Yufu Ning^{1,2} · Xiumei Chen^{1,2} · Zhiyong Wang^{1,2} · Xiangying Li^{1,2}

Received: 4 November 2015 / Accepted: 8 March 2016 / Published online: 21 March 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This paper discusses a parallel machine scheduling problem in which the processing times of jobs and the release dates are independent uncertain variables with known uncertainty distributions. An uncertain programming model with multiple objectives is obtained, whose first objective is to minimize the maximum completion time or makespan, and second objective is to minimize the maximum tardiness time. A genetic algorithm is employed to solve the proposed uncertain machine scheduling model, and its efficiency is illustrated by some numerical experiments.

Keywords Machine scheduling problem · Uncertain variable · Uncertain programming · Genetic algorithm

1 Introduction

Machine scheduling problem is concerned with finding an efficient schedule during an uninterrupted period of time for a set of machines to process a set of jobs. Machine scheduling problems arise in a wide field of real life and so are important from both theoretical and practical viewpoints. Since the pioneering work of Johnson [9] and Naughton [23], theory for machine scheduling has been greatly enlarged and improved. In 1977, Lenstra et al. [12] studied the complexity of machine scheduling problems,

and showed that most classical machine scheduling problems were NP-complete. In 1979 Graham et al. [6] proposed an expression of three parameters $\alpha | \beta | \gamma$ where α , β and γ represented machine environment, job characteristic and optimality criteria respectively to describe scheduling problems.

In a machine scheduling problem, processing time and release date are important factors that affect scheduling plan. Originally, release dates and processing times were assumed to be exact numbers. A great deal of models and algorithms have been developed, such as Baker and Scudder [2], Mokotoff [22], Lam and Xing [11], Pinedo [27] and Xing et al. [32]. Since in real life processing times and release dates are usually difficult to be crisp numbers, the indeterminacy is taken into account. With the proposition of probability theory by Kolmogorov in 1933, Banerjee [3] was the first who investigated a single machine scheduling problem with random processing times. Later, different machine scheduling problems with random processing times were researched. In 1977, Hodegson [7] described a single machine scheduling problem with random processing times, in which he showed that Lawler's efficient algorithm was optimal even when the processing times were random. Seo et al. [29] studied the single machine scheduling problem for the objective of minimizing the expected number of tardy jobs where the jobs had normally distributed processing times and a common deterministic due date. Scheduling problems with random release dates were also studied by many researchers. Pinedo [26] studied stochastic scheduling problems in which the processing times of jobs, the release dates and the due dates were random variables and the objectives were minimization of the expected weighted sum of completion times and the expected weighted number of late jobs. Ahmadizar et al. [1] dealt with a stochastic group

✉ Yufu Ning
yufuning@163.com

¹ School of Information Engineering, Shandong Youth University of Political Science, Jinan 250103, China

² Key Laboratory of Information Security and Intelligent Control in Universities of Shandong, Jinan 250103, China

shop scheduling problem in which both the release date of each job and the processing time of each job on each machine were random variables with known distributions, and the objective was to find a job schedule which minimized the expected makespan.

Since the concept of fuzzy set was proposed by Zadeh in 1965, many researchers began to study machine scheduling problems by fuzzy set theory. Prade [28] first used fuzzy set theory to deal with a real scheduling problem in 1979. Later, Ishibuchi et al. [8] proposed two fuzzy flowshop scheduling problems with fuzzy due dates and provided hybrid genetic algorithms and neighborhood search algorithms. Peng and Liu [24] presented three types of parallel machine scheduling models with fuzzy processing times in order to satisfy the different decision requirements, and used a revised genetic algorithm to solve the models in 2004. Petrovic and Fayad [25] described a fuzzy shifting bottleneck procedure hybridised with genetic algorithm for a real-world job shop scheduling problem where fuzzy sets were used to model uncertain processing times of jobs and their release dates and due dates. Gharehgozli et al. [5] presented a new mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates. Two objectives were considered in the model to minimize the total weighted flow time and the total weighted tardiness simultaneously in 2009.

As we know, in reality, there are cases that we lack observed data and all available are only belief degrees that are given by experts to describe the distributions of processing times. For example, when processing new types of products, there are no historical data of the processing times and the distributions of them can only be given by domain experts' evaluations expressed by belief degrees. Kehneman and Tversky [10] found that too much weight was given to the chance of unlikely events. A lot of surveys showed that *human beings usually estimate a much wider range of values than the object actually takes* (Liu [17]). For this reason, the belief degree is inappropriate to be treated as random variable or fuzzy variable. In order to deal with the belief degree mathematically, an uncertainty theory was founded by Liu [13] in 2007 and refined by Liu [16] in 2010 based on normality axiom, duality axiom, subadditivity axiom and product axiom.

Within the framework of uncertainty theory, a concept of uncertain variable was defined by Liu [13] as a measurable function on an uncertainty space. Meanwhile, a series of concepts such as uncertainty distribution, independence, expected value, variance and entropy, etc, were also proposed to describe an uncertain variable. And now, uncertainty theory has been deeply developed in many fields such as uncertain programming [18, 31, 33],

uncertain finance [4, 15] and uncertain differential equation [30, 34], etc.

Uncertain programming, as a type of mathematical programming involving uncertain variables, was first proposed by Liu [14] in 2009. Machine scheduling problem with uncertain processing times was also proposed as an example. An uncertain machine scheduling model [16] was built, of which the objective is to minimize the expected makespan. Since then uncertain multi-objective programming and uncertain goal programming were presented by Liu and Chen [18], and uncertain multilevel programming was proposed by Liu and Yao [19]. In 2013, Zhang and Meng [35] proposed an expected-variance-entropy model for uncertain parallel machine scheduling where the processing times are regarded as uncertain variables with known uncertainty distributions. Li and Liu [20] built an uncertain goal programming model for the machine scheduling problem and an intelligent algorithm was introduced to solve the equivalence based on a revised genetic algorithm.

Using uncertainty theory to solve machine scheduling problem, less effort has been done for studying the case where the release date of each job is indeterminate. In reality, the release dates of jobs are not determinate. So in many literatures, release dates are considered as random variables [1, 26] or fuzzy variables [5, 25]. However, for example, when there does not exist sample data for the release date of a new job, it is advisable to regard the release date as an uncertain variable. The domain experts are invited to estimate the distributions of the uncertain variables so as to be accordant with the practical situation. And it is well-known that any machine has different processing ability which is taken into account in this paper. So in this paper, the highlights is that we regard processing times and release dates as uncertain variables, meanwhile, we consider the machines with different processing capacity. The objective is to minimize the expected makespan and the tardiness time of the scheduling under the processing ability constraint.

In this paper, we further study machine scheduling problem, and build a multi-objective programming model for machine scheduling problem, of which the objectives are to minimize the makespan and the maximum tardiness time under the constraint of the capacity of a certain machine. The rest of the paper is organized as follows. In Sect. 2, some basic concepts and theorems about uncertain variables are introduced. Section 3 describes the uncertain machine scheduling problem with uncertain processing times, uncertain release dates and a common determinate due date. In Sect. 4, we will build an uncertain multi-objective machine scheduling model, and transform it into a crisp programming model. After that, we will introduce

an intelligent algorithm to solve the crisp programming model in Sect. 5, and illustrate the algorithm via some numerical experiments in Sect. 6. At last, some remarks are made in Sect. 7.

2 Preliminary

In this section, some basic concepts and theorems in uncertainty theory are introduced, which are used throughout this paper.

Definition 2.1 (Liu [13]) Let \mathcal{L} be a σ -algebra on a nonempty set Γ . A set function \mathcal{M} is called an *uncertain measure* if it satisfies the following axioms:

Axiom 1. (*Normality Axiom*) $\mathcal{M}\{\Gamma\} = 1$;

Axiom 2. (*Duality Axiom*) $\mathcal{M}\{\Lambda\} + \mathcal{M}\{\Lambda^c\} = 1$ for any $\Lambda \in \mathcal{L}$;

Axiom 3. (*Subadditivity Axiom*) For every countable sequence of $\{\Lambda_i\} \in \mathcal{L}$, we have

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

The triplet $(\Gamma, \mathcal{L}, \mathcal{M})$ is called an *uncertainty space*, and each element Λ in \mathcal{L} is called an *event*. In addition, in order to obtain an uncertain measure of compound event, a product uncertain measure is defined by Liu [15] via the following product axiom:

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

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

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

Definition 2.2 (Liu [13]) An *uncertain variable* ξ is a measurable function from an uncertainty space $(\Gamma, \mathcal{L}, \mathcal{M})$ to the set of real numbers, i.e., for any Borel set B of real numbers, the set

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

is an event.

Definition 2.3 (Liu [13]) The *uncertainty distribution* Φ of an uncertain variable ξ is defined by

$$\Phi(x) = \mathcal{M}\{\xi \leq x\}, \quad \forall x \in \mathfrak{R}.$$

Definition 2.4 (Liu [16]) An uncertainty distribution $\Phi(x)$ is said to be *regular* if it is a continuous and strictly increasing function with respect to x at which $0 < \Phi(x) < 1$, and

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

In addition, the inverse function $\Phi^{-1}(\alpha)$ is called the *inverse uncertainty distribution* of ξ .

An uncertain variable ξ is said to be linear if it has a linear uncertainty distribution

$$\Phi(x) = \begin{cases} 0, & \text{if } x < a \\ (x - a)/(b - a), & \text{if } a \leq x \leq b \\ 1, & \text{if } x > b \end{cases}$$

which is denoted by $\xi \sim \mathcal{L}(a, b)$. Apparently, the linear uncertain variable ξ is regular, and has an inverse uncertainty distribution

$$\Phi^{-1}(\alpha) = \alpha(b - a) + a.$$

An uncertain variable ξ is said to be normal if it has a normal uncertainty distribution

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

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

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

Definition 2.5 (Liu [15]) The uncertain variables $\xi_1, \xi_2, \dots, \xi_n$ are said to be *independent* if

$$\mathcal{M}\left\{\bigcap_{i=1}^n (\xi_i \in B_i)\right\} = \prod_{i=1}^n \mathcal{M}\{\xi_i \in B_i\}$$

for any Borel sets B_1, B_2, \dots, B_n of real numbers.

Definition 2.6 (Liu [13]) Let ξ be an uncertain variable. The expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathcal{M}\{\xi \geq r\} dr - \int_{-\infty}^0 \mathcal{M}\{\xi \leq r\} dr$$

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

For an uncertain variable ξ with an uncertainty distribution Φ , we have

$$E[\xi] = \int_0^{+\infty} (1 - \Phi(r)) dr - \int_{-\infty}^0 \Phi(r) dr.$$

If the inverse uncertainty distribution Φ^{-1} exists, then

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

Theorem 2.1 (Liu [16]) Assume $\xi_1, \xi_2, \dots, \xi_n$ are independent uncertain variables with regular uncertainty

distributions $\Phi_1, \Phi_2, \dots, \Phi_n$, respectively. If the function $f(x_1, x_2, \dots, x_n)$ is strictly increasing with respect to x_1, x_2, \dots, x_m and strictly decreasing with respect to $x_{m+1}, x_{m+2}, \dots, x_n$, then $\xi = f(\xi_1, \xi_2, \dots, \xi_n)$ has an inverse uncertainty distribution

$$\Psi^{-1}(\alpha) = f(\Phi_1^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1 - \alpha), \dots, \Phi_n^{-1}(1 - \alpha)).$$

In addition, Liu and Ha [21] proved that the uncertain variable ξ has an expected value

$$E[\xi] = \int_0^1 f(\Phi_1^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1 - \alpha), \dots, \Phi_n^{-1}(1 - \alpha)) d\alpha.$$

Theorem 2.2 (Liu [17]) *Let $\xi_1, \xi_2, \dots, \xi_n$ be independent uncertain variables with regular uncertainty distributions $\Phi_1, \Phi_2, \dots, \Phi_n$, respectively. If the function $g(\xi_1, \xi_2, \dots, \xi_n)$ is strictly increasing with respect to $\xi_1, \xi_2, \dots, \xi_m$ and strictly decreasing with respect to $\xi_{m+1}, \xi_{m+2}, \dots, \xi_n$, then $\mathcal{M}\{g(\xi_1, \xi_2, \dots, \xi_n) \leq 0\} \geq \alpha$ holds if and only if*

$$g(\Phi_1^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1 - \alpha), \dots, \Phi_n^{-1}(1 - \alpha)) \leq 0.$$

3 Uncertain parallel machine scheduling problem

Parallel machine scheduling is used to schedule jobs processed on a series of same-function machines, with optimized objective. For example, in surgery scheduling, we regard surgery patients as jobs, and regard the doctor, anesthesiologist, nurse and surgical equipment taken as a whole as a machine. If we regard the hospital operating rooms as parallel machines, a surgery is regarded as a job which need some machines to carry on at the same time. Then surgery scheduling is the parallel machine scheduling problem. Assigning the doctors and the surgery who will perform to the operating room in order to make the latest operation as early as possible which lead to all the procedure done as soon as possible. Obviously, in this surgery scheduling, the processing times and release dates of jobs are indeterminate and it is more reasonable for doctors to estimate the processing times and release dates.

In a parallel machine scheduling problem suppose that n independent jobs have to be processed on m machines. In addition, the following hypotheses are considered.

- H_1 There is only one operation for each job and this operation can be processed on any one of the machines without interruption.
- H_2 Uncertain release dates or ready times, and a common determinate due date D may be considered for jobs.
- H_3 Each machine can process only one job at a time.
- H_4 The processing times are uncertain variables with known uncertainty distributions.

H_5 The release dates are uncertain variables with known uncertainty distributions.

We utilize the representation method proposed by Liu ([16]) via two decision vectors x and y , where

$x = (x_1, x_2, \dots, x_n)$: integer decision vector representing n jobs with $1 \leq x_i \leq n$ and $x_i \neq x_j$ for all $i \neq j$, $i, j = 1, 2, \dots, n$. That is, the sequence $\{x_1, x_2, \dots, x_n\}$ is a rearrangement of $\{1, 2, \dots, n\}$;

$y = (y_1, y_2, \dots, y_{m-1})$: integer decision vector with $y_0 \equiv 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \equiv y_m$.

Note that the schedule is fully determined by the decision vectors x and y in the following way. For each $k (1 \leq k \leq m)$, if $y_k = y_{k-1}$, then the machine k is not used, and if $y_k > y_{k-1}$, then machine k is used and processes jobs $x_{y_{k-1}+1}, x_{y_{k-1}+2}, \dots, x_{y_k}$ in turn. In other words, the schedule of all machines are as follows:

Machine 1 : $x_{y_0+1} \rightarrow x_{y_0+2} \rightarrow \dots \rightarrow x_{y_1}$;

Machine 2 : $x_{y_1+1} \rightarrow x_{y_1+2} \rightarrow \dots \rightarrow x_{y_2}$;

...

Machine m : $x_{y_{m-1}+1} \rightarrow x_{y_{m-1}+2} \rightarrow \dots \rightarrow x_{y_m}$.

Let η_i denote the uncertain release dates of jobs i , $\xi_{i,k}$ denote the uncertain processing times of jobs i on machines k and $C_i(x, y, \xi)$ denote the completion times of jobs i , $i = 1, 2, \dots, n$, respectively. For each k with $1 \leq k \leq m$, if the machine k is used, (i.e., $y_k > y_{k-1}$), we can obtain

$$C_{x_{y_{k-1}+1}}(x, y, \xi) = \eta_{x_{y_{k-1}+1}} + \xi_{x_{y_{k-1}+1}, k}$$

and

$$C_{x_{y_{k-1}+j}}(x, y, \xi) = C_{x_{y_{k-1}+j-1}}(x, y, \xi) \vee \eta_{x_{y_{k-1}+j}} + \xi_{x_{y_{k-1}+j}, k}$$

for all $2 \leq j \leq y_k - y_{k-1}$. From the recursive formula, we can derive the value $C_{x_{y_k}}(x, y, \xi)$ which is the time that the machine k finishes all the jobs assigned to it. Thus the makespan of the schedule of (x, y) is determined by

$$f(x, y, \xi) = \max_{1 \leq k \leq m} C_{x_{y_k}}(x, y, \xi).$$

Thus the tardiness time of the scheduling is $(f(x, y, \xi) - D)^+$ where D is the common determinate due date of the scheduling.

4 Uncertain multi-objective machine scheduling model

There is a common situation in real applications where there are parallel machines with different capabilities. For simplicity, assume that the machine k_0 is supposed to stop working before time T_0 with a confidence level α_0 . Then we have the constraint

$$\mathcal{M}\{C_{x_{y_{k_0}}}(\mathbf{x}, \mathbf{y}, \xi) \leq T_0\} \geq \alpha_0.$$

In order to minimize the expected makespan and the tardiness time of the scheduling under the above constraint, we construct the following uncertain multi-objective machine scheduling model,

$$\begin{cases} \min_{\mathbf{x}, \mathbf{y}} (E[f(\mathbf{x}, \mathbf{y}, \xi)], E[(f(\mathbf{x}, \mathbf{y}, \xi) - D)^+]) \\ \text{subject to :} \\ \mathcal{M}\{C_{x_{y_{k_0}}}(\mathbf{x}, \mathbf{y}, \xi) \leq T_0\} \geq \alpha_0 \\ 1 \leq x_i \leq n, x_i \neq y_j, i \neq j, i, j = 1, 2, \dots, n \\ 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \\ x_i, y_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1, \text{ integers} \end{cases} \quad (1)$$

where $\alpha_0 \in [0, 1]$ is a predetermined confidence level.

Let $\Phi_{i,k}$ denote the uncertainty distribution of $\xi_{i,k}$ and Ψ_i denote the uncertainty distribution of η_i . In the premise of using the machine k , (i.e., $y_k > y_{k-1}$), $C_{x_{y_{k-1}+1}}(\mathbf{x}, \mathbf{y}, \xi) = \eta_{x_{y_{k-1}+1}} + \xi_{x_{y_{k-1}+1},k}$ which is the completion time of job $x_{y_{k-1}+1}$, is an increasing function with respect to $\eta_{x_{y_{k-1}+1}}$ and $\xi_{x_{y_{k-1}+1},k}$. By Theorem 2.1, it has an inverse uncertainty distribution $\Theta_{x_{y_{k-1}+1}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha)$ via

$$\Theta_{x_{y_{k-1}+1}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha) = \Psi_{x_{y_{k-1}+1}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha) + \Phi_{x_{y_{k-1}+1},k}^{-1}(\mathbf{x}, \mathbf{y}, \alpha).$$

Similarly, $C_{x_{y_{k-1}+j}}(\mathbf{x}, \mathbf{y}, \xi) = C_{x_{y_{k-1}+j-1}}(\mathbf{x}, \mathbf{y}, \xi) \vee \eta_{x_{y_{k-1}+j}} + \xi_{x_{y_{k-1}+j},k}$, the completion time of job $x_{y_{k-1}+j}$, has an inverse uncertainty distribution

$$\Theta_{x_{y_{k-1}+j}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha) = \Theta_{x_{y_{k-1}+j-1}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha) \vee \Psi_{x_{y_{k-1}+j}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha) + \Phi_{x_{y_{k-1}+j},k}^{-1}(\mathbf{x}, \mathbf{y}, \alpha)$$

for all $2 \leq j \leq y_k - y_{k-1}$. In fact, we can derive the inverse uncertainty distribution $\Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha)$ of the makespan $f(\mathbf{x}, \mathbf{y}, \xi)$ via

$$\Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) = \max_{1 \leq k \leq m} \Theta_{x_{y_k}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha).$$

It follows from Liu and Ha [21] that

$$E[f(\mathbf{x}, \mathbf{y}, \xi)] = \int_0^1 \Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) d\alpha$$

and

$$E[(f(\mathbf{x}, \mathbf{y}, \xi) - D)^+] = \int_0^1 [\Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) - D]^+ d\alpha.$$

Besides, by Theorem 2.2, the constraint $\mathcal{M}\{C_{x_{y_{k_0}}}(\mathbf{x}, \mathbf{y}, \xi) \leq T_0\} \geq \alpha_0$ is equivalent to

$$\Theta_{x_{y_{k_0}}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha_0) \leq T_0.$$

So the uncertain multi-objective machine scheduling model (1) is equivalent to

$$\begin{cases} \min_{\mathbf{x}, \mathbf{y}} \left(\int_0^1 \Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) d\alpha, \int_0^1 [\Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) - D]^+ d\alpha \right) \\ \text{subject to :} \\ \Theta_{x_{y_{k_0}}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha_0) \leq T_0 \\ 1 \leq x_i \leq n, x_i \neq y_j, i \neq j, i, j = 1, 2, \dots, n \\ 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \\ x_i, y_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1, \text{ integers.} \end{cases} \quad (2)$$

In order to solve the crisp model (2), we employ the weighted sum method, and solve the following programming model

$$\begin{cases} \min_{\mathbf{x}, \mathbf{y}} \left(\omega_1 \int_0^1 \Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) d\alpha + \omega_2 \int_0^1 [\Upsilon^{-1}(\mathbf{x}, \mathbf{y}, \alpha) - D]^+ d\alpha \right) \\ \text{subject to :} \\ \Theta_{x_{y_{k_0}}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha_0) \leq T_0 \\ 1 \leq x_i \leq n, x_i \neq y_j, i \neq j, i, j = 1, 2, \dots, n \\ 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \\ x_i, y_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1, \text{ integers} \end{cases} \quad (3)$$

where ω_1 and ω_2 are the weights of the first and second objectives, respectively.

5 Hybrid intelligent algorithm

In this section, we employ a genetic algorithm to solve the crisp programming model (3), which is performed in the following steps.

Step 1: Representation

In an uncertain machine scheduling problem, we use a vector $V = (\mathbf{x}, \mathbf{y}) = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_{m-1})$ as a chromosome to represent a solution, where $\mathbf{x} = (x_1, x_2, \dots, x_n)$, representing n jobs, is a rearrangement of $\{1, 2, \dots, n\}$, and $\mathbf{y} = (y_1, y_2, \dots, y_{m-1})$ satisfying $0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n$ represents an assignment of n jobs on m machines.

Step 2: Initialization

Give the population size of one generation pop_size and the confidence level α_0 in advance. Generate (\mathbf{x}, \mathbf{y}) randomly, and obtain a vector $(\mathbf{x}, \mathbf{y}) = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_{m-1})$. Calculate $\Theta_{x_{y_{k_0}}}^{-1}(\mathbf{x}, \mathbf{y}, \alpha_0)$ and verify whether it is less than T_0 or not. If so, the vector (\mathbf{x}, \mathbf{y}) is feasible, and we get a chromosome. Otherwise, regenerate another vector (\mathbf{x}, \mathbf{y}) . Repeat this process for pop_size times, and we get pop_size chromosomes $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_{pop_size}, \mathbf{y}_{pop_size})$.

Step 3: Evaluation

Give the weights ω_1 and ω_2 in advance. For each chromosome (x, y) , calculate the objective values

$$\omega_1 \int_0^1 \Upsilon^{-1}(x, y, \alpha) d\alpha + \omega_2 \int_0^1 [\Upsilon^{-1}(x, y, \alpha) - D]^+ d\alpha.$$

Rearrange $(x_1, y_1), (x_2, y_2), \dots, (x_{pop_size}, y_{pop_size})$ according to their objective values in an ascending order, and denote them by $(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_{pop_size}, y'_{pop_size})$. Then the fitness of the chromosome (x'_i, y'_i) is $f_i = a(1 - a)^{i-1}$ for $i = 1, 2, \dots, pop_size$, where $a \in [0, 1]$ is a given parameter. Calculate the total fitness $F = \sum_{i=1}^{pop_size} f_i$ and further the evaluation function is $Eval(x'_i, y'_i) = \frac{f_i}{F}$.

Step 4: Selection

Calculate the cumulative probability q_k for the k th chromosomes (x'_k, y'_k) by $q_k = \sum_{i=1}^k Eval(x'_i, y'_i)$, $k = 1, 2, \dots, pop_size$, and set $q_0 = 0$. Generate a random number $u \in (0, 1]$ and select the chromosome (x'_k, y'_k) if $q_{k-1} < u \leq q_k$. Repeat the process for pop_size times to get a new generation of the chromosomes.

Step 5: Crossover

Give a crossover probability P_c in advance. For each chromosome (x_i, y_i) , generate a random number $u_i \in [0, 1]$, $i = 1, 2, \dots, pop_size$. If $u_i \leq P_c$, then (x_i, y_i) is selected as a parent for crossover operation. Divide all the selected chromosomes into some groups such that each group contains only two chromosomes and perform crossover operations on these groups. Without loss of generality, suppose two chromosomes $V_1 = (x_1, y_1)$ and $V_2 = (x_2, y_2)$. And their children chromosomes are $V'_1 = (x_1, y_2)$ and $V'_2 = (x_2, y_1)$. If these two children are feasible, then we replace the parents with them. Otherwise, we use the original chromosomes $V_1 = (x_1, y_1)$ and $V_2 = (x_2, y_2)$.

Step 6: Mutation

Give a mutation probability P_m in advance. For each chromosome (x_i, y_i) , generate a random number $u_i \in [0, 1]$, $i = 1, 2, \dots, pop_size$. If $u_i \leq P_m$, then (x_i, y_i) is selected for mutation operation. For a chosen chromosome $V = (x, y)$, as for the gene x , generate two different integers n_1 and n_2 between 0 and n randomly. Replace n_1 and n_2 and obtain a new gene. As for the gene y , generate two integers k_1 and k_2 between 1 and $m - 1$ randomly, Without loss of generality, suppose $k_1 < k_2$. Generate randomly $k_2 - k_1 + 1$ integers, sort the sequence in ascending order, get $y'_{k_1}, y'_{k_1+1}, \dots, y'_{k_2}$ replacing $y_{k_1}, y_{k_1+1}, \dots, y_{k_2}$ and obtain a new gene. If it is feasible, then replace the chromosome with the new one. Otherwise, regenerate n_1, n_2, k_1 and k_2 until the new one is feasible.

Step 7: Repetition

Repeat Steps 3–6 for a number of cycles, and select the best chromosome as the optimal solution for model (3).

6 Experimental illustration

A numerical example is covered in this section to illustrate the feasibility and effectiveness of the designed hybrid algorithm. The numerical experiments are performed on a personal computer, and the parameters of genetic algorithm include the population size pop_size , the probability of crossover P_c , the probability of mutation P_m and the parameter a in the evaluation function. Assume 12 jobs will be processed on 4 machines with corresponding parameters of the programming model (3) are shown in Table 1.

In addition, assume machine $k_0 = 2$ tends to stop working before $T_0 = 20$ with a confidence level $\alpha_0 = 0.9$. The common determinate due date is $D = 30$ and the weights of the objectives are $\omega_1 = 1, \omega_2 = 100$ which indicate that more attention is paid to minimize the tardiness time. Then the crisp model (3) is

$$\left\{ \begin{array}{l} \min_{x,y} \left(\int_0^1 \Upsilon^{-1}(x,y,\alpha) d\alpha + 100 \int_0^1 [\Upsilon^{-1}(x,y,\alpha) - 30]^+ d\alpha \right) \\ \text{subject to:} \\ \Theta_{x,y_2}^{-1}(x,y,0.9) \leq 20 \\ 1 \leq x_i \leq n, x_i \neq y_j, i \neq j, i, j = 1, 2, \dots, n \\ 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \\ x_i, y_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1, \text{ integers.} \end{array} \right. \quad (4)$$

In view of identification of parameters’s influence on solution quality, we compare solution by careful variation of parameters of genetic algorithm with the same generation as a stopping rule. The computational results are collected in Table 2, where the parameters of genetic algorithm are given in table from the first column to the fourth column. ‘gen’ in fifth column is the generation in

Table 1 Jobs, uncertain processing times and uncertain release dates

Job i	$\Phi_{i,1}$	$\Phi_{i,2}$	$\Phi_{i,3}$	$\Phi_{i,4}$	Ψ_i
Job 1	$\mathcal{L}(2,3)$	$\mathcal{L}(3,5)$	$\mathcal{L}(2,4)$	$\mathcal{L}(3,4)$	$\mathcal{N}(2,1)$
Job 2	$\mathcal{L}(4,6)$	$\mathcal{L}(4,8)$	$\mathcal{L}(5,8)$	$\mathcal{L}(6,8)$	$\mathcal{N}(3,1)$
Job 3	$\mathcal{L}(6,9)$	$\mathcal{L}(7,9)$	$\mathcal{L}(7,8)$	$\mathcal{L}(6,8)$	$\mathcal{N}(2,0.5)$
Job 4	$\mathcal{L}(3,5)$	$\mathcal{L}(5,7)$	$\mathcal{L}(4,6)$	$\mathcal{L}(2,5)$	$\mathcal{N}(4,1)$
Job 5	$\mathcal{L}(7,11)$	$\mathcal{L}(10,13)$	$\mathcal{L}(9,12)$	$\mathcal{L}(8,12)$	$\mathcal{N}(3,0.5)$
Job 6	$\mathcal{L}(5,8)$	$\mathcal{L}(6,9)$	$\mathcal{L}(6,7)$	$\mathcal{L}(6,8)$	$\mathcal{N}(4,1)$
Job 7	$\mathcal{L}(1,5)$	$\mathcal{L}(3,4)$	$\mathcal{L}(2,4)$	$\mathcal{L}(3,5)$	$\mathcal{N}(2,1)$
Job 8	$\mathcal{L}(5,7)$	$\mathcal{L}(4,7)$	$\mathcal{L}(5,8)$	$\mathcal{L}(6,8)$	$\mathcal{N}(2,0.5)$
Job 9	$\mathcal{L}(8,11)$	$\mathcal{L}(8,12)$	$\mathcal{L}(9,12)$	$\mathcal{L}(10,13)$	$\mathcal{N}(1,0.5)$
Job 10	$\mathcal{L}(3,7)$	$\mathcal{L}(4,8)$	$\mathcal{L}(5,7)$	$\mathcal{L}(4,7)$	$\mathcal{N}(5,1)$
Job 11	$\mathcal{L}(2,6)$	$\mathcal{L}(4,6)$	$\mathcal{L}(3,5)$	$\mathcal{L}(3,6)$	$\mathcal{N}(1,1)$
Job 12	$\mathcal{L}(4,8)$	$\mathcal{L}(5,9)$	$\mathcal{L}(4,7)$	$\mathcal{L}(5,8)$	$\mathcal{N}(6,1)$

Table 2 Compare solutions

pop_size	P _c	P _m	a	gen	Objective schedule	Objective value	Relative error (%)
30	0.4	0.1	0.05	500	(8,12,9,11,10,7,6,2,4,5,1,3; 3,5,9)	23.8725	1.68
30	0.5	0.05	0.06	500	(3,5,1,9,8,2,12,4,7,6,10,11; 3,5,8)	24.0551	2.46
30	0.5	0.1	0.07	500	(3,5,10,11,9,7,8,1,6,12,2,4; 3,5,9)	23.9404	1.97
40	0.4	0.1	0.06	500	(9,2,12,1,11,8,5,3,7,10,4,6; 4,6,8)	24.4399	4.10
40	0.4	0.2	0.05	500	(9,4,5,2,8,1,11,6,12,10,7,3; 3,5,9)	23.4765	0
40	0.5	0.1	0.07	500	(8,2,4,11,5,1,7,3,6,9,12,10; 4,5,9)	24.4514	4.15

Table 3 Optimal schedule

Machines	Schedule
Machine 1	9 → 4 → 5
Machine 2	2 → 8
Machine 3	1 → 11 → 6 → 12
Machine 4	10 → 7 → 3

Table 4 The objective values under different weights

ω ₁	ω ₂	Optimal schedule	Objective value
0	100	(5,3,2,9,10,6,1,7,8,12,11,4; 3,4,8)	1.050
1	100	(9,4,5,2,8,1,11,6,12,10,7,3; 3,5,9)	23.4765
50	50	(11,5,9,4,8,10,1,3,7,6,12,3; 3,5,9)	1232.2
100	1	(5,1,11,7,3,8,4,2,12,9,10,6; 4,6,9)	2497.5
100	0	(1,3,7,2,8,4,9,10,12,11,5,6; 4,6,9)	2130

genetic algorithm, and ‘objective schedule’, ‘objective value’ are provided in the sixth column and seventh column respectively. In addition, the parameter ‘relative error’ in the last column is defined as

$$(\text{objective value} - \text{optimal value}) / \text{optimal value} \times 100\%$$

where the optimal value is the least one of the six objective values in the seventh column.

From Table 2, we can see that the relative error does not exceed 4.15 % when various parameters of genetic algorithm are selected, which implies the hybrid algorithm is effective to solve model (4). We can obtain the optimal value in the fifth line of Table 2, and optimal schedule is displayed in Table 3.

Since there are two parameters ω₁, ω₂ in objective function in model (3), we had better analysis the relationship between the objective value and the two parameters. In the following, we fix pop_size = 40, P_c = 0.4, P_m = 0.2 and a = 0.05. We consider one case as under different weights and fix other parameters as in the model (4) and the result of the objective value is displayed in Table 4. It can be seen from Table 4 that different weights affect the

Table 5 The objective values under different α₀ in model (5)

α ₀	0.6	0.7	0.8	0.85	0.9	0.95
Objective value	0	0	0.0014	0.0048	0.0105	0.0439

objective values. In the experimental environment, the objective values are not monotone.

In real life, once attention is paid only to reduce the losses caused by tardiness, we assume ω₁ = 0 and ω₂ = 1. Then the model (3) is changed into

$$\left\{ \begin{array}{l} \min_{x,y} \int_0^1 [\Upsilon^{-1}(x,y,\alpha) - D]^+ d\alpha \\ \text{subject to :} \\ \Theta_{xyk_0}^{-1}(x,y,\alpha_0) \leq T_0 \\ 1 \leq x_i \leq n, x_i \neq y_j, i \neq j, i, j = 1, 2, \dots, n \\ 0 \leq y_1 \leq y_2 \leq \dots \leq y_{m-1} \leq n \\ x_i, y_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m - 1, \text{ integers.} \end{array} \right. \quad (5)$$

In the model (5), we still employ the processing times and release dates showed in Table 1 and set k₀ = 2, T₀ = 20, and D = 30. Examine the sensitivity of confidence level α₀ in the constraint Θ_{xy₂}⁻¹(x, y, α₀) ≤ 20 and the result is displayed in Table 5. As is seen from Table 5, the objective values are increasing with respect to the confidence level α₀.

7 Conclusions

This paper proposed an uncertain machine scheduling problem, of which the release dates and processing times were described as uncertain variables with known uncertainty distributions. An uncertain multi-objective programming model was built for the problem, where the objectives are to minimize the makespan and the tardiness time of the scheduling simultaneously. The proposed uncertain machine scheduling model was transformed into a crisp equivalence by assigning different weights. An intelligent algorithm was designed to solve the crisp

equivalence, and was illustrated via some numerical experiments.

Acknowledgments This work is supported by Natural Science Foundation of Shandong Province (ZR2014GL002).

References

- Ahmadizar F, Ghazanfari M, Ghomi SM (2010) Group shops scheduling with makespan criterion subject to random release dates and processing times. *Comput Oper Res* 37:152–162
- Baker KR, Scudder GD (1990) Sequencing with earliness and tardiness penalties: a review. *Oper Res* 38:22–36
- Banerjee BP (1965) Single facility sequencing with random execution times. *Oper Res* 13(3):358–364
- Chen XW (2011) American option pricing formula for uncertain financial market. *Int J Oper Res* 8(2):32–37
- Gharehgozli AH, Moghaddam TR, Zaerpour N (2009) A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates. *Robot Comput Integr Manuf* 25:853–859
- Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann Discrete Math* 5:287–326
- Hodegson TJ (1977) A note on single machine sequencing with random processing times. *Manag Sci* 23(10):1144–1146
- Ishibuchi H, Yamamoto N, Murata T, Tanaka H (1994) Genetic algorithms and neighborhood search algorithms for fuzzy flow-shop scheduling problems. *Fuzzy Sets Syst* 67:81–100
- Johnson SM (1954) Optimal two and three-stage production schedules with setup times included. *Nav Res Logist Q* 1:61–68
- Kahneman D, Tversky A (1979) Prospect theory: an analysis of decision under risk. *Econometrica* 47(2):263–292
- Lam K, Xing WX (1997) New trends in parallel machine scheduling. *Int J Oper Prod Manag* 17(3):326–338
- Lenstra JK, Rinnooy Kan AHG, Brucker P (1977) Complexity of machine scheduling problems. *Ann Discrete Math* 1:343–362
- Liu B (2007) *Uncertainty theory*, 2nd edn. Springer-Verlag, Berlin
- Liu B (2009) *Theory and practice of uncertain programming*, 2nd edn. Springer-Verlag, Berlin
- Liu B (2009) Some research problems in uncertainty theory. *J Uncertain Syst* 3(1):3–10
- Liu B (2010) *Uncertainty theory: a branch of mathematics for modeling human uncertainty*. Springer-Verlag, Berlin
- Liu B (2015) *Uncertainty theory*, 5th edn. Springer-Verlag, Berlin
- Liu B, Chen X (2015) Uncertain multiobjective programming and uncertain goal programming. *J Uncertain Anal Appl* 3:10. doi:10.1186/s40467-015-0036-6
- Liu B, Yao K (2015) Uncertain multilevel programming: algorithm and applications. *Comput Ind Eng* 89:235–240
- Li RY, Liu G An uncertain goal programming model for machine scheduling problem. *J Intell Manuf*. doi:10.1007/s10845-014-0982-8
- Liu YH, Ha MH (2010) Expected value of function of uncertain variables. *J Uncertain Syst* 4(3):181–186
- Mokotoff E (2001) Parallel machine scheduling problems: a survey. *Asia Pac J Oper Res* 18:193–242
- Naughton MC (1959) Scheduling with deadlines and loss functions. *Manag Sci* 6(1):1–12
- Peng J, Liu B (2004) Parallel machine scheduling models with fuzzy processing times. *Inf Sci* 166:49–66
- Petrovic S, Fayad C (2004) A fuzzy shifting bottleneck hybridised with genetic algorithm for real-world job shop scheduling. In: *Proceedings of Mini-Euro Conference, managing uncertainty in decision support models*, Coimbra, Portugal, pp 1–6
- Pinedo M (1983) Stochastic scheduling with release dates and due dates. *Oper Res* 31:559–572
- Pinedo M (2008) *Scheduling: theory, algorithms and systems*. Springer, New York
- Prade H (1979) Using fuzzy set theory in a scheduling problem: a case study. *Fuzzy Sets Syst* 2:153–165
- Seo DK, Klein CM, Jang W (2005) Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Comput Ind Eng* 48:153–161
- Wang X, Ning YF, Moughal TA, Chen XM (2015) Adams–Simpson method for solving uncertain differential equation. *Appl Math Comput* 271:209–219
- Wu XL, Lan YF, Liu HT (2014) Optimal revenue-sharing contract based on forecasting effort for uncertain agency problem. *Int J Mach Learn Cybern* 5(6):971–979
- Xing LN, Chen YW, Yang KW (2009) An efficient search method for multi-objective flexible job shop scheduling problems. *J Intell Manuf* 20:283–293
- Yang K, Zhao RQ, Lan YF (2014) The impact of risk attitude in new product development under dual information asymmetry. *Comput Ind Eng* 76(10):122–137
- Yao K (2015) Uncertain differential equation with jumps. *Soft Comput* 19(7):2063–2069
- Zhang XF, Meng GW (2013) Expected-variance-entropy model for uncertain parallel machine scheduling. *Inf Int Interdiscip J* 16(2(A)):903–908