Expert Systems with Applications 39 (2012) 11276–11283

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Study on solution models and methods for the fuzzy assignment problems

Fachao Li [a], Li Da Xu [b,*], Chenxia Jin [a], Hong Wang [c]

[a] School of Economics and Management, Hebei University of Science and Technology, Shijiazhuang, Hebei 050018, China
[b] Department of Information Technology and Decision Science, Old Dominion University, Norfolk, VA 23529, USA
[c] School of Business and Economics, North Carolina A&T State University, Greensboro, NC 27411, USA

ARTICLE INFO

Keywords:
Fuzzy assignment problem
Level effect function
I_L-metric
U_L-dispersion
Genetic algorithm
Markov chain

ABSTRACT

In this study, commercing from the structural characteristics of fuzzy information, we propose the concept of level effect function, which can be used to describe fuzziness consciousness and to establish an I_L-metric method to measure all aspects of fuzzy information; further, we present an uncertainty metric model of concentrated quantification value; then, we establish two kinds of solution models based on the synthesizing effect of fuzzy assignment problems, by combining the genetic algorithm and assignment problems, and describe a concrete implementation strategy and algorithm to fuzzy assignment problem (denoted by GA⊕SE-FAM, for short); finally, we consider the algorithm's convergence using Markov chain theory, and analyze its performance through simulation of practical examples. All of these indicate that this algorithm possesses the advantages of higher feasibility and easier operationalization, as such, it can be widely used in many fuzzy assignment problems.

1. Introduction

Assignment problems in manufacturing and management processes were first studied in a 1952 paper. It is generally believed that this was the beginning of the study on classical assignment problems. An assignment problem is a typical optimization problem in which the objective is to assign n jobs to n persons at a minimum (maximum) time (benefit). Today, the most popular and widely used method for this problem is the Hungarian algorithm proposed by Kuhn (1955), but the Hungarian algorithm cannot directly solve all variations of assignment problems under different conditions, such as how to equally assign tasks to persons if the resources each person needs is limited, how to equally assign tasks to persons so that the maximum of the assignment costs could be minimized, with the condition that all the tasks must be finished smoothly, and so on. In practice, we often encounter problems similar to these. Therefore, many authors have proposed solutions for generalized assignment problems, bottleneck assignment problems and quadratic assignment problems among others from different angles (Burkard, 1984; Gross, 1959; Tang, Lim, Ong, & Er, 2006; Yang, Lu, Li, & Liao, 2008). Pentico has provided a comprehensive survey of different variations of the assignment problem that have appeared in the literature over the past 50 years (Pentico, 2007). Meanwhile, many different solution methods have been developed for those variations of assignment problems by other researchers in the past decades (Demirel & Toksan, 2006; Kumar, 2006). It is worth noting that all of the above works have been conducted for and focused on quantitative assignment problems. In reality, the benefit of each person by finishing the assigned task is difficult to be measured by exact quantities, but they could often be described in fuzzy numbers. Therefore we regard this type of assignment problem as fuzzy assignment problems. These problems have already been studied by some researchers in the past (Feng & Yang, 2006; Majumdar & Bhunia, 2007; Song, Wu, & Chen, 2001). They all transformed fuzzy assignment problems into traditional assignment problems and then established assignment scheme using the Hungarian algorithm. However, all of these solution methods have their weaknesses and cannot perfectly reflect the decision consciousness in the assignment process.

In view of these shortcomings, our main contributions in this study are as follows: (a) By using the structural characteristics of fuzzy information, we propose a level effect function that can describe fuzziness consciousness, and establish an I_L-metric to measure all aspects of fuzzy information. We also provide an uncertainty metric model of concentrated quantification value and discuss the relative operational properties of the I_L-metric; (b) We propose the concept of synthesizing effect functions which provides a theoretical basis for the systematic consideration of the centralized quantification value and the uncertainty of fuzzy information; (c) We establish two kinds of solution models to the fuzzy assignment problems by combining the genetic algorithm and assignment problems, and we describe a concrete implementation strategy and algorithm to solve fuzzy assignment problems; (d) We consider the convergence of the algorithm using Markov chain theory and analyze its performance through concrete practical examples. Our study demonstrates that this kind of algorithm not only reflects

* Corresponding author. Tel.: +1 757 6836138.
E-mail address: lidxu@hotmail.com (L.D. Xu).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.04.034

the decision consciousness effectively, but also has better convergence and higher computation efficiency.

2. Preliminaries

First of all, we introduce a set of definitions for the problem. Let R be the real number field and $F(R)$ the family of all fuzzy sets over R. For any $A \in F(R)$, the membership function of A is written as $A(x)$, the λ-cuts of A as $A_\lambda = \{x|A(x) \geqslant \lambda\}$, and the support set of A as sup - p$A = \{x|A(x) > 0\}$. The definitions of a fuzzy number and its basic operational properties are given as follows.

Definition 1. Goetschel and Voxman (1983) $A \in F(R)$ is called a fuzzy number if it satisfies the following conditions: (1) For any given $\lambda \in (0,1]$, A_λ are closed intervals; (2) $A_1 = \{x|A(x) = 1\} \neq \varphi$; (3) supp$A$ is bounded. The class of all fuzzy numbers is called fuzzy number space, which is denoted by E^1. In particular, if there exists a, b, $c \in$ R such that $A(x) = (x - a)/(b - a)$ for each $x \in [a,b)$, and $A(b) = 1$, and $A(x) = (x - c)/(b - c)$ for each $x \in (b, c]$, and $A(x) = 0$ for each $x \in (-\infty,a) \cup (c,+\infty)$, then we say that A is a triangular fuzzy number, written as $A = (a,b,c)$ for short.

For $A \in E^1$, it is easy to see that the closure of supp A is a closed interval. In what follows, we denote the closure of supp$A$ by A_0. By Definition 1, we can prove that $A_\lambda = [a + (b - a)\lambda, c - (c - b)\lambda]$ for any $A = (a,b,c)$ and $\lambda \in [0,1]$.

Obviously, if we regard real number a as a fuzzy set with membership function as $a(x) = 1$ for $x = a$ and $a(x) = 0$ for each $x \neq a$, then fuzzy numbers can be thought of as an extension of real numbers, so fuzzy numbers possess the properties of both numbers and sets, which is the broadest description of fuzzy information in many practical domains.

Fuzzy numbers have many good analytical properties, for the arithmetic operations, we draw the following conclusion:

Theorem 1 (Diamond and Kloeden (1994)). *Let A, $B \in E^1$, $k \in R$, $f(x,y)$ be a continuous binary function and A_λ, B_λ be the λ-cuts of A and B, respectively. Then $f(A,B) \in E^1$, and for any $\lambda \in (0,1]$, we have $(f(A,B))_\lambda = f(A_\lambda,B_\lambda)$. In particular, we have the following: (1) $A + B = B + A$, $A \cdot B = B \cdot A$, $k(A \pm B) = kA \pm kB$; (2) For $A = (a_1,b_1,c_1)$, $B = (a_2,b_2,c_2)$, $A + B = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$, and $A - B = (a_1 - c_2, b_1 - b_2, c_1 - a_2)$, $kA = (ka_1,kb_1,kc_1)$ for any $k \geqslant 0$, and $kA = (kc_1,kb_1,ka_1)$ for any $k < 0$. Here, $f(A_\lambda,B_\lambda) = \{f(x,y)|x \in A_\lambda, y \in B_\lambda\}$.*

3. Centralized quantification description of fuzzy numbers

3.1. I_L-metric of fuzzy numbers and its properties

The decomposition theorem of fuzzy sets provides us with a basic method to realize and process fuzzy information. But in many real world problems, we often depend on the global attributes of fuzzy information to make the decision. It is easy to see that the individuals with different membership characteristics play different roles during the process of decision-making. To establish a general theoretical model for this problem, we introduce the concept of level effect function.

Definition 2. We call $L(\lambda)$: $[0,1] \to [a,b] \subset [0,\infty)$ a level effect function if $L(\lambda)$ is piecewise continuous and monotone non-decreasing. For $A \in E^1$, let

$$I_L(A) = \frac{1}{2L^*}\int_0^1 L(\lambda)(\underline{a}(\lambda) + \bar{a}(\lambda))\mathrm{d}\lambda. \tag{1}$$

Here, $A_\lambda = [\underline{a}(\lambda), \bar{a}(\lambda)]$ is the λ- cuts of A, $L^* = \int_0^1 L(\lambda)\mathrm{d}\lambda$. $I_L(A)$ is called the I_L-metric of A, and when $L^* = 0$, we define $I_L(A) = [\underline{a}(1) + \bar{a}(1)]/2$.

In Definition 2, if we interpret the level effect function as the description for various levels of confidence degrees of information, A_λ as the intrinsic information of A and $L(\lambda)$ as a kind of decision parameter, then $I_L(A)$ is just a kind of synthesizing average value of A under $L(\lambda)$. Obviously, by the I_L-metric, we can establish an order relation on E^1, and we denote it by (E^1, I_L).

Definition 3. Let A, $B \in E^1$. If $I_L(A) < I_L(B)$, then we say A is less than B with respect to the I_L-metric, and it is written as $A < B$; If $I_L(A) = I_L(B)$, then we say A is equal to B with respect to the I_L-metric, and it is written as $A = B$; If $I_L(A) \leqslant I_L(B)$, then we say A is no more than B with respect to the I_L-metric, and it is written as $A \leqslant B$.

Remark 1. Order structure (E^1, I_L) provides a kind of model for describing the sequencial attributes of fuzzy information, and it generalizes the existing ranking methods of fuzzy numbers, so it has strong interpretability and operability. For example, (E^1, I_L) keeps the order relation $\leqslant_1$ defined by level cuts of fuzzy numbers (here $A \leqslant_1 B \iff A_\lambda \leqslant B_\lambda$ for each $\lambda \in [0,1]$, and $[a,b] \leq [c,d] \iff a \leq c, b \leq d$), that is, $I_L(A) \leqslant I_L(B)$) if $A \leqslant_1 B$; When $L(\lambda) \equiv 1$, (E^1, I_L) just coincides with the order relation proposed in Liou and Wang (1992).

Theorem 2. *Let A, $B \in E^1$, $k \in R$. Then*: (1) $I_L(A \pm B) = I_L(A) \pm I_L(B)$; (2) $I_L(kA) = kI_L(A)$

Proof. Let $A_\lambda = [\underline{a}(\lambda), \bar{a}(\lambda)], B_\lambda = [\underline{b}(\lambda), \bar{b}(\lambda)]$ be the λ-cuts of A, B respectively. By using the properties (Diamond & Kloeden, 1994) of fuzzy numbers, we have $(A+B)_\lambda = [\underline{a}(\lambda) + \underline{b}(\lambda), \bar{a}(\lambda) + \bar{b}(\lambda)]$ and $(A-B)_\lambda = [\underline{a}(\lambda) - \bar{b}(\lambda), \bar{a}(\lambda) - \underline{b}(\lambda)]$ for each $\lambda \in [0,1]$, $(kA)_\lambda = [k\underline{a}(\lambda), k\bar{a}(\lambda)]$ for each $\lambda \in [0,1]$ and $k \geqslant 0$, and $(kA)_\lambda = [k\bar{a}(\lambda), k\underline{a}(\lambda)]$ for each $\lambda \in [0,1]$ and $k < 0$. From this and the properties of the Lebesgue integral, we have:

$$\begin{aligned}I_L(A+B) &= \frac{1}{2L^*}\int_0^1 L(\lambda)\left[\underline{a}(\lambda) + \underline{b}(\lambda) + \bar{a}(\lambda) + \bar{b}(\lambda)\right]\mathrm{d}\lambda\\ &= \frac{1}{2L^*}\int_0^1 L(\lambda)[\underline{a}(\lambda) + \bar{a}(\lambda)]\mathrm{d}\lambda + \frac{1}{2L^*}\int_0^1 L(\lambda)\left[\underline{b}(\lambda) + \bar{b}(\lambda)\right]\mathrm{d}\lambda\\ &= I_L(A) + I_L(B);\end{aligned}$$

$$\begin{aligned}I_L(A-B) &= \frac{1}{2L^*}\int_0^1 L(\lambda)\left[\underline{a}(\lambda) - \bar{b}(\lambda) + \bar{a}(\lambda) - \underline{b}(\lambda)\right]\mathrm{d}\lambda\\ &= \frac{1}{2L^*}\int_0^1 L(\lambda)[\underline{a}(\lambda) + \bar{a}(\lambda)]\mathrm{d}\lambda - \frac{1}{2L^*}\int_0^1 L(\lambda)\left[\underline{b}(\lambda) + \bar{b}(\lambda)\right]\mathrm{d}\lambda\\ &= I_L(A) - I_L(B);\end{aligned}$$

$$\begin{aligned}I_L(kA) &= \frac{1}{2L^*}\int_0^1 L(\lambda)[k\underline{a}(\lambda) + k\bar{a}(\lambda)]\mathrm{d}\lambda\\ &= \frac{k}{2L^*}\int_0^1 L(\lambda)[\underline{a}(\lambda) + \bar{a}(\lambda)]\mathrm{d}\lambda = kI_L(A).\end{aligned}$$

3.2. U_L-dispersion on I_L-metric

For order structure (E^1, I_L), when $I_L(A) = I_L(B)$, we cannot further compare fuzzy numbers A and B using only the I_L-metric, which means that the I_L-metric cannot completely reflect the fuzzy numbers. In regards to the decision process in practical problems, we consider not only the decision solution itself, but also the degree of reliability of the solution. In order to abstract the quantitative attributes of fuzzy information more objectively, we introduce the concept of U_L-dispersion on I_L-metric.

Definition 4. Let $A \in E^1$, $\theta \in (0,\infty)$ $A_\lambda = [\underline{a}(\lambda), \bar{a}(\lambda)]$ be the λ-cuts of A, $L(\lambda)$ be a level effect function.

For

$$U_L(A) = \int_0^1 L(\lambda)(\bar{a}(\lambda) - \underline{a}(\lambda))d\lambda, \tag{2}$$

Then $U_L(A)$ is called the U_L-dispersion on I_L-metric based on $L(\lambda)$.

For $A, B, A^{(i)} \in E^1$, $k \in R$, $i = 1,2,\ldots,n$, we have $U_L(A \pm B)\ U_L(A) + U_L(B)$, $U_L(kA) = |k|U_L(A)$, $U_L(A(1) + A(2) + \cdots + A(n)) = U_L(A(1)) + U_L(A(2)) + \cdots + U_L(A(n))$ using the concept of fuzzy numbers and the properties of integrals.

According to the theories of integrals, we know that $U_L(A)$ is just the synthetic measurement for the uncertainty of the level cuts of A, and it is also the global description of uncertainty of A under $L(\lambda)$. The smaller $U_L(A)$ is, the lower the uncertain degree of $I_L(A)$ will be; and the larger $U_L(A)$ is, the higher the uncertainty degree of $I_L(A)$ will be.

3.3. Comparison of fuzzy information based on synthesizing effect

$(I_L(A)U_L(A))$ is a compound centralized quantification method for fuzzy information A, In processing fuzzy information, I_L-metric and U_L-dispersion can restrict and supplement each other. Generally speaking, in considering maximal (or minimum) fuzzy optimization problems, decision-makers always hope that the compound quantification of the objective function is as large (or small) as possible, and that the corresponding U_L-dispersion is as small as possible at the same time, but this cannot always be satisfied in real world problems. To establish a comparison method that can simultaneously consider the I_L-metric and U_L-dispersion, we introduce the concept of a synthesizing effect function.

Definition 5. A continuous function $S(x,y)$: $(-\infty,+\infty) \times [0,+\infty) \to (-\infty,+\infty)$ is called a maximum (minimum) synthesizing effect function, if it satisfies the following: (1) $S(x,y)$; is monotone non-decreasing on x for any $y \geqslant 0$; (2) $S(x,y)$ is monotone non-increasing (non-decreasing) on y for any $x \in (-\infty,\infty)$; (3) $S(x,0) = x$.

It is easy to verify that, for any a, b, $c \in [0,+\infty)$, $S_1(x,y) = x(1 + -ay)^{-b\delta(x)}$, $S_2(x,y) = x - ay^b$, $S_3(x,y) = x(1 + a)^{-by\delta(x)}$, and $S_4(x,y) = x - c\ln(1 + ay^b)$ are all the maximum synthesizing effect functions; and $S_1^*(x,y) = x(1+ay)^{b\delta(x)}, S_2^*(x,y) = x + ay^b, S_3^*(x,y) = x(1+a)^{by\delta(x)}, S_4^*(x,y) = x + c\ln(1+ay^b)$ are all the minimum synthesizing effect functions. Here, $\delta(x) = 1$ for any $x \geqslant 0$ and $\delta(x) = -1$ for any $x < 0$.

If we regard x and y as $I_L(A)$ and $U_L(A)$ of A respectively, then $S_L(A) = S(I_L(A), U_L(A))$ is a compound quantification method for fuzzy information with consideration of both the I_L-metric and U_L-dispersion. And this method not only contains I_L-metric method, but it also has better interpretability. In practical problems, we can choose different synthesizing effect functions to embody different uncertainty consciousnesses in the decision process.

4. The solution model of fuzzy assignment problems

4.1. The transformation model of fuzzy assignment problems

Let us assign n tasks to n people, let $c_{ij} \in E^1$ be the cost (or benefit) of assigning person i to task j and $i, j = 1,2,\ldots,n$, $C = (c_{ij})_{n\times n}$ be the fuzzy efficiency matrix. Then the mathematical model of the fuzzy assignment problem can be expressed as follows:

$$\begin{cases} \min(\text{or}\max) \quad f = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}, \\ \text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \\ \qquad \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n, \\ \qquad x_{ij} = 0,1. \end{cases} \tag{3}$$

where $x_{ij} = 1$ if person i is assigned to task j, $x_{ij} = 0$ if not.

Because fuzzy numbers do not have a complete order like the real numbers do, (3) is just a formal model and cannot be easily solved by existing methods. Therefore we should convert fuzzy information into numerical values using some strategy under certain consciousness, and then execute the solvable transformation of (3).

Based on the above analysis, using the I_L-metric, U_L-dispersion and their synthesizing effect given in Section 3, we can convert model (3) into the following models (4) or (5) with some decision consciousness:

$$\begin{cases} \min(\text{or}\max)E(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} S(I_L(c_{ij}), U_L(c_{ij}))x_{ij}, \\ \text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \\ \qquad \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n, \\ \qquad x_{ij} = 0,1. \end{cases} \tag{4}$$

$$\begin{cases} \min(\text{or}\max)E(Z) = S(I_L(Z), U_L(Z)), \\ \text{s.t.} \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}, \\ \qquad \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \\ \qquad \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n, \\ \qquad x_{ij} = 0,1. \end{cases} \tag{5}$$

where $x_{ij} = 1$ if person i is assigned to task j, $x_{ij} = 0$ if not, and $S(x,y)$ is the synthesizing effect function.

The guiding principles for solving (4) and (5) are essentially different, which implies that the recognition methods of feasible solutions are different. Model (4) emphasizes the concrete efficiency, while model (5) prefers total benefit. In terms of solution methods and problem complexity, model (4) is a linear programming problem, and it can be solved by using the Hungarian algorithm; and in general cases, model (5) is a nonlinear programming problem, and it cannot be solved by any fixed method. From the above analysis, we know that model (4) is easier than model (5), but model (5) reflects reality more accurately. In the following, we mainly discuss model (5) based on the minimum assignment problem.

Theorem 2. *When $S(x,y) = x$, model (5) degenerates into model (4), and the optimal solution must be the optimal solution to model (3) with respect to the I_L-metric.*

Proof. From Theorem 1, we have, when $S(x,y) = x$

$$S_L(Z) = S(I_L(Z), U_L(Z)) = I_L(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} I_L(c_{ij})x_{ij}$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} S(I_L(c_{ij}), I_L(c_{ij}))x_{ij}$$

Then we can obtain the conclusion. □

4.2. The solution procedures for fuzzy assignment problems

Based on the above analysis, we can solve fuzzy assignment problem (3) through the following steps:

Step 1. Select level effect function $L(\lambda)$ and synthesizing effect function $S(x,y)$.
Step 2. Compute the matrix $I_L(C) = (I_L(c_{ij}))_{n\times n}$ and $U_L(C) = (U_L(c_{ij}))_{n\times n}$ by (1) and (2).
Step 3. Solve model (3) using appropriate algorithms.

5. A solution to the fuzzy assignment problem based on genetic algorithms (GA⊕SE-FAM)

The genetic algorithm is a kind of self-adaptive, global optimization search technique simulating the evolution and inheritance process of living beings in a natural environment, and it is an effective tool for solving combinatorial optimization and intelligence optimization. During the process of using the genetic algorithm, we start from an initial population, search for the optimal solution generation by generation (or satisfactory solution), until the convergence conditions are satisfied or the specified iterative time achieved. Its main operations include selection, crossover, and mutation (Chaudhry, Varano, & Xu, 2000; Jiang, Xu, Wang, & Wang, 2009; Li, Xu, Jin, & Wang, 2011a; Li, Xu, Jin, & Wang, 2011b; Wang et al., 2011; Xu, 2000).

The core contents of the genetic algorithm include encoding for parameters, setting initial population, fitness function design, genetic operations design, and control parameter design. In the following, we propose the genetic strategies for GA⊕SE-FAM.

5.1. Coding

Coding is the most basic component of the genetic algorithm. And among the numerous methods available, binary coding and floating-point coding are the most commonly used methods. Solution vector $X = (x_1, x_2, \ldots, x_n)$ denotes that person i is assigned to task x_i. In the following, the permutations of n natural numbers are taken as code scheme to the solution.

5.2. Fitness function

The fitness function is the most basic tool for evaluating the fitness of individuals for the environment in the genetic algorithm. Generally speaking, a fitness function is nonnegative, relies on the objective function, and can also effectively distinguish the quality of the solution. Therefore, in real world problems, we should combine the range of the objective function to select the appropriate fitness function. Considering the benefits of the assignment problem are nonnegative, in this study, we take

$$F(x) = [M - E(f(x))]^p \tag{6}$$

as the fitness function, and $M, p \in (0,\infty)$ are fitness calibration parameters that can be determined according to the size of the assignment problem and the characteristics of benefits; $f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$ is the fuzzy objective benefit function, $E(f(x)) = S(I_L(f(x)), U_L(f(x)))$, and $S(x, y)$ is the synthesizing effect function.

5.3. Selection operator

Selection is an important component of the genetic algorithm. This process is dependent on the evolutionary principle "survival of the fittest". In this process, the better Roulette wheel selection is used for our genetic algorithm. This is just a proportional strategy based on fitness values, with the rule that better individuals possess proportionately larger survival probabilities, and all the individuals in the population have the opportunity to be selected.

5.4. Crossover operator

Crossover is a step that really powers the genetic algorithm. It produces new individuals by combining the information contained in parents. This operation aims at representing gene inheritance, while also ensuring that the population has some diversity. The crossover operation in this study is stated as follows: (a) Select one integral m in $1, 2, \ldots, n$ to be randomly determined as the crossover point, from which the two individuals X_1, X_2 of a pair of parents are each split into two parts; (b) Take the front partial genes of X_1 directly as the partial genes of individual X_1', rearrange the back genes of X_1 according to the sequence of those same genes in parent individual X_2, then take it as the remaining genes of individual X_1'; (c) Take the front partial genes of X_2 as the partial genes of individual X_2', rearrange the back genes of X_2 according to the sequence in parent individual X_1, then take it as the remaining genes of individual X_2'.

For example, for two parent individuals X_1 = 1 2 3 4 5 6 and X_2 = 2 5 6 3 4 1, if the randomly generated crossover point is 3, then through the crossover operator, the two child individuals should be $X_1' = 1\ 2\ 3\ 5\ 6\ 4$ and $X_2' = 2\ 5\ 6\ 1\ 3\ 4$.

5.5. Mutation operator

The aim of mutation is to increase genetic diversity (Zhou & Sun, 1999) into the population by introducing random variations in the members of the population. Mutation is applied to a single chromosome with a low probability. It attempts to bump the population gently into a slightly better one. In this study, we select the mutation operation as follows:

For the mutation probability p_m, execute the following operation incrementally from the first bit representing the individual: for the randomly generated number $r \in [0, 1]$, if $r \geqslant p_m$, then the corresponding gene does not mutate, but if $r < p_m$, then the corresponding gene mutates. The mutation puts the required number at the end of the encoding string. The same process is used for the second bit, the third bit, ..., and the nth bit.

For example, for parent X = 5 6 3 1 4 2, the mutation process is as follows: starting from the first bit 5, if randomly generated number $r \geqslant p_m$, then 5 keeps its original position; else, put 5 at the end of the encoding string, that is, X' = 6 3 1 4 2 5. Then repeat the process with the second bit 6, the third bit 3, and so on until the last.

5.6. Forced reserved strategy

To maintain good quality solutions in subsequent generations, we have used a forced reserved strategy. A forced reserved strategy is a strategy that can assure the optimal solution will be reached as soon as possible. Its primary function is to keep the optimal individual and suboptimal individual to fill the last entry of the new population of the current generation. The operation procedures are stated as follows:

(a) For two parent individuals X_1 and X_2, generate X_1' and X_2' through crossover.
(b) For child individuals X_1' and X_2', generate X_1'' and X_2'' through mutation.
(c) Compare the fitness values of parent individuals X_1, X_2 with child individuals X_1'', X_2'', and reserve the two individuals with the largest and the second largest fitness values. For example, if $f(X_1) = 0.6$, $f(X_2) = 0.8$, $f(X_1'') = 0.5, f(X_2'') = 0.9$, then we keep X_2 and X_2'' as the evolutionary results of X_1 and X_2.

Table 1
Time needed for each translator to finish each task in Example 1.

	Translator 1	Translator 2	Translator 3	Translator 4	Translator 5	Translator 6	Translator 7	Translator 8	Translator 9	Translator 10
Job 1	(9,10,12)	(6,8,12)	(9,10,13)	(7,10,14)	(15,17,19)	(5,9,10)	(5,8,10)	(11,12,13)	(12,15,16)	(11,16,18)
Job 2	(16,18,20)	(14,15,18)	(12,13,18)	(12,15,17)	(9,12,17)	(6,8,10)	(6,8,10)	(12,15,18)	(12,15,17)	(12,15,17)
Job 3	(4,7,9)	(15,17,19)	(16,19,21)	(6,8,9)	(9,10,12)	(4,5,8)	(5,7,9)	(6,9,11)	(13,15,16)	(9,10,12)
Job 4	(18,20,22)	(9,11,13)	(8,9,11)	(16,18,19)	(8,10,12)	(8,9,11)	(9,11,12)	(8,9,11)	(16,18,19)	(8,10,12)
Job 5	(16,18,20)	(5,6,9)	(15,17,19)	(10,13,14)	(7,10,12)	(6,8,10)	(15,16,19)	(5,7,9)	(9,12,14)	(11,13,15)
Job 6	(16,18,20)	(15,17,19)	(12,15,17)	(8,9,11)	(11,13,15)	(6,9,11)	(5,7,9)	(11,16,18)	(13,15,16)	(12,15,18)
Job 7	(14,16,18)	(8,10,12)	(13,14,16)	(7,8,11)	(9,11,13)	(4,6,8)	(8,10,12)	(7,8,11)	(12,15,17)	(9,11,13)
Job 8	(15,17,19)	(18,20,22)	(11,14,16)	(13,15,16)	(7,9,11)	(5,7,9)	(9,11,14)	(11,12,13)	(8,9,11)	(11,13,15)
Job 9	(11,13,15)	(13,15,16)	(12,15,18)	(9,10,12)	(11,12,13)	(11,13,15)	(13,15,16)	(11,13,15)	(9,10,12)	(11,14,16)
Job 10	(8,10,12)	(16,18,19)	(11,13,15)	(16,18,19)	(11,16,18)	(8,10,12)	(16,18,19)	(7,9,11)	(16,18,19)	(13,14,16)

Table 2
Result of 10 runs in Case 1.

	S.V.	I_L – M.V.	U_L – D.	C.G.	T.C.
1	91.1167	85.1667	5.9500	21	3.0780
2	91.4400	85	6.4400	30	3.1250
3	91.1167	85.1667	5.9500	31	3.4680
4	91.1167	85.1667	5.9500	16	3.1410
5	91.1167	85.1667	5.9500	25	3.3130
6	91.1167	85.1667	5.9500	17	3.2500
7	92.1167	86.1667	5.9500	28	3.2660
8	92.1133	85.3333	6.7800	29	3.3440
9	91.1167	85.1667	5.9500	12	3.2180
10	91.4400	85	6.4400	17	3.3590
A.V.	91.2810	85.1500	6.1310	22.6	3.2562

Table 3
Results of 10 runs Case 2.

	S.V.	I_L – M.V.	U_L – D.	C.G.	T.C.
1	83.5804	84.8334	6.2800	21	3.4380
2	84.3978	85.6667	6.4400	12	3.4380
3	83.3742	84.6665	6.6800	22	3.4530
4	83.3742	84.6665	6.6800	28	3.5000
5	83.2459	84.4999	6.2900	23	3.4680
6	83.7302	85	6.4500	30	3.4690
7	83.2459	84.4999	6.2900	25	3.2650
8	83.2459	84.4999	6.2900	28	3.5000
9	83.2459	84.4999	6.2900	19	3.3740
10	83.2459	84.4999	6.2900	10	3.4690
A.V.	83.4686	84.7333	6.3980	21.8	3.4374

6. Performance analysis of GA⊕SE-FAM

6.1. Convergence analysis

To analyze the performance of GA⊕SE-FAM theoretically, we first introduce the definitions of a Markov chain and the convergence of the genetic algorithm.

Table 5
Cost required in Example 2.

	A_1	A_2	A_3	A_4	A_5
Company 1	(11,12,13)	(5,6,13)	(3,10,11)	(1,8,9)	(7,9,11)
Company 2	(4,5,24)	(7,8,15)	(4,5,12)	(1,4,19)	(3,4,17)
Company 3	(5,6,13)	(13,17,21)	(8,10,24)	(8,15,16)	(3,10,11)
Company 4	(10,14,24)	(10,13,22)	(2,4,18)	(3,4,17)	(7,8,21)
Company 5	(2,4,6)	(7,8,21)	(1,8,9)	(8,9,16)	(7,8,15)

Definition 6 Han and Liao (1999). Let $X(n) = \{X_1(n), X_2(n), \ldots, X_N(n)\}$ be the nth population of the genetic algorithm, and Z_n denote the optimal value in the population $X(n)$, that is, $Z_n = \max\{f(X_i(n)) | i = 1, 2, \ldots, N\}$. If $\lim_{n\to\infty} P\{Z_n = f^*\} = 1$, then we say the genetic sequence $\{X(n)\}_{n=1}^{\infty}$ converges. Here, $f^* = \max\{f(X) | X \in S\}$ denotes the global optimal value of individuals.

Definition 7 Xiong (1991). For a Markov chain $\{X(n)\}_{n=1}^{\infty}$, if the transition probability starting from state i to state j

$$p_{ij}(t) = p\{X(t+1) = j | X(t) = i\} = p_{ij}(i, j \in I)$$

is independent of initiation time t, then $\{X(n)\}_{n=1}^{\infty}$ is called a homogeneous Markov chain.

Theorem 4. *The genetic sequence $\{X(n)\}_{n=1}^{\infty}$ of GA⊕SE-FAM is a homogeneous Markov chain.*

Proof. Through symbolic coding, the size of the population is $s = n!$ We know from the operating process of GA⊕SE-FAM that the Nth population $X(N)$ in the evolutionary process only depends on the $N-1$th population $X(N-1)$ and genetic operators, and it is independent of $X(N-2), X(N-3), \ldots, X(0)$. Therefore,

$$P\{X(N) = i_N | X(0) = i_0, X(1) = i_1, \ldots, X(N-1) = i_{N-1}\} = P\{X(N) = i_N | X(N-1) = i_{N-1}\},$$

and this implies that $\{X(n)\}_{n=1}^{\infty}$ is a Markov chain. □

Table 4
Result of optimal assignment with different parameters for Example 1.

	= 0.3, b = 0.3	$a = b = 0.5$	a = 0.5, b = 1	$a = b = 1$
$L(\lambda) = \lambda$	3 7 1 10 2	6 7 1 3 2	1 7 10 3 2	1 7 10
	4 6 5 9 8	4 10 5 9 8	4 6 5 9 8	3 24 6 5 9 8
$L(\lambda) = \begin{cases} \lambda & \lambda \geqslant 0.8 \\ 0 & \lambda < 0.8 \end{cases}$	3 7 6 5 2	3 7 6 10 2	3 7 6 10 2	3 7 6 10 2
	4 8 10 9 1	4 8 5 9 1	4 8 5 9 1	4 8 5 9 1
$L(\lambda) = \lambda^2$	3 7 10 8 2	2 7 6 3 8	1 7 6 3 2	1 7 6 3 2
	4 6 5 9 1	4 10 5 9 1	4 10 5 9 8	4 10 5 9 8
$L(\lambda) = \begin{cases} \lambda^2 & \lambda \geqslant 0.5 \\ \lambda & \lambda < 0.5 \end{cases}$	1 7 6 3 2	6 7 1 3 2	3 7 1 10 2	3 7 1
	4 10 5 9 8	4 10 5 9 8	4 6 5 9 8	10 24 6 5 9 8
$L(\lambda) = e^{\lambda}$	3 7 1 10 2	6 7 1 3 2	3 7 1 10 2	3 7 1 10 2
	6 4 5 9 8	4 10 5 9 8	6 4 5 9 8	6 4 5 9 8

Table 6
Corresponding data in Scheme 1 and Scheme 2.

Scheme 1			Scheme 2		
Synthesizing effect value	I_L-metric	U_L-dispersion	Synthesizing effect value	I_L-metric	U_L-dispersion
57.6	32	8	57.6	36	6

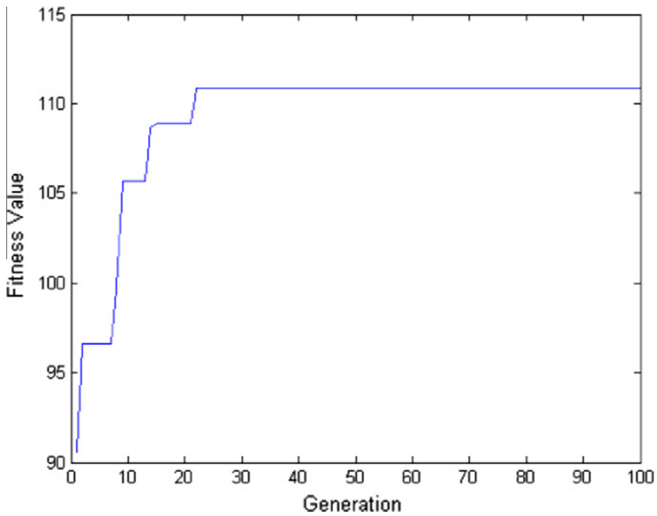

Fig. 1. Fitness curve in Case 1.

Let $p_{ij}^n(t) = P\{X_{t+n} = j|X_t = i\}$ denote the transition probability of state i to j after n steps from tth time. Because the transition probability of each generation only depends on the crossover probability, the mutation probability, as well as the population of the current generation, and it does not change with time (e.g., evolution generation) meaning $p_{ij}^n(t)$ is independent of t, $\{X(n)\}_{n=1}^{\infty}$ is a homogeneous Markov chain.

Theorem 5. GA⊕SE-FAM can converge to the global optimal solution.

Proof. Because a forced reserved strategy is used in GA⊕SE-FAM, there are some changes to the nature of a Markov chain. Suppose that the state of the current population (for example generation t) is j, when the genetic algorithm evolves to a new generation, all members of the parent population (generation t) that take part in generating the child population are compared and the most superior individual will replace the worst individual of the new generation (generation $t+1$). In other words, for some generation t' that precedes generation t, with i being the population state of generation t', a superior individual is produced in the evolution process from generation t' to generation t. It is obvious that $p_{ij}^{(n)} > 0$, which is to say it is reachable from i to j, but not reachable from j to i. That is, $p_{ji}^{(n)} = 0$, because an individual of generation t is forced to be replaced by the most superior individual of the previous generation. For arbitrary i and j, the fuzzy genetic algorithm GA⊕SE-FAM using a forced reserved strategy is an irreversible evolutionary process, and it will converge to the global optimal solution. □

6.2. Application Examples

In this section, we use two examples to analyze the performance of our algorithm for the fuzzy assignment problem. For the sake of simplicity, we assume all the elements in efficiency matrices are triangular fuzzy numbers.

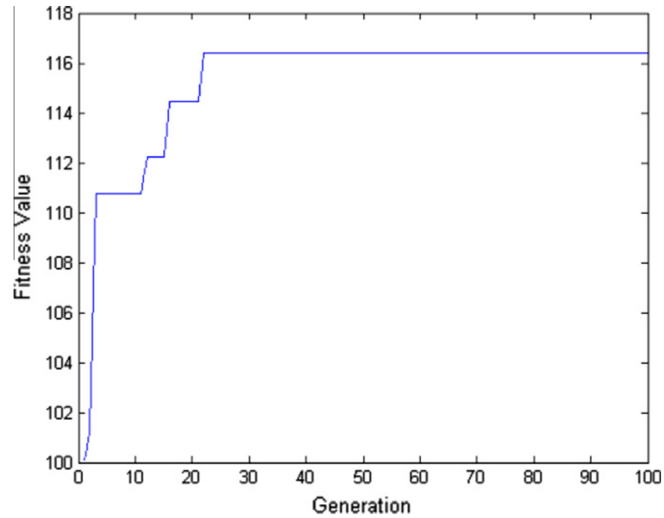

Fig. 2. Fitness curve in Case 2.

Example 1. Suppose a manuscript needs to be translated into 10 different languages, and this task will be assigned to 10 translators who have different skills. The times needed for each person to finish this task are shown in Table 1. We try to determine the optimal assignment plan.

For the sake of clarity, let $L(\lambda) = \lambda$, and for $A = (a,b,c)$, from (1), (2), $A_\lambda = [a + (b - a)\lambda, c - (c - b)\lambda]$, and based on the properties of integrals, we have $I_L(A) = (a + 4b + c)/6$ and $U_L(A) = (c - a)/6$. Therefore, $U_L(C)$ and $I_L(C)$ are as follows:

$$I_L(C) = \begin{pmatrix} 10.17 & 8.33 & 10.33 & 10.17 & 17 & 8.5 & 7.83 & 12 & 14.67 & 15.5 \\ 18 & 15.33 & 13.67 & 14.83 & 12.33 & 8 & 6 & 15 & 14.83 & 14.83 \\ 6.83 & 17 & 18.83 & 7.83 & 10.17 & 5.33 & 7 & 8.83 & 14.83 & 10.17 \\ 20 & 11 & 9.17 & 17.83 & 10 & 9.17 & 10.83 & 9.17 & 17.83 & 10 \\ 18 & 6.33 & 17 & 12.67 & 9.83 & 8 & 16.33 & 7 & 11.83 & 13 \\ 18 & 17 & 14.83 & 9.17 & 13 & 8.83 & 7 & 15.5 & 14.83 & 15 \\ 16 & 10 & 14.17 & 8.33 & 11 & 6 & 10 & 8.33 & 14.83 & 11 \\ 17 & 20 & 13.83 & 14.83 & 8.83 & 7 & 11.17 & 12 & 9.17 & 13 \\ 13 & 14.83 & 15 & 10.17 & 12 & 13 & 14.83 & 13 & 10.17 & 13.83 \\ 10 & 17.83 & 13 & 17.83 & 15.5 & 10 & 17.83 & 9 & 17.83 & 14.17 \end{pmatrix},$$

$$U_L(C) = \begin{pmatrix} 0.5 & 1 & 0.67 & 1.17 & 0.67 & 0.83 & 0.83 & 0.33 & 0.67 & 1.17 \\ 0.67 & 0.67 & 1 & 0.83 & 1.33 & 0.67 & 0.67 & 1 & 0.83 & 0.83 \\ 0.83 & 0.67 & 0.83 & 0.5 & 0.5 & 0.67 & 0.67 & 0.83 & 0.5 & 0.5 \\ 0.67 & 0.67 & 0.5 & 0.5 & 0.67 & 0.5 & 0.5 & 0.5 & 0.5 & 0.67 \\ 0.67 & 0.67 & 0.67 & 0.67 & 0.83 & 0.67 & 0.67 & 0.67 & 0.83 & 0.67 \\ 0.67 & 0.67 & 0.83 & 0.5 & 0.67 & 0.83 & 0.67 & 1.17 & 0.5 & 1 \\ 0.67 & 0.67 & 0.5 & 0.67 & 0.67 & 0.67 & 0.67 & 0.67 & 0.83 & 0.67 \\ 0.67 & 0.67 & 0.83 & 0.5 & 0.5 & 0.67 & 0.83 & 0.33 & 0.5 & 0.67 \\ 0.67 & 0.5 & 1 & 0.5 & 0.33 & 0.67 & 0.5 & 0.67 & 0.5 & 0.83 \\ 0.67 & 0.5 & 0.67 & 0.5 & 1.17 & 0.67 & 0.5 & 0.67 & 0.5 & 0.5 \end{pmatrix}.$$

According to the structure of GA⊕SE-FAM, if the genetic parameters are set as follows: the size of the population is 80, the number of evolution generations is 100, crossover probability $p_c = 0.6$, mutation probability $p_m = 0.1$, and the fitness function $F(x) = 200 - E(f(x))$, then we can discuss this problem from the following two approaches:Case 1 When $S(x,y) = x + y$, the optimal solution could be: translator 1-job 3, translator 2-job 7, translator-job 1, translator 4-job 10, translator 5-job 2, translator 6-job 4, translator 7-job 6, translator 8-job 5, translator 9-9, translator 10-job 8.

Case 2. When $S(x,y) = x - 0.5y^{0.5}$, the optimal solution could be: translator 1-job 3, translator 2-job 7, translator 3-job 6, translator 4-job 10, translator 5-job 2, translator 6-job 4, translator 7-job 8, translator 8-job 5, translator 9-job 9, translator 10-job 1.

It will converge to the above solution in about 20 generations. The fitness curves for Case 1 and Case 2 are shown in Figs. 1 and 2, respectively. We ran the experiments 10 times for each case, and the results of Case 1 and Case 2 are shown in Tables 2 and 3, respectively:

In Tables 2 and 3, S.V. denotes the synthesizing effect values, I_L-M.V. denotes the I_L-metric values, U_L-D. denotes the dispersion, C.G. denotes convergence generation, T.C. denotes time of convergence, and A.V. denotes average value.

In order to further analyze the performance of GA⊕SE-FAM, for the different parameters and data in Example 1, when $S(x,y) = x - ay^b$, we can obtain a different assignment scheme using a different $L(\lambda)$. Table 4 gives several concrete results.

The concrete data in Table 2 and Table 3 further prove the correctness of the theory for GA⊕SE-FAM in Section 6.1. At the same time, it also indicates that GA⊕SE-FAM has higher computational efficiency and a better practical value.

We see from Table 4 that: (1) The assignment schemes vary with level effect function $L(\lambda)$ (that is, decision consciousness). For example, when $a = 0.3$, $b = 0.3$, the corresponding assignment scheme for $L(\lambda) = \lambda$ is different from that for $L(\lambda) = \lambda^2$, because the variation of $L(\lambda)$ leads to the concentrated quantification value of c_{ij} in the efficiency matrix going beyond the variation range of the optimal solution; (2) For $L(\lambda) = \lambda^2$, the corresponding optimal solution is different with different a, b, which shows that the optimal solution vary with a, b when using the genetic algorithm for optimization problem (5).

For traditional assignment problems, sometimes the optimal solution is not unique, but they do not involve in uncertainty, as such we can select any solution. The same is true for fuzzy assignment problems as Example 2 will demonstrate.

Example 2. A business firm will establish five new sites A_i, $i = 1,2,\ldots,5$. In order to finish construction as soon as possible, the firm decides to use five construction companies to undertake the project. The costs $c_{ij}(i,j = 1,2,\ldots,5)$ required are as follows (Table 5). We try to determine the optimal assignment plan.

For this practical problem, if $L(\lambda) = \lambda$, $S(x,y) = x(1 + 0.1y)$, by using GA⊕SE-FAM with the following parameters: the size of the population is 40, the number of evolution generations is 100, the crossover probability $p_c = 0.6$ and the mutation probability $p_m = 0.1$, and the fitness function $F(x) = 120 - E(f(x))$. Then the results of the experiment are as follows:

(1) Company 1-A_2, Company 2-A_4, Company 3-A_5, Company 4-A_3, Company 5-A_1.
(2) Company 1-A_2, Company 2-A_3, Company 3-A_5, Company 4-A_4, Company 5-A_1.

For the above two assignment results, the corresponding data is shown in Table 6.

In order to help the reliability of the decision, we should select the scheme with the smallest dispersion, that is, take scheme 2 as the optimal solution of this assignment problem.

All the calculations above are based on Matlab 6.5 and a 2.00 GHz Pentium 4 processor, using WindowsXP professional Edition platform. If we use higher performance computers to run the experiment, we would expect even better results.

6.3. Several remarks

Remark 2. From the above theoretical analysis and experimental results, we can see that the solution scheme of fuzzy assignment problems based on the genetic algorithm can not only merge fuzzy consciousness perfectly into the decision process, but it also has higher computational efficiency and better convergence. Since different $L(\lambda)$ and $S(x,y)$ would directly lead to different assignment solutions, in practical problems, we should systematically consider all available data, views as well as the size of the assignment problem.

Remark 3. $L(\lambda)$ and $S(x,y)$ are parameters reflecting the uncertainty requirements from various aspects, and they can usually be divided into either a conservative type and a risk type. There exist both similarities and differences between them.

Remark 4. For any given parameters, assuming the optimal solution is obtained, we should also further consider the influence of parameter changes on the optimal solution; that is, we should determine the sensitivity of the optimal solution. This can provide suggestions for making more reasonable decisions, and we will plan to further discuss this in a forthcoming paper.

Remark 5. For a given $L(\lambda)$, if the optimal solution is not unique, we should select the assignment scheme with the least amount of uncertainty, so that it can improve the reliability of the decision.

Remark 6. According to the structural characteristics of fuzzy numbers, we know that the above scheme can also be applied to the general fuzzy assignment problem, and the computational complexity is similar to that of assignment problems with triangular fuzzy numbers.

7. Conclusion

The assignment problem is the most common problem in managerial decision processes, but the efficiency of assignments is often uncertain. In this paper, by using the structural characteristics of fuzzy information, we proposed a level effect function $L(\lambda)$, and established an instructive metric method to measure all aspects of fuzzy information. Further, we presented an uncertainty metric model of concentrated quantification value and established two kinds of solution models to the fuzzy assignment problem. By using the genetic algorithm, we proposed a new solution to fuzzy assignment problems and provided a concrete implementation strategy, with an analysis of the feasibility of this scheme using theory and simulation. These discussions not only provided a theoretical foundation for decision problems with different backgrounds, but they also showed easier ways to operationalize the theoretical concepts, so that they can be applied to many fields such as manufacturing, logistics and others (Li, 2012; Li, Xu, Wang, & Wang, 2012; Sepehri, 2012; Tao, Zhang, Lu, & Zhao, 2012; Xu, 2011; Zhang, Li, Xu, & Wang, 2011).

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC) Grant Nos. 71071049 and 71132008, Natural Science Foundation of Hebei Province Grant

No. F2011208056, Changjiang Scholar Program of the Ministry of Education of China, and US National Science Foundation Grant No. 1044845.

Reference

Burkard, R. (1984). Quadratic assignment problems. *European Journal of Operational Research, 15*(3), 283–289.

Chaudhry, S., Varano, M., & Xu, L. (2000). Systems research, genetic algorithms, and information systems. *Systems Research and Behavioral Science, 17*(2), 149–162.

Demirel, N. C., & Toksan, M. (2006). Optimization of the quadratic assignment problem using an ant colony algorithm. *Applied Mathematics and Computation, 183*, 427–435.

Diamond, P., & Kloeden, P. (1994). *Metric space of fuzzy set: Theory and applications*. Singapore: Word Scientific.

Feng, Y., & Yang, L. (2006). A two-objective fuzzy *k*-cardinality assignment problem. *Journal of Computational and Applied Mathematics, 197*(12), 233–244.

Goetschel, R., & Voxman, W. (1983). Topological properties of fuzzy number. *Fuzzy Sets and Systems, 10*, 87–99.

Gross, O. (1959). The bottleneck assignment problem. In *Paper P-1630, presented at the RAND symposium on mathematical programming (linear programming and extensions), 16–20 March*. Rand Corporation; 1959.

Han, W., & Liao, Z. (1999). Remark on the convergence of genetic algorithm. *Earthquake Engineering and Engineering Vibration, 19*(4), 13–16.

Jiang, Y., Xu, L., Wang, H., & Wang, H. (2009). Influencing factors for predicting financial performance based on genetic algorithms. *Systems Research and Behavioral Science, 26*(6), 661–673.

Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*(12), 83–97.

Kumar, A. (2006). A modified method for solving the unbalanced assignment problem. *Applied Mathematics and Computation, 176*(3), 76–82.

Li, L. (2012). Effects of enterprise technology on supply chain collaboration: analysis of China-linked supply chain. *Enterprise Information Systems, 6*(1), 55–77.

Li, F., Xu, L., Jin, C., & Wang, H. (2011a). Intelligent Bionic Genetic Algorithm (IB-GA) and its convergence. *Expert Systems with Applications, 38*(7), 8804–8811.

Li, F., Xu, L., Jin, C., & Wang, H. (2011b). Structure of multi-stage composite genetic algorithm (MSC-GA) and its performance. *Expert Systems with Applications, 38*(7), 8929–8937.

Li, S., Xu, L., Wang, X., & Wang, J. (2012). Integration of hybrid wireless networks in cloud services oriented enterprise information systems. *Enterprise Information Systems, 6*(2), 165–187.

Liou, T., & Wang, M. (1992). Ranking fuzzy numbers with integral value. *Fuzzy Sets and Systems, 50*(1), 247–255.

Majumdar, J., & Bhunia, A. (2007). Elitist genetic algorithm for assignment problem with imprecise goal. *European Journal of Operational Research, 177*, 684–692.

Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research, 176*, 774–793.

Sepehri, M. (2012). A grid-based collaborative supply chain with multi-product multi-period production–distribution. *Enterprise Information Systems, 6*(1), 115–137.

Song, Y., Wu, X., & Chen, M. (2001). A method for solving the multi-objective assignment problem with fuzzy information. *System Engineering, 19*(1), 28–33.

Tang, J., Lim, M., Ong, Y., & Er, M. (2006). Parallel memetic algorithm with selective local search for large scale quadratic assignment problems. *International Journal of Innovative Computing Information and Control, 2*(6), 1399–1416.

Tao, F., Zhang, L., Lu, K., & Zhao, D. (2012). Research on manufacturing grid resource service optimal-selection and composition framework. *Enterprise Information Systems, 6*(2), 237–264.

Wang, P., Zhang, J., Xu, L., Wang, H., Feng, S., & Zhu, H. (2011). How to measure adaptation complexity in evolvable systems – A new synthetic approach of constructing fitness functions. *Expert Systems with Applications, 38*(8), 10414–10419.

Xiong, D. (1991). *Stochastic processes – Theory and application*. Beijing: National Defence Industry Press.

Xu, L. (2000). The contribution of systems science to information systems research. *Systems Research and Behavioral Science, 17*(2), 105–116.

Xu, L. (2011). Enterprise systems: State-of-the-art and future trends. *IEEE Transactions on Industrial Informatics, 7*(4), 630–640.

Yang, X. F., Lu, Q., Li, C. D., & Liao, X. F. (2008). Biological computation of the solution to the quadratic assignment problems. *Applied Mathematics and Computation, 200*, 369–377.

Zhang, Y., Li, Z., Xu, L., & Wang, J. (2011). A new method for automatic synthesis of tolerances for complex assemblies based on polychromatic sets. *Enterprise Information Systems, 5*(3), 337–358.

Zhou, M., & Sun, S. (1999). *Genetic algorithm: Theory and applications*. Beijing: National Defence Industry Press.