Soft Comput (2016) 20:2245–2259
DOI 10.1007/s00500-015-1639-z

METHODOLOGIES AND APPLICATION

Multi-objective optimization problem under fuzzy rule constraints using particle swarm optimization

Debjani Chakraborty · Debashree Guha · Bapi Dutta

Published online: 17 March 2015

Abstract In this paper, a fuzzy multi-objective programming problem is considered where functional relationships between decision variables and objective functions are not completely known to us. Due to uncertainty in real decision situations sometimes it is difficult to find the exact functional relationship between objectives and decision variables. It is assumed that information source from where some knowledge may be obtained about the objective functions consists of a block of fuzzy if-then rules. In such situations, the decision making is difficult and the presence of multiple objectives gives rise to multi-objective optimization problem under fuzzy rule constraints. In order to tackle the problem, appropriate fuzzy reasoning schemes are used to determine crisp functional relationship between the objective functions and the decision variables. Thus a multi-objective optimization problem is formulated from the original fuzzy rule-based multi-objective optimization model. In order to solve the resultant problem, a deterministic single-objective non-linear optimization problem is reformulated with the help of fuzzy optimization technique. Finally, PSO (Particle Swarm Optimization) algorithm is employed to solve the resultant single-objective non-linear optimization model and the computation procedure is illustrated by means of numerical examples.

Communicated by V. Loia.

D. Chakraborty
Department of Mathematics, IIT Kharagpur,
Kharagpur 721302, India

D. Guha (✉) · B. Dutta
Department of Mathematics, IIT Patna, Patna 800013, India
e-mail: debashree@iitp.ac.in

Keywords Fuzzy rule · Linguistic variable · Multi-objective optimization · Takagi-Sugeno fuzzy reasoning · Tsukamoto's fuzzy reasoning · Particle swarm optimization (PSO) algorithm

1 Introduction

Decision makers in many areas, from industry to engineering and the social sector, face an increasing need to consider multiple, conflicting objectives in their decision processes. In this scenario the real-world decision problems can be formulated as multi-objective optimization models. The first note on multi-objective optimization was given by Pareto (Miettinen 1999); since then, the determination of the compromise set of solutions for a multi-objective problem is called Pareto optimization.

Conventional optimization methods assume that all the parameters and goals of an optimization model are precisely known. However, in reality problems arise in development of precise mathematical model of the system due to lack or abundance of information, subjective opinions, inadequate formulation of objectives and inability in evaluating the relative importance among the objectives. Under this circumstance to model imprecision and uncertainty of the system, fuzzy set theory (Zadeh 1965) becomes a natural choice since it can define the imprecise information in a more logical and meaningful fashion. Fuzzy multi-objective optimization problems have been studied in literature by several authors. Bellman and Zadeh (1970) developed fuzzy optimization problems by providing aggregation operators to combine fuzzy goals and fuzzy decision space. Zimmermann (1978) first used the fuzzified constraints and objective functions to solve the multi-objective linear programming problems. Chanas (1989) used the parametric programming technique to solve the fuzzy multi-objective linear programming problems. After this motivation and inspiration there came out a lot of literature dealing with the fuzzy optimization problems (e.g., Esogbue 1991; Lee and Li 1993; Sakawa and Sawada 1994; Nishizaki and Sakawa 1995; Mohan and Nguyen 1998; Sakawa and Kato 2000; Ali 2001; Wu 2004; Bector and Chandra 2005; Wu 2008; Yano and Sakawa 2009; Zhang et al. 2010). The collection of papers on the topic of the fuzzy optimization edited by Delgado et al. (1994) and Slowinski (1998) gave the main stream of this topic. On the other hand, the books by Zimmermann (1996); Lai and Hwang (1992, 1994) also gave the insightful survey.

In recent times, there are many other interesting articles concerning the fuzzy multi-objective programming problems. Thapar et al. (2012) presented genetic algorithm for solving multi-objective optimization problems with max-product fuzzy relation equations as constraints. Deep et al. (2011) proposed an interactive approach based method for solving multi-objective optimization problems by treating objectives as fuzzy goals and the satisfaction of constraints is considered at different α-level sets of the fuzzy parameter used. Kotinis (2014) developed a multi-objective version of the differential evolution optimization algorithm, which includes fuzzy adaptation of parameters and K-medoids clustering of solution vector. Li et al. (2013) proposed a multi-objective train scheduling model by minimizing the energy and carbon emission cost as well as the total passenger time, and to represent the fuzzy nature of failure they used the linear and non-linear fuzzy membership functions. Further, multi-objective optimization techniques have been applied in reliability optimization problems (Garg and Sharma 2013), portfolio selection problems (Liu and Zhang 2013; Khalili-Damghani and Sadi-Nezhad 2013), designing fuzzy random routing algorithms for wireless sensor networks (Lu et al. 2014) and solving method of several kinds of matrix games (Bector and Chandra 2005).

Most of the fuzzy multi-objective decision making (MODM) problems, currently available in literature, may be defined mathematically as finding the solution vector, or vector of crisp decision variables $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ that optimizes a vector of objective functions $\tilde{f}(x) = (\tilde{f}_1(x), \ldots, \tilde{f}_m(x))$ subject to $\tilde{A}x \underset{\sim}{<} \tilde{b}$, where $\tilde{A}, \tilde{b}$ are fuzzy quantities. In the conventional fuzzy optimization problems, the objective functions (say $\tilde{f}_i$, $i = 1, 2, \ldots, m$) are expressed as functions of decision variables. However, in solving practical decision problems, it may not be possible to get exact functional relationship between the decision variables and the objective functions (Chakraborty and Guha 2013). In this situation, we are only able to describe the casual link between objective functions and decision variables linguistically from past data. Let us take an example of an investment company, say Company ABC, which used to invest money in different amounts among four industries such as car industry, food industry, computer and arms industry. Suddenly, the company's performance in the business has fallen and also the company is facing a loss of over 10 % in business. In this situation, there is an urgent need for reviewing different investments so that from this analysis the company can get an idea to perform better. The company administrators, from past experience, can at best provide the information regarding its different past investments with corresponding net profits and as well as risk factors involved therein in terms of few production rules. For example, the manager collects the information given below.

> If the investment in car industry is "around \$80 million", the investment in food industry is "around \$40 million", the investment in computer industry is "around \$70 million" and the investment in arms industry is "around \$85 million" then the profit is "satisfactory" and risk is "somewhat", etc.

In this manner as presented above, the company collects the past records in the form of rules and constructs a rule base bearing the information regarding its different investment amounts, corresponding profits and risks for different time periods. Due to complexity of the studied system, the past information cannot be assessed with both precision and certainty and, thus, the company administrators feel comfortable to use linguistic assessments to express them in a more realistic manner. Now in this scenario the company is required to maximize the net profit while simultaneously minimizing the risk, which will be achieved based on its investments in different industries. Thus, to find the most suitable investments for the company, we need to formulate an optimization model, where the investment amounts for different industries are treated as decision variables with two objectives, i.e., maximizing profit and minimizing risk. In designing this optimization model the problems arise, as the functional relationship between the objective functions and the decision variables cannot be directly found in the given information. Here basically, the natures of objectives with respect to the decision variables are expressed linguistically by using rule-based system. Obviously the question arises as to how the company is able to reach a decision. The motivation of the presented work here is to provide a method for solving these types of practical MODM problems, in which the functional relationship between decision variables and objectives is not known.

Now in this respect, Carlsson and Fuller (1998) proposed a theory to find a compromise solution to the fuzzy multi-objective mathematical programming problem under fuzzy rule base constraints. However, in their solution procedure decision variables are restricted to assume only two linguistic terms {small, big} which are represented by complementary membership functions small$(x) = 1 - x$ and big$(x) = x$ for $x \in [0, 1]$. However, partitioning the domain of decision variables into two linguistic terms is an oversimplification and not practical from the application point of view. The number of the linguistic terms should be compatible with the number of conceptual entities a human being can efficiently store and utilize in his inference activities. Therefore, psychologist (Miller 1956) recommend the use of linguistic terms, less than 5 being not sufficiently informative and more than 9 being too much for proper understanding of their differences. Therefore, in the case of real decision situations a linguistic term set, with 7 ± 2 linguistic terms, is required for describing the objectives and decision variables more suitably. Thus, with the increment of linguistic terms for describing the decision variables as well as objectives, solving fuzzy-based multi-objective becomes a challenging task and requires an effective method to tackle this kind of problem with immense potential of practical application. With this analysis we present a novel fuzzy rule-based multi-objective optimization model which can model the linguistic dependencies between the decision variables and the objectives in an improved manner. It is assumed that the information source from where some knowledge may be obtained about the objective functions consists of a block of fuzzy if-then rules and represented as follows:

$$\max/\min \quad f(x) = (f_1(x), \ldots, f_m(x)); \\ \text{subject to}\{\mathfrak{R}_1(x), \mathfrak{R}_2(x), \ldots, \mathfrak{R}_p(x) | x \in Y\},$$

where $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ is the vector of decision variables, which are treated as linguistic variables, and $Y \subset \mathbb{R}^n$ is a (crisp or fuzzy) set of constraints on the domain of decision variables $x_1, x_2, \ldots, x_n$. The rule $\mathfrak{R}_j(x)$ constitutes only knowledge available about the values of $f(x) = (f_1(x), \ldots, f_m(x))$.

In the present study, to tackle the fuzzy rule-based MODM problem, we use appropriate fuzzy reasoning method (Takagi and Sugeno 1985; Tsukamoto 1979) for finding the crisp values of the objective functions at $y \in Y$. Subsequently, the original fuzzy rule-based optimization problem transforms into the following crisp non-linear multi-objective optimization problem:

$$\max/\min \quad f(y) = (f_1(y), \ldots, f_m(y)); \ \text{subject to} \quad y \in Y$$

We use aggregation operator for aggregating the different fuzzy goals of the objectives, and finally, to solve the resultant single-objective optimization problem which is non-linear, a robust global optimization technique is required. In this view, we employ a recently developed particle swarm optimization (PSO) (Clerc and Kennedy 2002) algorithm to find the compromise solution of the resulting deterministic non-linear multi-objective optimization problem.

This paper is organized as follows: Sect. 2 describes some useful results that are used to solve the proposed problem. Multi-objective optimization problem under fuzzy if-then rule base is introduced and the state of art of solving this kind of problem is developed in Sect. 3. In Sect. 4, the proposed methodology is illustrated using a numerical example. A practical application of the proposed method in production planning is also described via an example of a toy company, with detailed comparison analysis in Sect. 4. Finally, some conclusions are drawn in Sect. 5.

2 Preliminaries

In this section, we revise the concepts of linguistic variable and its representation. We then briefly review fuzzy reasoning schemes, which are the basis of our proposal.

2.1 Membership functions for the linguistic value of decision variable

In the present study, the decision variables $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of fuzzy rule-based multi-objective optimization model are considered as linguistic variables. A linguistic variable may be regarded either as a variable whose value is a fuzzy number or as a variable whose values are defined in linguistic terms (Zadeh 1975a, b, c). The mathematical formalism of linguistic variable may be presented in the following way:

Definition 1 (*Linguistic variable*) A linguistic variable is characterized by a quintuple $(x, \mathfrak{J}(x), X, G, M)$ in which x is the name of the linguistic variable; $\mathfrak{J}(x)$ is the term set of x, i.e., the set of names of linguistic values of x with each value being a fuzzy number defined on X; G is a syntactic rule for generating the names of values of x; and M is a semantic rule for associating with each value its meaning. The family of all fuzzy (sub) sets in X is denoted by $\mathfrak{J}(X)$.

In this paper, it is assumed that the values of each of the linguistic variables $x_1, x_2, \ldots, x_n$ are defined in the interval $[a, b] \subset \mathbb{R}$. Suppose that $X = [a, b]$ and $\mathfrak{J}(x)$ consists of $K + 1$, $(K \geq 2)$, terms (see Fig. 1) as given below.

$$\mathfrak{J} = \{\text{low}_1, \text{around}(a + \beta), \text{around}(a + 2\beta), \ldots, \text{around} \\ (a + (K - 1)\beta), \text{high}_K\}; \ \text{where } \beta = (b - a)/K,$$

where each term being a fuzzy number may be represented with the help of triangular membership functions $\{\tilde{\mu}_{A_1}, \ldots, \tilde{\mu}_{A_{K+1}}\}$ of the following form:

$$\mu_{\tilde{A}_1}(x) = \mu_{\text{low}_1}(x) \\ = \begin{cases} 1 - (x - a)/(b - a) & \text{if } \ a \leq x \leq b \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

The fuzzy number $\tilde{A}_1$ is denoted as $\tilde{A}_1 = (a; 0, b - a)$.

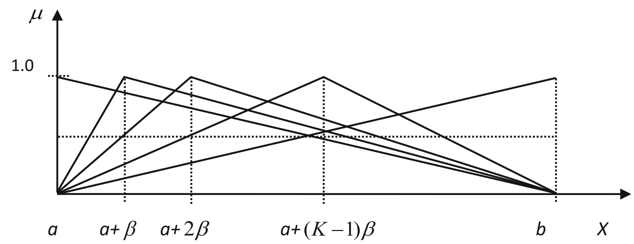

Fig. 1 Membership functions for the linguistic values with $(K + 1)$ terms

Table 1 The term set $\mathfrak{I}$(x)

$\mathfrak{I}$(x)	Fuzzy numbers
Very low (VL)	(0; 0, 100)
Low (L)	(25; 25, 75)
Medium (M)	(50; 50, 50)
High (H)	(75; 75, 25)
Very high (VH)	(100; 100, 0)

$$\mu_{\tilde{A}_k}(x) = \mu_{\text{around}(a+k\beta)}(x) = \begin{cases} 1-(a+k\beta-x)/k\beta & \text{if } a \le x \le a+k\beta \\ 1-(x-a-k\beta)/(b-a-k\beta) & \text{if } a+k\beta \le x \le b \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

for $1 \le k \le (K-1)$ and each of the corresponding fuzzy number $\tilde{A}_k$ is denoted as $\tilde{A}_k = (a + k\beta; k\beta, b - a - k\beta)$.

$$\mu_{\tilde{A}_{K+1}}(x) = \begin{cases} 1-(b-x)/(b-a) & \text{if } \ a \le x \le b \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The fuzzy number $\tilde{A}_{k+1}$ is denoted as $\tilde{A}_{k+1} = (b; b-a, 0)$. Here, for representing a fuzzy number in [0, 1] with membership function as described above, we have used the standard notation, $\tilde{A} = (m; \beta, \gamma)$ with mean value m, left spread β and right spread γ. The novelty of this term set arrangement is that when a very little knowledge is available about the boundaries of individual term, each term is stretched over the whole domain though the mid values of each of the terms are situated at a fixed distance apart.

Example 1 Suppose x is interpreted as one linguistic variable with term set $\mathfrak{I}(x) = \{\text{very low, low, medium, high,very high}\}$, where each term in $\mathfrak{I}(x)$ is characterized by a fuzzy number in the universe of discourse [0, 100]. Then by utilizing (1), (2) and (3) each term in $\mathfrak{I}(x)$ can be transformed to associate fuzzy numbers as provided in Table 1.

2.2 Fuzzy reasoning schemes

Mathematicians, specialists on fuzzy logic, developed many kinds of fuzzy inference systems. The basic difference between various models lies in the representation of the consequents of their fuzzy rules. Thus, for making decision from fuzzy rule-based system, depending on the nature of the consequent of fuzzy if-then rules, two traditional well-known fuzzy inference mechanisms are employed in this present study. Before describing these reasoning schemes, we recall the definition of triangular norms which is used to model logical connective ***and***. In fuzzy set theory, triangular norms (in short T-norm) are extensively used to model logical connective ***and***.

Definition 2 (*T-norm*) A triangular norm T is a function from $[0, 1] \times [0, 1]$ to $[0, 1]$ such that it is symmetric, associative and non decreasing in each argument and $T(a, 1) = a \forall a \in [0, 1]$.

The effectiveness of the product T-norm ($T_P(a, b) = ab$) operator has already been discussed in the literature (Deep et al. 2011) and this inspired us to use the product T-norms to model the ***and*** operator of the given rule base. We also recall that if x is a linguistic variable in the universe of discourse X and $y \in X$ then we write "x is $\bar{y}$" to indicate that $\bar{y}$ is a crisp value of the linguistic variable x.

2.2.1 Tsukamoto's inference scheme

This model is applied if the consequent of each fuzzy if-then rule is represented by a fuzzy set with a strictly monotone membership function. A brief description of Tsukamoto's fuzzy reasoning method is given below. For this purpose, consider the following fuzzy inference system:

$$\begin{array}{ll} \mathfrak{R}_j: & \text{If } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \dots \text{ and } x_n \text{ is } \tilde{A}_{jn} \\ & \text{then } z \text{ is } \tilde{C}_j. \\ \text{Input} & x_1 \text{is } \bar{y}_1 \text{ and } x_2 \text{ is } \bar{y}_2 \text{ and } \dots \text{ and } x_n \text{ is } \bar{y}_n \\ \text{Output} & \qquad z \text{ is } z_{TS}, \end{array} \tag{4}$$

where $\tilde{A}_{jk} \in \mathfrak{I}(X_k)$ is a value of the linguistic variable x_k defined in the universe of discourse$X_k \subset \mathbb{R}$ (for $k = 1, 2, \dots, n$), and $\tilde{C}_j \in \mathfrak{I}(Z)$ is a value of the linguistic variable z defined in the universe $Z \subset \mathbb{R}$ for $j = 1, 2, \dots, p$. It is assumed that Z is bounded. In Tuskamoto's (1979) fuzzy reasoning scheme, it is supposed that each $\tilde{C}_j$ has strictly monotonic membership functions on Z. The procedure for obtaining the crisp output z_{TS} from the crisp input vector $y = (y_1, y_2, \dots, y_n)$ and fuzzy rule base $\{\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_p\}$ is described below in a stepwise manner.

Step 1 The degree to which input matches the jth rule $\mathfrak{R}_j$ is computed by

$$l_j = T_P(\mu_{\tilde{A}_{j1}}(y_1), \mu_{\tilde{A}_{j2}}(y_2), \dots, \mu_{\tilde{A}_{jn}}(y_n)) \quad \text{for } j = 1, 2, \dots, p \tag{5}$$

Step 2 In this mode of reasoning, the individual crisp control actions z_j are derived from the relation

$$l_j = \mu_{\tilde{C}_j}(z_j), \text{ i.e., } z_j = \mu_{\tilde{C}_j}^{-1}(l_j) \tag{6}$$

The inverse of $\mu_{\tilde{C}_j}$ is well-defined because of its strict monotonicity.

Step 3 The overall system output is defined as the weighted average of the individual outputs, where associated weights are the firing levels. Therefore, the overall crisp control action is computed by

$$z_{TS} = \frac{l_1 z_1 + \cdots + l_p z_p}{l_1 + \cdots + l_p} \tag{7}$$

2.2.2 Takagi and Sugeno fuzzy reasoning scheme

Takagi and Sugeno (1985) introduced a fuzzy reasoning scheme. In this fuzzy reasoning method, a fuzzy rule is of the following form:

If x_1 is $\tilde{A}_1$ and x_2 is $\tilde{A}_2$ then $z = f(x_1, x_2)'$

where $\tilde{A}_1$ and $\tilde{A}_2$ are fuzzy sets in the antecedent, while $z = f(x_1, x_2)$ is a crisp function in the consequent, and $f(.,.)$ is very often a linear function with respect to x_1 and x_2. Let us consider the following architecture:

$$\begin{aligned} &\mathfrak{R}_j: \text{ If } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{A}_{jn} \\ &\text{then } z = a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jn}x_n + b_j. \\ &\text{Input } x_1 \text{ is } \bar{y}_1 \text{ and } x_2 \text{ is } \bar{y}_2 \text{ and } \ldots \text{ and } x_n \text{ is } \bar{y}_n \\ &\text{Output} \qquad\qquad z \text{ is } z_{TgS}. \end{aligned} \tag{8}$$

where $\tilde{A}_{jk}$s are defined as in (4), a_{jk} and b_j are real numbers for $j = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, n$. The procedure for obtaining the crisp output z_{TgS} from the crisp input vector $y = (y_1, y_2, \ldots, y_n)$ and fuzzy rule base $\{\mathfrak{R}_1, \mathfrak{R}_2, \ldots, \mathfrak{R}_p\}$ is are described below:

Step1 The degree to which input matches the jth rule $\mathfrak{R}_j$ is typically computed using the relation

$$\begin{aligned} l_j &= T_P(\mu_{\tilde{A}_{j1}}(y_1), \mu_{\tilde{A}_{j2}}(y_2), \ldots, \mu_{\tilde{A}_{jn}}(y_n)), \\ &\quad \text{for } j = 1, 2, \ldots, p \end{aligned} \tag{9}$$

Step 2 Then the individual rule outputs derived from the relationship

$$z_j(y) = \sum_{k=1}^{n} a_{jk} y_k + b_j \tag{10}$$

Step 3 Finally, the crisp control action is represented as

$$z_{TgS} = \frac{l_1 z_1 + \cdots + l_p z_p}{l_1 + \cdots + l_p} \tag{11}$$

Note The main difference between Tsukamoto's fuzzy reasoning scheme and Takagi–Sugeno fuzzy reasoning method lies in the consequent of the fuzzy rules: the former uses the fuzzy sets with strict monotonic membership functions where the latter employs the (linear) functions of input variables. Depending on the natures of the given information, which is described by fuzzy rules, it is important to realize which choice of the reasoning method is appropriate.

3 Multi-objective optimization problem under fuzzy if-then rules

3.1 Proposed fuzzy rule-based multi-objective optimization model

Let us consider the following multi-objective optimization problem in which the functional relationships between the decision variables and the objective function are not completely known:

$$\begin{aligned} &\max \quad f(x) = (f_1(x), \ldots, f_m(x)); \quad \text{subject to} \\ &\{\mathfrak{R}_1(x), \mathfrak{R}_2(x), \ldots, \mathfrak{R}_p(x) | x \in Y \subset \mathbb{R}^n\} \end{aligned} \tag{12}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the ith objective function, $x_1, x_2, \ldots, x_n$ are linguistic variables and $Y \subset \mathbb{R}^n$ is a (crisp or fuzzy) set of constraints on the domain of $x_k (k = 1, 2, \ldots, n)$. The causal link between x and $f(x)$ is described linguistically using fuzzy if-then rules $\mathfrak{R}_j(x)$. Moreover, depending on the nature of information available for the objective functions f_is, $\mathfrak{R}_j(x)$ may be either of following two types:

Type I $\mathfrak{R}_j(x)$: If x_1 is $\tilde{A}_{j1}$ and x_2 is $\tilde{A}_{j2} \ldots x_n$ is $\tilde{A}_{jn}$ then $f_1(x)$ is $\tilde{C}_{1j}, f_2(x)$ is $\tilde{C}_{2j}, \ldots, f_m(x)$ is $\tilde{C}_{mj}$

Type II $\mathfrak{R}_j(x)$: If x_1 is $\tilde{A}_{j1}$ and x_2 is $\tilde{A}_{j2} \ldots x_n$ is $\tilde{A}_{jn}$ then $f_i(x) = \sum_{k=1}^{n} a_{jik} x_k + b_{ji}, \;\; i = 1, 2, \ldots, m.$

It is important to note here that in Type I, the objective functions, namely f_is, are represented linguistically using fuzzy sets $\tilde{C}_{ij}$ with strictly monotone and continuous membership functions defined in the universe of discourse $Z_i \subset \mathbb{R}$ (for $i = 1, 2, \ldots, m$). In Type II, the objective function f_i is crisp function of the decision variables and a_{jik},b_{ji} are real numbers. In both Type I and II, fuzzy numbers $\tilde{A}_{jk} (j = 1, 2, \ldots, p; \; k = 1, 2, \ldots, n)$ representing the linguistic values of the decision variables x_k are defined in the universe of discourse $X_k \subset \mathbb{R}$ (for $k = 1, 2, \ldots, n$).

3.2 Formation of the equivalent deterministic multi-objective optimization model

We shall describe how the proposed model (12), based on the rule base of Type I and Type II, can be transformed into the equivalent deterministic multi-objective optimization model.

Type I In jth rule $\mathfrak{R}_j(x)$, f_i assumes linguistic values

In this situation, the procedure to find a deterministic model to the fuzzy multi-objective optimization problem (12) is illustrated stepwise below.

Step 1 Normalization of values of linguistic variables involved in antecedent and consequent part of rules

Since different types of information are measured in different ways, n variables $x_1, x_2, \ldots, x_n$ may be measured in different scales. Then the values of the linguistic variables $x_k (k = 1, 2, \ldots, n)$, i.e., $\tilde{A}_{jk} = (m_{jk}; \beta_{jk}, \gamma_{jk})$ need to be normalized. The purpose of the normalization method is to preserve the property such that the range of decision variables belongs to the closed interval [0, 1]. In the present study the linear scale transformation (Chen 2000; Li 2007) is used to transform the various decision variables into a comparable scale.

$$\tilde{a}_{jk} = \left(\frac{m_{jk}}{d_k^{\max}}; \frac{\beta_{jk}}{d_k^{\max}}, \frac{\gamma_{jk}}{d_k^{\max}} \right), \text{ where}$$
$$d_k^{\max} = \max_{1 \le j \le p} \{ m_{jk} + \gamma_{jk} | \tilde{A}_{jk} = (m_{jk}; \beta_{jk}, \gamma_{jk}) \} \tag{13}$$

The linguistic values $\tilde{C}_{ij}$ (for $i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, p$) of the objective functions $f_i(x)$ are normalized with the help of same normalization procedure (as mentioned above) and the corresponding normalized values are denoted as $\tilde{c}_{ij}$. Then the given fuzzy if-then rule base (i.e., Type I) is transformed into the following form:

$\mathfrak{R}_j(x)$: If x_1 is $\tilde{a}_{j1}$ and x_2 is $\tilde{a}_{j2} \ldots x_n$ is $\tilde{a}_{jn}$ then
$f_1(x)$ is $\tilde{c}_{1j}$, $f_2(x)$ is $\tilde{c}_{2j}, \ldots, f_m(x)$ is $\tilde{c}_{mj}$.

Step 2 Equivalent fuzzy rule base

The compositional rule of inference of individual objective functions will generate $p.m$ number of rules as follows:

$$\left.\begin{array}{ll} \mathfrak{R}_{1j}(x): & \text{If } x_1 \text{ is } \tilde{a}_{j1} \text{ and } x_2 \text{ is } \tilde{a}_{j2}, \ldots, x_n, \text{ is } \tilde{a}_{jn} \text{ then } f_1(x) \text{ is } \tilde{c}_{1j},\ j = 1, 2, \ldots, p \\ \mathfrak{R}_{2j}(x): & \text{If } x_1 \text{ is } \tilde{a}_{j1} \text{ and } x_2 \text{ is } \tilde{a}_{j2}, \ldots, x_n \text{ is } \tilde{a}_{jn} \text{ then } f_2(x) \text{ is } \tilde{c}_{2j},\ j = 1, 2, \ldots, p \\ & \ldots \\ \mathfrak{R}_{mj}(x): & \text{If } x_1 \text{ is } \tilde{a}_{j1} \text{ and } x_2 \text{ is } \tilde{a}_{j2}, \ldots, x_n \text{ is } \tilde{a}_{jn} \text{ then } f_m(x) \text{ is } \tilde{c}_{mj},\ j = 1, 2, \ldots, p \end{array}\right\} \tag{14}$$

Step 3 Aggregation of the rule outputs

In this step, on account of our discussion in Sect. 2.2.1 we can apply Tsukamoto's inference mechanism (4) to find out precise value of each of the objective function f_i (for $i = 1, 2, \ldots, m$). This means that the crisp value of ith objective function f_i from the crisp input vector $y \in Y \subset \mathbb{R}^n$ and the p rules $\mathfrak{R}_{ij}(x) (j = 1, 2, \ldots, p)$ may be obtained by employing (5), (6) and (7) as

$$f_i(y) = \frac{l_1 z_{i1} + \cdots + l_p z_{ip}}{l_1 + \cdots + l_p}, \tag{15}$$

where $l_j = T_P(\mu_{\tilde{a}_{j1}}(y_1), \mu_{\tilde{a}_{j2}}(y_2), \ldots, \mu_{\tilde{a}_{jn}}(y_n))$ is the degree to which the input vector match the jth rule $\mathfrak{R}_{ij}$ and $z_{ij} = \mu_{\tilde{c}_{ij}}^{-1}(l_j)$ is the output of the jth rule ($i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, p$).

Step 4 Construction of equivalent crisp multi-objective programming problem

Thus, the optimization problem (12) now converts to a multi-objective crisp mathematical programming problem as follows:

$$\max f(y) = (f_1(y), \ldots, f_m(y)) : \text{ subject to } y \in Y \tag{16}$$

Here the ith objective function may be non-linear in nature

Type II In jth rule $\mathfrak{R}_j(x)$, the crisp functional relationship between f_i s and x_k *s* exists

In this situation to find the compromise solution of the fuzzy multi-objective optimization problem (12), we have proceeded in a way similar to Type I and finally the crisp value of the objective function f_i at $y \in Y \subset \mathbb{R}^n$ is determined from the given fuzzy rule base using Takagi–Sugeno fuzzy reasoning scheme (8), (9), (10) and (11) as

$$f_i(y) = \frac{l_1 z_{i1} + \cdots + l_p z_{ip}}{l_1 + \cdots + l_p} \tag{17}$$

Here, l_j is computed in a way similar to Type I and z_{ij} representing the output of the jth rule, can be computed as $z_{ij} = \sum_{k=1}^{n} a_{jik} x_k + b_{ji}$. Hence, the equivalent crisp multi-objective non-linear programming (MONLP) problem of the form (16) is obtained for Type II also. Now our aim is to solve multi-objective programming problem (16).

3.3 Solution procedure of multi-objective optimization model (16)

The above multi-objective programming problem (16) yields not a single optimal solution, but a set of Pareto optimal solutions, in which one objective cannot be improved without sacrificing other objectives. For practical applications, however, one solution needs to be selected, which will satisfy the different goals to some extent. Such a solution is called best compromise solution. In this respect, considering the imprecise nature of decision makers' subjective judgment, it is natural to accept that decision makers may have fuzzy or imprecise goals for each of the objective functions (Sakawa and Yano 1985). Decision makers' fuzzy goals attached to each of the objectives can be described via fuzzy sets which are characterized by membership functions. To represent the nature of the fuzzy goal of each objective function the membership functions was first introduced by Zimmermann (1978). It was later followed by Narasimhan (1980), Hannan (1981), Lee and Li (1993), Rao et al. (1993), Huang (1997), Mohan and Nguyen (1998), Deep et al. (2011), Thapar et al. (2012), Garg and Sharma (2013), etc.

In the present study, for easy understanding and obvious computational advantage, simple linear membership functions $\mu_{f_i}, i = 1, 2, \ldots, m$ corresponding to the objective functions $f_i(y),\ i = 1, 2, \ldots, m$ are introduced to represent goal of each objective function as follows:

$$\mu_{f_i}(y) = \begin{cases} 1 & \text{if } f_i(y) \geq M_i \\ \frac{f_i(y)-m_i}{M_i-m_i} & \text{if } m_i \leq f_i(y) \leq M_i \\ 0 & \text{if } f_i(y) \leq m_i \end{cases} \tag{18}$$

Here M_i is the maximum value (or upper bound) of individual objective function f_i at which the decision maker is completely satisfied and M_i can be determined by $M_i = f_i(y_i^*)$ where y_i^* is the ideal solution for individual objective function obtained by solving $\max_{y \in Y} f_i(y)$. $(M_i - m_i)$ is the subjectively chosen constants of admissible violations, i.e., at m_i the decision maker is not-at-all satisfied. Basically, m_i is the lower bound of each objective which can be determined with the help of ideal solution as $m_i = \min_{1 \leq s \leq m} f_i(y_s^*),\ i = 1, 2, \ldots, m$. The membership function $\mu(f_i(y))$ represents the degree of satisfaction of the decision maker as a value between zero and one.

Having elicited the membership functions $\mu_{f_i}(y),\ i = 1, 2, \ldots, m$ for each of the objective functions $f_i(y),\ i = 1, 2, \ldots, m$ through the interaction of decision maker, the MONLP (16) can be converted into the fuzzy MONLP programming problem defined by

$$\max_{y \in Y} (\mu_{f_1}(y),\ \mu_{f_2}(y), \ldots, \mu_{f_m}(y)) \tag{19}$$

One of the crucial parts in fuzzy MONLP is the aggregation of membership values of the objective functions to convert it into a single-objective non-linear optimization problem. We can convert the preceding fuzzy MONLP into the following single-objective non-linear optimization problem based on the concept of aggregation operator in the following way:

$$\max_{y \in Y}\ A(\mu_{f_1}(y),\ \mu_{f_2}(y), \ldots, \mu_{f_m}(y)), \tag{20}$$

where A denotes an aggregation function (Grabisch et al. 2009). We recall that an aggregation function A on the scale [0, 1] is a non-decreasing function $A : [0, 1]^m \rightarrow [0, 1]$ such that $A(0, 0, \ldots, 0) = 0$ and $A(1, 1, \ldots, 1) = 1$. As a typical aggregation function, we may recall arithmetic mean, geometric mean, minimum operator, product operator. Among the several choices, weighted arithmetic mean (Lai and Lai 2000) and minimum operator (Cheng et al. 2013) are widely used in literature to aggregate the objectives. However, as mentioned earlier, due to the effectiveness of the product operator (Cheng and Li 1996; Deep et al. 2011) over the other existing aggregation functions, we adopt product operator as aggregation function which may be written as

$$\begin{aligned} &A(\mu_{f_1}(y),\ \mu_{f_2}(y), \ldots, \mu_{f_m}(y)) \\ &= \mu_{f_1}(y) \times \mu_{f_2}(y) \times \cdots \times \mu_{f_m}(y) \end{aligned} \tag{21}$$

One may observe that the value of $A(\mu_{f_1}(y)\, \mu_{f_2}(y), \ldots, \mu_{f_m}(y))$ can be interpreted as the overall degree of satisfaction of decision makers' fuzzy goals. After eliciting the choice of aggregation function A, the optimization system turns into the following single-objective non-linear optimization problem defined as follows:

$$\begin{aligned} &\text{Maximize } \alpha_1(y) \times \alpha_2(y) \times \cdots \times \alpha_m(y) \\ &\text{subject to} \\ &\qquad \mu_{f_i}(y) = \alpha_i(y),\ \ i = 1, 2, \ldots, m \\ &\qquad y \in Y, \end{aligned} \tag{22}$$

where $\alpha_i(y)$ is the degree of satisfaction of ith objective.

Since the crisp optimization problem (22) is non-linear in nature, it requires some effective methods and algorithms for finding global solution. Among various existing methods and algorithms, evolutionary algorithmic (EA) approaches are widely used to find the global solution of the non-linear optimization problem (Eiben and Smith 2003). The advantage of EA technique is that it does not require any kind of pre-assumptions, such as continuity, differentiability of objective functions and constraints. Particle swarm optimization (PSO) (Kennedy and Eberhart 1995; Shi and Eberhart 1998a) being prominent, one of the family of EAs has been applied widely in practical applications. From the applicability point of view, in this paper, we use PSO as a tool to solve the optimization model (22). The process of solving optimization problem under fuzzy if-then rules can be summarized in algorithmic form as given below.

Computational algorithm

Step 1 Calculate normalized values of the linguistic variables involved in given rule base representing linguistic relationship between the objectives and decision variables

Step 2 Use suitable reasoning scheme to compute the exact functional relationship of objectives and decision variables from the rule base with normalized values of linguistic variables

Step 3 Calculate the individual upper and lower bound for each objective function

Step 4 Keeping in view the functional form of the objectives and the ranges (upper and lower bounds) of each objective construct fuzzy membership functions using linear functions

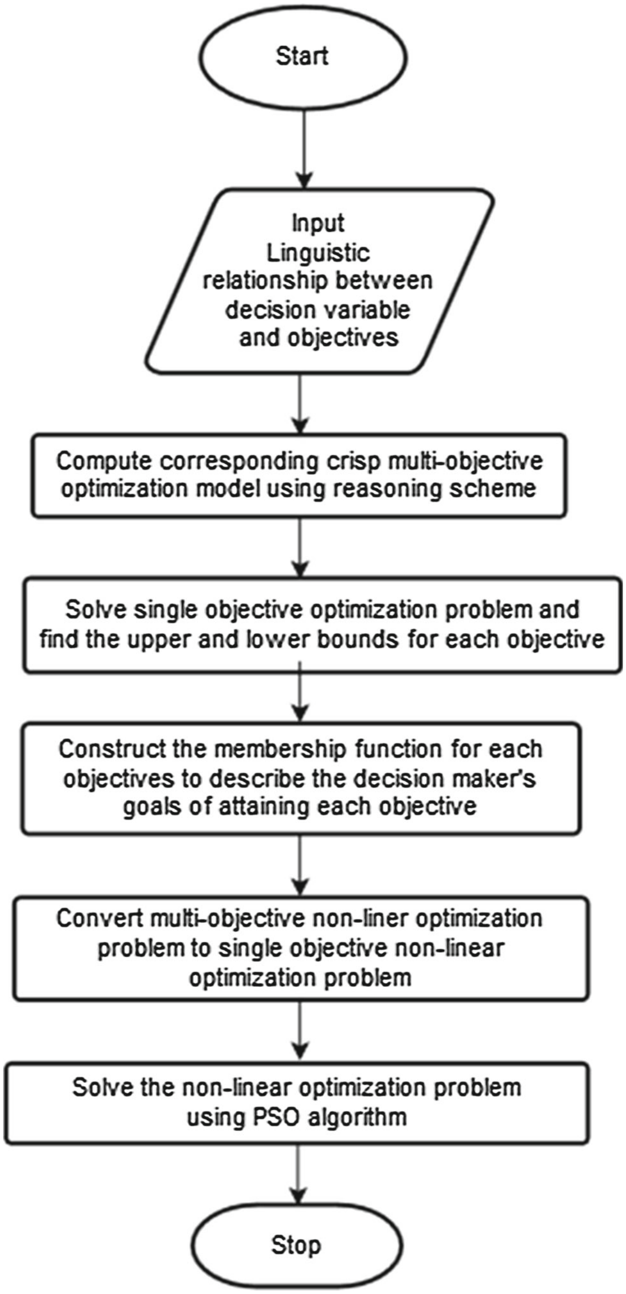

Fig. 2 Flowchart of the proposed method

Step 5 Employ suitable aggregation operator (as suggested product operator) to aggregate the fuzzy membership functions of each objective

Step 6 Use suitable non-linear optimization algorithm (such as, PSO) to solve the resulting crisp non-linear single-objective optimization problem

The flow chart of the proposed algorithm is shown in Fig. 2. A brief description of the PSO algorithm as used by us in solving the single-objective crisp non-linear optimization problem (22) is given in "Appendix".

Table 2 Linguistic value for x_1

$\mathfrak{J}(x_1)$	Fuzzy numbers	Normalized fuzzy numbers
Very low (VL)	(0; 0, 10)	(0; 0, 1.0)
Low (L)	(2.5; 2.5, 7.5)	(0.25; 0.25, 0.75)
Medium (M)	(5.0; 5.0, 5.0)	(0.5; 0.5, 0.5)
High (H)	(7.5; 7.5, 2.5)	(0.75; 0.75, 0.25)
Very high (VH)	(10.0; 10.0, 0)	(1.0; 1.0, 0)

Table 3 Linguistic value for x_2

$\mathfrak{J}(x_2)$	Fuzzy numbers	Normalized fuzzy numbers
Very low (VL)	(10; 0, 60)	(0.143; 0, 0.857)
Low (L)	(20; 10, 50)	(0.286; 0.143,0.714)
Medium low (ML)	(30; 20, 40)	(0.429; 0.286,0.571)
Medium (M)	(40; 30, 30)	(0.571; 0.428, 0.429)
Medium high (MH)	(50; 40, 20)	(0.714; 0.571, 0.286)
High (H)	(60; 50, 10)	(0.857; 0.714, 0.143)
Very high (VH)	(70; 60, 0)	(1.0; 0.857, 0)

4 Numerical illustration

In this section, two numerical examples are presented to illustrate the proposed methodology and its application in practice.

In the following example, we provide working of the proposed method in detail.

Example 1 Let us consider the following optimization problem:

Max $(f_1(x), f_2(x))$ subject to $\{0 \leq x_1 \leq 10, 10 \leq x_2 \leq 70, 6x_1 + x_2 \leq 90\}$, where

$\mathfrak{R}_1(x)$: If x_1 is high and x_2 is very high then $f_1(x) = -x_1 + x_2$ and $f_2(x) = x_1 + x_2/2$.
$\mathfrak{R}_2(x)$: If x_1 is low and x_2 is high then $f_1(x) = x_1 + x_2$ and $f_2(x) = -x_1 + x_2$.

Tables 2 and 3 represent the linguistic scales and their corresponding fuzzy numbers for different values of x_1 and x_2, respectively. The corresponding normalized values obtained using (13) are provided in third columns of respective tables.

First step of solving the above rule base multi-objective optimization problem is to derive corresponding crisp non-linear multi-objective optimization problem (16). As the consequent of each fuzzy if-then rule is described via linear function of input variables, Type II scheme is employed to find the functional form of objectives. For this purpose, let (y_1, y_2) be the input vector of the fuzzy system. Then the firing levels of the rules $\mathfrak{R}_1, \mathfrak{R}_2$ are computed by employing (9) as follows:

$$l_1 = \begin{cases} \frac{y_1}{0.75} \cdot \frac{y_2 - 0.143}{0.857} & \text{if } 0.143 \le y_1, y_2 \le 0.75 \\ \frac{1-y_1}{0.25} \cdot \frac{y_2 - 0.143}{0.857} & \text{if } 0.75 \le y_1, y_2 \le 1.0 \end{cases}$$

$$l_2 = \begin{cases} \frac{y_1}{0.25} \cdot \frac{y_2 - 0.143}{0.714} & \text{if } 0.143 \le y_1, y_2 \le 0.25 \\ \frac{1-y_1}{0.75} \cdot \frac{y_2 - 0.143}{0.714} & \text{if } 0.25 \le y_1, y_2 \le 0.857 \\ \frac{1-y_1}{0.75} \cdot \frac{1-y_2}{0.143} & \text{if } 0.857 \le y_1, y_2 \le 1.0 \end{cases}$$

It is clear that if $y_1 = 1.0$ and $y_2 = 0.143$, then no rule applies because $l_1 = l_2 = 0$. So, we can exclude the values $y_1 = 1.0$ and $y_2 = 0.143$ from the set of feasible solutions.

Table 4 Upper and lower bounds of f_1 and f_2

Objective functions	Upper bound (M_i)	Lower bound (m_i)
f_1	0.895	0.3227
f_2	0.726	0.5136

Therefore, the overall system output, interpreted as the crisp values of f_1 and f_2, at (y_1, y_2) is computed by (17) as follows:

$$f_1(y_1, y_2) = \begin{cases} \dfrac{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} .(y_2 - y_1) + \frac{y_1}{0.25} \cdot \frac{y_2-0.143}{0.714} .(y_1 + y_2)}{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} + \frac{y_1}{0.25} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.143 \le y_1 \le 0.25,\ 0.143 < y_2 \le 0.25 \\ \dfrac{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} .(y_2 - y_1) + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714} .(y_1 + y_2)}{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.25 \le y_1, y_2 \le 0.75 \\ \dfrac{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} .(y_2 - y_1) + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714} .(y_1 + y_2)}{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.75 \le y_1, y_2 \le 0.857 \\ \dfrac{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} .(y_2 - y_1) + \frac{1-y_1}{0.75} \cdot \frac{1-y_2}{0.143} .(y_1 + y_2)}{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{1-y_2}{0.143}} & \text{if } 0.857 \le y_1 < 1.0,\ 0.857 \le y_2 \le 1.0 \end{cases}$$

and

$$f_2(y_1, y_2) = \begin{cases} \dfrac{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} .(y_1 + y_2/2) + \frac{y_1}{0.25} \cdot \frac{y_2-0.143}{0.714} .(y_2 - y_1)}{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} + \frac{y_1}{0.25} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.143 \le y_1 \le 0.25,\ 0.143 < y_2 \le 0.25 \\ \dfrac{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} .(y_1 + y_2/2) + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714} .(y_2 - y_1)}{\frac{y_1}{0.75} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.25 \le y_1, y_2 \le 0.75 \\ \dfrac{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} .(y_1 + y_2/2) + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714} .(y_2 - y_1)}{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{y_2-0.143}{0.714}} & \text{if } 0.75 \le y_1, y_2 \le 0.857 \\ \dfrac{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} .(y_1 + y_2/2) + \frac{1-y_1}{0.75} \cdot \frac{1-y_2}{0.143} .(y_2 - y_1)}{\frac{1-y_1}{0.25} \cdot \frac{y_2-0.143}{0.857} + \frac{1-y_1}{0.75} \cdot \frac{1-y_2}{0.143}} & \text{if } 0.857 \le y_1 < 1.0,\ 0.857 \le y_2 \le 1.0 \end{cases}$$

Now, we employ the compositional rule of inference of individual objective functions to find out the equivalent fuzzy rule base system (14) for the given set of rules $\{\Re_1(x), \Re_2(x)\}$ as follows:

$\Re_{11}(x)$: If x_1 is high and x_2 is very high then $f_1(x) = -x_1 + x_2$.
$\Re_{21}(x)$: If x_1 is high and x_2 is very high then $f_2(x) = x_1 + x_2/2$.
$\Re_{12}(x)$: If x_1 is low and x_2 is high then $f_1(x) = x_1 + x_2$.
$\Re_{22}(x)$: If x_1 is low and x_2 is high then $f_2(x) = -x_1 + x_2$.

The individual output of the rules $\Re_{11}, \Re_{12}$ and $\Re_{21}, \Re_{22}$ are computed $z_{11} = -y_1 + y_2$ and $z_{12} = y_1 + y_2$; $z_{21} = y_1 + y_2/2$ and $z_{22} = -y_1 + y_2$.

Now, we may write the crisp multi-objective non-linear optimization model (16) for the given multi-objective optimization problem under fuzzy rule constraints as follows:

$$\begin{aligned} &\text{Maximize } f(y) = (f_1(y), f_2(y)) \\ &\text{subject to} \\ &\qquad 0.67 y_1 + 0.78 y_2 \le 1, \\ &\qquad y_1 \in [0, \ 1],\ y_2 \in [0.143, \ 1]. \end{aligned} \tag{23}$$

Now, we are going to employ the procedure describe in Sect. 3.3 to solve multi-objective optimization problem (23). First, we find the upper and lower bounds of the objectives and the results are presented in Table 4.

Now analyzing the ranges of functions $f_1(y_1, y_2)$ and $f_2(y_1, y_2)$ from Table 4, the decision maker constructs membership functions corresponding to each of the objectives. Both of the objectives being of maximizing type, their linear membership functions are constructed as follows:

Table 5 Pareto-optimal solution obtained using PSO and GA

Method	y_1	y_2	f_1	f_2	$\alpha_1(y)$	$\alpha_2(y)$	$\alpha_1(y) \times \alpha_2(y)$
GA	0.62529	0.74620	0.64409	0.63129	0.56135	0.55537	0.31176
PSO	0.61940	0.75	0.65643	0.62774	0.58292	0.53871	0.31402

Table 6 A sample of 5 rules surveyed the nature of objectives depending on the decision variables

Rule	x_1	x_2	x_3	x_4	x_5	x_6	f_1	f_2
R_1	Somewhat	Good	No	Good	High	Good	Not at all	Fully
R_2	Fully reliable	Good	No	Not at all	Not at all	Not at all	Fully	Not at all
R_3	Fully reliable	Good	Not so satisfied	Average	Very high	Good	Fully	Not at all
R_4	Somewhat	Average	Yes	Good	Below average	Average	Not at all	Fully
R_5	Somewhat	Average	No	Average	High	Average	Fully	Not at all

$$\mu_{f_1}(y) = \begin{cases} 1 & \text{for } f_1(y) > 0.895 \\ \frac{f_1(y)-0.323}{0.572} & \text{for } 0.895 > f_1(y) > 0.323 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{f_2}(y) = \begin{cases} 1 & \text{for } f_2(y) > 0.726 \\ \frac{f_2(x)-0.513}{0.213} & \text{for } 0.726 > f_2(y) > 0.513 \\ 0 & \text{otherwise} \end{cases}$$

Utilizing the above constructed membership functions for the satisfaction of the objectives at different points of the feasible regions, the equivalent single-objective non-linear optimization model for the system (23) is formulated as follows:

$$\begin{aligned} &\text{Maximize } \alpha_1(y) \times \alpha_2(y) \\ &\text{subject to} \\ &\qquad \mu_{f_i}(y) = \alpha_i(y), \quad i = 1, 2 \\ &\qquad g_1(y) = 0.67y_1 + 0.78y_2 - 1 \le 0, \\ &\qquad y_1 \in [0, \; 1], \; y_2 \in [0.143, \; 1] \end{aligned} \tag{24}$$

In order to solve the single-objective non-linear optimization problem (24), PSO algorithm is utilized and its result is compared with GA result (Table 5).

From Table 5, it is observed that both PSO and GA algorithms produce almost similar satisfactions for the objectives, however only notable fact is that GA requires more computational overhead in comparison to PSO.

Example 2 In this example, the proposed technique for solving multi-objective optimization under fuzzy rule constraints is illustrated through a problem of developing strategies for production of a company.

4.1 Problem description

Let us consider a company, say, Company XYZ, that sells various types of toys, such as dolls and doll houses, stuffed toys, puppets, transportation toys, play kitchens and house toys. The company finds that sales of its products are falling and the company is operating at a loss. In this situation, the company reviews its productions of the past several years and this review, based on mainly past experience and not on any mathematical formalism, is provided in the form of fuzzy rule-based system. As the past information may not be assessed with both precision and certainty, the information is available in the form of linguistic descriptors. To keep the matter simple let the amount of the production of six different types of toys be denoted as a set $x = (x_1, x_2, x_3, x_4, x_5, x_6)$; profit and goodwill are denoted as f_1 and f_2. The information acquired from past available data is presented in Table 6.

Table 7 Linguistic value for the decision variables x_1, x_2, x_3, x_5, x_6 and f_1, f_2

$\mathfrak{I}(x)$	Fuzzy numbers
Poor/Not at all/No	(0; 0, 100)
Below average/ Not so satisfactory/Rarely	(25; 25, 75)
Average/Moderate/Partial	(50; 50, 50)
Good/Somewhat/Satisfied/High	(75; 75, 25)
Very good/Fully/Very high/Yes	(100; 100, 0)

In the fuzzy rule-based system (given in Table 6 below), the manager, from experience, can at best provide the information about the amount of the past production of different types of toys (i.e., the linguistic values of $(x_1, x_2, \ldots, x_6)$) and the corresponding profit (f_1) and goodwill (f_2) of the company. What should be the optimal numbers of the productions of different types of toys so that the profit and goodwill earned by the company are maximum?

Fuzzy terms given in Table 6 are quantified using the membership functions given in (1)–(3). Tables 7 and 8 represent the linguistic scales and their corresponding fuzzy numbers for different values of the variables and objectives, respectively.

Table 8 Linguistic value for x_4

$\mathfrak{I}(x)$	Fuzzy numbers
Poor/Not at all/No	(10; 10, 90)
Below average/ Not so satisfactory/Rarely	(32.5; 22.5, 67.5)
Average/Moderate/Partial	(55; 45, 45)
Good/Somewhat/Satisfied/High	(77.5; 67.5, 22.5)
Very good/Fully/Very high/Yes	(100; 100, 0)

Table 9 Upper and lower bounds of f_1 and f_2

Objective functions	Upper bound (M_i)	Lower bound (m_i)
f_1	0.79786	0.31875
f_2	0.68125	0.20213

4.2 Results and discussion

The multi-objective optimization model under fuzzy rule-base constraints as described in Sect. 3 is formulated for the given system. As in the given rule-base constraints (Table 6), both the objectives are described using linguistic variables, the given system is transformed into corresponding crisp equivalent multi-objective non-linear optimization model (as formulated in (16)) using Type I approach. For this purpose, let $(y_1, y_2, \ldots, y_6)$ be the crisp input vector of the fuzzy system and the overall crisp outputs of the system, obtained with the help of (15), are denoted as $f_1(y_1, y_2, \ldots, y_6)$ and $f_2(y_1, y_2, \ldots, y_6)$. The multi-objective non-linear model for the system can be written as

$$\begin{aligned}
&\text{Maximize } f(y) = (f_1(y), f_2(y))\\
&\text{subject to}\\
&\qquad 0.15y_1 + 0.25y_2 + 0.2y_3 + 0.2y_4 + 0.1y_5\\
&\qquad + 0.1y_6 \le 1\\
&\qquad y_r \in [0, 1],\quad r = 1, 2, 3, 4, 5, 6.
\end{aligned} \tag{25}$$

where the first constraint is resource allocation constraint provided by the company. The single-objective non-linear optimization problem for (25) is constructed using the steps as described in Sect. 3.3. For each of the objectives, linear membership functions, which describe satisfaction of the objectives at different points in the feasible region, are constructed using the upper and lower bounds of each objective (provided in Table 9). By utilizing the constructed membership functions of the objectives, the single-objective non-linear optimization model for the system (25) can be formulated as follows:

$$\begin{aligned}
&\text{Maximize } \alpha_1(y) \times \alpha_2(y)\\
&\text{subject to}\\
&\qquad \mu_{f_i}(y) = \alpha_i(y),\quad i = 1, 2\\
&\qquad g_1(y) = 0.15y_1 + 0.25y_2 + 0.2y_3 + 0.2y_4\\
&\qquad\qquad + 0.1y_5 + 0.1y_6 - 1 \le 0\\
&\qquad y_k \in [0,\quad 1],\quad k = 1, 2, 3, 4, 5, 6.
\end{aligned} \tag{26}$$

Finally, the PSO algorithm is employed to solve this resultant non-linear optimization model (26) and the results are summarized in Table 10.

After transforming to the original scale, the *Company XYZ* should produce approximately 43 dolls and doll houses, 43 stuffed toys, 40 puppets, 26 transportation toys, 38 play kitchens and 36 house toys every time to achieve better profit and goodwill.

4.2.1 Comparison analysis

As mentioned earlier, the resulting single-objective non-linear optimization model (26) can be solved using any global optimization technique. Here we make a comparison of the results obtained by PSO (provided in Table 10) with the result obtained via GA. When decision makers' satisfaction corresponding to each objective is described using linear membership function, the optimal solution obtained by solving optimization model (26) with the help of PSO and GA is presented in Table 11.

One may observe from Table 11 that both PSO and GA algorithms produce almost same resultant satisfaction of objectives and the only notable fact is that GA required more computational overhead in comparison to PSO.

Another issue is the choice of membership function for describing decision makers' satisfaction of different objective functions. The use of linear membership function in describing decision makers' satisfaction may lead to loss of information (Watada 1997) since it is an approximation of non-linear membership function. Moreover, linear membership function does not allow decision maker to provide any kind of biasness towards objectives. In order to capture decision makers' biasness towards one or more objectives, we use exponential membership function as it can represent utility functions in a much more realistic way (Li and Lee 1991). An exponential membership function can be defined as follows:

Table 10 Pareto-optimal solution obtained using PSO

y_1	y_2	y_3	y_4	y_5	y_6	f_1	f_2	$\alpha_1(y)$	$\alpha_2(y)$	$\alpha_1(y) \times \alpha_2(y)$
0.4261	0. 4303	0. 4018	0. 2598	0. 3764	0. 3585	0. 5583	0.4416	0.50000	0.49998	0.24999

Table 11 Pareto-optimal solution obtained using PSO and GA with linear membership function

Method	$(y_1, y_2, y_3, y_4, y_5, y_6)$	f_1	f_2	$\alpha_1(y)$	$\alpha_2(y)$	$\alpha_1(y) \times \alpha_2(y)$
GA	(0.43875, 0.34276, 0.33438, 0.29652, 0.46322, 0.28238)	0.55831	0.44169	0.50001	0.50000	0.25001
PSO	(0.42611, 0.43035, 0.401893, 0.25981, 0.37646, 0.35851)	0.55831	0.44168	0.50000	0.49998	0.24999

Table 12 Pareto-optimal solution obtained using PSO with exponential membership function for different set of shape parameters

Shape parameters	$(y_1, y_2, y_3, y_4, y_5, y_6)$	f_1	f_2	$\alpha_1(y)$	$\alpha_2(y)$	$\alpha_1(y) \times \alpha_2(y)$
(0.1, 0.8)	(0.40519, 0.3520, 0.28348, 0.27171, 0.27982, 0.42038)	0.53587	0.46412	0.44072	0.44781	0.19736
(0.5, 1.5)	(0.38741, 0.38186, 0.39034, 0.29371, 0.27627, 0.31627)	0.52335	0.47665	0.36684	0.39117	0.14350
(1, 1)	(0.44918, 0.38471, 0.26072, 0.30521, 0.46157, 0.29590)	0.55835	0.44165	0.37756	0.37748	0.14252
(1.5, 1)	(0.42076, 0.49883, 0.28759, 0.27278, 0.32119, 0.45360)	0.57691	0.42309	0.35720	0.34104	0.12182
(0.8, 0.1)	(0.42253, 0.31141, 0.25591, 0.29221, 0.32075, 0.440919)	0.58083	0.41916	0.44788	0.44066	0.19736

Table 13 Pareto-optimal solution obtained using PSO with linear membership function for different set of weights

Weights	$(y_1, y_2, y_3, y_4, y_5, y_6)$	f_1	f_2	$\alpha_1(y)$	$\alpha_2(y)$	$\alpha_1(y)^{w_1} \times \alpha_2(y)^{w_2}$
(0.8, 0.2)	(0.35848, 0.40524, 0.39516, 0.44369, 0.44260, 0.37675)	0.70205	0.29795	0.8	0.2	0.60630
(0.6, 0.4)	(0.34755, 0.46059, 0.30779, 0.41441, 0.316604, 0.30237)	0.60622	0.39377	0.6	0.4	0.51017
(0.4, 0.6)	(0.36434, 0.27107, 0.34429, 0.28550, 0.35811, 0.29419)	0.51039	0.48960	0.4	0.6	0.51017
(0.2, 0.8)	(0.41158, 0.44484, 0.32821, 0.25107, 0.28042, 0.27625)	0.41457	0.58542	0.2	0.8	0.60620

$$\mu_{f_i}(x) = \begin{cases} 1 & \text{if } f_i(x) \geq M_i \\ \dfrac{e^{-a_i\left(\frac{f_i(x)-M}{-(M_i-m_i)}\right)} - e^{-a_i}}{1-e^{-a_i}} & \text{if } m_i \leq f_i(x) \leq M_i \\ 0 & \text{if } f_i(x) \leq m_i \end{cases}$$

where a_i ($a_i > 0$) is the shape parameter that determines the shape of membership function. Different values of shape parameter may describe decision makers' various types of desirability towards objectives (Gupta and Melhawat (2009)). Describing decision makers' satisfaction of different objectives by exponential membership functions, we solve the optimization model (26) for different values of the shape parameters with the help of PSO and the results are summarized in Table 12.

It is clear (from the Table 12) that with the changes of shape parameters in the exponential membership functions, individual satisfaction of the objectives and also the resulting overall satisfaction change significantly.

So far as, we have considered that decision maker has equal preferences for both of the objectives. In other words, decision makers insist on having no biasness towards the objectives and this case is basically uniform preference case. However, in practice decision makers may have different preferences for each of the objectives. Such preferences of the decision makers may be incorporated in the form of weights of the objectives. To capture the preference information in our model, we can reformulate (26) as follows (Chen 2001):

$$\begin{aligned} &\text{Maximize } \alpha_1(y)^{w_1} \times \alpha_2(y)^{w_2} \\ &\text{subject to} \\ &\qquad \mu_{f_i}(y) = \alpha_i(y), \quad i = 1, 2 \\ &\qquad g_1(y) = 0.15y_1 + 0.25y_2 + 0.2y_3 + 0.2y_4 \\ &\qquad\qquad + 0.1y_5 + 0.1y_6 - 1 \leq 0 \\ &\qquad y_k \in [0, \quad 1], \quad k = 1, 2, 3, 4, 5, 6. \end{aligned} \tag{27}$$

where w_1 and w_2 are the weights of the objectives f_1 and f_2, respectively, and satisfy the conditions $w_1, w_2 > 0$, $w_1 + w_2 = 1$. The varying nature of preferences of the objectives may produce different results. Table 13 depicts Pareto optimal results of the model (27) for different set of weights.

One may observe (from Table 13) that weighting factor directly impacts the overall satisfaction of individual objectives. When a lager weight is assigned to an objective in initial stage, the resulting satisfaction of that objective becomes higher. It gives the impression that uniform weights generate nearly uniform degree of satisfaction of individual objectives, as in model (26). So, by assigning appropriate weights to each of the objectives and solving the corresponding model (as (27)), decision maker can obtain the most satisfying solution.

5 Conclusion

In growing complexity of social and economic factors and rapid changes in business environment, it has become difficult for any company management to define the connection between various input decision variables and conflicting objectives of the company in precise crisp mathematical framework. Keeping this view in mind, in this paper, we have developed a fuzzy multi-objective mathematical programming problem in which the objective functions cannot be expressed precisely. It has been assumed that the information source from where some knowledge may be obtained about objective functions consists of a block of fuzzy if-then rules. The antecedent part of the rules contains some linguistic values of the decision variables and the consequence part is either a linear combination of crisp values of the decision variables or consists of linguistic values of the objective functions. We have focused in this work to solve the multi-objective optimization problem for the above situations by taking into account no-preference of decision maker regarding the objectives. Using suitable fuzzy reasoning scheme, we have obtained the exact functional form of the objective functions with respect to the decision variables and, thus, a crisp multi-objective optimization problem is formulated. The resultant problem has been solved by treating the goals of the objectives as fuzzy in nature. Linear membership function has been used for fuzzification. We have used product operator, a compensatory aggregation operator, to aggregate the membership functions of different objectives. Resulting nonlinear single-objective optimization problem has been solved using PSO and the obtained results are compared with GA results. The preferences of decision makers have also been taken into account and it has been observed from the analysis that the higher preference to any objective yields the larger degree of satisfaction of that objective in the resulting overall satisfaction.

The proposed model is a mathematical tool to the decision makers for dealing with such kind of complex scenario and aids the decision makers in making suitable decision. In future research, it will be interesting to see the application of the above model as a decision support system in may real-life problems, such as inventory management, scheduling, supplier selection, etc.

Appendix

Particle swarm optimization (PSO), introduced by (Kennedy and Eberhart 1995; Shi and Eberhart 1998a), is a nature-inspired heuristic global optimization technique. It simulates the social behavior of bird flocking or fish schooling to configure the heuristic learning mechanism. PSO normally starts with a set of initial solution (called swarm) of the decision making problems under consideration. Individual solutions are called particles and food is analogous to optimal solution. The particles are flown through a multi-dimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors. Let us assume that the dimension of the searching space is D and the number of particle present in initial solution set is l. We further assume that at sth generation the position of the rth particle is $x_r(s) = [x_{r1}(s),\ x_{r2}(s), \ldots, x_{rD}(s)]$, the velocity of the rth particle is denoted as $v_r(s) = [v_{r1}(s),\ v_{r2}(s), \ldots, v_{rD}(s)]$, the position vector of the rth particle at which best fitness encountered so far is denoted as $pbest_r(s) = [pbest_{r1}(s),\ pbest_{r2}(s), \ldots, pbest_{rD}(s)]$ and the best position of all the particle is denoted as $gbest(s) = [gbest_1(s),\ gbest_2(s), \ldots, gbest_D(s)]$. In $(s+1)^{\text{th}}$ generation the position and velocity of each particle are updated using the following rules:

$$v_r(s+1) = w.v_r(s) + c_1.uv_1. \times (pbest_r(s) - x_r(s)) + c_2.uv_2.(gbest_r(s) - x_r(s)) \quad (28)$$

$$x_r(s+1) = x_r(s) + v_r(s+1), \quad (29)$$

where the parameters c_1 and $c_2$are constants, uv_1 and $uv_2$are two random variables with uniform distribution in [0, 1] and $w(0 < w < 1)$ is called the inertia weight which controls the influence of previous velocity on new velocity. The pseudocode of the PSO algorithm is given below.

Algorithm: Pseudo code of PSO

1: Fitness function $f(x), x = (x_1, x_2, ..., x_D)$

2: Set the parameters: constant parameters c_1, c_2, inertia weight w, number of maximum generation M.

3: For each particle:
   Initialize particle position and velocity;

4: End For

5: DO:

6: For each particle:
   (a) Compute fitness value;
   (b) If the fitness value is better than the best fitness value (pbest) in history then set current value as the new pbest;

7: END For

8: Find the particle with best fitness value among the entire particles and set it as gbest

9: For each particle
   (a) Compute the particle velocity by utilizing (28);
   (b) Update particle position according to (29);

10: End For

11: WHILE maximum generation or minimum tolerance criteria is not attained.

Note For solving the optimization models given in Sect. 4, different parameters settings of the PSO algorithm and system environments are used. The optimization method is implemented in **MATLAB** and program is run on a Intel(R) Core(TM) i5-2500 CPU @ 3.30 GHz processor with 4 GB RAM under windows environment. To remove stochastic dependency 30 independent runs are made. In each run PSO parameters are set as follows: initial population size I = 20, maximum number of generation M = 100, the acceleration parameters $c_1 = c_2 = 1.5$, inertia weight w is obtained by putting $w_1 = 0.9$ and $w_2 = 0.4$ in the formula $w = (w_1 - w_2)(\frac{M-s}{M}) + w_1$ where M is the maximum number of generation and $s$denotes the current generation number (Shi and Eberhart 1998b; Clerc and Kennedy 2002). The termination criteria have been set as either limited to maximum number of 100 generations or the order of relative error 10^{-5}, whichever is achieved first. For solving each optimization problem program has been run 30 times and the best values are chosen.

References

Ali FM (2001) A differential equation approach to fuzzy vector optimization problems and sensitivity analysis. Fuzzy Sets Syst 119(1):87–95

Bector CR, Chandra S (2005) Fuzzy mathematical programming and fuzzy matrix games., Studies in fuzziness and soft computing-Springer, Berlin

Bellman RE, Zadeh LA (1970) Decision-making in a fuzzy environment. Manag Sci 17:141–164

Carlsson C, Fuller R (1998) Multiobjective optimization with linguistic variables. In: Proceedings of the sixth European congress on intelligent techniques and soft computing (EUFIT'98), Verlag Mainz, Aachen, pp 1038–1042

Chakraborty D, Guha D (2013), Multi-objective optimization based on fuzzy if-then rules. In: Proceedings of the IEEE international conference on fuzzy systems (FUZZ-IEEE 2013), Article no. 06622519

Chanas S (1989) Fuzzy programming in multi-objective linear programming-a parametric approach. Fuzzy Sets Syst 29(3):303–313

Chen C (2000) Extensions of the TOPSIS for group decision-making under fuzzy environment. Fuzzy Sets Syst 114(1):1–9

Chen L (2001) Multiobjective design optimization based on satisfaction metrics. Eng Optim 33:601–617

Cheng FY, Li D (1996) Multiobjective optimization of structures with and without control. J Guid Control Dynam 16(2):392–397

Cheng H, Huang W, Zhou Q, Cai J (2013) Solving fuzzy multi-objective linear programming problems using deviation degree measures and weighted max-min method. Appl Math Model 37:6855–6869

Clerc M, Kennedy JF (2002) The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. IEEE Trans Evol Comput 6(1):58–73

Deep K, Singh KP, Kansal ML, Mohan C (2011) An interactive method using genetic algorithm for multi-objective optimization problems modeled in fuzzy environment. Expert Syst Appl 38(3):1659–1667

Delgado M, Kacprzyk J, Verdegay JL, Vila MA (1994) Fuzzy optimization: recent advances. Physica, Wurzburg

Eiben AE, Smith JE (2003) Introduction to evolutionary computing. Springer, Berlin

Esogbue AO (1991) Computational aspects and applications of a branch and bound algorithms for fuzzy multistage decision processes. Comput Math Appl 21(11):117–127

Garg H, Sharma SP (2013) Multi-objective reliability-redundancy allocation problem using particle swarm optimization. Comput Ind Eng 64(1):247–255

Grabisch M, Marichal JL, Mesiar R, Pap E (2009) Aggregation functions. Cambridge University Press, Cambridge

Gupta P, Melhawat MK (2009) Bector-Chandra type duality in fuzzy linear programming with exponential membership functions. Fuzzy Sets Syst 160:3290–3308

Hannan EL (1981) On fuzzy goal programming. Decis Sci 12(3):522–531
Huang HZ (1997) Fuzzy multi-objective optimization decision-making of reliability of series system. Microelectron Reliab 37(3):447–449
Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, vol IV. Piscataway, NJ, Seoul, Korea, pp 1942–1948
Khalili-Damghani K, Sadi-Nezhad S (2013) A decision support system for fuzzy multi-objective multi-period sustainable project selection. Comput Ind Eng 64(4):1045–1060
Kotinis M (2014) Improving a multi-objective differential evolution optimizer using fuzzy adaptation and K -medoids clustering. Soft Comput 18(4):757–771
Lai L, Lai KK (2000) A fuzzy approach to the multiobjective transportation problem. Comput Oper Res 27:43–57
Lai YJ, Hwang CL (1992) Fuzzy mathematical programming: methods and applications. In: Proceedings of Lecture notes in economics and mathematical systems, vol 394. Springer, Berlin
Lai YJ, Hwang CL (1994) Fuzzy multiple objective decision making: methods and applications. In: Proceedings of Lecture notes in economics and mathematical systems, vol 404. Springer, Berlin
Lee ES, Li RJ (1993) Fuzzy multiple objective programming and compromise programming with Pareto optimum. Fuzzy Sets Syst 53(3):275–288
Li DF (2007) Compromise ratio method for fuzzy multi-attribute group decision making. Appl Soft Compt 7(3):807–817
Li RJ, Lee ES (1991) An exponential membership function for fuzzy multiple objective linear programming. Comput Math Appl 22(12):55–60
Li X, Wang D, Li K, Gao Z (2013) A green train scheduling model and fuzzy multi-objective optimization algorithm. Appl Math Model 37(4):2063–2073
Liu Y-J, Zhang W-G (2013) Fuzzy portfolio optimization model under real constraints. Insur Math Eco 53(3):704–711
Lu J, Wang X, Zhang L, Zhao X (2014) Fuzzy random multi-objective optimization based routing for wireless sensor networks. Soft Comput 18(5):981–994
Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer, Boston
Miller GA (1956) The magical number seven, plus or minus two: some limits on our capacity of processing information. Psychol Rev 63:81–97
Mohan C, Nguyen HT (1998) Reference direction interactive method for solving multiobjective fuzzy programming problems. Eur J Oper Res 107(3):599–613
Narasimhan R (1980) Goal programming in a fuzzy environment. Decis Sci 11(2):325–336
Nishizaki I, Sakawa M (1995) Equilibrium solutions for multiobjective bimatrix games incorporating fuzzy goals. J Optim Theory App 86(2):433–457
Rao JR, Tiwari RN, Chakraborty D (1993) Chance-constrained fuzzy goal programming. Int J Fuzzy Math 1:823–834
Sakawa M, Kato K (2000) An interactive fuzzy satisficing method for multiobjective block angular linear programming problems with fuzzy parameters. Fuzzy Sets Syst 111(1):55–69
Sakawa M, Sawada K (1994) An interactive fuzzy satisficing method for large-scale multiobjective linear programming problems with block angular structure. Fuzzy Sets Syst 67(1):5–17
Sakawa M, Yano H (1985) An interactive fuzzy satisficing method using augmented minimax problems and its application to environmental systems. IEEE Trans Syst Man Cybern 15(6):720–729
Shi Y, Eberhart RC (1998a) A modified particle swarm optimizer. In: Proceedings of IEEE international conference on evolutionary computation, IEEE Press, Piscataway, pp 69–73
Shi Y, Eberhart R C (1998b) Parameter selection in particle swarm optimization. In: Proceedings of evolutionary programming VII: EP 98. Springer, New York, pp 591–600
Slowinski R (1998) Fuzzy sets in decision analysis, operations research, and statistics. Kluwer Academic Publishers, New York
Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 15(1):116–132
Thapar A, Pandey D, Gaur SK (2012) Satisficing solutions of multi-objective fuzzy optimization problems using genetic algorithm. Appl Soft Comput 12(8):2178–2187
Tsukamoto Y (1979) An approach to fuzzy reasoning method. In: Gupta MM, Ragade RK, Yager RR (eds) Advances in fuzzy set theory and applications. North-Holland, New York
Watada J (1997) Fuzzy portfolio selection and its applications to decision making. Tatra Mt Math Publ 13:219–248
Wu HC (2004) A solution concept for fuzzy multiobjective programming problems based on convex cones. J Optim Theory Appl 121(2):397–417
Wu HC (2008) Using the technique of scalarization to solve the multiobjective programming problems with fuzzy coefficients. Math Comp Model 48(1–2):232–248
Yano H, Sakawa M (2009) A fuzzy approach to hierarchical multi-objective programming problems and its application to an industrial pollution control problem. Fuzzy Sets Syst 160(21):3309–3322
Zadeh LA (1965) Fuzzy sets. Inform Control 8:338–353
Zadeh LA (1975a) The concept of a linguistic variable and its application to approximate reasoning part I. Inform Sci 8(3):199–249
Zadeh LA (1975b) The concept of a linguistic variable and its application to approximate reasoning part II. Inform Sci 8(4):301–357
Zadeh LA (1975c) The concept of a linguistic variable and its application to approximate reasoning part II. Inform Sci 9(1):43–80
Zhang X, Huang GH, Chan CW, Liu Z, Lin Q (2010) A fuzzy-robust stochastic multiobjective programming approach for petroleum waste management planning. Appl Math Model 34(10):2778–2788
Zimmermann HJ (1978) Fuzzy programming and linear programming with several objective functions. Fuzzy Sets Syst 1(1):45–55
Zimmermann HJ (1996) Fuzzy set theory and its applications, 2nd edn. Kluwer Academic Publishers, New York