

Attribute weight computation in a decision making problem by particle swarm optimization

Satyajit Das¹ · Debashree Guha¹

Received: 30 November 2015 / Accepted: 4 October 2017 / Published online: 23 October 2017
© The Natural Computing Applications Forum 2017

Abstract The objective of this paper is to introduce a method for computing weights of attributes in a decision making problem under intuitionistic fuzzy environment. Many weight generation methods exist in the literature under intuitionistic fuzzy setting, but they have some limitations which can be pointed out as: the entropy measures used in entropy weight methods are invalid in many situations and also there are lots of entropy formulae for intuitionistic fuzzy sets, which will be better to use, and thus a confusion may arise; the other weight generation methods may lose some information since it needs to transform the intuitionistic fuzzy decision matrix into an interval-valued decision matrix. This conversion distorts experts original opinions. In this point of view, to overcome these demerits, we develop a weight generation method without changing the original decision information. The proposed method maximizes the average degree of satisfiability and minimizes the average degree of non-satisfiability of each alternative over a set of attributes, simultaneously. This leads to formulate a multi-objective programming problem (MOPP) to compute the final comprehensive value for each alternative. The scenario of an MOPP itself is subjective and can be modeled by fuzzy decision making problem due to the conflicting objectives and the way of human choice on conflict resolution. This problem is solved by using particle swarm optimization scheme, and the evaluation procedure is illustrated by

means of a numerical example. This work has also justified the proposed approach by analyzing a comparative study.

Keywords Intuitionistic fuzzy set · Multi-objective programming problem · Fuzzy optimization · Particle swarm optimization

1 Introduction

An important generalization of the classical fuzzy set theory is the theory of intuitionistic fuzzy set (IFS) introduced by Atanassov [1]. During the last decades, IFS theory has played a significant role for dealing with incomplete or inexact information present in real-world applications [2–6] including decision making problems. It is observed from the decision making problems that the weights of the attributes are determined usually beforehand. However, the proper assessment of attribute weights plays a vital role in multi-attribute decision making (MADM) problem as the variation of weights may affect on the final ranking order of the alternatives. In order to compute the weights of the attributes under intuitionistic fuzzy environment, several methods have been developed. For example, based on the technique for order of preference by similarity to ideal solution (TOPSIS) [7], a fractional programming model is developed by Li et al. [8] to determine attribute weights. By using TOPSIS method, Yue [9] presented a new method for group decision making problem with a weight determination process. On the basis of maximizing deviation method, Wei [10] introduced two optimization models to compute attribute weights where the information about attribute weights is completely unknown or partly known. In the same direction, Chou et al. [11] and Yeh and Chang [12] developed a weighting

✉ Satyajit Das
satyajitnit.das@gmail.com

Debashree Guha
debashree@iitp.ac.in

¹ Department of Mathematics, IIT Patna, Patna, India

system of attributes under group decision-making conditions. Li [13] defined several linear programming models by using IFSs to determine attribute weights by maximizing the comprehensive values of alternatives over attributes. Lin et al. [14] provide a linear programming model which is much simpler than Li's [13] method to generate weights of attributes. In terms of determining objective weights, one of the most representative approaches is the entropy method [15]. Ye [16] established a weight generation method where attribute weights are completely unknown based on weighted correlation coefficient measure with entropy. Wu and Zhang [17] introduced a linear programming model to determine the optimal weight of criteria based on intuitionistic fuzzy weighted entropy. By using gray relational analysis (GRA), Wei [18, 19] computed weights of criteria in a decision making problem.

Although numerous weight computation methods have been developed under intuitionistic fuzzy environment, they have some limitations which can be listed as follows: the entropy measure used in entropy weight methods [15–17] is invalid in many situations, and also there are lots of entropy formulae for IFSs, which will be better to use and thus should be improved; the other weight generation methods [13, 14] may loss some information since these processes need to be transformed the intuitionistic fuzzy decision matrix into an interval-valued decision matrix. This conversion distorts experts original opinions. In this point of view, without changing the original decision information, we have developed a weight generation method by maximizing the average degree of satisfiability and minimizing the average degree of non-satisfiability of each alternative over a set of attributes, simultaneously in the form of multi-objective programming problem (MOPP), which is nonlinear. A global optimization process is required to solve this nonlinear problem. In this work, we employ a recently developed optimization technique, namely particle swarm optimization (PSO) to find the compromise solution of MOPP under the intuitionistic fuzzy environment. Then, an approach to MADM problem in which the ratings of alternatives on the basis of attributes are expressed as IFSs is developed with incomplete attribute weights information.

The paper is organized as follows: in Sect. 2, primary discussions of the various useful ideas including MOPP, PSO, penalty function and IFSs are provided in brief. The proposed approach to determine attribute weights under intuitionistic fuzzy environment is presented in Sect. 3. In Sect. 4, a numerical example is studied for clarifying our proposed method and the comparison analysis is also conducted. Finally, a concrete conclusion is drawn in Sect. 5.

2 Preliminary

In this section, we carry out brief introductions to MOPP, PSO, penalty function and IFSs that will be required for our subsequent developments. We start by recalling the concepts of MOPP.

2.1 Multi-objective programming problem

An MOPP is a design process that optimizes a vector of objective functions in a systematic way by considering all the objective functions simultaneously. A general MOPP can be defined as follows:

$$\begin{aligned} \max(\min) f(x) &= (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{subject to} \quad &A_i x \leq b_i, \quad i = 1, 2, \dots, m \\ &x \geq 0 \end{aligned} \quad (1)$$

where $f_1(x), f_2(x), \dots, f_n(x)$ are n individual objective functions, $x \in \mathbb{R}^n$ is a vector of decision variables, $A_i \in \mathbb{R}^{m \times n}$ and $b_i \in \mathbb{R} (i = 1, 2, \dots, m)$. The MOPP is designed here to find the decision variables x which optimizes a vector of objective functions $f(x) = ((f_1(x), f_2(x), \dots, f_n(x)))$ in the feasible decision space. Due to the conflicting nature of objective functions, it is not necessary that MOPP has an optimal solution that maximizes/minimizes all the objective functions simultaneously. This is the scenario of tradeoff between objective functions. This leads us to find the best compromise solution, called Pareto optimal solution [20]. Generally, by a Pareto optimal solution, we mean a solution for which improvement of one objective can be achieved only at the expense of another [21]. Mathematically, it can be represented as follows:

Definition 1 [20] A solution vector $x^* \in X$ is said to be a Pareto optimal to MOPP if there does not exist any $x \in X$ such that $f_i(x) \geq f_i(x^*)$, for all i and $f_i(x) > f_i(x^*)$ for at least one $i = 1, 2, \dots, m$.

In general, the solution procedure of an MOPP is classified into following two categories: classical methods and evolutionary methods. From last few decades, many classical methods [20], such as weighted sum method, ϵ -constraint method, value function method, goal programming, interactive method, have been used to solve an MOPP. Most of the classical methods are not efficient for solving an MOPP due to get stuck at a suboptimal solution. To overcome this situation, over the last decade, evolutionary algorithms, such as genetic algorithm (GA) [22, 23], PSO [24–26], ant colony optimization [27], have been extensively used in the field of science and engineering disciplines.

In order to cope with imprecise information present in real-life problems, the study of MOPP is started in fuzzy environment. Fuzzy MOPPs are studied by Bellman and Zadeh [28], Zimmermann [29], Sakwa and Yano [30] and many others. Cheng et al. [31] solved fuzzy multi-objective linear programming problem by utilizing deviation method and weighted max–min method. By extending the concept of TOPSIS approach, a methodology [32] is developed for solving multiple objective decision making problems. Recently, Deep et al. [33] proposed an interactive method by using GA for solving MOPP in fuzzy environment. GA is also used by Sakwa and Yauchi [34] for the same purpose. A more comprehensive detail of evolutionary algorithms for MOPP can be found in [20].

2.2 Particle swarm optimization

In order to solve a nonlinear optimization problem, PSO is one of the most successful and widely used algorithms. PSO, introduced by Kennedy and Eberhart [24], is a population-based stochastic global optimization method which is inspired by bird flocking, fish schooling and swarming theory. PSO is developed on the basis of communication and interaction, i.e., information exchange between members, called ‘particles,’ of the population, called ‘swarm.’ During movement, each particle adjusts its best position (local best) according to its own previous best position and the best position attained by any neighbor particles (global best). PSO starts with a swarm of particles whose positions are initial solutions and the velocities are randomly initialized in the search space. Based on the local best and global best information, particles update their positions and velocities.

In each iteration, the velocity and position of the particle are updated according to the following equations.

$$v_i(k + 1) = \xi(k)(\omega v_i(k) + c_1 \cdot r_{i1}(k) \cdot (x_i^l(k) - x_i(k)) + c_2 \cdot r_{i2}(k) \cdot (x_i^g(k) - x_i(k))) \tag{2}$$

$$x_i(k + 1) = x_i(k) + v_i(k + 1) \tag{3}$$

where $i(=1,2,\dots,n)$ represents the number of particles in population and n is the population size; $v_i(k + 1)$ is the velocity of i th particle in the $(k + 1)$ th iteration; $\xi(k)$ is the constriction factor which controls and constricts velocities of particles; ω is the inertia weight; r_{i1} and r_{i2} are random numbers generated from a uniform distribution in the interval $[0,1]$; c_1 and c_2 , called ‘cognitive’ and ‘social’ parameter, respectively, are positive constants; $x_i(k + 1)$ is the position of i th particle in the $(k + 1)$ th iteration. So, at each generation, Eq. (2) is performed as a new velocity, whereas Eq. (3) represents the new position of particles.

The performance of each particle is evaluated according to fitness function which is problem dependent. Generally,

in optimization problem, the objective function is treated as the fitness function.

From the applicability point of view, in this work, we utilize PSO as a tool to solve the nonlinear constrained optimization problem. The constrained optimization problem is changed into the equivalent unconstrained problem by using penalty function approach which is discussed in the next subsection. Now, a brief description of a pseudo code of PSO algorithm is given below.

Algorithm: Pseudo code of particle swarm optimization (PSO)

-
- Step-1: Select objective function: $f(x); x = (x_1, x_2, \dots, x_n)$ as a fitness function
 - Step-2: Set iteration number $k = 0$
 - Step-3: Initialize particle position and velocity for each particle
 - Step-4: Initialize the particle’s best known position to its initial position, i.e., $x_i^l(k) = x_i(k)$
 - Step-5: Repeat
 - Step-6: Set iteration number $k = k + 1$
 - Step-7: Update the best known position $x_i^l(k)$ of each particle and the swarm’s best known position $x_i^g(k)$
 - Step-8: Evaluate particle velocity and position according to Eqs. (2) and (3)
 - Step-9: Continue until termination criterion met
-

2.3 Penalty function

Constrained optimization problems have two types of solutions, one is feasible which satisfies all the given constraints, whereas another one is infeasible which violates at least one of them. In order to solve a constrained optimization problem, one of the successful approaches is to use of penalty function [35]. In the penalty function approach, constrained optimization problem is reduced to unconstrained optimization problem by introducing a penalty function and making a single-objective function. The idea behind this method is that penalty function assigns zero penalty if the point is feasible one and it assigns positive penalty if the point is infeasible. The penalty function can be defined as

Definition 2 [36] A function $P : \mathbb{R}^n \rightarrow \mathbb{R}$ is called a penalty function if P satisfies following properties:

1. $P(x)$ is continuous;
2. $P(x) \geq 0 \quad \forall x \in \mathbb{R}^n$;
3. $P(x) = 0 \quad \forall$ feasible x ;
4. $\lim P(x) = \infty$ if $|x| \rightarrow \infty$.

Now, the constrained problem (1) can be transformed to unconstrained optimization problem by using a penalty function as

$$\max F(x) = f(x) + \alpha(k)P(x) \tag{4}$$

where k is the current iteration number, $\alpha(k)$ is the penalty parameter and $\alpha(k) > 0, \alpha(k) \leq \alpha(k + 1) \quad \forall k$, in practice $\{\alpha_k\}$ is taken as $\{\alpha_1 = 1, \alpha_2 = 10, \alpha_3 = 100, \alpha_4 = 1000, \dots\}$ and penalty function $P(x)$ is defined as

$$P(x) = \sum_{i=1}^m (q_i(x))^r \tag{5}$$

where $q_i(x) = \max\{0, g_i(x)\}$, $g_i(x) = A_i x - b_i$ and r is the power of a penalty function. In general r is taken as $r = 1$ or $r = 2$.

2.4 Intuitionistic fuzzy set

An important generalization of classical fuzzy set theory [37] is the theory of IFS [1] which comprehensively portrays the uncertainty of human beings when providing expert’s opinion over the objects. The definition of IFS is given in below.

An IFS a in the universe X can be expressed as a set of ordered triplets, $a = \{(x, \mu_a(x), \nu_a(x)) : x \in X\}$, where $\mu_a : X \rightarrow [0, 1]$ is the degree of belongingness and $\nu_a : X \rightarrow [0, 1]$ is the degree of non-belongingness of x in a . They satisfy the relation $0 \leq \mu_a(x) + \nu_a(x) \leq 1 \quad \forall x \in X$. Each fuzzy set is a special case of IFS, and it can be represented as $a = \{(x, \mu_a(x), 1 - \mu_a(x)) : x \in X\}$. The quantity $\pi_a(x) = 1 - \mu_a(x) - \nu_a(x)$ is called the degree of hesitation (indeterminacy) of x in a . When $\pi_a(x) = 0 \quad \forall x \in X$, IFS becomes fuzzy set.

For the sake of simplicity now onwards, we will write an IFS a as $a = (\mu_a, \nu_a)$ throughout the paper.

3 An approach to MADM problem with incomplete attribute weight information

In this section, we consider a framework for MADM problem with incomplete attribute weight information where ratings of alternatives over attributes are given as IFSs.

3.1 Problem formulation

Now, a MADM problem with IFSs is presented in which information of attribute weights is partly known. A multi-attribute single-expert decision process is designed with n attributes $C = \{C_1, C_2, \dots, C_n\}$ to evaluate m number of alternatives $A = \{A_1, A_2, \dots, A_m\}$. Assume that $w =$

$(w_1, w_2, \dots, w_n)^T$ be the weight vector of attributes, where $w_j \in [0, 1], j = 1, 2, \dots, n$ which are partly known and $\sum_{j=1}^n w_j = 1$.

Suppose, IFS $a_{ij} = (\mu_{a_{ij}}, \nu_{a_{ij}})$ denotes the expert’s assessment corresponding to j^{th} attribute $C_j (j = 1, 2, \dots, n)$ to evaluate i^{th} alternative $A_i (i = 1, 2, \dots, m)$.

The MADM process can be described in the following way:

Step 1: Construction of decision matrix

A MADM problem with IFSs can be represented concisely in a matrix form as follows:

$$D = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} & \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \dots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \end{matrix}$$

where each a_{ij} is IFS $a_{ij} = (\mu_{a_{ij}}, \nu_{a_{ij}})$.

Step 2: Computation of comprehensive values of each alternative

The comprehensive values for each alternative are calculated by weighted intuitionistic fuzzy arithmetic mean operator [38]. Let Z_i be the comprehensive values corresponding to the alternative $A_i (i = 1, 2, \dots, m)$, with

$$Z_i = (P_i, Q_i) = \left(1 - \prod_{j=1}^n (1 - \mu_{a_{ij}}^{w_j}), \prod_{j=1}^n \nu_{a_{ij}}^{w_j} \right) \tag{6}$$

where $P_i = 1 - \prod_{j=1}^n (1 - \mu_{a_{ij}}^{w_j})$ represents the membership degree of the comprehensive value (called the degree of satisfiability) and $Q_i = \prod_{j=1}^n \nu_{a_{ij}}^{w_j}$ represents the non-membership degree of the comprehensive value (called the degree of non-satisfiability) of the alternative A_i .

3.2 Estimation of attribute weights by constructing MOPP

In this subsection, an attribute weight determination method in a decision making process is developed by maximizing the degree of satisfiability and minimizing the degree of non-satisfiability. The methodology consists of following two steps:

Step 3: *MOPP formulation*

Sometimes, it may happen that decision maker only possesses partial information about attribute weights due to the increasing complexity of real-world decision situations. Based on the weighted intuitionistic fuzzy arithmetic values, we select the optimal weight vector $w = (w_j)^T (j = 1, 2, \dots, n)$ to maximize membership degree of the comprehensive values and minimize the non-membership degree of the comprehensive values of alternatives for all the attributes. For each alternative $A_i (i = 1, 2, \dots, m)$, the final comprehensive value can be computed through the following multi-objective programming model (Model-1):

$$\text{Model - 1 : } \alpha_i = \max_w P_i = 1 - \prod_{j=1}^n (1 - \mu_{a_{ij}}^{w_j})$$

$$\beta_i = \min_w Q_i = \prod_{j=1}^n \nu_{a_{ij}}^{w_j}$$

subject to $w \in H,$

$$\sum_{j=1}^n w_j = 1,$$

$$w_j \geq 0, \quad j = 1, 2, \dots, n$$

where H is the set of incomplete information about attribute weights. The objective function $\max_w P_i$ implicates that average membership values for each alternative are maximized, i.e., the degree of satisfiability of each alternative over a set of attributes is maximized. The objective function $\min_w Q_i$ represents that average non-membership values for each alternative are minimized, i.e., the degree of non-satisfiability of each alternative over a set of attributes is minimized.

Step 4: *Fuzzy MOPP formulation*

In the MOPP scenario, it is worth noticing that multiple objectives conflict with each other and thus, in multi-objective optimization there is no optimal solution for all the objectives. Thus, the notion of Pareto optimality [20] or efficiency has been introduced in MOPP instead of the optimality concept for single-objective optimization. Decisions with Pareto optimality or efficiency cannot be uniquely determined. However, for practical applications, 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 aspect, due to the imprecision or fuzziness inherent in human judgments, it is quite natural to assume that the decision maker may have a fuzzy goal for each of the objective functions. The fuzzy goals of the decision maker for each of the objective functions can be incorporated in the problem formulation of MOPP via fuzzy sets, which are characterized by membership functions [39].

In order to construct the membership function of each objective, first we solve the following two crisp single-objective optimization problems (one is maximization and one is minimization) separately. Therefore, for each objective $P_i (i = 1, 2, \dots, m)$, the following two single-objective optimization problems are solved:

$$\max/\min P_i$$

$$\text{subject to } w \in H, \sum_{j=1}^n w_j = 1, w_j \geq 0, \quad j = 1, 2, \dots, n$$

to get two individual values α_i^L (minimum value of the objective function P_i) and α_i^U (maximum value of the objective function P_i) for every objective function P_i . In the same way, for each objective $Q_i (i = 1, 2, \dots, m)$, the following two single-objective optimization problems are solved:

max/min Q_i

$$\text{subject to } w \in H, \sum_{j=1}^n w_j = 1, w_j \geq 0, \quad j = 1, 2, \dots, n$$

to generate two individual values β_i^L (minimum value of the objective function Q_i) and β_i^U (maximum value of the objective function Q_i) for every objective function Q_i .

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

$$\mu_{P_i}(w) = \begin{cases} 1, & \text{for } P_i \geq \alpha_i^U, \\ 1 - \frac{\alpha_i^U - P_i}{\alpha_i^U - \alpha_i^L}, & \text{for } \alpha_i^L \leq P_i \leq \alpha_i^U, \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

and

$$\mu_{Q_i}(w) = \begin{cases} 1, & \text{for } Q_i \leq \beta_i^L, \\ 1 - \frac{Q_i - \beta_i^L}{\beta_i^U - \beta_i^L}, & \text{for } \beta_i^L \leq Q_i \leq \beta_i^U, \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The membership functions $\mu_{P_i}(w)$ (Fig. 1) and $\mu_{Q_i}(w)$ (Fig. 2) represent the degree of satisfactions of the decision maker corresponding to the objective functions P_i and Q_i , respectively, as a value between zero and one. Having elicited the membership functions $\mu_{P_i}(w)$ and $\mu_{Q_i}(w)$ ($i = 1, 2, \dots, m$) for each of the objective functions P_i and Q_i ($i = 1, 2, \dots, m$), respectively, through the interaction of decision maker, the MOPP (Model 1) can be transformed into the fuzzy MOPP defined by

$$\max(\mu_{P_1}(w), \mu_{P_2}(w), \dots, \mu_{P_m}(w), \mu_{Q_1}(w), \mu_{Q_2}(w), \dots, \mu_{Q_m}(w))$$

One of the important steps in fuzzy MOPP is the aggregation of membership values of the objective functions to reduce it into a single-objective optimization problem. The

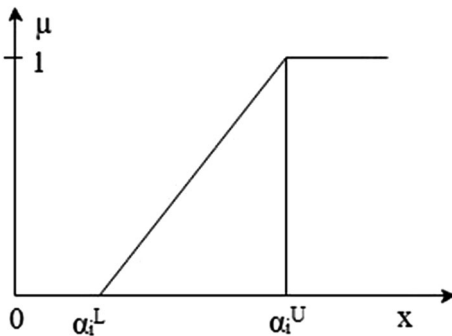


Fig. 1 Membership function for the min type objective functions

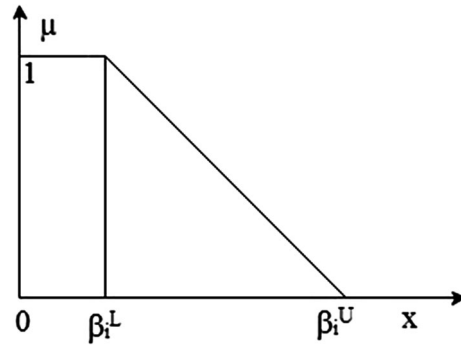


Fig. 2 Membership function for the min type objective functions

forementioned fuzzy MOPP can be reduced into the single-objective optimization problem based on the concept of aggregation operator in the following way:

$$\max f(\mu_{P_1}(w), \mu_{P_2}(w), \dots, \mu_{P_m}(w), \mu_{Q_1}(w), \mu_{Q_2}(w), \dots, \mu_{Q_m}(w))$$

where f represents an aggregation function. It is to be noted that the value of $f(\mu_{P_1}(w), \mu_{P_2}(w), \dots, \mu_{P_m}(w), \mu_{Q_1}(w), \mu_{Q_2}(w), \dots, \mu_{Q_m}(w))$ can be interpreted as the overall degree of satisfaction of decision makers' fuzzy goals. However, one may use arithmetic mean, geometric mean, minimum operator, product operator as an aggregation function f . To aggregate the objective functions, among the several choices, weighted arithmetic mean [40] and minimum operator [31] are often used in the literature. Inspired by the work [28, 29, 31], we use minimum operator as an aggregation function which may be given as follows:

$$\begin{aligned} f(\mu_{P_1}(w), \mu_{P_2}(w), \dots, \mu_{P_m}(w), \mu_{Q_1}(w), \mu_{Q_2}(w), \dots, \mu_{Q_m}(w)) \\ = \min(\mu_{P_1}(w), \mu_{P_2}(w), \dots, \mu_{P_m}(w), \mu_{Q_1}(w), \mu_{Q_2}(w), \dots, \mu_{Q_m}(w)) \end{aligned}$$

Hence, eliciting the choice of aggregation function f , the aforementioned optimization problem (Model 1) turns into the following single-objective nonlinear optimization problem (Model 2) defined as follows:

Model-2 : max λ

$$\text{subject to } \lambda \leq \mu_{P_i}(w) \quad (i = 1, 2, \dots, m),$$

$$\lambda \leq \mu_{Q_i}(w) \quad (i = 1, 2, \dots, m),$$

$$w \in H, \sum_{j=1}^n w_j = 1, w_j \geq 0, \quad j = 1, 2, \dots, n$$

$$\lambda \in [0, 1]$$

where $\lambda = \min_{i=1, \dots, m} \{\mu_{P_i}(w), \mu_{Q_i}(w)\}$ is the satisfactory level.

As Model-2 is nonlinear in nature with multiple variables, it is really a difficult task to find out the gradient vector and thus it requires some effective methods and algorithms for finding the global solution. One of the most useful processes that exists in the literature to find out the

global solution of the nonlinear optimization problem is the evolutionary algorithmic (EA) approach [41]. The advantage of this EA technique is that it does not require any kind of pre-assumptions, such as continuity, differentiability of objective functions [39].

In the present work, we employ a recently developed EA technique, namely PSO [24], to solve Model-2. In order to solve Model-2, first we convert the constrained optimization problem into unconstrained optimization problem by a penalty function. The PSO is implemented by utilizing the algorithm in Matlab R2013a. The solutions of the above optimization problem (Model-2) provide weights of the criteria. If (w^*, λ^*) is a global optimal solution of Model-2, then it implies that the maximum degree of overall satisfaction λ^* is achieved for the solution w^* .

3.3 Computation of the final aggregated values for each alternative and ranking of the alternatives

Final weighted aggregated values of alternatives A_i are determined by utilizing weights of the attributes derived from the above optimization problem and the weighted intuitionistic fuzzy arithmetic mean operator [38] (Eq. 6).

3.4 Ranking of the alternatives

Finally, ranking of all the alternatives is done according to the non-increasing order of the score function [42] defined as follows:

$$S(Z_i) = P_i - Q_i, \quad i = 1, 2, \dots, m \tag{9}$$

4 Numerical example

In this section, a numerical example is considered to illustrate the feasibility and effectiveness of the proposed weight generation method of attributes in a decision making problem under intuitionistic fuzzy environment.

An investment company wants to invest money in the best possible option (adapted from [2]). The three possible alternatives to invest money are: (1) A_1 is a car company; (2) A_2 is a food company; (3) A_3 is a computer company. The investment company must take a decision according to the following three attributes: C_1 : the risk analysis; C_2 : the growth analysis; C_3 : the environmental impact analysis. The ratings of the alternatives over the attributes are evaluated by utilizing IFSs. Now, the weights of attributes and ordering of the alternatives are computed as follows.

Step 1: Construction of decision matrix

Ratings of the alternatives over the attributes are given in the following matrix:

$$D = \begin{matrix} & C_1 & C_2 & C_3 \\ A_1 & (0.7, 0.2) & (0.5, 0.3) & (0.4, 0.4) \\ A_2 & (0.6, 0.3) & (0.8, 0.2) & (0.7, 0.1) \\ A_3 & (0.8, 0.1) & (0.6, 0.4) & (0.8, 0.2) \end{matrix}$$

Step 2: Computation of comprehensive values of each alternative

The comprehensive value of each alternative is calculated by weighted intuitionistic fuzzy arithmetic mean operator. Let Z_1, Z_2 and Z_3 be the comprehensive values corresponding to the alternatives A_1, A_2 and A_3 , respectively, and

$$\begin{aligned} Z_1 &= (P_1, Q_1) = (1 - 0.3^{w_1} 0.5^{w_2} 0.6^{w_3}, 0.2^{w_1} 0.3^{w_2} 0.4^{w_3}) \\ Z_2 &= (P_2, Q_2) = (1 - 0.4^{w_1} 0.2^{w_2} 0.3^{w_3}, 0.3^{w_1} 0.2^{w_2} 0.1^{w_3}) \\ Z_3 &= (P_3, Q_3) = (1 - 0.2^{w_1} 0.4^{w_2} 0.2^{w_3}, 0.1^{w_1} 0.4^{w_2} 0.2^{w_3}) \end{aligned}$$

where w_1, w_2 and w_3 are the weights of the attributes C_1, C_2 and C_3 , respectively. Attribute weights w_1, w_2 and w_3 are computed as follows:

Step 3: MOPP formulation

Assume that the information about attribute weights w_j is partly known and the partial weight information is given as follows:

$$H = \{0.25 \leq w_1 \leq 0.75, 0.15 \leq w_2 \leq 0.6, 0.2 \leq w_3 \leq 0.35\}.$$

By using Model-1, described in Sect. 3.2, we have,

$$\begin{aligned} \max P_1 &= 1 - 0.3^{w_1} 0.5^{w_2} 0.6^{w_3} \\ \max P_2 &= 1 - 0.4^{w_1} 0.2^{w_2} 0.3^{w_3} \\ \max P_3 &= 1 - 0.2^{w_1} 0.4^{w_2} 0.2^{w_3} \\ \min Q_1 &= 0.2^{w_1} 0.3^{w_2} 0.4^{w_3} \\ \min Q_2 &= 0.3^{w_1} 0.2^{w_2} 0.1^{w_3} \\ \min Q_3 &= 0.1^{w_1} 0.4^{w_2} 0.2^{w_3} \end{aligned}$$

$$\begin{aligned} \text{subject to } & 0.25 \leq w_1 \leq 0.75, \\ & 0.15 \leq w_2 \leq 0.60, \\ & 0.20 \leq w_3 \leq 0.35, \\ & w_1 + w_2 + w_3 = 1, \\ & w_j \geq 0, j = 1, 2, 3 \end{aligned}$$

Step 4: *Fuzzy MOPP formulation*

To get a maximum value and a minimum value of each objective function, we have solved $2 \times 6 = 12$ crisp single-objective optimization problems (one maximization and one minimization for each single optimization problem) by PSO scheme and the values are shown in Table 1.

The membership functions corresponding to objective functions, for describing the fuzziness involved in each objective, are defined as follows:

$$\mu_{P_1}(x) = \begin{cases} 1, & \text{for } P_1 \geq 0.6280, \\ 1 - \frac{0.6280 - P_1}{0.6280 - 0.5309}, & \text{for } 0.5309 \leq P_1 \leq 0.6280, \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$\mu_{P_2}(x) = \begin{cases} 1, & \text{for } P_2 \geq 0.7421, \\ 1 - \frac{0.7421 - P_2}{0.7421 - 0.6596}, & \text{for } 0.6596 \leq P_2 \leq 0.7421, \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$\mu_{P_3}(x) = \begin{cases} 1, & \text{for } P_3 \geq 0.7781, \\ 1 - \frac{0.7781 - P_3}{0.7781 - 0.7072}, & \text{for } 0.7072 \leq P_3 \leq 0.7781, \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$\mu_{Q_1}(x) = \begin{cases} 1, & \text{for } Q_1 \leq 0.2441, \\ 1 - \frac{Q_1 - 0.2441}{0.2997 - 0.2441}, & \text{for } 0.2441 \leq Q_1 \leq 0.2997, \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$\mu_{Q_2}(x) = \begin{cases} 1, & \text{for } Q_2 \leq 0.24, \\ 1 - \frac{Q_2 - 0.2400}{0.2823 - 0.2400}, & \text{for } 0.2400 \leq Q_2 \leq 0.2823, \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$\mu_{Q_3}(x) = \begin{cases} 1, & \text{for } Q_3 \leq 0.1414, \\ 1 - \frac{Q_3 - 0.1414}{0.2462 - 0.1414}, & \text{for } 0.1414 \leq Q_3 \leq 0.2462, \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

So, by utilizing Eqs. (10)–(15) and Model-2, the single-objective optimization problem can be formulated as:

Table 1 Optimization values

Optimization	Objective functions					
	P_1	P_2	P_3	Q_1	Q_2	Q_3
maximization	0.6280	0.7421	0.7781	0.2997	0.2823	0.2462
minimization	0.5309	0.6596	0.7072	0.2441	0.2400	0.1414

$$\begin{aligned} & \max \lambda \\ & \text{subject to } \lambda \leq \mu_{P_1}, \lambda \leq \mu_{P_2}, \lambda \leq \mu_{P_3}, \\ & \quad \lambda \leq \mu_{Q_1}, \lambda \leq \mu_{Q_2}, \lambda \leq \mu_{Q_3}, \\ & \quad 0.25 \leq w_1 \leq 0.75, \\ & \quad 0.15 \leq w_2 \leq 0.60, \\ & \quad 0.20 \leq w_3 \leq 0.35, \\ & \quad w_1 + w_2 + w_3 = 1, \\ & \quad w_j \geq 0, j = 1, 2, 3, \\ & \quad \lambda \in [0, 1] \end{aligned}$$

After simplification, we will get,

$$\begin{aligned} & \max \lambda \\ & \text{subject to } 0.0971\lambda + 0.3^{w_1}0.5^{w_2}0.6^{w_3} \leq 0.469 \\ & \quad 0.0825\lambda + 0.4^{w_1}0.2^{w_2}0.3^{w_3} \leq 0.3404 \\ & \quad 0.0709\lambda + 0.2^{w_1}0.4^{w_2}0.2^{w_3} \leq 0.2928 \\ & \quad 0.0556\lambda + 0.2^{w_1}0.3^{w_2}0.4^{w_3} \leq 0.2997 \\ & \quad 0.0423\lambda + 0.3^{w_1}0.2^{w_2}0.1^{w_3} \leq 0.2823 \\ & \quad 0.1048\lambda + 0.1^{w_1}0.4^{w_2}0.2^{w_3} \leq 0.2468 \\ & \quad 0.25 \leq w_1 \leq 0.75, \\ & \quad 0.15 \leq w_2 \leq 0.60, \\ & \quad 0.20 \leq w_3 \leq 0.35, \\ & \quad w_1 + w_2 + w_3 = 1, \\ & \quad w_j \geq 0, j = 1, 2, 3 \\ & \quad \lambda \in [0, 1] \end{aligned}$$

This is a nonlinear optimization problem. By using PSO scheme we get,

$$w_1 = 0.444, w_2 = 0.35 \text{ and } w_3 = 0.206.$$

Step 5: *Computation of the final aggregated values for each alternative.*

By utilizing the weight values of the attributes and the weighted intuitionistic fuzzy arithmetic mean operator [38], the final aggregated values of the alternatives $A_i (i = 1, 2, 3)$ are

$$\begin{aligned} Z_1 &= (0.6194, 0.2659), Z_2 = (0.7042, 0.2076), \\ Z_3 &= (0.7451, 0.1874) \end{aligned}$$

Table 2 Optimization values obtained using PSO and GA

Method	Weight vector	Ranking of alternatives
GA	$w = (0.4443, 0.3495, 0.2062)^T$	$A_3 > A_2 > A_1$
PSO	$w = (0.4440, 0.3500, 0.2060)^T$	$A_3 > A_2 > A_1$

Step 6: *Ranking of the alternatives.*

By utilizing Eq. (9), the score values of alternatives are computed as follows:

$$S(Z_1) = 0.3535, S(Z_2) = 0.4966, S(Z_3) = 0.5577$$

Ranking order is: $A_3 > A_2 > A_1$, *i.e.*, the most desirable alternative is A_3 .

4.1 Comparison analysis

4.1.1 Comparison analysis with GA

To further illustrate the effectiveness of the proposed method, we solve the above investment option selection problem (described in Sect. 4) by using GA [20]. In order to compare performance of the proposed weight generation method, we employ GA to solve above optimization problems and then compute the ranking of alternatives by following the same steps as in the proposed decision making process. Based on the alternatives’ final performances, ranking order of the alternatives in each of the cases is presented in Table 2.

It is observed from Table 2 that the ranking order of attribute weights as well as the ranking order of alternatives, obtained by using GA and PSO are same.

4.1.2 Comparison analysis with existing weight generation methods

In this section, we compare the proposed weight computation method with other existing approaches, such as Li’s process [13], Lin et al. process [14], GRA process [18], knowledge-based process [2] and maximizing deviation process [10] by using the investment option selection problem described in Sect. 4. The above MADM example

is solved by the aforementioned approaches, and the results are shown in Table 3.

As can be seen from Table 3, the ranking order of attribute weights and the ranking order of the alternatives obtained by different methods are same as it did by the proposed method and all of them identify A_3 as the most desirable company to invest money. The foregoing discussion ensures that in the case of weight generation in decision making problem, the proposed method works well. This observation verifies the effectiveness of the proposed method. Moreover, the ranking results obtained by different existing methods and proposed methods are depicted in Fig. 3.

Hence, from the aspects of theoretical analysis and particular application viewpoint it is clear that the proposed weight generation method is able to cope with the situations where attribute weight information is partly known.

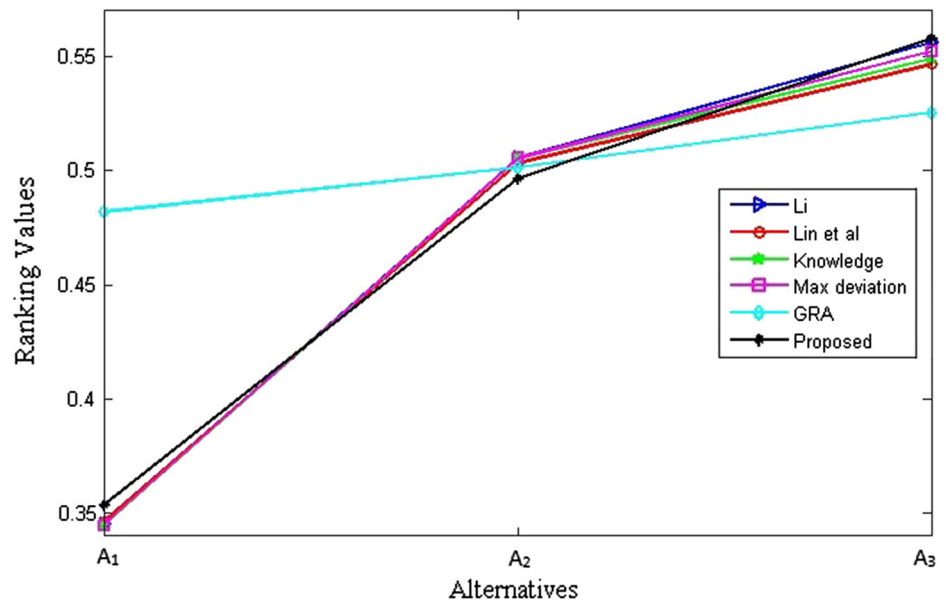
5 Conclusion

This study has presented a weight determination method in a decision making process with intuitionistic fuzzy setting. The weight generation methods that exist in the literature under intuitionistic fuzzy environment have some shortcomings: some weight generation methods are invalid in many situations and thus should be improved; the other weight generation methods may lose some information due to that they need to be transformed the intuitionistic fuzzy decision matrix into an interval-valued decision matrix. Under these circumstances, without changing the original decision information, we have developed a weight determination method by maximizing the average degree of satisfiability and minimizing the average degree of non-satisfiability of each alternative over a set of attributes, simultaneously. We have formulated an MOPP to compute the final comprehensive values for each alternative, and this MOPP is modeled by fuzzy decision making problem and solved by using PSO technique. Finally, we have considered a numerical example to describe effectiveness and applicability of the proposed weight generation method.

Table 3 Optimization values obtained using existing methods

Method	Weight vector	Ranking of alternatives
Max. deviation [10]	$w = (0.4150, 0.3551, 0.2299)^T$	$A_3 > A_2 > A_1$
Lin et al. [14]	$w = (0.4208, 0.3766, 0.2027)^T$	$A_3 > A_2 > A_1$
Li’s method [13]	$w = (0.4159, 0.3425, 0.2416)^T$	$A_3 > A_2 > A_1$
GRA method [18]	$w = (0.4294, 0.3211, 0.2495)^T$	$A_3 > A_2 > A_1$
Knowledge method [2]	$w = (0.4145, 0.3655, 0.220)^T$	$A_3 > A_2 > A_1$
Proposed method	$w = (0.4440, 0.3500, 0.2060)^T$	$A_3 > A_2 > A_1$

Fig. 3 Ranking results by different approaches in the IFS context when weight information is partly known



Although the proposed weight generation approach is illustrated by a selection problem of an investment company, it can also be applied to any other areas of decision problems where uncertainty and hesitation are involved in the evaluation process. This will be the topic of our future research work.

Acknowledgements The first author receives the research grant from the Ministry of Human Resource Development, Government of India. The second author acknowledges the support of Grant ECR/2016/001908.

Compliance with ethical standards

Conflict of interest The authors declare that they have no other conflict of interest.

References

- Atanassov KT (1986) Intuitionistic fuzzy sets. *Fuzzy sets Syst* 20(1):87–96
- Das S, Dutta B, Guha D (2015) Weight computation of criteria in a decision-making problem by knowledge measure with intuitionistic fuzzy set and interval-valued intuitionistic fuzzy set. *Soft Comput*. doi:10.1007/s00500-015-1813-3
- Wei G (2010) Some induced geometric aggregation operators with intuitionistic fuzzy information and their application to group decision making. *Appl Soft Comput* 10(2):423–431
- Wei G, Zhao X (2011) Minimum deviation models for multiple attribute decision making in intuitionistic fuzzy setting. *Int J Comput Intell Syst* 4(2):174–183
- Wei G, Wang H-J, Lin R, Zhao X (2011) Grey relational analysis method for intuitionistic fuzzy multiple attribute decision making with preference information on alternatives. *Int J Comput Intell Syst* 4(2):164–173
- Nguyen H (2015) A new knowledge-based measure for intuitionistic fuzzy sets and its application in multiple attribute group decision making. *Expert Syst Appl* 42(22):8766–8774
- Tzeng G-H, Huang J-J (2011) Multiple attribute decision making: methods and applications. CRC Press, Boca Raton
- Li D-F, Wang Y-C, Liu S, Shan F (2009) Fractional programming methodology for multi-attribute group decision-making using ifs. *Appl Soft Comput* 9(1):219–225
- Yue Z (2011) A method for group decision-making based on determining weights of decision makers using topsis. *Appl Math Modelling* 35(4):1926–1936
- Wei G-W (2008) Maximizing deviation method for multiple attribute decision making in intuitionistic fuzzy setting. *Knowl Based Syst* 21(8):833–836
- Chou S-Y, Chang Y-H, Shen C-Y (2008) A fuzzy simple additive weighting system under group decision-making for facility location selection with objective/subjective attributes. *Eur J Oper Res* 189(1):132–145
- Yeh C-H, Chang Y-H (2009) Modeling subjective evaluation for fuzzy group multicriteria decision making. *Eur J Oper Res* 194(2):464–473
- Li D-F (2005) Multiattribute decision making models and methods using intuitionistic fuzzy sets. *J Comput Syst Sci* 70(1):73–85
- Lin L, Yuan X-H, Xia Z-Q (2007) Multicriteria fuzzy decision-making methods based on intuitionistic fuzzy sets. *J Comput Syst Sci* 73(1):84–88
- Xia M, Xu Z (2012) Entropy/cross entropy-based group decision making under intuitionistic fuzzy environment. *Inf Fus* 13(1):31–47
- Ye J (2010) Fuzzy decision-making method based on the weighted correlation coefficient under intuitionistic fuzzy environment. *Eur J Oper Res* 205(1):202–204
- Wu J-Z, Zhang Q (2011) Multicriteria decision making method based on intuitionistic fuzzy weighted entropy. *Expert Syst Appl* 38(1):916–922
- Wei G-W (2010) GRA method for multiple attribute decision making with incomplete weight information in intuitionistic fuzzy setting. *Knowl Based Syst* 23(3):243–247
- Wei G-W (2011) Gray relational analysis method for intuitionistic fuzzy multiple attribute decision making. *Expert Syst Appl* 38(9):11671–11677
- Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. John Wiley & Sons, Hoboken

21. 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* 6:720–729
22. Goldberg DE, Korb B, Deb K (1989) Messy genetic algorithms: motivation, analysis, and first results. *Complex Syst* 3(5):493–530
23. Ding S, Xu L, Su C, Jin F (2012) An optimizing method of RBF neural network based on genetic algorithm. *Neural Comput Appl* 21(2):333–336
24. Kennedy J (2010) Particle swarm optimization. In: Sammut C, Webb GI (eds) *Encyclopedia of machine learning*. Springer, Berlin, pp 760–766
25. Marini F, Walczak B (2015) Particle swarm optimization (PSO). A tutorial. *Chemom Intell Lab Syst* 149:153–165
26. Hamza MF, Hwa HJ, Choudhury IA (2017) Recent advances on the use of meta-heuristic optimization algorithms to optimize the type-2 fuzzy logic systems in intelligent control. *Neural Comput Appl*. doi:10.1007/s00521-015-2111-9
27. Dorigo M, Caro G, Gambardella L (1999) Ant algorithms for discrete optimization. *Artif Life* 5(2):137–172
28. Bellman RE, Zadeh LA (1970) Decision-making in a fuzzy environment. *Manag Sci* 17(4):B-141
29. Zimmermann H-J (1978) Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets Syst* 1(1):45–55
30. Sakawa M, Yano H (1988) An interactive fuzzy satisficing method for multiobjective linear fractional programming problems. *Fuzzy Sets Syst* 28(2):129–144
31. 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 Modelling* 37(10):6855–6869
32. Baky IA, Abo-Sinna MA (2013) Topsis for bi-level modm problems. *Appl Math Modelling* 37(3):1004–1015
33. Deep K, Singh KP, Kansal M, 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
34. Sakawa M, Yauchi K (2001) An interactive fuzzy satisficing method for multiobjective nonconvex programming problems with fuzzy numbers through coevolutionary genetic algorithms. *IEEE Trans Syst Man Cybern Part B Cybern* 31(3):459–467
35. Chandra S, Jayadeva MA (2009) *Numerical optimization with applications*. Alpha Science International, Oxford
36. Berhe HW (2012) Penalty function methods using matrix laboratory (MATLAB). *Afr J Math Comput Sci Res* 5(13):209–246
37. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353
38. Xu Z (2007) Intuitionistic fuzzy aggregation operators. *IEEE Trans Fuzzy Syst* 15(6):1179–1187
39. Chakraborty D, Guha D, Dutta B (2016) Multi-objective optimization problem under fuzzy rule constraints using particle swarm optimization. *Soft Comput*. doi:10.1007/s00500-015-1639-z
40. Li L, Lai KK (2000) A fuzzy approach to the multiobjective transportation problem. *Comput Oper Res* 27(1):43–57
41. Eiben AE, Smith JE (2003) *Introduction to evolutionary computing*. Springer, Berlin
42. Chen S-M, Tan J-M (1994) Handling multicriteria fuzzy decision-making problems based on vague set theory. *Fuzzy Sets Syst* 67(2):163–172