Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

Maejo International Journal of Science and Technology

ISSN 1905-7873
Available online at www.mijst.mju.ac.th

Full Paper

Fuzzy preference of multiple decision-makers in solving multi-objective optimisation problems using genetic algorithm

Surafel Luleseged Tilahun* and Hong Choon Ong

School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia

* Corresponding author: e-mail:surafelaau@yahoo.com

Received: 13 July 2011 / Accepted: 7 June 2012 / Published: 15 June 2012

Abstract: Most real-life optimisation problems involve multiple objective functions. Finding a solution that satisfies the decision-maker is very difficult owing to conflict between the objectives. Furthermore, the solution depends on the decision-maker's preference. Metaheuristic solution methods have become common tools to solve these problems. The task of obtaining solutions that take account of a decision-maker's preference is at the forefront of current research. It is also possible to have multiple decision-makers with different preferences and with different decision-making powers. It may not be easy to express a preference using crisp numbers. In this study, the preferences of multiple decision-makers were simulated and a solution based on a genetic algorithm was developed to solve multi-objective optimisation problems. The preferences were collected as fuzzy conditional trade-offs and they were updated while running the algorithm interactively with the decision-makers. The proposed method was tested using well-known benchmark problems. The solutions were found to converge around the Pareto front of the problems.

Keywords: multi-objective optimisation, multiple decision-makers, fuzzy preference, genetic algorithm

INTRODUCTION

A decision has to be made when there are a set of possible actions to choose from in order to optimise an objective or objectives. Decision-making is involved in almost all human experience. One can compare the outcome of a decision in order to choose from a set of possible actions. An action that yields the optimal outcome will be the best action to choose. An optimisation focuses on finding the best action for a given objective function or functions. If the number of objective functions is

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

more than one, then the problem is called a multi-objective optimisation problem; otherwise, it is called a single-objective optimisation problem. Multi-objective optimisation problems are encountered in many real-life applications. They usually have conflicting objectives, such as maximising profit while improving the quality of the products. Hence, the solution to such problems involves a compromise between the objectives; this idea is known as the Pareto optimal solution after the Italian economist Vilfredo Pareto (1848–1923) [1]. A point in the feasible region for a multi-objective optimisation problem is said to be the Pareto optimal solution if it is not possible to find another feasible point which performs at least the same in all the objective functions and better in one or more of the objective functions [1]. There are usually many Pareto optimal solutions for a multi-objective optimisation problem. Choosing one from a set of Pareto solutions depends on the preference of the decision-maker. Ordering the objectives and trade-off are among the well-known and frequently used ways of expressing the preference of a decision-maker [2]. The trade-off can be expressed using fuzzy or crisp numbers. This preference helps to identify solutions according to the subjective judgment of the decision-maker [3].

Metaheuristic algorithms are useful in solving optimisation problems. Unlike classical approaches, these algorithms are not greatly affected by the behaviour of the problem. As a result, they have been widely used, especially in difficult optimisation problems. Although these algorithms do not guarantee optimality, they have been tested and proven to yield reasonable solutions [4]. Metaheuristic algorithms, especially genetic algorithms, have been used to solve many real problems formulated as multi-objective optimisation problems [5]. In a review paper, Coello [5] not only reviewed research trends in evolutionary algorithms for multi-objective optimisation problems, but also recommended further study on preference incorporation. Some studies of incorporating a decision-maker's preference involved the ranking of the objective functions [6, 7], which unfortunately lacks uniformity. Other studies involved the fuzzy trade-off of the decision-maker [8, 9] but did not consider situations involving multiple decision-makers or those where the trade-off varied from point to point in the feasible set. The impact of hedges, which uses a fuzzy trade-off as a preference, on fuzzy preferences has also not been dealt with in the previous papers. In many real-life problems, there may be multiple decision-makers with different decision-making powers [10].

In this paper, we introduce a genetic algorithm capable of embedding the fuzzy preferences of multiple decision-makers with different decision-making powers and solving multi-objective optimisation problems interactively. First, the fuzzy trade-off of each decision-maker is determined. From these trade-offs, a fuzzy weight for each decision-maker is constructed according to the randomly generated initial solutions. These solutions are incorporated in the fitness evaluation stage of the genetic algorithm by generating an appropriate probability density function. The probability density functions are constructed in such a way that they agree with the corresponding membership function of the fuzzy preferences and also with the hedges. After a specified number of iterations of the algorithm and depending on the solution at hand, a new preference is obtained, and the iteration is continued until a reasonable solution is achieved.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

PRELIMINARIES

Multi-objective Optimisation

A multi-objective minimisation problem can be written as:

$$\min_{x \in S \subseteq \Re^n} F(x) = (f_1(x), f_2(x), \ldots, f_k(x)) \quad (1)$$

The problem is to find x^* in the feasible set S that yields a minimum value for the k objective functions f_i's, where $i = 1, 2, \ldots, k$. Usually, a conflict exists between the objective functions. Minimising one of the objective functions after some limits will lead to an increase in the value of the other objective functions.

Metaheuristic solution methods have become common in solving these type of problems. For instance, the evolutionary algorithm has been used in different studies [9, 11-13]. Although these solution methods do not guarantee optimality, they generate sound and acceptable solutions. These solution methods yield multiple solutions within a single run, in contrast to the classical solution methods, which rely on the conversion of multi-objective optimisation problems into single- objective optimisation problems, for example, the weighting method [14], Benson's method [14], the utility function method [3] and the lexicographic method [14, 15]. Choosing the best solution among a given set of solutions depends on the preference of the decision-maker. As stated by Coello [5], incorporating the decision-maker's preference is an important issue requiring further exploration.

Fuzzy Preference

A preference is a way to express the subjective judgment of the decision-maker. One commonly used way of determining preferences is using trade-offs. A conditional trade-off of objective j for a unit decrease of objective i at a given point, $(y_1, y_2, \ldots, y_i, \ldots, y_j, \ldots y_k)$, is b, meaning that the decision-maker is indifferent between $(y_1, y_2, \ldots, y_i, \ldots, y_j, \ldots y_k)$ and $(y_1, y_2, \ldots, y_i\text{-}1, \ldots, y_j+b, \ldots y_k)$. This trade-off depends on the value $(y_1, y_2, \ldots, y_i, \ldots, y_j, \ldots y_k)$. If we ask the decision-maker to make a trade-off from another point, it is possible to obtain another trade-off number different from b. Although it is common to use trade-off preference, it is not an easy task for the decision-maker. However, if the decision-maker is allowed the flexibility of assigning a trade-off fuzzily, it will make the task easier. This means that for a unit decrease of objective i, the decision-maker is willing to give around b units of objective j, yielding:

$$\left(y_1, y_2, \ldots, y_i, \ldots, y_j, \ldots y_k\right) \sim \left(y_1, y_2, \ldots, y_i - 1, \ldots, y_j + t, \ldots y_k\right), \text{ for } t \le b + d. \quad (2)$$

The symbol ~ in equation (2) indicates equivalency. As t becomes larger, the degree of acceptability or the equivalency continues to decrease and becomes unacceptable after exceeding some limit, e.g. after $t = b + d$. If d is defined as the width of the fuzzy interval and b as the average trade-off, it is possible to consider this acceptability of preference as a function of t as a membership function in the fuzzy set theory with t as a fuzzy number [16]. So, 'acceptable' has the membership value 1 and 'unacceptable' has the membership value 0. Generally, the values for the membership function, $z(t)$, which is between 0 and 1, should agree with the degree of acceptability, as shown in Figure 1.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

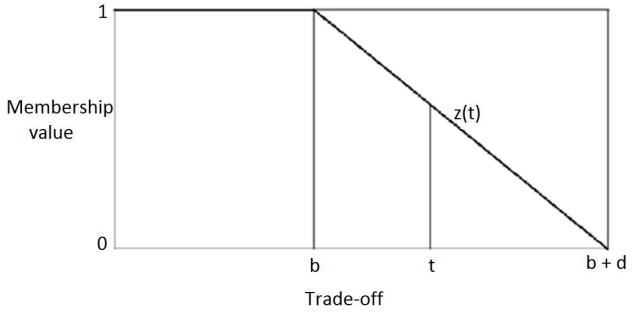


Figure 1. Fuzzy trade-off

Suppose that for the objective functions $f_i(x)$ and $f_j(x)$ with different i and j, the average trade-off given by the decision-maker is (b_{ij}) and the average width is (d_{ij}). Hence, we have two matrices: B = (b_{ij}) and D = (d_{ij}). It is meaningless to compute the trade-off of the i^{th} function for a unit decrease of the i^{th} function itself. Instead, we use $b_{ii} = 0$ and $d_{ii} = 0$. From B, it is possible to calculate the average weight, a weight with a high degree of acceptability, as follows:

$$\overline{b_p} = \frac{\sum_{\substack{i=1 \\ i\neq p}}^{k} b_{pi}}{\sum_{j=1}^{k}\sum_{\substack{i=1 \\ i\neq j}}^{k} b_{ij}} = \frac{\sum_{\substack{i=1}}^{k} b_{pi}}{\sum_{j=1}^{k}\sum_{i=1}^{k} b_{ij}} \tag{3}$$

$$\overline{B} = \begin{pmatrix} \overline{b_1} \\ \overline{b_2} \\ . \\ . \\ . \\ \overline{b_k} \end{pmatrix} \tag{4}$$

where $\overline{B}$ is the average weight for the k objective functions.

It is also possible to compute the average normalised fuzzy width as follows:

$$\overline{d_p} = \frac{\left(\sum_{\substack{i=1 \\ i\neq p}}^{k} d_{pi} \Big/ k-1\right)}{\sum_{i=1}^{k}\left(\sum_{\substack{j=1 \\ j\neq i}}^{k} d_{ij} \Big/ k-1\right)} = \frac{\sum_{i=1}^{k} d_{pi}}{\sum_{i=1}^{k}\sum_{j=1}^{k} d_{ij}} \tag{5}$$

$$\overline{D} = \begin{pmatrix} \overline{d_1} \\ \overline{d_2} \\ . \\ . \\ . \\ \overline{d_k} \end{pmatrix} \tag{6}$$

where $\overline{D}$ is the normalised average fuzzy width.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

For each function $f_i(x)$, the fuzzy weight, b_i, is around the corresponding average weight with some degree of acceptability, as shown Figure 2.

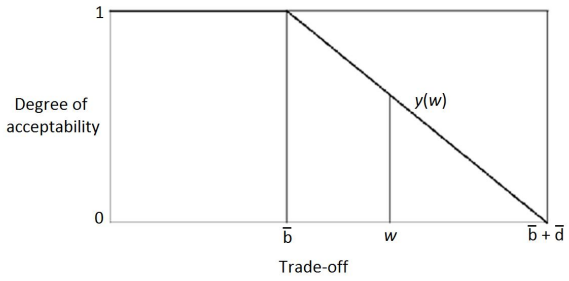
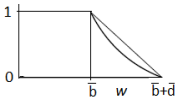
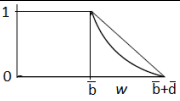
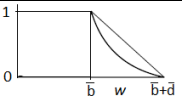
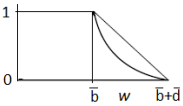
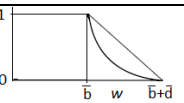
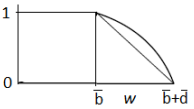
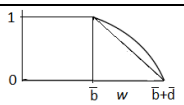
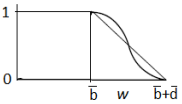


Figure 2. Fuzzy weight (where $\bar{b}$ stands for average weight of any of the objective functions, $\bar{b}_i$'s , and $\bar{d}$ is average width of any of the objective functions, $\bar{d}_i$'s)

Furthermore, the decision-maker may use hedges. Hedges are terms that modify the shape of the membership function of fuzzy sets. These include words such as very, somewhat, more or less, and slightly. The hedges, together with their graphical representations for our minimisation problem, are given in Table 1 and based on Negnevitsky [16].

Table 1. Graphical and functional representations of hedges for minimising the fuzzy trade-off

Hedges	The shape function (mathematical representation)	Graphical representation
A little	$h_{little}(w)=\left(y(w)\right)^{1.3}$	
Slightly	$h_{slightly}(w)=\left(y(w)\right)^{1.7}$	
Very	$h_{very}(w)=\left(y(w)\right)^{2}$	
Extremely	$h_{extremely}(w)=\left(y(w)\right)^{3}$	
Very very	$h_{very\_very}(w)=\left(y(w)\right)^{4}$	
More or less	$h_{more\_or\_less}(w)=\left(y(w)\right)^{1/2}$	
Somewhat	$h_{somewhat}(w)=\left(y(w)\right)^{1/2}$	
Indeed	$h_{indeed}(w)=\begin{cases}2(y(w))^2, \text{if } 0\le y(w)\le 0.5\\ 1-2(1-y(w))^2, \text{if } 0.5< y(w)\le 1\end{cases}$	

Note: $y(w)$ is the straight line joining $(b,1)$ and $(b+d, 0)$

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

If $h_{ij}(w)$ is the mathematical representation of the hedge, when one collects the fuzzy conditional trade-off of f_j for a unit decrease of f_i, then $h_{ij}(w)$ can be any of the functional representations of the hedges in Table (1). If no hedges are used, then $h_{ij}(w)=y(w)$, which will be linear. Hence, there will be a matrix of functions, $H(w) = (h_{ij}(w))$:

$$H:\Re^k \to \Re^k \times \Re^k,\ H(w) = \begin{pmatrix} h_{11}(w) & h_{12}(w) & . & . & h_{1k}(w) \\ h_{21}(w) & . & . & . & h_{2k}(w) \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ h_{k1}(w) & . & . & . & h_{kk}(w) \end{pmatrix} \tag{7}$$

If $h_{ii}(w)$ is taken as the identity mapping for all i, it is possible to combine the given shape functions from the given hedges to obtain the average shape functions as follows:

$$\overline{h_p}(w) = \frac{\sum_{i=1}^{k} h_{pi}(w)}{\sum_{j=1}^{k}\sum_{i=1}^{k} h_{ij}(w)} \tag{8}$$

and where

$$\overline{h}(w) = \begin{pmatrix} \overline{h_1}(w) \\ \overline{h_2}(w) \\ . \\ . \\ . \\ \overline{h_k}(w) \end{pmatrix} \text{ is the average shape function.} \tag{9}$$

Hence, $\overline{h_p}(w)$ determines the shape of the fuzzy membership function for the p^{th} objective function with average weight $\overline{b_p}$ and average fuzzy width $\overline{d_p}$.

Genetic Algorithm

Heuristic algorithms have become useful for dealing with a wide range of problems [17-19]. Based on the evolutionary ideas of natural selection, genetic algorithms are a type of adaptive metaheuristic search algorithms used to find a solution to optimisation problems. The genetic algorithm used in this study is an evolutionary algorithm in which an initial set of solutions are generated. According to the fitness of the solutions, a selection is performed via crossover and mutation in order to construct a new population. The old population is updated by the fittest members, and the same process is continued until the termination criterion is fulfilled. The termination criterion can be the maximum number of generations (fixed number of iterations) or the stage when there is no more improvement in the fitness [20]. A flow chart for a genetic algorithm is given in Figure 3.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

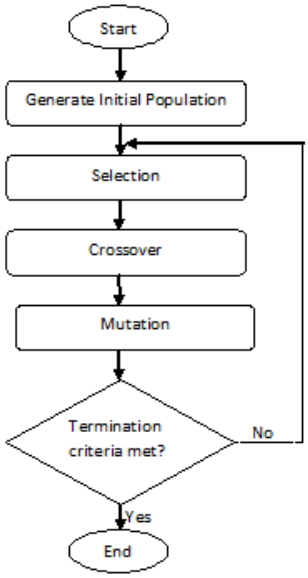


Figure 3. Flow chart of a genetic algorithm

MULTIPLE-PREFERENCE-INCORPORATED GENETIC ALGORITHM

Suppose there are *m* decision-makers, DM_1, DM_2, . . ., DM_m, with a specified authoritarian hierarchy dictating their preferences. DM_1 has the most authority and the level of authority decreases from DM_1 to DM_m. It is possible to assign a weight that describes the authority level of the decision-makers. One possible way to do this is to arrange the level of authority and assign a number which describes the level. We call it a vote. Let a vote be a numerical value which represents the degree of influence of the decision-maker in having his/her preference accepted. A decision-maker with a high level of authority will have a higher vote. When v_i represents the vote of DM_i, it is possible to construct a weight, c, for the decision-makers' preferences from the votes as follows:

$$c_i = \frac{v_i}{\sum_{j=1}^{m} v_j}, \; i \in \{1,2,...,m\} \tag{10}$$

Suppose a decision-maker can give a cumulative fuzzy preference by looking at a set of feasible solutions. After using the fuzzy preference of each decision-maker to construct the average weight matrix, the average width matrix and the average fuzzy shape function for all decision-makers, it is possible to construct a total cumulative weight, W, a total cumulative width, R, and a cumulative shape function, $g(x)$, by combining $\overline{B}$, $\overline{D}$ and $\overline{h}(x)$ using the weights for the decision-makers.

For each decision-maker DM_i, we have:

$$\overline{B^i} = \begin{pmatrix} \overline{b^i_1} \\ \overline{b^i_2} \\ . \\ . \\ . \\ \overline{b^i_k} \end{pmatrix}, \quad \overline{D^i} = \begin{pmatrix} \overline{d^i_1} \\ \overline{d^i_2} \\ . \\ . \\ . \\ \overline{d^i_k} \end{pmatrix} \quad \text{and} \quad \overline{h^i}(w) = \begin{pmatrix} \overline{h^i_1}(w) \\ \overline{h^i_2}(w) \\ . \\ . \\ . \\ \overline{h^i_k}(w) \end{pmatrix}.$$

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

$$W = \frac{1}{m}\sum_{i=1}^{m} c_i \overline{B^i} \Rightarrow \begin{pmatrix} \overline{w_1} \\ \overline{w_2} \\ . \\ . \\ . \\ \overline{w_k} \end{pmatrix} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} c_i \overline{b^i}_1 \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{b^i}_2 \\ . \\ . \\ . \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{b^i}_k \end{pmatrix} \tag{11}$$

$$R = \frac{1}{m}\sum_{i=1}^{m} c_i \overline{D^i} \Rightarrow \begin{pmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_k \end{pmatrix} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} c_i \overline{d^i}_1 \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{d^i}_2 \\ . \\ . \\ . \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{d^i}_k \end{pmatrix} \tag{12}$$

$$g(w) = \frac{1}{m}\sum_{i=1}^{m} c_i \overline{h^i}(w) \Rightarrow \begin{pmatrix} g_1(w) \\ g_2(w) \\ . \\ . \\ . \\ g_k(w) \end{pmatrix} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} c_i \overline{h^i}_1(w) \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{h^i}_2(w) \\ . \\ . \\ . \\ \frac{1}{m}\sum_{i=1}^{m} c_i \overline{h^i}_k(w) \end{pmatrix} \tag{13}$$

Once we have the total cumulative weight, width and shape function, it is possible to incorporate these parameters in the fitness evaluation step of the genetic algorithm by generating a dynamic weight from the given preference and taking the weighted sum of the objective functions. To generate a random weight that satisfies the given cumulative preference, it is necessary to specify an appropriate probability density function for each dynamic weight, $g_i(w)$ for $i \in \{1, 2, \ldots, k\}$, in such a way that a number with high acceptability needs to have a high probability, as shown in Figure 2. For instance, suppose no hedges are used, let the straight line joining $\left(\overline{w_i}, 1\right)$ and $\left(\overline{w_i} + r_i, 0\right)$ be $g_i(w) = p_i w + q_i$ for $i \in \{1,2, \ldots, k\}$, for some p_i and q_i. It is necessary to generate a random weight under the shape function with a high probability near $\overline{w_i}$. For such purpose, it is possible to use a sampling method. To make the curve a probability density function, the area under the curve needs to be 1. To do so, it may be necessary to adjust the end points of the curve. Suppose it passes

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

through $\left(\overline{w}_i,\ \mathrm{y}\right)$ and $\left(\overline{w}_i + r_i,\ 0\right)$, where y is an arbitrary number that is determined by setting the area under the curve to 1:

$$\text{Area} = \int_{\overline{w}_i}^{\overline{w}_i + r_i} g_i(w)dw = 1.$$

Furthermore, $g_i(\overline{w}_i + r_i) = 0$. Hence, $g_i(w)$ will be:

$$g_i(w) = \frac{-2w_i}{r_i^2} + \frac{2(\overline{w}_i + r_i)}{r_i^2}, \text{ for all } i. \tag{14}$$

In other words, w_i is a random variable with the probability density function $g_i(w)$. In a genetic algorithm, it is possible to incorporate this fuzzy preference and zoom to the area in the outcome space according to the cumulative fuzzy preference. To do that, we construct the fitness function in each iteration, j, by taking the weighted sum of the objective functions:

$$\text{Fitness function} = \sum_{i=1}^{k} w_i(j) f_i(x) \tag{15}$$

where $w_i(j)$ is the weight of f_i at iteration j.

The conditional fuzzy trade-off depends on the current value of the objective functions. Thus, it is necessary to return to the decision-makers after a number of iterations of the algorithm to determine whether a reasonable solution has been achieved. If the decision-makers are not satisfied with the solution, the preference will be updated by generating new preferences and rerunning the algorithm with new cumulative fuzzy preferences. Hence, the fuzzy weight may vary in the process until the decision-makers are satisfied with the solution or no further improvements can be made. This means that the algorithm will run interactively with the decision-makers until a termination criterion is fulfilled. The preference-incorporated interactive algorithm can be generalised as follows:

Initial step: set the parameter and the initial inputs as:

$f_j(x),\ x \in \Re^n,\ j \in \{1,2,...,k\}$ are the objective functions,

S is the feasibility condition,

P_r and P_m are the probability of crossover and mutation

and c_i is the vote of the decision-makers.

Main step:

1. $x_i, i \in \{1,2,...,m\}$, - generate the initial population and obtain a cumulative fuzzy preference from the DMs, depending on x_i's.
2. $\mathrm{g_i}\left(w\right)$, - construct a probability density function for the weight of each objective function using c_i and the cumulative fuzzy preference.
3. $children = \varnothing$, - set the children set empty

   for *t=1* to the number of the generation

   $parent = \{x_i, i \in \{1,2,...,m\}\}$.

   (3.1) For the fitness function construction

   $w_j\ \ j \in \{1,2,...,k\}$, generate a weight for the objective functions using the probability density function $\mathrm{g_i}\left(w\right)$.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

$fun_i = \sum_{j=1}^{m} w_j f_j(x_i)$ $i \in \{1,2,...,m\}$, - dynamic fitness function

(3.2) Choose x_1' and x_2' for crossover and mutation with the probability related to the fitness.

(3.3) Apply crossover and mutation with the probability P_r and P_m, on x_1' and x_2' to obtain y_1' and y_2'.

$children = children \cup \{y_1', y_2'\}$, and update the set children.

If the number of elements in children < *m*, then return to step (3.2).

(3.4) Choose and set the fittest members as parents from $children \cup parents$

end.

4. If the decision-makers are satisfied with the solution, stop; otherwise, update the preference and go back to step (2).

EXPERIMENTAL RESULTS

Simulations using Matlab were carried out on five bi-objective optimisation problems, as in equation (16), with different Pareto fronts and different objective functions, f_i's. The preferences were changed twice. In all the tests, the probability of crossover and mutation were taken to be 0.9 and 0.2 respectively. Forty initial solutions were generated for all the simulations.

$$\min_{x \in S \subseteq \Re^2} F(x) = (f_1(x), f_2(x)) \tag{16}$$

The results of the simulations are presented below:

1. The first optimisation problem, given in equation (17), exhibited a convex Pareto front. The total average weight and width were changed once and the shape functions were linear. The total average weight and width were 0.6 and 0.15 respectively for the first objective function, and 0.5 and 0.2 respectively for the second. After 15 iterations, the parameters were changed to 0.3 and 0.15 for the first test function and 0.8 and 0.2 for the second respectively. The iteration number was taken to be 15 before and 15 after the change of preference. The results are shown in Figure 4.

$$f_1(x) = \frac{1}{2}\sum_{i=1}^{2} x_i^2, f_2(x) = \frac{1}{2}\sum_{i=1}^{2}(x_i - 2.0)^2 \text{ and } S = \{(x_1, x_2) \in \Re^2 \mid -1 \le x_1, x_2 \le 1\} \tag{17}$$

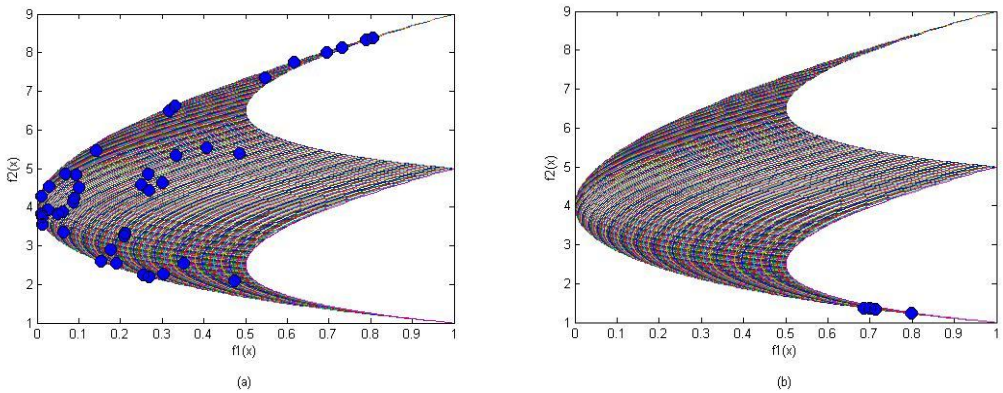


Figure 4. The outcome space for the first test problem: (a) with the population generated initially; (b) at the end of the algorithm

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

2. The second test problem exhibited a discontinuous Pareto front as shown in equation (18). The axis parallel to the hyper-ellipsoid function of the first objective function was negative, and the second objective function was a Rastrigin's function, with $n = 2$. The preferences were the same as in the first simulation, with the same number of iterations and shape functions. The number of iterations was 15 before and 15 after the change in the preferences. The results are shown in Figure 5.

$$f_1(x) = -\left(2x_1^2 + x_2^2\right),$$
$$f_2(x) = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) \quad (18)$$
$$\text{and } -5 \le x_1, x_2 \le 5.$$

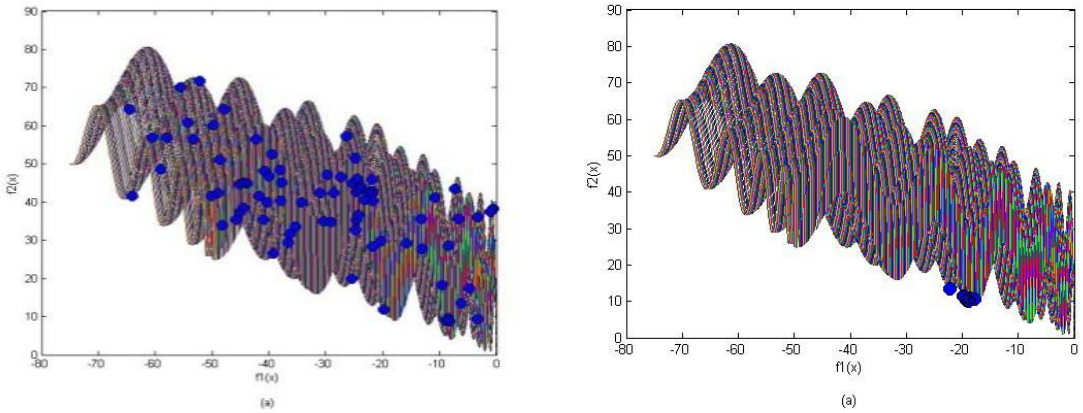


Figure 5. The outcome space for the second test problem: (a) with the population generated initially; (b) at the end of the algorithm

3. Here, the test problem had a point Pareto front. The first objective function was the first function of De Jong's, with $n = 2$, as shown in equation (19). The preferences were changed once with the same value used as in the previous cases, as well as the same shape functions and the same number of iterations. The results are shown in Figure 6.

$$f_1(x) = x_1^2 + x_2^2, f_2(x) = |x_1|^2 + |3x_2| \text{ and } 0 \le x_1, x_2 \le 1 \quad (19)$$

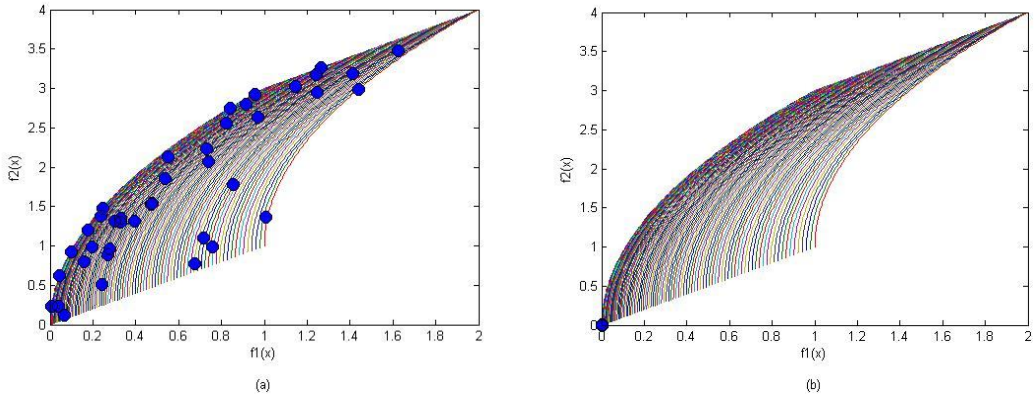


Figure 6. The outcome space for the third test problem: (a) with the population generated initially; (b) at the end of the algorithm

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

4. The fourth test problem exhibited a concave and convex Pareto front. The first objective function was a linear function (an identity function) as given in equation (20). The total average weight and width were 0.4 and 0.1 for the first objective functions, and 0.5 and 0.1 for the second respectively. These were changed after 15 iterations to 0.3 and 0.1 for the first objective functions, and 0.7 and 0.15 for the second respectively, with the shape functions left as a straight line. After changing the preferences, 15 iterations were performed. The results are shown in Figure 7.

$$f_1(x) = x_1,\ f_2(x) = (1+9x_2)\left(1-(\frac{x_1}{1+9x_2})^{1/4}-(\frac{x_1}{1+9x_2})^4\right) \text{and } 0 \le x_1, x_2 \le 1. \quad (20)$$

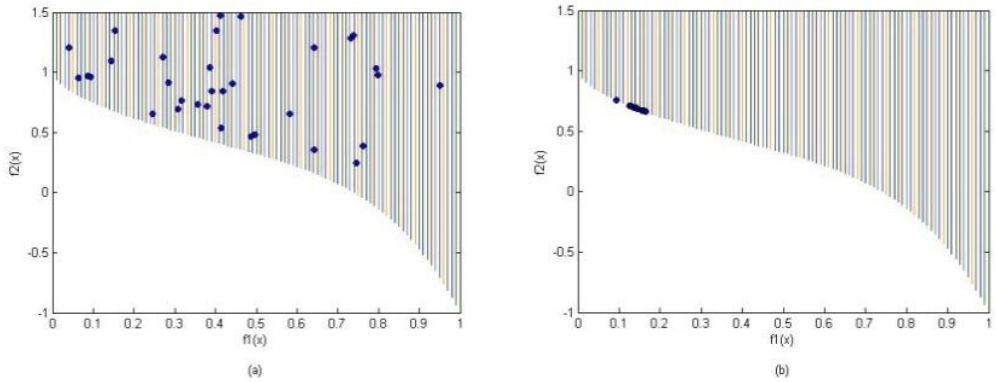


Figure 7. The outcome space for the fourth test problem: (a) with the population generated initially; (b) at the end of the algorithm

5. The last test problem showed a concave Pareto front as presented in equation (21). Fifteen iterations were performed before and after changing the preferences. The preferences and the shape functions were the same as in the previous case (test problem 4). The results are shown in Figure 8.

$$f_1(x) = x_1,\ f_2(x) = (1+9x_2)\left(1-(\frac{x_1}{1+9x_2})^2\right) \text{and } 0 \le x_1, x_2 \le 1. \quad (21)$$

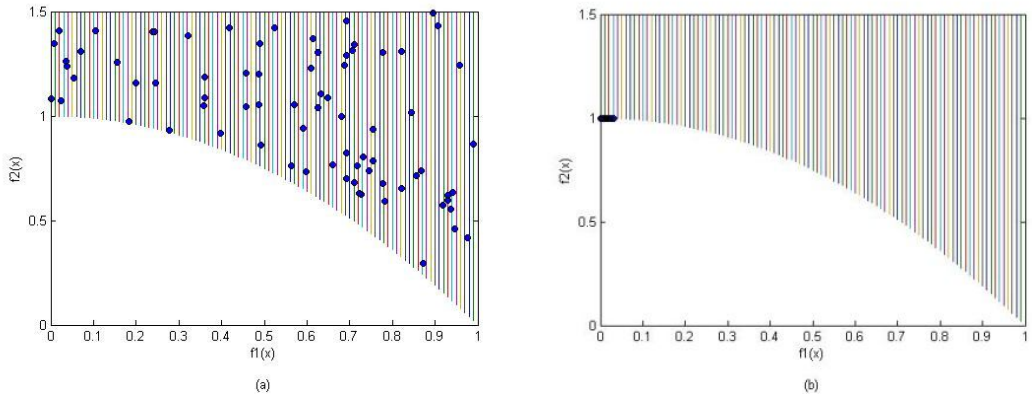


Figure 8. The outcome space for the fifth test problem: (a) with the population generated initially; (b) at the end of the algorithm

From the simulation results, one can see that the fuzzy preference incorporating an interactive genetic algorithm in this study is in agreement with the given preferences. When high preference is

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

placed on the first objective function, f_1, the points move to the left; when preference for the objective function 2, f_2, is higher, the points move to the right.

CONCLUSIONS

Solving multi-objective optimisation problems using a genetic algorithm with varying fuzzy preferences of multiple decision-makers was discussed and demonstrated. The fuzzy trade-off preference of each decision-maker was determined from initially generated solutions. These preferences were combined so that fuzzy weights could be constructed according to the fuzzy trade-offs of the decision-makers. Using the vote, a numerical value, of each decision-maker in having his/her preference accepted, a cumulative fuzzy weight vector of dimension k was constructed. The cumulative fuzzy weight was then expressed using the probability density function, which was in agreement with the membership function and with the hedges given, if any. These cumulative weights were then incorporated with the genetic algorithm, specifically in the fitness evaluation stage. This interactive method was tested on five selected bi-objective optimisation problems with different Pareto fronts and properties. Based on the simulation, it is clear that the algorithm yields a solution along the Pareto front that is in agreement with the given preference.

ACKNOWLEDGEMENTS

This work was supported in part by Universiti Sains Malaysia short-term grant number 304/PMATHS/6311126. The first author acknowledges support from a TWAS-USM 2008 fellowship.

REFERENCES

1. M. Ehrgott and X. Gandibleux, "Multiple Criteria Optimisation: State of the Art Annotated Bibliographic Surveys", Kluwer Acadamic Publishers, Boston, **2002**.
2. L. Rachmawati and D. Srinivasan, "Preference incorporation in multi-objective evolutionary algorithms: A survey", Proceedings of IEEE Congress on Evolutionary Computation, **2006**, Vancouver, Canada, pp.962-968.
3. R. L. Keeney and H. Raiffa, "Decision with Multiple Objectives: Preferences and Value Tradeoffs", John Wiley and Sons, New York, **1976**.
4. X.-S. Yang, "Nature-Inspired Metaheuristic Algorithms", Luniver Press, Frome, **2010**.
5. C. A. Coello, "Evolutionary multi-objective optimization: Some current research trends and topics that remain to be explored", *Front. Comp. Sci. China*, **2009**, *3*, 18-30.
6. Y. Jin, M. Olhofer and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?", Proceedings of Genetic and Evolutionary Computation, **2001**, San Francisco, USA, pp.1042-1049.
7. Y. Jin and B. Sendhoff, "Incorporating of fuzzy preferences into evolutionary multi-objective optimization", Proceedings of 4th Asia-Pacific Conference on Simulated Evolution and Learning, **2002**, Singapore, pp.26-30.

Maejo Int. J. Sci. Technol. **2012**, *6*(02), 224-237

8. S. L. Tilahun and H. C. Ong, "Fuzzy preference incorporated evolutionary algorithm for multi-objective optimization", Proceedings of International Conference on Advanced Science, Engineering and Information Technology, **2011**, Bangi-Putrajaya, Malaysia, pp.26-30.
9. H. C. Ong and S. L. Tilahun, "Integrating fuzzy preference in genetic algorithm to solve multi-objective optimization problems" *Far East J. Math. Sci.*, **2011**, *55*, 165-179.
10. P. Jaramillo, R. A. Smith and J. Andreu, "Multi-decision-makers equalizer: A multi-objective decision support system for multiple decision-makers", *Ann. Operat. Res.*, **2005**, *138*, 97-111.
11. A. Lahsasna, R. N. Ainon and T. Y. Wah, "Enhancement of transparency and accuracy of credit scoring models through genetic fuzzy classifier", *Maejo Int. J. Sci. Technol.*, **2010**, *4*, 136-158.
12. C. L. Karr and L. M. Freeman, "Industrial Applications of Genetic Algorithms", CRC Press, Boca Raton, **1999**.
13. A. Osyczka and S. Krenich, "Evolutionary algorithms for multi-criteria optimization with selecting a representative subset of Pareto optimal solutions. Evolutionary multi-criterion optimisation", Proceedings of 1st International Conference on Evolutionary Multi-Criterion Optimisation, **2001**, Zurich, Switzerland, pp.141-153.
14. M. Ehrgott, "Multicriteria Optimization", Springer, Berlin, **2005**.
15. V. A. Emelichev, M. K. Kravtsov and O. A. Yanushkevich, "Lexicographic optima in the multicriteria discrete optimization problem", *Math. Notes*, **1995**, *58*, 928-932.
16. M. Negnevitsky, "Artificial Intelligence: A Guide to Intelligent System", Henry Ling, Harlow, **2005**.
17. M. Uysal, "Using heuristic search algorithms for predicting the effort of software projects", *Appl. Comput. Math.*, **2009**, *8*, 251-262.
18. K. Y. Lee and M. A. El-Sharkawi, "Modern Heuristic Optimization Techniques: Theory and Application to Power System", John Wiley and Sons, Hoboken, **2008**.
19. I. H. Osman and J. P. Kelly, "Meta-Heuristics: Theory and Applications", Kluwer Academic Publishers, Norwell, **1996**.
20. R. L. Haupt and S. E. Haupt, "Practical Genetic Algorithms", John Wiley and Sons, Hoboken, **2004**.