A Multiobjective Evolutionary Algorithm for Deriving Final Ranking from a Fuzzy Outranking Relation

Juan Carlos Leyva-Lopez[1,2], Miguel Angel Aguilera-Contreras[2]

[1] Universidad de Occidente,
Carr. a Culiacancito Km. 1.5 Culiacan, Sinaloa, Mexico, 80020
jleyva@culiacan.udo.mx
[2] Universidad Autonoma de Sinaloa,
Prol. Josefa O. de Dominguez s/n, Ciudad Universitaria
Culiacan, Sinaloa, Mexico, 80040
{jleyva, aguilera}@uas.uasnet.mx

Abstract. The multiple criteria aggregation methods allow us to construct a recommendation from a set of alternatives based on the preferences of a decision maker. In some approaches, the recommendation is immediately deduced from the preferences aggregation process. When the aggregation model of preferences is based on the outranking approach, a special treatment is required, but some non-rational violations of the explicit global model of preferences could happen. In this case, the exploitation phase could then be treated as a multiobjective optimization problem. In this paper a new multiobjective evolutionary algorithm, which allows exploiting a known fuzzy outranking relation, is introduced with the purpose of constructing a recommendation for ranking problems. The performance of our algorithm is evaluated on a set of test problems. Computational results show that the multiobjective genetic algorithm-based heuristic is capable of producing high-quality recommendations.

1 Introduction

Multiple Criteria Decision Analysis provides two major approaches of constructing a global preference model from an actor involved in the decision process. The first one is the functional model, which has been widely used within the framework of multi-attribute utility theory (e.g. [10, 17, 30]). The second one is the relational model, which has its most known representation in the form of a fuzzy or crisp outranking relation (e.g. [26]). This paper is concerned with the outranking approach to Multiple Criteria Decision Aid. Methods related to this approach, including the well-known family of ELECTRE methods, are often presented as the combination of two phases: aggregation (or construction) and exploitation. The aggregation process corresponds to the operation, which transforms the marginal evaluations of separate criteria into a global outranking relation between every pair of alternatives, which is generally neither transitive nor complete. Outranking relations, in most methods, are built using a concordance-discordance principle.

It is well known that this principle does not, in general, lead to binary relations possessing “remarkable properties” such as transitivity and completeness [2]. The exploitation process deals with the outranking relation in order to clarify the decision through a partial or total preordering reflecting some of the irreducible indifferences and incomparabilities [8]. ELECTRE-III, PROMETHEE and other methods for decision aid (e.g. [25, 1, 8] build and exploit a fuzzy outranking relation.

Let A be the set of decision alternatives or potential actions and let us consider a fuzzy outranking relation S_A^σ defined on AXA ; this means that we associate with each ordered pair $(a,b) \in AXA$ a real number $\sigma(a,b)$ $(0 \le \sigma(a,b) \le 1)$ reflecting the degree of strength of the arguments favoring the crisp outranking aSb . The exploitation phase transforms the global information included in S_A^σ into a global ranking of the elements of A. Usually; three different ways are used [8]:

1: transform S_A^σ into another valued relation R that presents some interesting property needed for ranking purposes, i.e. transitivity,

2: determine a crisp binary relation, close to S_A^σ which presents crisp properties needed for ordering,

3: use a ranking method to obtain a score function.

Way 1 includes the process of finding the transitive closure or the intersection of traces. Way 3 is most commonly used in classical procedures like ELECTRE-III and PROMETHEE. But the main difficulty consists in finding reasonable ways of dealing with the intransitivities without losing too much of the contents of the outranking relation. In this sense, the methods included in ways 1 and 2 lose information coming from S_A^σ when exploiting a not so close transitive valued relation R, or a crisp binary relation with desirable properties for ranking purposes. On the other hand, the methods based in score functions do not perform well in presence of irrelevant alternatives or in case of complex graphs with many circuits. Nonrational situations could happen when the prescription is constructed. Most significant is the following: Suppose that a_i and a_j are two actions such that $\sigma(a_i, a_j) \ge \lambda$ and $\sigma(a_j, a_i) \le \lambda - \beta, \quad (\beta > 0)$; if $\lambda \ge c$ and $\beta \ge t$ (c and t representing consensus and threshold levels respectively), we should accept that " a_i outranks a_j " $(a_i S^\lambda a_j)$ and" a_j does not outrank a_i " $(a_j nS^\lambda a_i)$; in this case the global preference model captured in outranking relation is giving a presumed preference favoring a_i . However, a score function or any other similar method could lead to a final ordering in which a_j is ranked better. ELECTRE and PROMETHEE methods do not have a way to minimize this kind of irregularity. In any case, the exploitation phase could then be treated as a multiobjective optimization problem [19]. In this way, a number of solutions can be found which provide the decision maker with insight into the characteristics of the problem before a final solution is chosen.

Evolutionary Multiobjective Optimization (EMOO) seeks to optimize the components of a vector-valued cost function. Unlike single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set. Each point in this surface is optimal in the sense that no improvement can be achieved in one cost vector component that does not lead to degradation in at least one of the remaining components. Assuming, without loss of generality, a minimization problem, each element in the Pareto-optimal set constitutes a non-inferior solution to the EMOO problem. Non-inferior solutions have been obtained by solving appropriately formulated ranking problems. Methods used include the contained in the ways (**1**), (**2**) and (**3**) and recently a method based on a genetic algorithm [19, 7].

By maintaining a population of solutions, multiobjective evolutionary algorithms (MOEAs) can search for many non-inferior solutions in parallel. This characteristic makes MOEAs very attractive for solving EMOO problems.

In this paper, we propose a multiobjective evolutionary algorithm for improving the quality of recommendation when a fuzzy outranking relation is exploited, which is of particular interest for solving the Multiple Criteria Ranking Problem. This approach rests on the main idea of reducing differences between the global model of preferences and final ranking. In the next section the exploitation of a fuzzy outranking relation formulated as a multiobjective optimization problem is described, and on this background we present our proposal in section 3. A test problem and computational result is given in section 4, and finally, in section 5 some conclusions are discussed.

2 The Exploitation of a Fuzzy Outranking Relation as a Multiobjective Combinatorial Optimization Problem

Let *A* be a finite set of decision alternatives, which is the object of the decision process. This set is not the universe of the potentially feasible alternatives; it is only the set under consideration in a specific decision problem. Let $\sigma(a,b)$ be a valued binary relation defined on AXA with image in [0,1]. Note that the fuzzy outranking relation S_A^{σ} of the past section is a particular case of $\sigma(a,b)$. $\sigma(a,b)$ can be interpreted as the credibility degree of the predicate "*a* is at least as good as *b*". Let λ be a cut level such that if $\sigma(a,b) \geq \lambda$, we say that *a* outranks *b* with credibility λ, denoted by $aS^{\lambda}b$. Otherwise, the outranking is rejected $anS^{\lambda}b$.

We assume the existence of a threshold $\beta > 0$ such that if $aS^{\lambda}b$ and also $\sigma(b,a) \leq (\lambda - \beta)$, then there is an asymmetric preference relation favoring *a* that will be denoted by $aP^{\lambda,\beta}b$. One can agree that for some values of λ and β, the conditions defining $P^{\lambda,\beta}$ are good arguments for justifying a strict preference relation in the sense proposed by Roy (cf. [24]).

Let *E* be a way of exploiting σ and R_A the complete ranking derived from applying *E* to σ. *E* is a function assigning a ranking R_A to each σ defined on *AXA*. R_A defines a weak order *R* on *A*. $\forall(a,b) \in AXA$, aRb if and only if *b* is not ranked before *a* in R_A. We think that the quality of a final ranking should be judged according to the number of its discrepancies and concordances with σ and the crisp relations S^{λ}, and $P^{\lambda,\beta}$. Let *V* be the set of strong discrepancies (violations) defined as $V = \{(a,b) \in AXA\text{:}\ aP^{\lambda,\beta}b,\ \ bRa\ \}$ and $n_V = cardinality\ of\,(V)$. Note that n_V is a function of *R*, λ, and β.

We propose to consider the best ordering as the best compromise solution of the following multiple objective optimization problem:

$$\begin{aligned}
&Min(n_V), \quad Min(f), \quad Max(\lambda) \\
&Subject\ to \\
&R \subset AXA, \quad \lambda, \beta \in [0,1], \quad \lambda \geq \lambda_0 \\
&(\lambda_0 \textit{ is a minimum level of credibility, usually greater than } 0.5)
\end{aligned} \tag{1}$$

Where f is a measure counting the number of incomparable pairs i.e. counting all the pairs $(a,b) \in AXA$ such that $anS^{\lambda}b$ and $bnS^{\lambda}a$.

The minimization of n_V can be seen as a process of reducing the magnitude of the arguments against R. The minimization of f can be seen as a process of increasing the number of comparable pair of alternatives in S^{λ}. Increasing λ improves the credibility of $P^{\lambda,\beta}$ and S^{λ}, relations on which the ranking is based. The most important objective is n_V. A value of λ between 0.65 and 0.75 is often considered good and no further increments are necessary. The structure of (1) strongly suggests the use of evolutionary algorithms.

3 Multiobjective Evolutionary Algorithm for Exploiting a Fuzzy Outranking Relation

This section presents an evolutionary algorithm that solves (1) overcoming the limitations of the genetic algorithm presented by [19] by taking advantage of the structure of the objective space.

3.1 Multiobjective Evolutionary Optimization

Evolutionary algorithms are stochastic search techniques that mimic the natural selection process. The goal of evolutionary algorithms is to obtain better individuals (i.e., solutions) as the algorithm progresses. In any given generation (i.e., iteration), the population of individuals is combined and altered to obtain a children population. The parents and children undergo an evaluation and selection process, where the better individuals have a higher chance of survival. Algorithms sharing the same spirit and based on the same natural selection principle have been proposed in the fields of evolutionary strategies [28], evolutionary programming [9], and genetic algorithms [14].

Evolutionary algorithms have been applied to solve complex problems where traditional optimization methods have failed to provide a good solution. Solving optimization problems with multiple objectives is, generally, a very difficult task. Evolutionary algorithms are particularly well suited for multiobjective optimization due to their ability to explore a vast set of alternatives, partially because they are population-based and can evaluate several solutions in parallel [33]. Evolutionary algorithms can also be designed to search for the efficient frontier in a single run, without making assumptions about the shape and mathematical properties of the frontier [3]. Moreover, there are few competitive alternatives to multiobjective optimization with noise and uncertain objective functions [15].

Multiobjective evolutionary algorithms have become an active line of research since the first algorithm was proposed in the mid eighties [27]. Taxonomy of multiobjective evolutionary algorithms is possible based on the decision maker's preferences being made before, during, or after the optimization process. [32], present such a classification of multiobjective evolutionary algorithms based on prior, progressive, and posterior articulation of preferences. In the latter years, the algorithms based on posterior articulation of preferences have received the most attention. These evolutionary algorithms move the population of solutions toward and efficient frontier. Among these algorithms, the most prominent are the Multiobjective Genetic Algorithm (MOGA) [11], the Niched Pareto Genetic Algorithm (NPGA) [16], the Nondominated Sorting Genetic Algorithm (NSGA) [29, 6], the Strength Pareto Evolutionary Algorithm (SPEA2) [33], and the Pareto Archived Evolution Strategy (PAES) [18]. For a thorough exposition of multiobjective evolutionary algorithms the reader is referred to [3, 4, 32, 5].

3.2 The Multiobjective Evolutionary Algorithm

In this section we present a multiobjective evolutionary algorithm based on posterior articulation of preferences, able to exploit a known fuzzy outranking relation with the purpose of constructing a recommendation for the multiple criteria ranking problem. The algorithm borrows fundamental elements from MOGA [11], which has become one of the leading multiobjective evolutionary algorithms.

In the following subsections we present further detail on the fundamental aspects of the Multiobjective evolutionary algorithm.

Encoding the Solutions. A potential solution of a ranking problem is represented as an ordinal representation. In general, a potential solution is a ranking of the set of decision alternatives or actions by decreasing order of preference. These actions (known as genes) are joined together forming a string of values (known as chromosome). Any symbol in this string is refereed to as an allele [13, 20]. The chromosome is represented as the string of *m*-ary alphabet where *m* is the number of actions into the decision problem. In such representation, each action is coded into *m*-ary form. Actions are then linked together to produce one long *m*-ary string or chromosome. An action coded with value a_{k_i} in the *i*-th entry of the string means that the action coded with value a_{k_i} is ranked in the *i*-th place of the ordering and a_{k_i} is preferred to a_{k_j} if *i*<*j*, where $a_{k_i} \in A = \{a_1, a_2, ..., a_m\}$, $i = 1,2,...,m$, and $[k_1, k_2, ..., k_m]$ is a permutation of $[1,2,...,m]$.

Objective functions *f*, *u* and λ. Each potential solution or individual in the population is associated with a number λ ($0 \leq \lambda \leq 1$), which will be connected with the credibility level of a crisp outranking relation defined on the set of genes. The fitness of an individual with credibility level λ is calculated according to a given *fitness procedure*. The approach for defining individual's fitness involves the non dominated solutions in a similar form of MOGA [11]. In accordance with (1), we define the objective function f of an individual $\bar{p}$ with credibility level λ as follows: Let $\bar{p} = a_{k_1} a_{k_2} \ldots a_{k_m}$ be the schematic representation of an individual's chromosome and suppose that given a_{k_i} and a_{k_j}, two actions such that $\sigma(a_{k_i}, a_{k_j}) \geq \lambda$ and $\sigma(a_{k_j}, a_{k_i}) \leq \lambda - \beta$ ($\beta > 0$, representing a threshold level), we accept that "a_{k_i} outranks a_{k_j}" ($a_{k_i} S^{\lambda} a_{k_j}$) and "$a_{k_j}$ does not outrank a_{k_i}" ($a_{k_j} nS^{\lambda} a_{k_i}$). In this case, into the crisp outranking relation generated by λ, S_A^{λ}, a presumed preference favoring a_{k_i}, holds. Then:

$$f(\bar{p}) = \left| \left\{ (a_{k_i}, a_{k_j}) : a_{k_i} nSa_{k_j} \;\; and \;\; a_{k_j} nSa_{k_i}; \;\; i = 1,2,\ldots,m-1, \; j = 2,3,\ldots,m, \;\; i < j \right\} \right| \tag{2}$$

where $[k_1, k_2, \ldots, k_m]$ is a permutation of $[1,2,\ldots,m]$

$f(\bar{p})$ is the number of incomparabilities between pairs of actions (a_{k_i}, a_{k_j}) into the individual $\bar{p} = a_{k_1} a_{k_2} \ldots a_{k_m}$ in the sense of the crisp relation S_A^{λ}. Note that the quality of solution increases with decreasing *f* score.

The objective function u of an individual $\bar{p}$ measures the amount of unfeasibility (in relative terms) and we chose to define it as:

$$u(\bar{p}) = \left| \left\{ (a_{k_i}, a_{k_j}) : a_{k_i} Sa_{k_j} \;\; and \;\; a_{k_j} nSa_{k_i}; \quad i = 1,2,\ldots,m, \; j = 1,2,\ldots,m, \;\; i > j \right\} \right| \tag{3}$$

$u(\bar{p})$ is the number of preferences between actions into the individual $\bar{p}$ which are not "well-ordered" in the sense of S_A^{λ}.

An individual $\bar{p}$ is feasible if $u(\bar{p}) = 0$ and infeasible if $u(\bar{p}) > 0$. Defining the objective function *u* taking the zero minimum value if and only if the solution is feasible seems a natural approach. Each individual $\bar{p}$ can then be represented by a triad of values *f*, *u*, and λ.

We are interested in:

i) Individuals whose objective function *u* value is equal to zero. This assures us that the ordering represented by the individual is transitive; this is one of two characteristics that should be exhibited by all recommendation (solution) of ranking problems [31].

ii) Individuals whose objective function *f* value is equal (or near) to zero. This objective improves the comparability of *S* on *A*.

iii) Individuals whose credibility level λ is near to 1. This indicates us that the ordering represented by the individual with credibility level λ is trustier whenever the objective functions *u* and *f* values are zero or near to zero. In practice, the requirement connected to function *f* does not permit that λ values approach to 1 because in this case we could have many incomparable genes.

Then, we use an evolutionary search for solving the multiobjective problem:

$$\begin{aligned}&Min(u),\quad Min(f),\quad Max(\lambda)\\&Subject\ to\\&R_S\ ,\quad \lambda\in[0,1],\quad \lambda\geq\lambda_0\\&(where\ R_S\ is\ a\ strict\ total\ order\ of\ A\ and\ \lambda_0\ is\ a\\&\ minimum\ level\ of\ credibility)\end{aligned} \tag{4}$$

We can see that the objective function *u* coincides with n_V , (see (1) of section 2).

Fitness Assignment Procedure. Most of the approaches of the multiobjective decision making seek elements of *all* the *Pareto optimal set* P^* which in the jargon of Multiobjective Evolutionary Algorithms (MOEA) it is often denoted as P_{true} [4]. During MOEA execution, a "current" set of Pareto optimal solutions is determined at each EA generation and termed $P_{current}(t)$, where *t* represents the generation number. Many MOEA implementations also use a secondary population storing nondominated solutions found through the generations. This secondary population is named $P_{known}(t)$. This term is also annotated with *t* to reflect its possible changes in membership during MOEA execution. $P_{known}(0)$ is defined as the empty set (ϕ) and P_{known} only as the final set of solutions returned by the MOEA at termination. $P_{current}(t)$, P_{known} , and P_{true} are sets of MOEA genotypes; each set's corresponding phenotypes form an approximated Pareto front. The associated approximated Pareto front for each of these solution sets is called $PF_{current}(t)$, PF_{known} , and PF_{true}. Most of the methods based on MOEA attempt to evolve a population toward the *true Pareto frontier* PF_{true}. The hope is that by the end of the run, $P_{known} = P_{true}$, $P_{known} \subset P_{true}$, or $\{\bar{u}_i \in PF_{known}, \bar{u}_j \in PF_{true} : \forall i, \forall j \quad min[distance(\bar{u}_i, \bar{u}_j)] < \varepsilon\}$, where distance is defined over some norm (of course in an open problem we generally have no way of knowing P_{true}).

For solving the multicriteria ranking problem using a MOEA it is not necessary seek all the *Pareto optimal set* P_{true} or the associated Pareto front PF_{true} because of the fact that a lot of nondominated solutions are not of interest for the decision maker, we will use the strategy of attempting to find in each EA generation the most promising and attractive solutions for the decision maker which in our case are those individuals of which *u, f* score are near to value zero and has a sufficiently high value of

λ. Is sufficient to seek a *restricted Pareto optimal set*, which for our purpose it is defined as following.

$$P_{true}^{restricted} = \left\{ \bar{p} \in P_{true} : \| (u(\bar{p}), f(\bar{p})) \|_{\infty} \le \varepsilon \text{ , } where \ \varepsilon \ is \ a \ small \ no-negative \ \ number \right\} \quad \textbf{(5)}$$

Based on this strategy, the proposed method attempt to evolve a population toward the *true restricted Pareto frontier* $(PF_{true}^{restricted})$, by mean of a succession of restricted nondominated solutions subset $PF_{known}^{restricted}(t) = \{P_1^{(t)}, P_2^{(t)}, ..., P_n^{(t)}\}$. Note that the concepts *restricted Pareto optimal set* and *locally Pareto optimal set* are different [5].

Fitness Assignment Procedure. Main steps:

Step1. Let *N* be the population size. Choose a σ_{share} (a dynamically updated procedure for fixing σ_{share} is described later (Step 5)). Initialize $\lambda_j = c_j$, (c_j randomly chosen between 0 and 1), $\mu(j) = 0$ for all possible ranks, $j = 1,2,...,N$. Set solution counter $i = 1$.

Step2. Calculate the number of solutions (n_i) that dominates solution i. Compute the rank of the *i*-th solution as $r_i = 1 + n_i$. Increment the count for the number of solutions in rank r_i by one, that is, $\mu(r_i) = \mu(r_i) + 1$.

Step3. if $i < N$, increment i by one and go to step 2. Otherwise, go to step 4.

Step4. Identify the maximum rank $r*$ by checking the largest r_i, which has $\mu(r_i) > 0$. The sorting according to rank and fitness-averaging yields the following assignment of the average fitness F_i to any solution $i = 1,2,...N$:

$$F_i = N - \sum_{k=1}^{r_i^* - 1} \mu(k) - 0.5(\mu(r_i) - 1) \quad \textbf{(6)}$$

To each solution *i* with rank $r_i = 1$, the above equation assigns a fitness equal to

$$F_i = N - 0.5(\mu(1) - 1) \quad \textbf{(7)}$$

which is the average value of $\mu(1)$ consecutive integers from *N* to $N - \mu(1) + 1$. Set a rank counter $r = 1$.

Step5. For each solution *i* in rank *r*, calculate the distance count dc_i by using the equation:

$$dc(P_i^r) = dc_i = \begin{cases} \left\| P_i^r \right\|_\infty \big/ \sigma_{share}^r & , \quad if \quad \left\| P_i^r \right\|_\infty > \sigma_{share}^r \\ 1 & , \quad otherwise \end{cases} \tag{8}$$

where

$\sigma_{share}^r = \left\| P^{CM(r)} \right\|_\infty$, $P^{CM(r)} = Center\ of\ Mass\ of\ the\ set\ P^{(r)} = \{P_1^r, P_2^r, \dots, P_{\mu(r)}^r\}$ of solutions in rank *r*. The *Center of Mass* of a group of points is defined as the weighted mean of the points' positions. The weight applied to each point is the point's mass. $\left\| \bullet \right\|_\infty$ is the maximum holder metric. Note that $P^{(1)} = P_{current}^{restricted}$.

Calculate the shared fitness using $F_j^{'} = F_j \big/ dc_j$.

To preserve the same average fitness, scale the shared fitness as follows:

$$F_j^{'} \leftarrow \frac{F_j \mu(r)}{\sum_{k=1}^{\mu(r)} F_k^{'}} F_j^{'} \tag{9}$$

Step6. Increment λ_j by ε $(\lambda \leftarrow \lambda + \varepsilon)$ while $[u(\lambda) = u(\lambda + \varepsilon)]$ and $[f(\lambda) = f(\lambda + \varepsilon)]$.

Step7. If $r < r^*$, increment *r* by one and go to Step 5. Otherwise, the process is complete.

Crossover and Mutation Operators. Many crossover techniques exist in the literature (e.g. [21]), but, when working with ordinal (permutation) encoding, it is necessary to create both crossover and mutation operators that are specifics to this form of encoding. In this paper we make use of the crossover operator UX2 (Union Crossover #2) first introduced in [23]. The mutation operator works by interchanging two pairs of randomly chosen genes (actions), at each iteration under certain rules, in an individual.

Parent Selection Method. We developed a *Complement Selection (CS) method* for selecting parents that attempts to improve comparability as well as feasibility. The CS method is designed specifically for our problem and it takes into account the credibility level λ of the candidate parents.

In a complement selection, a parent $\bar{p}_i(\bar{p}_j)$ is first (second) selected using a *k*-ary tournament based on the *u* (*f*) function and credibility level; the rule is as follow: We selected the individual $\bar{p}_i(\bar{p}_j)$ which has lowest *u* (*f*) score and its credibility level $\lambda_{\bar{p}_i}(\lambda_{\bar{p}_j})$ is greater than λ_G or λ_K, where λ_G is the average credibility level of the population and λ_K is the average credibility level of the tournament. If *i* (*j*) is not unique, then we select the individual with higher credibility level score. If there is not such *i* (*j*) we tried the rule with the individual $\bar{p}_l$, which has next lower *u*(*f*) score; continue until the rule is satisfied. The logic here is that we would like the two parents together to cover as few amount of preference violations and incomparabilities between actions as possible, i.e. a low $u(\bar{p}_i)$ and $f(\bar{p}_j)$ with as high credibility level values $\lambda_{\bar{p}_i}$ and $\lambda_{\bar{p}_j}$ as possible.

Population Replacement Scheme. This MOEA part defines how new chromosomes will be put into the existing population. It is updating the current population continuously during the mating process, allowing new chromosomes act like parents to children in their own generation.
After that the child has been produced through the MOEA operators, it will replace the "less fitting" member of the population. The average of the population will improve if the child solution has lower scores than those of the solutions being replaced. In this algorithm, every new offspring is replacing the worst chromosome in the population. Each time that we replace a new offspring by the worst individual, the new population is sorted with the same criterion.

4 Computational Example

In order to benchmark the algorithm performance, some test problems were selected for numerical experiments. Here, only one test problem is shown. The proposed algorithm was coded in C++ and tested on a PC with processor Intel Pentium IV (1.5 GHz). In our computational study, 1 test problem instance of the MOEA heuristic was generated for solving the following test problem. The algorithm stopped when 10,000 populations had been generated. The population size was set to 40. The crossover probability was chosen 0.85 and the mutation probability was 0.30.

4.1 Test Problem 1

The fuzzy outranking relation is given by the following credibility matrix(10x10) between actions $A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9$ (Table 1).

Table 1. Credibility Matrix between actions $A_0,\ldots,A_9$.

	A0	*A1*	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*
A0	1.0	0.6	0.3	0.5	0.2	0.7	0.9	0.3	0.7	0.8
A1	0.2	1.0	0.4	0.3	0.4	0.5	0.5	0.5	0.6	0.3
A2	0.7	0.9	1.0	0.6	0.4	0.7	0.8	0.9	0.7	0.7
A3	0.6	0.8	0.3	1.0	0.4	0.7	0.8	0.4	0.3	0.8
A4	0.6	0.8	0.7	0.9	1.0	0.7	0.8	0.5	0.7	0.9
A5	0.4	0.65	0.4	0.4	0.2	1.0	0.4	0.3	0.5	0.5
A6	0.5	0.4	0.3	0.2	0.5	0.6	1.0	0.4	0.3	0.2
A7	0.6	0.8	0.3	0.8	0.2	0.8	0.9	1.0	0.8	0.7
A8	0.4	0.6	0.5	0.4	0.4	0.3	0.9	0.3	1.0	0.5
A9	0.5	0.8	0.3	0.6	0.3	0.5	0.4	0.3	0.7	1.0

Table 2. Restricted Pareto front found and the associated individual of the solutions space

Individual: Final set of solutions returned by the MOEA	$\widehat{p}_1$	$\widehat{p}_2$	$\widehat{p}_3$	$\widehat{p}_4$	$\widehat{p}_5$	$\widehat{p}_6$	$\widehat{p}_7$	$\widehat{p}_8$	$\widehat{p}_9$
	4	2	4	4	4	4	4	4	4
	2	7	2	7	2	7	2	0	2
	7	4	3	5	0	2	0	2	7
	3	3	7	2	7	0	7	7	0
	0	0	0	0	3	3	3	3	3
	9	9	9	3	5	9	9	9	9
	8	5	8	9	9	8	5	8	8
	1	1	1	8	8	1	1	1	1
	6	8	5	6	1	5	8	5	5
	5	6	6	1	6	6	6	6	6

Function *u*	01	02	03	07	00	02	00	01	00
Function *f*	06	06	06	06	13	13	14	14	26
Lambda	0.5992	0.5998	0.5999	0.6000	0.6497	0.6500	0.6997	0.7000	0.8000
Fitness value	41.44	41.44	41.44	41.44	31.30	31.30	29.06	29.06	15.65

Table 1 was processed with the proposal of section 3. The Restricted Pareto front $PF_{known}^{restricted}$ found and the associated final set of solutions returned by the MOEA at termination $P_{known}^{restricted}$ are presented in Table 2.

The number $T(i,j), \quad (1 \le i, j \le m)$, of times that an alternative was found at a certain place in the ranking of the individual associated to the members of the final restricted Pareto front is given in table 3. Based on the table 3 we found a compromise solution following the next procedure: as the ranking of the alternatives is of significant importance, the number of times that an alternative is found at a certain place in the ranking is weighted according to the importance of the alternatives to be ranked. It is reasonable to conclude that in certain cases, the rank of the alternatives would not be of equal importance.

Table 3. The number of times that an alternative was found at a certain place in the ranking

Weight w_i	Rank	A_4	A_2	A_7	A_0	A_3	A_9	A_8	A_5	A_1	A_6
10	1	8	1	0	0	0	0	0	0	0	0
9	2	0	5	3	1	0	0	0	0	0	0
8	3	1	2	2	2	1	0	0	1	0	0
7	4	0	1	4	2	2	0	0	0	0	0
6	5	0	0	0	4	5	0	0	0	0	0
5	6	0	0	0	0	1	7	0	1	0	0
4	7	0	0	0	0	0	2	5	2	0	0
3	8	0	0	0	0	0	0	2	0	7	0
2	9	0	0	0	0	0	0	2	4	1	2
1	10	0	0	0	0	0	0	0	1	1	7
$\sum_{i=1}^{m} w_i T(i,j)$		88	78	71	63	57	43	30	30	24	11
Minimum Lamda= 0.5992											

Relative importance could be reflected in a weighting $w_i = m - i + 1$ of each rank *i*, where *m* is the length of an individual. After that, we calculate the weighted sum $\sum_{i=1}^{m} w_i T(i,j)$, *j*=1,2, …, *m*. Finally, we obtaining a succession in decreasing order of preference, generating of this manner, a recommendation for the decision maker.

Table 3 suggests the following final ranking

$$A_4 \succ A_2 \succ A_7 \succ A_0 \succ A_3 \succ A_9 \succ A_8 \succ A_5 \succ A_1 \succ A_6 \qquad \mathbf{(10)}$$

where $A \succ B$ means that "alternative "A" is preferred to alternative "B"".

The Genetic algorithm of [19] obtains the following results: The number of times that an alternative was found at a certain place in the ranking is given in table 4 with respect to 100 variations in the seed parameter.

Table 4. Results of the Genetic Algorithm of [19].

	A_4	A_2	A_7	A_0	A_3	A_8	A_5	A_9	A_6	A_1
1	81	12	0	4	0	2	0	0	0	0
2	4	86	6	2	0	2	0	0	0	0
3	8	0	54	36	0	2	0	0	0	0
4	4	0	34	28	18	6	4	6	0	0
5	2	0	2	18	46	20	4	8	0	0
6	0	2	2	10	20	22	22	20	0	2
7	0	0	2	2	14	20	30	22	8	2
8	0	0	0	0	0	16	12	34	14	24
9	0	0	0	0	0	10	20	8	36	26
10	0	0	0	0	2	0	8	2	42	46

The best compromise solutions were individuals in which $u = 0$, $f = 13$, $\lambda = 0.69$ mostly corresponding to rankings $A_4 A_2 A_7 A_0 A_3 A_9 A_8 A_5 A_1 A_6$ and $A_4 A_2 A_0 A_7 A_3 A_9 A_8 A_1 A_5 A_6$. These results, combined with Table 4, suggest the final ranking:

$$A_4 \succ A_2 \succ A_7 \succ A_0 \succ A_3 \succ A_9 \succ A_8 \succ A_5 \succ A_1 \succ A_6 \quad \textbf{(11)}$$

Although without a clear preference between A_5 and A_1.

ELECTRE-III suggest the following ranking:

$$A_4 \succ (A_2, A_7) \succ A_0 \succ A_3 \succ (A_9, A_8) \succ A_5 \succ A_1 \succ A_6 \quad \textbf{(12)}$$

In this ordering, we have $u = 0$ with $\lambda = 0.69$, but we can hardly agree with the fact that A_2 and A_7 are posed in the same position. It is clear that $A_2 S^{\lambda} A_7$ and the contrary is false. The net flow score associated to A_2 is higher, and also, if we consider the set $\boldsymbol{A} - \{A_4\}$, A_2 is the best alternative in the sense of [22].

5 Conclusions

This paper presents a Multiobjective Evolutionary Algorithm suitable for exploiting fuzzy outranking relations. The basic concepts and formulations of the new method were given with an improved technique to handle the Pareto front. The intrinsic advantages of the method are that it is not sensitive to the shape of the Pareto front and that it is partially adaptive to inadequate input from a user as a complex graph with several cycles. The most similar method, the genetic algorithm of [19], was benchmarked against the new MOEA. The comparison study contrasted both methods in generation of efficient solutions and in the final recommendation. The MOEA present a wide area of efficient solutions whilst the other generates only one. Moreover, the recommendation for decision maker is obtained with only one run whilst the genetic algorithm procedure should be run n times for obtaining a final recommendation. In the numerical experiments carried out, our proposal performed very well, in the sense of quality of solution as well in the sense of computational effort. The Multiobjective Evolutionary Algorithm procedure also looks more effective than other approaches that exploit a fuzzy outranking relation; it could work easily with preference graphs of considerable size and it looks more robust than other dealing with irrelevant alternatives. Approaches based on a function score do not have a way to reduce nonrational violations of explicit preferences, while the evolutionary one may identify this kind of irregularity and try to control and minimize it.

References

1. Brans, J. P., Vincke, Ph.:A Preference Ranking Organization Method. Management Science 31 (1985) 647-656
2. Bouyssou, D., Vincke, Ph.: Ranking Alternatives on the Basis of Preference Relations: A Progress Report with Special Emphasis on Outranking Relations. Technical report *IS-MG 95/03.* Institut de Statistique et de Recherche Opérationnelle, Universite Libre de Bruxelles. Serie: Mathématiques de la Gestion (1995)
3. Coello, C.A.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems 1 (1999) 269-308
4. Coello, C.A., Van Veldhuizen, D.A., and Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers New York (2002)
5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Son, Chichester, UK (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 6 (2002) 182-197
7. Fernandez E., Leyva López J. C.: A method based on multiobjective optimization for deriving a ranking from a fuzzy preference relation. European Journal of Operational Research 154 (2004) 110-124
8. Fodor J., Roubens M: Fuzzy Preference Modeling and Multicriteria Decision Support. Kluwer, Dordrecht (1994)
9. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. John Wiley, New York (1966)
10. French S.: Decision Theory: An Introduction to the Mathematics of Rationality, Halsted Press, New York Chichester Brisbane Toronto (1986)
11. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Genetic Algorithms: Proceedings of the Fifth International Conference. Morgan Kaufmann (1993) 416-423
12. Fonseca, C.M., Fleming, P.J.: An Overview of Evolutionary Algorithm in Multiobjective Optimization. Evolutionary Computation. 3 (1995) 1-16
13. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley (1989)
14. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor Michigan (1975)
15. Horn, J.: Multicriterion Decision Making. In: Back, T., Fogel, D. B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. IOP Publishing Ltd and Oxford University Press, Bristol U.K. (1997)
16. Horn, J. Nafploitis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization". In: Michalewicz, Z. (ed.): Proceeding of the First IEEE Conference on Evolutionary Computation.: IEEE Service Center, Piscataway, New Jersey (1994) 82-87
17. Keeney R., Raiffa, H.: Decision with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York (1976)

18. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front using the Pareto Archived Evolution Strategy. Evolutionary Computation. 8 (2000) 149-172
19. Leyva-López J.C., Fernández-González, E.: A Genetic Algorithm for Deriving Final Ranking from a Fuzzy Outranking Relation. Foundations of Computing and Decision Sciences. 24 (1999) 33-47
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, Springer – Verlag, Berlin Heidelberg New York (1996)
21. Ordoñez Reinoso G., Valenzuela Rendón M.: Permutation Optimization with Genetic Algorithms: The Traveling Salesman Problem (in Spanish). Proc. of the 3th. Latin-American Congress of Artificial Intelligence. (1992) 271-282
22. Orlovski, S.A.: Decision-Making with a Fuzzy Preference Relation. Fuzzy Sets and Systems 1 (1978) 155-167
23. Poon P.W., Carter J.N.: Genetic Algorithm Crossover Operators for Ordering Applications. Computers & Operations Research 22 (1995) 135-147
24. Roy B.: Multicriteria Methodology for Decision Aiding. Kluwer (1996)
25. Roy B.: The Outranking Approach and the Foundations of ELECTRE Methods. In Bana e Costa, C.A. (ed.): Reading in Multiple Criteria Decision Aid. Springer-Verlag, Berlin (1990) 155-183
26. Roy B.: Decision-Aid and Decision-Making. European Journal of Operational Research. 45 (1990) 324-331
27. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Grefenstette, J.J. (ed.): Genetic algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms. Hillsdale, New Jersey: (1985) 93-100
28. Schwefel, H.P.: Numerical Optimization of Computer models. Wiley, Chichester UK (1981)
29. Srinivas, N., Deb, K.: Multiobjective Function Optimization using Nondominated Sorting Genetic Algorithms. Evolutionary Computation. 2 (1995) 221-248
30. Triantaphyllou, E.: Multicriteria Decision Making Methods: A Comparative Study. Kluwer Academic Publishers, Boston MA (2000)
31. Vanderpooten, D.: The Construction of Prescriptions in Outranking Methods. In: Bana e Costa, C.A. (ed.): Reading in Multiple Criteria Decision Aid, Springer-Verlag, Berlin (1990) 184-215
32. Van Veldhuizen, D.A., Lamont, G.A.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation. 8 (2000) 1-26
33. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, (2001)