Proceedings of the 5th WSEAS/IASME Int. Conf. on SYSTEMS THEORY and SCIENTIFIC COMPUTATION, Malta, September 15-17, 2005 (pp260-265)

An Interactive Procedure for Multiobjective Optimization

Inès Ben Jaafar Khaled Ghédira
Department of computer science
High Institute of management
41, rue de la liberté, 2000 cité Bouchoucha.
TUNISIA
http:// www.soie.isg.rnu.tn

Abstract: - The majority of approaches for finding best compromise solutions to multiobjective optimization problems (MOP) make use of the Pareto optimality concept. However, in modeling real world problems, we often encounter MOP with large Pareto optimal alternatives to choose from. This paper introduces the concept of a-efficiency, which provides a notion that is stronger than Pareto optimality and allows setting up a preference ordering amongst various alternatives that are Pareto optimal. If the user still has to process quite a large number of alternatives, we propose to arrange them using an interactive approach based on a plurality voting procedure. This interactive procedure is based on a binary preference relation to rank the solutions set. However, for more flexibility this interactive procedure is extended exploiting fuzzy preference relation.

Key-Words: - Multiobjective optimization, decision making, Pareto optimality, a-efficiency, vote plurality, interactive methods, fuzzy preference relation.

1 Introduction

In Multiobjective optimization problems, decision makers need to optimize more than one objective and to set up an order of preference among various available alternatives. Some tools consist in aggregating the different points of view into an unique function, which must be solved through standard single objective algorithms. Unfortunately this is very quite inadequate, because the objectives generally are incommensurable. They measure different properties that cannot be related to each other directly and cannot be combined into a single function [3]. Moreover, if the problem's objective space (i.e., the image through the objective functions of the set of solutions) is not convex, there is no guarantee that all the non-dominated points can be generated. Therefore, such scheme reflects neither real optimization nor the expected result from all objective points of view [4,15,16].

These methods try to translate a Multicriteria Optimization Problem (MOP) into one or more constrained optimization problems; they do not address directly the MOP problem. Other methods address the MOP but are incomplete (i.e., can't provide the hole non-dominated frontier), exploiting simulated annealing [3] genetic algorithms [5] and tabu search [6].

Also, there are a great number of multicriteria methods falling within the interactive local judgment with trial and error iterations [13]. In these methods, computation steps providing successive trade-offs and dialogue steps giving additional information on the decision-maker's preferences alternate.

Indeed, the majority of these methods miss axiomatic foundations, and it is difficult to choose the method to be applied to a given situation. In this paper, we propose an axiomatic approach for multiobjective optimization. This approach is based on concepts such as the extended efficiency [8] and the partial efficiency [10,12]. It can be used for solving various multiobjective situations such as the problems involving many incommensurable objectives and the public decision problems.

Very often in engineering applications when several-often contradictory-points of view must be taken into account, a reasonable approach is to generate the efficient, i.e. Pareto optimal solutions. Eliminating choices that are not Pareto optimal is a technique, which avoids the formation of a scalar measure in the process of optimization. However, Pareto optimality alone is not always adequate for generating the final decision because the set of Pareto optimal solutions is often very large [11,12]. So even after eliminating all the alternatives that are not Pareto optimal, the decision maker is usually left with a huge number of solutions to choose from. To discard this drawback, we introduce the notion of order of efficiency [11] which provides a concept that is stronger than Pareto optimality and allows to set up an order of preference among various available Pareto solutions and designate some Pareto points as being superior to others [12]. We also present an interactive process based on a plurality voting procedure that we have developed and which enhance the order of preference whenever the set of 'superior' Pareto optimal solutions is still large. Then, this interactive process is extended exploiting fuzzy preference relation in order to rank the solutions set.

This paper is structured as follows. In the section 2 we give the general basic concepts. In the section 3 we describe the pre-processing approach for generating the a-efficient solutions and an interactive procedure for extracting the most preferred Pareto optimal solutions. Section 4 illustrates the interactive procedure exploiting fuzzy preference relation and section 5 provides simulation results. Finally, section 6 deals with some concluding remarks and further researches.

2 General basic concepts

We introduce some concepts used in the rest of the paper.

Definition 2.1 (Multiobjective Optimization Problem)

A Multiobjective Optimization Problem (MOP) can be stated as:

$$\max_{x \in A} F(x) = [f_1(x), f_2(x),\ldots, f_n(x)] \quad n \geq 2$$

Where A denotes the feasible set of design alternatives, or the design space [14,16] and n the objective number.

Definition 2.2 (Pareto optimality)

A point $U^* \in A$ is said to be Pareto optimal or an efficient point for (MOP) if and only if for every $U \in A$ and $I = \{1,2,\ldots,n\}$ either;

$\forall i \in I\ f_i(U) = f_i(U^*)$

or there is at least one $i \in I$ such that

$f_i(U) < f_i(U^*)$

In words, this definition means that U^* is Pareto optimal if there exists no feasible solution U which would increase some criterion without causing a simultaneous decrease in at least one other criterion.

Definition 2.3 (Pareto dominance)

The vector $F(U)$ is said to dominate another vector $F(V)$, denoted $F(U)>F(V)$, if and only if $f_i(U) \geq f_i(V)$ and there exist at least one j such as $f_j(U)>f_j(V)$ $i, j \in \{1,2,...n\}$.

The dominance is the most used concept to reduce the set of the candidate solutions to the final decision. It is considered as the least controversial tool in MOP. Unfortunately, this tool is little discriminating and can be exploited only in the preliminary solving steps. For that reason, various forms of stronger dominances were proposed in literatures [11]. Among these extensions, we present the partial dominance [11,12] that is used in this work.

Definition 2.4 (Partial dominance)

The vector $F(U)$ is said to partially dominate another vector $F(V)$ for a criterion sub-set $B \subseteq I$, if and only if $f_i(U) \geq f_i(V)$ for all $i \in B$ and there exist at least one $j \in B$ such as $f_j(U)>f_j(V)$

The introduction of the partial dominance formalizes the following intuition: if a solution x dominates a solution y for a coalition of criteria, then x is better than y for this coalition.

Definition 2.5 (Partial efficiency)

Let $I' \subseteq I$. A point $x^* \in A$ is said to be *partially-efficient* for I' if and only if x^* is an efficient point for the problem P':

$$\max_{x \in A} F(x) = [f_1(x), f_2(x),\ldots, f_k(x)] \quad k \in I'$$

Hence, The partial efficiency corresponds to the efficiency of the solution for a restricted set of criteria.

Definition 2.6 (a-dominance)

Let a such that $(1 \leq a \leq n)$, The vector $F(U)$ is said to a-dominate another vector $F(V)$, if and only there exist $I_{(n+1-a)} \subseteq I$ such that: $f_k(U) \geq f_k(V)$ for all $k \in I_{(n+1-a)}$ and $f_k(U)>f_k(V)$ for at least one $k \in I_{(n+1-a)}$.

Note that $I_{(n+1-a)}$ is the index set of a subset of $(n+1-a)$ criteria.

A solution U *a-dominate* a solution V if U *partially dominate* V for a coalition with at least $(n+1-a)$ criteria.

Definition 2.7 (a-efficiency)

A point $x^* \in A$ is said to be ***a-efficient*** for (MOP) if and only if $F(x^*)$ is non ***a-dominated***.

An a-efficient point is the one that is partially-efficient for all the coalition with *(n+1-a)* criteria. ***a-efficiency*** means that it is not possible to increase anyone's utility without decreasing at least the utility of ***a*** criteria [12].

Definition 2.8 Let *W Î A* , ***a*** et *ā Î{ 1,2,...n}*. If ***W*** is a-***efficient*** and there does not exist *U Î A* such that *U* is ā-*efficient* and *ā>a* . The point ***W*** is then called a_{max}-*efficient.*

3. The proposed approach

3.1 Pre-processing procedure

The idea of the proposed approach is inspired from cooperative problem-solving methods, which distribute the problem and then allow the entities to work cooperatively on their local problems [2,7]. It benefits from the multi-agent techniques [2,7] that have opened an efficient way to solve diverse problems in terms of cooperation, conflict, coordination and concurrence within a society of agents. Each agent is an autonomous entity that is asynchronously able to acquire information from its environment and/or from other agents, to reason on the basis of this information and to act consequently. In this approach for dealing with multiobjective optimization, decision makers are autonomous and not required to know each other's value functions. They are assumed to be able to agree on the overall objectives although their opinions about the relative importance of each criterion may differ. With no loss of generality, we suppose that all the objectives have the same importance. Thus, each decision maker is responsible for a single objective that he aims to optimize and he interacts with the other decision makers and work together to reach a compromise set of solutions that satisfies the multiobjective optimization problem. Consequently, the objectives are separately optimized without any "scalarization" form, via interactions and communications. The approach results into an interactive procedure between the decision makers and a supra decision maker. Here the supra decision maker is a coordinator who tries to help the negotiating parties to find efficient agreements [9]. At the beginning of the procedure for generating Pareto optimal solutions, the supra decision maker decomposes the multiobjective optimization problem and assigns an objective for each decision maker. At the lower level each decision maker solves his own optimization problem. He detects all the feasible solutions and computes for each one its cost according to his own criterion. Then, he informs the supra decision maker about all his solutions. The role performed by a decision maker in this step is a pre-processing procedure that precedes the multiobjective optimization. When receiving all the generated solutions, the supra decision maker computes all the Pareto optimal solutions. To find out this set, he is not limited to any specific method, the main objective is to apply a method that leads to all the Pareto optimal solutions. Then, he uses the a-*efficiency,* which provides an intermediate concept of branding some Pareto optimal points as being perhaps superior or more desirable than others. One way to choose the final Pareto optimal points would be to compute progressively the ***a-efficient*** points and to increase the ***a*** value at each step ($a = 1..a_{max}$). When the set of the a-efficient solutions is empty the supra decision maker stops the process and considers the subset of the last found solutions as the set of the a_{max}-*efficient* solutions. It is clear that a_{max}-efficiency provides alternatives satisfying the strongest requirements and eliminating ones, which are inferior. The supra decision maker retains the following set of solutions and communicates it to the several decision makers as a result of the multiobjective optimization problem.

We show now a generic algorithm for generating a_{max}-*efficient* solutions (E_{a-max}):

1. let S_{eff} be the Pareto optimal solutions set.
2. let a =1
3. Let $E_a = S_{eff}$
4. $E_{a-max} = E_a$
5. While $E_a \neq \emptyset$
6. a = a+1
7. Find $S \subseteq E_{a-1}$, such as: $\forall s \in S$ and $s' \in E_{a-1} - \{S\}$, s a-dominate s'
8. $E_a = \{S\}$
9. If $E_a \neq \emptyset$, $E_{a-max} = E_a$
10. End while.
11. Return E_{a-max}

Algorithm1. Generating a_{max}-*efficient* solutions

3.2 Interactive procedure

Often the process of reducing the set of alternatives to retain points with the upper a-efficiency may not yield the desired reduction in the number of options. Though a reduced set, the decision maker still has to process quite a large number of alternatives. In the following, we introduce a plurality voting procedure, which will allow to further reducing the set of alternatives in such a case.

At this level, the supra decision maker informs the decision makers to start up the plurality voting procedure on the bases of the a_{max}-efficient solutions set found earlier and which we denote $E_{a\text{-}max}$.

Thus, the decision maker behavior is to iteratively compute, both its ideal id_i and anti-ideal aid_i costs as follows:
$aid_i = Min(f_i(x))$ (for maximization), $x \in E_{a\text{-}max}$, $i \in \{1,2,...n\}$, $id_i = Max(f_i(x))$ (for maximization), $x \in E_{a\text{-}max}$, $i \in \{1,2,...n\}$, and uses these parameters to compute its satisfaction level sl_i, $sl_i = aid_i + (idi - aidi)*e$
e is an adjustment parameter in $]0,1[$ used to gradually increase sl_i and consequently to set up an order of preference among the set of a_{max}-efficient solutions $E_{a\text{-}max}$. If the decision maker chooses greater values for this parameter, his satisfaction level will reach his ideal cost in a few number of iterations. In addition to this parameters, the decision maker has to separate the $E_{a\text{-}max}$ set into satisfied (respectively unsatisfied) solutions subsets noted *SSol* (respectively *USol*) such that $SSol = \{ x \in E_{a\text{-}max} / f_i(x) \geq sl_i\}$ and $USol = \{ x \in E_{a\text{-}max} / f_i(x) < sl_i\}$. The *USol* subset is considered as the solutions set that the decision maker prefers at least and which he would delete from the $E_{a\text{-}max}$ whenever the most of the other decision makers have the same agreements. So he informs every one's about his *USol* subset and asks them to establish their vote. For the others decision makers, if any received solution is belonging to their *USol* subset they agree to delete it from the $E_{a\text{-}max}$.. So, every one votes for each solution in the *USol* subset received, by assigning the value 1 to its vote vector if he agrees to delete the solution and 0 if he refuses.
As the expected answers are received, the decision maker sets up for every solution in his *USol* subset the related vote vector and removes from $E_{a\text{-}max}$ any solution the majority of the decision makers agree to delete. Then, he informs the supra decision maker about this removal.

When there are no agreements for discarding any solution from the *USol* subset, the decision maker keeps this solution in the $E_{a\text{-}max}$ set. It will be removed in the next iterations when the decision makers will increase their satisfaction level and this solution becomes unsatisfied for some of them.

Under the assumption that all of the decision makers agree to remove the least preferred solutions from $E_{a\text{-}max}$ the supra decision maker updates this set of solutions and arranges the removed solutions as the first subset of the least preferred a_{max}-*efficient* solutions. Next, he communicates the modified $E_{a\text{-}max}$ to all the decision makers and asks them to increase their satisfaction level and to restart a new plurality voting procedure. The process terminates whenever the size of the a_{max}-*efficient* solutions is equal to one or if after two successive iterations no new solution is removed. The last subset of solutions arranged $LPsol_k$ is considered as the most preferred subset among the various a_{max}-*efficient* solutions available because the satisfaction level of the decision makers are almost near to their ideal parameter and their *SSol* subset contain their "best" optimal solutions. Note that, for greater values of the adjustment parameter e, the decision maker is able to detect the most preferred solutions from the $E_{a\text{-}max}$ subset in a limited number of iterations, however, he will obtain a few number of $LPsol_k$ subsets containing solutions that could be further arranged.

In this manner, we can further reduce the set of available alternatives beyond that achieved by using the a-*efficiency*.

4 Interactive procedure exploiting fuzzy preference relation

The proposed interactive procedure is based on a binary preference relation in order to rank the solutions set. A solution is satisfying (preferred for) the decision maker only if it's cost is greater than the satisfaction level fixed, otherwise it is considered as unsatisfying one. In this extended version we will consider a fuzzy preference relation allowing the decision maker to have not only satisfied and unsatisfied solutions [12,13] but also solutions for which he is indifferent. For this purpose we will use the "pseudo- criteria" concept [13].
A pseudo-criterion is a preference model which includes two different thresholds : preference threshold and indifference threshold for each criterion f_i $(i = 1..n)$. These thresholds may be constant, linear or affine. For every criterion f_i, the double thresholds model is the following:
When $f_i(x) > f_i(y) + p_i(f_i(y))$, x is preferred to y.
When $f_i(y) + p_i(f_i(y)) \geq f_i(x) > f_i(y) + q_i(f_i(y)$, x is weakly preferred to y.
When $f_i(y) + q_i(f_i(y) \geq f_i(x)$ and $f_i(x) + q_i(f_i(x)) \geq f_i(y)$ x is indifferent to y.
Where $p_i(f_i(y))$ and $q_i(f_i(y))$ are preference and indifference thresholds, respectively. Weak preference is supposed to describe the decision maker's hesitation between indifference and preference.

Proceedings of the 5th WSEAS/IASME Int. Conf. on SYSTEMS THEORY and SCIENTIFIC COMPUTATION, Malta, September 15-17, 2005 (pp260-265)

In our work, solutions are not compared to each other, but each one is only compared to the id_i and aid_i solutions. However, we introduce some modifications in the indifference and preference thresholds definition.

The current interactive procedure differs from the earlier one by the use of the indifference threshold q_i, a constant value such as $q_i = aid_i + (id_i - aid_i)*e$, and the preference threshold $p_i = id_i - (id_i - aid_i)*e, e \in]0,1[$. Therefore, the decision maker is indifferent between all the solutions *x* such as $aid_i£f_i(x)£q_i$. They are unsatisfied and should be deleted from the $E_{a\text{-}max}$ whenever the most of the decision makers have the same indifference. The solutions *y* such as $p_i£f_i(y)<id_i$ are considered as the most preferred ones, while solutions *w* such as $q_i<f_i(w)<p_i$ are weakly preferred.

As stated before, the decision maker computes, the ideal id_i and anti-ideal aid_i costs, the indifference q_i and preference p_i thresholds. Next, these parameters are used to detect the set of unsatisfied solutions $USol = \{ x Î E_{a\text{-}max} / aid_i£f_i(x)£q_i \}$. The decision maker *i* would delete this set of solutions from the $E_{a\text{-}max}$, so he informs every one's about his $USol_i$ subset and asks them to establish their vote. For the other decision makers *j*, three cases may occurs:

Case1: $\forall$ y *Î* $USol_i$ such as $f_j(y)=aid_j$ or $aid_j<f_j(y)£q_j$
The decision maker *j* agrees to delete *y* from $E_{a\text{-}max}$
Case2: $\forall$ y *Î* $USol_i$ such as $f_j(y)=id_j$ or $p_j£f_j(y)<id_j$
The decision maker *j* doesn't agree to delete *y* from $E_{a\text{-}max}$
Case3: $\forall$ y *Î* $USol_i$ such as $q_j<f_j(y)<p_j$
The decision maker *j* hesitates, the solution is weakly preferred, he doesn't vote.

When the votes are received, the decision maker *i* computes a qualification score for each solution y *Î* $USol_i$. This qualification score is formally defined as:
$Qualification(y)=\sum j_{\, j|y Î [pj, idj]} - \sum j_{\, j| y Î [aidj, qj]}$
Consequently, each decision maker communicates to the SDM the set of solutions having the minimal qualification. The SDM compares the qualification score of each solution and removes those having minimal score from $E_{a\text{-}max}$. Hence, The removed solutions are ranked as the first subset of the least preferred a_{max}-*efficient* solutions.

5 Simulation results

The experiments are based on randomly generated binary CSMOPs (Constraint Satisfaction and Multicriteria Optimization Problems)[1]. Table 1 shows the average number of Pareto optimal solutions obtained for different values of the criterion number and the corresponding a-efficient solutions. The concept of a-*efficiency* has enabled us to reduce the number of possible options down to 1.5% of the original number of solutions.

Criterion number	Efficient solutions	2efi solutions	3efficient solutions	4efficient solutions	5efficient solutions	6efficient solutions	7efficient solutions
5	218	13	1	0	0	0	0
6	302	19	2	0	0	0	0
7	263	–	4	0	0	0	0
8	360	–	13	1	0	0	0
9	329	–	–	4	0	0	0
10	491	–	–	6	2	0	0
11	452	–	–	–	4	0	0
12	502	–	–	–	6	1	0
13	433	–	–	–	–	3	0
14	467	–	–	–	–	4	1
15	540	–	–	–	–	5	2

Table 1. Average number of the obtained solutions

The next experimentations illustrate that the concept of *a-efficiency* could reduce effectively the final decision. It should be noted that the average portion of the 2-efficient solutions (resp. 3-efficient solutions) represents 25.53% (resp. 6.87%) of the total number of efficient solutions.

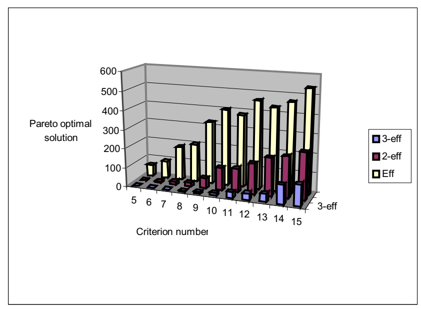

Figure1. The Improvements provided by the a-efficiency to reduce the Pareto optimal solutions set.

Next, we have applied the interactive process(1) and the interactive process(2) exploiting fuzzy preference relation, on the same problems. For each criterion number we have generated 15 problem instances and calculated the average number of solutions. Table2 shows that applying the interactive process(1) doesn't eliminate any solution from the a_{max} efficient solutions. However, the interactive process(2) table3 allows the decision maker's to eliminate yet other points from the set of a_{max} efficient solutions. Here the number of cycles for the interactive process(2) varies from 2 to 4 iterations. At each cycle there is a number of solutions that is removed, and the size of the final choice can be equal to only one solution. So this proves that the decision maker can set up preference ordering among a_{max}-efficient solutions.

Criterion number	*Efficient Solutions*	$E_{a\text{-}max}$	$E^k_{a\text{-}max}$	*Interactive procedure(1) cycles*	*CPU(ms)*
5	91	7	7	1	3149
6	130	14	14	1	4158
7	179	5	5	1	12127
8	186	5	5	1	20581
9	246	9	9	1	39843
10	226	6	6	1	24560
11	251	5	5	1	27212
12	255	5	5	1	43942
13	273	5	5	1	51267
14	252	3	3	1	34231
15	245	3	3	1	36158

Table 2. Average number of the obtained solutions by the interactive procedure

Criterion number	*efficient Solutions*	$E_{a\text{-}max}$	$E^k_{a\text{-}max}$	*Interactive procedure(2) cycles*	*CPU(ms)*
5	91	7	1	4	4093
6	130	14	1	3.66	5195
7	179	5	1	3.33	10703
8	186	5	1	3.34	19036
9	246	9	2	2.73	39132
10	226	6	2	3.02	23074
11	251	5	2	2.61	20833
12	255	5	1	3.26	40478
13	273	5	2	2.69	48103
14	252	3	1	2.38	35766
15	245	3	1	2.38	38409

Table 3. Average number of the obtained solutions by the interactive procedure exploiting fuzzy preference relation

6 Conclusion and future work

In this paper, we have developed an interactive procedure for dealing with multiobjective optimisation problem. Each decision maker is responsible for a single objective. They interact in order to find out and to reach their 'best' Pareto optimal solutions. Next, this interactive procedure is extended exploiting fuzzy preference relation. The effectiveness of the proposed approaches is demonstrated and tested on randomly generated CSMOP examples. Using the notion of a-*efficiency* and the interactive procedure in succession could help eliminate some alternatives among the multiple non–inferior ones. However, we shall focuses on an efficient algorithm for identifying points with a-*efficiency*.

References:

[1] BenJaafar I., Khayati N., Ghedira K. Multicriteria Optimization in CSPs : Foundations and Distributed Solving Approach, *In the Proceedings of the 11th International Conference on Artificial Intelligence: Methodology, Systems, and Applications, AIMSA*, Lecture Notes in Computer Science 3192 Springer, 2004.

[2] S.H.Clearwater, B.A.Huberman and T.Hogg, Cooperative problem Solving, *In Huberman, B., editor, Computation: The Micro and the View*, 1992,

[3] Y.Collette and P.Siarry, Multiobjective optimisation, EYROLLES, 2004.

[4] P.C.Fishburn, Utility theory for Decision-Making, WILEY, 1970.

[5] C.Fonsesca and P.Fleming, An overview of evolutionary algorithms in Multiobjective optimisation, *Evolutionary Computation*, 3(1), 1995.

[6] M.Hansen, Tabu search for Multiobjective optimization: MOTS, *in 13th International Conference on multiple criteria decision making*, 1997.

[7] B.A.Huberman, The performance of cooperative process, *Phisica D*, 1990.

[8] R.L.Keeny and H.Raiffa, Decision with multiple objectives: preferences and value tradoffs, WILEY, 1976

[9] R.L.Keeny and C.W. Kirkwood, Group decision making using cardinal social Welfare functions, *Management Science*, vol(22), 1975.

[10] B.Matarazzo, Multicriteria analysis of preferences by means of pair wise actions and criteria comparisons MAP-PACC, *Applied mathematics and computation,* vol(18), 1986.

[11] B.Matarazzo, PCCA and K-dominance in MCDM, *Belgian Journal of Operations Research, Statics and Computer Science,* vol(30), 1990.

[12] I.Othmani, A new concept for multiple criteria optimization: the a-efficient solution, *International conference on methods and applications of Multicriteria Decision Making*, FUCAM, 1997.

[13] B.Roy and D.Bouyssou, Aide multicritère à la décision: Méthodes et Cas, ECONOMICA, 1993.

[14] R.B. Statnikov and J.B. Matusov, Multicriteria Optimization and Engineering, Chapman and Hall, New York, 1995.

[15] R.E. Steuer, Multiple criteria optimization: Theory, computation and application, WILEY, 1986

[16] P.Vincke, Multicriteria Decision -Aid, JHON WILEY & SONS,1989.