Applied Mathematics and Computation 213 (2009) 455–465

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems

George Mavrotas

Laboratory of Industrial and Energy Economics, School of Chemical Engineering, National Technical University of Athens, Zographou Campus, Athens 15780, Greece

ARTICLE INFO

Keywords:
Multi-Objective Programming
ε-Constraint method
GAMS

ABSTRACT

As indicated by the most widely accepted classification, the Multi-Objective Mathematical Programming (MOMP) methods can be classified as a priori, interactive and a posteriori, according to the decision stage in which the decision maker expresses his/her preferences. Although the a priori methods are the most popular, the interactive and the a posteriori methods convey much more information to the decision maker. Especially, the a posteriori (or generation) methods give the whole picture (i.e. the Pareto set) to the decision maker, before his/her final choice, reinforcing thus, his/her confidence to the final decision. However, the generation methods are the less popular due to their computational effort and the lack of widely available software. The present work is an effort to effectively implement the ε-constraint method for producing the Pareto optimal solutions in a MOMP. We propose a novel version of the method (augmented ε-constraint method – AUGMECON) that avoids the production of weakly Pareto optimal solutions and accelerates the whole process by avoiding redundant iterations. The method AUGMECON has been implemented in GAMS, a widely used modelling language, and has already been used in some applications. Finally, an interactive approach that is based on AUGMECON and eventually results in the most preferred Pareto optimal solution is also proposed in the paper.

1. Introduction

The solution of Mathematical Programming (MP) problems with only one objective function is a straightforward task. The output is the optimal solution and all the relevant information (dual prices, etc.). In Multi-Objective Mathematical Programming (MOMP) there are more than one objective functions and, in general, there is no single optimal solution that simultaneously optimizes all the objective functions. In these cases the decision makers are looking for the "most preferred" solution, in contrast to the optimal solution. In MOMP the concept of optimality is replaced with that of Pareto optimality or efficiency. The Pareto optimal (or efficient, non-dominated, non-inferior) solutions are the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest. The set of the Pareto optimal solutions is the Pareto set. The mathematical definition of the efficient solution is the following (without loss of generality assume that all the objective functions f_i, $i = 1,\ldots,p$ are for maximization): A feasible solution $\mathbf{x}$ of a MOMP problem is efficient if there is no other feasible solution $\mathbf{x}'$ such as $f_i(\mathbf{x}') \geqslant f_i(\mathbf{x})$ for every $i = 1,2,\ldots,p$ with at least one strict inequality. Every efficient solution corresponds to a non-dominated or non-improvable vector in the criterion space. If we replace the condition $f_i(\mathbf{x}') \geqslant f_i(\mathbf{x})$ with $f_i(\mathbf{x}') > f_i(\mathbf{x})$ we obtain the weakly efficient solutions. Weakly efficient solutions are not usually pursued in MOMP because they may be dominated by other efficient solutions. The rational decision maker is looking for the most preferred solution among the Pareto optimal solutions of the MOMP.

E-mail address: mavrotas@chemeng.ntua.gr

0096-3003/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2009.03.037

According to Hwang and Masud [1], the methods for solving MOMP problems can be classified into three categories, based on the phase in which the decision maker involves in the decision making process expressing his/her preferences: The a priori methods, the interactive methods and the a posteriori or generation methods. In a priori methods the decision maker expresses his/her preferences before the solution process (e.g. setting goals or weights to the objective functions). The criticism about the a priori methods is that it is very difficult for the decision maker to know beforehand and to be able to accurately quantify (either by means of goals or weights) his/her preferences. In the interactive methods phases of dialogue with the decision maker are interchanged with phases of calculation and the process usually converges, after a few iterations, to the most preferred solution. The decision maker progressively drives the search with his answers towards the most preferred solution. The drawback is that he/she never sees the whole picture (the Pareto set) or an approximation of it. Hence, the most preferred solution is "most preferred" in relation to what he/she has seen and compare so far. In the a posteriori methods the efficient solutions of the problem (all of them or a sufficient representation) are generated and then the decision maker involves, in order to select among them, the most preferred one.

The generation methods are the less popular due to their computational effort (the calculation of the efficient solutions is usually a time consuming process) and the lack of widely available software. However, they have some significant advantages. The solution process is divided into two independent phases: First, the generation of the efficient solutions and subsequently the involvement of the decision maker when all the information is on the table. Hence, they are favorable whenever the decision maker is hardly available and the interaction with him is difficult, because he is involved only in the second phase, having at hand all the possible alternatives (the Pareto set or an adequate representation). Besides, the fact that none of the potential solutions has been left undiscovered, reinforces the decision maker's confidence on the final decision.

For special kind of MOMP problems (mostly linear problems) of small and medium size, there are also methods that produce the entire efficient set (see e.g. [2–4]). Here, we will focus on the general case, where even relatively large MOMP problems can be addressed. In general, the most widely used, generation methods are the *weighting method* and the ε-constraint method. These methods can provide a representative subset of the Pareto set which in most cases is adequate. An essential step towards further penetration of the generation methods in MOMP applications is to provide appropriate codes for MP solvers that are widely used by people in engineering, economics, agriculture etc. In this context we propose the use of the augmented ε-constraint method (AUGMECON) which is a novel version of the conventional ε-constraint method that provides remedies for its well-known pitfalls. AUGMECON has been implemented in the widely used modeling language GAMS (General Algebraic Modeling Language, www.gams.com) and is available for the interested readers in GAMS library. Besides the generation of the Pareto optimal solutions the current work proceeds also to the support of decision maker in selecting his/her most preferred among them. For this reason, an interactive process based on AUGMECON is also developed and presented in the current paper.

The rest of the paper is organized as follows: In Section 2 the ε-constrained method is described. In Section 3 the proposed method (AUGMECON) is presented, while in Section 4 the implementation of the method in a modeling language is briefly described. In Section 5, an interactive approach based on AUGMECON is proposed in order to assist the decision maker in finding the most preferred among the Pareto optimal solutions. Finally in Section 6 the basic concluding remarks are discussed.

2. The ε-constraint method

Assume the following MOMP problem:

$$
\begin{aligned}
\max \quad & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_p(\mathbf{x})) \\
\text{st} \quad & \\
& \mathbf{x} \in S,
\end{aligned} \tag{1}
$$

where $\mathbf{x}$ is the vector of decision variables, $f_1(\mathbf{x}),\ldots,f_p(\mathbf{x})$ are the p objective functions and S is the feasible region.

In the ε-constraint method we optimize one of the objective functions using the other objective functions as constraints, incorporating them in the constraint part of the model as shown below [5,6]

$$
\begin{aligned}
\max \quad & f_1(\mathbf{x}) \\
\text{st} \quad & \\
& f_2(\mathbf{x}) \geqslant e_2, \\
& f_3(\mathbf{x}) \geqslant e_3, \\
& \ldots \\
& f_p(\mathbf{x}) \geqslant e_p, \\
& \mathbf{x} \in S.
\end{aligned} \tag{2}
$$

By parametrical variation in the RHS of the constrained objective functions (e_i) the efficient solutions of the problem are obtained.

The ε-constraint method has several advantages over the weighting method.

1. For linear problems, the weighting method is applied to the original feasible region and results to a corner solution (extreme solution), thus generating only efficient extreme solutions. On the contrary, the ε-constraint method alters the original feasible region and is able to produce non-extreme efficient solutions. As a consequence, with the weighting method we may spend a lot of runs that are redundant in the sense that there can be a lot of combination of weights that result in the same efficient extreme solution. On the other hand, with the ε-constraint we can exploit almost every run to produce a different efficient solution, thus obtaining a more rich representation of the efficient set.
2. The weighting method cannot produce unsupported efficient solutions in multi-objective integer and mixed integer programming problems, while the ε-constraint method does not suffer from this pitfall [2,3].
3. In the weighting method the scaling of the objective functions has strong influence in the obtained results. Therefore, we need to scale the objective functions to a common scale before forming the weighted sum. In the ε-constrained method this is not necessary.
4. An additional advantage of the ε-constraint method is that we can control the number of the generated efficient solutions by properly adjusting the number of grid points in each one of the objective function ranges. This is not so easy with the weighting method (see point 1 above).

In the literature, several versions of the ε-constraint method have been appeared trying to improve its performance or adapt it to a specific type of problems [7–9].

However, despite its advantages over the weighting method, the ε-constraint method has three points that need attention in its implementation: (a) the calculation of the range of the objective functions over the efficient set, (b) the guarantee of efficiency of the obtained solution and (c) the increased solution time for problems with several (more than two) objective functions. We try to address these three issues with a novel version of the ε-constraint method that is presented in the next section.

3. The augmented ε-constraint method (AUGMECON)

The augmented ε-constraint method will be illustrated using an educational example in order to better explain the differences with the conventional ε-constraint. The example comprises two variables in order to be easily depicted graphically:

Problem P

$$
\begin{aligned}
&\max \quad f1 = x1\\
&\max \quad f2 = 3x1 + 4x2\\
&\text{st}\\
&\qquad x1 <= 20,\\
&\qquad x2 = 40,\\
&\qquad 5x1 + 4x2 <= 200.
\end{aligned}
$$

The feasible region and the direction of the two objective functions are shown in Fig. 1. The Pareto set (or efficient frontier) for this problem is depicted with the heavy line (segment QR).

3.1. Lexicographic optimization for the payoff table

In order to properly apply the ε-constraint method we must have the range of every objective function, at least for the $p-1$ objective functions that will be used as constraints. The calculation of the range of the objective functions over the efficient set is not a trivial task (see e.g. [10–12]). While the best value is easily attainable as the optimal of the individual optimization, the worst value over the efficient set (nadir value) is not. The most common approach is to calculate these ranges from the payoff table (the table with the results from the individual optimization of the p objective functions). The nadir value is usually approximated with the minimum of the corresponding column (see e.g. [2,3,6]). However, even in this case, we must be sure that the obtained solutions from the individual optimization of the objective functions are indeed Pareto optimal solutions. In the presence of alternative optima the obtained by a commercial software optimal solution is not a guaranteed Pareto optimal solution. In order to overcome this ambiguity, we propose the use of lexicographic optimization for every objective function in order to construct the payoff table with only Pareto optimal solutions. A simple remedy in order to bypass the difficulty of estimating the nadir values of the objective functions is to define reservation values for the objective functions. The reservation value acts like a lower (or upper for minimization objective functions) bound. Values worse than the reservation value are not allowed.

Following the conventional ε-constraint method in our educational example, we first calculate the payoff table by simply calculating the individual optima of the objective functions. A conventional LP optimizer will produce the payoff table of Table 1.

It can be noticed that the optimal solution obtained for f_1 ($f_1 = 20, f_2 = 60$) that corresponds to point P is a dominated solution in the problem due to alternative optima (see e.g. point Q). However, it is almost sure that a conventional LP optimizer

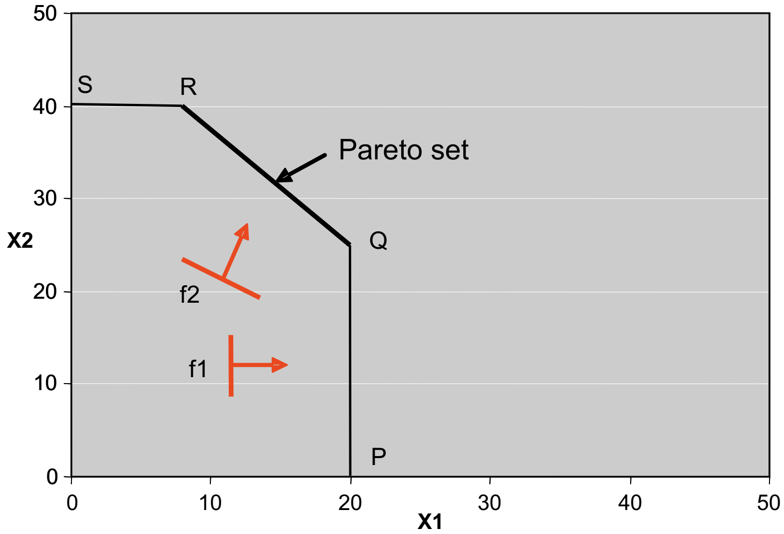

Fig. 1. Feasible region and directions of objective functions.

Table 1
Payoff table obtained by a conventional LP optimizer.

	f_1	f_2
maxf_1	20	60
maxf_2	8	184

will calculate the solution of point P first and will stop the searching giving this solution as output. In order to avoid this situation we proceed to the lexicographic optimization of the objective functions and the results are shown in Table 2.

With the lexicographic optimization we obtain as the solution that maximizes f_1 the solution that corresponds to point Q which is a Pareto optimal (non-dominated) solution.

In general, the lexicographic optimization of a series of objective functions is to optimize the first objective function and then among the possible alternative optima optimize for the second objective function and so on. Practically, the lexicographic optimization is performed as follows: we optimize the first objective function (of higher priority), obtaining max $f_1 = z_1^*$. Then we optimize the second objective function by adding the constraint $f_1 = z_1^*$ in order to keep the optimal solution of the first optimization. Assume that we obtain max $f_2 = z_2^*$. Subsequently, we optimize the third objective function by adding the constraints $f_1 = z_1^*$ and $f_2 = z_2^*$ in order to keep the previous optimal solutions and so on, until we finish with the objective functions.

After the calculation of the payoff table we divide the ranges of the objective functions to four equal intervals and we use the five grid points as the values of e_2 in the ε-constraint method. In the first case we apply the model (2) of the conventional ε-constraint method. The results are shown in Fig. 2.

The solutions that correspond to the points A, B, C, D, E are the output of the method. In fact only point E is efficient while the other 4 are weakly efficient point (dominated by point Q). On the other hand, if we use the payoff table from the lexicographic optimization (Table 2), the results of the ε-constraint method are more meaningful. They provide a much more dense representation of the efficient set (see Fig. 3). We can see that points A′, B′, C′, D′ and E′ are all efficient points that adequately describe the efficient set.

3.2. Guaranteed efficiency of the obtained solutions

The second point of attention is that the optimal solution of problem (2) is guaranteed to be an efficient solution only if all the $(p-1)$ objective functions' constraints are binding [3,13]. Otherwise, if there are alternative optima (that may improve

Table 2
Payoff table obtained by the lexicographic optimization of the objective functions.

	f_1	f_2
maxf_1	20	160
maxf_2	8	184

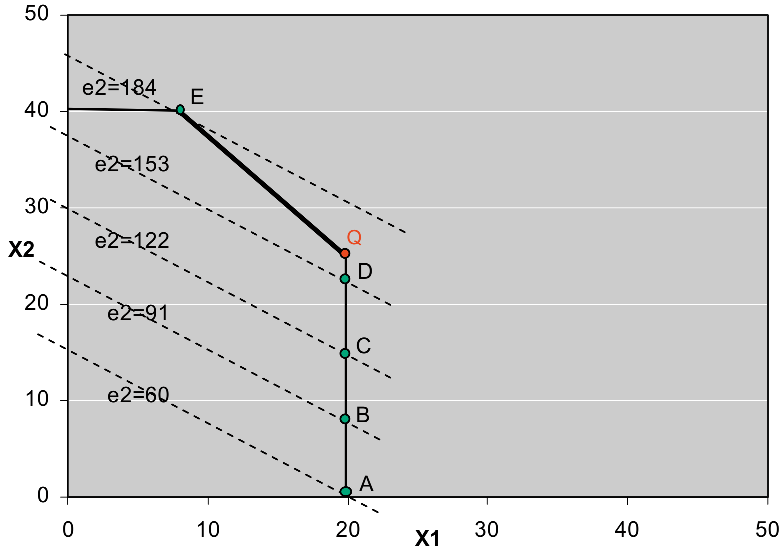

Fig. 2. Results of the conventional ε-constraint method.

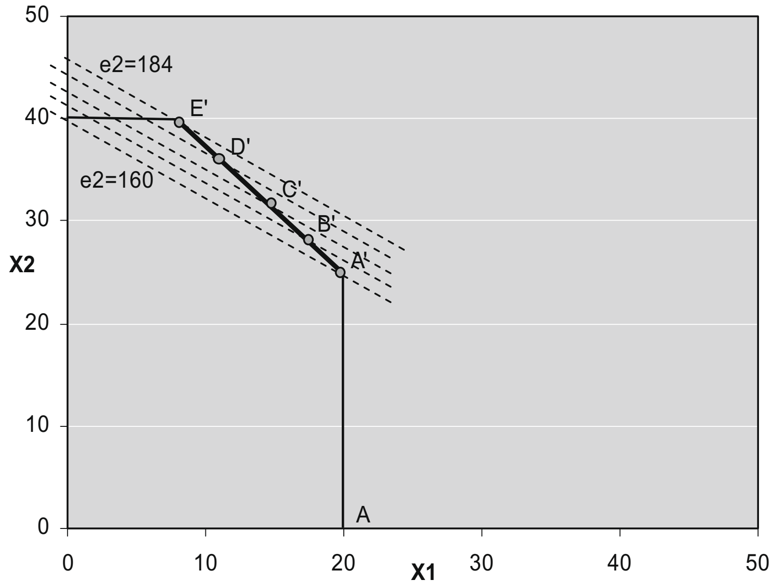

Fig. 3. Results of the ε-constraint method from model using lex-optimization in the payoff table.

one of the non binding constraints that correspond to an objective function), the obtained optimal solution of problem (2) is not in fact efficient, but it is a *weakly* efficient solution. In order to overcome this ambiguity we propose the transformation of the objective function constraints to equalities by explicitly incorporating the appropriate slack or surplus variables. In the same time, these slack or surplus variables are used as a second term (with lower priority in a lexicographic manner) in the objective function, forcing the program to produce only efficient solutions. The new problem becomes:

$$
\begin{aligned}
&\max \quad (f_1(\mathbf{x}) + eps \times (s_2 + s_3 + \cdots + s_p))\\
&\text{st}\\
&\qquad f_2(\mathbf{x}) - s_2 = e_2,\\
&\qquad f_3(\mathbf{x}) - s_3 = e_3,\\
&\qquad \ldots\\
&\qquad f_p(\mathbf{x}) - s_p = e_p,\\
&\qquad \mathbf{x} \in S \text{ and } s_i \in R^+,
\end{aligned}
\tag{3}
$$

where *eps* is an adequately small number (usually between 10^{-3} and 10^{-6}).

Proposition. *The above formulation (3) of the ε-constraint method produces only efficient solutions (it avoids the generation of weakly efficient solutions).*

Proof. Assume that the problem (2) has alternative optima and one of them (depicted as $\mathbf{x}'$) dominates the optimal solution (depicted as $\mathbf{x}$) obtained from problem (3). This means that the vector $(z_1, e_2 + s_2, \ldots, e_p + s_p)$ is dominated by the vector $(z_1, e_2 + s'_2, \ldots, e_p + s'_p)$ or in other words:

$$\begin{aligned} & e_2 + s_2 \leqslant e_2 + s'_2, \\ & e_3 + s_3 \leqslant e_3 + s'_3, \\ & \ldots \\ & e_p + s_p \leqslant e_p + s'_p \end{aligned} \tag{4}$$

with at least on strict inequality (note that $z_1 = \max f_1(\mathbf{x})$ is the same for the two cases as we have alternative optima in problem(2)). Taking the sum of these relations and based on the fact that there is at least one strict inequality we conclude that:

$$\sum_{i=2}^{p} s_i < \sum_{i=2}^{p} s'_i \tag{5}$$

But this contradicts the initial assumption that the optimal solution of (3) maximizes the sum of s_i. Hence, there is no solution $\mathbf{x}'$ that dominates the obtained solution $\mathbf{x}$, or, in other words the obtained solution $\mathbf{x}$ from problem (3) is efficient. □

In order to avoid any scaling problems it is recommended to replace the s_i in the second term of the objective function by s_i/r_i, where r_i is the range of the ith objective function (as calculated from the payoff table). Thus, the objective function of the ε-constraint method becomes:

$$\max(f_1(\mathbf{x}) + eps \times (s_2/r_2 + s_3/r_3 + \ldots + s_p/r_p)). \tag{6}$$

The proposed version of the ε-constraint method that corresponds to model (3) with the objective function (6) will be called hereafter augmented ε-constraint method or AUGMECON method.

It must be noted that in the educational example, even in the case where we don't use lexicographic optimization for the payoff table (Fig. 2), the application of model (3) will provide only E and Q (4 times) as the Pareto optimal solutions, avoiding the weakly Pareto optimal solutions A, B, C and D. It means that, although there was a dominated solution in the payoff table, the augmented ε-constraint method produces the correct results due to the corrective use of the maximization (in a second level) of the surplus variables in the objective function.

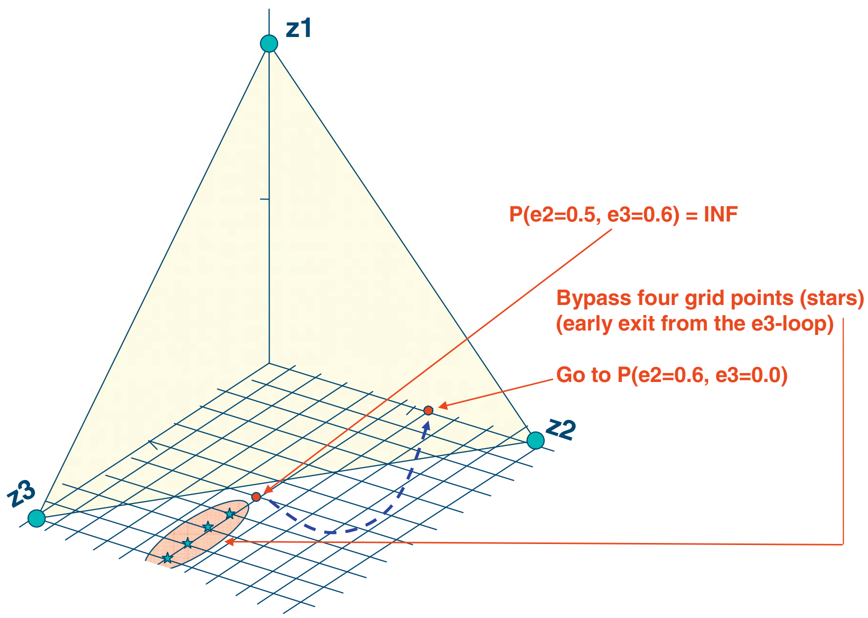

Fig. 4. Graphical representation of the Pareto front of problem (7) and the AUGMECON's feature of the early exit from the nested loops.

3.3. Acceleration of the algorithm with early exit from the loops

An innovative addition to the algorithm is the early exit from the nested loop when the problem becomes infeasible, an addition that significantly accelerates the algorithm in the case of several (more than three) objective functions. The algorithm starts with the more relaxed version of the constrained objective functions and gradually restricts the bounds. This means that for maximization objective functions, it starts from the minimum and gradually increases the RHS of the respective constraint (the opposite for minimization objective functions). In this process, when the problem becomes infeasible, it means that there is no need to further restrict the corresponding objective function, as it will result in also infeasible solutions. Therefore, the algorithm exits from the innermost loop and proceeds with the next waiting grid point of the previous objective function that corresponds to the outer loop.

The acceleration of the algorithm is shown in Fig. 4 with a graphical example. Assume the following simple MOLP with three objective functions:

$$\begin{aligned} &\max \quad z_1 = x_1 \\ &\max \quad z_2 = x_2 \\ &\max \quad z_3 = x_3 \\ &\text{st} \\ &\qquad x_1 + x_2 + x_3 <= 1. \end{aligned} \tag{7}$$

The above MOLP problem is transformed to the ε-constraint equivalent which is a parametric problem on e_2 and e_3 that provides the Pareto optimal solutions by varying e_2 and e_3:

$$\begin{aligned} &\max \quad z_1 = x_1 \\ &\text{st} \\ &\qquad x_2 >= e_2, \\ &\qquad x_3 >= e_3, \\ &\qquad x_1 + x_2 + x_3 <= 1. \end{aligned} \tag{8}$$

In Fig. 4 the Pareto front and the steps of the ε-constraint method are depicted. The range of each one of the two objective functions is divided in 10 intervals (11 grid points). In the course of the method, when e_2 = 0.5 and e_3 = 0.6 the problem becomes infeasible. In this case there is no need to check for e_3 = 0.7, e_3 = 0.8, etc. (marked with star in Fig. 4). So the algorithm exits from the e_3 loop and directly goes to e_2 = 0.6 and e_3 = 0.

The early exit saves a lot of computational time in problems with more than 2–3 objective functions. This technique is especially beneficial when there are several objective functions in the problem. In a real application problem (described in [14]) with six objective functions, 236 binary variables and 92 constraints, the early exit from the loops reduce the computation time by 45%. Namely, the computation time for five grid points per objective function (5^5 = 3125 optimization problems to be solved) was approximately 21 min in a Pentium M 1.7 GHz computer. Using the early exit technique the computation time reduced to 11 min and 30 s by just avoiding the redundant iterations that lead to infeasible solutions, solving actually only 1705 optimization problems.

4. Implementation

Practically, the ε-constraint method is applied as follows: From the payoff table we obtain the range of each one of the $p-1$ objective functions that are going to be used as constraints. Then we divide the range of the ith objective function to q_i equal intervals using $(q_i - 1)$ intermediate equidistant grid points. Thus we have in total $(q_i + 1)$ grid points that are used to vary parametrically the RHS (e_i) of the ith objective function. The total number of runs becomes $(q_2 + 1) \times (q_3 + 1) \times \cdots \times (q_p + 1)$. A desirable characteristic of the ε-constraint method is that we can control the density of the efficient set representation by properly assigning the values to the q_i. The higher the number of grid points the more dense is the representation of the efficient set but with the cost of higher computation time. A trade off between the density of the efficient set and the computation time is always advisable. The flowchart of the algorithm is shown in Fig. 5.

The AUGMECON method has been coded in GAMS, a widely used modeling language [15]. The code is available in the GAMS library (http://www.gams.com/modlib/libhtml/epscm.htm) with an educational example and some supporting documentation. It is an effort to provide the multi-criteria community with access to very powerful optimization tools and, vice versa, to provide the many GAMS users with tools for effectively dealing with multi-objective optimization.

The interested reader can use AUGMECON in his/her own problems by modifying only the part of code that has to do with the example (the specific objective functions and constraints), as well as the parameters of AUGMECON (number of grid points per objective function). The part of the code that performs the calculation of the payoff table with lexicographic optimization and the production of the Pareto optimal solutions is fully parameterized in order to be ready to use. The GAMS version of AUGMECON can be used in multi-objective linear programming (LP), mixed integer programming (MIP) or even nonlinear programming (NLP) problems (given that the necessary solvers are installed).

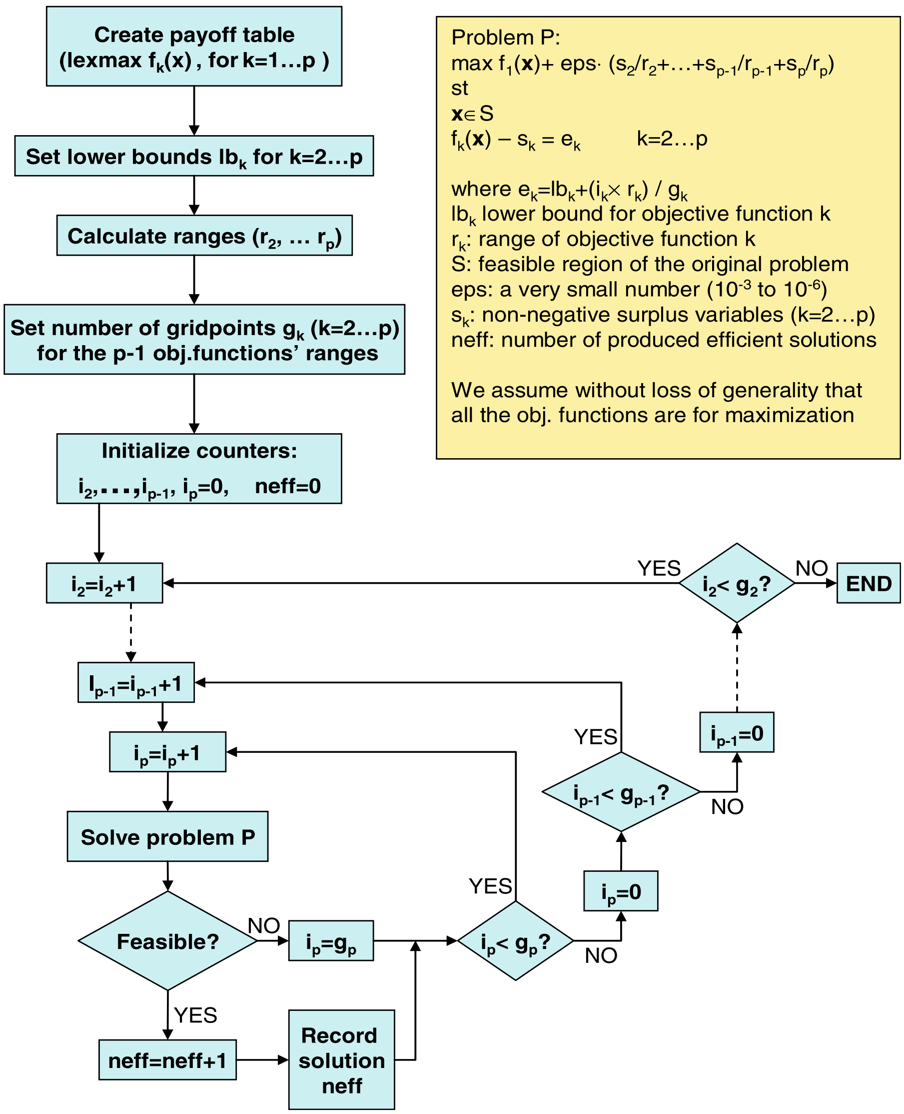

Fig. 5. Flowchart of the AUGMECON method.

5. Use of AUGMECON in an interactive context

In the absence of any other information, none of the generated Pareto optimal solutions can be said to be better than the other. Usually a decision maker is needed to provide additional preference information and to identify the "most preferred" solution. Multi-criteria optimization thus has to combine two aspects: optimization and decision support. Consequently, whenever a single solution-decision is pursued, the next step, beyond the generation of the Pareto optimal solutions, is to assist the decision maker in selecting his/her most preferred solution among them. One option is to consider the Pareto optimal vectors as discrete alternatives in a Multiple Criteria Decision Making (MCDM) problem and use one of the available methods. However, the description of such MCDM methods is task is beyond the scope of the current paper. The interested reader can find more information in [16] and [17] among others, that extensively cover the current trends in the field.

In this paper, we suggest an interactive approach that is based on AUGMECON and can help the decision maker to identify his/her most preferred Pareto optimal solution after a number of iterations. As in all interactive procedures, phases of computation will be followed by decision phases and vice versa. The decision maker's decisions will guide the search towards the most preferred Pareto optimal solution after a number of iterations. The proposed interactive process works as follows (assume, without loss of generality, that all the objective functions have to be maximized): The AUGMECON is applied in the specific multi-objective problem using a coarse grid. The derived Pareto optimal solutions are filtered down to 5, using the forward filtering process that is described in [2, p. 314]. These five solutions, which are representative of the Pareto set, are shown to the decision maker. The decision maker selects his/her most preferred. After the selection, the method exploits this information, in order to prune the search space around the selected solution. This is achieved by imposing lower bounds to the objective functions (for the next Pareto solution generation), calculated as:

$$LB_j^{(i+1)} = z_{*j}^{(i)} - a^i(z_{*j}^{(i)} - z_j^{min}) \tag{9}$$

where $LB_j^{(i+1)}$ is the lower bound of the jth objective function, in the next $(i+1)$ iteration, $z_{*j}^{(i)}$ is the jth element of the selected Pareto optimal solution's criterion vector in the ith iteration, z_j^{min} is the minimum of the jth objective function over the entire Pareto set as obtained from the initial payoff table and, finally, a is the contraction parameter that takes values in [0,1] and controls the rate of search space contraction. The higher the a, the slower the contraction of the search space, i.e. more iterations needed to converge to the most preferred solution. It must be noted that if the decision maker wants to select two solutions in the ith iteration, say solution A and B, then the lower bound for the next iteration is calculated as:

$$LB_j^{(i+1)} = \min\{z_{Aj}^{(i)} - a^i(z_{Aj}^{(i)} - z_j^{min}), \quad z_{Bj}^{(i)} - a^i(z_{Bj}^{(i)} - z_j^{min})\} \tag{10}$$

In this way the danger of excluding interesting areas is reduced. The method may continue until the decision maker is satisfied with the obtained Pareto optimal solution. The flowchart of the algorithm is illustrated in Fig. 6.

The method is inspired by the Interactive Weighted Tchebycheff method of Steuer [2, p. 419]. However it differs in the following points:

1. The generation of the Pareto optimal solutions is performed using AUGMECON and not the augmented weighted Tchebycheff method.
2. The contraction of the search space around the selection of the decision maker in each iteration is performed directly in the criteria space and not in the weight space.

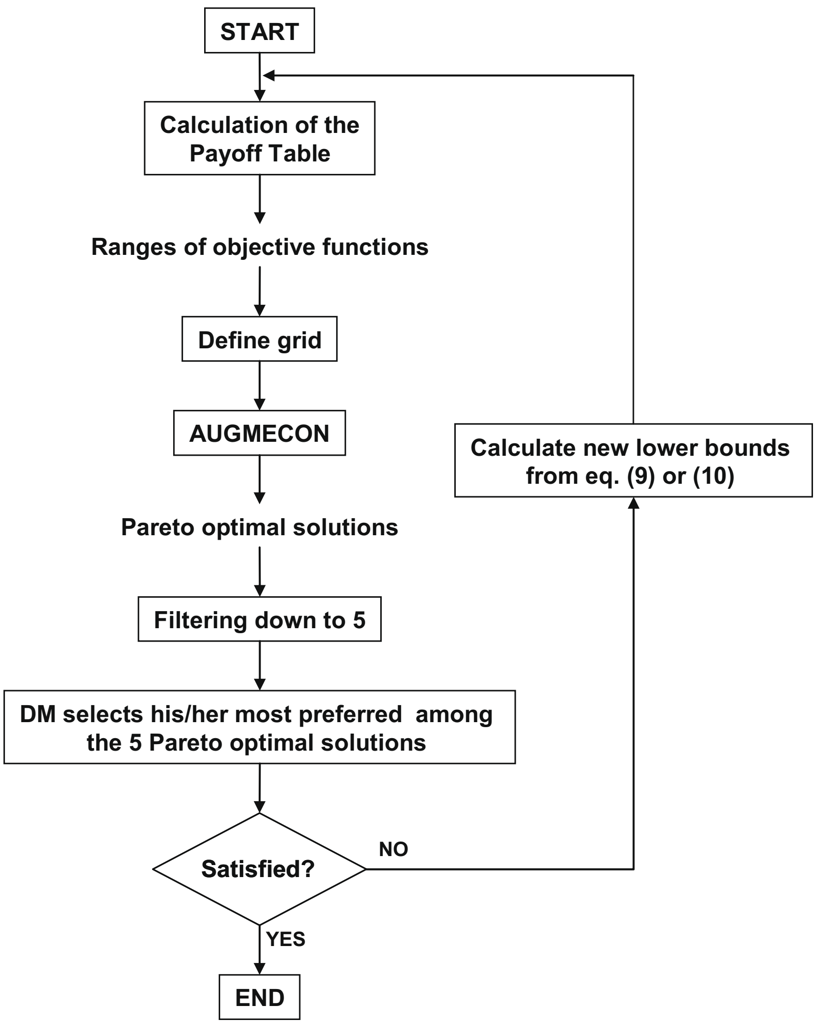

Fig. 6. Flowchart of the interactive process based on AUGMECON.

3. In each iteration the decision maker may select more than one Pareto optimal solutions.
4. It is not necessary to define the number of iterations a priori. The decision maker may continue the process until she/he is satisfied with the obtained solution.

It must also be noted that the grid for the objective functions in the AUGMECON does not need to be constant throughout the whole process, but may be adjusted. In this way the decision maker may start with a coarse grid in the early iterations, in order to cover quickly the whole Pareto front. Subsequently, in the late iterations, when the decision maker is making fine tuning, he/she can accordingly use a denser grid for investigating more thoroughly the interesting areas. This option is

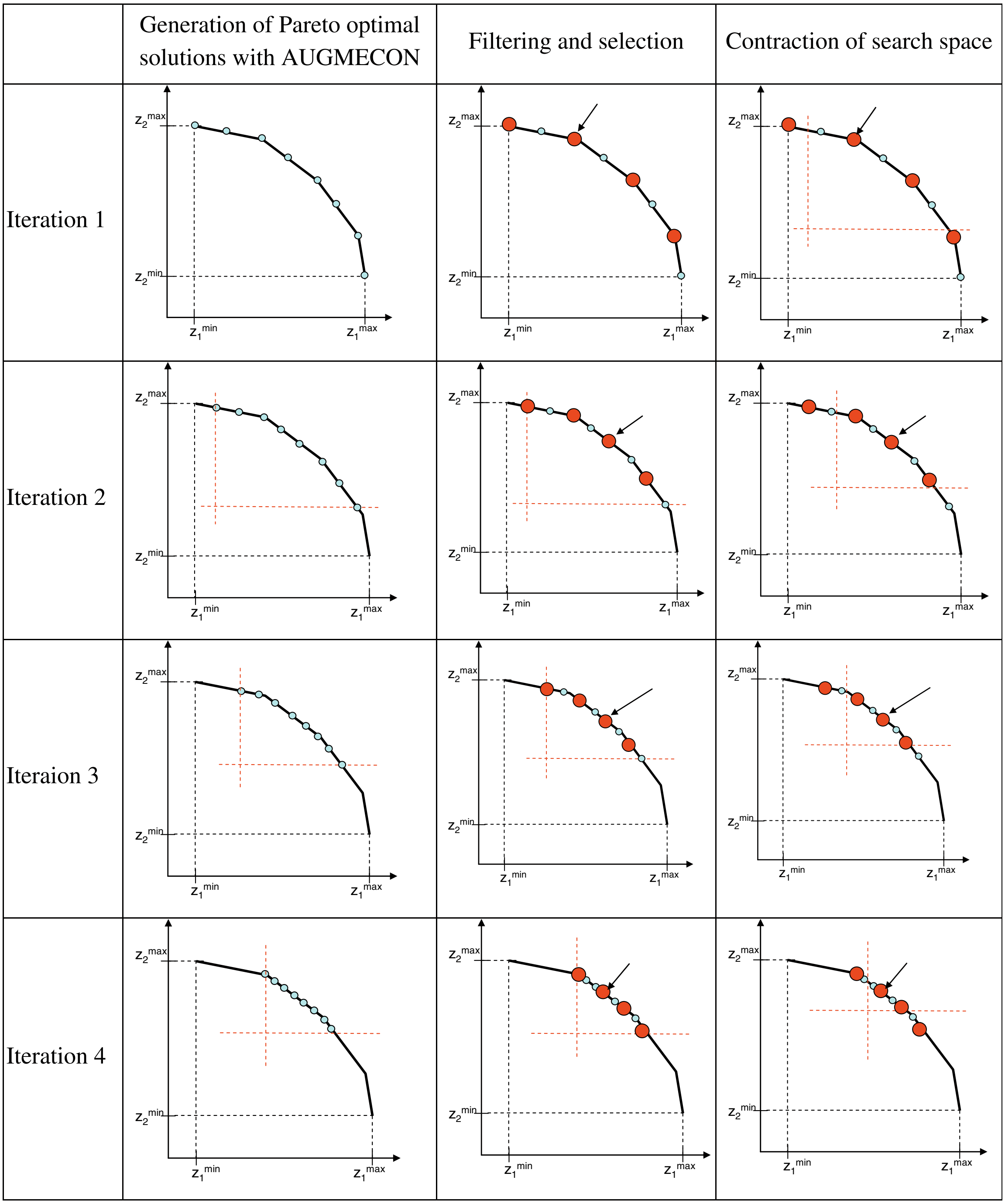

Fig. 7. Graphical representation of the interactive use of AUGMECON.

extremely useful whenever the decision and criteria space are discontinuous (i.e. Integer or Mixed Integer Programming problems). The whole process is depicted graphically in the following figure using a bi-objective maximization problem in Fig. 7.

In the graphical example, the contraction parameter α is set to 0.7, while the representative set that is shown to the decision maker comprises four Pareto optimal solutions. The small cyan circles are the Pareto optimal solutions as obtained by AUGMECON and the big red circles are the Pareto optimal solutions obtained from the forward filtering process (representative set). The changing dashed lines are the new lower bounds, updated at the final column of each iteration. The first four iteration are shown in the chart. The decision maker may continue further until he is satisfied with the obtained solution, or until the lower bounds to the objective functions eventually determine the final solution. It must be noted that the lower bounds in every iteration are saved, so that, it is possible for the decision maker to backtrack to a previous iteration and reconsider his/her choice.

6. Concluding remarks

In the present text we propose an effective implementation of the ε-constraint method (the AUGMECON method) which is already implemented using commercial software (GAMS) and is available for the interested reader. Special care is taken in order to secure the efficiency of the obtained solutions by using the augmented ε-constraint method. The code can be easily adapted to the needs of the user (number and direction of the objective functions, density of the efficient set representation, reservation values for the objective functions). We also incorporate some acceleration issues (early exit from the loops) which are particularly useful when there are several objective functions. Due to its characteristics, the AUGMECON method can be successfully applied to multi-objective problems with continuous and discrete variables.

AUGMECON can effectively be used also as a part of an interactive procedure that gradually guides the decision maker to his/her most preferred Pareto optimal solution, through an iterative process where phases of computation and dialogue are interchanged. In the present situation the interactive features of AUGMECON are performed manually, while a more user-friendly version is under development for using in combination with the GAMS code.

References

[1] C.L. Hwang, A. Masud, in: Multiple Objective Decision Making. Methods and Applications: A State of the Art Survey, Lecture Notes in Economics and Mathematical Systems, vol. 164, Springer-Verlag, Berlin, 1979.

[2] R.E. Steuer, Multiple Criteria Optimization, Theory, Computation and Application, Krieger, Malabar, FL, 1986.

[3] K.M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston, 1998.

[4] G. Mavrotas, D. Diakoulaki, Multi-criteria branch and bound: a vector maximization algorithm for Mixed 0–1 Multiple Objective Linear Programming, Applied Mathematics and Computation 171 (2005) 53–71.

[5] V. Chankong, Y.Y. Haimes, Multiobjective Decision Making: Theory and Methodology, North-Holland, New York, 1983.

[6] J.L. Cohon, Multiobjective Programming and Planning, Academic Press, New York, 1978.

[7] M. Ehrgott, D.M. Ryan, Constructing robust crew schedules with bicirteria optimization, Journal of Multi-Criteria Decision Analysis 11 (2002) 139–150.

[8] M. Laumanns, L. Thiele, E. Zitzler, An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, European Journal of Operational Research 169 (2006) 932–942.

[9] H.W. Hamacher, C.R. Pedersen, S. Ruzika, Finding representative systems for discrete bicriterion optimization problems, Operations Research Letters 35 (2007) 336–344.

[10] G.R. Reeves, R.C. Reid, Minimum values over the efficient set in multiple objective decision making, European Journal of Operational Research 36 (1988) 334–338.

[11] R.E. Steuer, Non-fully resolved questions about the efficient/nondominated set, in: J. Climaco (Ed.), Multicriteria Analysis, Springer, 1997, pp. 585–589.

[12] H. Isermann, R.E. Steuer, Computational experience concerning payoff tables and minimum criterion values over the efficient set, European Journal of Operational Research 33 (1987) 91–97.

[13] M. Ehrgott, M. Wiecek, Multiobjective programming, in: J. Figueira, S. Greco, M. Ehrgott (Eds.), Multiple Criteria Decision Analysis. State of the Art Surveys, Springer, 2005, pp. 667–722.

[14] G. Mavrotas, E. Georgopoulou, S. Mirasgedis, Y. Sarafidis, D. Lalas, V. Hontou, N. Gakis, An integrated approach for the selection of Best Available Techniques (BAT) for the industries in the greater Athens area using multi-objective combinatorial optimization, Energy Economics 29 (2007) 953–973.

[15] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, GAMS. A User's Guide, GAMS Development Corporation, Washington, 1998.

[16] V. Belton, T. Stewart, Multiple Criteria Decision Analysis. An Integrated Approach, Kluwer Academic Publishers, 2002.

[17] J. Figueira, S. Greco, M. Ehrgott, Multiple Criteria Decision Analysis. State of the Art Surveys, Springer, 2005.