FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 27, No 1 (2012), 55–66

COMPENDIOUS LEXICOGRAPHIC METHOD FOR MULTI-OBJECTIVE OPTIMIZATION

Ivan P. Stanimirović*

Abstract. A modification of the standard lexicographic method, used for linear multi-objective optimization problems, is presented. An algorithm for solving these kind of problems is developed, for the cases of two and three unknowns. The algorithm uses the general idea of indicating the lexicographic order to objective functions, combined with the graphical method of linear programming. Implementation details and some illustrative examples are provided by means of symbolic processing of the package `MATHEMATICA`. Also, some comparative processing times showed that our method was highly efficient on every 2D and 3D linear optimization problem.

1. Introduction

Multi-objective optimization is the problem of optimizing several objective functions that can often be conflicted [3]. Therefore, lexicographic method is a way to handle multi-objective optimization problems in general, where a pre-defined order can be established amongst the objective functions. Then a sequence of single-objective optimization problems is solved, where each objective is optimized at a single point of time. Often, the decision-maker needs to determine some preferences in order to establish the lexicographic ordering of the objectives, which can be difficult task. Some attempts have been made to dynamically change the ordering of objectives, which reflects their preferences [5], eliminating the need for the decision making.

The general multi-objective optimization problem is considered as an ordered sequence of real objective functions with the set of constrains:

$$\begin{aligned} &\text{Maximize:} \quad && \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), \dots, F_l(\mathbf{x})]^T, \quad \mathbf{x} \in \mathbb{R}^n \\ &\text{Subject to:} && g_i(\mathbf{x}) \le 0,\ i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0,\ i = 1, \dots, k. \end{aligned} \tag{1.1}$$

Received April 23, 2012.
2010 *Mathematics Subject Classification.* Primary 90C29; Secondary 90C05
*The author was supported in part by research project 174013 of the Serbian Ministry of Science

The feasible design space (often called the constraints set) of (1.1) is denoted by $\mathbf{X} \subset \mathbb{R}^n$, where the following expression holds:

$$\mathbf{X} = \{\mathbf{x}|\ g_i(\mathbf{x}) \le 0,\ i = \overline{1,m};\ h_i(\mathbf{x}) = 0,\ i = \overline{1,k}\}.$$

The feasible criterion space $\mathbf{Z}$ (also called the feasible cost space) is defined as the set

$$\mathbf{Z} = \{\mathbf{F}(\mathbf{x})|\ \mathbf{x} \in \mathbf{X}\}.$$

Feasibility implies that no constraints are violated. The term attainability means that a point in the criterion space maps to a point in the design space. Notice that each point $\mathbf{x} \in \mathbf{X}$ is mapped to a point from the criterion space. Also, the point $\mathbf{x_j}^*$ that maximizes the j-th objective function subject to constraints in (1.1), is obtained.

Pareto optimal solution is the adopted concept in multi-objective optimization with applications in economics, engineering and social sciences. Generally, Pareto optimal points are the ones where it is impossible to make one objective function better without necessarily making some else worse. For the sake of completeness we restate the definition of Pareto-optimal solution (see [2, 7, 8]).

Definition 1.1. A solution $\mathbf{x}^*$ is said to be **Pareto optimal solution** of multi-objective optimization problem (1.1) iff there does not exist another feasible solution $\mathbf{x} \in \mathbf{X}$ such that $Q_j(\mathbf{x}) \ge Q_j(\mathbf{x}^*)$ for all $j = 1, \dots, l$, and $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$ for at least one index j.

Informally, a point is Pareto optimal if there is no other point improving at least one objective function without deterioration of another function. All Pareto optimal points lie on the boundary of the feasible criterion space $\mathbf{X}$.

Here we restate the main idea of the lexicographic method. The objective functions are arranged in order of importance, either by the decision maker or by an algorithm. Then the sequence of single-objective optimization problems is solved, one problem at a time. It is assumed that the ordering of the objective functions is given by the sequence $\{Q_1(\mathbf{x})\ \cdots\ Q_l(\mathbf{x})\}$. Then the stated problem (1.1) is equivalent to the following sequence of single-objective programming problems, associated to the priority levels k, $k = 1, \dots, l$:

$$\begin{array}{lll} \text{Maximize} & F_k(\mathbf{x}) & \\ \text{Subject to:} & F_j(\mathbf{x}) \ge F_j(x_j^*), & j = 1, \dots, k-1,\ k \ge 2, \\ & \mathbf{x} \in \mathbf{X}, & x_1 \ge 0, \dots, x_n \ge 0, \end{array} \tag{1.2}$$

where $F_j(x_j^*)$, $j = \overline{1, k-1}$, represent optimal values for the previously stated problems on the priority levels $j = \overline{1, k-1}$, $k \ge 2$.

It is obvious that after each solved single-objective optimization problem, the set of constraints is enlarged. This drastically increases the processing time of the algorithm. Also, the implementation of the lexicographic algorithm is of great importance and can greatly influence processor times (see [10, 11]).

Our motivation is to reduce the constraints set in each step of the lexicographic method. Therefore, the graphical method of linear programming is applied in each step, where the output is used as the constrains set in the next step. Instead of enlarging the constraints set by one inequality in each step, it can be reduced to the set of optimal points generated by the graphical method. Since the majority of MOO problems consists of maximally three unknowns and implies linear objective functions, a modification of the lexicographic method based on graphical method will be very applicable and effective. Also, since other methods for solving MOO problems, such as weighted sum method, have some drawbacks ([4, 8]), our method can be wealthy option for the specific case of problems.

We introduce the modification of standard lexicographic method, which uses the graphical method for solving each single-objective programming problem. The paper is organized as follows. In the second section a relation with linear programming problem is discussed, where some necessary results are repeated and generalized. The third section is organized to give main algorithms and theorems proving that our algorithms generate only Pareto optimal points. The efficiency of introduced method is reported through several illustrative examples. Finally, in appendix some implementation details are provided in order to produce better processing times as well as fine graphical representation of the results.

2. A relation to the linear programming problem

Relations between the linear programming and single and multi-objective optimizations have been well observed. Some known results were provided in [6], which we mention here for the sake of completeness.

Firstly, let us observe the basic linear programming problem in the space $\mathbb{R}^n$, considering only the inequality constraints. Without loss of generality, we will consider only the maximization problem:

$$\begin{aligned} &\text{Maximize:} && f(\mathbf{x}) = f(x_1,\ldots,x_n) = c_1x_1 + \ldots + c_nx_n \\ &\text{Subject to:} && \sum_{j=1}^{n} a_{ij}x_j \geq b_i,\ i = 1,\ldots,m, \end{aligned} \tag{2.1}$$

where the constants $a_{ij}, b_i, c_j \in \mathbb{R}$, $i = 1,\ldots,m$, $j = 1,\ldots,n$ are given. Let us mention some well-known results used for the graphical method.

An arbitrary solution $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of the system of inequalities given above is called a *feasible solution*. In geometric interpretation, this is a point of n-dimensional space $\mathbb{R}^n$. By $\Omega \subset \mathbb{R}^n$ we denote the set of all feasible solutions of the problem (2.1). Here we assume that $\Omega \neq \emptyset$. Thus, the following lemma was proven to be valid.

Lemma 2.1. *The boundary set* $\Omega_P = \{\mathbf{x} | \sum_{j=1}^{n} a_{ij}x_j = b_i,\ i = 1,\ldots,m\}$ *of the feasible solutions set* Ω *is the convex set.*

The *optimal point* $x^* = (x_1^*, \ldots, x_n^*) \in \Omega$ of the linear programming problem (2.1) is the feasible solution which maximizes the objective function $f(x_1, \ldots, x_n)$.

If the restricting conditions are given in the form of inequalities, each of the corresponding straight lines divides the plane to two areas. The permissible conditions are located in the sub-space Ω representing the region of feasible solutions. The following lemma is well-known from the literature.

Lemma 2.2. *The optimal value of the objective function f in (2.1) is gained in one of the extreme points of the set of feasible solutions Ω.*

The significance of this lemma is the reduction of the set of possible optimal points from Ω (infinite in general case) to the finite set of extreme points. For a linear programming problem with a bounded, feasible region defined by m linearly independent constraints expressed by m linear equations, and with n variables, a goal is to find one of the $\frac{n!}{m!(n-m)!}$ basic solutions which maximizes the goal function (linear in this case). Here we generalize some known results with the following theorem.

Theorem 2.1. *If two optimal solutions of the problem (2.1) are gained, then each point on the line segment between these two points represents the optimal solution of (2.1), i.e. if $x^{(1)}, x^{(2)} \in \Omega_P$ are two optimal solutions, then each point x_λ of the form*

$$x_\lambda = \lambda x^{(1)} + (1-\lambda)x^{(2)}, \quad 0 \le \lambda \le 1$$

is the optimal solution of the problem (2.1).

Proof. Since $x^{(1)}$ and $x^{(2)}$ are the optimal points of the problem (2.1), considering the objective function f, it is obviously satisfied that

$$f(x^{(1)}) \geqslant f(x) \ \wedge \ f(x^{(2)}) \geqslant f(x), \quad \forall x \in \Omega.$$

Therefore, for a point $x_\lambda = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ the following inequality is valid:

$$\begin{aligned} f(x_\lambda) &= f(\lambda x^{(1)} + (1-\lambda)x^{(2)}) = \lambda f(x^{(1)}) + (1-\lambda) f(x^{(2)}) \\ &\ge \lambda f(x) + (1-\lambda) f(x) \\ &= f(x), \quad \forall x \in \Omega_P. \end{aligned}$$

Then, each point x_λ, $0 \le \lambda \le 1$, maximizes the objective function f, representing the optimal solution of the problem (2.1). $\square$

According to this statement we conclude that after the first step, the area of feasible solutions will be a segment or a point. If the solutions are graphically represented in the Gaussian coordinate system, a convex polygon is generated. Amongst the vertices of this polygon are the possible optimal solutions of our MOO problem.

3. Main results

Let us introduce the modification of lexicographic method for solving the linear MOO problem for the case of two or three unknowns. In spite of the classical lexicographic method, in each single step, the feasible design space is reduced to the set of optimal solutions gained in previous step, by implementing the graphical method.

Now we can introduce the modification of the lexicographic method procedure. The input parameters are the list of objective functions f, ordered by importance and the list of constraints g.

Algorithm 3..1 Solving the multi-objective optimization problem (1.1), in the case of two unknowns x_1, x_2.
Compendious Lexicographic Method 2D

Require: List of ordered objective functions f and the list of constraints g.

1: Set index $i = 1$ and the set $R = \emptyset$. Use the first function f_1 as the objective function and the list g as the set of constraints.
2: Solve the single-objective optimization problem for the given objective function and the set of constraints, using the graphical method. Store the solution in the set R.
3: If the solution from R is a line segment, represented by $\lambda x^{(1)} + (1-\lambda)x^{(2)}$, where $0 \leq \lambda \leq 1$, go to Step 4. Vice versa, if the solution R is a single point, go to Step 5.
4: If $i < l$ perform the following: increase index i by 1, use f_i as the objective function and the set R as the constraints set, and go to Step 2. In the case of $i = l$, go to Step 5.
5: **return** the set R as the solution of the problem.

We conclude that the solution of multi-objective optimization problem using the Algorithm 3..1 (Step 5) is provided as a line segment or a single point. After each single-objective optimization problem solving in Step 2, the best solution (which maximizes the objective function) is used. So, the following theorem can easily be proven.

Theorem 3.1. *Consider the multi-objective optimization problem* (1.1)*, where* $\mathbf{x} \in \mathbb{R}^2$*. For the list of ordered objective functions* f *and the list of constraints* g*, the solution gained by the Algorithm 3..1 is Pareto optimal point of the observed MOO problem* (1.1)*.*

Proof. Firstly, observe that by applying the algorithm we get the array of solutions of maximally l single-objective optimization problems. Denote these sets of solutions by R_i $i \leq l$, in the i-th step. For each index $1 \leq i \leq l$, there are two possibilities:

1) the set R_i consists of only one point.

2) the set R_i consists of a line segment.

Observe that $f_k(x^{(l)}) \geq f_k(x^{(k)})$ is satisfied for each $k = \overline{1, l-1}$. Suppose contrary, that the solution $x^{(l)}$ is not a Pareto optimal point. Then there exists another point x^*, such that

$$f_k(x^*) \geq f_k(x^{(l)}),\ \forall k \in \{1, \ldots, l\},$$

where at least one inequality is strong. Suppose that, for example, for some index t the inequality $f_t(x^*) > f_t(x^{(l)})$ holds. Whereas $f_t(x^{(l)}) \geq f_t(x^{(t)})$ holds, the inequality $f_t(x^*) > f_t(x^{(t)})$ is satisfied. This implies that $x^* \notin \Omega'_P$, where

$$\Omega'_P = \Omega_P \cap \{x \mid f_1(x) \geq f_1(x^{(1)}), \ldots, f_{t-1}(x) \geq f_{t-1}(x^{(t-1)}).$$

Therefore, the point $x^* \in \Omega_P \setminus \Omega_{Pt}$, which implies that for some index $i \in \{1, \ldots, k-1\}$ the inequality $f_i(x^*) < f_i(x^{(i)})$ holds. This is contradictory to the primordial statement that $f_k(x^*) \geq f_k(x^{(k)})$, for each $k = 1, \ldots, l$. Obviously, the solution $x^{(l)}$ must be Pareto optimal. $\square$

Since the graphical method is considered for the case of two and three variables, the analogous algorithm to the Algorithm 3..1 can be stated, for the case of three unknowns.

Algorithm 3..2 Solving the multi-objective optimization problem (1.1), in the case of three unknowns x_1, x_2, x_3.

Compendious Lexicographic Method 3D

Require: List of ordered objective functions f and the list of constraints g.

1: Set index $i = 1$ and the set $R = \emptyset$. Use the first function f_1 as the objective function and the list g as the set of constraints.

2: Solve the single-objective optimization problem for the given objective function and the set of constraints, using the graphical method. Memorize this solution in the set R.

3: If the solution set R consists of a single point, go to Step 5. Vice versa, if the solution set R is represented by $\lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)}$, where $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1,\ \lambda_1 + \lambda_2 + \lambda_3 = 1$, and at least two coefficients are greater than zero, go to Step 4.

4: If $i < l$ perform the following: increase index i by 1, use f_i as the objective function and the set R as the constraints set, and go to Step 2. In the case of $i = l$, go to Step 5.

5: **return** the set R as the solution of the problem.

Theorem 3.2. *Consider the multi-objective optimization problem* (1.1)*, where* $\mathbf{x} \in \mathbb{R}^3$*. For the list of ordered objective functions* f *and the list of constraints* g*, the solution gained by the Algorithm 3..1 is Pareto optimal point of the observed MOO problem* (1.1)*.*

Proof. The proof is similar to the proof of the Theorem 3.1, with the difference that for each index $1 \leq i \leq l$, the solutions set R_i can consist of only one point, a line segment or a plane segment. It is represented by a linear combination of three optimal points, i.e. by $\lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)}$, where $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$. $\square$

4. Numerical examples

When the objective function is of two or three variables, a graphical procedure for solving the linear programming problems can be applied [9]. The optimal solution is found by drawing the graphic of the modified objective function $f(x_1, x_2) = 0$ and parallel shifting of this line in the direction of the gradient vector.

If two different solutions are found, then in accordance to Theorem 2.1 all points on the line segment between these two points are optimal solutions. Therefore, the modification of lexicographic method does not limit at finding only one Pareto optimal solution.

Example 4.1. Solve the following multi-objective optimization problem:

$$
\begin{aligned}
\text{Maximize :} \quad & [8x + 12y, 14x + 10y, x + y] \\
\text{Such that :} \quad & 2x + y \leq 150 \\
& 2x + 3y \leq 300 \\
& 4x + 3y \leq 360 \\
& x + 2y \geq 120 \\
& x, y \geq 0
\end{aligned}
$$

using the lexicographic order as given in the constraints set.

Firstly we observe the first function by lexicographic order: $f_1 = 8x+12y$. By applying the graphical method we get that each point on the line segment $\lambda\{30, 80\}+(1-\lambda)\{0, 100\}$, $0 \leq \lambda \leq 1$, maximizes the objective function f_1. This result is depicted on the left picture of the Figure 4.1.

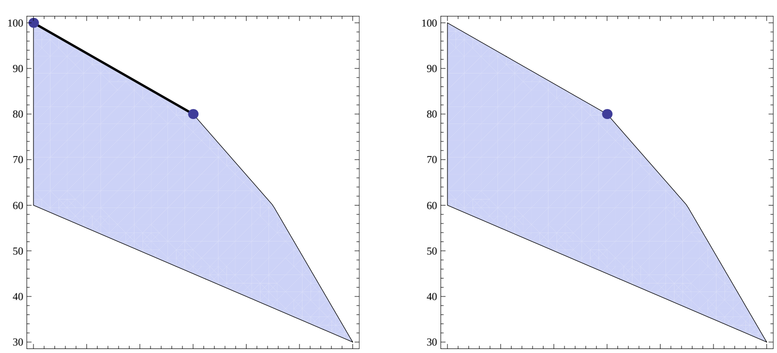

FIG. 4.1: The results on the first and the second iteration of Compendious lexicographic algorithm

After that, we search for points belonging to the given line segment, which maximize the second objective function in the lexicographic order $f_2 = 14x + 10y$. Respecting that the line $f_2 = 0$ is not parallel to given line segment, the maximal point is between the vertices of the line segment. By testing, we conclude the maximal point is $\{30, 80\}$. The subsequent computation stops here because the solution of the single-criteria optimization problem is the single point.

So, the solution of this problem is the point $\{30, 80\}$ (see the picture on the right-hand side of the Figure 4.1), in which case the three given objective functions have the following values, respectively: $1200, 1220, 110$.

We developed a procedure `LexicModif` in programming package `MATHEMATICA` for testing and verification purposes (see Appendix for details). The following instruction can be used for solving this problem:

```
LexicModif[{8x + 12y, 14x + 10y, x + y},
2x + y<= 150, 2x + 3y<= 300, 4x + 3y<= 360, x + 2y>= 120, x>= 0, y>= 0]
```

Therefore, the following result is obtained

$$\{\{1200, 1220, 110\}, \{x \to 30, y \to 80\}\}.$$

Example 4.2. Solve the three-dimensional multi-objective optimization problem with the constraints set determined in [12]:

$$\begin{aligned} \text{Maximize :} \quad & [x - 2y - z, x + y + z] \\ \text{Such that :} \quad & -x + 2y + z \geq -3, \\ & x - y + 2z \leq 4, \\ & x \geq 0. \end{aligned}$$

We determined, by the graphical method, that each point on the line $\lambda\{1, -\frac{7}{5}, \frac{4}{5}\} + (1 - \lambda)\{0, -2, 1\}$, $0 \leq \lambda \leq 1$ maximizes the objective function $f_1 = x - 2y - z$. This result is depicted on the left picture of the Figure 4.2.

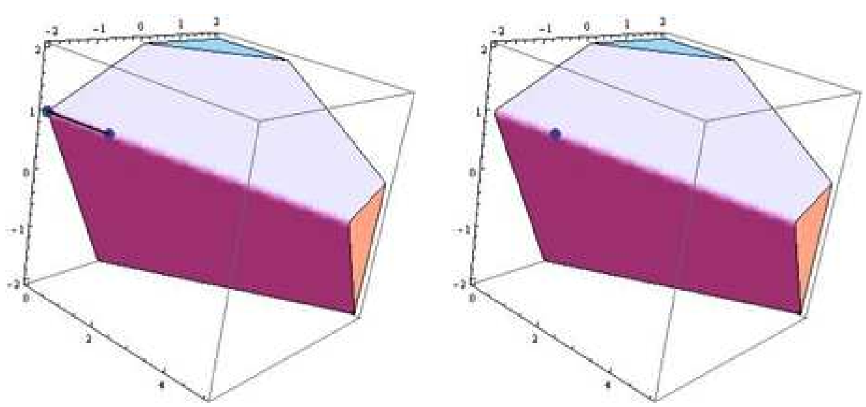

FIG. 4.2: The first two iterations of Compendious lexicographic algorithm depicted by `MATHEMATICA`

Table 4.1: Average processing times in seconds for randomly generated linear MOO problems for $n = 2$.

n	2					
l	3	10	25	100	250	1000
m	5	10	20	50	100	200
Lexicographic M.	0.1	0.4	1.0	4.8	14.0	57.3
Compend. Lexic. M.	0.1	0.2	0.4	0.9	2.2	8.5

Table 4.2: Average processing times in seconds for $n = 3$.

n	3					
l	3	10	25	100	250	1000
m	5	10	20	50	100	200
Lexicographic M.	0.2	0.8	1.9	7.8	22.4	90.0
Compend. Lexic. M.	0.2	0.5	1.1	2.3	4.9	14.2

Next, search for points in the obtained line segment, which maximize the second objective function in the lexicographic order $f_2 = x+y+z$. The following command for solving this problem is used: `LexicModif[{x-2y-z, x+y+z}, {-x+2y+z+3>=0, x-y+2z<=4, x>=0}]`

The solution of this problem is the point $\{1, -\frac{7}{5}, \frac{4}{5}\}$ (the picture on the right side of the Figure 4.2), in which case the three given objective functions have the following values, respectively: $\{3, \frac{2}{5}\}$.

Example 4.3. We have observed various random linear multi-objective optimization problems with two and three unknowns. Therefore, the coefficients in the objective functions and the ones in the inequality set are randomly generated from the set $[-1000, 1000]$. All processing times were evaluated as the average of 10 independent run times. Some of the results are depicted in Table 4.1 and Table 4.2. Notice that the number of variables is denoted by n, the number of objective functions is represented by l and that m denotes the number of inequalities of the constraints set.

Obviously, Compendious lexicographic algorithm improves the standard lexicographic algorithm for bigger MOO problems, considering $l, m > 3$. For large test examples, Compendious lexicographic algorithm is drastically better than the standard algorithm and several times more efficient. The reason for this is that Compendious lexicographic algorithm finishes the computation at some objective function (usually at the second or the third objective function) depending of the construction of the constraints set. Despite this, the basic lexicographic algorithm continues the computation until the last objective and enlarges the constraints set in each iteration (notice that at the final iteration the size of the constraints set is nearly doubled!). The benefits of using Compendious lexicographic algorithm for the mentioned class of problems are obvious, even though it applies the graphical method procedure. Its efficiency is based on the reduction of the constraints set in each iteration and the ending of the computation when the result is a single Pareto optimal solution.

5. Conclusions

A modification of the lexicographic method for solving multi-objective optimization problems is developed. Therefore, two algorithms, for the cases of two and three unknowns, are introduced in order to deal with MOO problems. The general idea of the lexicographic ordering of objective functions combined with the graphical method of linear programming produces Pareto optimal points more effectively. This is proved via theorems, where some well known results of linear programming theory are taken into consideration.

Notice that great effort is taken to present implementation details and technics to handle the introduced algorithms. Illustrative examples are provided by means of the package MATHEMATICA. Therefore, one of our goals is to demonstrate and use symbolic features of the programming language MATHEMATICA in the considered MOO problem. The lacking of the functions to deal with multi-objective optimization problems in MATHEMATICA and other symbolic-based languages is an additional motivation in this paper.

Appendix

In accordance to the discussions in the third and fourth section, the implementation of graphical method in the case of two unknowns can be written as follows. In this function formal parameters are criterion function f and the set of constraints g.

```
LinProgGraphicalM[f_, g_List] :=
 Module[{res = {}, var = Variables[f], i, j, h, gE, t, r},
  For[i = 1, i ≤ Length[g], i++, var = Union[Variables[First[g[[i]]]], var]];
  h = g /. {List → And};
  h = Reduce[h, var];
  If[h == False, Print["The problem is inadmissible"]; Break[];];
  gE = g /. {LessEqual → Equal, GreaterEqual → Equal, Less → Equal, Greater → Equal};
  For[i = 1, i ≤ Length[gE] - 1, i++,
   For[j = i + 1, j ≤ Length[gE], j++, t = FindInstance[gE[[i]] && gE[[j]] && h, var];
     If[t ≠ {}, AppendTo[res, {ReplaceAll[f, t[[1]]], t[[1]]}];];];];
  For[i = 1, i ≤ Length[res] - 1, i++,
   For[j = i + 1, j ≤ Length[res], j++, If[res[[i, 1]] < res[[j, 1]], r = res[[i]];
       res[[i]] = res[[j]];
       res[[j]] = r;];];];
  If[res[[1, 1]] == res[[2, 1]], r = {res[[1, 2]], res[[2, 2]]}, r = {res[[1, 2]]};];
  Return[r];]
```

By solving the system of inequalities given by g we obtain the region of the feasible solutions, represented with the variable h. At that point the function FindInstance (similar way of determining the points satisfying the system of inequalities is used in [1]). The extreme points are saved in the list R. These points are found in the intersection of each two straight lines gained by compilation of inequalities from the list of constraints to equalities. After that, the list of extreme points R is sorted in non-descending order depending of the distance from the line $f(x_1, x_2) = 0$. The optimal solution is the maximal element of this list.

Here we provide the implementation of the modification of lexicographic method using the graphical method for linear programming.

```
LexicModif[f_List, g_List] :=
 Module[{l = Length[f], var = Variables[f], r, q, q1, q2, i, m1, m2},
  r = LinProgGraphicalM[f[[1]], g];
  Print[r];
  q = RegionPlot[g, {var[[1]], 0, 10}, {var[[2]], 0, 10}, AspectRatio → 1];
  For[i = 2, (i ≤ l) && (Length[r] == 2), i++, m1 = ReplaceAll[f[[i]], r[[1]]];
   m2 = ReplaceAll[f[[i]], r[[2]]];
   If[m1 > m2, r = {r[[1]]}];
   If[m2 > m1, r = {r[[2]]}];];
  If[Length[r] > 1, Print["Pareto optimal solutions have the form: [Lambda]*",
    r[[1]], "+(1-[Lambda])*", r[[2]], ", 0<=[Lambda]<=1"];
   q1 = Graphics[{Thick, Line[r]}],
   Print["Pareto optimal solution is: ", r[[1]]]];
  q2 = ListPlot[r, PlotStyle → {PointSize[0.03]}, DisplayFunction → Identity];
  Return[{ReplaceAll[f, r[[1]]], r[[1]]}];]
```

Here we use the function `RegionPlot` (see [12]), in order to represent the set of feasible solutions, on which the search for possible optimal solutions are carried out. This set is after each step reduced to the set of optimal solutions from the previous problem of single-criteria problem. This idea provides the speeding-up of the program execution.

REFERENCES

1. M.A. Bhatti: *Practical optimization methods with Mathematica® applications*, Springer-Verlag, New York, 2000.
2. V. Chankong, Y. Haimes: *Multiobjective decision making: Theory and methodology Series, Volume 8*, North-Holland, New York, Amsterdam, Oxford, 1983.
3. Y. Collette, P. Siarry: *Multiobjective optimization: Principles and case studies*, Decision Engineering, Springer, 2004.
4. I. Das, J.E. Dennis: *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*, Struct. Optim. **14** (1997) 63-69.
5. J.C. Gutierrez, D.L. Silva, J.M. Perez: *Improved dynamic lexicographic ordering for Multi-objective Optimization*, Parralel Problem Solving from Nature - PPSN XI, Lectures Notes in Computer Science, **6239**, Springer, (2010) 31–40.
6. J.P. Ignizio: *Linear programming in single-multiple-objective systems*, Englewood Cliffs: Prentice Hall, 1982.
7. K. Miettien: *Nonlinear Multiobjective Optimization*, Kluver Academic Publishers, Boston, London, Dordrecht, 1999.

8. R.T. Marler: *Survey of multi-objective optimization methods for engineering*, Struct. Multidisc. Optim. **26** (2004) 369–395.
9. M. Sakaratovitch: *Linear programming*, Springer-Verlag, New York (1983).
10. P.S. Stanimirović, I.P. Stanimirović: *Implementation of polynomial multi-objective optimization in MATHEMATICA*, Struct. Multidisc. Optim. **36** (2008) 411–428.
11. P.S. Stanimirović, S. Rančić: *Symbolic implementation of lexicographic multicriteria program*, Filomat **12** (1998) 1–8.
12. S. Wolfram: *The Mathematica Book, 4th ed.*, Wolfram Media/Cambridge University Press, 1999.

Ivan P. Stanimirović
Faculty of Sciences and Mathematics
Department of Computer Science
P. O. Box 224, Visegradska 33,
18000 Niš, Serbia
ivan.stanimirovic@gmail.com