FACTA UNIVERSITATIS (NIŠ)
S . M . I . 26 (2011), 49–63

ON THE LINEAR WEIGHTED SUM METHOD FOR MULTI-OBJECTIVE OPTIMIZATION *

Ivan P. Stanimirović, Milan Lj. Zlatanović, Marko D. Petković

Abstract. A method providing the efficient way of construction of weighted coefficients for linear weighted sum method is provided. By applying this method, all of the resulting points are Pareto optimal points of the corresponding multi-objective optimization problem. A method for the efficient construction of weighting coefficients $w_i > 0$ in programming package `MATHEMATICA` is presented. Run-time symbolic transformations of the objective functions and constraints into the corresponding single-objective constrained problem are emphasized. The implementation details and the graphical representations of two and three variables case are given, in order to depict the introduced method.

1. Introduction

Pareto optimal solutions denote a concept in economics with some applications in engineering and social sciences. Informally, Pareto efficient situations are those in which it is impossible to make one person better off without necessarily making someone else worse off.

The general multi-objective optimization problem is posed as follows. We consider an ordered sequence of real objective functions with a set of constrains:

$$\begin{array}{lll} & \text{Maximize:} & Q(\mathbf{x}) = [Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})], \quad \mathbf{x} \in \mathbf{R}^n \\ (1.1) & \text{Subject to:} & f_i(\mathbf{x}) \le 0, \ i = 1, \dots, m \\ & & h_i(\mathbf{x}) = 0, \ i = 1, \dots, k. \end{array}$$

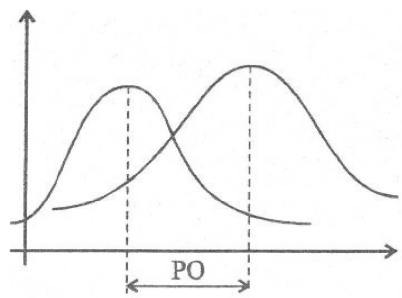
The feasible design space (often called the constraint set) in (1.1) we simply denote by $\mathbf{X}$. Therefore, the set $\mathbf{X}$ is defined by $\mathbf{X} = \{\mathbf{x} | f_i(\mathbf{x}) \le 0, \ i = 1, \dots, m; \ h_i(\mathbf{x}) = 0, \ i = 1, \dots, k\}$. In the sequel, the notation $\mathbf{x} \in \mathbf{X}$ will mean that $\mathbf{x}$ satisfies inequality and

Received March 23, 2011.
2010 *Mathematics Subject Classification.* Primary 90C29; Secondary 90C05
*The authors were supported in part by research project 174013 of the Serbian Ministry of Science.

equality constraints in (1.1). By $\mathbf{x_j}^*$ we denote the point that maximizes the jth objective function subject to constraints in (1.1).

For the sake of completeness we restate the definitions of some types of non-inferior (Pareto-optimal) solutions and ideal (utopia) point from [1], [4] and [7].

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 $Q_j(\mathbf{x}) \geq Q_j(\mathbf{x}^*)$ for all $j = 1, \ldots, l$, and $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$ for at least one index j.



F . 1.1: Representation of the region containing Pareto optimal points

All Pareto optimal points lie on the boundary of the feasible criterion space $\mathbf{X}$. Often, algorithms provide solutions that may not be Pareto optimal but may satisfy other criteria, making them significant for practical applications. For instance, weakly Pareto optimal is defined as follows:

Definition 1.2. A solution $\mathbf{x}^*$ is said to be **weakly 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 $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$ for all $j = 1, \ldots, l$.

A solution is weakly Pareto optimal if there is no other point that improves all of the objective functions simultaneously. In contrast, a point is Pareto optimal if there is no other point that improves at least one objective function without detriment to another function. It is obvious that each Pareto optimal point is weakly Pareto optimal, but weakly Pareto optimal point is not Pareto optimal.

All Pareto optimal points may be categorized as being either proper or improper. The idea of *proper Pareto optimality* and its relevance to certain algorithms is discussed in [3] and [4]. It is defined as follows:

Definition 1.3. A solution $\mathbf{x}^* \in \mathbf{X}$ is said to be **properly Pareto optimal solution** (in the sense of Geoffrion [3]) if it is Pareto optimal and there is some real number $M > 0$ such that for each $Q_j(\mathbf{x})$, $\mathbf{x} \in \mathbf{X}$ satisfying $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$ for all $j = 1, \ldots, l$, there exist at least one $Q_i(\mathbf{x})$ such that $Q_i(\mathbf{x}^*) > Q_i(\mathbf{x})$ and $\frac{Q_j(\mathbf{x}^*) - Q_j(\mathbf{x})}{Q_i(\mathbf{x}) - Q_i(\mathbf{x}^*)} \leq M$. If a Pareto optimal point is not proper called improper.

A few functions for constrained numerical optimization are available in the programming package MATHEMATICA (see [5], [10]). Functions *Maximize* and *Minimize* allow to specify an objective function to maximize or minimize, together with a set of constrains. In all cases it is assumed that the variables are constrained to have non-negative values.

`Minimize[f, {cons}, {x, y,...}]` or `Minimize[{f, cons}, {x, y,...}]`, minimize f in the region specified by the constraints *cons*;

`Maximize[f, {cons}, {x, y,...}]` or `Maximize[{f, cons}, {x, y,...}]`, find the maximum of f, in the region specified by *cons*.

`Minimize` and `Maximize` can in principle solve any polynomial programming problem in which the objective function f and the constraints *cons* involve arbitrary polynomial functions of the variables [10]. An important feature of `Minimize` and `Maximize` is that they always find global minima and maxima [10].

The main idea in the weighted sum method is to choose the weighting coefficients ω_i corresponding to objective functions $Q_i(x)$, $i = \overline{1, l}$.. So, the multi-criteria optimization problem is transformed to a single-objective one. Many authors have developed systematic approaches to selecting weights. One of difficulties with the weighted sum method is that varying the weights consistently and continuously may not necessarily result in an accurate, complete representation of the Pareto optimal set. Also, some drawbacks of minimizing weighted sums of objectives in multi-criteria optimization problems were observed in [2].

Our motivation is to develop a specific conditions for the weighted coefficients, such that each solution gained by the linear weighted sum method is Pareto optimal. A goal is to provide a practical criteria for the construction of the weighted coefficients, in order to generate the Pareto set efficiently. Therefore, we want to use the benefits of symbolic computation of MATHEMATICA to depict the generated Pareto set and the feasible design space. This will give the better insight in the multi-objective decision making process and the position of the Pareto optimal points on the boundary of the feasible solutions set.

The paper is organized as follows. In the second section we observe the properties of linear weighted sum method and provide the conditions under which the gained solution of MOO problem is Pareto optimal. Therefore, the practical method for the construction of the weighted coefficients is presented. Some implementation details for the two variables case are depicted in the third section, as well as some illustrative examples. In the last section we give the implementation details for the three variables case and observe two 3D multi-objective optimization problems, which are graphically represented via the programming package MATHEMATICA.

2. Linear weighted sum method

Let us observe the following normalized single-objective optimization problem:

$$\text{(2.1)} \qquad \text{Maximize:} \quad f(x) = \sum_{k=1}^{l} \omega_k Q_k^o(x),$$

$$\text{(2.2)} \qquad \text{Subject to:} \quad x \in X,$$

where the weights w_i, $i = 1, \ldots, l$ corresponding to objective functions satisfy the following conditions:

$$\text{(2.3)} \qquad \sum_{i=1}^{l} w_i = 1, \quad w_i \geq 0, \ i = 1, \ldots, l,$$

and $Q_k^o(x)$ is normalized k-th objective function $Q_k(x)$, $k = \overline{1, l}$.

For the case of the linear weighted sum, we consider the MOO problem (1.1) with linear objective functions, having the next form:

$$Q_i(x) = \sum_{k=1}^{l} a_{ki} x_i, \quad a_{ki} \in \mathbf{R}.$$

Therefore, normalized objective functions have the following forms:

$$Q_k^o(x) = \frac{Q_k(x)}{S_k} = \frac{a_{k1}}{S_k} x_1 + \frac{a_{k2}}{S_k} x_2 + \ldots + \frac{a_{kn}}{S_k} x_n,$$

in which case the floating-point values S_k are evaluated in the following way:

$$S_k = \sum_{j=1}^{n} |a_{kj}| \neq 0.$$

Obviously, in many practical problems, the objective functions are represented by various measure units (for exam. if Q_1 is measured in kilos, Q_2 in seconds, etc.). For this reasons the objective functions normalization is required. It's obvious that now the coefficients have values from the segment $[0, 1]$. Denote that we now have the linear programming problem

$$\text{(2.4)} \quad \text{Maximize:} \quad f(x) = \sum_{k=1}^{l} \omega_k \frac{Q_k(x)}{S_k} = \omega_1 \frac{a_{k1}}{S_k} x_1 + \frac{a_{k2}}{S_k} x_2 + \ldots + \omega_n \frac{a_{kn}}{S_k} x_n,$$

$$\text{Subject to:} \quad x \in X,$$

The following theorem gives the practical criteria for the detection of some Pareto optimal solutions of the problem (2.4).

Theorem 2.1. *The solution of the MOO problem (1.1) in the case of linear objective functions, generated by the weighted sum method (2.4) is Pareto optimal if the following conditions are satisfied:* $\frac{\omega_k}{S_k} > 0$ *for all* $k \in \{1, \dots, l\}$.

Proof. Denote with x^* the solution of the MOO problem (2.4), gained by maximizing the function $f(x) = \sum_{k=1}^{l} \omega_k Q_k^o(x)$. Obviously, it is satisfied that $f(x^*) \geq f(x), \forall x \in X$. Next, we get the following statements

$$\begin{aligned} & \sum_{k=1}^{l} \omega_k Q_k^o(x^*) \geq \sum_{k=1}^{l} \omega_k Q_k^o(x), \forall x \in X \\ \Leftrightarrow \quad & \sum_{k=1}^{l} \omega_k (Q_k^o(x^*) - Q_k^o(x)) \geq 0, \forall x \in X \\ \Leftrightarrow \quad & \sum_{k=1}^{l} \tfrac{\omega_k}{S_k}(Q_k(x^*) - Q_k(x)) \geq 0, \forall x \in X \qquad (2.5) \end{aligned}$$

Suppose contrary, that the solution x^* of the problem (1.1) is not Pareto optimal. Then there exists some feasible solution x' of the problem (1.1) for which is satisfied: $Q_k(x') \geq Q_k(x^*)$, which implies that

$$Q_k(x^*) - Q_k(x') \leq 0 \ \text{for all} \ k \in \{1, \dots, l\}.$$

Thereat there exists at least one index k_i for which the inequality is strong. By summing this inequalities and by considering the assumption of the theorem that values $\frac{\omega_k}{S_k}$ are all positive we get

$$\sum_{k=1}^{l} \frac{\omega_k}{S_k} [Q_k(x^*) - Q_k(x')] < 0.$$

Off course, this inequality stands in a contradiction with the statement (2.5). In this way, the observed solution x^* must be Pareto optimal. □

This theorem presents a way of construction of the weighted coefficients ω_i, $i = \overline{1, l}$ in order to generate only Pareto optimal solutions by applying the weighted sum method. That is, if a decision maker choose a positive real number c, weighted coefficients are automatically generated as $\omega_i = c \cdot S_i$, $i = \overline{1, l}$.

Corollary 2.1. *The solution of the MOO problem (1.1) in the case of non-normalized linear objective functions, generated by the weighted sum method (2.4) is Pareto optimal if the following conditions are satisfied:* $\omega_k > 0$ *for all* $k \in \{1, \dots, l\}$.

Mechanizing the process of constructing the Pareto optimal set, can be accomplished by a computer-aided construction of weighting coefficients w_i satisfying (2.3). This method is based on the standard `MATHEMATICA` function *Compositions*[]. For any chosen positive integer k, the function `Compositions[k,l]` can be used for the construction of the list which contains "l-dimensional points" (lists of l elements, $l = Length[q]$), such that the sum of their coordinates is equal to k. If such a list is divided by k, we obtain a p-element list whose elements are sublists representing compositions of 1 into l parts. Denote this list by $W = \{W_1, \dots, W_p\} = Compositions[k,l]/k$. It is easy to verify that $p = Length[W]$ is equal to the binomial coefficient of $k+l-1$ over $l-1$. Later we solve the problem (2.1) for each list W_i, $i = 1 \dots, p$, using $w_j = W_{i,j}$, $j = 1, \dots, l$.

According to the Theorem 2.1, for positive weights and convex problem, the optimal solutions of the substitute problem (2.1) are Pareto optimal (similar result is obtained in [11]). Minimizing (2.1) with strictly positive weights is the sufficient condition for the Pareto optimality. However, the formulation does not provide a necessary condition for Pareto optimality [12]. When the multicriteria problem is convex, an application of the function `W=Compositions[k,l]` produces b Pareto optimal solutions, where the integer b satisfies $1 \le b \le \frac{(k+l-1)!}{k!(l-1)!}$.

Example 2.1. In the case $k = 5, l = 2$, the expression `W=Compositions[k,l]/k` produces the following list W:

$$\{\{0,1\},\{\frac{1}{5},\frac{4}{5}\},\{\frac{2}{5},\frac{3}{5}\},\{\frac{3}{5},\frac{2}{5}\},\{\frac{4}{5},\frac{1}{5}\},\{1,0\}\}.$$

The number of Pareto optimal points is an integer between 1 and 6.

We also admit an explicit selection of coefficients w_i, $i = 1, \dots, l$ by the decision maker.

3. Implementation details for the two variables case

Consider the general form of multi-objective optimization problem in $\mathbb{R}^2$:

$$\begin{array}{ll} \text{Maximize:} & Q(\mathbf{x}) = [Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})], \quad \mathbf{x} \in \mathbf{R}^2 \\ \text{Subject to:} & a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i \le 0, \quad i \in I_1 \\ & a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i \ge 0, \quad i \in I_2 \\ & x, y \ge 0. \end{array} \tag{3.1}$$

where $I_1 \cup I_2 = \{1, \dots, m\}$, $I_1 \cap I_2 = \emptyset$ and a_{ij}, b_i, c_j are given real numbers and $m = |I_1| + |I_2|$, as explained in [8]. Each inequality constraint from (3.1) determines a subset $D_i \subset \mathbb{R}^2$, $i = 1, \dots, m$, representing the set of points on the one side of corresponding real algebraic curve $a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i = 0$. Therefore, the set of feasible solutions (denoted as Ω_P in $\mathbb{R}^2$) is determined as the intersection

$$\Omega_p = D_1 \cap D_2 \cap \dots \cap D_m \cap D_{m+1} \cap D_{m+2},$$

where subsets D_{m+1}, D_{m+2} of $\mathbb{R}^2$ are derived from the conditions $x \ge 0, y \ge 0$.

Formal parameters of the function *MultiW* are used in the following sense:

*q_*, *constr_List*, *var_List*: The list of unevaluated expressions (representing objective functions), the list of given constraints and the list of unassigned variables, respectively (the internal form of the problem).

*w*1_*List*: The empty list used as the value of the parameter $w1$ means that the weighting coefficients will be generated by means of the function *Compositions*. Otherwise, it is assumed that each element $w1[[i]], 1 \leq 1 \leq Length[w1]$ of the list $w1$ is a possible set of the coefficients w_j, $j = 1, \ldots, l$: $w[[j]] = w1[[i, j]]$, $j = 1, \ldots, l$.

Local variable *res* in the function *MultiW* collects the intermediate results. Also, the local variable *fun* represents the expression $Q(\mathbf{x}) = \sum w_i Q_i(\mathbf{x})$ in (2.1).

```
MultiW[q_, constr_List, var_List, w1_List] :=
    Module[{i = 0, k, l = Length[q], res = {}, w = {}, fun, sk = {},
   qres = {}, mxs = {}, Paretos = {}, m, ls = {}},
      If[w1 == {},
        k = Input["Initial sum of weighting coefficients? "];
        w = Compositions[k, l] / k,
        w = w1;
      ];
      Print["Weighting Coefficients: "]; Print[w];
      Print["Single-objective problems: "];
      k = Length[w];
      For[i = 1, i <= k, i++,
            fun = Simplify[Sum[w[[i, j]] * q[[j]], {j, l}]];
            Print[{fun, constr, var}];
            Temp = Maximize[fun, constr, var];
            AppendTo[res, Temp];
            If[IsPareto[q, constr, First[Rest[Temp]], var] == 1,
                  AppendTo[Paretos, List[Temp[[2, 1, 2]], Temp[[2, 2, 2]]]]]];
      ];
      Print["Solutions of single-objective problems: "];
      Print[res];
      For[i = 1, i <= k, i++,
         AppendTo[qres, q /. res[[i, 2]]];  AppendTo[mxs, res[[i, 1]]];
      ];
      Print["Values of objectives corresponding to solutions
             of single-objective problems: "];
      Print[qres];
      m = Max[mxs];
      For[i = 1, i ≤ Length[mxs], i++,
         If[m == mxs[[i]], AppendTo[ls, {qres[[i]], res[[i, 2]]}]; ]
      ];
      (* Print["Paretos = ",Paretos]; *)
      DrawParetos[Paretos, constr, var];
      Return[ls];
 ]
```

Example 3.1. Solve the following multi-objective optimization problem:

$$\begin{array}{rc} \max & [40x+10y; x+y] \\ p.o. & 2x+y \le 6 \\ & x+y \le 5 \\ & x \le 2 \\ & x, y \ge 0 \end{array}$$

Let us construct the coefficients w_i:

```
pts = MultiW[{40x + 10y, x + y}, {2x + y <= 6, x + y <= 5, x <= 2,
x>= 0, y >= 0}, {x, y}, {}]
```

In the case of $k = 5$ we have $w = \{\{0, 5\}, \{\frac{1}{5}, \frac{4}{5}\}, \{\frac{2}{5}, \frac{3}{5}\}, \{\frac{3}{5}, \frac{2}{5}\}, \{\frac{4}{5}, \frac{1}{5}\}, \{5, 0\}\}$. It is necessary to solve six problems of linear programming, for every sublist of w. Our module firstly enlists the linear programming problems, and then the appropriate solutions.

$$\begin{array}{ll} \max \frac{5(x+y)}{2}, & \{\{80, 5\}, \{x \to 1, y \to 4\}\} \\ \max \frac{14x+11y}{5}, & \{\{80, 5\}, \{x \to 1, y \to 4\}\} \\ \max \frac{31x+19y}{10}, & \{\{80, 5\}, \{x \to 1, y \to 4\}\} \\ \max \frac{17x+8)}{5}, & \{\{100, 4\}, \{x \to 2, y \to 2\}\} \\ \max \frac{37x+13y}{10}, & \{\{100, 4\}, \{x \to 2, y \to 2\}\} \\ \max \frac{4x+y}{1}, & \{\{100, 4\}, \{x \to 2, y \to 2\}\} \end{array}$$

By taking the weighted coefficients $w_1 = \frac{1}{5}$ i $w_2 = \frac{4}{5}$, we solve the problem of maximization of the function $\frac{14x+11y}{5}$ on the given constraints set. The solution $x = 1, \ y = 4$ is obtained. In that point the first objective function has the value 80, and the second 5. The solution $\{x = 1, y = 4\}$ is Pareto optimal by the definition. In the case of weighted coefficient $w_1 = \frac{3}{5}$ and $w_2 = \frac{2}{5}$, by maximizing the function $\frac{14x+11y}{5}$ we gain the solution $\{x = 2, y = 2\}$, which is also Pareto optimal.

A function for verification of Pareto optimality conditions and function for graphical illustration of the Pareto optimal points are described in the paper [9]. All generated solutions satisfying the function `IsPareto` implemented in [9] are used to generate the set of Pareto optimal points, denoted by `Paretos`. Standard `MATHEMATICA` function `ListPlot` plots points is contained in the list `Paretos.` Function `RegionPlot[ineqs, vars]` gives a graphical representation of the set of inequalities `ineqs`, with the variables `vars`. By means of the graphics functions `Show` we combine graphics of Pareto optimal points and the graphics of constraint set.

Each of the corresponding real algebraic curve divides the area into a range which is possible for these conditions and a range impossible for these conditions.

The permissible conditions are located in the range Ω_P that is permissible for all conditions (the region of feasible solution). Function `DrawParetos` uses the following parameters:

1. `Paretos_List`: Pareto points;
2. `constr_List`: the feasible set;
3. `var_List`: the list of given constraints.

```
DrawParetos[Paretos_List, constr_List, var_List]:= Block[{p1, p2},
    p1=ListPlot[Paretos, PlotStyle->{PointSize[0.033], Hue[1]},
        DisplayFunction->Identity];
    p2=RegionPlot[constr,{var[[1]]},{var[[2]]}, AspectRatio->1,
        DisplayFunction->Identity];
    Show[p2,p1 ,DisplayFunction->$DisplayFunction ];
]
```

Firstly we plot set of Pareto optimal points `Paretos` applying the standard `MATHEMATICA` function `ListPlot` (see [6]).

```
p1=ListPlot[Paretos, (* Display the set Paretos *)
         PlotStyle->{PointSize[0.033], Hue[1]},
         DisplayFunction->Identity       ]
```

In the next step, we plot the constraints set (graphics `constr`) applying the standard `MATHEMATICA` function `RegionPlot`. It is done by the following piece of code:

```
p2=RegionPlot[constr, (* Display the constraint set *)
          {var[[1]]},{var[[2]]}, AspectRatio->1,
          DisplayFunction->Identity       ]
```

Example 3.2. Solve the following problem:

$$\begin{array}{ll} \text{Maximize:} & Q_1(x1,x2) = 8x1 + 12x2,\ Q_2(x1,x2) = 14x1 + 10x2,\ Q_3(x1,x2) = x1 + x2 \\ \text{Subject to:} & 8x1 + 4x2 \ \le 600,\ 2x1 + 3x2 \ \le 300, \\ & 4x1 + 3x2 \ \le 360,\ x \ge 0, y \ge 0. \end{array}$$

We use, for example, the weighted sum method, and choose the following weighting coefficients:

$$\{\{0,0,1\},\{0,\frac{2}{5},\frac{3}{5}\},\{1,0,0\}\}$$

This problem can be solved by the expression

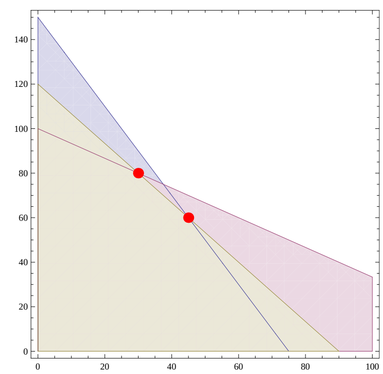
```
MultiW[{8x1+12x2, 14x1+10x2, x1+x2},
{8x1+4x2<=600,2x1+3x2<=300,4x1+3x2<=360,x1>=0,x2>=0},
   {x1,x2},    {{0,0,1}, {0,2/5,3/5}, {1,0,0}}]
```

Program generates three solutions, corresponding to each selection of weighting coefficients. For the first choice of these coefficients the solution of the problem is $\{\{1200, 1220, 110\}, \{x1{-}{>}\ 30, x2{-}{>}\ 80\}\}$, and is Pareto optimal.

The solution $\{\{1080, 1230, 105\}, \{x1{-}{>}\ 45, x2{-}{>}\ 60\}\}$, gained using the weighting coefficients $\{0, \frac{2}{5}, \frac{3}{5}\}$ is also Pareto optimal.

The optimal solution corresponding to the weighting coefficients $\{1, 0, 0\}$, is equal to $\{\{1200, 1000, 100\}, \{x1{-}{>}\ 0, x2{-}{>}\ 100\}\}$. This solution is not Pareto optimal, because the solution corresponding to the first choice of weights, where $\{x1{-}{>}\ 30, x2{-}{>}\ 80\}$, gives greater values for the functions Q_2 and Q_3, and the values for the function Q_1 in points $\{x1{-}{>}\ 30, x2{-}{>}\ 80\}$ and $\{x1{-}{>}\ 0, x2{-}{>}\ 100\}$ are equal.

This result is illustrated on the Figure 3.1, where the feasible design space is depicted as the region on which boundary the Pareto optimal points lye.



F . 3.1: Graphical representation of the solution

4. Implementation details for the three variables case

Consider the general form of the multi-objective optimization problem in $\mathbb{R}^3$ ($3D$ problem):

$$
\begin{array}{ll}
\text{Maximize:} & Q(\mathbf{x}) = [Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})], \quad \mathbf{x} \in \mathbf{R}^3 \\
\text{Subject to:} & a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + \\
 & 2a_{23}yz + 2a_1x + 2a_2y + 2a_3z + a_0 \le 0, \quad i \in I_1 \\
 & a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + \\
 & 2a_{23}yz + 2a_1x + 2a_2y + 2a_3z + a_0 \ge 0, \quad i \in I_2 \\
 & x, y, z \ge 0.
\end{array}
$$

Similar to the $2D$ case, the set of feasible solutions (in $\mathbb{R}^2$ denoted by Ω_P) is determined as the intersection

$$\Omega_p = D_1 \cap D_2 \cap \cdots \cap D_m \cap D_{m+1} \cap D_{m+2} \cap D_{m+3},$$

where $D_i \subset \mathbb{R}^3$ is set of the solutions of the i-th inequality and $D_{m+1}, D_{m+2}, D_{m+3} \subset \mathbb{R}^3$ are derived from the conditions $x \geq 0, y \geq 0, z \geq 0$.

```
MultiW3D[q_, constr_List, var_List, w1_List] :=
    Module[{i = 0, k, l = Length[q], res = {}, w = {}, fun,
           sk = {}, cfs, qres = {}, mxs = {}, m, ls = {}},
     If[w1 == {},
       k = Input["How many times? "];
       w = Compositions[k, l] / k,
       k = Length[w1] - 1;  w = w1;
     ];
     Print["Weighting Coefficients: "]; Print[w];
      (* Normalization *)
      For[i = 1, i ≤ l, i++,
         cfs = Coefficient[q[[i]], var];
         AppendTo[sk, Sum[cfs[[j]], {j, Length[cfs]}]   ];
      ];
      Print["Single-objective problems: "];
     For[i = 1, i <= k + 1, i++,
       fun = Simplify[Sum[w[[i, j]] * q[[j]] / sk[[j]], {j, l}]];
       Print[{fun, constr, var}];
       AppendTo[res, Penalty[fun, constr, var, {5, 5},
                10^-4, 10^-4, 30, Interior]];
     ];
     Print["Solutions of single-objective problems: "];
     Print[res];
     For[i = 1, i <= k + 1, i++,
       AppendTo[qres, q /. res[[i, 2]] ];  AppendTo[mxs, res[[i, 1]] ];
     ];
     Print["Values of objectives corresponding to
           solutions of single-objective problems: "];
     Print[qres];
     m = Max[mxs];
     For[i = 1, i ≤ Length[mxs], i++,
       If[m == mxs[[i]],  AppendTo[ls, {qres[[i]], res[[i, 2]]}]; ]
     ];
     Return[ls];
 ]
```

Corresponding algorithm is implemented `MATHEMATICA` function `DrawParetos3D` which solves given $3D$ problem and gives the interactive visualization of the Pareto optimal solution. This function has the following form:

```
DrawParetos3D[Paretos_List, constr_List, var_List, {}] :=
  Module[{p1,p2},
   {p1 =Graphics3D[Table[{Blue, PointSize[0.08], Point[Paretos[[i]]]},
        {i, Length[Paretos]}],  DisplayFunction -> Identity],
   p2 = RegionPlot[constr,(*Display the constraint set*)
```

```
{var[[1]]}, {var[[2]]}, {var[[3]]}, AspectRatio -> 1,
     DisplayFunction -> Identity],
     Show[p2, p1, DisplayFunction -> dip[EdgeForm[]],
     ViewPoint -> {3, 4, 0}, AspectRatio -> 1]}]
```

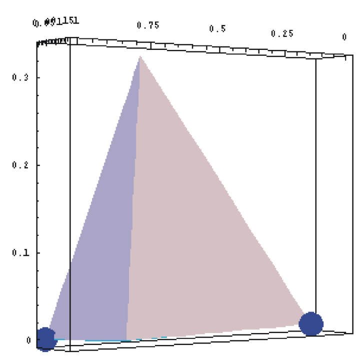
Example 4.1. Consider the following MOO problem, for the three variables case

$$\begin{array}{ll} \text{Maximize:} & Q(\mathbf{x}) = [x+y+z, -x-3y-6z, x+y+z] \\ \text{Subject to:} & x+y+z \leq 1 \\ & x+3y+z \geq 0 \\ & -x+5y+2z \leq 0 \\ & x \geq 0,\ y \geq 0,\ z \geq 0 \end{array}$$

```
pts = MultiW3D[{x + y + z, -x - 3y - 6z, x + y + z},
               { x + y + z <=1, x + 3y + z >= 0, -x + 5y + 2z <= 0 ,
                    z >= 0, y >= 0, x >= 0}, {x,y,z}, {}]
```

In case where $k = 6$, we have more possibilities for better graphical illustration (see Figure 4.1).

```
DrawParetos3D[pts, { x + y + z <= 1, x + 3y >= -z, -x + 5y <= -2z,
                     z>= 0, y >= 0, x >= 0}, {x, y, z}, {}]
```



F . 4.1: The representation of the feasible design space and the Pareto set

Example 4.2. Let us observe the following $3D$ multi-objective problem

$$\begin{array}{ll} \text{Maximize:} & Q(\mathbf{x}) = [x+y+z, -x-3y-6z, x+y+z] \\ \text{Subject to:} & x^2+y^2+z^2 \leq 1 \\ & x \geq 0,\ y \geq 0,\ z \geq 0 \end{array}$$

By applying the next function

```
pts =MultiW3D[{x+y+z, -x-3y-6z, x+y+z},
              {x^2+y^2+z^2<=1, z>=0, x>=0}, {x,y,z}, {}]
```

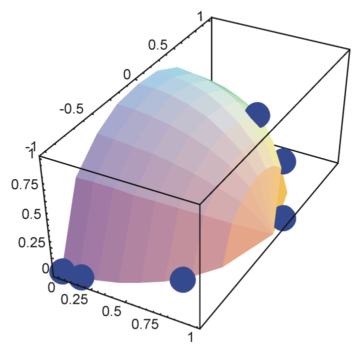
when $k = 9$, we get next Pareto optimal points:

$$\{0,-1,0\},\{-\tfrac{1}{\sqrt{2}}+\sqrt{2},-\tfrac{1}{\sqrt{2}},0\},\{-\tfrac{49}{5\sqrt{2}}+5\sqrt{2},-\tfrac{7}{\sqrt{2}},0\},$$
$$\{-\tfrac{2}{\sqrt{3}}+\sqrt{3},\tfrac{1}{\sqrt{3}},\tfrac{1}{\sqrt{3}}\},\{\tfrac{1}{5}(-\tfrac{1}{\sqrt{26}}+\sqrt{26}),\tfrac{1}{\sqrt{26}},0\},\{\tfrac{1}{7}(-2\sqrt{\tfrac{2}{\sqrt{39}}}-\tfrac{25}{\sqrt{78}}+\sqrt{78}),\tfrac{5}{\sqrt{78}},\sqrt{\tfrac{2}{39}}\}.$$

Using the following function

```
DrawParetos3D[pts, { x^2 + y^2 + z^2 <= 1, z >= 0, x >= 0}, {x, y,z}, {}]
```

we obtain graphical representation of Pareto optimal points



F . 4.2: The Pareto optimal points lye on the boundary of the feasible design set

By applying the previous code we obtain the nice interactive demonstration. Demonstration also provides ability to the user to rotate image and see from different angles and points of view, and possibility to see all Pareto optimal points.

5. Conclusion

We have introduced the result giving the necessary conditions for multi-objective optimization problem to have Pareto optimal solutions. The practical criteria for the construction of weighted coefficients in order to gain only the solutions which are Pareto optimal, is provided. Some graphical representations of 2D and 3D case are given in order to depict the introduced method.

REFERENCES

1. V. C , Y. H , *Multiobjective decision making: Theory and methodology Series, Volume 8*, North-Holland, New York, Amsterdam, Oxford, 1983.
2. I. D , J.E. D , *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.
3. A.M. G , *Proper efficiency and the theory of vector maximization*, J. Math. Anal. Appl. **22**, (1968), 618–630.
4. K. M , *Nonlinear Multiobjective Optimization*, Kluver Academic Publishers, Boston, London, Dordrecht, 1999.
5. R.E. M , *Programming in Mathematica, Third Edition* Redwood City, California: Adisson-Wesley, 1996.
6. R.E. M , *Computer Science with Mathematica* Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, Sao Paulo, 2000.
7. R.T. M , *Survey of multi-objective optimization methods for engineering* Struct. Multidisc. Optim. **26** (2004) 369–395.
8. P.S. S ´, M.L . Z ´, M.D.P ´, *Visualisation in optimization with MATHEMATICA*, Filomat **23:2** (2009), 68–81.
9. P.S. S ´, I.P. S ´, *Implementation of polynomial multi-objective optimization in MATHEMATICA* Struct. Multidisc. Optim. **36** (2008), 411–428.
10. S. W , *The Mathematica Book, 4th ed.*, Wolfram Media/Cambridge University Press, 1999.
11. L.A. Z , *Optimality and non-scalar-valued performance criteria*, IEEE Trans. Autom. Control AC-8, (1963) 5960.
12. S. Z , *Multiple criteria mathematical programming: an updated overview and several approaches*, In: Mitra, G. (ed.) Mathematical Models for Decision Support, 1988, pp. 135167, Berlin: Springer-Verlag.

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`

Milan Lj. Zlatanović
Faculty of Sciences and Mathematics
Department of Mathematics
P. O. Box 224, Visegradska 33,
18000 Niš, Serbia
`zlatmilan@yahoo.com`

Marko D. Petković
Faculty of Sciences and Mathematics
Department of Computer Science
P. O. Box 224, Visegradska 33,
18000 Niš, Serbia
dexterofnis@gmail.com