IIASA

International Institute for Applied Systems Analysis • A-2361 Laxenburg • Austria
Tel: +43 2236 807 • Fax: +43 2236 71313 • E-mail: info@iiasa.ac.at • Web: www.iiasa.ac.at

INTERIM REPORT IR-97-58 /September

Quadratic Pareto Race

Pekka Korhonen (korhonen@iiasa.ac.at)
GuangYuan Yu (yu@hkkk.fi)

Approved by

Gordon MacDonald (macdon@iiasa.ac.at)
Director, IIASA

Contents

Abstract

This paper presents a dynamic and visual “free-search” type of a decision support system - Quadratic Pareto Race, which enables a decision maker (DM) to freely search any part of the efficient frontier of a multiple objective quadratic-linear programming problem by controlling the speed and direction of motion. The values of the objective functions are presented in a numeric form and as bar graphs on a display. The implementation of Quadratic Pareto Race is based on the theoretical foundations developed by Korhonen and Yu [1996]. The system is implemented on a microcomputer and illustrated using a numerical example.

Keywords: Multiple Objective Programming, Quadratic Programming, Reference Direction, Linear Complementarity Model, Interactive Procedure, Decision Support Systems.

Acknowledgments

The research was supported, in part, by grants from the Foundation of the Helsinki School of Economics and the Academy of Finland.

About the Authors

Pekka Korhonen is Project Leader of the Decision Analysis and Support Project at IIASA, and also Professor of Economics at the Department of Economics and Management Science, Helsinki School of Economics and Business Administration.

Guang Yuan Yu is a Ph.D student at the Department of Economics and Management Science, Helsinki School of Economics and Business Administration.

Quadratic Pareto Race

Pekka Korhonen
GuangYuan Yu

1. Introduction

Pareto Race was developed by Korhonen and Wallenius [1988] to solve multiple objective linear programming problems and it was characterized as follows: "*Pareto Race represents a dynamic, visual, and interactive procedure for multiple objective linear programming. In Pareto Race, a decision maker (DM) can freely search the efficient frontier of a multiple objective linear programming problem by controlling the speed and direction of motion. On a display, the DM sees the objective function values in numeric form and as bar graphs whose lengths are dynamically changing as he moves about on the efficient frontier. The keyboard controls include gears, an accelerator, brakes, and a steering mechanism.*" A key feature in Pareto Race is that it does not dictate to the DM what to do. The system only controls that all the solutions the DM evaluates are "reasonable", i.e. efficient, but the DM is free to choose any from among those reasonable solutions. The method behind the system is based on the use of the reference direction approach which was developed by Korhonen and Laakso [1986]. In Pareto Race, the reference direction is specified implicitly in a dynamic manner. No assumptions concerning the DM's behavior is required.

Pareto Race is an essential part of VIG, which is a computer system developed by Korhonen [1987] to support the structuring and solving of multiple criteria decision making and planning problems. VIG is widely applied to many multiple criteria decision problems, for instance, in determining the changes of optimal prices in an alcohol sales monopoly in Finland (see, Korhonen and Soismaa [1988]), in stockpiling critical materials for a national emergency in Finland (see, Kananen, Korhonen, Wallenius and Wallenius [1990]), and in selecting a flexible manufacturing system (see, Stam and Kuula [1991]), etc.

Pareto Race has been developed to solve multiple objective linear programming problems. This restricts its use to the problems which include non-linear features. Unfortunately, nonlinear multiple criteria problems are not an exception. For excellent surveys, see e.g., White [1990], Saber and Ravindran [1993], and Boyan [1995]. However, not many interactive software systems exist for solving multiple objective nonlinear programming problems. Typically, in those systems one or several nondominated solutions are generated and presented for the DM's evaluation at each iteration. One example of those procedures is a "Light Beam Search" – procedure developed by Jaszkiewicz and Slowinski [1994]. The procedure first finds a nondominated solution and then defines a subregion of the nondominated set according to the current solution by using the form of an outranking relation which takes into account the preference information of the DM. Finally, a sample of nondominated points in the subregion, composed of the current solution and a number of alternative proposals, is presented to the DM. The LBS procedure is a typical nonlinear multiple criteria procedure in the sense that the sample set of nondominated solutions is finite. For this reason, some important nondominated solutions may not be available to the DM. Generally, in a nonlinear case, it is very time consuming to generate – actually to approximate – a continuous path on the nondominated frontier like, for instance, in Pareto Race.

Following the original idea of Pareto Race, our aim is to develop a similar system – called Quadratic Pareto Race – for dealing with a special case of a multiple objective non-linear problem: multiple objective quadratic-linear programming problem (MOQLP). Our problem consists of one quadratic objective, several linear objectives, and a set of linear constraints. For instance, an extension of the classical portfolio selection problem is of this type. The problem can be transformed into a parametric linear complementarity problem and solve using techniques developed for solving multiple objective linear programming problems. The underlying theory is developed in two papers by Korhonen and Yu [1996] and [1997].

Quadratic Pareto Race is based on the use of reference directions (Korhonen and Laakso [1986]) and weighted-sums. Reference directions for the linear functions, and weighted-sums for combining the quadratic function with the linear ones are used as parameters to implement a free search on the nondominated frontier. The interface of Quadratic Pareto Race is very similar to the original Pareto Race.

The rest of the paper is organized as follows: In section 2, we briefly review theoretical foundations. In section 3, we describe the parametric linear complementarity formulation of the problem and its solution procedure. In section 4, Quadratic Pareto Race is developed and a numerical example is given in section 5. In the last section, we present some concluding remarks and discuss future applications.

2. Theoretical Foundations

2.1 Problem Formulation

The multiple objective quadratic-linear programming problem is defined as follows:

$$\text{Min} \quad V(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}'D\mathbf{x}$$

$$\text{"Max"} \quad \mathbf{l}(\mathbf{x}) = C\mathbf{x}$$

$$\text{Subject to:} \qquad (2.1)$$

$$A\,\mathbf{x} \le \mathbf{b}$$

$$\mathbf{x} \ge \mathbf{0},$$

where D is a symmetric positive semi-definite matrix of order $n \times n$, C is the $k \times n$ matrix of coefficients of the linear objective functions, A is an $m \times n$ matrix (rank m) of coefficients of the constraints, $\mathbf{b}$ is an m-vector of the rhs, and $\mathbf{x}$ is an n-vector of the decision variables. By "Max" we mean that all linear objective functions have to be maximized while the quadratic objective function is to be minimized.

We will denote the objective function vector by $\mathbf{f}(\mathbf{x})$ and refer to its components as follows:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} -V(\mathbf{x}) \\ \mathbf{l}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_0(\mathbf{x}) \\ f_1(\mathbf{x}) \\ \dots \\ f_k(\mathbf{x}) \end{pmatrix}.$$

The problem can now be written as:

$$\begin{aligned} &\text{"Max" } \mathbf{f}(\mathbf{x}) \\ &\text{Subject to:} \qquad\qquad (2.2) \\ &\qquad \mathbf{x} \in X = \{\mathbf{x} \mid A\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0}\} \end{aligned}$$

We define efficiency and weak efficiency for a point $\mathbf{x}^0 \in X$ in the usual manner:

Definition 1. $\mathbf{x}^0 \in X$ is **efficient** iff $\nexists\, \mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) \ge \mathbf{f}(\mathbf{x}^0)$ and $\mathbf{f}(\mathbf{x}) \ne \mathbf{f}(\mathbf{x}^0)$.

Definition 2. $\mathbf{x}^0 \in X$ is **weakly efficient** iff $\nexists\, \mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) > \mathbf{f}(\mathbf{x}^0)$.

The objective function vectors $\mathbf{q} \in Q = \{\mathbf{q} \mid \mathbf{q} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in X\}$ corresponding to **(weakly) efficient** points are called **(weakly) nondominated** solutions.

2.2 A Parametric Model for Searching Efficient Frontier

To search (weakly) nondominated solutions of a problem (2.1), Korhonen and Yu [1996] introduced the following parametric model:

$$\begin{aligned} &\text{Min} \quad \tfrac{1}{2}\mathbf{x}'D\mathbf{x} + (\lambda + t\Delta\lambda)\varepsilon \\ &\text{Subject to:} \qquad\qquad (2.3) \\ &\qquad C\mathbf{x} + \varepsilon\mathbf{w} \ge \mathbf{g} + t\Delta\mathbf{g} \\ &\qquad A\mathbf{x} \le \mathbf{b} \\ &\qquad \mathbf{x} \ge \mathbf{0}, \end{aligned}$$

where $t \in [0, t^u)$, $t^u = \begin{cases} \infty, & \text{if } \Delta\lambda \geq 0 \\ \dfrac{\lambda}{-\Delta\lambda}, & \text{if } \Delta\lambda < 0 \end{cases}$, $\lambda > 0$, and thus $(\lambda + t\Delta\lambda) > 0$ for all $t \in [0, t^u)$.

By varying t, $\Delta\lambda$ and $\Delta\mathbf{g}$ we may search the nondominated frontier of problem (2.1).

Parameter $\lambda \leftarrow \lambda + t\Delta\lambda$ combines the quadratic objective function with the linear ones. When λ increases the "relative meaning" of the quadratic function in the combined objective function in model (2.3) decreases, and when it approaches zero, the solution approaches to the minimum of the quadratic function. Vector $\Delta\mathbf{g} \in \mathfrak{R}^k$ is a so-called reference direction vector and is used to control desirable changes in linear objective functions. Respectively, parameter $\Delta\lambda$ is used to change the "relative meaning" of the quadratic part. By varying the values of $\Delta\lambda$ and $\Delta\mathbf{g}$, the DM may control a search direction on the nondominated frontier, and parameter t is used to control a 'speed' on that frontier.

In model (2.3), we made a clear distinction between linear objective functions and linear constraints. However, from the technical point of view, the distinction is rather artificial as we can see from the following formulation:

$$\text{Min} \quad \tfrac{1}{2}\mathbf{x}'D\mathbf{x} + (\lambda + t\Delta\lambda)\varepsilon$$

$$\text{Subject to:} \qquad\qquad (2.4)$$

$$H\mathbf{x} + \varepsilon\mathbf{w}^+ \geq \mathbf{g}^+ + t\Delta\mathbf{g}^+$$

$$\mathbf{x} \geq \mathbf{0}$$

$$t : 0 \rightarrow t^* \in [0, t^u),$$

where $H = \begin{pmatrix} C \\ -A \end{pmatrix} \in \mathfrak{R}^{(k+m)\times n}$, $\mathbf{w}^+ = \begin{pmatrix} \mathbf{w} \\ \mathbf{0} \end{pmatrix} \in \mathfrak{R}^{k+m}$, $\mathbf{g}^+ = \begin{pmatrix} \mathbf{g} \\ \mathbf{-b} \end{pmatrix} \in \mathfrak{R}^{k+m}$, $\Delta\mathbf{g}^+ = \begin{pmatrix} \Delta\mathbf{g} \\ \mathbf{0} \end{pmatrix} \in \mathfrak{R}^{k+m}$,

and $t^u = \begin{cases} \infty, & \text{if } \Delta\lambda \geq 0 \\ \dfrac{\lambda}{-\Delta\lambda}, & \text{if } \Delta\lambda < 0 \end{cases}$.

Which linear inequalities are standing for linear objective functions and which for linear constraints depends on the components of vectors $\mathbf{w}^+ \in \mathfrak{R}^{k+m}$ and $\Delta\mathbf{g}^+ \in \mathfrak{R}^{k+m}$:

$$w_i^+ \begin{cases} = 0, & \text{if i refers to a constraint} \\ > 0, & \text{if i refers to an objective} \end{cases}$$

and

$$\Delta g_i^+ \begin{cases} = 0, & \text{if i refers to a constraint} \\ \neq 0, & \text{if i refers to an objective} \end{cases}.$$

Formulation (2.4) now provides us with a simple formulation to change the role of linear objectives and constraints. When a constraint is changed into an objective, a certain positive value is given to the corresponding component of the weight vector $\mathbf{w}^+$, and a non-zero value is given to the corresponding component of vector $\Delta\mathbf{g}^+$. When the change is made in the opposite direction, the corresponding components of vectors $\mathbf{w}^+$ and $\Delta\mathbf{g}^+$ are set to zero.

Using the unified formulation for linear objective functions and constraints, we do not need to fix their role in the beginning of the search process, but we may freely change it during the search process. The quadratic function is the only function, which is fixed to be one of the objective functions. The decision support system described in the following section, supports this kind of an evolutionary approach. We will use the common term ‘goal’ to refer to objectives and constraints. Following Ignizio [1983] objectives are sometimes called ‘flexible goals’ and constraints ‘inflexible or rigid goals’.

Model (2.4) does not provide an operational way to continuously search an efficient frontier. For that purpose, we transform the model (2.4) into a linear complementarity form, which makes it possible to apply the methods developed for multiple objective linear programming.

3. Development of the Linear Complementarity Model

3.1 Linear Complementarity Formulation

To simplify notation, we drop the superscript “$^+$” from the vectors $\mathbf{w}^+$, $\mathbf{g}^+$, and $\Delta\mathbf{g}^+$. For each given t, $\Delta\lambda$, and $\Delta\mathbf{g} \in \Re^{k+m}$, the quadratic problem (2.4) can be transformed to a linear complementarity problem as shown in Korhonen and Yu [1996]:

Find a solution for the problem:

$$\begin{aligned}
-D\mathbf{x} + H'\mathbf{y} + \alpha &= \mathbf{0} \\
\mathbf{w}'\mathbf{y} + \beta &= \lambda + t\Delta\lambda \\
-H\mathbf{x} - \varepsilon\mathbf{w} + \delta &= -\mathbf{g} - t\Delta\mathbf{g} \qquad (3.1)\\
\mathbf{x}, \mathbf{y}, \alpha, \delta \geq \mathbf{0}&, \beta = 0 \\
\alpha_i x_i = 0,&\ \ i = 1,2, ..., n \\
\delta_j y_j = 0,&\ \ j = 1,2, ..., k+m, \\
t : 0 \rightarrow t^* &\in [0, t^u),
\end{aligned}$$

$$\text{where } t^u = \begin{cases} \infty, & \text{if } \Delta\lambda \geq 0 \\ \dfrac{\lambda}{-\Delta\lambda}, & \text{if } \Delta\lambda < 0 \end{cases}.$$

Above we have transformed the original MOQLP-problem into a parametric linear complementarity problem. The model (3.1) now provides us with a straightforward way to make search similar to Pareto Race in multiple objective linear programming. The search is controlled by means of the problem parameters: t, $\Delta\lambda$, and $\Delta\mathbf{g} \in \Re^{m+k}$.

The search is started with t = 0. After finding an initial solution, we use a parametric principal pivoting method to move on the efficient frontier. The method is briefly reviewed in the following subsection. For a more detailed description, see, e.g., Cottle, Pang, and Stone [1992, pp. 288–308]. The implemented procedure is called Quadratic Pareto Race and explained in more detail in section 4.

3.2 Parametric Principal Pivoting Algorithm

In this subsection, we briefly describe the parametric linear complementarity problem and the parametric principal pivoting method for solving it. To keep the text readable, we will use notation commonly used in the context of the linear complementarity problem.

Given a $p \times p$ matrix M, and a vector $\mathbf{q} \in \Re^p$, the *linear complementarity problem* is to find nonnegative vectors $\mathbf{w} \in \Re^p$ and $\mathbf{z} \in \Re^p$ such that

$$\mathbf{w} - M\mathbf{z} = \mathbf{q} \tag{3.2}$$

$$\mathbf{w} \geq \mathbf{0},\ \ \mathbf{z} \geq \mathbf{0}$$

$$\mathbf{w}'\mathbf{z} = \mathbf{0}.$$

For our purposes, it is sufficient to assume that M is positive semi-definite. This assumption simplifies considerations.

Correspondingly, a *parametric linear complementary problem* is the family of linear complementarity problems:

$$\mathbf{w} - M\mathbf{z} = \mathbf{q} + t\Delta\mathbf{q} \tag{3.3}$$

$$\mathbf{w} \geq \mathbf{0},\ \ \mathbf{z} \geq \mathbf{0}$$

$$\mathbf{w}'\mathbf{z} = \mathbf{0}$$

where $\Delta\mathbf{q} \in \Re^p$ is a parametric vector and parameter t is running over a closed interval I = [a, b] in $\Re$. We will consider the case, where $0 \in I$.

First, we solve the problem (3.2), which corresponds to the case t = 0. If the problem has a solution, then under the assumptions we made about M, the problem has a solution for all $t \in I$ provided that the problem has a solution for t = a and t = b. To solve the problem (3.2), we may use the traditional *Complementary Pivoting Algorithm* proposed by Lemke [1968], or the parametric principal pivoting method in the way as explained later on.

Following Cottle, Pang, and Stone [1992, pp. 294–295], we describe the *Symmetric Parametric Principal Pivoting Algorithm*:

Step 0. Let's assume that problem (3.2) has a nondegenerate solution. Without loss of generality, we may assume that $\mathbf{q} > \mathbf{0}$ implying that $(\mathbf{w}, \mathbf{z}) = (\mathbf{q}, \mathbf{0})$ is the solution of problem (3.2). We define h:= 0, $\mathbf{q}^h = \mathbf{q}$, $\Delta\mathbf{q}^h = \Delta\mathbf{q}$, $\mathbf{w}^h = \mathbf{w}$, $\mathbf{z}^h = \mathbf{z}$, and $M^h = M$. We also specify $t_{max} \in [0, \infty)$.

Step 1. Determine the next critical value of t:

$$t^{h+1} = \min\{\min_i \{\frac{-q_i^h}{\Delta q_i^h} \mid \Delta q_i^h < 0\}, t_{max}\}$$

and set

$$(\mathbf{w}^h(t), \mathbf{z}^h(t)) = (\mathbf{q}^h + t\Delta\mathbf{q}^h, \mathbf{0}) \text{ for all } t \in [t^h, t^{h+1}].$$

If $t^{h+1} = t_{max}$, Stop. Otherwise let

$$r = \arg\min_i \{\frac{-q_i^h}{\Delta q_i^h} \mid \Delta q_i^h < 0\}.$$

The new critical value of t is $t^{h+1} = \dfrac{-q_r^h}{\Delta q_r^h}$.

Step 2. If $m_{rr}^h > 0$, then pivot matrix M^h using as a pivot element m_{rr}^h, and denote a resulting matrix by M^{h+1}. Set

$$w_r^{h+1} = z_r^h, \quad z_r^{h+1} = w_r^h,$$

$$w_i^{h+1} = w_i^h, \quad z_i^{h+1} = z_i^h, \; i \neq r,$$

and h := h+1. Return to Step 1.

If $m_{rr}^h = 0$, determine the index of the leaving variable as usually in the parametric linear programming. Let it be s. Pivot M^h twice using as pivot elements m_{sr}^h and m_{rs}^h, and denote a resulting matrix by M^{h+1}. Set

$$w_s^{h+1} = z_r^h, \qquad z_s^{h+1} = w_r^h,$$

$$w_r^{h+1} = z_s^h, \qquad z_r^{h+1} = w_s^h,$$

$$w_i^{h+1} = w_i^h, \qquad z_i^{h+1} = z_i^h, \; i \neq r, s$$

and h := h+1. Return to Step 1.

See below the Remark following the algorithm.

Remark. Assumption **q** > **0** made in **Step 0**, is not restrictive, because we can transform the problem (3.2) into a parametric form. Choose an arbitrary vector **q*** > **0**, and formulate the problem as follows:

Find a solution for the problem, when $\tau = 1$:

$$\mathbf{w} - M\mathbf{z} = \mathbf{q}^* + \tau(\mathbf{q} - \mathbf{q}^*) \tag{3.4}$$

$$\mathbf{w} \geq \mathbf{0}, \; \mathbf{z} \geq \mathbf{0}$$

$$\mathbf{w}'\mathbf{z} = \mathbf{0}.$$

Because **q*** > **0**, the solution of the problem (3.4) is **w** = **q*** when τ=0. To solve the problem, when τ = 1, we use the algorithm above. The problem has always a solution, when τ = 0, and if the problem has a solution with τ=1, then under the assumptions we made about M, the problem has a solution for all $\tau \in [0, 1]$.

4. Quadratic Pareto Race

The description of Quadratic Pareto Race follows the description of Pareto Race as proposed by Korhonen and Wallenius [1988] and it is also implemented in a similar way.

Step 0. *Problem Setup*

Provide the rows and variables of the problem with names; define the constraint matrix H and give a symmetric positive semi-definite matrix D in a lower-triangular form (see, Figure 1); specify the initial aspiration levels (**g** vector) for linear goals; and define the goal categories (≥, ≤, or =) (see, Figure 2). (To simplify the notation. we assume that all inequality goal categories are “greater than or equal to”. When a goal with the category “greater than or equal to” is defined as a flexible goal, the optimization is interpreted to mean “maximize”.)

Step 1. *Classification of Goals*

Specify the goals you consider objectives (flexible goals), and denote the corresponding index set by G. (Initially, G = {0}, where index 0 refers to the quadratic objective function.) The set of constraints is denoted by R = {1, 2, …, k+m}. (This step is implemented exactly as in Pareto Race (Korhonen and Wallenius [1988]).

Step 2. *Specification of Ranges*

Specify approximate (subjective) lower and upper bounds g_i^L and g_i^U for all objectives, $i \in G$, and require $g_i^U > g_i^L$. Set $w_i = g_i^U - g_i^L$, when $i \in G - \{0\}$and $w_i = 0$, when $i \in R+\{0\}$. (The initial ranges are used for the initialization of the race and for the specification of reference directions for linear objectives. The ranges are automatically updated during the race if necessary.) If the DM is unable to specify the ranges, the system will generate them by using previously (in **Step 0**) specified goal values **g**: $g_i^U = 1.5g_i$ and $g_i^L = 0.5g_i$. If $g_i = 0$, then $g_i^U = 1$ and $g_i^L = -1$.

Define an initial reference direction vector $\Delta\mathbf{g}^0$ and $\Delta\lambda^0$ as follows:

$$\Delta g_i^0 = \begin{cases} 0 & , i \notin G\text{-}\{0\} \\ \dfrac{g_i^U - g_i^L}{\sum\limits_{j\in G} (g_j^U - g_j^L)} & , i \in G\text{-}\{0\} \end{cases}$$

and

$$\Delta\lambda^0 = \frac{g_0^U - g_0^L}{\sum\limits_{j\in G} (g_j^U - g_j^L)} .$$

Step 3. *An Initial Solution*

Find an efficient (possibly weakly-efficient) solution to the following linear complementarity problem:

$$\begin{aligned} -D\mathbf{x} \;+\; H'\mathbf{y} \;+\; \alpha &= \mathbf{0} \\ \mathbf{w}'\mathbf{y} \;+\; \beta &= \lambda \\ -H\mathbf{x} \;-\; \varepsilon\mathbf{w} \;+\; \delta &= -\mathbf{g} \qquad (4.1) \\ \mathbf{x}, \mathbf{y}, \alpha, \delta \ge \mathbf{0}, \; \beta = 0 & \end{aligned}$$

$\alpha_i x_i = 0, \; i = 1,2, ..., n$

$\delta_j y_j = 0, \; j = 1,2, ..., k+m.$

Initially, let λ be a small positive number. (We use the value $\lambda := \Delta\lambda^0 \bullet 10^{-7}$). This guarantees that the initial solution is the efficient solution of problem (2.3) (see, e.g., Geoffrion [1968]), but it might still be a weakly efficient solution of the original problem (2.1). If λ is small enough, the race is started from the optimum of the quadratic term or from its neighborhood.

Specify $\Delta\mathbf{g} = \Delta\mathbf{g}^0$ and $\Delta\lambda = \Delta\lambda^0$.

Step 4. *Moving About the Efficient Frontier (Quadratic Pareto Race*)

The use of Quadratic Pareto Race is illustrated in **Figures 1–5**.

To implement the moving mechanism, we have to initialize several sets of constants and have found the following experimented values most suitable:

ω	$:= 2.5$	the multiplier determining the degree of change in the direction of the motion
ρ	$:= 3$	Multiplier determining the increase in speed
Δt^0	$:= 10^{-5}$	“base speed”; i.e. a step size.

There are three parameters which can be used to control the motion on the efficient frontier: the reference vector $\Delta\mathbf{g}$ for linear objectives, the parameter $\Delta\lambda$ for varying the “relative meaning” of the quadratic objective, and the scalar t. By varying $\Delta\mathbf{g}$ and $\Delta\lambda$ we change the direction of motion, and “speed” is controlled by t. The value of t is varied using the step-size Δt, which is initially set $\Delta t := \Delta t^0$.

The DM does not need to explicitly specify these parameters. We have built a mechanism that changes them. When the DM wants to “travel” faster, the step size parameter Δt is increased. When he/she wants to improve some objectives, we change $\Delta\mathbf{g}$ and $\Delta\lambda$ in such a way that the DM can see improvements in the value of objectives in question.

For each given $\Delta\mathbf{g}$ and $\Delta\lambda$, we can find the range $[t_L, t_U]$ for which the basis of the model (3.1) is the same, when $t \in [t_L, t_U]$. When the new values of the parameters are specified or the basis change is performed, then a new range has to be determined. When the DM reaches the point, from which he/she cannot continue the search with current parameter values, he/she is guided to respecify new values for parameters.

Quadratic Pareto Race is controlled by using the following function keys which are the same as already used in Pareto Race, see for more details Korhonen and Wallenius [1988]:

(SPACE) BAR: An "Accelerator"

Proceed in the current direction at constant speed. (The step-size is constant.)

On the screen, the DM will see the solutions corresponding to the following values of t:

$$t := \begin{cases} \min\{t+\Delta t, t_U\}, \text{ if } \Delta t > 0 \\ \max\{t-\Delta t, t_L\}, \text{ if } \Delta t < 0 \end{cases} .$$

If $t = t_U$, change the basis, and update the range. If $t_U = \infty$ or $t_U = t_{max}$ or ($\Delta t < 0$ and $t = t_L$), ask the DM to initiate a turn.

When $t \in [t_L, t_U]$, i.e. the basis remains the same, we can present the values of the quadratic function and the linear functions as follows:

$$V(\mathbf{x}(t)) = V(\mathbf{x}^0) + t\mathbf{x}^{0\,\prime}D\Delta\mathbf{x} + \tfrac{1}{2}t^2\Delta\mathbf{x}^{0\,\prime}D\Delta\mathbf{x}$$

$$\mathbf{l}(\mathbf{x}(t)) = C\mathbf{x}^0 + tC\Delta\mathbf{x},$$

where $\Delta\mathbf{x} = \Delta\mathbf{x}(\Delta\lambda, \Delta\mathbf{g})$ indicates how the values of the vector of decision variables **x** depend on the direction parameters $\Delta\lambda$ and $\Delta\mathbf{g}$. (Actually, we will use in the implemented algorithm a so-called indirect method for computing $V(\mathbf{x}(t))$ and $\mathbf{l}(\mathbf{x}(t)$ as explained in Korhonen and Yu [1997]. However, an indirect method is less illustrative. That’s why the direct method is used here to illustrate the idea.)

F1: "Gears (Backward)"

Increase speed in the backward direction. (The absolute value of the step-size is increased and the sign is negative.)

$$\Delta t := \begin{cases} -\Delta t^0 \ , \text{ if } \Delta t > 0 \\ \rho\bullet\Delta t, \text{ if } \Delta t < 0 \end{cases}$$

F2: "Gears (Forward)"

Increase speed in the forward direction. (The absolute value of the step-size is increased and the sign is positive.)

$$\Delta t := \begin{cases} \Delta t^0 \ , \text{ if } \Delta t < 0 \\ \rho\bullet\Delta t, \text{ if } \Delta t > 0 \end{cases}$$

F5: "Brakes"

Reduce speed as follows. (The absolute value of the step-size is decreased.)

$$\Delta t := \begin{cases} \max\{\Delta t/\rho, \Delta t^0\}, \text{ if } \Delta t > 0 \\ \min\{\Delta t/\rho, -\Delta t^0\}, \text{ if } \Delta t < 0 \end{cases}$$

F10: "Exit"

Return to Main Menu.

num: "Turn"

Change the direction of motion by increasing the component of the reference direction corresponding to the objective function's ordinal number $j \in [1,h]$ pressed by the DM. Each $j \in [1,h]$ and $i \in G$ has a one-to-one correspondence. We define that ordinal number h corresponds to the index of the quadratic objective function (0).

The DM can continue pressing various number keys, until he/she would like to see a new search direction. Then he/she will press "Space Bar". How many times the DM will press a specific number key, indicates how strongly he/she would like to improve the objective in question. Before $\Delta\mathbf{g}$ and $\Delta\lambda$ are updated, the current values of the aspiration levels are updated $\mathbf{g} := \mathbf{g} + t^c\Delta\mathbf{g}$ and $\lambda := \lambda + t^c\Delta\lambda$, where t^c corresponds to the current value of t. Based on the information received from the DM, the system will respecify the parameter $\Delta\lambda$ and the components of the reference direction $\Delta\mathbf{g}$. A new range for parameter t is needed, and the search will continue.

Let n_i be the integer number indicating how many times the DM pressed the ordinal number j corresponding to the objective i. The reference direction $\Delta\mathbf{g}$ will be updated as follows:

$$\Delta g_i := \begin{cases} 0 & , i \notin G\text{-}\{0\} \\ \dfrac{n_i\omega\Delta g_i + \Delta g_i^0}{\sum\limits_{j\in G\text{-}\{0\}} (n_j\omega\Delta g_j + \Delta g_j^0) + \left|\Lambda n_0\omega|\Delta\lambda| + \Delta\lambda^0\right|} & , i\in G\text{-}\{0\} \end{cases} .$$

Parameter $\Delta\lambda$ is updated as:

$$\Delta\lambda := \begin{cases} \dfrac{-n_0\omega|\Delta\lambda| + \Delta\lambda^0}{\sum\limits_{j\in G\text{-}\{0\}} (n_j\omega\Delta g_j + \Delta g_j^0) + \left|-n_0\omega|\Delta\lambda| + \Delta\lambda^0\right|}, & \text{if } s < 0 \\ \dfrac{n_0\omega|\Delta\lambda| + \Delta\lambda^0}{\sum\limits_{j\in G\text{-}\{0\}} (n_j\omega\Delta g_j + \Delta g_j^0) + \left|n_0\omega|\Delta\lambda| + \Delta\lambda^0\right|}, & \text{if } s > 0 \end{cases}$$

where $s := \left(\dfrac{\sum_{i\in G} n_i}{|G|} - n_0\right)$, $|G|$ refers to the number of elements in set G and Λ is defined as $\Lambda = -1$ when $s < 0$ or $\Lambda = 1$ when $s > 0$.

The changes in the reference direction are made in proportion to the current value of each linear objective function. The term Δg_i^0 is required to prevent the corresponding component of $\Delta\mathbf{g}$ becoming zero. The scale is estimated on the basis of the range given in the beginning, and updated later on as far as new information is received.

Parameter $\Delta\lambda$ is updated – as $\Delta\mathbf{g}$ – in proportion to its current value, and it is scaled in such a way that it is in relation to the sum of the new parameters.

Parameter λ is used to control the "relative meaning" of a quadratic objective function in proportion to linear objective functions. We have to vary a magnitude of this meaning, but we also need a mechanism which makes it possible to the DM to increase or decrease this relative meaning. Parameter s tells how much n_0 deviates from the average key presses when a new direction is specified. If $s > 0$, then the relative meaning of the quadratic function starts to decrease in a new search direction, and when $s < 0$, it starts to increase. The magnitude $|s|$ of the deviation from the average key presses specifies how strongly the DM would like to change the effect of the quadratic term.

Using the above mentioned function keys the DM can fully control the motion on the efficient frontier. When he/she reaches a "corner point" on the frontier, from which he/she cannot continue with the current reference direction, the system will inform the DM about the situation by blinking the "**num**"-key. This is a sign that the DM has to respecify a reference direction until a new direction to proceed is found.

Step 5. *The Values of Decision Variables and Goals*

Display the current values of decision variables and goals to the DM for evaluation. Provide him/her with a possibility to save the results. If the DM is satisfied with the results, stop; otherwise he/she can go back to

- Update the aspiration levels of the inflexible goals (**Step 0**),
- Reclassify the goals (**Step 1**)
- Respecify the ranges (**Step 2**) or
- Continue the Race (**Step 4**).

In Quadratic Pareto Race, the quadratic objective function plays an essential role. It is not controlled by means of a reference direction, but using a weighted-sum. The Race will start from its optimum or the neighborhood of the optimum. Initially, the direction which improves all linear objective functions simultaneously at the cost of the quadratic function is chosen. The DM may traverse across the efficient frontier in this direction until a boundary is reached. To reach the boundary – in this context – means that it is not possible anymore to improve all linear objective functions simultaneously. He/she can continue the search beyond this point using the current values of direction parameters $\Delta\mathbf{g}$ and $\Delta\lambda$ or to specify new values for them. However, he/she has to be willing to accept the fact that he cannot improve all linear functions beyond this point, simultaneously. The value of the quadratic function may improve or get worse.

5. A Numerical Example

Consider the following simple multiple objective quadratic-linear programming problem:

$$\text{Min} \qquad V(\mathbf{x}) = \frac{1}{2}\mathbf{x}'\begin{pmatrix} 6.26 & -0.76 & 3.72 & -1.91 & 1.27 \\ -0.76 & 8.30 & 1.36 & -1.17 & -1.75 \\ 3.72 & 1.36 & 2.78 & -1.17 & 0.88 \\ -1.91 & -1.17 & -1.17 & 2.72 & -1.12 \\ 1.27 & -1.75 & 0.88 & -1.12 & 3.84 \end{pmatrix}\mathbf{x}$$

$$\text{"Max"} \ \mathbf{l}(\mathbf{x}) = \begin{pmatrix} -1 & 19 & -2 & -3 & 17 \\ 0 & 5 & 4 & -1 & 19 \\ 5 & 20 & 3 & 16 & -1 \end{pmatrix}\mathbf{x}$$

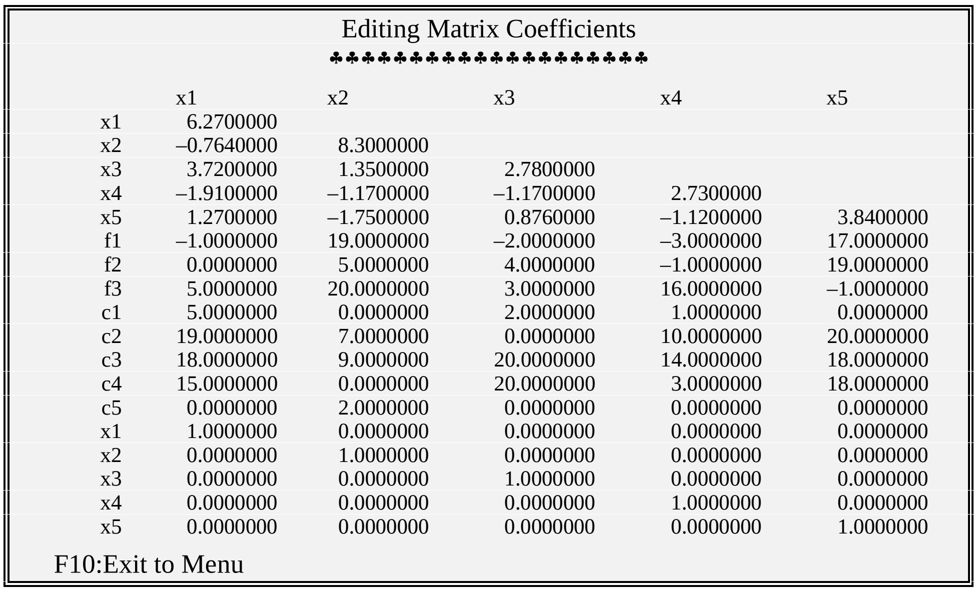
Subject to: (5.1)

$$\begin{pmatrix} 5 & 0 & 2 & 1 & 0 \\ 19 & 7 & 0 & 10 & 20 \\ 18 & 9 & 20 & 14 & 18 \\ 15 & 0 & 20 & 3 & 18 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}\mathbf{x} \le \begin{pmatrix} 76 \\ 72 \\ 93 \\ 89 \\ 31 \end{pmatrix} \text{and } \mathbf{x} \ge \begin{pmatrix} 0.10 \\ 0.10 \\ 0.10 \\ 0.10 \\ 0.10 \end{pmatrix}.$$

This problem is randomly generated by using the ADBASE package (see Steuer [1992]) on microcomputers to generate an MOLP-model, and then the quadratic term was randomly built using Microsoft EXCEL. The same idea to generate random multiple objective quadratic-linear programming problems is used in Korhonen and Yu [1997], in which the detailed description of the method is given.

To solve problem (5.1), we first name the rows and columns and then input numerical data on the screen entitled “Editing Matrix Coefficients” as shown in **Figure 1** (**Step 0**). Next we input right-side values for all goals of the model (see, **Figure 2**). Note that so far, we have not specified which goals are flexible (= objectives) and which inflexible (= constraints). Next, we will define the goals f_1, f_2, and f_3 flexible. In addition, the quadratic function called “Risk” is the fourth objective function. (The corresponding screen – similar to Pareto Race – is not shown here.) (**Step 1**) The goal values 100, 10, and 70 specified for the goals f_1, f_2, and f_3 in **Figure 2**, are now taken as the aspiration levels for linear objective functions.

Before starting Quadratic Pareto Race, we have to define the initial ranges for flexible goals. These ranges are used for the initialization of the race and for specification of reference directions for linear objectives (see, **Step 2**). The ranges are automatically updated during the race if necessary. The system will propose initial ranges as explained in **Step 2**. (The race is started with the ranges the system will propose for the linear functions. For the quadratic function, the bounds [0, 376.76] are used.)

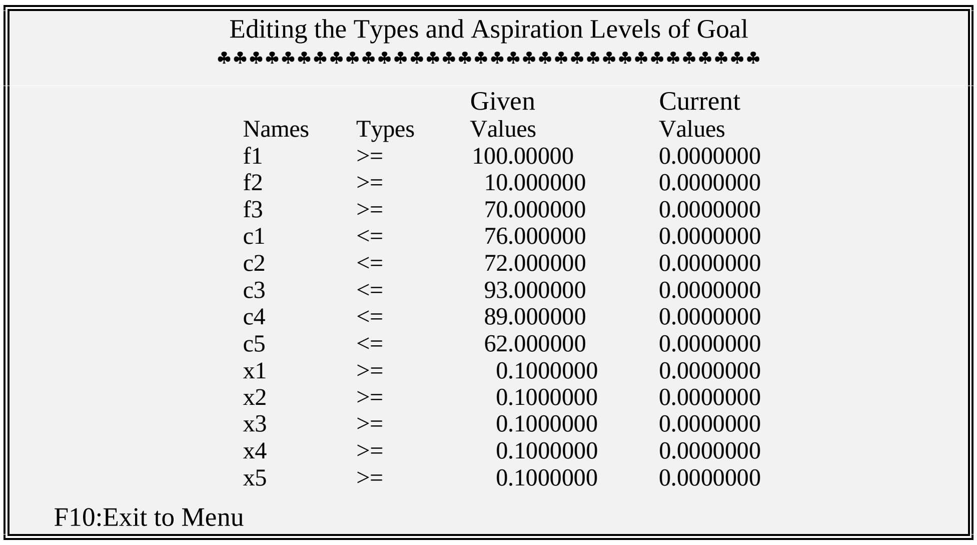


Editing Matrix Coefficients

	x1	x2	x3	x4	x5
x1	6.2700000				
x2	–0.7640000	8.3000000			
x3	3.7200000	1.3500000	2.7800000		
x4	–1.9100000	–1.1700000	–1.1700000	2.7300000	
x5	1.2700000	–1.7500000	0.8760000	–1.1200000	3.8400000
f1	–1.0000000	19.0000000	–2.0000000	–3.0000000	17.0000000
f2	0.0000000	5.0000000	4.0000000	–1.0000000	19.0000000
f3	5.0000000	20.0000000	3.0000000	16.0000000	–1.0000000
c1	5.0000000	0.0000000	2.0000000	1.0000000	0.0000000
c2	19.0000000	7.0000000	0.0000000	10.0000000	20.0000000
c3	18.0000000	9.0000000	20.0000000	14.0000000	18.0000000
c4	15.0000000	0.0000000	20.0000000	3.0000000	18.0000000
c5	0.0000000	2.0000000	0.0000000	0.0000000	0.0000000
x1	1.0000000	0.0000000	0.0000000	0.0000000	0.0000000
x2	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000
x3	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000
x4	0.0000000	0.0000000	0.0000000	1.0000000	0.0000000
x5	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000

F10:Exit to Menu

Figure 1. Data of the Model



Editing the Types and Aspiration Levels of Goal

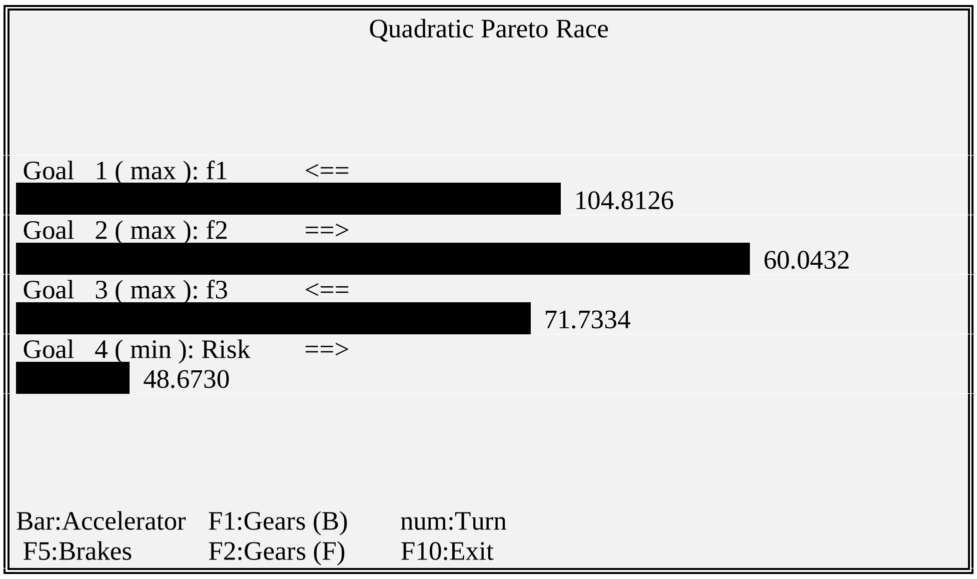
Names	Types	Given Values	Current Values
f1	>=	100.00000	0.0000000
f2	>=	10.000000	0.0000000
f3	>=	70.000000	0.0000000
c1	<=	76.000000	0.0000000
c2	<=	72.000000	0.0000000
c3	<=	93.000000	0.0000000
c4	<=	89.000000	0.0000000
c5	<=	62.000000	0.0000000
x1	>=	0.1000000	0.0000000
x2	>=	0.1000000	0.0000000
x3	>=	0.1000000	0.0000000
x4	>=	0.1000000	0.0000000
x5	>=	0.1000000	0.0000000

F10:Exit to Menu

Figure 2. The Aspiration Levels and Types of Goals

The search is started from the minimum of the quadratic function or close to the minimum. In this example, the minimum is not interesting, because the values of all objective functions are (approximately) zeroes. Therefore the corresponding screen is not displayed here. Starting from an initial solution, we may proceed in the direction specified by an initial reference direction (see, **Step 2**). The values of all linear objective functions are improved until the “boundary” solution shown in **Figure 3** is reached. Passing through this “boundary” solution means that it is impossible to find a feasible direction to improve all linear objectives simultaneously. However, the DM may continue the search using the current values of $\Delta\lambda$ and $\Delta\mathbf{g}$. The objective function f_2 still increases, whereas f_1 and f_3 start to decrease. Continuing the search further, we will reach a solution after that the objective functions f_1 and f_3 turn to increase, whereas f_2 start to decrease. Finally, we reach the solution where objective functions f_1 and f_3 have maximal values. At this point, the objective function f_4 has the worst value we have found. Now we are at the corner point (**Figure 4**). It means that we cannot continue the search with current values of parameters Δλ and Δ**g**. The system ask the DM to make a "turn" by blinking "num" – indicator at the bottom of the display.

Assume that the DM is not satisfied with this solution, and is willing to improve at least the objective function f_2. He/she can indicate this desire by pressing number "2" several times. The system will generate a new search direction. To move in this direction objective functions f_1, f_3, and f_4 go down and f_2 is going up until the optimal solution of objective function f_2 is reached (**Figure 5**). If the DM is not satisfied with the solution, he/she is welcome to respecify a new search direction. The DM can control the direction and speed of motion until he/she finds a solution for which no direction of improvement can be found.

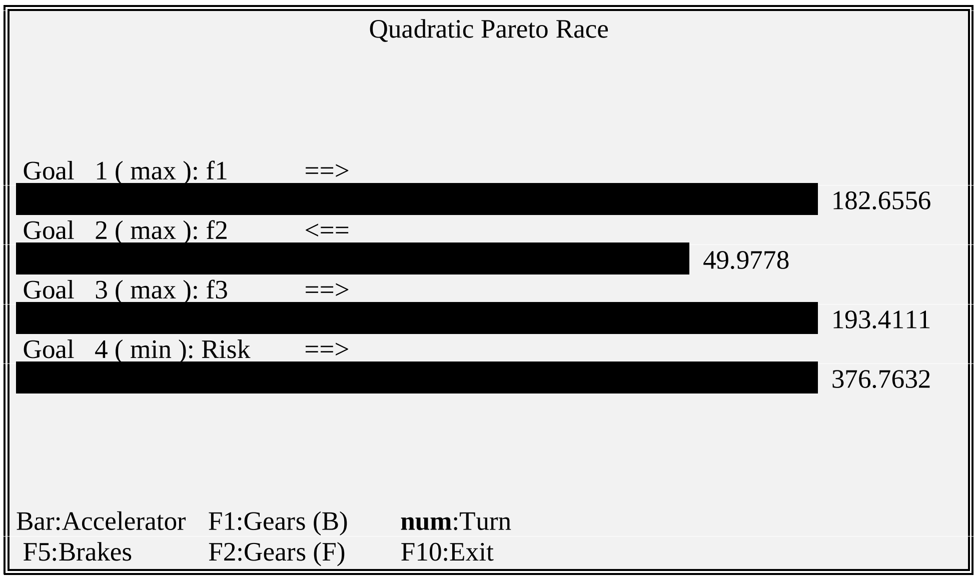


```
                         Quadratic Pareto Race

 Goal   1 ( max ): f1          <==
                                          104.8126
 Goal   2 ( max ): f2          ==>
                                                        60.0432
 Goal   3 ( max ): f3          <==
                                        71.7334
 Goal   4 ( min ): Risk        ==>
          48.6730

Bar:Accelerator   F1:Gears (B)     num:Turn
 F5:Brakes        F2:Gears (F)     F10:Exit
```

Figure 3. The Solution After it is Impossible to Improve All Linear Objectives

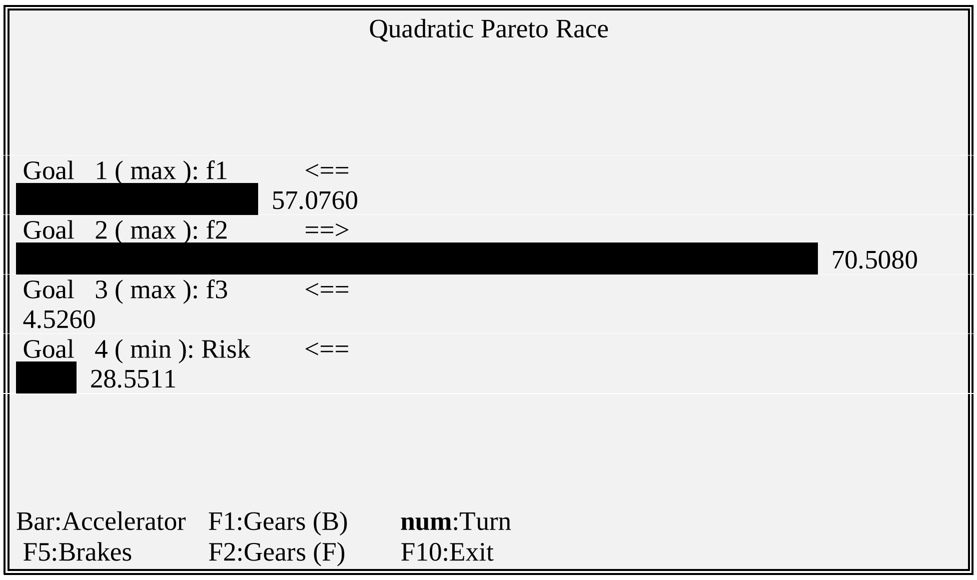


```
                         Quadratic Pareto Race

 Goal   1 ( max ): f1          ==>
                                                              182.6556
 Goal   2 ( max ): f2          <==
                                                      49.9778
 Goal   3 ( max ): f3          ==>
                                                              193.4111
 Goal   4 ( min ): Risk        ==>
                                                              376.7632

Bar:Accelerator   F1:Gears (B)     num:Turn
 F5:Brakes        F2:Gears (F)     F10:Exit
```

Figure 4. The Optimum of the First Objective Function



```
                         Quadratic Pareto Race

Goal   1 ( max ): f1          <==
                  57.0760
Goal   2 ( max ): f2          ==>
                                                        70.5080
Goal   3 ( max ): f3          <==
4.5260
Goal   4 ( min ): Risk        <==
     28.5511

Bar:Accelerator   F1:Gears (B)      num:Turn
 F5:Brakes        F2:Gears (F)      F10:Exit
```

Figure 5. The Optimum of the Second Objective Function

6. Conclusion

In this study we have described a dynamic and visual "free-search" type of a decision support system for solving multiple objective quadratic-linear programming problems. The system is implemented under the name Quadratic Pareto Race. Its purpose is to provide a DM with the possibility to freely explore the efficient frontier of a multiple objective quadratic-linear programming problem by controlling the direction and speed of motion.

A computer program written in TURBO PASCAL was developed to implement Quadratic Pareto Race on a microcomputer running under DOS-operating system. We call the whole program PORFO (Portfolio Selection), because a target user of the system is obviously the person who would like to solve multiple criteria portfolio problems, where one of the linear criteria is an expected return and the quadratic criterion is the variance of the expected return. The other linear criteria are describing the other features of the problem.

The current version of the program is capable of solving problems with a maximum of 20 variables with 50 goals, from which at most 10 may be flexible. The program consists of 6000 lines of code. The interface is based on the menus, spreadsheets, and visual interaction. "Visual interaction" means that the DM communicates with the system using visual representation in an interactive manner.

References

Boyan, M. (1995), "Use of Reference Points for Solving MONLP Problems". *European Journal of Operational Research***,** 80, pp. 193–203.

Cottle, R., Pang, J.-S., and Stone, R. (1992), *The Linear Complementarity Problem*, Academic Press.

Geoffrion, A. (1968), "Proper Efficiency and the Theory of Vector Maximization", *Journal of Mathematical Analysis and Applications*, 22, pp. 618–630.

Ignizio, J.P. (1983), "Generalized Goal Programming", *Computers and Operations Research*, 10, 277–289.

Jaszkiewicz, A., and Slowinski, R. (1994), "The Light Beam Search Interactive Procedure for Multiple Objective Non-linear Mathematical Programming", *The Lecture Notes at the Fifth International Summer School on Multicriteria Decision Aid.*

Kananen, I., Korhonen, P., Wallenius, H. and Wallenius, J. (1990), "Multiple Objective Analysis of Input–Output Models for Emergency Management", *Operations Research*, 38, pp. 193–201.

Korhonen, P. (1987), "VIG – A Visual Interactive Support System for Multiple Criteria Decision Making", *Belgian Journal of Operations Research, Statistics and Computer Science*, 27, pp. 3–15.

Korhonen, P., and Laakso, J. (1986), "A Visual Interactive Method for Solving the Multiple Criteria Problem", *European Journal of Operational Research*, 24, pp. 277–287.

Korhonen, P., and Soismaa, M. (1988), "A Multiple Criteria Model for Pricing Alcoholic Beverages", *European Journal of Operational Research*, 37, pp. 165–175.

Korhonen, P., and Yu, G.-Y. (1996), "A Reference Direction Approach to Multiple Objective Quadratic-linear Programming", Forthcoming in *European Journal of Operational Research*.

Korhonen, P., and Yu, G.-Y. (1997), "On Computing Objective Function Values in Multiple Objective Quadratic-Linear Programming", Forthcoming in *European Journal of Operational Research.*

Korhonen, P., and Wallenius, J. (1988)**,** "A Pareto Race", *Naval Research Logistics*, 35, pp. 615–623.

Lemke, C. (1968), "On Complementary Pivot Theory", in G. Dantzig and A. Veinott (Eds.): *Mathematics of the Decision Sciences*, Part I, Amer. Math. Society, pp. 95–114.

Saber, H. M., and Ravindran, A. (1993**),** "Nonlinear Goal Programming Theory and Practice: A Survey", *Computers & Operations Research*, 20, pp. 275–291.

Stam, A. and Kuula, M. (1991), "Selecting A Flexible Manufacturing System Using Multiple Criteria Analysis", *International Journal of Production Research*, 29, pp. 803–820.

Steuer, R. E. (1992), “Manual for the ADBASE Multiple Objective Linear Programming Package”, Department of Management Science, University of Georgia, Athens, Georgia, USA.

White, D. J. (1990), “A Bibliography on the Applications of Mathematical Programming Multiple-Objective Methods”, *The Journal of the Operational Research Society*, 41, pp. 669–691.