Control and Cybernetics

vol. **34** (2005) No. 2

A logarithmic barrier function method for solving nonlinear multiobjective programming problems

by

M. Tlas and B. Abdul Ghani

Scientific Services Department, Atomic Energy Commission
P.O. Box 6091, Damascus, Syria

Abstract: An interior point method for solving nonlinear multiobjective programming problems, over a convex set contained in the real space $\mathbf{R}^n$, has been developed in this paper. In this method a new strictly concave logarithmic barrier function has been suggested in order to transform the orginal problem into a sequence of unconstrained subproblems. These subproblems can be solved using Newton method for determining Newton's directions along which line searches are performed. It also has been proved that the number of iterations required by the suggested algorithm to converge to an ε-optimal solution is $O(m\,|\ln \varepsilon|)$, depending on predetermined error tolerance ε and the number of constraints m.

Keywords: multiobjective programming, interior methods, Newton method, barrier functions

1. Introduction

Karmarkar (1984) proposed an algorithm for solving linear programming problems in at most $o(n^{3.5}L)$ arithmetic operations. This proposal has led to a new cycle in optimization research. His algorithm is based on transfering the proposed problem to an equivalent one. Its objective function is a logarithmic potential function, which was immediately recognized as a penalized objective function. This method is found to be superior to the simplex method proposed by George Dantzig because it is faster and capable of treating problems of greater sizes.

G. de Ghellinck and J.Ph. Vial (1986) presented an algorithm for solving linear programming problem in polynomial time based on the Newton method and an other one in 1987 for solving a set of linear equations on the non-negative orthant. By this algorithm, the problem can made equivalent to a maximization of a simple concave function with a similar set of linear equations.

Renegar (1988), and Iri and Imai (1986) proposed algorithms for solving linear programming problems in polynomial time. The algorithms were based on two main ideas: the analytical centre concept and the Newton's method. Recently, many approaches to convex programming using the analytical centre idea and Newton's method have been reported by Mehrotra and Sun (1990) for convex quadratic programming and by Hertog, Roos, and Terlaky (1991, 1992) for linear programming and for a class of smooth convex programming problems.

Following these proposals it is useful to generalize these ideas of interior points technique to the domain of multiobjective programming. Therefore two algorithms are proposed for solving multiobjective programming problems based on the interior points and on the substitution rates concept.

An interior points method for solving general nonlinear multiobjective programming problems is developed in this paper. The method is based on a line search along the Newton's direction with respect to a certain strictly concave potential function (barrier function). It is proven that, after each line search, the potential function value is reduced by at least a certain constant amount. Using this result, it can be shown that the number of iterations required for the algorithm to converge to a good (compromise) solution is bounded.

Two main cases for solving multiobjective programming problems (MOP) were treated in this paper. The first case was based on certain and precise knowledge of substitution rates (Steuer, 1986). In this case, the decision-maker has to be capable of presenting his global preferences through a utility function $U(f_1, \ldots, f_p)$ (where p is the number of objective functions). This function is supposed to satisfy certain conditions (continuously differentiable, concave or quasi-concave, and strictly increasing in f) on the space $F(X)$ which is the image of the feasible set X by the objective functions $f_i(x)$, $(i = 1, \ldots, p)$. The substitution rates in this case are exactly the quotients of partial derivatives of $U(f_1, \ldots, f_p)$ or can simply be evaluated by the comparison method of Dyer (1973) and Steuer (1986). The second case is based on uncertain and imprecise knowledge of substitution rates. In this case also the utility function is known but practice has shown that the precise values of substitution rates are very difficult to obtain (Dyer, 1973, and Wallenius, 1975). It is easier to estimate them with intervals. Here, it is assumed that the decision-maker is capable of evaluating lower and upper bounds for all substitution rates (including the case when only their estimates are available) which allows for the construction of a polar cone in the objectives space $\mathbf{R}^p$. The cone will be sharp, polyhedral, and included in the positive orthant of $\mathbf{R}^p$.

2. The problem

The problem considered, in this work, is to maximize the objective functions $f_i(x)$ $(i = 1, \ldots, p)$ under the inequality constraints $g_i(x) \geq 0$ $(i = 1, \ldots, m)$ where the functions $f_i(x)$ $(i = 1, \ldots, p)$ and $g_i(x) \geq 0$ $\ (i = 1, \ldots, m)$ are concave with continuous first and second-order derivatives. Suppose that the interior of the feasible region $X = \{x \in \mathbf{R}^n \backslash g_i(x) \geq 0 \quad (i = 1, \ldots, m)\}$ (denoted

Int X) is non-empty and bounded in the real space $\mathbf{R}^n$. In mathematical form, the considered problem becomes:

$$\begin{aligned} &\textit{Maximize} \quad f_i = f_i(x) \quad (i = 1, \ldots, p)\,, && \textit{MOP} \\ &\textit{subject to} \quad g_i(x) \geq 0 \quad (i = 1, \ldots, m)\,. \end{aligned}$$

The previously mentioned cases of substitution rates will be now fully discussed.

3. Certain and precise knowledge of substitution rates

In this case, the decision-maker would suggest his global preferences through a utility function, $U(f_1, \ldots, f_p)$, which is continuously differentiable, concave within $F(X)$ and strictly increasing in f. The substitution rates will be given by the following equalities, see Steuer, 1986:

$$w_i = \frac{\dfrac{\partial U(f(x))}{\partial f_i}}{\dfrac{\partial U(f(x))}{\partial f_1}} \qquad (i = 1, \ldots, p)$$

where f_1 is considered as a reference criterion.

3.1. The concavity of the utility function on the decision space X

Let $V(x) = U(f_1(x), \ldots, f_p(x))$, where functions $f_i(x)$ $(i = 1, \ldots, p)$ are continuously differentiable and concave on X. If the function $U(f_1, \ldots, f_p$ is continuously differentiable, concave and strictly increasing on $F(X)$, then the function $V(x)$ is concave on X.

Consider the following relation:

$$\begin{tikzcd} \mathbf{R}^p \supseteq F(X) \arrow[dr, "V"'] & & X \subseteq \mathbf{R}^n \arrow[ll, "F"'] \arrow[dl, "U"] \\ & \mathbf{R} & \end{tikzcd}$$

where $V = UoF$ and

$$\nabla_x V(x) = \sum_{j=1}^{p} \frac{\partial U(f(x))}{\partial f_j}\, \nabla_x f_j(x)\,.$$

Since U is strictly increasing in f on $F(X)$, then $\frac{\partial U}{\partial f_j} > 0$ $(j = 1, \ldots, p)$. The functions $f_i(x)$ $(i = 1, \ldots, p)$ are concave on X. Therefore:

$$\forall\, x, x^* \in X;\ f_j(x^*) \leq f_j(x) + \nabla_x^T f_j(x)(x^* - x) \qquad (j = 1, \ldots, p)\,,$$

then:

$$\sum_{j=1}^{p} \frac{\partial U}{\partial f_j} \left(f_j(x^*) - f_j(x) \right) \leq \sum_{j=1}^{p} \frac{\partial U}{\partial f_j} \nabla_x^T f_j(x)(x^* - x) \, .$$

Using the last inequality, it can be found that:

$$\begin{aligned} \nabla_x^T V(x)(x^* - x) = & \left(\sum_{j=1}^{p} \frac{\partial U}{\partial f_j} \nabla_x^T f_j(x) \right) (x^* - x) \geq \sum_{j=1}^{p} \frac{\partial U}{\partial f_j} \left(f_j(x^*) - f_j(x) \right) \\ & = \nabla_x^T U(f(x))(f(x^*) - f(x)). \end{aligned}$$

As the function U is concave on $F(X)$, then:

$$\begin{aligned} \nabla_x^T V(x)(x^* - x) & \geq \nabla_x^T U(f(x))(f(x^*) - f(x)) \\ & \geq U(f(x^*)) - U(f(x)) = V(x^*) - V(x) \end{aligned}$$

so that $V(x^*) - V(x) \leq \nabla_x^T V(x)(x^* - x)$, which signifies that the function $V(x)$ is concave on X.

3.2. A logarithmic barrier function and its derivatives

Associate the following multiplicative barrier function with the primal problem (MOP):

$$\psi^k(x) = \left[\left(\sum_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k \right]^s \prod_{i=1}^{m} g_i(x) \qquad (k = 0, 1, \ldots)$$

where $x^k \in \mathrm{Int}\, X$, $z^k \in \mathbf{R}$ is an arbitrary lower bound so that $U(f(x^k)) > z^k$, k is the number of iterations and s is an integer number greater or equal to m (the number s plays the role of a weight). The function $\psi^k(x)$ is defined on X, strictly concave (Renegar, 1988), Hertog et al., 1991,1992), and close to zero when x goes to the boundary of X. It is difficult to find the first and second derivatives of $\psi^k(x)$, therefore, it is useful to use the first and second derivatives of $\ln \psi^k(x)$:

$$\begin{aligned} \phi^k(x) = \ln \psi^k(x) = & s \ln \left[\left(\sum_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k \right] \\ & + \sum_{i=1}^{m} \ln \, g_i(x) \qquad (k = 0, 1, \ldots) \, . \end{aligned}$$

This function is also defined only in the interior $\mathrm{Int}\, X$ of the feasible region X, twice continuously differentiable, strictly concave and close to $-\infty$ when x

goes to the boundary of X. Hence this logarithmic barrier function (potential function) attains the optimal value in its domain (for fixed z) at a unique point denoted x. The necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions for this optimum are:

$$\begin{aligned} & g_i(x) \geq 0 \qquad (i = 1, \ldots, m) \\ & \nabla U(f(x)) + \sum_{i=1}^{m} u_i \, \nabla g_i(x) = 0, \quad u_i \geq 0 \qquad (i = 1, \ldots, m) \\ & g_i(x) u_i = c \times \frac{U(f(x)) - z}{s} \qquad (c > 0) \end{aligned} \tag{1}$$

where u_i $(i = 1, \ldots, m)$ denote the dual variables of the problem (MOP).

Differentiating the function $\phi^k(x)$ gives:

$$\begin{aligned} G^k(x) = & \nabla \phi^k(x) = s \frac{\sum\limits_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k)}{\left(\sum\limits_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k} \\ & + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x) \qquad (k = 0, 1, \ldots). \end{aligned}$$

The vector $G^k(x)$ will simply be called the gradient of $\phi^k(x)$.

Further differentiation will yield:

$$\begin{aligned} H^k(x) = & \nabla^2 \phi^k(x) = - s \frac{\left(\sum\limits_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right) \left(\sum\limits_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right)^T}{\left[\left(\sum\limits_{i=1}^{p} w_i^k \, \nabla_x f_i(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k \right]^2} \\ & + \sum_{i=1}^{m} \left(\frac{1}{g_i(x)} \nabla^2 g_i(x) - \frac{1}{(g_i(x))^2} \nabla g_i(x) \, \nabla^T g_i(x) \right) \quad (k = 0, 1, \ldots). \end{aligned}$$

The matrix $H^k(x)$ will simply be called the Hessian matrix of $\phi^k(x)$.

The following algorithm is designed to work in the relative interior of the feasible set X and solving nonlinear multiobjective problems (MOP) with certain and precise knowledge of substitution rates.

3.3. Algorithm 1

Step 0. Initialization Let $k = 0$, error tolerance $\varepsilon > 0$, starting point $x^0 \in \operatorname{Int} X$, and lower bound $z^0 \in \mathbf{R}$ so that $U(f(x^0)) > z^0$.

Step 1. Calculating the potential function, gradient and Hessian

$$w_i^k \quad (i = 1, \ldots, p), \quad \phi^k(x^k), \quad G^k(x^k) \text{ and } H^k(x^k).$$

Step 2. Determining the feasible direction Find the unique solution of the following system of linear equations: $H^k(x^k)d^k = -G^k(x^k)$, where d^k denotes the solution of this linear system.

Step 3. Length of step Find the scalars:

$$\lambda^* = \arg\max_{\lambda \geq 0} \ \phi^k(x^k + \lambda d^k) \quad \text{and}$$

$$\lambda^k = \arg\max_{0 \leq \lambda \leq \lambda^*} \ U(f(x^k + \lambda d^k)) .$$

Step 4. Updating Define the new point $x^{k+1} = x^k + \lambda^k d^k$.

Step 5. Stopping criterion If $\left\|d^k\right\| < \varepsilon$ then stop. The point x^{k+1} is then considered as a compromise solution (efficient solution) in X of problem (MOP) and consequently the point $(f_1(x^{k+1}), \ldots, f_p(x^{k+1}))$ is considered as a non dominated solution in $F(X)$ of (MOP), else the new bound will be defined as follows: $z^{k+1} = z^k + \theta\ (U(f(x^{k+1}) - z^k)$ where θ is an arbitrary number chosen from the interval (0,1), $k = k + 1$ and go to Step 1.

Note: In practice it would probably be wise to choose $0 < \theta < 1$ initially large and then reduce it in later iterations if Newton's method begins having trouble in approximating centers, where the center is the point maximizing the function $\phi^k(x)$.

3.4. The easily demonstrable properties

1. The direction d^k, determined in Step 2 of the Algorithm 1, is a strict ascent direction of $\phi^k(x)$ at $x^k \in \text{Int}\ X$.
From Step 2 of the Algorithm 1, it can be seen that:

$$[G^k(x^k)]^T d^k = -[d^k]^T [H^k(x^k)]^T d^k .$$

From the strict concavity of $\phi^k(x)$, it follows that: $[d^k]^T[H^k(x^k)]^T d^k < 0$, so $[G^k(x^k)]^T d^k > 0$.

2. The point $x^{k+1} = x^k + \lambda^k d^k$ is feasible.
The proof can be completely derived from Step 3 of the Algorithm 1.

3.5. The reduction of the potential function value

It is known that:

$$\phi^k(x) = \ln\left[\left(\sum_{i=1}^{p} w_i^k\, \nabla_x f_i(x^k)\right)^T (x - x^k) + U(f(x^k)) - z^k\right]^s + \sum_{i=1}^{m} \ln g_i(x) \quad (k = 0, 1, \ldots) ,$$

$$\phi^{k+1}(x) = \ln\left[\left(\sum_{i=1}^{p} w_i^{k+1}\,\nabla_x f_i(x^{k+1})\right)^T (x - x^{k+1}) + U(f(x^{k+1})) - z^{k+1}\right]^s + \sum_{i=1}^{m} \ln g_i(x) \qquad (k = 0, 1, \ldots)$$

and $\phi^{k+1}(x) - \phi^k(x) =$

$$= s \ln \frac{\left(\sum_{i=1}^{p} w_i^{k+1}\,\nabla_x f_i(x^{k+1})\right)^T (x - x^{k+1}) + U(f(x^{k+1})) - z^{k+1}}{\left(\sum_{i=1}^{p} w_i^{k}\,\nabla_x f_i(x^{k})\right)^T (x - x^{k}) + U(f(x^{k})) - z^{k}}\,.$$

Now when $x = x^{k+1}$, then: $\phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) =$

$$= s \ln \frac{U(f(x^{k+1})) - z^{k+1}}{\left(\sum_{i=1}^{p} w_i^{k}\,\nabla_x f_i(x^{k})\right)^T (x^{k+1} - x^{k}) + U(f(x^{k})) - z^{k}}$$

and $\phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) =$

$$= s \ln \frac{(1-\theta)(U(f(x^{k+1})) - z^{k})}{\left(\sum_{i=1}^{p} w_i^{k}\,\nabla_x f_i(x^{k})\right)^T (x^{k+1} - x^{k}) + U(f(x^{k})) - z^{k}}\,.$$

As the function U is concave, then:

$$\phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) \le s\ln(1-\theta) \qquad \text{and} \qquad 0 < \theta < 1\,.$$

3.6. The available solution after $O(m\,|\ln\varepsilon|)$ iterations can be converted to an ε-optimal solution

Wolfe's formulation of the dual problem associated with the primal problem (MOP) is defined as follows:

$$\begin{aligned} &\textit{Minimize} && U(f(x)) + \sum_{i=1}^{m} u_i g_i(x) \\ &\textit{subject to} && \nabla U(f(x)) + \sum_{i=1}^{m} u_i\,\nabla g_i(x) = 0 \qquad\qquad DMOP \\ & && u_i \ge 0 \qquad (i = 1, \ldots, m) \end{aligned}$$

where the vectors x and u are the primal and dual variables, respectively.

It is well-known that, if x is a feasible solution of the primal problem (MOP) and (x^*, u) is a feasible solution of the dual problem $(DMOP)$, then:

$$U(f(x)) \leq U(f(x^*)) + \sum_{i=1}^{m} u_i g_i(x^*) . \tag{2}$$

Let z^* denote the value of the utility function at the optimal solution of problem (MOP) and let z^k be the value of the utility function at the point x^k, then:

$$\begin{aligned} \frac{z^* - z^{k+1}}{z^* - z^k} &= \frac{z^* - z^k - \theta(U(f(x^{k+1})) - z^k)}{z^* - z^k} \\ &= 1 - \theta \frac{U(f(x^{k+1})) - z^k}{z^* - z^k} \qquad (0 < \theta < 1) \end{aligned} \tag{3}$$

Using inequality (2), it can be seen that:

$$z^* \leq U(f(x^{k+1})) + \sum_{i=1}^{m} u_i g_i(x^{k+1}) \Rightarrow z^* - z^k \leq U(f(x^{k+1})) - z^k + \sum_{i=1}^{m} u_i g_i(x^{k+1})$$

Using (1), it follows that:

$$\begin{aligned} &z^* - z^k \leq U(f(x^{k+1})) - z^k + m\, \frac{U(f(x^{k+1})) - z^k}{s} \Rightarrow \\ &z^* - z^k \leq \left(1 + \frac{m}{s}\right) (U(f(x^{k+1})) - z^k) . \end{aligned}$$

Substitution of the last inequality into (3) gives:

$$\frac{z^* - z^{k+1}}{z^* - z^k} \leq 1 - \theta \left(1 + \tfrac{m}{s}\right)^{-1} = 1 - \theta \tfrac{s}{m+s} \Rightarrow$$

$$\left.\begin{aligned} z^* - z^{k+1} &\leq \left(1 - \theta \tfrac{s}{m+s}\right) (z^* - z^k) \\ z^* - z^k &\leq \left(1 - \theta \tfrac{s}{m+s}\right) (z^* - z^{k-1}) \\ z^* - z^{k-1} &\leq \left(1 - \theta \tfrac{s}{m+s}\right) (z^* - z^{k-2}) \\ &\vdots \\ z^* - z^1 &\leq \left(1 - \theta \tfrac{s}{m+s}\right) (z^* - z^0) \end{aligned}\right\} \Rightarrow z^* - z^{k+1} \leq \left(1 - \theta \tfrac{s}{m+s}\right)^{k+1} (z^* - z^0) .$$

As $z^* - U(f(x^{k+1})) \leq z^* - z^{k+1}$, so: $z^* - U(f(x^{k+1})) \leq (1 - \theta\frac{s}{m+s})^{k+1}(z^* - z^0)$. Since $\ln(1 - v) < -v$ for $v < 1$, then

$$\begin{aligned} &\ln(z^* - U(f(x^{k+1})) \leq (k + 1) \ln(1 - \theta \tfrac{s}{m+s}) + \ln(z^* - z^0) \Rightarrow \\ &\ln(z^* - U(f(x^{k+1})) \leq (k + 1)(-\theta \tfrac{s}{m+s}) + \ln(z^* - z^0) . \end{aligned}$$

The aim is to find the number of iterations K so that: $\ln(z^* - U(f(x^{k+1})) < \ln\varepsilon$ then:

$$-\theta(k+1) < \frac{m+s}{s}\ln\frac{\varepsilon}{z^*-z^0} \Rightarrow \theta(k+1) > -\frac{m+s}{s}\ln\frac{\varepsilon}{z^*-z^0} \Rightarrow$$
$$k+1 > -\frac{m+s}{\theta\, s}\ln\frac{\varepsilon}{z^*-z^0} \Rightarrow k > -1-\frac{m+s}{\theta\, s}\ln\frac{\varepsilon}{z^*-z^0}\,.$$

From this inequality, it can be seen that the number of iterations K for an ε-optimal solution is at most: $K = \left\lfloor -1 - \frac{m+s}{\theta\, s}\ln\frac{\varepsilon}{z^*-z^0}\right\rfloor + 1$ [1]. It is clear from this equality that the number of iterations K reduces when s goes to infinity and θ goes to one. Taking $z^* - z^0 < \frac{1}{\varepsilon}$ the number of iterations K described as follows: $K = O(m\,|\ln\varepsilon|)$.

3.7. Convergence of Algorithm 1

From Algorithm 1 it is found that $\left\|d^k\right\| = 0$, which implies

$$G^k(x^k) = 0 \Rightarrow$$
$$\frac{s}{U(f(x^k))-z^k}\sum_{i=1}^{p} w_i^k\,\nabla_x f_i(x^k) + \sum_{i=1}^{m}\frac{1}{g_i(x^k)}\,\nabla g_i(x^k) = 0\,,$$
$$\frac{1}{\frac{\partial U(f(x^k))}{\partial f_1}} \times \frac{s}{U(f(x^k))-z^k}\sum_{i=1}^{p}\frac{\partial U(f(x^k))}{\partial f_i}\,\nabla_x f_i(x^k)$$
$$+\sum_{i=1}^{m}\frac{1}{g_i(x^k)}\,\nabla g_i(x^k) = 0\,,$$
$$\frac{1}{\frac{\partial U(f(x^k))}{\partial f_1}} \times \frac{s}{U(f(x^k))-z^k}\,\nabla U(f(x^k)) + \sum_{i=1}^{m}\frac{1}{g_i(x^k)}\,\nabla g_i(x^k) = 0\,,$$
$$\nabla U(f(x^k)) + \frac{\partial U(f(x^k))}{\partial f_1} \times \frac{U(f(x^k))-z^k}{s}\sum_{i=1}^{m}\frac{1}{g_i(x^k)}\,\nabla g_i(x^k) = 0\,.$$

Taking

$$c = \frac{\partial U(f(x^k))}{\partial f_1} > 0 \text{ and } u_i = c \times \frac{U(f(x^k))-z^k}{s} \times \frac{1}{g_i(x^k)} \quad (i=1,\ldots,m)$$

we find:

$$\nabla U(f(x^k)) + \sum_{i=1}^{m} u_i\,\nabla g_i(x^k) = 0, \qquad g_i(x^k) > 0 \quad (i=1,\ldots,m)\,,$$
$$g_i(x^k)\,u_i = c \times \frac{U(f(x^k))-z^k}{s} \quad (i=1,\ldots,m),$$

[1] $\lfloor u \rfloor$ denotes the integer part of the real number.

which means that the accumulation point x^k satisfies the KKT conditions.

As the proposed algorithm creates a sequence of interior points $\{x^k\}_{k=0,1,\ldots}$ contained in $\operatorname{Int} X$ and converges to a solution satisfying the KKT conditions, under the assumptions used in the paper, then by the general theory of convergence (Minoux, 1983) we conclude that the accumulation point x^k which is found by the algorithm is an optimal solution of the problem (MOP) in X and consequentely the point $(f_1(x^k), \ldots, f_p(x^k))$ is a non-dominated point of the problem in $F(X)$.

3.8. Some notes on Algorithm 1

1. The convergence of Algorithm 1, to a good (compromise) solution, is assured under some hypotheses on the proposed utility function (strictly increasing, concave, and continuously differentiable).
2. The substitution rates provided by the decision-maker are exactly the quotients of partial derivatives of the utility function.
3. The substitution rates are utilized for determining the feasible direction.
4. The determination of the feasible direction will be calculated by the determination of the solution of a system of linear equations at every iteration.
5. The calculations are very simple to carry out.
6. The convergence to a satisfying (compromise) solution is relatively fast.

4. Uncertain and imprecise knowledge of substitution rates

In this case, the utility function is also known as in the first case but, as practice has shown, the values of substitution rates w_i $(i = 1, \ldots, p)$ are very difficult to obtain (Dyer, 1973, and Wallenius, 1975). Then it is very easy and preferable for the decision-maker to evaluate lower and upper bounds for all substitution rates w_i $(i = 2, \ldots, p)$ and $w_1 = 1$ (this, by the way, is well-suited to the situations when the the substitution rates are not known with high accuracy).

Taking into consideration the function f_1 as a reference criterion, all the above mentioned proposals allow the construction of a polar cone in the space of objective functions $\mathbf{R}^p$.

Let m_i and M_i $(i = 2, \ldots, p)$ be the lower and upper bounds proposed by the decision-maker, respectively, so that: $0 < m_i \leq w_i \leq M_i$ $(i = 2, \ldots, p)$ and $w_1 = 1$. Define the following polar cone:

$$\Lambda = \{z \in \mathbf{R}^p \backslash z = \sum_{j=1}^{2(p-1)} \lambda_j q^j, \quad \lambda_j \geq 0 \quad (j = 1, \ldots, 2(p-1))\}$$

in the space $\mathbf{R}^p$ generated by the Cartesian products of intervals $[m_j, M_j]$ $(j = 2, \ldots, p)$ where q^j $(j = 2, \ldots, p)$ are the generators of this cone. The generators of Λ are just the rows of the following matrix:

$$A = \begin{bmatrix} 1 & m_2 & 0 & 0 & \cdots & 0 & 0 \\ 1 & M_2 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & m_3 & 0 & \cdots & 0 & 0 \\ 1 & 0 & M_3 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & 0 & 0 & \cdots & 0 & m_p \\ 1 & 0 & 0 & 0 & \cdots & 0 & M_p \end{bmatrix} \in \mathbf{R}^{2(p-1)\times p}.$$

The following algorithm is designed to work in the relative interior of the feasible set X and to solve the nonlinear multiobjective problems (MOP) with uncertain and imprecise knowledge of substitution rates.

4.1. Algorithm 2

Step 0. Initialization Let k =0, error tolerance $\varepsilon > 0$, starting point $x^0 \in \operatorname{Int} X$, and lower bound $z^0 \in \mathbf{R}$ so that $U(f(x^0)) > z^0$.

Step 1. Calculating the potential function, gradient, Hessian and polar cone Let the decision-maker evaluate the lower and upper bounds m_j^k and M_j^k $(j = 2, \dots, p)$ for all substitution rates w_j^k $(j = 2, \dots, p)$, where $0 < m_j^k \leq w_j^k \leq M_j^k$ $(j = 2, \dots, p)$ and $w_1^k = 1$. Construct the matrix:

$$A^k = \begin{bmatrix} 1 & m_2^k & 0 & 0 & \cdots & 0 & 0 \\ 1 & M_2^k & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & m_3^k & 0 & \cdots & 0 & 0 \\ 1 & 0 & M_3^k & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & 0 & 0 & \cdots & 0 & m_p^k \\ 1 & 0 & 0 & 0 & \cdots & 0 & M_p^k \end{bmatrix} \in \mathbf{R}^{2(p-1)\times p}.$$

Let $q^k(i)$ be the i-th row of the matrix A^k $(i = 1, \dots, 2(p-1))$,

$$\begin{aligned} \phi_i^k(x) = s \ln \left[\left(\sum_{j=1}^{p} q_j^k(i) \, \nabla_x f_j(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k \right] \\ + \sum_{j=1}^{m} \ln g_j(x) \qquad (i = 1, \dots, 2(p-1)), \end{aligned}$$

$$\begin{aligned} G_i^k(x) = s \frac{\sum\limits_{j=1}^{p} q_j^k(i) \, \nabla_x f_j(x^k)}{\left(\sum\limits_{j=1}^{p} q_j^k(i) \, \nabla_x f_j(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k} \\ + \sum_{j=1}^{m} \frac{1}{g_j(x)} \nabla g_j(x) \quad (i = 1, \dots, 2(p-1)), \end{aligned}$$

$$H_i^k(x) = -\, s\, \frac{\times \left(\sum\limits_{j=1}^{p} q_j^k(i)\, \nabla_x f_j(x^k) \right) \left(\sum\limits_{i=1}^{p} q_j^k(i)\, \nabla_x f_j(x^k) \right)^T}{\left[\left(\sum\limits_{j=1}^{p} q_j^k(i)\, \nabla_x f_j(x^k) \right)^T (x - x^k) + U(f(x^k)) - z^k \right]^2}$$
$$+ \sum_{j=1}^{m} \left(\frac{1}{g_j(x)}\, \nabla^2 g_j(x) - \frac{1}{(g_j(x))^2}\, \nabla g_j(x)\, \nabla^T g_j(x) \right) \quad (i = 1, \ldots, 2(p-1))\,.$$

Step 2. Determining the feasible direction Find the unique solutions of the following systems of linear equations:

$$H_i^k(x^k)\, d = -G_i^k(x^k)\ (i = 1, \ldots, 2(p-1))\,.$$

Let $d^k(i)$ denote the solution of the system number i.

Step 3. Length of step Find the scalars:

$$\lambda^*(i) = \underset{\lambda \geq 0}{\arg\max}\ \phi_i^k(x^k + \lambda d^k(i)) \quad \text{and}$$
$$\lambda^k(i) = \underset{0 \leq \lambda \leq \lambda^*(i)}{\arg\max}\ U(f(x^k + \lambda d^k(i))) \qquad (i = 1, \ldots, 2(p-1))\,.$$

Step 4. Updating Define the new points $v(i) = x^k + \lambda^k(i) d^k(i)$ $(i = 1, \ldots, 2(p-1))$. Choose $u^* \in \{1, 2, \ldots, 2(p-1)\}$ such that $U(f(v(u^*))) \geq U(f(v(i)))$ $(i = 1, \ldots, 2(p-1))$. Put $x^{k+1} = v(u^*)$.

Step 5. Stopping criterion If $\left\| d^k(u^*) \right\| < \varepsilon$ then stop. The point x^{k+1} is then considered as a satisfying or compromise solution (efficient solution) in X of problem (MOP) and consequently the point $(f_1(x^{k+1}), \ldots, f_p(x^{k+1}))$ is considered as a nondominated solution in $F(X)$ of (MOP), else define the new bound as follows: $z^{k+1} = z^k + \theta\, (U(f(x^{k+1}) - z^k)$, where $0 < \theta < 1$, $k = k + 1$ and go to Step 1.

4.2. Some notes on Algorithm 2

1. This method does not need precise knowledge of substitution rates. It is enough to know lower and upper bounds for all substitution rates.
2. The convergence, to a satisfiying (compromise) solution, is assured.
3. The calculation of steps is very simple.
4. The improvement direction is determined by the lower and upper bounds of substitution rates and by solving systems of linear equations.

5. Numerical results

A multiobjective example has been solved. The example was chosen to satisfy the assumptions used in the paper with the objective functions $f_1(x)$, $f_2(x)$ and the constraint functions $g_i(x)$ $(i = 1, \ldots, 4)$ concave with continous first and second order-derivatives. The utility function $U(f_1, f_2)$ is continously differentiable, concave withn $F(X)$ and strictly increasing in f.

$$\begin{aligned}
&\textit{Maximize} \quad U(f_1(x), f_2(x)) \\
&g_1(x) = x_1 + x_2 - 8 \geq 0 \\
&g_2(x) = -x_1 - x_2 + 20 \geq 0 \\
&g_3(x) = x_1 - 2 \geq 0 \\
&g_4(x) = x_2 - 3 \geq 0 \\
&f_1(x) = -(x_1 - 1) \\
&f_2(x) = -(x_2 - 2) \\
&U(f_1, f_2) = -f_1^2 - f_2^2
\end{aligned}$$

The example will now be solved using the Algorithm 1 (i.e., certain knowledge of substitution rates) with starting point $x^0 = (9, 7)^T$, $s = 16$ and $z^0 = -100$.

Table 1. Results from Algorithm 1 for the example

k	z^k	x^k	d^k	λ^k	x^{k+1}	$U(x^{k+1})$	θ
0	-100	9 7	-8.107589 2.701107	0.703216	3.298614 8.899462	-52.886202	
1	-57.597582	3.298614 8.899462	1.34596 -1.95943	1.84484	5.781695 5.284627	-33.653382	0.9
2	-36.047802	5.781695 5.284627	-2.9517 1.27273	0.961414	2.943889 6.508247	-24.102995	0.9
3	-25.297476	2.943889 6.508247	1.07758 -0.964711	1.07774	4.105240 5.468539	-21.673278	0.9
4	-22.035698	4.105240 5.468539	0.376034 -0.653433	1.93319	4.832185 4.205329	-19.549118	0.9
5	-19.797776	4.832185 4.205329	-0.933974 1.19822	0.405839	4.453142 4.691613	-19.168970	0.9
6	-19.483373	4.453142 4.691613	-1.0374 0.938943	0.538876	3.894112 5.197587	-18.600447	0.5
7	-19.04191	3.894112 5.197587	0.336447 -0.686932	2.09001	4.597290 3.761892	-16.044759	0.5
8	-17.543335	4.597290 3.761892	-1.70956 1.60948	0.601132	3.569619 4.729402	-14.052577	0.5
9	-15.797956	3.569619 4.729402	0.543029 8 -0.859918	0.922009	4.070297 3.936550	-13.176950	0.5
10	-14.487453	4.070297 3.936550	-1.92906 1.93568	0.291137	**3.508676** **4.500098**	**-12.543945**	

The solution obtained using Algorithm 1 is $x^* = (3.508676, 4.500098)^T$ (for $s = 16$, the values of objective functions at this point are $f_1(x^*) = -2.508676$ and $f_2(x^*) = -2.500098$) and the value of the utility function at this point is $U(x^*) = -12.543945$. The optimal solution computed by the method of Geoffrion, Dyer and Feinberg (see Steuer, 1986) is $x^* = (3.5, 4.5)^T$ and the value of the utility function at this point is -12.5.

The same example will now be solved using the Algorithm 2 (i.e., uncertain knowledge of substitution rates) with starting point $x^0 = (9, 7)^T$, $s = 16$, $z^0 = -100$ and $w^0 = (1, 0.625)^T$.

Illustration of functioning of Algorithm 2:

$q^0(1) = (1, 0.5)^T$	$q^0(2) = (1, 1)^T$
$d^0(1) = (-8.43893, 3.54073)^T$	$d^0(2) = (-6.001954, -0.245536)^T$
$\lambda^0(1) = 0.594703$	$\lambda^0(2) = 0.908809$
$v(1) = (3.981343, 9.105683)^T$	$v(2) = (3.545370, 6.776855)^T$
$U(v(1)) = -59.379137$	$U(v(2)) = -29.297252$

- -

$x^1 = (3.545370, 6.776855)^T$
$U(x^1) = -29.297252$

- -

$w^1 = (1, 1.876684)^T, \ \theta = 0.9, \ z^1 = -36.367527$

$q^1(1) = (1, 1)^T$	$q^1(2) = (1, 2)^T$
$d^1(1) = (0.379907, -3.18451)^T$	$d^1(2) = (1.73729, -3.48318)^T$
$\lambda^1(1) = 0.577112$	$\lambda^1(2) = 0.806338$
$v(1) = (3.764619, 4.939036)^T$	$v(2) = (4.946213, 3.968235)^T$
$U(v(1)) = -16.281051$	$U(v(2)) = -19.446546$

- -

$x^2 = (3.764619, 4.939036)^T$
$U(x^2) = -16.281051$

- -

$w^2 = (1, 1.063089)^T, \ \theta = 0.9, \ z^2 = -18.289699$

$q^2(1) = (1, 0.5)^T$	$q^2(2) = (1, 1.5)^T$
$d^2(1) = (-2.43983, 2.32887)^T$	$d^2(2) = (1.73705, -2.13573)^T$
$\lambda^2(1) = 0$	$\lambda^2(2) = 0.194586$
This means $d^2(1)$is not feasible	$v(2) = (4.102625, 4.523453)^T$
direction	$U(v(2)) = -15.994097$

- -

$x^3 = (4.102625, 4.523453)^T$
$U(x^3) = -15.994097$

- -

$w^3 = (1, 0.813328)^T,\ \theta = 0.9,\ z^3 = -16.223657$

$q^3(1) = (1, 0.5)^T$
$d^3(1) = (-0.863161, 1.27463)^T$
$\lambda^3(1) = 0$

$q^3(2) = (1, 1)^T$
$d^3(2) = (-0.419999, 0.199151)^T$
$\lambda^3(2) = 2.59149$
$v(2) = (3.014202, 5.039551)^T$
$U(v(2)) = -13.29588$

- -

$x^4 = (3.014202, 5.039551)^T$
$U(x^4) = -13.29588$

- -

$w^4 = (1, 1.50906)^T,\ \theta = 0.9,\ z^4 = -13.588658$

$q^4(1) = (1, 1)^T$
$d^4(1) = (0.395569, -0.462255)^T$
$\lambda^4(1) = 0$

$q^4(2) = (1, 2)^T$
$d^4(2) = (0.403449, -0.348167)^T$
$\lambda^4(2) = 0.864968$
$v(2) = (3.363172, 4.738398)^T$
$U(v(2)) = -13.083406$

- -

$x^5 = (3.363172, 4.738398)^T$
$U(x^5) = -13.083406$

- -

$w^5 = (1, 1.158781)^T,\ \theta = 0.9,\ z^5 = -13.133931$

$q^5(1) = (1, 1)^T$
$d^5(1) = (0.163878, -0.211998)^T$
$\lambda^5(1) = 1.92294$
$v(1) = (3.678210, 4.330739)^T$
$U(v(1)) = -12.605153$

$q^5(2) = (1, 1.5)^T$
$d^1(2) = (0.408839, -0.306208)^T$
$\lambda^5(2) = 0$

- -

$x^6 = (3.678210, 4.330739)^T$
$U(x^6) = -12.605153$

- -

$w^6 = (1, 0.87026)^T,\ \theta = 0.9,\ z^6 = -12.658031$

$q^6(1) = (1, 0.5)^T$
$d^6(1) = (-0.114788, 0.123788)^T$
$\lambda^6(1) = 0.663474$
$v(1) = (3.602051, 4.412869)^T$
$U(v(1)) = -12.592606$

$q^6(2) = (1, 1)^T$
$d^6(2) = (-0.175578, 0.165129)^T$
$\lambda^6(2) = 0.509191$
$v(2) = (3.588807, 4.414821)^T$
$U(v(2)) = -12.533282$

- -

$x^7 = (3.588807, 4.414821)^T$
$U(x^7) = -12.533282$

- -

$w^7 = (1, 0.932793)^T,\ \theta = 0.9,\ z^7 = -12.545757$

$q^7(1) = (1, 0.5)^T$	$q^7(2) = (1, 1)^T$
$d^7(1) = (-0.028588, 0.032225)^T$	$d^7(2) = (-0.089551, 0.083922)^T$
$\lambda^7(1) = 0$	$\lambda^7(2) = 0.47625$
	$v(2) = (3.546158, 4.454789)^T$
	$U(v(2)) = -12.50891$

- -

$x^8 = (3.546158, 4.454789)^T$
$U(x^8) = -12.50891$

- -

$w^8 = (1, 0.964115)^T,\ \theta = 0.9,\ z^8 = -12.512595$

$q^8(1) = (1, 0.5)^T$	$q^8(2) = (1, 1)^T$
$d^8(1) = (-0.008318, 0.009266)^T$	$d^8(2) = (-0.046360, 0.044927)^T$
$\lambda^8(1) = 0$	$\lambda^8(2) = 0.470701$
	$v(2) = (3.524336, 4.475936)^T$
	$U(v(2)) = -12.502531$

- -

$x^9 = (3.524336, 4.475936)^T$
$U(x^9) = -12.502531$

- -

$w^9 = (1, 0.9808267)^T,\ \theta = 0.9,\ z^9 = -12.503537$

$q^9(1) = (1, 0.5)^T$	$q^9(2) = (1, 1)^T$
$d^9(1) = (-0.002283, 0.002555)^T$	$d^9(2) = (-0.024403, 0.023986)^T$
$\lambda^9(1) = 0$	$\lambda^9(2) = 0.472045$
	$v(2) = (3.512817, 4.487258)^T$
	$U(v(2)) = -12.500702$

- -

$x^{10} = (3.512817, 4.487258)^T$
$U(x^{10}) = -12.500702$

- -

$w^{10} = (1, 0.989829)^T,\ \theta = 0.9,\ z^{10} = -12.500986$

$q^{10}(1) = (1, 0.5)^T$	$q^{10}(2) = (1, 1)^T$
$d^{10}(1) = (-0.000644, 0.000719)^T$	$d^{10}(2) = (-0.012836, 0.012722)^T$
$\lambda^{10}(1) = 0$	$\lambda^{10}(2) = 0.472464$
	$v(2) = (3.506752, 4.493269)^T$
	$U(v(2)) = -12.500196$

- -

$x^{11} = (3.506752, 4.493269)^T U(x^{11}) = -12.500196$

The solution obtained using Algorithm 2 is $x^* = (3.506752, 4.493269)^T$ (for $s = 16$, the values of objective functions at this point are $f_1(x^*) = -2.506752$ and $f_2(x^*) = -2.493269$) and the value of the utility function at this point is $U(x^*) = -12.500196$.

6. Conclusion

An interior point algorithm for nonlinear multiobjective programming was successfully developed using a logarithmic barrier function method. Each iteration, in the suggested algorithm, consists of a Newton step followed by a reduction in the value of the logarithmic barrier function. The algorithm requires a small (almost constant) number of iterations to solve the multiobjective programming problems. Further investigations are needed to compare the efficiency of the proposed algorithm with other methods in the domain of multiobjective programming.

Acknowledgment

Authors would like to express their thanks to the Director General of AECS Prof. I. Othman for his continuous encouragement, guidance and support. They also thank Dr A. Al-Mohamad for his revision. Special thanks go to the reviewers for their constructive suggestions aimed at improving the quality of this paper.

References

De Ghellinck, G. and Vial, J. Ph. (1986) A polynomial Newton method for linear programming. *Algorithmica* **1**, 425–453.

De Ghellinck, G. and Vial, J. Ph. (1987) An extention of Karmarkar's algorithm for solving a system of linear homogenous equations on the simplex. *Mathematical Programming* **39**, 79–92.

Dyer, J. (1973) A time-sharing computer program for the solution of the multiple criteria problem. *Management Science* **19** (12), 1379-1383.

Hertog, D., Roos, C. and Terlaky, T. (1992) A large-step analytic center method for a class of smooth convex programming problems. *SIAM J. Optimization* **2** (1), 55-70.

Hertog, D., Roos, C. and Terlaky, T. (1991) A potential-reduction variant of Renegar's short- following method for linear programming. *Linear Algebra and its Applications* **152**, 43-68.

Iri, M. and Imai, H. (1986) A multiplicative barrier function method for linear programming. *Algorithmica* **1**, 455-482.

Karmarkar, N. (1984) A new polynomial time algorithm for linear programming. *Combinatorica* **4**, 373-395.

Mehrotra, S. and Sun, J. (1990) An algorithm for convex quadratic programming that requires $O(n^{3.5}L)$ arithmetic operations. *Mathematics of Operations Research* **15** (2), 342-363.

Minoux, M. (1983) *Programmation mathématique. Théorie et algorithmes.* Dunod, Paris.

Renegar, J. (1988) A polynomial-time algorithm based on Newton's method for linear Programming. *Mathematical Programming* **40**, 59-93.

Steuer, R. (1986) *Multiple Criteria Optimization: Theory, Computation, and Application. Wiley series in probability and mathematical statistics applied*, John Wiley & Sons.

Wallenius, J. (1975) Comparative evaluation of some interactive approaches to multicriterion optimization. *Management Science* **21** (12), 1387-1396.