Article

Using the analytic hierarchy process in an interactive interior point algorithm for mathematical multiple-objective nonlinear programming problems

M. Tlas
Scientific Services Department, Atomic Energy Commission, P. O. Box 6091, Damascus, Syria
E-mail: pscientific31@aec.org.sy

Received 30 June 2024; *Accepted 26 July 2024*; *Published online 25 August 2024*; *Published 1 June 2025*

Abstract
An interactive interior point method for solving multiple-objective nonlinear programming problems has been proposed. The method uses a single-objective nonlinear variant based on both logarithmic barrier function and Newton's method in order to generate, at each iterate, interior search directions. New feasible points are found along these directions which will be later used for deriving best-approximation to the gradient of the implicitly-known utility function at the current iterate. Using this approximate gradient, a single feasible interior direction for the implicitly-utility function could be found by solving a set of linear equations. It may be easily taken an interior step from the current iterate to the next one along this feasible direction. During the execution of the algorithm, a sequence of interior points will be generated. It has been proved that this sequence converges to an $\varepsilon-$optimal solution, where ε is a predetermined error tolerance known a priori. A numerical multiobjective example is illustrated using this algorithm.

Keywords multiobjective mathematical programming; multi-criteria optimization; interactive methods; interior point methods; barrier function; Newton's method; analytic hierarchy process.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology /rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

In this paper, an interior point algorithm for solving general nonlinear convex programming problems is presented. The algorithm generalizes a logarithmic barrier function proposed by (Renegar, 1988) for solving linear programming problems in polynomial time. This algorithm is based on both the analytical center concept and the Newton's method. Recently, many important approaches to convex programming based on the idea of an analytic center are also proposed by (Mehrotra and Sun, 1990) for convex quadratic programming and the distinct works of (Hertog et all.,1991, 1992) for linear programming and for a class of smooth convex programming problems. Our work is mainly influenced by the works of (Renegar, 1988; Mehrotra and Sun, 1990; Hertog et al., 1991, 1992; Tlas, 2013, 2024) in their study of analytical center methods for linear and nonlinear programming.

In this proposed method, the line search is performed along the Newton's direction which can be found by solving a system of linear equations in polynomial time, with respect to a certain strictly concave potential function in each iterate. It is proven that, after a line search the potential function value decreases with at least a certain constant. Using this result, it can be proved that the number of iterations required by the algorithm to converge to an ε-optimal solution is at most $O(m\,|ln\varepsilon|)$ iterations where m denotes the number of constraints of problem and ε is the predetermined error tolerance.

In addition, it is useful to generalize this interior point technique for solving a single nonlinear objective function to the domain of multiobjective nonlinear programming. Therefore, an interactive interior point algorithm is proposed to solve nonlinear multiple-objective programming problems based on both logarithmic barrier functions and approximate gradient.

The method uses the single-objective nonlinear variant proposed before in order to generate, at each iterate, interior search directions. New feasible points are found along these directions which will be later used for deriving best-approximation to the gradient of the implicitly-known utility function at the current iterate. Using this approximate gradient, a single feasible interior direction for the implicitly-utility function could be found by solving a set of linear equations by Gaussian elimination method. It may be easily taken an interior step from the current iterate to the next one along this feasible direction. During the execution of the algorithm, a sequence of interior points will be generated. It has been proved that this sequence converges to an $\varepsilon-$ optimal solution, where ε is a predetermined error tolerance known a priori.

The multiobjective nonlinear programming problem is ambiguous since, usually, the objective functions are conflicting and pursuing the optimum with respect to each objective. This will lead to different solutions. The ambiguity may be solved by introducing a utility function (or preference function) defined over the space of objectives. It is supposed that the decision-maker is capable of presenting his global preferences through this function, which is not necessarily being explicitly known but it is supposed to satisfy certain conditions as being continuously differentiable, concave and strictly increasing on the objective space in order to ensure the global convergence and to reach a global optimum. If the utility function is explicitly available, then it is easy to find the approximate gradient through the values of the utility function and the values of the objective functions at the current iteration. In the opposite case, when the utility function is implicitly known the approximate gradient could be found through the values of the objective functions and the analytic hierarchy process (AHP) (Zhang, 2019) technique at the current iteration. For more details about the AHP technique, the reader is referred to consult the following references: (Saaty, 1988; Arbel, 1994; Arbel and Oren, 1996).

2 Statement of the Nonlinear Programming Problem (*NLP*)

Consider the nonlinear programming problem (*NLP*) given in a standard form through:

$$\begin{aligned} &Maximize\ \ f(x) \\ &Subject\ to\ \ g_i(x) \geq 0\ \ (i=1,\dots,m) \end{aligned}$$

$$(NLP)$$

Where $x \in \mathbf{R}^n$, n is the number of unknown (decision) variables and m is the number of constraints. The functions $f(x)$ and $g_i(x) \geq 0$ $(i=1,\dots,m)$ are concave with continuous first and second-order derivatives.

It is supposed that, the interior of the feasible region $X = \left\{ x \in \mathbf{R}^n \setminus g_i(x) \geq 0 \;\; (i = 1,\dots,m) \right\}$, denoted as *Int* (*X*) is non-empty, compact and convex in the real space $\mathbf{R}^n$.

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

$$\begin{aligned} &Minimize \;\; f(x) + \sum_{i=1}^{m} u_i g_i(x) \\ &Subject\ to \;\; \nabla f(x) + \sum_{i=1}^{m} u_i \nabla g_i(x) = 0 \qquad\qquad (DNLP) \\ &\qquad u_i \geq 0 \;\; (i = 1,\dots,m) \end{aligned}$$

Where the vectors *x* and *u* are primal and dual variables consequently. It is well-known that, if *x* is a feasible solution of the primal problem (NLP) and $(\bar{x}, u)$ is a feasible solution of the dual problem $(DNLP)$, then the following inequality:

$$f(x) \leq f(\bar{x}) + \sum_{i=1}^{m} u_i g_i(\bar{x}) \tag{1}$$

is correct.

2.1 Logarithmic barrier function and its derivatives

The following multiplicative barrier function is associated with the primal problem (NLP) as follows:

$$\psi^k(x) = (f(x) - z^k)^{m+k} \prod_{i=1}^{m} g_i(x) \qquad (k = 0,1,\dots)$$

Where, z^k is a lower bound for the optimal value z^* and *k* is the number of iteration. This function is inspired from the work of Iri and Imai (1986) with some modifications. The function $\psi^k(x)$ is defined on the feasible region *X*, strictly concave, $\psi^k(x) > 0$ on *Int* (*X*) and $\psi^k(x)$ tends to zero when *x* goes to the boundary of *X*. It is difficult to derive the first and second derivatives of $\psi^k(x)$, for this reason it is interesting to see the first and second derivatives of $ln(\psi^k(x))$ which have nice expressions, as follows:

$$\phi^k(x) = ln(\psi^k(x))$$

$$\phi^k(x) = (m+k) ln(f(x) - z^k) + \sum_{i=1}^{m} ln(g_i(x)) \;\; (k = 0,1,\dots).$$

The function $\phi^k(x)$ is also defined only on the interior *Int* (*X*) of the feasible region *X*, twice-continuously differentiable, strictly concave and $\phi^k(x)$ tends to $-\infty$ when *x* goes to the boundary of *X*. Hence this logarithmic barrier function (potential function) achieves the maximal value in its domain (for fixed z^k) at a unique point denoted x^k . The necessary and sufficient Karush-Kuhn-Tucker (*KKT*) conditions for this maximal are:

$$g_i(x^k) \geq 0 \quad (i = 1,\dots,m),$$
$$\nabla f(x^k) + \sum_{i=1}^{m} u_i \nabla g_i(x^k) = 0, \; u_i \geq 0 \quad (i = 1,\dots,m), \tag{2}$$
$$g_i(x^k)u_i = \frac{f(x^k) - z^k}{m+k} \quad (i = 1,\dots,m) \;\; and \quad (k = 0,1,\dots\dots).$$

To begin with, we differentiate the function $\phi^k(x)$ to get:

$$G^k(x) = \nabla \phi^k(x) = \frac{m+k}{f(x) - z^k} \nabla f(x) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x) \quad (k = 0,1,\dots).$$

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

Further differentiations will yield:

$$H^k(x) = \nabla^2 \phi^k(x) = \frac{m+k}{f(x) - z^k} \nabla^2 f(x) - \frac{m+k}{(f(x) - z^k)^2} \nabla f(x) \nabla^T f(x) +$$
$$\sum_{i=1}^{m} \left(\frac{1}{g_i(x)} \nabla^2 g_i(x) - \frac{1}{(g_i(x))^2} \nabla g_i(x) \nabla^T g_i(x)\right) \qquad (k = 0,1,\dots).$$

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

Now, we will describe the basic algorithm for solving the problem (NLP) .

The following algorithm is designed to work in the relative interior of the feasible set X and solving the nonlinear programming problem (NLP) .

2.2 Algorithm for solving *NLP*

Step 1: *Initialization*: Let: k = 0 the iteration counter, $\varepsilon > 0$ the tolerance level, $x^0 \in Int(X)$ the starting interior point, and $z^0 \in R$ the lower bound where $f(x^0) > z^0$.

Step 2: *Feasible direction*: Find the unique solution of the following system of linear equations:

$H^k(x^k)\left(y - x^k\right) = -G^k(x^k)$. Let y^k denote the solution of this system.

This problem is purely linear and can be solved in polynomial time by Gaussian elimination requiring computations of order $O(nm^2)$ arithmetic operations.

Step 3: *Length of step:* Find the scalar:

$$\lambda^k = \underset{0 \leq \lambda \leq 1}{\arg\max} \; \phi^k(x^k + \lambda(y^k - x^k))$$

Step 4: *Updating*: Define the new point: $x^{k+1} = x^k + \lambda^k(y^k - x^k)$.

Step 5: *Termination test*: If $\left\| x^{k+1} - x^k \right\| < \varepsilon$, then stop, the point x^k is an optimal solution of NLP else define a new lower bound as follows: $z^{k+1} = z^k + \theta(f(x^{k+1}) - z^k)$, where $0 < \theta < 1$. Set $k = k + 1$ (increment the iteration counter) and return again to *step 2*.

2.3 The easily demonstrable properties

1. The direction $y^k - x^k$, determined in *Step 2* of the algorithm is a strict assent direction of $\phi^k(x)$ at $x^k \in Int(X)$.

From *Step 2* of the algorithm, it can be seen that: $H^k(x^k)\left(y^k - x^k\right) = -G^k(x^k)$.

Using the strict concavity of $\phi^k(x)$, it follows that,

$\left(y^k - x^k\right)^T H^k(x^k)\left(y^k - x^k\right) < 0$ so $\left(G^k(x^k)\right)^T \left(y^k - x^k\right) > 0$.

2. The point $x^{k+1} = x^k + \lambda^k\left(y^k - x^k\right)$ is feasible.

Being the feasible set X convex in $\mathbf{R}^n$, the proof can be completely derived from *Steps 3* and 4 of the algorithm.

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 $\phi^k(x)$.

2.4 The reduction of the potential function value

It is known that:

$$\phi^k(x) = (m + k)ln(f(x) - z^k) + \sum_{i=1}^{m} ln(g_i(x)) \qquad (k = 0,1,...),$$

$$\phi^{k+1}(x) = (m + k + 1)ln(f(x) - z^{k+1}) + \sum_{i=1}^{m} ln(g_i(x)) \qquad (k = 0,1,...),$$

$z^{k+1} = z^k + \theta(f(x^{k+1}) - z^k)$, $0 < \theta < 1$ and

$$\phi^{k+1}(x) - \phi^k(x) = ln\left(\frac{(f(x) - z^{k+1})^{m+k+1}}{(f(x) - z^k)^{m+k}}\right)$$

Now when $x = x^{k+1}$, then:

$$\phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) = ln\left(\frac{f(x^{k+1}) - z^{k+1}}{f(x^{k+1}) - z^k}\right)^{m+k}\left(f(x^{k+1}) - z^{k+1}\right).$$

But $f(x^{k+1}) - z^{k+1} = (1-\theta)\left(f(x^{k+1}) - z^k\right)$, then:

$$\phi^{k+1}(x^{k+1})-\phi^{k}(x^{k+1})=ln\left((1-\theta)^{m+k+1}(f(x^{k+1})-z^{k})\right).$$

Let z^* denote the value of the objective function $f(x)$ at the optimal solution of NLP, then:

$$\phi^{k+1}(x^{k+1})-\phi^{k}(x^{k+1})\le ln((1-\theta)^{m+k+1}(z^{*}-z^{k}))\le ln((1-\theta)^{m+k+1}(z^{*}-z^{0})).$$

Choosing $z^{*}-z^{0}\le(1-\theta)^{-m}$, where $(0<\theta<1)$, then we can see $\phi^{k+1}(x^{k+1})-\phi^{k}(x^{k+1})\le ln(1-\theta)^{k+1}$,

this means that the function $\phi^{k}(x)$ goes to $-\infty$ when k goes to $+\infty$.

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

Let z^* denote the value of the objective function $f(x)$ at the optimal solution of NLP and let z^k be the value of the objective value at the point x^k, then:

$$\frac{z^{*}-z^{k+1}}{z^{*}-z^{k}}=\frac{z^{*}-z^{k}-\theta(f(x^{k+1})-z^{k})}{z^{*}-z^{k}}=1-\theta\frac{f(x^{k+1})-z^{k}}{z^{*}-z^{k}}\quad(0<\theta<1)\tag{3}$$

Using inequality (1), it can be seen that:

$$z^{*}\le f(x^{k+1})+\sum_{i=1}^{m}u_{i}g_{i}(x^{k+1})\Rightarrow\ z^{*}-z^{k}\le f(x^{k+1})-z^{k}+\sum_{i=1}^{m}u_{i}g_{i}(x^{k+1})$$

Using (2), it follows that:

$$z^{*}-z^{k}\le f(x^{k+1})-z^{k}+\frac{m}{m+k}(f(x^{k+1})-z^{k})\Rightarrow z^{*}-z^{k}\le(1+\frac{m}{m+k})(f(x^{k+1})-z^{k}).$$

Substituting of the last inequality into (3) gives:

$$\frac{z^{*}-z^{k+1}}{z^{*}-z^{k}}\le 1-\theta(1+\frac{m}{m+k})^{-1}=1-\theta\frac{m+k}{2m+k}\le 1-\theta\frac{1}{2m+k}\ \Rightarrow$$

$$\left.\begin{array}{l}
z^{*}-z^{k+1}\le(1-\theta\dfrac{1}{2m+k})(z^{*}-z^{k})\\
z^{*}-z^{k}\le(1-\theta\dfrac{1}{2m+k-1})(z^{*}-z^{k-1})\\
.\\
.\\
.\\
z^{*}-z^{1}\le(1-\theta\dfrac{1}{2m})(z^{*}-z^{0})
\end{array}\right\}\Rightarrow$$

$$z^{*}-z^{k+1}\le\prod_{i=0}^{k}(1-\theta\frac{1}{2m+i})(z^{*}-z^{0})\le\prod_{i=0}^{k}(1-\theta\frac{1}{2m+k+1})(z^{*}-z^{0})\ .$$

As: $z^{*}-f(x^{k+1})\le z^{*}-z^{k+1}$, so:

$$z^* - f(x^{k+1}) \le (1-\frac{\theta}{2m+k+1})^{k+1}(z^* - z^0),$$

$$ln(z^* - f(x^{k+1})) \le ln((1-\frac{\theta}{2m+k+1})^{k+1}(z^* - z^0)) = (k+1)ln(1-\frac{\theta}{2m+k+1}) + ln(z^* - z^0).$$

Since $ln(1-\gamma) \le -\gamma, 0 < \gamma \le 1$, then:

$$ln(z^* - f(x^{k+1})) \le (k+1)\frac{-\theta}{2m+k+1} + ln(z^* - z^0).$$

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

$$(k+1)\frac{-\theta}{2m+k+1} + ln(z^* - z^0) < ln\varepsilon \Rightarrow (k+1)\frac{-\theta}{2m+k+1} \le ln\left(\frac{\varepsilon}{z^* - z^0}\right) \Rightarrow$$

$$-\frac{1}{\theta}\frac{2m+k+1}{k+1} \ge ln\left(\frac{z^* - z^0}{\varepsilon}\right) \Rightarrow -\frac{1}{\theta}(1+\frac{2m}{k+1}) \ge ln\left(\frac{z^* - z^0}{\varepsilon}\right) \Rightarrow -(1+\frac{2m}{k+1}) \ge \theta ln\left(\frac{z^* - z^0}{\varepsilon}\right)$$

$$\Rightarrow$$

$$1+\frac{2m}{k+1} \le -\theta ln\left(\frac{z^* - z^0}{\varepsilon}\right) \Rightarrow \frac{2m}{k+1} \le -1-\theta ln\left(\frac{z^* - z^0}{\varepsilon}\right) \le -\theta ln\left(\frac{z^* - z^0}{\varepsilon}\right) \Rightarrow$$

$$\frac{k+1}{2m} \ge -\frac{1}{\theta} ln\left(\frac{\varepsilon}{z^* - z^0}\right) \Rightarrow k+1 \ge -\frac{2m}{\theta} ln\left(\frac{\varepsilon}{z^* - z^0}\right) \Rightarrow k \ge -1-\frac{2m}{\theta} ln\left(\frac{\varepsilon}{z^* - z^0}\right).$$

From this inequality, it can be seen that the number of iterations K for an $\varepsilon-$ optimal solution is at most: $K = \left\lfloor -1-\frac{2m}{\theta} ln\left(\frac{\varepsilon}{z^* - z^0}\right) \right\rfloor + 1$ where $\lfloor u \rfloor$ denotes the integer part of the real number u .

Taking $z^* - z^0 < \frac{1}{\varepsilon}$, the number of iterations K can be described as follows $K = O(m\,|ln\varepsilon|)$.

2.6 Convergence analysis

From the algorithm, it is found that $\|x^{k+1} - x^k\| < \varepsilon$, which implies

$$G^k(x^k) = 0 \Rightarrow \frac{m+k}{f(x^k) - z^k}\nabla f(x^k) + \sum_{i=1}^{m}\frac{1}{g_i(x^k)}\nabla g_i(x^k) = 0 \Rightarrow$$

$$\nabla f(x^k) + \frac{f(x^k) - z^k}{m+k}\sum_{i=1}^{m}\frac{1}{g_i(x^k)}\nabla g_i(x^k) = 0.$$

Taking $u_i = \frac{f(x^k) - z^k}{m+k}\frac{1}{g_i(x^k)} \quad (i = 1,\ldots,m)$, it can be found:

$$\nabla f(x^k)+\sum_{i=1}^{m}u_i\nabla g_i(x^k)=0,\ u_i\geq 0\ \ (i=1,...,m),$$
$$g_i(x^k)\geq 0\ \ (i=1,...,m)\ and$$
$$g_i(x^k)u_i=\frac{f(x^k)-z^k}{m+k}\ \ (i=1,...,m).$$

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

As the proposed algorithm creates a sequence of interior points $\left\{x^k\right\}_{k=0,1,...}$ contained in $Int(X)$ and converges to a solution satisfying the KKT conditions and under the assumptions used in this paper then, by the general theory of convergence (Minoux, 1983), it can be concluded that the accumulation point x^k which is found by the algorithm is an $\varepsilon-$ optimal solution of NLP in X .

3 Statement of the Multiobjective Nonlinear Programming Problem (*MONLP*)

A multiobjective nonlinear programming problem (*MONLP*) is generally described through the standard formulation:

$$\begin{array}{ll} maximize & v_1=v_1(x) \\ maximize & v_2=v_2(x) \\ . & \\ . & \\ maximize & v_r=v_r(x) \\ subject\ to & g_i(x)\geq 0 \quad (i=1,...,m) \end{array}$$

(*MONLP*)

Where the functions: $v_i(x)\ (i=1,...,r)$ and $g_i(x)\ (i=1,...,m)$ are concave with continuous first and second-order derivatives. The first derivatives of $v_i(x)\ (i=1,...,r)$ satisfy Lipchitz's condition in x on X , where the feasible set $X=\left\{x\in\mathbf{R}^n\ /\ g_i(x)\geq 0\ (i=1,...,m)\right\}$ is compact and convex in the real space $\mathbf{R}^n$. The interior of the feasible region (denoted $Int(X)$) is non-empty and bounded, n is the number of unknown or decision variables, m is the number of constraints such that ($m<n$), and r is the number of objective functions.

In multiobjective programming, it is supposed that, the decision-maker has to be capable of presenting his global preferences through a utility function $U(v)=U(v_1,...,v_r)$. This function is not necessarily being explicitly known but it is supposed to satisfy certain conditions (continuously differentiable, concave, and strictly increasing in v on the objective space $V(X)$. $V(X)$ is the image of the feasible set X (decision space) by the objective functions $v_i(x)\ (i=1,...,r)$. It is also assumed that the first derivative of $U(v)=U(v_1,...,v_r)$ satisfies Lipchitz's condition in v on $V(X)$.

Lemma 1: If the utility function $U(v) = U(v_1,...,v_r)$ is concave and strictly increasing in v on the objective space $V(X)$, then the function $\varphi(x) = U(v_1(x),...,v_r(x))$ is concave in x on the decision space X .

Consider the following relation:

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

Where $\varphi = UoV$ and the gradient of the utility function with respect of x is given as follows $\nabla_x \varphi(x) = \sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j} \nabla_x v_j(x)$.

Since $U(v) = U(v_1,...,v_r)$ is strictly increasing in v on $V(X)$, then $\frac{\partial \mathrm{U}}{\partial v_j} > 0 \ (j = 1,...,r)$. The functions $v_i(x) \ (i = 1,...,r)$ are concaves on X . Therefore:

$\forall x, x^* \in X \ ; v_j(x^*) \le v_j(x) + \nabla_x^T v_j(x)(x^* - x) \quad (j = 1,...,r)$, then:

$$\sum_{j=1}^{r} \frac{\partial U}{\partial v_j}(v_j(x^*) - v_j(x)) \le \sum_{j=1}^{r} \frac{\partial U}{\partial v_j} \nabla_x^T v_j(x)(x^* - x) \quad .$$

Using the last inequality, it can be found that:

$$\begin{aligned} \nabla_x^T \varphi(x)(x^* - x) &= \left(\sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j} \nabla_x^T v_j(x) \right)(x^* - x) \ge \sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j}(v_j(x^*) - v_j(x)) \\ &= \nabla_v^T U(v(x))(v(x^*) - v(x)) \end{aligned}$$

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

$$\nabla_x^T \varphi(x)(x^* - x) \ge \nabla_v^T U(v(x))(v(x^*) - v(x)) \ge U(v(x^*)) - U(v(x)) = \varphi(x^*) - \varphi(x)$$

So that $\varphi(x^*) - \varphi(x) \le \nabla_x^T \varphi(x)(x^* - x)$, which means that the function $\varphi(x)$ is concave on X .

Lemma 2: If the derivative of the utility function $U(v)=U(v_1,...,v_r)$ is strictly increasing and satisfies Lipchitz's condition on the objective space $V(X)$, then the derivative of the function $\varphi(x)=U(v_1(x),...,v_r(x))$ satisfies Lipchitz's condition on the decision space X .

It is easy to see that:

$$\left|\nabla_x\varphi(x^2)-\nabla_x\varphi(x^1)\right|=\left|\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}\nabla_x v_j(x^2)-\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}\nabla_x v_j(x^1)\right|$$
$$=\left|\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}(\nabla_x v_j(x^2)-\nabla_x v_j(x^1))\right|$$

The derivatives of the functions v_j $(j=1,...,r)$ satisfy Lipchitz's condition on X , it can be seen that, there is $L\geq 0$ such that:

$$\left|\nabla_x v_j(x^2)-\nabla_x v_j(x^1)\right|\leq L\left\|x^2-x^1\right\|.$$

The function $U(v)=U(v_1,...,v_r)$ is strictly increasing $\frac{\partial \mathrm{U}}{\partial v_j}>0$ $(j=1,...,r)$, and then it can be found $\left|\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}(\nabla_x v_j(x^2)-\nabla_x v_j(x^1))\right|\leq L\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}\left\|x^2-x^1\right\|\leq L'\left\|x^2-x^1\right\|$. So $\left|\nabla_x\varphi(x^2)-\nabla_x\varphi(x^1)\right|\leq L'\left\|x^2-x^1\right\|$, then the derivative of the function $\varphi(x)$ satisfies the condition of Lipchitz on the decision space X .

3.1 Approximate gradient

The multiobjective nonlinear programming problem (*MONLP*) is ambiguous since the objectives usually are conflicting and pursuing the optimum with respect to each objective. This leads to different solutions. The ambiguity may be solved by introducing a utility function $U(v)=U(v_1,...,v_r)$, defined over the space of objectives $V(X)$ and presented by the decision-maker. This function has to satisfy certain conditions as being continuously differentiable, concave, and strictly increasing on the objective space and its derivative satisfies Lipchitz's condition in order to ensure the global convergence and to reach a global optimum.

If $U(v)=U(v_1,...,v_r)$ is explicitly available then, we have to find a way to approximate the gradient of the utility function based on the values of the utility function at the current iteration.

The gradient of the utility function in the decision space X could be given as follows:

$$\nabla_x\varphi(x)=\frac{\partial U(v)}{\partial v_1}\nabla_x v_1(x)+....+\frac{\partial U(v)}{\partial v_r}\nabla_x v_r(x)$$

$$= \frac{\partial U(v)}{\partial v_1}\begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_1(x)}{\partial x_n} \end{pmatrix} + \ldots + \frac{\partial U(v)}{\partial v_r}\begin{pmatrix} \frac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_r(x)}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} + \ldots + \frac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_1(x)}{\partial x_n} + \ldots + \frac{\partial v_r(x)}{\partial x_n} \end{pmatrix} \times \nabla_v U(v).$$

In matrix form, the gradient can be written as: $\nabla_x \varphi(x) = C \times \nabla_v U(v)$.

Where $\nabla_v U(v) = \left(\frac{\partial U(v)}{\partial v_1}, \ldots., \frac{\partial U(v)}{\partial v_r}\right)^T$, $\nabla_x \varphi = \left(\frac{\partial \varphi(x)}{\partial x_1}, \ldots., \frac{\partial \varphi(x)}{\partial x_n}\right)^T$ and

$$C = \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} + \ldots + \frac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_1(x)}{\partial x_n} + \ldots + \frac{\partial v_r(x)}{\partial x_n} \end{pmatrix}.$$

Therefore, to find the approximate gradient $\nabla_x \varphi(x)$ in the decision space we have to evaluate the gradient of the utility function $\nabla_v U(v)$ in the objective space. Since the derivatives objectives matrix C is $n \times r$ matrix, considering each of the r objective functions by themselves, results in stepping from the current iteration, x^0 along a specific step direction to r end points x^i $(i = 1, \ldots, r)$ with their respective values for the r objective functions. The change in the utility function in decision space $\varphi(x)$ in stepping from the current iteration x^0 to the set of r new iterations can be approximated through a first order Taylor's expansion as follows:

$$\begin{aligned} &\varphi(x^1) = \varphi(x^0) + \nabla_x^T \varphi(x) \times (x^1 - x^0) \\ &. \\ &. \\ &\varphi(x^r) = \varphi(x^0) + \nabla_x^T \varphi(x) \times (x^r - x^0) \end{aligned}$$

Or

$$\begin{aligned} &\varphi(x^1) = \varphi(x^0) + \nabla_v^T U(v) \times C^T \times (x^1 - x^0) \\ &. \\ &. \\ &\varphi(x^r) = \varphi(x^0) + \nabla_v^T U(v) \times C^T \times (x^r - x^0) \end{aligned}$$

These equations can be rewritten as follows:

$$\varphi(x^1) = \varphi(x^0) + \nabla_v^T U(v) \times \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} + \ldots + \frac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_1(x)}{\partial x_n} + \ldots + \frac{\partial v_r(x)}{\partial x_n} \end{pmatrix} \times (x^1 - x^0)$$

.

.

$$\varphi(x^r) = \varphi(x^0) + \nabla_v^T U(v) \times \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} + \ldots + \frac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial v_1(x)}{\partial x_n} + \ldots + \frac{\partial v_r(x)}{\partial x_n} \end{pmatrix} \times (x^r - x^0)$$

In a matrix form, we can write:

$$\varphi(x^1) - \varphi(x^0) = \nabla_v^T U(v) \times \begin{pmatrix} \nabla_x^T v_1(x) \times (x^1 - x^0) \\ . \\ . \\ \nabla_x^T v_r(x) \times (x^1 - x^0) \end{pmatrix}$$

.

.

$$\varphi(x^r) - \varphi(x^0) = \nabla_v^T U(v) \times \begin{pmatrix} \nabla_x^T v_1(x) \times (x^r - x^0) \\ . \\ . \\ \nabla_x^T v_r(x) \times (x^r - x^0) \end{pmatrix}$$

$$\varphi(x^1)-\varphi(x^0)=\nabla_v^T U(v)\times\begin{pmatrix} v_1(x^1)-v_1(x^0) \\ . \\ . \\ v_r(x^1)-v_r(x^0) \end{pmatrix}$$

.

Or

.

$$\varphi(x^r)-\varphi(x^0)=\nabla_v^T U(v)\times\begin{pmatrix} v_1(x^r)-v_1(x^0) \\ . \\ . \\ v_r(x^r)-v_r(x^0) \end{pmatrix}$$

$$\Delta\varphi=\begin{pmatrix} v_1(x^1)-v_1(x^0),...,v_r(x^1)-v_r(x^0) \\ . \\ . \\ v_1(x^r)-v_1(x^0),...,v_r(x^r)-v_r(x^0) \end{pmatrix}\times\nabla_v U(v)$$

$$\Delta\varphi=\Delta V\times\nabla_v U(v)\Rightarrow\nabla_v U(v)=\left(\Delta V\right)^{-1}\times\Delta\varphi .$$

But we have $\nabla_x\varphi(x)=C\times\nabla_v U(v)$, then $\nabla_x\varphi(x)=C\times\left(\Delta V\right)^{-1}\times\Delta\varphi$.

From this relation, it could be concluded that, the Taylor's series approximation for the gradient of the utility function $\varphi(x)$ in the decision space involves the value of the utility function at the initial point x^0 and the value at the r new iterations.

In the absence of an explicit utility function, these values are unavailable and have to be approximated. One way of assessment of relative preferences for the $(r+1)$ value vectors is through the analytic hierarchy process (AHP) (Saaty, 1988; Arbel, 1994; Arbel and Oren, 1996).

To obtain an approximate measure for the utility function at points of interest we proceed as follows:

While the value of the utility function at the $(r+1)$ points $\{x^0,x^1,...,x^r\}$ is unknown, we can evaluate the complete $r-$dimensional vector of objective functions value, $v_i(x)$ $(i=1,...,r)$ at each of these points. This information is now presented, in objective space, to the decision maker and seeks to obtain relative preference for these points. This is accomplished by using AHP and involves filling a comparison matrix whose principal eigenvector provides the priority vector showing the relative preference for these points. The priority vector $pr\in\mathbf{R}^{r+1}$ provides an approximate measure of the vector $\Delta\varphi$ given through:

$$\Delta\varphi\approx\Delta pr=\left(pr_1-pr_0,...,pr_r-pr_0\right)$$

Where $pr_i\ (i=0,...,r)$ is the priority of the $i-th$ iteration as derived by using the method of AHP, the gradient of the utility function with respect to x, $\nabla_x \varphi(x)$ is evaluated through: $\nabla_x \varphi(x) = C \times \left(\Delta V\right)^{-1} \times \Delta\varphi$.

3.2 Summary of the analytic hierarchy process (*AHP*)

The application of *AHP* technique is for $r-$ dimensional vector $v_i(x)\ (i=1,...,r)$ value obtained at each of the $(r+1)$ points $\{x^0, x^1,...,x^r\}$, at the current iteration.

- Create $(r+1)\times(r+1)$ comparison matrix for $r+1$ requirements with the aid of the decision maker to provide relative preferences

  (Requirements here are the r - dimensional vector $v_i(x)\ (i=1,...,r)$ value obtained

  at each of the points $\{x^0, x^1,...,x^r\}$, at the current iteration)

  The creation of the matrix is as follows:

  For element (*x* , *y*) in the comparison matrix enter:

  $\mapsto$ 1- If *x* and *y* are of equal value (equal importance)

  $\mapsto$ 3- If *x* is slightly more preferred than *y* (weak importance of one over the other)

  $\mapsto$ 5- If *x* is strongly more preferred than *y* (strong importance)

  $\mapsto$ 7- If *x* is very strongly more preferred than *y* (demonstrated importance over the other)

  $\mapsto$ 9- If *x* is extremely more preferred than *y* (absolute importance)

  $\mapsto$ 2, 4, 6, 8- intermediate values between

  $\mapsto$ And for (*y*, *x*) enter the reciprocal.

- Estimate the eigenvalues (eigenvector) as follows:

  E.g. “averaging over normalized columns”

  $\mapsto$ Calculate the sum of each column

  $\mapsto$ Divide each element in the matrix by the sum of its column

  $\mapsto$ Calculate the sum of each row

  $\mapsto$ Divide each row sum by the number of rows

  This gives a value of relative priority for each requirement (priority vector $pr \in \mathbf{R}^{r+1}$).

Remark: If the utility function is available, we could use, at the current iteration, the normalized utility function values at the points $\{x^0, x^1,...,x^r\}$ as components of the priority vector pr.

3.3 Logarithmic barrier function and its derivatives concerning the problem (*MONLP*)

The following logarithmic barrier function is associated with the primal problem *MONLP* (Tlas and Abdul Ghani, 2005) as follows::

$$\omega^k(x) = (m+k)ln(\nabla_x^T \varphi(x^k)(x-x^k) - z^k) + \sum_{i=1}^{m} ln(g_i(x)) \quad (k=0,1,...)$$

Where, z^k is a real negative number and k is the number of iteration. The function $\omega^k(x)$ is defined on the interior of the feasible region X, twice-continuously differentiable, strictly concave and $\omega^k(x)$ tends to $-\infty$ when x goes to the boundary of X.

To begin with, we differentiate the function $\omega^k(x)$ to get $\nabla\omega^k(x)$ the gradient of $\omega^k(x)$:

$$\nabla\omega^k(x) = \frac{m+k}{\nabla^T\varphi(x^k)(x-x^k)-z^k}\nabla\varphi(x^k) + \sum_{i=1}^{m}\frac{1}{g_i(x)}\nabla g_i(x) \quad (k=0,1,...).$$

Further differentiations will yield to get $\nabla^2\omega^k(x)$ the Hessian matrix of $\omega^k(x)$:

$$\nabla^2\omega^k(x) = -\frac{m+k}{(\nabla^T\varphi(x^k)(x-x^k)-z^k)^2}\nabla\varphi(x^k)\nabla^T\varphi(x^k) +$$
$$\sum_{i=1}^{m}\left(\frac{1}{g_i(x)}\nabla^2 g_i(x) - \frac{1}{(g_i(x))^2}\nabla g_i(x)\nabla^T g_i(x)\right) \quad (k=0,1,...).$$

We associate also the following logarithmic barrier functions with the objective functions $v_i(x)\ (i=1,...,r)$:

$$\psi_i^k(x) = (m+k)ln(v_i(x)-\beta_i^k) + \sum_{i=1}^{m} ln(g_i(x)) \quad (i=1,...,r),(k=0,1,...).$$

Where $\beta_i^k\ (i=1,...,r)$ are the lower bounds for the optimal values β_i^* related to the objective functions $v_i(x)\ (i=1,...,r)$.

The gradient vectors and the Hessian matrixes of $\psi_i^k(x)\ (i=1,...,r)$ are given as:

$$G_i^k(x) = \nabla\psi_i^k(x) = \frac{m+k}{v_i(x)-\beta_i^k}\nabla v_i(x) + \sum_{i=1}^{m}\frac{1}{g_i(x)}\nabla g_i(x) \quad (i=1,...,r),(k=0,1,...).$$

$$H_i^k(x) = \nabla^2\psi_i^k(x) = \frac{m+k}{v_i(x)-\beta_i^k}\nabla^2 v_i(x) - \frac{m+k}{(v_i(x)-\beta_i^k)^2}\nabla v_i(x)\nabla^T v_i(x) +$$
$$\sum_{i=1}^{m}\left(\frac{1}{g_i(x)}\nabla^2 g_i(x) - \frac{1}{(g_i(x))^2}\nabla g_i(x)\nabla^T g_i(x)\right) \quad (i=1,...,r),(k=0,1,...).$$

Now, we will describe the basic algorithm for solving *MONLP*.

The following algorithm is designed to work in the relative interior of the feasible set X and solving the muliobjective nonlinear programming problem (*MONLP*).

3.4 Algorithm for solving *MONLP*

Step 1: *Initialization*. Let: k=0 the iteration counter, $\varepsilon > 0$ the tolerance level, $x^0 \in Int(X)$ the starting interior point $z^0 < 0$, $\beta_i^0 \in \mathbf{R}^r$ lower bounds where $v_i(x^0) > \beta_i^0$ $(i = 1,...,r)$ and $L \geq 0$ (Lipchitz's constant).

Step 2: *Feasible directions.* For $i = 1,...,r$, find the unique solution y_i^k of the following system of linear equations:

$$H_i^k(x^k)\left(y - x^k\right) = -G_i^k(x^k) \; .$$

This problem is purely linear and can be solved in polynomial time by Gaussian elimination requiring computations of order $O(nm^2)$ arithmetic operations.

Step 3: *Length of steps.* For $i = 1,...,r$, find the scalar:

$$\lambda_i^k = \underset{0 \leq \lambda \leq 1}{\arg\max} \quad \psi_i^k(x^k + \lambda(y_i^k - x^k))$$

Step 4: *Updating.* For $i = 1,...,r$, define the new point: $x_i = x^k + \lambda_i^k(y_i^k - x^k)$, and consequently find:

$$x_0 = x^k, C = \begin{pmatrix} \dfrac{\partial v_1(x)}{\partial x_1} + ... + \dfrac{\partial v_r(x)}{\partial x_1} \\ . \\ . \\ . \\ \dfrac{\partial v_1(x)}{\partial x_n} + ... + \dfrac{\partial v_r(x)}{\partial x_n} \end{pmatrix}, \Delta V = \begin{pmatrix} v_1(x_1) - v_1(x_0),...,v_r(x_1) - v_r(x_0) \\ . \\ . \\ v_1(x_r) - v_1(x_0),...,v_r(x_r) - v_r(x_0) \end{pmatrix},$$

$$\Delta\varphi = \left(\varphi(x_1) - \varphi(x_0),...,\varphi(x_r) - \varphi(x_0)\right)^T \text{ or } \Delta\varphi = \left(pr_1 - pr_0,..., pr_r - pr_0\right)^T \text{ and}$$

$$\nabla_x \varphi(x^k) = C \times \left(\Delta V\right)^{-1} \times \Delta\varphi \; .$$

Step 5: *Feasible direction*. Find $y^k \in \mathbf{R}^n$ which solve the linear system of equations:

$$\nabla^2 \omega^k(x^k)\left(y - x^k\right) = -\nabla\omega^k(x^k)$$

Step 6: *Length of step*. Find the scalar:

$$\rho_1 = \underset{0 \leq \lambda \leq 1}{\arg\max} \quad \omega^k(x^k + \lambda(y^k - x^k)) \;, \; \rho_2 = \frac{\nabla_x^T \varphi(x^k)(y^k - x^k)}{L\left\|y^k - x^k\right\|^2} \text{ and } \; \lambda^k = \min\left\{\rho_1, \rho_2\right\}$$

Define the new point $x^{k+1} = x^k + \lambda^k(y^k - x^k)$

Step 7**:** *Termination test.* If $\left\| x^{k+1} - x^k \right\| < \varepsilon$, then stop, the point x^k is an optimal solution of *MONLP* else define new lower bounds as follows: $\beta_i^{k+1} = \beta_i^k + \theta_i \times (v_i(x^{k+1}) - \beta_i^k)\ (i = 1, \dots, r)$, where $0 < \theta_i < 1\ (i = 1, \dots, r)$ and $z^{k+1} = (1-\theta) z^k$, $0 < \theta < 1$. Set $k = k+1$ (increment the iteration counter) and return again to *step 2*.

3.5 Convergence analysis

The direction $y^k - x^k$, determined in *Step 5* of the algorithm is a strict assent direction of $\omega^k(x)$ at x^k because, from *Step 5* of the algorithm, it can be seen that: $\nabla^2 \omega^k(x^k)\left(y - x^k\right) = -\nabla \omega^k(x^k)$.

Using the strict concavity of $\omega^k(x)$, it follows that: $\left(y - x^k\right)^T \nabla^2 \omega^k(x^k)\left(y - x^k\right) < 0$,so $\left(\nabla \omega^k(x^k)\right)^T \left(y^k - x^k\right) > 0$.

The point $x^{k+1} = x^k + \lambda^k \left(y^k - x^k\right)$ is feasible because the feasible set X is convex in $\mathbf{R}^n$.

Now, from the mean value theorem, it can be seen that:

$\varphi(x^k) - \varphi(x^{k+1}) = \nabla^T \varphi(\xi)(x^k - x^{k+1})$, where $\xi \in \left[x^k, x^{k+1}\right]$ and $x^{k+1} = x^k + \lambda^k (y^k - x^k)$

$$\varphi(x^k) - \varphi(x^{k+1}) = \left(\nabla^T \varphi(\xi) - \nabla^T \varphi(x^k)\right)(x^k - x^{k+1}) + \nabla^T \varphi(x^k)(x^k - x^{k+1})$$

$$\varphi(x^k) - \varphi(x^{k+1}) \le \left\| \nabla^T \varphi(\xi) - \nabla^T \varphi(x^k) \right\| \left\| x^k - x^{k+1} \right\| + \nabla^T \varphi(x^k)(x^k - x^{k+1})$$

Using the following condition of Lipchitz $\left\| \nabla^T \varphi(\xi) - \nabla^T \varphi(x^k) \right\| \le L \left\| \xi - x^k \right\|$, it can be found:

$$\varphi(x^k) - \varphi(x^{k+1}) \le L \left\| \xi - x^k \right\| \left\| x^k - x^{k+1} \right\| + \nabla^T \varphi(x^k)(x^k - x^{k+1})$$

$$\varphi(x^k) - \varphi(x^{k+1}) \le L \left\| x^k - x^{k+1} \right\|^2 + \nabla^T \varphi(x^k)(x^k - x^{k+1})$$

$$\varphi(x^k) - \varphi(x^{k+1}) \le L \lambda^2 \left\| y^k - x^k \right\|^2 - \lambda \nabla^T \varphi(x^k)(y^k - x^k)$$

$$\varphi(x^{k+1}) - \varphi(x^k) \ge \lambda \nabla^T \varphi(x^k)(y^k - x^k) - L \lambda^2 \left\| y^k - x^k \right\|^2.$$

Choosing $0 \le \lambda \le \dfrac{\nabla_x^T \varphi(x^k)(y^k - x^k)}{L \left\| y^k - x^k \right\|^2}$, then it can be seen that: $\varphi(x^{k+1}) \ge \varphi(x^k)$, this means that the value of the function $\varphi(x)$ increase in each iteration.

From the algorithm it is found that $\left\| x^{k+1} - x^k \right\| < \varepsilon$, which implies

$$\frac{m+k}{\nabla^T \varphi(x^k)(x^{k+1}-x^k)-z^k}\nabla\varphi(x^k)+\sum_{i=1}^{m}\frac{1}{g_i(x^k)}\nabla g_i(x^k)=0 \qquad \Rightarrow$$

$$\nabla\varphi(x^k)+\frac{\nabla^T \varphi(x^k)(x^{k+1}-x^k)-z^k}{m+k}\sum_{i=1}^{m}\frac{1}{g_i(x^k)}\nabla g_i(x^k)=0\,.$$

Taking $u_i = \frac{\nabla^T \varphi(x^k)(x^{k+1}-x^k)-z^k}{m+k}\frac{1}{g_i(x^k)} \quad (i=1,\ldots,m)$, we find:

$$\nabla\varphi(x^k)+\sum_{i=1}^{m}u_i\nabla g_i(x^k)=0,\; u_i \geq 0 \;\; (i=1,\ldots,m),$$
$$g_i(x^k)\geq 0 \;\; (i=1,\ldots,m),$$
$$g_i(x^k)u_i = \frac{\nabla^T \varphi(x^k)(x^{k+1}-x^k)-z^k}{m+k} \;\; (i=1,\ldots,m).$$

Where z^k goes to zero when k goes to the infinity, this 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 $Int(X)$ and converges to a solution satisfying the KKT conditions, under the assumptions used in this paper, then by the general theory of convergence, it can be concluded that the accumulation point x^k which is found by the algorithm is an $\varepsilon-$ optimal solution of the *MONLP* in X .

4 Conclusions

An algorithm for solving multiobjective nonlinear programming problems has been proposed. The algorithm is based on a single-objective nonlinear variant of interior point method using logarithmic barrier function in order to generate interior search directions. New feasible points are found along these directions which will be later used for deriving best-approximation to the gradient of the implicitly-known utility function at the current iterate. Using this approximate gradient, a single feasible interior direction for the implicitly-utility function could be found by solving a set of linear equations. It may be easily taken an interior step from the current iterate to the next one along this feasible direction. During the execution of the algorithm, a sequence of interior points will be generated. It has been proved that this sequence converges to an $\varepsilon-$ optimal solution, where ε is a predetermined error tolerance known a priori.

For assuring the global convergence of the algorithm and to reach a global optimum, it is supposed that the utility function has to satisfy certain conditions as being continuously differentiable, concave and strictly increasing on the objective space and its derivative satisfies Lipchitz's condition. A simple formula is derived to approximate the gradient of the utility function based on the objective values and also on the utility function values, when it is known explicitly. In the absence of an explicit utility function, these values are unavailable and have to be approximated. The best way of approximating is through the use of the analytic hierarchy process (AHP) technique. Further deeply research in this new area of multiobjective programming is needed and should be concentrated on the ways of developing more rapid and robust interactive methods for solving multi-objective nonlinear programming problems.

5 Illustrative Examples

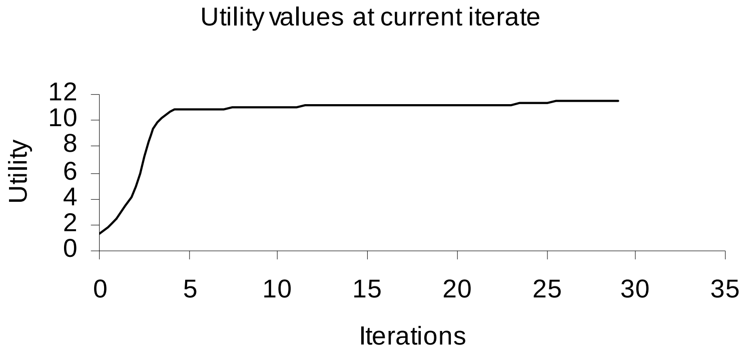
The demonstration of the proposed algorithm will be done through the following numerical example. Consider the following $MONLP$ problem:

$$\begin{aligned}
&\max v_1 = v_1(x) = x_1 \\
&\max v_2 = v_2(x) = x_2 \\
&Subject\ \ to: \\
&\quad g_1(x) = -3x_1 - 2x_2 + 6 \geq 0 \\
&\quad g_2(x) = -x_1 + 2 \geq 0 \\
&\quad g_3(x) = -x_2 + 2 \geq 0 \\
&\quad g_4(x) = x_1 \geq 0 \\
&\quad g_5(x) = x_2 \geq 0
\end{aligned}$$

For this example, an initial point is available through $x^0 = \begin{pmatrix}0.1 & 0.1\end{pmatrix}^T$, Lipchitz's constant $L = 2$, $\theta = 0.2$ and $z^0 = -25$, assuming that the decision maker's utility function is given through $U(v) = 5v_1 - v_1^2 + 8v_2 - 2v_2^2$. This vector optimization problem has an optimal solution given through $x^* = \begin{pmatrix}1 & 1.5\end{pmatrix}^T$, $v_1^* = 1,\ v_2^* = 1.5,\ U(v_1^*, v_2^*) = 11.5$.

Solution results (Current iteration)

k	x_1	x_2	U	k	x_1	x_2	U
0	0.1	0.1	1.27	15	0.9720	1.3910	11.1734
1	0.1995	0.2047	2.5114	16	0.8605	1.5662	11.1859
2	0.4020	0.4417	4.9922	17	0.9652	1.4094	11.1968
3	0.7802	1.0360	9.4337	18	0.8723	1.5555	11.2054
4	0.6710	1.6930	10.7164	19	0.9574	1.4266	11.2127
5	0.9169	1.3047	10.7768	20	0.8785	1.5518	11.2189
6	0.6958	1.7091	10.8257	21	0.9473	1.4453	11.2236
7	0.9783	1.2827	10.9055	22	0.8727	1.5683	11.2292
8	0.7504	1.6597	10.9574	23	0.9313	1.4725	11.2326
9	0.9851	1.3135	11.0124	24	0.9110	1.5876	11.3850
10	0.7901	1.6286	11.0504	25	0.9703	1.4893	11.3885
11	0.9834	1.3435	11.0878	26	0.9609	1.5516	11.4791
12	0.8209	1.6022	11.1142	27	1.0091	1.4776	11.4815
13	0.9782	1.3694	11.1389	28	0.9909	1.5046	11.4818
14	0.8437	1.5818	11.1570	29	0.9974	1.4949	11.4819



Acknowledgments
Author wishes to thank Prof. I. Othman, the DG of the AECS for his valuable support and encouragement throughout this work. The anonymous reviewers are cordially thanked for their critics, remarks and suggestions that considerably improved the final version of this paper.

References
Arbel A. 1994. An interior multiobjective primal-dual linear programming algorithm using approximated gradients and sequential generation of anchor points. Optimization, 30: 137-150

Arbel A, Oren SS. 1996. Using approximate gradients in developing an interactive interior primal-dual multiobjective linear programming algorithm. European Journal of Operational Research, 89: 202-211

Hertog D, Roos C, Terlaky T. 1991. A potential-reduction variant of Renegar's short-following method for linear programming. Linear algebra and its applications, 152: 43-68

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

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

Mehrotra S, 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, France

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

Saaty TL. 1988. Multicriteria decision making: The analytic hierarchy process. RSW publication, Pittsburgh, PA

Tlas M. 2013. An interactive interior point method for multiobjective nonlinear programming problems. Journal of Mathematics, Statistics and Operations Research. 2(1):34-44

Tlas M. 2024. An interactive interior point method for mathematical multi-objective linear optimization problems based on approximate gradients. Selforganizology, 11(3-4): 54-75

Tlas M. Abdul Ghani B. 2005. A logarithmic barrier function method for solving nonlinear multiobjective programming problems. Control and Cybernetics, 34 (2): 487-504

Zhang WJ. 2019. Analytic Hierarchy Process (AHP): Matlab computation software. Network Biology, 9(1): 10-17