Abstract—An interactive interior point algorithm for solving a multiobjective nonlinear programming problem has been proposed in this paper. The algorithm 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 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 $\varepsilon$ is a predetermined error tolerance known a priori. A numerical multiobjective example is illustrated using this algorithm.

Index Terms—Barrier function, Interior point method, Newton’s method, Multicriteria optimization, Multiobjective programming

I. INTRODUCTION

After the seminal algorithm of Karmarkar (1984) [6] for solving linear programming problems in polynomial time, a great number of the so-called interior point methods for both linear and nonlinear programming have been reported in the literature.

Renegar (1988)[9], and Iri and Imai (1986)[5] proposed interior point algorithms for solving linear programming problems in polynomial time. The algorithms were based on two main ideas: the analytical center concept and the Newton’s method. Recently, many approaches of interior point to convex programming using the analytical center idea and Newton’s method have been reported by Mehrotra and sun (1990) [7] for convex quadratic programming and by Hertog, Roos, and Terlaky (1991, 1992) [3], [4] for linear programming and for a class of smooth convex programming problems.

Following these proposals, it is useful to generalize these ideas of interior point technique to the domain of multiobjective programming. Therefore, two algorithms were proposed for solving single and multiple-objective nonlinear programming problems based on these ideas.

The first algorithm developed in this paper is an interior point variant for solving single-objective nonlinear programming problems. In this algorithm the line search, in each iterate, is performed along Newton’s direction which can be found by solving a set of linear equations in polynomial time using Gaussian elimination method which requires arithmetic operations of order $O(n^2)$, 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 an $\varepsilon$-optimal solution is at most $O(m \log \varepsilon)$ iterations where $m$ denotes the number of constraints of problem and $\varepsilon$ is the predetermined error tolerance.

The second algorithm proposed is an interactive interior point variant for multiple-objective nonlinear programming problems. The algorithm is mainly based on both logarithmic barrier functions and approximate gradients. The algorithm 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 $\varepsilon$ 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, will lead to different solutions. This ambiguity may be solved by introducing a utility function (or preference function) defined onto the space of objectives. It is supposed that the decision-maker is capable to present his global preferences through this function. This function is not necessarily being explicitly known but it is supposed to satisfy certain conditions as being continuously differentiable, concave and strictly increasing onto 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 iterate. In the contrary case, when the utility function is implicitly known the approximate gradient could be evaluated through the values of the objective functions and the analytic hierarchy process (AHP) technique at the current iterate. For more details about the AHP technique, the reader is invited to consult the following references: Saaty (1988) [10], Arbel (1994) [1], and Arbel and Oren (1996) [2].

II. STATEMENT OF THE NONLINEAR PROGRAMMING PROBLEM (NLP)

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

Maximize $f(x)$

Subject to $g_i(x) \geq 0 \ (i = 1, ..., m)$

where $x \in \mathbb{R}^n$, $n$ is the number of unknown (decision) variables and $m$ is the number of constraints. The functions $f(x)$ and $g_i(x)$ are concave with continuous first and second-order derivatives, the first derivative of the objective function $f(x)$ satisfies the Lipschitz’s condition on the decision space $X = \{x \in \mathbb{R}^n : g_i(x) \geq 0 \ (i = 1, ..., m)\}$.

It is supposed that, the interior of the feasible region $X$, denoted as $\text{Int} (X)$ is non-empty, compact and convex in the real space $\mathbb{R}^n$.

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

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

Subject to $\nabla f(x) + \sum_{i=1}^{m} u_i \nabla g_i(x) = 0 \ (DNLP)$

$u_i \geq 0 \ (i = 1, ..., m)$

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:

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

is correct.

A. A logarithmic barrier function and its derivatives

We associate the following suggested multiplicative barrier function with the primal problem (NLP):

$$\psi^k(x) = \left(\nabla f(x^k) (x - x^k) - z^k\right) \times \prod_{i=1}^{m} g_i(x) \ (k = 0, 1, ...)$$

where $x^k \in \text{Int} (X)$, $s$ is an integer number greater or equal to $m$, this number plays the role of a weight, $z^k$ is an arbitrary negative real number and $k$ is the number of iteration. This function is inspired from the work of Tlas and Abdul Ghani (2005) [11] with some modifications. The function $\psi^k(x)$ is defined on the feasible region $X$, strictly concave, strictly positive on $\text{Int} (x)$ and close to zero when $x$ goes to the boundary of $X$. It is difficult to find the first and second derivatives of $\psi^k(x)$, then it is useful to use the first and second derivatives of $\ln(\psi^k(x))$:

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

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

This function is also defined only on the interior $\text{Int} (X)$ 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 $z^k$) at a unique point denoted $x^*$. The necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions for this optimum are:

$$g_i(x^*) \geq 0 \ (i = 1, ..., m),$$

$$\nabla f(x^*) + \sum_{i=1}^{m} \nabla g_i(x^*) = 0 \ (i = 1, ..., m),$$

$$g_i(x^*) u_i = \frac{\nabla f(x^*) (x^* - x^k) - z^k}{s} \ (i = 1, ..., m) \quad (k = 0, 1, ...,).$$

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

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

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

The vector $G^k(x)$ will simply be called the gradient of $\phi^k(x)$. Further differentiation will yield:
\[ H^k(x) = \nabla^2 \phi^k(x) = \frac{s}{(\nabla f(x^k)(x-x^k) - z^k)} \nabla f(x^k) \nabla f(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla^2 g_i(x) - \frac{1}{g_i(x)} \nabla g_i(x) \nabla g_i(x) \] 

\[(k = 0, 1, \ldots)\]

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 \((NLP)\).

**B. Algorithm for solving NLP**

**Step 1. Initialization.** Let: \( k = 0 \) the iteration counter, \( \varepsilon > 0 \) the tolerance level, \( x^0 \in \text{Int} (X) \) the starting interior point, \( z^0 < 0 \) the arbitrary negative real number, \( s \) the integer number greater or equal to \( m \) and \( L \geq 0 \) the Lipschitz’s constant.

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

\[ H^k(x^k)(y - x^k) = -G^k(x^k). \]

Let’s \( y^k \) denotes the solution of this set. This problem is purely linear and can be solved in polynomial time by Gaussian elimination method requiring computations of order \( O(m^3) \) arithmetic operations.

**Step 3. Length of step.** Find the scalars:

\[ \lambda^k = \arg \max_{0 \leq \lambda \leq 1} \phi^k(x^k + \lambda(y^k - x^k)) \]

\[ \lambda^k = \frac{\nabla^T f(x^k)(y^k - x^k)}{L \| y^k - x^k \|^2} \]

\[ \lambda^k = \min \{ \lambda^k_1, \lambda^k_2 \} \]

**Step 4. Updating.** Define the new point:

\[ x^{k+1} = x^k + \lambda^k(y^k - x^k). \]

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

**C. The easily demonstrable properties**

1. The direction \( y^k - x^k \), determined in step 2 of the algorithm is a strict ascent direction of \( \phi^k(x) \) at \( x^k \in \text{Int} (X) \).

From step 2 of the algorithm, it can be seen that:

\[ (y^k - x^k)^T H^k(x^k)(y^k - x^k) < 0, \] 

using the strict concavity of \( \phi^k(x) \), it follows that:

\[ (G^k(x^k))^T (y^k - x^k) > 0. \]

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

Being the feasible set \( X \) convex in \( \mathbb{R}^n \), the proof can be completely derived from steps 3 and 4 of the algorithm.

**D. The reduction of the potential function value**

It is known that:

\[ \phi^1(x) = s \ln(\nabla f(x^k)(x-x^k) - z^k) + \sum_{i=1}^{m} \ln(g_i(x)) \]

\[ \phi^{k+1}(x) = s \ln(\nabla f(x^{k+1})(x-x^{k+1}) - z^{k+1}) + \sum_{i=1}^{m} \ln(g_i(x)) \]

\[ z^{k+1} = (1 - \theta)z^k, \quad 0 < \theta < 1, \quad s \geq m \]

\[ \phi^{k+1}(x) - \phi^k(x) = \ln \left( \frac{\nabla f(x^{k+1})(x-x^{k+1}) - z^{k+1}}{\nabla f(x^k)(x-x^k) - z^k} \right) \]

now when \( x = x^{k+1} \), then:

\[ \phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) = \ln \left( \frac{-z^{k+1}}{\nabla f(x^k)(x^{k+1}-x^k) - z^k} \right)^x \]

\[ \phi^{k+1}(x^{k+1}) - \phi(x^{k+1}) = \ln \left( \frac{-(1-\theta)z^k}{\nabla f(x^k)(x^{k+1}-x^k) - z^k} \right)^x \]

\[ \phi^{k+1}(x^{k+1}) - \phi(x^{k+1}) = \ln(1-\theta) \left( \frac{-z^k}{\nabla f(x^k)(x^{k+1}-x^k) - z^k} \right)^x \]

being \( 0 < \theta \leq 1 \) then, it can be seen that:

\[ \phi^{k+1}(x^{k+1}) - \phi^k(x^{k+1}) \leq \ln(1-\theta)^x \leq 0, \]

which means that the function \( \phi^k(x) \) goes to \(-\infty\) when \( k \) goes to \(+\infty\).

**E. The available solution at O(m |ln ε|) iterations can be converted to an ε-optimal solution**

Let’s \( z^* \) denotes the value of the objective function \( f(x) \) at the optimal solution of \((NLP)\) then:
\[
\frac{z^* - f(x^{k+1})}{z^* - f(x^k)} = \frac{z^* - f(x^{k+1}) + f(x^k) - f(x^k)}{z^* - f(x^k)} .
\]

Being the objective function \( f(x) \) concave then:
\[
f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)(x^{k+1} - x^k) \quad \text{and} \quad z^* - f(x^{k+1}) \leq 1 - \frac{f(x^{k+1}) - f(x^k)}{z^* - f(x^k)} .
\]

Using the inequality (1) and the concavity of \( f(x) \), it can be seen that:
\[
z^* \leq f(x^{k+1}) + \sum_{i=1}^{m} u_i g_i(x^{k+1}) , \quad \text{which implies to:}
\]
\[
z^* \leq f(x^k) + \nabla^T f(x^k)(x^{k+1} - x^k) + \sum_{i=1}^{m} u_i g_i(x^{k+1}) .
\]

But we have \( u_i g_i(x^{k+1}) = \frac{\nabla^T f(x^k)(x^{k+1} - x^k)}{s} \), when
\[
z^k \xrightarrow{k \to \infty} 0 , \quad \text{then}
\]
\[
z^* - f(x^k) \leq \nabla f(x^k)(x^{k+1} - x^k) + \sum_{i=1}^{m} u_i g_i(x^{k+1}).
\]

We have \( u_i g_i(x^{k+1}) = \frac{\nabla^T f(x^k)(x^{k+1} - x^k)}{s} \), when
\[
z^k \xrightarrow{k \to \infty} 0 , \quad \text{then}
\]
\[
z^* - f(x^k) \leq (1 + \frac{m}{s}) \nabla f(x^k)(x^{k+1} - x^k) .
\]

\[
z^* - f(x^{k+1}) - \frac{1}{1 + \frac{m}{s}} \nabla f(x^k)(x^{k+1} - x^k) = z^* - f(x^k) \leq \frac{1}{1 + \frac{m}{s}} \left( 1 - \frac{m}{s} \right) \nabla f(x^k)(x^{k+1} - x^k) .
\]

Being \( f(x) \) concave and \( 0 < \theta \leq 1 \), then:
\[
z^* - f(x^{k+1}) = 1 - \left( 1 + \frac{m}{s} \right)^{-1} \nabla f(x^k)(x^{k+1} - x^k) .
\]

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

The aim is to find the number of iterations \( K \) so that:
\[
\ln(z^* - f(x^{k+1})) \leq \ln \varepsilon \quad \Rightarrow \quad - \frac{(k+1)\theta}{m+s} + \ln \varepsilon \leq - \frac{(k+1)\theta}{m+s} + \ln \varepsilon \quad \Rightarrow \quad - \frac{(k+1)\theta}{m+s} \leq \ln \left( \frac{\varepsilon}{z^* - f(x^0)} \right) \quad \Rightarrow \quad k \geq -1 \frac{m+s}{\theta} \ln \left( \frac{\varepsilon}{z^* - f(x^0)} \right).
\]

F. Convergence analysis

From the mean value theorem, it can be seen that:
\[
f(x^k) - f(x^{k+1}) = \nabla f(x^k)(x^k - x^{k+1}), \quad \text{where} \quad \xi \in [x^k, x^{k+1}] \quad \text{and} \quad x^{k+1} = x^k + \lambda (x^k - x^k)
\]
\[
f(x^k) - f(x^{k+1}) = \nabla f(x^k)(x^k - x^{k+1}) + \nabla f(x^k)(x^{k+1} - x^k).
\]

Being the first derivative of the objective function \( f(x) \) satisfies the following condition of Lipschitz: there is \( L \geq 0 \) such that:
\[
\| \nabla^T f(\xi) - \nabla^T f(x^k) \| \leq L \| \xi - x^k \| , \quad \text{it can be found:}
\]
\[
f(x^k) - f(x^{k+1}) \leq L\| \xi - x^k \| \| x^k - x^{k+1} \| + \nabla f(x^k)(x^k - x^{k+1})
\]
\[
f(x^k) - f(x^{k+1}) \leq L \| x^k - x^{k+1} \|^2 + \nabla f(x^k)(x^k - x^{k+1})
\]
\[
f(x^k) - f(x^{k+1}) \leq L\| \lambda \| \| y^k - x^k \| \| \lambda^2 \| \| y^k - x^k \| \| y^k - x^k \| .
\]

Choosing \( 0 \leq \lambda \leq \frac{\nabla^T f(x^k)(y^k - x^k)}{L\| y^k - x^k \|^2} , \text{ then it can be seen that:} \quad f(x^{k+1}) \geq f(x^k) , \quad \text{this means that the value of} \quad f(x) \quad \text{increase in each iteration.}
\]

Being the function \( f(x) \) is upper bounded and monotonically, then \( f(x^{k+1}) - f(x^k) \xrightarrow{k \to \infty} 0 \) and
consequently $\|x^{k+1} - x^k\| - \veps \to 0$, this means that the point $x^{k+1} \approx x^k$ is an accumulation point in the decision space. 

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

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

Taking 

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

it can be found:

$$\nabla f(x^k) + \sum_{i=1}^m u_i \nabla g_i(x^k) = 0, \quad u_i \geq 0 \quad (i = 1, \ldots, m),$$

$$g_i(x^k) \geq 0 \quad (i = 1, \ldots, m) \text{ and }$$

$$g_i(x^k)u_i = \frac{-\nabla f(x^k)(x^{k+1} - x^k) - z^k}{s} \quad (i = 1, \ldots, m).$$

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 $\text{Int}(X)$ 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) [8], it can be concluded that the accumulation point $x^k$ which is found by the algorithm is an $\veps$-optimal solution of $\text{NLP}$ in $X$.

III. STATEMENT OF THE MULTIOBJECTIVE NONLINEAR PROGRAMMING PROBLEM (MONLP)

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

$$\begin{align*}
\text{maximize} & \quad v_1 = v_1(x) \\
\text{maximize} & \quad v_2 = v_2(x) \\
\text{maximize} & \quad v_r = v_r(x) \\
\text{subject to} & \quad g_i(x) \geq 0 \quad (i = 1, \ldots, m) \\
\end{align*}$$

(MONLP)

where the functions $v_i(x) (i = 1, \ldots, r)$ and $g_i(x) (i = 1, \ldots, m)$ are concave with continuous first and second-order derivatives. The first derivatives of $v_i(x) (i = 1, \ldots, r)$ satisfy the Lipschitz’s condition in $x$ on $X$, where the feasible set $X = \{x \in \mathbb{R}^n / g_i(x) \geq 0 (i = 1, \ldots, m)\}$ is compact and convex in the real space $\mathbb{R}^n$. The interior of the feasible region (denoted $\text{Int}(X)$) 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 $U(v) = U(v_1, \ldots, v_r)$. 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 $v_i(x) (i = 1, \ldots, r)$. It is also assumed that the first derivative of $U(v) = U(v_1, \ldots, v_r)$ satisfies the Lipschitz’s condition in $v$ on $V(X)$.

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

Consider the following relation:

$$\begin{align*}
R' \ni V(X) \leftrightsquigarrow V \ni X \subseteq \mathbb{R}^n \\
\varphi \ni U \ni R \\
\end{align*}$$

where $\varphi = UoV$ and the gradient of the utility function with respect of $x$ is given as follows:

$$\nabla_x \varphi(x) = \sum_{j=1}^r \frac{\partial U(v)}{\partial v_j} \nabla_v v_j(x).$$

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

$$\forall x, x^* \in X, v_j(x) \leq v_j(x^*) + \nabla_v v_j(x)(x^* - x) \quad (j = 1, \ldots, r),$$

then:

$$\sum_{j=1}^r \frac{\partial U}{\partial v_j}(v_j(x^*) - v_j(x)) \leq \sum_{j=1}^r \frac{\partial U}{\partial v_j} \nabla_v v_j(x^*)(x^* - x).$$

Using the last inequality, it can be found that:

$$\nabla \varphi(x)(x^* - x) = \sum_{j=1}^r \frac{\partial U(v)}{\partial v_j} \nabla_v v_j(x)(x^* - x) \geq \sum_{j=1}^r \frac{\partial U(v)}{\partial v_j} (v_j(x^*) - v_j(x)) = \nabla U(v(x))(v(x^*) - v(x))$$

As the function $U$ is concave on $V(X)$, then:
\[ \nabla \varphi(x^* - x) \geq \nabla U(v(x)) (v(x^*) - v(x)) \geq U(v(x^*)) - U(v(x)) = \varphi(x^*) - \varphi(x) \]

So that \( \varphi(x^*) - \varphi(x) \leq \nabla \varphi(x^*) (x^* - x) \), 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 the Lipschitz’s condition on the objective space \( V(X) \), then the derivative of the function \( \varphi(x) = U(v_1(x), ..., v_r(x)) \) satisfies the Lipschitz’s condition on the decision space \( X \).

It is easy to see that:

\[
\nabla \varphi(x) = \sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j} \nabla v_j(x) = \sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j} (\nabla v_j(x) - \nabla v_j(x^*))
\]

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

\[
|\nabla v_j(x^2) - \nabla v_j(x^1)| \leq L \| x^2 - x^1 \|
\]

The function \( U(v) = U(v_1, ..., v_r) \) is strictly increasing \( \frac{\partial U}{\partial v_j} > 0 \) \( (j = 1, ..., r) \), then it can be found:

\[
\left| \sum_{j=1}^{r} \frac{\partial U(v)}{\partial v_j} (\nabla v_j(x^2) - \nabla v_j(x^1)) \right| \leq L \sum_{j=1}^{r} \left| \frac{\partial U(v)}{\partial v_j} \right| \| x^2 - x^1 \| \leq L \| x^2 - x^1 \|
\]

So \( |\nabla \varphi(x^2) - \nabla \varphi(x^1)| \leq L \| x^2 - x^1 \| \), then the derivative of the function \( \varphi(x) \) satisfies the condition of Lipschitz on the decision space \( X \).

**A. Approximate gradient**

The multiobjective nonlinear programming problem (MONLP) is ambiguous since usually the objective are conflicting and pursuing the optimum with respect to each objective will lead to different solutions. This ambiguity may be solved by introducing a utility function \( U(v) = U(v_1, ..., v_r) \), defined onto the space of objectives \( V(X) \) and presented by the decision-maker. This function have to satisfy certain conditions as being continuously differentiable, concave, and strictly increasing on the objective space and his derivative satisfies the Lipschitz’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 iterate. The gradient of the utility function in the decision space \( X \) could be given as follows:

\[
\nabla \varphi(x) = \frac{\partial U(v)}{\partial v_1} \nabla v_1(x) + + \frac{\partial U(v)}{\partial v_r} \nabla v_r(x)
\]

\[
= \left( \frac{\partial v_1(x)}{\partial x_1}, ..., \frac{\partial v_r(x)}{\partial x_r} \right) \times \nabla U(v)
\]

In matrix form, the gradient can be written as:

\[
\nabla \varphi(x) = C \times \nabla U(v)
\]

Where \( \nabla U(v) = \left( \frac{\partial U(v)}{\partial v_1}, ..., \frac{\partial U(v)}{\partial v_r} \right)^T \),

\[
\nabla \varphi = \left( \frac{\partial \varphi(x)}{\partial x_1}, ..., \frac{\partial \varphi(x)}{\partial x_n} \right)^T \text{ and }
\]

\[
C = \left( \frac{\partial v_1(x)}{\partial x_1}, ..., \frac{\partial v_r(x)}{\partial x_n} \right)
\]

Therefore, to find the approximate gradient \( \nabla \varphi(x) \) in the decision space we have to evaluate the gradient of the utility function \( \nabla U(v) \) 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 iterate, \( x^0 \) along a specific step direction to \( r \) end points \( x^i \) \( (i = 1, ..., r) \) 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 iterate $x^0$ to the set of $r$ new iterates can be approximated through a first order Taylor’s expansion as follows:

$$\varphi(x^1) = \varphi(x^0) + \nabla_x \varphi(x) \times (x^1 - x^0)$$

$$\varphi(x^r) = \varphi(x^0) + \nabla_x \varphi(x) \times (x^r - x^0)$$

$$\varphi(x^1) = \varphi(x^0) + \nabla^T U(v) \times C^T \times (x^1 - x^0)$$

$$\varphi(x^r) = \varphi(x^0) + \nabla^T U(v) \times C^T \times (x^r - x^0)$$

It can write these equations as:

$$\varphi(x^1) - \varphi(x^0) = \nabla^T U(v) \times \left( v_1(x^1) - v_1(x^0) \right)$$

$$\varphi(x^r) - \varphi(x^0) = \nabla^T U(v) \times \left( v_r(x^r) - v_r(x^0) \right)$$

In matrix form, we can write:

$$\Delta \varphi = \nabla^T U(v) \times \left( v_1(x^1) - v_1(x^0), \ldots, v_r(x^r) - v_r(x^0) \right)$$

But we have $\nabla \varphi(x) = C \times \nabla^T U(v) \times (\Delta V)^{-1} \times \Delta \varphi$. From this relation, it could be concluded that, the Taylor’s series approximation for the gradient 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 iterates. In the absence of an explicit utility function, these values are unavailable and have to be approximated. One way of assessing relative preferences for the $(r+1)$ value vectors is through the analytic hierarchy process (AHP) (see Saaty(1988) [10], Arbel(1994) [1], and Arbel and Oren(1996)) [2]. To obtain an approximate measure for the utility function at the points of interest we proceed as follows. While the value
of the utility function at the \((r+1)\) points \(\{x^0, x^1, \ldots, x^r\}\) is unknown, we can still evaluate the complete \(r\) - dimensional vector of objective functions value, \(v_i(x) (i = 1, \ldots, r)\) at each of these points. We now present this information in objective space to the decision maker and seek to obtain relative preference for these points. This is accomplished by using the \(AHP\) and involves filling a comparison matrix whose principal eigenvector provides the priority vector showing the relative preference for these points. The priority vector \(pr \in R^{r+1}\) provides now an approximate measure of the vector \(\Delta \varphi\) given through

\[
\Delta \varphi \approx \Delta pr = (pr_1 - pr_o, \ldots, pr_r - pr_o)
\]

Where \(pr_i (i = 0, \ldots, r)\) is the priority of the \(i\) - th iterate as derived by using the \(AHP\) technique. The gradient of the utility function with respect to \(B\). Where \((0, \ldots, r)\) (requirements here are the importance)

The creation of the matrix is as follows:

- Create \((r+1) \times (r+1)\) comparison matrix for \(r+1\) requirements with the aide of the decision maker to provide relative preferences (Requirements here are the \(r\) - dimensional vector \(v_i(x) (i = 1, \ldots, r)\) value obtained at each of the points \(\{x^0, x^1, \ldots, x^r\}\), at the current iterate.

- Estimate the eigenvalues (eigenvector) as follows: E.g. “averaging over normalized columns”

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

**Remark.** If the utility function is available, we could use, at the current iterate, the normalized utility function values at the points \(\{x^0, x^1, \ldots, x^r\}\) as components of the priority vector \(pr\).

\(C. \) A logarithmic barrier function and its derivatives concerning the problem (MONLP)

We associate the following logarithmic barrier function with the primal problem \(MONLP\) (Tlas and Abdul Ghani, 2005):

\[
\psi(x) = \sum_{i=1}^{m} \ln(g_i(x)) (k = 0, 1, \ldots),
\]

where \(x^k \in Int(X)\), \(s\) is an integer number greater or equal to \(m\), this number plays the role of a weight, \(z^k\) is an arbitrary negative real 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 \(\omega^k(x)\):

\[
\nabla \omega^k(x) = \frac{s}{\nabla^2 \varphi(x^k)} \nabla \varphi(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x) (k = 0, 1, \ldots).
\]

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

\[
\nabla^2 \omega^k(x) = -\frac{s}{(\nabla^2 \varphi(x^k)) (x-x^k)^2} \nabla \varphi(x^k) \nabla^2 \varphi(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla^2 g_i(x) - \frac{1}{(g_i(x))^2} \nabla g_i(x) \nabla^2 g_i(x) (k = 0, 1, \ldots).
\]

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

\[
\psi_i^k(x) = s h(\nabla v_i(x^k)(x-x^k) - \beta_i^k) + \sum_{i=1}^{m} \ln(g_i(x)) (i = 1, \ldots, r), (k = 0, 1, \ldots).
\]

Where \(\beta_i^k (i = 1, \ldots, r)\) are arbitrary negative real numbers. The gradient vectors and the Hessian matrixes of \(\psi_i^k(x) (i = 1, \ldots, r)\) are given as:

\[
G_i^k(x) = \nabla \psi_i^k(x) = \frac{s}{\nabla v_i(x^k)(x-x^k) - \beta_i^k} \nabla v_i(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x) (i = 1, \ldots, r), (k = 0, 1, \ldots).
\]
$$H_f(x) = \nabla^2 \psi_f(x) = \frac{s}{(\nabla \psi_f(x^*) - \beta_f)} \nabla \psi_f(x^*) + \sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x) \nabla g_i(x)$$

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 multiobjective nonlinear programming problem (MONLP).

### D. Algorithm for solving MONLP

**Step 1. Initialization.** Let $k=0$ the iteration counter, $\varepsilon > 0$ the tolerance level, $x^0 \in \text{Int } (X)$ the starting interior point, $z^0 < 0$, $\beta^0 < 0 (i = 1, \ldots, r)$, $s \geq m$ and $L \geq 0$ (Lipschitz’s constant).

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

$$H_i^k(x^k) (y - x^k) = -G_i^k(x^k).$$

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 $i = 1, \ldots, r$, find the scalar:

$$\alpha_i = \arg \max \psi_i^k(x^k + \lambda(y_i - x^k))$$

$$0 \leq \lambda \leq 1$$

$$\alpha_2 = \frac{\nabla^T \psi_i(x^k) (y_i - x^k)}{L \| y_i - x^k \|^2}$$

$$\lambda^k = \min \{ \alpha_1, \alpha_2 \}$$

**Step 4. Updating.** For $i = 1, \ldots, r$, define the new point:

$$x_i = x^k + \lambda_i (y_i - x^k),$$

and consequently find:

$$x_0 = x^k, C = \begin{pmatrix}
\frac{\partial v_1(x)}{\partial x_1} + \cdots + \frac{\partial v_r(x)}{\partial x_1} \\
\vdots \\
\frac{\partial v_1(x)}{\partial x_n} + \cdots + \frac{\partial v_r(x)}{\partial x_n} \\
\end{pmatrix},$$

$$\Delta V = \begin{pmatrix}
v_1(x_1) - v_1(x_0), \ldots, v_r(x_1) - v_r(x_0) \\
\vdots \\
v_1(x_r) - v_1(x_0), \ldots, v_r(x_r) - v_r(x_0) \\
\end{pmatrix}.$$
\[ \phi(x^k) - \phi(x^{k+1}) \leq L \| z - x \| \| x^k - x^{k+1} \| + \nabla \phi(x^k)(x^k - x^{k+1}) \]

\[ \phi(x^k) - \phi(x^{k+1}) \leq L \| x^k - x^{k+1} \|^2 + \nabla \phi(x^k)(x^k - x^{k+1}) \]

\[ \phi(x^k) - \phi(x^{k+1}) \leq L \lambda^2 \| y^k - x^k \|^2 - \lambda \nabla \phi(x^k)(y^k - x^k) \]

\[ \phi(x^{k+1}) - \phi(x^k) \geq \lambda \nabla \phi(x^k)(y^k - x^k) - L \lambda^2 \| y^k - x^k \|^2. \]

Choosing \( 0 \leq \lambda \leq \frac{\nabla^T \phi(x^k)(y^k - x^k)}{\| y^k - x^k \|^2} \), then it can be seen that: \( \phi(x^{k+1}) \geq \phi(x^k) \), this means that the value of the function \( \phi(x) \) increase in each iteration.

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

\[ \frac{1}{s} \nabla \phi(x^k)(x^{k+1} - x^k) - z^k \phi(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x^k)} \nabla g_i(x^k) = 0 \]

\[ \nabla \phi(x^k) + \frac{1}{s} \nabla \phi(x^k)(x^{k+1} - x^k) - z^k \phi(x^k) + \sum_{i=1}^{m} \frac{1}{g_i(x^k)} \nabla g_i(x^k) = 0. \]

Taking

\[ u_i = \frac{1}{s} \nabla \phi(x^k)(x^{k+1} - x^k) - z^k \phi(x^k) + \frac{1}{g_i(x^k)} \]

we find:

\[ \nabla \phi(x^k) + \sum_{i=1}^{m} u_i \nabla g_i(x^k) = 0, \quad u_i \geq 0 \quad (i = 1, \ldots, m), \]

\[ g_i(x^k) \geq 0 \quad (i = 1, \ldots, m), \]

\[ g_i(x^k) u_i = \frac{1}{s} \nabla \phi(x^k)(x^{k+1} - x^k) - z^k \phi(x^k) \quad (i = 1, \ldots, m). \]

Where \( z^k \) goes to zero when \( k \) goes to the infinity, this means that the accumulation point \( x^* \) satisfies the KKT conditions. As the proposed algorithm creates a sequence of interior points \( \{ x^k \}_{k=0}^{\infty} \) contained in \( \text{Int}(X) \) 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 \( X \).

IV. CONCLUSION

An algorithm for solving multiobjective nonlinear programming problems is 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 \( \varepsilon \) 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 Lipschitz’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.

V. AN ILLUSTRATIVE EXAMPLE

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

max \( v_1(x) = x_1 \)

max \( v_2(x) = x_2 \)

Subject to:

\[ 3x_1 + 2x_2 \leq 6 \]

\[ x_1 \leq 2 \]

\[ x_2 \leq 2 \]

\[ x_1 \geq 0 \]

\[ x_2 \geq 0 \]

For this example, an initial point is available through \( x^0 = (0.3, 0.7)^T \). Lipschitz’s constant \( L = 4, \quad \theta = 0.2 \), \( s = 12 \) and \( z^0 = -25 \). Assuming that the decision maker’s utility function is given through

\[ U(v) = -v_1^2 - 2v_2^2 + 5v_1 + 8v_2. \]

This vector optimization problem has an optimal solution given through

\[ x^* = (1, 1.5)^T, \quad v_1^* = 1, \quad v_2^* = 1.5, \quad U(v_1^*, v_2^*) = 11.5. \]
Solution results (Current iterate)

<table>
<thead>
<tr>
<th>κ</th>
<th>x_1</th>
<th>x_2</th>
<th>U</th>
<th>κ</th>
<th>x_1</th>
<th>x_2</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
<td>6.03</td>
<td>6</td>
<td>1.0011</td>
<td>1.4418</td>
<td>11.3802</td>
</tr>
<tr>
<td>1</td>
<td>0.5827</td>
<td>1.7212</td>
<td>10.4186</td>
<td>7</td>
<td>0.9918</td>
<td>1.4644</td>
<td>11.4016</td>
</tr>
<tr>
<td>2</td>
<td>0.7540</td>
<td>1.4343</td>
<td>10.5613</td>
<td>8</td>
<td>0.9755</td>
<td>1.4959</td>
<td>11.4176</td>
</tr>
<tr>
<td>3</td>
<td>0.8976</td>
<td>1.3187</td>
<td>10.7541</td>
<td>9</td>
<td>0.9887</td>
<td>1.4789</td>
<td>11.4228</td>
</tr>
<tr>
<td>4</td>
<td>0.9445</td>
<td>1.3218</td>
<td>10.9103</td>
<td>10</td>
<td>0.9807</td>
<td>1.4908</td>
<td>11.4233</td>
</tr>
<tr>
<td>5</td>
<td>0.8931</td>
<td>1.4892</td>
<td>11.1459</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT

The author would like to express his thanks to the director general of the AECS Prof. I. Othman for his continuous encouragement, guidance and support.

REFERENCES


M. Tlas received his M. Sc. in graph theory from Louvain University, Belgium in 1989 and he received the Ph.D. in applied statistics-orientation operations research from Brussels University, Belgium in 1993. From 1993 up till now, he is a principal and senior researcher at the Atomic Energy Commission of Syria. The main topics are operation research, applied statistics, inverse problem, multi-objective optimization and mathematical modeling. He published many papers in the above mentioned fields and he was awarded the ABDEL SALAM prize in mathematics for young Syrian scientists in 2005.