Article

An interactive interior point method for mathematical multi-objective linear optimization problems based on approximate gradients

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

Received 15 January 2024; *Accepted 20 February 2024*; *Published online 10 March 2024*; *Published 1 December 2024*

Abstract
An interactive interior point method for solving multiple-objective linear programming problems has been proposed. The method uses a single-objective linear variant 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 generated by projecting this approximate gradient onto the null space of the feasible region. An interior step can be taken 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 multi-objective example is illustrated using this algorithm.

Keywords multi-objective mathematical programming; multi-criteria decision making; multi-criteria optimization; interactive methods; interior point methods.

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

After the seminal algorithm of (Karmarkar, 1984), for solving linear programming problems in polynomial time $O(n^{3.5}L)$ arithmetic operations, where n is the number of unknown variables including slack (surplus) variables and L is the length of the input data (total number of bits used in the description of the problem data), a great number of the so-called interior point methods for linear programming have been proposed. These methods can be classified in two main categories: The first category is the extensions and variants of Karmarkar's algorithm which can be divided also into two subgroups:

1. The projective algorithms (Karmarkar, 1984; de Ghellinck and Vial, 1986, 1987; Anstreicher, 1986; Darvay, 2003; Naseri and Malek , 2004 ;Yu and Sun, 2009; Wang and Luo, 2015 and Todd and Burell, 1986).
2. The “affine” methods (Barnes, 1986; Gay, 1987; Gill et al, 1986 and Vanderbei et al, 1986).

The second category is the path following approaches as: (Gonzaga, 1989 and Renegar, 1988).

The methods in the second group are polynomially bounded and require $O(n^{0.5}L)$ iterations. The overall complexity is $O(n^3L)$. The projective methods in the first group are also polynomially bounded. They require $O(nL)$ iterations and $O(n^{3.5}L)$ operations.

Following these proposals, it is useful to generalize these ideas of interior point technique to the domain of multi-objective programming. Therefore, two algorithms are proposed for solving single and multiple-objective linear programming problems based on projecting of gradients onto the null space of the feasible region of problem.

The first algorithm developed in this paper is an interior point method for solving single objective linear programming problems. The main idea focuses on a projection operation onto the null space of the feasible region which computes a feasible direction (line search) in each iteration in at most $O(nm^2)$ arithmetic operations, where m is the number of constraints ($m < n$). It is proven that the number of iterations required for the algorithm to converge to a good solution is bounded and estimated to be no more than $O(nL)$ iterations and $O(n^2m^2L)$ arithmetic operations.

The second algorithm proposed is an interior point method for interactive multi-objective linear programming problems. The method uses a single-objective linear variant 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 generated by projecting this approximate gradient onto the null space of the feasible region. It can 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 ε is a predetermined error tolerance known a priori.

The multi-objective linear programming problem is ambiguous since usually the objective functions are conflicting and pursuing the optimum, with respect to each objective, leads to different solutions. This 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 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 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 iterate. In the contrary case, when the utility function is implicitly known the approximate gradient can be found 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 references (Saaty, 1988; Arbel, 1994; Arbel and Oren, 1996; Zhang, 2019).

2 Statement of the Linear Programming Problem (*LP*)

Consider the linear programming problem given in standard form through

$$\begin{aligned} &maximize \quad c^T x \\ &subject\ to \quad Ax = b \\ &\qquad\qquad\quad x \geq 0 \end{aligned} \tag{1}$$

where $c, x \in \mathbf{R}^n, b \in \mathbf{R}^m$ and A is $m \times n$ matrix, n is the number of unknown (decision) variables including slack variables and m is the number of linear constraints so that $m < n$.

Assuming that the feasible set: $X = \left\{x \in \mathbf{R}^n / Ax = b\ and\ \ x \geq 0\right\}$ is compact and convex in the nonnegative orthant of $\mathbf{R}^n$.

It is easy to transform the canonical feasibility problem (1) into the following equivalent affine feasibility problem:

$$\begin{aligned} &maximize \quad \tilde{c}^T y \\ &subject\ to \quad d^T y = 1 \\ &\qquad\qquad\quad \tilde{A}y = 0 \\ &\qquad\qquad\quad y \geq 0 \end{aligned} \tag{2}$$

where $y \in \mathbf{R}^{n+1}$ is the new unknown variables, $\tilde{c}^T = (0, c_1, ..., c_n) \in \mathbf{R}^{n+1}$, $\tilde{A}$ is an $m \times (n+1)$ matrix defined as: $\tilde{a}_{i0} = -b_i\ (i = 1, ..., m)$, $\tilde{a}_{ij} = a_{ij}\ (i = 1, ..., m, j = 1, ..., n)$, and $d^T \in R^{n+1}$ defined as:

$$d_j = \begin{cases} 1\ \ for\ \ j = 0 \\ 0\ \ for\ \ j = 1, ..., n \end{cases}$$

All the constraints in (2) are linear, i.e., homogenous, except the first one $d^T y = 1$ which is affine and can be interpreted as a normalization constraint.

The feasible set of (2): $Y = \left\{y \in \mathbf{R}^{n+1} / \tilde{A}y = 0, d^T y = 1 and\ \ y \geq 0\right\}$ is also compact and convex in the nonnegative orthant of $\mathbf{R}^{n+1}$.

Let Y^k be the diagonal matrix $Y^k = diag\left(y^k\right)$. Using the linear transformation $z = \left(Y^k\right)^{-1} y$, we obtain a transformed problem:

$$\begin{aligned} &maximize \quad \tilde{c}^T Y^k z \\ &subject\ to \quad d^T Y^k z = 1 \\ &\qquad\qquad\quad \tilde{A} Y^k z = 0 \\ &\qquad\qquad\quad z \geq 0 \end{aligned} \tag{3}$$

where $z \in \mathbf{R}^{n+1}$ and the feasible set of (3): $Z = \left\{ z \in \mathbf{R}^{n+1} / \tilde{A} Y^{k} z = 0, d^{T} Y^{k} z = 1 and \ z \geq 0 \right\}$ is also compact and convex in the nonnegative orthant of $\mathbf{R}^{n+1}$.

The following algorithm is designed to work in the relative interior of the feasible set X of (1). It is capable of solving problem (2), where in each iteration, $k \geq 0$, problem (3) is solved on a sphere centered at the point $e^{T} = (1, \ldots, 1) \in \mathbf{R}^{n+1}$ and inscribed in the feasible set Z of problem (3).

2.1 Algorithm for solving *LP* problem

Step 1. *Initialization*. Let $\varepsilon > 0$ be a tolerance level. Let $\left(x^{0}\right)^{T} = \left(x_{1}^{0}, \ldots, x_{n}^{0}\right) \in Int(X)$.

Let $\left(y^{0}\right)^{T} = \left(1, x_{1}^{0}, \ldots, x_{n}^{0}\right)$, $e^{t} = (1, \ldots\ldots, 1) \in R^{n+1}$ and $k = 0$ the iteration counter.

Step 2. *Change*. Let $Y^{k} = diag\left(y^{k}\right)$, $B^{k} = \begin{pmatrix} d^{T} \\ \tilde{A} \end{pmatrix} Y^{k}$

Step 3. *Projection*. Find $p^{k} \in \mathbf{R}^{n+1}$ and $u^{k} \in \mathbf{R}^{m+1}$ which solve the linear system of equations

$$\begin{aligned} p^{k} + \left(B^{k}\right)^{T} u^{k} &= Y^{k} \tilde{c} \\ B^{k} p^{k} &= 0 \end{aligned}$$

$$“P_{k}”$$

where p^{k} is the projection of $Y^{k} \tilde{c}$ onto the null space of B^{k} and u^{k} is the dual variable.

Step 4. *Termination test*. If $\left\| p^{k} \right\| < \varepsilon$ then stop, the point y^{k} is an optimal solution of problem (2) and consequently the point given by $x_{j}^{k} = y_{j}^{k} \ (j = 1, \ldots, n)$ is an optimal solution of problem (1).

Step 5. *Normalization.* Define $q^{k} = \dfrac{p^{k}}{\left\| p^{k} \right\|}$

Step 6. *Line search step*. Find α^{k} which satisfies the following inequalities:

$$\begin{aligned} &\alpha^{k} > 0 \\ &e + \alpha^{k} q^{k} > 0 \\ &e^{T} q^{k} - \frac{\alpha^{k}}{2} + \frac{\left(\alpha^{k}\right)^{2}}{3\left(1 - \alpha^{k}\right)} < 0 \end{aligned}$$

A possible choice for α^{k} which enforces these conditions is $\alpha^{k} \in \left(0,\ 0.6\right]$.

Step 7. *New iterate*. Let

$$z^{k+1} = e + \alpha^k q^k$$
$$y^{k+1} = Y^k z^{k+1}$$

set $k = k+1$ (increment the iteration counter) and return to *step 2*.

Remark. p_k is the set of so-called normal equations whose solution p^k is unique and is the projection of the vector $Y^k \tilde{c}$ onto the null space of the matrix B^k. This problem is purely linear and can be solved in $O(nm^2)$ arithmetic operations.

Lemma 0. When $\left\| p^k \right\| \xrightarrow[k\to\infty]{} 0$, then $e^T p^k \le 0 \quad (\forall k \ge 0)$.

Proof. Taking into consideration the Holder inequality $\left\| p^k \right\| \ge \frac{1}{\sqrt{n+1}} \sum_{i=0}^{n} \left| p_i^k \right|$ and the condition $\left\| p^k \right\| \xrightarrow[k\to\infty]{} 0$, it can be concluded that:

$$\sum_{i=0}^{n} p_i^k \xrightarrow[k\to\infty]{} 0. \qquad \text{(a)}$$

From step 3 of the previous algorithm, we can write the following:

$\sum_{i=0}^{n} p_i^k = \sum_{i=0}^{n} \tilde{c}_i y_i^k$ and $\sum_{i=0}^{n} p_i^{k+1} = \sum_{i=0}^{n} \tilde{c}_i y_i^{k+1}$.

From step 7, we can find that: $y_i^{k+1} - y_i^k = \alpha^k y_i^k q_i^k \quad (i = 0,...,n)$.

So $\sum_{i=0}^{n} p_i^{k+1} - \sum_{i=0}^{n} p_i^k = \sum_{i=0}^{n} \tilde{c}_i \left(y_i^{k+1} - y_i^k \right) = \alpha^k \sum_{i=0}^{n} \tilde{c}_i y_i^k q_i^k = \alpha^k \left\| p^k \right\| \ge 0$, then

$$\sum_{i=0}^{n} p_i^{k+1} \ge \sum_{i=0}^{n} p_i^k \qquad (k = 0,1,...) \qquad \text{(b)}$$

From (a) and (b), it can be concluded that $\sum_{i=0}^{n} p_i^k \le 0 \quad (\forall k = 0,1,...)$, and the proof of the lemma is completed.

2.2 Complexity calculation

First, we propose some propositions needed for calculation the complexity of the algorithm.

Definition. If A is $n \times n$ nonsingular matrix, $b \in \mathbf{R}^n$ and $T_A : \mathbf{R}^n \to \mathbf{R}^n$ so that $T_A(x) = Ax + b$, then T_A is called an affine transformation. Affine transformations have several important priorities. One of them is the affine transformations preserve set inclusion if:

$$T_A(W) = \{\xi \in \mathbf{R}^n : \xi = Ax + b, x \in W\}$$

Proposition 1. If $W \subseteq W' \subseteq \mathbf{R}^n$, then $T_A(W) \subseteq T_A(W') \subseteq \mathbf{R}^n$. So volumes are changed by a constant factor and the relative volumes are preserved.

Proposition 2. If $W \subseteq \mathbf{R}^n$ is full dimensional and convex, then $vol(T_A(W)) = \left|\det A\right| \times vol(W)$.

Considering the propositions 1 and 2, it can be concluded that for any ellipsoid there is an affine transformation which gives a sphere centered at the point $e^T = (1,1,...,1)$ when it is mapping on the given ellipsoid.

Suppose that E^k is an ellipsoid centered at y^k in the iteration number k and E^{k+1} is an ellipsoid centered at y^{k+1} in the iteration number $k+1$.

The inequalities: $\left(\prod_{j=0}^{n} v_j\right)^{\frac{1}{n+1}} \le \frac{1}{n+1}\sum_{j=0}^{n} v_j \quad and\ v_j \ge 0 (j = 0,...,n)$, express the relation between the geometric mean and the arithmetic mean on the non negative orthant of $\mathbf{R}^{n+1}$.

In view of the previous inequality it followed that:

$$\left(\prod_{j=0}^{n}\left(1+\alpha^k q_j^k\right)\right)^{\frac{1}{n+1}} \le \frac{1}{n+1}\sum_{j=0}^{n}\left(1+\alpha^k q_j^k\right) = 1 + \frac{\alpha^k}{n+1}\sum_{j=0}^{n} q_j^k$$

As $\sum_{j=0}^{n} p_j^k \le 0$, then $\sum_{j=0}^{n} q_j^k \le 0$. The last inequality becomes:

$$\left(\prod_{j=0}^{n}\left(1+\alpha^k q_j^k\right)\right)^{\frac{1}{n+1}} \le 1 + \frac{\alpha^k}{n+1}\sum_{j=0}^{n} q_j^k \le 1$$

By applying the proposition 2, it can be concluded that:

$\frac{vol(B^k)}{vol(B^{k+1})} = \frac{\frac{1}{\prod_{j=0}^{n} y_j^k}}{\frac{1}{\prod_{j=0}^{n} y_j^{k+1}}} \frac{vol(E^k)}{vol(E^{k+1})}$, where B^k and B^{k+1} are spheres centered at $e^T = (1,1,...,1)$ with radius $r^k = \alpha^k q^k$ and $r^{k+1} = \alpha^{k+1} q^{k+1}$ $\left(\left\|r^{k+1}\right\| \le \left\|r^k\right\|\right)$, then $\frac{vol(B^k)}{vol(B^{k+1})} = \frac{\prod_{j=0}^{n} y_j^{k+1}}{\prod_{j=0}^{n} y_j^k} \frac{vol(E^k)}{vol(E^{k+1})}$. So

$$\frac{vol\,(E^{k+1})}{vol\,(E^{k})}=\frac{\prod_{j=0}^{n} y_j^{k+1}}{\prod_{j=0}^{n} y_j^{k}}\frac{vol\,(B^{k+1})}{vol\,(B^{k})}\leq\prod_{j=0}^{n}\left(1+\alpha^k q_j^k\right)\leq 1.$$

It follows that $vol\,(E^{k+1})\leq vol\,(E^{k})$, then $vol\,(E^{k})\xrightarrow[k\to\infty]{}0$.

Using the development of Taylor for the function $\log(1+\lambda)$ around $\lambda=0$ with $|\lambda|<1$, it can be seen that:

$\log(1+\lambda)=\lambda-\frac{\lambda^2}{2}+\xi(\lambda)$, where $\xi(\lambda)=\sum_{j=3}^{\infty}\frac{\lambda^j(-1)^{j+1}}{j}$.

So: $|\xi(\lambda)|=\sum_{j=3}^{\infty}\left|\frac{\lambda^j(-1)^{j+1}}{j}\right|=\sum_{j=3}^{\infty}\frac{|\lambda^j|}{j}|-1|^{j+1}=\sum_{j=3}^{\infty}\frac{|\lambda^j|}{j}$.

Since $j\geq 3$, then $\frac{1}{j}\leq\frac{1}{3}$. It follows that: $|\xi(\lambda)|\leq\sum_{j=3}^{\infty}\frac{|\lambda|^j}{3}=\frac{|\lambda|^3}{3\left(1-|\lambda|\right)}$.

So $\log(1+\lambda)\leq\lambda-\frac{\lambda^2}{2}+\frac{|\lambda|^3}{3\left(1-|\lambda|\right)}$.

Let $\lambda=\alpha^k q_j^k$, then $\log(1+\alpha^k q_j^k)\leq\alpha^k q_j^k-\frac{\left(\alpha^k\right)^2}{2}\left(q_j^k\right)^2+\frac{\left|\alpha^k q_j^k\right|^3}{3\left(1-\left|\alpha^k q_j^k\right|\right)}$.

It follows that $\sum_{j=0}^{n}\log(1+\alpha^k q_j^k)\leq\alpha^k\sum_{j=0}^{n}q_j^k-\frac{\left(\alpha^k\right)^2}{2}\sum_{j=0}^{n}\left(q_j^k\right)^2+\sum_{j=0}^{n}\frac{\left|\alpha^k q_j^k\right|^3}{3\left(1-\left|\alpha^k q_j^k\right|\right)}$.

As: $\sum_{j=0}^{n}\left(q_j^k\right)^2=1$, $\sum_{j=0}^{n}\left|\alpha^k q_j^k\right|^3\leq\left\|\alpha^k q^k\right\|^3$, and $\left|\alpha^k q_j^k\right|\leq\left\|\alpha^k q^k\right\|$ $(j=0,\dots,n)$, then the last inequality becomes: $\sum_{j=0}^{n}\log(1+\alpha^k q_j^k)\leq\alpha^k\sum_{j=0}^{n}q_j^k-\frac{\left(\alpha^k\right)^2}{2}+\frac{\left\|\alpha^k q^k\right\|^3}{3\left(1-\left\|\alpha^k q^k\right\|\right)}$

Hence $\sum_{j=0}^{n}\log(1+\alpha^k q_j^k)\leq\alpha^k\sum_{j=0}^{n}q_j^k-\frac{\left(\alpha^k\right)^2}{2}+\frac{\left(\alpha^k\right)^3}{3\left(1-\alpha^k\right)}$.

The value of α^k should be chosen so that $\sum_{j=0}^{n}q_j^k-\frac{\alpha^k}{2}+\frac{\left(\alpha^k\right)^2}{3\left(1-\alpha^k\right)}<0$, then it can be written that:

$\sum_{j=0}^{n}\log(1+\alpha^k q_j^k)\leq-\frac{\beta}{n},\ \beta>0$. So $\prod_{j=0}^{n}(1+\alpha^k q_j^k)\leq e^{-\frac{\beta}{n}},\ \beta>0$.

From this relation and since $vol(E^{k+1}) \le vol(E^k) \times \prod_{j=0}^{n} \left(1+\alpha^k q_j^k\right)$, it follows that

$vol(E^{k+1}) \le vol(E^k) \times e^{-\frac{\beta}{n}} \le ... \le vol(E^0) \times e^{-\beta\frac{k+1}{n}}$, that gives $vol(E^{k+1}) \le V^0 \times e^{-\beta\frac{k+1}{n}}$.

Choosing $\varepsilon = e^{-dL}$, where d is a positive number and L is the length of the input data (total number of bits used in the description of the problem data), then $vol(E^{k+1}) \le \varepsilon$ after K iterations, where $K = \left\| \left\lfloor -1 + \frac{n}{\beta} \log\left(\frac{\varepsilon}{V^0}\right)^{-1} \right\rfloor \right\| + 1 = O(nL)$, and $\lfloor u \rfloor$ denotes the integer part of $u \ge 0$.

From step 3 of the algorithm, the problem P_k is a set of the so-called normal equations which has a unique solution. It is known that this problem is purely linear and can be solved in $O(nm^2)$ arithmetic operations. The proposed algorithm stops in no more than $O(nL)$ iterations, then it can be seen that the complexity of the algorithm is $O(n^2m^2L)$. Consequently, the solution of the problem LP is reached in polynomial time.

2.3 Convergence analysis

From the proposed algorithm, we have $y^k \in Y$ so that $p^k = 0$. Suppose that there is another point $y^* \in Y$ so that $\tilde{c}^T y^* > \tilde{c}^T y^k$. This implies $\tilde{c}^T (y^* - y^k) > 0$ which means that the vector $d^* = y^* - y^k$ is an improvement direction. Since $y^{k+1} = y^k + \alpha^k Y^k q^k \in Y$, then let $w = y^k + \alpha d^*$ where $0 < \alpha \le 1$. If $d^* \ne Y^k q^k$ (matrix $\tilde{A}$ is full rank), then $\tilde{A} d^* \ne \tilde{A} Y^k q^k = 0$, this implies $\tilde{A} d^* \ne 0$ which means the point w is not feasible.

For $\alpha = 1$,we get $y^* = w$ and that conflicts with the fact that y^* is a feasible point. Then it must be $d^* = Y^k q^k$ which means clearly $y^{k+1} = w$.

For $\alpha = 1$, we have $y^{k+1} = y^k = w$ and this means that y^k is an accumulation point in Y, the condition of optimality of Karush-Kuhn-Tuker (*KKT*) in accumulation point y^k are given as follows:

$\exists u^k \in \mathbf{R}^{m+1}, \lambda \in \mathbf{R}_+^{n+1}$:

$$\tilde{c} - \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k - I\lambda = 0$$

$$\lambda_i y_i^k = 0 \quad (i = 0,...,n)$$

To demonstrate the verification of these conditions, from step 3 of the algorithm, we can write the following equation (at $p^k = 0$):

$$Y^k \tilde{c} - Y^k \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k = 0$$

As the proposed algorithm creates a sequence of points $\left\{y^k\right\}_{k=0,1,...}$ contained in Y set with $y_j^k > 0\ (j = 0,...,n)$ and $k \geq 0$, then it can be concluded that:

$$\tilde{c} - \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k = 0$$

As a result of taking $\lambda_j = 0\ (j = 0,...,n)$, the conditions of optimality of *KKT* in point y^k are satisfied. Sequence $\left\{y^k\right\}_{k=0,1,...}$ converges to a solution that satisfies the conditions of optimality of *KKT* of problem (2). Consequently, this succession creates a sequence of points $\left\{x^k\right\}_{k=0,1,...}$ contained in X and converges to an optimal solution of problem (1).

3 Statement of the Multi-Objective Linear Programming Problem (*MOLP*)

A multi-objective linear programming (*MOLP*) problem is generally described through the standard formulation:

$$\begin{aligned} &maximize \quad v_1 = c_1^T x \\ &maximize \quad v_2 = c_2^T x \\ &. \\ &. \\ &maximize \quad v_r = c_r^T x \\ &subject\ to \quad Ax = b \\ &\qquad\qquad\quad x \geq 0 \end{aligned}$$

(4)

where $c_1,..,c_r, x \in \mathbf{R}^n, b \in \mathbf{R}^m$ and A is $m \times n$ matrix, n is the number of unknown or decision variables including slack variables, m is the number of linear constraints such that ($m < n$), and r is the number of objective functions. Assuming that the feasible set: $X = \left\{x \in \mathbf{R}^n / Ax = b\ and\ \ x \geq 0\right\}$ is compact and convex in the nonnegative orthant of $\mathbf{R}^n$.

In multi-objective 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 $V(X)$ which is the image of the feasible set X (decision space) by the objective functions v_i $(i=1,\ldots,r)$, its derivative 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:

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

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

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

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

The derivatives of the functions v_j $(j=1,...,r)$ satisfy the Lipschitz's condition on X because they are linear, 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 $\dfrac{\partial \mathrm{U}}{\partial v_j}>0\ (j=1,...,r)$, then it can be found

$\left|\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}(\nabla_x v_j(x^2)-\nabla_x v_j(x^1))\right|\leq L\sum_{j=1}^{r}\frac{\partial U(v)}{\partial v_j}\left\|x^2-x^1\right\|\leq L'\left\|x^2-x^1\right\|$. 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 Lipschitz on the decision space X .

Lemma 3. If the derivative of any function $\phi(x)$ satisfies the following Lipschitz's condition $\left\|\phi'(x)-\phi'(y)\right\|\leq L\left\|x-y\right\|$, then $\phi(x)-\phi(y)\leq\langle\phi'(x),x-y\rangle+\frac{1}{2}L\left\|x-y\right\|^2$.

The following inequality is always true in the Euclidean space:

$\langle x,y\rangle\leq\|x\|\|y\|$ (Cauchy-Bunyakovskii inequality) and

$$\phi(x)-\phi(y)=\int_0^1\langle\phi'(y+\tau(x-y)),(x-y)\rangle d\tau \quad \text{(Cauchy-Bunyakovskii equality)}$$

$$=\langle\phi'(x),x-y\rangle+\int_0^1\langle\phi'(y+\tau(x-y))-\phi'(x),(x-y)\rangle d\tau$$

$$\leq\langle\phi'(x),x-y\rangle+\int_0^1\left\|\phi'(y+\tau(x-y))-\phi'(x)\right\|\left\|x-y\right\|d\tau$$

$$= \langle \phi'(x), x-y \rangle + \|x-y\| \int_0^1 \|\phi'(y+\tau(x-y)) - \phi'(x)\| d\tau$$

$$\le \langle \phi'(x), x-y \rangle + L\,\|x-y\| \int_0^1 \|y+\tau(x-y) - x\| d\tau$$

$$= \langle \phi'(x), x-y \rangle + L\,\|x-y\|^2 \int_0^1 (1-\tau) d\tau$$

$$= \langle \phi'(x), x-y \rangle + L\,\|x-y\|^2 \left[\frac{(1-\tau)^2}{2} \right]_1^0 \Rightarrow$$

$\phi(x) - \phi(y) \le \langle \phi'(x), x-y \rangle + \frac{1}{2} L\,\|x-y\|^2$. The proof of the lemma is completed.

3.1 Approximate gradient

The multi-objective linear programming (*MOLP*) problem (4) is ambiguous since the objective are conflicting and pursuing the optimum with respect to each objective leads to different solutions. This 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 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 is given as follows:

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

$$\nabla_x \varphi(x) = \frac{\partial U(v)}{\partial v_1} c_1 + ... + \frac{\partial U(v)}{\partial v_r} c_r$$

where $c_i = \begin{pmatrix} c_i^1 \\ . \\ . \\ c_i^n \end{pmatrix} (i = 1, ..., r)$

in matrix form, the gradient can be written as

$$\nabla_x \varphi(x) = \begin{pmatrix} c_1^1 \ldots c_r^1 \\ . \\ . \\ c_1^n \ldots c_r^n \end{pmatrix} \begin{pmatrix} \frac{\partial U(v)}{\partial v_1} \\ . \\ . \\ \frac{\partial U(v)}{\partial v_r} \end{pmatrix} = C \times \nabla_v U(v)$$

where $C = (c_1 \ldots c_r)$, $\nabla_v U(v) = \left(\frac{\partial U(v)}{\partial v_1}, \ldots, \frac{\partial U(v)}{\partial v_r} \right)^T$, and $\nabla_x \varphi = \left(\frac{\partial \varphi(x)}{\partial x_1}, \ldots, \frac{\partial \varphi(x)}{\partial x_n} \right)^T$.

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 objective 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 $(i = 1, \ldots, r)$ 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 iterate x^0 to the set of r new iterates can be approximated through a first order Taylor's expansion as follows:

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

These equations can be rewritten as:

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

In matrix form, we can write

$$\varphi(x^1)-\varphi(x^0)=\nabla_v^T U(v)\times\begin{pmatrix} c_1^T x^1 - c_1^T x^0 \\ . \\ . \\ c_r^T x^1 - c_r^T x^0 \end{pmatrix}$$

.

.

$$\varphi(x^r)-\varphi(x^0)=\nabla_v^T U(v)\times\begin{pmatrix} c_1^T x^r - c_1^T x^0 \\ . \\ . \\ c_r^T x^r - c_r^T x^0 \end{pmatrix}$$

Or

$$\Delta\varphi=\begin{pmatrix} c_1^T x^1 - c_1^T x^0 .....c_r^T x^1 - c_r^T x^0 \\ . \\ . \\ c_1^T x^r - c_1^T x^0 .....c_r^T x^r - c_r^T x^0 \end{pmatrix}\times\nabla_v U(v)$$

$$\Delta\varphi=\Delta V \times\nabla_v U(v)$$

$$\Delta V_{ij} = c_j^T x^i - c_j^T x^0 \quad \begin{pmatrix} i=1,...,r \\ j=1,...,r \end{pmatrix}$$

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

$$\nabla_x\varphi(x)=C\times\nabla_v U(v)$$

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

From this relation, it can 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 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) (Saaty, 1988; Arbel, 1994; Arbel and Oren, 1996; Zhang, 2019).

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 $\{x^0, x^1,..., x^r\}$ is unknown, we can still evaluate the complete $r-$dimensional vector of objective functions value, $v_i(x)=c_i^T x$ $(i=1,...,r)$ 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 $pr \in \mathbf{R}^{r+1}$ provides now 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 $i-th$ iterate as derived by using the AHP technique. 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 $v_i(x) = c_i^T x\ (i = 1,...,r)$ value obtained at each of the $(r+1)$ points $\{x^0, x^1, ..., x^r\}$, at the current iterate.

- 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 $v_i(x) = c_i^T x\ (i = 1,...,r)$ value obtained at each of the points $\{x^0, x^1, ..., x^r\}$, at the current iterate)

  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. Notice that, if the utility function is available, we can 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 .

3.3 Algorithm for solving *MOLP* problem

Step 1. *initialization*. Let $\varepsilon > 0$ be a tolerance level. Let $\left(x^0\right)^T = \left(x_1^0, \ldots, x_n^0\right) \in Int\,(X\,)$

Let $L \geq 0$ (Lipschitz's constant), $\left(y^0\right)^T = \left(1, x_1^0, \ldots, x_n^0\right)$, $e^t = (1,, \ldots\ldots, 1) \in R^{n+1}$ and $k = 0$ (iteration counter)

Step 2. *Change*. Let $Y^k = diag\left(y^k\right)$, $B^k = \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix} Y^k$

Step 3. *Projection.* For $i = 1, \ldots, r$, find $p_i \in \mathbf{R}^{n+1}$ and $u_i \in \mathbf{R}^{m+1}$ which solve the linear system of equations

$$p_i + \left(B^k\right)^T u_i = Y^k \tilde{c}_i$$
$$B^k p_i = 0$$

where p_i is the projection of $Y^k \tilde{c}_i$ onto the null space of B^k and u_i is the dual variable.

Step 4. *Normalization.* For $i = 1, \ldots, r$, define $q_i = \dfrac{p_i}{\left\| p_i \right\|}$

Step 5. *Line search step*. For $i = 1, \ldots, r$, find α_i which satisfies the following inequalities:

$$\alpha_i > 0$$
$$e + \alpha_i q_i > 0$$
$$e^T q_i - \frac{\alpha_i}{2} + \frac{\alpha_i^2}{3\left(1 - \alpha_i\right)} < 0$$

A possible choice for α_i $(i = 1, \ldots, r)$ which enforces these conditions is $\alpha_i \in \left(0,\ 0.6\right]$.

Step 6. *New iterate*. For $i = 1, \ldots, r$, find

$$z_i = e + \alpha_i q_i$$
$$y_i = Y^k z_i$$
$$x_i = y_i$$

And consequently find

$$x_0 = x^k \ , \ C = (c_1 \ldots c_r) = \begin{pmatrix} c_1^1 \ldots c_r^1 \\ . \\ . \\ c_1^n \ldots c_r^n \end{pmatrix}, \quad \Delta V = \begin{pmatrix} c_1^T x_1 - c_1^T x^k \ldots . c_r^T x_1 - c_r^T x^k \\ . \\ . \\ c_1^T x_r - c_1^T x^k \ldots . c_r^T x_r - c_r^T x^k \end{pmatrix} ,$$

$$\Delta\varphi = \left(\varphi(x_1) - \varphi(x^k), \ldots, \varphi(x_r) - \varphi(x^k)\right)^T \text{ or } \Delta\varphi = \left(pr_1 - pr_0, \ldots, pr_r - pr_0\right)^T ,$$

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

$$\nabla_y \varphi(y^k) = \begin{pmatrix} 0 \\ \nabla_x \varphi(x^k) \end{pmatrix}$$

Step7. *Projection.* Find $p^k \in \mathbf{R}^{n+1}$ and $u^k \in \mathbf{R}^{m+1}$ which solve the linear system of equations

$$p^k + \left(B^k\right)^T u^k = Y^k \nabla_y \varphi(y^k)$$
$$B^k p^k = 0$$

Step 8. *Termination test.* If $\left\|p^k\right\| < \varepsilon$ then stop, the point x^k is an optimal solution of problem (4)

Step 9. *Updating*. Choose an arbitrary $\alpha^k \in (0\ 1]$ and let

$$y^{k+1} = Y^k \left(e + \alpha^k \frac{p^k}{L + \left\|p^k\right\|}\right)$$

$$x^{k+1} = y^{k+1}$$

Set $k = k + 1$ (increment the iteration counter)and return to *step 2*.

Lemma 4. If the derivative of the utility function φ satisfies the Lipschitz's condition on the decision space, then $\left\|p^k\right\| \xrightarrow[k \to \infty]{} 0$

Applying the lemma 3, the following could be written

$$\varphi(y^k) - \varphi(y^{k+1}) \le \left\langle \nabla_y^T \varphi(y^k), y^k - y^{k+1} \right\rangle + \frac{1}{2} L \left\|y^k - y^{k+1}\right\|^2$$

$$\varphi(y^k) - \varphi(y^{k+1}) \le -\alpha^k \nabla_y^T \varphi(y^k) \frac{Y^k p^k}{L + \left\|p^k\right\|} + \frac{1}{2} L \left(\alpha^k\right)^2 \left(\frac{Y^k p^k}{L + \left\|p^k\right\|}\right)^2$$

$$\varphi(y^k) - \varphi(y^{k+1}) \le -\alpha^k \frac{\left\|p^k\right\|^2}{L + \left\|p^k\right\|} + \frac{1}{2} L \left(\alpha^k\right)^2 \frac{\left\|Y^k p^k\right\|^2}{\left(L + \left\|p^k\right\|\right)^2}$$

$$\varphi(y^{k+1})-\varphi(y^{k})\geq\alpha^{k}\frac{\left\|p^{k}\right\|^{2}}{L+\left\|p^{k}\right\|}-\frac{1}{2}L\left(\alpha^{k}\right)^{2}\frac{\left\|Y^{k}p^{k}\right\|^{2}}{\left(L+\left\|p^{k}\right\|\right)^{2}}$$

Since $0\leq y_{i}^{k}\leq 1\ (i=0,...,n)$ then $\left\|Y^{k}p^{k}\right\|^{2}\leq\left\|p^{k}\right\|^{2}$ and

$$\varphi(y^{k+1})-\varphi(y^{k})\geq\alpha^{k}\frac{\left\|p^{k}\right\|^{2}}{L+\left\|p^{k}\right\|}-\frac{1}{2}L\left(\alpha^{k}\right)^{2}\frac{\left\|Y^{k}p^{k}\right\|^{2}}{\left(L+\left\|p^{k}\right\|\right)^{2}}\geq\left(1-\frac{1}{2}\alpha^{k}\frac{L}{L+\left\|p^{k}\right\|}\right)\alpha^{k}\frac{\left\|p^{k}\right\|^{2}}{L+\left\|p^{k}\right\|}$$

then

$$\varphi(y^{k+1})-\varphi(y^{k})\geq\left(1-\frac{1}{2}\alpha^{k}\right)\alpha^{k}\frac{\left\|p^{k}\right\|^{2}}{L+\left\|p^{k}\right\|}$$

Choosing $0<\alpha^{k}\leq 1$ and the function φ is upper bounded and monotonically, then

$\varphi(y^{k+1})-\varphi(y^{k})\xrightarrow[k\to\infty]{}0$ and consequently $\left\|p^{k}\right\|\xrightarrow[k\to\infty]{}0$, then $\left\|y^{k}-y^{k+1}\right\|\xrightarrow[k\to\infty]{}0$, this means that the point y^{k} and consequently $x_{j}^{k}=y_{j}^{k}\ (j=1,...,n)$ is an optimal solution in the decision space X .

3.4 Convergence analysis

The proposed algorithm produces a feasible point y^{k} so that $p^{k}=0$. Suppose now that there is another feasible point y^{*} so that $\varphi(y^{*})>\varphi(y^{k})$. As the function φ is concave, so it can be concluded that $\nabla_{y}^{T}\varphi(y^{k})(y^{*}-y^{k})>0$ which implies that the vector $d^{*}=y^{*}-y^{k}$ is an improvement direction. Besides, $y^{k+1}=y^{k}+\alpha^{k}\dfrac{Y^{k}p^{k}}{L+\left\|p^{k}\right\|}$ is feasible. Let $w^{*}=y^{k}+\alpha d^{*}$; $0<\alpha\leq 1$. Now if $d^{*}\neq\dfrac{Y^{k}p^{k}}{L+\left\|p^{k}\right\|}$ (matrix $\tilde{A}$ is full rank), then $\tilde{A}d^{*}\neq\tilde{A}\dfrac{Y^{k}p^{k}}{L+\left\|p^{k}\right\|}=0$. It follows that $\tilde{A}d^{*}\neq 0$. So the point w^{*} is not feasible for any $0<\alpha\leq 1$.

For $\alpha=1$,it follows that $y^{*}=w^{*}$. This conflicts with the fact that y^{*} is a feasible point. Then it must be $d^{*}=\dfrac{Y^{k}p^{k}}{L+\left\|p^{k}\right\|}$ which means clearly that $y^{k+1}=w^{*}$.

For $\alpha=1$,it can be seen that $y^{k+1}=y^{k}=w^{*}$ and this means that y^{k} is an accumulation point, the condition of optimality of Karush-Kuhn-Tuker (*KKT*) in accumulation point y^{k} are given as follows:

$\exists u^k \in \mathbf{R}^{m+1}, \lambda \in \mathbf{R}_+^{n+1}$:

$$\nabla_y \varphi(y^k) - \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k - I\lambda = 0$$

$$\lambda_i y_i^k = 0 \quad (i = 0,...,n)$$

To demonstrate the verification of these conditions, from step 3 of the algorithm, we can write the following equation (at $p^k = 0$):

$$Y^k \nabla_y \varphi(y^k) - Y^k \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k = 0$$

As the proposed algorithm creates a sequence of feasible points $\{y^k\}_{k=0,1,...}$ with $y_j^k > 0\ (j = 0,...,n)$ and $k \geq 0$, then it can be concluded that:

$$\nabla_y \varphi(y^k) - \begin{pmatrix} d^T \\ \tilde{A} \end{pmatrix}^T u^k = 0$$

As a result of taking $\lambda_j = 0\ (j = 0,...,n)$, the conditions of optimality of *KKT* in point y^k are satisfied. Sequence $\{y^k\}_{k=0,1,...}$ converges to a solution that satisfies the conditions of optimality of *KKT* of problem. Consequently, this succession creates a sequence of points $\{x^k\}_{k=0,1,...}$ converges to an optimal solution of problem in the decision space X .

4 Conclusions

An algorithm for solving multi-objective linear programming problems has been proposed in this paper. The method uses a single-objective linear variant 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 generated by projecting this approximate gradient onto the null space of the feasible region. It can 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 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 his derivative satisfies the 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 using of the analytic hierarchy process (AHP) technique. Further deeply research in this new area of multi-objective programming is needed and should be concentrated on the ways of developing interactive methods for solving multi-objective nonlinear programming problems.

5 Illustrative Example

The demonstration of the proposed algorithm will be done through the following numerical example (Table 1, Fig.1). Consider the following $MOLP$ problem:

$$
\begin{aligned}
&\max v_1 = c_1^T x = x_1 \\
&\max v_2 = c_2^T x = x_2 \\
&Subject\ \ to \\
&\quad x_1 + 5x_2 \le 41 \\
&\quad 2x_1 + 3x_2 \le 33 \\
&\quad 4x_1 + x_2 \le 41 \\
&\quad x_1 - 2x_2 \le 8 \\
&\quad x_1 + x_2 \ge 2 \\
&\quad -4x_1 + x_2 \le 4 \\
&\quad x_1, x_2 \ge 0
\end{aligned}
$$

Adding the necessary slack and surplus variables, it can be found that:

$c_1 = (10000000)^T$,$c_2 = (01000000)^T$, $b = (41\ 33\ 41\ 8\ 2\ 4)^T$

$$
A = \begin{pmatrix}
1 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & -2 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
-4 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

For this example, an initial point is available through $x^0 = (2\ \ 1\ \ 34\ \ 26\ \ 32\ \ 8\ 1\ 11)^T$. Assuming that the decision maker's utility function is given through $U(v) = (v_1 + 4)(v_2 + 1)$. This vector optimization problem has an optimal solution given through

$x^* = (7\ \ 6.3)^T$ and $v_1^* = 7$, $v_2^* = 6.3$, $U(v_1^*, v_2^*) = 80.6$.

Table 1 Solution results (current iterate).

k	x_1	x_2	U	k	x_1	x_2	U
0	2	1	12	11	6.924	6.138	77.971
1	1.948	1.317	13.784	12	7.517	5.801	78.324
2	2.310	1.505	15.808	13	6.906	6.230	78.853
3	2.6	1.97	19.601	14	7.487	5.873	78.945
4	3.187	2.534	25.4	15	6.940	6.250	79.314
5	3.942	3.364	34.657	16	6.964	6.259	79.591
6	4.968	4.266	47.221	17	6.981	6.265	79.775
7	6.007	5.021	60.249	18	6.992	6.269	79.906
8	6.708	5.500	69.596	19	7.001	6.272	80.003
9	6.957	5.809	74.608	20	7.008	6.275	80.078
10	7.247	5.836	76.885				

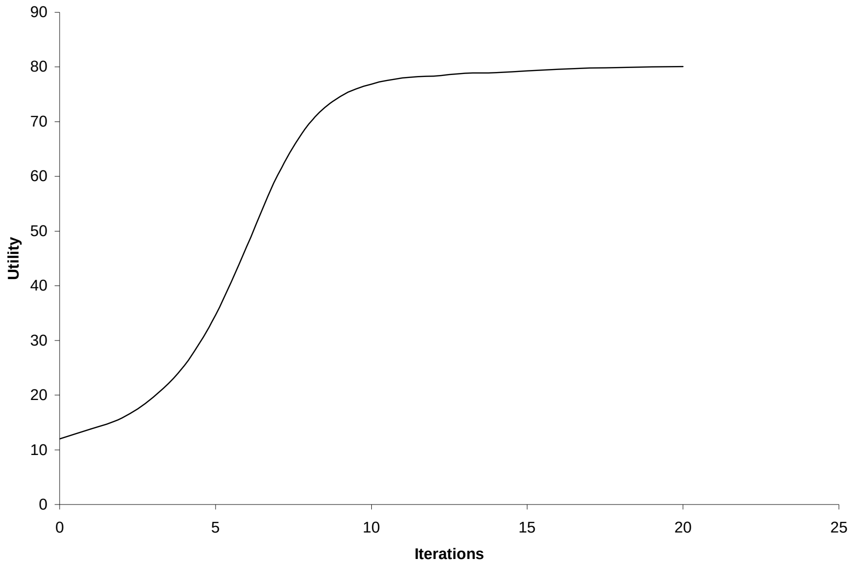

Fig. 1 Utility values at current iterate.

Acknowledgment

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

References

Anstreicher KM. 1986. A monotonic projective algorithm for fractional linear programming. Algorithmica, 1: 483-498

Arbel A. 1994. An interior multi-objective 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 multi-objective linear programming algorithm. European Journal of Operational Research, 89: 202-211

Barnes ER. 1986. A variation on Karmarkar's algorithm for solving linear programming problems. Mathematical Programming, 36: 174-182

De Ghellinck G, Vial JPh. 1986. A polynomial Newton method for linear programming. Algorithmica , 1: 425-453

De Ghellinck G, Vial JPh.1987. An extension of Karmarkar's algorithm for solving a system of linear homogenous equations on the simplex. Mathematical Programming, 39: 79-92

Darvay Z. 2003. New interior point algorithms in linear programming. AMO - Advanced Modeling and Optimization, 5(1): 51-92

Gay D. 1987. A variant of Karmarkar's linear programming algorithm for problems in atandard form. Mathematical Programming, 37: 81-90

Gill PE, Murray W, Saunders MA, Tomlin JA, Wright MH. 1986. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. Mathematical Programming, 36: 183-209

Gonzaga C. 1989. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In: Progress in Mathematical Programming: Interior point and related methods (Megiddo N, editor). Springer Verlag

Karmarkar NK. 1984. A new polynomial time algorithm for linear programming. Combinatorica, 4: 373-395

Naseri R, Malek A. 2004. A new fast algorithm based on Karmarkar's gradient projected method for solving linear programming problem, AMO - Advanced Modeling and Optimization, 6(2): 43-51

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, USA

Todd MJ, Burell BP. 1986. An extension of Karmarkar's algorithm for linear programming using dual variables. Algorithmica, 1: 409-424

Vanderbei RJ, Meketon MS, Freedman BA.1986. A modification of Karmarkar's linear programming algorithm. Algorithmica, 1: 395-407

Wang L, Luo Z. 2015. An improved gradient projection method for constrained optimization problems. AMO - Advanced Modeling and Optimization, 17(2): 335-348

Yu Z, Sun J. 2009. Solving linear programming problem via a reduced dimension method. AMO - Advanced Modeling and Optimization, 11(2): 169-176

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