Abstract

In this work, we studied the ``Path-Following Algorithm``, which is one of the family algorithms, called ``Interior-Point Algorithms``.

We are discussed two modifications, the first one concerned with the path solution, while the second one is concerned with the feasibility solution. These two modifications are combined in a new manner, to construct a hybrid method. The same test problem had been run for all the algorithms, as well as, number of tested problems had been implemented for comparison, shown that our modifications give better results in the number of iterations and the accuracy of the results.

Acknowledgements

Pries to Allah for providing me the willingness and strength to accomplish this work.

I would like to express my deep appreciation to my supervisor, Dr.Alauldin Noori Ahmad, for his patience and appreciable advice. I wish I will give him what he disservice from gratitude.

Its pleasure to record my indebtedness to the staff members of the department of mathematics and computer applications for their great help and efforts in the research work.

I am grateful for the al-Nahrain university, college of science, department of mathematics and computer applications for providing me the chance to further my study.

Finally, I would like to say "Thank you" to my family, for their care, sacrifice and love.

Yasamen
september, 2008 ✍

Beyond any doubt linear programming has been successful from a practical point of view. It is therefore perhaps surprising that various fundamental theoretical questions concerning linear programming have remained open for a very long time. Several of these questions are related to the inherent computational complexity of linear programming problems.

In this chapter, a brief introduction to the linear programming problems, is presented, in which its standard mathematical form and needed basic related techniques are presented.

(1.1) Linear-Programming Problem Formulation[5]:

A linear-programming problem differs from the general variety in that a mathematical model or description of the problem can be state using relationships which are called ``straight-line`` or linear, mathematically, these relationships are of the form:

$$a_1x_1 + a_2x_2 + ... + a_jx_j + ... + a_nx_n = b$$

where the a_j`s and b are known coefficients and the $x_j$`s are unknown variables.

The general linear programming problem is to find a vector which minimize the linear form (i.e., the objective function) $(x_1, x_2, ..., x_n)$

$$c_1x_1 + c_2x_2 + ... + c_nx_n \quad (1.1)$$

subject to the linear constraints

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + ... + a_{1j}x_j + ... + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + ... + a_{2j}x_j + ... + a_{2n}x_n = b_2 \\ &\vdots \\ &a_{m1}x_1 + a_{m2}x_2 + ... + a_{mj}x_j + ... + a_{mn}x_n = b_m \end{aligned} \quad (1.2)$$

and

$x_j \geq 0,$ j=1,2,…,n (1.3)

where the a_{ij} , b_i and c_j are given constants and m<n. The above general linear problem can be written commonly stated in the following matrix form

$$Minimize \quad C^T X \tag{1.4}$$

subject to

$$AX = B \tag{1.5}$$

$$X \geq 0 \tag{1.6}$$

where C and X are vectors in R^n , B is avector in R^m , and A_{mxn} is an matrix. $m \times n$

Next, considering the primal problem (1.4)-(1.6), we can define another, closely related dual problem, as follows:

$$MaximizeB^T y \tag{1.7}$$

subject to

$$A^T Y + S = C \tag{1.8}$$

$$S \geq 0 \tag{1.9}$$

where is a vector of unrestricted values, and $Y = (y_1, y_2, ..., y_n)$ $S = (s_1, s_2, ..., s_n)$ is a vector of slack variables.

Therefore, we can state the following definition:

Definition[13]:

A feasible solution to the linear programming problems is a vector X=(x_1 ,x_2,…, x_n) which satisfies conditions (1.2) and (1.3).

(1.2) Optimality Conditions[11]:

To construct the optimality conditions, we start by combining the primal and dual problem, using the following Lagrangian function [13]:

$$L(X,Y,S) = C^T X - Y^T(AX - B) - S^T X \tag{1.10}$$

and applying the 1st-order necessary conditions [20] for x^* to be a solution of the primal and dual problems such that

$$A^T Y + S = C \tag{1.11a}$$

$$AX = B \tag{1.11b}$$

(1.11c) $X \geq 0$

(1.11d) $S \geq 0$

$x_i s_i = 0$ (1.11e) $i = 1,\ldots, n$

Which are called ``Karush-Kuhn-Tucker`` conditions denoted by KKT conditions.

The complementarity condition (1.11e), which essentially says that $.i = 1,\ldots,n$ must be zero for each s_i and x_i at least one of the components denoted a vector triple that satisfies (1.11). By (x^*, y^*, s^*) Let combining the three equalities (1.11a),(1.11d), and (1.11e), we find that

$$c^T x^* = (A^T y^* + s^*)^T x^* = (Ax^*)^T y^* = B^T y^* \tag{1.12}$$

is also the objective function for the $B^T y$ As we shall see in a moment, problem formulated by eqs. (1.4)-(1.6), so (1.12) indicates that the primal which satisfy the dual objectives are equal for vector triples (x,y,s) that satisfy (1.11).

It is easy to show directly that the conditions (1.11) are sufficient for x* be any other feasible $\bar{x}$ to be a global solution of (1.4)-(1.6), by letting then $\bar{x} \geq 0$ and $A\bar{x} = B$point, so that

$$c^T \bar{x} = (Ay^* + s^*)^T \bar{x} = B^T y^* + \bar{x}^T s^* \geq B^T y^* = c^T x^* \qquad (1.13)$$

We have used (1.11) and (1.12) here; the inequality relation follows. The inequality (1.13) tells us that no other feasible $s^* \geq 0$ and $\bar{x} \geq 0$ trivially from . We can say more: $c^T x^*$ point can have a lower objective value than is optimal if and only $\bar{x}$ The feasible point

$$\bar{x}^T s^* = 0$$

since otherwise the inequality in (1.13) is strict. In other words, when of (1.4)-(1.6). $\bar{x}$ for all solutions $\bar{x}_i = 0$,then we must have $s_i^* > 0$

(1.3) Methods For Solving Linear Programming Problems:

Many methods are available in which the simplest way is to follow a direct mathematical search through calculating the objective function value at each vertex [2].

An algorithm for a certain problem is said to be `good` if it solves an instance of that problem in a number of computations that is bounded from above by a polynomial function of the size of the problem instance (Edmonds [1965]). Informally we say that an algorithm is `good` if it runs in polynomial time.

G.B. Dantzing and others in 1947 (see[21]), developed an algorithm for solving the linear programming problem, by constracting a path along the edges of the

polytype defined by (1.2) and (1.3). In each iteration, the method examines if further improvement in the objective function is attainable by moving from the current extreme points to one of its neighbours; if so, one of these neighbours is selected by means of a pivot selection rule and another step is executed. On the other hand, in practice the number of iterations in the simplex method usually ranges from m to 3m, (see Wolfe and Cutler [1963], Quandt and Kuhn [1964]).

Karmarkar [1] presented a new polynomial time algorithm for linear programming problems. Also, several so called interior point algorithms for linear and convex quadratic programming have been proposed, (see [12] and [21]), which become very competitive to the simplex method. They fall into three main groups:

(a) Projective algorithms.

(b) Affine scaling algorithms.

(c) Path following algorithms.

The above three categories are similar in concept but differ in problem transformation and internal search direction, (see[21]).

This work, consist of three chapters, as well as, this chapter. The second chapter, discussed in details, the class of path following method. Chapter three, discussed two modifications on the path following method. Chapter four build a hybrid method, based on the modifications presented in chapter three. The programming steps for all algorithms are presented, and the same tested problem, is utilized for all methods for the saek of the accuracy and comparision of the results. As well as number of tested problems had been utilized for comparison.

By the early 1990s, the class of path following algorithms had distinguished itself as the most efficient practical approach and proved to be a strong competitor to the simplex method on large problems [14]. This method is the focus of this chapter.

The motivation for interior-point method arose from the desire to find algorithms with better theoretical properties than the simplex method. As it has been mentioned, (see [7]), the simplex method can be quite inefficient on certain problems (see [15]). Roughly speaking, the time required to solve a linear program may be exponential in the size of the problem, as measured by the number of unknowns and the amount of storage needed for the problem data. In practice, the poor worst case complexity motivated the development of new algorithms with better guaranteed performance.

Interior-point method shere common features that distinguish them from the simplex method (see [14,20]). Each interior-point iteration is expensive to compute and can make significant progress towards the solution, while the simplex method usually requires a larger number of inexpensive iterations (see [10,21]). The simplex method works its way around the boundary of the feasible polytope, testing a sequence of vertices in turn until it finds the optimal one. Interior-point method approach the boundary of the feasible set only in the limit. They may approach the solution either from the interior or the exterior of the feasible region, but they never actually lie on the boundary of this region.

In this chapter we outline some of the basic ideas behind the path following algorithms, called Primal-Dual Interior-Point method (see[19,20], including its relationship to Newton`s method.

Therefore, before we are presented the details of the Primal-Dual Method, we will give a short explination to the Newton`s method for nonlinear algebric equations as following, (see[9]):

If the mathematical model involves two (or more) simultaneous nonlinear algebraic equations in two (or more) unknowns, the Newton-Raphson method can be extended to solve, these equations simultaneously, by considering the following model for two unknowns variables [6,16]

$$f_1(x_1, x_2) = 0$$
$$f_2(x_1, x_2) = 0$$

are nonlinear function of variables x_1 and x_2. f_2 and f_1 where
Both functions may be expanded in to Taylor series expansion around an initial estimate of x_1^1 and x_2^1:

$$\left.\begin{aligned} f_1(x_1,x_2) &= f_1(x_1^{\,1},x_2^{\,1}) + \left.\frac{\partial f_1}{\partial x_1}\right|_{x^1}(x_1 - x_1^{\,1}) + \left.\frac{\partial f_1}{\partial x_2}\right|_{x^1}(x_2 - x_2^{\,1}) + \ldots \\ f_2(x_1,x_2) &= f_2(x_1^{\,1},x_2^{\,1}) + \left.\frac{\partial f_2}{\partial x_1}\right|_{x^1}(x_1 - x_1^{\,1}) + \left.\frac{\partial f_2}{\partial x_2}\right|_{x^1}(x_2 - x_2^{\,1}) + .. \end{aligned}\right\} \tag{2.1}$$

Setting the left hand sides of eqs.(2.1) to zero and truncating the second-order and higher derivatives of the Taylor series , we obtain the following equations:

$$\left.\begin{aligned} \left.\frac{\partial f_1}{\partial x_1}\right|_{x^1}(x_1 - x_1^{\,1}) + \left.\frac{\partial f_1}{\partial x_2}\right|_{x^1}(x_2 - x_2^{\,1}) &= -f_1(x_1^{\,1},x_2^{\,1}) \\ \left.\frac{\partial f_2}{\partial x_1}\right|_{x^1}(x_1 - x_1^{\,1}) + \left.\frac{\partial f_2}{\partial x_2}\right|_{x^1}(x_2 - x_2^{\,1}) &= -f_2(x_1^{\,1},x_2^{\,1}) \end{aligned}\right\} \tag{2.2}$$

as Δx If we define the correction variables

$$\Delta x_1 = x_1 - x_1^{\,1}$$
$$\Delta x_2 = x_2 - x_2^{\,1}$$

then eqs.(2.2) can be rewritten as:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = - \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_{2)} \end{bmatrix}$$

Now if we consider the system:

$$\begin{aligned} & f_1(x_1,\ldots,\ x_k) = 0 \\ & \vdots \\ & f_k(x_1,\ldots,\ x_k) = 0 \end{aligned}$$

The linearization of this system obtained by applying the Taylor series expansion, we have eqs.(2.3).

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_k} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_k} \\ \vdots & & & \\ \frac{\partial f_k}{\partial x_1} & \frac{\partial f_k}{\partial x_2} & \cdots & \frac{\partial f_k}{\partial x_k} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_k \end{bmatrix} = - \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix} \tag{2.3}$$

In matrix -vector notation this condenses to

$$J\Delta x = -f$$

is the Jacobian matrix containing the partial derivatives, is the J where is the vector of functions. f correction vector, and

Primal-dual problem is to find solution (x^*, y^*, s^*) of this system by applying variants of Newton’s method (which is described later in this chapter), and modifying search directions and step lengths so that the inequalities (x , s) ≥ 0 are satisfied strictly at every iterations. The equations (1.11a), (1.11b), (1.11c) are only mildly nonlinear and so are not difficult to solve by themselves. However, the problem becomes much more difficult when we add the nonnegativity requirement (1.11d), which is the source of all the complications in the design and analysis of interior-point method [4].

To derive Primal-Dual Interior-Point method see[13], we restate the optimality conditions (1.11) in a slightly different form by means of mapping F from R^{2n+m} to R^{2n+m}

$$F(x,y,s)=\begin{bmatrix} A^T y+s-c \\ Ax-b \\ XSe \end{bmatrix}=0 \qquad (2.4a)$$

$$(x,s)\geq 0 \qquad (2.4b)$$

Where

$$\left.\begin{array}{l} X = diag\ (x_1, x_2,..., x_n) \\ S = diag\ (s_1, s_2,..., s_n) \end{array}\right\} \qquad (2.5)$$

and

$$e=(1,...,1)^T$$

Primal-Dual method generate an iterates (x^k, y^k, s^k) that satisfy the bounds (2.4b) strictly, that is, $x^k > 0$ and $s^k > 0$. This property is the origin of the term interior-point. By respecting these bounds , the methods avoid spurious solution, that is, points that satisfy $F(x,y,s)=0$ but $(x,s)\geq 0$.

The Primal-Dual problem of (1.4)-(1.6) and (1.11) and (2.5) are characterized by KKT conditions which we are restarted here as equations (1.11).

Interior-Point method actually require the iterates to be strictly feasible, that is, each (x^k, y^k, s^k) must satisfy the linear equality constraints for the primal and dual problems (see [20,22]). If we define the primal-dual feasible set F and strictly feasible set F^o by:

$$F = \{(x, y, s) / Ax = b, A^T y + s = c, (x, s) \geq 0\} \qquad (2.6a)$$

$$F^0 = \{(x, y, s) / Ax = b, A^T y + s = c, (x, s) > 0\} \qquad (2.6b)$$

The strict feasibility condition can be written concisely as

$$(x^k, y^k, s^k) \in F^0$$

Like most iterative algorithms in optimization, primal-dual interior -point method have two basic ingredients: a procedure for determining the step direction and its length at each point in the search space. The search direction procedure has its origins in Newton's method for the nonlinear equations (2.4a). Newton's method forms a linear model for F around the current point and obtains the search direction (Δx , Δy, Δs) by solving the following system of linear equations[3]:

$$J(x, y, s) \times \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = -F(x, y, s)$$

where J is the jacobian of F, which is equivalent to the system (2.3). If the current point is strictly feasible (that is $(x, y, s) \in F^0$), the Newton step equation become

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix} \qquad (2.7)$$

A full step along direction usually is not permissible, since it would violate the bound (x , s) ≥ 0. To avoid this difficulty, we perform a line search along the Newton direction so that the new iterate is

$$(x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$$

for some line search parameter $\alpha \in (0,1]$. Unfortunatly we often can take only a small step along the direction ($\alpha << 1$) before violating the condition $(x, s) > 0$. Hence, the pure Newton direction (2.7), which is known as the affine scaling direction, often does not allow us to make much progress toward a solution (see [13]). Primal-dual methods modify the basic Newton procedure in the following two ways:

1. They bias the search direction toward the interior of the nonnegative $(x, s) \geq 0$, so that we can move further along the direction before one of the components of (x, s) becomes negative.

2. They keep the components of (x, s) from moving " too close " to the boundary of the nonnegative.

Algorithm:

Given $(x^0, y^0, s^0) \in F^0$

For $k = 0,1,2,...$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e \end{bmatrix}$$

Set

$$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (\Delta x^k, \Delta y^k, \Delta s^k)$$

choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$

Now, we are present an example taken from [15], and considered it in all modifications presented in chapters three and four for the results verifying.

Example:

Minimize

$$-6x_1 - 3x_2 + 4x_3$$

subject to

$$x_1 + 4x_2 - 2x_3 \le 1$$
$$2x_1 - 2x_2 + 6x_3 \le 2$$
$$-2x_1 + 3x_2 + x_3 \le 5$$

we have

$$A = \begin{bmatrix} 1 & 4 & -2 \\ 2 & -2 & 6 \\ -2 & 3 & 1 \end{bmatrix}$$

Iteration1

and suppose that

$$X^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad S^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Therefor, from eq.(2.7)

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix}$$

where

$$-X^0 S^0 e = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Thus, we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.053 \\ 0.053 \\ 0.053 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

choose $\alpha_0 = 0.01$

$$(x^1, y^1, s^1) = (x^0, y^0, s^0) + \alpha_0(\Delta x^0, \Delta y^0, \Delta s^0)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0005 \\ 0.0005 \\ 0.0005 \\ 0.9900 \\ 0.9900 \\ 0.9900 \end{bmatrix}$$

Iteration2

Since we get $(X, Y, S) > 0$ in the last iteration, then $\alpha_1 = \alpha_0$

$$X^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} , \qquad S^1 = \begin{bmatrix} 0.9900 & 0 & 0 \\ 0 & 0.9900 & 0 \\ 0 & 0 & 0.9900 \end{bmatrix}$$

where

$$-X^{1}S^{1}e=\begin{bmatrix}-0.9900\\-0.9900\\-0.9900\end{bmatrix}$$

Thus, we get

$$\begin{bmatrix}0&0&0&1&2&-2&1&0&0\\0&0&0&4&-2&3&0&1&0\\0&0&0&-2&6&1&0&0&1\\1&4&-2&0&0&0&0&0&0\\2&-2&6&0&0&0&0&0&0\\-2&3&1&0&0&0&0&0&0\\0.9900&0&0&0&0&0&1&0&0\\0&0.9900&0&0&0&0&0&1&0\\0&0&0.9900&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}\Delta x_1\\\Delta x_2\\\Delta x_3\\\Delta y_1\\\Delta y_2\\\Delta y_3\\\Delta s_1\\\Delta s_2\\\Delta s_3\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0\\0\\0\\-0.9900\\-0.9900\\-0.9900\end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix}\Delta x_1\\\Delta x_2\\\Delta x_3\\\Delta y_1\\\Delta y_2\\\Delta y_3\\\Delta s_1\\\Delta s_2\\\Delta s_3\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0.0390\\0.0290\\0.0010\\-0.9900\\-0.9900\\-0.9900\end{bmatrix}$$

$$(x^{2},y^{2},s^{2})=(x^{1},y^{1},s^{1})+\alpha_0(\Delta x^{1},\Delta y^{1},\Delta s^{1})$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0005 \\ 0.0005 \\ 0.0005 \\ 0.9890 \\ 0.9890 \\ 0.9890 \end{bmatrix}$$

The same procedure will be repeated for 15 different values of α, satisfying condition (2.6b) until no significant difference values in X and S , will be appeared. And the final results is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0025 \\ 0.0022 \\ 0.0007 \\ 0.9799 \\ 0.9799 \\ 0.9799 \end{bmatrix}$$

The comparsion results for the interior-point and all its modifications are presented in a table to show the efficiency of the hybrid method, based on the number of iterations.

The path modification, shifting the path ρ, by a parameter $\tau \in [0,1)$, with $(\hat{x}(0), \hat{y}(0), \hat{s}(0)) = (x, y, s)$ and, if the limit exists as $\tau \to 1$ we have

$\lim_{\tau \to 1} (\hat{x}(\tau), \hat{y}(\tau), \hat{s}(\tau)) \in \Omega$, where Ω is the set of the solution points.

Then after we perform this shifting, we are choosing the parameter σ adaptively, which assign a value to σ prior to calculating the search direction. At each iteration, the method first calculates the scaling direction (with $\sigma = 0$) and assesses its usefulness as a search direction. If this direction yields a large reduction in μ without violating the positivity condition $(x,s)>0$, the processing concludes that it need to choose σ close to zero and calculates a search direction with this small value. If the scaling direction is not so productive, the algorithm enforces choosing a value of σ closer to 1.

The computation of the search direction $(\Delta x, \Delta y, \Delta s)$ proceeds as follows.

First, we calculate the predictor step $(\Delta x^p, \Delta y^p, \Delta s^p)$ (where $"P"$ denote the "scalling direction") by setting $\sigma = 0$ in (3.9), that is

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta y^p \\ \Delta s^p \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix} \tag{4.1}$$

To measure the effectiveness of this direction, we find α_p^{pri} and α_p^{dual} (where ``pri`` and ``dual`` denote the ``primal`` and ``dual`` problem) to be the longest step lengths that can be taken along this direction before violating the nonnegativity conditions $(x,s)\geq 0$, with an upper bound of 1. where [20]

$$\alpha_p^{pri} = \min(1, \min_{i:\Delta x_i<0} \frac{-x_i}{\Delta x_i^{\ p}}) \tag{4.2.a}$$

$$\alpha_p^{dual} = \min(1, \min_{i:\Delta s_i<0} \frac{-s_i}{\Delta s_i^{\ p}}) \tag{4.2.b}$$

we define μ_p to be the value of μ that would be obtained by a full step (i.e $\alpha_k = 1$) to the boundary that is [20]

$$\mu_p = \frac{(x + \alpha_p^{pri}\,\Delta x^p)^T (s + \alpha_p^{dual}\,\Delta s^p)}{n} \tag{4.3}$$

and set the parameter σ to be

$$\sigma = \left(\frac{\mu_p}{\mu}\right)^j$$, where j is any suitable integer.

Second, the corrector step is obtained by replacing the right-hand-side of eqs.(4.1) by $(0,0,-\Delta x^p \Delta s^p e)$, while the parameter σ_k requires a right-hand-side of $(0,0,\sigma\mu\ e)$.

Now, we can built a hybrid algorithm step which includes the path and step length modifications, by changing the right-hand-sides for these three components and solving the following modified system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe - \Delta x^p \Delta s^p e + \sigma\mu e \end{bmatrix} \tag{4.4}$$

We calculate the maximum steps lengths that can be taken along these directions before violating the nonnegativity condition $(x, s) > 0$ by a formula similar to (4.2); namely,

$$\alpha_{\max}^{pri} = \min(\,1, \min_{i:\Delta x_i < 0} \frac{x_i^{\;k}}{\Delta x_i}) \tag{4.5.a}$$

$$\alpha_{\max}^{dual} = \min(\;1, \min_{i:\Delta s_i < 0} \frac{s_i^{\;k}}{\Delta s_i}) \tag{4.5.b}$$

and then choose the primal and dual step lengths as follows

$$\left.\begin{aligned} \alpha_k^{pri} &= \min(\,1, \eta\,\alpha_{\max}^{pri}) \\ \alpha_k^{dual} &= \min(\,1, \eta\,\alpha_{\max}^{dual}) \end{aligned}\right\} \tag{4.6}$$

Where $\eta \in [0.9, 1.0)$ (see [20]) is chosen so that $\eta \to 1$ near the solution, to accelerate the convergence. We summarize this discussion by specifying the following algorithm in the usual format.

The Hybrid Algorithm:

(x^0, y^0, s^0) Given with $(x^0, s^0) > 0$

for $k = 0,1,2,...$

Set $(x, y, s) = (x^k, y^k, s^k)$

and solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta y^p \\ \Delta s^p \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}$$

Calculate

$$\alpha_p^{pri} = \min(1, \min_{i:\Delta x_i < 0} \frac{-x_i}{\Delta x_i^p})$$

$$\alpha_p^{dual} = \min(1, \min_{i:\Delta s_i < 0} \frac{-s_i}{\Delta s_i^p})$$

and

$$\mu_p = (x + \alpha_p^{pri} \Delta x^p)^T (s + \alpha_p^{dual} \Delta s^p) / n$$

Set centering parameter to solve $\sigma = ({}^{\mu_p}\!/\!{}_{\mu})^j$

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^0 & 0 & X^0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -X^0 S^0 e - \Delta x^p \Delta s^p e + \sigma \mu e \end{bmatrix}$$

Calculate

$$\alpha_{\max}^{pri} = \min(1, \min_{i:\Delta x_i < 0} \frac{x_i^k}{\Delta x_i})$$

$$\alpha_{\max}^{dual} = \min(\ 1, \min_{i:\Delta s_i<0} \frac{s_i^k}{\Delta s_i})$$

$$\eta \to 1$$

$$\alpha_k^{pri} = \min(1, \eta\, \alpha_{\max}^{pri})$$

$$\alpha_k^{dual} = \min(1, \eta\, \alpha_{\max}^{dual})$$

set

$$x^{k+1} = x^k + \alpha_k^{pri} \Delta x$$

$$(y^{k+1}, s^{k+1}) = (y^k, s^k) + \alpha_k^{dual} (\Delta y, \Delta s)$$

end(for).

The above algorithm, can be demonstrated by solving the same preceding example:

The solution of the example:

Iteration1

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta y^p \\ \Delta s^p \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}$$

where

$$r_b = \begin{bmatrix} 2 \\ 4 \\ -3 \end{bmatrix}$$

$$r_c = \begin{bmatrix} 8 \\ 8 \\ -5 \end{bmatrix}$$

$$-X^0S^0e=\begin{bmatrix}-1\\-1\\-1\end{bmatrix}$$

Therefor, from eq.(4.1) we get

$$\begin{bmatrix}0&0&0&1&2&-2&1&0&0\\0&0&0&4&-2&3&0&1&0\\0&0&0&-2&6&1&0&0&1\\1&4&-2&0&0&0&0&0&0\\2&-2&6&0&0&0&0&0&0\\-2&3&1&0&0&0&0&0&0\\1&0&0&0&0&0&1&0&0\\0&1&0&0&0&0&0&1&0\\0&0&1&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}\Delta x_1{}^p\\\Delta x_2{}^p\\\Delta x_3{}^p\\\Delta y_1{}^p\\\Delta y_2{}^p\\\Delta y_3{}^p\\\Delta s_1{}^p\\\Delta s_2{}^p\\\Delta s_3{}^p\end{bmatrix}=\begin{bmatrix}-8\\-8\\5\\-2\\-4\\3\\-1\\-1\\-1\end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix}\Delta x_1{}^p\\\Delta x_2{}^p\\\Delta x_3{}^p\\\Delta y_1{}^p\\\Delta y_2{}^p\\\Delta y_3{}^p\\\Delta s_1{}^p\\\Delta s_2{}^p\\\Delta s_3{}^p\end{bmatrix}=\begin{bmatrix}-0.0651\\-0.0104\\-0.0556\\-0.0659\\-0.0558\\0.0035\\-0.9985\\-0.9998\\-0.9996\end{bmatrix}$$

Calculate α_p^{pri} , α_p^{dual} and μ_p

$$\alpha_p^{pri}=\min(1,\min_{i:\Delta x_i<0}\frac{-x_i}{\Delta x_i{}^p})$$

$$\alpha_p^{pri}=1$$

$$\alpha_p^{dual} = \min(1, \min_{i:\Delta s_i<0} \frac{-s_i}{\Delta s_i^{\ p}})$$

$$\alpha_p^{dual} = 1$$

$$\mu_p = \frac{(x+\alpha_p^{pri}\,\Delta x^p)^T (s+\alpha_p^{dual}\,\Delta s^p)}{n}$$

$$\mu_p = 0.0030$$

$\sigma = (\dfrac{\mu_p}{\mu})^3$ set centering parameter

$$\sigma = 0.0909$$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^0 & 0 & X^0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -X^0S^0e-\Delta x^p\Delta s^p e+\sigma\mu e \end{bmatrix}$$

where

$$-X^0S^0e-\Delta x^p\Delta s^p e+\sigma\mu e = \begin{bmatrix} -0.9962 \\ -0.9987 \\ -0.9899 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -8 \\ -8 \\ 5 \\ -2 \\ -4 \\ 3 \\ -0.9962 \\ -0.9987 \\ -0.9899 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\\ \Delta y_1\\ \Delta y_2\\ \Delta y_3\\ \Delta s_1\\ \Delta s_2\\ \Delta s_3\end{bmatrix} = \begin{bmatrix}-0.0054\\ -0.0156\\ -0.0580\\ 0.0067\\ 0.0005\\ 0.0003\\ -0.9983\\ -0.9989\\ -0.9988\end{bmatrix}$$

Calculate

$$\alpha_{\max}^{pri} = \min(1, \min_{i:\Delta x_i<0} \frac{-x_i}{\Delta x_i})$$

$$\alpha_{\max}^{pri} = 1$$

chosen

$$\eta = 0.95$$

$$\alpha_k^{pri} = \min(1, \eta\, \alpha_{\max}^{pri})$$

$$\alpha_k^{pri} = 0.95$$

$$\alpha_{\max}^{dual} = \min(1, \min_{i:\Delta s_i<0} \frac{-s_i}{\Delta s_i})$$

$$\alpha_{\max}^{dual} = 1$$

$$\alpha_k^{dual} = \min(1, \eta\, \alpha_{\max}^{dual})$$

$$\alpha_k^{dual} = 0.95$$

set

$$x^{k+1} = x^k + \alpha_k^{pri} \Delta x$$

$$x_1 = 0.9948$$

$$x_2 = 0.9899$$

$$x_3 = 0.9649$$

$$(y^{k+1}, s^{k+1}) = (y^k, s^k) + \alpha_k^{dual} (\Delta y, \Delta s)$$

$$(y_1, s_1) = (1.0063, 0.9986)$$

$$(y_2, s_2) = (0.0006, 0.9978)$$

$$(y_3, s_3) = (0.0007, 0.9988)$$

Iteration2

Now we obtain new X and S

$$X^1 = \begin{bmatrix} 0.9948 & 0 & 0 \\ 0 & 0.9899 & 0 \\ 0 & 0 & 0.9649 \end{bmatrix}$$

$$S^1 = \begin{bmatrix} 0.9986 & 0 & 0 \\ 0 & 0.9978 & 0 \\ 0 & 0 & 0.9988 \end{bmatrix}$$

where

$$r_b = \begin{bmatrix} -0.0004 \\ -0.0005 \\ -0.6789 \end{bmatrix}$$

$$r_c = \begin{bmatrix} 7.8602 \\ 6.9623 \\ -5.9635 \end{bmatrix}$$

$$-X^1 S^1 e = \begin{bmatrix} -0.9934 \\ -0.9877 \\ -0.9796 \end{bmatrix}$$

Therefor, from eq.(4.1) we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9986 & 0 & 0 & 0 & 0 & 0 & 0.9948 & 0 & 0 \\ 0 & 0.9978 & 0 & 0 & 0 & 0 & 0 & 0.9899 & 0 \\ 0 & 0 & 0.9988 & 0 & 0 & 0 & 0 & 0 & 0.9649 \end{bmatrix} \begin{bmatrix} \Delta x_1{}^p \\ \Delta x_2{}^p \\ \Delta x_3{}^p \\ \Delta y_1{}^p \\ \Delta y_2{}^p \\ \Delta y_3{}^p \\ \Delta s_1{}^p \\ \Delta s_2{}^p \\ \Delta s_3{}^p \end{bmatrix} = \begin{bmatrix} -7.8602 \\ -6.9623 \\ 5.9635 \\ 0.0004 \\ 0.0005 \\ 0.6789 \\ -0.9934 \\ -0.9877 \\ -0.9796 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1{}^p \\ \Delta x_2{}^p \\ \Delta x_3{}^p \\ \Delta y_1{}^p \\ \Delta y_2{}^p \\ \Delta y_3{}^p \\ \Delta s_1{}^p \\ \Delta s_2{}^p \\ \Delta s_3{}^p \end{bmatrix} = \begin{bmatrix} -0.0646 \\ -0.0101 \\ -0.0545 \\ -0.0329 \\ -0.0457 \\ 0.0032 \\ -0.9983 \\ -0.9994 \\ -0.9993 \end{bmatrix}$$

Calculate α_p^{pri} , α_p^{dual} and μ_p

$$\alpha_p^{pri} = \min(1, \min_{i:\Delta x_i<0} \frac{-x_i}{\Delta x_i{}^p})$$

$$\alpha_p^{pri} = 1$$

$$\alpha_p^{dual} = \min(1, \min_{i:\Delta s_i<0} \frac{-s_i}{\Delta s_i^{\,p}})$$

$$\alpha_p^{dual} = 1$$

$$\mu_p = (x + \alpha_p^{pri} \Delta x^p)^T (s + \alpha_p^{dual} \Delta s^p) \Big/ n$$

$$\mu_p = 0.0026$$

$$\mu = 0.3311$$

$\sigma = (\mu_p / \mu)^3$ set centering parameter

$$\sigma = 0.0000$$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^0 & 0 & X^0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -x^0 s^0 e - \Delta x^p \Delta s^p e + \sigma \mu e \end{bmatrix}$$

where

$$-X^0 S^0 e - \Delta x^p \Delta s^p e + \sigma \mu e = \begin{bmatrix} -0.9961 \\ -0.9985 \\ -0.9896 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9986 & 0 & 0 & 0 & 0 & 0 & 0.9948 & 0 & 0 \\ 0 & 0.9978 & 0 & 0 & 0 & 0 & 0 & 0.9899 & 0 \\ 0 & 0 & 0.9988 & 0 & 0 & 0 & 0 & 0 & 0.9649 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -7.8602 \\ -6.9623 \\ 5.9635 \\ 0.0004 \\ 0.0005 \\ 0.6789 \\ -0.9961 \\ -0.9985 \\ -0.9896 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -0.0052 \\ -0.0153 \\ -0.0579 \\ 0.0060 \\ 0.0004 \\ 0.0002 \\ -0.9981 \\ -0.9987 \\ -0.9986 \end{bmatrix}$$

Calculate

$$\alpha_{\max}^{pri} = \min(1, \min_{i:\Delta x_i < 0} \frac{-x_i}{\Delta x_i})$$

$$\alpha_{\max}^{pri} = 1$$

$$\alpha_k^{pri} = \min(1, \eta\, \alpha_k^{pri})$$

$$\alpha_k^{pri} = 0.95$$

$$\alpha_{\max}^{dual} = \min(1, \min_{i:\Delta s_i < 0} \frac{-s_i}{\Delta s_i})$$

$$\alpha_{\max}^{dual} = 1$$

$$\alpha_k^{dual} = \min(1, \eta\,\alpha_{\max}^{dual})$$

$$\alpha_k^{dual} = 0.95$$

set

$$x^{k+1} = x^k + \alpha_k^{pri}\,\Delta x$$

$$x_1 = 0.9945$$

$$x_2 = 0.9896$$

$$x_3 = 0.9646$$

$$(y^{k+1}, s^{k+1}) = (y^k, s^k) + \alpha_k^{dual}(\Delta y, \Delta s)$$

$$(y_1, s_1) = (1.0060, 0.9983)$$

$$(y_2, s_2) = (0.0010, 0.9975)$$

$$(y_3, s_3) = (0.0006, 0.9986)$$

The above procedure, will be repeated for 4 more different values of α , satisfying the condition (2.6b), having the final results:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0023 \\ 0.0019 \\ 0.0007 \\ 0.9795 \\ 0.9793 \\ 0.9797 \end{bmatrix}$$

Computation Results:

Twenty problems have been tested for a comparison study. The number of iterations have been considered as a measurement of the efficiency to all methods, as shown in the following table(1).

Problem number	Size n×m	Number of iteration			
		Path Following	Path Modification	Step Length Modification	Hybrid Modification
١	٥×٤	١٦	١١	٩	٦
٢	٨×٧	١٧	١٢	١٠	٦
٣	3×4	١٥	١١	٨	٥
٤	٤×٢	١٤	١٠	٧	٤
٥	٧×٥	٢٠	١٥	٩	٦
٦	٨×٥	٢١	١٦	١٠	٦
٧	٩×٩	٢٤	١٨	١١	٦
٨	٩×٨	٢٣	١٧	١٠	٥
٩	١٠×٩	٢٥	٢٠	١٤	٧
١٠	٨×٢	١٨	١٣	٩	٥
١١	٧×٣	١٩	١٥	١٠	٥
١٢	٨×٤	٢٠	١٦	١١	٦
١٣	٩×٧	٢٢	١٧	١٢	٧
١٤	٨×٨	٢٣	١٨	١٣	٧
١٥	٧×٧	٢٢	١٨	١٢	٦
١٦	٦×٦	١٧	١٣	١٠	٥
١٧	٥×٥	١٦	١١	٨	٤
١٨	٤×٤	١٥	١١	٧	٥
١٩	٦×٤	١٩	١٤	١٠	٥
٢٠	٩×٦	٢١	١٧	١١	٧

Table(1)

Comparison of the Results for the Interior-Point method and their modifications

(3.1) The Path Modification:

Since the path in primal-dual algorithms plays an effective rule in solving the problem [15], we suggested a modification on the path ρ based on the nonnegative condition (2.4), in which we parametrized it by a scalar $\tau > 0$ and each point $(x_\tau, y_\tau, s_\tau) \in \rho$ solve the following system:

$$A^T y + s = c \tag{3.1}$$

$$Ax = b \tag{3.2}$$

$$x_i s_i = \tau \tag{3.3}$$

$$(x, s) > 0 \tag{3.4}$$

The above conditions differ from KKT condition (1.11) only in the term τ on the right-hand-side of (3.3). Instead of the complementarity condition (1.11c), we require that the pairwise products x_i s_i have the same value for all indices i. Therefore, another way of defining ρ is to use the mapping F defined in (2.4)-(2.5) as:

$$F(x_t, y_t, s_t) = \begin{bmatrix} 0 \\ 0 \\ \tau_e \end{bmatrix} \quad , \quad (x_\tau, s_\tau) > 0 \tag{3.5}$$

Equations (3.1)-(3.4) approximate (1.11) more and more closely as τ goes to zero. If ρ converges to anything as $\tau \to 0$, it must converge to a primal-dual solution of the linear program[15]. This path thus guides us to a solution along a route that solutions keeping all x and s components strictly positive and decreasing the pairwise products $x_i s_i$, $(i = 1,2,...,n)$ to zero.

To describe this, we introduce a parameter $\sigma \in [0,1]$ and a measurement μ defined by [19]

$$\mu = \frac{1}{n\sum_{i=1}^{n} x_i s_i} = \frac{x^T s}{n} \qquad (3.6)$$

which measure the average value of the pairwise products $x_i s_i$, by writing $\tau = \sigma\mu$ and applying Newton`s method to the system (3.5) (which is a modification to (2.7)), we obtain

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{bmatrix} \qquad (3.7)$$

and, we call this modification as ``path modification `` .

To demonstrate this modification, we solve the same example presented in chapter 2, by the following modified algorithm:

Given $(x^0, y^0, s^0) \in F^0$

For $k = 0,1,2,...$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma\mu e \end{bmatrix}$$

where

$\sigma \in [0,1]$

and

$\mu_k = (x^k)^T s^k / n$

Set

$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (\Delta x^k, \Delta y^k, \Delta s^k)$

choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$

end(for).

The solution of the example:

Iteration 1

Let

$$X^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad S^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad y^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

where

$$\sigma\mu e = \begin{bmatrix} 0.0033 \\ 0.0033 \\ 0.0033 \end{bmatrix}$$

$$-X^0 S^0 e + \sigma\mu e = \begin{bmatrix} -0.9966 \\ -0.9966 \\ -0.9966 \end{bmatrix}$$

Therefor, from eq.(3.7) we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.9966 \\ -0.9966 \\ -0.9966 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.0399 \\ 0.0299 \\ 0.0001 \\ -0.9970 \\ -0.9970 \\ -0.9970 \end{bmatrix}$$

choose $\alpha_0 = 0.01$

$$(x^1, y^1, s^1) = (x^0, y^0, s^0) + \alpha_0 (\Delta x^0, \Delta y^0, \Delta s^0)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0003 \\ 0.0002 \\ 0.0002 \\ 0.9900 \\ 0.9900 \\ 0.9900 \end{bmatrix}$$

Iteration 2

Now we obtain new X and S

$$X^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S^1 = \begin{bmatrix} 0.9900 & 0 & 0 \\ 0 & 0.9900 & 0 \\ 0 & 0 & 0.9900 \end{bmatrix}$$

where

$$\sigma\mu e = \begin{bmatrix} 0.0032 \\ 0.0032 \\ 0.0032 \end{bmatrix}$$

$$-X^1 S^1 e + \sigma\mu e = \begin{bmatrix} -0.9867 \\ -0.9867 \\ -0.9867 \end{bmatrix}$$

Therefor, from eq.(3.7) we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9900 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0.9900 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.9900 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.9867 \\ -0.9867 \\ -0.9867 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\\ \Delta y_1\\ \Delta y_2\\ \Delta y_3\\ \Delta s_1\\ \Delta s_2\\ \Delta s_3\end{bmatrix}=\begin{bmatrix}0\\ 0\\ 0\\ 0.0395\\ 0.02960\\ 0.0001\\ -0.9870\\ -0.9870\\ -0.9870\end{bmatrix}$$

Since we get $(X,Y,S)>0$ in the last iteration, then $\alpha_1=\alpha_0$

$$(x^2,y^2,s^2)=(x^1,y^1,s^1)+\alpha_0(\Delta x^1,\Delta y^1,\Delta s^1)$$

$$\begin{bmatrix}x_1\\ x_2\\ x_3\\ y_1\\ y_2\\ y_3\\ s_1\\ s_2\\ s_3\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 1\\ 1.0006\\ 0.0004\\ 0.0002\\ 0.9801\\ 0.9801\\ 0.9801\end{bmatrix}$$

The above procedure, will be repeated for 10 more different values of α , satisfying the condition (2.6b), having the final results:

$$\begin{bmatrix}x_1\\ x_2\\ x_3\\ y_1\\ y_2\\ y_3\\ s_1\\ s_2\\ s_3\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 1\\ 1.0011\\ 0.0015\\ 0.0003\\ 0.9791\\ 0.9791\\ 0.9791\end{bmatrix}$$

(3.2) The Feasibility Modification:

Since the choices of parameter σ and step length α_k are crucial to the performance of the method. Techniques for controlling these parameters, directly and indirectly, give rise to a wide variety of method (see [13]). In the original algorithm and its modification, we have assumed that the starting point is strictly feasible and in particular it satisfies the linear equations $Ax^0 = b$, $A^T y^0 + s^0 = c$.

Since in most problems, however, a strictly feasible starting points are difficult to find, while infeasible-interior-point require only that the components of x^0 and s^0 be strictly positive. We modified the search direction so that it improves feasibility at each iteration, but this requirement entails only a slight change to equations (3.1) and (3.2). Let (x^*, y^*, s^*) be an approximate solution to system (3.1)-(3.4) we define the residuals for the two linear equations as:

$$r_b = Ax^* - b \quad , \quad r_c = A^T y^* + s^* - c \tag{3.8}$$

then the modified system become:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe + \sigma\mu e \end{bmatrix} \tag{3.9}$$

The search direction is still a Newton step toward the point $(x_\tau, y_\tau, s_\tau) \in \rho$. It tries to correct the infeasibility in the equality constraints in a single step. If a full step is taken at any iteration (that is, $\alpha_k = 1$ for some k), the residuals r_b and r_c become zero, and all subsequent iterates remain strictly feasible.

Now, to demonstrate this modification, we solve the same example, by the following modified algorithm:

Given $(x^0, y^0, s^0) \in F^0$

for $k = 0,1,2,\ldots$

and

$r_b = Ax - b \qquad r_c = A^T y + s - c$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -X^k S^k e + \sigma \mu e \end{bmatrix}$$

Where $\sigma \in [0,1]$ and $\mu_k = (x^k)^T s^k / n$

Set

$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k (\Delta x^k, \Delta y^k, \Delta s^k)$

choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$

end(for).

The solution of the example:

Iteration 1

Let

$$S^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad X^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$r_b = Ax - b$

$$r_b = \begin{bmatrix} 0 \\ 0 \\ -7 \end{bmatrix}$$

$$r_c = A^T y + s - c$$

$$r_c = \begin{bmatrix} 8 \\ 7 \\ -6 \end{bmatrix}$$

where

$$\sigma\mu e = \begin{bmatrix} 0.0033 \\ 0.0033 \\ 0.0033 \end{bmatrix}$$

$$-X^0 S^0 e = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

$$-X^0 S^0 e + \sigma\mu e = \begin{bmatrix} -0.9967 \\ -0.9967 \\ -0.9967 \end{bmatrix}$$

Therefor, from eq.(3.٩) we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -8 \\ -7 \\ 6 \\ 0 \\ 0 \\ 7 \\ -0.9967 \\ -0.9967 \\ -0.9967 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -0.0110 \\ -0.0087 \\ -0.0087 \\ 0.0328 \\ 0.0204 \\ 0.0025 \\ -0.9881 \\ -0.9901 \\ -0.9901 \end{bmatrix}$$

choose $\alpha_0 = 0.01$

$$(x^1, y^1, s^1) = (x^0, y^0, s^0) + \alpha_0 (\Delta x^0, \Delta y^0, \Delta s^0)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0.9998 \\ 0.9999 \\ 0.9999 \\ 1.0003 \\ 0.0002 \\ 0.0002 \\ 0.9901 \\ 0.9901 \\ 0.9901 \end{bmatrix}$$

Iteration 2

Now we obtain new X and S

$$X^1 = \begin{bmatrix} 0.9998 & 0 & 0 \\ 0 & 0.9999 & 0 \\ 0 & 0 & 0.9999 \end{bmatrix}, \qquad S^1 = \begin{bmatrix} 0.9901 & 0 & 0 \\ 0 & 0.9901 & 0 \\ 0 & 0 & 0.9901 \end{bmatrix}$$

where

$$r_b = \begin{bmatrix} -0.0001 \\ -0.0001 \\ -0.6989 \end{bmatrix}$$

$$r_c = \begin{bmatrix} 7.9983 \\ 6.8911 \\ -5.9381 \end{bmatrix}$$

$$-X^{1}S^{1}e = \begin{bmatrix} -0.9899 \\ -0.9899 \\ -0.9899 \end{bmatrix}$$

$$\sigma\mu e = \begin{bmatrix} 0.0032 \\ 0.0032 \\ 0.0032 \end{bmatrix}$$

$$-X^{1}S^{1}e + \sigma\mu e = \begin{bmatrix} -0.9867 \\ -0.9861 \\ -0.9861 \end{bmatrix}$$

Therefor, from eq.(3.٩) we get

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & -2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 6 & 1 & 0 & 0 & 1 \\ 1 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9901 & 0 & 0 & 0 & 0 & 0 & 0.9998 & 0 & 0 \\ 0 & 0.9901 & 0 & 0 & 0 & 0 & 0 & 0.9999 & 0 \\ 0 & 0 & 0.9901 & 0 & 0 & 0 & 0 & 0 & 0.9999 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -7.9983 \\ -6.8911 \\ 5.9381 \\ 0.0001 \\ 0.0001 \\ -6.988 \\ -0.9867 \\ -0.9861 \\ -0.9861 \end{bmatrix}$$

By solving the above system we get:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \end{bmatrix} = \begin{bmatrix} -0.0173 \\ -0.0086 \\ -0.0086 \\ 0.0326 \\ 0.0210 \\ 0.0036 \\ -0.9822 \\ -0.9898 \\ -0.9898 \end{bmatrix}$$

Since we get $(X,Y,S) > 0$ $\alpha_1 = \alpha_0$ in the last iteration, then $(x^2, y^2, s^2) = (x^1, y^1, s^1) + \alpha_0(\Delta x^1, \Delta y^1, \Delta s^1)$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0.9999 \\ 0.9998 \\ 0.9998 \\ 1.0006 \\ 0.0004 \\ 0.0002 \\ 0.9879 \\ 0.9895 \\ 0.9895 \end{bmatrix}$$

The above procedure, will be repeated for 5 more different values of α_k, satisfying the condition (2.6b), having the final results:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0.9996 \\ 0.9998 \\ 0.9998 \\ 1.0009 \\ 0.0080 \\ 0.0004 \\ 0.9790 \\ 0.9789 \\ 0.9789 \end{bmatrix}$$

Conclusions and Future work

From the computational results of this work, we could have the following conclusions:

1. The path modification takes less number of iterations than the path-following method, because of the path in primal-dual algorithms plays an effective rule in solving the problem.

2. The feasibility modification takes less number of iterations than the path-following method and the predictor phase, because of the choices of the parameter as well as the positivity of x^0 σ_k and step length α_k and s^0.

3. The hybrid method shows less number of iterations than the path-following method, path modification and the feasibility modification, because of the way of calculating the parameter σ, as well as the change of the right-hand side in (4.1) to be as in (4.4).

For the future work, we are suggested the following:

1. Comparision study between our hybrid method and the simplex method for different structure of linear programming problems.

2. Studying the convergence theory of our hybrid method.

3. Reformulation of the linear systems, especially when the coefficient matrix usually large and sparse, in which implementation of the algorithms to be easier than the original form.

4. Extended our algorithms to a wider class of problems, such as convex quadratic programming problems.

Contents

Examining Committee's Certification

We certify that we read this thesis entitled ***"Modified Algorithms for Solving Linear Programming Problems"***and as examining committee examined the student, ***Yasamen Mueen Mohamed*** in its contents and in what it connected with, and that is in our opinion it meet the standard of thesis for the degree of Master of Science in Mathematics.

(Chairman)	*(Member)*
Signature:	*Signature:*
Name: Dr. Ahmed Molod Al-Ani	*Name:* Dr. Shatha Ahmed Aziz
Date: / /2009	*Date: / /2009*
(Member)	*(Member and Supervisor)*
Signature:	*Signature:*
Name: Dr. Fadhil Sbhi Fadhil	*Name:* Dr. Alauldin Noori Ahmad
Date: / /2009	*Date: / /2009*

Approved by the Dean of College of Science / Al-Nahrain University

Signature:

Name: Assist. Prof. Dr. Laith Abdul-Aziz Al-Ani

Dean of College of Science of Al-Nahrain University

Date: / / 2009

1

Introduction

The development of linear programming started about sixty years ago when G.B. Dantzig and others formulated the simple method (see [2]), in which various surveys confirm that it is still the most widely technique used in practice. Its fields of application range from oil refinery management to hospital diet planning; problems with thousands of variables and constraints are solved routinely by sophisticated commercial codes.

2

Path-Following Method

In the 1980s it was discovered that many large linear programs may be solved efficiently by formulating them as nonlinear problems and solving them with various modifications of nonlinear algorithms such as Newton`s method [11,17,23]. One characteristic of these methods was that they require all iterates to satisfy the inequality constraints in the problem strictly, so they soon became known as interior-point methods.

3

Modifications

In [15, 8, 18, 19], several notes had been suggested, some ideas concerning the path solution and the step length, hopping better performance.

In this chapter, we discussed two modifications to the interior method which have been presented in chapter two. The first modification, concern on the path solution, called path modification, while the second modification, concern the feasibility solution, called feasibility modification.

4

Hybrid Method

In this chapter, we are presenting a hybrid method, based on the results of path and feasibility solution modifications, discussed in chapter three, by combining these two modifications together, we call it as ``hybrid`` method.

References

[1] A. Karmarkar; `` A New Polynomial Time Linear Programming``, Proc. of the Sixteenth Annual ACM Symposium on Theory of Computing, held in Washington. The Association for Computing Machinery, New York, N. (1984), PP. 302-311.

[2] Di R. and G.M.F; ``Introduction To Linear Programming Processes``, Soth Werstern Publishing Company, USA (1967).

[3] E. polak, optimization; ``Algorithms and Consistent In Applied Mathematical Sciences``, Springer, 1997.

[4] Gass S. I.; `` Linear Programming, Methods and Applications ``, 5th edition, Mcgraw. Hill Book Company (1975).

[5] J. Gondzio; ``Multiple Centrality Corrections In a Primal-Dual Method For Linear Programming Computational Optimization and Applications``, 6(1996), pp.137-156.

[6] Jorge N., Stephan J. W.; ``Numerical Optimization``, 1999.

[7] Katta G.Murty; ``Interior Point Method For Linear Programming``, department of Industrial and operation Engineering university of Michigan, 2003.

[8] L. Vandenber Ghe and S.Boyd; `` Semidefinite Programming``, siam review 38(1996), pp.49-95.

[9] M. H. Wright; ``ILL Conditioning and Computating Error In Interior Methods For Nonlinear Programming``, slam journal on optimization, 9(1999), pp. 84-111.

[10] M. H. Wright; ``Interior Methods For Constrained Optimization``, in Acta Numerica, 1992, pp. 341-407.

[11] M. J. Todd; ``Potential Reduction Methods In Mathematical Programming``, Mathematical Programming Series B, 76(1997), pp.3-45.

[12] M. J. Todd and Y.YE ; ``A Centered Projective Algorithm For Linear Programming``, mathematics of operations research, 15(1990), pp.508-529.

[13] Monteiro, R. D. C., Adler,I. Resende and M.G.C.;``A Polynomial Time Primal Dual Affine Scaling Algorithm for Linear and Convex Quadratic Programming and Its Power Series Extension``, Mathematics of Operations Research, The Institute of Management Sciences, Operations Research Society of America, (1990) , Vol.15, No.2, pp.191-214.

[14] N.Megiddo; ``Interior-Point and Related Methods ``, pathways to the optimal set in linear programming in progress in mathematical programming, Springer-Verlag, New York, 1989, 131-158.

[15] R. Chandras Ekaran; ``Linear Programming and Extensions`` , the university of Texas at Dallas 1998.

[16] R.W. Cottle, J-S.Pang, and R.E.Stone; `` The Linear Complementarity Problem``, Academic Press, San Diego, 1992.

[17] Saul I. Gass; `` Linear Programming ``, college of Business and Management University of Maryland.

[18] S. J. Wright; ``Applying New Optimization Algorithms To Model Predictive Control``, in chemical process control-V, J.C. Kantor, ed., 1997.

[19] S. J. Wright; ``Primal-Dual Interior-Point Methods``, SIAM publications, Philadelphia, 1997.

[20] S. Mehrotra; `` On the Implementation of a Primal-Dual Interior Point Method`` , Siam Journal on optimization, 2(1992), pp.575-601.

[21] S.Mizuno, M.Todd and Y.Ye; ``On Adaptive Step Primal-Dual Interior-Point Algorithms For Linear Programming``, mathematics of operations research, 18(1993), pp.964-981.

[22] Todd, M.J. (1989); `` Recent Development and New Directions in Linear Programming``, in Mathematical Programming, Tokyo, PP109-157.

[23] W.L.Winston; ``Operations Research``, wadsworth publishing Co. 3rd ed., 1997.

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Al-Nahrain University, College of Science, in partial fulfillment of the requirement for the degree of Master of Science in mathematics

Signature:

Name: Dr. Alauldin Noori Ahmad

Date: / / 2008

In view of the available recommendations I forward this thesis for debate by the examining committee.

Signature:

Name: Assist. Prof. Dr. Akram M. Al-Abood

Head of the Department mathematics and computer applications

Date: / / 2008

الاهــداء

الى من اتخذه مثالا أقتدي به...

الى الشمس التي تنير لي الطريق
والدي العزيز

الى من شجعتني وضحت من أجلي...

الى البحر الذي يغمرني بحنانه
والدتي العزيزة

الى اقرب الناس لي وخيرهم نصحا وعونا اخي (محمد)

الى من رسمت على شفتي ابتسامة الامل اختي (نسرين)

الى مكمن قوتي في هذه الحياة اخوتي الاعزاء (علي ، حسن)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ

أُوتُوا الْعِلْمَ دَرَجَتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴾

صَدَقَ اللَّهُ العَظِيم

(سورة المُجادلة /جزء من الاية 11)

Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science

Modified Algorithms for Solving Linear Programming Problems

A Thesis

Submitted to the College of Science, AL-Nahrain University
as a Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mathematics

By

Yasamen Mueen Mohammed Al_Asadi
(B.Sc., Al-Nahrain University, 2003)

Supervised by
Dr. Alauldin Noori Ahmed

September 2008 Shawwal 1429

وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

خوارزميات محورة لحل مسائل البرمجة الخطية

رسالة
مقدمة إلى كلية العلوم – جامعة النهرين
وهي جزء من متطلبات نيل درجة ماجستير علوم
في الرياضيات

من قبل

ياسمين معين محمد الاسدي

(بكلوريوس علوم ، جامعة النهرين، ٢٠٠٣)

بأشراف

د. علاء الدين نوري احمد

ايلول ٢٠٠٨ شوال ١٤٢٩

المستخلص

في هذا العمل تم دراسة خوارزمية Path-Following والتي هي احد الخوارزميات المسماة ``Interior-Point Algorithms`` والتي تعالج حلول النماذج الرياضية الخطية.

تم مناقشة هذه الخوارزمية بأجراء تحويرين. التحوير الاول يتعلق Path Solution أما التحوير الثاني فيتعلق Feasibility Solution. وقد تم ربط هذين التحويرين بأسلوب جديد للحصول على طريقة مهجنة سميت ``Hybrid Method`` وقد اعتمدت مسألة رياضية واحدة في اختيا ر واعتماد الحل لكل خوارزمية. باللأضافة الى ذلك، تم اعتماد عدد من المسائل لغرض المقارنة حيث اظهرت الخوارزمية الهجينة نتائج افضل من حيث عدد التكرارات والدقة في النتائج.