

## OPTIMIZACIÓN DE SISTEMAS LINEALES USANDO MÉTODOS DE PUNTO INTERIOR

### RESUMEN

La técnica de optimización denominada *puntos interiores* evoluciona por el *interior* de la región factible a diferencia del método SIMPLEX que evoluciona por sus puntos extremos, disminuyendo considerablemente el tiempo de solución de los problemas. En problemas de optimización no lineal de gran tamaño, las técnicas existentes para su solución deben resolver muchos problemas de programación lineal (PL) sucesivos, cada uno de los cuales requiere de gran tiempo de procesamiento. Este artículo presenta un método alternativo denominado *punto interior* aplicable a esta clase de problemas, con el propósito de disminuir los tiempos de procesamiento.

**PALABRAS CLAVES:** Modelos Lineales, Métodos de Puntos Interiores.

### ABSTRACT

*The optimization techniques that use interior points take advantage of the characteristics of the problem to move inside the feasible region. In contrast, simplex method move on the boundary. The method of interior point reduces the execution times in large size problems. In techniques that generally use successive PL large problems that demand relatively large processing time, this method is a good alternative. This paper presents an implementation of an interior point algorithm for this problems for the reduction of processing time.*

**KEYWORDS:** Interior point Method, Linear Programming.

### 1. INTRODUCCIÓN

Las técnicas de optimización se utilizan en diversas áreas del conocimiento en el manejo de procesos, recursos, ganancias, inversión, entre otros y estos pueden ser formulados como problemas de programación no lineal restringidos, los cuales asumen la siguiente forma:

$$\begin{aligned} \min f(x) & \quad (1) \\ \text{s.a.} & \\ g_i(x) \leq 0 \quad i=1,2,\dots,m & \end{aligned}$$

donde  $x \in R^n$  y  $f(x)$  y  $g(x)$  son funciones no lineales. Otros problemas son lineales y para su solución se emplean técnicas basadas en el método SIMPLEX [1].

Se asume que un problema lineal toma la forma:

$$\begin{aligned} \min b^T y & \quad (2) \\ \text{s.a.} & \\ A^T y = c & \\ y \geq 0 & \end{aligned}$$

donde  $A$  es una matriz de  $m \times n$ ,  $b$  y  $y$  son vectores de dimensión  $m$  y  $c$  es un vector de dimensión  $n$ .

### EDGAR M. CARREÑO F.

Ingeniero Electricista, M. Ss  
Universidad Tecnológica de Pereira  
emfra@utp.edu.co

### ELIANA M. TORO OCAMPO

Ingeniera Industrial.  
Estudiante Maestría IE  
Universidad Tecnológica de Pereira  
mirledyt@yahoo.com

### ANTONIO ESCOBAR Z.

Ingeniero Electricista, M.Sc  
Profesor Asociado  
Universidad Tecnológica de Pereira  
aescobar@utp.edu.co

Grupo de Planeamiento Eléctrico,  
área de Investigación Operativa.

Los problemas no lineales pueden ser resueltos como una secuencia de problemas de programación lineal (PL) [2] en sistemas de complejidad pequeña o media. En sistemas de gran complejidad se requieren técnicas, como por ejemplo las combinatoriales, que también requieren de la solución de subproblemas con miles de PL.

Tradicionalmente los PL se resuelven usando el método SIMPLEX, el cual evoluciona por los puntos extremos de la región factible hasta encontrar el óptimo del problema. En sistemas de gran tamaño el número de vértices aumenta exponencialmente con el número de variables, y por lo tanto, el proceso de búsqueda puede requerir gran tiempo de cómputo. El método de *puntos interiores*, surge como una forma alternativa para la solución de problemas lineales y no lineales con tiempos de cómputo inferiores, debido a que evoluciona por el interior de la región factible.

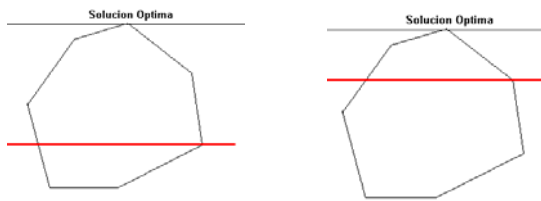
Aunque el método fué propuesto inicialmente para resolver problemas no lineales, los sistemas de cómputo disponibles en la época no permitieron su avance y en cambio se desarrolló una idea para problemas lineales propuesto por Karmarkar [4] denominado algoritmo de escalamiento proyectivo. El método de Karmarkar [4,5] ha sido estudiado e implementado en varias formas que son considerablemente más rápidas que el método SIMPLEX estándar. Posteriormente se presentaron nuevos métodos que básicamente eran simplificaciones

de los algoritmos de escalamiento y posteriormente se presentaron los métodos conocidos como primales-duales que usan el concepto de barrera logarítmica [6].

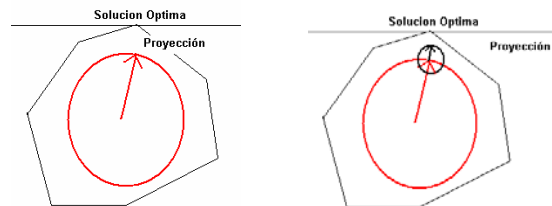
En este artículo se presentan dos métodos de *punto interior*, el método de escalamiento proyectivo [4] y un método primal-dual [6], los cuales se ilustran con un ejemplo gráfico para su entendimiento.

**2. MÉTODO DE ESCALAMIENTO PROYECTIVO.**

Geoméricamente, la diferencia entre el SIMPLEX y este método de punto interior se muestra en las figuras 1 y 2



Primera Iteración Simplex Segunda Iteración Simplex  
Figura 1 Dos iteraciones SIMPLEX



Primera Iteración Interior Segunda Iteración Interior  
Figura 2 Dos iteraciones de Escalamiento

Teniendo el problema mostrado en (2), es más fácil resolver su dual:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b \end{aligned} \tag{3}$$

Una propiedad importante de los problemas dados por (2) y (3) es que su solución óptima es igual, esto es:

$$b^T y^* = c^T x^*$$

donde  $x^*$  es la solución óptima del problema dual (3) y  $y^*$  es la solución óptima del problema primal (2). Se considera una variación de [5] con el fin de obtener la solución óptima  $x^*$  de (3). Se asume que  $c$  y  $x$  son vectores de dimensión  $n$ , y  $b$  es vector de dimensión  $m$ ,  $A$  es una matriz de  $m \times n$  de rango completo ( $m \geq n$ ) y  $c \neq 0$ . También se requiere un punto inicial  $x^0$  factible.

Comenzando con  $x^0$ , el algoritmo genera una secuencia de puntos interiores factibles  $\{x^1, x^2, \dots, x^k, \dots\}$  con un valor de función objetivo creciente monótonamente.

$$Ax^k < 0 \text{ y } c^T x^{k+1} > c^T x^k \tag{4}$$

El algoritmo termina cuando se cumple un criterio de parada. Introduciendo las variables de holgura  $v$  al problema dual (3), se tiene:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.a.} \quad & Ax + v = b \\ & v \geq 0 \end{aligned} \tag{5}$$

donde  $v$  es el vector de dimensión  $m$  de las variables de holgura. En cada iteración,  $k$ , con  $v^k$  y  $x^k$  como la iteración actual, una transformación lineal se aplica al espacio de solución:

$$\hat{v} = D_k v$$

con:

$$D_k = \text{diag} \left[ \frac{1}{v_1^k}, \frac{1}{v_2^k}, \dots, \frac{1}{v_m^k} \right]$$

Las variables de holgura se escalan de tal manera que  $x^k$  sea equidistante a todos los hiperplanos, generando el espacio cerrado cuya intersección forma el espacio factible del conjunto de restricciones. Reescribiendo (5) en términos de las variables de holgura escaladas, se tiene:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.a.} \quad & Ax + D_k^{-1} \hat{v} = b \\ & \hat{v} \geq 0 \end{aligned} \tag{6}$$

El conjunto de las variables de holgura escaladas factibles están en el conjunto:

$$S = \left\{ \hat{v} \in R^n \mid Ax + D_k^{-1} \hat{v} = b \text{ para } x \in F \right\}$$

Si la matriz  $A$  es de rango completo, entonces

$$\hat{v} = D_k (b - Ax)$$

y

$$x(\hat{v}) = (A^T D_k^2 A)^T D_k (D_k b - \hat{v})$$

La nueva dirección factible  $d_x^k$  es obtenida de la solución de un sistema lineal de ecuaciones:

$$(A^T D_k^2 A) d_w^k = c \tag{7}$$

La dirección correspondiente de las variables de holgura es:

$$d_v^k = -Ad_x^k \quad (8)$$

Esto lleva a la siguiente fórmula de actualización:

$$x^{k+1} = x^k + \alpha d_x^k \quad (9)$$

Una propiedad importante es que  $x^k$  converge a la solución óptima  $x^*$  del problema (3) y converge a la solución primal  $y^*$  del problema (2). Lo anterior puede ser sistematizado a través del siguiente pseudo código:

1. Sea  $x^0$  un punto interior, escoger un  $\gamma$  de tal manera que:

$$Ax^0 < b \text{ y } 0 < \gamma < 1, k = -1$$

2. Calcular la holgura (distancia del punto a cada restricción).  $k = k + 1$ :

$$v^k = b - Ax^k = [v_1^k, v_2^k, \dots, v_m^k]^T$$

3. Calcular la matriz diagonal:

$$D_k = \text{diag} \left[ \frac{1}{v_1^k}, \frac{1}{v_2^k}, \dots, \frac{1}{v_m^k} \right]$$

4. Calcular la proyección del punto actual sobre la periferia de la esfera. Resolver el sistema de ecuaciones para  $d_x^k$ :

$$(A^T D_k D_k A) d_x^k = c \quad (10a)$$

5. Escalar la dirección proyectada  $d_x^k$

$$d_v^k = -Ad_x^k = [(d_v)_1, (d_v)_2, \dots, (d_v)_m]^T$$

6. Calcular la longitud del paso a tomar en la dirección  $d_x^k$ :

$$\alpha = \gamma \max \left\{ \frac{v_i^k}{(d_v)_i} < 0 : (d_v)_i < 0 : i = 1, 2, \dots, m \right\}$$

Calcular el nuevo punto interior:

$$x^{k+1} = x^k - \alpha d_x^k$$

7. Para detener el algoritmo, se calcula la solución primal:

$$y^k = D_k^2 d_v^k$$

si  $\frac{|b^T y^k - c^T x^k|}{\max \{1, |c^T x^k|\}} < \varepsilon = 10^{-6}$  FIN, de lo contrario, ir al paso 2.

El paso 2 asegura que el punto interior se encuentra dentro de la región factible, esto es,  $v_i > 0$ . Aunque  $v_i$  se acerque a 0 por muchas restricciones, nunca es igual a 0. Los pasos 5 y 6 ajustan el elipsoide interior para que sea tan grande como sea posible en la región factible. El paso 7 asegura que el *gap* de la dualidad vaya a 0, esto es, la diferencia entre los valores de las funciones objetivo primal y dual sea 0.

### 3. METODO PRIMAL-DUAL

El método primal-dual presentado soluciona un problema de programación lineal de la forma:

$$\text{Min } c^T x \quad (10)$$

s.a.

$$g(x) = 0$$

$$h(x) \leq 0$$

Donde  $c$ ,  $x$ ,  $g(x)$ ,  $h(x)$ , representan, respectivamente, el vector de coeficientes de costo  $R^n$ , variables del problema en  $R^n$ , vector de restricciones lineales de igualdad  $R^n \rightarrow R^m$ , vector de restricciones lineales de desigualdad  $R^n \rightarrow R^p$ .

El primer paso consiste en transformar todas las restricciones de desigualdad en igualdad con la adición de variables de holgura (11) de la siguiente manera:

$$\text{Min } c^T x \quad (11)$$

s.a.

$$g(x) = 0$$

$$h(x) + s \leq 0$$

$$s \geq 0$$

La condición de no negatividad puede ser considerada introduciendo un término de barrera logarítmica, imponiendo una condición de positividad en las variables de holgura de una manera implícita:

$$\text{Min } f_\mu = c^T x - \mu \sum_{i=0}^p \ln(s_i) \quad (12)$$

s.a.

$$g(x) = 0$$

$$h(x) + s = 0$$

Donde  $\mu^k \geq 0$  y el parámetro de barrera es monótonamente decreciente con el proceso iterativo hasta llegar a un valor de cero. Una secuencia de parámetros  $\{\mu^k\}_{k=0}^\infty$  genera una serie de subproblemas dados por (12), y como  $\mu \rightarrow 0$ , una secuencia de soluciones

$\{x(\mu^k)\}_{k=0}^{\infty}$  del problema (12) se aproxima a  $x^*$ , óptimo local del problema (11) [7].

Gráficamente, el comportamiento de la barrera logarítmica se puede observar en la figura 3.

La función lagrangeana resultante del subproblema (12) con función logarítmica es:

$$L_{\mu} = c^T x - \mu \sum_{i=1}^p \ln(s_i) - \lambda^T \cdot g(x) - \pi^T \cdot (-h(x) - s) \quad (13)$$

donde  $\lambda$  y  $\pi$ , representan respectivamente el vector de los multiplicadores de lagrange de las restricciones de igualdad  $R^m$  y de desigualdad  $R^p$ . Estos multiplicadores de lagrange son las denominadas variables duales del problema.

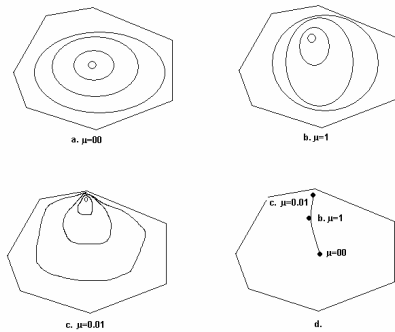


Figura 3. Barrera Logarítmica

Un mínimo local del problema (12) es expresado en términos de un punto estacionario de  $L_{\mu}$ , el cual debe satisfacer las condiciones necesarias de primer orden de Karush - Kuhn Tucker (KKT):

$$\nabla_x L_{\mu} = c + \nabla_x g(x) \cdot \lambda + \nabla_x h(x) \cdot \pi = 0 \quad (14)$$

$$\nabla_{\lambda} L_{\mu} = g(x) = 0 \quad (15)$$

$$\nabla_{\pi} L_{\mu} = h(x) + s = 0 \quad (16)$$

$$\nabla_s = -\mu^k S^{-1} \cdot e + \pi = 0 \quad (17)$$

donde  $\nabla g(x)$ ,  $\nabla h(x)$ ,  $S$  y  $e$ , representan: la matriz de los vectores gradientes de las restricciones de igualdad, la matriz de los vectores gradiente de las restricciones de desigualdad, la matriz diagonal definida por los componentes de  $S$  y el vector de "unos" de dimensión apropiada.

En las condiciones de KKT, las ecuaciones (15) y (16) junto con  $s \geq 0$  asegura la factibilidad del primal, la ecuación (14) en conjunto con  $\pi \geq 0$  garantiza la

factibilidad dual, y la ecuación (17) representa las ecuaciones de complementariedad perturbadas.

Las iteraciones del MPI primal-dual aplican invariablemente una iteración del método de newton para sistemas no lineales para resolver el sistema (14)-(17), calcula un tamaño de paso en la dirección de newton, actualiza las variables y reduce el parámetro  $\mu_k$ . El algoritmo termina cuando las infactibilidades primal y dual, y la distancia de complementariedad son menores que una tolerancia predeterminada.

#### A. Cálculo de la dirección de búsqueda.

Para un sistema de KKT no lineal, la solución es aproximada por una iteración del método de Newton (la dirección de Newton es sólo un medio para seguir el camino de minimización parametrizado por  $\mu$ . Entonces, al aplicar el método de Newton al sistema (18)-(21), para la determinación de la dirección de búsqueda, se obtiene el siguiente sistema simétrico:

$$\begin{bmatrix} 0 & \nabla_x g(x) & \nabla_x h(x) & 0 \\ \nabla_x g(x)^T & 0 & 0 & 0 \\ \nabla_x h(x)^T & 0 & 0 & I \\ 0 & 0 & I & \Pi/S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta \pi \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla_x L_{\mu} \\ \nabla_{\lambda} L_{\mu} \\ \nabla_{\pi} L_{\mu} \\ \nabla_s L_{\mu} \end{bmatrix} \quad (18)$$

Donde  $\Pi/S$  representa una matriz diagonal definida por los componentes de  $\pi$  divididos por los componentes correspondientes de  $s$ .

El sistema (18) posee las siguientes características: dimensión elevada, alto grado de dispersión, simetría, casi invariable en el proceso iterativo por que sólo los elementos de la matriz  $\Pi$  varían en el proceso.

#### B. Actualización de las variables

Las nuevas variables primales y duales son calculadas como:

$$x^{k+1} = x^k + k_s \cdot \alpha_p \cdot \nabla x \quad (19)$$

$$s^{k+1} = s^k + k_s \cdot \alpha_p \cdot \nabla s$$

$$\lambda^{k+1} = \lambda^k + k_s \cdot \alpha_D \cdot \nabla \lambda$$

$$\pi^{k+1} = \pi^k + k_s \cdot \alpha_D \cdot \nabla \pi$$

donde  $\alpha_p$ ,  $\alpha_D$ ,  $k_s$ , representan: el paso primal, el paso dual, y el factor de seguridad que garantiza estricta positividad de las variables  $k_s = 0.99995$ .

#### C. Cálculo del tamaño de los pasos primal y dual.

Una vez resuelto el sistema lineal (18), se obtiene una dirección la cual encuentra un nuevo punto dentro del proceso. Para hacer una actualización de las variables, es preciso saber el tamaño de los pasos primal y dual. Estos pasos son calculados de tal forma que garantizan la no negatividad de las variables  $s$  y  $\pi$ , lo que quiere decir que existen pasos primal y dual máximos, que son calculados de la siguiente manera:

$$\alpha_{P_{\max}} = \min \left[ \min_{\Delta s_j < 0} \frac{s_j}{|\Delta s_j|}, 1 \right] \quad (20)$$

$$\alpha_{D_{\max}} = \min \left[ \min_{\Delta \pi_j < 0} \frac{\pi_j}{|\Delta \pi_j|}, 1 \right] \quad (21)$$

Los valores de los pasos primal y dual pueden ser seleccionados entre los intervalos  $\alpha_P \in [0, \alpha_{P_{\max}}]$  y  $\alpha_D \in [0, \alpha_{D_{\max}}]$ , respectivamente y de forma independiente, por lo tanto, cada valor puede ser diferente. Generalmente, en el caso de PL, son aplicados los pasos máximos en la actualización de las variables.

#### D. Actualización del parámetro de barrera.

Un paso importante dentro del MPI es escoger un parámetro de barrera  $\mu$ . Existen varias propuestas para seleccionar este parámetro, todas basadas en el *gap* de dualidad o en el *gap* de complementariedad del problema. En este trabajo se usa el de complementariedad por que está directamente relacionado con  $\mu$  [8], [9].

$$\rho = \pi^T \cdot s \quad (22)$$

$$\mu = \sigma \cdot \frac{\rho}{p} \quad (23)$$

donde  $\rho$ ,  $p$  y  $\sigma$  representan respectivamente: el *gap* de complementariedad, el número de restricciones de desigualdad y el parámetro de combinación de direcciones que en el MPI primal-dual es  $\sigma = 0.1$ .

#### E. Criterio de parada

Definiendo:

$$v_1^k = \max \left\{ \max \{h(x^k)\}, \|g(x^k)\|_{\infty} \right\} \quad (28)$$

$$v_2^k = \frac{\|c + \nabla_x g(x^k) \cdot \lambda^k + \nabla_x h(x^k) \cdot \pi^k\|_{\infty}}{1 + \|x^k\|_2} \quad (29)$$

$$v_3^k = \frac{\rho^k}{1 + \|x^k\|_2} \quad (30)$$

$$v_4^k = \mu^k \quad (31)$$

El proceso finaliza si los cuatro valores anteriores son menores a unas tolerancias definidas para convergencia. La ecuación (28) representa el error primal, la ecuación (29) es el error dual, las condiciones (26) y (27) indican que el problema parametrizado por  $\mu$  es igual al problema original sin el parámetro.

#### F. Punto Inicial

La única condición que debe cumplir el punto inicial es  $(s, \pi) \geq 0$ , la cual tiene que ser cumplida durante todo el proceso iterativo [11].

#### F. Algoritmo de MPI para PL

1. Inicialización: punto inicial  $(x^0, \lambda^0, s^0, \pi^0)$ , con  $(s^0, \pi^0) \geq 0$  y  $\mu^0 \geq 0$ .
2. Cálculo de la dirección de búsqueda: solucionar el sistema (18) para el punto actual.
3. Cálculo del tamaño de los pasos primal y dual usando (20) y (21).
4. Actualizar las variables primal y dual con (19).
5. Prueba de convergencia: usar (24)-(27), si se cumple el criterio de convergencia, FIN, de lo contrario, retornar al paso 2 actualizando el parámetro de barrera  $\mu$  con (23).

### 4. PRUEBAS Y RESULTADOS

Para ilustrar la forma en que cada uno de estos métodos soluciona un PL, se ha seleccionado el siguiente sistema de prueba:

$$\begin{aligned} &\max \quad X1 + 3X2 \\ &s.a. \\ &\quad X1 - 3X2 \leq 3 \\ &\quad -2X1 + X2 \leq 2 \\ &\quad -3X1 + 4X2 \leq 12 \\ &\quad 3X1 + X2 \leq 9 \\ &\quad X1, X2 \geq 0 \end{aligned}$$

La solución óptima de esta Problema de programación lineal (PL) es:  $X1 = 1.6$  y  $X2 = 4.2$ .

Al resolver este problema mediante escalamiento, se obtienen las siguientes respuestas:

Iteración	0	1	2	3	4
X1	0.1	1.4463	1.6052	1.5986	1.600
X2	0.1	4.0549	4.1783	4.1987	4.1998

Tabla 1. Iteraciones por el método de escalamiento

El camino seguido se muestra en la figura 4

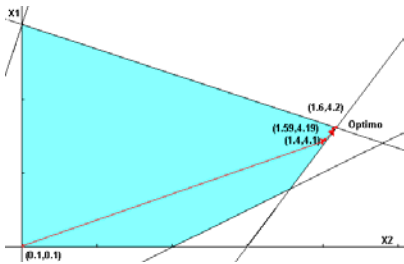


Figura 4.

Una de las desventajas de este método, es que el punto inicial debe de estar dentro de la región factible, no puede ser la solución trivial como en la mayoría de los métodos para evitar la singularidad de la matriz  $D_k$ . Obsérvese como en la primera iteración la solución se acerca al óptimo.

Utilizando el método de barrera logarítmica se obtienen las siguientes respuestas:

Iteración	0	1	2	3	4	5
X1	0	0.7350	0.7876	1.3232	1.5894	1.6
X2	0	1.1749	1.9970	3.4474	4.1714	4.2

Tabla 2. iteraciones por el método de barrera logarítmica

La figura 5 muestra su evolución. A diferencia del método anterior, el punto inicial no necesariamente tiene que ser un punto interior, podría ser cualquier punto. Para el ejemplo se adoptó como punto inicial (0,0) para fines de comparación. En este método, llegar a la vecindad del óptimo toma más tiempo. La verdadera fortaleza de este método es la solución de problemas no lineales, los cuales se analizarán en artículos posteriores.

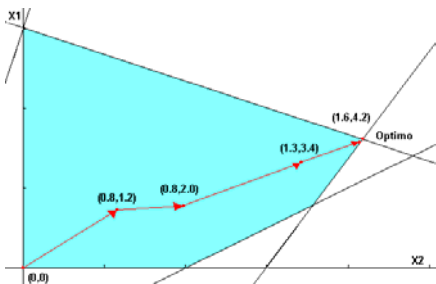


Figura 5.

**5. CONCLUSIONES**

Se han presentado dos métodos de programación lineal usando puntos interiores, estos métodos presentan un comportamiento distinto al SIMPLEX, lo que se muestra como alternativa interesante para resolver problemas de gran tamaño y complejidad matemática.

**6. AGRADECIMIENTOS**

Los autores expresan su agradecimiento a la Universidad Tecnológica de Pereira por su apoyo al grupo de investigación en planeamiento de sistemas eléctricos.

**7. BIBLIOGRAFÍA**

[1] G. Dantzig. Linear Programming and extensions . Princeton, NJ: Princeton Unive Press, 1963.

[2] M.S. Bazaraa and C.M Shetty, Nonlinear Programing: Theory and algorithms, Toronto: Wiley 1970

[4] N. Karmarkar, “A new polynomial –time algorithm for linear programming”, Combinatorica, vol 4, no 4, pp 373-395, 1984.

[3] Vanelli, A. “Teaching Large Scale Optimization by an Interior Point Approach”, IEEE trans on Education, Vol 36, No 1, pp 204-209, Feb 1993,

[5] I. Adler, N. Karmarkar, M.G.C. Resende, and G. Veiga, “An implementation of Karmarkar’s algorithm for linear programming,” Mathematical Programming, vol 44, pp 297-335, 1989

[6] Mehrotra, S. “on the implementation of a Primal Dual Interior Point Method”, SIAM Journal on Optimization, vol 2, pp 575-61, 1992.

[7] A. V. Fiacco, G.P. McCormick, Nonlinear programming: sequential unconstrained minimization techniques, Jhon Wiley & Sons, 1968.

[8] S. Granville, "Optimal reactive dispatch through interior point methods", IEEE Transactions on Power System, vol 9, pp136-146, Feb 1994.

[9] Y.Ch. Wu, A.S Debs, R.E Marsten, "A direct predictor-corrector primal dual interior point algorithm for optimal power flows," IEEE Transactions on Power systems, vol 9, pp 876-883, May 1994

[10] G.L Torres, V.H. Quintana, "An interior point method for nonlinear optimal power flow using voltage rectangular coordinates", IEEE Transactions on Power Systems, vol 13, pp 1211-1218, Nov 1998.

[11] R.J. Vanderbei, Linear Programming: foundations and extensions, Second Edition, <http://www.princeton.edu/~rvdb/LPbook/>, Dec 2000