İÇ NOKTA ALGORİTMALARI VE DOĞRUSAL PROGRAMLAMAYA UYGULANMASI

Gülnur KEÇEK*

ÖZET

Karmarkar-İç Nokta Algoritması, 1984'te Narendra Karmarkar tarafından geliştirilmiş olup; iç nokta algoritmalarının ana algoritması olarak bilinen bir algoritmadır. İç nokta algoritmalarında, uygun bölge içerisindeki bir noktadan başlayıp; her bir adımda uygun bölgenin iç noktalarında var olan daha iyi bir çözüme gidilerek optimal çözüme ulaşılmaya çalışılır. Çalışmamızın amacı, doğrusal programlama probleminin kısa sürede çözülmesinde Karmarkar- İç Nokta Algoritmasının ve etkin bir iç nokta algoritması olan Mehrotra tahminci-düzeltici Algoritması'nın etkinliğinin gösterilmesidir. Bir üretim işletmesine ilişkin 4500 karar değişkeni ve 180 kısıtlayıcı içeren bir Doğrusal programlama modeli oluşturulmuştur. Model, iç nokta algoritmaları ve Simpleks algoritması ile çözülerek, çözüm sonuçları karşılaştırılmıştır. Modelin çözümü için MOSEK, PCx, XPRESS-MP/Barrier ve XPRESS-MP/Simplex yazılımlarından yararlanılmıştır.

Anahtar Kelimeler: İç nokta algoritmaları, Birim Simpleks.

ABSTRACT

Karmarkar's Interior Point Algorithm is an algorithm developed by Narendra Karmarkar in 1984 and known the major algorithm of interior point algorithms. In Interior point algorithms, it is tried to reach for the optimum solution by starting from an available one and gradually continuing for better ones which lie in the interior points of the available area. The purpose of our study is to present the efficiency of Karmarkar's Interior Point Algorithm and Mehrotra Predictor-Corrector Algorithm, which is an other effective interior point algorithm in solving the linear programming problem in a very short time. A linear programming model has been formulated that has 4500 decision variables and 180 constraints. This model has been solved by interior point algorithms and Simplex Algorithm and they have been compared. MOSEK, PCx, XPRESS-MP/Barrier and XPRESS-MP/Simplex softwares are used for the solution of the model.

Keywords: Interior point algorithms, unit simplex.

GİRİŞ

İç nokta algoritmaları, Yöneylem Araştırması'nda teorik ve uygulama yönünden büyük etkilere yol açmıştır. Yöneylem Araştırmasında yeni bir çalışma alanı ortaya konulmuş ve hızla ilerlemeler sağlanmıştır.

---

* Yrd.Doç.Dr. DPÜ İİBF İşletme Bölümü Sayısal Yöntemler Anabilim Dalı

İç nokta algoritmaları, Yöneylem Araştırması'nın yeni araştırma alanlarından biridir. Araştırmacılar, doğrusal olmayan programlama, kuadratik programlama ve yarı belirli (semidefinite) programlama alanlarında da iç nokta algoritmalarını uygulamaya yönelmişlerdir(Nash ve Sofer, 1996; Wright, 1997).

1. İÇ NOKTA ALGORİTMALARI

1947 yılında Dantzig Simpleks tekniğini geliştirdikten sonra bazı araştırmacılar, iç nokta algoritmaları önermişlerdir. Bu araştırmacıların başta gelenleri, Von Neuman (1947), Hoffman v.d. (1953) ve Frisch (1955) olarak sayılabilir. 1979 yılında Khachian tarafından ortaya konan Ellipsoid tekniği, uygulamada oldukça yavaş bir tekniktir. Bununla birlikte, 1947-1955 yıllarında önerilen bu teknikler, teknolojik yetersizlikler sebebiyle Simpleks tekniği ile rekabet edecek düzeyde değildir(Andersen ve diğerleri,1996; Bland ve diğerleri,1981; Ye,1987; Goldfarb ve Todd, 1989).

N. K. Karmarkar, 1984'te AT & T Laboratuarlarında DP için yeni bir polinom zamanlı algoritma önermiştir. Günümüzde Karmarkar Algoritması olarak biline bu algoritma, Ellipsoid tekniğinin tersine, Simpleks tekniğine rakip olabilecek bir potansiyele sahiptir(Goldfarb ve Todd, 1989).
Daha sonra araştırmacılar, Karmarkar algoritmasından hareketle, bazıları yeni fikirler içeren, bazıları sadece küçük değişiklikler içeren polinom zamanlı yeni teknikler önermişlerdir(Hertog, 1994; Kojima ve diğerleri, 1991).

İç nokta algoritmaları, affine scaling algoritmalar , merkezi yörüngeyi izleyen algoritmalar, izdüşümsel algoritmalar ve potansiyel azaltma algoritmaları olmak üzere dört grupta incelenebilir(Hertog ve Roos,1991; Nash ve Sofer, 1996).

2. KARMARKAR ALGORİTMASI VE DEĞİŞTİRİLMİŞ TÜRLERİ

İç nokta algoritmalarından Ana algoritma olarak bilinen ve büyük ölçekli modellerde etkin olan Karmarkar algoritması ve ona bağlı geliştirilen Mehrotra tahminci-düzeltici Algoritması açıklanmaya çalışılacaktır.

2.1. Karmarkar Algoritması

Karmarkar algoritması, uygun bölgede belirlenen bir alanı tamamen eniyilemek için izlenen izdüşümsel dönüşümlerin yinelenmesine dayanır(Ye ve Kojima, 1987; Tardos, 1986; Knowles,1989).

Delta Havayolları, 400'den fazla uçak ve 7000 pilot için aylık uçuş çizelgelemesini Karmarkar Algoritması'nın çözümü için AT&T tarafından geliştirilen KORBX adlı ticari yazılım ile gerçekleştirmiştir(Render ve Stair,1997).

2.1.1. Temel Kavramlar

Karmarkar Algoritması'nın hesaplama sürecine geçmeden önce iç nokta ve birim simpleks kavramları açıklanmaya çalışılacaktır.
İç Noktalar: Standart formdaki bir DP modelinin uygun çözüm alanı; $P = \{\mathbf{x} \in R^n | \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$ olmak üzere, modelin iç nokta uygun çözüm alanı, $P^0 = \{\mathbf{x} \in R^n | \mathbf{Ax} = \mathbf{b}, \mathbf{x} > 0\}$ şeklinde ifade edilir. Herhangi bir iç nokta yaklaşımı için $P^0 \neq \phi$ temel varsayımı yapılır**(**Fang ve Puthenpura,1993; Gay,1987;Rardin,1998).

Birim Simpleks : Birim Simpleks, Karmarkar Algoritması'nda önemli bir rol oynar. Birim simpleks, S_x, $\sum_{i=1}^{n} x_i = x_1 + x_2 + ... + x_n = 1$ ve $x_j \geq 0$ koşullarını sağlayan $[x_1, x_2, ..., x_n]^T$ noktalar kümesidir ve $S_x = \{\mathbf{x} | \mathbf{1x} = 1, x \geq 0\}$ ifadesi, (n – 1) boyutlu simpleksi gösterir(Fang ve Puthenpura,1993; Goldfarb ve Todd,1989; Roos ve diğerleri,1997).

2.1.2. Karmarkar Algoritmasının Matematiksel Yapısı

Karmarkar Algoritması'nın değişik şekilleri bulunmaktadır. Standart Karmarkar Algoritması, bir doğrusal programlama probleminin aşağıdaki gibi verildiğini varsaymaktadır(Moon ve Stirling, 2000; Schrijver,1998).

$$Min \ \mathbf{cx}$$

Kısıtlayıcılar

$$\mathbf{Ax} = 0$$
$$\mathbf{1x} = 1$$

Burada, $\mathbf{A}$: m x n boyutlu katsayılar matrisi olup $\mathbf{x} \in R^n$ dir.

Karmarkar uygun bölgeyi bir politop(çokyüzlü) varsayar. Mevcut olan iç çözüm, politopun merkezine yakınsa, bir minimum değere ulaşmak için amaç fonksiyonunun dik iniş(steepest descent) doğrultusunda hareket etmesi uygun olur. Bir iç çözümü, dönüştürülmüş uzayda politopun merkezine yakın yerleştirmek için dönüşüm yapılır.

Karmarkar Algoritması'nın Varsayımları
Karmarkar Algoritması, Karmarkar'ın standart biçimindeki DP problemini aşağıdaki varsayımlarla çözer(Chong ve Zak,1996; Goldfarb ve Todd,1989; Gonzaga, 1991);

a) Problemin amaç fonksiyonunun optimal değeri sıfırdır.
b) Birim simpleksin(S_x) merkezi $x_0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$ olup; uygun bir noktadır.
c) Bir L parametresi verildiğinde, $\mathbf{cx}_{k+1} < 2^{-L}$ eşitsizliğini sağlayan bir uygun nokta elde edildiğinde optimal çözüme ulaşılmıştır. Bilinmeyen vektör $\mathbf{x} \in R^n$ ve $\mathbf{A}$ matrisi m x n boyutludur.
d) $(m+1)xn$ boyutlu matris $\begin{bmatrix} \mathbf{A} \\ \mathbf{1} \end{bmatrix}$ olup; rankı $(m+1)$ olur.

2.1.3. Doğrusal Programlama Modelinin Karmarkar Algoritması İle Çözümü

Karmarkar uygun hale getirilmekte, algoritma problemin çözümüne uygulanmaktadır.

2.1.3.1 DP Modelinin Karmarkar Algoritmasına Uygun Hale Getirilmesi

Bilinen minimum amaçlı standart biçimdeki DP modelini, Karmarkar'ın standart biçimine dönüştürmek için aşağıdaki adımlar izlenir(Fang ve Puthenpura,1993);

Adım 1: Karmarkar'ın standart biçiminin temel özelliği, sınırlı bir uygun bölgede sonuçlanan birim simpleks yapısıdır. Bu sebeple, modele $\sum_{j=1}^{n} \mathbf{x}_j \leq Q$ şeklinde sınırlayıcı bir kısıtlayıcı eklenir.

Burada Q pozitif bir tamsayıdır. L problemin büyüklüğünü göstermek üzere, $Q = 2^L$ olarak seçilebilir(Fang ve Puthenpura,1993; Taha,2000).

Adım 2: Modele x_{n+1} aylak değişken eklenir ve DP modeli aşağıdaki şekle dönüşür;

$$\begin{aligned}
&Min\ \ \mathbf{c}^T\mathbf{x} \\
&\quad \textbf{Kısıtlayıcılar} \\
&\qquad \mathbf{Ax} = \mathbf{b} \\
&\qquad \mathbf{e}^T\mathbf{x} + x_{n+1} = Q \\
&\qquad \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0 \text{ dır.}
\end{aligned}$$

Adım 3: Kısıtlayıcıların tamsayı olan katsayıları toplamı sıfır olacak şekilde (Değişkenlerin aldığı değerler toplamı=1 kısıtlayıcısı dışında kalanlar) $x_{n+2} = 1$ dolgu değişkeninin uygun katları kısıtlayıcılara eklenir(Bazaraa ve diğerleri,1990).

$$\begin{aligned}
&Min\ \ \mathbf{c}^T\mathbf{x} \\
&\quad \textbf{Kısıtlayıcılar} \\
&\qquad \mathbf{Ax} - \mathbf{b}x_{n+2} = 0 \\
&\qquad \mathbf{e}^T\mathbf{x} + x_{n+1} - Qx_{n+2} = 0 \\
&\qquad \mathbf{e}^T\mathbf{x} + x_{n+1} + x_{n+2} = Q + 1 \\
&\qquad \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,\ x_{n+2} \geq 0 \text{ dır.}
\end{aligned}$$

Adım 4: Üçüncü adımdaki modelin son eşitlik kısıtlayıcısının sağ taraf sabitini 1 olarak elde etmek için(Karmarkar'ın standart formunda değişkenlerin aldığı değerler toplamı 1 olduğundan) $x_j = (Q+1)y_j$, $j = 1,2,...,n+2$ dönüşümü uygulanır ve model;

$$\begin{aligned}
&Min\ \ (Q+1)(\mathbf{c}^T\mathbf{y}) \\
&\quad \textbf{Kısıtlayıcılar} \\
&\qquad \mathbf{A}y - \mathbf{b}y_{n+2} = 0 \\
&\qquad \mathbf{e}^T\mathbf{y} + y_{n+1} - Qy_{n+2} = 0 \\
&\qquad \mathbf{e}^T\mathbf{y} + y_{n+1} + y_{n+2} = 1 \\
&\qquad \mathbf{y} \geq \mathbf{0},\ y_{n+1} \geq 0,\ \ y_{n+2} \geq 0
\end{aligned}$$

biçiminde yazılabilir(Fang ve Puthenpura,1993; Bazaraa ve diğerleri ,1999).

Adım 5: Karmarkar Algoritmasının ikinci varsayımını sağlamak amacıyla, yapay bir y_{n+3} değişkeni alınır ve amaç fonksiyonuna M katsayısı eklenir. Böylece n+3 değişkenli model aşağıdaki şekle dönüşür;

$$Min \ (Q+1)(\mathbf{c}^T\mathbf{y}) + My_{n+3}$$

Kısıtlayıcılar

$$\begin{aligned}
&\mathbf{A}y - \mathbf{b}y_{n+2} - [\mathbf{A}\mathbf{e} - \mathbf{b}]y_{n+3} = 0 \\
&\mathbf{e}^T\mathbf{y} + y_{n+1} - Qy_{n+2} - (n+1-Q)y_{n+3} = 0 \\
&\mathbf{e}^T\mathbf{y} + y_{n+1} + y_{n+2} + y_{n+3} = 1 \\
&y_j \geq 0, \quad j = 1,2,\ldots,n+3
\end{aligned}$$

Yukarıdaki model için $\mathbf{y} = \mathbf{e}/(n+3)$ bir başlangıç iç nokta uygun çözümdür.

Karmarkar Algoritmasının birinci varsayımının(Amaç fonksiyonunun sıfır değerini alması) sağlanması için aşağıdaki işlemler yapılır (Bazaraa ve diğerleri, 1990).

DP dualiteden hareketle, standart formdaki bir DP modeli için optimallik koşulları, $\{\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \boldsymbol{\lambda}\mathbf{A} \leq \mathbf{c}, \mathbf{cx} = \boldsymbol{\lambda}\mathbf{b}\}$ sistemine bir çözüm gerektirir. Bu kısıtlayıcılar, $\mathbf{A}'\mathbf{x}' = \mathbf{b}'$ ve $\mathbf{x}' \geq \mathbf{0}$ şeklinde standart forma dönüştürülerek, Adım 5'teki modeli(Burada amaç fonksiyonu, y_{n+3} yapay değişkeninin minimizasyonudur) elde etmek için yukarıda sözü edilen dönüşümler kullanılabilir. Eğer ilk verilen problemin optimal çözümü varsa, Adım 5'teki modelin amaç fonksiyonunun optimal değeri sıfırdır.

2.1.3.2. DP Probleminin Karmarkar Algoritması İle Çözümü

Karmarkar Algoritması'nın uygulanabileceği yapıya dönüştürülen DP probleminin Karmarkar algoritması ile çözümü, $x_0 = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$ noktası ile başlar. Birinci ve ikinci varsayımları sağlayacak ve daha küçük bir amaç fonksiyon değeri veren yeni bir nokta bulunması gerekir. Karmarkar, x_0 'dan en iyi hareket doğrultusunu bulmak için amaç fonksiyonunu kullanır: $\Omega = \{\mathbf{x} | \mathbf{AX} = \mathbf{0}\}$olmak üzere, $\mathbf{c} \in R^{n+1}$, $\Omega \cap S^n$ bölgesinde bir doğrultu vermediği için bölgede ortogonal olarak izdüşümü alınır. Böylece, algoritmanın bir iç noktadan diğer bir iç noktaya hareketi sağlanmış olur (Taha,2000;Rockett ve Stevenson,1987; Schrijver, 1998).

Simpleksin (S_x) içine merkezi x_0 olan bir küre çizilir; daha sonra $\Omega = \{\mathbf{x} | \mathbf{AX} = \mathbf{0}\}$ kümesi ile bu kürenin arakesiti, daha küçük bir küre olacaktır. Minimum ilerlemeyi sağlamak için Karmarkar algoritması, her bir adımda mümkün olduğu kadar uzak bir noktaya ilerlemeyip; yukarıda belirtilen simpleksin içine çizilen küreden daha küçük bir küre kullanır (Rockett ve Stevenson,1987; Dodani ve Babu, 1989).

Karmarkar Algoritması'nın çözüm adımları aşağıdaki gibi özetlenebilir (Karmarkar,1984;Winston,1994; Tepecik,1998);

Adım 1: Başlangıç noktası $\mathbf{x}_0 = \left(\frac{1}{n},...,\frac{1}{n}\right)$ ve yineleme sayacı k = 0 olarak alınır. Simpleksin içine çizilen en büyük kürenin yarıçapı, $r = 1/\sqrt{n(n-1)}$ ile hesaplanır. $0 < \alpha < 1$ olmak üzere bir α sabiti belirlenir(Genellikle $\alpha = \frac{1}{4}$ veya $\alpha = \frac{n-1}{3n}$ alınır)

$L = [1 + \log(1 + /c_{j\max}/) + \log(/\det_{\max}/)]$ sınır değeri belirlenir. Burada $|c_j max|$, c_j maliyet katsayısının en büyük sayısal değeridir ve $|detmax|$ ise, Karmarkar'ın standart formundaki problemin çözümünde karşılaşılan matrislerin determinantlarının en büyük sayısal değeridir.

Adım 2: $\mathbf{D}_k = diag\{x_{k1},...,x_{kn}\}$ oluşturulur.

$\mathbf{y}o = (\frac{1}{n},..., \frac{1}{n})^t$ $\quad \mathbf{P} = \begin{bmatrix} \mathbf{AD}_k \\ \mathbf{1} \end{bmatrix}$ matrisi oluşturulur.

$\bar{\mathbf{c}} = \mathbf{cD}_k$ hesaplanır.

$\mathbf{c}_p = [\mathbf{I} - \mathbf{P}^t(\mathbf{PP}^t)^{-1}\mathbf{P}]\bar{\mathbf{c}}^t$ maliyet vektörü hesaplanır. Bu vektörün uzunluğu,

$\|\mathbf{c}_p\| = \sqrt{(\mathbf{c}_{P1})^2 + ... + (\mathbf{c}_{Pn})^2}$ ile hesaplanır.

Adım 3: $\mathbf{y}_{yeni} = \mathbf{y}_0 - \alpha r \frac{\mathbf{c}_p}{\|\mathbf{c}_p\|}$ hesaplanır.

Adım 4: Ters dönüşümle, $\mathbf{x}_{k+1} = \dfrac{\mathbf{D}_k \mathbf{y}_{yeni}}{\mathbf{1D}_k \mathbf{y}_{yeni}}$ ve karşı gelen amaç fonksiyon değeri hesaplanır.

Adım 5: Eğer $\mathbf{cx}_k < 2^{-L}$ ise, optimal çözüme ulaşılmıştır. Aksi takdirde, adım 2' ye dönülür.

2.2. Mehrotra Tahminci–Düzeltici Algoritması

Son zamanlarda Karmarkar algoritmasının değiştirilmiş türleri ile karşılaşılmaya başlanmıştır. Bu algoritmalardan Mehrotra Tahminci-Düzeltici Algoritması, yazılım olanağı fazla olan etkin bir algoritmadır. 1990 dan beri pek çok iç nokta yazılımları(LIPSOL, LOQO ve PCx gibi), Mehrotra'nın Tahminci–Düzeltici algoritmasına (MPC) dayanır(Arbel, 1993;Wright,1997, Andersen ve diğerleri , 1996).

Bu algoritmanın iki anahtar özelliği, aşağıdaki gibi sıralanabilir(Wright, 1997);

a) Algoritmanın Ω çözüm grubuna daha yakında bir yörünge izlemesi amacıyla, PD arama doğrultusunda düzeltici bir adımın eklenmesi.
b) Merkezi parametre σ 'nın uygun bir biçimde seçimi.

Mehrotra algoritması, her bir adımda öncelikle affine-scaling doğrultusunu hesaplar ve eğer arama doğrultusu, (x,s) >0 koşulunu bozmadan μ'da büyük bir düşüş yaratıyorsa, algoritma, ufak bir merkezlemenin gerekliliği sonucunu çıkarır ve böylece σ_k 'yı sıfıra yakın olarak seçer ve bu küçük değerle merkezleşmiş bir arama doğrultusu hesaplanır. Eğer affine-scaling doğrultusu yararlı değilse, algoritma, σ_k 'ya 1'e daha yakın bir değer vererek daha yüksek miktarlı bir merkezleme uygular(Nocedal ve Wright,1999).

Çözüm Adımları:
Mehrotra tahminci düzeltici algoritması'nın çözüm adımları, aşağıdaki gibi sıralanabilir(Wright,1997;Nocedal ve Wright,1999);

Adım 1: $(x^0, s^0) > 0$ olmak üzere (x^0, λ^0, s^0) başlangıç nokta olarak alınır.

Adım 2: k=0,1,2,.... için $(\mathbf{x},\boldsymbol{\lambda},\mathbf{s})=(\mathbf{x}^k,\boldsymbol{\lambda}^k,\mathbf{s}^k)$ alınır ve aşağıdaki adımlar izlenir;

Adım 3:
$$\begin{bmatrix} \mathbf{0} & \mathbf{A}^{\mathrm{T}} & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x}^{aff} \\ \Delta\boldsymbol{\lambda}^{aff} \\ \Delta\mathbf{s}^{aff} \end{bmatrix}=\begin{bmatrix} -\gamma_c \\ -\gamma_b \\ -\mathbf{X}\,\mathbf{Se}+\sigma\mu\mathbf{e} \end{bmatrix}$$
sistemi, $(\Delta\mathbf{x}^{aff},\Delta\boldsymbol{\lambda}^{aff},\Delta\mathbf{s}^{aff})$ için çözülür.

Adım 4: α_{aff}^{pri} ve α_{aff}^{dual} ve μ_{aff} hesaplanır.

$$\alpha_{aff}^{pri}=\arg\max\{\alpha\in[0,1]\;\; x^k+\alpha\Delta x^{aff}\geq 0\;\}$$
$$\alpha_{aff}^{dual}=\arg\max\{\alpha\in[0,1]\;\; s^k+\alpha\Delta s^{aff}\geq 0\}$$
$$\mu_{aff}=\left(x^k+\alpha_{aff}^{pri}\Delta x^{aff}\right)^T\left(s^k+\alpha_{aff}^{dual}\Delta s^{aff}\right)/n$$

Adım 5: Merkezi parametre, $\sigma=(\mu_{aff}/\mu)^3$ alınır.

Adım 6:
$$\begin{bmatrix} 0 & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S} & 0 & \mathbf{X} \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x}^{cc} \\ \Delta\boldsymbol{\lambda}^{cc} \\ \Delta\mathbf{s}^{cc} \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ \sigma\mu\mathbf{e}-\Delta\mathbf{X}^{aff}\Delta\mathbf{S}^{aff}\mathbf{e} \end{bmatrix}$$
sistemi, (Δx^{cc} , $\Delta\lambda^{cc}$, Δs^{cc}) için çözülür. Arama doğrultusu ve adımı aşağıdaki gibi hesaplanır;

$$\left(\Delta x^k,\Delta\lambda^k,\Delta s^k\right)=\left(\Delta x^{aff},\Delta\lambda^{aff},\Delta s^{aff}\right)+\left(\Delta x^{cc},\Delta\lambda^{cc},\Delta s^{cc}\right)$$

Adım 7: $\alpha_{\max}^{pri}$ ve $\alpha_{\max}^{dual}$ aşağıdaki gibi hesaplanır;

$$\alpha_{\max}^{pri}=\arg\max\{\alpha\geq 0\;\; x^k+\alpha\Delta x^k\geq 0\}$$
$$\alpha_{\max}^{dual}=\arg\max\{\alpha\geq 0\;\; s^k+\alpha\Delta s^k\geq 0\}$$

Adım 8: $\alpha_k^{pri}=\min(0.99*\alpha_{\max}^{pri},1)$ ve $\alpha_k^{dual}=\min(0.99*\alpha_{\max}^{dual},1)$ hesaplanır;

Adım 9: $x^{k+1}=x^k+\alpha_k^{pri}\Delta x^k$ $\left(\lambda^{k+1},s^{k+1}\right)=\left(\lambda^k,s^k\right)+\alpha_k^{dual}\left(\Delta\lambda^k,\Delta s^k\right)$ alınır ve adım 2'ye gidilir.

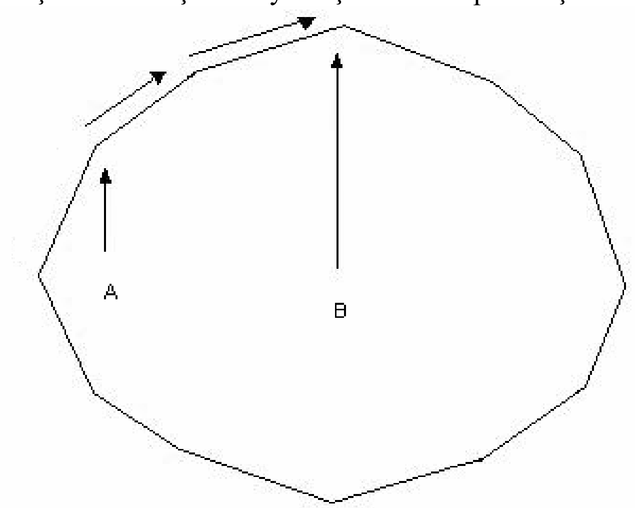
2.3. İç Nokta Algoritmalarının Simpleks Algoritması İle Karşılaştırılması

Simpleks algoritması, uygun bölge sınırlarında uzun bir yol izlemek zorunda kalabilir ve optimal çözüme ulaşmadan önce 2^n uç nokta ile karşılaşmaktadır(Murray,1989;Powel,1993;Cavalier ve Schall,1987).
Karmarkar Algoritması ise, uygun bölgenin içindeki birim simpleksin merkezinden başlayıp; yarıçapından daha büyük olmayan mesafede yinelemelerle optimum köşeye doğru gidecek iç noktaları bulmaya çalışır (Tepecik, 1998).

Simpleks algoritmasının tersine, iç nokta algoritmaları kesin bir optimal çözüm oluşturmak yerine optimal çözüme yakınsayan sınırsız bir dizi oluştururlar(Andersen ve diğerleri,1996; Andersen ve Ye, 1996; Güler ve Ye, 1993).

Şekil 2.1.'de iç nokta ve uç nokta yaklaşımları ile optimal çözüme ilerleme gösterilmeye çalışılmıştır. 1984'te yapılan araştırmalarda 5.000 değişkenli bir problemin optimal çözümü, Karmarkar Algoritması kullanıldığında Simpleks algoritmasından 50 kez daha hızlı olarak elde edilmiştir. Daha sonraki araştırmalarda da Karmarkar algoritmasının büyük ölçekli problemlerde daha az yineleme gerektiren etkin bir teknik olduğu gözlenmiştir(Lee ve diğerleri, 1990; Fang ve Puthenpura, 1993).

Şekil 2.1. İç nokta ve uç nokta yaklaşımları ile optimal çözüme ilerleme



Karmarkar Algoritması ve çoğu iç nokta algoritmaları , polinom zamanlı algoritmalardır; dolayısıyla herhangi bir DP probleminin çözümü için gereken zaman, problemin boyutunun polinom bir fonksiyonu ile sınırlanabilir. Simpleks algoritması ise, üstel zamanlı bir algoritmadır(Hillier ve Lieberman, 1995;Dodani ve Babu,1989; Knowles, 1989).

Bazı iç nokta algoritmaları, tek amaçlı ve çok amaçlı DP problemlerine uygulanabilmektedir. İç nokta algoritmalarının Simpleks algoritması karşısında etkinliğinin ve hızının, bilgisayar teknolojisindeki gelişmeler ve yapılan araştırmalar ile, daha da artacağı düşünülmektedir(Hillier ve Lieberman, 1995; Arbel,1994; Bal,1995; Jansen,1997).

3. DOĞRUSAL BİR ÜRETİM MODELİNDE İÇ NOKTA ALGORİTMALARININ UYGULANMASI

Aşağıda hipotetik bir işletmenin üretim planlamasına ilişkin bir doğrusal programlama modelinin varsayımları ve formülasyonu verilecektir. DP modeli iç nokta algoritmaları ile ve Simpleks algoritması ile çözülerek çözüm sonuçları karşılaştırılacaktır.

3.1. Modelin Formülasyonu

DP modeli kurulmadan önce aşağıdaki varsayımlarda bulunulmuştur:

1) Malzemenin temini ve taşıma konusunda herhangi bir problem bulunmadığı,
2) Planlama dönemi boyunca talep ve kapasitelerin belirli olduğu,
3) Planlama dönemi boyunca üretim hatlarının çalışır durumda oldukları, herhangi bir arızanın söz konusu olmadığı,
4) Fabrikanın stok alanı konusunda herhangi bir kısıtı bulunmadığı,
5) Malzemenin taşınmasıyla ilgili herhangi bir kısıtlayıcı bulunmadığı,
6) Ürünlerde zayiat olmadığı,
7) Ürünlerin tümünün 30 üretim hattından her birinde üretilebildikleri ve hatlar arasında farklılık bulunmadığı varsayılmıştır.

Karar modelinin bileşenleri olan karar değişkenleri, amaç fonksiyonu ve kısıtlayıcılar belirlenmiştir.

Karar Değişkenleri: Fabrikada 150 farklı ürün olup; bu ürünler 30 üretim hattında işlem görmektedir. Üretim hatlarının kapasiteleri sınırlıdır ve ürünlere olan talep karşılanmalıdır. Ürünlerin toplam üretim maliyeti minimize edilmelidir. Belirlenen kısıtlayıcılar altında toplam maliyeti minimize etmek için her bir üründen her bir süreçte üretilmesi gereken miktarlar belirlenecektir.

Karar değişkenleri;

$X_{i,j}$:j inci üründen i inci hatta üretilecek miktardır.

(i=1,2,...,30; j=1,2,...,150)

DP modelinde 4500 adet karar değişkeni bulunmaktadır.

Amaç Fonksiyonu: İşletmenin toplam üretim maliyetinin minimize edilmesidir ve aşağıdaki gibi ifade edilir;.

$$Minz = \sum_{i=1}^{30} \sum_{j=1}^{150} c_{i,j} X_{i,j}$$

Parametreler: Ürünlerin üretim hatlarındaki birim işlem süreleri, hatların kapasiteleri, ürünlerin talep miktarı ve ürünlerin birim maliyetleri parametreler olarak alınmıştır.

Kısıtlayıcılar: Modelin kısıtlayıcıları, aşağıda verilmeye çalışılmıştır.

a) Her bir ürün için belirlenen talep karşılanmalıdır. Bu grupta 150 adet kısıtlayıcı bulunmaktadır. Bu gruptaki kısıtlayıcılar kısaca aşağıdaki gibi ifade edilebilir;

$$\sum_{i=1}^{30} X_{i,j} \geq b_j \qquad j=1,2,...,150$$

b) Her bir hatta 150 adet ürünün üretimi için geçen toplam süre(saat) sınırlıdır Bu gruptaki kısıtlayıcılar 30 adettir. Bu gruptaki kısıtlayıcılar kısaca aşağıdaki gibi ifade edilebilir;

$$\sum_{j=1}^{150} t_{i,j} X_{i,j} \leq s_i$$

c) Karar değişkenleri negatif olamaz.

$$X_{i,j} \geq 0$$

Burada;

i : hat no(i=1,2,...,30)

j : ürün no(j=1,2,...,150)

$t_{i,j}$: j ürününün i üretim hattında üretilmesi için gerekli birim süre

$c_{i,j}$: j ürününün i üretim hattında üretilmesinin birim maliyeti

s_i : i.inci hatta ürünlerin üretimi için geçen toplam süreler

b_j : j.inci ürünün talebidir.

MODEL: Uygulamada oluşturulan DP modeli, kısaca aşağıdaki gibi yazılabilir;

$$Minz = 14X_{1,1} + 10X_{1,2} + 14X_{1,3} + 18X_{1,4} + 10X_{1,5} + 11X_{1,6} + 14X_{1,7}$$

$$+10X_{1,8} + 15X_{1,9} + 11X_{1,10} + ... + 14X_{30,148} + 16X_{30,149} + 13X_{30,150}$$

Kısıtlayıcılar

$X_{1,1} + X_{2,1} + X_{3,1} + ... + X_{29,1} + X_{30,1} \geq 18$

$X_{1,2} + X_{2,2} + X_{3,2} + ... + X_{29,2} + X_{30,2} \geq 11$

$X_{1,3} + X_{2,3} + X_{3,3} + ... + X_{29,3} + X_{30,3} \geq 15$

$X_{1,4} + X_{2,4} + X_{3,4} + ... + X_{29,4} + X_{30,4} \geq 12$

$X_{1,5} + X_{2,5} + X_{3,5} + ... + X_{29,5} + X_{30,5} \geq 11$

$X_{1,6} + X_{2,6} + X_{3,6} + ... + xX_{29,6} + X_{30,6} \geq 11$

$X_{1,7} + X_{2,7} + X_{3,7} + ... + X_{29,7} + X_{30,7} \geq 12$

$X_{1,8} + X_{2,8} + X_{3,8} + ... + X_{29,8} + X_{30,8} \geq 17$

$X_{1,9} + X_{2,9} + X_{3,9} + ... + X_{29,9} + X_{30,9} \geq 17$

$X_{1,10} + X_{2,10} + X_{3,10} + ... + X_{29,10} + X_{30,10} \geq 12$

$X_{1,11} + X_{2,11} + X_{3,11} + ... + X_{29,11} + X_{30,11} \geq 13$

$X_{1,12} + X_{2,12} + X_{3,12} + ... + X_{29,12} + X_{30,12} \geq 15$

$X_{1,13} + X_{2,13} + X_{3,13} + ... + X_{29,13} + X_{30,13} \geq 14$

$X_{1,14} + X_{2,14} + X_{3,14} + ... + X_{29,14} + X_{30,14} \geq 13$

$X_{1,15} + X_{2,15} + X_{3,15} + ... + X_{29,15} + X_{30,15} \geq 14$

....

$X_{1,149} + X_{2,149} + X_{3,149} + ... + X_{29,149} + X_{30,149} \geq 13$

$X_{1,150} + X_{2,150} + X_{3,150} + ... + X_{29,150} + X_{30,150} \geq 17$

b. $8X_{1,1} + 5X_{1,2} + 4X_{1,3} + ... + 6X_{1,149} + 4X_{1,150} \leq 571$

$5X_{2,1} + 5X_{2,2} + 7X_{2,3} + ... + 3X_{2,149} + 3X_{2,150} \leq 551$

$3X_{3,1} + 7X_{3,2} + 3X_{3,3} + ... + 5X_{3,149} + 5X_{3,150} \leq 506$

$8X_{4,1} + 7X_{4,2} + 8X_{4,3} + ... + 7X_{4,149} + 5X_{4,150} \leq 578$

$8X_{5,1} + 4X_{5,2} + 6X_{5,3} + ... + 7X_{5,149} + 5X_{5,150} \leq 538$

$$5X_{6,1} + 8X_{6,2} + 3X_{6,3} + ... + 5X_{6,149} + 3X_{6,150} \leq 512$$
$$5X_{7,1} + 4X_{7,2} + 7X_{7,3} + ... + 3X_{7,149} + 7X_{7,150} \leq 595$$
$$8X_{8,1} + 6X_{8,2} + 8X_{8,3} + ... + 3X_{8,149} + 4X_{8,150} \leq 575$$
$$5X_{9,1} + 4X_{9,2} + 4X_{9,3} + ... + 4X_{9,149} + 7X_{9,150} \leq 582$$
$$..........$$
$$7X_{28,1} + 3X_{28,2} + 7X_{28,3} + ... + 3X_{28,149} + 3X_{28,150} \leq 593$$
$$6X_{29,1} + 6X_{29,2} + 8X_{29,3} + ... + 5X_{29,149} + 3X_{29,150} \leq 536$$
$$3X_{30,1} + 4X_{30,2} + 3X_{30,3} + ... + 6X_{30,149} + 7X_{30,150} \leq 523$$
$$X_{1,1}, X_{1,2}, X_{1,3}, ..., X_{30,149}, X_{30,150} \geq 0$$

3.2 Modelin Çözümü

Çalışmamızda oluşturulan DP(Doğrusal Programlama) modeli, 4500 karar değişkeni ve 180 kısıtlayıcı içermektedir. Bu modelin iç nokta algoritmaları ve Simpleks algoritması ile çözülerek çözüm sonuçları karşılaştırılmıştır.

3.2.1 Modelin İç Nokta Algoritmaları İle Çözümü

Doğrusal programlama modeli, Karmarkar tekniğine dayalı bir yazılım olan MOSEK ile çözülmüş ve 11 yineleme sonunda ve 0.47 sn'de optimum çözüme ulaşılmıştır ve amaç fonksiyonu 21636 milyon p.b. olarak belirlenmiştir.

Daha sonra, doğrusal karar modeli, Mehrotra Tahminci-Düzeltici Algoritması (PCx, doğrusal programlama modelini Mehrotra tekniği ile çözen bir yazılımdır) ile çözülmeye çalışılmıştır. Model, PCx yazılımı kullanılarak çözüldüğünde, 7 yinelemede ve 0.21 sn. de çözüme ulaşılmıştır.

Son olarak DP modeli, yine Mehrotra tahminci düzeltici algoritması'na dayalı bir yazılım olan XPRESS-MP/Barrier ile çözülmeye çalışılmıştır. XPRESS-MP/Barrier, 9 yineleme sonunda optimum çözüme çok yaklaşmış; ancak işlemler 54 yinelemeye kadar sürdürülmüştür ve amaç fonksiyonunun optimum değeri 21636 milyon p.b. olarak belirlenmiştir.

3.2.2. Modelin Simpleks Algoritması İle Çözümü

XPRESS-MP/Simplex doğrusal programlama modelini Simpleks algoritması ile çözen bir yazılımdır. Model, XPRESS-MP/Simplex yazılımı kullanılarak çözüldüğünde 202 yineleme sonunda optimum çözüme ulaşılmıştır ve amaç fonksiyonunun optimum değeri 21636 milyon p.b. olarak belirlenmiştir

3.3. Çözüm Sonuçlarının Karşılaştırılması

Doğrusal programlama modeli; Karmarkar Algoritması, Mehrotra tahminci düzeltici algoritması ve Simpleks algoritması ile ayrı ayrı çözüldüğünde elde edilen çözüm sonuçları ve yineleme sayıları, Tablo 3.5 ’de toplu olarak görülmektedir.

Tablo 3.5. Modelin Üç Farklı Yazılım İle Çözüm Sonuçları

YAZILIM	AMAÇ FONK. DEĞERİ (milyon p.b.)	YİNELEME SAYISI (f)	ÇÖZÜM SÜRESİ (t-sn)	OPT.ÇÖZÜM NOKTASI (İÇ/UÇ)
MOSEK	21636	11	0.47	Uç nokta
XPRESS-MP/ Barrier	21636	54	2	Uç nokta
XPRESS-MP/Simplex	21636	202	2	Uç nokta
PCx	21636	7	0.21	İç nokta

Tablo 3.6’da yazılımların yineleme sayılarının ve çözüm sürelerinin oranları görülmektedir. XPRESS-MP/Simplex’in yineleme sayısının PCx’in 28.8 katı olması, MOSEK’in yineleme sayısının ise XPRESS-MP/Simplex’in yineleme sayısının yaklaşık 18 katı olması özellikle dikkat çekicidir.

Tablo 3.6. Yazılımların Yineleme Sayılarının ve Çözüm Sürelerinin Oranları

Yazılımların yineleme sayılarına ilişkin oranlar	$f_{XMPSimlex}/f_{Mosek}$	202/11 = 18.363
	$f_{XMPSimlex}/f_{PCx}$	202/7 = 28.857
	$f_{XMPSimlex}/f_{XMPBarrier}$	202/54 = 3.741
Yazılımların Çözüm sürelerine	$t_{XMPSimlex}/t_{Mosek}$	2/0.47 = 4.255

ilişkin oranlar (Sn.)	$t_{XMPSimlex}/t_{PCx}$	2/0.21 = 9.524

Tablo 3.5 ve Tablo 3.6'ya göre 4500 karar değişkeni ve 180 kısıtlayıcı içermekte olan doğrusal programlama modelimizin çözümünde iç nokta algoritmalarının, Simpleks algoritmasından yineleme sayısı itibarı ile daha etkin olduğu sonucuna varılmıştır. Burada, problemin değişken ve kısıtlayıcılarının sayısı, önemli rol oynamaktadır.

SONUÇ

Uygulamada elde edilen sonuçlara göre, problemin değişken ve kısıtlayıcı sayısı fazla ise, iç nokta algoritmalarının Simpleks algoritmasından daha kısa sürede çözüme ulaştığı görülmektedir. Bunun yanında aşırı büyük modellerde Simpleks algoritmasının uygun çözüm bulamadığı durumlarda iç nokta algoritmaları uygun çözüme ulaşabilmektedir. DP test problemleri ile yapılan çeşitli çalışmalarda, Simpleks algoritması ile iç nokta algoritmaları karşılaştırılmıştır. Bu çalışmalarda da özellikle büyük problemlerde iç nokta algoritmalarının gerek yineleme sayısı ve gerekse çözüm süresi yönünden etkinliği gösterilmiştir.

KAYNAKÇA

Kitaplar

Arbel, A.(1993). *Exploring Interior – Point Linear Programming* (Algorithms and Software). Hong Kong: Times Roman by Asco Trade Typesetting Ltd.

Bazaraa, M. S., Jarvis, J.J., Sherali, H.D.(1990).*Linear Programming and Network Flows.* USA: Second Edition, John Wiley&Sons, Inc.

Chong, E. K. P., Zak, S. H.(1996).*An Introduction to Optimization*, USA: John Wiley & Sons, Inc.

Fang, S., Puthenpura, S.(1993).*Linear Optimization and Extensions*, Theory and Algorithms. USA: Prentice –Hall,Inc.

Hertog, D.den(1994). *Interior Point Approach to Linear, Quadratic and Convex Programming.* Netherlands: Kluwer Academic Publishers.

Hillier, F.S., Lieberman, G.J.(1995). *Introduction to Operations Research*. , Singapore: Mc Graw-Hill,Inc., Sixth Edition.

Knowles, T. W.(1989).*Management Science Building and Using Models.,* USA:Richard D.Irwin, Inc.

Kojima, M., Megiddo, N., Noma, T., Yoshise, A.(1991).*A Unified Approach To Interior Point Algoirthms for Linear Complementarity Problems*. Germany: Springer Verlag.

Moon, T.K., Stirling, W.C.(2000).*Mathematical Methods and Algorithms for Signal Processing*. USA: Prentice-Hall,Inc.

Nash, G., Sofer, A.(1996).*Linear and Nonlinear Programmming*.Singapore: McGraw-Hill Companies,Inc.,.

Nocedal, J., Wright, S.(1999).*Numerical Optimization*. Springer Series in Operations Research, New York : Springer.

Rardin, R.L.(1998).*Optimization in Operations Research*. New Jersey: Prentice Hall,Inc.

Render, B., Stair, R.M.(1997).*Quantitative Analysis for Management.* New Jersey : Prentice-Hall,Inc., Sixth Edition.

Roos, C., Terlaky, T., Vial, J. Ph.(1997).*Theory and Algorithms for Linear Optimization*. An Interior Point Approach, England: John Wiley & Sons Ltd.

Schrijver, A.(1998).*Linear and Integer Programming*. England:John Wiley&Sons Ltd.

Taha, H.(2000).*Yöneylem Araştırması*. 6. Basımdan Çeviri, Çev:Ş.Alp Baray, Şakir Esnaf, İstanbul: Literatür Yayıncılık.

Winston, W. L.(1994*). Operations Research Applications and Algorithms. –* Belmont, Calif. : Duxbury Press.

Wright, S. J.(1997).*Primal - Dual Interior Points Methods*. The Society for Industrial and Applied Mathematics, USA:Siam.

Makaleler

Andersen, E. D., Gondzio, J., Meszaros, C., Xu, X.(1996). "Implementation of Interior Point Methods For Large Scale Linear Programs", Interior Point Methods of Mathematical Programming**,** Netherlands: *Kluwer Academic Publishers*.

Andersen, E. D., Ye, Y.(1996). "Combining Interior-point and Pivoting Algorithms for Linear Programming", Management Science, Vol. 42.,No.12, Institute for Operations Research and Management Sciences.

Arbel, A.(1994)."A Multiobjective Interior Primal – Dual Linear Programming Algorithm" *Computers Operations Research*, Vol. 21, No. 4, Great Britain: Elsevier Science Ltd., s. 433 – 445.

Bal, H.(1995). "Using of Karmarkar Algorithm in Linear Goal Programming and an application*", Gazi Üniversitesi Fen Bilimleri Ens. Der.* C.8., S.1.

Bland, R.G., Goldfarb, D., Todd M.(1981).“The Ellipsoid Method : A Survey”, *Operations Research*, Vol. 29, No.6, Operations Research Society of America.

Cavalier, T. M., C. Schall, K.(1987). “Implementing An Affine Scaling Algorithm For Linear Programming”, *Comput. Operations Research,* Vol.14, No.5, Great Britain :Pergamon Journals Ltd.

Dodani, M. H., Babu, A. J. G.(1989).“Karmarkar”s Projective Method For Linear Programming : A Computational Appraisal*”, Computers Industrial Engineering,* Vol. 16, No.1, ., Great Britain: Pergamon Press plc.

Gay, D. M.(1987).”A Variant of Karmarkar”s Linear Programming Algorithm For Problems In Standart Form*”, Mathematical Programming* 37,North Holland.

Goldfarb, D., Todd, M.J.(1989). “Linear Programming”, *Handbooks in Operations Research and Management Science*, Volume 1, Elsevier Science Publishers B.V., Netherlands.

Gonzaga, C. C.(1991). “ Large Step Path – Following Methods For Linear Programming, Part I : Barrier Function Method ”, *SIAM Journal on Optinization*, Vol.1, No.2, Society for Industrial and Applied Mathematics.

Gonzaga, C. C.(1991). “ Large Step Path – Following Methods For Linear Programming, Part II : Potential Reduction Method *”, SIAM Journal on Optinization,* Vol.1, No.2, SIAM.

Güler, O., Ye, Y.(1993). “Convergence Behavior of Interior-point Algorithms”, *Mathematical Programming* 60, North-Holland.

Hertog, D.Den, Roos, C.(1991). “A Survey of search directions in interior point methods for linear programming”, *Mathematical Programming* 52, North-Holland.

Karmarkar, N.(1984). “ A New Polynomial Time Algorithm For Linear Programming”, *Combinatorica*, Vol.4.

Murray, Walter(1989).“Methods For Large – Scale Linear Programming”, *Algorithms and Model Formulations in Mathematical Programming* Ed: Stein W. Wallace, Berlin : Springer – Verlag.

Powell, M.J.D.(1993).“On The Number of Karmarkar”s Algorithm for Linear Programming”, *Mathematical Programming* 62, North-Holland.

Rockett, A. M., Stevenson J.C.(1987). “Karmarkar Algorithm”, *Byte,* September.

Tardos, E.(1986).“A Strongly Polynomial Algorithm To Solve Combinatorial Linear Programs”, *Operations Research*, Vol.34, No.2, Operations Research Society of America.

Tepecik, A.(1998).”Geliştirilmiş Karmarkar Algoritması”,*Gazi Ünv. Fen Bil.Ens.Der.*, Cilt.11, No.4,Ankara.

Ye, Y., Kojıma, M.(1987).“Recovering Optimal Dual Solutions in Karmarkar”s Polynomial Algorithm For Linear Programming”, *Mathematical Programming* 39, North-Holland.