İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

LİNEER PROGRAMLAR VE BİLGİSAYAR ÇÖZÜMLERİ

N. Beylem SİNOPLU

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman: Yrd.Doç.Dr. Nuri ÖZALP

Bu tez üç bölümden oluşmaktadır.

Birinci bölümde, bu tez çalışmasının seçimindeki temel motivasyon ortaya konulmuş olup, lineer programlamanın matematiksel yapısı için gerekli temel tanım ve kavramlar tanıtılmıştır.

İkinci bölümde, lineer programlama problemlerinin en güçlü çözüm tekniklerinden biri olan simpleks algoritması için gerekli teoremler ispatlanmış olup, bir örnek problem üzerinde bu algoritma incelenmiştir. Bu bölümde, ayrıca maksimum, minimum ve dual problemler arasındaki ilişkiler verilmiştir.

Üçüncü bölümde ise, lineer programlamaların gerçek hayat problemlerine uygulamaları dört temel problemle analiz edilmiş olup, bölüm sonunda ise bilgisayar çözümleri için gerekli bir kaynak kod günümüzdeki en güçlü ve yaygın kullanılan dillerden biri olan C++ programla diliyle verilmiştir.

2003, 70 sayfa

ANAHTAR KELİMELER : Lineer programlama, konveks kümeler, uygun küme, uygun çözüm, kısıt, maksimum, minimum, optimal çözüm, dual problem, Simpleks algoritması, Gauss eliminasyonu.

ABSTRACT

Master Thesis

LINEAR PROGRAMMING AND COMPUTER SOLUTIONS

N. Beylem SİNOPLU

Ankara University
Graduate School of Natural and Applied Sciences
Department of Mathematics

Supervisor: Asst. Prof. Dr. Nuri ÖZALP

This thesis consists of three chapters.

In the first chapter, the basic motivation in selection of the subject matter of the thesis is explained, and fundamentals definitions and concepts needed for the mathematical background of linear programming are given.

In the second chapter, by proving some important theorems for the simplex method which is one of the powerfull techniques for solving linear programming problems, the simplex algorithm is explained in details on a clear example. In addition, relations between maximum, minimum and dual problems are given.

Finally, in the third chapter, applications of linear programming to real life problems are analysed by using for fundamental example of the subject matter. At the end of the chapter, a source code for computer solutions are given in C++ language which is one of the powerfull and most popular languages.

2003, 70 pages

Key Words: Linear programming, convex sets, feasible set, feasible solution, constraints, maximum, minimum, optimal solution, dual problem, simplex method, Gauss elimination.

TEŞEKKÜR

Bu çalışmayı bana vererek, çalışmalarım sırasında benden yardımlarını ve bilgisini esirgemeyen, ilgi ve önerileri ile yanımda olan danışman hocam, Sayın Yrd. Doc. Dr. Nuri ÖZALP 'e sonsuz teşekkürlerimi sunmayı bir borç bilirim.

N. Beylem SİNOPLU

Ankara, Temmuz 2003

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGE DİZİNİ

1. GİRİŞ

1.1. Motivasyon

Son yıllarda, matematik ve mühendislik, fiziksel veya organizasyonel sistemlerde karar verme problemleri üzerinde önem arz etmektedir. Bunun temel nedeni, pahalı kaynakların dağılımında uygun bir karar verme gerekliliğinden ortaya çıkan belirgin ekonomik kazançların oluşmasında, ve iyi karar vermede, bu tür problemlerin gerçekçi matematiksel formülasyonunun yapılabilmesidir.

Yüksek hızlara sahip digital bilgisayarın geliştirilmesi, karar verme biliminin gelişmesinde ayrıca önemli bir rol oynamış olup, aynı zamanda uygulamalı matematikte de bir devrim yaratmıştır. Karar verme problemlerinin formülasyonu için temel yaklaşımda, en iyi ya da optimal karar kavramının oluşması doğaldır. Bu tür bir yaklaşımda, performans veya bir kararın değerini özetleyen tek bir reel nicelik, değişik alternatifler içinden yalıtılarak, duruma göre ya minimize ya da maksimize yani optimize edilir. Ortaya çıkan optimal karar, karar verme probleminin çözümü olarak alınır. Aşağıdaki problem bu türden bir probleme örnektir.

Örnek 1.1.1. Bir üreticinin ham madde envanterini göz önüne alalım. Üretici, ham maddelerden n farklı tip ürün elde edebilen üretici araca sahip olsun. Üreticinin problemi, kazancını maksimum yapacak şekilde, ham maddelerini olası ürünlerine göre ayarlamasıdır.

Problemin ideal şekliyle, üretim ve kazanç modelinin lineer olduğunu kabul edelim. Her bir j ürününün birim satış fiatı p_j, $j = 1, 2, ..., n$ olsun. Eğer x_j, j ürününün üretilecek miktarını, b_i, halihazırdaki i ham maddesinin miktarını, ve a_{ij}, bir birim j ürünündeki i maddesinin miktarını gösterirse bu durumda üretici, ham maddelerin miktarı üzerindeki üretim kısıtları

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\le b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\le b_2 \\
\vdots \qquad\qquad\qquad &\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\le b_m
\end{aligned}
$$

ve

$$x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0$$

üzerinden

$$p_1x_1 + p_2x_2 + \ldots + p_nx_n$$

kazancını maksimum yapmak istemektedir.

Lineer eşitsizlik kısıtları altında bir lineer objektif fonksiyonu ile karakterize edilen yukarıdaki tipten bir problem, bir *lineer (doğrusal) programlama* problemi olarak adlandırılır. Lineer programlama terimi bilgisayar programlamasından ziyade, iş ve ekonomik yatırımların programlanması anlamına gelmektedir.

1.2. Temel Tanım ve Kavramlar

Lineer programlama optimizasyon problemlerini çözmek için bir araçtır. 1947 'de George Dantzig [1] doğrusal programlama problemlerini çözmek için Simpleks algoritması adında etkin bir yöntem geliştirmiştir.

Lineer programlama en yaygın olarak sınırlı miktardaki kaynakların çeşitli faaliyetlere optimal şekilde paylaştırılmasına yönelik problemlerde kullanılır. Bu kapsamda lineer programlama başta endüstri, ulaştırma, eğitim, ekonomi, bankacılık, askeri faaliyetler, tarım ve sağlık olmak üzere pek çok alanda uygulama imkanı bulan bir yaklaşım olarak karşımıza çıkmaktadır. Bu bölümde lineer programlamanın genel tanımlamasında kullanılan temel kavramlar tanıtılacaktır.

1.2.1. Konvekslik ve lineer eşitsizlikler

Lineer eşitsizlikler teorisi, iyi bilinen lineer eşitlikler teorilerine paraleldir. Burada konunun temel kısımlarını geliştirip, bazı uygulamalarına değineceğiz.

Teori, doğal olarak temelde sıralı reel doğru yapısı içerdiğinden, tezde kullanılan tüm vektörler ve matrisleri reel değerli olarak kabul edeceğiz. $\mathbb{R}$, reel sayılar kümesini göstermek üzere, $\mathbb{R}^n$ deki herhangi iki x ve y noktaları (vektörleri) için, iki vektörün eşitsizliğini

$$x \geq y \text{ olması için gerek ve yeter koşul } x_i \geq y_i \quad (1 \leq i \leq n)$$

şeklinde ve benzer olarak $x \leq y, x > y$, veya $x < y$ eşitsizliklerini bileşen bazında tanımlayalım. $x > y$ nin ($x \geq y$ ve $x \neq y$) ile aynı anlama gelmediğini özellikle belirtelim. (Bunun nedeni $x \neq y$ nin her i için $x_i \neq y_i$ anlamına gelmeyip, sadece bazı i ler için $x_i \neq y_i$ anlamına gelmesidir.)

Bir n değişkenli m zayıf lineer eşitsizlik sistemi; A nın $m \times n$, x in $n \times 1$ ve b nin de $m \times 1$ boyutlu olduğu

$$Ax \geq b \tag{1.1}$$

şeklinde bir eşitsizlik olarak yazılabilir. Bu sistemdeki her bir bireysel eşitsizlik;

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \geq b_i \qquad (1 \leq i \leq m)$$

veya $A_{satır\ i}$, A matrisinin i yinci satır vektörünü göstermek üzere

$$A_{satır\ i}x \geq b_i$$

şeklindedir. Böyle bir sistemi ilgilendiren en temel soru sistemin ***tutarlı*** olup olmamasıdır. Diğer bir ifadeyle $Ax \geq b$ olacak şekilde bir x var mıdır? Eğer sistem tutarlı ise bu durumda çözümleri bulabilmek için algoritmalar arayabiliriz. Genel olarak,

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \geq b_i & (1 \leq i \leq m_1) \\ \sum_{j=1}^{n} a_{ij} x_j > b_i & (m_1 + 1 \leq i \leq m) \end{cases}$$

formunda daha genel tipten lineer eşitsizlik sistemlerini incelemek mümkün olmakla birlikte, basitlik için şimdilik sadece (1.1) tipindeki sistemleri göz önüne alalım. Ne olacağını görmek için öncelikle aşağıdaki basit sistemi inceleyelim.

$$\begin{cases} x_1 + x_2 \geq 2 \\ x_1 - x_2 \geq -1 \\ -3x_1 + x_2 \geq -6 \end{cases} \tag{1.2}$$

$n = 2$ ve $m = 3$ olmak üzere bu sistem matris formunda

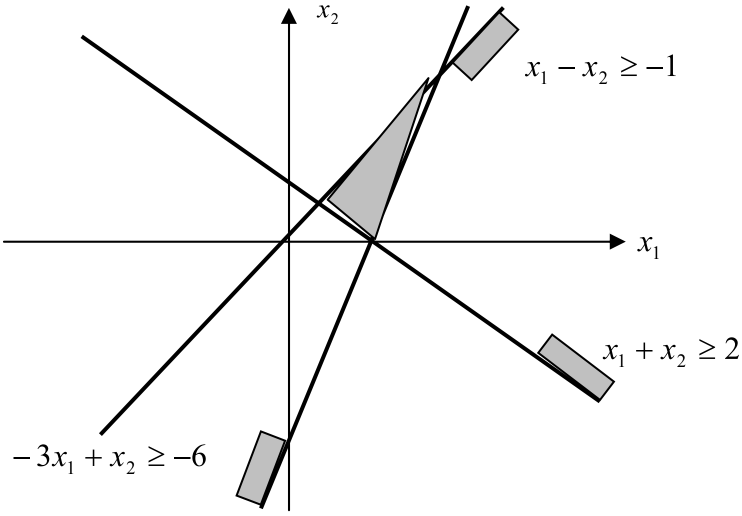
$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 2 \\ -1 \\ -6 \end{bmatrix}$$

şeklinde yazılabilir. $x_1 + x_2 \geq 2$ eşitsizliğini sağlayan noktalar, $x_1 + x_2 = 2$ doğrusunun bir tarafında bulunmaktadır. Bu doğru ile birlikte, Sistem (1.2) den elde edilen diğer iki doğrunun grafiğinin çizilmesi ile, bu sistemin çözümü olan bütün noktaların kümesine karar verebiliriz. Bu küme Şekil 1.1 de gösterilen üçgensel bölgedir. Eğer, Sistem (1.2) deki eşitsizliklerin hepsini tersine çevirirsek, Şekil 1.1 den, *tutarlı olmayan* bir sonuç elde edeceğimizi kolayca görebiliriz. Aynı zamanda, katsayılar matrisinde yapılan ufak değişikliklerle çözüm kümesi *sınırsız* olan bir sistem de üretebiliriz. (Örneğin, üçüncü eşitsizlikte -3 yerine -1 alalım.) Grafik, ayrıca çözüm kümesinin aşağıda tanımlayacağımız bir konveks küme olduğunu göstermektedir.

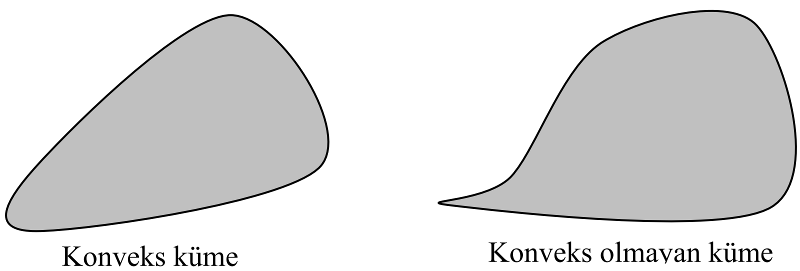
Tanım 1.2.1. K, bir lineer (vektör) uzayın bir alt kümesi olsun. Eğer, K daki herhangi iki noktayı birleştiren doğru parçası tümüyle K nın içinde kalıyor ise K ya bir ***konveks*** küme denir. Matematiksel olarak, eğer

$$\left.\begin{array}{l} x, y \in K \\ 0 \le \theta \le 1 \end{array}\right\} \Rightarrow \theta x + (1-\theta) y \in K$$

oluyorsa, K ya konveks denir. Şekil 1.2, $\mathbb{R}^2$ de konveks ve konveks olmayan kümelere birer örnek göstermektedir.



Şekil 1.1. Sistem (1.2) nin çözüm kümesi



Şekil 1.2. Konveks olmayan küme

Eğer $\sum_{i=1}^{k} \theta_i = 1$ ve $\theta_i \ge 0$ olması durumunda, vektörlerin $\sum_{i=1}^{k} \theta_i x^{(i)}$ formundaki bir lineer birleşimine bir ***konveks birleşim*** *(kombinasyon)* denir. Verilen bir S kümesinden seçilen noktaların tüm konveks birleşimlerinin kümesi S nin ***konveks kabuğu*** olarak adlandırılır ve $co(S)$ ile gösterilir. Böylece,

$$co(S)=\left\{\sum_{i=1}^{k}\theta_i x^{(i)}\ :k\in\mathbb{N},\ x^{(i)}\in S,\ \theta_i\geq 0,\ \sum_{i=1}^{k}\theta_i=1\right\}$$

kümesidir.

Konveks kümelerle ilgili bazı temel gerçekler aşağıdaki sonuçlarda belirtilmiştir.

Lemma 1.2.1. K, $\mathbb{R}^n$ de (veya herhangi bir Hilbert uzayında) bir kapalı konveks küme olsun. Bu durumda K minimum norma sahip tek bir nokta içerir. Dahası aşağıdaki özellikler denktir:

(i) $0\notin K$

(ii) Her $u\in K$ için $\langle v,u\rangle>0$ olacak şekilde bir v vektörü vardır.

Lemma 1.2.2. (Caratheodory teoremi) S n-boyutlu bir lineer uzayın bir alt kümesi olsun. S nin konveks örtüsünün her noktası S nin en fazla $n+1$ noktasının bir konveks birleşimidir.

Lemma 1.2.3. Sonlu boyutlu bir normlu lineer uzayda bir kompakt kümenin konveks örtüsü kompaktır.

Lemma 1.2.4. (Lineer eşitsizlikler teoremi) $\mathbb{R}^n$ deki bir kompakt K kümesi için aşağıdakiler denktir.

(i) Her $u\in K$ için $\langle v,u\rangle>0$ olacak şekilde bir v vektörü vardır.

(ii) 0, K nın konveks örtüsüne ait değildir.

Teorem 1.2.1. Eğer K konveks bir küme ise o zaman K daki noktaların herhangi bir konveks birleşimi yine K ya aittir.

İspat. Teoremin ispatını tümevarım yöntemiyle verelim. $m = 1$ için durum aşikar olup, $m = 2$ nin doğruluğu ise konveksliğin tanımından dolayı açıktır. m durumunu m -1 durumundan ispatlamak için, $x^{(i)}$ lerin K ya ait noktalar olmak üzere

$$\sum_{i=1}^{m} \lambda_i x^{(i)} = \lambda_m x^{(m)} + (1-\lambda_m) \sum_{i=1}^{m-1} \frac{\lambda_i}{1-\lambda_m} x^{(i)}$$

olur. Burada $0 \le \lambda_m \le 1$ ve $1-\lambda_m = \sum_{i=1}^{m-1} \lambda_i$ olup, katsayılar $\lambda_i / (1-\lambda_m)$ negatif değildir ve toplamları 1'dir.

Teorem 1.2.2. Konveks kümelerin bir ailesinin kesişimi yine bir konveks kümedir.

İspat. İndeks kümesindeki her α için K_α nın konveks küme olduğunu kabul edelim. Eğer $x, y \in \bigcap K_\alpha$ ve $0 \le \theta \le 1$ ise, bu durumda K_α konveks olduğundan her α için $\theta x + (1-\theta) y \in K_\alpha$ dır. Böylece, $\theta x + (1-\theta) y \in \bigcap K_\alpha$ dır.

Şimdi, Sistem (1.2) nin çözüm kümesi ile ilgili bazı teoremler verelim.

Teorem 1.2.3. Bir lineer eşitsizlikler sisteminin çözüm kümesi konveks bir kümedir.

İspat. $a^T x \ge \beta$ şeklindeki tek bir lineer eşitsizliğin çözüm kümesi **;**

$$a^T (\theta x + (1-\theta) y) = \theta a^T x + (1-\theta) a^T y \ge \theta \beta + (1-\theta) \beta = \beta$$

olduğundan dolayı konvekstir. Burada x ve y ler çözüm kümesinin elemanlarıdır. Bir lineer eşitsizlikler sisteminin çözüm kümesi bu sistemi meydana getiren lineer eşitsizliklerin çözüm kümelerinin kesişimidir. Böylece, konveks kümelerin herhangi bir ailesinin kesişimi de konveks olduğundan dolayı bu küme de konvekstir.

Teorem 1.2.4. S kümesinin konveks kabuğu S yi içeren en küçük konveks kümedir.

İspat. K kümesi S'nin konveks kabuğu ve T de S'yi içeren herhangi bir başka konveks küme olsun. $K \subseteq T$ olduğunu ispatlamalıyız. K'nın herhangi bir elemanı, $x_i \in S, \lambda_i \geq 0$ ve $\sum_{i=1}^{n} \lambda_i = 1$ olmak üzere, $x = \sum_{i=1}^{n} \lambda_i x_i$ formundadır. T S'yi içerdiğinden $x_i \in T$ olduğu açıktır. T konveks olduğundan $x \in T$ dir. x K'nın keyfi bir elemanı olduğundan $K \subseteq T$ dir.

Teorem 1.2.5. (Ayırma Teoremi I) X R^n de kapalı konveks bir küme olsun. Eğer p X'e ait olmayan bir nokta ise bu durumda, bazı $v \neq 0$ için

$$\langle v, p \rangle < \inf_{x \in X} \langle v, x \rangle$$

dir.

İspat. S, p merkezli ve X kümesi ile kesişebilecek büyüklükte yarıçapa sahip kapalı bir küme olsun. Bu durumda $S \cap X$ kompakttır. $x \mapsto \|x - p\|$ sürekli bir fonksiyon olduğundan minimum değerini $S \cap X$ kümesinde bir ξ noktasında alır. Eğer $x \in X$ ve $0 < \theta < 1$ ise $y \equiv \theta x + (1-\theta)\xi \in K$ olup, dolayısıyla $\|y - p\| \geq \|\xi - p\|$ olduğu kolayca görülür. Böylece,

$$\begin{aligned} \|\xi - p\|^2 &\leq \|\theta x + (1-\theta)\xi - p\|^2 \\ &= \|\xi - p + \theta(x - \xi)\|^2 \\ &= \|\xi - p\|^2 + 2\theta\langle \xi - p, x - \xi \rangle + \theta^2 \|x - \xi\|^2 \end{aligned}$$

olup, buradan

$$0 \leq 2\langle \xi - p, x - \xi \rangle + \theta \|x - \xi\|^2$$

bulunur. Bu ise, $\theta \to 0$ için

$$0 \leq \langle \xi - p, x - \xi \rangle$$

demektir. $v = \xi - p$ dersek, son eşitsizliği

$$0 \le \langle v, x - p + p - \xi \rangle = \langle v, x - p - v \rangle = \langle v, x - p \rangle - \|v\|^2$$

şeklinde yazabiliriz. Böylece iddia edilenden daha güçlü olan

$$0 < \|v\|^2 \le \langle v, x \rangle - \langle v, p \rangle$$

eşitsizliğini elde etmiş oluruz.

Ayırma teoreminin uygun versiyonları daha genel uzaylar için de geçerlidir. Şimdi bir geometrik uygulamasından bahsedelim. R^n de bir *kapalı yarı-uzay,* $a \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ ve $a \neq 0$ olmak üzere

$$\{x : \langle a, x \rangle \ge \lambda\}$$

formundaki bir küme olarak tanımlanır.

Teorem 1.2.6. $\mathbb{R}^n$ deki her kapalı konveks küme, bu kümeyi içeren bütün kapalı yarı-uzayların kesişimidir.

İspat. X in kapalı ve konveks bir küme olduğunu kabul edelim. X in kendisini içeren kapalı yarı uzayların kesişimi tarafından kapsandığı açıktır. Tersi için, p X'e ait olmayan bir nokta olsun. Ayırma teoreminden,

$$\langle v, p \rangle < \inf_{x \in X} \langle v, x \rangle$$

olacak şekilde bir $v \neq 0$ vektörü vardır.

$$\lambda = \inf_{x \in X} \langle v, x \rangle$$

dersek $X \subseteq \{x : \langle v, x\rangle \geq \lambda\}$ olur ki, bu yarı-uzay p 'yi içermez.

Teorem 1.2.7. (Ayırma Teoremi II) ($\mathbb{R}^n$ de) X kapalı konveks bir küme ve Y de kompakt konveks bir küme olsun. Eğer X ve Y ayrık ise bu durumda

$$\inf_{x\in X}\langle v, x\rangle > \sup_{y\in Y}\langle v, y\rangle$$

olacak şekilde bir $v \in \mathbb{R}^n$ mevcuttur.

İspat. İlk olarak $Z = X - Y$ kümesinin kapalı olduğunu gösterelim. $z_k \in Z$ olmak üzere $z_k \to z$ olsun. Acaba $z \in Z$ midir? $x_k \in X$ ve $y_k \in Y$ olmak üzere $z_k = x_k - y_k$ alalım. Y 'nin kompaktlığından dolayı $y_{k_i} \to y \in Y$ olacak şekilde yakınsak bir alt dizi vardır. Böylece $z_{k_i} \to z$ ve $x_{k_i} \to z + y$ dir. X kapalı olduğunda $z + y \in X$ dir. O halde $z \in X - Y$ dir. $0 = x - y$ ($x \in X, y \in Y$) eşitliği X ve Y nin ortak bir nokta içerdiğini ifade edeceğinden, $0 \notin X - Y$ dir. Z nin konveks olduğunu göstermek kolaydır. Böylece Ayırma Teoremi I i Z 'ye uygularsak,

$$\langle v, 0\rangle < \inf_{x\in X, y\in Y}\langle v, x - y\rangle$$

olacak şeklinde bir v vektörünün var olduğu sonucuna varabiliriz. Eğer eşitsizlikteki infimumu ε ile gösterirsek, $x \in X$ ve $y \in Y$ için $\langle v, x - y\rangle \geq \varepsilon$ olur ki buradan da istenen eşitsizlik elde edilir.

1.3. Ekstrem Noktalar

$0 < \theta < 1$, $y \in K$, $z \in K$ ve $y \neq z$ olmak üzere $x = \theta y + (1-\theta)z$ formunda yazılamayan bir $x \in K$ noktasına K konveks kümesinin bir ***ekstrem noktası*** denir. Diğer bir ifadeyle, bir ekstrem nokta, K ya ait olan herhangi bir doğru parçasının bir *iç noktası* olmayan bir noktadır. Denk bir tanım; eğer $K \setminus \{x\}$ konveks ise, x K konveks kümesinin bir ekstrem noktasıdır. Örneğin, bir küpün ekstrem noktaları sadece köşe noktalarıdır. Bir kürenin ekstrem noktaları ise kürenin tüm sınır noktalarıdır.

Aşağıdaki teorem Krein-Milman Teoreminin sonlu boyutlu versiyonudur. (Royden [1968])

Teorem 1.3.1. n-boyutlu uzaydaki her kompakt konveks küme ekstrem noktalarının kümesinin konveks kabuğunun kapanışıdır.

İspat. Bu teoremin ispatını tümevarımla yapacağız. Eğer $n=1$ ise konveks küme kapalı ve sınırlı bir aralıktır. Böylece ekstrem noktalar aralığın iki uç noktası olup, bu durumda teoremin doğru olduğu açıktır. Şimdi n boyuttan az olan durumlar için teoremin doğru olduğunu ve K nın n-boyutlu uzayda kompakt konveks bir küme olduğunu kabul edelim. E, K'nın ekstrem noktalarının kümesini H'da E'nin konveks kabuğunu göstersin. $\overline{H}=K$ olduğu ispatlanmalıdır.(Burada $\overline{H}$, H'nın kapanışını temsil etmektedir.) K konveks ve $E \subseteq K$ olduğunda $H \subseteq K$ elde edilir. K kapalı olduğunda $\overline{H} \subseteq K$ olur. Şimdi, $p \in K / \overline{H}$ kabul edelim.($x \mapsto x-p$) dönüşümü ile $p=0$ olduğunu kabul edebiliriz. $0 \notin \overline{H}$ olduğunda Lemma 1.2.4 den her $u \in \overline{H}$ için $\langle v,u \rangle > 0$ olacak şekilde bir v vektörü vardır.

$$c=\sup\{\langle v,x \rangle : x \in K\}$$

diyelim. $0 \in K$ olduğunda $c \geq 0$ dır. K kompakt olduğunda bu supremum vardır. Yani,

$$K' = \{x \in K : \langle v,x \rangle = c\}$$

kümesi boş değildir. K' kümesi de bir hiperdüzlem üzerinde yattığından; kompakt, konveks ve $n-1$ boyutludur. Tümevarım hipotezinden dolayı K', en azından bir z ekstrem noktasını içerir. Şimdi z nin aslında K'nın bir ekstrem noktası olduğunu ispatlayalım. $0<\theta<1$ ve $z_i \in K$ olmak üzere $z=\theta z_1+(1-\theta)z_2$ olsun. Buradan

$$c=\langle v,z \rangle = \theta\langle v,z_1 \rangle + (1-\theta)\langle v,z_2 \rangle \leq \theta c + (1-\theta)c = c$$

olup, böylece $\langle v, z_1 \rangle = \langle v, z_2 \rangle = c$ ve $z_i \in K^{/}$ dir. Fakat z, $K^{/}$nün bir ekstrem noktası olduğundan $z_1 = z_2$ dir. Bu da $z \in E$ olduğunu ispatlar. Böylece $\langle v, z \rangle < 0 \leq c$ olur ki bu da çelişkidir.

Optimizasyon problemlerinde ekstrem noktaların önemi, bir kompakt konveks küme üzerinde bir lineer fonksiyonun minimumunu bulmak için ekstrem noktaları araştırma gerekliliğinden çıkmaktadır. Aşağıdaki sonuç bununla ilgilidir.

Teorem 1.3.2. K, $\mathbb{R}^n$ de bir kompakt, konveks küme ve f de $\mathbb{R}^n$de bir lineer fonksiyonel (yani $\mathbb{R}^n$ den $\mathbb{R}$ ye bir lineer dönüşüm) olsun. f fonksiyonelinin K daki maksimum ve minimum değerleri K nın ekstrem noktalarıdır.

İspat.

$$c = \sup\{f(x) : x \in K\}$$

olsun. f fonksiyoneli sürekli ve K da kompakt olduğundan

$$K^{/} = \{x \in K : f(x) = c\}$$

kümesi boş küme değildir. Aynı zamanda bu küme konveks ve kompakttır. O halde, Krein-Milman teoreminden $K^{/}$, z gibi bir ekstrem noktaya sahiptir. Krein-Milman Teoreminin ispatındakine benzer şekilde, z K nın bir ekstrem noktasıdır. K daki f fonksiyonunun minimumunun ispatı, $-f$ fonksiyonunun maksimumunun gözönünde bulundurulması ile elde edilir.

1.4. Lineer Eşitsizlikler

X reel cisim üzerinde bir vektör uzayı olsun. Bir ***lineer fonksiyonel*** X den $\mathbb{R}^n$ ye bir lineer dönüşümdür. Bir ***lineer eşitsizlik*** ise f nin lineer bir fonksiyonel olduğu $f(x) \geq \alpha$ eşitsizliğidir. Bu tipten tek bir eşitsizlik çözüm kümesi olarak $\{x : f(x) \geq \alpha\}$ biçiminde bir ***yarı-uzaya*** sahiptir. Burada asıl ilgi alanımız; I sonlu olması gerekmeyen bir indeks kümesi olmak üzere $f_i(x) \geq \alpha_i$, ($i \in I$) ***lineer eşitsizlikler sistemi*** olacaktır.

1.4.1. Homojen denklem sistemleri

Lineer cebirdeki homogen denklem sistemleri ile ilgili bir teorem ile başlayayıp, daha sonra da eşitsizlikler için benzer yapıları vereceğiz. Bir f ***fonksiyonelinin sıfır uzayını***

$$f^0 = \{x \in X : f(x) = 0\}$$

kümesi ile gösterelim. $\{f_1, f_2, ..., f_m\}$ kümesinin ***linear gerenini*** *de* $L(f_1, f_2, ..., f_m)$ ile belirtelim.

Teorem 1.4.1. (Lineer Bağımlılık Teoremi) Bir lineer fonksiyoneller kümesi için aşağıdakiler denkdir.

(i) $f^0 \supseteq \bigcap_{i=1}^{m} f_i^0$

(ii) $f \in L(f_1, f_2, ..., f_m)$

İspat. (ii)$\Rightarrow$(i) olduğunu göstermek kolaydır. Gerçekten, eğer $f = \sum_{i=1}^{m} \lambda_i f_i$ ve $x \in \bigcap_{i=1}^{m} f_i^0$ ise bu durumda $\forall i$ için $f_i(x) = 0$ olup böylece $f(x) = 0$ olduğu açıktır.

Tersinin ispatı için m üzerinde tümevarım kullanacağız. Kabul edelim ki $m = 1$ ve $f^0 \supseteq f_1^0$ olsun. Eğer $f_1 = 0$ ise $f_1^0 = X$ olup, dolayısıyla $f^0 = X$ ve $f = 0$ dır. Bu durumda $f \in L(f_1)$ dir. Eğer $f_1 \neq 0$ ise $f_1(y) = 1$ olacak şekilde bir y noktası alalım ve kabul edelim ki x, X in herhangi bir noktası olsun. Buradan,

$f_1[x - f_1(x)y] = f_1(x) - f_1(x)f_1(y) = 0$ ve böylece $x - f_1(x)y \in f_1^{\,0}$ dır. Hipotezden dolayı, $x - f_1(x)y \in f^{\,0}$ dır. O halde, $f(x) - f_1(x)f(y) = 0$ yani $f = f(y)f_1$ dir. Böylece $f \in L(f_1)$ olur.

Şimdi *m* tamsayısı için teorem doğru olsun.

$$f^{\,0} \supseteq \bigcap_{i=1}^{m+1} f_i^{\,0}$$

kabul edelim. $Y = f_{m+1}^{\,0}$ diyelim. Bu, X in bir altuzayıdır. Y nin kısıtlamasını $f \mid Y$ notasyonu ile göstererek,

$$(f \mid Y)^0 \supseteq \bigcap_{i=1}^{m} (f_i \mid Y)$$

dir. Tümevarım hipotezinden uygun λ_i ler için,

$$f \mid Y = \sum_{i=1}^{m} \lambda_{i.} f_i \mid Y$$

dir. Şimdi, iki denk denklem; $(f - \sum_{i=1}^{m} \lambda_i f_i) \mid Y = 0$ ve $(f - \sum_{i=1}^{m} \lambda_i f_i)^0 \supseteq f_{m+1}^{\,0}$ dır. Teoremin $m = 1$ durumunu kullanarsak, bazı λ_{m+1} ler için ,

$$f - \sum_{i=1}^{m} \lambda_i f_i = \lambda_{m+1} f_{m+1}$$

sonucuna varırız.

1.4.2. Lineer eşitsizlikler

Şimdi yukarıdaki teoremin lineer eşitsizliklere karşılık gelenini verelim. Fonksiyonların sıfır uzaylarına karşılık gelen kavram bu durumda

$$f^+ = \{x \in X : f(x) \geq 0\}$$

yarı-uzayları olacaktır. $f_1, f_2, ..., f_m$ lerin sıradan lineer gereni yerine, bunların doğurduğu

$$C\{f_1, f_2 ..., f_m\} = \left\{\sum_{i=1}^{m} \lambda_i f_i : \lambda_i \geq 0\right\}$$

konisini göz önüne alacağız.

Teorem 1.4.2. (Farkas Teoremi) f ve f_i lineer fonksiyonları için aşağıdakiler denktir.

(i) $f^+ \supseteq \bigcap_{i=1}^{m} f_i^+$

(ii) $f \in C(f_1, f_2, ..., f_m)$

İspat. Eğer (ii) doğru ise buradan kolayca (i) elde edilir. Gerçekten, $\lambda_i \geq 0$ olmak üzere,

$$f = \sum_{i=1}^{m} \lambda_i f_i$$

olsun. Eğer $x \in \bigcap_{i=1}^{m} f_i^+$ ise, bu durumda her i için $f_i(x) \geq 0$ ve böylece açıkça $f(x) \geq 0$ dır.

Tersinin ispatını $X = \mathbb{R}^n$ kabulü altında verelim. Her lineer fonksiyon, uygun bir v için $f(x) = \langle v, x \rangle$ formundadır. Şimdi C, $C(f_1, f_2, ..., f_m)$ konisini göstermek üzere, $f \notin C$ olduğunu kabul edelim. Ayırma teoreminden,

$$\langle v, f \rangle < \inf_{u \in C} \langle v, u \rangle$$

olacak şekilde bir v vektörü vardır.

$$k = \inf_{u \in C} \langle v, u \rangle$$

olsun. $0 \in C$ olduğunda, $k \leq 0$ dır. Eğer $k < 0$ ise $k \leq \langle v, u \rangle < 0$ olacak şekilde bir $u \in C$ seçelim. Tüm pozitif t ler için, $tu \in C$ dir. Eğer t yeterince büyük ise $\langle v, tu \rangle < k$ olacaktır ki bu da çelişkidir. O halde $k = 0$ dır ve her $u \in C$ için,

$$\langle v, f \rangle < 0 \leq \langle v, u \rangle$$

dır. Buradan $f(v) < 0 \leq f_i(v)$ elde edilir ki bu da (i) nin yanlış olduğunu gösterir.

Farkas Teoremi aşağıdaki matris-vektör formülasyonuna sahiptir.

Teorem 1.4.3. Verilen bir *A* matrisi ve bir *c* vektörünün aşağıdaki özellikleri denktir.

(i) Her x için, eğer $Ax \geq 0$ ise $c^T x \geq 0$ dır.

(ii) Bazı y ler için, $y \geq 0$ ve $c = A^T y$ dir.

1.4.3. Tutarlı ve tutarlı olmayan sistemler

Şimdi Teorem 1.4.3 ün homogen olmayan bir benzerini vereceğiz. Bunun içinde şu terminolojiyi kullanacağız: Eğer $g_i(x) \geq \beta_i$ sisteminin her bir çözümü $f_i(x) \geq \alpha_i$ şeklindeki bir eşitsizlik sisteminin de bir çözümü ise ikinci sisteme birincinin bir *sonucudur* denir. Böylece,

$$\{x : g_i(x) \geq \beta_i \ , \ \forall i\} \subseteq \{x : f_i(x) \geq \alpha_i \ \ \forall i\}$$

dir.

Teorem 1.4.4. (Homojen olmayan Farkas teoremi) Eğer $f(x) \geq \alpha$ lineer eşitsizliği

$$g_i(x) \geq \beta_i \quad (1 \leq i \leq n)$$

tutarlı lineer eşitsizlikler sisteminin bir sonucu ise bu durumda, uygun $\theta_i \geq 0$ lar için ;

$$f = \sum_{i=1}^{n} \theta_i g_i \quad \text{ve} \quad \sum_{i=1}^{n} \theta_i \beta_i \geq \alpha$$

dır.

İspat.

$$g_i(x) - \lambda\beta_i \geq 0 \quad (\lambda > 0) \tag{1.3}$$

sistemini gözönüne alalım. Eğer (x, λ) ikilisi (1.3) sistemini çözüyorsa, bu durumda $g_i(x) \geq \lambda\beta_i$ ve $g_i(x/\lambda) \geq \beta_i$ elde edilir. Teoremdeki hipotezden $f(x/\lambda) \geq \alpha$ dır. Böylece,

$$f(x) - \lambda\alpha \geq 0 \tag{1.4}$$

olur. Bu ise (1.4) eşitsizliğinin (1.3) eşitsizliğinin bir sonucu olduğunu gösterir. Şimdi,

$$g_i(x) - \lambda\beta_i \geq 0 \quad (\lambda \geq 0) \tag{1.5}$$

sistemini gözönüne alalım. Eğer (1.4), (1.5)'in bir sonucu ise Farkas Teoreminin homojen formu uygulanırsa

$$(f; -\alpha) = \sum_{i=1}^{n} \theta_i (g_i; -\beta_i) + \theta_0 (0;1) \qquad (\theta_i \geq 0)$$

elde edilir. Burada $(f; -\alpha)$ ikilisi basitçe bir vektör çiftini göstermektedir. Bu sonuç aşağıdaki formda da yazılabilir.

$$f = \sum_{i=1}^{n} \theta_i g_i \qquad \alpha = \sum_{i=1}^{n} \theta_i \beta_i - \theta_0 \leq \sum_{i=1}^{n} \theta_i \beta_i$$

Bu ise ispatlanmak istenen şeydir. Fakat ispatımız bununla bitmemektedir. Çünkü (1.4) ün (1.3) ün sonucu olduğunu gösterdiğimiz halde (1.5) in sonucu olmayabilir. Bu durumda (1.5) in (1.3) ün ya da (1.4) ün de çözümü olmayan bir çözümü vardır. Bu şekilde bir çözümü ifade eden (u,λ) için mutlaka $\lambda = 0$ dır. Çünkü diğer durumda bu çift (1.3) ü ve dolayısıyla (1.4) ü de çözer. O halde, $g_i(u) \geq 0 > f(u)$ dur. Hipotezden dolayı $g_i(v) \geq \beta_i$ olacak şekilde bir v vektörü vardır. $f(u+\lambda v) < \lambda\alpha$ olacak şekilde pozitif bir λ sayısı seçelim. Bu mümkündür çünkü, $\lambda \downarrow 0$ olduğunda sol taraf negatif $f(u)$ ya yakınsarken sağ taraf da 0'a yakınsar. Böylece, $g_i(u/\lambda + v) \geq g_i(v) \geq \beta_i$ için $f(u/\lambda + v) < \alpha$ olduğundan hipotez ile bir çelişki elde etmiş oluruz.

1.4.4. Matris-vektör formları

Teorem 1.4.4 ün matris-vektör biçiminde ifade edilişi aşağıdaki gibidir.

Teorem 1.4.5. Eğer ,

$$Ax \geq b$$

sistemi tutarlı ve,

$$Ax \geq b \qquad c^T x < \alpha$$

sistemi de tutarlı değil ise,

$$A^T y = c \qquad y^T b \geq \alpha \qquad y \geq 0$$

sistemi tutarlıdır.

Teorem 1.4.6. Eğer,

$$Ax = b \qquad x \geq 0 \tag{1.6}$$

sistemi tutarlı değil ise,

$$A^T y \geq 0 \quad b^T y \leq 0 \tag{1.7}$$

sistemi tutarlıdır.

İspat. Eğer (1.6) tutarlı değil ise buradan,

$$b \notin K \equiv \{Ax : x \geq 0\}$$

dır. K kapalı ve konveks olduğunda, Ayırma teoremi uygulanır ve aşağıdaki şekilde bir y vektörü vardır.

$$\langle y, b \rangle < \inf_{x \geq 0} \langle y, Ax \rangle$$

Bu hesaplamada x 0 olabileceğinden $\langle y, b \rangle < 0$ elde ederiz. Geriye $A^T y \geq 0$ olduğunu göstermek kalır. Eğer bu doğru değilse, bu durumda bazı α indeksi için $\left(A^T y\right)_\alpha < 0$ dır. x, $x_j = \lambda \delta_{\alpha j}$ koordinatlı bir vektör olsun. Bu durumda,

$$\langle y, Ax \rangle = \sum_{j=1}^{n} \left(A^T y\right)_j x_j = \lambda \left(A^T y\right)_\alpha \to -\infty \qquad (\lambda \to +\infty)$$

olur. Bu durumda uygun λ için, $\langle y, Ax \rangle < \langle y, b \rangle$ olup bu ise çelişkidir. O halde, y (1.7) sistemini çözer.

Teorem 1.4.7. Eğer,

$$Ax \leq b, \quad x \geq 0 \tag{1.8}$$

sistemi tutarsız ise, bu durumda

$$A^T y \geq 0, \quad b^T y < 0, \quad y \geq 0 \tag{1.9}$$

sistemi tutarlıdır.

İspat. Eğer (1.8) sistemi tutarsız ise, bu durumda

$$Ax + z = b, \quad x \geq 0, \quad z \geq 0 \tag{1.10}$$

sistemi de tutarsızdır. (1.10) u,

$$[A, I]\begin{bmatrix} x \\ z \end{bmatrix} = b \quad \begin{bmatrix} x \\ z \end{bmatrix} \geq 0 \tag{1.11}$$

şeklinde yazarsak, Teorem 1.4.6 dan

$$\begin{bmatrix} A^T \\ I \end{bmatrix} y \geq 0 \quad b^T y < 0 \tag{1.12}$$

sistemi tutarlıdır. Bu ise Sistem (1.9) ile aynıdır.

2. LİNEER PROGRAMLAMA

Lineer eşitsizlik kısıtları altında bir lineer objektif fonksiyonu ile karakterize edilen Kısım 1.1, Örnek 1.1.1 deki tipten bir problem, bir *lineer (doğrusal) programlama* problemi olarak adlandırılır. Lineer programlama terimi bilgisayar programından ziyade, iş ve ekonomik yatırımların programlanması anlamına gelmektedir. En açık ve teknik anlamda, lineer programlama aşağıdaki şekilde ifade edilebilir

2.1. İlk Standart Form

$c \in R^n, b \in R^m$ ve $A \in R^{m\times n}$ olsun. $x \in R^n$, $Ax \le b$ ve $x \ge 0$ kısıtlarına altında $c^T x$ in maksimum değerini bulunuz.

Burada $x = (x_1, x_2, ..., x_n)^T$ olmak üzere, $x \ge 0$ vektör eşitsizliğinden kasıt, $\forall i \in \{1,2,...,n\}$ için $x_i \ge 0$ olması anlamındandır. Benzer şekilde

$$Ax \le b$$

eşitsizliği; her $i \in \{1,2,...,m\}$ için $\sum_{j=1}^{n} a_{ij} x_j \le b_i$ anlamına gelmektedir. Şimdi, lineer programlamada yaygın olarak kullanılan bazı terminolojileri tanımlayalım:

$$K = \{x \in \mathbb{R}^n : Ax \le b, x \ge 0\}$$

kümesine ***uygun küme*** denir.

$$v = \sup\{c^T x : x \in K\}$$

sayısına ***problemin değeri*** denir. K nın herhangi bir noktası bir ***uygun nokta*** olarak adlandırılır. $c^T x = v$ olacak şekildeki herhangi bir $x \in K$ noktasına bir ***çözüm*** veya bir ***optimal uygun nokta*** denir. $x \mapsto c^T x = \sum_{j=1}^{n} c_j x_j$ fonksiyonu ***objektif (amaç)***

fonksiyonu olarak adlandırılır. Problem, A, b, c verileri ile tam olarak belirlendiğinden problem genellikle (A, b, c) lineer programlama problemi olarak ifade edilir.

2.2. Problemleri Dönüştürme Teknikleri

Değişkenleri lineer eşitsizlik kısıtları olan bir lineer fonksiyonun optimizasyonunu içeren hemen hemen bütün problemler, lineer programlama formatı ile verilebilir. Bunun için çoğu zaman aşağıdakilerden bir veya birkaçına gereksinim vardır.

(i) Eğer $c^T x$ in minimum yapılması isteniyorsa bu, $-c^T x$ in maksimum yapılması ile aynı anlama gelmektedir.

(ii) $a^T x \geq \beta$ formundaki herhangi bir kısıt $-a^T x \leq -\beta$ ya denktir.

(iii) $a^T x = \beta$ formundaki herhangi bir kısıt $a^T x \leq \beta$, $-a^T x \leq -\beta$ ya denktir.

(iv) $\left|a^T x\right| \leq \beta$ formundaki herhangi bir kısıt $a^T x \leq \beta$, $-a^T x \leq \beta$ ya denktir.

(v) Eğer amaç fonksiyonu artı bir sabiti içeriyorsa, bunun çözümde hiçbir etkisi yoktur. Böylece, $c^T x + \beta$ nın maksimumu, $c^T x$ in maksimumu ile aynı noktaları verir.

(vi) Eğer verilen bir problem bir x_j değişkeninin negatif olmama koşuluna gereksinim duymuyorsa x_j yi, $x_j = u_j - v_j$ şeklinde negatif olmayan iki değişkenin farkı olarak ifade edebiliriz.

Örnek 2.2.1. Aşağıdaki problemi standart formda lineer programlama problemine dönüştürelim.

$$\text{Minimum: } 7x_1 - x_2 + x_3 - 4$$

$$\text{Kısıtlar:} \begin{cases} x_1 + x_2 - x_3 \geq 2 \\ 3x_1 + 4x_2 + x_3 = 6 \\ \left|x_1 - 2x_2 + 3x_3\right| \leq 5 \\ x_1 \geq 0, x_2 \leq 0 \end{cases}$$

Çözüm. $u_1 = x_1, -u_2 = x_2$ ve $u_3 - u_4 = x_3$ olarak kabul edelim. Böylece bize verilen lineer programlama problemi aşağıdaki şekle dönüşmüş olur.

$$\text{Maksimum: } -7u_1 - u_2 - u_3 + u_4$$

$$\text{Kısıtlar: } \begin{cases} -u_1 + u_2 + u_3 - u_4 \leq -2 \\ 3u_1 - 4u_2 + u_3 - u_4 \leq 6 \\ -3u_1 + 4u_2 - u_3 + u_4 \leq -6 \\ u_1 + 2u_2 + 3u_3 - 3u_4 \leq 5 \\ -u_1 - 2u_2 - 3u_3 + 3u_4 \leq 5 \\ u_1 \geq 0, u_2 \geq 0, u_3 \geq 0, u_4 \geq 0 \end{cases}$$

Verilen bir (A, b, c) lineer programlama probleminin bir çözümü olabilir de olmayabilir de. Eğer, K uygun kümesi boş küme olursa hiç bir çözüm varolmaz. Eğer, uygun küme boş olmayan ve sınırsız bir küme ise objektif fonksiyonunun K da bir üst sınırı olmayabilir. Bu durumda $v = +\infty$ olup çözüm yoktur. Eğer, K boş küme ise $v = -\infty$ olur çözüm yine yoktur. Eğer, K boş küme değilse ve sınırlı ise, bu durumda en az bir çözüm vardır. Bu, K nın bu durumda kompakt ($\mathbb{R}^n$ de kapalı ve sınırlı) olmasının ve böylece, objektif fonksiyonunun (ki sürekli fonksiyondur) supremumunu K da almasının sonucudur.

2.3. Dual Problem

Herhangi bir (A, b, c) lineer programlama problemi ile $(-A^T, -c, -b)$ şeklinde bir başka problem ilişkilendirilebilir. Bu probleme orijinal problemin ***duali*** denir. Örneğin,

$$\text{Maksimum: } 3x_1 - 2x_2$$

$$\text{Kısıtlar: } \begin{cases} 7x_1 + x_2 \leq 18 \\ -3x_1 + 5x_2 \leq 25 \\ 6x_1 - x_2 \leq 13 \\ x_1 \geq 0,\ x_2 \geq 0 \end{cases}$$

probleminin duali aşağıdaki gibidir.

$$\text{Maksimum:} \quad -18y_1 - 25y_2 - 13y_3$$

$$\text{Kısıtlar:} \quad \begin{cases} -7y_1 + 3y_2 - 6y_3 \le -3 \\ -y_1 - 5y_2 + y_3 \le 2 \\ y_1 \ge 0,\ y_2 \ge 0,\ y_3 \ge 0 \end{cases}$$

Aşağıdaki teoremler bir lineer programlama problemi ile onun dual problemi arasındaki ilişkileri vermektedir.

Teorem 2.3.1. Eğer *x*, (A, b, c) lineer programlama problemi için ve *y* de $(-A^T, -c, -b)$ dual problemi için uygun noktalar ise, bu durumda

$$c^T x \le y^T Ax \le b^T y$$

olur. Eşitlik durumunda, *x* ve *y* ler sırasıyla kendi problemlerinin çözümleri olur.

İspat. Hipotezden, *x* ve *y* noktaları aşağıdaki eşitsizlikleri sağlar.

$$x \ge 0 \qquad Ax \le b \qquad y \ge 0 \qquad -A^T y \le -c\,.$$

Buradan,

$$c^T x \le (A^T y)^T x = y^T Ax \le y^T b = b^T y$$

elde edilir. Böylece, her iki problemdeki v_1 ve v_2 değerleri aşağıdakileri sağlamak zorundadır.

$$c^T x \le v_1 \le b^T y$$

$$-b^T y \le v_2 \le -c^T x$$

Eğer $c^T x = b^T y$ ise $c^T x = v_1 = b^T y = -v_2$ olduğu açıktır.

Teorem 2.3.1 çoğunlukla lineer programlama problemindeki v_1 değerini tahmin etmek için kullanılır. Eğer x ve dual problemdeki y uygun noktaları biliniyorsa, $c^T x \le v_1 \le b^T y$ eşitsizliği v_1 ‘i içeren bir aralık tanımlar.

Lemma 2.3.1. Eğer,

$$Ax \ge b$$

sistemi *tutarlı* ise (yani bir çözüme sahipse), ve

$$Ax \ge b \qquad c^T x < \alpha$$

sistemi *tutarsız* ise (yani çözümü yoksa), bu durumda

$$A^T y = b \qquad y^T b \ge \alpha \qquad y \ge 0$$

sistemi tutarlıdır.

Lemma 2.3.2. Eğer,

$$Ax = b \qquad x \ge 0$$

sistemi tutarsız ise, bu durumda

$$A^T y \ge 0 \qquad b^T y \le 0$$

sistemi tutarlıdır.

Lemma 2.3.3. Eğer,

$$Ax \le b \quad x \ge 0$$

sistemi tutarlı değilse, bu durumda

$$A^T y \ge 0 \quad b^T y < 0 \quad y \ge 0$$

sistemi tutarlıdır.

Teorem 2.3.2. Eğer bir lineer programlama problemi ve onun duali uygun noktalara sahipseler, her iki problemin de çözümü vardır ve bu çözümlerin biri diğerinin negatifidir.

İspat. Teorem 2.3.1 den

$$x \geq 0 \qquad Ax \leq b \qquad y \geq 0 \qquad -A^T y \leq -c \qquad c^T x \geq b^T y$$

olacak şekilde x ve y nin var olduğunu ispatlamak yeterlidir. Gerçekten, bu koşullardaki (x, y) ikilisi, orijinal problem için x, ve dual problem için de y çözümlerini verir. Aşağıdaki lineer eşitsizlik sisteminin tutarlı olduğunu ispatlamalıyız.

$$\begin{bmatrix} A & 0 \\ 0 & -A^T \\ -c^T & b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} b \\ -c \\ 0 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Aksine, bu sistemin tutarlı olmadığını kabul edelim ve bu durumda bir çelişki elde etmeye çalışalım. Lemma 2.3.3 den

$$\begin{cases} \begin{bmatrix} A^T & 0 & -c \\ 0 & -A & b \end{bmatrix} \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} b^T & -c^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} < 0 \\ \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} \geq 0 \end{cases}$$

sistemi tutarlıdır. Burada u ve v vektörler ve λ bir sabittir. (u, v, λ) nın bu sistemi sağladığını kabul edelim. Böylece,

$$\begin{cases} A^T u - \lambda c \geq 0 & -Av + \lambda b \geq 0 & b^T u - c^T v < 0 \\ u \geq 0 & v \geq 0 & \lambda \geq 0 \end{cases}$$

olur. İlk olarak $\lambda > 0$ olduğunu kabul edelim. Bu durumda $\lambda^{-1} v$, (A,b,c) problemi için $\lambda^{-1} u$ da $\left(-A^T,-c,-b\right)$ dual problemi için uygundur. Böylece, Teorem 2.3.1 den dolayı $c^T\left(\lambda^{-1} v\right) \leq b^T\left(\lambda^{-1} u\right)$ ve $b^T u - c^T v \geq 0$ olur ki bu da çelişkidir.

Eğer $\lambda = 0$ ise $A^T u \geq 0 \geq Av$ olur. x'i orijinal problem y'yi de dual problem için uygun aldığımızda daha önceki eşitsizlikte çelişki elde ederiz:

$$c^T v \leq \left(A^T y\right)^T v = y^T A v \leq 0 \leq \left(A^T u\right)^T x = u^T (Ax) \leq u^T b = b^T u$$

Teorem 2.3.3. Eğer bir doğrusal programlama probleminin veya onun dualinin çözümü var ise diğerinin de çözümü vardır.

İspat. Dual problemin duali orijinal problem olduğundan dolayı teoremin sadece bir durumunu ispatlamak yeterlidir. Kabul edelim ki $\left(-A^T,-c,-b\right)$ dual problemi bir y_0 çözümüne sahip olsun. Bu durumda,

$$-A^T y \leq -c \qquad y \geq 0 \quad -b^T y \geq -b^T y_0$$

eşitsizlik sistemi tutarlı değildir. Kuşkusuz, üçüncü eşitsizliğin ihmal edildiği karşılık gelen sistem tutarlıdır. Tutarlı olmayan sistemi aşağıdaki formda yazalım.

$$\begin{bmatrix} A^T \\ I \end{bmatrix} y \geq \begin{bmatrix} c \\ 0 \end{bmatrix} \qquad b^T y \leq b^T y_0 .$$

Lemma 2.3.1 den

$$\begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = b \qquad \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{bmatrix} c \\ 0 \end{bmatrix} \geq b^T y_0 \qquad \begin{bmatrix} x \\ u \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

sistemi tutarlıdır. Böylece,

$$Ax + u = b \qquad x^T c \geq b^T y_0 \qquad x \geq 0 \qquad u \geq 0$$

ve buradan,

$$Ax \leq b \qquad cx^T \geq b^T y_0 \qquad x \geq 0$$

elde edilir. Böylece, Teorem 2.3.1 den x, orijinal problemin çözümüdür.

Teorem 2.3.4. x ve y sırasıyla doğrusal programlama problemi ve onun duali için uygun noktalar olsun. Bu noktaların sırasıyla kendi problemlerinin çözümleri olması çin gerek ve yeter koşul, $y_i > 0$ olacak şekildeki $\forall i$ için $(Ax)_i = b_i$, ve $x_i > 0$ olacak şekildeki $\forall i$ için $(A^T y)_i = c_i$ olmasıdır.

İspat. Eğer x ve y ler çözümler ise, Teorem 2.3.1 ve 2.3.2 den

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

olup, buradan $y^T(b - Ax) = 0$ dır. $y \geq 0$ ve $b - Ax \geq 0$ olduğunda $\forall i$ için $y_i(b_i - (Ax)_i) = 0$ olur. Böylece i, $y_i > 0$ için bir indeks olduğunda, $(Ax)_i = b_i$ olur. Diğer durum simetriden elde edilir. Tersine, $\forall i$ için $y_i(b_i - (Ax)_i) = 0$ ve $x_i(c_i - (A^T y)_i) = 0$ olduğunu kabul edelim. Buradan

$$b^T y \;=\; y^T b = y^T Ax = x^T A^T y = x^T c = c^T x$$

olur. Teorem 2.3.1 den x ve y sırasıyla kendi problemlerinin çözümleridir.

2.4. Simpleks Algoritması

Lineer programlama problemlerinin çözümlerinde en güçlü algoritmalardan birisi Simpleks algoritmsıdır. Bu kısımda bu algoritma tanıtılacaktır.

2.4.1. İkinci standart form

Standart teknikler anlamında, bir lineer programlama problemi aşağıdaki forma dönüştürülebilir.

$x \in \mathbb{R}^n, c \in \mathbb{R}^n, b \in \mathbb{R}^m$ ve $A \in \mathbb{R}^{m \times n}$ olmak üzere, $Ax = b$ ve $x \geq 0$ kısıtları altında $c^T x$ in maksimum değerini bulunuz.

$c^T x$ in objektif (amaç) fonksiyonunu tanımladığını hatırlayalım. Ayrıca, eğer bir lineer programlama probleminin

$$\mathrm{Ax} \leq b$$

formunda kısıtları varsa,

$$Ax + u = b$$

olacak şekilde bir $u \geq 0$ vektörü tanımlanarak, yukarıda verilen ikinci standart formun elde edilebileceğini not edelim.

İkinci standart formdaki problem için uygun küme

$$K = \left\{ x \in \mathbb{R}^n : Ax = b,\ x \geq 0 \right\}$$

dır. Herhangi $x \in K$ için, x in bileşenleri pozitif olacak şekilde

$$I(x) = \left\{ i : 1 \leq i \leq n \quad \text{ve} \quad x_i > 0 \right\}$$

indis kümesini tanımlayalım. A matrisinin kolonları $A_1, A_2, ..., A_n$ olsun. Böylece,

$$Ax = b$$

denklemi

$$\sum_{i=1}^{n} x_i A_i = b$$

şekline dönüşür. Şimdi aşağıdaki önemli teoremi verelim

Teorem 2.4.1. $x \in K$ olsun. x in aşağıdaki özellikleri denktir.

(i) $x,\ K$ nın bir ekstrem(uç) noktasıdır.

(ii) $\{ A_i : i \in I(x) \}$ lineer bağımsızdır.

İspat. (ii) nin doğru olduğunu kabul edelim. (i) nin doğruluğunu ispatlayacağız. $u \in K$, $v \in K$ ve $0 < \theta < 1$ olmak üzere $x = \theta u + (1-\theta)v$ olsun. Her bir $i \notin I(x)$ için $0 = x_i = \theta u_i + (1-\theta)v_i$ olur. Böylece, $u_i \geq 0$, $v_i \geq 0$ olduğundan, her $i \notin I(x)$ için $u_i = v_i = 0$ sonucu ortaya çıkar. Buradan,

$$0 = Au - Av = \sum_{i=1}^{n} (u_i - v_i) A_i = \sum_{i \in I(x)} (u_i - v_i) A_i$$

elde edilir. (ii) den, $u_i = v_i$ olur. O halde $x,\ K$ nın bir ekstrem noktasıdır.

Şimdi kabul edelimki (i) doğru olsun. Eğer, $\sum_{i \in I(x)} w_i A_i = 0$ ise, bu durumda $i \notin I(x)$ için $w_i = 0$ alalım. Açıkça

$$\sum_{i \in I(x)} (x_i \mp \lambda w_i) A_i = b$$

dir. $i \in I(x)$ için $x_i > 0$ olduğundan, $x_i + \lambda w_i > 0$ ve $x_i - \lambda w_i > 0$ olacak şekilde yeterince küçük bir $\lambda \neq 0$ alabiliriz. Böylece, $u = x + \lambda w$ ve $v = x - \lambda w$ lar uygun noktalardadır. $x = \frac{1}{2}(u + v)$ ve $x,\ K$ nın bir ekstrem noktası olduğunda, $u = v$ ve böylece $w = 0$ olur. Bu (ii)yi ispatlar.

Sonuç 2.4.1. K uygun kümesi sadece sayılabilir çoklukta ekstrem noktası içerir.

İspat. E, K nın ekstrem noktalarının kümesi olsun. Her $x \in E$ için $I(x) \subseteq \{1,2,...,n\}$ ve böylece $I : E \to 2^{\{1,2,...,n\}}$ dir (2^S gösterimi S nin tüm alt kümelerinin ailesini göstermektedir). Bu şekilde (E üzerinde) tanımlanan I dönüşümü birebirdir. Bunu göstermek için her $x, y \in E,\ x \neq y$ olsun. Bu durumda,

$$b = Ax = \sum_{i=1}^{n} x_i A_i = \sum_{i \in I(x)} x_i A_i$$

$$b = Ay = \sum_{i=1}^{n} y_i A_i = \sum_{i \in I(y)} y_i A_i$$

olur. Eğer $I(x) = I(y)$ ise, bu durum $\{A_i : i \in I(x)\}$ kümesinin lineer bağımsızlığı ile çelişir. Böylece, $I : E \to 2^{\{1,2,...,n\}}$ içine olan bu dönüşüm E nin elemanlarının 2^n den fazla olamayacağını gösterir.

Dantzig [1] in simpleks algoritması iki kısım içerir. İlk kısımda, K nın bir $x^{(1)}$ başlangıç ekstrem noktası bulunur. İkinci kısımda ise, oluşturulan her noktada objektif fonksiyonu artan olacak şekilde, $x^{(1)}$ den başlanarak sonlu sayıda ekstrem noktaları oluşturulur. Bu diziyi $x^{(1)}, x^{(2)}, x^{(3)}...$ şeklinde gösterirsek $c^T x^{(1)} < c^T x^{(2)} < c^T x^{(3)} < ...$ olur. Eğer problemin bir çözümü yoksa bu durum algoritmanın akışından anlaşılır. Eğer, lineer programlama problemi bir çözüme sahipse, bu durumda algoritmanın sonlu bir adımında objektif fonksiyonunu maksimum yapan bir x^k ekstrem değeri yani çözüm üretilir.

2.4.2. Soyut form

Bu kısımda, algoritmanın ikinci kısmının soyut formunu açıklayacağız. Öncelikle aşağıdaki hipotezi kabul edelim.

Kabul 2.4.1. (Düzgünlük kabülü) K uygun kümesinin her bir x ekstrem noktası tam olarak m tane pozitif bileşene sahiptir.

Kabul edelim ki *x, K* nın bir ekstrem noktasıdır. Teorem 2.4.1 den, $\{A_i : i \in I(x)\}$ kümesi lineer bağımsızdır. O halde düzgünlük kabülünden dolayı küme $\mathbb{R}^m$ için bazdır. (Bu nedenden dolayı bu şekilde bulunan *x* noktasına literatürde ***temel uygun nokta*** denilmektedir.)

$$A_j = \sum_{i \in I(x)} D_{ij} A_i \qquad (1 \le j \le n)$$

olacak şekilde D_{ij} katsayıları varolmalıdır. Eğer $i \notin I(x)$ ise $D_{ij} = 0$ alırsak, bu durumda

$$A_j = \sum_{i=1}^{n} D_{ij} A_i$$

veya

$$A = AD^T$$

olur. *c,* objektif fonksiyonunda yer alan vektör olmak üzere $d = D^T c$ tanımlayalım. *D* ve *d* nin *x* e bağlı olduğunu ve algoritma boyunca değişeceğini not edelim.

Herhangi $q \notin I(x)$ indeksi ve herhangi bir $\lambda \in \mathbb{R}$ için

$$\begin{aligned} b = Ax &= \sum_{i=1}^{n} x_i A_i + \lambda A_q - \lambda \sum_{i=1}^{n} D_{iq} A_i \\ &= \sum_{i=1}^{n} (x_i - \lambda D_{iq} + \lambda \delta_{iq}) A_i \\ &= \sum_{i=1}^{n} y_i A_i \end{aligned}$$

dir. Böylece, $y = x - \lambda D_q + \lambda d_q$ olmak üzere

$$Ay = b$$

dir. Burada D_q, D nin q. kolonunu göstermektedir. Şimdi amacımız y in K nın bir ekstrem noktası (yani $y \geq 0$) ve $c^T y > c^T x$ olacak şekilde q ve λ yı seçmektir.

$q \notin I(x)$ olduğunda $D_{qq} = 0$ ve $x_q = 0$ olup, böylece $y_q = \lambda$ dır. y bir uygun nokta olacağından, $\lambda \geq 0$ olmalıdır. Şimdi,

$$
\begin{aligned}
c^T y &= \sum_{i=1}^{n} c_i x_i - \lambda \sum_{i=1}^{n} c_i D_{iq} + \lambda \sum_{i=1}^{n} c_i \delta_{iq} \\
&= c^T x - \lambda c^T D_q + \lambda c_q \\
&= c^T x - \lambda (c_q - d_q)
\end{aligned}
$$

olur. Objektif fonksiyonunun değerini artırmak için q yu $c_q > d_q$ olacak şekilde seçelim. Eğer $c \leq d$ ise, q varolamaz ve hesaplama durur: x çözümdür. Diğer durumda $c_q - d_q$ yu mümkün olduğu kadar büyük yapacak şekilde q yu seçelim ve bu adımdan sonra sabitleyelim.

λ nın seçimi, $\lambda > 0$ ve $I(y)$ en fazla m elemana sahip olacak şekilde yapılır. y nin tanımında $I(y) \subseteq I(x) \cup \{q\}$ olduğu görülür. $I(x)$ in tam olarak m elemanı olduğundan, λ yı, $x_i - \lambda D_{iq}$ nun terimlerinden bir tanesi 0 ve diğerleri ≥ 0 olacak şekilde seçelim. Bununla beraber, dikkat edilirse eğer her $1 \leq i \leq n$ için $D_{iq} \leq 0$ ise, bu durumda $\forall \lambda > 0$ için $y \in K$ dır. $c^T y$ nin yukarıdaki formülünden bu durumda, $\lim_{\lambda \to \infty} c^T y = +\infty$ dur. Böylece, objektif fonksiyonu K da sınırsız olduğundan, bir çözüm yoktur. Eğer i nin bir veya birkaç değeri için $D_{iq} > 0$ ise, λ yı sıfırdan artan olarak düşünebiliriz. Başlangıçta, $i \in I(x)$ için $x_i - \lambda D_{iq} > 0$ dır. Bu terimlerin bazıları azalmaktadır ve bunlardan bir tanesi 0 olduğunda, karşılık gelen λ yı alıyoruz. Yani,

$$
\lambda = \min \left\{ \frac{x_i}{D_{iq}} : x_i > 0, D_{iq} > 0 \right\}
$$

dır. Böylece y yi tamamı ile belirlemiş olarak, bunun bir ekstrem nokta olduğunu belirlemek istiyoruz. İlk olarak $\lambda = \dfrac{x_p}{D_{pq}}$ olduğunu kabul edelim. Burada p, $x_p > 0$ ve $D_{pq} > 0$ olacak şeklindedir. Buradan

$$\mathrm{I(y)}\, I(y) \subseteq I(x) \bigcup \{q\} \setminus \{p\}$$

Teorem 2.4.1 den,

$$\left\{A_i : i \in I(x) \cup \{q\} \setminus \{p\}\right\}$$

kümesinin lineer bağımsız olduğunu ispatlamak yeterlidir.

$$\sum_{i \in I(x)} \beta_i A_i + \beta_q A_q = 0\,, \qquad \beta_p = 0$$

olduğunu kabul edelim. Eğer $\beta_q = 0$ ise, bu denklem

$$\sum_{i \in I(x)} \beta_i A_i = 0$$

denklemine indirgenir. $\{A_i : i \in I(x)\}$ in lineer bağımsızlığından, $i \in I(x)$ için $\beta_i = 0$ olur. O halde, bu durumda bütün β_i ler 0 dır. Bu yüzden $\beta_q \neq 0$ duruma bakacağız. Homojenlik nedeni ile $\beta_q = -1$ olduğunu kabul edelim. Böylece, denklem

$$A_q = \sum_{i \in I(x)} \beta_i A_i \quad , \quad \beta_p = 0$$

olur. Aynı zamanda,

$$A_q = \sum_{i=1}^{n} D_{iq} A_i = \sum_{i \in I(x)} D_{iq} A_i$$

dir. $\{A_i : i \in I(x)\}$ in lineer bağımsızlığından, $\beta_i = D_{iq}$ sonucu ortaya çıkar. Fakat, $\beta_p = 0$ ve $D_{pq} > 0$ olduğunda bu doğru olamaz. Böylece, β_q mutlaka 0 olmalıdır. O halde $y \in K$ olup bir ekstrem noktadır.

İspatlanması gereken bir ayrıntı daha vardır: Eğer $c \leq d$ ise, bu durumda x çözümdür. Bunun için u nun bir uygun nokta olduğunu kabul edelim. Buradan, $Ax = b = Au = A(D^T u)$ dur. Böylece, Ax ve $AD^T u$, $\{A_i : i \in I(x)\}$ in lineer birleşimleri olduklarından $x = D^T u$ olur. Buradan, ispatlamak istediğimiz gibi, $c^T u \leq d^T u = c^T Du = c^T x$ bulunur.

Şimdi yukarıda soyut olarak açıklanan simpleks algoritmasına somut bir örnek verelim.

Örnek 2.4.1.

$$\text{Maksimum: } F(x) = x_1 + 2x_2 + x_3$$

$$\text{Kısıtlar: } \begin{cases} \quad x_1 + x_2 + x_5 = 1 \\ x_1 + x_3 + x_4 + x_5 = 1 \\ x_i \geq 0, \quad (1 \leq i \leq 5) \end{cases}$$

problemini simpleks algoritması ile çözünüz.

Çözüm. Standart formda, veriler

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad c^T = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 \end{bmatrix}$$

dır. $x = (0,1,0,1,0)^T$ ve $I(x) = \{2,4\}$ ile başlayalım. $\{A_2, A_4\}$ ün $\mathbb{R}^2$ için bir baz olduğuna dikkat edelim. O halde, Teorem 2.4.1 den x, uygun kümenin bir ekstrem noktası yani temel uygun noktadır.

A nın her kolonu A_2 ve A_4 ün bir lineer birleşimidir. Aslında,

$$A_1 = A_2 + A_4, \quad A_2 = A_2, \quad A_3 = A_4, \quad A_4 = A_4 \text{ ve } A_5 = A_2 + A_4$$

dür. Böylece D matrisi

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

olur. d^T vektörü, D matrisindeki D^i satırlarının bir lineer birleşimidir. Yani,

$$d^T = c^T D = \sum_{i=1}^{n} c_i D^i = D^1 + 2D^2 + D^3 = \begin{bmatrix} 2 & 2 & 0 & 0 & 2 \end{bmatrix}^T$$

dir. $c-d$ vektörü

$$c-d = \begin{bmatrix} -1 & 0 & 1 & 0 & -2 \end{bmatrix}^T$$

olup, sadece bir bileşen pozitif ve $q=3$ dür. D_q, D nin q yüncü kolonunu göstermek üzere, $x = \lambda D_q$ vektörü

$$x-\lambda D_q = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T - \lambda \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 0 & 1-\lambda & 0 \end{bmatrix}^T$$

şeklindedir. $\lambda = 1$ alırsak, y vektörü,

$$y = x - \lambda D_q + \lambda d_3 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}^T$$

bulunur. Aynı işlemler, x yerine y alınarak tekrarlanırsa, ikinci adımda $d = (3,2,1,1,3)^T$ elde edilir. $c \le d$ olduğundan y bir çözümdür ve $F(y) = 3$ dür.

2.5. Tablo Yöntemi

Simpleks algoritmasının pratik kullanımında genellikle veriler bir tablo üzerinde gösterilip, daha sonra belli kurallara göre ardışık adımlarla düzenlenir. Aşağıdaki basit örnek bu algoritmanın tablo yöntemi ile nasıl oluşturulacağını göstermektedir.

$$\text{Maksimum : } F(x) = 6x_1 + 14x_2$$

$$\text{Kısıtlar : } \begin{cases} 2x_1 + \ \ x_2 \le 12 \\ 2x_1 + 3x_2 \le 15 \\ x_1 + \ \ 7x_2 \le 21 \\ x_1 \ge 0,\ x_2 \ge 0 \end{cases}$$

Simpleks algoritmasını kullanmak için geçici değişkenlerle problemi yeniden düzenlersek;

$$\text{Maksimum: } F(x) = 6x_1 + 14x_2 + 0x_3 + 0x_4 + 0x_5$$

$$\text{Kısıtlar: } \begin{cases} 2x_1 + \ x_2 + x_3 = 12 \\ 2x_1 + 3x_2 + x_4 = 15 \\ x_1 + 7x_2 + x_5 = 21 \\ x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0, x_5 \ge 0 \end{cases}$$

olup, böylece

$$\text{Maksimum: } F(x) = (6,\ 14,\ 0,\ 0,\ 0)^T x$$

$$\text{Kısıtlar: } \begin{cases} \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 2 & 3 & 0 & 1 & 0 \\ 1 & 7 & 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 12 \\ 15 \\ 21 \end{bmatrix} \\ x = (x_1, x_2, x_3, x_4, x_5)^T \ge 0 \end{cases}$$

elde edilir. İlk vektör $x = (0,0,12,15,21)^T$ olup, bu verilerin tümü aşağıdaki gibi bir tabloda yazılır:

6	14	0	0	0	0
2	1	1	0	0	12
2	3	0	1	0	15
1	7	0	0	1	21
0	0	12	15	21	

Simpleks yönteminin her adımı bir tablo ile başlar. İlk satır objektif fonksiyonu F nin katsayılarını içermektedir. $F(x) = c^T x$ in o anki değeri sağ üst köşede gösterilmektedir. Tablodaki sonraki m satır, eşitlik kısıtlarını oluşturan denklem sistemini temsil etmektedir. Bu sisteme uygulanan elemanter satır operasyonlarının, sistemin çözüm kümesini değiştirmeyeceğini hatırlayalım. Tablonun son satırı, o anki x vektörünü içerir. Dikkat edilirse, en üst ve en alt satır kullanılarak, $F(x) = c^T x$ değeri kolaylıkla hesaplanabilir. Tablonun genel formu;

C^T	0	$F(x)$
A	I	b
x(temel olmayan)	x(temel)	

şeklindedir.

2.5.1. Tablo kuralları

Simpleks yönteminde yer alan her tablo mutlaka aşağıdaki 5 kuralı sağlamalıdır.

(i) x vektörü mutlaka $Ax=b$ eşitlik kısıtlarını sağlamalıdır.

(ii) x vektörü mutlaka $x \geq 0$ eşitsizliğini sağlamalıdır.

(iii) x in (temel olmayan değişkenlere karşılık gelen) 0 değerini alan n bileşeni ve geriye kalan (temel değişkenlere karşılık gelen) genellikle sıfırdan farklı m bileşeni vardır. (Burada n ve m, geçici değişkenler sunulmadan önceki, orjinal problemle ilgili değerlere karşılık gelmektedir.)

(iv) Kısıtları tanımlayan matrisdeki her temel değişken sadece bir satırda yer alır.

(v) Objektif fonksiyonu F, mutlaka temel olmayan değişkenler cinsinden ifade edilmelidir.

Şimdi tablo yöntemine devam edelim. Örnekteki, ilk tabloda temel değişkenler x_3, x_4 ve x_5 ve temel olmayan değişkenler de x_1 ve x_2 dir. Dikkat edilirse bu tablo için yukarıdaki beş kural da doğrudur.

Her adımda, temel olmayan bir değişkeni bir temel değişken yapmakla, $F(x)$ in değerinin artıp armayacağını görmek için tablo incelenir. Buradaki örnekte (x_3, x_4 ve x_5 uygun şekilde ayarlanarak), x_1 veya x_2 nin artırılmasıyla $F(x)$ in değerinin gerçekten arttığı görülmektedir. F deki 14 katsayısı 6 katsayısından daha büyük olduğundan x_2 deki birim artış, $F(x)$ i x_1 deki birim artıştan daha fazla artıracaktır. O halde, x_1 i sıfırda sabit tutup, x_2 nin mümkün olduğunca fazla artmasına izin vereceğiz. Böylece:

$$0 \le x_3 = 12 - x_2$$

$$0 \le x_4 = 15 - 3x_2$$

$$0 \le x_5 = 21 - 7x_2$$

olur. Bu kısıtlar sırası ile, $x_2 \le 12$, $x_2 \le 5$, $x_2 \le 3$ olmasını gerektirir. Bunların her üçünü sağlayacak şekilde, $x_2 \le 3$ olmalıdır. O halde, x_2 yi 3 e kadar artırabiliriz. x_3, x_4 ve x_5 için karşılık gelen yeni değerler de verilen üç kısıttan elde edilir. Böylece, yeni x vektörü;

$$x = (0,\ 3,\ 9,\ 6,\ 0)^T$$

olur. Yeni temel değişkenler x_2, x_3 ve x_4 olup, şimdi daha önce verilen 5 kurala uyacak şekilde yeni tabloyu belirleyelim. (v) inci kuralı sağlamak için $x_2 = (21 - x_5)/7$ olduğuna dikkat edelim. Bunu F de yerine koyduğumuzda, objektif fonksiyonu;

$$F(x) = 6x_1 + 14x_2$$

$$= 6x_1 + 14(21 - x_5)/7 = 6x_1 - 2x_5 + 42$$

şeklini alır. (iv) kuralını sağlamak için 7 yi pivot eleman alarak Gauss eliminasyon yöntemi (elemanter satır operasyonu) uygulayalım. Böylece x_2, bir denklem hariç, tüm denklemlerden yok edilir. Bütün bu işlemler bittikten sonra birinci adım bitmiş olup, ikinci adım aşağıdaki tablo ile başlar:

6	0	0	0	-2	42
13/7	0	1	0	-1/7	9
11/7	0	0	1	-3/7	6
1	7	0	0	1	21
0	3	9	6	0	

Şimdi, ortadaki durum başlangıçtaki tabloya benzerdir. Temel olmayan değişkenler x_1 ve x_5 dir. x_5 deki herhangi bir artış, $F(x)$ de azalmaya neden olur. O halde, yeni temel değişken x_1 olmalıdır. Aynı kuralla, x_5 i 0 da sabit tutup x_1 in mümkün olduğu kadar artmasına izin verirsek;

$$0 \le x_3 = 9 - (13/7)x_1$$
$$0 \le x_4 = 6 - (11/7)x_1$$
$$0 \le 7x_2 = 21 - x_1$$

kısıtlarından

$$x_1 \le 63/13 \qquad x_1 \le 42/11 \qquad x_1 \le 21$$

elde edilir. O halde yeni temel değişken x_1, en fazla 42/11 ye kadar artabilir. x_3, x_4 ve x_2 nin karşılık gelen yeni değerleri tablodan veya yukarıdaki kısıtlardan heaplanabilir. Böylece yeni x vektörü:

$$x = (42/11, \quad 27/11, \quad 21/2, \quad 0, \quad 0)^T$$

bulunur. Yeni temel olmayan değişkenler şimdi x_4 ve x_5 dir. (v) inci kuralı sağlamak için $x_1 = (7/11)(6 - x_4)$ alırsak,

$$F(x) = 6x_1 - 2x_5 + 42$$
$$= (42/11)(6 - x_4) - 2x_5 + 42$$
$$= -(42/11)x_4 - 2x_5 + 714/11$$

bulunur. F deki her iki katsayı da negatif olduğu için, üçüncü tabloyu tamamlamak gereksizdir. Bu ise, x_4 ve x_5 temel olmayan değişkenlerinin her ikisinin de $F(x)$i azaltmadan temel değişken olamayacağından dolayı, şimdiki x vektörünün bir çözüm olduğunu gösterir. O halde orijinal problemde maksimum değer $F(42/11, 27/11) = 630/11$ dir.

2.6. Özet

Yukarıdaki örnek ve açıklamalar ışığında, tablo yönteminin herhangi bir adımıdaki tabloda yapılması gereken işi aşağıdaki gibi özetleyebiliriz:

(i) Eğer F deki bütün katsayılar (yani, tablodaki en üst satır) ≤ 0 ise o anki x çözümdür.

(ii) F deki katsayısı pozitif olan ve mümkün olduğu kadar büyük olan temel olmayan değişken seçilir. Bu değişken yeni x_j temel değişkenidir.

(iii) Her bir b_i, bulunduğu satırdaki yeni temel değişkenin katsayısı a_{ij} ile bölünür. Yeni temel değişkene atanan değer bu oranların en küçüğüdür. Böylece eğer b_k / a_{kj} en küçük değer ise $x_j = b_k / a_{kj}$ alınır.

(iv) a_{kj} pivot elemanını kullanarak, Gauss eliminasyon ile A nın j kolonunda 0 lar oluşturulur.

3. UYGULAMA PROBLEMLERİ VE BİLGİSAYAR KODU

Bu bölümde, Bölüm 2 de matematiksel çözüm yapısı verilen lineer programlama problemlerinin gerçek hayat problemlerine uygulamaları somut örneklerle incelenecektir. Bölüm sonunda ise problemlerin çözümleri için C dilinde yazılmış bir bilgisayar kodu verilecektir.

3.1. Fabrika Üretim Programlaması (Maksimum problemi)

Bir oyuncak şirketi askerler ve trenler olmak üzere iki tip tahta oyuncak üretiyor. 27 dolara satılan bir oyuncak asker için 10$ değerinde ham madde kullanıyor. Üretilen her oyuncak asker şirketin işçilik değişkenini arttırıp, maliyete ilave olarak 14$ ekliyor. 21$'a satılan bir oyuncak tren için 9$ değerinde ham madde kullanılıyor. Yapılan her oyuncak tren şirketin işçilik değişkenini artırıp, maliyete ilave olarak 10$ ekliyor. Tahta askerlerin ve trenlerin üretiminde marangozluk ve cilalama olmak üzere iki tip işçiliğe gereksinim duyuluyor. Bir askerin üretimi için 2 saat cilalama ve 1 saat marangozluk gerekmektedir. Bir trenin yapımında ise 1 saat cilalama ve 1 saat marangozluk gerekiyor. Şirket haftalık ihtiyacı olan ham maddeleri bulabildiği halde ancak 100 saat cilalama ve 80 saat marangozluk yapabiliyor. Trenler için talep sınırsız olduğu halde haftada en fazla 40 asker satılıyor. Şirketin haftalık karını (gelirler - giderler) maksimum yapmak için bir matematiksel model kurunuz.

Çözüm.

Karar Değişkenleri : Lineer programlama modelinde karar değişkenleri, üzerinde karar verilmesi gereken iki veya daha fazla sayıdaki faaliyeti ifade eder. Açık olarak, şirket her hafta kaç tane oyuncak asker ve trenin üretileceğine karar vermelidir. Bu düşünceyle,

x_1 : her hafta üretilen asker sayısı

x_2 : her hafta üretilen tren sayısı

olarak tanımlansın.

Objektif (Amaç) Fonksiyonu: Herhangi bir lineer programlama probleminde, karar verici, karar değişkenlerinin bir fonksiyonunu (genellikle gelir veya karı) maksimum veya (genellikle maliyeti) minimum yapmak ister. Maksimum veya minimum yapılmak istenen bu fonksiyona *objektif fonksiyonu* denir.

Şirket problemi için sabit maliyetlerin (kiralama ve sigorta gibi) x_1 ve x_2 değerlerine bağlı olmadığını kabul edelim. Böylece şirket ,

(Haftalık gelir) - (Ham maddenin satın alınış maliyeti) - (Diğer değişken maliyetler)

değerlerini nasıl maksimum yapacağı üzerine yoğunlaşmalıdır.

Şirketin haftalık geliri ve maliyeti x_1 ve x_2 karar değişkenleri cinsinden ifade edilebilir. Şirket için satılacak miktardan daha fazla oyuncak asker üretmek gereksizdir. O halde üretilen bütün oyuncakların satılacağını kabul edersek,

$$\begin{aligned} \text{Haftalık gelir} &= \text{Askerden sağlanan haftalık kazanç} + \text{Trenden sağlanan haftalık kazanç} \\ &= \left(\frac{\text{dolar}}{\text{asker}}\right)\left(\frac{\text{askerler}}{\text{hafta}}\right)+\left(\frac{\text{dolar}}{\text{tren}}\right)\left(\frac{\text{trenler}}{\text{hafta}}\right) \\ &= 27x_1 + 21x_2 \end{aligned}$$

olur. Ayrıca,

$$\text{Haftalık ham madde maliyeti} = 10x_1 + 9x_2$$

$$\text{Diğer haftalık değişken maliyetleri} = 14x_1 + 10x_2$$

olur. Böylece şirket,

$$(27x_1 + 21x_2) - (10x_1 + 9x_2) - (14x_1 + 10x_2) = 3x_1 + 2x_2$$

değerini maksimum yapmaya çalışıyor. O halde şirket $3x_1 + 2x_2$ ‘yi maksimum yapacak x_1 ve x_2 değerlerini bulmalıdır. Böylece, matematiksel olarak amaç fonksiyonu;

$$\text{maks } z = 3x_1 + 2x_2 \qquad (3.1)$$

olur. Amaç fonksiyonundaki değişkenin katsayısına değişkenin ***amaç fonksiyonu katsayısı*** denir. Örneğin; x_1 için amaç fonksiyonunun katsayısı 3, x_2 için de 2'dir. Bu örnekte, (ve diğer birçok problemde) her değişkenin amaç fonksiyonu katsayısı, en basit anlamda değişkenin şirketin karına olan katkısıdır.

Kısıtlar: x_1 ve x_2 arttığı sürece şirketin amaç fonksiyonunun değeride artar. Böylece, eğer şirket x_1 ve x_2 değerlerini seçmede özgür olursa bu değerleri çok yüksek tutarak karını keyfi bir şekilde büyütebilir. Fakat x_1 ve x_2 değerleri aşağıdaki üç kısıtlama ile sınırlıdır.

Kısıt 1: Her hafta 100 saatten fazla cilalama yapılamıyor.

Kısıt 2: Her hafta 80 saatten fazla marangozluk yapılamıyor.

Kısıt 3: Sınırlı talepten dolayı, her hafta en fazla 40 oyuncak asker üretiliyor.

Elde bulunan ham madde miktarı sınırsız olarak kabul edildiğinden bu konuda kısıtlama yoktur.

Şirket probleminin matematiksel modelini kurmak için bir sonraki basamak Kısıt 1-2-3'ü x_1 ve x_2 karar değişkenleri cinsinden ifade etmektir.

Kısıt 1'i x_1 ve x_2 karar değişkenleri cinsinden ifade edelim:

$$\begin{aligned}\frac{\text{toplam cilalama saati}}{\text{hafta}} &= \left(\frac{\text{cilalama saati}}{\text{asker}}\right)\left(\frac{\text{üretilen askerler}}{\text{hafta}}\right) \\ &+\left(\frac{\text{cilalama saati}}{\text{tren}}\right)\left(\frac{\text{üretilen trenler}}{\text{hafta}}\right) \\ &= 2(x_1)+1(x_2)=2x_1+x_2\end{aligned}$$

olup böylece Kısıt 1

$$2x_1+x_2 \le 100 \tag{3.2}$$

şeklini alır. (3.2) deki her bir terimin biriminin bir haftalık cilalama saati olduğuna dikkat edelim. Bir kısıtın kabul edilebilir olması için kısıttaki her terimin aynı birimden olması gerekir. Aksi halde bu elmalarla armutları toplamak gibi olur ki bu durumda kısıt bir anlam ifade etmez. Benzer şekilde Kısıt 2'yi x_1 ve x_2 karar değişkenleri cinsinden ifade edersek

$$\begin{aligned}\frac{\text{toplam marangozluk saati}}{\text{hafta}} &= \left(\frac{\text{marangozluk saati}}{\text{asker}}\right)\left(\frac{\text{üretilen askerler}}{\text{hafta}}\right) \\ &+ \left(\frac{\text{marangozluk saati}}{\text{tren}}\right)\left(\frac{\text{üretilen trenler}}{\text{hafta}}\right) \\ &= 1(x_1) + 1(x_2) = x_1 + x_2\end{aligned}$$

olup, böylece Kısıt 2

$$x_1 + x_2 \leq 80 \tag{3.3}$$

ile ifade edilir.

Bir haftada en fazla 40 oyuncak asker satılabileceğinden, haftalık oyuncak asker üretimini 40 ile sınırlıyoruz. Bu da aşağıdaki kısıtı verir:

$$x_1 \leq 40 \tag{3.4}$$

Böylece (3.2), (3.3) ve (3.4) şirketin lineer programlama probleminin kısıtlarının karar değişkenleri cinsinden ifadesidir. Kısıtlardaki karar değişkenlerinin katsayılarına genellikle "teknolojik katsayılar" denilmektedir. Bunun nedeni, bu katsayıların sıklıkla farklı ürünlerin üretiminde kullanılan teknolojiyi yansıtmalarındandır. Örneğin, (3.3)'deki x_2 nin teknolojik katsayısı 1; bir oyuncak askerin yapımında 1 saatlik marangozluk işine ihtiyaç duyulduğunu belirtmektedir.

İşaret Sınırlamaları: Lineer programlama probleminin kurulmasını tamamlamak için son aşama her karar değişkeni için aşağıdaki sorunun yanıtlanmasıdır:

Karar değişkeni sadece negatif olmayan değerleri mi kabul ediyor yoksa hem pozitif hem de negatif değerler alabilir mi?

Eğer x_i karar değişkeni sadece negatif olmayan değerleri kabul ediyorsa $x_i \geq 0$ işaret kısıtlamasını ekleyeceğiz. Eğer x_i değişkeni hem pozitif hemde negatif (yada sıfır) değerlerini kabul ediyorsa, x_i'nin işaret kısıtlaması olmadığını söyleyeceğiz. Şirket problemi için, $x_1 \geq 0$ ve $x_2 \geq 0$ olduğu açıktır. Bazı problemler de karar değişkenleri için işaret sınırlaması olmayabilir. Örneğin x_i bir firmanın nakit dengesini temsil etsin. Eğer firmanın borcu, elindeki parasından daha fazla ise x_i negatif bir değer alır. Bu durumda x_i'nin işaret sınırlaması olmadığını kabul edeceğiz.

$x_1 \geq 0$, $x_2 \geq 0$ işaret sınırlamalarını, (3.1) amaç fonksiyonu ve (3.2),(3.3),(3.4) kısıtları ile birleştirerek aşağıdaki optimizasyon problemini elde ederiz:

$$2x_1 + x_2 \leq 100 \quad \text{(cilalama kısıtı)}$$

$$x_1 + x_2 \leq 80 \quad \text{(marangozluk kısıtı)}$$

$$x_1 \leq 40 \quad \text{(askerlere talep kısıtı)}$$

$$x_1 \geq 0 \quad \text{(işaret kısıtı)} \tag{3.5}$$

$$x_2 \geq 0 \quad \text{(işaret kısıtı)} \tag{3.6}$$

kısıtları altında

$$z = 3x_1 + 2x_2$$

fonksiyonunu maksimum yapan x_1 ve x_2 değerlerini bulunuz.

Uygun Çözüm Bölgesi ve Optimum Çözüm: Bu problemin çözümünü öncelikle Simpleks algoritması ile oluşturup, problem iki değişkenli olduğu için çözümü daha sonra grafiksel olarak elde edeceğiz.

Bölüm 2 deki yöntemle ilk tablo

3	2	0	0	0	0
2	1	1	0	0	100
1	1	0	1	0	80
1	0	0	0	1	40
0	0	100	80	40	

olup, ilk başlangıç vektörü $x = (0, 0, 100, 80, 40)^T$ olarak alınabilir. $F(x) = 3x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5$ olduğundan, $F(x)$ i x_1 deki birim artış daha fazla artıracaktır. O halde x_2 yi sıfırda sabit tutup, x_1 in mümkün olduğunca fazla artmasına izin vereceğiz. Böylece:

$$0 \le x_3 = 100 - 2x_1$$

$$0 \le x_4 = 80 - x_1$$

$$0 \le x_5 = 40 - x_1$$

olur. Bu kısıtlar sırası ile, $x_1 \le 50$, $x_1 \le 80$, $x_1 \le 40$ olmasını gerektirir. Bunların her üçünü sağlayacak şekilde, $x_1 \le 40$ olmalıdır. O halde, x_1 i 40 a kadar artırabiliriz. x_3, x_4 ve x_5 için karşılık gelen yeni değerler de verilen üç kısıttan elde edilir. Böylece, yeni x vektörü;

$$x = (40,\ 0,\ 20, 40,\ 0)^T$$

ve $F(x) = 3(40 - x_5) + 2x_2 = 0x_1 + 2x_2 + 0x_3 + 0x_4 - 3x_5 + 120$ olur. O halde ikinci tablo

0	2	0	0	-3	120
2	1	1	0	0	100
1	1	0	1	0	80
1	0	0	0	1	40
40	0	20	40	0	

dır. Gauss eliminasyon uygulanırsa

0	2	0	0	-3	120
0	1	1	0	-2	20
0	1	0	1	-1	40
1	0	0	0	1	40
40	0	20	40	0	

olup, birinci adıma benzer şekilde bu kez x_2 yi temel değişken yaparsak

$$0 \le x_3 = 20 - x_2$$

$$0 \le x_4 = 40 - x_2$$

$$0 \le x_1 = 40$$

ve böylece x_2 yi 20 ye kadar artırabiliriz. x_1, x_3 ve x_4 için karşılık gelen yeni değerler verilen üç kısıt cinsinden elde edilirse, yeni x vektörü

$$x = (40,\ 20,\ 0, 20,\ 0)^T$$

ve $F(x) = 2(20 + 2x_5 - x_3) - 3x_5 + 120 = -2x_3 + x_5 + 160$ olur. Böylece,

0	0	-2	0	1	160
0	1	1	0	-2	20
0	1	0	1	-1	40
1	0	0	0	1	40
40	20	0	20	0	

olur. Gauss eliminasyon uygulanırsa

0	0	-2	0	1	160
0	1	1	0	-2	20
0	0	-1	1	1	20
1	0	0	0	1	40
40	20	0	20	0	

bulunur. Şimdi x_5 temel değişken yapılırsa

$$0 \le x_2 = 20 + 2x_5$$

$$0 \le x_4 = 20 - x_5$$

$$0 \le x_1 = 40 - x_5$$

olur. Bu kısıtlar sırası ile, $x_5 \ge -10$, $x_5 \le 20$, $x_5 \le 40$ olmasını gerektirir. Kural gereği birinci eşitsizlik gözardı edileceğinden $x_5 \le 20$ olmalıdır. O halde, x_5 i 20 ye kadar artırabiliriz. x_1, x_2 ve x_4 için karşılık gelen yeni değerler de verilen üç kısıttan yazılırsa, yeni x vektörü;

$$x = (20,\ 60, 0, 0,\ 20)^T$$

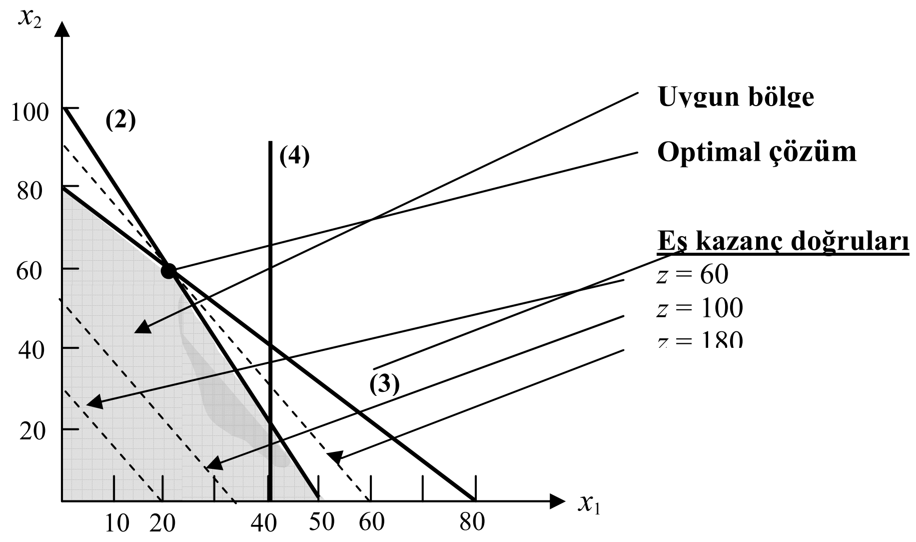
ve

$$F(x) = -2x_3 + (20 + x_3 - x_4) + 160 = -x_3 - x_4 + 180$$

olur. Son ifadede her iki değişkenin katsayısı negatif olduğundan maksimum değer $x_3 = x_4 = 0$ da oluşacağından, $x_1 = 20$ ve $\mathrm{x}_2 = 60$ olup ve böylece amaç fonksiyonunun değeri $z = 3x_1 + 2x_2 = 180$ dolardır. Uygun çözüm bölgesi Şekil 3.1 de gösterilmiştir.

(x_1= 20 , x_2=60)'ın şirket probleminin optimum çözümü olduğunu söylediğimizde, uygun çözüm bölgesindeki hiçbir noktanın amaç fonksiyonu değerinin 180'ni geçemeyeceğini söylemiş oluyoruz. Şirket haftada 20 oyuncak asker ve 60 oyuncak tren üretirse karını maksimum yapacaktır. Bu durumda haftalık kazanç 180 dolar – haftalık sabit giderler olacaktır. Eğer örneğin sabit giderleri haftada 100 dolar ise bu durumda şirketin haftalık kazancı 180-100 = 80 dolar olur.

Bu örnekte olduğu gibi iki değişkenli (düzlemsel) veriler içeren problemleri grafikler yardımı ile de çözebiliriz. Problemin (3.2)-(3.6) kısıtlarını sağlayan uygun çözüm bölgesi Şekil 3.1 de gösterilmiştir. O halde amaç fonsiyonu (3.1) i maksimum yapan nokta bu bölge içindedir. (Bu bölge konveks olduğundan optimal çözümün bölgenin uç noktalarından oluşacağını not edelim.) Optimal çözümü bulmak için öncelikle üzerindeki tüm noktaların aynı z değerini verdiği bir doğru çizelim (kopuk çizgili doğrular). Bir maksimum probleminde bu tür doğrulara ***eş kazanç*** doğruları denir. Eğer problem bir minimum problemi ise bu tür doğrulara ***eş gider*** doğruları denir. Bir eş kazanç doğrusu çizmek için uygun bölgeden bir nokta seçelim. Örneğin bu nokta (20,0) olsun. Bu noktada $z = 3(20) + 2(0) = 60$ olur. O halde (20,0) noktası $z = 3x_1 + 2x_2 = 60$ doğrusu üzerindedir. Tüm eş kazanç eğrileri $3x_1 + 2x_2 = sabit$ formunda olduklarından, diğer eş kazanç eğrileri bu doğruya paralele olan eğrilerdir. Böylece z değeri artacak şekilde paralel doğrular çizildiğinde, uygun bölgeye değen en son doğrununun uygun bölgeyi kestiği nokta G, aranılan maksimum çözüm değerini verecektir. Bu noktada, $2x_1 + x_2 = 100$ ve $x_1 + x_2 = 80$ doğruları kesiştiğinden, G noktası bu iki denklemin ortak çözümü olan ((20, 60) noktası olup, böylece çözüm $z = 3(20) + 2(60) = 180$ dir.



Şekil 3.1. Fabrika probleminin grafik çözümü

3.2. Reklam Giderleri (Minimum Problemi)

Bir otomobil firması lüks arabalar ve jeepler üretmektedir. Şirket müşterilerinin büyük olasılıkla yüksek gelirli kadınlar ve erkekler olacağına inanıyor. Bu kitlelere ulaşmak için firma televizyon reklamlarına girişimde bulunuyor ve iki tip program içinde 1 dakikalık kısa reklamlar satın almayı düşünüyor: Komedi programları ve futbol maçları. Televizyondaki her komedi programı 7 milyon yüksek gelirli kadın ve 2 milyon yüksek gelirli erkek tarafından izleniyor. Televizyondaki her futbol programı ise 2 milyon yüksek gelirli kadın ve 12 milyon yüksek gelirli erkek tarafından izleniyor. Bir komedi programı içinde yayınlanacak 1 dakikalık reklam için 50,000 dolarlık, futbol programı içinde yayınlanacak 1 dakikalık reklam için ise 100,000 lık bir maliyet gerekiyor. Firma, reklamların en az 28 milyon yüksek gelirli kadın ve 24 milyon yüksek gelirli erkek tarafından görülmesini istiyor. Oto firmasının reklam giderlerini nasıl minimum maliyette tutacağına dair bir doğrusal programlama problemi geliştiriniz.

Çözüm.

Firma komedi programları ve futbol maçları için ne kadar reklam satın alacağına karar vermelidir. O halde karar değişkenleri,

$x_1 =$ satın alınan 1 dakikalık komedi programı reklamı sayısı

$x_2 =$ satın alınan 1 dakikalık futbol maçı reklamı sayısı

Firma toplam reklam giderlerini .(bin dolar bazında) minimum yapmak istiyor.

Toplam reklam gideri = komedi programı reklam gideri + futbol maçları reklam gideri

$$= \left(\frac{\text{gider}}{\text{komedi reklamı}} \right) \left(\begin{matrix} \text{toplam} \\ \text{komedi reklamı} \end{matrix} \right) + \left(\frac{\text{gider}}{\text{futbol reklamı}} \right) \left(\begin{matrix} \text{toplam} \\ \text{futbol reklamı} \end{matrix} \right)$$

$$= 50x_1 + 100x_2$$

olur. Böylece firmanın amaç fonksiyonu,

$$\min z = 50x_1 + 100x_2 \tag{3.7}$$

dir. Firma aşağıdaki kısıtlarla karşı karşıya kalacaktır.

Kısıt 1: Reklamlar mutlaka en az 28 milyon yüksek gelirli kadına ulaşmalı.

Kısıt 2: Reklamlar mutlaka en az 24 milyon yüksek gelirli erkeğe ulaşmalı.

Yazım kolaylığı için yüksek gelirli kadın sayısını (milyon bazında) YGK ve yüksek gelirli erkek sayısını da (milyon bazında) YGE ile gösterirsek,

$$\text{YGK} = \left(\frac{\text{YGK}}{\text{komedi reklamı}} \right) \left(\begin{matrix} \text{toplam} \\ \text{komedi reklamı} \end{matrix} \right) + \left(\frac{\text{YGK}}{\text{futbol reklamı}} \right) \left(\begin{matrix} \text{toplam} \\ \text{futbol reklamı} \end{matrix} \right)$$

$$= 7x_1 + 2x_2$$

$$\text{YGE} = \left(\frac{\text{YGE}}{\text{komedi reklamı}}\right)\left(\begin{array}{c}\text{toplam}\\ \text{komedi reklamı}\end{array}\right) + \left(\frac{\text{YGE}}{\text{futbol reklamı}}\right)\left(\begin{array}{c}\text{toplam}\\ \text{futbol reklamı}\end{array}\right)$$

$$= 2x_1 + 12x_2$$

olup, x_1 ve x_2 cinsinden Kısıt 1' i

$$7x_1 + 2x_2 \geq 28 \tag{3.8}$$

ve Kısıt 2'yi de

$$2x_1 + 12x_2 \geq 24 \tag{3.9}$$

şeklinde ifade edebiliriz.

Ayrıca işaret sınırlamaları $x_1 \geq 0$ ve $x_2 \geq 0$ da gereklidir. Böylece, firmanın doğrusal programlaması,

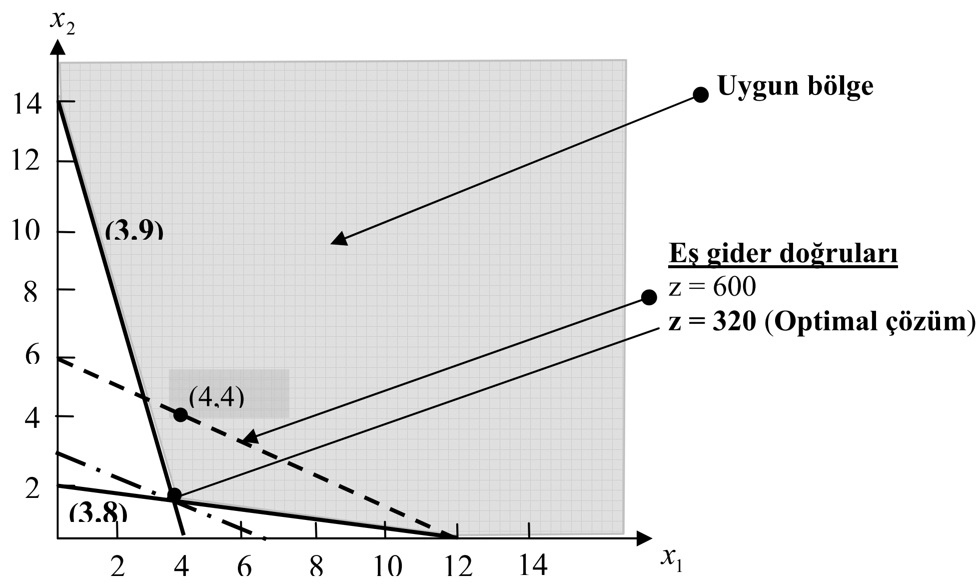
$$\begin{aligned} 7x_1 + 2x_2 &\geq 28 \quad \text{(YGK)} \\ 2x_1 + 2x_2 &\geq 24 \quad \text{(YGE)} \\ x_1, x_2 &\geq 0 \end{aligned}$$

kısıtları altında

$$\min z = 50x_1 + 100x_2$$

olur.

Bu problem, karar vericilerin; belli gereksinimlerin maliyetini minimum yapmak istedikleri tipik bir doğrusal programlama probleminin uygulamasıdır. Şimdi, bu doğrusal programlamanın grafik çözümü için, uygun çözüm bölgesini çizerek, çözümü grafiksel olarak analiz edelim.



Şekil 3.2. Reklam giderleri grafik çözümü (minimum problemi)

Kısıtların oluşturduğu bölge Şekil 3.2. de gösterilen taralı bölgedir. Oyuncak firması probleminde olduğu gibi, bu problemin de bir konveks uygun çözüm bölgesi vardır. Fakat bu problemin uygun çözüm bölgesi, oyuncak şirketi probleminden farklı olarak, en az bir değişkenin değeri keyfi büyüklükte değerler alabilecek noktalar içeriyor. Bu şekildeki uygun çözüm bölgesine "*sınırsız uygun çözüm bölgesi*" denir.

Firma, toplam reklam maliyetini minimum yapmak istediğinden, problemin optimal çözümü uygun çözüm bölgesinde en küçük z değerini alan noktadır. Optimal sonucu bulmak için, uygun çözüm bölgesi ile kesişen eş gider doğrusunu çizmemiz gerekmektedir. Eş gider doğrusu, üzerindeki bütün noktaların aynı z değerine (ya da aynı maliyet) sahip olduğu herhangi bir doğrudur. Uygun bölgedeki her hangi bir noktadan geçen, (örneğin $x_1 = 4,\ x_2 = 4$ noktasından geçen) eşgider doğrusunu keyfi olarak seçiyoruz. Bu nokta için, $z = 50(4) + 100(4) = 600$ olur. Böylece, bir eşgider doğrusu $z = 50x_1 + 100x_2 = 600$ olur. Diğer eşgider doğruları, uygun çözüm bölgesinden geçen ve bu doğruya paralel olan doğrulardır. z değeri azalacak şekilde, uygun bölgeden geçen en son eşgider doğrusu böylece (3.8) ve (3.9) a ait doğruların kesiştiği, yani $7x_1 + 2x_2 = 28$ ve $2x_1 + 12x_2 = 24$ doğrularının kesiştiği ($x_1 = 3{,}6$ ve $x_2 = 1{,}4$) noktasından geçen doğru olup, bu noktada minimum z değeri elde edilmiş olur. Böylece, optimal z değeri, $z = 50(3,6) + 100(1,4) = 320 = 320,000\$$ olur.

3.3. Diyet Problemi (Minimum Problemi)

Bir kişinin tükettiği tüm gıdaların dört "temel besin grubu" nun (çikolatalı kek, dondurma, soda ve peynirli kek) birinden gelmesi istenmektedir. Kişi, bu gruplardan kakaolu kek, çikolatalı dondurma, kola ve ananaslı-peynirli kek tüketmek istemektedir. Her bir kakaolu kekin maliyeti 800 bin TL, bir külah çikolatalı dondurmanın maliyeti 600 bin TL, bir şişe kolanın maliyeti 700 bin TL ve ananaslı-peynirli kekin bir diliminin maliyeti de 1 milyon TL dir. Kişi her gün mutlaka en az 500 kalori, 40 gr çikolata, 30 gr şeker ve 20 gr yağ tüketecektir. Her gıdanın birim başına düşen besin içeriği Tablo 3.1 de gösterilmiştir. Buna göre kişinin günlük besin gereksinimini en az maliyetle sağlayabileceği bir doğrusal programlama modelini formüle ediniz.

Çözüm.

İlk olarak karar değişkenlerimizi oluşturalım. Her bir gıdadan günlük ne kadar tüketilmelidir? Böylece karar değişkenlerini,

$x_1 =$ bir günde tüketilen kakaolu kek sayısı

$x_2 =$ bir günde tüketilen dondurmanın külah sayısı

$x_3 =$ bir günde tüketilen kolanın şişe sayısı

$x_4 =$ bir günde tüketilen ananaslı-peynirli keklerin dilim sayısı

olacak şekilde oluşturabiliriz.

	Kalori	Çikolata (gr)	Şeker (gr)	Yağ (gr)
Kakolu Kek	400	30	20	20
Dondurma (1 Külah)	200	20	20	10
Kola (1 şişe)	150	0	40	10
Peynirli kek (1 dilim)	500	0	40	15

Tablo 3.1. Diyet için besin değerleri

Amacımız diyetin maliyetini minimuma indirmektir. Diyetin toplam maliyeti aşağıdaki gibi verilebilir.

(Diyetin toplam maliyeti) = (Kakaolu keklerin maliyeti)
+ (Dondurmaların maliyeti)
+ (Kolanın maliyeti)
+ (Peynirli kekin maliyeti)

Diyetin toplam maliyetinin değeri için (bin TL bazında), örneğin;

$$\text{Kolanın maliyeti}=\left(\frac{\text{fiatı}}{\text{bir şişe kola}}\right)\left(\begin{array}{c}\text{tüketilen}\\ \text{şişe sayısı}\end{array}\right)=700x_3$$

olacağından, benzer şekilde diğerlerinin maliyetleri de yazılırsa

$$\text{Toplam diyet maliyeti}=800\text{x}_1+600x_2+700x_3+1000x_4$$

elde edilir. Böylece amaç fonksiyonu

$$\text{Minimum: } z=800\text{x}_1+600x_2+700x_3+1000x_4$$

bulunur. Karar değişkenleri aşağıdaki kısıtları sağlamalıdır.

Kısıt 1: Alınan günlük kalori en az 500 kalori olmalıdır.
Kısıt 2: Alınan günlük çikolata en az 40 gr olmalıdır.
Kısıt 3: Alınan günlük şeker en az 30 gr olmalıdır.
Kısıt 4: Alınan günlük yağ en az 20 gr olmalıdır.

Kısıt 1 i karar değişkenleri cinsinden ifade etmek için;

(Alınan günlük kalori) = (Kakaolu kekteki kalori)
+ (Çikolatalı dondurmadaki kalori)
+ (Koladaki kalori)
+ (Ananaslı-peynirli kekteki kalori)

olacağından ve örneğin kakaolu kek tüketimindeki kalori;

$$\text{Kakolu kekteki kalori=}\left(\frac{\text{kalori}}{\text{bir kek}}\right)\left(\begin{array}{c}\text{tüketilen}\\ \text{kek sayısı}\end{array}\right)=400x_1$$

olacağından, sonuç olarak

$$\text{Alınan günlük kalori} = 400x_1 + 200x_2 + 150x_3 + 500x_4$$

olup, böylece

Kısıt 1:

$$400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500 \quad \text{(Kalori kısıtı)} \qquad (3.10)$$

ve benzer şekilde,

Kısıt 2:

$$30x_1 + 20x_2 \geq 40 \quad \text{(Çikolata kısıtı)} \qquad (3.11)$$

Kısıt 3:

$$20x_1 + 20x_2 + 40x_3 + 40x_4 \geq 30 \quad \text{(Şeker kısıtı)} \qquad (3.12)$$

Kısıt 4:

$$20x_1 + 40x_2 + 10x_3 + 15x_4 \geq 20 \quad \text{(Yağ kısıtı)} \qquad (3.13)$$

ve işaret kısıtları $x_i \geq 0 \quad (i = 1,2,3,4)$ elde edilir. O halde diyet için lineer programlama modeli

$$\min z = 800x_1 + 600x_2 + 700x_3 + 1000x_4$$

$$\text{Kısıtlar : }\begin{cases} 400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500 \\ 30x_1 + 20x_2 \geq 40 \\ 20x_1 + 20x_2 + 40x_3 + 40x_4 \geq 30 \\ 20x_1 + 10x_2 + 10x_3 + 15x_4 \geq 20 \\ x_i \geq 0 \end{cases}$$

şeklini alır. Bu lineer programlama için optimal çözüm için grafik yöntem kullanılamayacaktır. O halde problemi Kısım 2.2 deki kurallar yardımı ile bir maksimum problemine dönüştürürsek;

$$\text{Maksimum: } -z = -800x_1 - 600x_2 - 700x_3 - 1000x_4$$

$$\text{Kısıtlar : } \begin{cases} -400x_1 - 200x_2 - 150x_3 - 500x_4 \leq -500 \\ -30x_1 - 20x_2 \leq -40 \\ -20x_1 - 20x_2 - 40x_3 - 40x_4 \leq -30 \\ -20x_1 - 10x_2 - 10x_3 - 15x_4 \leq -20 \\ x_i \geq 0 \end{cases}$$

elde ederiz. Bu problemin simpleks algoritması ile çözümü $x_1 = 1,\ x_2 = 1/2,\ x_3 = 0,\ x_4 = 0$ olup, ve böylece $z = 1100$ olarak elde edilir. O halde istenilen diyet için sadece 1 adet kakaolu kek ve yarım külah çikolatalı dondurma yenilmelidir. Bu minimum maliyetli (1 milyon 100 bin TL) diyetle günlük

400(1) + 200(1/2) = 500 kalori

30(1)+20(1/2) =40 gr çikolata

20(1) + 20(1/2) = 30 gr şeker

20(1) + 10(1/2) =15 gr yağ

alınmış olur.

3.4. İş Planlama Problemi

Lineer programlama problemlerinin çoğu uygulaması iş gücü gereksinimlerini sağlayacak şekilde minimum gider yöntemi belirlemeyi içermektedir. Aşağıdaki örnek bu tür uygulamalrın çoğunda ortak olan temel özellikleri içermektedir.

Bir posta firması haftanın farklı günlerinde, tam gün çalışacak farklı sayıda işçiye gereksinim duymaktadır. Her gün için tam zamanlı çalışmasına gereksinim duyulan en az işçi sayısı şu şekildedir: P.tesi: 17, Salı: 13, Çarşamba: 15, Perşembe: 19, Cuma: 14, C.tesi: 16, Pazar: 11. Işçilerin çalışmasında ele alınacak temel sendika yasaları şu şekildedir: Bir işçi aralıksız 5 gün çalışacak ve sonra iki gün dinlenecektir. Örneğin; Pazartesiden Cumaya kadar çalışan bir işçi mutlaka Cumartesi ve Pazar günü tatil yapmalıdır. Posta firması günlük ihtiyaçlarını sadece tam zamanlı işçileri çalıştırarak sağlamak istemektedir. Postanenin işe alması gereken en az işçi miktarının bulunmasını sağlayan bir lineer programlama problemi modelini kurunuz.

Çözüm. Bu problemin doğru olarak modelini vermeden önce, birçoğumuzun yapabileceği bir yanlış çözüm modelini tartışarak işe başlayalım. Model için ilk akla gelen, çoğunlukla, 1.gün = Pazartesi , 2.gün = Salı ,v.b. olmak üzere, x_i yi i. günde çalışan işçilerin sayısı olarak tanımlamaktır. Böylece,

Tam gün çalışan işçi sayısı = (Pazartesi günü çalışan işçi sayısı)
+ (Salı günü çalışan işçi sayısı)
:
+ (Pazar günü çalışan işçi sayısı)

olup, amaç fonksiyonu

$$\min z = x_1 + x_2 + ... + x_6 + x_7$$

şeklindedir. Her gün çalışmak için yeterli sayıda tam zamanlı işçi olmasını sağlamak için $x_i \geq$ (i günde gerekli olan işçi sayısı) kısıtını da eklemek gerekir. Örneğin; Pazartesi günü için $x_1 \geq 17$ kısıtını ekleyelim. $x_i \geq 0$ $(i = 1,2,...,7)$ işaret kısıtlamalarını da eklersek aşağıdakine benzer bir lineer programlama problemini elde ederiz.

$$\text{Minimum: } z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

$$\text{Kısıtlar: } \begin{cases} x_1 & \geq 17 \\ x_2 & \geq 13 \\ x_3 & \geq 15 \\ x_4 & \geq 19 \\ x_5 & \geq 14 \\ x_6 & \geq 16 \\ x_7 & \geq 11 \\ x_i & \geq 0 \quad (i = 1, 2, ..., 7) \end{cases}$$

Bu formülasyonda en az iki hata vardır. Birincisi, amaç fonksiyonu tam zamanlı çalışan işçilerin sayısını vermez. Mevcut olan amaç fonksiyonu her işçiyi bir kere saymak yerine 5 kez saymaktadır. Örneğin, Pazartesi günü çalışmaya başlayan her işçi Pazartesi gününden Cuma gününe kadar çalışmaktadır, ve bahsedilen bu işçi x_1, x_2, x_3, x_4 ve x_5'e dahildir. Ikinci olarak $x_1, x_2, ..., x_7$ değişkenleri birbirleri ile ilişkilidir ve değişkenler arasındaki ilişki yukarıdaki kısıtların kümesi ile elde edilemez. Örneğin; Pazartesi günü çalışan bazı işçiler (x_1 işçileri) Salı günü de çalışacaklardır. Bu x_1 ve x_2 nin ilişkili olduğu anlamına gelmektedir. Fakat yukarıdaki kısıtlar x_1 in değerinin x_2 nin değerine bir etkisi olduğunu belirtmemektedir.

Bu problemi doğru olarak formüle etmenin anahtarı; firmanın temel kararının her gün kaç kişinin çalışıyor olması değil, haftanın her günü kaç kişinin çalışmaya başlayacağıdır. O halde bu şekilde düşünürsek,

$$x_i = i \text{ . günde çalışmaya başlayan işçi sayısı}$$

olmalıdır. Örneğin; x_1 = Pazartesi günü çalışmaya başlayan işçi sayısı (Bu kişiler Pazartesi gününden Cuma gününe kadar çalışıyorlar) gibi.

Uygun şekilde tanımlanan değişkenlerle birlikte, doğru amaç fonksiyonunu ve kısıtları belirlemek kolay olacaktır. Amaç fonksiyonunu tanımlamak için,

Tam gün çalışan işçi sayısı = (Pazartesi günü işe başlayan işçi sayısı)
+ (Salı günü işe başlayan işçi sayısı)
:
+ (Pazar günü işe başlayan işçi sayısı)

alınmalıdır. Her işçi çalışmaya haftanın sadece bir gününde başladığı için, yukarıdaki eşitlik işçileri iki kere saymaz. Böylece, değişkenlerin doğru olarak tanımlanmış haliyle, amaç fonksiyonu,

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

olur. Firma, haftanın her günü yeterli sayıda işçi çalışmasını sağlamak zorundadır. Örneğin; Pazartesi günü en az 17 işçi çalışmalıdır. Pazartesi günü kim çalışmaktadır? Salı veya Çarşamba günü çalışmaya başlayan işçiler dışındaki bütün işçiler. (Salı günü çalışmaya başlayanlar Pazar ve Pazartesi günü çalışmıyor, Çarşamba günü çalışmaya başlayanlar Pazartesi ve Salı günü çalışmıyor v.s.) Böylece, Pazartesi günü en az 17 işçi çalışacaksa

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$$

kısıtı sağlanmalıdır. Benzer şekilde, $x_i \geq 0 \quad (i = 1,2,...,7)$ işaret sınırlamaları ile birlikte diğer altı günün kısıtlarını da yazarsak, aşağıdaki lineer programlama problemini elde etmiş oluruz.

Minimum: $z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

$$\text{Kısıtlar:} \left\{ \begin{array}{llllllll} x_1 & & & +x_4 & +x_5 & +x_6 & +x_7 & \geq 17 \\ x_1 & +x_2 & & & +x_5 & +x_6 & +x_7 & \geq 13 \\ x_1 & +x_2 & +x_3 & & & +x_6 & +x_7 & \geq 15 \\ x_1 & +x_2 & +x_3 & +x_4 & & & +x_7 & \geq 19 \\ x_1 & +x_2 & +x_3 & +x_4 & +x_5 & & & \geq 14 \\ & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \geq 16 \\ & & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \geq 11 \\ & & & & & \multicolumn{3}{l}{x_i \geq 0 \quad (i = 1,2,...,7)} \end{array} \right.$$

Bu minimum probleminine karşılık gelen maksimum problemi

Maksimum: $-z = -x_1 - x_2 - x_3 - x_4 - x_5 - x_6 - x_7$

$$
\text{Kısıtlar:}\begin{cases}
-x_1 & & & -x_4 & -x_5 & -x_6 & -x_7 & \leq -17 \\
-x_1 & -x_2 & & & -x_5 & -x_6 & -x_7 & \leq -13 \\
-x_1 & -x_2 & -x_3 & & & -x_6 & -x_7 & \leq -15 \\
-x_1 & -x_2 & -x_3 & -x_4 & & & -x_7 & \leq -19 \\
-x_1 & -x_2 & -x_3 & -x_4 & -x_5 & & & \leq -14 \\
 & -x_2 & -x_3 & -x_4 & -x_5 & -x_6 & & \leq -16 \\
 & & -x_3 & -x_4 & -x_5 & -x_6 & -x_7 & \leq -11 \\
 & & & & & x_i \geq & 0 & (i = 1, 2, \ldots, 7)
\end{cases}
$$

olup, simpleks yöntemi ile çözüm

$$z = \frac{67}{3}, x_1 = \frac{4}{3}, x_2 = \frac{10}{3}, x_3 = 2, x_4 = \frac{22}{3}, x_5 = 0, x_6 = \frac{10}{3} x_7 = 5$$

Değişkenler tamsayı iken anlam kazanacağından bütün değişkenlerin tam sayı olduğu bir cevap için kesirli değişkenler tamsayıya yuvarlanırsa

$$z = 25,\ x_1 = 2,\ x_2 = 4,\ x_3 = 2,\ x_4 = 8,\ x_5 = 0,\ x_6 = 4,\ x_7 = 5$$

uygun çözümünü elde ederiz.

3.5. Lineer Programlama İçin C Kodu

Bu kısımda en genel anlamda aşağıdaki lineer programlama problemi için bir C kodu üretilecektir. Bilgisayar çözümü üretilecek olan problem:

Maksimum: $z = a_{01}x_1 a_{02}x_2 + \ldots + a_{0n}x_n$

$$
\text{Kısıtlar:}\begin{cases}
a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i \quad (b_i \geq 0) \quad i = 1, \ldots, m_1 \\
a_{j1}x_1 + a_{j2}x_2 + \ldots + a_{jn}x_n \geq b_j \quad (b_j \geq 0) \quad j = m_1 + 1, \ldots, m_1 + m_2 \\
a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n = b_k \quad (b_k \geq 0) \quad k = m_1 + m_2 + 1, \ldots, m_1 + m_2 + m_3 \\
x_i \geq 0 \quad (i = 1, \ldots, n)
\end{cases}
$$

şeklinde verilmelidir. Programda n, m, m_1, m_2, m_3 ve

$$a = \begin{bmatrix} 0 & a_{01} & \cdots & a_{0n} \\ b_1 & a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & & \vdots \\ b_{m_1}+1 & a_{m_1+1,1} & \cdots & a_{m_1+1,n} \\ \vdots & \vdots & & \vdots \\ b_{m_1+m_2+m_3} & a_{m_1+m_2+m_3,1} & \cdots & a_{m_1+m_2+m_3,n} \end{bmatrix}$$

katsayılar matrisi girdi olarak sağlanmalıdır. Program çıktısı olarak **icase** değeri (eğer sonlu bir çözüm var ise 0, amaç fonksiyonu sınırsız ise 1 ve kısıtları sağlayan bir çözüm yok ise -1 değerlerini döndürmekte) ve çözüm matrisi a yı temsil eden **ipsv[j]** ve **izrov[j]** indeks vektörleri döndürülmektedir.

```
#define EPS 1.0e-6
#define FREEALL free_ivector(l3,1,m);free_ivector(l2,1,m);\
        free_ivector(l1,1,n+1);
void simplex(a,m,n,m1,m2,m3,icase,izrov,iposv)
int m,n,m1,m2,m3;
int *icase;
int izrov[],iposv[];
float **a;
{
int i,ip,ir,is,k,kh,kp,m12,nl1,nl2,j;
int *l1,*l2,*l3,*ivector();
float q1,bmax;
void simp1(),simp2(),simp3(),nrerror(),free_ivector();
m=4;n=4;m1=2;m2=1;m3=1;
if(m!=(m1+m2+m3)) nrerror("Simpleks için yanlış kısıt sayısı");
for (i=0;i<=m;i++)
for(j=0;j<=n;j++)
scanf("%f",&a[i][j]);
```

```
l1=ivector(1,n+1);
l2=ivector(1,m);
l3=ivector(1,m);
nl1=m;
for(k=1;k<=n;k++) l1[k]=izrov[k]=k;
nl2=m;
for (i=1;i<=m;i++){
if(a[i+1][1]<0.0) nrerror("Simpleks de kötü girdi tablosu");
l2[i]=i;
iposv[i]=n+i;
}
for (i=1;i<=m2;i++) l3[i]=1;
ir=0;
if(m2+m3){
        ir=1;
        for(k=1;k<=(n+1);k++){
            q1=0.0;
            for(i=m1+1;i<=m;i++) q1+=a[i+1][k];
            a[m+2][k]=-q1;
            }
do{
  simp1(a,m+1,l1,nl1,0,&kp,&bmax);
  if(bmax<=EPS && a[m+2][1]<-EPS){
    *icase=-1;
    printf("durum: %d",*icase);

    for (i=0;i<=m;i++)
        for(j=0;j<=n;j++)
        printf("\n%.2f %.2f %.2f %.2f\n",a[i][j]);

    FREEALL return;
    } else if(bmax<=EPS && a[m+2][1]<=EPS){
        m12=m1+m2+1;
        if(m12<=m){
```

```
         for(ip=m12; ip<=m;ip++){
           if(iposv[ip]==(ip+n)){
            simp1(a,ip,l1,nl1,1,&kp,&bmax);
               if(bmax>0.0)
                    goto bir;
           }
         }
        }
  ir=0;
  --m12;
  if(m1+1<=m12)
        for(i=m1+1;i<=m12;i++)
           if(l3[i-m1]==1)
            for(k=1;k<=n+1;k++)
                 a[i+1][k]=-a[i+1][k];
  break;
 }
 simp2(a,n,l2,nl2,&ip,kp,&q1);
 if(ip==0){
  *icase=-1;
  printf("durum1: %d",*icase);

  for (i=0;i<=m;i++)
       for(j=0;j<=n;j++)
       printf("\n%.2f %.2f %.2f %.2f\n",a[i][j]);

  FREEALL return;
 }
bir:   simp3(a,m+1,n,ip,kp);
  if(iposv[ip]>=(n+m1+m2+1)){
       for(k=1;k<=nl1;k++)
                 if(l1[k]==kp)break;
       --nl1;
       for(is=k;is<=nl1;is++) l1[is]=l1[is+1];
```

```
            a[m+2][kp+1] +=1.0;
            for(i=1;i<=m+2;i++) a[i][kp+1]=-a[i][kp+1];
        } else {
              if(iposv[ip]>=(n+m1+1)){
                        kh=iposv[ip]-m1-n;
                        if(l3[kh]){
                         l3[kh]=0;
                         a[m+2][kp+1] +=1.0;
                         for (i=1;i<=m+2;i++)
                                  a[i][kp+1]= -a[i][kp+1];
                        }
              }
        }
        is=izrov[kp];
        izrov[kp]=iposv[ip];
        iposv[ip]=is;
    }  while (ir);
}
for(;;){
            simp1(a,0,l1,nl1,0,&kp,&bmax);
            if(bmax<=0.0){
                        *icase=0;

                        printf("durum2: %d",*icase);
                        for (i=0;i<=m;i++)
                        for(j=0;j<=n;j++)
                        printf("%f",&a[i][j]);

                        FREEALL return;
            }
            simp2(a,n,l2,nl2,&ip,kp,&q1);
            if(ip==0){
                        *icase=1;
                        printf("durum3: %d",*icase);
```

```
                printf(" %f ",a);
                FREEALL return;
        }
        simp3(a,m,n,ip,kp);
        is=izrov[kp];
        izrov[kp]=iposv[ip];
        iposv[ip]=is;
        }
}

#include <math.h>
void simp1(a,mm,ll,nll,iabf,kp,bmax)
float **a,*bmax;
int mm, ll[],nll,iabf,*kp;
{
int k;
float test;
*kp=ll[1];
*bmax=a[mm+1][*kp+1];
for (k=2;k<=nll;k++){
        if(iabf==0)
                test=a[mm+1][ll[k]+1]-(*bmax);
                else
                test=fabs(a[mm+1][ll[k]+1])-fabs(*bmax);
                if(test>0.0){*bmax=a[mm+1][ll[k]+1];
                *kp=ll[k];
                }
        }
}

#define EPS 1.0e-6
void simp2(a,n,l2,nl2,ip,kp,q1)
int n,l2[],nl2,*ip,kp;
float **a,*q1;
```

```
{
int k,ii,i;
float qp,q0,q;
*ip=0;
for(i=1;i<=nl2;i++){
if(a[l2[i]+1][kp+1]<-EPS){
        *q1=-a[l2[i]+1][1]/a[l2[i]+1][kp+1];
        *ip=l2[i];
        for(i=i+1;i<=nl2;i++){
        ii=l2[i];
        if(a[ii+1][kp+1]<-EPS){
                ii=l2[i];
                q=-a[ii+1][1]/a[ii+1][kp+1];
                if(q<*q1){
                        *ip=ii;
                        *q1=q;
             }else if(q==*q1){
                        for(k=1;k<=n;k++){
                                qp=-a[*ip+1][k+1]/a[*ip+1][kp+1];
                                q0=-a[ii+1][k+1]/a[ii+1][kp+1];
                                if(q0!=qp)break;
                        }
                        if(q0<qp) *ip=ii;
                }
        }
}}}}

void simp3(a,i1,k1,ip,kp)
int i1,k1,ip,kp;
float **a;
{
int kk,ii;
float piv;
piv=1.0/a[ip+1][kp+1];
```

```
for(ii=1;ii<=i1+1;ii++)
if(ii-1 !=ip){
a[ii][kp+1] *=piv;
for(kk=1;kk<=k1+1;kk++)
        if(kk-1 !=kp)
                a[ii][kk] -= a[ip+1][kk]*a[ii][kp+1];
}
for(kk=1;kk<=k1+1;kk++)
        if(kk-1 !=kp) a[ip+1][kk] *=-piv;
        a[ip+1][kp+1]=piv;
}


void nrerror(error_text)
char error_text[];
{
void exit();
fprintf(stderr,"run-time error... \n");
fprintf(stderr,"%s\n", error_text);
fprintf(stderr,"...sistemden çıkıyor \n");
exit(1);  }

int *ivector(nl,nh)
int nl,nh;
{  int *v;
v=(int *)malloc((unsigned) (nh-nl+1)*sizeof(int));
if(!v) nrerror("vekt”rde Allocation hatas□");
return v-nl; }

void free_ivector(v,nl,nh)
int *v,nl,nh;
{
free((char*) (v+nl));
}
```

KAYNAKLAR

Dantzig, G.B. 1963. Linear Programming and Extensions, Princeton Univ. Press.

Dantzig, G.B. 1983. Reminiscences about the origins of linear programming. Mathematical Programming: The State of the Art, Springer Verlag, Berlin, pp.78-86.

Kahn, B.P. 1990. Mathematical Methods for Scientists and Engineers, Linear and Nonlinear Systems, John Wiley&Sons, New York.

Kincaid, D. - Cheney, W. 1996. Numerical Analysis, Brooks/Cole Publ. Co., New York.

Luenberger, D.G. 1984. Introduction to Linear and Nonlinear Programming, Second Edition, Addison-Wesley, California.

Nazareth, J.L. 1987. Computer Solutions of Linear Programs, Monographs on Numerical Analysis, Oxford Science Publ., New York.

Nyhoff, L. - Leestma, S. 1988. FORTRAN 77 for Engineers and Scientists, Macmillan Publishing, New York.

Press, W.H.- Flannery, B.P., Teukolsky, S.A.,Vetterling, W.T. 1988. Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press, New York.

Stewart, G.W. 1973. Introduction to Matrix Computations, Academic Press, NewYork.

Winston, W.L. 1994. Operations Research, Applications and Algorithms, Duxbury Press, Belmont CA.

Press, W.H. - Flannery, B.P. – Teukolsky, S.A. – Vetterling, W.T., 1988. Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press, New York.

Royden, H.L., 1968. Real Analysis, MacMillan Publ., New York.

ÖZGEÇMİŞ

1978 yılında Ankara'da doğdu. İlkokulu Özel Neşe İlkokulu'nda, Ortaokul ve lise öğrenimini de Özel Arı Lisesinde tamamladı. 1996 yılında girdiği Ankara Üniversitesi Fen Fakültesi Matematik Bölümünden 2000 yılında mezun oldu. Aynı yıl Ankara Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim dalı yüksek lisans sınavını kazandı. Halen yüksek lisans öğrenimini sürdürmekte olup, Ankara Üniversitesi Eğitim Bilimleri Fakültesi İlköğretim Bölümünde araştırma görevlisi olarak çalışmaktadır.