YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK AMAÇLI LİNEER PROGRAMLAMA PROBLEMİ VE NETWORK ÜZERİNE BİR UYGULAMA

YÜKSEK LİSANS TEZİ

COŞKUN GÜLER
MATEMATİK MÜHENDİSİ

YÖNETİCİ
Y.DOÇ.DR. MEHMET AHLATÇIOĞLU

İSTANBUL - 1990

TEŞEKKÜR

Bu çalışma konusunu seçmemde yardımcı olan ve çalışmam süresince her türlü desteğini gördüğüm tez yürütücüm Sayın Yrd.Doç.Dr.Mehmet AHLATÇIOĞLU'na, Yüksek lisans öğrenimime başlamam için teşvik eden Matematik Mühendisliği Bölüm Başkanı Sayın Prof.Tahir ŞİŞMAN'a en içten duygularla teşekkür ederim.

İÇİNDEKİLER

ÖZET

Lineer Programlama Problemi, belirli bir amacın gerçeklenme derecesini etkileyen, bazı kısıtlayıcı koşulların lineer eşitlik veya eşitsizlikler halinde verilmesi durumunda, istenen amaca ulaşılabilmesi için kaynakların optimal kullanımını sağlayan matematiksel bir yöntemdir.

Birinci bölümde, tek amaçlı Lineer Programlama problemi üzerinde durulmuştur. Problemin çözümü için en çok kullanılan "Simpleks yöntem" hakkında bilgi verilmiştir.

İkinci bölümde, amaç fonksiyonlarının birden fazla olması halinde karşımıza çıkan "Çok Amaçlı Lineer Programlama Problemi" tanıtılmıştır. Problemin Çözümü için İki Kişili Sıfır Toplamlı Oyunlar ve Hedef Programlama Yöntemleri açıklanmıştır.

Üçüncü bölüm çalışmamızın esasını oluşturmaktadır. Burada Network Problemi Lineer Programlama problemi olarak ele alınmış ve minimum fiyatla maksimum akışı gerçekleştirecek en iyi çözüm araştırılmıştır. Güncelliği göz önünde bulundurularak Network Problemine örnek olarak bir doğal gaz boru hattı problemi ele alınmış, çalışma bu şekilde tamamlanmıştır.

SUMMARY

The Lineer programming problem is a mathematical method that ensures optimal use of resources for achieving a given purpose with some restraints given in the form of lineer equalities or inequalities.

In the first part of this work, single purpose of Lineer Programming problem is studied the most commonly used method know as the simplex method is described.

In the second part introduced "multi purpose lineer programming problem" which comes up when the purpose functions are more than one. For solving the problem "two person zero-sum game" method and the "target programming" method are described.

The third part is the main part of this study. Here the Network problem is studied as a lineer programming problem and the optimum solution achieving maximum flow at minimum cost is searched. A natural gas pipeline is taken as a network example, because of the recent importance of this topic.

I. BÖLÜM

TEK AMAÇLI LİNEER PROGRAMLAMA

1.1. LİNEER PROGRAMLAMANIN TANIMI

Belirli bir amacın gerçeklenme derecesini etkileyen bazı kısıtlayıcı koşulların bulunması ve bunların lineer eşitlik veya eşitsizlikler olarak verilmesi durumunda bu amaca ulaşılması için kıt kaynakların en verimli şekilde kullanılmasını sağlayan bir matematik yöntemdir. Bu şekilde varılması istenen hedef (amaç), karın maksimizasyonu, maliyetin minimizasyonu, vs. olarak belirlenebilir.

Genel olarak bir Tek Amaçlı Lineer Programlama Problemi, Amaç fonksiyonu, Esas kısıtlar, Pozitiflik Kısıtları olmak üzere üç ana kısımdan oluşur. Problemin amaçlanan, verilen kısıtlar altında amaç fonksiyonunu maksimum (veya minimum) yapan çözümü bulmaktır. Problemi matematiksel olarak aşağıdaki gibi gösterebiliriz.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned} \qquad (1.1)$$

Esas kısıtlar

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots, \quad x_n \geq 0 \qquad (1.2)$$

Pozitiflik kısıtları.

Kısıtları altında,

$$Z = C_1X_1 + C_2X_2 + \ldots + C_nX_n \qquad (1.3)$$

Amaç fonksiyonu

Amaç fonksiyonunu maksimum (ya da minimum) yapan $X_1, X_2, \ldots, X_n$ bilinmeyenlerini belirleme işlemlerine Lineer Programlama Problemi denir.

Burada ;

$C_1X_1 + C_2X_2 + \ldots + C_nX_n$ amaç fonksiyonu

$i = 1,2, \ldots, n$ olmak üzere ; C_i amaç fonksiyonu katsayıları

Z amaç fonksiyon değeri

$i = 1,2, \ldots, n$ olmak üzere; X_i karar değişkenleri

$j = 1,2, \ldots, m$ olmak üzere; b_j sağ taraf değerleri

$i = 1,2, \ldots, n$

$j = 1,2, \ldots, m$ olmak üzere; a_{ij} kısıt katsayıları

olarak tanımlanır.

Bir lineer programlama problemini aşağıdaki gibi gösterebiliriz.

(1.1) koşulları : $\sum_{j=1}^{n} a_{ij} X_j = b_i \qquad i = 1,2, \ldots, m$

(1.2) koşulları : $X_j \geq 0 \qquad j = 1,2, \ldots, n$

(1.3) Amaç : $Z = \sum_{j=1}^{n} C_jX_j$ maks.veya min.

Bir lineer programlama problemine örnek verecek olursak;

$$5X_1 + 2X_2 - X_3 + 7X_4 = 8$$

$$-2X_1 - X_2 + 5X_3 + 9X_4 \leq 7 \qquad (1.1)$$

$$X_1 + X_2 + 2X_3 + X_4 \leq 2$$

$$X_1 \geq 0, \quad X_2 \geq 0, \quad X_3 \geq 0, \quad X_4 \geq 0 \qquad (1.2)$$

Kısıtları altında

$$\text{Min } Z = 2X_1 + X_2 + 3X_3 + X_4 \qquad (1.3)$$

Amaç fonksiyonunu minimum yapan X_1, X_2, X_3, X_4 değişkenlerinin belirlenmesi lineer programlama problemidir.

Bir lineer programlama probleminde önce gerekli veriler toplanır, probleme ait bir model kurulur ve daha sonra bu modelin çözümleri araştırılır. Bu çözümler kurulmuş olan modelin yapısına bağlı olarak birden fazla olabilir ya da hiçbir çözüm bulunmayabilir. Zaman zaman amaç denklemine aynı değeri kazandıran farklı (Alternatif) çözümlere de rastlanır.

Karar verme durumunda bulunan bir yöneticinin bu farklı seçenekler arasından, işletme için en uygun olanını uygulamaya koyması gerekir. Bazı durumlarda ise, modele sokulmamış olan başka bir çözümün benimsenmesi de gerekebilecektir. Ama her durumda lineer programlama, yönetim için önemli bir araç olarak kullanılabilecektir.

Örnek: Bir fabrika üç mamul imal etmektedir. Bu mamuller üç farklı üretim kademesinde işlem görmektedir. Her mamulün bir birimini imal etmek için gerekli zaman ve atölyelerin günlük kapasitesi aşağıdaki gibi verilmektedir.

	(Dakika - birim mamul)			
Atölye	1.Mamul	2.Mamul	3.Mamul	Atölye kapasitesi (Dak/gün)
I	1	2	1	430
II	3	-	2	460
III	1	4	-	420

Her bir mamulün birim karı sırasıyla 3, 2 ve 5 TL. verilerek imal edilecek günlük miktarın belirlenmesi gerekmektedir. (Üretilen malların pazarlandığı varsayılmaktadır).

X_1, X_2, X_3 sırayla 1., 2. ve 3. mamullerin üretim miktarlarını göstersin. Toplam kar; $3X_1 + 2X_2 + 5X_3$ olur. Bu problemde kapasite sınırlayıcı koşulu ve pozitiflik koşulu vardır. Nitekim üç mamulün alacağı zaman herbiri için atölyelerin kapasitesini aşamayacaktır.

I. atölye için : $X_1 + 2X_2 + X_3 \leq 430$

II. atölye için : $3X_1 + 2X_3 \leq 460$

III. atölye için : $X_1 + 4X_2 \leq 420$

İncelediğimiz bu örnekte lineer programlama problemi ;

$$\text{Maks } Z = 3X_1 + 2X_2 + 5X_3$$

Kısıtlar :

$$X_1 + 2X_2 + X_3 \leq 430$$

$$3X_1 + 2X_3 \leq 460$$

$$X_1 + 4X_2 \leq 420$$

$$X_1 \geq 0, \quad X_2 \geq 0, \quad X_3 \geq 0$$

şeklinde yazılır.

1.2. SİMPLEKS YÖNTEM

Tek amaçlı lineer programlama problemini çözmek için pek çok yöntem geliştirilmiştir. En çok kullanılan ve çok bilinen bir yöntem "Simpleks Yöntem" dir.

Bu yöntem 1947 yılında George B.Dantzing tarafından geliştirilmiştir. Yöntemin en önemli özelliği, değişken sayısı çok olan problemlere de uygulanabilmesidir.

Şimdi Simpleks yöntemin kullanıldığı uygulama alanlarını ve çözüm yöntemlerini açıklayalım (1).

1.2.1. Simpleks Yöntemin Uygulama Alanları

Burada, lineer programlama problemlerinin çözümlerinde simpleks yöntemin kullanılması gereken bazı problemlere kısaca değinilecektir. Bunlar ;

---

(1) Tulunay,Y., "İşletme Matematiği", 1982, s.177-178.

1. Karışım Problemleri : Burada gıda sanayiinde, Petrol sanayiinde, metalurjide maliyetleri de gözönüne alınarak yapılacak karışımların bünyelendirilmesinde en uygun yol araştırılır.

2. Optimum üretim programının saptanması: Burada ise sınırlı kaynakların (Kapasite, ham madde, iş gücü, taşımalar vs.) optimum kullanımı ile karı maksimize veya maliyetleri minimize etme problemleri söz konusudur.

3. İş ve ücret değerlemesi: Lineer programlama değerleme işleminde açık olarak ilgilenen faktörler için uygun ağırlıkların saptanması çoklu korelasyon analizi yerine kullanılabilir.

4. Depolama Problemleri: Belirli bir zaman içinde saptanmış depo kapasitesinin karı maksimize (veya maliyetleri minimize) edecek şekilde satışlarını, depolamasını veya satın almaları saptamada optimum uygulamayı sağlamaya çalışır.

5. Malzeme kullanımını optimize eder: Esasında bu optimum üretim programının saptanmasının başka bir görünümüdür. Ana konu, standart şekilli hammaddelerin (çelik levhalar, yassı veya yaprak metaller vs.) kayıplar en az (minimum) olacak şekilde kullanımını sağlamaktır.

6. Uzun dönem planlama çalışmaları: Gelecekteki talepleri karşılayabilmek üzere verimli kapasitenin saptanması, bunun verimli bir program içinde sağlanması ve hatta eldeki optimum çözümlerin duyarlılık analizlerinin öngörülerdeki hataları, fiyat değişimlerini de dikkate alarak yapılmasına olanak verir.

7. Yapısal modellerin optimize edilmesi: Yapısal modellerde dayanıklılık, ağırlık ve çevre koşullarını dikkate alarak optimum planın hazırlanmasındaki kullanımlardır.

Bütün bunların yanında, bu yöntemin uygulanabildiği daha değişik problemlere de rastlanabilir.

1.2.2. Simpleks Yöntemin Algoritması

Simpleks yöntemin uygulanmasında ilk aşamayı, sınırlayıcı koşullardaki eşitsizliklerin eşitlik haline getirilmesi oluşturacaktır. Bunun için eşitsizliklere ya aylak değişkenlerin eklenmesi ($\leqslant$ durumunda) ya da artık değişkenlerin çıkarılması ($\geqslant$ durumunda) gerekecektir. Bu işlemlerin yapıldığını varsayarsak yöntemi şöyle özetleyebiliriz (2).

1. Bir başlangıç mümkün temel çözümü bulunur.

2. Temel matrisin tersi (B^{-1}) bulunur.

3. $Y_j = B^{-1}.A_j$ ($j = 1,2, \ldots, r$ için) bulunur.

(Burada r aylak değişkenleri de kapsayan değişken sayısıdır).

4. $Z_j - C_j = \sum_{i=1}^{m} C_{Bi} Y_{ij} - C_j$ hesap edilir ($j = 1,2,\ldots,r$)

5. Negatif değerli ($Z_j - C_j$) bulunup bulunmadığı araştırılır. Eğer varsa, bir sonraki adıma geçilir, aksi halde 11. adıma atlanır. Böylece optimum çözüme ulaşılır.

---

(2) Tulunay,Y., "İşletme Matematiği", 1982, s.213, 214.

6. $(Z_j - C_j) < 0$ ve ilgili sütundaki bütün $y_{ij} < 0$ olan bir sütunun bulunup bulunmadığı araştırılır. Eğer böyle bir sütun varsa işlemlere son verilir. Aksi halde bir sonraki adıma geçilir.

7. Maks $\{|Z_j - C_j|\} = |Z_h - C_h|$ bulunur. (Bütün j'ler için). A_h vektörü temel değişken yapılır.

8. Min $\{X_{Bi} / Y_{ih}\} = X_{Bg} / Y_{gh}$ hesap edilir. (Bütün $Y_{ih} > 0$ için). Eğer X_{Bg} / Y_{gh} en küçük değere sahip ise, Bg temeli terk eder.

9. $X^*_{Bg} = X_{Bg} / Y_{gh}$ ve $X^*_{Bi} = X_{Bi} - X_h Y_{ih}$ hesap edilir. ($i \neq g$ dışında bütün i'ler için). Burada X^*_{Bi} yeni temel değişkenin çözüm değeridir.

10. A_h 'yı temele ekleyerek ve B_g'yi temelden çıkararak yeni temel saptanır ve ikinci adıma dönülür.

11. $(Z_j - C_j) = 0$ olan herhangi bir temel olmayan vektörün bulunup bulunmadığı araştırılır. Eğer varsa elde edilen optimum çözüm tek değildir, alternatif çözümün karakterlerini incelemek üzere söz konusu vektör temele girer, yeni optimum mümkün temel çözüm bulunur. Aksi halde optimum çözüm tektir, işlemler sona erer.

2. BÖLÜM

ÇOK AMAÇLI LİNEER PROGRAMLAMA

2.1. ÇOK AMAÇLI LİNEER PROGRAMLAMANIN TANIMI

Çok Amaçlı Lineer Programlama Problemi, amaç fonksiyonlarının birden fazla olması halinde karşımıza çıkar. Genel olarak Çok Amaçlı Lineer Programlama Problemi aşağıdaki gibi tanımlanır.

$$\text{maks } C(\vec{X}) = \text{maks } \{ \vec{C}^{1}\vec{X} , \vec{C}^{2}\vec{X} , \ldots , \vec{C}^{K}\vec{X} \}$$

$$X = \{ \vec{X} \mid A\vec{X} \leqslant \vec{b} , \quad \vec{X} = 0 \}$$

Burada K amaç fonksiyonu sayısı, C, satırları amaç fonksiyonu katsayılarından oluşan kxn boyutlu bir matris, A standart eniyileme (optimizasyon) problemlerindeki gibi m x n boyutlu teknolojik katsayılar (Kısıt kaynakları) matrisi, $\vec{b}$ m boyutlu kaynak vektörüdür. X, $A\vec{X} \leqslant \vec{b}$ kısıtlar sisteminin meydana getirdiği uygun çözüm alanı (uzayı)'nı gösterir.

Çok Amaçlı Lineer Programlama Problemini açık olarak yazacak olursak,

$$\left.\begin{array}{l} a_{11}X_1 + a_{12}X_2 + \ldots\ldots + a_{1n}X_n \leqslant b_1 \\ a_{21}X_1 + a_{22}X_2 + \ldots\ldots + a_{2n}X_n \leqslant b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad\qquad \vdots \\ a_{m1}X_1 + a_{m2}X_2 + \ldots\ldots + a_{mn}X_n \leqslant b_m \end{array}\right\} \text{Esas Kısıtlar}$$

$$X_1, X_2, \ldots, X_n \geqslant 0 \quad \text{Pozitiflik kısıtları}$$

Kısıtları altında,

$$\text{Maks } Z_1 = C_{11}X_1 + C_{12}X_2 + \ldots + C_{1n}X_n$$

$$\text{Maks } Z_2 = C_{21}X_1 + C_{22}X_2 + \ldots + C_{2n}X_n$$

$$\vdots \quad \vdots \quad \vdots \qquad \vdots \qquad\qquad \vdots$$

$$\text{Maks } Z_K = C_{K1}X_1 + C_{K2}X_2 + \ldots + C_{Kn}X_n$$

amaç fonksiyonlarını maksimum (ya da minimum) yapan, $X_1, X_2, \ldots, X_n$ bilinmeyenlerini belirleme işlemine Lineer Programlama probleminin çözümü denir.

Not: Esas kısıtlardaki eşitsizlikler, $\leqslant$, $=$, $\geqslant$ olabilir.

X uzayı Lineer Programlama Problemlerinde konveks olduğuna göre X'teki her nokta için $C(\vec{X})$ vektörü hesaplanarak amaç fonksiyonları uzayı S bulunur.

$$S = \{ C(\vec{X}) \mid \vec{X} \varepsilon X \}$$

Yani amaç vektörünü "maksimum" yapan $\vec{X} \varepsilon S$ vektörü elde etmektir. Ancak bütün amaçları aynı anda maksimum yapacak bir çözümün bulunması mümkün olmadığından bir uzlaşık çözüm bulmaya çalışılır.

Örnek: X şirketi iki tip oyuncak bebek üretmektedir. A tipindeki bebek B'ye göre kalite seviyesi daha yüksektir. Sırasıyla nispi karlar 0.4 TL, 0.30 TL. dir. Her bir A bebeği için gerekli olan teçhizat miktarı B için gerekenin iki katıdır. Şirket günde en fazla 500 birim faktör kullanabilmektedir. A ve B ürünü için ortak malzeme ihtiyacı ise günlük 400 birimdir. Firmanın satış problemi yoktur. Ancak

A ürününden mümkün olduğunca daha fazla satmak istemektedir. Bu arada karını da maksimum seviyede tutmak istemektedir.

Buna göre problemi

1. Amaç: Karın maksimizasyonu

2. Amaç: A'nın üretim maksimizasyonu olarak düşünüp Amaç fonksiyonlarını aşağıdaki gibi yazabiliriz :

$$\text{Maks } f_1 = 0.4\, X_1 + 0.3\, X_2$$

$$\text{Maks } f_2 = X_1$$

Kısıtlar :

$$2\, X_1 + X_2 \leq 500$$

$$X_1 + X_2 \leq 400$$

$$X_1\,,\, X_2 \geq 0$$

Böylece Problem Çok Amaçlı Lineer Programlama problemi olarak ifade edilmiş olur.

2.2. ÇOK AMAÇLI LİNEER PROGRAMLAMA PROBLEMLERİNİ ÇÖZMEK İÇİN KULLANILAN YÖNTEMLER

Ç.A.L.P. Problemlerini çözmek için bir çok yöntem geliştirilmiştir. Biz bunlardan sadece ikisini ele alacağız. Bunlardan birincisi Belenson ve Kapur'un iki kişili Sıfır toplamlı oyunlar yöntemi diğeri ise Hedef programlama yöntemidir.

2.2.1. İki Kişili Sıfır Toplamlı Oyunlar Yöntemi

Belenson ve Kapur'un geliştirdikleri bu yöntemde amaç fonksiyonları teker teker maksimize edilerek bir ödemeler matrisi kurulmaktadır (1). Bu matris bir oyun matrisi olarak düşünülmekte, z_i^j , birinci oyuncu i ninci, ikinci oyuncu j ninci stratejiyi uyguladığında beklenen kazancı göstermektedir. λ_i birinci oyuncunun i ninci stratejiyi kullanma ağırlığı, μ_j de ikinci oyuncunun j ninci stratejiyi kullanma ağırlığını göstermek üzere,

$$\sum_{i=1}^{K} \lambda_i = 1 \qquad \lambda_i \geqslant 0 \ , \quad \text{her i için}$$

$$\sum_{j=1}^{K} \mu_j = 1 \qquad \mu_j \geqslant 0 \ , \quad \text{her j için}$$

olmaktadır. Oyunun beklenen ödemesi,

$$P = \sum_{i=1}^{K} \sum_{j=1}^{K} z_i^j \lambda_i \mu_i$$

dir. P_o ile P nin minimum değeri, P^o ile de maksimum değeri gösterilmekte, oyunun çözümü aşağıdaki primal-dual lineer programlama çiftlerinden birini çözerek elde edilmektedir.

1. $\sum_{i=1}^{K} z_i^j \ r_i \geqslant 1, \quad r_i \geqslant 0, \quad$ her i için

kısıtları altında

(1) Belenson, M., Kapur, K.C., "An Algorithm for Solving Multi-criterion Lineer Programming, Problems with examples", O.R. Quarterly, 1973, s.65-77.

$$\min \left(\frac{1}{P_o} \right) = \min \sum_{i=1}^{K} r_i$$

Lineer programlama problemi, veya

$$2. \quad \sum_{j=1}^{K} Z_i^j \ S_j \leqslant 1 \ , \ S_j \geqslant 0 \quad \forall_j \quad \text{için}$$

Kısıtları altında

$$\text{maks} \ \left(\frac{1}{P^o} \right) = \sum_{j=1}^{K} S_j$$

Lineer programlama problemi

Optimizasyon halinde

$$P_o^* = P^{o^*} = P^*$$

ve

$$\lambda_i^* = r_i^* \ P^* \quad , \quad \mu_j^* = S_j^* \ P^*$$

olmaktadır, burada P^* oyunun optimal ödemesidir. Oyun matrisine iki kişili sıfır toplamlı oyun teorisinin uygulanması sonucu elde edilen ağırlıklar kullanılarak baskın çözümlerden biri elde edilmektedir.

Pratik hayatta çok amaçlı Lineer programlama problemlerinin çoğunda amaç fonksiyonlarının değerleri arasında büyük farklılıklar mevcuttur. Aynı zamanda boyut farkı da olmaktadır. Örneğin üretim güvenilirliğini ve karı maksimize etme problemi düşünülürse, güvenilirlik sıfır ile bir arasında değişen bir değer iken, genel olarak kar, TL.olarak ifade edilmektedir. Eğer $\vec{C}^1 \vec{X}$ güvenilirliği $\vec{C}^2 \vec{X}$ ise karı gösteriyorsa, oyun matrisindeki ikinci satır daima birinci satıra göre baskın olacaktır. Bu nedenle ödemeler matrisi elemanları bu farklılığı giderecek şekilde normalize edilmektedir.

Genel olarak normalizasyon işlemi aşağıdaki şekilde yapılmaktadır. μ_i (i= 1,2,...,K)'ler $\vec{C}^i \vec{X}$ in maksimum değerleri olmak üzere, ödeme matrisinin yeni elemanları z_i^j ler

$$z_i^j = \frac{z_i^j}{\mu_i} = \frac{\vec{C}^i\vec{X}}{\mu_i} \qquad i= 1,2,\ldots,K\ , \quad j= 1,2,\ldots,K$$

şekline dönüştürülmektedir. Matrisin yeni elemanları her bir amacın maksimum değerine göre normalize edilmiş değerlerdir. Bunun sonucu olarak oyun aşağıdaki normalize amaç fonksiyonlarına karşı gelmektedir.

$$\text{maks } \vec{C}^i \vec{X} = \frac{1}{\mu_i} \vec{C}^i \vec{X}^i \qquad i= 1,2,\ldots,K \text{ ve}$$

$$\vec{X} \in X$$

Dolayısıyla oyunun çözümü sonucunda bulunan optimal ağırlık vektörü, orjinal amaçlar kümesinin değil normalize amaçlar kümesinin optimal ağırlıkları olmaktadır. Bundan dolayı normalize olmayan esas amaç fonksiyonları kümesi için eş değer en iyi (optimal) ağırlık vektörünü bulmak gerekmektedir. λ 'normalize oyunun çözümü sonucu bulunan en iyi (optimal) ağırlık vektörü olmak üzere baskın nokta $\vec{X}^{*p+1}$ i bulmak için eşdeğer doğrusal program,

$$\vec{X} \in X$$

şartları altında

$$\sum_{i=1}^{K} \lambda_i \vec{C}^i \vec{X} = \sum_{i=1}^{K} \frac{\lambda_i}{\mu_i} \vec{C}^i \vec{X}$$

olmaktadır. Normalize amaç fonksiyonları esasında

$\frac{\lambda_i}{\mu_i}$ (i= 1,2,...,K) çarpanları ile "tartılanmıştır".

Ancak

$$\sum_{i=1}^{K} \frac{\lambda_i}{\mu_i} \neq 1$$

olması λ_i' / μ_i (i= 1,2,...,K) değerlerinin optimal ağırlık vektörünü oluşturmasını engellemektedir. Bu aşağıdaki şekilde düzeltilmektedir.

$$n_i = \frac{\lambda_i'}{\mu_i}$$

alınarak ve orjinal amaçlar kümesi için optimal ağırlık vektörü $\vec{\lambda}^*$,

$$\lambda_i^* = \frac{n_i}{\sum_{i=1}^{K} n_i} \qquad i= 1,2,...,K$$

olacak şekilde belirlenmektedir. Bu durumda eşdeğer doğrusal proğram

$\vec{X} \in X$

olmak üzere

$$\text{maks } \vec{C}^{p+1} \vec{X} = \sum_{i=1}^{K} \lambda_i^* \vec{C}^i \vec{X}$$

$\vec{X}^{p+1}$ i verecek şekilde düzeltilmektedir.

Yöntemde Z_i^j lerin negatif olması sorunuyla karşılaşılabilir. Örneğin kullanılan algoritma maksimizasyon için geliştirilmiş ise ve minimize edilmek istenen fonksiyonları varsa, amaç fonksiyonlarının maksimizasyon için aldığı değerlerin negatifleri ödemeler matrisinin elemanları olarak alınabilmektedir. Bazı i'ler için $\mu_i = 0$ olması sorun yaratmakta, bu durumda normalizasyon gerçekleştirilememektedir. Bu sorunu ortadan kaldırmak için matrisin elemanlarına gerekli büyüklükte pozitif bir K sabiti eklenmektedir. Yani matris ilk matrise denk bir matris olacağından ödemeler matrisinin elemanlarına eklenecek K sabiti optimal ağırlıkları değiştirmemekte, dolayısıyla yöntemde herhangi bir değişiklik yapmaya gerek kalmamaktadır.

Belenson;Kapur algoritmasını kısaca özetlemek gerekirse,

1. Verilen Ç.A.L.P. Problemini her bir amaç için tek amaçlıya indirerek çöz ve P tane amaç için P tane karar değişkeni çözüm kümesi sapta.

2. Birinci adımda elde edilen karar değişkenleri için amaç fonksiyonu değerlerini hesapla ve P x P boyutlu bir ödemeler matrisi oluştur.

3. Eğer herhangi bir f_L değeri (L= 1,2,...,P) sıfırdan küçükse ($f_L < 0$) bu durumda,

K= $-\min f_L$ olarak K sabiti al ve ödemeler matrisi elemanlarına bu K sabitini ekle. Eğer $f_L > 0$ ise (L= 1,2,...,P)

$K = 0$ al.

4. Ödemeler matrisinin bütün elemanlarını kontrol et, eğer herhangi bir $f_L < 0$ $(L= 1,2,\ldots,P)$ ise adım-3'e git.

5. Negatif değerlerden kurtarılmış ödemeler matrisini,

$$f_L' = \frac{f_L}{\max f_{iL}}$$

i ninci satırdaki en büyük f_L değerine f_L $(L= 1,2,\ldots,P)$ yi böl.

6. Normalize edilmiş ödemeler matrisinden A oyuncusu için oyunun λ strateji olasılıklarını hesapla.

7. Optimal λ ların hesabı için;

$$n_K = \frac{\lambda_K}{\max f_{iL}} \qquad (K= 1,2,\ldots,P)$$

$$\bar{\lambda}^* = \frac{n_K}{\Sigma\, n_K} \qquad (K= 1,2,\ldots,P)$$

8. Maks $U= \bar{\lambda}^*.f_K(\bar{X})$ vektörünü aynı kısıtlar altında Lineer programlama problemini çöz.

9. $\bar{X}^{u*}$ (U Uzlaşık çözüm) uzlaşık optimal $\bar{X}^{u*}$ değerleri-ni hesapla ve bu vektörü her bir amaç fonksiyonlarında yerine koyarak optimal uzlaşık çözüm kümesini bul

$$f_K^{u*} = \bar{X}^{u*} \; (\bar{C})$$

Örnek: Amaçlar : Maks $Z_1 = 0.1\ X_1 + 0.2\ X_2$

Maks $Z_2 = 10\ X_1 - 5\ X_2$

Kısıtlar : $X_1 + X_2 \geqslant 1$

$X_1 + X_2 \leqslant 7$

$X_1 \leqslant 5$

$X_2 \leqslant 3$

$X_1 \geqslant 0,\ X_2 \geqslant 0$

Çok Amaçlı Lineer Programlama Problemini Belenson-Kapur Algoritmasını kullanarak çözelim.

Çözüm :

1. Amaç: Maks $Z_1 = 0.1\ X_1 + 0.2\ X_2$ için Simpleks yöntemle bulunan optimal çözüm;

Maks $Z_1 = 1$

$X_1 = 4$, $X_2 = 3$ bulunur.

2. Amaç fonksiyonunda aldığı değer;

Maks $Z_2 = 25$ olur.

2. Amaç: Maks $Z_2 = 10\ X_1 - 5\ X_2$ için Simpleks yöntemle bulunan optimal çözüm :

Maks $Z_2 = 50$

$X_1 = 5$, $X_2 = 2$ bulunur.

1. Amaç fonksiyonunda aldığı değer;

Maks $Z_1 = 0.5$ olur.

Bu sonuçlardan yararlanarak ödemeler matrisini oluşturalım.

Ödemeler Matrisi :

$$\begin{bmatrix} 1 & 0.5 \\ 25 & 50 \end{bmatrix}$$

Şimdi ödemeler matrisini normalize edelim. (Yani her bir satırı amaçların maksimum değerine bölelim.)

$m_1 = 1$ 1.satırın maksimum değeri

$m_2 = 50$ 2.satırın maksimum değeri

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$ olur. Oyunlar kuramı uygulanırsa;

Strateji olasılıkları : $\lambda_1 = 0.5$, $\lambda_2 = 0.5$ bulunur.

$$n_1 = \frac{\lambda_1}{m_1} = \frac{0.5}{1} = 0.5 = \frac{1}{2}$$

$$n_2 = \frac{\lambda_2}{m_2} = \frac{0.5}{50} = 0.01 = \frac{1}{100}$$

$$\Sigma n_k = n_1 + n_2 = \frac{1}{2} + \frac{1}{100} = \frac{51}{100}$$

$$\lambda_1^* = \frac{n_1}{\Sigma n_k} = \frac{\frac{1}{2}}{\frac{51}{100}} = \frac{50}{51}$$

$$\lambda_2^* = \frac{n_2}{\Sigma n_k} = \frac{\frac{1}{100}}{\frac{51}{100}} = \frac{1}{51}$$

$$\lambda^* = \left(\frac{50}{51} , \frac{1}{51} \right) \text{ bulunur.}$$

Şimdiye kadar hangi amaca ne ağırlık verileceğini bulduk. Şimdi fayda fonksiyonunu tanımlayalım.

$$\text{Maks } Z_1 = 0.1\ \lambda_1^*\ X_1 + 0.2\ \lambda_1^*\ X_2$$

$$\text{Maks } Z_1 = \frac{5}{51} X_1 + \frac{10}{51} X_2$$

$$\text{Maks } Z_2 = 10\ \lambda_2^*\ X_1 - 5\ \lambda_2^*\ X_2$$

$$\text{Maks } Z_2 = \frac{10}{51} X_1 - \frac{5}{51} X_2$$

$$\text{Maks } U = \frac{15}{51} X_1 + \frac{5}{51} X_2 \quad \text{bulunur. (Fayda Fonksiyonu)}$$

Bulduğumuz bu fayda fonksiyonunu başlangıçta verilen kısıtlar altında optimum çözümünü aşağıdaki gibidir.

$$\text{Maks } U = \frac{85}{51} \quad X_1 = 5, \quad X_2 = 2 \quad \text{bulunur.}$$

Sonuçlar :

Maks $Z_1 = 0.9$, Maks $Z_2 = 40$, $X_1 = 5$, $X_2 = 2$ olur.

2.2.2. Hedef Programlama ile Çözüm Yöntemi

Hedef programlama, çok sayıda hedef veya amaçların bulunduğu Lineer Programlama problemlerine uygulanan bir yöntemdir.

Firma yönetiminde çok sayıda hedef söz konusu ise bu hedefler sıraya konabilir. Diğer bir deyişle hedeflere öncelik sırası verilebilir. Şayet bu türde bilgiler yönetimce sağlanabilirse bu hedefleri gerçeklemek yönetimin görevi olacaktır. Sonuçta ise sapmalar, yani bir hedefin gerçekleşmesi (= pozitif sapma) ve hedefin altında seyretmesi (= negatif sapma) değerleri toplamının minimize edilmesi tek bir amaç olarak ortaya koyulur.

Hedef Programlama Problemleri tek hedefli model, Eş değerli hedefler modeli ve öncelikli hedefler modeli olarak karşımıza çıkar.

Tek hedefli modelde amaç tek olduğundan bu çalışmada ele alınmamıştır. Şimdi kısaca Eşdeğerli hedefler modelini ve Öncelikli Hedefler modelini açıklayalım.

2.2.2.1. Eşdeğerli Hedefler Modeli ile Çözüm

$\vec{X} \in S$ ve $\vec{X} \geqslant \vec{0}$ olmak üzere,

$$
\begin{array}{lll}
\text{hedef } \{ \vec{C}^{1} \vec{X} = Z_1 \} & (Z_1 \geqslant t_1) & \\
\text{hedef } \{ \vec{C}^{2} \vec{X} = Z_2 \} & (Z_2 \leqslant t_2) & (1) \\
\text{hedef } \{ \vec{C}^{3} \vec{X} = Z_3 \} & (Z_3 = t_3) &
\end{array}
$$

şeklinde genel olarak verilen (1) Hedef Programlama Probleminin çözümü bu yöntem ile aşağıdaki gibi formüle edilerek

Tek Amaçlı Lineer Programlama gibi çözülür. Burada sapma değişkenlerinin ağırlıklı toplamlarının minimizasyonu ile hedefleri en iyi gerçekleyen çözümler bulunur.

$$\begin{aligned} \vec{C}^{1}\vec{X} + d_1^- \quad & \geq t_1 \\ \vec{C}^{2}\vec{X} - d_2^+ \quad & \leq t_2 \\ \vec{C}^{3}\vec{X} - d_3^+ + d_3^- & = t_3 \end{aligned} \qquad \begin{array}{c}(2)\\ \text{Hedef Kısıtları}\end{array}$$

$$\vec{X} \varepsilon S \quad \text{ve } d_1^-,\ d_2^+,\ d_3^+,\ d_3^- \geq 0$$

olmak üzere;

$$\min \{ W_1^- d_1^- + W_2^+ d_2^+ + W_3^- d_3^- + W_3^+ d_3^+ \}$$

olur.

Burada W_i ağırlıklarının önceden belirlendiğini belirtelim. Aşağıda bu modele ait bir örnek verilmiştir.

Örnek: Mamul Bileşim Problemi Örneği

Masa ve sandalye imal eden bir mobilya şirketinde montaj ve son işlem daireleri mevcuttur. Belirlenen üretim devresinde bir birimin imali için gerekli süreler ve dairelerin kullanılabilir zamanları aşağıdaki tabloda verilmiştir. Zaman birimi saattir.

	Masa	Sandalye	Toplam Saat
Montaj	4	2	60
Son İşlem	2	4	48
Birim kar (TL)	8	6	

Hedefimiz 100 TL. kar etmek ve 10 adet masa talebini karşılamak üzere üretim yapmak olsun. Hedef Programlama için şu notasyonu yazalım.

y^- = kar hedefinden negatif sapma

y^+ = kar hedefinden pozitif sapma

t^- = Masa üretimi hedefinden negatif sapma

t^+ = Masa üretimi hedefinden pozitif sapma

olmak üzere Hedef Programlama aşağıdaki gibi yazılır.

Amaç : Min $Z = y^- + t^-$

1. Kar Hedefi : $8X_1 + 6X_2 + y^- - y^+ = 100$
2. Masa üretim Hedefi: $X_1 + t^- - t^+ = 10$
3. Montaj hattı kısıtı: $4X_1 + 2X_2 \leq 60$
4. Son işlem hattı kısıtı: $2X_1 + 4X_2 \leq 48$

$$X_1, X_2, y^-, y^+, t^-, t^+ \geqslant 0$$

Problemin Hedef Programlama ile çözümü Simpleks yöntem ile yapılmıştır. Başlangıç tablosu ve Optimal tablo aşağıda verilmektedir. Optimal çözüm tablosundan 10 adet masa üretim hedefinin gerçeklendiği ve (12.5 - 10 = 2.5) adet daha iyi olduğu anlaşılır. 2.5 değeri optimum çözümde $t^{+} = 2.5$ olarak görülür.

100 TL. lık kar hedefinin de $y^{-} = 0$, $y^{+} = 0$ ile kesin olarak gerçeklendiği optimal çözümden anlaşılır.

Başlangıç Simpleks tablosu

C_j			0	0	0	0	1	0	1	0
			X_1	X_2	S_1	S_2	y^{-}	y^{+}	t^{-}	t^{+}
1	y^{-}	100	8	6	0	0	1	-1	0	0
1	t^{-}	10	1	0	0	0	0	0	1	-1
0	S_1	60	4	2	1	0	0	0	0	0
0	S_2	48	2	4	0	1	0	0	0	0
	Z_j	110	9	6	0	0	1	-1	1	-1
	C_j-Z_j		-9	-6	0	0	0	1	0	1

Optimal Simpleks Tablosu

C_j			0	0	0	0	1	0	1	0
			X_1	X_2	S_1	S_2	y^-	y^+	t^-	t^+
0	t^+	2.5	0	0.75	0	0	0.125	-0.125	-1	-1
0	X_1	12.5	1	0.75	0	0	0.125	-0.125	0	0
0	S_1	10	0	-1	1	0	-0.5	0.5	0	0
0	S_2	23	0	2.5	0	1	-0.25	0.25	0	0
	Z_j	0	0	0	0	0	0	0	0	0
	C_j-Z_j		0	0	0	0	1	0	1	0

2.2.2.2. Öncelikli Hedefler Modeli

Bu yöntemde hedefler önceliklere göre gruplandırılır. Yüksek öncelik seviyesindeki hedefler ikinci derecede öncelik seviyesi olan hedeflere nazaran daha fazla önemle gözönüne alınır. İkinci derece öncelik seviyeleri olan hedefler üçüncü derece öncelik seviyeleri olan hedeflere nazaran daha fazla önemle gözönüne alınırlar ve bu böylece sürdürülerek gruplandırma yapılır. Örneğin;

$\vec{X} \in S$ olmak üzere

$$\text{hedef } \{\vec{C}^{1}\vec{X} = Z_1\} \qquad P_1(Z_1 \leq t_1)$$

$$\text{hedef } \{\vec{C}^{2}\vec{X} = Z_2\} \qquad P_2(Z_2 \geq t_2)$$

$$\text{hedef } \{\vec{C}^{3}\vec{X} = Z_3\} \qquad P_3(Z_3 = t_3)$$

Burada P_j, $(j= 1,2,3)$ j ninci hedefin öncelik seviyesini belirtmektedir. P_j ler "Öncelik çarpanlarıdır".

$$\vec{X} \in S \quad \text{ve} \quad \vec{d}^{+}, \vec{d}^{-} \geq 0$$

$$\vec{C}^{1}\vec{X} - d_1^{+} \leq t_1$$

$$\vec{C}^{2}\vec{X} + d_3^{-} \geq t_2$$

$$\vec{C}^{3}\vec{X} - d_3^{+} + d_3^{-} = t_3$$

olmak üzere,

$$\min \{P_1 (d_1^{+}) + P_2 (d_3^{-}) + P_3 (d_3^{+} + d_3^{-})\}$$

şeklinde yazılır.

Örnek: Bundan önceki Mamul bileşim Problemi örneğini göz-önüne alalım.

Mamul bileşim probleminde aşağıdaki hedeflerin sıraya konulduğunu varsayarak hedeflerin önem sırası P_1, P_2, P_3 olsun.

Hedefler	Öncelik
1. 13 adet masa üretmek	P_1
2. 135 TL. kar sağlamak	P_2
3. 5 adet sandalye üretmek	P_3

problemi formüle etmek için şu notasyonu yazalım.

y^- = kar hedefinden negatif sapma

y^+ = kar hedefinden pozitif sapma

t^- = Masa üretimi hedefinden negatif sapma

t^+ = Masa üretimi hedefinden pozitif sapma

U^- = Sandalye üretimi hedefinden negatif sapma

U^+ = Sandalye üretimi hedefinden pozitif sapma.

$$Z_{min} = P_1 t^- + P_2 y^- + P_3 U^-$$

$8X_1 + 6X_2 + y^- - y^+ = 135$ Kar hedefi

$X_1 + t^- - t^+ = 13$ Masa üretim hedefi

$X_2 + U^- - U^+ = 5$ Sandalye üretim hedefi

$$4X_1 + 2X_2 \leqslant 60$$

$$2X_1 + 4X_2 \leqslant 48$$

$$X_1,\ X_2,\ y^-,\ y^+,\ t^-,\ t^+,\ U^-,\ U^+ \geqslant 0$$

Başlangıç Simpleks Tablosu

			0	0	0	0	P_2	0	P_1	0	P_3	0
C_j			X_1	X_2	S_1	S_2	y^-	y^+	t^-	t^+	U^-	U^+
P_2	y^-	135	8	6	0	0	1	-1	0	0	0	0
P_1	t^-	13	1	0	0	0	0	0	1	-1	0	0
P_3	U^-	5	0	1	0	0	0	0	0	0	1	-1
0	S_1	60	4	2	1	0	0	0	0	0	0	0
0	S_2	48	2	4	0	1	0	0	0	0	0	0
P_3 {	Z_j		0	1	0	0	0	0	0	0	1	-1
	C_j-Z_j		0	-1	0	0	0	0	0	0	0	1
P_2 {	Z_j		8	6	0	0	1	-1	0	0	0	0
	C_j-Z_j		-8	-6	0	0	0	1	0	0	0	0
P_1 {	Z_j		1	0	0	0	0	0	1	-1	0	0
	C_j-Z_j		-1	0	0	0	0	0	0	1	0	0

Tabloda şu özellikler vardır :

1. P_1, P_2 ve P_3 önceliklerinin herbiri için ayrı bir Z_j ve C_j-Z_j satırı vardır. Sandalye üretimi hedefinden

sapmalar, birimlerin farklı olması nedeniyle kar hedefinden sapmalara ilave edilemediği için ayrı ayrı öncelik satırlarına ihtiyaç vardır. Önceliklerin tablonun altından yukarıya doğru sıra takip etmesi pratik bir uygulamadır.

2. Sütunların $C_j - Z_j$ değerleri tablonun altında bulunan öncelik satırlarında verilmektedir. Örneğin, X_1 sütunu C_j-Z_j değeri tablo altında bulunan P_2 ve P_1 satırlarında bulunmaktadır ve $(-8P_2 - 1P_1)$ olarak okunur; benzer şekilde X_2 sütunu için C_j-Z_j değeri $(-1P_3 - 6P_2)$ olur.

3. Optimal çözüme giren değişkenin bulunması, en fazla önceliğe sahip olan P_1 satırında en büyük negatif değerli elemanın seçilmesi ile olur. Bu örnekte X_1 optimal çözüme giren değişken olmaktadır. Şayet ilk öncelikli satırda negatif eleman olmasa idi ikinci öncelikli satıra bakılacaktı, v.b.

4. Optimal çözümden çıkan değişkenin bulunması (135/8, 13/1 , 5/0, 60/4, 48/2) oranları arasında en küçük pozitif elemanın seçimi ile yapılır. Böylece (13/1) oranı en küçük orandır ve (t^-) optimal çözümden çıkartılır.

5. P satırlarından birinde pozitif C_j-Z_j değeri ile birlikte negatif C_j-Z_j değeri bulunursa negatif değerler gözönüne alınmaz. Bu türdeki pozitif değerler, o değişkenin çözüme girmesi halinde hedefteki sapmayı ifade eder. Örnekte +1 değerli P_1 satırı elde edildiği için P_2 ve P_3 de bulunan negatif değerlere dikkat edilmemiştir.

Başlangıç simpleks tablo kurulduktan sonra, verilen beş kural uygulanarak simpleks işlemlere devam edilir. İncelediğimiz problemin optimal simpleks tablosu aşağıda verilmektedir. Tabloda bulunan P_2 ve P_3 satırlarında sıra ile -4 ve -2 değerli negatif $C_j - Z_j$ elemanları bulunmaktadır. Halbuki P_1 satırında P_1 sütunu altındaki eleman +1 dir ve her iki satırın negatif değerleri gözönüne alınmaz.

Optimal Simpleks Tablo

			0	0	0	0	P_2	0	P_1	0	P_3	0
C_j			X_1	X_2	S_1	S_2	y^-	y^+	t^-	t^+	u^-	u^+
P_2	y^-	7	0	0	-3	0	1	-1	4	-4	0	0
0	X_1	13	1	0	0	0	0	0	1	-1	0	0
P_3	u^-	1	0	0	-1/2	0	0	0	2	-2	1	-1
0	X_2	4	0	1	1/2	0	0	0	-2	2	0	0
0	S_2	6	0	0	-2	1	0	0	6	-6	0	0
P_3 {	Z_j		0	0	-1/2	0	0	0	2	-2	1	-1
	C_j-Z_j		0	0	1/2	0	0	0	-2	2	0	1
P_2 {	Z_j		0	0	-3	0	1	-1	4	-4	0	0
	C_j-Z_j		0	0	3	0	0	1	-4	4	0	0
P_1 {	Z_j		0	0	0	0	0	0	0	0	0	0
	C_j-Z_j		0	0	0	0	0	0	1	0	0	0

Tablodan $X_1 = 13$, kar= 135 TL (Daha önce 128 TL.lık kar elde edilmişti ve fark $y^- = 7$ TL dır.) $X_2 = 4$ adet sandalye ($U^- = 1$) olarak hedefler gerçekleşmiştir. En son hedef % 20 sapma ile belirlenir. $S_2 = 6$ saat; son işlem hattının kullanılmayan boş zamanıdır. Bu sürede tüketilmek istenirse, montaj hattından yeterince zaman verilmelidir.

III. BÖLÜM

NETWORK PROBLEMİ

3.1. NETWORK AKIŞLARI VE LİNEER PROGRAMLAMA ARASINDAKİ BAĞINTILAR

Network akış problemi Lineer Programlama Probleminin özel bir sınıfıdır. Her network akış problemi;

$$\text{Amaç : Maks } Z = \vec{C}^{\,t}\,\vec{X} = \text{maks } V$$

$$\text{Kısıtlar : } A\,\vec{X} = 0$$

$$0 \leqslant \vec{X} \leqslant \vec{b}$$

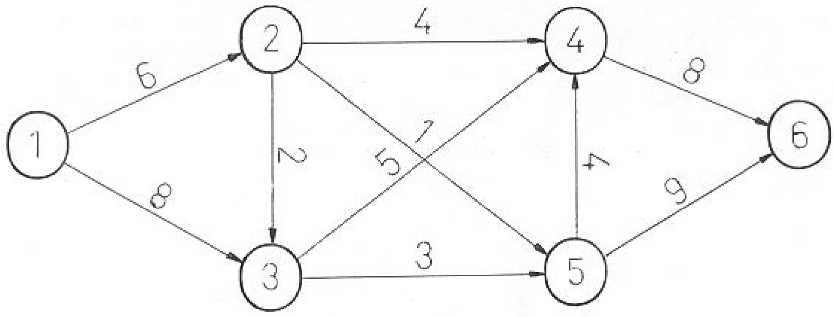
Lineer Programlama problemi olarak formüle edilebilir.

$A\vec{X} = 0$ kısıtı noktalardaki akımın korunumu kısıtıdır. $0 \leqslant \vec{X} \leqslant \vec{b}$ kısıtı arklardaki kapasite kısıtıdır. A matrisinin bileşenleri +1, -1 ve 0 sayılarından meydana gelmiştir.

$\vec{b}$ bileşen sayıları tam sayılar olduğundan çözümler tam sayılı olarak çıkar. Ancak çoğunlukla dejenere çözümler elde edilir.

$$\sum_i X_{ij} - \sum_K X_{jK} = \begin{cases} -V \text{ dir.} & j = s \\ 0 \text{ dır.} & j \neq s\,,\; j \neq t \\ v \text{ dir.} & j = t \end{cases}$$

Örnek :



Networkteki maksimum akışı belirleyelim :

Çözüm : Önce kısıtları belirleyelim.

1 nolu kaynağa gelen akış X_1 olsun. O takdirde gelen akış çıkan akışa eşit olacağından,

$X_1 = X_{12} + X_{13}$ yazılabilir. Aynı şekilde diğer noktalar için de yazacak olursak,

2 noktası için;

$$X_{12} = X_{23} + X_{24} + X_{25}$$

3 noktası için;

$$X_{13} + X_{23} = X_{34} + X_{35}$$

4 noktası için;

$$X_{24} + X_{34} + X_{54} = X_{46}$$

5 noktası için;

$$X_{25} + X_{35} = X_{54} + X_{56}$$

6 noktası için;

$X_{46} + X_{56} = X_1$

Şimdi de arklardaki kapasite kısıtlarını yazalım.

$X_{12} \leq 6$

$X_{13} \leq 8$

$X_{23} \leq 2$

$X_{24} \leq 4$

$X_{25} \leq 1$

$X_{34} \leq 5$

$X_{35} \leq 3$

$X_{46} \leq 8$

$X_{54} \leq 4$

$X_{56} \leq 9$

Burada $x_{ij} \geq 0$ dır.

Amaç fonksiyonumuz Maks $Z = X_1$ olur.

Amaç fonksiyonu ve kısıtlar belli olduğuna göre problemin çözümüne geçebiliriz.

Problemin Çözümü için yapmış olduğumuz bilgisayar programı kullanılmış ve aşağıdaki sonuçlar elde edilmiştir.

Maksimum Akış 12 birim olarak bulunmuştur. Maksimum akışı sağlayan alternatif çözümler aşağıdaki gibidir.

Maks Z = 12

Alternatif Çözümler

	1	2	3	4
X_{12}	5	6	6	4
X_{13}	7	6	6	8
X_{23}	0	1	2	0
X_{24}	4	4	3	3
X_{25}	1	1	1	1
X_{34}	4	4	5	5
X_{35}	3	3	3	3
X_{46}	8	8	8	8
X_{54}	0	0	0	0
X_{56}	4	4	4	4

Minimal Fiyatlı Akışlar :

Her arkın bir kapasitesi vardır. Dolayısıyle kapasiteyi aşmak mümkün değildir. Aynı zamanda her arkta birim başına bir C_{ij} taşıma fiyatı vardır. Amaç kaynaktan kuyuya tüm akışı minimum fiyatla gerçekleştirmektir.

Problemin Lineer modeli ;

Amaç : $\min Z = \Sigma\, C_{ij}\; X_{ij}$

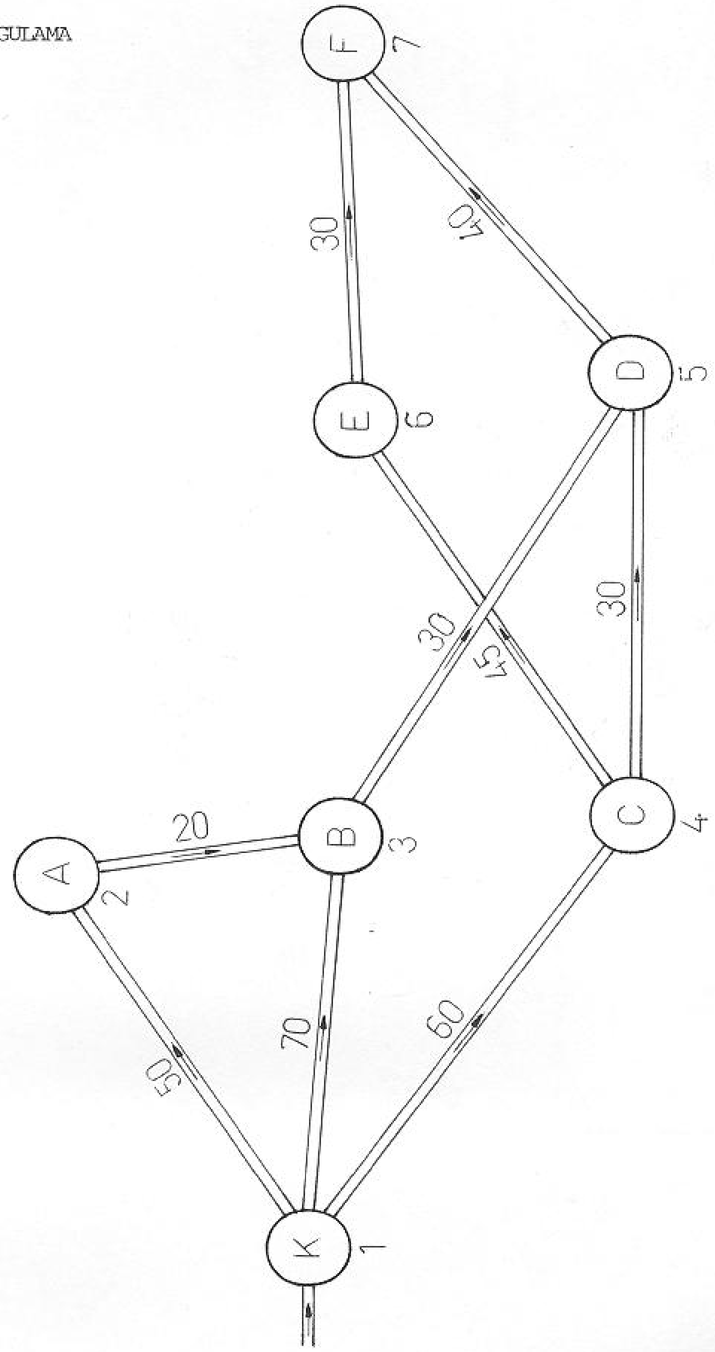
Kısıtlar : $$\sum_{i} X_{ij} - \sum_{K} X_{jK} = \begin{cases} -v & j = s \\ 0 & j \neq s\,,\; j \neq t \\ v & j = t \end{cases}$$

Kapasite kısıtları : $0 \leqslant X_{ij} \leqslant b_{ij}$

olarak formüle edilebilir.

Burada C_{ij} bir birimin A_{ij} arkı boyunca taşıma fiyatıdır.

3.2. UYGULAMA



Şekil 1

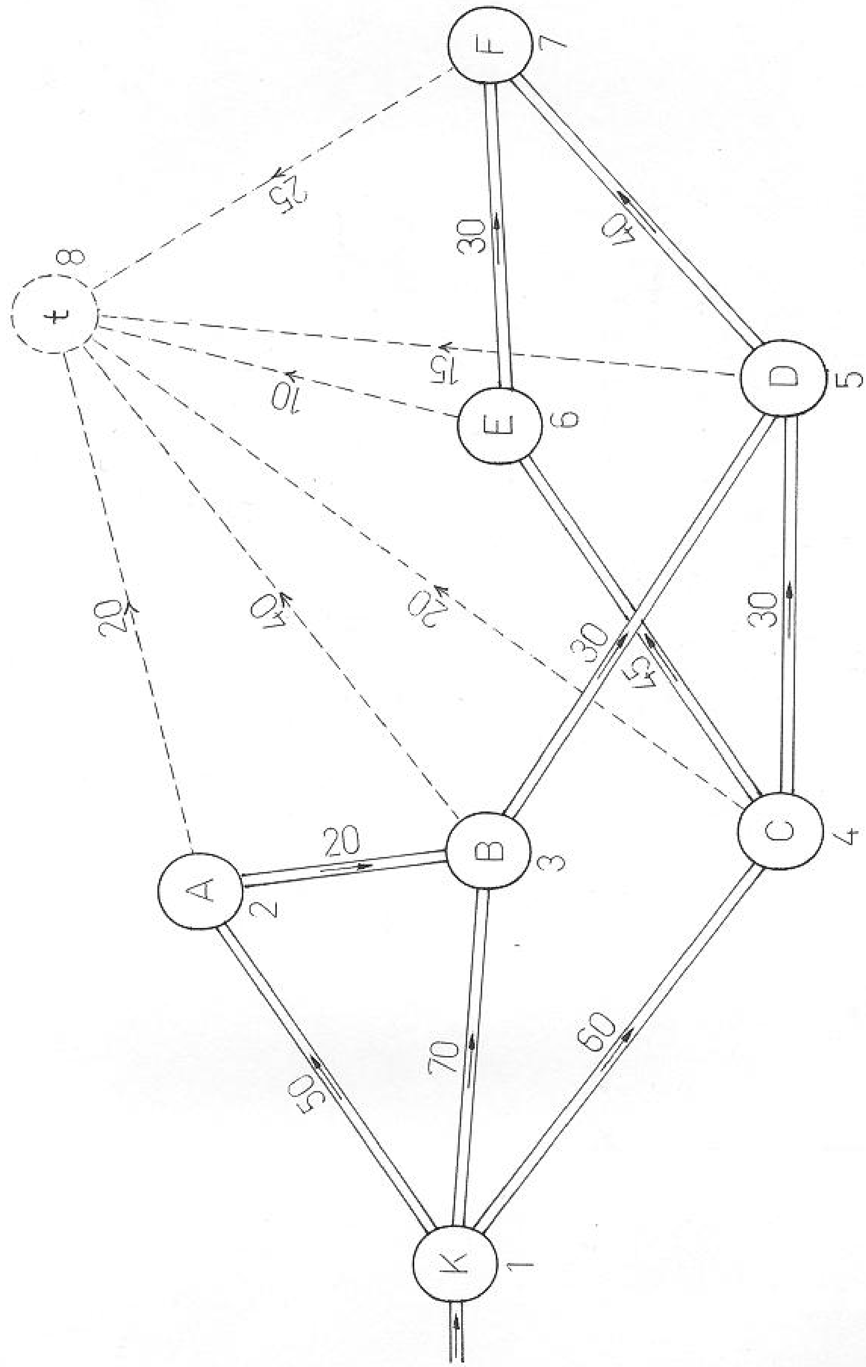
Şekilde görüldüğü gibi Kaynaktan A, B, C, D, E ve F bölgelerine doğal gaz dağıtımı yapılmaktadır. Kapasiteler ve birim taşıma maliyetleri aşağıda verilmiştir. Buna göre Maksimum akışı minimum fiyatla gerçekleştirecek Optimum çözümü bulalım :

	Kapasiteler b_{ij}	Fiyatlar c_{ij}
1-2	50	10
1-3	70	18
1-4	60	14
2-3	20	5
3-5	30	10
4-5	30	6
4-6	45	13
5-7	40	10
6-7	30	5

Şimdi amaç fonksiyonlarını belirleyelim :

Kaynağa gelen akış X_1 olsun. O takdirde ;

Maks $Z_1 = X_1$ olur.



Şekil 2

Maliyetin minimum yapılması istendiğinden ;

$$\text{Min } Z_2 = C_{12}\, X_{12} + C_{13}\, X_{13} + C_{14}\, X_{14} + C_{23}\, X_{23} + C_{35}\, X_{35}$$

$$+ C_{45}\, X_{45} + C_{46}\, X_{46} + C_{57}\, X_{57} + C_{67}\, X_{67}$$

şeklinde yazabiliriz.

Buna göre Amaç fonksiyonlarımızı aşağıdaki gibi yazabiliriz :

1\. Amaç : Maks $Z = X_1$

2\. Amaç : Min $Z = 10\, X_{12} + 18\, X_{13} + 14\, X_{14} + 5\, X_{23} + 10\, X_{35}$

$$+ 6\, X_{45} + 13\, X_{46} + 10\, X_{57} + 5\, X_{67}$$

Şimdi kısıtları belirleyelim :

Kaynağa gelen akış X_1 olduğuna göre ;

$X_1 = X_{12} + X_{13} + X_{14}$ olur. Aynı şekilde diğer noktalar için de yazacak olursak :

2 noktası (A Bölgesi) için ;

$$X_{12} = X_{23} + X_{28}$$

3 noktası (B-Bölgesi) için;

$$X_{13} + X_{23} = X_{35} + X_{38}$$

4 noktası (C-Bölgesi) için ;

$$X_{14} = X_{45} + X_{46} + X_{48}$$

5 noktası (D-Bölgesi) için ;

$$X_{35} + X_{45} = X_{57} + X_{58}$$

6 noktası (E-Bölgesi) için ;

$$X_{46} = X_{67} + X_{68}$$

7 noktası (F-Bölgesi) için ;

$$X_{57} + X_{67} = X_{78}$$

8 noktası için ;

$$X_{28} + X_{38} + X_{48} + X_{58} + X_{68} + X_{78} = X_{1}$$

olur.

Şimdi de arklardaki kapasite kısıtlarını yazalım :

$X_{12} \leqslant 50$	$X_{45} \leqslant 30$
$X_{13} \leqslant 70$	$X_{46} \leqslant 45$
$X_{14} \leqslant 60$	$X_{57} \leqslant 40$
$X_{23} \leqslant 20$	$X_{67} \leqslant 30$
$X_{35} \leqslant 30$	

Bölgelerdeki tüketim miktarlarını kısıt olarak yazarsak;

$X_{28} \leqslant 20$ $\qquad$ $X_{58} \leqslant 15$

$X_{38} \leqslant 40$ $\qquad$ $X_{68} \leqslant 10$

$X_{48} \leqslant 20$ $\qquad$ $X_{78} \leqslant 25$

olur.

Buradaki X_{28}, X_{38}, X_{48}, X_{58}, X_{68}, X_{78} değişkenlerinin birim taşıma fiyatları sıfır'dır.

Amaç denklemlerimizi ve kısıtları düzenli bir şekilde yazacak olursak ;

Amaçlar : Maks $Z = X_1$

$$\text{Min } Z = 10\,X_{12} + 18\,X_{13} + 14\,X_{14} + 5\,X_{23} + 0\,X_{28} +$$
$$+ 10\,X_{35} + 0\,X_{38} + 6\,X_{45} + 13\,X_{46} + 0\,X_{48} +$$
$$+ 10\,X_{57} + 0\,X_{58} + 5\,X_{67} + 0\,X_{67} + 0\,X_{78}$$

olur.

Kısıtlar :

$$X_1 - X_{12} - X_{13} - X_{14} = 0$$

$$X_{12} - X_{23} - X_{28} = 0$$

$$X_{13} + X_{23} - X_{35} - X_{38} = 0$$

$$X_{14} - X_{45} - X_{46} - X_{48} = 0$$

$$X_{35} + X_{45} - X_{57} - X_{58} = 0$$

$$X_{46} - X_{67} - X_{68} = 0$$

$$X_{57} + X_{67} - X_{78} = 0$$

$$-X_1 + X_{28} + X_{38} + X_{48} + X_{68} + X_{78} = 0$$

$X_{12} \leqslant 50$	$X_{23} \leqslant 20$	$X_{38} \leqslant 40$	$X_{48} \leqslant 20$	$X_{67} \leqslant 30$
$X_{13} \leqslant 70$	$X_{28} \leqslant 20$	$X_{45} \leqslant 30$	$X_{57} \leqslant 40$	$X_{68} \leqslant 10$
$X_{14} \leqslant 60$	$X_{35} \leqslant 30$	$X_{46} \leqslant 45$	$X_{58} \leqslant 15$	$X_{78} \leqslant 25$

Amaç denklemler ve kısıtlar bilgisayar programı ile değerlendirilmiş ve Maksimum akışın minimum fiyatla gerçeklendiği optimum çözüm bulunmuştur.

Problemin çözümü iki değişik yolla değerlendirilmiştir.

1. Sistemin önce maksimum akış değeri bulunmuş (130 birim), sonra bu değer kısıt olarak alınıp maliyeti minimum yapan optimal çözüm hesaplanmıştır.

2. Sistemin 1.amaca göre (Maksimum akış) tüm alternatif çözümleri bulunup 2.amaç için aldığı değerler hesaplanmış ve bu alternatif (Minimum maliyet) çözümler içinden maliyeti düşük olan optimal çözüm olarak alınmıştır.

Buna göre aşağıdaki sonuçlar elde edilmiştir.

Alternatif Çözümler

	1	2	3	4
X_{12}	20	40	40	20
X_{13}	70	50	30	50
X_{14}	40	40	60	60
X_{23}	0	20	20	0
X_{28}	20	20	20	20
X_{35}	30	30	10	10
X_{38}	40	40	40	40
X_{45}	10	10	30	30
X_{46}	10	10	10	10
X_{48}	20	20	20	20
X_{57}	25	25	25	25
X_{58}	15	15	15	15
X_{67}	0	0	0	0
X_{68}	10	10	10	10
X_{78}	25	25	25	25
Maks Z_1	130	130	130	130
Min Z_2	2760	2700	2540	2600

Buna göre sistemin optimal çözümü aşağıdaki gibi olmaktadır.

Maks $Z_1 = 130$ "Maksimum akış"

Min $Z_2 = 2540$ "Minimum maliyet"

$X_{12} = 40$

$X_{13} = 30$

$X_{14} = 60$

$X_{23} = 20$

$X_{28} = 20$

$X_{35} = 10$

$X_{38} = 40$

$X_{45} = 30$

$X_{46} = 10$

$X_{48} = 20$

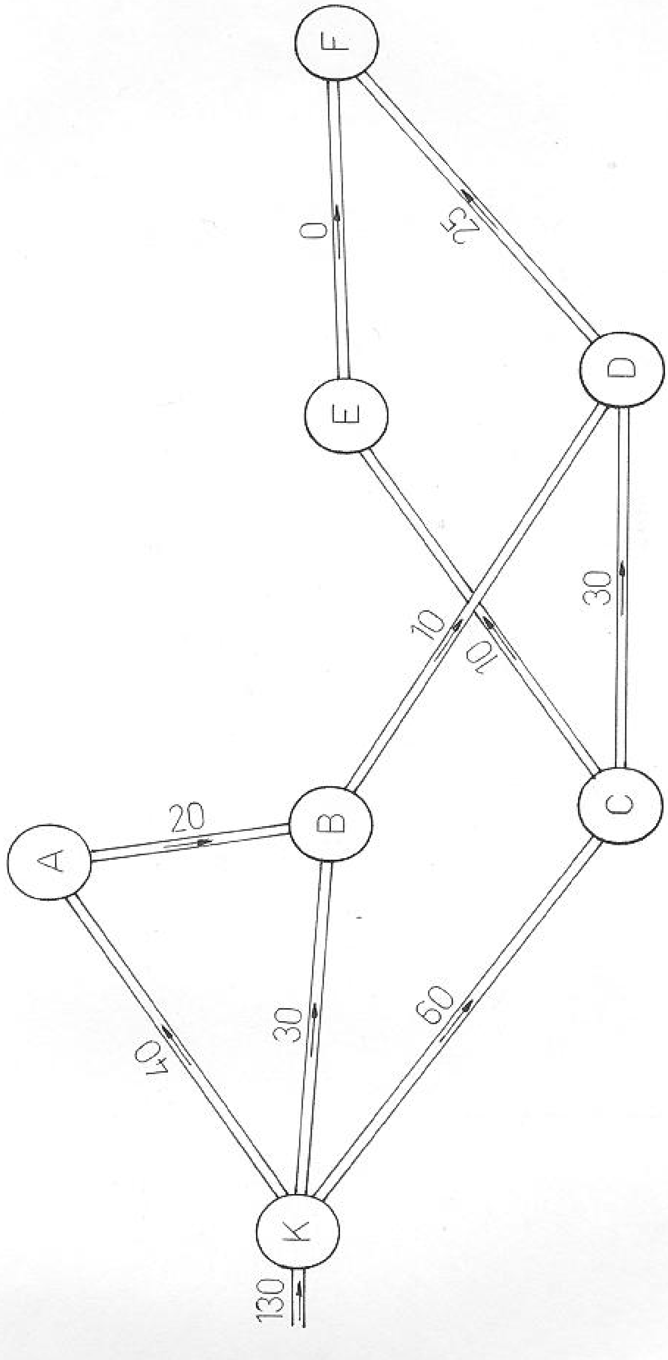
$X_{57} = 25$

$X_{58} = 15$

$X_{67} = 0$

$X_{68} = 10$

$X_{78} = 25$



Şekil 3

4. BÖLÜM

S O N U Ç L A R

Bu çalışmada bir Network Problemi lineer programlama problemi olarak ifade edilerek en iyi çözüm araştırılmıştır. Network olarak da, doğal gaz boru hattı gözönüne alınmıştır.

Şekil 1 den de görüldüğü gibi kaynaktan A, B, C, D ve F yerleşim birimlerine doğal gaz dağıtımı yapılmaktadır. Kapasiteler ve birim taşıma maliyetleri verilmiştir. Bu verilere göre maksimum akışı minimum fiyatla gerçekleştirecek optimum çözüm hesaplanmıştır. Yapılan hesaplama sonucunda maksimum akış değeri 130 birim olarak bulunmuştur. Bu da bize kaynaktan toplam 130 birimlik akış çıkacağını ve tüm yerleşim birimleri için yeterli olacağını göstermektedir. Daha sonra bu 130 birimlik akışı A, B, C, D, E ve F yerleşim bölgelerine taşıma maliyeti minimum olacak şekilde dağıtımı yapılmıştır (Şekil 3).

Optimum çözümü bulmak için iki değişik yolla işlem yapılmıştır. Birinci yol olarak iki adımda sonuca gidilmiştir. Önce sistemin maksimum akış değeri hesaplanmıştır. Sonra bulunan maksimum akış değeri kısıt olarak alınıp ikinci amaç için tekrar işlem yapılarak optimum çözüm bulunmuştur. İkinci yol olarak sistemin birinci amaca göre tüm alternatif çözümleri bulunup ikinci amaç için aldığı değerler hesaplanmış ve bu alternatif optimal çözümler içinden en iyisi (maliyeti minimum olan çözüm) optimum çözüm olarak alınmıştır.

E K

YENİLENMİŞ SİMPLEKS YÖNTEM

YENİLENMİŞ SİMPLEKS YÖNTEM

Bir Lineer Programlama problemini simpleks yöntemle çözerken, her iterasyonda (yani bir tablodan diğerine geçişte) tablonun gövdesi tümüyle işlem görür. Küçük boyutlu problemlerde bu husus fazla sakınca doğurmaz. Fakat Lineer Programlama Problemleri genellikle büyük boyutlarda olup bilgisayarla çözüldüklerinden simpleks yöntem ekonomik değildir.

Yenilenmiş Simpleks Yöntem

Bir iterasyonda baz

$$B_p = (V_1, V_2, \ldots, V_p, \ldots, V_m) = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1p} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2p} & \ldots & a_{2m} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mp} & \ldots & a_{mm} \end{bmatrix}$$

olsun ve bu iterasyon sonunda V_p vektörünün bazdan çıkıp yerine V_r vektörünün girmesine karar verilsin; yani

$B_p = (V_1, V_2, \ldots, V_p, \ldots, V_m)$ bazından

$B_r = (V_1, V_2, \ldots, V_r, \ldots, V_m)$ bazına atlanmasına karar verilsin. Sorun, B_p^{-1} matrisi biliniyorken B_r^{-1} matrisinin bulunmasıdır. Aşağıdaki işlemler bu amaca yöneliktir.

$$B_p^{-1} B_p = B_p^{-1} \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1p} & \ldots & a_{1m} \\ \vdots & & & & & \\ \vdots & & & & & \\ a_{m1} & a_{m2} & \ldots & a_{mp} & \ldots & a_{mm} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \ldots & 0 \\ \vdots & & & & & \\ 0 & 0 & \ldots & 0 & \ldots & 1 \end{bmatrix} = I_m$$

dir. B_p ve B_r matrislerinin sadece birer sütunları farklı olduklarından $B_p^{-1} B_p = I_m$ ve $B_p^{-1}.B_r$ matrislerinin de sadece birer sütunları farklıdır. Yani,

$$B_p^{-1} \begin{bmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{pr} \\ \vdots \\ a_{mr} \end{bmatrix} = \begin{bmatrix} a'_{1r} \\ a'_{2r} \\ \vdots \\ a'_{pr} \\ \vdots \\ a'_{mr} \end{bmatrix}$$

olmak üzere,

$$B_p^{-1} . B_r = B_p^{-1} \begin{bmatrix} a_{11} & a_{12} \ldots & a_{1r} \ldots & a_{1m} \\ \vdots & & & \\ a_{m1} & a_{m2} \ldots & a_{mr} \ldots & a_{mm} \end{bmatrix} = \begin{bmatrix} 1 & 0 \ldots & a'_{1r} \ldots & 0 \\ 0 & 1 \ldots & a'_{2r} \ldots & 0 \\ \vdots & & & \\ 0 & 0 \ldots & a'_{mr} \ldots & 1 \end{bmatrix}$$

Burada $p \neq r$ olmakla beraber a'_{pr} elemanının $B_p^{-1} B_r$ matrisinin asal köşegeni üzerinde bulunduğuna dikkat edilmelidir.

$B_p^{-1} . B_r$ matrisini soldan,

$$E = \begin{bmatrix} 1 & 0 \ldots & a'_{1r}/a'_{pr} & \ldots 0 \\ 0 & 1 \ldots & a_{2r}/a_{pr} & \ldots 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 \ldots & 1/a'_{pr} & \ldots 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 \ldots & a_{mr}/a'_{pr} & \ldots 1 \end{bmatrix}$$

matrisiyle çarparsak;

$E(B_p^{-1} B_r) = I$ olur ve bu eşitliği de sağdan B_r^{-1} ile çarparsak,

$$E\,B_p^{-1}\,(B_r B_r^{-1}) = I\,B_r^{-1}$$

$B_r^{-1} = E\,B_p^{-1}$ sonucu elde edilir.

Bu yöntem ile yapılan Program listesi aşağıda verilmektedir.

```
10 REM              **************************************************
20 REM              *                                                *
30 REM              *     Y I L D I Z   U N I V E R S I T E S I      *
40 REM              *           FEN BILIMLERI ENSTITUSU              *
50 REM              *         MATEMATIK MUH.ANABILIM DALI            *
60 REM              *                                                *
70 REM              *           Aras.Gor.COSKUN GULER                *
80 REM              *                                                *
90 REM              **************************************************
100 REM
110 REM             ********  S I M P L E K S   Y O N T E M  ********
120 REM
130 REM
140 REM
150 READ M,N,LLL
160 DIM A(N,M+2*N),B(N),C(M+2*N),C0(2*N+M),D(N),Q1(N+M),Q2(N),CCC(LLL,M),Z(LLL,L
LL),CX(N*3+N),CC(N*3+M),O(N*3+M),XX(M*5),QB(NN+N,NN+N),QD(NN+N,NN+N),E(NN+N,NN+N
)
170 DIM A2(LLL,LLL),B1(LLL,LLL),P(LLL,LLL),AA(LLL+1,LLL+1),AB(LLL+1,LLL+1),F(LLL
),X(LLL+1),DDD(LLL*3,M*3),AZ(LLL,LLL),RR(NN+N),NN1(M*3),BP(NN+N,NN+N),S(NN+N)
180 DIM BR(NN+N,NN+N),XY(M*5,M*5),FG1(20,10),BIR(N,N),FQ1(N),FQ2(N),C05(2*N+M)
190 FOR I=1 TO LLL
200 READ SV$(I)
210 NEXT I
220 FOR I=1 TO M-1
230 READ X1$(I)
240 NEXT I
250 FOR I=1 TO LLL
260 FOR J=1 TO M
270 READ DDD(I,J)
280 NEXT J:NEXT I
290 FOR I=1 TO LLL:FOR J=1 TO M
300 CCC(I,J)=DDD(I,J):NEXT J:NEXT I
310 FOR I=1 TO N
320 FOR J=1 TO M
330 READ A(I,J)
340 NEXT J
350 NEXT I
360 FOR I=1 TO N
370 READ B(I)
380 NEXT I
390 FOR I=1 TO N
400 READ D(I)
410 NEXT I
420 REM ****** A MATRISININ YENI SEKLI ******
430 FOR P=1 TO LLL
440 FOR J=1 TO M
450 C(J)=CCC(P,J)
460 NEXT J
470 NN=M
480 FOR I=1 TO N
490 IF D(I)<>3 THEN 530
500 NN=NN+1
510 A(I,NN)=-1
520 C(NN)=0
530 NEXT I
```

```
540 FOR I=1 TO N
550 ON D(I) GOTO 580,630,680
560  PRINT"    --- HATALI KOD GIRILMIS --- (KOD * 1,2,3 OLACAK) --- "
570 GOTO 3850
580 BP(I,I)=1
590 C0(I)=0
600 Q2(I)=NN+I
610 A(I,NN+I)=1
620 GOTO 720
630 BP(I,I)=1
640 C0(I)=-1000000!
650 Q2(I)=NN+I
660 A(I,NN+I)=1
670 GOTO 720
680 BP(I,I)=1
690 C0(I)=-1000000!
700 Q2(I)=NN+I
710 A(I,NN+I)=1
720 NEXT I
730 FOR I=1 TO NN
740 Q1(I)=I
750 NEXT I
760 V=0
770 FOR I=NN+1 TO NN+N
780 V=V+1
790 C(I)=C0(V)
800 NEXT I
810 FOR I=1 TO N
820 C05(Q2(I))=C0(I)
830 NEXT I
840 FOR I=1 TO N
850 C0(Q2(I))=C05(Q2(I))
860 NEXT I
870 FOR I=1 TO NN+N
880 CX(I)=C(I)
890 NEXT I
900 REM ***** S DIZISININ BULUNMASI *****
910  FOR I=1 TO NN-1
920  FOR J=I+1 TO NN
930  IF Q1(I)<Q1(J) THEN 950
940  FF=Q1(I):Q1(I)=Q1(J):Q1(J)=FF
950  NEXT J
960  NEXT I
970 T=0
980 FOR I=1 TO N
990 FOR J=1 TO N
1000 T=T+C0(Q2(J))*BP(J,I)
1010 NEXT J
1020 S(I)=-T:T=0
1030 NEXT I
1040 REM ***** CC DIZISININ BULUNMASI *****
1050 FOR J=1 TO NN
1060 FOR I=1 TO N
```

```
1070 T=T+A(I,Q1(J))*S(I)
1080 NEXT I
1090 CC(Q1(J))=C(Q1(J))+T
1100 IF ABS(CC(Q1(J)))<=.00001 THEN CC(Q1(J))=0
1110 T=0
1120 NEXT J
1130 REM ***** CC DIZISININ BULUNMASI *****
1140 PS=0:SS=0
1150 FOR I=1 TO NN
1160 IF CC(Q1(I))<0 THEN 1200
1170 IF CC(Q1(I))>0 THEN 1190
1180 SS=SS+1:GOTO 1200
1190 PS=PS+1
1200 NEXT I
1210 Q=0
1220 IF ALTER1=1 THEN 1330
1230 IF PS=0 THEN 2860
1240 REM ***** BAZA GIRECEK VEKTORUN YERI ICIN HESAPLAMA *****
1250 REM ***** CC DIZISININ EN BUYUK ELEMANININ YERINI BULUR *****
1260 IF PS=0 THEN 1330
1270 EB=CC(Q1(1)):Y=Q1(1)
1280 FOR I=2 TO NN
1290 IF EB>=CC(Q1(I)) THEN 1310
1300 EB=CC(Q1(I)):Y=Q1(I)
1310 NEXT I
1320 IF FB1=0 THEN 1680
1330 REM **** ALTERNATIF COZUM VARSA YAPILACAK ****
1340 W=0:W1=W1+1:I1=I1+1
1350 IF Q8=0 THEN W5=W5+1
1360 FOR I=1 TO NN
1370 IF CC(Q1(I))=0 AND C(Q1(I))<>-1000000! THEN W=W+1:Q=1:FG(W1,W)=Q1(I):FG1(I1
,W)=Q1(I)
1380 NEXT I
1390 REM **** BAZA GIRECEK VEKTORUN BELIRLENMESI ****
1400 IF W1=1 AND W2=0 THEN Y=FG(1,1):GOTO 1530
1410 L1=1
1420 FOR K=1 TO W
1430 FOR I=2 TO W5
1440 FOR J=1 TO W
1450 IF FG(1,K)<>FG(I,J) THEN 1470
1460 L1=L1+1
1470 NEXT J:NEXT I
1480 IF L1=W5 THEN Y=FG(1,K):GOTO 1530
1490 L1=1
1500 NEXT K
1510 PRINT"HATA VAR"
1520 INPUT FB$
1530 W2=1
1540 IF W1=W THEN W1=0:Q8=1
1550 ALTER1=1
1560 J1=0
1570 IF I1=1 THEN 1680
1580 FOR K=1 TO I1-1
1590 FOR I=K+1 TO I1
1600 FOR L=1 TO W
1610 FOR J=1 TO W
```

```
1620 IF FG1(K,L)<>FG1(I,J) THEN 1640
1630 J1=J1+1
1640 NEXT J:NEXT L
1650 IF J1=W THEN 3030
1660 J1=0
1670 NEXT I:NEXT K
1680 REM ***** BAZDAN CIKACAK VEKTORUN BELIRLENMESI *****
1690 REM ***** QB MATRISININ OLUSTURULMASI *****
1700 FOR I=1 TO N
1710 QB(I,1)=B(I)
1720 QB(I,2)=A(I,Y)
1730 NEXT I
1740 REM ***** BP*QB MATRISININ HESABI *****
1750 C=0
1760 FOR I=1 TO N
1770 FOR J=1 TO 2
1780 FOR K=1 TO N
1790 C=C+BP(I,K)*QB(K,J)
1800 NEXT K
1810 QD(I,J)=C:C=0
1820 NEXT J
1830 NEXT I
1840 REM ***** ORAN DIZISININ BULUNMASI *****
1850 FOR I=1 TO N
1860 IF QD(I,2)=0 THEN O(I)=1000000!:GOTO 1890
1870 IF QD(I,1)>0 AND QD(I,2)<0 OR QD(I,1)<0 AND QD(I,2)>0 OR QD(I,1)=0 AND QD(I
,2)<0 THEN O(I)=1000000!:GOTO 1890
1880 O(I)=QD(I,1)/QD(I,2)
1890 NEXT I
1900 REM ***** BAZDAN CIKACAK VEKTORUN YERINI BULUR *****
1910 REM ***** O DIZISININ EN KUCUK ELAMANININ YERININ BULUNMASI *****
1920 EK=O(1):Y5=1:Y1=Q2(1)
1930 FOR I=2 TO N
1940 IF EK<=O(I) THEN 1960
1950 EK=O(I):Y1=Q2(I):Y5=I
1960 NEXT I
1970 J6=0
1980 FOR I=1 TO N
1990 IF EK<>O(I) THEN 2010
2000 J6=J6+1
2010 NEXT I
2020 IF J6=1 THEN 2400
2030 REM ***** DEJENERE COZUM VARSA YAPILACAK *****
2040 FOR I=1 TO N:FQ1(I)=I:FQ2(I)=Q2(I):NEXT I
2050 FOR I=1 TO N-1
2060 FOR J=I+1 TO N
2070 IF FQ2(I)<FQ2(J) THEN 2100
2080 E1=FQ2(I):FQ2(I)=FQ2(J):FQ2(J)=E1
2090 E1=FQ1(I):FQ1(I)=FQ1(J):FQ1(J)=E1
2100 NEXT J:NEXT I
2110 FOR I=1 TO N:FOR J=1 TO N:BIR(I,J)=0:NEXT J:NEXT I
2120 FOR I=1 TO N
2130 BIR(FQ1(I),I)=1
2140 NEXT I
2150 J7=0
```

```
2160 FOR I=1 TO N
2170 IF EK<>O(I) THEN 2190
2180 J7=J7+1:FB7(J7)=I
2190 NEXT I
2200 FOR I=1 TO N
2210 FOR J=1 TO J7
2220 IF QD(FB7(J),2)<=0 THEN O7(J)=10000000#:GOTO 2240
2230 O7(J)=BIR(FB7(J),I)/QD(FB7(J),2)
2240 NEXT J
2250 EK=O7(1):Y1=FB7(1)
2260 FOR K=2 TO J7
2270 IF EK<=O7(K) THEN 2290
2280 EK=O7(K):Y1=FB7(K)
2290 NEXT K
2300 Y5=Y1:Y1=Q2(Y1)
2310 J8=0
2320 FOR K=1 TO J7
2330 IF EK<>O7(K) THEN 2350
2340 J8=J8+1
2350 NEXT K
2360 IF J8=1 THEN 2400
2370 NEXT I
2380 PRINT"  HATA VAR DIKKAT "
2390 INPUT JHJ$
2400 C0(Y)=C(Y)
2410 FOR I=1 TO N
2420 IF Q2(I)<>Y1 THEN 2440
2430 Q2(I)=Y
2440 NEXT I
2450 FOR I=1 TO NN
2460 IF Q1(I)<>Y THEN 2480
2470 Q1(I)=Y1
2480 NEXT I
2490 REM ***** E MATRISININ OLUSTURULMASI *****
2500 Y2=1/QD(Y5,2)
2510 FOR J=1 TO N
2520 FOR I=1 TO N
2530 IF J=Y5 THEN 2540 ELSE 2590
2540 IF I=Y5 THEN 2550 ELSE 2570
2550 E(I,J)=Y2
2560 GOTO 2610
2570 E(I,J)=-QD(I,2)/QD(Y5,2)
2580 GOTO 2610
2590 IF I=J THEN 2600 ELSE 2610
2600 E(I,J)=1
2610 NEXT I
2620 NEXT J
2630 REM ***** BR MATRISININ OLUSTURULMASI *****
2640 T=0
2650 FOR I=1 TO N
2660 FOR J=1 TO N
2670 FOR K=1 TO N
2680 T=T+E(I,K)*BP(K,J)
2690 NEXT K
```

```
2700 BR(I,J)=T:T=0
2710 NEXT J:NEXT I
2720 FOR I=1 TO N
2730 FOR J=1 TO N
2740 BP(I,J)=BR(I,J)
2750 NEXT J:NEXT I
2760 CLS
2770 FOR I=1 TO NN+N
2780 CC(I)=0:O(I)=0
2790 NEXT I
2800 FOR I=1 TO N
2810 FOR J=1 TO N
2820 BR(I,J)=0:E(I,J)=0
2830 NEXT J:NEXT I
2840 IF PS=0 THEN 2860
2850 GOTO 900
2860 REM ***** OPTIMAL COZUME ULASILDI *****
2870 T=0
2880 FOR I=1 TO N
2890 FOR J=1 TO N
2900 T=T+BP(I,J)*B(J)
2910 NEXT J
2920 XX(Q2(I))=T:T=0
2930 NEXT I
2940 IF SS=0 THEN 3570
2950 IF LW=0 THEN 3460
2960 P5=0
2970 FOR I=1 TO LW
2980 FOR J=1 TO NN+N
2990 IF XY(I,J)<>XX(J) THEN 3010
3000 P5=P5+1
3010 NEXT J
3020 IF P5=NN+N THEN 3030 ELSE 3440
3030 FB5=1
3040 IF LLL=1 THEN 3850
3050 IF P=1 THEN P6=2 ELSE P6=1
3060 FOR I=1 TO LW
3070 FOR J=1 TO NN+N
3080 T=T+XY(I,J)*DDD(P6,J)
3090 NEXT J
3100 BB(I)=T:T=0
3110 NEXT I
3120 BB=BB(1):KQ=0
3130 FOR I=2 TO LW
3140 IF BB>BB(I) THEN 3160
3150 BB=BB(I)
3160 NEXT I
3170 FOR I=1 TO LW
3180 IF BB<>BB(I) THEN 3200
3190 KQ=KQ+1:YY(KQ)=I
3200 NEXT I
3210 FOR JJ=1 TO KQ
3220 FOR I=1 TO NN+N
3230 XX(I)=XY(YY(JJ),I)
3240 NEXT I
```

```
3250 FOR I=1 TO LLL
3260 FOR J=1 TO NN+N
3270 T=T+XX(J)*DDD(I,J)
3280 NEXT J
3290 AA(P,I)=T:T=0
3300 NEXT I
3310 REM ***** YAZDIRMA BOLUMU *****
3320 LPRINT:LPRINT:LPRINT:LPRINT :LPRINT
3330 LPRINT "                              OPTIMAL COZUM              "
3340 LPRINT "            -----------------------------------------"
3350 FOR I=2 TO M
3360 LPRINT USING "                         \    \= ###.##";X1$(I-1),XX(I)
3370 NEXT I
3380 LPRINT:LPRINT
3390 FOR K=1 TO LLL
3400 LPRINT USING "                    \      \Z# = #####.## ";SV$(K),K,ABS(AA(P,K
))
3410 NEXT K
3420 NEXT JJ
3430 GOTO 3850
3440 P5=0
3450 NEXT I
3460 LW=LW+1
3470 FOR J=1 TO NN+N
3480 XY(LW,J)=XX(J)
3490 NEXT J
3500 T=0
3510 FOR I=1 TO LLL
3520 FOR J=1 TO NN+N
3530 T=T+XX(J)*DDD(I,J)
3540 NEXT J
3550 AA(P,I)=T:T=0
3560 NEXT I
3570 IF GURSEL=1 THEN 3840
3580 LPRINT:LPRINT:LPRINT
3590 LPRINT "                   ";P;". AMAC ICIN OPTIMAL COZUM     "
3600 LPRINT "            -----------------------------------------"
3610 FOR I=2 TO M
3620 LPRINT USING "                         \    \= ###.##";X1$(I-1),XX(I)
3630 NEXT I
3640 LPRINT:LPRINT
3650 FOR K=1 TO LLL
3660 LPRINT USING "                    \      \Z## = #####.## ";SV$(K),K,ABS(AA(P,K
))
3670 NEXT K
3680 PRINT "TAMAM "
3690 FOR I=1 TO NN+N
3700 XX(I)=0
3710 NEXT I
3720 IF FB5=1 THEN 3760
3730 FB1=1
3740 IF ALTER1=1 THEN 900
3750 IF Q=1 THEN 900 ELSE 1240
3760 REM ****** BAZI MATRIS VE DIZILERI SIFIRLA *****
3770 INPUT "   AMACLAR BITTI ",GHG$
3780 D=0:W=0:Q=0:LW=0:FB1=0:W1=0:W5=0:Q8=0:W2=0:I1=0:FB5=0:SS=0:PS=0:ALTER1=0
3790 FOR I=1 TO NN+N
```

```
3800 XX(I)=0
3810 NEXT I
3820 FOR I=1 TO N:FOR J=1 TO N
3830 BP(I,J)=0:NEXT J:NEXT I
3840 FB=0:NEXT P
3850 END
3860 DATA 16,23,2
3870 DATA MAKS,MIN
3880 DATA X12,X13,X14,X23,X28,X35,X38,X45,X46,X48,X57,X58,X67,X68,X78
3890 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3900 DATA 0,-10,-18,-14,-5,0,-10,0,-6,-13,0,-10,0,-5,0,0
3910 DATA 1,-1,-1,-1,0,0,0,0,0,0,0,0,0,0,0,0
3920 DATA 0,1,0,0,-1,-1,0,0,0,0,0,0,0,0,0,0
3930 DATA 0,0,1,0,1,0,-1,-1,0,0,0,0,0,0,0,0
3940 DATA 0,0,0,1,0,0,0,0,-1,-1,-1,0,0,0,0,0
3950 DATA 0,0,0,0,0,0,1,0,1,0,0,-1,-1,0,0,0
3960 DATA 0,0,0,0,0,0,0,0,0,1,0,0,0,1,-1,0
3970 DATA 0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,-1
3980 DATA -1,0,0,0,0,1,0,1,0,0,1,0,1,0,1,1
3990 'DATA 1,0,0,0,0,0,0,0,0,0,0
4000 DATA 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4010 DATA 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
4020 DATA 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
4030 DATA 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
4040 DATA 0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
4050 DATA 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
4060 DATA 0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
4070 DATA 0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
4080 DATA 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
4090 DATA 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
4100 DATA 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
4110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
4120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
4130 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
4140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
4150 DATA 0,0,0,0,0,0,0,0,50,70,60,20,20,30,40,30,45,20,40,15,30,10,25
4160 DATA 2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
```

KAYNAKLAR

1. Belenson, M., Kapur,K.C.," An Algorithm for Solving Multicriterion Lineer Programming, Problems with Examples", O. R. Quarterly,1973.

2. Dantzig,G.B., "Lineer Programming and Extensions",Princeton University Press,New Jersey,1974.

3. Gass,S.I.,"Lineer Programming", Mc Graw Hill Book Company, Inc. New York, 1958

4. Giresunlu,İ.M., "Çok Amaçlı Lineer Kesirli Programlama'yı Hedef Programlamaya dönüştüren üst sınırlar önerisi", Doktora Tezi, 1987.

5. Halaç,O., "Kantitatif Karar Verme Teknikleri" 1984.

6. Kuruüzüm,A., "Çok Amaçlı Karar Verme" Doktora Tezi.

7. Sezginman,İ., "Lineer Programlama", 1986.

8. Tulunay,Y., "İşletme Matematiği", 1982.

ÖZGEÇMİŞ

Coşkun GÜLER 14.08.1965 tarihinde Kastamonu'da doğmuştur. İlk öğrenimini Şişli Kaptanpaşa İlkokulunda 1971-1976 yılları arasında tamamlamıştır. Orta öğrenimini Şişli Kaptanpaşa Ortaokulu'nda 1976-1979 yılları arasında, Şişli Kaptanpaşa Lisesi'ne ise 1979-1982 yılları arasında devam ederek tamamlamıştır. 1982 yılında ise Yıldız Üniversitesi Mühendislik Fakültesi Matematik Mühendisliği Bölümünde yüksek öğrenimine başlamıştır. Lisans öğreniminin son iki yılında Ercanlar Otomotiv Tic.A.Ş. Bilgi İşlem Merkezinde çalışmıştır. 1986-1987 öğretim yılında lisans öğrenimini tamamlamış ve aynı yıl Fen Bilimleri Enstitüsü Matematik Mühendisliği Bölümünde Yüksek Lisans öğrenimine başlamıştır. Yüksek öğrenimine devam ederken 1988 yılında aynı enstitüde Araştırma Görevlisi olarak çalışmaya başlamış olup halen görevine devam etmektedir.