YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

PROSES TASARIMINDAKİ MODELLEME ÇALIŞMALARINA UYGULANABİLECEK GEOMETRİK PROGRAMLAMA İÇİN YENİ BİR YAKLAŞIM

Kim.Müh.Yavuz SALT

F.B.E.Kimya Mühendisliği Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof.Dr.Edip BÜYÜKKOCA

İSTANBUL,1995

İÇİNDEKİLER

Sayfa no

Sayfa no

Sayfa no

SEMBOL LİSTESİ

A,B,C,E,G,P = reaktördeki kimyasal maddeler

F = fonksiyon

M = toplam kısıt sayısı

N = toplam değişken sayısı

T = reaktör sıcaklığı (K)

V = reaktör hacmi (m^3)

A_i = k_i ifadesinde frekans faktörü (i=1,2,3)

B_i = k_i ifadesinde aktivasyon enerjisi (i=1,2,3)

C_n = doğrusal fonksiyonlarda terimlerin katsayıları (n=1,2,...,N)

F_A = A maddesinin besleme debisi (kg/saat)

F_B = B maddesinin besleme debisi (kg/saat)

F_D = atık akımın akış debisi (kg/saat)

F_G = yan ürün olan katranın akış debisi (kg/saat)

F_P = P ürününün üretim debisi (kg/saat)

F_R = reaktörden çıkan akımın toplam akış debisi (kg/saat)

H_i = reaksiyon ısısı (kcal/kg)

M_B, M_C, M_E, M_G, M_P = mol ağırlıkları

T_0 = amaç fonksiyonundaki terim sayısı

T_m = kısıtlardaki terim sayısı (m=1,2,...,M)

P_t = fonksiyonlardaki terimler (t=1,2,...,T_0 veya T_m)

x_n = değişken (n=1,2,...,N)

c_{0t} = amaç fonksiyonundaki terimlerin katsayıları (t=1,2,...,T_0)

c_{mt}, c_{mn} = kısıtlardaki terimlerin katsayıları (t=1,2,...,T_m)

$F_{RA}, F_{RB}, F_{RC}, F_{RE}, F_{RP}$ = reaktörden çıkan akımdaki bileşenlerin akış hızı (kg/saat)

w_{mt} = kısıtlardaki ilgili terimin ağırlığı

b = çözüm vektörü

b_m = kısıtlarda eşitsizliklerin sağ tarafı

d(w) = amaç fonksiyonunun dual çözümü

g_0 = amaç fonksiyonu

g_m = kısıtlar

k_i = reaksiyon hız sabiti (saat^{-1}) (i=1,2,3)

W_t = kısıtsız problemler için ortalama

w_{m0} = kısıtlar için terimlerin dualdeki çarpımları

w_{mt} = kısıttaki terimlere karşılık gelen yardımcı değişkenler

a_{0tn} = amaç fonksiyondaki primal değişkenlerin üsleri

a_{mtn} = kısıtlardaki primal değişkenlerin üsleri

v_{dmt} = doğrusal bağımsız çözümler (d=1,2,...,D)

ρ = yoğunluk (kg/m^3)

σ_m = ilgili kısıtın işareti

σ_{0t} = amaç fonksiyonunda ilgili terimin işareti

σ_{mt} = kısıtlardaki ilgili terimin işareti

ŞEKİL LİSTESİ

Sayfa no

TABLO LİSTESİ

Sayfa no

TEŞEKKÜR

Bu tez çalışmasına olanak sağlayan Bölüm Başkanımız Prof.Dr.Salih DİNÇER'e, tezin hazırlanmasında bilgi ve tecrübelerinden yararlandığım tez danışmanım Prof. Dr.Edip BÜYÜKKOCA'ya teşekkürlerimi sunarım.

Ayrıca, değerli vakitlerini ayırarak bana yardımcı olan Yrd.Doç.Dr.Abdullah YILDIZ'a, tez çalışmalarımda ve tezin hazırlanmasında gösterdikleri itina ve katkılarından dolayı Araş.Gör.Dr.Nalan ADAŞOĞLU, Araş.Gör.Mesut AKGÜN ve Araş. Gör.H.İBrahim ÜNAL'a teşekkürü bir borç bilirim.

ÖZET

Bu çalışmada, kimya mühendisliği açısından önemli etkilerin birçoğunu içeren Williams-Otto prosesi ele alınarak optimizasyonu, Holotransformasyon yönteminin yapay dönüşüm algoritmasıyla gerçekleştirilmiştir. Bu algoritma, denklem sisteminin Newton-Raphson yöntemiyle iteratif olarak çözümü sırasında Jakobiyen matrisi üzerine uygulanmıştır.

Holotransformasyon yöntemi kullanılarak maksimize edilmek istenen amaç fonksiyonu değeri optimuma yakın olarak elde edilmiştir. Bu değer, Williams-Otto prosesinin Geometrik Programlama ile optimizasyonunu gerçekleştiren Rijckaert ve Martens'in çözümünden daha büyüktür.

ABSTRACT

In this study, optimization of Williams-Otto process containing most of the effects important from chemical engineering point of view have been carried out by the artifical variation algorithm of Holotransformation method. This algorithm has been applied to Jakobien matrix during iterative solution of equation system by Newton-Raphson method.

The value of the maximized objective function was attained near the optimum by Holotransformation method. This value is better than solution obtained by Rijckaert and Martens who carried out the optimization of Williams-Otto process applying Geometric Programming.

1.GİRİŞ

Geometrik programlama, doğrusal veya doğrusal olmayan kısıtlara bağlı doğrusal olmayan cebirsel programlama problemlerini çözmek için geliştirilmiş bir tekniktir ve optimizasyon problemlerinin çözümünde kullanılır.

İlk olarak 1961 yılında Clarence Zener tarafından ortaya atılmış ve matematiksel temeli, dualite teorisiyle Richard Duffin tarafından aritmetik-geometrik ortalama eşitsizliğine dayandırılmıştır. Bu yüzden Duffin bu tekniğe **geometrik programlama** demiştir.

Bu teknikte, doğrusal olmayan yapıdaki optimizasyon problemi kısıtları da kapsayacak şekilde, aritmetik ortalamanın genel haliyle yazılır ve bu genel halin geometrik ortalama karşılığını oluşturarak problem, çarpım fonksiyonu haline getirilir. Böylelikle, oluşturulan aritmetik ortalamanın geometrik ortalama karşılığına **predual** denir. Predual üzerinden genel işlem izlenerek, problem, normalite ve ortogonalite koşullarından oluşturulan doğrusal denklem sistemine indirgenir. Bu doğrusal denklem sisteminin çözümünden elde edilen ortalama değerler yardımıyla fonksiyonun değeri dual'den bulunur. En son olarak ortalama ve fonksiyon değerleri kullanılarak değişkenlerin optimum değerleri hesaplanır.

Problemimizin amaç fonksiyonu ve kısıtları sadece pozitif katsayılı terimlerden oluşuyorsa bir pozinomal geometrik programlama; bir veya daha fazla negatif katsayılı terim mevcutsa signomal geometrik programlama söz konusudur.

Amaç fonksiyonunun maksimizasyonu problemiyle karşı karşıya olduğumuzda dual'in minimizasyonu, minimizasyon probleminde ise dual'in maksimizasyonu gerçekleştirilir.

Geometrik programlama genel işlemi ile normalite ve ortogonalite koşullarından elde edilen denklem sisteminin çözümüne yeni bir yaklaşım olarak **holotransformasyon** önerilmektedir.

Holotransformasyon Prof.Dr.Edip Büyükkoca [1974] tarafından geliştirilmiş bir yöntemdir ve yapay dönüşüm, özdeş normal denklemler ve doğrudan yaklaşım olmak üzere üç algoritma bulunmaktadır.

Holotransformasyonda problem, öklit uzayından öklit olmayan uzaya uygun bir dönüştürme matrisi ile yansıtılır. Problemin çözümü daima öklit olmayan uzayda klasik yöntemlerle yapılır. Geri yansıma orijinal problemin çözümü olan öklit uzayındaki çözümü verir.

Geometrik programlama ve geometrik programlama için yeni bir yaklaşım olarak önerilen holotransformasyonun ayrı ayrı uygulandığı Williams-Otto prosesinin oluşturulmasındaki amaç, zaman zaman bazı bilgisayar üreticilerinin kimya ve petrol şirketlerinin bir çoğu ile ortak çalışma programları üstlenmesi ve var olan bilgisayarlar ve onların yeteneklerinin doğrudan karşılaştırılmasına temel oluşturması ümidiyle, bilgisayar kontrolünü içine alarak açıklayan kimyasal proses modelinin araştırılmasıdır. Bu çalışma 1960 yılında Monsanto Kimya Şirketi bünyesinde gerçekleştirilmiştir.

Williams-Otto prosesi karmaşık tasarım probleminin özelliklerinin çoğunu taşımakta ve tipik kimyasal üretim fabrikalarında olası etkilerin bir çoğunu (reaksiyon, ısı aktarımı, toplam ayırma adımları ve geri döngü akımı) kapsamaktadır.

18 143 606 kg/yıl üretim amaçlanmaktadır ve tesis, iki saf besleme ve bir geri döngü akımının girdiği, geri beslemeli reaktör, ısı değiştirici, dekantör ve bir destilasyon kolonundan oluşmaktadır.

1.1.Temel Kavramlar

1.1.1.Optimizasyon

Optimizasyon, bir amaç doğrultusunda, denge şartlarında, seçilen boyut ve birimlerde bir olayı meydana getiren etkenlerin değerlerinin bulunmasıdır. En basit anlamda optimizasyon uygulamaları, seçilen farklı hedeflere göre, alternatif çözümlerin üretilmesi çalışmaları olarak kabul edilebilir. Başka bir ifade ile üretilen ve üretilebilecek bir çok alternatif çözümden seçilen hedef doğrultusunda tek bir çözüm bulmaktır.

Her optimizasyon probleminde her zaman bir amaç bazen alt amaçlar vardır. Örneğin, çevre mühendisliği faaliyetlerimizdeki amaç, kirletici etkenlerin kontrolü, yani bu etkenlerin ölçülen değerlerinin kötü etki sınırlarının altında tutulmasıdır. Bu amaç, modeldeki kirletici faktörlerin toplam etkisinin minimum olması veya kirletici etkinin azlığını ifade eden büyüklüklerin maksimum olmasıdır.

1.1.2.Matematiksel Programlama

Çeşitli yollarla kısıtlanmış bir veya daha fazla değişkenden oluşan sayısal fonksiyonun optimizasyonu problemi, bir matematiksel programlama problemi olarak adlandırılır. Açık olarak, bu tür problemlerin amacı, $x_1, x_2, ..., x_n$ gibi n tane değişkenin değerini, aşağıdaki fonksiyonu verilen kısıtları altında optimize edecek şekilde belirlemektir.

$$z = f(x_1, x_2, ..., x_n)$$

Kısıtlar,

$$g_m(x_1, x_2, ..., x_n) \qquad m = 1, 2, ...M$$

Genellikle n değişkeninin değerinin sayısal olarak negatif olamayacağı kabul edilir. Negatif olmama kısıtı değişkenler üzerinde,

$$x_n \geq 0 \quad n = 1, 2, ..., N$$

şeklinde ifade edilebilir. Bununla beraber aranan, amaç fonksiyon adı verilen "z" fonksiyonunun optimal değeridir.

1.1.3.Model Sınıflandırma

Gerçek hayat problemleri beş ayrı yolla sınıflandırılabilir :

1- Problemdeki fonksiyonel ilişkiler kesin olarak bilinir (deterministik) veya kesin olmayabilir (probabilistik).

2- Yukarıdaki bağıntılardaki $f(x_1, x_2, ..., x_n)$ ve $g_m(x_1, x_2, ..., x_n)$; m=1,2,...M fonksiyonları $x_1, x_2, ..., x_n$ değişkenleri için doğrusaldır veya fonksiyonlardaki en az bir eleman doğrusal olmayabilir.

3- Fonksiyonlar sürekli türevlenebilir (düzgün) veya türevlenemez (düzgün olmayan) olabilirler.

4- Matematiksel programlama problemindeki $x_1, x_2, ..., x_n$ sürekli veya tamsayı değerlerle kısıtlandırılmış olabilirler.

5- Optimizasyon zamanda sabit bir noktada (statik) veya bir zaman aralığında (dinamik) yer alabilirler.

Birçok matematiksel programlama modeli deterministiktir; verilen $x_1, x_2, ...,$

x_n'ler ve $f, g_1, g_2, ..., g_m$'lerin değeri tek olarak belirlenmiştir. Ticari ve ekonomik alanlar için matematiksel programlamanın birçok uygulamasında model fonksiyonlar doğrusaldır. Bunun nedeni, Dantzing [1947] tarafından oluşturulmuş simpleks yönteminin doğrusal programlama problemlerinin çözümü için oldukça etkin bir işlem oluşudur. Ancak bir veya birkaç fonksiyon doğrusal olmayan ise çözüm daha güç olacaktır. Matematiksel programlama problemlerini çözmek için oluşturulmuş algoritmaların büyük çoğunluğu, modeldeki fonksiyonun sürekli türevlenebilir olmasını gerektirir. Dolayısıyla bütün fonksiyonlar düzgün olmalıdır. Aynı zamanda z fonksiyonunun optimizasyonu çoğunlukla zaman içinde sabit bir noktada düşünülmüştür, yani model statiktir.

Matematiksel programlama yöntemlerinin, genel hayat problemlerine uygulanmasındaki birçok örnekte, fonksiyonlar yapıları gereği doğrusal olmayan, düzgün olmayan ve probabilistik olup, problem değişkenlerinin tamsayı değerleri ve problemdeki amaç fonksiyonun optimizasyonunun dinamik olarak belirlenmesi zorunluğu ile karşılaşılır. Buna karşın oluşturulan model sürekli olarak doğrusal-düzgün-deterministik-sürekli-statik bir modeldir. Model kurucunun beklentisi, oluşturulmuş olan matematiksel programlama modelinin gerçek hayat problemlerinin çözümlenmesi sırasında kullanışlı bilgi sağlayabilmesidir.

Yukarıda anlatılan matematiksel programlama modeli aşağıdaki gibi iki kısma ayrılabilir :

1- Amaç fonksiyon ve kısıtlar altında,

$$z = f(x_1, x_2, ..., x_n) = \sum_{n=1}^{N} C_n x_n$$

$$g_i(x_1, x_2, ..., x_n) = \sum_{n=1}^{N} a_{mn} x_n$$

$$\{\leq, =, \geq\} b_m \; ; \; m = 1, 2, ..., M$$

$$x_n \geq 0 \; ; \; n = 1, 2, ..., N$$

yapısının optimizasyonu doğrusal programlama modelidir. Burada c_n, a_{mn}, b_m'ler bilinen sabitlerdir.

2- Amaç fonksiyon ve kısıtlar altında

$$z = f(x_1, x_2, ..., x_n)$$

$$g_m(x_1, x_2, ..., x_n) \quad \{\leq, =, \geq\} b_m \; ; \; m = 1, 2, ..., M$$

$$x_n \geq 0 \; ; \; n = 1, 2, ..., N$$

şeklindeki yapısının optimizasyonu, ancak $f, g_1, g_2, ..., g_m$ fonksiyon kümesinde en az bir fonksiyonun doğrusal olmağı durum için doğrusal olmayan programlama modelidir.

2.DOĞRUSAL OLMAYAN PROGRAMLAMA

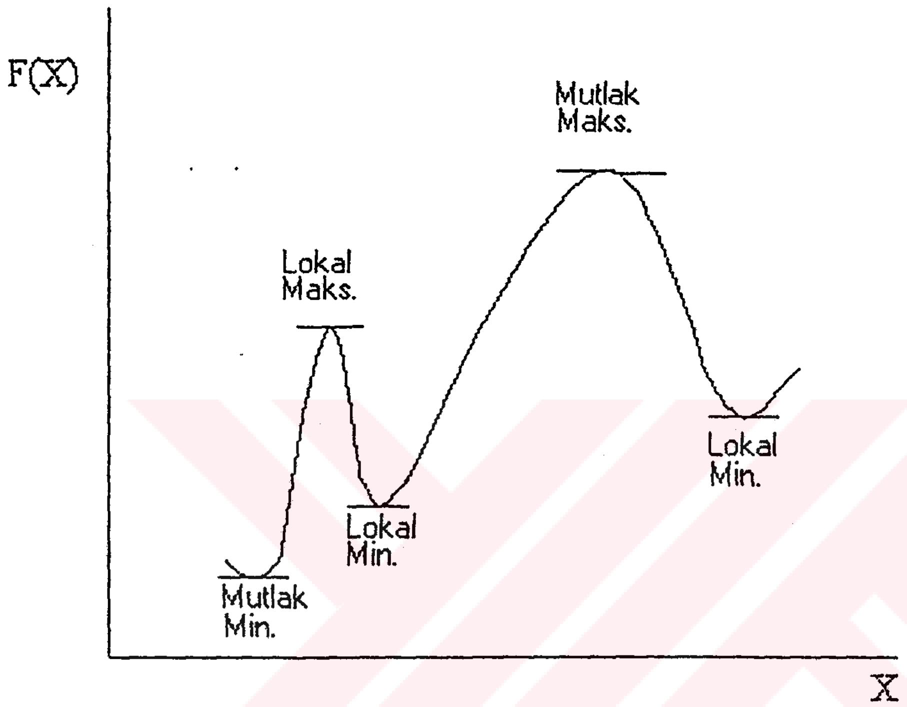
Pratikte karşılaşılan modellerin birçoğunda gerek amaç denklemi ve gerekse sınırlayıcı koşullar doğrusal değildir. Bazı durumlarda doğrusal olmayan bu ifadeler yerine, yaklaşık doğrusal ifadeler kullanılabilir ya da bu eğriyi yaklaşık olarak tanımlayan doğrulardan yararlanılır. Ancak bu yaklaşımlar her zaman kullanılamaz.

2.1.Klasik Optimizasyon

Verilmiş olan herhangi bir fonksiyonu maksimize veya minimize eden değişken veya değişkenler setine ait değerlerin bulunması mühendislikte önemli bir yere sahiptir. Çünkü mühendislik problemlerinin çoğunda maksimize veya minimize edilmesi gereken bir amaç denklemi vardır.

Genel halde verilmiş olan herhangi bir fonksiyonun birden çok maksimum veya minimum noktası bulunabilir. Örneğin, tek değişkenli bir y=f(x) fonksiyonunun eğrisi Şekil 2.1'deki gibiyse, burada birden çok maksimum veya minimum nokta vardır. Ancak bu ekstremumlardan sadece bir tanesi asıl minimum noktayı (mutlak minimum), bir tanesi de asıl maksimum noktayı (mutlak maksimum) gösterecektir. Bunların dışında kalan maksimum ve minimum noktalar ise lokal maksimum ve lokal minimum olarak adlandırılırlar.

y = f(x) tek değişkenli fonksiyonunda $x = x''$ değerinin bir ekstremum tanımlayabilmesi için gerekli koşul;



Şekil 2.1. y=f(x) fonksiyonu eğrisi

$$\frac{df}{dx} = 0 \qquad x = x''$$

olarak verilir. Dikkat edilirse, maksimum ve minimum noktalarda eğriye çizilen teğet doğrular x eksenine paralel olmakta ve dolayısıyla fonksiyonun birinci türevi ile de belirlenebilen eğrinin o noktasındaki eğimi yani df/dx ifadesi sıfır olmaktadır. Bu ifadenin sıfıra eşitlenmesiyle elde edilen denklemin çözümü, genel halde birden çok x^* değeri verecektir. Bunlardan hangisinin maksimum veya hangisinin minimum noktalar gösterdiğini anlamak için ikinci türevlerinin incelenmesi gerekmektedir. Buna göre, eğer ;

$$\frac{d^2f(x)}{dx^2} > 0 \text{ ise } (x = x^*)$$

incelenmekte olan x^* değerinin bir minimum (mutlak veya bölgesel) olduğu anlaşılır. Yani x^*'a yeterince yakın bütün x değerleri için $f(x^*) \leq f(x)$ olacaktır. Eğer ;

$$\frac{d^2f(x)}{dx^2} < 0 \text{ ise } (x = x^*)$$

x^*'a yeterince yakın bütün x değerleri için $f(x^*) \geq f(x)$ olacağından, incelenmekte olan x^* değeri bir maksimum gösterir.

2.1.1.Ekstremumlar

Eğer $f(x)$ bir x^* komşuluğu ile tam konveks ise, bu x^* değeri için bir minimum nokta söz konusu olacaktır. Bu nokta bölgesel veya mutlak olabilir. Benzer şekilde, x^* değerinin bir maksimum gösterebilmesi için, gerekli koşullar gerçekleştikten sonra, yeter koşul $f(x)$ fonksiyonunun bir x^* komşuluğu ile tam konkav olmasıdır. O halde ;

$$\text{maksimum icin} \quad \frac{d^2f(x)}{dx^2} < 0 \qquad (x = x^*)$$

$$\text{minimum icin} \quad \frac{d^2f(x)}{dx^2} > 0 \qquad (x = x^*)$$

İkinci türev sıfır olduğu taktirde belirsizlik vardır ve daha yüksek mertebeden türevlerin incelenmesi gerekir.

Bütün x'ler için $f(x^*) \leq f(x)$ koşulunu gerçekleyen x^* mutlak minimum (global minimum) değerini bulabilmek için, bulunmuş olan minimum noktalar içinden, $f(x)$ fonksiyonuna en küçük değeri kazandıran x^* değerinin seçilmesi gerekir. Bu şekilde bulunan $f(x)$ değeri, $x = -\infty, +\infty$ veya fonksiyonun bitim noktalarının belirlediği aralıkta bütün x değerlerinden küçük olmalıdır.

Eğer $f(x)$ fonksiyonunun konveks mi yoksa konkav mı olduğu önceden biliniyorsa yapılan işlemler basitleşecektir. $f(x)$ fonksiyonu Şekil 2.2a'daki gibi konveks ise,

$$\frac{df}{dx} = 0 \qquad x = x^*$$

koşulunu gerçekleyen herhangi bir x^* değeri doğrudan doğruya mutlak minimum gösterecektir.

Eğer $f(x)$ fonksiyonu Şekil 2.2b'deki gibi konkav ise,

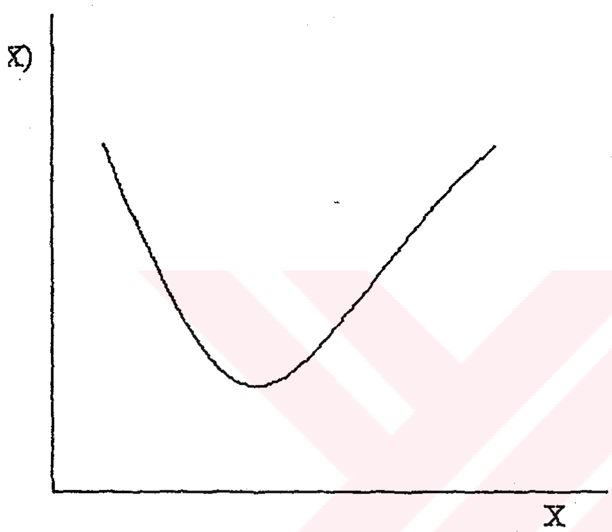
$$\frac{df}{dx} = 0 \qquad x = x^*$$

koşulu, x^* değerinin mutlak maksimum olması için gerek ve yeter koşul olacaktır.

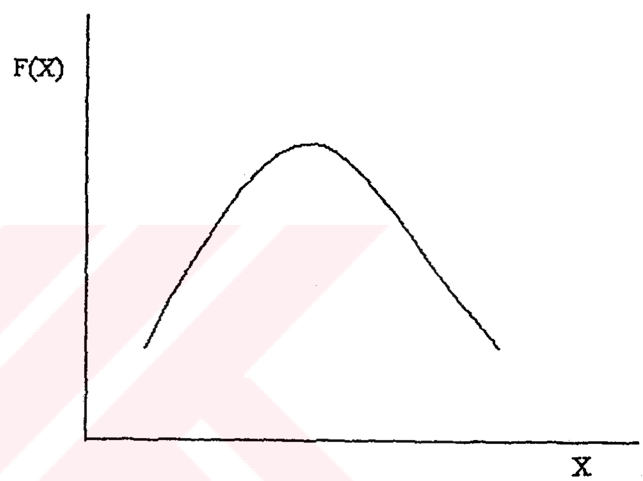
Sınırlanmış çok değişkenli fonksiyonlarda da durum benzerdir. Yani bir $f(x_1, x_2, ..., x_n)$ fonksiyonunda, $(x_1, x_2, ..., x_n) = (x_1^*, x_2^*, ..., x_n^*)$ setinin bir maksimum veya minimum gösterebilmesi için ;

$$\frac{\partial f(x_1, x_2, ..., x_n)}{\partial x_n} = 0 \qquad (n = 1, 2, ..., N \text{ icin})$$

koşulları gerçeklenmelidir.



Şekil 2.2a.Konveks fonksiyon



Şekil 2.2b.Konkav fonksiyon

3.GEOMETRİK PROGRAMLAMA

İlk olarak Duffin Peterson ve Zener (1967) tarafından ortaya atılıp geliştirilen geometrik programlama, herhangi bir dereceden fonksiyonu ve sınırlayıcı koşulları bulunduran, toplam ve farklar içeren bir optimizasyon problemini; çarpımlar içeren problemlere dönüştürülebilmektedir.

Ayrıca, aritmetik ve geometrik ortalama eşitsizliğinden yararlanılarak elde edilen çarpım fonksiyonunun logaritmasından kısmi türevler oluşturularak, problem doğrusal olmayan yapıdan, kolaylıkla çözülebilecek doğrusal denklem sistemine de indirgenebilmektedir.

3.1.Aritmetik ve Geometrik Ortalamalar

İki sayı veya fonksiyon verildiğinde bunun aritmetik ortalaması $(1/2)U_1+(1/2)U_2$ dir. Bu iki sayının geometrik ortalaması ise $U_1^{1/2}.U_2^{1/2}$'dir. Bazı fonksiyonların ve sayıların aritmetik ve geometrik ortalamaları aşağıda verilmiştir ;

ARİTMETİK	GEOMETRİK
$(1/3)x_1+(1/3)x_2+(1/3)x_3$	$x_1^{1/3}.x_2^{1/3}.x_3^{1/3}$
$(1/2)U_1+(1/2)U_2$	$U_1^{1/2}.U_2^{1/2}$

Negatif olmayan sayı gruplarının veya fonksiyonların aritmetik ortalaması, her zaman geometrik ortalamasından büyüktür veya geometrik ortalamasına eşittir.

$$(1/2)U_1 + (1/2)U_2 \geq U_1^{1/2}.U_2^{1/2} \qquad [3.1]$$

Eğer U_1 ve U_2 negatif değilse,

$$(U_1 - U_2)^2 \geq 0$$

$$U_1^2 - 2U_1U_2 + U_2^2 \geq 0$$

$$U_1^2 + 2U_1U_2 + U_2^2 \geq 4U_1U_2$$

$$\left[(1/4)U_1^2 + (1/2)U_1U_2 + (1/4)U_2^2\right]^{1/2} \geq (U_1U_2)^{1/2}$$

veya

$$(1/2)U_1 + (1/2)U_2 \geq (U_1U_2)^{1/2}$$

Böylece [3.1] eşitsizliği kanıtlanmış oldu. Örneğin, aşağıdaki

$$f(x) = 4x + \frac{16}{x} \qquad [3.2]$$

fonksiyonu ele alındığında,

$$(1/2)8x + (1/2)32x^{-1}$$

şeklinde yazılırsa daha kolay çözülebilir hale gelir. Bu ifadeye karşılık gelen geometrik ortalama,

$$(8x)^{1/2}.(32/x)^{1/2}$$

olur.

3.1.1.Genel Bağıntı

Her aritmetik ortalama için buna karşılık gelen çeşitli geometrik ortalamalar oluşturulabilir. [3.2] eşitliğindeki aritmetik ortalama ve bunun geometrik ortalama karşılığı ve bunların arasındaki ilişki şu şekildedir :

$$w_1\left(\frac{4x}{w_1}\right)+w_2\left(\frac{16x^{-1}}{w_2}\right)\geq\left(\frac{4x}{w_1}\right)^{w_1}.\left(\frac{16x^{-1}}{w_2}\right)^{w_2}$$

buradaki ağırlıklar w_1 ve w_2'nin toplamı 1'dir.

Aritmetik ve geometrik ortalamalar arasındaki genel ilişki, aritmetik ortalama $U_1+U_2+...$ şeklindeyse

$$w_1\left(\frac{U_1}{w_1}\right)+w_2\left(\frac{U_2}{w_2}\right)+...\geq\left(\frac{U_1}{w_1}\right)^{w_1}.\left(\frac{U_2}{w_2}\right)^{w_2}...$$

Her aritmetik ortalama için buna karşılık gelen birçok geometrik ortalama vardır. İşte bu yüzden iki ortalamanın eşit olmasını sağlayacak $w_1, w_2, ..., w_n$ değerler kümesi vardır. Genelde, eğer geometrik ortalamanın ağırlıksız terimleri $(U_1, U_2, ..., U_n)$ eşitse aritmetik ortalama geometrik ortalama karşılığına eşittir.

3.1.2. Aritmetik ve Geometrik Ortalamalarda Ekstrem Durum

Bazı spesifik ortalamaların işlenmesinde, U_i şeklinde gösterilen her terimi bir çarpım fonksiyonu (bir veya daha fazla değişkenli) olarak temsil edilir. U_i her zaman şu şekilde olacaktır ;

$$cx_1^{a1}x_2^{a2}x_3^{a3}...,x_n^{an}$$

$c, a_1, a_2, ..., a_n$ sabittir. Şu şekilde bir aritmetik ortalama ele alınırsa :

$$2x_1x_2^{-1} + 5x_2x_3 + x_1^{-2}x_2x_3^{-1}$$

Geometrik ortalaması :

$$\left(\frac{2x_1x_2^{-1}}{w_1}\right)^{w_1} . \left(\frac{5x_2x_3}{w_2}\right)^{w_2} . \left(\frac{x_1^{-2}x_2x_3^{-1}}{w_3}\right)^{w_3}$$

Bu ifade şu şekilde de yazılabilir :

$$\left(\frac{2}{w_1}\right)^{w_1} . \left(\frac{5}{w_2}\right)^{w_2} . \left(\frac{1}{w_3}\right)^{w_3} .x_1^{(w_1-2w_3)} .x_2^{(-w_1+w_2+w_3)} .x_3^{(w_2-w_3)} \qquad [3.3]$$

Ekstremlerde kontrol değişkenlerinin üsleri sıfıra gidecektir. Böylece ;

$$w_1 - 2w_3 = 0$$

$$-w_1 + w_2 + w_3 = 0$$

$$w_2 - w_3 = 0$$

Eğer bu üsler sıfıra giderse x_1^0, x_2^0, ve x_3^0 olarak 1 değerlerini alırlar. Yukarıdaki doğrusal denklem sistemi çözülürse w_1, w_2, w_3 ağırlıkları bulabiliriz. Böylece [3.3] bağıntısının kalan sol tarafı hesaplanarak (g_0) ekstremi elde edilir.

[3.2] bağıntısında, geometrik ortalama şekli tekrar düzenlendiğinde,

$$\left(\frac{4}{w_1}\right)^{w_1} . \left(\frac{16}{w_2}\right)^{w_2} . x^{(w_1 - w_2)}$$

Buradan,

$$w_1 - w_2 = 0$$

$$w_1 + w_2 = 1$$

olduğu görülebilir.

Bu doğrusal denklem sisteminin çözümünden $w_1 = w_2 = 1/2$ olarak elde edilir. [3.2]'nin ekstremi böylece,

$$\left(\frac{4}{1/2}\right)^{1/2} . \left(\frac{16}{1/2}\right)^{1/3} = 16$$

x'in ne olabileceğine bakılmaksızın [3.2] bağıntısı 16'da ekstrem değerindedir.

Herhangi bir aritmetik ortalama için, buna karşılık gelen çeşitli geometrik ortalamalar oluşturulabilmekte ve oluşturulan geometrik ortalamalardan en az biri aritmetik ortalamaya eşit olmaktadır. $w_1 = w_2 = 1/2$'de aritmetik ve geometrik ortalamalar eşitse [3.2] için şu yazılabilir :

$$(1/2)8x + (1/2)32x^{-1} = 16$$

Buradan $x = \pm 2$ olarak elde edilir ve bir minimumdur. Bu durumda geometrik ortalamanın minimum değeri olduğu kanıtlanmıştır. Aritmetik ortalama 16'dan küçük olamayacağından geometrik ortalama da 16'dan büyük olamaz ve 16 ekstrem değeri geometrik ortalamanın maksimum değeridir.

Bu yöntem önemli bir optimizasyon aracıdır. Pozitif katsayılı terimlerin çarpım fonksiyonları olan, doğrusal olmayan fonksiyonları minimize etmek için, bu fonksiyonun karşılığı olan geometrik ortalamanın maksimize edilmesi gerekir.

3.1.3.Dual ve Predual Fonksiyonu

Eğer minimize edilecek fonksiyon, $U_1 + U_2 + ... + U_n$ ise aritmetik ortalama,

$$w_1 \left(\frac{U_1}{w_1}\right) + w_2 \left(\frac{U_2}{w_2}\right) + ... + w_n \left(\frac{U_n}{w_n}\right) = \sum_{n=1}^{N} w_n \left(U_n / w_n\right) \qquad [3.4]$$

olacaktır. Geometrik ortalama karşılığı;

$$\left(\frac{U_1}{w_1}\right)^{w_1} . \left(\frac{U_2}{w_2}\right)^{w_2} ..... \left(\frac{U_n}{w_n}\right)^{w_n} = \prod_{n=1}^{N} \left(U_n / w_n\right)^{w_n} \qquad [3.5]$$

[3.5] ifadesine [3.4] ifadesinin **preduali** denir. Yani aritmetik ortalamanın geometrik ortalama karşılığı predual'dir.

Predual fonksiyonunun, kontrol değişkenlerini sağ tarafa ayırmak suretiyle kalan kısma **dual** denir. Ağırlıklar optimum değerini aldığında dualin değeri minimize edilen orijinal fonksiyonun minimum değeridir.

[3.3] ifadesine bakacak olursak;

$$\left(\frac{2}{w_1}\right)^{w_1} . \left(\frac{5}{w_2}\right)^{w_2} . \left(\frac{1}{w_3}\right)^{w_3} .x_1^{(2w_1 - 2w_3)} .x_2^{(-w_1 + 3w_2 + w_3)} .x_3^{(w_2 + 2w_3)}$$

ifadenin tamamı **predual** olarak adlandırılır.

3.2.Geometrik Programlama Problemlerinin Genel Yapısı

Terminoloji ve notasyonlara bağlı olarak kurulan en genel şekil aşağıda olduğu gibi ifade edilebilir.

$$\text{min. } g_0(x) = \sum_{t=1}^{T_0} \sigma_{0t} c_{0t} \prod_{n=1}^{N} (x_n)^{a_{0tn}}$$

$$\text{kisitlar } g_m(x) = \sum_{t=1}^{T_m} \sigma_{mt} c_{mt} \prod_{n=1}^{N} (x_n)^{a_{mtn}} \leq \sigma_m$$

m = 1,2,...,M, $x_i > 0$ olup burada,

$$\sigma_{mk} = \pm 1 \qquad c_{mk} > 0$$

$$\sigma_{0k} = \pm 1 \qquad c_{0k} > 0$$

$$\sigma_m = \pm 1$$

şeklinde tanımlanır. Ayrıca,

σ_{mt}, σ_{0t} : işaret fonksiyonları,

T_m : m'inci kısıttaki terim sayısı,

T_0 : amaç fonksiyondaki terim sayısı,

Eğer tüm σ değerleri pozitif ise yapı **pozinomal** olarak adlandırılır. Örneğin,

$$f(x_1, x_2, x_3) = x_1^2 x_2 x_3^3 - \sqrt{2} x_2^4 + (5/7) x_1^5 x_2^3 + x_2$$

fonksiyonu üç değişken ve dört terimden oluşmuş, derecesi sekiz olan bir yapıdadır. Pozinomal fonksiyonda a_{mtn} kuvvet değerleri herhangi reel sayı olabilir. Ancak

c_{mt} katsayılarına pozitiflik şartları getirilmiştir. Buna göre, yukarıdaki fonksiyon $-\sqrt{2}$, katsayısı bulunduran terim içerdiğinden pozinomal olamaz. Ayrıca,

$$f(x_1, x_2) = 2.31x_1x_2^2 + x_1^{-1}x_2^{0.5} + x_1^{0.19}$$

fonksiyonu bir pozinomal olup, tamsayı olmayan kuvvetler içerdiği için bir polinomal değildir. Sonuç olarak, $f(x) = \pi\, x^5$ hem pozinomal hem de polinomal iken $f(x) = -x^{0.5}$ ne pozinomal ne de polinomaldir. Ancak a_{mtn}'lerin yine herhangi reel değerler alabildiği "signomal fonksiyonlar" sınıfına dahildir. Signomaller kuvvet terimleri dışında polinomallere benzerler.

3.3.Pozinomal Geometrik Programlama

Bütün terimlerin katsayılarının işareti pozitif olan fonksiyonlara **pozinomal fonksiyonlar** denir. Bu tip fonksiyonların genel şekli :

$$g \equiv g(\underline{x}) = \sum_{t=1}^{T} c_t P_t(\underline{x}) \qquad [3.6]$$

c_t pozitif sayılar ve $\underline{x} \equiv (x_1, x_2, ..., x_n)$'dir. Ayrıca $P_t(\underline{x})$ fonksiyonları :

$$P_t(\underline{x}) = \prod_{n=1}^{N} x_n^{a_{tn}} \qquad [3.7]$$

şeklinde tanımlanmaktadır. a_{tn} gerçek sayılardır. g bir polinomal değildir ve terimlerin katsayıları pozitif olmak zorundadır. Üsler ise pozitif tamsayılarla sınırlanmamıştır.

Amaç, g amaç fonksiyonunu minimize etmektir. Bu bileşenlerin maliyetlerinin tanımlarından oluşan bir toplam maliyet hesabı olarak düşünülebilir.

3.3.1.Kısıtsız Pozinomal Fonksiyonlar

Kısıtlanmamış pozinomal bir fonksiyon için dual geometrik program ;

$$\text{Optimize } \; d(w) = \prod_{t=1}^{T} \left[\frac{c_t}{w_t}\right]^{w_t} \qquad [3.8]$$

ve

$$\sum_{t=1}^{T} a_{tn} w_t = 0 \quad \text{ortogonalite kosulu} \qquad [3.9]$$

$$\sum_{t=1}^{T} w_t = 1; \quad w_t > 0; \quad \text{normalite kosulu} \qquad [3.10]$$

Dual doğrusal kısıt seti konveks bir bölge gösterir. Dual amaç fonksiyonunun yapısını inceleyebilmek için z(w) = ln [d(w)] alınırsa;

$$z(w) = ln \prod_{t=1}^{T} \left[\frac{c_t}{w_t}\right]^{w_t} = -\sum_{t=1}^{T} w_t \, ln \left[\frac{w_t}{c_t}\right] \qquad [3.11]$$

Fonksiyon z(w) ağırlıklara göre konkavdır. Dual problem, konveks kısıtlarla beraber konkav bir amaç fonksiyon için sabit bir nokta gösterir. Fonksiyonu bir mutlak maksimum gösteren sabit bir noktaya götürür. Böylece, dual problem normalite ve ortogonalite koşullarıyla maksimize z(w)'dir.

Aritmetik-geometrik eşitsizlik temelini yazacak olursak ;

$$\sum_{t=1}^{T} c_t P_t \;\geq\; \prod_{t=1}^{T} \left[\frac{c_t}{w_t}\right]^{w_t} \qquad [3.12]$$

Sonuç olarak, dual problem için elde edilen mutlak maksimum primal problemin mutlak minimumuna eşittir denilebilir.

3.3.2.Kısıtlı Pozinomal Fonksiyonlar

Kısıtlanmış bir pozinomal yapının genel şekli aşağıdaki gibidir ;

$$\text{Minimize} \quad g_0(\underline{x}) = \sum_{t=1}^{T_0} c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} \qquad [3.13]$$

Kısıtlar

$$g_m(\underline{x}) = \sum_{t=1}^{T_m} c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \leq 1 \quad \text{m} = 1, 2, ...\text{M} \qquad [3.14]$$

Amaç fonksiyon için bir, kısıtlar için M tane ifade vardır. Her bir pozinomal terimi, m=1,2,...,M olmak üzere T_m gösterir.

Optimizasyon problemi doğrusal veya doğrusal olmayan eşitsizlik kısıtları altında $y_0(\underline{x})$'i minimize etmektir. Bu eşitlik ve eşitsizliklerden yola çıkarak genel pozinomal için dual geometrik program ;

$$\text{Maksimize} \quad d(\underline{w}) = \prod_{m=0}^{M} \prod_{t=1}^{T_m} \left[\frac{c_{mt} w_{m0}}{w_{mt}} \right]^{w_{mt}} \qquad [3.15]$$

$$\sum_{t=1}^{T_0} w_{0t} = 1 \qquad [3.16]$$

$$\sum_{m=0}^{M} \sum_{t=1}^{T_m} a_{mtn} w_{mt} = 0 \ ; \quad \text{n} = 1, 2, ...\text{N} \qquad [3.17]$$

$$w_{m0} = \sum_{t=1}^{T_m} w_{mt} \ ; \quad \text{m} = 1, 2, ...\text{M} \qquad [3.18]$$

$$w_{00} = 1$$

Optimal primal amaç fonksiyonu, optimal ağırlıklarla,

$$d(w^*) = g_0(x^*)$$

olmak üzere,

$$w_{0t}^*[g_0(x^*)] = c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} \quad ; \quad \mathrm{t} = 1, 2, \ldots \mathrm{T_0} \qquad [3.19]$$

Genelleştirilmiş ağırlıklar, herbiri bir terime karşılık gelmek üzere,

$$w_{mt} = \frac{w_{mt}}{\sum_{t=1}^{T_m} w_{mt}} = c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \qquad [3.20]$$

$m = 1, 2, \ldots, M$ ve $\mathrm{t} = 1, 2, \ldots, \mathrm{T_m}$

3.4.Signomal Geometrik Programlama

Bazı geometrik programlama problemlerinde negatif katsayılara sahip terimler vardır. Maksimizasyon problemi böyle bir olaya iyi bir örnek olabilir. Eğer bir geometrik program bir veya daha fazla negatif katsayılı terimler içerirse, bir signomal geometrik programlama sözkonusu demektir.

Aşağıdaki yapıda T terimli ve N değişkenli doğrusal olmayan amaç fonksiyonu incelensin:

$$\text{Minimize} \quad g_0(\underline{x}) = \sum_{t=1}^{T_0} \sigma_{0t} c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} \qquad [3.21]$$

Bu fonksiyon M kadar eşitsizlik kısıtıyla sınırlandırılmıştır ;

$$g_m(\underline{x}) = \sum_{t=1}^{T_m} \sigma_{mt} c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \leq \sigma_m \quad \mathrm{m} = 1, 2, ..., \mathrm{M} \qquad [3.22]$$

Burada ;

$\sigma_{mt} = \pm 1$; $t = 1, 2, ..., T_m$; $m = 0, 1, ..., M$

$\sigma_m = \pm 1$; $m = 1, 2, ..., M$

$c_{mt} > 0$; $t = 1, 2, ..., T_m$; $m = 0, 1, ..., M$

$x_n > 0$; $n = 1, 2, ..., N$

şeklindedir.

3.4.1.Pozitif Katsayılı Amaç Fonksiyonu

Amaç fonksiyondaki bütün işaret fonksiyonları pozitif olduğunda [3.21] ifadesi aşağıdaki yapıya indirgenir ;

$$\text{Minimize} \quad g_0(\underline{x}) = \sum_{t=1}^{T_0} c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} \qquad [3.23]$$

Kısıtlar

$$g_m(\underline{x}) = \sum_{t=1}^{T_m} \sigma_{mt} c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \leq \sigma_m \qquad [3.24]$$

$x_n > 0$; $n = 1, 2, ..., N$

$\sigma_m = \pm 1$; $m = 1, 2, ..., M$

$\sigma_{mt} = \pm 1$; $t = 1, 2, ..., T_m$; $m = 0, 1, ..., M$

Bu geometrik programlama probleminin çözümü için eş **dual** geometrik program,

$$\text{Optimize} \quad d(\underline{w}) = \prod_{m=0}^{M} \prod_{t=1}^{T_m} \left[\frac{c_{mt} w_{m0}}{w_{mt}} \right]^{\sigma_{mt} w_{mt}} \qquad [3.25]$$

Kısıtlar ;

$$\sum_{t=1}^{T_0} w_{0t} = 1 \qquad [3.26]$$

$$\sum_{m=0}^{M} \sum_{t=1}^{T_m} \sigma_{mt} a_{mtn} w_{mt} = 0 \ ; \mathrm{n} = 1, 2, ...\mathrm{N} \qquad [3.27]$$

$$w_{m0} = \sigma_m \sum_{t=1}^{T_m} \sigma_{mt} w_{mt} \quad \geq 0 \qquad [3.28]$$

σ_m ve $\sigma_{mt} = 1 \quad \mathrm{m} = 0$ için

$w_{mt} \geq 0$ bütün m ve t için

$w_{00} = 1$

$$W_{mt} = \frac{w_{mt}}{\sigma_m \sum_{t=1}^{T_m} \sigma_{mt} w_{mt}} = \frac{w_{mt}}{w_{m0}} \qquad [3.29]$$

w^* optimal dual değişkenleri bilindiğinde x_n^* primal değişkenlerin değerleri aşağıdaki ifadeden bulunabilmektedir.

$$c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} = w_{0t} g_0(x^*) \qquad [3.30]$$

$$c_{mt}\prod_{n=1}^{N}x_n^{a_{mtn}} = \frac{w_{mt}}{\sigma_m\sum_{m=1}^{T_m}\sigma_{mt}w_{mt}} \quad [3.31]$$

$t = 1, 2, ..., T_m \quad ve \quad m = 1, 2, ..., M$

Her iki eşitlikte de eşitliğin her iki tarafının da logaritması alınırsa, oluşan doğrusal-logaritmik sistem çözülmek suretiyle değişkenler hesaplanabilir.

3.4.2.Negatif Katsayılı Amaç Fonksiyonu

Amaç fonksiyonunda negatif katsayılı terimler mevcut olduğunda, signomal programlar için en genel durum ;

$$\text{Minimum} \quad g_0(\underline{x}) = \sum_{t=1}^{T_0}\sigma_{0t}c_{0t}\prod_{n=1}^{N}x_n^{a_{0tn}} \quad [3.32]$$

Kısıtlar

$$g_m(\underline{x}) = \sum_{t=1}^{T_m}\sigma_{mt}c_{mt}\prod_{n=1}^{N}x_n^{a_{mtn}} \leq \sigma_m \quad [3.33]$$

m = 1,2,...,M

$x_n > 0 \quad ; n = 1, 2, ..., N$

3.4.3.Pozitif Değerli Amaç Fonksiyonu

Signomal amaç fonksiyonunun optimal değeri pozitif $(g_0^*(x) > 0)$ olduğunda durum;

$$\text{Minimize} \quad x_0$$

Kısıtlar

$$\hat{g}_0(\underline{x}) = \sum_{t=1}^{T_0} \sigma_{0t} c_{0t} x_0^{-1} \prod_{n=1}^{N} x_n^{a_{0tn}} \leq 1 \qquad [3.34]$$

$$g_m(\underline{x}) = \sum_{t=1}^{T_m} \sigma_{mt} c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \leq \sigma_m \qquad [3.35]$$

m = 1,2,...,M

$x_n > 0 \quad ; n = 1, 2, ..., N$

Burada

$\hat{g}_0(\underline{x}) = x_0^{-1} g_0(\underline{x})$

şeklindedir.

$w_{00} = 1$ (normalite koşulu)

$w_{00} - \sigma_{01} w_{01} - \sigma_{02} w_{02} - ... - \sigma_{0,T_0} w_{0,T_0} = 0$

Bu iki eşitlik birleştirilirse ;

$$\sigma_{01} w_{01} + \sigma_{02} w_{02} + ... + \sigma_{0,T_0} w_{0,T_0} = 1 \qquad [3.36]$$

elde edilir.

Karşılık gelen işaret fonksiyonları negatif olduğundan toplamları yerine farkları alınan dual değişkenlerle bağıntısı, orijinal amaç fonksiyon için bir normalite kısıtı gibi görünür. Bu eşitlik **genelleştirilmiş normalite koşulu** olarak adlandırılır.

Problemin çözümünde orijinal problemdeki sayıda denklem, dual değişken ve zorluk derecesi bulunacaktır. Negatif terimlerden gelen kuvvetlerin işaret değişiminden emin olunmalıdır. Bu zorluktan, problemin orijinal formu yerine amaç fonksiyon için,

$$\sum_{t=1}^{T_0} \sigma_{0t} w_{0t} = 1 \qquad [3.37]$$

normalite koşulu kullanılarak kurtulunabilir. Bu durum ortogonalite koşulu için,

$$\sum_{m=0}^{M} \sum_{t=1}^{T_m} \sigma_{mt} a_{mtn} w_{mt} = 0 \quad \mathrm{n} = 1, 2, ..., \mathrm{N} \qquad [3.38]$$

şeklindedir.

3.4.4.Negatif Değerli Amaç Fonksiyonu

Amaç fonksiyonunun optimal değerinin negatif ($g_0(\underline{x}^*) < 0$) olduğu varsayalırsa;

$$x_0 \leq g_0(\underline{x}) \qquad [3.39]$$

bağıntısını koruyacak şekilde dönüştürülmelidir. Buradan,

$$x_0^{-1} g_0(\underline{x}) \geq 1$$

veya

$$-1.0[x_0^{-1}] g_0(\underline{x}) \leq -1 \qquad [3.40]$$

elde edilir. [3.40] bağıntısı, her σ_{0t}, t=1,2,...T_0 ve eşitsizliğin her iki tarafındaki işaret farkı dışında [3.34] ile aynı yapıdadır. Amaç fonksiyon yine x_0'ın minimizasyonu ve dual kısıtlar kümesi,

$$w_{00} = 1$$

$$w_{00} + (-1.0)^2 \sum_{t=1}^{T_0} \sigma_{0t} w_{0t} = 0$$

bağıntılarını korur. Buradan,

$$\sum_{t=1}^{T_0} \sigma_{0t} w_{0t} = -w_{00} = -1$$

olduğu görülebilir.

Daha önce olduğu gibi problemi x_0'ın minimizasyonu şeklinde yeniden düzenlemeye gerek yoktur. Dikkat edilmesi gereken, genelleştirilmiş normalite koşulunun,

$$\sum_{t=1}^{T_0} \sigma_{0t} w_{0t} = -1 \qquad [3.41]$$

bağıntısı ile verilecek olmasıdır. Dual geometrik program için ortogonalite koşulu yine aynıdır. Bununla beraber optimallikte $g_0^*(\underline{x})$'in değeri,

$$g_0^*(\underline{x}) = -[\text{minimum } x_0]^{-1} = -[x_0^*]^{-1} \qquad [3.42]$$

olduğu için $d^*(w)$ dual çözümü değildir. Optimallikte primal ve dual problemlerin eşitliğini hedeflediğimiz için dual problem ;

$$d(\underline{w}) = -1.0\left[\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left[\frac{c_{mt}w_{m0}}{w_{mt}}\right]^{\sigma_{mt}w_{mt}}\right]^{-1.0} \qquad [3.43]$$

Her iki durumda $y_0^*(\underline{x})$'in işareti olan yeni bir σ_0 fonksiyonu tanımlayarak tek fonksiyon ile ifade edilebilir. O zaman genelleştirilmiş normalite koşulu ;

$$\sum_{t=1}^{T_0}\sigma_{0t}w_{0t} = \sigma_0(=\pm 1) \qquad [3.44]$$

ile verilebilir.

Birçok problemde σ_0'ın değeri genelde bilinir. Ortogonalite koşulu homojen olduğundan σ_0'ın işareti değiştirildiğinde w_{mt} dual değişkenlerinin tümünün işareti değişir. Dolayısıyla σ_0'ın yanlış seçiminde bütün dual değişkenlerinin negatif değer almalarına neden olacaktır. Ancak mutlak değerce doğru olacaklardır. σ_0 ve dual değişkenler belirlendiğinde dual fonksiyon;

$$d(\underline{w}) = \sigma_0\left[\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left[\frac{c_{mt}w_{m0}}{w_{mt}}\right]^{\sigma_{mt}w_{mt}}\right]^{\sigma_0} \qquad [3.45]$$

şeklinde hesaplanabilir.

Passy ve Wilde (1969) bazı ek koşullar geliştirmişlerdir. Bunları da, kütle hareketi yasasını tanımlayan denklemlere benzerliklerinden dolayı denge denklemleri (koşulları) olarak tanımlamışlardır. Yukarıda oluşturulmuş normalite ve ortogonalite koşullarına ek olarak, zorluk derecesi sayısında doğrusal olmayan denge koşulları değişkenleri kadar denklemden oluşan bir sistem verir. Bunlar ;

$$\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left[\frac{w_{mt}}{w_{m0}}\right]^{\sigma_{mt}v_{dmt}} = \prod_{m=0}^{M}\prod_{t=1}^{T_k}c_{mt}^{\sigma_{mt}v_{dmt}} \qquad [3.46]$$

d = 1,2,...,D

$$D = \sum_{m=0}^{M} T_m - (n+1)$$

Burada V_d, d= 1,2,...,D (V_{dmt} bileşenli) $\sigma_0 = 0$ için normalite ve ortogonalite koşullarının doğrusal bağımsız çözümleridir.

4.GEOMETRİK PROGRAMLAMA İÇİN YENİ BİR YAKLAŞIM : HOLOTRANSFORMASYON

Holotransformasyon, Prof.Dr.Edip Büyükkoca tarafından geliştirilmiş ve önceki çalışmalarında Büyükkoca Algoritması ve Büyükkoca Dönüşümü adları ile adlandırılmış yeni bir dönüştürme yöntemidir. Bu yöntem, daha sonra Prof.Dr.Edip Büyükkoca (1974) tarafından **Holotransformasyon** adı ile yeniden adlandırılmıştır. Çünkü, lazer ışınları tarafından yapılan, üç boyutlu bir görüntü olan hologram gibi saklı değişkenler fikrine sahiptir.

Holotransformasyonda problem öklit uzayından öklit olmayan uzaya uygun bir dönüştürme matrisi ile yansıtılır. Problemin çözümü daima öklit olmayan uzayda klasik yöntemlerle yapılır. Geri yansıma (dönüştürme) orijinal problemin çözümü olan öklit uzayındaki çözümü verir. Yansıtılan uzay ne Riemann ne de Lobatchevky uzayıdır. Bir helis uzay temeline dayandırılmıştır (küresel ve semer biçiminde değildir). Holotransformasyon, matematiksel modellemede her çeşit problemin nümerik ve analitik yolla çözümünde uygulanabilir.

Holografi, bilinen fotoğrafik levhalar ile uygun ışık kullanıldığında özel bir optik eleman kullanmadan üç boyutlu görüntü üreten parazitik bir fotoğraftır. Hologramın hazırlanması sırasında uygun ışık, nesneyi (bilinmeyen "cisim" dalgası) ve fotoğrafik levhayı (bilinen "referans" dalgası) aydınlatmak için kanalize edilmiştir. Işının yarısı sahnedeki cisimden yayılmaktadır ve fotoğrafik levhada parazitik bir örnek oluşturmak için ışının öbür yarısı ile yeniden birleşir. Görüntü levha üzerine veya hologram ışığı üzerine doğrudan gönderilerek orijinale yakın bir şekilde üretilir. Negatif holografik fotoğraf daima şeffaftır ve iki boyutlu cisimler için açık ve renkli paralel çizgiler, üç boyutlu cisimler için karmaşık daireler vardır. Cismin şekli ile hologram görüntü arasında hiçbir benzerlik yoktur.

Bir hologramın çok küçük parçası hemen hemen bütün hologramın verdiği kadar iyi bir görüntü verir.

Holotransformasyonun anlamı, yansıtılmış uzayda her noktanın diğer noktalar hakkında bilgi taşımasıdır. Bu, yansıtılmış uzayın bütününü küçük farklarla tanımlayabilmesi anlamına gelir. Holotransformasyonun çok yönlü uygulamalarında yansıtılmış uzayın her bir noktası, diğer noktalar hakkında bilgi içermektedir. Holotransformasyon, fiziksel geometri kavramını kapsar ve kendi bağımsız yapısının özelliklerini doğurur.

4.1.Holotransformasyon Algoritmaları

Prof.Dr.Edip Büyükkoca (1990, 1994), doğrusal en küçük kareler probleminin çözümünde üç yeni yaklaşım geliştirmiştir.

4.1.1.Yapay Dönüşüm Yöntemi

Matematiksel modellemede daima, uydurulan modelde bazı artıklar veya açıklanmayan değişmeler vardır. Bazen en iyi modeli elde etmek için bu dönüşüme ihtiyaç vardır. Yapay dönüştürme, modelin tipini değiştirmeden gözlenen ve ölçülen bilgiler için daha iyi model çıkartabilme yeteneğine sahip olan holotransformasyonun çok yönlü uygulamaları ile yapılır.

Aşağıda verilen teorem ile normal denklemlerin çözümü üç adımda gerçekleştirilir :

1- Orijinal normal denklemleri yardımcı normal denklemlere dönüştürmek,
Bunun için orijinal doğrusal sistemin katsayılar matrisi A^T olarak devrikleştirilir ve $(A.A^T)$ matris çarpımı, daima simetrik olan doğrusal sistemin katsayılar matrisini tanımlar.

2- Yardımcı normal denklemleri bilinen yöntemleri kullanarak çözmek,
Tanımlanan $(A.A^T)X = b$ yardımcı doğrusal sistem herhangi bir klasik yöntemle çözülür. Orijinal doğrusal sistemdeki (b) vektörü, yardımcı doğrusal sistemdeki (b) vektörünün aynısıdır.

3- Yardımcı normal denklemlerin sonucunu kullanarak son çözümü elde etmek,
Yardımcı doğrusal sistemin çözümü X, X^T şekline dönüştürülür ve $X^T.A$ vektör matris çarpımı orijinal doğrusal sistemin önerilen (X) çözümünü belirtir.

4.1.2.Özdeş Normal Denklemler Yöntemi

Bu yöntem problemi iki adımda çözer.

1- Orijinal normal denklemleri, özdeş normal denklemlere dönüştürmek için orijinal denklemlerden biri çıkartılır. Çıkartılan bu denklem diğerlerinden daha fazla hatalı bilgiye sahiptir.

2- Özdeş normal denklemler, holotransformasyon yöntemi kullanılarak çözülür. Özdeş normal denklemler daima tanımlanmış doğrusal sistemlerdir.

4.1.3.Doğrudan Yaklaşım Yöntemi

Bu yöntem, lineer ve toplam türevleri kullanmadan doğrusal modeller için yeni hesaplama yolları üretir ve önerilen yeni dönüştürmenin tanımlanmayan doğrusal sistemlere bir uzantısıdır.

En küçük kareler (EKK) tahmininde, uygun bir matematiksel ifade seçmek gereklidir. Uygun ve basit matematiksel formülü bulmak çok güçtür. Eğer EKK probleminin gözlenen değişken değerleri veya deneysel olarak ölçülen değerleri yapısında istatiksel hatalar (rastgele veya belirsiz) varsa EKK tahminini kullanmak güvenli değildir. Çünkü EKK normu, normal dağılımlı (çan eğrisi şekilli) rastgele hatalar tahmini üzerine kurulmuştur. Gerçek hayattaki problemlerde hataların dağılımı üzerine çok fazla bilgi yoktur. Holotransformasyon normu, gözlenen ve ölçülen verilerle var olan sistem hakkında hatalı bilgilerden çok, hatasız bilgi olması esasına dayanır.

5.WILLIAMS-OTTO PROSESİNİN GEOMETRİK PROGRAMLAMA İLE OPTİMİZASYONU

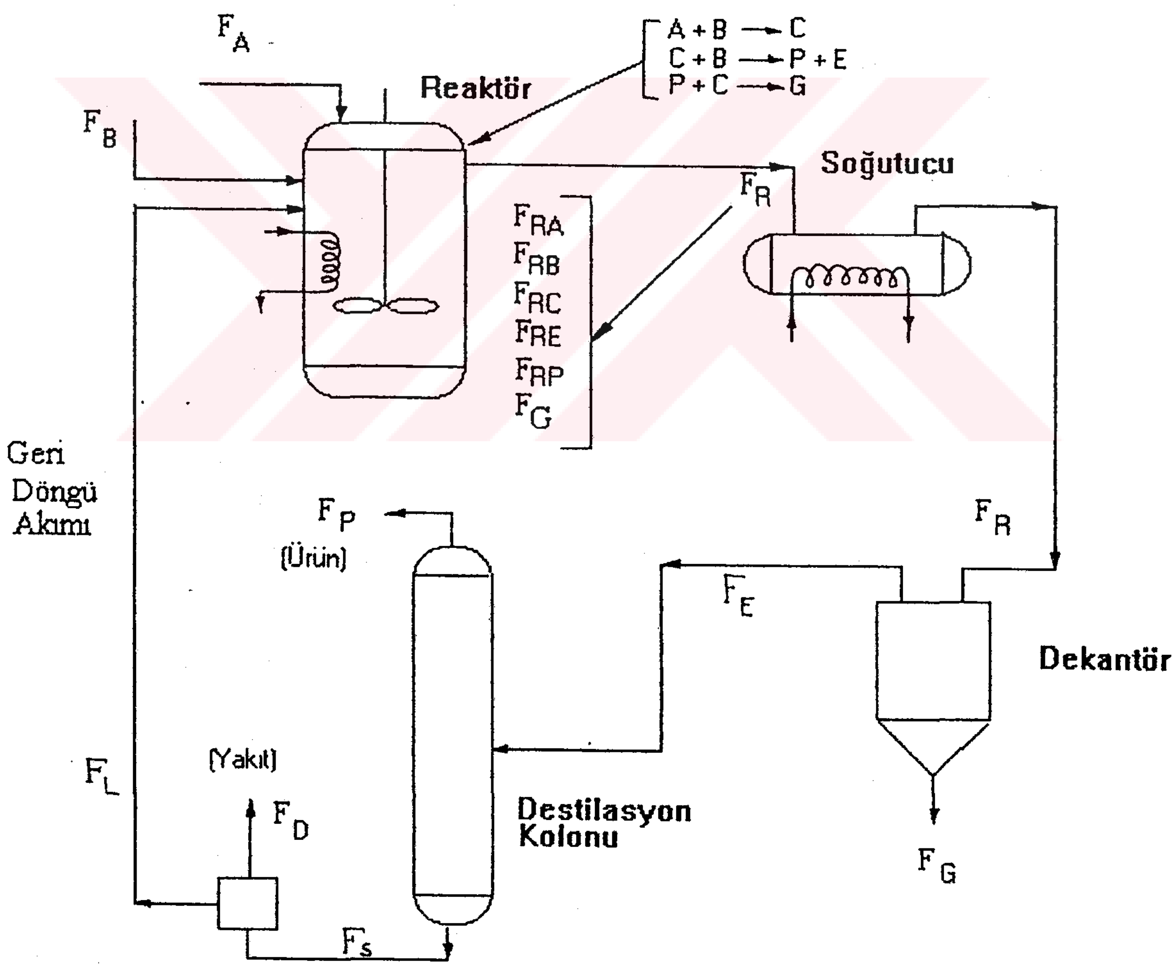
Williams-Otto prosesinin geometrik programlama ile optimizasyonundaki amaç, % dönüş = 100 (işletme karı/toplam yatırım)'ı maksimize etmek ve proses değişkenlerinin optimum değerlerini bulmaktır. Amaç fonksiyon ve kısıtlar negatif terimler içerdiğinden signomal programlama adımları izlenecektir.

5.1.Williams-Otto Prosesi

Karmaşık bir tasarım probleminin özelliklerinin çoğunu taşıyan ve tipik kimyasal üretim fabrikalarında olası etkilerin bir çoğunu (reaksiyon, ısı değişimi, toplam ayırma adımları ve geri döngü akımı) kapsayan Williams-Otto prosesinin blok diyagramı, sadeleştirilmiş olarak Şekil 5.1'de gösterilmektedir.

Atık ağır yağ maddesi G'nin fazla üretiminden sakınılmalı ve bir sonraki üründe azaltılmalıdır. Ek olarak sıcaklık belli bir değerin altına düştüğünde G çözünmez olur. Bu yüzden soğutucu gereklidir.

Atık ağır yağ maddesi G, 37.78°C veya aşağı akım sıcaklıklarında çözünmez olur ve taşıyıcı akımda daha yüksek bir yoğunluğa sahip olur. Dekantör yardımıyla uzaklaştırılabilir. Dekantördeki sıcaklık, G'nin çözünmez olduğu sıcaklıkta muhafaza edilmelidir. Kalan besleme maddesi ve ortak yüzey arasında, G, var olan hacme tamamen karışacaktır. Bu anda, reaktör çıkışındaki F_R akımında



Şekil 5.1.Williams-Otto prosesi

karışmış olacaktır ve kolon beslemesi G içererek F_L akımında reaktöre dönecektir. Böylece uygun sıcaklık kontrolü altında G, bütün akımlarda kabul edilebilir düzeyde mevcut olacaktır.

P ürünü, dekantörden çıkan F_E akımından destilasyonla ayrılmaktadır. Destilasyon kolonu, maksimum soğutma sıcaklığı 23.89°C için tasarım üretim hızında sınırlanan kolon yoğuşturucusu gözönüne alınarak tasarlanmaktadır. Soğutma suyu sıcaklığı günlük (gece ve gündüz) ve aylık olarak değişmektedir. Tesis kapasitesi de buna bağlı olarak değer almaktadır.

Gerekli soğutma sıcaklığı, yoğuşturucuda var olan buhar basıncının bir fonksiyonudur ve P ürünü kısmi basınca bağlı olarak yoğuşur.

P ürünü, ara ürün E üretimiyle yüksek sıcaklıkta kaynayan bir azeotrop oluşturur. Böylece P'nin bir kısmı kolon dip akımı F_S'de, bundan dolayı da atık akımı F_D'de ortaya çıkar. Atık akım F_D, kolon dip akımının fraksiyonu olarak ayrılmaktadır.

Asıl zorluk, tesisin işletme kararlılığını sınamak ve kararlı bir işletim sağlamak için bilgisayar kontrolünün olabilirliliğininin araştırılmasında yatmaktadır. Aynı zamanda, yatırımın maksimum dönüşünün elde edilmesini temin eden, tesis işletme koşullarının optimum ayarının bulunup, bilgisayarla sürdürülmesidir.

Tesisin kararlı halde yatırımın optimum dönüşüne çok yakın bir noktada çalıştığı bulunmuştur.

5.2.Kimyasal Reaksiyonun Ayrıntıları

P, aşağıdaki reaksiyonlarla üretilmektedir :

$A + B \rightarrow C$; reaksiyon katsayısı, k_1

$C+B \rightarrow P+E$; reaksiyon katsayısı, k_2

$P+C \rightarrow G$; reaksiyon katsayısı, k_3

Başlangıç reaktanları A ve B, tesisin bir önceki ünitelerinde saf olarak elde edilmektedir.

C ve E bileşenleri ara ve/veya yan ürünlerdir. Bunların kimyasal ürün olarak satış değeri yoktur. Ancak yakıt olarak hazırlanabilirler. G yan ürünü ise atık olan ağır yağlı bir maddedir.

Reaksiyon katsayıları k_1, k_2, k_3 Arrhenius eşitliği ile ifade edilmektedir.

$$k_i = A_i e^{-B_i/T} \quad i = 1,2,3$$

$A_1 = 5.9755.10^9/\text{saat}$

$B_1 = 6666.67K$

$A_2 = 2.5962.10^{12}/\text{saat}$

$B_2 = 8333.33K$

$A_3 = 9.6283.10^{15}/\text{saat}$

$B_3 = 11111.11K$

Reaksiyon katsayısında, k_i, hacim birimi m^3'den ziyade kg karışım cinsindendir (yada diğer hacim birimlerinde). Yoğunluk reaktörde sabit olarak düşünülmektedir ($\rho = 801.64\,kg/m^3$). Üç reaksiyonun reaksiyon ısıları (hepsi ekzotermik):

Reaksiyon 1 : H_1 = -68.83 kcal/kg C üretimi

Reaksiyon 2 : H_2 = -27.53 kcal/kg (E+P) üretimi

Reaksiyon 3 : H_3 = -78.74 kcal/kg G üretimi

5.3.Amaç Fonksiyonu

Esas amaç dönüşüm yüzdesini (amaç fonksiyonu) maksimize etmektir. Amaç fonksiyonu, madde dengesi eşitlikleri, akış hızındaki kısıtlar; ayırma verimi eşitlikleri ve reaksiyon sıcaklıklarındaki kısıtlarla sınırlandırılmıştır. Bunlarla ilgili ayrıntılar Luus ve Jaakola (1973), Adelman ve Stevens (1972) ve Williams ve Otto (1960) tarafından açıklanmaktadır.

Burada kullanılan amaç fonksiyonu, doğrudan Rijckaert ve Martens (1974) den alınan bir sonuçtur.

Amaç fonksiyonu,

$$\%\text{Dönüş} = \max\left[\frac{100[8400X - 2.2F_R - 0.12 * 8400(0.3F_P + 0.0068F_D - 60V\rho)]}{600V\rho}\right]$$

$$X = 0.3F_P + 0.0068F_D - 0.02F_A - 0.03F_B - 0.01F_G$$

X değeri asıl fonksiyonda yerine yazılırsa sonuç amaç fonksiyonumuz :

$$\%\text{Dönüş} = \max\left[\frac{84F_A - 201.96F_D - 336F_G + 1955.5F_P - 2.2F_R - 60V\rho}{6V\rho}\right] \quad [5.1]$$

olacaktır.

5.4.Proses Kısıt Eşitlikleri

Proses kısıtları, sistemdeki değişik birimlerdeki madde denklikleridir.

A için madde denkliği

$$F_A - k_1 F_{RA} F_{RB} \frac{V\rho}{F_R^2} - \frac{F_D F_{RA}}{F_R - F_P - F_G} = 0 \qquad [5.2]$$

B için madde denkliği

$$F_B - (k_1 F_{RA} F_{RB} + k_2 F_{RB} F_{RC}) \frac{V\rho}{F_R^2} - \frac{F_D F_{RB}}{F_R - F_P - F_G} = 0 \qquad [5.3]$$

C için madde denkliği

$$\left(\frac{M_C}{M_B} k_1 F_{RC} F_{RB} - \frac{M_E}{M_B} k_2 F_{RB} F_{RC} - k_3 F_{RP} F_{RC} \right) \frac{V\rho}{F_R^2} - \frac{F_D F_{RC}}{F_R - F_P - F_G} = 0 \qquad [5.4]$$

E için madde denkliği

$$\frac{M_E}{M_B} k_2 F_{RB} F_{RC} \frac{V\rho}{F_R^2} - \frac{F_D F_{RE}}{F_R - F_P - F_G} = 0 \qquad [5.5]$$

G için madde denkliği

$$-F_G + \frac{M_G}{M_C} k_3 F_{RC} F_{RP} \frac{V\rho}{F_R^2} = 0 \qquad [5.6]$$

P için madde denkliği

$$-F_P + \left(k_2 F_{RB} F_{RC} - \frac{M_P}{M_C} k_3 F_{RC} F_{RP} \right) \frac{V\rho}{F_R^2} - \frac{F_D (F_{RP} - F_P)}{F_R - F_P - F_G} = 0 \qquad [5.7]$$

Reaktördeki akış hızındaki kısıt,

$$F_{RA} + F_{RB} + F_{RC} + F_{RE} + F_G + F_{RP} - F_R = 0 \qquad [5.8]$$

Destilasyon kolonunda ayırma etkinliğindeki kısıt,

$$F_{RP} - 0.1F_{RE} - F_P = 0 \qquad [5.9]$$

Toplam madde denkliği,

$$F_A + F_B - F_G - F_P - F_D = 0 \qquad [5.10]$$

Sıcaklıklardaki kısıt,

$$322.22K(580R) \leq T \leq 377.78K(680R) \qquad [5.11]$$

Bu denkliklerde M_A, M_B ve M_P=100, M_C, M_E=200 ve M_G=300 olarak alınabilir. Toplam madde denkliği diğer denkliklerden bağımsızdır. F_P sabit olmak üzere, sekiz eşitlik kısıtı, iki eşitsizlik kısıtı ve 12 değişken vardır.

5.5.Amaç Fonksiyonunun ve Kısıtlarının Düzenlenmesi

Standart geometrik programlama düzeninde primal problemde sadece m=1,2,...,M olmak üzere $g_m(x) \leq \sigma_g$ eşitsizlik kısıtlarına yer verilmiştir. Bu yüzden eşitlik kısıtları eşitsizliklere dönüştürülmelidir.

Problemde yer alan eşitlik kısıtları için birinci adım, optimal durumdaki yapılarından hareketle zorlayıcı mı yoksa bağlayıcı mı olduklarına karar vermektir. Herhangi eşitlik kısıtı $\{\leq, \geq\}$'dan uygun eşitsizliğin seçimi ile optimallikte bağlayıcı olmaya zorlar. Dolayısıyla problemdeki her eşitlik için bir eşitsizlik bağıntısı kurulur ve elde edilen problem çözülür.

Eğer, optimallikte bir veya daha fazla kısıt zorunlu veya bağlayıcı olmaktan çıkarsa eşitlik kısıtı için yapılan tahmin değiştirilmeli ve yeniden çözüm yapılmalıdır. Kısıt için doğru tahmin, eldeki problemin fiziksel yapısı ile genellikle oluşturulabilir.

Dönüşüm için genel düzen,

1- Amaç fonksiyonunda $(GV\rho^{-1})$ ile bölme ve eksi işaretle çarpma,

2- Sekiz eşitlik kısıtını doğru bir şekilde eşitsizliklere dönüştürme,

3- $g_m(\phi) \leq \sigma_m$ yapısında herbir kısıtı yeniden düzenleme,

Problem üzerinde yapılacak bu dönüşümde ağırlık ikinci adımda olacaktır.

Eşitlik kısıt seti Blau ve Wilde (1969) tarafından açıklanan kurallar kullanılarak eşitsizliklere dönüştürülebilir. Eğer ana program kısıtlarda bir kez, modelde iki kez, amaç fonksiyonda bir kez olacak şekilde tamamen var olan karar değişkenleri cebirsel olarak yeniden düzenlenirse, bu dönüşüm kolaylıkla başarılabilir.[5.2] Eşitliğinden [5.10]'a kadar cebirsel beceriyle elde edilen kısıt seti aşağıdaki gibidir:

$$F_A = k_1 F_{RA} F_{RB} \beta + 0.2\beta k_2 F_{RA} F_{RB} F_{RC} b^{-1} \quad [5.12]$$

$$F_{RA} = k_2 k_1^{-1} F_{RC} + 0.5 k_3 k_1^{-1} F_{RC} F_{RP} F_{RP}^{-1} + 0.1 k_2 k_1^{-1} F_{RC}^2 b^{-1} \quad [5.13]$$

$$F_G = 1.5 k_3 F_{RC} F_{RP} \beta \quad [5.14]$$

$$F_P = 0.8\beta k_2 F_{RB} F_{RC} - 0.5 k_3 \beta F_{RC} F_{RP} \quad [5.15]$$

$$\alpha = k_2 k_1^{-1} F_{RC} + 0.5 k_3 k_1^{-1} F_{RC} F_{RP} F_{RB}^{-1} + 0.1 k_2 k_1^{-1} F_{RC} b^{-1} + F_{RB} + F_{RC} + 11b \quad [5.16]$$

$$b = F_{RP} - 0.8\beta k_2 F_{RB} F_{RC} + 0.5 k_3 \beta F_{RC} F_{RP} \qquad [5.17]$$

$$\beta = V \rho F_R^{-2} \qquad [5.18]$$

$$F_R = \alpha + 1.5 k_3 F_{RC} F_{RP} \beta + F_{RP} - b \qquad [5.19]$$

$$F_D = 0.2 \alpha \beta k_2 F_{RB} F_{RC} b^{-1} \qquad [5.20]$$

$$\alpha = F_{RA} + F_{RB} + F_{RC} + 11b \qquad [5.21]$$

$$b = F_{RP} - 0.2(4\beta k_2 F_{RB} F_{RC} - 2.5 k_3 \beta F_{RC} F_{RP} \qquad [5.22]$$

$$F_B = F_G + F_P + F_D - F_A \qquad [5.23]$$

$$F_{RE} = 10(F_{RP} - F_P) \qquad [5.24]$$

$$322.22K(580R) \leq T \leq 377.78K(680R) \qquad [5.25]$$

Eğer F_P sabitse bu bağıntılar 11 eşitlik kısıtına, iki eşitsizlik kısıtına ve 15 değişkene sahiptir. Blau ve Wilde'nin (1969) kuralları, eşitlikleri eşitsizliklere dönüştürmek için [5.12] eşitliğinden [5.24] eşitliğine kadar uygulanabilir.

[5.12], [5.14], [5.15] ve [5.20] eşitliklerinde sol taraf değişkenleri sadece amaç fonksiyonda görünmekte ve sağ taraf değişkenleri tarafından sınırlanmaktadır.

Eşitsizlik haline getirilmiş eşitlikler aşağıdaki gibidir :

$$F_A \leq k_1 F_{RA} F_{RB} \beta + 0.2\beta k_2 F_{RA} F_{RB} F_{RC} b^{-1} \quad [5.26]$$

$$F_{RA} \leq k_2 k_1^{-1} F_{RC} + 0.5 k_3 k_1^{-1} F_{RC} F_{RP} F_{RB}^{-1} + 0.1 k_2 k_1^{-1} F_{RC}^2 b^{-1} \quad [5.27]$$

$$F_G \geq 1.5 k_3 F_{RC} F_{RP} \beta \quad [5.28]$$

$$F_P \leq 0.8\beta k_2 F_{RB} F_{RC} - 0.5 k_3 \beta F_{RC} F_{RP} \quad [5.29]$$

$$\alpha \geq k_2 k_1^{-1} F_{RC} + 0.5 k_3 k_1^{-1} F_{RC} F_{RP} F_{RB}^{-1} + 0.1 k_2 k_1^{-1} F_{RC} b^{-1} + F_{RB} + F_{RC} + 11b \quad [5.30]$$

$$b \leq F_{RP} - 0.8\beta k_2 F_{RB} F_{RC} + 0.5 F_{RP} F_{RC} k_3 \beta \quad [5.31]$$

$$\beta \leq V \rho F_R^{-2} \quad [5.32]$$

$$F_R \geq \alpha + 1.5 k_3 F_{RC} F_{RP} \beta + F_{RP} - b \quad [5.33]$$

$$F_D \geq 0.2\alpha\beta k_2 F_{RB} F_{RC} b^{-1} \quad [5.34]$$

$$T \geq 322.22K \quad [5.35]$$

$$T \leq 377.78K \quad [5.36]$$

5.6.Geometrik Programlama ile Çözüm

Rijckaert ve Martens (1974), geliştirilmiş olan Williams-Otto prosesinin matematiksel modelinin var olan geometrik programlama düzenini geliştirmişlerdir. Rijckaert ve Martens aşağıda verilen ifadelerle k_2 ve k_3'ü k_1 değişkeni ile ifade ederek T değişkeninin konumunu değiştirmişlerdir.

$$k_2 = 1.56268k_1^{1.25} \tag{5.37}$$

$$k_3 = 0.48932k_1^{1.67} \tag{5.38}$$

[5.26]-[5.36] eşitsizliklerine [5.37] ve [5.38] eşitlikleri yazılır ve geleneksel geometrik programlama düzeni için [5.26]-[5.36] eşitsizlikleri yeniden düzenlenirse elde edilen sonuç ifadeler aşağıdaki gibidir :

$$\text{Min } g_0 = \min(-84F_A + 201.96F_D + 336F_G - 1955.52F_P + 2.22F_R)(6V\rho)^{-1} + 0.4\epsilon_1^{-0.01} + 0.4\epsilon_2^{-0.01} \tag{5.39}$$

$$g_1 = 0.73398k_1^{1.67}F_{RC}F_{RP}F_G^{-1}\beta \leq 1 \tag{5.40}$$

$$g_2 = -(-1.56268k_1^{0.25}F_{RC}F_{RA}^{-1} - 0.24466k_1^{0.67}F_{RC}F_{RB}^{-1}F_{RA}^{-1}F_{RP} - 0.15627k_1^{0.25}F_{RC}^2b^{-1}F_{RA}^{-1})^{-1} \leq 1 \tag{5.41}$$

$$g_3 = -(-1.25014k_1^{1.25}F_{RB}F_{RC}F_P^{-1}\beta + 0.24466k_1^{1.67}F_{RC}F_{RP}F_P^{-1}\beta)^{-1} \leq 1 \quad [5.42]$$

$$g_4 = 0.31254k_1^{1.25}F_{RB}F_{RC}F_D^{-1}b^{-1}\alpha\beta \leq 1 \quad [5.43]$$

$$g_5 = -(-k_1F_{RA}F_{RB}F_A^{-1}\beta - 0.31254k_1^{1.25}F_{RA}F_{RB}F_{RC}F_A^{-1}b^{-1}\beta)^{-1} \leq 1 \quad [5.44]$$

$$g_6 = \rho^{-1}\beta V^{-1}F_R^2 \leq 1 \quad [5.45]$$

$$g_7 = -(-F_{RP}b^{-1} + 1.2504k_1^{1.25}F_{RB}F_{RC}b^{-1}\beta - 0.24466k_1^{1.67}F_{RC}F_{RP}b^{-1}\beta)^{-1} \leq 1 \quad [5.46]$$

$$g_8 = \alpha F_R^{-1} + 0.73398k_1^{1.67}F_{RC}F_{RP}F_R^{-1}\beta + F_{RP}F_R^{-1} - bF_R^{-1} \leq 1 \quad [5.47]$$

$$g_9 = \alpha^{-1}F_{RC} + 0.24466k_1^{0.67}F_{RC}F_{RP}F_{RB}^{-1}\alpha^{-1} + 0.15627k_1^{0.25}F_{RC}^2b^{-1}\alpha^{-1} +$$
$$F_{RB}\alpha^{-1} + 11b\alpha^{-1} + 1.5628k_1^{0.25}F_{RC}\alpha^{-1} \leq 1 \quad [5.48]$$

$$g_{10} = 0.00772k_1 + \epsilon_1 \leq 1 \quad [5.49]$$

$$g_{11} = 6.17975k_1^{-1} + \epsilon_2 \leq 1 \quad [5.50]$$

$F_A, F_D; F_G, F_P, F_R, V, F_{RC}, F_{RP}, F_{RB}, k_1, \alpha, \beta > 0$

Bu optimizasyon probleminde 34 terim, 11 kısıt ve 15 değişken vardır. Zorluk derecesi 18'dir. Dual amaç fonksiyonda $\sigma = -1$ olarak seçilmiştir.
[3.41] eşitliğinden elde edilen normalite koşulu,
$-w_{01} + w_{02} + w_{03} - w_{04} + w_{05} + w_{06} + w_{07} = -1$

[3.38] eşitliğinden elde edilen ortogonalite koşulları,

$F_A : -w_{01} + w_{51} + w_{52} = 0$

$F_D : w_{02} - w_{41} = 0$

$F_G : w_{03} - w_{11} = 0$

$k_1 : 1.67w_{11} - 0.25w_{21} - 0.67w_{22} - 0.25w_{23} - 1.25w_{31} + 1.67w_{32} + 1.25w_{41} - w_{51} - 1.25w_{52} + 1.25w_{72} - 1.67w_{73} + 0.25w_{96} + 1.67w_{82} + 0.67w_{92} + 0.25w_{93} + w_{101} - w_{111} = 0$

$F_R : w_{05} + 2w_{61} - w_{81} - w_{82} - w_{83} + w_{84} = 0$

$V_\rho : w_{01} - w_{02} - w_{03} + w_{04} - w_{05} - w_{61} = 0$

$F_{RC} : w_{11} - w_{21} - w_{22} - 2w_{23} - w_{31} + w_{32} + w_{41} - w_{52} + w_{72} - w_{73} + w_{82} + w_{91} + w_{92} + 2w_{93} + w_{96} = 0$

$F_{RP} : w_{11} - w_{22} + w_{32} - w_{71} - w_{73} + w_{82} + w_{83} + w_{92} = 0$

$\beta : w_{11} - w_{31} + w_{32} + w_{41} - w_{51} - w_{52} + w_{61} + w_{72} - w_{73} + w_{82} = 0$

$F_{RA} : w_{21} + w_{22} + w_{23} - w_{51} - w_{52} = 0$

$F_{RB} : w_{22} - w_{31} + w_{41} - w_{51} - w_{52} + w_{72} - w_{92} + w_{94} = 0$

$b: w_{23} - w_{41} + w_{52} + w_{71} - w_{72} + w_{73} - w_{84} - w_{93} + w_{95} = 0$

$\alpha: w_{41} + w_{81} - w_{91} - w_{92} - w_{93} - w_{94} - w_{95} - w_{96} = 0$

$\epsilon_1: -0.01w_{06} + w_{102} = 0$

$\epsilon_2: -0.01w_{07} + w_{112} = 0$

[3.28] eşitliğinden elde edilen doğrusal eşitlikler,

$w_{11} - w_{10} = 0$

$w_{21} + w_{22} + w_{23} - w_{20} = 0$

$w_{31} - w_{32} - w_{30} = 0$

$w_{41} - w_{40} = 0$

$w_{51} + w_{52} - w_{50} = 0$

$w_{61} - w_{60} = 0$

$w_{71} - w_{72} + w_{73} - w_{70} = 0$

$w_{81} + w_{82} + w_{83} - w_{84} - w_{80} = 0$

$w_{91} + w_{92} + w_{93} + w_{94} + w_{95} + w_{96} - w_{90} = 0$

$w_{101} + w_{102} - w_{100} = 0$

$w_{111} + w_{112} - w_{110} = 0$

Dual problemimizde 45 bilinmeyen olmasına karşın 27 denklem vardır. Bu yüzden bazı ek koşulların getirilmesi gerekmektedir. Zorluk derecesi kadar denklem üreten bu koşullara Passy ve Wilde (1969) tarafından **denge koşulları** adı verilmiştir. Denge koşulları,

$$\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left(\frac{w_{mt}}{w_{m0}}\right)^{V_{mtd}\sigma_{mt}} = \prod_{m=0}^{M}\prod_{t=1}^{T_m} c_{mt}{}^{V_{mtd}\sigma_{mt}}$$

$$d = 1, ..., D$$

$$t = 1, ..., T_m$$

$$m = 0, ..., M$$

V_{mtd}: σ_0 için normalite ve ortogonalite koşullarının doğrusal bağımsız çözümleridir. Sıfırdan farklı V_{mtd} değerleri d=1,...,D için Tablo 5.1'de listelenmiştir.

Tablo 5.1.Sıfırdan farklı V_{mtd} değerleri

d	mt : v_{mtd}								
1	01:1	06:-1	21:1	51:1	61:-1	81:-2	91:4	94:-1	96:-5	102:-0.01
2	01:1	07:-1	21:1	51:1	61:-1	81:-2	91:4	94:-1	96:-5	112:-0.01
3	01:1	03:-1	11:-1	22:1	51:1					
4	21:-1	23:1	91:-1	95:1						
5	01:-1	04:1	21:-1	31:1	51:-1					
6	03:1	04:-1	11:1	32:-1						
7	01:1	02:-1	21:1	41:-1	51:1	95:-1				
8	51:-1	52:1	95:1	96:-1						
9	01:1	03:-1	11:-1	21:1	51:1	71:1	91:-1.67	94:-1	95:1	96:1.67
10	01:1	05:-1	21:1	51:1	72:-1	81:-1	95:-1			
11	03:-1	05:1	11:-1	73:1	81:1	95:1				
12	03:1	05:-1	11:1	82:-1						
13	01:-1	03:1	11:1	21:-1	51:-1	81:1	83:-1	91:1.67	94:1	96:-1.67
14	81:-1	84:1	95:-1							
15	01:-1	03:1	11:1	21:-1	51:-1	92:-1	96:1			
16	91:1	93:-1	95:-1	96:1						
17	91:-4	96:4	101:-1							
18	91:4	96:-4	111:-1							

Bu ek koşullarla elde edilen doğrusal olmayan denklem sistemi;

$w_{01}^{-1}.w_{06}^{-1}.w_{21}^{-1}.w_{51}^{-1}.w_{61}^{-1}.w_{81}^{-2}.w_{91}^{4}.w_{94}^{-1}.w_{96}^{-5}.w_{102}^{-0.01}.w_{20}.w_{50}.w_{60}.$
$w_{80}^{2}.w_{90}^{2}.w_{100}^{0.01} = 0.01226$

$w_{01}^{-1}.w_{07}^{-1}.w_{21}^{-1}.w_{51}^{-1}.w_{61}^{-1}.w_{81}^{-2}.w_{91}^{4}.w_{94}^{-1}.w_{96}^{-5}.w_{112}^{-0.01}.w_{20}.w_{50}.w_{60}.$
$w_{80}^{2}.w_{90}^{2}.w_{110}^{0.01} = 0.01226$

$w_{01}^{-1}.w_{03}^{-1}.w_{11}^{-1}.w_{22}^{-1}.w_{51}^{-1}.w_{10}.w_{20}.w_{50} = 0.07103$

$w_{21}.w_{23}^{-1}.w_{91}^{-1}.w_{95} = 10$

$w_{01}.w_{04}^{-1}.w_{21}.w_{31}^{-1}.w_{51}.w_{20}^{-1}.w_{30}.w_{50}^{-1} = 0.0537$

$w_{03}.w_{04}.w_{11}.w_{32}^{-1}.w_{10}^{-1}.w_{30} = 54754.56$

$w_{01}^{-1}.w_{02}^{-1}.w_{21}^{-1}.w_{41}^{-1}.w_{51}^{-1}.w_{95}^{-1}.w_{20}.w_{40}.w_{50}.w_{90} = 1.2345 * 10^{-4}$

$w_{51}.w_{52}^{-1}.w_{95}.w_{96}^{-1} = 22.521$

$w_{01}^{-1}.w_{03}^{-1}.w_{11}^{-1}.w_{21}^{-1}.w_{51}^{-1}.w_{71}^{-1}.w_{91}^{-1.67}.w_{94}^{-1}.w_{95}.w_{96}^{1.67}.w_{10}.w_{20}.w_{50}.w_{70} = 0.02578$

$w_{01}^{-1}.w_{05}^{-1}.w_{21}^{-1}.w_{51}^{-1}.w_{72}^{-1}.w_{81}^{-1}.w_{95}^{-1}.w_{20}.w_{50}.w_{70}.w_{80}.w_{90} = 8.983 * 10^{-3}$

$w_{03}^{-1}.w_{05}.w_{11}^{-1}.w_{73}^{-1}.w_{81}.w_{95}.w_{10}.w_{70}.w_{80}^{-1}.w_{90}^{-1} = 0.099$

$w_{03}.w_{05}^{-1}.w_{11}.w_{82}^{-1}.w_{10}^{-1}.w_{80} = 151.351$

$w_{01}.w_{03}.w_{11}.w_{21}.w_{51}.w_{81}.w_{83}^{-1}.w_{91}^{1.67}.w_{94}.w_{96}^{-1.67}.w_{10}^{-1}.w_{20}^{-1}.w_{50}^{-1}.w_{90}^{-1} = 426.63$

$w_{81}^{-1}.w_{84}^{-1}.w_{95}^{-1}.w_{80}^{2}.w_{90} = 0.0909$

$w_{01}.w_{03}.w_{11}.w_{21}.w_{51}.w_{92}^{-1}.w_{96}.w_{10}^{-1}.w_{20}^{-1}.w_{50}^{-1} = 5743.95$

$w_{91}.w_{93}^{-1}.w_{95}^{-1}.w_{96} = 0.909$

$w_{91}^{-4}.w_{96}^{4}.w_{101}^{-1}.w_{100} = 772.674$

$w_{91}^{4}.w_{96}^{-4}.w_{111}^{-1}.w_{110} = 0.02713$

Denge koşullarının logaritmasını alıp, normalite koşulu ve ortogonalite koşullarıyla beraber elde edilen doğrusal-logaritmik denklem sistemi 45 denklem ve 45 bilinmeyenden oluşur. Newton-Raphson yöntemi ile bu sistem iteratif olarak çözüldüğünde bulunan optimal ağırlıklar Tablo 5.2'de gösterilmektedir.

Hesaplanan W_{mt} değerlerini,

$$d(\underline{w}) = -1.0\left[\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left(\frac{c_{mt}w_{m0}}{w_{mt}}\right)^{w_{mt}\sigma_{mt}}\right]^{-1}$$

denkleminde yerine yazarsak elde edilen dual çözüm :

$$d(\underline{w}) = -121.54$$

Dual çözüm, primal çözüme eşit olacağından,

Maksimum % dönüş $= g_0 = d(\underline{w}) = 121.54$

Primal değişkenler aşağıdaki eşitlikler yardımıyla bulunabilir :

$$c_{0t}\prod_{n=1}^{N} x_n^{a_{0tn}} = g_0(\underline{x})w_{0t} \quad t = 1, 2, ..., t_0$$

Tablo 5.2.Optimumda w_{mt} değerleri

m	t							
	1	2	3	4	5	6	7	0
0	1.012	6.540	0.962	8.308	0.733	0.042	0.042	1.000
1	0.962							0.962
2	0.843	0.124	0.452					1.012
3	10.170	1.866						8.308
4	6.540							6.540
5	0.656	0.356						1.012
6	1.084							1.084
7	2.872	0.867	0.159					2.164
8	2.839	0.025	0.151	0.114				2.901
9	0.202	0.150	0.225	3.805	4.144	1.024		9.379
10	0.002	$0.4(10^{-3})$						$0.3(10^{-3})$
11	$0.2(10^{-4})$	$0.4(10^{-3})$						$0.4(10^{-3})$

$$c_{mt}\prod_{n=1}^{N} x_n^{a_{mtn}} = W_{mt} = \frac{w_{mt}}{w_{m0}}$$

$m = 1, 2, ..., M$

$t = 1, 2, ..., T_m$

c_{0t} amaç fonksiyonundaki terimlerin katsayıları, c_{mt} ise kısıtlardaki terimlerin katsayılarıdır. n = 1,2,...,N olmak üzere x_n primal değişkenlerdir. Bu değişkenlerin değerlerini bulmak için eşitliklerin logaritmaları alınırsa doğrusal olmayan denklem sistemi, doğrusal-logaritmik denklem sistemine indirgenir. 15 bilinmeyenli 34 denklem çözülürse elde edilen sonuç Tablo 5.3'te görüldüğü gibidir.

Tablo 5.3. Optimum primal çözüm

DEĞİŞKEN	kg/h	lb/saat
F_A	6125.90	13505.99
F_B	13.962.27	30782.41
F_D	16471.10	36312.35
F_G	1457.10	3212.35
F_P	2160.46	4763.00
F_R	168016.00	370411.86
F_{RA}	21499.91	47399.18
F_{RB}	66695.79	147039.03
F_{RC}	3534.97	7793.27
F_{RE}	66042.24	145598.20
F_{RP}	8764.69	19322.83
T	374.46 K	674.03 R
V	0.8707 m^3	30.75 ft^3
g_0(%)	121.54	121.54

6.WILLIAMS-OTTO PROSESİNİN HOLOTRANSFORMASYON İLE OPTİMİZASYONU

Williams-Otto prosesinin **Holotransformasyon** ile optimizasyonunda, geometrik programlama ile elde edilen normalite koşulu, ortogonalite koşulları ve denge koşullarından oluşan doğrusal-logaritmik denklem sistemi kullanılmıştır. Hesaplamalarda geometrik programlamada olduğu gibi Newton-Raphson yöntemi kullanılmıştır.

Dual çözüm vektörünü belirleyebilmek için bu sistemin, 4.bölümde anlatıldığı gibi (45,45) boyutlarında A katsayılar matrisi oluşturulur. Katsayılar matrisinin, devriği ile çarpımı $(A.A^T)$, çözüm vektörü değişmeksizin yardımcı doğrusal sistemi $(A.A^T)\underline{X} = b$ oluşturur. A katsayılar matrisi, her bir dual değişkene göre birinci türevlerin oluşturduğu Jakobiyen matrisi göstermektedir.

b çözüm vektörü aşağıdaki denklemlerden oluşmaktadır :

$$F_1 = -w_{01} + w_{02} + w_{03} - w_{04} + w_{05} + w_{06} - w_{07} + 1$$

$$F_2 = -w_{01} + w_{51} + w_{52}$$

$$F_3 = w_{02} - w_{41}$$

$$F_4 = w_{03} - w_{11}$$

$$F_5 = 1.67w_{11} - 0.25w_{21} - 0.67w_{22} - 0.25w_{23} - 1.25w_{31} + 1.67w_{32} + 1.25w_{41} + 1.25w_{72} - w_{51} - 1.25w_{52} - 1.67w_{73} + 1.67w_{82} + 0.25w_{96} + w_{101} - w_{111}$$

$$F_6 = w_{05} + 2w_{61} - w_{81} - w_{82} - w_{83} + w_{84}$$

$$F_7 = w_{01} - w_{02} - w_{03} + w_{04} - w_{05} - w_{61}$$

$$F_8 = w_{11} - w_{21} - w_{22} - 2w_{23} - w_{31} + w_{32} + w_{41} - w_{52} + w_{72} - w_{73} + w_{82} + w_{91} + w_{92} + 2w_{93} + w_{96}$$

$$F_9 = w_{11} - w_{22} + w_{32} - w_{71} - w_{73} + w_{82} + w_{83} + w_{92}$$

$$F_{10} = w_{11} - w_{31} + w_{32} + w_{41} - w_{51} - w_{52} + w_{61} + w_{72} - w_{73} + w_{82}$$

$$F_{11} = w_{21} + w_{22} + w_{23} - w_{51} - w_{52}$$

$$F_{12} = w_{22} - w_{31} + w_{41} - w_{51} - w_{52} + w_{72} - w_{92} + w_{94}$$

$$F_{13} = w_{23} - w_{41} + w_{52} + w_{71} - w_{72} + w_{73} - w_{84} - w_{93} + w_{95}$$

$$F_{14} = w_{41} + w_{81} - w_{91} - w_{92} - w_{93} - w_{94} - w_{95} - w_{96}$$

$$F_{15} = -0.01w_{06} + w_{102}$$

$$F_{16} = -0.01w_{07} + w_{112}$$

$$F_{17} = w_{11} - w_{10}$$

$$F_{18} = w_{21} + w_{22} + w_{23} - w_{20}$$

$$F_{19} = w_{31} - w_{32} - w_{30}$$

$$F_{20} = w_{41} - w_{40}$$

$$F_{21} = w_{51} + w_{52} - w_{50}$$

$$F_{22} = w_{61} - w_{60}$$

$$F_{23} = w_{71} - w_{72} + w_{73} - w_{70}$$

$$F_{24} = w_{81} + w_{82} + w_{83} - w_{84} - w_{80}$$

$$F_{25} = w_{91} + w_{92} + w_{93} + w_{94} + w_{95} + w_{96} - w_{90}$$

$$F_{26} = w_{101} + w_{102} - w_{100}$$

$$F_{27} = w_{111} + w_{112} - w_{110}$$

$$F_{28} = -logw_{01} - logw_{06} - logw_{21} - logw_{51} - logw_{61} - 2logw_{81} + 4logw_{91} - logw_{94} - 5logw_{96} - 0.01logw_{102} + logw_{20} + logw_{50} + logw_{60} + 2logw_{80} + 2logw_{90} + 0.01logw_{100} + 1.91$$

$$F_{29} = -logw_{01} - logw_{07} - logw_{21} - logw_{51} - logw_{61} - 2logw_{81} + 4logw_{91} - logw_{94} - 5logw_{96} - 0.01logw_{112} + logw_{20} + logw_{50} + logw_{60} + 2logw_{80} + 2logw_{90} + 0.01logw_{110} + 1.91$$

$$F_{30} = -logw_{01} - logw_{03} - logw_{11} - logw_{22} - logw_{51} + logw_{10} + logw_{20} + logw_{50} + 1.15$$

$$F_{31} = logw_{21} - logw_{23} - logw_{91} + logw_{95} - 1$$

$$F_{32} = logw_{01} - logw_{04} + logw_{21} - logw_{31} + logw_{51} - logw_{20} + logw_{30} - logw_{50} + 1.27$$

$$F_{33} = logw_{03} + logw_{04} + logw_{11} - logw_{32} - logw_{10} + logw_{30} - 4.74$$

$$F_{34} = -logw_{01} - logw_{02} - logw_{21} - logw_{41} - logw_{51} - logw_{95} + logw_{20} + logw_{40} + logw_{50} + logw_{90} + 3.91$$

$$F_{35} = logw_{51} - logw_{52} + logw_{95} - logw_{96} - 1.35$$

$$F_{36} = -logw_{01} - logw_{03} - logw_{11} - logw_{21} - logw_{51} - logw_{71} - 1.67logw_{91} - logw_{94} + 1.67logw_{96} + logw_{10} + logw_{20} + logw_{50} + logw_{70} + 1.58$$

$$F_{37} = -logw_{01} - logw_{05} - logw_{21} - logw_{51} - logw_{72} - logw_{81} - logw_{95} + logw_{20} + logw_{50} + logw_{70} + logw_{80} + logw_{90} + 2.04$$

$$F_{38} = -logw_{03} + logw_{05} - logw_{11} - logw_{73} + logw_{81} + logw_{95} + logw_{10} + logw_{70} - logw_{80} - logw_{90} + 1$$

$$F_{39} = logw_{03} - logw_{05} + logw_{11} - logw_{82} - logw_{10} + logw_{80} - 2.18$$

$$F_{40} = logw_{01} + logw_{03} + logw_{11} + logw_{21} + logw_{51} + logw_{81} - logw_{83} + 1.67logw_{91} + logw_{94} - 1.67logw_{96} - logw_{10} - logw_{20} - logw_{50} - logw_{90} - 2.63$$

$$F_{41} = -logw_{81} - logw_{84} - logw_{95} + 2logw_{80} + logw_{90} + 1.04$$

$$F_{42} = logw_{01} + logw_{03} + logw_{11} + logw_{21} + logw_{51} - logw_{92} + logw_{96} - logw_{10} - logw_{20} - logw_{50} - 3.76$$

$$F_{43} = logw_{91} - logw_{93} - logw_{95} + logw_{96} + 0.041$$

$$F_{44} = -4logw_{91} + 4logw_{96} - logw_{101} + logw_{100} - 2.89$$

$$F_{45} = 4logw_{91} - 4logw_{96} - logw_{111} + logw_{110} + 1.56$$

b çözüm vektörü, uygun başlangıç koşulları alınarak her iterasyon için bu denklemlerden belirlenir.

142 iterasyon sonunda optimum çözüme ulaşılmıştır. Bulunan optimum ağırlıklar Tablo 6.1'de verilmiştir.

Tablo 6.1. Optimumda w_{mt} ağırlıkları

m	t							
	1	2	3	4	5	6	7	0
0	0.9757	6.484	0.929	8.191	0.706	0.0226	0.0217	1.000
1	0.929							0.931
2	0.8115	0.1283	0.035					0.9817
3	9.956	1.764						8.1940
4	6.484							6.4880
5	0.642	0.334						0.9810
6	1.05							1.0500
7	2.723	0.838	0.155					2.0404
8	2.747	0.0221	0.149	0.112				2.8061
9	0.2117	0.1401	0.0313	3.625	4.205	1.01		9.2301
10	0.02327	$4.825(10^{-4})$						$2.327(10^{-2})$
11	$2.545(10^{-3})$	$1.138(10^{-3})$						$4.243(10^{-3})$

Hesaplanan w_{mt} değerlerini,

$$d(\underline{w}) = -1.0\left[\prod_{m=0}^{M}\prod_{t=1}^{T_m}\left(\frac{c_{mt}w_{m0}}{w_{mt}}\right)^{w_{mt}\sigma_{mt}}\right]^{-1}$$

denkleminde yerine yazarsak elde edilen dual çözüm :

$$d(\underline{w}) = -138.65$$

Dual çözüm, primal çözüme eşit olacağından,

Maksimum % dönüş $= g_0 = d(\underline{w}) = -138.65$

Primal değişkenleri bulmak için geometrik programlamada olduğu gibi aşağıdaki eşitlikler kullanılırsa,

$$c_{0t} \prod_{n=1}^{N} x_n^{a_{0tn}} = g_0(\underline{x}) w_{0t} \quad t = 1, 2, ..., T_0$$

$$c_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} = W_{mt} = \frac{w_{mt}}{w_{m0}}$$

$$m = 1, 2, ..., M$$

$$t = 1, 2, ..., T_m$$

Bu eşitliklerin sol· tarafı amaç fonksiyonu ve kısıtlardaki terimlere karşılık gelmektedir. c_{0t} ve c_{mt} sırasıyla amaç fonksiyondaki ve kısıtlardaki terimlerin katsayılarıdır.

Bu eşitliklerin logaritmaları alınıp doğrusallaştırılırsa, 34 denklem ve 16 bilinmeyenden oluşan logaritmik bir sistem ortaya çıkar.

$$logF_A - logV\rho = logw_{01} + logg_0(\underline{x}) - log14$$

$$logF_D - logV\rho = logw_{02} + logg_0(\underline{x}) - log33.66$$

$$logF_G - logV\rho = logw_{03} + logg_0(\underline{x}) - log56$$

$$logF_P - logV\rho = logw_{04} + logg_0(\underline{x}) - log325.92$$

$logF_R - logV\rho = logw_{05} + logg_0(\underline{x}) - log0.37$

$-0.01log\epsilon_1 = logw_{06} + logg_0(\underline{x}) - log0.4$

$-0.01log\epsilon_2 = logw_{07} + logg_0(\underline{x}) - log0.4$

$0.67logk_1 + logF_{RC} + logF_{RP} - logF_G + log\beta = logw_{11} - logw_{10} - log0.73398$

$0.25logk_1 + logF_{RC} - logF_{RA} = logw_{21} - logw_{20} - log1.56268$

$0.67logk_1 + logF_{RC} + logF_{RB} - logF_{RA} + logF_{RP} = logw_{22} - logw_{20} - log0.24466$

$0.25logk_1 + 2logF_{RC} - logb - logF_{RA} = logw_{23} - logw_{20} - log0.15627$

$1.25logk_1 + logF_{RB} + logF_{RC} - logF_P + log\beta = logw_{31} - logw_{30} - log1.25014$

$1.67logk_1 + logF_{RC} + logF_{RP} - logF_P + log\beta = logw_{32} - logw_{30} - log0.24466$

$1.25logk_1 + logF_{RB} + logF_{RC} - logF_D - logb + log\alpha + log\beta = logw_{41} - logw_{40} - log0.31254$

$logk_1 + logF_{RA} + logF_{RB} - logF_A + log\beta = logw_{51} - logw_{50}$

$1.25logk_1 + logF_{RA} + logF_{RB} + logF_{RC} - logF_A - logb - log\beta = logw_{52} - logw_{50} - log0.31254$

$-logV\rho + log\beta + 2logF_R = logw_{61} - logw_{60}$

$logF_{RP} - logb = logw_{71} - logw_{70}$

$1.25logk_1 + logF_{RB} + logF_{RC} - logb + log\beta = logw_{72} - logw_{70} - log1.25014$

$1.67logk_1 + logF_{RC} + logF_{RP} - logb + log\beta = logw_{73} - logw_{70} - log0.24466$

$$log\alpha - logF_R = logw_{81} - logw_{80}$$

$$1.67logk_1 + logF_{RC} + logF_{RP} - logF_R + log\beta = logw_{82} - logw_{80} - log0.73398$$

$$logF_{RP} - logF_R = logw_{83} - logw_{80}$$

$$logb - logF_R = logw_{84} - logw_{80}$$

$$-log\alpha + logF_{RC} = logw_{91} - logw_{90}$$

$$0.67logk_1 + logF_{RC} + logF_{RP} - logF_{RB} - log\alpha = logw_{92} - logw_{90} - log0.24466$$

$$0.25logk_1 + 2logF_{RC} - logb - log\alpha = logw_{93} - logw_{90} - log0.15627$$

$$logF_{RB} - log\alpha = logw_{94} - logw_{90}$$

$$logb - log\alpha = logw_{95} - logw_{90} - log11$$

$$0.25logk_1 + logF_{RC} - log\alpha = logw_{96} - logw_{90} - log1.5628$$

$$logk_1 = logw_{101} - logw_{100} - log0.00772$$

$$log\epsilon_1 = logw_{102} - logw_{100}$$

$$-logk_1 = logw_{111} - logw_{110} - log6.17975$$

$$log\epsilon_2 = logw_{112} - logw_{110}$$

Bu logaritmik denklem sisteminin çözümü Tablo 6.2'de görüldüğü gibidir.

Tablo 6.2. Optimum primal çözümler

DEĞİŞKEN	kg/h	Rijckaert ve Martens Çözümü (kg/h)
F_A	6009.544	6125.90
F_B	14136.541	13962.67
F_D	16559.530	16471.01
F_G	1426.092	1457.10
F_P	2160.46	2160.46
F_R	164029.872	168016.00
F_{RA}	21608.851	21499.91
F_{RB}	62928.82	66695.79
F_{RC}	3675.043	3534.97
F_{RE}	65465.052	66042.24
F_{RP}	8706.965	8764.69
T	375.68 K	374.46 K
V	0.7734 m^3	0.8707 m^3
g_0(%)	138.65	121.54

SONUÇ VE ÖNERİLER

Williams-Otto prosesinin geometrik programlama ile optimizasyonunda izlenen genel işlem, aritmetik-geometrik ortalama eşitsizliğinden yararlanarak problemin, çarpım fonksiyonları haline getirilmesiyle dual uzaya taşınmasıdır. Ağırlıklı ortalamaların oluşturduğu dual değişkenlerin değerleri Newton-Raphson yöntemi kullanılarak bulunmaktadır. Bu dual değişkenlerden yararlanılarak optimum primal değişkenler hesaplanmaktadır. Rijckaert ve Martens [1970] bu adımları izleyerek maksimize edilmek istenen amaç fonksiyonunun değerini 121.54 olarak bulmuştur.

Geometrik Programlama için yeni bir yaklaşım olarak önerilen **Holotransformasyon** yönteminin yapay dönüş algoritması kullanılmıştır. Bu algoritma dual değişkenlerin oluşturduğu denklem sisteminin jakobiyen matrisi üzerine uygulanmıştır. Bu uygulamada Jakobiyen matrisi (A) devriği ile (A^T) çarpılarak $(A.A^T)$X = b yardımcı denklem sistemi elde edilmiştir. Bu denklem sisteminin Gauss Eliminasyonu ile X çözümü elde edilmiş ve $(X^T.A)$ vektör matris çarpımı ile asıl çözümü bulunmuştur. Bu işlem her bir iterasyon için tekrarlanmıştır. 142. iterasyondan sonra elde edilen ΔX değerlerinin çok küçülmesi nedeni ile iterasyon kesilmiş ve sonuç olarak amaç fonksiyonu değeri 138.65 olarak bulunmuştur.

Bu sonuç tam olarak optimum olmamasına rağmen optimuma oldukça yakın bir noktadır ve genel olarak ortalama %2.54 hata getirmektedir. Ancak amaç fonksiyonunun yüksek değeri gözönüne alındığında bu hata kabul edilebilir sınırlar içindedir.

Oldukça fazla sayıda başlangıç noktaları denenmiş ve en uygun başlangıç değerlerinin elde edilen optimum değerlere yakın olduğu gözlenmiştir. Daha küçük başlangıç değerleri seçiminde, optimuma yaklaşılamadan ΔX değerleri küçülmektedir. Bu yüzden kademe kademe başlangıç değerleri belli oranlarda arttırılmıştır. Ancak uygun ve dengeli başlangıç değerlerinin seçimi son derece güçtür.

Karşılaştırma yapmak ve Rijckaert ve Martens'in elde ettikleri sonucu bulabilmek amacıyla bu başlangıç değerleri, Holotransformasyon yöntemi kullanılmadan Newton-Raphson yöntemine uygulanmıştır. Bu uygulamada jakobiyen matrisinin inversi hesabı, Geliştirilmiş Gauss-Jordan metodu ile yapılmış ve verilerin zayıf veya tekil olduğu mesajı alındığından çözüme ulaşılamamıştır.

Holotransformasyon ile bulunan dual değişkenlerden yararlanılarak primal değişkenlerin değerlerini hesaplayabilmek için Geometrik Programlama adımlarıyla elde edilmiş, 34 denklem ve 16 bilinmeyenden oluşan denklem sisteminin çözümü, Holotransformasyon bünyesinde Gauss Eliminasyon yöntemi ile mümkün olmamıştır. Köşegen elemanları kendileri ile ilgili işlemler sırasında sıfır değerini almıştır. Bu yüzden Geliştirilmiş Gauss Eliminasyonu kullanılmış ve verinin zayıf veya tekil olduğu sonucu ile karşılaşılmıştır. Gauss-Siedel yöntemi ile iteratif olarak çözülmeye çalışıldığında elde edilen çözümlerin primal denklemleri sağlamadığı görülmüştür. Bunun nedeni (34,16) boyutlarındaki denklem sisteminin logaritmik yapıda olmasından kaynaklanmaktadır. Bu denklem sisteminin Gauss-Siedel yöntemi ile çözümü logaritmik yapıyı sağlamasına rağmen anti-logaritmaları alındığında sağlamamaktadır. Bu geri dönüş sırasında ortaya çıkan durum, hem hataların birikimi hem de logaritmik yapıdaki çok küçük bir sapmanın, akış hızları gibi büyük değerler üzerinde büyük sapmalara yol açmasından kaynaklanmaktadır. Bu nedenlerden ötürü, yeterli denklem sayısı da var olduğundan primal çözümler elle bulunmuştur.

Doğrusal Olmayan Programlama dalı olarak Geometrik Programlama, doğrudan primali çözen diğer yöntemlere göre, toplam veya farklar içeren problemleri çarpım fonksiyonları haline getirdiğinden daha olumlu sonuçlar elde edilmesine rağmen; Williams-Otto prosesi gibi sadece 12 değişkenden ibaret bir prosesin optimizasyonunda, büyük çapta problemlerle karşılaşılması pratiklik ve çabukluk açısından olumsuzluk yaratmaktadır. Geometrik programlama ana başlığı altında birçok yöntem ortaya atılmıştır. Yeni geliştirilmiş yöntemler arasında, Mancini ve Wilde'nin [1978] *Combined Interval Newton Method*; Ecker ve Gochet'in [1978] *Modified Reduced Gradient Method* ve Barr, Sarın ve Bishara'nın [1989] *Gpall Algorithm Method* sayılabilir. Ayrıca, Dembo [1978], Rijckaert ve Martens [1978], Rajkopal ve Bricker [1990, 1992] tarafından birçok yeni gelişme ortaya atılmıştır.

Bütün bu yeni yöntemler ve gelişmeler Geometrik Programlama'nın gelişimini devam ettirdiğini ve ortaya çıkan problemler, sadece genel işlemin takip edilmesinin yeterli olmadığını göstermektedir. Ancak, az değişkenli problemler için diğer yöntemlere göre uygulanabilirliği basit ve sonuçlar daha olumludur. Diğer yöntemlere göre önemli bir üstünlüğü ise değişkenlerin optimal değerlerini bilmeden amaç fonksiyonu değerinin bulunabilmesidir.

Yeni yaklaşım olarak önerilen Holotransformasyon ile her iterasyonda köşegen elemanları sıfırdan farklı olması büyük bir sorunu gidermiş ve sisteme daha rahat hakim olunarak optimuma yakın bir nokta elde edilmiştir. Ortaya çıkan bu sapmada, model denklemlerin oluşturulmasından, model denklemlerinden yola çıkılarak elde edilen eşitsizliklerden veya problemin bir psedomaksimizasyon problemi olmasından kaynaklanabilir.

Sonuç olarak, Geometrik Programlama'da uygulanabileceği kanıtlanmış olan Holotransformasyon'un yeni gelişmeler üzerinde denenmesinin de çok daha olumlu sonuçlar vereceği düşünülmektedir.

KAYNAKLAR

1- Adelman,A., Stevens,W.F., 1972. Process optimization by the complex method, AIChe Journal, Vol 18, no 1: 20-24.

2- Blau,G.E., Wilde,D.J., 1969. Generalized polynomial programming, Canadian Journal of Chemistry Engineering: 47.

3- Büyükkoca,E., 1974. Kimya Sanayinde planlama prosesinin incelenmesi ve lineer programlamanın etkinliğini arttıran iki algoritma, Doktora Tezi, İstanbul Teknik Üniversitesi, Makina Fakültesi.

4- Büyükkoca,E., Adalı,S., Ikuho,Y., Hastaoğlu,M.A., 1990. Stratejik üretim ve savunma endüstrilerinde holotransformasyon uygulamalarıyla spesifik entegre modellerin kurulması-çözüm tekniklerinin araştırılması, Marmara Üniversitesi, Yayın no : 90-2.

5- Büyükkoca,E., 1994. Holotransformation and its Applications, Yıldız Teknik Üniversitesi, Kimya-Metalurji Fakültesi, Kimya Mühendisliği Bölümü, İstanbul.

6- Demiriz,A., 1992. Geometrik Programlama Teori ve Çözüm Teknikleri, YTÜ FBE, Yüksek Lisans Tezi, İstanbul.

7- Duffin,R.J., Peterson,E.L., Zener,C., 1967. Geometric Programming, John Wiley and Sons Inc., NewYork.

8- Luus,R., Jaakola,T.H.I., 1973. a direct approach to optimization of a complex system, AIChe Journal, 19 : 645.

9- Passy,U, Wilde,D.J., 1969. Mass action and polynomial optimization, Journal of Engineering Mathematics, 3 : 325-335.

10- Rijckaert,M.J., Martens,X.M., 1974. Analysis and optimization of the Williams-Otto process by geometric programming, AIChe Journal, Vol 20, no 4: 742-750.

11- Williams,T.J., Otto,R.E., 1960. A generalized chemical processing model for the investigation of computer control, AIEE Trnas., Part I, Communication and Electronics, 79 : 458-473.

Vol 26, no 2: 205-241.

20- Rijckaert,M.J., Martens,X.M., 1978. Comparison of generalized geometric programming algorithms, Journal of Optimization Theory and Applications,

21- Sposito,V.A., 1975. Linear and Nonlinear Programming, The Iowa State University Press.

22- Stoecker,W.F., 1986. Design of thermal systems, Mc Graw Hill.

23- Stiram,M., Steven,W.F., 1972. An example of the application of nonlinear programming to chemical process optimization, North Western University.

24- Tangwill,W.I., 1985. Nonlinear Programming, A varified Approach, Prentice Hall.

25- Tolunay,Y., 1980. Matematik Programlama ve İşletme Uygulamaları, İstanbul Üniversitesi Yayınları.

26- Williams,T.J., Otto,R.E., 1960. A generalized chemical processing model for the investigation of computer control, AIEE Trnas., Part I, Communication and Electronics, 79 : 458-473.

EK : KULLANILAN BİLGİSAYAR PROGRAMI

```
REM  BU BİLGİSAYAR PROGRAMINDA, NEWTON-RAPHSON YÖNTEMİ  İÇİNDE
REM  OLUŞTURULAN  JAKOBİEN MATRİSİ ÜZERİNE HOLOTRANSFORMASYON
REM  YÖNTEMİ UYGULANMAKTADIR. BU UYGULAMA SIRASINDA DENKLEM
REM  SİSTEMİNİN ÇÖZÜMÜNDE  GAUSS ELİMİNASYONU, PRIMAL ÇÖZÜM
REM  SIRASINDA İSE, GAUSS SEIDEL  YÖNTEMİ KULLANILMAKTADIR.
REM
REM                        YILDIZ TEKNİK ÜNİVERSİTESİ
REM               F.B.E.  KİMYA MÜHENDİSLİĞİ  ANA BİLİM DALI
REM
REM                        KİMYA MÜHENDİSİ  YAVUZ  SALT.
REM                                 1995
REM  BU PROGRAM  Q-BASIC 4.5 SÜRÜMÜ İLE YAZILMIŞ VE 486-DX,4MB RAM
REM  ÖZELLİKLERİNE SAHİP BİR BİLGİSAYAR KULLANILMIŞTIR.
CLS
CLEAR
DIM A(45, 46), YS#(45, 46), P#(45, 45), AT#(45, 45)
DIM ATX#(45, 45), SC#(45, 45), F(45)
DIM GC#(46, 46), B(35, 35), C(35), X(35), CG#(1, 45)

FOR I = 1 TO 27
FOR J = 1 TO 45
READ A(I, J)
NEXT J, I


FOR I = 1 TO 35
FOR J = 1 TO 16
READ B(I, J)
NEXT J, I

W01 = .7084: W02 = 4.578: W03 = .6734: W04 = 5.8156: W05 = .5131: G=.0294
W07 = .0294
W11 = .6734: W21 = .5901: W22 = .0868: W23 = .03164: W31 = 7.119: W32 = 1.3062
W41 = 4.578: W51 = .4592: W52 = .2492: W61 = .7588: W71 = 2.0104: W72 = .6069
W73 = .1113: W81 = 1.9873: W82 = .0175: W83 = .1057: W84 = .0798: W91 = .1414
W92 = .105: W93 = .0385: W94 = 2.6635: W95 = 2.9008: W96 = .7168
W101 = .0014: W102 = .00028: W111 = .000014: W112 = .00028
W10 = .6734: W20 = .7084: W30 = 5.8156: W40 = 4.578: W50 = .7084
W60 = .7588: W70 = 1.5148: W80 = 2.0307: W90 = 6.5653: W100 = .00021
W110 = .00028
T = 1
50   FOR I = 28 TO 45
FOR J = 1 TO 45
A(I, J) = 0
NEXT J, I

F(1) = -W01 + W02 + W03 - W04 + W05 + W06 + W07 + 1

F(2) = -W01 + W51 + W52
```

F(3) = W02 - W41

F(4) = W03 - W11

F51 = (1.67 * W11) - (.25 * W21) - (.67 * W22) - (.25 * W23)
F52 = (-1.25 * W31) + (1.67 * W32) + (1.25 * W41) + (-1 * W51)
F53 = (-1.25 * W31) + (1.67 * W32) + (1.25 * W41) + (-1 * W51)
F54 = (1.25 * W52) + (1.25 * W72) - (1.67 * W73) + (1.67 * W82)
F55 = (.67 * W92) + (.25 * W93) + (.25 * W96) + (W101 - W111)

F(5) = F51 + F52 + F53 - F54 + F55

F(6) = W05 + (2 * W61) - W81 - W82 - W83 + W84

F(7) = W01 - W02 - W03 + W04 - W05 - W61

F81 = W11 - W21 - W22 - (2 * W23) - W31 + W32 + W41 - W52 + W72
F82 = -W73 + W82 + W91 + W92 + (2 * W93) + W96

F(8) = F81 + F82

F(9) = W11 - W22 + W32 - W71 - W73 + W82 + W83 + W92

F(10) = W11 - W31 + W32 + W41 - W51 - W52 + W61 + W72 - W73 + W82

F(11) = W21 + W22 + W23 - W51 - W52

F(12) = W22 - W31 + W41 - W51 - W52 + W72 - W92 + W94

F(13) = W23 - W41 + W52 + W71 - W72 + W73 - W84 - W93 + W95

F(14) = W41 + W81 - W91 - W92 - W93 - W94 - W95 - W96

F(15) = (-.01 * W06) + W102

F(16) = (-.01 * W07) + W112

F(17) = W11 - W10

F(18) = W21 + W22 + W23 - W20

F(19) = W31 - W32 - W30

F(20) = W41 - W40

F(21) = W51 + W52 - W50

F(22) = W61 - W60

F(23) = W71 - W72 + W73 - W70

F(24) = W81 + W82 + W83 - W84 - W80

F(25) = W91 + W92 + W93 + W94 + W95 + W96 - W90

F(26) = W101 + W102 - W100

F(27) = W111 + W112 - W110

F281 = (-LOG(W01)) - LOG(W06) - LOG(W21) - LOG(W51) - LOG(W61)
F282 = (2 * LOG(W81)) + (4 * LOG(W91)) - (LOG(W94))
F283 = (5 * LOG(W96)) + (-.01 * LOG(W102)) + LOG(W20) + LOG(W50)
F284 = LOG(W60) + (2 * LOG(W80)) + (2 * LOG(W90)) + (.01 * LOG(W100))
F(28) = ((F281 - F282 - F283 + F284) / LOG(10)) - .2

F291 = (-LOG(W01)) - LOG(W07) - LOG(W21) - LOG(W51) - LOG(W61)
F292 = (2 * LOG(W81)) + (4 * LOG(W91)) - (LOG(W94))
F293 = (5 * LOG(W96)) + (-.01 * LOG(W112)) + LOG(W20) + LOG(W50)
F294 = LOG(W60) + (2 * LOG(W80)) + (2 * LOG(W90)) + (.01 * LOG(W110))
F(29) = ((F291 - F292 - F293 + F294) / LOG(10)) - .2

F301 = (-LOG(W01)) - LOG(W03) - LOG(W11) - LOG(W22) - LOG(W51)
F(30) = ((F301 + LOG(W10) + LOG(W20) + LOG(W50)) / LOG(10)) - 1.1

F(31) = ((LOG(W21) - LOG(W23) - LOG(W91) + LOG(W95)) / LOG(10)) - 2.6

F321 = LOG(W01) - LOG(W04) + LOG(W21) - LOG(W31) + LOG(W51)
F(32) = ((F321 - LOG(W20) + LOG(W30) - LOG(W50)) / LOG(10)) + 1.27

F331 = LOG(W03) + LOG(W04) + LOG(W11) - LOG(W32) - LOG(W10)
F(33) = ((F331 + LOG(W30)) / LOG(10)) - 1.55

F341 = (-LOG(W01)) - LOG(W02) - LOG(W21) - LOG(W41) - LOG(W51)
F342 = LOG(W95) + LOG(W20) + LOG(W40) + LOG(W50)
F(34) = ((F341 - F342 + LOG(W90)) / LOG(10)) + .2

F(35) = ((LOG(W51) - LOG(W52) + LOG(W95) - LOG(W96)) / LOG(10)) - .9

F361 = (-LOG(W01)) - LOG(W03) - LOG(W11) - LOG(W21) - LOG(W51)
F362 = LOG(W71) - (1.67 * LOG(W91)) + (-1 * LOG(W94))
F363 = LOG(W95) + (1.67 * LOG(W96)) + LOG(W10)
F(36) = ((F361 - F362 + F363) / LOG(10)) - 1.37

F371 = (-LOG(W01)) - LOG(W05) - LOG(W21) - LOG(W51) - LOG(W72)
F372 = LOG(W81) - LOG(W95) + LOG(W20) + LOG(W50) + LOG(W70)
F(37) = ((F371 - F372 + LOG(W80) + LOG(W90)) / LOG(10)) - 1.16

F381 = (-LOG(W03)) + LOG(W05) - LOG(W11) - LOG(W73) + LOG(W81)
F382 = LOG(W95) + LOG(W10) + LOG(W70) - LOG(W80)
F(38) = ((F381 + F382 - LOG(W90)) / LOG(10)) - .65

F391 = LOG(W03) - LOG(W05) + LOG(W11) - LOG(W82) - LOG(W10)
F(39) = ((F391 + LOG(W80)) / LOG(10)) - 2.2

F401 = LOG(W01) + LOG(W03) + LOG(W11) + LOG(W21) + LOG(W51) + LOG(W81)
F402 = LOG(W83) + (1.67 * LOG(W91)) + LOG(W94) - (1.67 * LOG(W96))
F403 = LOG(W10) - LOG(W20) - LOG(W50)

```
F(40) = ((F401 - F402 - F403 - LOG(W90)) / LOG(10)) + .57

F411 = LOG(W10) - LOG(W20) - LOG(W50)
F(41) = ((F411 + (2 * LOG(W80))) / LOG(10)) - 1.77

F421 = LOG(W01) + LOG(W03) + LOG(W11) + LOG(W21) + LOG(W51)
F422 = (LOG(W92)) + LOG(W96) - LOG(W10) - LOG(W20)
F(42) = ((F421 - F422 + (-1 * LOG(W50))) / LOG(10)) - .55

F(43) = ((LOG(W91) - LOG(W93) - LOG(W95) + LOG(W96)) / LOG(10)) + .042

F441 = (-4 * LOG(W91)) + (4 * LOG(W96)) - LOG(W101)
F(44) = ((F441 + LOG(W100)) / LOG(10)) - 2

F451 = (4 * LOG(W91)) - (4 * LOG(W96)) - LOG(W111)
F(45) = ((F451 + LOG(W110)) / LOG(10)) + 1.52


CLS
PRINT "FONKSIYON DEGERLERI"
PRINT "------------------------------------------"
FOR I = 1 TO 15
PRINT "F"; I; "= "; F(I)
NEXT I

FOR I = 16 TO 30
LOCATE I - 13, 24: PRINT "F"; I; "= "; F(I)
NEXT I
FOR I = 31 TO 45
LOCATE I - 28, 46: PRINT "F"; I; "= "; F(I)
NEXT I

PRINT T; ".ITERASYON"

FOR I = 1 TO 45
A(I, 46) = -F(I)
NEXT I

A(28, 1) = -1 / W01: A(28, 6) = -1 / W06: A(28, 9) = -1 / W21
A(28, 15) = -1 / W51: A(28, 17) = -1 / W61: A(28, 21) = -2 / W81
A(28, 25) = 4 / W91: A(28, 28) = -1 / W94: A(28, 30) = -5 / W96
A(28, 32) = -.01 / W102: A(28, 36) = 1 / W20: A(28, 39) = 1 / W50
A(28, 40) = 1 / W60: A(28, 42) = 2 / W80: A(28, 43) = 2 / W90
A(28, 44) = .01 / W100

A(29, 1) = -1 / W01: A(29, 7) = -1 / W07: A(29, 9) = -1 / W21
A(29, 15) = -1 / W51: A(29, 17) = -1 / W61: A(29, 21) = -2 / W81
A(29, 25) = 4 / W91: A(29, 28) = -1 / W94: A(29, 30) = -5 / W96
A(29, 34) = -.01 / W112: A(29, 36) = 1 / W20: A(29, 39) = 1 / W50
A(29, 40) = 1 / W60: A(29, 42) = 2 / W80: A(29, 43) = 2 / W90
A(29, 45) = .01 / W110

A(30, 1) = -1 / W01: A(30, 3) = -1 / W03: A(30, 8) = -1 / W11
A(30, 10) = -1 / W22: A(30, 15) = -1 / W51: A(30, 35) = 1 / W10
```

A(30, 36) = 1 / W20: A(30, 39) = 1 / W50

A(31, 9) = 1 / W21: A(31, 11) = -1 / W23: A(31, 25) = -1 / W91
A(31, 29) = 1 / W95

A(32, 1) = 1 / W01: A(32, 4) = -1 / W04: A(32, 9) = 1 / W21
A(32, 12) = -1 / W31: A(32, 15) = 1 / W51: A(32, 36) = -1 / W20
A(32, 37) = 1 / W30: A(32, 39) = -1 / W50

A(33, 3) = 1 / W03: A(33, 4) = 1 / W04: A(33, 8) = 1 / W11
A(33, 13) = -1 / W32: A(33, 35) = -1 / W10: A(33, 37) = 1 / W30

A(34, 1) = -1 / W01: A(34, 2) = -1 / W02: A(34, 9) = -1 / W21
A(34, 14) = -1 / W41: A(34, 15) = -1 / W51: A(34, 29) = -1 / W95
A(34, 36) = 1 / W20: A(34, 38) = 1 / W40: A(34, 39) = 1 / W50
A(34, 43) = 1 / W90

A(35, 15) = 1 / W51: A(35, 16) = -1 / W52: A(35, 29) = 1 / W95
A(35, 30) = -1 / W96

A(36, 1) = -1 / W01: A(36, 3) = -1 / W03: A(36, 8) = -1 / W11
A(36, 9) = -1 / W21: A(36, 15) = -1 / W51: A(36, 18) = -1 / W71
A(36, 25) = -1.67 / W91: A(36, 28) = -1 / W94: A(36, 29) = 1 / W95
A(36, 30) = 1.67 / W96: A(36, 35) = 1 / W10: A(36, 36) = 1 / W20
A(36, 39) = 1 / W50: A(36, 41) = 1 / W70

A(37, 1) = -1 / W01: A(37, 5) = -1 / W05: A(37, 9) = -1 / W21
A(37, 15) = -1 / W51: A(37, 19) = -1 / W72: A(37, 21) = -1 / W81
A(37, 29) = -1 / W95: A(37, 36) = 1 / W20: A(37, 39) = 1 / W50
A(37, 41) = 1 / W70: A(37, 42) = 1 / W80: A(37, 43) = 1 / W90

A(38, 3) = -1 / W03: A(38, 5) = 1 / W05: A(38, 8) = -1 / W11
A(38, 20) = -1 / W73: A(38, 21) = 1 / W81: A(38, 29) = 1 / W95
A(38, 35) = 1 / W10: A(38, 41) = 1 / W70: A(38, 42) = -1 / W80
A(38, 43) = -1 / W90

A(39, 3) = 1 / W03: A(39, 5) = -1 / W05: A(39, 8) = 1 / W11
A(39, 22) = -1 / W82: A(39, 35) = -1 / W10: A(39, 42) = 1 / W80

A(40, 1) = 1 / W01: A(40, 3) = 1 / W03: A(40, 8) = 1 / W11
A(40, 9) = 1 / W21: A(40, 15) = 1 / W51: A(40, 21) = 1 / W81
A(40, 23) = -1 / W83: A(40, 25) = 1.67 / W91: A(40, 28) = 1 / W94
A(40, 30) = -1.67 / W96: A(40, 35) = -1 / W10: A(40, 36) = -1 / W20
A(40, 39) = -1 / W50: A(40, 43) = -1 / W90

A(41, 21) = -1 / W81: A(41, 24) = -1 / W84: A(41, 29) = -1 / W95
A(41, 43) = 1 / W90: A(41, 42) = 2 / W80

A(42, 1) = 1 / W01: A(42, 3) = 1 / W03: A(42, 8) = 1 / W11
A(42, 9) = 1 / W21: A(42, 15) = 1 / W51: A(42, 26) = -1 / W92
A(42, 30) = 1 / W96: A(42, 35) = -1 / W10: A(42, 36) = -1 / W20
A(42, 39) = -1 / W50

A(43, 25) = 1 / W91: A(43, 27) = -1 / W93: A(43, 29) = -1 / W95

```
A(43, 30) = 1 / W96

A(44, 25) = -4 / W91: A(44, 30) = 4 / W96: A(44, 31) = -1 / W101
A(44, 44) = 1 / W100

A(45, 25) = 4 / W91: A(45, 30) = -4 / W96: A(45, 33) = -1 / W111
A(45, 45) = 1 / W110

FOR I = 28 TO 45
FOR J = 1 TO 45
A(I, J) = .4343 * A(I, J)
NEXT J, I

FOR I = 1 TO 45
FOR J = 1 TO 45
P#(I, J) = A(I, J)
NEXT J, I

FOR I = 1 TO 45
FOR J = 1 TO 45
AT#(I, J) = A(J, I)
NEXT J, I
N = 45
M = 45
L = 45
GOSUB 100

REM ******SANAL COZUM *******
FOR I = 1 TO 45
FOR J = 1 TO 45
YS#(I, J) = ATX#(I, J)
NEXT J, I

FOR I = 1 TO 45
YS#(I, 46) = A(I, 46)
NEXT I

GOSUB 250

CLS
PRINT "SANAL COZUMLER......................................"

FOR I = 1 TO 23
SC#(1, I) = YS#(I, 46) / YS#(I, I)
PRINT "SC("; 1; ","; I; ")= "; SC#(1, I)
NEXT I

FOR I = 24 TO 45
SC#(1, I) = YS#(I, 46) / YS#(I, I)
LOCATE I - 22, 35: PRINT "SC("; 1; ","; I; ")= "; SC#(1, I)
NEXT I

FOR I = 1 TO 45
FOR J = 1 TO 45
```

```
    PR1# = PR1# + SC#(1, J) * A(J, I)
    NEXT J
    GC#(1, I) = PR1#
    PR1# = 0
    NEXT I

22  CLS : PRINT "DX COZUMLERI..................................."
    FOR I = 1 TO 23
    LOCATE I + 1, 1: PRINT GC#(1, I)
    NEXT I
    FOR I = 24 TO 45
    LOCATE I - 22, 35: PRINT GC#(1, I)
    NEXT I

    W01 = ABS(W01 + GC#(1, 1)): W02 = ABS(W02 + GC#(1, 2))
    W03 = ABS(W03 + GC#(1, 3)): W04 = ABS(W04 + GC#(1, 4))
    W05 = ABS(W05 + GC#(1, 5)): W06 = ABS(W06 + GC#(1, 6))
    W07 = ABS(W07 + GC#(1, 7)): W11 = ABS(W11 + GC#(1, 8))
    W21 = ABS(W21 + GC#(1, 9)): W22 = ABS(W22 + GC#(1, 10))
    W23 = ABS(W23 + GC#(1, 11)): W31 = ABS(W31 + GC#(1, 12))
    W32 = ABS(W32 + GC#(1, 13)): W41 = ABS(W41 + GC#(1, 14))
    W51 = ABS(W51 + GC#(1, 15)): W52 = ABS(W52 + GC#(1, 16))
    W61 = ABS(W61 + GC#(1, 17)): W71 = ABS(W71 + GC#(1, 18))
    W72 = ABS(W72 + GC#(1, 19)): W73 = ABS(W73 + GC#(1, 20))
    W81 = ABS(W81 + GC#(1, 21)): W82 = ABS(W82 + GC#(1, 22))
    W83 = ABS(W83 + GC#(1, 23)): W84 = ABS(W84 + GC#(1, 24))
    W91 = ABS(W91 + GC#(1, 25)): W92 = ABS(W92 + GC#(1, 26))
    W93 = ABS(W93 + GC#(1, 27)): W94 = ABS(W94 + GC#(1, 28))
    W95 = ABS(W95 + GC#(1, 29)): W96 = ABS(W96 + GC#(1, 30))
    W101 = ABS(W100 + GC#(1, 31)): W102 = ABS(W102 + GC#(1, 32))
    W111 = ABS(W111 + GC#(1, 33)): W112 = ABS(W112 + GC#(1, 34))
    W10 = ABS(W10 + GC#(1, 35)): W20 = ABS(W20 + GC#(1, 36))
    W30 = ABS(W30 + GC#(1, 37)): W40 = ABS(W40 + GC#(1, 38))
    W50 = ABS(W50 + GC#(1, 39)): W60 = ABS(W60 + GC#(1, 40))
    W70 = ABS(W70 + GC#(1, 41)): W80 = ABS(W80 + GC#(1, 42))
    W90 = ABS(W90 + GC#(1, 43)): W100 = ABS(W100 + GC#(1, 44))
    W110 = ABS(W110 + GC#(1, 45))
    T = T + 1
    GOSUB 40
    INPUT "ITERASYONA DEVAM ETMEK ISTIYORMUSUNUZ"; A$
    IF A$ = "H" THEN GOTO 21
    IF A$ = "E" THEN GOTO 50
    GOTO 50
21  C(1) = LOG((W01 * FMIN) / 14)
    C(2) = LOG((W02 * FMIN) / 33.66)
    C(3) = LOG((W03 * FMIN) / 56)
    C(4) = LOG((W04 * FMIN) / 325.92)
    C(5) = LOG((W05 * FMIN) / .37)
    C(6) = LOG((W06 * FMIN) / .4)
    C(7) = LOG((W07 * FMIN) / .4)
    C(8) = LOG(W11 / (W10 * .73398))
    C(9) = LOG(W21 / (W20 * 1.56268))
    C(10) = LOG(W22 / (W20 * .24466))
    C(11) = LOG(W23 / (W20 * .15627))
```

```
C(12) = LOG(W31 / (W30 * 1.25014))
C(13) = LOG(W32 / (W30 * .24466))
C(14) = LOG(W41 / (W40 * .31254))
C(15) = LOG(W51 / W50)
C(16) = LOG(W52 / (W50 * .31254))
C(17) = LOG(W61 / W60)
C(18) = LOG(W71 / W70)
C(19) = LOG(W72 / (W70 * 1.25014))
C(20) = LOG(W73 / (W70 * .24466))
C(21) = LOG(W81 / W80)
C(22) = LOG(W82 / (W80 * .73398))
C(23) = LOG(W83 / W80)
C(24) = LOG(W84 / W80)
C(25) = LOG(W91 / W90)
C(26) = LOG(W92 / (W90 * .24466))
C(27) = LOG(W93 / (W90 * .15627))
C(28) = LOG(W94 / W90)
C(29) = LOG(W95 / (W90 * 11))
C(30) = LOG(W96 / (W90 * 1.56268))
C(33) = LOG(W101 / (W100 * .00772))
C(32) = LOG(W102 / W100)
C(31) = LOG(W111 / (W110 * 6.17975))
C(34) = LOG(W112 / W110)
C(35) = LOG(2160.46)

FOR I = 1 TO 45
FOR J = 1 TO 45
P#(I, J) = 0
AT#(I, J) = 0
ATX#(I, J) = 0
SC#(I, J) = 0
GC#(I, J) = 0
NEXT J, I

FOR I = 1 TO 35
FOR J = 1 TO 16
P#(I, J) = B(I, J)
NEXT J, I

FOR I = 1 TO 16
FOR J = 1 TO 35
AT#(I, J) = B(J, I)
NEXT J, I

N = 35
M = 35
L = 16

GOSUB 100

REM ****** SANAL ÇÖZÜM ******

FOR I = 1 TO N
FOR J = 1 TO N
```

```
YS#(I, J) = ATX#(I, J)
NEXT J, I

GOSUB 550

FOR I = 1 TO 16
FOR J = 1 TO 35
PR1# = PR1# + X(J) * B(J, I)
NEXT J
CG#(1, I) = PR1#
PR1# = 0
NEXT I

FOR I = 1 TO 16
GC#(1, I) = EXP(CG#(1, I))
NEXT I
T1 = -6666.67 / (LOG(GC#(1, 12) / 5.9755E+09))

CLS
PRINT
PRINT "E1    ="; GC#(1, 1)
PRINT "E2    ="; GC#(1, 2)
PRINT "FA    ="; GC#(1, 3)
PRINT "FD    ="; GC#(1, 4)
PRINT "FG    ="; GC#(1, 5)
PRINT "FP    ="; GC#(1, 6)
PRINT "FR    ="; GC#(1, 7)
PRINT "FRA   ="; GC#(1, 8)
LOCATE 2, 35: PRINT "FRB   ="; GC#(1, 9)
LOCATE 3, 35: PRINT "FRC   ="; GC#(1, 10)
LOCATE 4, 35: PRINT "FRP   ="; GC#(1, 11)
LOCATE 5, 35: PRINT "k1    ="; GC#(1, 12)
LOCATE 6, 35: PRINT "V     ="; GC#(1, 13)
LOCATE 7, 35: PRINT "B     ="; GC#(1, 14)
LOCATE 8, 35: PRINT "b     ="; GC#(1, 15)
LOCATE 9, 35: PRINT "ALFA  ="; GC#(1, 16)
LOCATE 10, 35: PRINT "T    ="; T1
INPUT "", QWER
GOTO 50
END

40  CLS
PRINT "Dual Çözümler......................................."
PRINT "W01 = "; W01: PRINT "W02 = "; W02: PRINT "W03 = "; W03
PRINT "W04 = "; W04: PRINT "W05 = "; W05: PRINT "W06 = "; W06
PRINT "W07 = "; W07: PRINT "W11 = "; W11: PRINT "W21 = "; W21
PRINT "W22 = "; W22: PRINT "W23 = "; W23: PRINT "W31 = "; W31
PRINT "W32 = "; W32: PRINT "W41 = "; W41: PRINT "W51 = "; W51
PRINT "W52 = "; W52
PRINT "W61 = "; W61
LOCATE 2, 32: PRINT "W71 = "; W71
LOCATE 3, 32: PRINT "W72 = "; W72
LOCATE 4, 32: PRINT "W73 = "; W73
LOCATE 5, 32: PRINT "W81 = "; W81
```

```
LOCATE 6, 32: PRINT "W82 = "; W82
LOCATE 7, 32: PRINT "W83 = "; W83
LOCATE 8, 32: PRINT "W84 = "; W84
LOCATE 9, 32: PRINT "W91 = "; W91
LOCATE 10, 32: PRINT "W92 = "; W92
LOCATE 11, 32: PRINT "W93 = "; W93
LOCATE 12, 32: PRINT "W94 = "; W94
LOCATE 13, 32: PRINT "W95 = "; W95
LOCATE 14, 32: PRINT "W96 = "; W96
LOCATE 15, 32: PRINT "W101= "; W101
LOCATE 16, 32: PRINT "W102= "; W102
LOCATE 17, 32: PRINT "W111= "; W111
LOCATE 18, 32: PRINT "W112= "; W112

INPUT "", QWER

11  CLS
    PRINT "UYGULAMA SAYISI.....:"; IT
    PRINT
    PRINT "——————————"
    PRINT "Wmt CARPANLARI      "
    PRINT "——————————"
    PRINT
    PRINT "W10  =  "; W10
    PRINT "W20  =  "; W20
    PRINT "W30  =  "; W30
    PRINT "W40  =  "; W40
    PRINT "W50  =  "; W50
    PRINT "W60  =  "; W60
    PRINT "W70  =  "; W70
    PRINT "W80  =  "; W80
    PRINT "W90  =  "; W90
    PRINT "W100 =  "; W100
    PRINT "W110 =  "; W110

    INPUT "", QWER

    MINY1 = (14 / W01) ^ (-W01) * (33.66 / W02) ^ (W02) * (56 / W03) ^ (W03) * (325.92 / W04) ^
(-W04) * (.37 / W05) ^ (W05) * (.4 / W06) ^ (W06) * (.4 / W07) ^ (W07)
    MINY2 = ((.73398 * W10) / W11) ^ (W11) * ((1.56268 * W20) / W21) ^ (-W21) * ((.24466 * W20)
/ W22) ^ (-W22) * ((.15627 * W20) / W23) ^ (-W23) * ((1.25014 * W30) / W31) ^ (-W31)
    MINY3 = ((.24466 * W30) / W32) ^ (W32) * ((.31254 * W40) / W41) ^ (W41) * ((1 / W51) * W50)
^ (-W51) * ((.31254 * W50) / W52) ^ (-W52) * ((1 / W61) * W60) ^ (W61)
    MINY4 = ((1 / W71) * W70) ^ (-W71) * ((1.25014 * W70) / W72) ^ (W72) * ((.24466 / W73) *
W70) ^ (-W73) * ((1 / W81) * W80) ^ (W81) * ((.73398 / W82) * W80) ^ (W82)
    MINY5 = ((1 / W83) * W80) ^ (W83) * ((1 / W84) * W80) ^ (-W84) * ((1 / W91) * W90) ^ (W91) *
((.24466 / W92) * W90) ^ (W92) * ((.15627 / W93) * W90) ^ (W93) * ((1 / W94) * W90) ^ (W94) * ((11
/ W95) * W90) ^ (W95)
    MINY6 = ((1.56268 / W96) * W90) ^ (W96) * ((.00772 / W101) * W100) ^ (W101) * ((6.17975 /
W111) * W100) ^ (W111) * ((W100 / W102) ^ (W102)) * ((W110 / W112) ^ (W112))


    FMIN = 1 / (MINY1 * MINY2 * MINY3 * MINY4 * MINY5 * MINY6)
```

```
    CLS
    PRINT "MINY1 ="; MINY1
    PRINT "MINY2 ="; MINY2
    PRINT "MINY3 ="; MINY3
    PRINT "MINY4 ="; MINY4
    PRINT "MINY5 ="; MINY5
    PRINT "MINY6 ="; MINY6
    PRINT
    PRINT "FMIN  ="; FMIN

    INPUT "", QWER


    RETURN
    END
100 REM ***** MATRİS ÇARPIM *****
    FOR I = 1 TO N
    FOR J = 1 TO M
    FOR K = 1 TO L
    ATX#(I, J) = ATX#(I, J) + (P#(I, K) * AT#(K, J))
    NEXT K
    NEXT J
    NEXT I
    RETURN
250 REM ***** GAUSS ELİMİNASYONU *****
    FOR I = 1 TO 45
    FOR J = I TO 45
    IF I = J THEN GOSUB 251
    NEXT J, I
    RETURN
251 FOR K = 1 TO 45
    IF K = I THEN GOTO 252
    RT# = YS#(K, J) / YS#(I, J)
    FOR X = 1 TO 46
    YS#(K, X) = YS#(K, X) - ((YS#(I, X)) * RT#)
    NEXT X
    YS#(K, J) = 0
252 NEXT K
    RETURN

550 REM ***** GAUSS - SEIDEL *****

    ITR = 0
    EPS = .01
    MITR = 300
    CLS
551 PRINT ITR
    FARK = 0
    FOR I = 1 TO N
    TOP = 0
    FOR J = 1 TO N
    IF J = I THEN GOTO 555
    TOP = TOP - YS#(I, J) * X(J)
```

```
555  NEXT J
     XYE = (C(I) + TOP) / YS#(I, I)
     KAR = ABS((X(I) - XYE) / XYE)
     IF KAR > FARK THEN FARK = KAR
     X(I) = XYE
     NEXT I
     ITR = ITR + 1
     IF ITR < MITR THEN GOTO 600
     PRINT " MAX  ITR  SAYISI = MITR ="; MITR; "RAĞMEN"; EPS; "HASSASLI"
     PRINT "ÇÖZÜME ULAŞMADAN EN SON ELDE EDİLEN ÇÖZÜM AŞAĞIDADIR:"
     FOR I = 1 TO N
     PRINT X(I);
     NEXT I
     INPUT "", QWER
     GOTO 660
600  IF FARK > EPS THEN GOTO 551
     PRINT "ITR SAYISI = "; ITR
     PRINT "EPS       = "; EPS
     FOR I = 1 TO N
     PRINT X(I);
     NEXT I
     PRINT "FARK  = "; FARK
     INPUT "", QWER
660  RETURN

     DATA -1,1,1,-1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA -1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,1,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,1,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,1.67,-0.25,-0.67,-0.25,-1.25,1.67,1.25,-1,-1.25,0,0,1.25,-
          .67,0,1.67,0,0,0,0.67,0.25,0,0,0.25,1,0,-1,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,1,0,0,0,0,0,0,0,0,0,2,0,0,0,-1,-1,-1,1,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 1,-1,-1,1,-1,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,1,-1,-1,-2,-1,1,1,0,-1,0,0,1,-1,0,1,0,0,1,1,2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,1,0,-1,0,0,1,0,0,0,0,-1,0,-1,0,1,1,0,0,1,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,1,0,0,0,-1,1,1,-1,-1,1,0,1,-1,0,1,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,1,1,1,0,0,0,-1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,1,0,-1,0,1,-1,-1,0,0,1,0,0,0,0,0,0,-1,0,1,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,1,0,0,-1,0,1,0,1,-1,1,0,0,0,-1,0,0,-1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,-1,-1,-1,-1,-1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,-0.01,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,1,0,0,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,-0.01,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,1,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,-1,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,-1,0,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,-1,0,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,-1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0, 0,0,0,0,0,0,0,0,0,0,-1,0,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,-1,0
     DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,-1
```

```
REM MODEL DENKLEMLERIN DATASI

DATA 0,0,1,0,0,0,0,0,0,0,0,0,-1,0,0,0
DATA 0,0,0,1,0,0,0,0,0,0,0,0,-1,0,0,0
DATA 0,0,0,0,1,0,0,0,0,0,0,0,-1,0,0,0
DATA 0,0,0,0,0,1,0,0,0,0,0,0,-1,0,0,0
DATA 0,0,0,0,0,0,1,0,0,0,0,0,-1,0,0,0
DATA -0.01,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,-0.01,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,-1,0,0,0,0,1,1,0.67,0,1,0,0
DATA 0,0,0,0,0,0,0,-1,0,1,0,0.25,0,0,0,0
DATA 0,0,0,0,0,0,0,-1,-1,1,1,0.67,0,0,0,0
DATA 0,0,0,0,0,0,0,-1,0,2,0,0.25,0,0,-1,0
DATA 0,0,0,0,0,-1,0,0,1,1,0,1.25,0,1,0,0
DATA 0,0,0,0,0,-1,0,0,0,1,1,1.67,0,1,0,0
DATA 0,0,0,-1,0,0,0,0,1,1,0,1.25,0,1,-1,1
DATA 0,0,-1,0,0,0,0,1,1,0,0,1,0,1,0,0
DATA 0,0,-1,0,0,0,0,1,1,1,0,1.25,0,1,-1,0
DATA 0,0,0,0,0,0,2,0,0,0,0,0,-1,1,0,0
DATA 0,0,0,0,0,0,0,0,0,0,1,0,0,0,-1,0
DATA 0,0,0,0,0,0,0,0,0,1,1,0,1.25,0,1,-1,0
DATA 0,0,0,0,0,0,0,0,0,0,1,1,1.67,0,1,-1,0
DATA 0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,1
DATA 0,0,0,0,0,0,-1,0,0,1,1,1.67,0,1,0,0
DATA 0,0,0,0,0,0,-1,0,0,0,1,0,0,0,0,0
DATA 0,0,0,0,0,0,-1,0,0,0,0,0,0,0,1,0
DATA 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,-1
DATA 0,0,0,0,0,0,0,0,-1,1,1,0.67,0,0,0,-1
DATA 0,0,0,0,0,0,0,0,0,0,2,0,0.25,0,0,-1,-1
DATA 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,-1
DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,-1
DATA 0,0,0,0,0,0,0,0,0,0,1,0,0.25,0,0,0,-1
DATA 0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0
DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
DATA 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
```

ÖZGEÇMİŞ

Doğum Tarihi : 27 Nisan 1968

Doğum Yeri : Belen

İlk Öğrenimi : İzmit-Derince Cumhuriyet İlkokulu (1974-1979)

Orta Öğrenimi : İzmit-Derince Lisesi (1979-1985)

Yüksek Öğrenimi : Ankara Üniversitesi Fen Fakültesi Kimya Mühendisliği Bölümü (1986-1992)

Görevi : Yıldız Teknik Üniversitesi Kimya-Metalurji Fakültesi Kimya Mühendisliği Bölümü Temel İşlemler ve Termodinamik Ana Bilim Dalı'nda Araştırma Görevlisi