

ÇOK AMAÇLI KARAR VERMEDE ETKİLEŞİMLİ PROGRAMLAMA

VE

KALİTELİ KANATLI YEMLERİ İÇİN BİR UYGULAMA

DOKTORA TEZİ

(İSTATİSTİK)

T. C.
Yükseköğretim Kurulu
Dokümantasyon Merkezi

İhsan ALP

Eylül 1990

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

[Signature]

Danışman

Prof. Dr. Özkan UNVER

Sınav Jürisi

Başkan : Prof. Dr. Alptekin ESİN

Üye : Prof. Dr. Özkan UNVER

Üye : Doç. Dr. Fikri ÖZTÜRK

Üye :

[Signatures]

Bu Tez Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez yazım yönetmeliğine uygundur.

[Signature]

COK AMAÇLI KARAR VERMEDE/ETKİLEŞİMLİ PROGRAMLAMA
VE KALİTELİ KANATLI YEMLERİ İÇİN BİR UYGULAMA

(Doktora Tezi)

İhsan ALP

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Eylül 1990

ÖZ

70 ve 80'li yıllarda Çok amaçlı karar vermenin etkileşimli programlama dalında pek çok algoritma geliştirildi.

Bu araştırmada sözü edilen algoritmaların önce teorik temelleri incelendi. Ardından algoritmanın adımları ve diğer bilgiler verildi.

Çalışmanın üçüncü bölümünde, Korhonen ve Laakso'nun görsel etkileşimli algoritmasında adım büyüklüğü sağ taraf parametrik programlama kullanılarak çok amaçlı bir karar destekleme sistemi oluşturuldu.

Geliştirilen çok amaçlı karar destekleme sistemi kullanılarak kaliteli kanatlı yemleri için bir uygulama yapıldı.

INTERACTIVE PROGRAMMING IN MULTIOBJECTIVE DECISION
MAKING AND AN APPLICATION FOR CHICKEN FEEDS QUALITY

(Ph.D Thesis)

Ihsan ALP

GAZI UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

September 1990

ABSTRACT

Many algorithms have been developed at the interactive branches of Multiobjective Decision Making in the last decades.

In this research, firstly, the theoretical fundamentals of described algorithms are investigated. And then it has been given the steps of algorithms and some related indicators.

At the third section of the study it has been developed a new multiobjective decision support system using the step-size right hand side parametric programming on the Korhonen and Laakso's visual interactive algorithm.

At the end of the study, using the developed multiobjective decision support system it has been applied to the chicken feeds quality.

TEŞEKKÜR

Tez çalışmamı destekleyen ve yardımlarını esirgemeyen sayın Prof.Dr. Özkan ÜNVER, Yrd.Doç.Dr. Hasan BAL ve Yrd.Doç.Dr. Salih CELEBİOĞLUNA, özel çalışma ve yayınlanmamış makalelerini gönderme lütfunda bulunan Pekka KORHONEN, Ralph E. STEUER ve H. ISERMANN'a sonsuz teşekkürlerimi sunarım.

Tez yazımını ihtimamla gerçekleştiren Ali Necip KARABEY ve Tacettin ÇETİNKAYA'ya teşekkürler.

TABLolar

Sayfa No

Tablo 1. Çok amaçlı karar verme tekniklerinin Sınıflandırması.....	18
Tablo 2. Ödenti Tablosu.....	33
Tablo 3. Değişik P değerleri için d uzaklıkları..... p	47
Tablo 4. Karma yemde kullanılan hammadde ve besin değerleri.....	112

SEKİLLER

	Sayfa No
Sekil 1. Baskın küme örnekleri.....	20
Sekil 2. Farksız Bölüşümler.....	26
Sekil 3. (Z_1^0, Z_2^0) noktasından geçen farksız eğri...	28
Sekil 4. Dik çıkış ve F-W algoritmalarının işleyişi	39
Sekil 5. İdeal çözüm ve uzlaşık küme.....	48
Sekil 6. Komşuluklar dışında ilk nokta örneği.....	65
Sekil 7. Komşuluklar dışında en yakın nokta örneği.	68
Sekil 8. Küme sayısını sağlayamama.....	69
Sekil 9. Düzgün ve Weibull dağılımından bileşimler.	72
Sekil 10. Karar üretim şebekesi.....	98

İÇİNDEKİLER

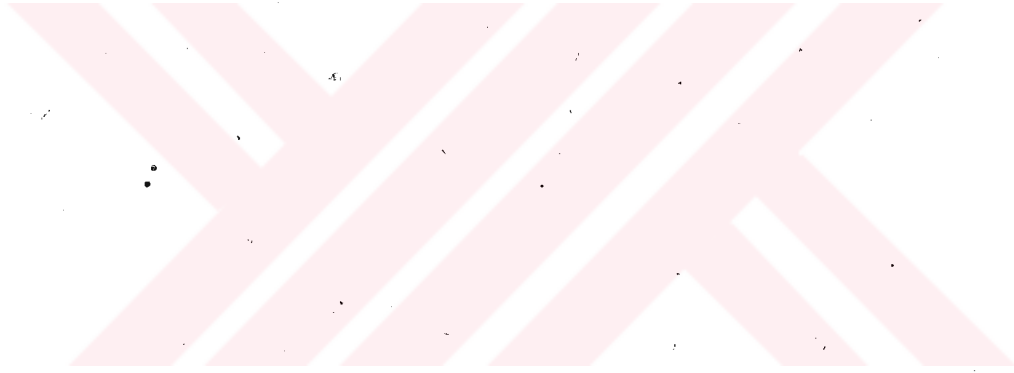
	Sayfa No:
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
TABLOLAR.....	iv
ŞEKİLLER.....	v
1. BÖLÜM	
ÇOK AMAÇLI KARAR VERMEYE GENEL BAKIŞ.....	1
GİRİŞ.....	1
1.1 Çok Amaçlı Karar Verme ve Etkileşimli Programlamanın Tarihçesi.....	4
1.2 Tek Amaçlı D Programlamadan Çok Amaçlı Programlama Geçiş.....	7
1.3 Çok Amaçlı Karar Verme Süreci ve Elemanları...	10
1.4 Çok Kriterli Problem Örnekleri.....	13
1.5 ÇKKV Probleminin Dalları.....	14
1.6 Çok Amaçlı Karar Vermede Etkileşimli Programlama	15
1.7 Çok Amaçlı Karar Verme Yöntemlerinin Sınıflandırılması.....	17
1.8 Baskın Çözümler ve Tanımlar.....	17
1.9 Fayda Fonksiyonu ve türleri.....	20
1.10 Gradyent ve Düzey Egrileri.....	21
1.11 Normlar ve Lp Norm Ailesi.....	22
1.12 Bölüşüm (Trade-off) ve Çeşitleri.....	23
1.13 Farksızlık Egrilerinin Çizimi.....	26
1.14 Etkileşimli Algoritmaların Değerlendirmesi....	28

2. BÖLÜM

ETKİLEŞİMLİ ALGORİTMALAR.....	31
2.1 Giriş.....	31
2.2 Stem Yöntemi.....	32
2.2.1 Odenti Matrisi.....	33
2.2.2 Stem Algoritması.....	34
2.3 GDF Yöntemi.....	37
2.3.1 Hızlı Dik Çıkış Algoritması.....	38
2.3.2 Frank-Wolfe Algoritması.....	40
2.3.3 Uygun Ağırlıklar Üretme Algoritması.....	41
2.3.4 GDF Algoritması.....	43
2.4 Etkileşimli Uzlaşık Programlama.....	45
2.4.1 En-iyi Uzlaşık Çözümler.....	48
2.4.2 Kayan İdeal Nokta Yöntemi Adımları.....	49
2.4.3 Etkileşimli Uzlaşık Programlama.....	51
2.5 Zions-Wallenius (Z-W) Yöntemi.....	55
2.5.1 CKS Yöntemi.....	56
2.5.2 ZW Algoritması.....	59
2.6 Aralık Kriter Ağırlıkları / Vektör	
Enbüyükleme Yaklaşımı.....	61
2.6.1 Süzgeçleme ve Küme Kesikleme.....	62
2.6.2 Aralık Eşitleme Ağırlıkları.....	63
2.6.3 Süzgeçleme Mekanizması.....	64
2.6.4 Küme Sayısını Sağlayamama.....	69
2.6.5 Geriye Doğru Süzgeçleme.....	70

2.6.6 Küme İçi Noktalar Üretme ya da Küme Kesikleme	70
2.6.7 Baskın Çözümleri Azaltmanın Diğer Yolları..	72
2.6.8 Amaç Fonksiyonları Ölçekleme.....	73
2.6.9 Algoritma.....	74
2.7 Etkileşimli Ağırlıklı Toplamlar / Süzgeçleme Yönt.	76
2.7.1 Algoritma.....	79
2.8 Korhonen ve Laakso'nun Görsel Etkileşimli Yöntemi	81
2.8.1 Ölçekleme Fonksiyonları.....	82
2.8.2 Sağ Taraf Parametrik Programlama.....	87
2.8.3 Algoritma.....	89
2.9 Diğer Etkileşimli Yöntemler.....	91
2.10 Etkileşimli Yöntemlerin Karşılaştırılması için yapılan Çalışmalar.....	94
 3. BÖLÜM	
ÇOK AMAÇLI BİR KARAR DESTEKLEME SİSTEMİ.....	96
3.1 Giriş.....	96
3.2 ÇAKV ve Karar Destekleme Sistemleri.....	97
3.3 Yeni Çok Amaçlı Karar Destekleme Sistemi	99
3.4 Yeni Çok Amaçlı Karar Destekleme Sistemi Elm.....	100
 4. BÖLÜM	
KALİTELİ KANATLI YEMLERİ İÇİN BİR UYGULAMA.....	106
4.1 Giriş.....	106
4.2 Yem Maddelerinin Besin Değeri.....	108
4.3 Karma Yemde Kalite Karakteristikleri.....	109

4.4 Karma Yem Problemi Bir Çok Amaçlı Karar	
Problemidir.....	110
4.5 Uygulama.....	111
SONUÇ.....	116
KAYNAKLAR.....	119
PROGRAM.....	129



1.BÖLÜM

ÇOK AMAÇLI KARAR VERMEYE GENEL BAKIŞ

GİRİŞ

Çok amaçlı karar verme (ÇAKV), son onbeş yıldır yöneylem araştırmalarının en hızlı ve en fazla gelişen alanıdır [1,2]. Çünkü, çevremizdeki dünyayı tek boyutlu görmek ve olayları tek kriterle açıklamak artık yeterli görülmemektedir. Karar vericiler bir çok nedenden dolayı, yalnız tek amaçla yetinmemekte ve gerçek dünya problemleri çok amaç dikkate alınarak daha esnek, anlamlı ve doğru modellenmektedir.

Kriterlerin çelişmesi sebebiyle tüm amaçları birlikte enbüyükleyen ya da enküçükleyen tek çözüm, genellikle yoktur. Bu noktada baskın, etkin, Pareto-optimal çözümler önem kazanmaktadır.

İki veya daha fazla amaç dikkate alındığında etkin küme ya bir kenar ya da bir çok yüzlünün (polyhedron) kenar ve yüzey(ler)inden oluşmaktadır. Bu durumda baskın küme sonsuz çözüm içerdiğinden tümünü elde etmek de imkansızdır [3]. Bu aşamada karar vericiye, etkin kümeden tercih edeceği çözümler getiren yöntemler önem kazanmaktadır.

Etkileşimli programlama diye adlandırılan bu yöntemlerde model ile karar verici arasında bilgi alışverişi gerçekleştirilerek problem çözülmemektedir [4,5,6,7].

Bu çalışmada, karar verici ile bilgisayar aracısız doğrudan etkileşerek çok amaçlı karar verme problemlerinin çözümünde kullanılacak bir karar destekleme sistemi geliştirilmiştir.

Yeni çok amaçlı karar destekleme sisteminde çözüm algoritması olarak, Korhonen ve Laakso'nun etkileşimli yönteminde, sağ taraf parametrik programlama yerine sağ taraf adım büyüklüğü parametrik programlama kullanılmıştır.

Çalışma dört bölümden oluşmaktadır. Birinci bölümde konu tanıtılarak gerekli tanımlar, kavramlar ve temel bilgiler verilmektedir. Tek amaçlı karar problemlerinden çok amaçlı karar vermeye, etkileşimli programlamaya geçiş ve tarihçesi verilerek çok amaçlı karar yöntemlerinin sınıflandırılması yapılmaktadır. Algoritmaların değerlendirme kriterleri belirtilmektedir.

İkinci bölümde, literatürde sıkça geçen, önemli etkileşimli algoritmalar için önce , gerekli teorik temel dayanaklar; bu algorithmada kullanılan yöntemler incelenerek ardından, algoritma, adım adım sıralanmaktadır. Bu çalışma kapsamı içine yayınlarda görülen tüm etkileşimli algoritmaları almak, çoğu algorithmada benzer mantıklar kullanılması nedeniyle de gereksizdi. Bölümün sonunda diğer etkileşimli algoritmalarından da bahsedilmiştir.

Üçüncü bölümde karar destekleme sistemi kavramı araştırılarak önemi vurgulanmaktadır. Çok amaçlı karar problemlerine yalnızca uç noktalarda değil, etkin yüzeylerde de dolaşarak çözüm getiren bir algoritma atlama taşı olarak kullanılıp; amaçları birlikte eniyileyen çok amaçlı bir karar

destekleme sistemi oluşturulmuştur. Yeni çok amaçlı karar destekleme sisteminin elemanları ve algoritma ile programda geçen bazı tuşların fonksiyonları bu kısımda verilmektedir.

Dördüncü bölümde, tek amaçla incelenen [8,9,10] kaliteli kaliteli kanatlı yemleri çok amaçlı karar problemi olarak dikkate alınıp , yeni çok amaçlı karar destekleme sistemi kullanılarak çözülmektedir.

1.1 Çok Amaçlı Karar Verme ve Etkileşimli Programlamanın

Tarihçesi

Bu önemli alanda araştırmalar 1970 yılından beri görülmektedir. Gerçektende Zeleny 1970'li yıllarda DR, yöneylem araştırması alanları içinde en hızlı gelişen dalın çok amaçlı karar verme olduğunu belirtmektedir. Bununla birlikte 1970 ve 1980'lerdeki çalışmaların kökü çok önceki yıllara uzanır [1].

Halk, profesyonel karar vericiler, çok amaçlı karar verme olgusundan beri bahsetmektedirler. Örneğin Benjamin Franklin arkadaşı Joseph Priestly'ye 1772 yılında gönderdiği mektupta birçok amaç arasında değiş-tokuştan (trade-off) söz etmektedir. Ancak kavram üzerinde ilk bilimsel çalışmalar II. Dünya savaşı yıllarına rastlar. John von Neumann ve Oscar Morgenstern "birçok çelişen enbüyüklenme problemleri" çıkmazına işaret etmektedirler [11].

J.C. Koopmans 1951 yılında ilk olarak "etkin vektör" yani modern ÇAKV probleminin baskın çözümü, kavramını kullandı. Aynı yılda H.W. Kuhn ve A.W. Tucker vektör enbüyüklenme problemini formüle ederek etkin çözümlerin varlığı için optimumluk şartlarını türettiler. Arrow, Barankin ve Blackwell [12] aynı yönde devam ederek vektör enbüyüklenme problemi ile çok parametrelili doğrusal programlama arasındaki ilişkiyi göstermişlerdir.

Ellili yılların en önemli ürünü Abraham Charnes ve William W.Cooper tarafından 1957 yılında yazdıkları bir makaleye dayanan [13] ilk defa kitaplarında [14] yayınladıkları (Goal programming) Hedef (Amaç) programlamasıdır.

Diğer bir yönde, bugün " çok nitelikli fayda teorisi " (Multiattribute utility teory) olarak bilinen alanın temelleri May (1954) Adams ve Fagot (1959) ve özellikle Yntema ve Torgerson (1961) çalışmalarına dayanır.

Bugün " çok kriterli simplex " diye bilinen yöntemle doğrusal çok amaçlı programlama alanında ilk çalışmaları Peter Bod (1963) yapmıştır.

Altmışlı yılların ortalarında Kuhn - Tucker'in vektör enbüyükleme teorisi bu defa mühendisler, özellikle Lotfi Zadeh [15] bulanık kümeler (fuzzy sets) ve Klinger [16] tarafından yeniden ele alındı.

Yine altmışlı yıllarda çok nitelikli fayda bozunması daha çok ilgi çekti Fishburn [17] ve Pollak [18], toplanabilir fayda bozunması için gerek ve yeter şartları açıkladılar. Bu alanın teorisi üzerinde çok etkili çalışmaların birisi Howard Raiffa'nın 1969 RAND memorandumudur [19]. Benzer etkili bir RAND raporunda Mac Crimmon'a [20] aitti. Dikkate değer diğer bir rapor Keeney'in ayrılabilir fayda fonksiyonları [21] üzerine ilk çalışmasıydı.

Çok amaçlı programlama, Süskü [22], Radzikowski [23] ve Jüttler'in [24] çalışmaları ile yeni boyutlar kazandı. Dinkelbach [25] ve Dinkelbach ve Dürr [26] vektör enbüyükleme probleminin uzantılarını sağladılar. Philip [27], Evans ve Steuer [28] ve Zeleny monografileri [29] doğrusal çok amaçlı programlamayı kesin biçimde ortaya koydu.

CAKV üzerine ilk uluslararası konferans 1972' de South Carolina Üniversitesinde yapıldı. 250 bilim adamı katıldı ve altmışüç çalışma sunuldu. Bu çalışmalar Cochrane ve Zeleny'nin editörlüğü altında hacimli bir kitapta toplandı [30].

Bu konferans CAKV araştırma ve uygulamaları için bir dönüm noktası oldu. Birçok ünlü ve yakında ünlenecek araştırmacılar katıldı: Chuchman, Dawes, MacCrimmon, Fishburn, Keeney, Blin, Roy, Dyer, Yu, Zeleny, Dinkelbach, Isermann, Steuer, Iyiri, Green, Zadeh, Easton, Briskin, Wagner ve Ben-Israel, bunlardan birkaçının ismi.Yetmişlerde CAKV dalında yayın patlaması yaşandı. Yayınlanan makale ve kitap sayısı 1000'in çok üstünde gerçekleşti [31].

Management Science 1977 yılında, Computers & Operations Research dergileri 1980 yılında CAKV üstüne özel sayılar çıkardı.

Yayın patlaması seksenli yıllarda artarak devam etti. CAKV konusunda düzenli yayın yapan peryodik sayısı onu geçti: Computer & Operations Research European Journal of Operational Research Society, Journal of Optimization Theory and Applications, Omega Naval Research Quarterly, Mathematical Programming, Journal of Mathematical Analysis and Applications, Management Science, Operations Research ve Water Resources Research.

Uluslararası konferanslar artık yıl bazında yapılmaya başlanmıştır. Yerel ve bölgesel konferanslar, seminerler ve çalışma gruplarının sayısı hayret vericidir [31].

Etkileşimli programlama algoritmalarından ilki Benoyoun, De Montgolfier ve arkadaşları tarafından 1971 yılında yayınlanmıştır [32].

Geoffrion, Dyer, Feinberg algoritması, kısaca GDF algoritması (1972) karar vericinin amaçlar üzerindeki değiş-tokuş bilgisine dayanmaktadır [33].

Zionts ve Wallenius yöntemi ilk defa 1976 yılında sunulmuş ancak 1983'te yeniden gözden geçirilmiştir [34,35].

Steuer, biri daralan koniler kullanarak, diğeri, ağırlıklı-toplamlar / süzgeçleme işlemiyle ağırlıklandırma uzayını daraltıp etkin çözümleri azaltan, iki önemli etkileşimli algoritma geliştirmiştir [36].

Korhonen ve Laakso, başarı ölçekleme fonksiyonu ve parametrik programlama yardımıyla etkin kümeden çözümler getiren görsel etkileşimli algoritmalarını 1986 yazında EJOR'da yayınlamışlardır.

Son on yılda bilgisayar teknolojisinin gelişmesi ve hafıza büyüklüğüne göre göreceli ucuzlaması etkileşimli programlama algoritmalarının atlama taşı olarak kullanıldığı karar destekleme sistemlerinin ön plana çıkmasına neden olmaktadır.

1.2 Tek Amaçlı Doğrusal Programlamadan Çok Amaçlı

Programlamaya Geçiş

Çok iyi bilinen standart doğrusal programlama problemi,

$$\text{Enb } C^T x, \quad \text{Enb } \{f(x)=z\}$$

$$\text{Kısıtlar : } A x \leq b \quad (1.1) \quad \text{Kısıtlar : } x \in S \quad (1.1^*)$$

$$x \geq 0$$

x karar değişkenleri vektörüdür. C katsayılar vektörü
öyleki $C^T x$ maliyet, fiyat veya kar gibi basit bir amaç
fonksiyonuna karşılık gelir. Kısıt katsayıları A matrisini, b
vektörü ise problemin (kapasite) kısıtlarını oluşturur [37].

$x \geq 0$ kısıtları negatif olmama kısıtları olarak adlandırılır.

Karar değişkenlerinden hiçbirinin negatif olamayacağını belirtir. Literatürde (1.1) yazılımı yerine (1.1*) biçimi de kullanılır. Tüm uygun noktaların (feasible points) kümesi büyük S harfi ile gösterilmektedir. Amaç fonksiyonu $f(x)$ doğrusal ve S uygun çözüm kümesi koordinatları tamsayı ise, tek amaçlı tamsayılı programlama, amaç fonksiyonu veya kısıtlardan bir ya da birkaçı doğrusal değilse tek amaçlı doğrusal olmayan programlamadan söz edilir.

Geçen kırk yılda, tek amaçlı matematiksel programlama problemlerinin çözümü için yöntemler yaygınca çalışıldı. Bununla birlikte tek amaçlı karar verme yöntemleri başlangıç ve ilk dönemleri yansıtır. Dünya daha karmaşıkmaktadır. Bilgi çağına girerken hemen hemen her gerçek dünya problemi birden fazla amaç içermektedir. Bu yüzden karar vericiler, öncekinden daha fazla, çok kritere göre alternatif çözümleri hesaplamanın zorunlu olduğunu gördüler [38,39].

Zeleny ise "çok amaçlı karar verme" için şunları söylemektedir : Çevremizdeki dünyayı tek boyutlu görmek ve görülen herşeyi tek kritere göre değerlendirmek giderek zorlaşmaktadır. Her zaman seçim kriterlerine göre amaçları karşılaştırır, derecelendirir ve sıralarız. Fakat yalnızca basit, açık ve alışılmış durumlarda tek bir seçim kriteri tamamiyle geçerli olabilir.

Şimdi gelecek cümle ile okuyucuya meydan okuyacağız : Enaz iki kriter olmadan karar verme yoktur. Eğer yalnızca tek kriter varsa, bir seçim yapmak için sadece ölçme ve araştırma yetecektir [31].

Pratikte pek çok karar problemi çoklu amaçları dikkate almadan esnek biçimde analiz edilemez [24].

Kar gibi sadece bir amaçla ilgilenen karar verici yoktur [3].

Seksenlerin karar vericileri "öteki taraf" a daha fazla dikkat etmeliler; onlardan beklenen hareketlerini belirlerken çok sayıda değişik kriteri dayandırmaktır. Çoklu ve genellikle karşılaştırılamaz ilgileri, ekonomik, politik, çevresel, sosyal, sağlık, estetik ve öteki kriter kategorilerini kapsar.

Matematiksel olarak ÇAKV problemi k amaç fonksiyonu ve m kısıt ile aşağıdaki biçimde formüle edilebilir [26] :

$$\begin{aligned} \text{enb } \sum_{i=1}^n C_j X_i, \quad & j=1,2,\dots,k \\ \text{Kısıtlar :} \quad & \end{aligned} \quad (1.2)$$

$$\sum_{i=1}^n a_{ki} X_i \leq b_k \quad k=1,2,\dots,m$$

$$X \in R$$

Matris gösterimiyle [5] :

$$\text{enb } Z=Cx$$

Kısıtlar :

$$Ax = b$$

$$x \geq 0$$

A : $m \times n$ boyutlu kısıt matrisi ($m < n$),

C : $k \times n$ boyutlu amaç katsayıları matrisi,

X : $n \times 1$ karar değişken vektörü (Aylak değişkenler de dahil),

b : $m \times 1$ sağ taraf vektörü,

Z : i . amacın kriter değeri,

c : i . amacın gradiyent vektörü.

i

Daha önce de belirtildiği gibi, tüm amaçları birlikte (simultaneously) eniyileyen tek bir çözümün olması nadirdir. Bunun yerine her amaç "mümkün olduğunca" enbüyüklenmeye çalışılır.

1.3 Çok Amaçlı Karar Verme Süreci ve Elemanları

Karar verme, günlük hayatımızın tamamlayıcı bir parçasıdır. Kişiden büyük gruplara ve topluluklara, uluslara ve büyük ölçekte, bütünsel düzeyde uluslararası organizasyonlara varan yayılma alanı vardır [5].

Karar verme, özünde, birden fazla seçenek içinden seçim yapma işlemidir. Genelde, yönetimin karar verme ve yöneticinin karar verici ile eşanlamda kullanıldığı hatırlanırsa karar vermenin süreklilik gösteren bir işlev olduğu görülür. Bu nedenle karar verme üzerinde çalışırken öncelikle karar süreci kavramı ele alınır [40].

Bir çok amaçlı karar verme işleminin (CAKV) dört önemli anahtarı vardır :

- 1- Karar verme birimi,
- 2- Alternatifler kümesi,
- 3- Amaçlar kümesi,
- 4- Karar kuralı.

1.3.1 Karar Birimi

Churchman [41], karar vericiyi , sistemi değiştirme kabiliyeti olan; yani böyle bir değişiklik için, kendisinde otorite olan ve sorumluluk taşıyan kişi olarak tanımlamaktadır.

Karar verici, doğrudan ya da dolaylı yoldan son değer yargılarını düzenleyip, eldeki alternatifleri sıralayarak "eniye"yi belirleyen kişi veya gruplar olabilir. Buradan enküçük karar verme biriminin tek kişi olduğu çıkarılabilir. Büyük bir karar ünitesi ise sistem analistleri, grafik ve hesaplama aletleri, baskı grupları gibi insan ve makinalar topluluğudur. Bilgileri alır, kendi içinde bilgi üretir; bilgiyi fikre dönüştürür ve karar üretir.

Analistin görevi, tüm ilgili bilgileri ; özellikle kriterlerin erişebilir- değerleri ve uzlaşık çözümleri temin etmektir [42].

1.3.2 Alternatifler Kümesi

Kararları içinden seçeceğimiz alternatifler kümesi S , n boyutlu karar değişkenleri (X) vektörlerinden oluşur. Her alternatif X 'in değerlerine göre tamamen tanımlıdır. Alternatifler kümesi boş olmayan, tek ya da sonlu sayıda elemandan oluşabildiği gibi, özellikle ÇAKV problemlerinde sonsuz sayıda elemandan oluşması da sözkonusudur.

Alternatifler olunca, eniyi için, karşılaştırmalar gerekecektir. Bu nedenle amaçlar ve değişkenler kümesi ölçümleri açık ve net olmalıdır.

1.3.3 Amaçlar Kümesi

Amaçlar (Objectives), nitelikler (Attributes) ve hedeflerin (goal) özelliklerini, yapısını anlamak oldukça zordur. Bu kavramlar hakkında oldukça bol örnek ve açıklamaları Zeleny [29], Keeny ve Raiffa [43] vermektedirler.

Nitelikler, bu terim, açık gerçeklğin tanımlayıcısı biçiminde tanımlanabilir. Gerçek, açık özellikler veya subjektif atanmış özellikler alabilir, ama "dışardaki" dünyada objelerin karakteristikleri olarak anlaşılır. Örneğin karar verici yeni bir otomobil seçiyor. Otomobilin nitelikleri beygirgücü, ağırlık, fiyat ve renk gibi subjektif olmayan, görünüş, stil ve statü imajı gibi objektif olmayan, özelliklerdir. Tüm bu özellikler az-çok ölçülebilirdir.

Amaçlar karar vericinin istekleri ve ihtiyaçları ile yakın, olarak belirlenabilir. Tek veya karışık niteliklerle ilgili tercih ve gelişim yönünü gösterir. Yalnız iki yön vardır: az veya çok, yani enküçük veya enbüyük. Bundan dolayı yükseklik kendi başına niteliktir, ama seçeneklerden enuzununu bulmak veya yüksekliği enbüyüklemek bir amaçtır [29].

Hedefler ise, karar vericinin istekleri ve ihtiyaçları ile tamamen belirlidir. Hedefler, önceden belirli nitelik ve amaçlarla tanımlanan özel değer veya düzeylerdir. Ağrı'nın zirvesine ulaşmak hedef ise, bunu enkısa sürede gerçekleştirmek amaçtır.

1.3.4 Karar Kuralı

Bir karar verme işleminde, elde edilebilir alternatifler arasından "eniysi" seçilmeye çalışılır. Bu, ÇAKV problemi için seçilen tüm niteliklerin ölçülen değerlerine bağlı kalite ve performansa göre alternatiflerin hepsinin sıralanmasını gerektirir. Alternatiflerin tam sıralanmasını kolaylaştıran bir kurallar kümesi, karar kuralıdır. Karar kuralı, amaçlar cümlesiyle hemen hemen tamamiyle belirlenebilir.

Pratikte, karar kuralları, kabaca iki kategoriye ayrılabilir: Optimum ve sağlama (satisficing) kuralı. Bir karar kuralı optimum sınıfında ise, tüm elde edilebilir alternatifler tamamiyle sıralanmalıdır. Bu küme içinde, karar kuralına göre eniyi alternatif vardır. Diğer yandan sağlama sınıfındaki bir karar kuralı iddialı değildir. Sağlama kuralına göre "sağlanan" bir alternatif arandığında, optimumluk, para ve zamanda önemli tasarrufa feda edilir.

Laplace, rasyonellik, optimistik, pessimistik gibi başka karar kuralları da vardır [29].

1.4. Çok Kriterli Problem Örnekleri

Gerçek dünya problemlerinde amaçların tümü aynı yönde değildir, yani hepsi enbüyüklenme veya enküçüklenme tipinde olmaz. Çoğu zaman amaçlar çelişir. Örneğin bir imalat probleminde minimum maliyet maksimum kalite karakteristiğiyle çelişir. Bazı çok kriterli örnekler aşağıdadır.

Yayıncılık : enb {yayın başımı}, enk {stok miktarı}, enb {gelir}.

Taşımacılık: enb {kar}, enb {taşınan madde}, enk {yakıt tüketimi}
enk {katedilecek yol}.

Kimyasal Karışım : enb {kar}, enk {tehlikeli madde}, enk {ham
madde}, enb {üretim}.

Hot Dog, Bologna
ve

Sucuk Karışımı : enk {maliyet}, enk {yağ}, enk {renk sapması},
enk {protein}, enk {nem hedefinden sapma}[36].

Yatırım Seçimi : enb {gelir}, enb {yatırım güvenliği}, enk {top-
lam maliyet}, enb {faiz} [44].

Keeney ve Raiffa [43] ÇAKV alanında yeni ufuklar açan kitaplarının giriş bölümünde yaklaşık beş sayfalık güzel örnekler vermişlerdir. Bunlar : Hava kirlenmesine karşı elektrik gücü, Havaalanı yeri seçimi, Newyork'ta eroin alışkanlığını önleme, Tıbbi teşhis ve iş problemlerini kapsar.

Zeleny , [29] kitabının ilk bölümünde, üretim, vergi, ekonomik göstergeler, tıp, nükleer güç, köpek maması, planlama, milli ekonomi, çevre kirlenmesi gibi alanlarda amaçları sıralamıştır.

Steuer, [36] petrol rafinerisi çizelgelemesi, üretim planlaması, portföy seçimi, orman yönetimi v.b. problemler için uygun amaçlar belirlemiştir.

Sürekli yayınlarda ÇAKV ile ilgili her makalede en az bir örnek verilmektedir. Ancak bunları sıralamak uzun olacaktır. Değişik diğer örnekler [30,31,32] nolu kaynaklarda bulunabilir.

1.5 ÇKKV Probleminin Dalları

Değişik ortamlarda karar durumları; "çok amaçlı" , "çok nitelikli", "çok boyutlu" terimleri ile ifade edilmektedir [4,38,29,36]. Evrensel bir tanımlama olmadığından bu terimler birbiri yerine de kullanılabilir.

Çok kriterli karar verme, iki farklı yarımından oluşmaktadır. Birinci yarım, çok nitelikli karar verme (analizi) diğeri ise çok amaçlı karar vermedir.

ÇKKV ; belirsizlik ortamında, az sayıda alternatif arasından seçim problemlerine uygulanır. Keeney ve Raiffa'nın [43] verdikleri örneklerde olduğu gibi, havaalanı, nükleer reaktör yerleşim planlaması, sağlık problemleri ve benzerlerinde kullanılan analiz tekniklerini kapsar.

CAKV ise, kısıtlarla, belirli uygun çözümlerin fazla sayıda, çoğu zaman sonsuz olduğu deterministik problemlere uygulanır.

1.6 Çok Amaçlı Karar Vermede Etkileşimli Programlama

Birden fazla amacı dikkate alma, uygun analiz gereği, çok sayıda, bazen sonsuz çözümü gerektirmektedir. Matematiksel optimum şart olmadığından, karar verici, çözümlerin hesaplanma, sıralanma ve karşılaştırma aşamalarında önemli rol oynar. Bu nedenle "Çok kriterli optimum" terimi kullanmak doğru değildir ve kaynaklarda bunun yerine "baskın çözümler", "iyi" veya kabul edilebilir çözümler, uzlaşık veya yüksek güvenli çözümler gibi kavramlar geçmektedir.

Amaçların çokluğu yanında, birde, karar vericinin çözümleri gördükçe genişleyen tercih problemi vardır. Sistem böyle geliştiğinde karar vericinin bir dereceye kadar ilgisi, formülasyon, çözüm ve hesaplama adımlarında sadece istenmez, zorunludur da. Böylece karar verici, çözüm süreci ardıştırım döngülerinin tamamlayıcısı haline gelir. Bu tür ilgilere "etkileşim" denir. Adımlarında etkileşim bulunan algoritmalar etkileşimli yöntemler, etkileşimli programlama diye adlandırılır.

"Optimum" cevaplar yerine, etkileşim planlanarak, gelişen şartlara uygun, daha iyi çözümler, birbiri ardından elde edilir. Matematiksel programlama teknikleri, böylece, daha gerçekçi, daha esnek ve yöneticiler tarafından çok daha kabul edilir konuma yükselir.

Etkileşim, davranışsal, fonksiyonel ve nitelikli analizle daha iyi bütünleşmek, insani değerlerin önemine dolu değerlendirmeyi ve varsayımları açık biçimde anlamayı gerektirir. Bunların tümü çok yararlı ve hızlı büyüyen yöneylem araştırması alanının gerekli etkin parçalarıdır.

Matematiksel programlamanın etkileşimli felsefesi oldukça yenidir. Karar vermede, insanın rolünü, şimdiki düzeyin ötesine götürmeye çalışır.

Karmaşıklıkla karşılaşınca, insanoğlu, yeri doldurulmaz karar vericidir. Yeteneklerine hala matematiksel analizin başdöndüren süper yapılarıyla ulaşamadı [44].

Cözüm işlemi ilerledikçe karar verici, yalnız tercih bilgisini belirtmekle kalmaz; problemini tanıyarak, amaçlarının sağlama düzeylerini de görür.

Etkileşimli yöntemlerin avantajları şöyle sıralanabilir [4] :

- 1- Bu yöntemlerde öncel tercih bilgisine gerek yoktur.
- 2- Karar verici için sistemin davranışını anlama işlemidir.
- 3- Yalnız yerel tercih bilgisi ister.
- 4- Karar verici çözüm işleminin bir parçası olduğundan beklenenden daha iyi çözüm elde edilebilir.
- 5- Varsayımları diğer yöntemlere göre daha azdır.

Dezavantajları ise,

- 1- Çözümlerin doğruluğunun karar vericiden alınan yerel tercih bilgisine bağlı olması.
- 2- Birçok yöntemin belli sayıda adımla, tercih edilen çözüme ulaşmayı garanti etmemesi.

3- Etkileşimli olmayan yöntemlere göre, karar vericiden daha fazla çaba harcamasını istemesidir.

1.7 Çok Amaçlı Karar Yöntemlerinin Sınıflandırılması

Çok amaçlı karar verme yöntemleri yayınlarda değişik şekillerde sınıflandırılmaktadır.

Hwang ve arkadaşları [4] sınıflandırmada dört durumu baz almışlardır. Bunlar : (1) Karar vericiden vericiden alınacak tercih bilgisi gerektirmeyen, (2) Öncel (a priori) tercih bilgisi gerektiren, (3) Adım adım tercih bilgisi gerektiren, (4) Sonsal (a posteriori) tercih bilgisi gerektiren yöntemler olmak üzere ayrılmıştır.

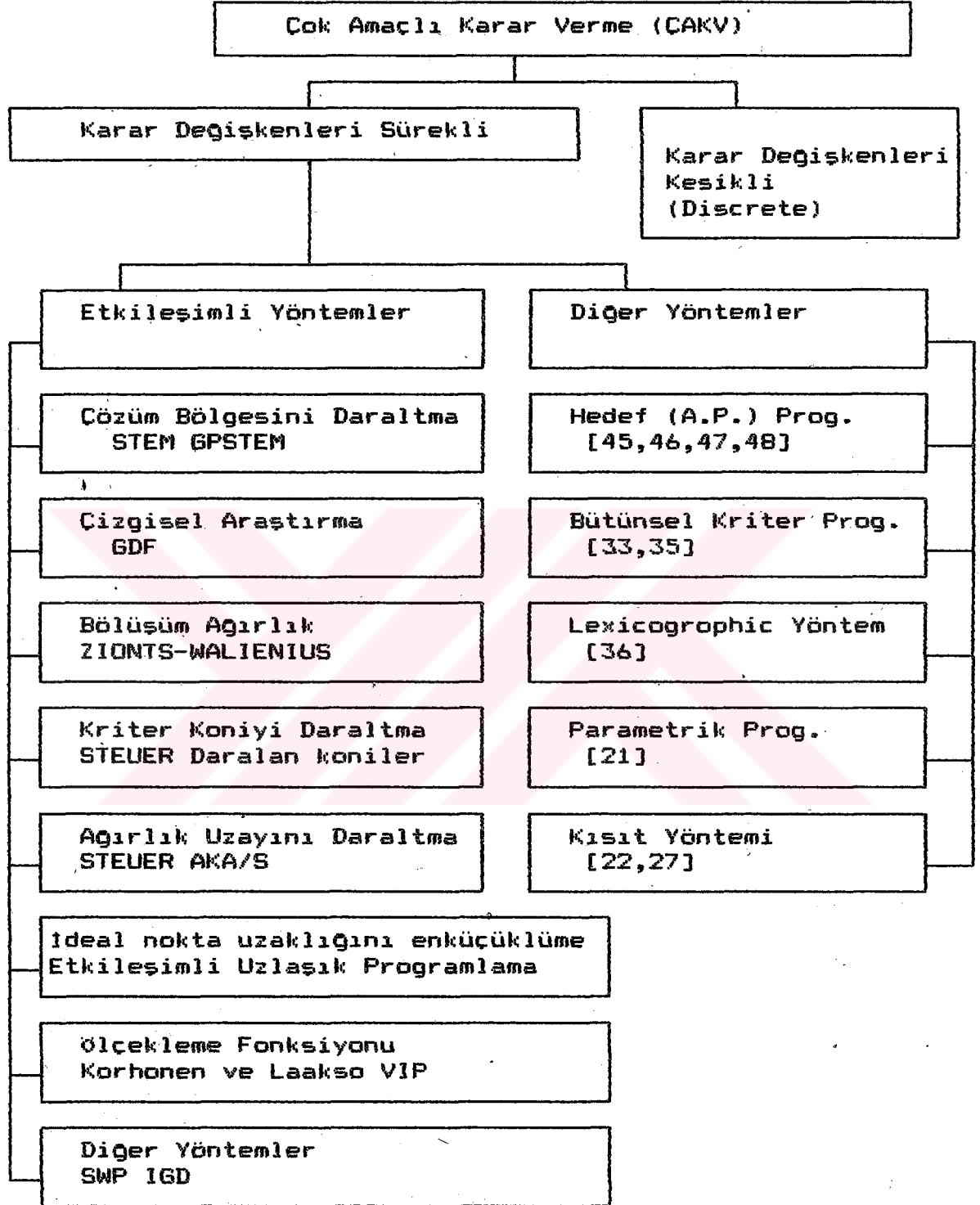
Evans [1] ise, yöntemleri, karar değişkenlerinin sürekli olup olmadığını dikkate alarak, tercih bilgisinin öncel, adım adım, sonsal ve birleşimlerine göre sınıflandırmaktadır.

Aşağıda vereceğimiz sınıflamada (Tablo 1) algoritmalarda kullanılan yöntemin dayandığı mantık esas alınmıştır. Çalışma etkileşimli yöntemler üzerine olması nedeniyle diğer algoritmalar ayrıntılı incelenmemiştir. Yöntemler karar değişkenlerine göre ikiye ayrılmış, sürekli değişkenler için geliştirilen yöntemler etkileşimli ve diğer yöntemler biçiminde gruplandırılmıştır. Her algoritmada geçen temel mantık belirtilmiştir.

1.8 Baskın Çözüm Kümesi ve Tanımlar

Literatürde çok amaçlı karar problemlerinin çözümleri için baskın (nondominant), etkin (efficient), Pareto - optimal deyimleri geçmektedir.

Tablo 1. Çok amaçlı Karar Verme Tekniklerinin Sınıflandırılması



Baskın çözüm kavramı, geleneksel ekonomi teorisinin kurucularından olan Pareto (1906)'ya kadar uzanır. Kuhn ve Tucker etkin çözüm için yeterli ve gerekli şartları 1950 li yıllarda yayınlamışlardır. Genellikle Pareto ya da Pareto en iyileme prensibi diye adlandırılan bu kural : A dünyası B dünyasına, eğer, A dünyasında en az bir kişi B dünyasındaki kişilerden iyi ve hiçkimse kötü değilse tercih edilir biçiminde ifade edilir [21].

Çok amaçlı karar probleminin çözümü ile ilgili bazı tanımlar:

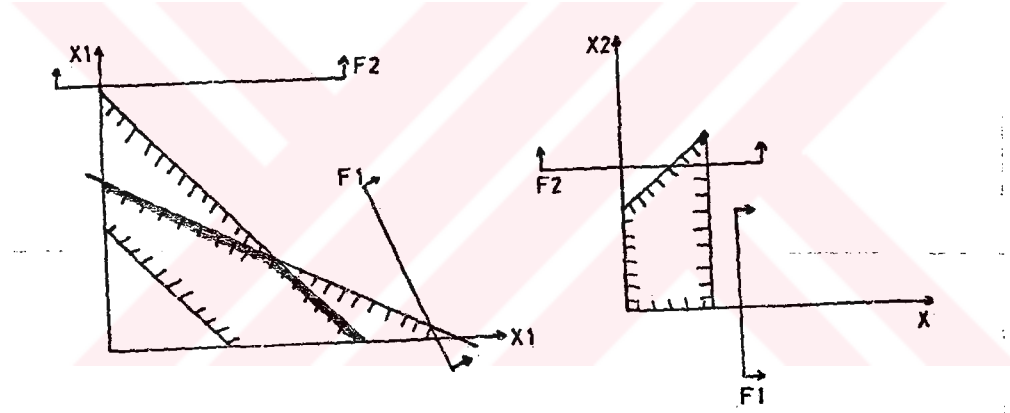
1.8.1 Tanım : Çok amaçlı karar probleminin, $X \in S$ için bir çözüm X^* , süper çözümdür $\Leftrightarrow f_i(X^*) \geq f_i(X)$, $i=1,2,...,k$ ve $x \in S$.

Süper çözümle elde edilen amaç fonksiyon değerleri "ideal" i oluşturur. Tipik olarak k amaçtan enaz ikisi tabii olarak çelişir. Örneğin bir sistemin maliyetini düşürmek, sistemin güvenilirliğini veya kalitesini olumsuz yönde etkiler. Bu tür problemlerde karar verici etkin, baskın çözümler ile ilgilenir.

1.8.2 Tanım : Çok amaçlı karar probleminin, $X \in S$ için bir çözüm X^* , etkin bir çözümdür eğer başka uygun çözüm yok ise öyleki $f_i(X^*) \geq f_i(X)$ tüm $i=1,2,...,k$ ve $f_i(X^*) > f_i(X)$ bazı $i=1,2,...,k$ için.

Doğrusal çok amaçlı karar verme probleminin baskın çözümler kümesi (X^*) ; genellikle uygun çözüm bölgesinin (S) sınırı (efficient frontier), bazen tamamı ve nadiren bir noktasıdır. Amaç fonksiyonlarının hepsi doğrusal bağımsız ise, X^* kümesi uygun çözüm bölgesinin sınırında uzanır. Doğrusal problem için çözüm

kümesi, dışbükey çokyözlü ise, sınırlar, uç kenarlar ya da kenarın iç noktası baskın ise uç noktaları ve (iki ardışık uç noktayı birleştiren) kenarı kapsayan tüm yüz de baskın *
 olmaktadır. X baskın çözümler kümesinin herhangi iki noktasına bu *
 kümeden ayrılmadan geçilebilir, yani X kümesinde bir uç noktadan *
 diğerine X kümesinin kenarlarından çıkmadan ulaşılabilir [5,29].
 İki ve üç boyutlu problemlerde, baskın çözümler kümesi, grafikte amaç fonksiyonları gradiyent yönünde hareket ettirilerek belirlenebilir. Aşağıdaki örneklerde baskın küme koyu çizgi ile gösterilmiştir.



Sekil 1. Baskın küme örnekleri

1.9 Fayda Fonksiyonu ve Türleri

Fayda fonksiyonu, $U: R^k \rightarrow R$ biçiminde tanımlanan ve kriter vektörlerini gerçekte sayılar kümesine taşıyan bir fonksiyondur. Büyük değerli kriter vektörü yüksek faydaya sahip olduğundan tercih edilir. Algoritmalarda değişik fayda fonksiyonları geçmesi nedeniyle bazı tanımlamalar yapılacaktır [36].

1.9.1 Tanım : $U:R \xrightarrow{k} R$ azalmayan bir fonksiyondur
 $\Leftrightarrow$ tüm $Z, Z \in R$ öyleki $Z \leq Z, U(z_1) \leq U(z_2)$

1.9.2 Tanım : $U:R \xrightarrow{k} R$ artan bir fonksiyondur
 $\Leftrightarrow$ tüm $Z, Z \in R$ öyleki $Z < Z, U(z_1) < U(z_2)$

1.9.3 Tanım : $U:R \xrightarrow{k} R$ dışbükey bir fonksiyondur
 $\Leftrightarrow$ tüm $Z, Z \in R$
 $U(wz_1 + (1-w)z_2) \geq wU(z_1) + (1-w)U(z_2) \quad (1.4)$

tüm $w \in [0,1]$.

1.9.4 Tanım : $U:R \xrightarrow{k} R$ kesin dışbükeydir
 $\Leftrightarrow$ tüm $Z, Z \in R$ öyleki $Z = Z$
 $U(wz_1 + (1-w)z_2) > wU(z_1) + (1-w)U(z_2) \quad (1.5)$

tüm $w \in [0,1]$

1.9.5 Tanım : $U:R \xrightarrow{k} R$ yarım dışbükeydir (quasiconcave) bir fonksiyondur $\Leftrightarrow$ tüm $Z, Z \in R$ öyleki $U(z_1) < U(z_2)$
 $U(z_1) \leq U(wz_1 + (1-w)z_2)$ tüm $w \in [0,1]$. (1.6)

1.9.6 Tanım : $U:R \xrightarrow{k} R, R$ da türevlenebilir olsun.

U yalancı dışbükey (pseudoconcave) bir fonksiyondur $\Leftrightarrow$ $Z, Z \in R$ öyleki
 $\nabla U(z_1)(z_2 - z_1) \leq 0, U(z_1) \leq U(z_2) \quad (1.6)$

1.10 Gradyent ve Düzey Egrileri [36]

Doğrusal programlamada, terimler, bilinmeyenleri sol tarafta (LHS) ve sabitleri sağ tarafta (RHS) bırakarak düzenlenir.

Grafik gösterimlerinde "gradyent ve düzey egrileri" düzeninden sık sık yararlanılır. Bir doğrusal ifadenin düzey eğrisi (veya contour) sağ taraf değerini sağlayan noktaların

R^2 hattıdır. R^2 'de düzey eğrileri (bir boyutlu) düzgün doğrulardır. R^3 'de düzey eğrileri (iki boyutlu) yüzeylerdir. R^n 'de düzey eğrileri $n-1$ boyutlu çok yüzlüdür (hyperplanes). Verilen bir doğrusal ifadenin bütün düzey eğrilerinin birbirine paralel olması düzey eğrilerinin yararlı özelliklerinden biridir.

Bir doğrusal ifadenin gradiyenti, sol taraf bilinmeyenlerinin katsayılarının, oluşturduğu vektörden elde edilir. Tüm gradiyentler kendi düzey eğrilerine normal (dik) dir. Gradiyentler, bir ok çizildiğinde artan büyüme yönünü gösterdiğinden önemlidir.

1.11 Normlar ve Lp-Norm Ailesi

Normlar vektörlerin uzunluklarını ölçer. Bu norm R^n üzerinde her $V \in R^n$ vektörünü $\|V\| \in R$ skalaya dönüştüren bir fonksiyondur. Aşağıdaki aksiyomları sağlaması gerekir.

- 1- $\|V\| \geq 0$
- 2- $\|V\| = 0 \iff V = 0 \in R^n$
- 3- $\|kV\| = k\|V\|, k \in R$

$V \in R^n$ olmak üzere Lp-norm ailesi

$$\|V\|_p = \left[\sum_{i=1}^n |V_i|^p \right]^{1/p} \quad p \in (1, 2, 3, \dots) \cup \{\infty\} \quad (1.7)$$

Çok kullanılan $p=1, 2, \infty$ Lp-normları,

$$\|V\|_1 = \sum_{i=1}^n |V_i| \quad (1.8)$$

$$\|V\|_2 = \left[\sum_{i=1}^n |V_i|^2 \right]^{1/2} \quad (1.9)$$

$$\|V\|_\infty = \max_{i=1,2,\dots,n} (|V_i|) \quad (1.10)$$

Bir vektörü L_p -normuna göre normalleştirmek için

$$\frac{v}{\|v\|_p} \quad \text{bağıntısı kullanılır.}$$

1.11.1 L_p -Ölçüm Ailesi

R^n ölçüm R^n içinde noktalar arasındaki uzaklıkları ölçer.

$X, Y \in R^n$ iki nokta olmak üzere L_p -ölçüm ailesi

$$\|x-y\|_p = \left[\sum_{i=1}^n |x_i - y_i|^p \right]^{1/p} \quad P \in \{1, 2, 3, \dots\} \cup \{\infty\} \quad (1.11)$$

$p=\infty$ ölçümü Tchebycheff ölçümü olarak isimlendirilir.

$x, y \in R^n$ olmak üzere iki nokta olsun. Ağırlıklı L_p -ölçüm ailesi,

$$\|x-y\|_p^w = \left[\sum_{i=1}^n (W_i |x_i - y_i|)^p \right]^{1/p} \quad (1.12)$$

$$P \in \{1, 2, 3, \dots\} \cup \{\infty\}$$

Burada $W \in R^n$ pozitif ağırlıklar vektörüdür.

1.12 Bölüşüm (trade-off) ve Çeşitleri

Etkileşimli algoritmaların çözüm aşamasında karar verici amaçların gerçekleşen değerlerine göre bazı amaç değerlerinde artma için diğer amaç değerlerinde belli bir oranda azalmayı kabul eder. Bu işlem algoritmada değişik yollarla gerçekleştirilir. Gelen kesimlerde bu kavram üzerinde duracağız.

Webster'in yeni dünya sözlüğüne göre bölüşüm : bir değişim, birini daha fazla istenir kılmak için diğerinin fayda ve avantajını azaltmaktır [49]. Bu kavram ÇAKV problemlerinde sıkça ve serbestçe kullanılmaktadır.

Toplam bölüşüm, kısmi bölüşüm, ferksiz bölüşüm oranı gibi çeşitli kavramların tanımları bu kısımda verilecektir.

1.12.1 Toplam ve Kısmi Bölüşümler :

İkiden çok amaç olduğunda toplam bölüşümden söz etmek gerekir. İki uygun alternatif x^o ile x^* ve amaçların karşılık geldikleri düzey sırasıyla,

$$f(x^o) = \{f_1(x^o), f_2(x^o), \dots, f_n(x^o)\} \text{ ve}$$

$$f(x^*) = \{f_1(x^*), f_2(x^*), \dots, f_n(x^*)\} \text{ verilsin. } f_k \text{ ve } f_j$$

arasındaki değişim oranı $T_{kj}(x^o, x^*)$ ile gösterilsin.

$$T_{kj}(x^o, x^*) = [f_k(x^o) - f_k(x^*)] / [f_j(x^o) - f_j(x^*)] \quad (1.13)$$

1.12.1.1 Tanım : Tüm $l=1, \dots, n$, $l \neq k$ ve $l \neq j$ için

$$f_l(x^o) = f_l(x^*) \text{ ise } x^o \text{ ve } x^* \text{ arasındaki } f_k \text{ ve } f_j \text{ ile}$$

tanımlanan $T_{kj}(x^o, x^*)$ kısmi bölüşüm (partial trade-off) olarak adlandırılır.

Diğer yandan en az bir $l=1, \dots, n$ ve $l \neq k$, $l \neq j$ için

$$f_l(x^o) \neq f_l(x^*) \text{ ise } T_{kj}(x^o, x^*) \text{ 'ye toplam bölüşüm denir.}$$

Kısmi ve toplam terimleri kısmi ve toplam türeve benzer olarak kullanılır. Kısmi bölüşüm farksız eğri aileleri kurarken bulunabilir. Bu kavramların önemi, karar vericiye bir defada iki amaç değerini karşılaştırarak tercih yapmasını kolaylaştırmasıdır.

1.12.2 Farksız Bölüşüm

x^o ve x^* verilen iki alternatif ise $f(x^o)$ ve $f(x^*)$ aynı farksız (aynı tercihtli) eğri üzerinde uzanıyorsa karşılık gelen bölüşüm olarak bilinir. Bu terim karar vermede çok yaygın kullanılmaktadır.

$U(f_1, f_2, \dots, f_n)$ fayda fonksiyonunun farksız eğriler

$$\text{ailesi } U(f_1, f_2, \dots, f_n) = \text{sabit} \quad (1.14)$$

ile verilir. Varsayalımki farksızlık eğrisi üzerinde (farksızlık eğrisi çizimi bir sonraki kısımda anlatılacaktır) sabit bir f^* noktasında f_j ve f_k arasında farksız bölüşüm oranını bulmak istiyoruz.

$$U(f_1, f_2, \dots, f_n) = U_1 \quad (1.15)$$

f_j amacı da dahil olmak üzere ve serbestlik derecesi bir eksilterek geri kalan amaçların fonksiyonu biçiminde ifade edilir.

$$f_k = f_k(f_1, \dots, f_{k-1}, f_{k+1}, \dots, f_n) \quad (1.16)$$

f_k noktasında f_k ve f_j arasındaki farksız bölüşüm oranı

(indifference trade-off rate), M_{kj} ,

$$M_{kj} = \frac{\partial f_k / \partial (f_1, \dots, f_{k-1}, f_{k+1}, \dots, f_n)}{\partial f_j} \bigg|_{f=f^*} \quad (1.17)$$

Alternatif olarak

$$U(f_1, f_2, \dots, f_n) = U_1 \quad (1.18)$$

eşitliğini kullanarak ve zincirleme kuralını uygulayarak

$$M_{kj} = \frac{\partial U(f^*) / \partial f_j}{\partial U(f^*) / \partial f_k} \quad (1.19)$$

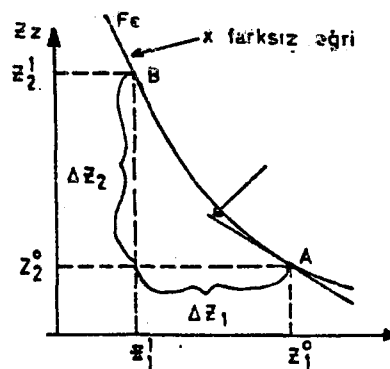
bulunabilir.

1.12.3 Objektif ve Subjektif Bölüşümler

" Objektif " ve " subjektif " terimleri, bölüşümlerde kişisel yargıların karışma derecesini belirtmek için kullanılır [50]. Bölüşüm yalnızca sistemin içsel yapısı altında çalışılarak hesaplanmış ve uygunsa objektif bölüşüm (ya da bazen dönüşümün marjinal oranı) denir. Diğer yandan bölüşüm, sistemin içsel davranışı dikkate alınmaksızın bazı tercih kriterlerini sağlar biçimde karar verici tarafından belirlenirse subjektif bölüşüm denir. Subjektif bölüşümler çoğu zaman tercih ya da fayda fonksiyonu kavramına dayalı karar vermeye kullanılır. Örneğin GDF algoritması subjektif bölüşüm analizine dayanır [4].

1.13 Farksızlık Eğrilerinin Çizimi

Aynı farksız eğri üzerinde herhangi iki nokta için, karar verici, iki noktanın tercih derecesi aynı kalmak üzere; bir amaç değerinde belli miktarda azalmaya karşılık öteki amaçta belli miktarda artma sağlanıyorsa bölüşüm (trade-off) yapma eğiliminde olur. Bu, farksız bölüşüm veya marjinal değiştirme oranı diye adlandırılır.



Sekil 2. Farksız bölüşümler

$A(Z_1, Z_2)$ alternatifinden $B(Z_1, Z_2)$ alternatifine giderken Z_1

amaç değerinden ΔZ_1 birim kadar azalmaya karşılık Z_2 amaç değerinde ΔZ_2 birim artış olacaktır. Bu değişimde farksız bölüşüm $\Delta Z_2 / \Delta Z_1$. Herhangi (Z_1, Z_2) referans noktasından

geçen farksızlık eğrisini belirlemek (Z_1, Z_2) noktası ile farksız bölüşümlere sahip tüm noktaların bulunması demeye

denktir. $-dZ_2 / dZ_1$ ile verilen (Z_1, Z_2) noktasında

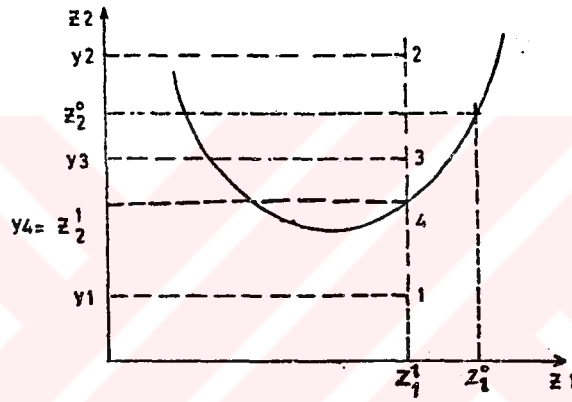
eğriye tanjantın eğimi Z_2 ve Z_1 arasında marjinal değişim oranı olarak adlandırılır.

Maccrimman ve Wehrung [51] iki amaçlı problemler için farksız eğrileri oluşturmak üzere çizgi yöntemi, kare yöntemi ve karo veya daire yöntemi diye adlandırdıkları üç yöntem kullanmaktadırlar.

Tüm bu yöntemler tercih-farksızlık yargılarına dayanmaktadır. Çizgi yönteminde karar vericiden iki alternatiften birini tercih etmesi istenir. Referans notlarından biri (Z_1, Z_2) ve ötekisi (Z_1, Z_2) veya (Z_1, Z_2) olsun. Z_1 (veya Z_2) değeri uygun değerde sabit alınırken Z_2 (veya Z_1) değeri uygun farksızlık sağlanıncaya kadar değiştirilir. (3 nolu Şekile bakınız). Örneğin $Z_1 = y$ değerini seçtik ve karar vericiden (Z_1, Z_2) ve (Z_1, y) noktalarını karşılaştırmasını istedik. Eğer karar verici öncekini sonrakine tercih ederse Z_2 y 'den y 'ye yükseltilir. (Z_1, Z_2) ile (Z_1, y) noktası teklif edilir. Bu defa

sonrakini öncekine tercih ederse y azaltılarak y noktasına gelinir. İşlem tekrarlanır. Varsayalımki karar verici (Z_1, Z_2) ve (Z_1, y) arasında tercihi farksızsa $(Z_1, Z_2 = y)$ noktası (Z_1, Z_2) noktasından geçen farksız eğrinin üzerinde aradığımız noktadır.

İşlemi değişik Z_1 değerleri için (örneğin 1 2 3 4 5 Z_1, Z_2, Z_1, Z_2, Z_1) yinelersek, yeterli sayıda nokta belirlendiğinden istenilen farksızlık eğrisine karşılık gelen kesin eğri ortaya çıkmış olur.



Sekil 3. (Z_1, Z_2) noktasından geçen farksız eğri

1.14 Etkileşimli Algoritmaların Karşılaştırılması ve Değerlendirilmesi

Algoritmaların göreceli performansı, karşılaştırılması ve değerlendirilmesi, eldeki probleme en uygununu seçmek için karar verici, karar verici-analist, makine ile çözüm ve problem özellikleri açısından aşağıdaki sorular sorulabilir [4,36,39,52, 53,54,55]:

1-Karar Verici açısından,

a-Karar vericinin iyi bir matematik ve bilgisayar bilgisine gerek duyuyor mu ?

b-Karar vericiden, yardımcı ağırlık,bölüşüm ve ikili karşılaştırmalar yapması isteniyor mu ?

c-Karar vericiden yargılarını hemen mi yoksa belli bir süre sonra mı istiyor ?

d-Karar vericinin ardıştırma sırasında sağlama düzeylerini görerek,gerekirse algoritmanın başına dönmeden, değiştirme imkanı veriyor mu ?

e-Algoritma, karar vericinin amaçlara ilişkin fayda fonksiyonu hakkında özel varsayımlar yapıyor mu ?

2-Karar verici-Analist ve Analist-Bilgisayar arasındaki diyalog bakımından,

a-Kolay anlaşılır mı ?

b-Bilgisayar programına veri girişi bakımından kullanımı kolay mı ?

c-Tecrübeli analist ve teknik eleman gerektiriyor mu ?

d-Özel yazılım (software) gerektiriyor mu ?

Hazır yazılımlardan yararlanarak çözüme ulaşılabilir mi ?

e-Cok bilgisayar zamanına (CPU) gerek duyuyor mu?

f-Karar vericinin amaç fonksiyonlarının gerçekleşme düzeyleri üzerinde kolayca değişiklik yapabilmesi için yardımcı özelliklerden yararlanıyor mu ?

g-Yöntemin ve ardışım adımlarının temel mantığı kolay anlaşılır mı ?

3-Cözümleri açısından,

- a-Her ardıştırmada karar vericiye tek mi yoksa belli sayıda çözüm mü getiriliyor ?
- b-Algoritmanın çalışması sonunda elde edilen çözüm baskın çözümler kümesinin bir elemanı mı ?
- c-Algoritma ad hoc,yani her zaman doğru sonuç vermeyen bir algoritma mıdır ?
- d-Elde edilen bilgiler faydalı ve karar verici tarafından kolay anlaşılır mı ?

4-Problem özellikleri bakımından ,

- a-Algoritma yalnız ÇADP problemlerini mi yoksa ÇADOP (çok amaçlı doğrusal olmayan programlama) ve CATP (Çok amaçlı tamsayılı programlama) problemlerinin çözümünde de kullanılabilir mi ?
- b-Amac kısıt ve değişken sayısı sınırları nedir ? Büyük ölçekli gerçek dünya problemlerini çözebilir mi ?
- c-Ardıştırma sayısı sabit mi yoksa problemlerinin özelliklerine göre değişmekte mi ?

2. BÖLÜM

ÇOK AMAÇLI ETKİLEŞİMLİ ALGORİTMALAR

2.1 Giriş

Çok amaçlı programlamanın geleceği, onun etkileşimli uygulamalarıdır [36]. Yakın zamanlarda, karar analizinde etkileşimli yöntemler daha popüler olmuştur [42].

Genelde , etkileşimli yöntemler karar verici, analist ve model arasında karşılıklı ve başarılı görüş ve bilgi alış-verişine dayanır.

Bilgisayar teknolojisinin gelişmesi; hafıza birimlerinin reel ucuzlaması, depolama kapasitelerinin büyümesi ve hızın artması olaya yeni boyutlar kazandırmıştır.

Bir etkileşimli yaklaşım üç temel adımla karakterize edilebilir [5]:

(1)- Bazı başlangıç parametrelerine göre problem çözülerek uygun, baskın çözüm bulunur.

(2)- Çözüm(ler) karar vericiye sunularak tepkisi, kararı istenir.

(3)-Karar vericinin cevabına göre, ya yeni parametre kümesiyle yeni problem kurularak ikinci adıma dönülür, ya da çözüm bulunmuştur, durulur.

Etkileşimli algoritmalar bölüşüm (trade-off) bilgisinin açık, kapalı alınışına göre sınıflandırıldığı [1 , 4] gibi, etkin küme üzerinde baskın çözümleri elde etme yollarına göre de sınıflandırılmaktadır [36].

izleyen kısımlarda, uygun çözüm alanını, ağırlıklandırma uzayını, kriter koniyi daraltma, çizgisel araştırma, ideal çözümden sapmayı minimum yapan, ölçekleme fonksiyonu kullanan algoritmalarından örnekler verilecektir.

2.2 Step Yöntemi (STEM)

Çok amaçlı doğrusal programlama problemleri için R. Benayoun, J. de Montgolfier ve arkadaşları tarafından geliştirilen ilk etkileşimli yöntemdir[1,32]. Bu tip problemlerde genellikle "Optimum" kavramı yerine çoğunlukla "eniye-uzlaşık" düşüncesinden hareket edilmiştir. Amaç fonksiyonunun öncelikli ağırlıklı toplamları yerine ardışık adımlarla sonuca ulaşılacak düşünülmüştür. Algoritma bu arayış sırasında karar vericiye çözümle ilgili sorular yöneltip tercihlerini alır. Bu tercihlere göre ya döngü başına dönüp arama devam ettirilir ya da işlem sonlandırılır. Böylece etkileşimli insan-model ilişkisi, hesaplama ve karar aşamalarıyla sürer. Bu arayış sırasında karar verici problemin çözüm kümesini ve amaçlarının göreceli önemini de öğrenir.

Genellikle tüm amaçları verilen kısıtlar üzerinden birarada eniyileyen süper veya ideal çözüm yoktur. STEM süper veya ideal çözüm yerine eniye-uzlaşık çözümü, etkin çözüm bölgesini daraltarak gerçekleştirir.

Etkin kümenin daraltılması işlemi, ardışıklarda $X \in R$ ideal kriter vektörüne ağırlıklı L_∞ (Tchebycheff) uzaklık ölçümüne göre en yakın nokta hesaplanarak yapılır. Başta söylendiği gibi ÇADP için geliştirilen bu yöntem doğrusal olmayan ve tamsayılı çok amaçlı programlama problemlerine de uygulanabilir [36].

2.2.1 Ödenti (Pay-off) Matrisi

Çok amaçlı optimizasyonda anahtar kavramlardan biri ödenti matrisidir [56].

$$\text{Enb } \{C^j X = Z_j\} \quad j = 1, 2, \dots, k \quad (2.1)$$

$$\text{Kısıtlar: } AX \leq b \quad X \geq 0$$

Problemin ideal çözümleri Z^* olmak üzere, J . satır x çözüm vektörüne karşılık f amaç fonksiyonunun alacağı değerlere karşılık gelir. Böylece tablo,

Tablo 2. Ödenti Tablosu

	Z_1	Z_2	...	Z_j	...	Z_k
1 Z	Z_{11}	Z_{12}	...	Z_{1j}	...	Z_{1k}
2 Z	Z_{21}	Z_{22}	...	Z_{2j}	...	Z_{2k}
...	...	...	...	...	...	...
k Z	Z_{k1}	Z_{k2}	...	Z_{kj}	...	Z_{k}

Ödenti matrisi eldeki karar probleminin çatışan amaçlarının değerli bir aletidir. Bu matristen iki önemli vektör türetilir.

$$Z = (Z_1^*, \dots, Z_k^*) \quad (2.2)$$

İdeal (utopia) vektörü ve

$$n = (n_1^*, \dots, n_k^*) \quad (2.3)$$

en büyük kriter değerleri (nadir) vektörüdür. Burada n_j değerleri

$$n_j^* = \max_j z_j \quad (2.4)$$

ilişkisiyle bulunur.

Değişik çözüm vektörlerinin aynı z^* değerlerini üretebileceğinden ödenti matrisi ve nadir vektörü her zaman tek değildir [56].

2.2.2 Stem Algoritması

1. Adım : Herbir amaç fonksiyonunu kısıtlar üzerinden enbüyükleyen etkin çözüm kümeleri bulunur. Bu çözüm diğer amaç fonksiyonlarında yerlerine konarak 2.2.1 de verilen ödenti tablosu oluşturulur. Ödenti tablosunun j . satır elemanları x_j çözüm vektörüne karşılık amaç fonksiyonundan elde edilen değerlerdir. Tablonun esas köşegeninde yer alan z_j^* değerleri ise ideal çözüm vektörünü oluşturur. Karar verici ideal çözüm vektörüne yakın bir çözüm aramaktadır.

2. Adım : Artıdırma adım sayacı $h = 0$ ve M_i ödenti tablosunun i . sütunun en küçük değeri olsun.

n_i değerleri,

$$n_i = \begin{cases} \begin{array}{c|c|c} z_i^* - M_i & n_i^2 & -1/2 \\ \hline \Sigma(C_{ij}) & & \\ j=1 & & \end{array} & z_i^* > 0 \text{ ise} \\ \begin{array}{c|c|c} M_i - z_i^* & n_i^2 & -1/2 \\ \hline \Sigma(C_{ij}) & & \\ j=1 & & \end{array} & z_i^* \leq 0 \text{ ise} \end{cases} \quad (2.5)$$

Terim 1 Terim 2

hesaplanır.

Burada 1.terimin fonksiyonu, enbüyük görelî uzaklı amaçlara daha fazla ağırlık vermektir. 2. terim ise L uzaklık ölçümüne göre amaç fonksiyonu gradiyentini normalleştirir.

(1) *
3. Adım : S = S ve J = ∅ olsun.

S(1) = S ile başlamak demek henüz daraltılmamış orjinal uygun çözüm bölgesiyle başlamak demektir. J gevşetilecek çözüm kümesi elemanlarının indislerinden oluşur. Başlangıçta daraltılacak küme boş olduğundan J kümesi de boşdur.

(h)
4. Adım : Adım sayacı h = h + 1 olsun W enkenb (minimax) ağırlıkları

$$W^{(h)} = \begin{cases} 0 & i \in J \\ \frac{n_i}{\sum_{j=1}^k n_j} & i \in J \end{cases} \quad (2.6)$$

göre hesaplanır.

Bu ağırlıklar $\| \bar{z} = \bar{z} \|_{\infty} = \text{enb} \{ W_i^{(h)} | \bar{z}_i = \bar{z}_i \}$ (2.7)

(2.7) Eşitliği 5. adımda h. ardışım çözümünü üretmek üzere ağırlıklı Tchebycheff ölçümüdür.

(h)
ilk ardıştırmada W ağırlıkları toplamı birdir. Sonraki ardıştırmalarda J indis kümesi boş olmadığından W ağırlıkları toplamı birden küçüktür.

5. Adım : Ağırlıklı enkenb (minimax) programlaması,

enk { α }

$$\text{Kısıtlar : } \alpha \geq W_i^{(h)} (Z_i - C X) \quad 1 \leq i \leq k \quad (2.8)$$

$$X \in S$$

$$0 \leq \alpha \in R$$

çözülür.

(h) k
Bu adımda $W^{(h)}$ R ile tanımlı ağırlıklı Tchebycheff metriğine

(h) *
uygun $S^{(h)}$ daraltılmış uygun çözüm bölgesinde kriter vektörü $Z^{(h)}$ ideal çözümüne en yakın nokta bulunur.

(h) (h) (h) *
6. Adım : $Z^{(h)} = Z(X^{(h)})$ olsun. $Z^{(h)}$ ile Z^* karşılaştırılır.

Bu karşılaştırmaların amacı elde edilen kriter vektörünün Z^* ideal vektörüne göre durumunu belirlemektir.

(h)
7. Adım : $Z^{(h)}$ kriter vektörünün tüm elemanları karar vericiyi sağlıyorsa, dur. Son çözüm $(Z^{(h)}, X^{(h)})$ en iyi uzlaşık çözümdür. Aksi halde 8. adımla devam edilir.

Kriter vektörünün bazı elemanları karar vericinin sağlama düzeylerine ulaşmamışsa ardıştırmaya devam edilir. Çünkü ödünleşim (değiş-tokuş) yaparak eniyi uzlaşık çözüm kümesi elde edilebilir. Eğer karar verici ödünleşim yapmaya isteksiz ise algoritmadan çıkılır.

*
8. Adım : J gevşetilecek kriter değerlerinin indis kümesi ve bu küme elemanlarının gevşeme miktarları λ_j azaltma miktarları bu adımda belirlenir.

9. Adım : Uygun çözüm bölgesini

$$S^{(h+1)} = \left\{ X \in S \mid \begin{array}{l} EX \geq Z_j(X^{(h)}) - \lambda_j \quad j \in J^* \\ EX \geq Z_j(X^{(h)}) \quad j \in J \end{array} \right\} \quad (2.9)$$

ilişkisine göre daralt ve 4. adıma dön.

Ardıştırmayla ek kısıtlar ilave ederek S kümesinin daha alt kümelerine ilerlenir. Benayoun ve ark. j . kriterin bir defa

gevşetildikten sonra gelecek arđıřtırmalarda dođru deđer veremeyebileceđi nedeniyle algoritmanın en çok amaç sayısı kadar çalıştırılmasını salık verir [36].

Stem yönteminin mantığını kullanarak geliştirilen birçok etkileşimli yöntemi yayınlarda görmek mümkündür. Fichet, Hedef (goal) programlanmasının temel özellikleriyle stem algoritmasını birleştirerek Gpstem yöntemini geliřtirmiřtir [57]. Bu algoritmanın Stem yöntemine üstünlüğü "sađlayan üzlaşık çözüme" daha az arđıřtırmayla ulařılabilmesidir [1].

Choo ve Atkins, Step yöntemine benzer bir algoritmayı etkileşimli kesirli doğrusal programla problemlerini çözmek üzere kullanmışlardır [58].

Nijkamp ile Spronk [59] ve Spronk ile Telgen [60] hedef programlaması ve Step yönteminden yararlanarak etkileşimli çok amaçlı programlama ve etkileşimli elipsoid çok amaçlı programlama algoritmalarını oluşturmuşlardır.

Stem, uygun yazılım (software) ile doğrusal olmayan ve tamsayılı programlamaya da uygulanabileceđi belirtilmiştir [36].

2.3 GDF Yöntemi

1970 yılında Geoffrion [61] tarafından geliştirilen bu yöntem 1972'de Geoffrion, Dyer, Feinberg [33] tarafından genel ÇAKV problemlerini çözmek için yeniden düzenlendi. GDF algoritması "çizgisel araştırma" ve subjektif farksız bölüşüm kavramı üzerine kurulmuřtur. Varsayımlarından biri amaçlar üzerinde karar vericinin tercihini yansıtan fayda fonksiyonunun

varlığıdır. Enbüyükleme problemi,

$$\text{enb} \quad U(f(x), f(x), \dots, f(x)) \quad (2.10)$$

1 2 k

Kısıtlar: $X \in S$

Burada $S = \{ X \in R^n \mid g_i(x) \leq 0, i=1,2,\dots,n \}$ biçiminde ifade edilebilir. Bu yöntem için temel diğer varsayımlar ise,

- i- S doğrusal eşitsizliklerle tanımlanmış sınırlı, tıkkız,
- ii- Tüm fonksiyonlar diferensiyellenebilir,
- iii- Fayda fonksiyonu uygun çözüm bölgesinde dışbükey, olmalıdır.

GDF yöntemi çizgisel araştırma için hızlı yakınsaması nedeniyle doğrusal olmayan programlama tekniklerinden Frank-Wolfe algoritmasını kullanır.

2.3.1 Hızlı Dikcıkış Algoritması

Frank-Wolfe algoritmasının hızlı yakınsamasını kavramak için önce hızlı dikcıkış algoritmasının optimum araştırma biçimini görelim. Burada $F: R^n \rightarrow R$, matematiksel olarak bilinen, türevlenebilir bir fonksiyondur.

Algoritmanın adımları:

1. Adım : Bir başlangıç noktası $x^{(0)} \in S$ ve adım sayacı $h=0$ olsun.

2. Adım : Hareket yönü,

$$d^{(h+1)} = - \nabla_x F(x^{(h)}) \quad (2.11)$$

belirlenir. Burada (2.11), F fonksiyonunun $x^{(h)}$ noktasında gradiyentidir.

3. Adım : Belirlenen $d^{(h+1)}$ yönünde çizgisel araştırma için

$$\text{enb} \quad (F(x^{(h)} + td^{(h+1)}))$$

$$\text{Kısıtlar} \quad (x^{(h)} + td^{(h+1)}) \in S \quad (2.12)$$

$$0 \leq t \in R$$

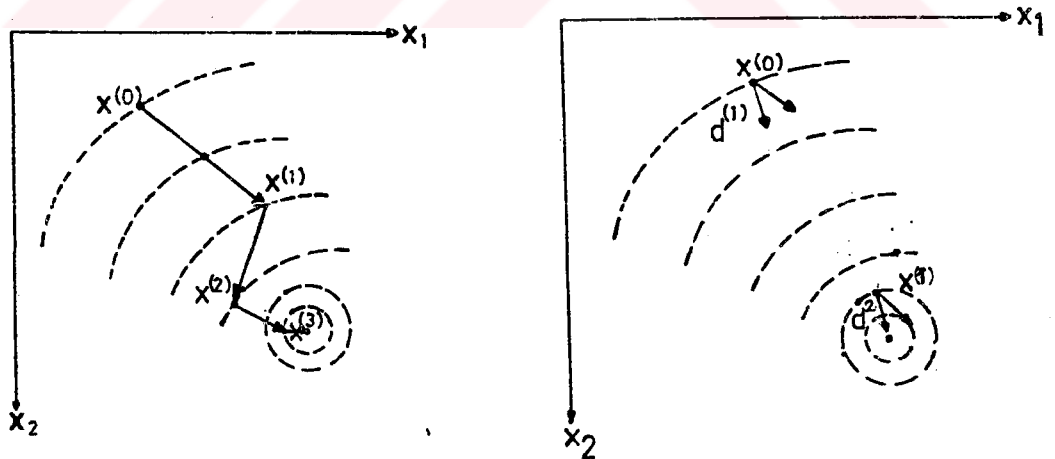
adım büyüklüğü problemi çözülür. t^* optimal adım büyüklüğü ile

$$x^{(h+1)} = x^{(h)} + t^* d^{(h+1)} \quad (2.13)$$

bulunur.

4. Adım : Son iki adımda elde edilen ilerleme test durdurma parametresinden küçük ise ya da karar verici araştırmayı sürdürmek istenmezse durulur. Aksi halde adım sayacı $h=h+1$ alınarak 2. Adıma döndülür.

Dik çıkış algoritması aşağıdaki şekilde görüleceği gibi çok inis çıkış yapar. F fonksiyonunun düzey eğrilerinin kesikli çizgilerle gösterildiği grafikte $x^{(0)}$ başlangıç noktasından optimuma ulaşmak için fazla sayıda adım atmak gereklidir.



Sekil 4. Dikcıkis ve Frank-Wolfe algoritmalarının işleyişi

Frank-Wolfe algoritması başka çizgisel araştırma tekniği kullanarak optimuma daha az sayıda adımla ulaşmayı gerçekleştirir.

2.3.2 Frank-Wolfe Algoritması

$F: \mathbb{R}^n \rightarrow \mathbb{R}$ matematiksel olarak bilinen, türevlenebilir bir fonksiyon olmak üzere, F-W algoritması aşağıdaki biçimde çalışır:

1. Adım : Başlangıç noktası $x^{(0)} \in S$ ve adım sayacı $h=0$ olsun.

2. Adım : Yön bulma programı,

$$\text{enb } \left\{ \left[\nabla_x F(x^{(h)}) \right]^T x \right\}$$

Kısıtlar: $x^{(h+1)} \in S$ çözülür. $y^{(h+1)} \in S$ bir optimum çözüm ile hareket yönü $d^{(h+1)} = y^{(h+1)} - x^{(h)}$ bulunur.

3. Adım : Belirlenen $d^{(h+1)}$ yönünde çizgisel araştırma için adım büyüklüğü programı,

$$\begin{aligned} \text{enb } \{ F(x^{(h)} + t d^{(h+1)}) \} & \quad (2.15) \\ 0 \leq t \leq 1 \end{aligned}$$

*
çözülür. t optimum adım büyüklüğü ile $x^{(h+1)} = x^{(h)} + t d^{(h+1)}$ elde edilir.

4. Adım : Uygun bir durdurma kuralına göre durulur, ya da adım sayacı $h=h+1$ alınarak 2. adıma dönülür.

F-W algoritması için durdurma kriterleri [62],

$$\left| \frac{F(x^{(h+1)}) - F(x^{(h)})}{\text{enb } \{ \epsilon_1, |f(x^{(h+1)})|, |f(x^{(h)})| \}} \right| \leq \epsilon_2 \quad (2.16)$$

veya

$$\left| \frac{\begin{matrix} (h+1) & (h) \\ x & - x \end{matrix}}{\text{enb } \{ \epsilon_3, \| x^{(h+1)} \|, \| x^{(h)} \| \}} \right| \leq \epsilon_4 \quad (2.17)$$

seçilebilir. Burada $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ önceden belirlenen sayılardır. Bu kriterler ile $f(x)$ ve x 'deki göreceli değişimler ölçülür.

Dik çıkış ve F-W algoritmalarının ikisinde de belli noktalarda F fonksiyonunun gradiyenti kullanıldı. Dik çıkış algoritmasında gradiyent araştırma yönünü belirlerken F-W algoritmasında yön bulma programının gradiyenti olarak kullanılır.

Şekil 4.' de görüldüğü gibi başlangıç noktasında yön bulma

programı ile $y^{(1)}, y^{(1)}$ kullanarak $d^{(1)}$ yönü $\nabla x F(x^{(0)})$ gradiyent yönünden farklı $y^{(1)}$ bulunur. Adım büyüklüğü programı ile $x^{(1)}$ noktasına atlanır. Şekilde görüldüğü gibi az sayıda adımla optimuma yaklaşılar.

Frank-Wolfe algoritması doğrusal olmayan programlamada kullanılan bir tekniktir. Çok amaçlı programlamada, F fonksiyonunun yerini gerçekte bilinmeyen $[1,36,63]$ U, fayda fonksiyonu alır. GDF algoritması karar vericinin tercihlerini ardışık biçimde tahmin etmede, yön bulma programında U fayda fonksiyonun yerel gradiyent yönünü kullanır.

2.3.3 Uygun Ağırlıklar Üretme Algoritması

GDF algoritması fayda fonksiyonunda karar vericinin tercihlerini yansıtan ağırlıkları elde etmek için ikili karşılaştırmalar yapar. İlk amacı referans seçip i . amacın ağırlığını hesaplamada $\frac{f_i}{f_1}$ oranı kullanılır. Burada f_i , i öteki amaçların değerlerine karışmaksızın i . amactaki artmaya

karşılık ilk amactaki $\wedge$ kadar azalmayı telafi etme miktarıdır.
 İkili karşılaştırmaların amacı verilen bir $\wedge_1$ için $\wedge_i$ miktarını
 ve buradan

$$W = \frac{\wedge_i}{\wedge_1} \quad (2.18)$$

tercih ağırlıklarını hesaplamaktır. Bilinmeyen $\wedge_i$ değerlerini
 hesaplamak bölme ve katlama algoritması ile mümkündür:

1.Adım : $Z = (Z_1, Z_2, \dots, Z_i, \dots, Z_n)$ kriter vektörü $i=2$ ve
 $W = 1$ olsun. $\wedge_1$ belirlenir.

2.Adım : $l = 0$ olsun ve $\wedge_i^{(1)}$ belirlenir.

3.Adım : $l = l + 1$ ve $Z = (Z_1 - \wedge_i^{(1)}, \dots, Z_i + \wedge_i^{(1)}, \dots, Z_k)$
 Z^a ve Z^b vektörleri karşılaştırılır.

4.Adım : Üç durum söz konusu olabilir:

i- Eger Z^a tercih edilirse Z^b vektörünün tercih edilebilir-
 lüğünü arttırmak için $\wedge_i^{(1+1)} = 2 \wedge_i^{(1)}$ al, 3. Adıma dön.

ii- Eger Z^b tercih edilirse $\theta = \wedge_i^{(1)} / 2$ al 6. Adıma git.

iii-Eger karar verici Z^a ve Z^b arasında bir tercih yapamazsa
 8. adıma git.

5. Adım : $\theta^{(1+1)} = \theta^{(1)} / 2$ ve $l = l+1$ olsun. Z^a ve Z^b
 vektörlerini karşılaştıır,

i- Eger Z^b tercih edilirse 6. adıma git.

ii- " Z^a " " 7. adıma git.

iii- Kararsız ise 8. adımdan devam et.

6. Adım : Z^b vektörünün tercih edilebilirliğini azaltmak
 için,

$$\begin{matrix} (1+1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix} = \begin{matrix} (1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix} - \theta \quad /2 \text{ al ve 5. adıma dön.}$$

7. Adım : Z^b vektörünün tercih edilibilirliğini arttırmak için ,

$$\begin{matrix} (1+1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix} = \begin{matrix} (1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix} + \theta \quad /2 \text{ olsun ve 5.adıma dön.}$$

8. Adım : i . amacın ağırlığı $W_i = \begin{matrix} (1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix} / \begin{matrix} (1) \\ \backslash \end{matrix} \begin{matrix} (1) \\ i \end{matrix}$ olsun.

9. Adım : Eğer $i=h$ ise $i=i+1$ al ve 2.adıma dön.

10.Adım : Dur [36].

2.3.4 GDF Algoritması

1. Adım : Başlangıç noktası $x^{(0)} \in S$ ve adım sayacı $h = 0$ alınır.Uygun çözüm bölgesi S doğrusal eşitsizliklerle tanımlandığında, başlangıç noktası DP ile elde edilebilir. Amaç fonksiyonları da doğrusal ise karar vericiden amaçların sağlama düzeyleri (Z) alınarak,

$$\begin{aligned} \text{enb } \{ & 0 \ x \} \\ & i \\ \text{Kısıtlar : } & C \ x = Z \quad i=1,2, \dots, k \quad (2.19) \\ & x \in S \end{aligned}$$

tek amaçlı DP problemi çözülür. Eğer uygun bir çözüm varsa $x^{(0)}$ başlangıç noktası olarak kullanılabilir.

2. Adım : U fayda forksiyonunun x^h noktasında gradiyent yönünü belirlemek için karar verici, ağırlık vektörünü ya doğrudan belirler ya da bir dizi karşılaştırmalarla (bir önceki kısımda verildiği biçimde) uygun ağırlıklar belirlenir.

3. Adım : i . amacın x^h noktasında gradiyenti,

$$\begin{aligned} \backslash / f \quad & (x^h) \text{ ve } g = \sum_{i=1}^k W_i \backslash / f(x^h) \quad (2.20) \\ x \quad & i \end{aligned}$$

olmak üzere

$$\text{enb } \{ g^T x \} \quad (2.21)$$

$$x \in S$$

yön bulma programı çözülür, $y^{(h+1)} \in S$ elde edilir.

$$\text{Yön } d^{(h+1)} = y^{(h+1)} - x^{(h)} \text{ eşitliğiyle bulunur.}$$

2. adımda amaç fonksiyonları için uygun ağırlıklar bulunmuş ise fayda fonksiyonunun $x^{(h)}$ noktasındaki gradiyent yönü ile kabaca aynı yönde olacaktır.

4. Adım : Eğer $y^{(h+1)} = x^{(h)}$ ise $x^{(h+1)} = x^{(h)}$ alınıp 10. adıma gidilir. Bulunulan çözümden herhangi uygun yönde fayda fonksiyonu değişmezse optimum çözüm bulunmuştur. Bu ise optimum için birinci derecede gerekli şartların sağlanması demektir [5]. Fayda fonksiyonunun içbükey olması optimum olmayı garanti eder. Pratikte $\| x^{(h+1)} - x^{(h)} \|$ değeri belli bir ϵ sayısından küçük olmasıyla eşitlik sağlanmış kabul edilir.

5. Adım : Kesim sayısı $P > 1$ olmak üzere belirlenir. Böylece her ardıştırmada kaç kriter vektörü hazırlanacağı da belirlenmiş olur.

6. Adım : Kesim noktalarında P tane kriter vektörü bulunur. (h) $d^{(h)}$ yönünde çizgisel araştırma düzenlemek üzere (h) $(h+1)$ $Y(x^{(h)}, y^{(h+1)})$ doğru parçası boyunca eşit aralıklı kesim noktalarında kriter vektörleri hesaplanır.

$$\text{Bunlar } P-2 \text{ adet } z(x^{(h)} + \frac{j-1}{P-1} d^{(h+1)}) \quad j=2, \dots, P-1 \quad (2.22)$$

vektörleri ile ilk kriter vektörü $z(x^{(h)})$ ve $z(y^{(h+1)})$ son kriter vektörüdür.

7. Adım : Karar verici kesim noktalarındaki kriter vektörlerinden tercih ettiği $(z^{(h+1)}, x^{(h+1)})$ çiftini belirler.

8. Adım : Ardıştırma sürdürülecekse $h=h+1$ alınarak 2. adıma dönülür.

9. Adım : Çözüm vektörü $X^{(h+1)}$ ve amaç değerlerinden oluşan $Z^{(h+1)}$ ile işlem sonlanır.

GDF algoritması, S doğrusal olmayan kısıtlarla tanımlandığında da, doğrusal olmayan bir yazılım ile kullanılabilir [36].

Wallenius [53] farksız bölüşümlerin veya marjinal yerdeğiştirme oranlarının hesaplanmasının zorluğunu belirtmiştir. Dyer [64] problemin bu zorluğunu azaltmak amacıyla bir algoritma geliştirmiştir.

Openheimer, [65] karar vericinin öncel tercih bilgisine dayanarak daha doğru fayda fonksiyonları kurulacağını ileri sürmektedir.

2.4 Etkileşimli Uzlaşık Programlama

Bazı ÇAKV problemlerinde , özellikle özel sektörde, ağırlıklar, öncelikler, hedefler ve idealler karar vermede alışılmış yollar olarak sürekli kullanılır. Karar verici tercihlerini bu terimlerle daha rahat verebilir.

Eniyi kavramı, her amacın gerçekleşme değeriyle, istenen hedef ve ideallerden sapma derecesine göre tanımlanır. Daha açık olarak, eniyi-uzlaşık alternatif genellikle istenen hedef ve amaçlardan minumum sapmalı olandır [31].

Minumum sapmayı belirleyecek uzaklık ölçümü tek değildir. Genellikle matematiksel, ağırlıklı normlardan biri kullanılır:

$$d_p(f(x), f^*) = \left(\sum_{j=1}^k W_j [f_j(x) - f_j^*]^p \right)^{1/p} \quad (2.23)$$

$p=1,2, \dots$

f^* : ideal veya hayali (utopia) çözüm vektörü,

W_j : j. amaca verilen öncelik ya da ağırlık,

$d(.)$: $f(x)$ ve f^* arasındaki uzaklık.

Buna göre ÇAKV problemi,

$$\text{enk } \left\{ d_p(f(x), f^*) = \left(\sum_{j=1}^k W_j [f_j(x) - f_j^*]^p \right)^{1/p} \right\} \quad (2.24)$$

Kısıtlar : $x \in S$

Minumum sapmayı kullanabilmek için üç elemanın belirlenmesi gereklidir:

- i- ideal vektör f^*
- ii- Öncelik yapısının ağırlıkları
- iii- Sapma ölçümünün (P) seçimi.

İlk iki elemanın anlamlı biçimde belirlenmesini karar vericiden bekleyebiliriz. Ancak uygun P ve böylece sapma ölçümü uzaklık kavramına bağlı olarak tümüyle açık değildir [4].

Uzaklık yalnız, sırf geometrik anlamda değil, karar vericinin tercihi için bir vekil ölçüm olarak da kullanılır. Uzaklığı tek tek koordinatlara, boyutlara ve niteliklere göre yakınlık veya benzerliğin bir ölçümü diye tanımlayabiliriz. Bu nedenle bakış açımızı bir p ile daraltamayız [31].

Herhangi iki nokta $((5,4), (2,2))$ arasında, değişen p değerlerine göre d_p uzaklıkları, ağırlıklar eşit ve bir alınarak aşağıdaki tabloda verilmektedir.

Tablo 3. Değişik P değerleri için d uzaklıkları

p	1 1 p (X - X) 1 2	2 2 p (X - X) 1 2	Toplam	d p
1	3	2	5	5
1.2	3.73	2.29	6.02	4.46
1.6	5.79	3.03	8.82	3.89
2	9	4	13	3.61
5	243	32	275	3.10
10	59049	1024	60073	3.01
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
∞	3 ∞	2 ∞	∞	3

Tabloda görüldüğü gibi, toplamı oluşturan sapmaların büyüğü daha fazla etkilenmektedir. Sonunda $p=\infty$ için sapmaların büyüğü uzaklık hesaplamalarını tamamen aşar. $p=1,2$ ve ∞ değerlerinin niçin stratejik öneme sahip olduğu görülebilir. $p=1$ geometrik anlamda iki nokta arasındaki uzaklığın en uzununu verir. Bu d ölçümüne ekseriya "city blok" ya da "Manhattan blok" uzaklık ölçümü¹ denir. Herhangi iki nokta arasındaki en kısa yol düzgün doğrudur ve bu $p=2$ ile elde edilir. $P>2$ düzgün doğrudan daha kısa uzaklık ölçümlerini vermektedir.

Bazı durumlarda uzaklık, seçilen ölçü biriminden etkilenebilir. Bu gibi istenmeyen durumları önlemek üzere göreceli uzaklık ölçümü,

$$d_p(f(x), f) = \left[\sum_{j=1}^k \left(\frac{f_j - f_j^*}{f_j} \right)^p \right]^{1/p} \quad (2.25)$$

alınabilir.

2.4.1 En-iyi Uzlaşık Çözümler

2.24 Problemi veya $1 \leq p < \infty$ için esiti,

$$\min_{x \in S} \left(d_p(f(x), f^*) \right) = \sum_{j=1}^p w_j (f_j(x) - f_j^*)^p \quad (2.26)$$

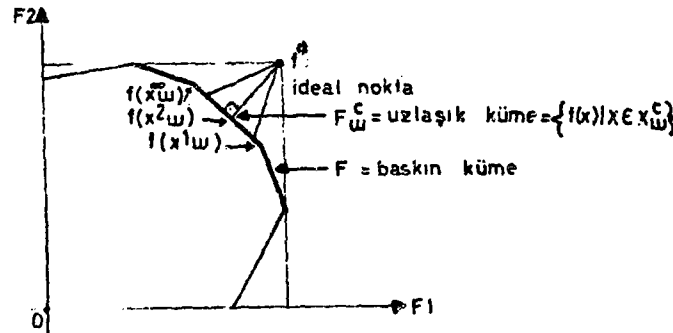
$$x \in S$$

problemini çözen X_w^p uzlaşık bir çözümdür. Problemlerden birinin çözümü diğerinin de çözümüdür [5]. Verilen w ağırlık vektörü ve $1 \leq p < \infty$ için tüm uzlaşık çözümlerin kümesi X_w^c olsun.

X_w^c kümesi açık biçimde,

$$X_w^c = \{ x \in S \mid x \text{ 2.24 veya 2.26 problemlerini verilen } w \text{ ve bazı } 1 \leq p < \infty \text{ için çözer} \} \quad (2.27)$$

X_w^p çözümü, herhangi $1 \leq p < \infty$ ve $w > 0$ ise her zaman baskın bir çözümdür. Dinkelbach ve Dürr X_w^p çözümünün baskın olduğunu göstermiştir [5]. Baskın çözümler kümesinin bir alt kümesi olan uzlaşık çözümler kümesi (X_w^c) , X_w^p çözümünü içerirse baskındır.



Şekil 5. İdeal çözüm ve uzlaşık küme

Şekil 5' de enbüyükleme problemi için uzlaşık küme grafikte gösterilmektedir. Tüm amaçlara eşit ağırlık verildiğinde, yani $w = (1/n, \dots, 1/n)$ ise X_w^c ile ideal çözüm

arasındaki geometrik uzaklık enküçüktür. Doğrusal durumda, X^1 ve X^0 biliniyorsa X^C uzlaşık çözüm kümesi tamamen belirlidir. Gerçekten iki amaçlı problemde $w > 0$ ise, X^C , baskın çözümler kümesi üzerinde, X^1 ve X^0 arasında uzanan noktaların kümesidir. X^1 ve X^0 aynı baskın kenar üzerinde ise

$$X^C = \{ X | X = \alpha X^1 + (1-\alpha) X^0 \}$$

olacaktır.

Zeleny üç veya daha çok amaçlı problemlerde X^C uzlaşık kümeye aşağıda verilen iki-amaçlı problemin baskın çözümleriyle yaklaşılabileceğini salık vermektedir [74].

$$\min_{X \in S} [d(f(x), f^*), d_0(f(x), f^*)] \quad (2.28)$$

X^C uzlaşık kümesi, baskın çözümler kümesinin bir azaltılmış kümesidir. X^C kümesi elde edildiğinde, bu kümeden eniyi-uzlaşık çözümün seçilebilmesi bilinen tercih yapısını gerektirir. Eğer X^C , eniyi uzlaşık çözümü seçmek için, hala çok büyük ise, daha da daraltılabilir.

Zeleny ve Steuer bu amaçla, iki ayrı yol önermektedir. Birincisinde, karar vericiden doğrudan, X^C kümesinden, kendisine basılgın gelen çözümlerin atılması istenmektedir. İkincisinde ise, w_j ağırlıkları daralan $[a_j, b_j]$ aralıklarından seçilerek, X^C kümesi eniyi uzlaşık çözümü kapsayınca kadar azaltılmaktadır. Karar vericiden gerekli bilgi kolayca temin edilemezse, aşağıda verilen azaltma işlemi, analist tarafından uygulanabilir.

İlkinde, X^C kümesi, f^* ideal noktadan uzaklığı enküçük yapar iken, benzer biçimde X^C uzlaşık kümesi anti-ideal (enaz istenen)

noktadan (f^0) uzaklığı enbüyük yapar. Daha açık ifade edilirse,

$$\bar{X}_w^c = \{ X | X \text{ bazı } 1 \leq p < \infty \text{ için} \\ \text{enb } \sum_{j=1}^k (W_j [f_j^0 - f_j(X)]^p)^{1/p} \quad X \in S, \text{ pb çözer.} \} \quad (2.29)$$

Burada $f_j^0 = \text{enb } f_j(X)$.

$$X \in S$$

$X \cap \bar{X}_w^c$ kesişimi uygulanabilir ve bu daraltılmış uzlaşık kümeyi verecektir.

Zeleny tarafından önerilen ikinci yöntem ise değişen, hareket eden (displaced) ideal kavramına dayanır. Ideal nokta (f^*) , S kümesine bağlıdır. Verilen algoritmalarla S kümesi, $\bar{X}_w^c$ uzlaşık daraltılmış kümeye indirgendiğinde, bu yeni kümeye göre belirlenen ideal nokta öncekinin yerini alır.

$$\begin{aligned} &\text{Yeni ideal nokta } f^*, \\ &\bar{f}_j^* = \text{enk } f_j(X) \\ &X \in \bar{X}_w^c \end{aligned} \quad (2.30)$$

ilişkisinden hesaplanır. Yeni ideal nokta kullanılarak yeni $\bar{X}_w^c$ uzlaşık kümesi, $\bar{X}_w^c$ kümesine benzer biçimde kurulur.

$$\begin{aligned} \bar{X}_w^c &= \{ X | X, \text{enk } \sum_{j=1}^k W_j [f_j^* - f_j(X)]^p, \quad 1 \leq p < \infty \text{ veya} \\ &\text{enk enb } (W_j [f_j^* - f_j(X)]^p), \quad 1 \leq p < \infty \quad X \in \bar{X}_w^c \end{aligned} \quad (2.31)$$

problemlerinden birini çözer}

2.4.2 Kayan Ideal Nokta Yöntemi Adımları

1. Adım : Karar vericiden W ağırlık vektörünü al. $\bar{X} = S$ olsun.

$$2. \text{ Adım : } \min_j f_j(x) \quad j=1,2,\dots,k$$

(2.32)

$$\text{Kısıtlar: } x \in \bar{X}$$

problemini çözerek f^* ideal nokta bulunur. Ağırlıklar $W = W_j$

f^* ifadesiyle ölçeklenir.

$$3. \text{ Adım : } \min_{x \in \bar{X}} (d(f(x), f^*), d(w(f(x), f^*))) \quad (2.33)$$

problemin baskın çözümlerini bularak X^C uzlaşık kümesi oluşturur.

4. Adım : Karar verici X^C uzlaşık kümeden eniyi-uzlaşık çözümü seçebilir ise ya da X^C kümesi f^* ideal noktayı kapsarsa durulur; eniyi-uzlaşık çözüm bulunmuştur. Aksi taktirde 5. adımla devam edilir.

5. Adım : $\bar{X} = X^C$ alınarak yeni ideal noktayı bulmak için X^C uzlaşık kümesi daraltılır. 2. Adıma dönülür.

Zeleny'nin bu yöntemi, doğrusal problemler olduğu kadar genel ÇAKV problemlerine de uygulanabilir.

2.4.3 Etkileşimli Uzlaşık Programlama

Önemli karar problemlerinde, uzlaşık çözümü yalnız bir kez uygulayarak optimal çözüme ulaşılmayabilir. Bu nedenle aşağıdaki etkileşimli yöntem Yu[7] tarafından geliştirilmiştir.

1. Adım : S , f^* ve f belirleme, dikkatli gözlem ve karar vericiyle görüşmek gerektirir. Hedef (ideal) noktası f^* , w ağırlık vektörü ve l -uzaklık ölçümü için p değerlerini kesin biçimde belirlemek daha zordur. Bununla birlikte ideal nokta, w vekil ağırlıklar ve ölçüm için $p = 1,2, \dots$ değerleri seçilebilir.

2. Adım : Belirlenmiş f^* , w ve p değerleriyle uzlaşık programlama çözümleri bulunur.

Bu adımla kurulan uzlaşık programlama problemleri kısım (2.4.3.1)' de verilmektedir.

3. Adım : 1 ve 2 adımlarda elde edilen tüm bilgiler karar vericiye sunulur. Bunlar S, f_1, f_2 ve y_1, y_2, y_3 uzlaşık çözümleri ile ağırlıklıdır.

4. Adım : 3. Adımda optimal çözüm elde edilmiş ve karar verici sağlanmış ise 6. adıma git.

5. Adım : Karar vericiyle konuşarak S, f, y , ağırlık vektörü(w) ve p ile ilgili yeni bilgiler alınır. Zaman içinde, çözümler görüldükçe, bu beş eleman değişebilir. Analist, karar vericiye uygun sorular sorarak problem elemanlarında değişme ve azaltma yapılmasını sağlar. 2. Adıma dönülür.

6. Adım : y uzlaşık çözümü bu çözümün asıl değişken değerleri p eniyi-uzlaşık çözümdür. Durulur.

2.4.3.1 y Uzlaşık Programlama Problemleri

Cokkamaçlı karar problemi,

$$\begin{aligned} \text{enb } f_1(x) &= Z_1 \\ \text{enb } f_2(x) &= Z_2 \\ &\vdots \\ \text{enb } f_k(x) &= Z_k \end{aligned} \quad (2.34)$$

$$\begin{aligned} \text{Kısıtlar : } x \in S \quad S &= \{ x \mid Ax = b, x \geq 0 \} \\ \text{ve ideal çözüm } f(x) &= \text{enb } f_j \quad j=1,2,\dots,k. \end{aligned} \quad (2.35)$$

$$x \in S$$

$$\begin{aligned} \text{i- } y \text{ Uzlaşık programlama problemi,} \\ \text{enk } \{ [f_1 - f_1(x)] + [f_2 - f_2(x)] + \dots + [f_k - f_k(x)] \} \end{aligned} \quad (2.36)$$

Kısıtlar : $X \in S$

Tek amaçlı DP problemidir.

ii- y Uzlaşık programlama problemi bir Quadratik programlama problemidir.

$$\text{enk } \left\{ \sum_{i=1}^k [f_i - f_i(x)]^2 + \dots + \sum_{k=1}^k [f_k - f_k(x)]^2 \right\} \quad (2.37)$$

Kısıtlar : $X \in S$

Quadratik programlama problemi için geliştirilen algoritmalarından biri Frank-Wolfe algoritmasıdır. Kısım 2.4.3.2'de verilecektir.

iii- y uzlaşık programlama problemi,

$$\text{enbenk } \left\{ \sum_{i=1}^k [f_i - f_i(x)], \dots, \sum_{k=1}^k [f_k - f_k(x)] \right\} \quad (2.38)$$

$X \in S$

Bu problem Y_u 'nun aşağıdaki sonuç teoremine göre (2.41) problemine dönüştürülür.

Sonuç teorem :

$$\text{Enk } \text{Enb } \left\{ \sum_{i=1}^k [f_i - f_i(x)] \mid i=1,2,\dots,k, \quad x \in S \right\} \iff$$

$$x \text{ ve } v = \text{enb } \left\{ \sum_{i=1}^k [f_i - f_i(x)] \right\} \text{ için} \quad (2.40)$$

$$\text{enk } \{v \mid v \geq \sum_{i=1}^k [f_i - f_i(x)], i=1,2,\dots,k, x \in S\}$$

problemi çözülür. Buradan (2.38) problemi

$\text{enk } v$

$$\text{Kısıtlar : } \sum_{i=1}^k [f_i - f_i(x)] \leq v \quad i=1,2,\dots,k \quad (2.41)$$

$x \in S$

ile çözülür.

2.4.3.2 Quadratik Programlama için F-W Algoritması

1. Adım : Kısıtları dikkate almadan , kısıtsız problemin minimum değerini bul. Bu nokta uygun ise problemin optimal çözümü bulunmuştur. Durulur. Aksi halde 2. adımdan devam edilir.

2. Adım : Kısıtları sağlayan herhangi bir uygun çözüm x^1 ve adım sayacı $h=1$ olsun. Uygun bir çözüm yoksa problemde çözümü yoktur.

3. Adım : Gradyent $\nabla f(x^h)$ hesaplanır.

$$\text{enk } [\nabla f(x^h)]^T y \quad (2.42)$$

Kısıtlar: $y \leq b$
 $y \in R^n$

Doğrusal programlama problemi çözümü y^k bulunur.

4. Adım : Bir boyutlu araştırma ardistirması uygulanır.

Yani,

$$\text{enk } f(w x^h + (1-w)y^k) \quad (2.43)$$

$$0 \leq w \leq 1$$

probleminin optimal sonucu bulunur. Optimal sonuç w^h olsun.

5. Adım : $x^{h+1} = w^h x^h + (1-w^h)y^k$ hesaplanır.

Aşağıdaki durdurma kriterlerinden biri gerçekleşirse çalışma kesilir.

$$i- \left| \frac{f(x^{h+1}) - f(x^h)}{\text{enb } \{ \epsilon_1, |f(x^{h+1})|, |f(x^h)| \}} \right| < \frac{\epsilon}{2} \quad (2.44)$$

$$ii- \left| \frac{x^{h+1} - x^h}{\text{enb } \{ \epsilon_3, \|x^{h+1}\|, \|x^h\| \}} \right| < \frac{\epsilon}{4} \quad (2.45)$$

Burada $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ önceden belirlenmiş küçük pozitif sayılardır.

Durdurma kriterleri gerçekleşmez ise, $h=h+1$ alınır, 2. adıma dönülür.

2.5 Zionts-Wallenius (Z-W) Yöntemi

Zionts ve Wallenius [34,35] aşağıda verilecek karakteristiklere sahip çok amaçlı karar problemlerini çözen bir algoritma geliştirmişlerdir:

- i- Kısıtları doğrusal,
- ii- Amaç fonksiyonları doğrusal veya dışbükey,
- iii- Gerçekte belli olmayan değer fonksiyonu ya doğrusal toplanabilir ya da amaç fonksiyonlarının içbükey bir fonksiyonudur. Her amacın tekdüze azalan (artan) bir fonksiyon olduğu da varsayılır [5].

Etkileşimli programlama ve buna bağlı eniyi uzlaşık çözüm değiş tokuş (trade-off) kullanılarak üretilir.

X doğrusal kısıtlarla tanımlanmış dışbükey çokyüzlü (polyhedron) olsun:

$$X = \{ x \in \mathbb{R}^N \mid Ax = b, x \geq 0 \} \quad (2.46)$$

Burada A , $(m \times N)$ boyutunda tam ranklı, amaç fonksiyonları doğrusal ve

$$f_i(x) = \sum_{j=1}^k c_{ij} x_j \quad i=1,2,\dots,n \quad (2.47)$$

biçiminde tanımlı olsun. Fayda fonksiyonu ise,

$$U(f_1, f_2, \dots, f_k) = - \sum_{i=1}^k w_i f_i \quad (2.48)$$

şeklindedir. Burada $\sum_{i=1}^k W_i = 1$ ve $W_i \geq 0 \quad i=1,2,\dots,k$

Fayda fonksiyonuna dayalı eniyi çözümü aramak için X çokyüzlüsünün etkin uç noktalarından birine veya etkin kenarda bir noktaya ulaşılmalıdır. Ardışık baskın uç noktaları keşfetme yöntemi bu amaç için çok yararlıdır. Çok amaçlı simpleks yöntemi ile ardışık uç noktalar elde edilebilir. Temelde olmayan herhangi bir sütun için, C_{ij} , temelde olmayan X_j değişkeninin temele girerken, bir birim artışa karşılık f^j 'deki değişim oranını gösterir. Sonuç olarak temelde olmayan C_j sütunu bu değişken temele girmesine karşılık meydana gelecek toplam bölüşümü gösterir. Karar verici C_j sütununa bakarak bölüşüm yapmak ister, istemez ya da çekimser davranabilir. Z-W yöntemi için önce Çok kriterli simpleks yöntemi verelim.

2.5.1 ÇKS Yöntemi [7]

Bu kısımda simpleks yönteminin genelleştirilmiş olanı çok kriterli simpleks yöntemini açıklayacağız. Yöntem yalnız baskın uç noktaları değil aynı zamanda her baskın uç noktaya karşılık gelen optimum ağırlıkları da verir.

Gösterimi basitleştirmek için,

$$S = \{ X \in \mathbb{R}^n \mid AX \leq b, X \geq 0 \} \quad (2.49)$$

A $m \times n$ matris, i . satır C_i ile gösterilirse

$$C_i^T x \quad i=1,2,\dots,k \quad (2.50)$$

amaç fonksiyonları ve $C = C_i$, amaç katsayıları matrisidir. $k \times n$

Böylece,

$$Y = \{ Cx \mid x \in X \} \quad (2.51)$$

ve (2.56) , wC (2.57) ile toplanarak (2.58) elde edilir.

(2.57) ve (2.58) 'den B $b \geq 0$ ise , $(X, X') = (B^{-1} b, 0)$ wC $B^{-1} b$ değeri ile temel uygun çözümdür. (2.57) ve (2.58) 'den değişkenler çıkarılarak çok kriterli simplex yönteminin formatı,

$$\begin{array}{c|cc|cc} & -1 & & -1 & \\ I & B & B' & B & b \\ \hline 0 & C & B' - C & C & b \\ & B & B & B & \end{array} \quad (2.60)$$

elde edilir. (A, I) matrisi (2.54) 'e göre düzenlenmez ise, (2.55) - (2.58) 'e karşılık gelen tablolar (ilk sütun çıkarılarak):

$$\begin{array}{c|cc} A & I & b \\ \hline -wC & 0 & 0 \end{array} \quad (2.61)$$

$$\begin{array}{c|cc|cc} & -1 & & -1 & \\ & B & A & B & \\ \hline & & & & -1 \\ & & & B & b \end{array} \quad (2.62)$$

$$\begin{array}{c|cc|cc} wC & -1 & & -1 & \\ B & B & A - wC & B & \\ \hline & & & & -1 \\ & & & B & b \end{array} \quad (2.63)$$

$$\begin{array}{c|cc|cc} & -1 & & -1 & \\ & B & A & B & \\ \hline & & & & -1 \\ & & & B & b \end{array} \quad (2.64)$$

$$\begin{array}{c|cc|cc} C & -1 & & -1 & \\ B & B & A - C & B & \\ \hline & & & & -1 \\ & & & B & b \end{array} \quad (2.65)$$

(2.64) ve (2.65) B temel ise, ÇKS tablosudur. (2.64) ve (2.65) yeniden

$$\left[\begin{array}{c|c} Y & Y^0 \\ \hline Z & V \end{array} \right]$$

(2.66)

biçiminde yazılır ise,

$$Y = \begin{bmatrix} -1 & -1 \\ B & A, B \end{bmatrix}, Y^0 = \begin{bmatrix} 0 & -1 \\ B & b \end{bmatrix}$$

(2.67)

$$Z = \begin{bmatrix} -1 & -1 \\ C & A - C, C \end{bmatrix}$$

ve

$$V = \begin{bmatrix} -1 \\ C & B \\ B \end{bmatrix} d.$$

(2.68)

2.5.2 ZW Algoritması

1. Adım : Her elemanı kesin sıfırdan büyük olmak ve toplamaları bir etmek kaydı ile bir ağırlık vektörü seçilir.

$$\text{Enb } U(f_1, f_2, \dots, f_k) = - \sum_{i=1}^k w_i f_i \quad (2.69)$$

$$x \in X$$

problemi çözülerek çözüm, başlangıç uç noktası olarak alınır. Ardıştıma adım sayacı $h=0$ olsun.

Ağırlık vektörü elemanları kesin sıfırdan büyük seçilerek elde edilen çözümün, baskın çözümler kümesi elemanı olduğunu Ecker ve Kouada [66] göstermiştir.

2. Adım : x^h noktasına ardışık baskın uç noktalar çok kriterli simpleks algoritması kullanılarak bulunur. Daha önce dikkate alınmamış temelde olmayan uç noktalar indis kümesi $J(x^h)$ olsun.

3. Adım : Karar vericiye her $j \in J(x^h)$ için C_j sütunları ile bölüşümler hazırlanır. $J.$ sütunda bölüşüm yapılacak ise;

$$-\sum_{i=1}^k w_i C_{ij} > 0 \quad (2.70)$$

yapılmayacak ise,

$$\sum_{i=1}^k w_i C_{ij} < 0 \quad (2.71)$$

ve kararsızlık,

$$\sum_{i=1}^k w_i C_{ij} = 0 \quad (2.72)$$

biçiminde kısıtlarla belirlenir. Karar verici temelde olmayan tüm sütunlar için değiş-tokuş yapmak istemiyor ya da bölüşümde kararsız ise durulur. Bulunan x^h noktası eniyi-uzlaşık çözümdür. Karar verici enaz bir sütun için de olsa bölüşüm yapmak isterse dördüncü adımdan devam edilir.

4. Adım : Karar vericinin bölüşüm kararlarına göre belirlenen kısıtlara,

$$\sum_{i=1}^k w_i = 1 \quad w_i \geq 0 \quad i=1,2,\dots,k \quad (2.73)$$

şartları da eklenerek w^{h+1} ağırlık vektörü bulunur.

$w = w^{h+1}$ alınarak

$$\text{Enb } U(f_1, f_2, \dots, f_k) = - \sum_{i=1}^k w_i f_i \quad (2.74)$$

$x \in X$

problemi çözülür.

5. Adım : Adım sayacı bir arttırılarak ikinci adımdan devam edilir.

2.6. Aralık Kriter Ağırlıklar / Vektör Enbüyükleme

Yaklaşımı

Steuer [36,67] tarafından geliştirilen bu yöntem kriter koninin daraltılmasına dayanır. ÇAKV problemi ,

$$\begin{aligned} & \text{enb } \{ C^1 X = Z^1 \} \\ & \text{enb } \{ C^2 X = Z^2 \} \\ & \vdots \\ & \text{enb } \{ C^k X = Z^k \} \end{aligned} \quad (2.75)$$

Kısıtlar : $X \in S$

modelindedir. Bu algoritmanın amacı enbüyük faydaya sahip etkin uç noktanın tayin edilmesidir. Etkileşimli yöntem, karar vericiye, her ardıştırmadan sonra belli sayıda baskın kriter vektörü sunar. Karar verici hazırlanan her gruptan hangi vektörü tercih ettiğini belirtmelidir.

Karar vericiye iletilen kriter vektörleri bir dizi DP probleminin enbüyüklenmesi ile üretilir. DP amaç fonksiyonu gradiyentleri (2.75) probleminin amaç fonksiyonları gradiyentlerinin özel dışbükey bileşimleri alınarak oluşturulur. Orijinal amaç fonksiyon gradiyenlerinden oluşan koni ardıştırma boyunca daraltılır. Sonuçta etkin uç noktası az olacak vektör enbüyükleme problemi kalır. Tüm etkin uç noktaları veren yöntemler kullanılarak problem çözülür. Steuer'in Aralık kriter ağırlıkları ve ağırlıklı toplamalar (2.7) algoritmalarında geçen süzgeçleme ve küme kesikleme izleyen kısımda incelenecektir.

2.6.1 Süzgeçleme ve Küme Kesikleme (Discretization)

Süzgeçleme sonlu sayıda büyük bir noktalar kümesinden noktalara ait altkümeler seçme işlemidir. Süzgeçleme ile büyük sonlu miktarda bilgi değerlendirilir. Sonlu sayı da noktalar kümesi üzerinde iki tip süzgeçleme söz konusu olabilir:

- 1- İleriye doğru süzgeçleme,
- 2- Geriye doğru süzgeçleme.

İleriye doğru süzgeçlemede, P , süzgeçte kalacak altküme eleman sayısını göstermek üzere, V vektör uzayından birbirinden farklı vektörler seçilir. Geriye doğru süzgeçleme de ise, P , geriye doğru süzgeçte kalacak küme eleman sayısını göstermek üzere V vektör uzayındaki vektörlere, seçilen bir vektöre göre sıra vererek, bu vektöre en yakın $P-1$ vektör seçilir.

Burada kullanılacak L_p ölçümleri daha önce (1.11) kesiminde verilmişti. Bu kısımda kullanılacak notasyonda p ölçüm parametresi, P ise, küme sayısını göstermektedir.

Ağırlıklı L_p uzaklık ölçümü :

$V \in R^q$ sonlu sayıda vektörler kümesi olsun. İleri ve Geri süzgeçlemede V^t ve $V^h \in V$ vektör çiftleri arasında uzaklığı ağırlıklı L_p ölçümüne göre V^t ve V^h vektörleri arasındaki uzaklık

$$\|V^t - V^h\|_p = \left[\sum_{i=1}^q (\pi_i |V_i^t - V_i^h|)^p \right]^{1/p} \quad (2.76)$$

ile ölçülür.

Bu ifadede,

q : Süzgeçlemeye alınacak vektör eleman sayısı.

π_i : Süzgeçlenmekte olan vektörün i . kısmıyla çarpılacak aralık eşitleme ağırlığı. Kısım (2.6.2)' de bahsedilecek.

p : ölçüm parametresi , $p \in (\{ 1,2,... \} \cup \{ \infty \})$.

Yukarıda verilen eşitlik, ağırlıklı L_p uzaklık ölçümü olarak adlandırılır.

2.6.2 Aralık Eşitleme Ağırlıkları

π_i , V vektör kümesinde vektörlerin elemanlarının aralıklarını eşitlemek amacıyla kullanılan ağırlıklardır. Böylece süzgeçleme işlemi, elemanların büyük aralığından yanlı olarak etkilenmeyecektir. π_i aralık eşitleme ağırlıkları, ağırlıkların görelî önemi ile karıştırılmamalıdır.

Aralık eşitleme ağırlıkları,

$$\pi_i = \frac{1}{R_i} \left[\sum_{j=1}^q \frac{1}{R_j} \right]^{-1} \quad (2.77)$$

ifadesiyle hesaplanır. Bu ifade de R_i , V kümesindeki vektörlerin i . kısımlarının aralığını verir.

$$R_i = \max \{ V_i \} - \min \{ V_i \} \quad v \in V \quad (2.78)$$

Örneğin,

$V_1 = (5,4,9)$, $V_2 = (12,3,10)$, $V_3 = (8,15,19)$ ise $R_1 = 7$, $R_2 = 12$, $R_3 = 10$. Aralık eşitleme ağırlıkları $\pi_1 = 0.04$, $\pi_2 = 0.02$ ve $\pi_3 = 0.03$.

Buradan V_1 ile V_2 ve V_1 ile V_3 arasındaki ağırlıklı L_p uzaklık ölçümü, sırasıyla,

$$\left[\sum_{i=1}^3 (\pi_i |V_i - V_i|)^2 \right]^{-1/2} = 0.6928$$

$$\left[\sum_{i=1}^3 (\pi_i |V_i^2 - V_i^1|) \right]^{-1/2} = 0.7874$$

Cıkan sonuclara göre V^2 vektörü V^1 vektörüne V^3 vektöründen daha yakındır.

2.6.3 Süzgeçleme Mekanizması

İleriye ve geriye doğru süzgeçlemede ağırlıklı L_p uzaklık ölçümü bir d test-uzunluk parametresiyle karşılaştırılır. Aralarındaki fark sıfırdan küçük ise vektör elenir, aksi durumda süzgeçte kalır. Süzgeçleme ilişkisi,

$$\left[\sum_{i=1}^q (\pi_i |V_i^t - V_i^h|)^p \right]^{1/p} - d < 0 \quad (2.79)$$

biçiminde ifade edilir. Burada

t : Süzgeç başlangıç noktası indisi

h : Süzgeçlemeye alınan nokta indisi

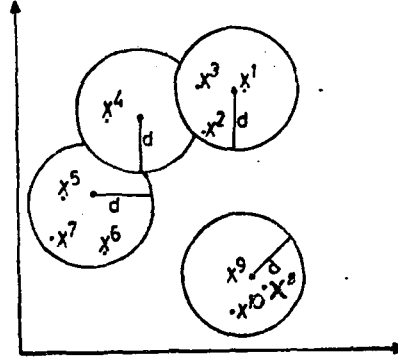
V_t ile V_h arasındaki uzaklık d test parametresinden büyük ise iki nokta önemli derecede birbirinden farklı, aksi takdirde önemli derecede farksız olarak kabul edilir.

2.6.3.1 İleriye Doğru Süzgeçleme

İleriye doğru süzgeçleme üç değişik yolla yapılabilir:

- 1- Komşuluk dışında ilk nokta yöntemi,
- 2- Komşuluk dışında enyakın nokta yöntemi,
- 3- Komşuluk dışında enuzak nokta.

Birinci yöntem en hızlı, ikinci ve üçüncü yöntemler ise hesaplama fazlalığına karşılık iyi sonuçlar verdiği Steuer tarafından bildirilmektedir.



Şekil 6. Komşuluklar dışında ilk nokta örneği

2.6.3.1.1 Komşuluklar Dışında İlk Nokta Yöntemi

Yöntemi bir örnekle açıklayalım.

Bu yöntemle yukarıda grafikte görülen 10 vektörden 4 tanesini seçmek isteyelim. Uygun bir test-uzaklık parametresi d ve 10 vektör süzgeçleme algoritması için listelenir. Listenin başındaki ilk vektör ileri seed noktasıdır. İleri hareket noktası süzgeçlemeyi başlatır ve süzgeçte kalan ilk noktadır. Süzgeçte kalacak ikinci noktayı bulmak için geride kalan dokuz nokta tek tek işleme alınır. Süzgeçte kalan ilk nokta ile diğer vektörler sırasıyla süzgeçleme ifadesi yardımıyla karşılaştırılır. x_2 ve x_3 vektörleri x_1 ile önemli derecede farklı olmadığından elenir. Komşuluk dışında önemli derecede farklı olan ilk nokta x_4 vektörüdür. Bundan sonra süzgeçte kalacak nokta x_1 ve x_4 vektörlerinden önemli derecede farklı olan ilk noktadır. x_5 vektörü x_1 ve x_4 vektörlerinden yeterince farklı olan x_5 , seçilen üçüncü vektördür. x_6 ve x_7 vektörleri x_1 vektöründen önemli derecede farklı olmadığı için elenir. Şimdi x_1 , x_4 ve x_5 vektörlerine

komsuluk dışında önemli derecede uzak ilk nokta X_9 vektörüdür.
Süzgeçte kalır. X_{10} noktası X_9 noktasından önemli derecede farklı
değildir. Elenir. Böylece $\{X_1, X_2, \dots, X_{10}\}$ vektör kümesinden
 $\{X_1, X_4, X_5, X_9\}$ süzgeçte kalmıştır.

Secilen test parametresi her zaman süzgeçlenecek vektör kümesinden istenilen miktarda vektör seçimine uygun olmayabilir. Secilen test parametresi büyük olduğunda süzgeçleme ile eksik elemanlı vektör kümesi, küçük olduğunda ise fazla elemanlı vektör kümesi elde edilir. Bu durumu önlemek için d test parametresi eksiltme ve arttırma işlemlerine tabi tutularak uygun büyüklük sağlanır.

Uygun Test Parametresi İçin Algoritma

İleriye doğru süzgeçleme için kural;

1- Süzgeçte kalan küme eleman sayısı istenenden az ise d test parametresini küçült.

2- Süzgeçte kalan küme eleman sayısı istenenden fazla ise d test parametresini büyült.

P sayıda noktalar kümesi elde etmek için test parametresi d uygun değilse bölme ve katlama algoritması yardımıyla düzeltilir. Algoritma başlangıçta d test parametresini ve $\wedge d$ arttırma ve eksiltme miktarını ister. Uygun seçim formülleri bir sonraki kesimde verilecektir. Algoritmada geçecek olan BÜYÜKX=1 ileriye doğru süzgeçleme ile edilen noktalar kümesinin çok büyük, KÜÇÜKX=1 ise süzgeçleme sonunda elde edilen azaltılmış kümesinin çok küçük olduğunu gösterir.

Bölme ve katlama algoritması aşağıdaki şekildedir.

1. Adım : d ve /\d başlangıç değerlerini al.

2. Adım : BÜYÜKZ=0 ve KÜÇÜKZ=0 olsun.

3. Adım : Vektörleri ileriye doğru süzgeçle.

4. Adım : Süzgeçte kalan vektör sayısı VS olsun.

i- VS=P ise dur. İstenen sayıda vektör kümesi elde edilmiştir.

ii- VS>P ise BÜYÜKZ=1 olsun. 5. adıma git.

iii- VS<P ise KÜÇÜKZ=1 olsun.

5. Adım : BÜYÜKZ=1 ve KÜÇÜKZ=1 değilse 6. adıma git.

/\d=/\d/2 olsun. 8. adıma git.

6. Adım : İlk ardıştırmayı mı ? Evet ise 8. adıma git.

7. Adım : /\d=2/\d olsun.

8. Adım : VS>P ise d=d+/\d al. 3. adıma git.

9. Adım : d=d-/\d al. 3. adıma git.

Başlangıç d ve /\d değerlerinin seçimi :

Süzgeçleme işlemi sonucunda kesin olarak istenen sayıda vektör kümesi elde etmeyi garantileyen öncel d test parametresi hesaplamak mümkün değildir. Ancak Steuer aralık eşitleme ağırlıklarından hareketle d ve /\d değerlerini aşağıdaki biçimde seçip, uygulamada iyi çalıştığını görmüştür:

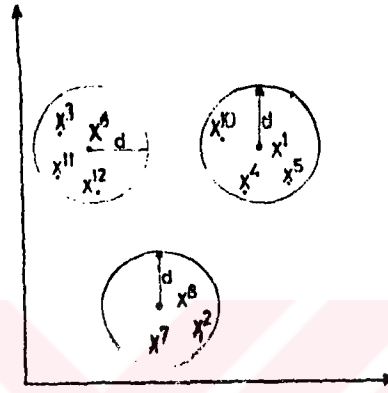
$$\text{başlangıç } d = \frac{\left[\sum_{i=1}^q (\pi R_i)^P \right]}{4} \quad (2.80)$$

ve

$$\text{başlangıç } /\d = \frac{\text{başlangıç } d}{2} \quad (2.81)$$

2.6.3.1.2 Komşuluklar Dışında En Yakın Nokta Yöntemi

Bu yöntemi açıklamak için 12 nokta içinden ileriye doğru 3 elemanlı bir kümenin seçileceğini varsayalım. Noktalar Şekil 7'de olduğu gibi dağılmış iseler,



Şekil 7. Komşuluklar dışında en yakın nokta örneği

Komşuluklar dışında ilk nokta yöntemindeki gibi ilk nokta hareket noktası olarak seçilir. Bu nokta-süzgeçte kalmış sayılır. Geri kalan 11 noktanın hareket noktasına L_1 normunda uzaklığı bulunur. Test parametresi d ile karşılaştırılır. Önemli derecede farklı olmayan noktalar atılır. Örneğimizde x_1, x_{10} vektörleri x_1 vektöründen önemli derecede farklı değildir. Komşuluklar dışında ilk nokta süzgeçte kalmış sayılır. Örneğimizde bu nokta x_6 vektörüdür. x_1 ve x_6 vektörlerine önemli derecede uzak olmayan x_3, x_{11}, x_{12} noktaları elenir. Geri kalan 6 nokta $x_2, x_4, x_5, x_7, x_8, x_9$ dışında geriye kalan 3 nokta x_1, x_6, x_8 bu noktanın komşuluğu içinde olması nedeniyle elenir. Elde edilen 3 elemanlı küme x_1, x_6, x_8 vektörlerinden oluşur. Eğer komşuluklar dışında ilk nokta yöntemi kullanılsaydı süzgeçte kalacak küme

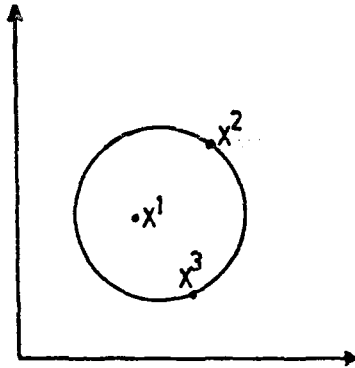
$\{ X^1, X^2, X^3 \}$ olacaktı.

2.6.3.1.3- Komşuluklar Dışında En Uzak Nokta Yöntemi

Bu yöntem komşuluklar dışında enyakın noktanın tersine olarak seçilen hareket noktasından enuzaktaki noktayı süzgeçte bırakarak ilerler. Listenin başındaki ilk vektör hareket noktası seçilir ve süzgeçte kalır. Süzgeçte kalacak ikinci nokta, X^1 noktasına önemli derecede farklı uzaklıktaki noktalardan enuzağı olur. Şekil 7'deki X^7 vektörüdür. Üçüncü nokta ise X^1 ve X^7 vektörlerine en uzak X^3 noktasıdır. Böylece $\{ X^1, X^7, X^3 \}$ kümesi süzgeçte kalmış olur.

2.6.4 Küme Sayısını Sağlayamama

Komşuluklar dışında ilk nokta, en yakın nokta veya en uzak nokta yöntemleriyle istenen sayıda azaltılmış küme, vektör dağılımının uygun olmaması nedeni ile mümkün olmayabilir. Bu durumda test parametresi d azaltılarak veya arttırılarak istenen az farklı sayıda kümeye ulaşılabilir. Şekil 7'de verilen örnekte istenen iki elemanlı küme ileriye doğru süzgeçleme yöntemlerinden hiçbirisiyle sağlanamaz.



Şekil 8. Küme sayısını sağlayamama

2.6.5 Geriye Doğru Süzgeçleme

V sonlu sayıda vektör kümesini gösterebilir. Bu kümeden P sayıda altküme seçilecek ise, geriye doğru süzgeçleme için hareket noktası olmak üzere bir $v \in V$ vektörü seçilir. Diğer vektörlerin bu vektöre göre ağırlıklı L_p ölçümleri alınarak artan büyüklükte sıralanır (ranked). v vektörü ile geriye kalan $P-1$ en yakın komşu vektörler alınır. Bu listeye geriye liste denir.

2.6.6 Küme-içi Noktalar Üretme ya da Küme Kesikleme

Bir matematiksel programlama algoritması, bir probleme uygulandığında, uygun çözüm bölgesinin, sürekli, çok boyutlu bir alt setinin çözüm kümesi olduğunu varsayalım. CADP tüm etkin küme için bu karakteristik olarak gerçekleştirilecektir. Bu çeşit problemlerde çözüm kümesi uygun bölgenin dışbükey olmayan bir parçası olabilir. Bu çözüm kümesini üretilen sonlu etkin uç nokta ile göstermek muhtemelen riskli ve sağlıksızdır [68]. Çünkü bu noktalar etkin kümenin üstünde veya altında olabilirler. Bazen, beklenmediği halde, yeterli kriter vektörleri de bulunmayabilir. Tüm bu nedenlerle kriter vektörlerinin dışbükey bileşimleri alınarak küme içi (intra-set) noktalar üretilir.

Küme-içi nokta üretme ya da küme kesikleme (discretization) ile sürekli bir küme sonlu sayıda nokta ile temsil edilmiş olur.

Dışbükey bileşimler üç yolla elde edilebilir :

(1) Önceden belirlenmiş dışbükey bileşimler : Bu yaklaşımda bileşim ağırlıkları birbirinden mümkün olduğunca farklı seçilir. Çünkü ağırlık vektörleri yaygın dağılmış ise sonuç bileşim vektörleri de yaygın dağılacaktır. Ancak bu her zaman gerçekleşmeyebilir.

(2k+1) ağırlık,

$$\left. \begin{array}{l} W^1 = (1, 0, \dots, 0) \\ W^2 = (0, 1, \dots, 0) \\ \vdots \\ W^k = (0, 0, \dots, 0) \end{array} \right\}$$

Uç ışık
dışbükey bileşimler

(2.82)

$$\left. \begin{array}{l} W^{k+1} = (1/k^2, r, \dots, r) \\ \vdots \\ W^{2k} = (r, r, \dots, 1/k^2) \end{array} \right\}$$

Merkez dışı
dışbükey bileşimler

$$W^{2+1} = (1/k, 1/k, \dots, 1/k) \text{ } \text{Merkez dışbükey bileşimler}$$

(2) Düzgün dağılımından çekilen dışbükey bileşimler :

Birinci yöntemin eksiklerini önlemek için fazla miktarda tesadüfi ağırlık vektörü üretilerek çok benzer olanlar atılır. Bu yöntemin etkinliği ağırlıkların alınacağı dağılıma bağlıdır. W_i dışbükey birleşim ağırlıkları,

$$W_i = V_i / \left(\sum_{j=1}^n V_j \right), \quad (2.83)$$

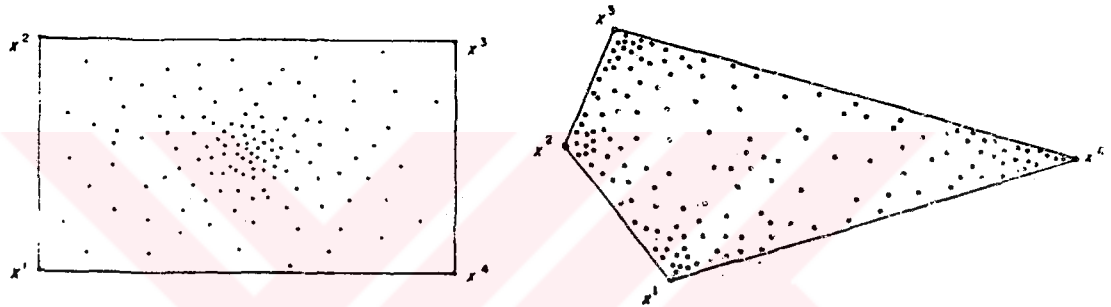
oluşturan V_i değerleri (0,1) aralığında düzgün dağılımdan çekilir. Şekil 9a'da görüldüğü gibi bu ağırlıklar kullanılarak elde edilen dışbükey birleşimler daha çok merkezde toplanır.

(3) Weibull dağılımından çekilen birleşimler : Ağırlık formülünde geçen V_i ler b ve c parametrelili Weibull dağılımından çekilirse dışbükey birleşimler çok yüzlünün (polihedron) köşelerine yakın dağılacaktır. Şekil 9b' Tesadüfi ağırlıkları üretmek için,

$$b \left[-\ln(R;0,1) \right]^{1/c} \quad (2.84)$$

ifadesinden yararlanılır. $(R;0,1)$, sıfır bir arasında düzgün dağılımından üretilen bir sayı, b ve c parametreleri ise küçük değerler seçilmelidir.

Üretilecek dışbükey birleşimlerin daha iyi bir dağılımı, yarısı ikinci, yarısı da üçüncü yöntem ile seçilerek sağlanabilir.



Sekil 9. Düzgün ve Weibull dağılımından bileşimler

2.6.7 Baskın Çözümleri Azaltmanın Diğer Yolları

Baskın kümeyi temsil eden alt kümeleri elde etmek için başka bir yöntem de Morse [69] tarafından geliştirilmiştir. Morse'yi yönlendiren, karar vericinin karar vermesi değil, iki çok boyutlu çözüm arasındaki farkı fark edip etmeyeceğiydi. Morse, baskın kümeyi azaltmak için istatistiksel kümeleme (Cluster) analizini kullandı.

Baskın kümenin elemanları nitelik ve nicelik bakımından birbirine benzer. Kümeler birimlerin karışımıdır ve birimler gruplara göre birbirine daha yakındır. Morse'nin tekniğinde Steuer'in süzgeçleme yönteminde kullandığı test uzaklık

parametresine gerek olmayan blok kümelemeye başvurulmuştur. Blok kümeleme şu şekilde çalışır : Her boyut (kriter, nitelik) erişilen endüşük ve enyüksek değerler arası (genişlik, range) aralıklara bölünür. Her x değeri 1,2,3,..., rank sayıları ile sıralandırılır. Çözümler dikey, kriterler yatay ekseninde yerleştirilirse verilerin sınıflandırılmış matrisi elde edilir. Satırları ve sütunları, benzer değerler birbirinin yanına gelinceye kadar değiştirerek çözümlerin ve amaçların kümesi elde edilir. Bu arada bazı kriterler bağlantılı ise çıkartılır veya yer değiştirilir.

Törn'ün [70] makalesi ise üçüncü kısmı oluşturur. Törn, verilen çözümde, subjektif güvene dayalı, karar vericiye bağımlı, optimum kavramını üreterek Steuer ve Morse'nin tekniklerini birleştirmiştir.

2.6.8 Amaç Fonksiyonlarını Ölçekleme

Amaç fonksiyonları üç yolla yeniden düzenlenebilir.

- 1- Birimleştirme,
- 2- Uygun dereceden 10'un kuvvetleri ile çarpma,
- 3- Aralık eşitleme faktörü uygulayarak düzeltme.

Amaç fonksiyonu katsayıları çok farklı büyüklüklerden oluştuğunda ya da ağırlıklı toplamlar DP kullanılacak ise L_p normlarından biri kullanılarak ölçeklenebilir. Steuer'in önerdiği diğer bir yolda amaç fonksiyonlarını 10'un uygun kuvvetleri ile çarparak, katsayıları bilgisayar anlamında normalize etmektir. Bu yolla yapılan ölçekleme ile katsayıların sadece kesir noktası hareket eder. Böylece, birinci yolun aksine sayılar hala tanınabilir. Üçüncü yöntemde ise

$$\pi_i = \frac{1}{R_i} \left[\sum_{j=1}^k \frac{1}{R_j} \right]^{-1} \quad (2.85)$$

ifadesiyle hesaplanan π_i aralık eşitleme faktörleri kullanılır [36,68].

2.6.9 Algoritma

1. Adım : Algoritmanın parametreleri, P ve t belirlenir. P parametresi, karar vericiye her ardıştırmadan sonra sunulacak vektör sayısıdır. Yine önceden belirlenen t, algoritmanın toplam adım sayısını göstermektedir. P ve t parametreleri analistle karar vericinin karşılıklı görüşmesiyle belirlenebilir. Amaç sayısı k olmak üzere, P ve t,

$$k \leq P \leq 2k+1$$

$$t \approx k$$

ilişkilerine uygun seçilmesi önerilebilir.

2. Adım : Satırları amaç fonksiyonundan oluşan C matrisi normalleştirilir.

Normalleştirme, Lp-ölçümlerini kullanarak yapılabileceği gibi, amaç fonksiyonları 10'un uygun kuvvetleri ile çarpılarak da sağlanabilir. Bu yaklaşım ile yapılan tek değişiklik amaç fonksiyonu katsayılarının kesir noktasının hareket etmesidir.

3. Adım : Ardıştırma adım sayacı $h = 1$ olsun. Dışbükey birleşim ağırlıklandırma vektörleri,

$$W_1 = (1, 0, \dots, 0)$$

$$W_2 = (0, 1, \dots, 0)$$

.

.

.

.

$$W_k = (0, 0, \dots, 1)$$

$$W_{k+1} = (1/k, r, \dots, r)$$

(2.86)

$$W = (r, r, \dots, 1/k) \quad (2.86)$$

$$W = (1/k, 1/k, \dots, 1/k) \quad (2.87)$$

Burada $r=(k+1)/k$ seçilebilir.

Ağırlıklandırma vektörlerinde sıfır yerine, çok küçük, ama hesaplamalarda duyarlı değerler kullanılır.

4. Adım : $(2k+1)$ dışbükey birleşim DP problemi,

$$\text{enb } \{(W_i) \in \mathbb{R}^n | W_i \in \mathbb{S}\} \quad i=1,2,\dots,2k+1 \quad (2.87)$$

çözülür. C satırları h . daraltılmış kriter koniyi sarmak için üreticileri belirleyen $k \times n$ boyutlu matristir.

$$C_h = \begin{cases} \begin{bmatrix} T_{h-1} & T_{h-2} & \dots & T_1 \\ & & & 1 \end{bmatrix} & h > 1 \text{ ise} \\ C_1 & h = 1 \text{ ise} \end{cases} \quad (2.88)$$

Birinci ardıştırmada orijinal kriter koni ile, takip eden ardıştırmalarda ise, daraltılmış kriter koninin alt kümeleriyle çalışılır. Öncarpım T matrisleri kısım [36,9.10.5]'de verilmiştir. Bir kriter koninin üretkenleri uygulandığında, ağırlıklandırma vektörü kriter koni üzerinde merkezzel dağılan bir grup üçlü gradiyent oluşturur.

5. Adım : Sonuç kriter vektörleri süzgeçlenerek çok farklı P tanesi karar vericiye sunulur.

Alternatif optimum olasılığı ve verilen bir uç noktanın birden fazla dışbükey birleşim DP problemlerinin çözümü olabileceği nedeniyle bu adımda elde edilen farklı kriter vektörlerinin sayısı P 'den az olabilir. Bu nedenle P sayısının $2k+1$ alınmasında yarar vardır.

6. Adım : Karar verici, P baskın kriter vektörlerinden en çok tercih ettiklerini seçip $Z^{(h)}$ teşkil edilir.

7. Adım : $(h+1)$ ardıştırma daraltılmış koniyi oluşturmak için $Z^{(h)}$ 'ı üreten üçlü gradiyentler ile h ardıştırma kriter koni kurulur.

8. Adım : Ardıştırma adım sayacı $h = h + 1$ alınır.

9. Adım : Eğer $h < t-1$ ise 4. adıma dön.

10. Adım : Azaltılmış vektör enbüyükleme probleminin C_t matrisine göre tüm etkin uç noktaları bulunur.

Esas kriter koni $(t-1)$ defa daraltıldığından, t. ardıştırmada azaltılmış koni oldukça küçülmüş olacaktır. Bu nedenle bu adımda üretilecek etkin uç nokta sayısı az olacaktır.

11. Adım : 10. adımda, üretilen kriter vektörlerini süzgeçleyerek çok farklı P tanesi karar vericiye sunulur.

12. Adım : Karar verici P baskın kriter vektöründen en çok tercih ettiğini seçer. Bu kriter vektörüne ait asıl değişkenlerin oluşturduğu $X^{(t)}$ son çözümdür.

2.7 Etkileşimli Ağırlıklı Toplamlar / Süzgeçleme Yöntemi

Steuer ve Wood tarafından kullanılan etkileşimli ağırlıklı toplamlar / süzgeçleme yaklaşımı ÇADP problemini çözmek için ağırlıklandırma vektör uzayını daraltan bir yöntemdir.

Yaklaşım, kriter ağırlıklandırması yöntemi felsefesinin bir benzeridir. Bu yöntemde de her ardıştırma sonunda, belli, sabit sayıda çözüm, karar vericiye sunulularak en büyük faydaya sahip etkin uç noktayı belirlemesi istenir. Yöntemin karar vericinin fayda fonksiyonunun sürekli düzenli artan bir fonksiyon olmasından

başka herhangi bir varsayımı yoktur. Bu algoritmada kriter koninin daraltılması yerine,

$$\Lambda = \{ W \in \mathbb{R}^k \mid W_i \geq 0, \sum_{i=1}^k W_i = 1 \}$$

ağırlıklandırma vektör uzayını daraltır. Başka bir farkta önceden hesaplanmış dışbükey kombinasyon w-vektörleri kullanılmamasıdır. Bunun yerine düzgün ve Weibull dağılımından tesadüfi üretilir.

2.7.1 Algoritma Parametrelerinin Değer ve Sınırlarının Belirlenmesi

Bu etkileşimli algoritma ardıştırmalarında w-vektörleri

$$\Lambda^{(h)} = \{ W \in \mathbb{R}^k \mid W_i \in \left[\frac{1}{i^{(h)}}, \frac{u_i^{(h)}}{i^{(h)}} \right], \sum_{i=1}^k W_i = 1 \} \quad (2.89)$$

ağırlıklandırma uzayından alınarak ağırlıklı toplamalar DP problemleri çözülür, baskın kriter vektörleri elde edilir. Ağırlıkların alındığı

$$\left[\frac{1}{i^{(h)}}, \frac{u_i^{(h)}}{i^{(h)}} \right] \text{ aralığı}$$

ardıştırma adım sayacı arttıkça daraltılır. Algoritmada kullanılan parametreler şunlardır :

t : Algoritmanın toplam ardıştırma sayısı.

P : Her ardıştırma sonunda karar vericiye sunulacak çözüm sayısı.

W : Son ardıştırmada, $\left[\frac{1}{i^{(h)}}, \frac{u_i^{(h)}}{i^{(h)}} \right]$ aralığının ulaşacağı genişlik.

r : Λ ağırlıklandırma uzayını daraltma faktörü.

P ve t parametrelerini, analistle karar verici tartışarak kararlaştırılabilir. Son aralık genişliğini analist tecrübelerine göre seçebilir. Analistin takdirine göre W son aralık genişliği $1/(2k)$ ile $1/(3k)$ arasında bir değer genelde yeterli olur. W ile bağıntılı olarak r daraltma faktörü için uygun değer bulunur.

Steuer [36] bu parametrelerin kabaca aşağıdaki standart ilişkilere göre seçilmesini önermektedir :

$$P \geq k$$

$$t \approx k$$

$$1/(2k) \leq W \leq 1/(3k) \quad (2.90)$$

$$k \frac{t-1}{\sqrt{1/P}} \leq r \leq \frac{t-1}{\sqrt{W}}$$

Burada " $\approx$ " yaklaşık anlamında kullanılmıştır. Dördüncü ilişki k-boyutlu çoklu-küpten çıkarılmıştır. Birim çoklu-küpten P tane örneğin herbiri $1/P$ büyüklüğünde alt çoklu-küple temsil etmek düşünülmüştür. Eğer bir tarafta çoklu-küpün alt kümesinin uzunluğu r olursa $r > 1/P$ ilişkisinden

$$r \geq k \frac{t-1}{\sqrt{1/P}} \quad (2.91)$$

elde edilir.

Son ardıştırmada h ağırlıklarının alındığı $\begin{bmatrix} t & t \\ 1 & u \\ i & i \end{bmatrix}$

aralık genişliği W'ye eşit veya küçük olduğundan $r^{t-1} \leq W$

sağlanmalıdır. Buradan $r \leq \frac{t-1}{\sqrt{W}} \quad (2.92)$

bulunur. İki sınır birarada dördüncü ilişkiyi oluşturur.

2.7.1 Algoritma

1. Adım : Algoritma parametreleri ; örnek çapı P, toplam ardıştırma sayısı t, son

$$\begin{bmatrix} 1 \\ i \end{bmatrix}, |u_i$$

aralık genişliği W ve

ağırlıklandırma uzayı (/), daraltma faktörü r, belirlenir.

Parametreler 2.7.1 kesiminde önerilen biçimde seçilir.

2. Adım : Amaç fonksiyonları normalleştirilir. (Yeniden ölçeklenir).

Amaçlar kesim (2.6.8) de belirtilen yollardan biriyle yeniden ölçeklendirilir.

3. Adım : Ardıştırma adım sayacı h = 0 ve

$$\begin{bmatrix} 1 \\ i \end{bmatrix}, |u_i = \begin{bmatrix} 0,1 \end{bmatrix} \text{ tüm } i\text{'ler için alınır.}$$

4. Adım : h = h + 1 olsun. Ağırlıklandırma uzayı,

$$\wedge^{(h)} = \{ W \in \mathbb{R}^k \mid w_i \in \text{rel} \begin{bmatrix} 1 \\ i \end{bmatrix}, |u_i^{(h)} \}, \sum_{i=1}^k W_i = 1 \} \quad (2.93)$$

oluşturulur.

(1)
İlk ardıştırmada $\wedge = \wedge^{(1)}$, takip eden ardıştırmalarda ise,
(h)
 $\wedge^{(h)} \wedge^{(h-1)}$ olur, çünkü,

$$\begin{bmatrix} 1 \\ i \end{bmatrix}^{(h)}, |u_i^{(h)}$$

aralıkları açık birim aralığın alt aralıklarıdır.

(h)
5. Adım : $\wedge^{(h)}$ ağırlık uzayından rassal $50 \times k$ ağırlık vektörü seçilir.

Steuer $50 \times k$ vektörünün yarısını düzgün dağılımından yarısını da Weibull dağılımından çekmeyi öğretmektedir.

6. Adım : $50 \times k$ ağırlık vektöründen çok farklı $3 \times P$ tanesi belirlenir.

5. Adımdaki düşünce, W vektörlerinden bir havuz oluşturup
6. adımda bunları süzgeçleyerek, $\wedge^{(h)}$ ağırlık uzayını temsil eden, farklı dağılmış belli sayıda h ağırlık vektörü elde etmektir.

7. Adım : $\wedge^{(h)}$ ağırlıklandırma uzayını temsil eden $3 \times p$ ağırlık vektörlerini kullanarak oluşturulan ağırlıklı toplamalar DP problemleri çözülür.

W ağırlık vektörü elemanları, açık birim aralığın alt bölümlerinden gelmesinden dolayı tüm enbüyüklenen uç noktalar etkindir. Bu nedenle tüm sonuç kriter vektörleri baskındır.

8. Adım : 7. adımda elde edilen kriter vektörleri süzgeçlenir. Çok farklı P tanesi seçilir. Seçilen P baskın kriter vektörü karar vericiye sunulur.

Özellikle son ardıştırmalarda, bir uç noktanın, birden çok ATDP problemini enbüyükleyebilmesi nedeniyle fazla sayıda ATDP problemi çözmek gerekli olur. Fazla sayıda ATDP problemini çözmenin başka bir nedeni de h ağırlık vektörlerinin iyi dağılmış olmasına karşılık sonuç kriter vektörlerinin iyi olmayan bir dağılım gösterebileceğidir.

9. Adım : P tane baskın kriter vektörü içinden karar verici çok beğendiği vektörü seçer. Böylece $Z^{(h)}$ belirlenmiş olur.

10. Adım : Eğer ardıştıрма adım sayacı, toplam ardıştıрма sayısından küçük ($h < t$) ise, 11. adımdan, değilse 12. adımdan devam edilir.

11. Adım : ATDP ile $Z^{(h)}$ kriter vektörünün üreten ağırlık vektörü $w^{(h)}$ ise,

$$\wedge \begin{matrix} (h+1) \\ \end{matrix} = \{ w \in \mathbb{R}^k \mid w_i \in \text{rel} \left[1, \frac{u_i^{(h+1)}}{u_i^{(h)}} \right] \right\} \quad (2.94)$$

$$\sum_{i=1}^k w_i = 1 \}$$

oluşturulur.

Buradan

$$\begin{bmatrix} 1 \\ \frac{u_i^{(h+1)}}{u_i^{(h)}} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0, r^h \\ 1 - r^h, 1 \end{bmatrix} w_i^{(h)} - \frac{r^h}{2} \leq 0 \text{ ise} \\ \begin{bmatrix} 1 - r^h, 1 \end{bmatrix} w_i^{(h)} + \frac{r^h}{2} \geq 1 \text{ ise} \\ \begin{bmatrix} w_i^{(h)} - \frac{r^h}{2}, w_i^{(h)} + \frac{r^h}{2} \end{bmatrix} \text{ d.h.} \end{cases} \quad (2.95)$$

ve r^h daraltma faktörüdür. 4. Adıma dönülür.

12. Adım : Son çözüm $X^{(h)}$ ve amaç fonksiyon değerleri $Z^{(h)}$ ile durulur.

2.8. Korhonen ve Laakso'nun Görsel Etkileşimli Yöntemi

Bu yöntemin temel yaklaşımı, kriter uzayında bir doğru parçasını adım adım baskın yüzeye izdüşürmektir. Baskın yüzeye izdüşüm başarı ölçekleme fonksiyonu ve sağ taraf parametrik programlama kullanılarak gerçekleştirilir [36].

Algoritma fayda fonksiyonuna dayalı kurulmakla birlikte gerçekte karar vericinin tercihlerini yansıtan bu fonksiyon hakkında herhangi bir varsayım yapmaz. Hatta fayda fonksiyonunun

etkileşimli işlem boyunca, çözümler görüldükçe, değişebileceğini varsayar.

Algoritmanın genel çalışma alanı GDF yöntemine benzer. Geoffrion ve arkadaşlarının yönteminde kullanılan Frank-Wolfe algoritması bazen etkin olmayan çözümler vermektedir [71]. Bu yüzden yeni yöntemde Frank-Wolfe algoritması yerine Hemming [72] ve Winkels ve Meikka'nın [73] bahsettiği düzeltilmiş gradyent izdüşüm metodu (modified gradyent projection method) kullanılmıştır. Bununla birlikte fayda fonksiyonu gerçekte bilinmediğinden gradyenti alınamaz. Yeni araştırma yönü karar vericinin tercihlerini yansıtan referans yönü kullanılarak belirlenir. Her yön etkin sınıra (efficient frontier) izdüşülür ve etkin yüzeyden geçen bir eğri elde edilir. Bu çizgi boyunca etkileşimli çizgisel araştırma gerçekleştirilir [71]. Başarı ve diğer ölçekleme fonksiyonları (2.8.1) kısmında verilecektir.

2.8.1 Ölçekleme Fonksiyonları

Etkileşimli yöntemlerde ölçekleme fonksiyonu, (etkin) çözümleri hesaplamamanın teknik bir yoludur, karar verici ile model arasında değiştirilen bilgiyi güçlü biçimde etkiler. Etkileşimli programlamada kullanılan genel ölçekleme fonksiyonlarından bazıları

1- Ağırlıklandırma yöntemi

Amac fonksiyonlarının doğrusal bir kombinasyonunu bir ağırlıklandırma vektörü $W \in R^k$ ile dikkate almak, geleneksel bir yoldur. Varsayımımız, $w_i \geq 0$, $i=1,2,...,k$ ve $\sum_{i=1}^k w_i = 1$.

Bu takdirde CADP problemi,

$$\begin{aligned} \text{enb} \quad & \sum_{j=1}^n w_j C_j X_j \\ \text{kısıtlar: } & A X \leq b \\ & X \geq 0 \end{aligned} \quad (2.96)$$

DP problemine dönüşür. (2.96) probleminin çözümünün etkinliği ile ilgili iyi bilinen iki teorem [5]:

2.8.1.1 Teorem : Uygun bir $\bar{X}$ noktası etkindir $\Leftrightarrow$

$$\text{her } w \geq 0 \text{ öyleki } x \in S \text{ için } w^T C \bar{X} \geq w^T C X$$

2.8.1.2 Teorem : (2.96) probleminin çözümü tektir $\Rightarrow$ $\bar{X}$ etkindir.

2- Sınır yöntemi

Etkin çözümler üretmenin diğer bir yolu sınır yöntemidir.

Bu yöntemde bir amaç fonksiyonu optimize edilirken diğerleri $k-1$ alt sınır vektörleri ile kısıtlanır.

Bu alt sınır vektörü $1 \in R$ ile problem,

$$\begin{aligned} \text{enb} \quad & (C) X + \sum_{j=1}^k \epsilon_j (C_j) X_j \\ & j \neq i \end{aligned} \quad (2.97)$$

$$\text{kısıtlar : } A X \leq b$$

$$\begin{aligned} & (C_j) X \geq 1_j \quad j=1,2,\dots,k \quad j \neq i \\ & X \geq 0 \end{aligned}$$

Amaç fonksiyonunda küçük bir artık (perturbation) terimi eklenerek $\bar{X}$ çözümünün problemin etkin çözümü olması garantilenir [39]. ϵ_i değerleri tüm i 'ler için sıfıra yakın pozitif büyüklüklerdir.

2.8.1.3 Teorem : Uygun bir $\bar{X}$ noktası etkindir $\iff \bar{X}$ (2.97) problemini $1 = C \bar{X}$ sınır vektörü ile tüm j 'ler için çözer [5].

2.8.1.4 Teorem : Uygun bir $\bar{X}$ noktası, (herhangi alt sınır vektörü ile (2.97) probleminin tek çözümüdür $\implies \bar{X}$ etkindir. Bu ölçekleme fonksiyonunu kullanan etkileşimli yöntemlerden biri Sprong'un Etkileşimli çok amaçlı Hedef programasıdır [39,74].

3- Ağırlıklı Tchebycheff Ölçüm Yöntemi

Bu yöntemde temel, tüm uygun noktalar, amaç uzayında sabit bir noktaya göre ölçülür. Genellikle bu referans noktayla uygun çözümler kümesi arasındaki uzaklık en küçüklenir.

Bu uzaklık, çeşitli "norm" lar ile ölçülür. Bununla birlikte norm seçimi (veya herhangi başka fonksiyon referans noktayı uygun bölgenin etkin sınırına (efficient frontier) izdüşürür.) etkileşimli programlamada problemin başka bir boyutudur. QADP problemi,

$$\min \left\{ \sum_{i=1}^k w_i \left(\bar{Z}_i - (C_i)^T X \right) + \sum_{j=1}^n \epsilon_j \left(\bar{Z}_j - (C_j)^T X \right) \right\}$$

$$i = 1, 2, \dots, k$$

$$\text{kısıtlar: } A X \leq b \quad (2.98)$$

$$X \geq 0$$

DP problemine dönüşür. Burada $\bar{Z}$ referans noktası ve w ağırlıklandırma vektörüdür. Amaç fonksiyonundaki ikinci terim ϵ_i küçük, pozitif değerleriyle etkinliği garantileyen artık terimlerdir.

Karar verici, (2.98) probleminde, bir derece serbestliğe sahiptir referans vektör veya Z ideal vektörü referans noktası seçilebilir ve ağırlandırma vektörü w , ideal vektörün elemanlarında istenmeyen sapmayı açıklamak için kullanılabilir [5]. Ideal vektör referans noktası seçildiğinde aşağıdaki teorem ispatlanabilir [75].

2.8.1.4 Teorem : Uygun bir $\bar{X}$ noktası etkin ve $(C) \bar{X} < Z^*$
 $i=1,2,\dots,k \implies \bar{X}$, bazı pozitif w_i ve bazı kesin pozitif ϵ_j
 değerleri için 2.98 probleminin çözümüdür.

Bu teorem, tüm etkin çözümlerin herhangi amaç değerleri ideal değerlerine eşit olamaz demektir. Ek olarak eğer $\bar{X}_i$, $i=1,2,\dots,k$ (i . amaç fonksiyonunu optimize eden çözüm) tek ise, tüm çözümler $w \geq 0$ ağırlıklarıyla elde edilebilir.

Bu ölçekleme fonksiyonu Steuer ve Choo'nun Weighted augmented Tchebycheff, Nahayama ve Savaragi'nin sağlayan bölüşümler [76], Kok ve Lootsma'nın ikili karşılaştırmalar [77] yöntemlerinde kullanılmıştır.

4- Başarı ölçekleme fonksiyonu (Achievement Scalarizing Function)

$\bar{q} \in R^k$ kriter uzayında bir referans noktası olsun. Referans noktası uygun çözüm bölgesinin bir elemanı olabilir veya olmayabilir. w ağırlandırma vektörü $\wedge$ ağırlıklandırma uzayından olmak üzere başarı ölçekleme fonksiyonu $S(\bar{q}, \bar{Z}, w)$, $\bar{q}$ referans noktasını baskın kriter vektörleri kümesi (N) üzerine izdüşürür. N üzerinde izdüşürülen nokta $S(\bar{q}, \bar{Z}, w)$ fonksiyonunun endüşük değerli düzey eğrisinin kestiği an, Z 'ye tanjant olduğu noktadır.

Böyle bir kriter vektörü,

$$\text{enk } \{ S(\bar{q}, Z, w) \}$$

$$\text{kısıtlar: } Z \in Z$$

başarı ölçekleme programını çözerek bulunabilir. Başarı ölçekleme fonksiyonu,

$$S(\bar{q}, Z, w) = \max_i \{ w_i (\bar{q}_i - Z_i) \} - \epsilon \sum_{i=1}^k Z_i \quad (2.99)$$

$$i=1, 2, \dots, k$$

ve burada ϵ çok küçük pozitif değerdir. Bu fonksiyon ile başarı ölçekleme programı,

$$\text{enk } \{ \alpha - \epsilon \sum_{i=1}^k Z_i \}$$

$$\text{kısıtlar: } \alpha \geq w_i (\bar{q}_i - Z_i) \quad 1 \leq i \leq k \quad (2.100)$$

$$f(x) = Z \quad 1 \leq i \leq k$$

$$X \in S$$

(2.100) probleminde $\alpha \in R$ ve $Z \in R$ ve genellikle işaret kısıtı yoktur.

(2.100) Başarı ölçekleme program formülasyonu yeniden düzenlenirse,

$$\min \{ \alpha - \epsilon \sum_{i=1}^k Z_i \} \quad (2.101)$$

Kısıtlar:

$$Z_i + \frac{\alpha}{w_i} \geq \bar{q}_i \quad 1 \leq i \leq k$$

$$f(X) = Z \quad 1 \leq i \leq k$$

$$X \in S$$

$\bar{q}$ referans noktasından d yönünde çizgi parçası, N baskın kriter

vektörleri üzerine izdüşürmek için başarı ölçekleme parametrik programı,

$$\text{enk } \left\{ a - \epsilon \sum_{i=1}^k Z_i \right\} \quad (2.102)$$

$$\text{kısıtlar } Z_i + \frac{a}{w_i} > \bar{q} + \theta d_i \quad 1 \leq i \leq k$$

$$f(X) = Z_i \quad 1 \leq i \leq k$$

$$X \in S$$

çözülür.

2.8.2 Sağ Taraf Parametrik Programlama

Gerçeğe uygun programlama modellerinin kurulmasında önemli adımlardan biri katsayılar için doğru ve güvenli sayısal değerleri toplamaktır. Bazı durumlarda katsayıların yalnız ortalama değerleri veya tahminleri bulunabilir. Bu nedenle problemin katsayılarının değişimi sözkonusu olsa, acaba doğrusal programlama probleminin çözümünün davranışları nasıl olur? Yani problem katsayıları bir θ parametresinin doğrusal fonksiyonuna bağlı değişse hangi aralıklarda problemin çözümü hala optimaldir. Bu kısımda, sadece kısıtların sağ taraf sabitlerinin bir θ parametresinin doğrusal fonksiyonu olması durumunda çözümün uygunluk şartlarını araştıracağız [78].

Sağ taraf parametrik problemin matematiksel tanımı :

$$\sigma < \theta < q \quad \text{aralığında her } \theta \text{ için}$$

$$\sum_{j=1}^n a_{ij} x_j = b_i + \theta b'_i \quad i = 1, 2, \dots, m \quad (2.103)$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

kısıtları altında $\sum_{j=1}^n C_j X_j$ toplamını enküçükleyen $X=(x_1, x_2, \dots, x_n)$

vektörünü bulmaktır. $\theta = \sigma$ iken enküçük uygun çözüm

$X^0 = (x_{10}, x_{20}, \dots, x_{n0})$ olduğunu varsayalım. Her x_{10}, b_{10} parametresinin bir kombinasyonu biçimde yazılabilir:

$$x_{10} = q_i + \theta p_i \geq 0 \quad \text{ve buradan} \quad x_{10} = q_i + \theta p_i \geq 0 \quad (2.104)$$

eşitsizlikleri tutarlıdır. Tüm $p_i = 0$ değerleri için enküçük

uygun çözümdür. Eger tüm $p_i \geq 0$ ise X^0 çözümü tüm $\theta \geq \sigma$ için

enk uygun çözümdür. Her $p_i \leq 0$ ise X^0 çözümü $\theta < 0$ için de enk

uygun çözümdür. p_i değerleri pozitif ya da negatif olabilirler.

p_i değerlerinin pozitif olması durumunda θ parametresinin hangi

aralıkları için X^0 çözümünün enk olduğuna bakalım.

$$\theta \geq - \frac{q_i}{p_i} \quad (2.105)$$

θ parametresi için alt sınır,

$$\theta = \begin{cases} \text{enb} - \frac{q_i}{p_i} & \text{her } p_i > 0 \text{ ise} \\ -\infty & \text{her } p_i \leq 0 \text{ ise} \end{cases} \quad (2.106)$$

θ parametresi için üst sınır,

$$\theta = \begin{cases} \text{enk} - \frac{q_i}{p_i} & \text{her } p_i < 0 \text{ ise} \\ +\infty & \text{her } p_i \geq 0 \text{ ise} \end{cases} \quad (2.107)$$

2.104 çözümü $\theta < \bar{\theta} < \theta$ aralığı için minumdur. θ parametresinin üst sınırı ise $+\infty$ değilse, θ parametresi arttıkça çözüm optimal kalır. Yani her $Z_j - C_j \leq 0$, ancak uygun olmak zorunda değildir.

θ parametresinin yeterince büyük artışlarında değerlerden biri, $X_{io} = q_i + \theta p_i$ negatif olabilecektir. Eğer X_{io} negatife giden değerlerden ilki ise $P_1 \leq 0$ olmak üzere,

$$\bar{\theta} = - \frac{q_1}{p_1} \quad (2.109)$$

$\theta > \bar{\theta}$ aralığı için yeni enk uygun çözüm bulunmalıdır. Temele öyle vektörler girmeli ve çıkmalı ki sağ taraf elemanları negatif olmayan eşitliklere dönüşmeli $Z_j - C_j$ elemanları ise negatif kalmalıdır.

2.8.3 Algoritma

Aşağıda verilen problemi dikkate alalım:

enb $U(z)$

kısıtlar: $z \in Z = \{ f(x) | x \in S \}$.

burada $f(x) = (f_1(x), f_2(x), \dots, f_k(x))$ amaç fonksiyonlarının k

boyutlu vektörü, U karar vericinin bilinmeyen fayda fonksiyonu,

x karar değişkenleri vektörü ve S uygun kararlar kümesidir.

Genelliği kaybetmeden tüm amaç fonksiyonlarının enbüyükleneceğini varsayalım. Z^* etkin vektörler kümesi,

$$Z^* = \{ z | z \in Z \text{ ve } z \text{ etkin} \} \quad (2.110)$$

biçiminde tanımlansın. Algoritmanın adımları :

(a)

1. Adım : Kriter uzayında Z bir başlangıç noktasını al ve adım sayacı $h=1$ olsun.

(o)
Z vektörü etkin ve hatta uygun (feasible) olmak zorunda da değildir. Çünkü 3. adımda nasıl olsa etkin sınıra izdüşürülecektir.

2. Adım : Bir $g^{(h)}$ vektörü al ve $d^{(h)} = g^{(h)} - z^{(h-1)}$ yeni referans vektörü olsun.

$g^{(h)}$ vektörü Wierzbicki'nin referans noktası yaklaşımına uyarak referans hedefler seçilebilir. Başka bir yolda sınır noktası sıralama yöntemi olabilir. Geoffrion ve ark. [33] önerdiği karar vericinin marjinal yerine koyma oranını tahmin eden referans yönleri de kullanılabilir. Wallenius [53] bu oranların hesaplanmasının zor olduğunu rapor etmiştir.

3. Adım : Z etkin vektörler kümesini bulmak için başarı ölçekleme parametrik programlama problemi

$$\text{enk } \{ \alpha - \epsilon \sum_{i=1}^k Z_i \} \quad (2.111)$$

$$\begin{aligned} \text{kısıtlar : } & Z_i + \frac{\alpha}{w_i} < Z^{(h-1)} + \theta d^{(h)} \quad 1 \leq i \leq k \\ & f(x) = Z_i \quad 1 \leq i \leq k \\ & x \in S \end{aligned}$$

çözülür.

Parametrik sağ taraf programlama ile $z^{(h)}$ noktasından çıkan ve etkin kümeden geçen eğriler elde edilir. Elde edilen eğri ekranda karar vericiye sunulur.

4. Adım : Karar verici $z^{(h)}$ içinde tercih ettiği $z^{(h)}$ kriter vektörünü belirtir.

$(z^{(h)}, d^{(h)})$ yönünde, $z^{(h)}$ noktasından çıkan doğrunun uygun çözüm bölgesinde, baskın çözüm vektörleri kesikli

bir doğru parçası oluşturur. Korhonen ve Laakso bilgisayar grafiklerinden yararlanarak bu doğruyu ekranda görüntülemişlerdir. Ekranın alt kısmında körsir sağa sola hareket ettikçe üstte amaç fonksiyonlarının değerleri verilmektedir [71]. Bu aşamada karar vericiden doğru parçası üzerinde tercih ettiği kriter vektörü istenir.

5. Adım : Eger $Z^{(h-1)} \neq Z^{(h)}$ ise $h = h+1$ al ve 2. adıma dön. Aksi halde optimum şartlarını kontrol et. Sağlıyorsa $Z^{(h)}$ optimum çözümdür, dur. Diğer durumda $h=h+1$ al, yeni d araştırma yönü ile ikinci adıma dön.

2.9 Diğer Etkileşimli Yöntemler

Yöneylem araştırmasının en hızlı gelişen bu alanında algoritmaların tümünü bu çalışmada toplamak çok zaman alıcı ve hacimli olacağı açıktır. Bu nedenle yayınlarda sıkça geçen önemli algoritmaları verdik. Ancak yine de diğer algoritmaların bazı önemli noktalarını vurgulamadan geçemeyeceğiz.

SWT (Surrogate Wort Trade-off) yöntemi, Haimes, Hall ve Freedman [5,79] tarafından önerilmiştir. Karmaşık bir algoritmadır. Diğer amaçlar aynı kalmak üzere, i. amaçtaki bir birim değişime karşılık j. amactaki değişim dikkate alınarak karar vericinin vekil değer fonksiyonu bWb değerleri bulunur. Eger $W_{ij} > 0$ ise bölüşüm rededilir. Bölüşüm oranları tüm amaçlar için bölüşüm oranları,

$W_{ij} = - \frac{df_i(x)}{df_j(x)}$ ilişkisiyle bulunur. Algoritma, tüm vekil değer fonksiyonları sıfır olduğunda son çözüme ulaşmış demektir.

Basit deneme-yanılma yapısal olmayan yöntemde, karar vericiden, her ardıştırmada, kriter vektörü değerleri alınır. Algoritma, seçimin uygun olup olmadığını, kontrol eder. Bu yolla kriter uzayı keşfedilmeye çalışılır. Daha çok tercih edilen değerler bulununcaya kadar işlem sürer [4].

Nakayama ve Sawaragi'nin [76,77] Sağlayan bölüşümler yönteminde, ağırlıklı Tchebychff ölçekleme fonksiyonundaki ağırlıklar her ardıştırmada değişir.

w_i değerleri, $\bar{f}_i$ amaçların sağlama düzeyleri ve $\bar{f}_i^* < f_i^*$ olmak

üzere $w_i = 1 / (f_i^* - \bar{f}_i)$, $i=1,2,...,k$ ilişkisi ile hesaplanır.

Karar verici $\bar{f}_i$ değerlerini değiştirerek hangi amaç sağlama düzeyleri yükseltilecek, hangileri gevsetilecek, belirler. Sabit tutulanlar haric diğer değişim miktarları alınarak,

$$Y > w_i (f_i^* - (C_i)^T X) \quad i=1,...,k \quad (2.119)$$

her ardıştırmada kısıtlar ile ilgili dual değişkenler Y ile enküçüklenir. Ardıştırma amaçlardan hiçbiri geliştirilemediğinde sonlanır.

Kok ve Lootsma'nın [70] ikili karşılaştırmalar yönteminde yukarıdaki w_i ağırlıkları geleneksel karşılaştırmalar yoluyla bulunur. İdeal ve nadir vektör belirlenerek bir amacın belli oranda, örneğin % 10, değişimine karşılık başka bir amaçta aynı oranda değişim olup olmadığı hesaplanır. Eğer değilse, soru değiştirilerek karar vericiden hangi amaç fonksiyonunu tercih ettiği sorulur. Bu soruşturmalar ışığında w ağırlıklar vektörü tutarlı değilse logaritmik regresyon uygulanır. Buna göre etkin bir çözüm bulunur. Çözüm kabul edilmezse amaçların alt sınırları

belirlenerek başa dönülür. Sağlayan bir çözüm bulununcaya kadar algoritma çalıştırılır.

Hedef (amaç) programlaması etkileşimli yöntemleri de yayınlarda önemli yer tutmaktadır.

Franz'ın IGP (Interactive goal programming) tekniği, karar vericinin ilerleyen tercihinine dayanır. Etkileşim, amaçlar için sağlama düzeylerini belirleme, sapma değişkenlerine uygun ağırlık atama, hedef başarılarını sıralama gibi alanlarda istenir.

Model,

$$\text{enk } \sum_{j=1}^k d_j$$

kısıtlar :

$$g_i(x) \leq 0$$

$$f_1(x) + d_1 = a_1$$

$$f_2(x) + d_2 = a_2$$

$$\vdots$$

$$\vdots$$

$$f_k(x) + d_k = a_k$$

Burada $a_1, a_2, \dots, a_k$, ideale karşılık gelen sağlama düzeyleri $d_1, d_2, \dots, d_k$ sapmalara karşılık gelir. Modelin tek varsayımı; amaçların ulaşılmamış değerleri sağlama düzeylerini aşamaz. Franz yönteminde, amaçlar arasındaki bölüşüm değerleri dikkate alınarak sıralama yapılır. En önemli amaçtan başlayarak faydası en büyük * x ciyarında bir çözüme ulaşıncaya kadar ardıştırmaya devam edilir [80].

Etkileşimli çok amaçlı hedef programlaması [74,39] kısıt yöntemini kullanır. Her adımda ideal ve nadir vektör

hesaplanarak karar vericiye sunulur. Karar verici hangi amaç fonksiyonlarının eşik değerlerinden yüksek olduğuna bakarak alt sınır vektörünün elemanlarını güncelleştirir. Yeniden ideal ve nadir vektörler hesaplanır. Karar vericinin kabul edeceği bir çözüme ulaşıncaya kadar ardıştırmaya sürdürülür.

2.10 Etkileşimli Yöntemlerin Karşılaştırılması İçin

Yapılan Çalışmalar

Etkileşimli yöntemlerin sistematik karşılaştırması [52,53,39,55,80] nolu çalışmalarda yapılmıştır. Ancak her çalışmada birkaç yöntem üzerinde durulmuştur. ÇAKV etkileşimli yöntemlerinin tümünü dikkate alan en geniş perspektiften bir inceleme yapılmamıştır.

İlk çalışma Wallenius tarafından 18 yönetim bilimleri öğrencisi ve aynı sayıda karar verici ile 3 amaç, 19 kısıt, 25 karar değişkenli problem kullanılarak yapıldı. Araştırmada GDF, STEM ve basit deneme-yanılma algoritmaları parametre dışı, Friedman iki yönlü varyans analizi, Wilcoxon birleştirilmiş çiftler işaret sıralaması gibi istatistik teknikleri yardımıyla karşılaştırıldı.

Değerlendirme kriterleri :

- 1- Karar vericinin uzlaşık çözümde güvenliği (ordinal sıralama),
- 2- Yöntemin kullanım kolaylığı (ordinal sıralama),
- 3- Adımların ve yöntemin mantığının anlaşılabilirliği (ordinal sıralama),
- 4- Karar vericiye sunulan bilginin yararlılığı (ordinal sıralama),

5- Yakınsama hızı, ardıştırma sayısı ve problemi çözmek için harcanan toplam zamanla ölçüldü [53].

Bu çalışmaya göre basit deneme-yanılma yöntemi, STEM ve GDF'e göre daha üstün görünmektedir.

Brockhoff [52] Stem ile Steuer'in algoritmasının karşılaştırma deneyimi çeşitli bölümlerden seçilen 142 öğrenci ile yaptı. Problemler değişik boyutta, tesadüfi seçildi. Yedi hipotez, Mann-Witney U testiyle test edildi. Sonuçlar gerek hipotezler ve gerekse kullanıcı davranışları yönünden çok farklı çıktı. Bu nedenle araştırmacı sonuç için bu çalışmanın yeterli olmadığını daha geniş çaplı araştırmaların gerekliliğini vurgulamaktadır.

Kok [39], Z-W, Etkileşimli çok amaçlı hedef programlaması, ikili karşılaştırmalar, ağırlıklı arttırılmış Tchebycheff yöntemlerini uzun dönemli stratejik planlama probleminde kullandı. Çıkardığı sonuçlar çok kısa biçimde şöyle sıralanabilir: Klasik ağırlandırma, ÇADP için geçerli değildir. ECAHP ve AAT yöntemleri yüksek bilgisayar fiyatlıdır. IK, diğerlerine göre, karar vericinin bölüşümlerini öğrenmesi bakımından daha zayıftır. AAT ve Z-W algoritmalarında yüklenilen bilgi fazladır.

Bu çalışmada da en uygun yöntemin hangisi olduğu belirtilememektedir.

3.BÖLÜM

ÇOK AMAÇLI BİR KARAR DESTEKLEME SİSTEMİ

3.1 Giriş

Çok amaçlı karar verme problemlerine klasik yaklaşım, yönetici ile etkileşerek bir yarar fonksiyonu kurma esasına dayanır. Yarar fonksiyonu kurulabilirse problem bir optimizasyon problemine dönüşür. Bazı araştırmacılar yarar fonksiyonu kurma aksiyomlarının belirli durumlarda geçerli olmadığını, bazı durumlarda ise karar vericinin çelişen amaçlar arası değiş-tokuşları, problemi çözmeden belirlemede güçlük çektikleri gözönünde bulundurarak, karar vericilere tüm baskın çözümleri vermeyi önermişlerdir. Böylece karar verici yarar fonksiyonunu açık olarak ifade etmeden optimumu bulma imkanına kavuşmaktadır.

Bu alanda ilk yapılan araştırmalar genellikle yöneticiye tüm baskın çözümleri verir, yönetici bunlardan kendi yarar fonksiyonuna en uygun olanını seçer felsefesiyle yapılmıştır. Son zamanlarda ise araştırmacıların çoğu, yöneticinin yarar fonksiyonuna uyan etkin çözümü bulmasında destek olmayı savunmaktadır. Bu aşamada bilgisayarın aracısız olarak karar verici ile etkileşmesi hem optimal çözümü bulma sürecini kısaltacak hem de bu süreci daha sağlıklı kılacaktır [81]. Yeni çok amaçlı karar destekleme sistemi (ÇAKDS) bilgisayar ile karar verici arasında doğrudan etkileşimi sağlamaktadır.

3.2 ÇAKV ve Karar Destekleme Sistemleri

ÇAKV için her ne kadar birçok algoritma ve program üretilmiş ise de kullanıcının kullanım yüzüne (interface) pek az önem verilmiştir. Halbuki kullanım yüzünün kalitesi, ÇAKV yöntemlerinin kabul edilirligini etkileyen baş faktörlerden birisidir. Zayıf düzenlenmiş kullanım yüzü genellikle yöntemin kabülünü önler. Bugünkü bilgisayar teknolojisi, sistem yerine, karar vericinin bakış açısından karar problemlerinin çözümünü destekleyen kullanım yüzüne ilişkin yazılım (software) paketlerinin planlanlanmasına izin vermektedir [82,83].

Weber'in [84] işaret ettiği gibi, karar destekleme sistemleri [KDS], kavramı son on yıldan beri var. KDS geliştirildiği yıllardan beri pek çok işletmede kişisel ve değişik organizasyon seviyelerinde başarıyla uygulandı. Karar destekleme için değişik tanımlamalar sunuldu. Ginzberg ve Stohr [85] bu kavramı şu şekilde tanımlar : Tüm karar sürecini mümkün olmayan ya da istenmeyen makinalaşmış sistem icrası yerine, karar verme hareketlerini desteklemede kullanılan bilgisayar tabanlı bilgi sistemidir. Çokamaçlı karar destekleme sistemi (ÇAKDS) ise Jelassi ve ark. [86] tarafından geniş KDS ailesi içinde "özel tip" bir sistem olarak dikkate alındı. Hatta ÇAKDS, geleneksel KDS öğelerinin çoğunu aynen kullanılır.

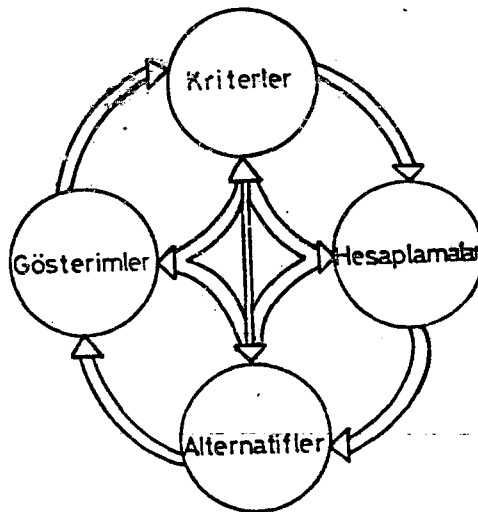
KDS yöntembilimi üç safhayı birleştirir : (1) KDS ekran yazılımı; (2) Hesaplama Üretici; (3) özel KDS planı [87].

Yeni etkileşimli KDS üzerinde durulan nokta, karar verme sürecini daha iyi tanımsal anlamayı gerektirir. İnsanlar kararlarını "Üretirken" nasıl başlarlar? Şimdi önerilen karar

verme sürecinin hiçbir boyutu ve yz nceden sabit olmamalı; kriterler, alternatifler, kısıtlar, lmler, hesaplamalar ve gsterimlerin tm deęisebilmeli ve tm modelde uyum ve denge iin srekli yeniden formle edilebilmelidir [88].

Bir karar; tm karar birimlerini dengeleyen, " uyumlu " model olarak tanımlanmakta ve kabul edilmektedir. Nevralji ve onunla birleşik psikolojik veriler gstermektedir ki yalnız basmakalıp rasyonel akıl kesin yeterli deęildir.

Karar destekleme sistemlerinin geliřimi ve ilerlemesi ile karar verici, karar şebekesinin (network) tekrarlı yeniden formlasyonu boyunca, gerek karar retme srecinden uygun biimde yararlanacaktır. İnsanoęlu bilgi ve kararları oluřturur veya kurar. Karar vermenin tm nemli paraları; kriterler, alternatifler, gsterimler ve hesaplamalar birbirine baęlı, birleşik belirlenir ve yapıcı karřılıklı dzeltme akıřında tutulur. Hibirsey nceden, sabit deęildir. Şekil 10'de enkk karar retim şebekesinin paraları ve baęlantıları gsterilmektedir.



Şekil 10. Karar retim şebekesi

3.3 Yeni Çok Amaçlı Karar Destekleme Sistemi

Yeni çok amaçlı karar destekleme sistemi, Zeleny'nin de belirttiği gibi amaçları, kısıtları, değişkenleri, hesaplamaları birbiriyle ilintili olarak değişime, yeniden formülasyona izin verebilecek yapıda kurulmuştur.

Yeni çok amaçlı karar destekleme sisteminde, karar verme süreci ile ilgili bir dizi varsayım yapılmıştır.

1- Karar problemlerinde amaç sayısı en fazla 9 seçilmelidir. Her çözüm bir vektör olduğundan hazırlanan bilgi genellikle büyük miktardadır. Bu nedenle üretilen bilginin kavranması zordur.

Problem kümesinde amaç sayısı bir karar vericinin ilgilenebileceği, gerçeğe uygun sayıya dayanmalıdır. George Miller'in meşhur iddiasına göre, insan hafızasının kısa vadeli kapasitesi yaklaşık olarak yedi giriş ile sınırlıdır. Bu nedenle ÇAKV problemleri için kabul edilebilir amaç kümesi 7 ± 2 olabilir [25,89].

2- Karar verici, amaçlar üzerinde bir dizi kanaate sahip olsa da, kişisel tercih fonksiyonu ya da fayda fonksiyonu analitik olarak açıklanamaz [27,22,1,64,87,90].

3- Sağlama düzeyleri ve istekler, öğrenme ve deney sonucu değişebilir. Değer yargıları zaman içinde değişik oranlarda değişir [64,90]. Karar verici, çözümleri, gördükçe amaçlar arası değiş-tokuşları dinamik olarak gerçekleştirebilir.

4- Karar probleminin çözümü herhangi kabul edilebilir hareket yönüdür [90].

Yeni ÇAKDS, dinamik, görsel etkileşimli bir algoritmadır.

Yeni ÇAKDS ile karar verici, ÇADP probleminin etkin kümesinde hız ve yönü kontrol ederek serbestçe dolaşabilir. Etkin kenar ve yüzeylerde gezdikçe amaç, fonksiyon ve değerleri, sayısal ve dikdörtgen grafiklerle görüntülenir. Etkin küme üzerinde dolaşım da klavyeden kontrol edilebilmektedir. Sabit hız ya da istenen adım büyüklükleriyle geriye, ileriye hareket etmek de mümkündür. Yeni ÇAKDS ile yönetici, baskın çözümlerden birini bulup, bundan sonra yeterli bulmadığı bir amacı, istediği yönde değiştirebilmekte ve bu değişimin diğer amaçlar üzerindeki etkisini anında grafiklerle görebilmektedir. Böylece karar verici amaçlar üzerindeki fayda fonksiyonunu açık biçimde ifade etmeden kendisi için optimal çözümü bulmaktadır.

3.4 Yeni Çokamaçlı Karar Destekleme Sisteminin Elemanları

Sistemin başlıca dört ana fonksiyonu vardır :

- 1- Model yönetim fonksiyonu
- 2- Model geliştirme fonksiyonu
- 3- Problem çözüm fonksiyonu
- 4- Çözüm çıktı fonksiyonu.

Her ana fonksiyon menüden seçilen bir veya daha fazla altfonksiyondan oluşmaktadır. Her aşamada seçimlikler görüntülenmekte ve belli aralıkta girilen tamsayılar ile seçilmektedir. Kısaca her ana fonksiyonu açıklayalım :

1- Model yönetim fonksiyonu : " Yeni Model Girişi ", " Disk(et)ten Giriş " ve " Model Verilerini Saklama " seçimliklerinden oluşmaktadır.

İlk defa model, " Yeni Model Girişi " seçilerek kurulur. Yeni model girişi seçildiğinde, modele verilecek ad istenir. Girilen ad altında model kurumu, üç aşamada gerçekleşir. Her aşamanın kendi özel tabloları vardır. İlk tabloda amaç sayısı, kısıt sayısı ve değişken sayısı alınır. Değişken adları ilgili değişkeni hatırlatacak üç karakterlik bir kısaltma ile temsil edilebileceği gibi programa da bırakılabilir. Eğer programa bırakılırsa otomatik olarak $x_1, x_2, \dots, x_n$ biçiminde seçilmiş olur.

İkinci ve üçüncü tablolar sırasıyla kısıt ve amaç sayısınınca tekrarlanır. Amaç ve kısıt katsayıları girildiğinde körsür bir sonraki katsayı pozisyonunda görüntülenir. Katsayıları kontrol ve düzeltme her amaç fonksiyonu veya kısıt girişi tamamlandığında tanındığı gibi, model girişi bittiğinde de seçilebilmektedir. Kısıt ilişki operatörleri otomatik olarak $>$ biçimindedir. İlişki işareti farklı kısıtlar için gerekli düzeltme, girişte yapılır.

Model verilerini tekrar kullanmak için menüden, "Model verilerini Saklama" alt fonksiyonu seçilmelidir. İsteğe bağlı olarak, model, eski veya yeni bir isimle istenen yardımcı hafıza ünitesinde saklanır.

Önceden, birinci alt fonksiyon yardımıyla oluşturulmuş bir model "Disk(et)ten Giriş" seçimliğiyle, tekrar kullanılabilir.

2- Model Geliştirme Fonksiyonu : " Model Düzeltme-Değiştirme " ve " Model Verilerini Görüntüleme " alt fonksiyonlarından oluşur. Yeni model girişi seçimliğindeki edit ve kontrol işlemlerine ek olarak, bu alt fonksiyonlar ile sisteme, yeni amaç(lar) ekleme, modelden amaç(lar) çıkartma

mümkün olmaktadır. Aynı işlemler kısıtlar için de sözkonusudur.

3- Model çözüm Fonksiyonu : " Model çözümü " alt fonksiyonundan oluşur. Problem çözümü birkaç program ile gerçekleştirilir.

Çok amaçlı modeller için ilk aşamada karar vericinin amaçlar için düşündüğü sağlama düzeyleri (aspiration levels) istenir. Weirzbecki'nin başarı ölçekleme fonksiyonu (achievement scalarizing function) ile baskın çözümlerin etkin sınırına (efficient frontier) izdüşüm yapılır. Sağlama düzeyleri uygun çözüm bölgesinde olsun veya olmasın, bu nokta etkin küme üzerine izdüşürülür. Amaçların gerçekleşme değerleri ve göreceli büyüklükleri dikdörtgen barlar ile ekranda görüntülenir. Bu aşamada 1- Sabit Hız, 2- İleriye Doğru Hız, 3- Geriye Doğru Hız 4- Katsayı Değiştirme, 5- Yön Değiştirme, 6- Çözüm Vektörü, 7- Otomatik yön değiştirme 8- Ana Menüye Dönüş seçimliklerinden yön değiştirme seçilerek amaçlar arası sağlama düzeyleri bölüşümleri yapılır. Gerekirse parametre değiştirerek, sabit veya istenen hızla etkin kümede, ileriye geriye, istenilen biçimde dolaşılır.

Etkin kümede gezintiyi düzenleyen üç katsayı vardır :

1- Başarı ölçekleme sağ taraf parametrik programlama da adım büyüklüğü t , başlangıçta $t = 0.0001$ seçilmiştir. Amaç fonksiyonlarının aldığı değerler, ardışık iki iterasyonda önemli derecede değişmezse dört seçiciliğiyle arttırılabilir.

2- Hızlandırma katsayısı k , etkin küme gezintisinde adımları büyütme katsayısıdır. Başlangıçta $k=3$ seçilmiştir. Dört seçiciliği ile, isteğe bağlı olarak değiştirilebilir.

3- Otomatik yön deęiřtirme katsayısı β , bařlangıçta $\beta = 0.4$ seçilmiřtir. Çözüm bulunduğdan sonra bu noktanın çevresinde amaçların alacağı deęerler otomatik yön deęiřtirme seçimlilięiyle görülebilir.

Etkin kümede gezintiyi düzenleyen tuřlar ve fonksiyonları :

1 : "Sabit Hız"

Bulunulan yönde sabit adımla devam eder, $\theta_t = \theta_t + 0.0001$.

2 : "İleriye Doğru Hız"

Adım büyüklüğünü k ve t (temel hız) katsayılarına göre büyültür. İleriye doğru adım büyüklüğü $\theta_t = \theta_t + k \cdot t$ olur.

3 : "Geriye Doğru Hız"

Adım büyüklüğünü k ve t katsayılarına göre küçültür. Geriye doğru adım büyüklüğü $\theta_t = \theta_t - k \cdot t$ olur. $\theta_t < 0$ olduğunda $\theta_t = t$ alınır.

4 : "Katsayı Deęiřtirme"

Programda kullanılan katsayılar t, k ve β isteęe baęlı olarak deęiřtirilebilir.

5 : "Yön Deęiřtirme"

Bulunulan andaki amaç fonksiyon deęerleri görüntülenerek yeni saęlama düzeyleri ile yön, $D(I) = G(I) - Z(I)$ $I=1, \dots, \text{amac sayısı}$ seçilir.

6 : "Çözüm Vektörü"

Karar deęiřkenlerinin $(x_1, x_2, \dots, x_n)$ ve sabit sisteme eklenen dięer deęiřkenlerin deęerleri görüntülenir.

7 : "Otomatik Yön Deęiřtirme"

Saęlama düzeylerine göre yön seçiminden farklı olarak istenirse yön deęiřtirme katsayısı kullanılarak yön seçilebilir.

$$D(I) = D(I) + \beta \quad I=1, 2, \dots, \text{amac sayısı}$$

Programda kullanılan çözüm algoritması :

1. Adım : Kriter uzayında $Z = (0,0,...,0)$ noktası alınır.

Adım sayacı $h=1$ olsun.

2. Adım : Yön vektörünü belirlemek için bir $g^{(h)}$ vektörü seçilir. Böylece yön,
 $d^{(h)} = g^{(h)} - Z^{(h-1)}$ olur. $g^{(h)}$ vektörü elemanları amaçların
sağlama düzeyleridir.

3. Adım : $Z^{(h)}$ etkin vektörler kümesini bulmak için başarı
ölçekleme adım büyüklüğü parametrik programlama problemi,

$$\text{enk } \{ \alpha - \epsilon \sum_{i=1}^k Z_i \}$$

Kısıtlar :

$$Z_i + \frac{\alpha}{w_i} \leq Z^{(h-1)} + \theta d^{(h)} \quad 1 \leq i \leq k$$

$$f_i(x) = Z_i \quad 1 \leq i \leq k$$
$$x \in S$$

$\theta \rightarrow 0$ iken çözülür. Bu çözümle amaç fonksiyonunun aldığı
değerler sayısal olarak ve dikdörtgen grafikler ile ekranda
görüntülenir. Etkin kümede dolaşımı sağlayan fonksiyonlar bu
aşamada kullanılabilir.

Eğer $h = 1$ ise 4. Adımdan devam edilir.

Uygun alt fonksiyonlar ile etkin kümede gezerek tercih
edeceği $Z^{(h)}$ bulunur.

4. Adım : Eğer $Z^{(h)} \neq Z^{(h-1)}$ ise ikinci adıma dönülür.

Aksi halde karar vericiyi sağlayan optimum çözüm bulunmuştur.
Dur.

Bu algoritmada Korhonen ve Laakso'nun etkileşimli algoritmasında olduğu gibi sağ taraf parametrik programlama ile tüm değişim aralığı belirlenmeyip, farklı olarak bu eğriler üzerinde istenen adım büyüklüğü ile gezilerek karar vericinin tercih ettiği $Z^{(h)}$ kümesi belirlenmektedir. Bu ise araştırılan çözüme daha çabuk ulaşmayı sağlayacaktır.

4- Çözüm Çıktı Fonksiyonu : Amaç fonksiyonları ve karar değişkenlerinin değerleri raporda kullanılmak üzere ekranda ve kağıtta çoğaltılmak üzere yardımcı hafıza ünitelerinde saklanabilir.

4.BÖLÜM

KALİTELİ KANATLI YEMLERİ İÇİN BİR UYGULAMA

4.1 Giriş

Dünya nüfusunun özellikle ülkemiz nüfusunun hızla artmasına bağlı olarak yeterli ve dengeli beslenme için gıda maddelerine olan talep her geçen gün artmaktadır [91].

Türkiyenin coğrafi, jeolojik ve iklim durumu çok değişik özelliklere sahiptir. Bu nedenle de ülkemizde bitkisel ve hayvansal olarak çok çeşitli ürünlerin elde edilmesine uygun bir ortam vardır. Buna karşılık, vatandaşlarımızın beslenmesi, ileri ülkelerdeki aksine, hayvansal ürünlerden daha çok bitkisel ürünlere dayanmaktadır. Halbuki, planlı ve programlı hareket edilirse, gerek besleme değeri ve gerekse ekonomik değeri çok daha yüksek bulunan ürünlerin üretimlerinin oldukça arttırılması mümkündür [92].

Hayvancılıkta karlılığın temeli yüksek verim elde etmeye dayanır. Hayvanlardan yüksek verim elde etmenin temel iki ana kuralından biri, verim gücü yüksek hayvanlar kullanmak diğeri ise, en uygun bakım ve besleme sağlamaktır. Bu ikisinden herhangi biri yerine getirilmedikçe beklenen verim düzeyine ulaşamaz.

Hayvanları sadece kaba yemlerle beslemek doğru değildir. Aksine bunların yüksek verimlerini sağlamak için kaba yemleri azaltıp, mümkün olduğunca fazla kesif yem vermek gerektir [93]. Hayvanları yeterli ve dengeli beslemek için ihtiyaç duydukları protein, karbonhidrat, yağ, vitaminler ve mineraller gibi besin öğelerini uygun oranlarda içeren yem karışımları [94] yedirilmelidir.

Hayvan beslemede, yem en önemli girdiyi oluşturmaktadır. Bununla önemli kısmını, hatta tavukçulukta tamamını, diğer hayvancılık dallarında ise yarısından fazlasını karma yemler oluşturmaktadır. Bu itibarla hayvancılığın gelişmesi ve hayvansal ürün üretiminin arttırılması diğer faktörlerin yanında karma yem endüstrisi ile yakından ilgilidir.

Yem sanayine ilk adım bu yüz yılın başlarında Amerika'da 1916 yılında, Almanya'da ise 1920 yılında çıkarılan yem kanunları ile ciddi kontrol düzenleri getirilmiştir. Daha sonra bu durum diğer ülkelerde yaygınlaşarak karma yem üretimleri artmıştır.

Ülkemizde 1955 yılından itibaren yem sanayi fabrikaları kurulmuştur [91].

Tavuk eti fiyatının diğer etlere oranla düşük bulunması ve sevilerek yenmesi tavuk eti tüketiminin giderek artmasına neden olmaktadır. Hayvancılığı ileri ülkelerde, toplam et tüketiminde olduğu gibi, tavuk eti tüketimi de oldukça fazladır. Ülkemizde ise gerek toplam et tüketimi ve gerekse bunun içinde tavuk etinin payı çok düşüktür.

Planlı kalkınma dönemlerinde tavuk eti ve yumurta üretimine büyük bir önem verilmiştir. Ayrıca son yıllarda dış satım imkanlarının artışı, bu sektörün gelişmesini hızlandırmıştır [95]. Tavukçuluğun gelişmesi için yem sanayinde buna paralel olarak geliştirilmesi gerekir. Zira üretim maliyetini etkileyen en önemli unsur yemdir.

4.2 Yem Maddelerinin Besin Değeri

Bir yemin yapısında bulunan organik ve inorganik bileşiklere genel olarak besin maddeleri denilmektedir. Besin maddeleri, organizmanın normal çalışmasını ve yaşamını sürdürmesi için gerekli ihtiyaçları karşıladıkları gibi et, süt ve yumurta gibi verimlerin de oluşumunu sağlarlar.

Hayvan vücuduna yemlerle alınması gereken besin maddeleri fizyolojik etkilerine göre kabaca üç grupta toplanabilir.

1- Vücuda lüzumlu enerjiyi sağlayan besin maddeleri: yağlar, karbonhidratlar ile proteinlerin fazlası da enerji sağlanmasında kullanılır.

2- Vücudun yapı taşını meydana getiren besin maddeleri: protein ve inorganik bileşiklerdir.

3- Vücutta normal metabolik ve fizyolojik olaylar için lüzumlu etkilil maddelerdir.

Hayatın devamı ve optimal verimleri verebilmeleri için bu besin maddelerini yemlerle almaları gerekmektedir. Bu besin maddelerinden özellikle proteinler vücuttaki özel fizyolojik fonksiyonları ve hayvansal ürünlerin yapısına girmeleri nedeniyle önemli bulunmaktadır. Hayvanlar yemlerle verilen proteini kendi özel vücut proteinlerine çevirme yeteneğine sahiptirler. Bu çevirme işlemi, rasyon proteinlerinin sindirimle en küçük yapı taşları olan aminoasitlerine kadar parçalanması ile olur. Kümes kanatlıları bu aminoasitlerin büyük bir kısmını sentezleyemediklerinden ihtiyaçların yemlerle karşılanması gerekir [96].

4.3 Karma Yemde Kalite Karakteristikleri

Bilindiği üzere kalite kontrolü için üç önemli nokta vardır. Bunlar sırasıyla hammadde, yarımamül ve mamül kontrolleridir. Yem sanayinde ise yarı mamül olmadığından kalite kontrol çalışmaları hammadde ve mamül üzerinde yoğunlaşmıştır [97].

Rasyonların kalitesi ve uygunluğu, her sınıftan kanatlıların performansı üzerinde ve her tavukçuluk faaliyetinin karlılığı üzerinde büyük bir etkiye sahiptir. Genel olarak, düşük kalitedeki yemler performansı geriletir ve üretim masraflarını arttırır [98].

Kanatlılar için hazırlanacak rasyonlarda kalite karakteristikleri dört grup altında toplanabilir.

1- Protein : Proteinlerin hayvanların beslenmesindeki önemi çok uzun zamandır bilinmektedir [99], her türlü karma yem belirli koşullarla daha yüksek protein içerdikçe o kadar değerli yem olur [92]. Kümes hayvanları özellikle tavuk ve hindiler civciv ve piliç döneminde yüksek proteine ihtiyaç duyarlar [96].

2- Metabolik enerji : Genel olarak piliçler yüksek enerjili dietler üzerinde daha yüksek performans gösterirler [98]. Yemin metabolik enerji düzeyi düştükçe yem tüketimi artmaktadır.

3- Ham Selüloz : Kanatlı hayvanların sindirim sistemlerinin güçlü olmaması nedeniyle yem rasyonlarının içerdiği bazı hammaddelerin çeperlerinde bulunan selülozun sindirilmediği belirlenmiştir [100]. Bu nedenle rasyonun protein düzeyini arttırır iken ham selüloz düşük oranda tutulmalıdır.

4- Ham kül : Ham selüloz gibi, karma yemin içeriğinde bulunan besin maddelerinin yandığında geriye kalan ve sindirilemeyen artıklardır. Karma yem oluşturulur iken ham kül oranının düşük tutulması gerekir.

4.4 Karma Yem Problemi Bir Çok Amaçlı Doğrusal Karar Problemidir

Türkiye’de çok sayıda karma yem fabrikası bulunduğu ve yanısıra hayvancılığın gerekli yem ihtiyacının da yüksek olduğu bilinmektedir. Halk arasında fenni yem, fabrika yemi gibi isimler takılan karma yemin verileceği hayvana faydalı olabilmesi için teknik şartname ve standartlarda [101] belirtilen, araştırma ve bilgi birikimi sonucu ortaya çıkmış, belirli besin değeri sınırlarını tutacak kalitede bulunmalıdır.

Karma yem teşkilinde iki önemli sorun vardır. Birincisi, gerekli sınırların sağlanması, ikincisi de maliyetin minimum olması [8].

Şimdiye kadar yem problemine minimum maliyet tek amacı ile yaklaşmıştır [8,9,10]. Gerçekte Steuer’in sucuk, salam, sosis karışımı probleminde olduğu gibi çok amaç dikkate alınmalıdır. Çünkü endüstri karma yemin seçiminde fiyat, gözönünde bulundurulması gereken tek nokta değildir. Sadece ucuz karma yem satan imalatçılar görmek mümkündür. Hem kaliteli ve pahalı ve hem de ikinci kalitede ve ucuz karma yemi yapan şirketler de görülmüştür. Fakat ikinci kalitedeki yem imalatından vazgeçmek zorunda kalmışlardır [92].

Bal [48], yem problemine amaçlara öncelik vererek hedef programlaması ile yaklaşmıştır.

(4.3) kesiminde verilen, kanatlılar için karma yemde kalite karakteristikleri de dikkate alınarak çelişen amaçlar aşağıdaki şekilde belirlenmiştir.

- 1- Minimum maliyet,
- 2- Maksimum protein,
- 3- Maksimum metabolik enerji,
- 4- Minimum ham selüloz,
- 5- Minimum ham kül.

4.5 Uygulama

Şimdiye değin, minimum maliyet tek amacıyla yaklaşılan kanatlı yem problemi yukarıda belirtilen yem kalite karakteristikleri amaç seçilerek yeni çokamaçlı karar destekleme sistemi ile çözülecektir. Aşağıdaki tabloda karma yem maddeleri ve bunların besin değerleri verilmektedir. Bu tablodan yararlanarak amaç ve ilgili yönetmenlikler [101] dikkate alınarak kısıtlar belirlenmiştir.

Kaliteli kanatlı piliç büyüme yemi için amaçlar:

- 1- Fiat (Min) $330X + 346X + 325X + 220X + 850X + 350X + 350X + 1400X + 125X + 1000X + 40X + 120X + 2700 + 900X + 8500X + 13000X + 11000X$
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
- 2- Ham selüloz (Min) $.03X + .035X + .065X + .10X + .07X + .23X + .17X$
1 2 3 4 5 6 7
- 3- Ham Kül (Min) $.02X + .03X + .03X + .05X + .07X + .07X + .07X + .19X + .10X + .85X + .99X + .99X + .01X + .01X$
1 2 3 4 5 6 7 8 9 10 12 13 16 17
- 4- Protein (Max) $.085X + .115X + .105X + .130X + .41X + .30X + .31X + .63X + .08X$
1 2 3 4 5 6 7 8 9

$$5- M.Enerji (Max) \quad 3.350X + 3.050X + 2.750X + 1.700X + 2.300X \\ +2.000X + 2.100X + 3.100X + 2.000X + 3.990X + 3.990X \\ \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 16 & 17 \end{matrix}$$

Tablo 4 . Yem maddeleri ve besin degerleri

HAMMADDE ADI	FİYAT	MİKT	KURU MADD	HAM PROTE	HAM SELÜL	HAM KÜL	HAM YAĞ	METD + LİSİN	LİSİN
MISIR	330	1	.88	.085	.030	.02	.04	.0038	.0028
BUGDAY	346	1	.90	.115	.035	.03	.02	.0045	.0040
ARPA	325	1	.90	.105	.065	.03	.02	.0037	.0025
KEPEK	220	1	.89	.130	.100	.05	.06	.0043	.0060
SOYA KÜSP	850	1	.89	.410	.070	.07	.01	.0110	.0240
AYÇİÇEK K	350	1	.90	.300	.230	.07	.015	.0130	.0110
PAM TOH K	350	1	.91	.310	.170	.07	.015	.0110	.0130
BALIK UNU	1400	1	.92	.630		.19	.10	.0260	.0500
ET-KEM UN	750	1	.94	.400		.32	.10	.0125	.0165
MELAS	125	1	.80	.080		.10			
DCP	1000	1	.98			.85			
MERMER TO	40	1	.98			.99			
TUZ	120	1	1			.99			
PREMİ 206	2700	1							
İZMİN 1	900	1							
ANTİKOKSİ	8500	1							
SEN LİZİ	13000	1	.98			.01			
SEN METH	11000	1	.98			.01		.9800	

HAMMADDE ADI	LİON ASİD	NİŞA STA	FOS FOR	SOD YUM	KALSİ YUM	HAZMOLUR PROTEİN	METOBİ ENERJİ	METHO NİN
MISIR	.020	.78	.0022	.0001	.0004	.0008	3350	
BUGDAY	.006	.78	.004	.0002	.0005	.0027	3050	
ARPA	.006	.73	.004	.0002	.0007	.0017	2750	
KEPEK		.50	.012	.0002	.0010	.0040	1700	
SOYA KÜSP.	.0045	.70	.0065	.0002	.0015	.0020	2300	
AYCIÇEK K.	.0045	.58	.012	.0001	.0044	.0030	2000	
PAM TOH K.	.0045	.58	.012	.0004	.0020	.0032	2100	
BALIK UNU		.78	.025	.0040	.0450	.0250	3100	
ET-KEM UN	.005	.60	.050	.0073	.0100	.0500	2400	
MELAS		.52	.0002	.0114	.0010	.0001	2000	
DCP			.180		.2300	.1800		
MERMER TO					.3500			
TUZ				.3900				
PREMİ 206								
İZMİN 1								
ANTİKOKSİ								
SEN LİZİN							3990	
SEN METHO							3990	.9800

Kısıtlar:

$$1- \text{Kalsiyum} \quad .0004X + .0005X + .0007X + .001X + .0015X \\ + .0044X + .002X + .045X + .001X + .23X + .35X \leq 6$$

1 2 3 4 5
6 7 8 9 10 11

$$2- \text{Kalsiyum} \quad .0004X + .0005X + .0007X + .001X + .0015X \\ + .0044X + .002X + .045X + .001X + .23X + .35X \leq 15$$

1 2 3 4 5
6 7 8 9 10 11

Yeni ÇAKDS ile çözüm sonunda amaçların gerçekleşme değerleri ve istenen düzeyleri:

AMAÇLAR	İSTENEN	GERÇEK-LEŞEN
MALİYET (MIN)		590064.5
HAM SELÜLOZ (MIN)	≤ 70	56.36
HAM KÜL (MIN)	≤ 80	70.43
PROTEİN (MAX)	≥ 160	249.71
METABOLİK ENJ. (MAX)	≥ 2750	2866.61

Yeni ÇAKDS ile çözüm sonunda kısıtların gerçekleşme değerleri ve istenen düzeyleri:

KISITLAR	İSTENEN	GERÇEK-LEŞEN
KALSIYUM	$6 \leq \text{KALSI} \leq 15$	13.208
FOSFOR	$\text{FOSFOR} \geq 6$	8.775
METHOİNİN + SİSTİN	$M+S \geq 5.5$	9.408
SODYUM	$1 \leq \text{SODY} \leq 3$	1.891
LYSİN	$\text{LYSİN} \geq 7$	14.596
KURU MADDE	$K + M \geq 880$	890.026
TOPLAM MİKTAR	$T M = 1000$	1000.0
BALIK UNU	$B U \geq 20$	150
TUZ	$\text{TUZ} = 3$	3
PREMİKS 206	$P206 = 2$	2
İZMİN-I	$\text{İZ-I} = 1$	1
ANTİKOKSİDİAL	$\text{ANTİ} = 1$	1

Uygulamada, enk. fiat, enb. protein, enb. metebolik enerji, enk. ham selüloz, enk. ham kül amaçları ile ele aldığımız kaliteli kanatlı yem probleminin yeni ÇAKDS ile bulunan optimum çözümü, ilgili bakanlığın [101] verdiği normlara göre; istenilen yönde, protein 89, metebolik enerji 116 puan daha yüksek , yine istenilen yönde, ham selüloz ve ham kül ise sırasıyla 14 ve 10 puan daha az gerçekleşmiştir.

SONUC

Gelişen bilim ve teknoloji nedeniyle artık, olayları ve gerçek dünya problemlerini tek kriter ile açıklamak tam ve yeterli görülmemektedir. Örneğin; çalışmamızda ele aldığımız kaliteli kanatlı yemleri probleminde olduğu gibi, enküçük maliyet amacı yanında enbüyük protein, enbüyük metabolik enerji, enküçük ham selüloz, enküçük ham kül gibi başka kriterler de gözönünde bulundurulmalıdır.

Karar verme probleminde ikiden fazla amaç bulunduğunda, çözüm kümesi nadiren tek noktadır. Fakat genellikle bu tür karar verme problemlerinde çözüm kümesi, bir çokyüzlünün (polihedron) kenar,köşe veya yüzeyleridir ve çözüm kümesi sonsuz nokta içerir. Bu aşamada uygun çözümün seçimi için karar vericinin tercihi önem kazanmakta ve bu durum etkileşimi zorunlu kılmaktadır.

Özellikle son yıllarda bilgisayar kullanımının görece ucuzlayışı , hafıza kapasitelerinin korkunç artışı yeni bir alan olan ve karar sürecini istenmeyen makinalaşmış sistem icrası durumundan çıkarıp, karar vericiyi de sisteme dahil ederek, karar verme hareketlerine yardım eden Karar Destekleme Sistemleri dediğimiz bilgisayar tabanlı bilgi sistemlerinin gelişmesine yol açmıştır.

1970' li yıllardan beri etkileşimli programlama dalında algoritma ve programlar üretilmektedir. ÇAKV için geliştirilen

programların önemli eksikliği, kullanıcının kullanım yüzüne (interface) özen gösterilmeyişiydi; Oysa belirtildiği gibi, bugünkü bilgisayar teknolojisi karar vericinin bakış açısından, problemlerin çözümü yanında , model yönetim, geliştirme ve çözüm çıktı fonksiyonu gibi kullanım yüzüne ilişkin yazılım (software) paketlerinin planlanmasına izin vermektedir. Bu ise, yöntemin kabul edilirliliğini etkileyen önemli bir faktördür. Zayıf düzenlenmiş kullanım yüzü yöntemin kabulünü de engeller [82,83].

Çok amaçlı karar verme alanında ilk yapılan araştırmalar "yöneticiye tüm baskın çözümleri verin, yönetici bunlardan kendi yarar fonksiyonuna en uygun olanı seçer" felsefesiyle yapılmıştır. Son zamanlarda ise araştırmacıların çoğu yöneticinin yarar fonksiyonuna uyan etkin çözümü bulmasında destek olmayı savunmaktadır [81] . Yeni ÇAKDS bu düşünceden yola çıkılarak geliştirilmiştir. Bu sistemde karar verici, karar destekleme sistemi ile aracısız, bilgisayarla doğrudan etkileşerek problemine çözüm bulacaktır. Söz konusu durum, çözüm zamanını kısaltmakta ve çözüm sürecini daha sağlıklı kılmaktadır.

Biz, bu çalışmada geliştirdiğimiz ÇAKDS'inde Korhonen ve Laakso' nun görsel etkileşimli algoritmalarında kullandıkları sağ taraf parametrik programlama yerine, adım büyüklüğü sağ taraf parametrik programlamayı kullandık. Bu algorithmada uygun bölgenin içinde veya dışında bir nokta önce etkin küme üzerine izdüşürülmekte, sonra bu etkin küme üzerinde istenilen hız ve yönde gezim ile amaçların göreceli büyüklükleri bilgisayarın grafik özelliklerinden de yararlanılarak kullanıcıya sunulmaktadır.

Böylece karar verici doğrudan bilgisayarla etkileşerek amaçların aldığı değerleri görecektir ve problemini daha iyi tanıyacaktır.

Yeni ÇAKDS'in veri tabanı, Zeleny'nin önerdiği biçimde kriterler, kısıtlar, değişkenlerin tümünü değiştirebilecek ve tümünü modelde denge içinde sürekli formüle edebilecek yapıda kurulmuştur.

Kaliteli kanatlı yem problemini yeni ÇAKDS kullanarak çözdük ve ilgili bakanlığın verdiği normlar ile karşılaştırıldığında oldukça iyi sonuçlar elde ettik. Bu çözümde istenilen yönde protein 89, metabolik enerji 116 puan daha yüksek, ham selüloz 14, ham kül 10 puan istenilen yönde daha az gerçekleşmiştir.

KAYNAKLAR

- 1- Gerald, W.E, "An Overview of Techniques for Solving Multiobjective Mathematical programs", Management Science, vol: 30, No:11, (1984).
- 2- Gibson, M. et al, "A Comparison of Interactive Multiple-Objective Decision Making Procedures", Comput. Opns. Res., vol:14, No:2, pp.97-105, (1987).
- 3- French. S., "Interactive Multi-Objective Programming : It's Aims, Applications and Demands", J. Opl. Res. Soc. Vol:35, No:9, pp.827-834, (1984).
- 4- Hwang, C.L., Masud A.S.M. et al , "Mathematical Programming with Multiple Objectives: A Tutorial", Comput. & Ops. Res. vol:7, pp.5-31, (1980).
- 5- Chankong, V., Haimes, Y.Y. , Multiobjective Decision Making Theory and Methodology, Elsevier Science Publishing Co. Inc. Newyork, (1983).
- 6- Aksoy, Y. , "Interactive Multiobjective D. M. : A Bibliography (1965-1988)". Management Research News V: 13 , (1990).
- 7- Yu, P.L., Multiple Criteria Decision Making Concepts, Techniques and Extensions, Plenum Press, Newyork, (1985).
- 8- Haktanir. T., "Türkiyede Üretilen Karma Hayvan Yemleri Optimum Rasyonu..", Yem Sanayi Dergisi, S:49, (1985).
- 9- Eck , R.D., Operation Research For Business, Wordsworth Pub. California, (1980).
- 10- Çakır , C. , DP yön. end. mal. karma yem for. sap. üze. bir araş. (Doçentlik tezi), Bornova,(1978)

- 11- Von Neuman J. and Morgenstern O., Theory of Games and Economic Behavior, Princeton, N.J., (1944).
- 12- Arrow, K.J., E.W. Barankin et al., "Admissible Points of Convex Sets" in H.W. Kuhn and A.W. Tucker (eds.), Contributions to the Theory of Games, Princeton, pp. 87- 91, (1953).
- 13- Charnes, A. and Cooper W.W. , Management Models and Industrial Applications of Linear Programming, Management Sciences, Vol:4, No:1, pp. 81-87, (1957).
- 14- _____ , _____ , Management Model and Industrial Applications of Linear Programming, Vol:1 and 2, Wiley, Newyork, (1961).
15. Zadeh, L.A. , "Optimality and Nonscalar-Valued Performance Criteria", IEEE Transactions on Automatic Control, AL-8, No:1, (1963).
- 16- Klinger, A., "Vector-valued Performance Criteria", IEEE Transactions on Automatic Control, AC-9, No:1, (1964).
- 17- Fishburn, P.C., "Independence in Utility Theory with Whole Product Sets", Operation Research, Vol:13, (1965).
- 18- Pollak, R.A., "Additive von Neumann-Morgenstern Utility Functions", Econometrica, Vol:35, (1967).
- 19- Raiffa, H., "Preference for Multi-attributed Alternatives", Rand Memorandum, RM-5868-DDT, Santa Monica, Calif., (1969).
- 20- MacCrimman, K.R., "Decision Making Among Multiple-Attribute Alternatives : A survey and Consolidated Approach", RAND Memorandum, RM-4823-ARPA, Santa Monica, Calif., (1969).
- 21- Keeney, R.L., "Quasi-Separable Utility Functions", Naval

- Research Quarterly, Vol:15, pp. 551-565, (1968).
- 22- Süskü, J. , " Linearni Multiprogramovani", Ekonomicko Matematicky Obzor., Vol:4, No:3, (1968).
- 23- Radzikowski, O., "Die Berücksichtigung mehrerer Ziel..", Wirtschaftswissenschaft, No:5, (1967).
- 24- Jüttler, H., "Lineinaja Modelj neskol'kimi celevimi funktsijami", Ekonomika i Matematicheskoe Metody, Vol:3, No:3, (1967).
- 25- Dinkelbach, W., " Über einen Lösungsansatz zum Vektor maximumproblem", in M. Beckmann (ed.) Unternehmen sforschung Heute, Springer-Verlag, Berlin, (1971).
- 26- ———— , Durr W. , "Effizienenzaussagen bei Ersatzprogrammen zum Vektormaximumproblem" , in R. Henn, H.P. Künzi (eds), Operations Research-Verfahren, Vol:12, Verlag Anton Hain, (1971).
- 27- Philip, J.P. and Steuer R.E., "Algorithms for Vector Maximization Problems," Mathematical Programming, Vol:2 No:2, pp.207-229, (1972).
- 28- Evans, J.P. and Steuer R.E., "A revised Simplex Method for Linear Multiple Objective Programs", Mathematical Programming, Vol:5, No:1, (1973).
- 29- Zeleny. M. , Linear Multiple objective Programming, Springer-Verlag, Newyork, (1974).
- 30- Cochrane, J.L., Zeleny M., Multiple Criteria Decision Making, University of South Carolina Press, Columbia, (1973).
- 31- Zeleny, M. , Multiple Criteria Decision Making, McGraw-Hill Inc., Newyork, (1982).
- 32- Benayoun R., et al. , " Linear Programming with

- Multiobjective Functions: Step Method (STEM)", Mathematical Programming Vol:1, pp.366-375, (1971).
- 33- Geoffrion, A.M., Dyer, J.S. and Feinberg, A., "An Interactive Approach for Multi-criterion Optimization with an Application to the Operation of Academic Department", Management Science 19, (1972).
- 34- Zionts, S. and Wallenius, J., "An Interactive Programming Method for Solving the Multiple Criteria Problem", Management Sciences 22, pp.652-663, (1976).
- 35- Zionts, S. and Wallenius, J., "An Interactive Multiple Objective Linear Programming Methods....", Management Science, Vol:29, No:5, 1983.
- 36- Steuer, R.E., Multiple Criteria Optimization, Theory, Computation and Application, John Wiley & Sons, Newyork, 1986.
- 37- Esin, A., Yöneylem Araştırmasında Kullanılan Doğrusal Programlama Teknikleri, Ankara, (1984).
- 38- Vincke P., "Analysis of Multicriteria Decision Aid in ...", European Journal of Oper. Res. 25, pp.160-168, (1986).
- 39- Kok, M., "The Interface with Decision Makers and some Experimental results in IMOPM", European Journal of Oper. Res., 26, pp. 96-107, (1986).
- 40- Kara, İ., Yöneylem Araştırması Doğrusal Olmayan Modeller, Anadolu Üniversitesi, Eskişehir, 1986.
- 41- Churchman, C.W., The system Approach, Dell, Newyork, (1986).
- 42- Spronk, J. and Zambruna G., "IMGP for Bank Portfilo Selection", Report 8317/F, Erasmus University, Rotterdam, (1983).

- 43- Keeney, R.L. and Raiffa, H., Decision with Multiple Objectives, Wiley, Newyork, (1976).
- 44- Zeleny, M., "Multiple Objectives in Mathematical Programming : letting the Man in", Comput. & Ops. Res. Vol:7, pp. 1-4, (1980).
- 45- Ignizio, J.P., Goal Programming and Extensions, Heath, Lexington, pp. 193-219, Mass., 1976.
- 46- Lee, S.M., Goal Programming for Decision Analysis, Aurbach Publishers, Philadelphia, pp.191-350, (1972).
- 47- Lin, W.T., "A Survey of Goal Programming Applications", Omega, Vol:8, No:1, (1980).
- 48- Bal, H.: Cok Amaçlı Karar Vermede Amaç Programlaması ve Yem Sanayiine bir Uygulama, Doktora tezi, G.Ü. Fen Bilimleri Enstitüsü, Ankara, (1986).
- 49- Webster's New World Dictionary
- 50- Achoff, R.L. and Sasieni, M.W., Fundamentals of Operations Research, Wiley, Newyork, (1968).
- 51- MacCrimman, K.R. and Wehrung, D.A., Trade-off Analysis : " The indifference and preferred proportions appr...", Wiley, Chichester Great Britain, (1977).
- 52- Brockhoff, K., "Experimental Test of MCDM Algorithm in a modular Approach", European Journal of Oper. Resh. 22, pp. 159-166, (1985).
- 53- Wallenius, J. , " Comparative Evaluation of Some Interactive Approaches to Multicriterion Optimization", Management Science, Vol:21, No:12, (1975).
- 54- Gibson, M. et al., "A Comparison of Interactive Multiple-

- Objective Decision Making Procedures", Comput. Opns. Res.
Vol:14, No:2, pp. 97-105, (1987).
- 55- Buchanan, J.T. and Daellenbach, H.G. , "A comparative Evaluation of Interactive Solution Methods for Multiple Objective Decision Models",
- 56- Kok, M., Spronk J. , "A note on the Pay-off Matrix in Multiple Objective programming", Omega Vol:13, No:6, (1985).
- 57- Fichet, J., " GPSTEM : An Interactive Multiobjective Optimization method", Progress in Operations Research, North-Holland, Vol:1, (1976).
- 58- Choo, E.U. and Atkins, D.R. , " An Interactive Algorithm for Multicriteria Programming", Comput. & Ops. Res. Vol:7, (1980).
- 59- Nijkamp, P. and Spronk, J., "Interactive Multiple Goal Programming...", Multiple Decision Making, West Germany, (1979).
- 60- Spronk, J. and Telgen, J., "An Ellipsoidal Interactive MGP", Multiple Criteria Decision Making, Newark, (1980).
- 61- Geoffrion, A.M. , " Vector Maximal Decomposition Programming", Working Paper No:165, Western Management Science Institute University of California, (1986).
- 62- Giselt, H.A., Pederzali G., O.R. Continuous Optimization Models, Walter de Gruyter, Newyork (1987).
- 63- Cochrane, J.L. and M.Zeleny (eds) , Multiple Criteria Decision Making, University of South Carolina, Colombia, (1973).
- 64- Dyer, J.S. , "Interactive Goal Programming", Management

- Science 19, 62-70, (1972).
- 65- Openheimer, K.K. : "A proxy Approach to Multi-attribute Decision Making", Management Science, 24, 1978.
- 66- Ecker, J.G. and Kouada ,I,A., "Finding Efficient Points for Linear Multiple Objective Programs", Mathematical Programming, 8, 375-377, (1975).
- 67- Steuer, R.E., "Multiple objective Linear Programming with Interval criterion Weights", Management Science, Vol:23, No:3, (1983).
- 68- Steuer, R.E. and Harris W,F., "Intra-set Point Generation and Filtering in Decision and Criterion Space", Comput. & Ops. Res. Vol:7, pp. 67-79, (1980).
- 69- Morse, J.N., "Reducing the size of the nondominated set: Pruning by clustering", Comput. & Ops. Res. Vol:7, (1980).
- 70- Torn A.A., "A sampling search-clustering approach for exploring The feasible/efficient solutions of MCDM problems" Comput.& Ops.Res, Vol:7,pp 67-79, (1980).
- 71- Korhonen, P.J. and Laakso, J., "A Visual Interactive Method for solving the Multiple Criteria Problem", European Journal of Operational Research 24, (1986).
- 72- Hemming T., "Some Modifications of a large step gradient method for Interactive Multicriterion Optimization" in J. Morse (ed) Organizations : Multiple Agent with Multiple Criteria, Springer, Newyork, 1981.
- 73- Winkels, H.M. and Meikka, M., "An Integration of Efficiency projections...", European Journal of Operational Research 16, (1984).

- 74- Spronk, J., Interactive Multiple Goal Programming Applications to Financial Planning, Nichof, Boston, (1981).
- 75- Steuer, R.E. and Choo, E.U., "An Interactive weighted Tchbycheff procedure for Multiple Objective Programming", Mathematical Programming, 26, (1983).
- 76- Nakayama H., Sawaragi Y., "Satisficing trade-off method for multi-objective programming and its Applications", 9. IFAC World Congress, Budapeest, Hungary.
- 77- Kok, M. and Lootsma, F.A. , "Pairwise Comparision methods in multiple objective programming with Applications in a long-term energy-planning model", European Journal of Operational Research 22, (1985).
- 78- Gass, S.I. , Linear Programming Methods and Applications, McGraw-Hill, Newyork, (1984).
- 79- Sucu G., Çok Amaçlı Karar Teknikleri ve Yeni Bir Yaklaşım, Doktora , H.Ü., (1985).
- 80- Gibson, M. et al. , "A comporison of Interactive Multiple-objective Decision Making Procedures", Comput. Opns. Res. Vol:14, No:2, (1987).
- 81- Karpak, B. : "Üretim planlamada Pareto Yarışı" , YA/EM ulusal kongresi, Bilkent Üniversitesi, Ankara , (1989).
- 82- Korhonen, P.J. and Wallenius J., "A Pareto Race", Working papers, Forthcoming in Naval Research Logistics in (1989).
- 83- Korhonen P. J., Vig- A visual Interactive Support System for multiple Criteria Decision Making , Helsinki School of Economics, Helsinki, Finland, (1988).
- 84- Weber E.S., "System to thing with: A response to A vision

- for Decision Support Systems" , Journal of Managament Information System II, pp,85-89, 1986.
- 85- Ginzberg, M. and Stohr E., "Decision Support Systems : Issues and Perspectives", in M.Ginzberg et al. (Eds) , Decision Support Systems, Amsterdam, 1982.
- 86- Jelassi, M.T. , " Designing a Generalized Multiple Criteria Decision Support Systems",Journal of Management Information System I, pp:24-43, (1985).
- 87- Le Blank, L. and Jelassi M.T., "DSS software Selection: A Multiple Criteria Decision Methodolgy", Information of Management 17,pp 49-65, 1989.
- 88- Zeleny, M., "Stable Patterns from Decision-Producing Networks ,New Interfaces of DSS and MCDM ", MCDM World Scan, vol:3, No: 2 & 3, October (1989).
- 89- Miller, G., "The magical number seven plus or minus two: Some limits on our capacity for processing information" , Psychol Rev, 63, 81-87, (1956).
- 90- Monarchi, D.E. etal., "Interactive Multiobjective Programming in Water Resources: A case study" , Water Resources Research, vol. 9, No: 4, (1973).
- 91- Büyüksahin, H., "Dünden bugüne Türkiye Karma Yem Sanayi", Yem Sanayi dergisi, Sayı:59, (1989).
- 92- Akyıldız, R., Karma Yemler Endüstrisi, San Matbaası, Ankara, (1979).
- 93- Özen, N., "Geviş getiren Hayvanlarda ...", Yem Sanayi Dergisi Dergisi, Sayı:53, 1986.
- 94- Gökçebağ, M., "Yem Sanayiinde Kalite Kontrol Sorunları",

Yem Sanayi Dergisi ,Sayı 53, (1986).

- 95- Öztürkcan, O., "Ürenin Azot Kaynağı olarak Hayvan Beslemede Kullanılması", Yem Sanayi Dergisi, Sayı:59, (1988).
- 96- Thomason, D. et al., Tavuk Yetiştirme (Çev:Prof.Dr. Kamil Dogan), Ankara, (1987).
- 97- Karakuş, M., Aran, N., "Yem Sanayimizin Proteinli hammadde sorunu", Yem Sanayi dergisi, Sayı:49, (1985).
- 98- Sevgican, F. ve ark., "Kasaplık Piliç Karma Yemlerinde.", Yem Sanayi dergisi, Sayı:49, (1985).
- 99- Büyüksahin, H., "Hayvansal Kökenli Yemlerin Hayvan Beslemedeki Yeri", Yem Sanayi Dergisi,Sayı 49, (1985).
- 100- Özerdem, O., Bülbül, C., " Yem Endüstrisinde Enzim Kullanımı...", TUBİTAK Araş. Proj. Yar. Öz., Ankara, (1989).
- 101- Resmi Gazete, "Karma Yem Normları", No:19501, 28 Haziran (1987).

PROGRAM LISTESI

```
10 CLS
20 SCREEN 2
30 LINE (25,15)-(500,250),,BF
40 KEY OFF
50 LOCATE 3,15:PRINT "...A.N.A....M.E.N.U....."
60 LOCATE 5,15:PRINT "1==>VENI.MODEL.GIRISI....."
70 LOCATE 6,15:PRINT "2==>DISK(ET)TEN.GIRIS....."
80 LOCATE 7,15:PRINT "3==>MODEL.COZUMU....."
90 LOCATE 8,15:PRINT "4==>MODEL.DUZELTME-DEGISTIRME....."
100 LOCATE 9,15:PRINT "5==>MODEL.VERILERINI.GORUNTULEME...."
110 LOCATE 10,15:PRINT "6==>COZUM.VERILERINI.GORUNTULEME...."
120 LOCATE 11,15:PRINT "7==>MODEL.VERILERINI.SAKLAMA....."
130 LOCATE 12,15:PRINT "8==>PROGRAMDAN.CIKIS....."
140 LOCATE 15,15:PRINT ".....SECIMiniz=====>"
150 W$=INKEY$
160 IF W$="1" THEN RUN "YGBR.BAS"
170 IF W$="2" THEN RUN "DOKU7.BAS"
180 IF W$="3" THEN RUN "MCDZ"
190 IF W$="4" THEN RUN "AKDD.BAS"
200 IF W$="5" THEN RUN "YAZGD.BAS"
210 IF W$="6" THEN RUN "CDZ.BAS"
220 IF W$="7" THEN RUN "MSAK.BAS"
230 IF W$="8" THEN SYSTEM
240 GOTO 150
```

```

250 SCREEN 2
260 DIM A(9,100),D(80,100),F(90),RHS(90),AD$(90),ISA$(90)
270 KEY OFF
280 CLS:LOCATE 5,10:PRINT "YENI MODELİN ADINI GİRİNİZ....."
290 LOCATE 6,10:PRINT "(EN FAZLA 4 KRAKTER. OR.:YENİ)"
300 LOCATE 5,45:INPUT KUT$
310 IF KUT$="" THEN 280
320 IF LEN(KUT$)>4 THEN KUT$=LEFT$(KUT$,4)
330 CLS:REM BASARI OLCEKLEME GİRİŞİ İÇİN
340 REM LINE (180,25)-(570,170),,B
350 CLS
360 LINE (10,30)-(540,200),,BF
370 LOCATE 5,5:PRINT "*****YENİ MODEL GİRİŞİ*****"
380 LOCATE 7,5:PRINT "* AMAC FONKSİYONU SAYISINI GİRİNİZ      (<      >#)"
390 LOCATE 7,48:INPUT ASZ
400 IF ASZ>9 OR ASZ<1 THEN 350
410 LOCATE 8,5:PRINT "* KİSİT SAYISINI GİRİNİZ              (<      >#)"
420 LOCATE 8,48:INPUT KWZ
430 IF KWZ<1 OR KWZ>50 THEN 410
440 KSZ=KWZ
450 LOCATE 9,5:PRINT "* DEĞİSKEN SAYISINI GİRİNİZ          (<      >#)"
460 LOCATE 9,48:INPUT KZZ
470 IF KZZ<1 OR KZZ>50 THEN 450
480 DSZ=KZZ
490 LOCATE 11,5:PRINT "* 1- DEĞİSKEN İSİMLERİ (X1,X2...) [DEFAULT]      "
500 LOCATE 12,5:PRINT "* 2- İSTEGE BAĞLI [1/2]              (<      >#)"
510 LOCATE 12,48:
520 W$=INKEY$
530 IF W$="1" THEN 560
540 IF W$="2" THEN 580
550 GOTO 520
560 IS=1
570 GOTO 750
580 IS=2
590 CLS:LOCATE 2,5:PRINT "*** DEĞİSKEN ADLARI ENFAZLA 3 KRAKTER !!! ***"
600 II=1
610 FOR I=1 TO KZZ STEP 2
620 LOCATE II+3,10:PRINT I;".DEĞ ADI=";:INPUT AD$(I)
630 IF LEN (AD$(I))>3 THEN AD$(I)=LEFT$(AD$(I),3)
640 IF I+1 >KZZ THEN 680
650 LOCATE II+3,40:PRINT I+1;".DEĞ ADI=";:INPUT AD$(I+1)
660 IF LEN (AD$(I+1))>3 THEN AD$(I+1)=LEFT$(AD$(I+1),3)
670 II=II+1
680 NEXT I
690 LINE (50,290)-(550,310),,BF
700 LOCATE 22,10:PRINT "* 1- DÜZELTME                2- DEVAM #"
710 W$=INKEY$
720 IF W$="1" THEN 590
730 IF W$="2" THEN 780
740 GOTO 710
750 FOR I=1 TO KZZ
760 AD$(I)="X"+STR$(I)
770 NEXT I
780 FOR I=1 TO KWZ

```

```

790 ISA$(I)=">"
800 NEXT I
810 CLS:LOCATE 1,5:PRINT "% MAX icin 1, MIN icin 2 GIRINIZ! %"
820 FOR I=1 TO ASZ
830 GOSUB 3280
840 IF TIP(I)<1 OR TIP(I)>2 THEN 830
850 NEXT I
860 CLS:LINE (70,15)-(550,170),,BF
870 LOCATE 3,15:PRINT "*****"
880 LOCATE 4,15:PRINT "### DUZELTME MENUSU ###"
890 LOCATE 5,15:PRINT "### 1- TEK AMAC TIPI DUZELTME ###"
900 LOCATE 6,15:PRINT "### 2- TEK TEK KONTROL VE DUZELTME ###"
910 LOCATE 7,15:PRINT "### 3- DEVAM ###"
920 LOCATE 10,15:PRINT "***** SECIMINIZ **[1/2/3]**"
930 LOCATE 8,15:PRINT "*****"
940 W$=INKEY$
950 IF W$="1" THEN 990
960 IF W$="2" THEN 1040
970 IF W$="3" THEN 1110
980 GOTO 940
990 CLS :LOCATE 10,10:PRINT "AMAC NO:";INPUT I
1000 IF I>ASZ THEN 990
1010 CLS:LOCATE 1,5:PRINT "% MAX icin 1, MIN icin 2, DEGISTIRMEDEN DEVAM icin
RETURN GIRINIZ! %" :GOSUB 3260
1020 GOSUB 3290
1030 GOTO 860
1040 CLS :
1050 CLS:LOCATE 1,5:PRINT "% MAX icin 1, MIN icin 2, DEGISTIRMEDEN DEVAM icin
RETURN GIRINIZ! %"
1060 FOR I=1 TO ASZ
1070 GOSUB 3260
1080 GOSUB 3290
1090 NEXT I
1100 GOTO 860
1110 FOR J=1 TO ASZ :CLS
1120 LOCATE 2,1:PRINT J;". AMAC :";
1130 IF TIP(J)=1 THEN PRINT "MAX"; ELSE PRINT "MIN";
1140 II=1:III=1
1150 FOR I=1 TO KZZ
1160 LOCATE 2+II,3+III:PRINT "_____";AD$(I)
1170 III=III+16:IF III>64 THEN III=1:II=II+2
1180 NEXT I
1190 II=1:III=1
1200 FOR I=1 TO KZZ
1210 LOCATE 2+II,3+III:INPUT A(J,I)
1220 III=III+16:IF III>64 THEN III=1:II=II+2
1230 NEXT I
1240 LINE (50,290)-(550,310),,BF
1250 LOCATE 22,10 :PRINT "1- DUZELTME 2- DEVAM
1260 W$=INKEY$
1270 IF W$="1" THEN 1300
1280 IF W$="2" THEN 1310
1290 GOTO 1260
1300 GOSUB 2390
1310 NEXT J
1320 CLS

```

```

1330 LINE (60,25)-(550,170),,BF
1340 LOCATE 4,15:PRINT "*****"
1350 LOCATE 5,15:PRINT "###      AMAC DUZELTME MENUSU      ###"
1360 LOCATE 6,15:PRINT "### 1- TEK AMACI KONTROL VE DUZELTME ###"
1370 LOCATE 7,15:PRINT "### 2- TEK TEK KONTROL VE DUZELTME ###"
1380 LOCATE 8,15:PRINT "### 3- DEVAM          ###"
1390 LOCATE 9,15:PRINT "*****"
1400 LOCATE 11,15:PRINT "***** SECIMINIZ ##[1/2/3]*****"
1410 W%=INKEY$
1420 IF W%="1" THEN 1460
1430 IF W%="2" THEN 1520
1440 IF W%="3" THEN 1570
1450 GOTO 1410
1460 CLS
1470 LOCATE 5,5 :PRINT "KACINCI AMAC=";:INPUT J
1480 IF J>ASZ THEN 1470
1490 KD=J
1500 GOSUB 2390
1510 GOTO 1320
1520 FOR J=1 TO ASZ
1530 KD=J
1540 GOSUB 2390
1550 NEXT J
1560 GOTO 1320
1570 IF MM=2 THEN RETURN
1580 FOR I=1 TO KWZ
1590 CLS
1600 II=1
1610 LOCATE 11,1:PRINT I;". KISIT:"
1620 II=II+1 :III=1
1630 FOR J=1 TO KZZ
1640 LOCATE 2+II,3+III:PRINT "      ":AD$(J)
1650 III=III+16:IF III>64 THEN III=1:II=II+1
1660 NEXT J
1670 LOCATE 2+II,5+III:PRINT ISA$(I);"      "
1680 II=2
1690 II=II+1 :III=1
1700 FOR J=1 TO KZZ
1710 LOCATE 1+II,3+III:INPUT D(I,J)
1720 III=III+16:IF III>64 THEN III=1:II=II+1
1730 NEXT J
1740 LOCATE 1+II,3+III:INPUT S$
1750 IF S%="" THEN 1780
1760 IF S%="=" OR S%="<" OR S%=">" THEN 1770 ELSE 1740
1770 ISA$(I)=S$
1780 LOCATE 1+II,7+III:INPUT RNS(I)
1790 II=II+1:III=1
1800 IF II>20 THEN 1810
1810 LINE (50,290)-(550,310),,BF
1820 LOCATE 22,10:PRINT "1- DUZELTME      2- DEVAM      "
1830 W%=INKEY$
1840 IF W%="1" THEN 1870

```

```

1840 IF W$="1" THEN 1870
1850 IF W$="2" THEN 1880
1860 GOTO 1830
1870 GOSUB 2640
1880 NEXT I
1890 GOSUB 3040
1900 IF KUT$="" THEN CLS:LOCATE 5,10:INPUT "Lutfen problem adini giriniz !!!";KU
T$:GOTO 1900
1910 OPEN "o",#1,"kont.bas"
1920 WRITE #1,KUT$
1930 CLOSE #1
1940 KUTA$=KUT$+"1"
1950 OPEN "o",#1,KUTA$
1960 WRITE #1,ASZ,KIZ
1970 FOR I=1 TO ASZ
1980 WRITE #1,TIP(I)
1990 NEXT I
2000 FOR P=1 TO ASZ
2010 FOR P1=1 TO KZZ
2020 WRITE #1,A(P,P1)
2030 NEXT P1
2040 NEXT P
2050 FOR P=1 TO KZZ
2060 WRITE #1,AD$(P):REM PRINT "D";AD$(P)
2070 NEXT P
2080 CLOSE #1 :IF MM=2 THEN MM=0:RETURN
2090 KUTK$=KUT$+"2"
2100 OPEN "o",#1,KUTK$
2110 WRITE #1,KWZ,KZZ
2120 FOR P=1 TO KWZ
2130 FOR P1=1 TO KZZ
2140 WRITE #1,D(P,P1):REM PRINT "LKIS";D(P,P1);
2150 NEXT P1
2160 WRITE #1,ISA$(P),RHS(P):REM LPRINT ISA$(P),RHS(P),"ISA"
2170 NEXT P
2180 FOR P=1 TO KZZ
2190 WRITE #1,AD$(P)
2200 NEXT P
2210 CLOSE #1 :IF MM=2 THEN MM=0:RETURN
2220 CLS
2230 LINE (50,25)-(550,170),,BF
2240 LOCATE 4,15:PRINT "*****"
2250 LOCATE 5,15:PRINT "SON MENU"
2260 LOCATE 6,15:PRINT "1- AMAC MENUSUNE DONUS"
2270 LOCATE 7,15:PRINT "2- KISIT MENUSUNE DONUS"
2280 LOCATE 8,15:PRINT "3- ANA MENUYE DONUS"
2290 LOCATE 9,15:PRINT "*****"
2300 LOCATE 11,15:PRINT "***** SECIMINIZ **[1/2/3]**"
2310 W$=INKEY$
2320 IF W$="1" THEN 2360
2330 IF W$="2" THEN 2370
2340 IF W$="3" THEN 2380
2350 GOTO 2310
2360 MM=2:GOSUB 1320 :GOSUB 1910:GOTO 2220
2370 GOSUB 3040 :MM=2 :GOSUB 2090:GOTO 2220

```



```

2380 CLOSE :RUN "AMENU.BAS"
2390 CLS
2400 LOCATE 2,1:IF T1P(J)=1 THEN PRINT J;" . AMAC =MAX"; ELSE PRINT J;" . AMAC =MI
N";
2410 I1=1:III=1
2420 FOR I=1 TO KZ%
2430 LOCATE 2+I1,3+III:PRINT A(J,1);:FF=LEN (STR$(A(J,1))):PRINT STRING$(13-FF,"
_");:LOCATE 2+I1,16-FF+III:PRINT AD$(I)
2440 III=III+16:IF III>64 THEN III=1:I1=I1+2
2450 NEXT I
2460 IF KD=1 THEN KD=0:GOTO 2580
2470 IF W$="0" THEN 2480
2480 LINE (60,280)-(570,310),,BF
2490 LOCATE 22,10 :PRINT "!!!! sifirlanacak katsayi icin 999 giriniz !!!!!"
2500 I1=1:III=1
2510 FOR I=1 TO KZ%
2520 LOCATE 2+I1,2+III:INPUT FD
2530 IF FD=999 THEN A(J,1)=0 :GOTO 2560
2540 IF FD=0 THEN 2560
2550 A(J,1)=FD
2560 III=III+16:IF III>64 THEN III=1:I1=I1+2
2570 NEXT I
2580 LINE (50,290)-(550,310),,BF
2590 LOCATE 22,10 :PRINT "1- DUZELTME          2- DEVAM
2600 W$=INKEY$
2610 IF W$="1" THEN 2390
2620 IF W$="2" THEN RETURN
2630 GOTO 2600
2640 I1=2 :III=1 :CLS:LOCATE 11,III:PRINT I;" . KISIT"
2650 FOR J=1 TO KZ%
2660 LOCATE 2+I1,3+III:PRINT D(I,J);:FF=LEN(STR$(D(I,J))):PRINT STRING$(16-FF,"
_");:LOCATE 2+I1,16-FF+III:PRINT AD$(J)
2670 III=III+16:IF III>64 THEN III=1:I1=I1+1
2680 NEXT J
2690 LOCATE 2+I1,5+III:PRINT ISA$(I);"
2700 LOCATE 2+I1,8+III:PRINT RMS(I)
2710 I1=I1+1:III=1
2720 LINE (50,290)-(550,310),,BF
2730 LOCATE 22,10 :PRINT "1- DUZELTME          2- DEVAM
2740 W$=INKEY$
2750 IF W$="1" THEN 2780
2760 IF W$="2" THEN RETURN
2770 GOTO 2740
2780 I1=1
2790 LINE (50,290)-(550,310),,B
2800 LOCATE 22,10:PRINT "!!!! Sifirlanacak deger icin 999 giriniz !!!!!"
2810 I1=I1+1 :III=1
2820 FOR J=1 TO KZ%
2830 LOCATE 2+I1,2+III:INPUT DD:
2840 IF DD=0 THEN 2870
2850 IF DD=999 THEN D(I,J)=0 :GOTO 2870
2860 D(I,J)=DD
2870 III=III+16:IF III>64 THEN III=1:I1=I1+1
2880 NEXT J
2890 LOCATE 2+I1,1+III:INPUT S$
2900 IF S$="" THEN 2930

```

```

2910 IF S$="=") OR S$="=" OR S$="<=" THEN 2920 ELSE 2890
2920 ISA$(I)=S$
2930 LOCATE 2+II,7+III:INPUT RH
2940 IF RH=0 THEN 2970
2950 IF RH=999 THEN RHS(I)=0 :GOTO 2970
2960 RHS(I)=RH
2970 II=II+1:III=1
2980 LINE (50,290)-(550,310),,BF
2990 LOCATE 22,10:PRINT "1- DUZELTME          2- DEVAM
3000 W$=INKEY$
3010 IF W$="1" THEN 2640
3020 IF W$="2" THEN RETURN
3030 GOTO 3000
3040 CLS
3050 LINE (50,25)-(550,170),,BF
3060 LOCATE 4,15:PRINT "*****"
3070 LOCATE 5,15:PRINT "###      KISIT  DUZELTME  MENUSU      ###"
3080 LOCATE 6,15:PRINT "### 1- TEK KISITI KONTROL VE DUZELTME ###"
3090 LOCATE 7,15:PRINT "### 2- TEK TEK KONTROL VE DUZELTME ###"
3100 LOCATE 8,15:PRINT "### 3- DEVAM          ###"
3110 LOCATE 9,15:PRINT "*****"
3120 LOCATE 11,15:PRINT "***** SECIMINIZ **[1/2/3]*****"
3130 W$=INKEY$
3140 IF W$="1" THEN 3180
3150 IF W$="2" THEN 3220
3160 IF W$="3" THEN RETURN
3170 GOTO 3130
3180 CLS
3190 LOCATE 5,5 :PRINT "KACINCI KISIT=";:INPUT I
3200 GOSUB 2640
3210 GOTO 3040
3220 FOR I=1 TO KW%
3230 GOSUB 2640
3240 NEXT I
3250 GOTO 3040
3260 LOCATE I+2,10:PRINT I;". AMAC=";:IF TIP(I)=1 THEN PRINT "MAX" ELSE PRINT "K
IN"
3270 RETURN
3280 LOCATE I+2,10:PRINT I;". AMAC=";:INPUT TIP(I):RETURN
3290 LOCATE I+2,30:INPUT TP
3300 IF TP>2 THEN 3290
3310 IF TP=0 THEN RETURN
3320 TIP(I)=TP
3330 RETURN

```

```

3440 OPEN "I",#1,"kont.bas"
3450 INPUT #1,KUT$
3460 CLOSE #1
3470 KUTA$=KUT$+"1"
3480 OPEN "I",#1,KUTA$
3490 INPUT #1,ASZ,KZZ
3500 FOR I=1 TO ASZ
3510 INPUT #1,TIP(I)
3520 NEXT I
3530 FOR P=1 TO ASZ
3540   FOR P1=1 TO KZZ
3550     INPUT #1,A(P,P1)
3560   NEXT P1
3570   INPUT #1,AD$(P)
3580   NEXT P
3590   CLOSE #1
3600   KUTK$=KUT$+"2"
3610   OPEN "I",#1,KUTK$
3620   INPUT #1,KWZ,KZZ
3630   FOR P=1 TO KWZ
3640     FOR P1=1 TO KZZ
3650       INPUT #1,D(P,P1)
3660     NEXT P1
3670     INPUT #1,ISA$(P),RHS(P)
3680     NEXT P
3690     FOR P=1 TO KZZ
3700       INPUT #1,AD$(P)
3710     NEXT P
3720     CLOSE #1
3730   LOCATE 22,10:PRINT "MODEL GİRİŞİ DISK(ET)TEN YAPILMISTIR."
3740   LOCATE 23,10:INPUT "DEVAM İÇİN BİR TUSA BASINIZ=====>";T$
3750   RUN "AMENU.BAS"
3760   CLS:PRINT "BU DOSYA DISK(ET)TE YOK"
3770   PRINT "1-YENİ GİRİŞ  2-ANA MENU"
3780   W$=INKEY$
3790   IF W$="1" THEN RUN
3800   IF W$="2" THEN RUN "AMENU"
3810   GOTO 3780

```

```

3820 DIM D(100,100),F(100),AD$(100),TIP(10),ISA$(100),RHS(100)
3830 OPEN "i",#1,"kont.bas"
3840 INPUT #1,KUT$
3850 CLOSE #1
3860 KUTA$=KUT$+"1"
3870 OPEN "i",#1,KUTA$
3880 INPUT #1,SZ,KZZ
3890 KUTK$=KUT$+"2" : OPEN "i",#2,KUTK$
3900 INPUT #2,KWZ,KZZ
3910 KSZ=KWZ:DSZ=KZZ
3920 FOR I=1 TO SZ
3930 INPUT #1,TIP(I)
3940 NEXT I
3950 FOR I=1 TO SZ
3960 FOR J=1 TO KZZ
3970 INPUT #1,D(I+SZ,J)
3980 NEXT J
3990 NEXT I
4000 FOR I=1 TO KZZ
4010 INPUT #1,AD$(I)
4020 NEXT I
4030 CLOSE #1
4040 FOR I=2*SZ+1 TO KWZ+2*SZ
4050 FOR J=1 TO KZZ
4060 INPUT #2,D(I,J)
4070 NEXT J
4080 INPUT #2,ISA$(I),RHS(I)
4090 NEXT I
4100 FOR I=1 TO SZ
4110 CLOSE
4120 IF TIP(I)=1 THEN 4160
4130 FOR J=1 TO KZZ
4140 D(I+SZ,J)=-D(I+SZ,J)
4150 NEXT J
4160 NEXT I
4170 FOR I=1 TO SZ
4180 IF TIP(I)=1 THEN ISA$(I)="=>" ELSE ISA$(I)="(<="
4190 NEXT I
4200 FOR I=SZ+1 TO SZ*2
4210 ISA$(I)="="
4220 NEXT I
4230 LL=1
4240 FOR I=1 TO SZ
4250 D(I,KZZ+LL)=1
4260 D(I,KZZ+LL+1)=-1
4270 LL=LL+2
4280 NEXT I:LL=1
4290 FOR I=1 TO SZ
4300 D(SZ+I,KZZ+LL)=-1
4310 D(SZ+I,KZZ+LL+1)=1
4320 LL=LL+2
4330 NEXT I
4340 FOR I=1 TO SZ

```

```

4350 D(I,KZZ+2*SZ+1)=1
4360 D(I,KZZ+2*SZ+2)=-1
4370 NEXT I
4380 LL=1
4390 FOR I=1 TO SZ
4400 F(KZZ+LL)=.001*(-1)
4410 F(KZZ+LL+1)=.001
4420 LL=LL+2
4430 NEXT I
4440 F(KZZ+2*SZ+1)=-1
4450 F(KZZ+2*SZ+2)=1
4460 FOR I=1 TO SZ*2+2+KZZ
4470 NEXT I
4480 KWZ=KWZ+2*SZ
4490 SON1=KWZ
4500 SON2=KZZ+2*SZ+2
4510 FOR I=KZZ+1 TO KZZ+2*SZ STEP 2
4520 AD$(I)="Z"+STR$(I-KZZ)
4530 AD$(I+1)=AD$(I)
4540 NEXT I
4550 AD$(I)="ALF"
4560 J=KZZ+2*SZ+3 :II=1:AA=1
4570 FOR I=1 TO KWZ
4580 IF ISA$(I)("<=") THEN 4700
4590 IF ISA$(I)(">=") THEN 4760
4600 IF ISA$(I)("=") THEN 4620
4610 LOCATE 20,5:PRINT " * * * HATA * * * ":PRINT I: KISIT ISARETINI DOGRU G
IRINIZ": END
4620 FOR G=1 TO KWZ
4630 IF I(>G) THEN D(G,J)=0 ELSE D(G,J)=1
4640 NEXT G
4650 AD$(J)="A" + STR$(AA)
4660 F(J)=-9999
4670 IF KON=1 THEN RETURN
4680 J=J+1:AA=AA+1
4690 GOTO 4830
4700 FOR G=1 TO KWZ
4710 IF I(>G) THEN D(G,J)=0 ELSE D(G,J)=1
4720 NEXT G
4730 F(J)=0:AD$(J)="S"+STR$(II)
4740 J=J+1:II=II+1
4750 GOTO 4830
4760 FOR G=1 TO KWZ
4770 IF I(>G) THEN D(G,J)=0 ELSE D(G,J)=-1
4780 IF I(>G) THEN D(G,J+1)=0 ELSE D(G,J+1)=1
4790 NEXT G
4800 F(J)=0:AD$(J)="S"+STR$(II):AD$(J+1)="A"+STR$(AA)
4810 F(J+1)=-9999
4820 J=J+2 :II=II+1:AA=AA+1
4830 NEXT I
4840 KZZ=J-1
4850 IF KUT#="" THEN CLS:LOCATE 5,10:INPUT "Lutfen problem adini giriniz !!!":KU
T#:GOTO 4850
4860 SDSZ=DSZ+2*SZ+3
4870 KUT1#=KUT#+"3"

```

```

4880 CLOSE
4890 OPEN "d",#1,KUT1$
4900 WRITE #1,KWZ,KZZ+1,SDSZ,SZ,KSZ,DSZ
4910 FOR P=1 TO KZZ
4920 WRITE #1,F(P)
4930 NEXT P
4940 FOR P=1 TO KWZ
4950 FOR P1=1 TO KZZ
4960 WRITE #1,D(P,P1)
4970 NEXT P1
4980 WRITE #1,RHS(P)
4990 NEXT P
5000 FOR I=1 TO SZ
5010 WRITE #1,TIP(I)
5020 NEXT I
5030 FOR P=1 TO KZZ
5040 WRITE #1,AD$(P)
5050 NEXT P
5060 FOR P=1 TO KWZ
5070 WRITE #1,ISA$(P)
5080 NEXT P
5090 CLOSE #1
5100 RUN "p26"
5110 OPEN "i",#1,KUT$
5120 INPUT #1,KWZ,DSZ,KZZ,SZ,KSZ
5130 KZZ=KZZ-1
5140 FOR I=1 TO SZ
5150 INPUT #1,TIP(I)
5160 NEXT I
5170 FOR P=1 TO KZZ
5180 INPUT #1,F(P)
5190 NEXT P
5200 FOR P=1 TO KWZ
5210 FOR P1=1 TO KZZ
5220 INPUT #1,D(P,P1)
5230 NEXT P1
5240 INPUT #1,RHS(P)
5250 NEXT P
5260 FOR P=1 TO KZZ
5270 INPUT #1,AD$(P)
5280 NEXT P
5290 FOR P=1 TO KWZ
5300 INPUT #1,ISA$(P)
5310 NEXT P
5320 CLOSE #1
5330 RETURN

```

```

5350 CLS
5360 SCREEN 2
5370 BK=3:TEMHIZ=.001:YK=.4
5380 KEY OFF
5390 DIM D(40,90),F(90),IBV(90),SC(90),DS(90),B(10),DB(10),ALF(10)
5400 DIM YON(10),TIP(10),CIZ(10)
5410 GOSUB 6760
5420 GOSUB 6160
5430 GOSUB 6370
5440 GOSUB 6620
5450 GOSUB 6680
5460 GOSUB 6720
5470 CLS:LINE (240,140)-(440,180),,BF:LDCATE 12,30:PRINT "LUTFEN BEKLEYINIZ"
5480 GOSUB 5530:GOSUB 8150
5490 GOSUB 6990
5500 D=DT
5510 GOSUB 6760
5520 GOTO 5440
5530 FOR N=KY TO IX
5540 FOR L=1 TO KW
5550 IF (D(L,N)-1)= 0 THEN 5580
5560 NEXT L
5570 GOTO 5590
5580 IBV(L)=N
5590 NEXT N
5600 Z=0
5610 FOR M=1 TO KW
5620 IBVM=IBV(M)
5640 NEXT M
5650 NOTAB=1
5660 SCHAX=0
5670 FOR N=1 TO IX
5680 FOR I=1 TO KW
5690 IF (N-IBV(I))=0 THEN 5800
5700 NEXT I
5710 SUM=0!
5720 FOR I=1 TO KW
5730 J=IBV(I)
5740 SUM=SUM+F(J)*D(I,N)
5750 NEXT I
5760 SC(N)=F(N)-SUM
5770 IF (SC(N)+SCHAX)<=0 THEN 5800
5780 SCHAX=SC(N)
5790 IPIVCO=N
5800 NEXT N
5810 FOR M=1 TO KW
5820 IBVM=IBV(N-M)
5830 SC(IBVM)=0
5840 NEXT M
5850 ZZ#=Z
5860 IF (SCHAX-.000001)<=0 THEN 6150
5870 NOTAB=NOTAB+1
5880 SHLVAL#=-99999999#
5890 IPIVRO=0
5900 FOR M=1 TO KW

```

```

5910 IF (D(M,IPIVCO)-.000001) <=0 THEN 5960
5920 QUONT=D(M,K1)/D(0,IPIVRO)
5930 IF (QUONT-SHLVAL#)>=0 THEN 5960
5940 IPIVRO=M
5950 SHLVAL#=QUONT
5960 NEXT M
5970 IF IPIVRO>=0 THEN 6000
5980 PRINT TAB(15);"UNBOUNDED FEASIBLE REGION"
5990 GOTO 5840
6000 IBV(IPIVRO)=IPIVCO
6010 DIV=D(IPIVRO,IPIVCO)
6020 FOR N=1 TO K1
6030 D(IPIVRO,N)=D(IPIVRO,N)/DIV
6040 NEXT N
6050 FOR M=1 TO KW
6060 IF (M-IPIVRO)=0 THEN 6120
6070 CM=-D(M,IPIVCO)
6080 FOR N=1 TO K1
6090 TM=D(IPIVRO,N)*CM
6100 D(M,N)=D(M,N)+TM
6110 NEXT N
6120 NEXT M
6130 Z=Z+SHLVAL#*SCMAX
6140 GOTO 5660
6150 RETURN
6160 CLS:LOCATE 5,5
6170 IF KD=0 THEN PRINT "q VEKTORU ELEMAN DEGERLERINI GIRINIZ"
6180 IF KD=1 THEN PRINT "DUZELTME VE KONTROL"
6190 IF KD=1 THEN LOCATE 22,5:PRINT "SIFIRLANACAK ELEMAN icin 999 giriniz !!!"
6200 FOR I=1 TO SZ
6210 LOCATE 1+5,5:PRINT "q( ";I; ")=";:IF KD=1 THEN PRINT D(I)
6220 NEXT I
6230 FOR I=1 TO SZ
6240 IF KD=1 THEN 6260
6250 LOCATE 1+5,15:INPUT Q(I):GOTO 6300
6260 LOCATE 1+5,15:INPUT AL
6270 IF AL=0 THEN 6300
6280 IF AL=999 THEN Q(I)=0 :GOTO 6300
6290 Q(I)=AL
6300 NEXT I
6310 REM LINE (60,285)-(570,315),,BF
6320 LOCATE 22,10:PRINT "1- DUZELTME VE KONTROL 2- DEVAM "
6330 W%=INKEY$
6340 IF W%="1" THEN KD=1:GOTO 6160
6350 IF W%="2" THEN KD=0:RETURN
6360 GOTO 6330
6370 FOR I=1 TO SZ
6380 ALF(I)=Q(I)
6390 NEXT I
6400 RETURN
6410 CLS:LOCATE 5,5
6420 IF KD=0 THEN PRINT "YON VEKTORU ELEMANLARINI GIRINIZ"
6430 IF KD=1 THEN PRINT "DUZELTME VE KONTROL"

```



```

6440 IF KO=1 THEN LOCATE 22,5:PRINT "SIFIRLANACAK ELEMAN icin 999 giriniz !!!"
6450 FOR I=1 TO SZ
6460 LOCATE I+5,5:PRINT "YDN(";I;")=";:IF KO=1 THEN PRINT YDN(I)
6470 NEXT I
6480 YDN(1)=3:YDN(2)=10:YDN(3)=-1:RETURN
6490 FOR I=1 TO SZ
6500 IF KO=1 THEN 6520
6510 LOCATE I+5,15:INPUT YDN(I):GOTO 6560
6520 LOCATE I+5,15:INPUT AL
6530 IF AL=0 THEN 6560
6540 IF AL=999 THEN YDN(I)=0 :GOTO 6560
6550 YDN(I)=AL
6560 NEXT I
6570 LOCATE 20,10:PRINT "1- DUZELTME VE KONTROL          2- DEVAM      "
6580 W$=INKEY$
6590 IF W$="1" THEN KO=1:GOTO 6160
6600 IF W$="2" THEN KO=0:RETURN
6610 GOTO 6580
6620 DSZ=DSZ+2*SZ+1
6630 FOR I=1 TO SZ
6640 D(I,DSZ)=D(I,DSZ)*ALF(I)
6650 D(I,DSZ+1)=D(I,DSZ+1)*ALF(I)
6660 NEXT I
6670 RETURN
6680 FOR I=1 TO SZ
6690 D(I,KZ)=D(I)
6700 NEXT I
6710 RETURN
6720 FOR I=1 TO SZ
6730 D(I,KZ)=D(I,KZ)+D*YDN(I)
6740 NEXT I
6750 RETURN
6760 OPEN "i",#1,"kont.has"
6770 INPUT #1,KUT$
6780 CLOSE #1
6790 KUT1$=KUT$+"3"
6800 OPEN "I",#1,KUT1$
6810 INPUT #1 ,KW,KZ,KY,SZ,KSZ,DSZ
6820 IF KW>999999! THEN END
6830 IX=KZ-1
6840 FOR M=1 TO IX
6850 INPUT #1,F(M)
6870 NEXT M
6880 REM PRINT
6890 FOR M=1 TO KW
6900 FOR N=1 TO KZ
6910 INPUT #1,D(M,N)
6920 NEXT N
6930 NEXT M
6940 FOR I=1 TO SZ
6950 INPUT #1,TIP(I)
6960 NEXT I
6970 CLOSE #1
6980 RETURN

```

```

6990 BDSUB 7530:LOCATE 18,2:PRINT "1- SABIT HIZ"
7000 LOCATE 18,15:PRINT "2- ILERIYE DOGRU HIZ"
7010 LOCATE 18,37:PRINT "3- GERIYE DOGRU HIZ"
7020 LOCATE 18,60:PRINT "4- PARAMETRE DEGT."
7030 LOCATE 19,2 :PRINT "5- YON DEGISTIRME "
7040 LOCATE 19,22:PRINT "6- COZUM VEKTORU "
7050 LOCATE 19,42:PRINT "7- OTOMATIK YON DEG."
7060 LOCATE 19,65:PRINT "8- CIXIS "
7070 LOCATE 20,10:PRINT "SECIMINIZ (1/2/3/4/5/6/7/8)?";
7080 LINE (1,236)-(700,285),,8
7090 W$=INKEY$
7100 IF W$="1" THEN 7190
7110 IF W$="2" THEN 7210
7120 IF W$="3" THEN 7230
7130 IF W$="4" THEN 7250
7140 IF W$="5" THEN 7570
7150 IF W$="6" THEN 7880
7160 IF W$="7" THEN 8630
7170 IF W$="8" THEN RUN "amenu"
7180 GOTO 7090
7190 QT=QT+.001
7200 RETURN
7210 QT=QT+QK*TEMHIZ
7220 RETURN
7230 QT=QT-QK*TEMHIZ :IF QT<0 THEN QT=TEMHIZ
7240 RETURN
7250 BDSUB 7530
7260 LOCATE 18,5
7270 PRINT "ADIM BUYUKLUGU=";
7280 PRINT USING "##.###";TEMHIZ
7290 LOCATE 19,5
7300 PRINT "BUYULTME KATS.=";
7310 PRINT USING "###";QK
7320 LOCATE 20,5
7330 PRINT "YON DEGT.KATS.=";
7340 PRINT USING "##.###";YK
7350 LOCATE 18,28
7360 INPUT YTEMHIZ
7370 IF YTEMHIZ=0 THEN 7390
7380 TEMHIZ=YTEMHIZ
7390 LOCATE 19,28
7400 INPUT YQK
7410 IF YQK=0 THEN 7430
7420 QK=YQK
7430 LOCATE 20,28
7440 INPUT YYK
7450 IF YYK=0 THEN 7470
7460 YK=YYK
7470 LOCATE 19,40:PRINT STRING$(40," ")
7480 LOCATE 19,40:PRINT "1- DUZELTME 2- DEVAM "
7490 W$=INKEY$
7500 IF W$="1" THEN 7250
7510 IF W$="2" THEN BDSUB 7530 :GOTO 5490
7520 GOTO 7490
7530 FOR I=18 TO 25

```

```

7540 LOCATE 1,1:PRINT STRING$(80," ");
7550 NEXT 1
7560 RETURN
7570 CLS
7580 FOR I=1 TO SZ
7590 B(I)=DB(I)
7600 NEXT I
7610 LOCATE 2,1:PRINT "AMAC FONK. NO"
7620 LOCATE 2,17:PRINT "DEGERLERI"
7630 LOCATE 3,1 :PRINT " _____"
7640 LOCATE 3,17:PRINT " _____"
7650 FOR I=1 TO SZ
7660 LOCATE 3+I,5:PRINT I
7670 LOCATE 3+I,17
7680 PRINT USING "#####.###";B(I)
7690 NEXT I
7700 QT=0
7710 LOCATE 2,30:PRINT "YENI REFERANS DEGL. ?"
7720 LOCATE 3,30:PRINT " _____"
7730 FOR I=1 TO SZ
7740 LOCATE 3+I,31
7750 INPUT YDEB
7760 IF YDEB=0 THEN ALF(I)=B(I) :GOTO 7780
7770 ALF(I)=YDEB
7780 NEXT I
7790 LOCATE 22,10:PRINT "1- DUZELTME      2- DEYAM "
7800 W$=INKEY$
7810 IF W$="1" THEN 7570
7820 IF W$="2" THEN 7840
7830 GOTO 7800
7840 FOR I=1 TO SZ
7850 YON(I)=ALF(I)-B(I)
7860 NEXT I
7870 RETURN
7880 GOSUB 7530
7890 K4$=KUT$+"E"
7900 OPEN "C",#1,K4$
7910 SS=0:AA1=1:II=0
7920 FOR I=1 TO INT(KZ/5)+1
7930 IF II>5 THEN II=0:GOTO 8070
7940 II=II+1
7950 IF SS>5 THEN 8070
7960 AA=1
7970 FOR J=1 TO 5
7980 IF AA1>KZ THEN 8090
7990 LOCATE 18+II,AA:PRINT "X";AA1;
8000 WRITE #1,"X",AA1,DS(AA1)
8010 AA=AA+5
8020 LOCATE 18+II,AA
8030 PRINT USING "#####.###";DS(AA1);
8040 AA=AA+11:AA1=AA1+1
8050 NEXT J
8060 GOTO 8090
8070 LOCATE 25,10 :INPUT "SAYFANIN DEVAMI ICIN BIR TUSA BASINIZ !!.";T$
8080 SS=0 :GOSUB 7530

```

```

8090 NEXT I
8100 CLOSE #1
8110 LOCATE 24,10:PRINT "MENU ICIN BIR TUSA BASINIZ !!!";
8120 W$=INKEY$
8130 IF W$(">") THEN 8990
8140 GOTO 8120
8150 LO=0
8160 CLS
8170 FOR I=1 TO K2
8180 DS(I)=0
8190 NEXT I
8200 FOR M=1 TO KM
8210 LL=IBV(M)
8220 DS(LL)=D(M,K2)
8230 NEXT M
8240 FOR I=1 TO KM
8250 NEXT I
8260 REM PRINT
8270 FOR I=1 TO SZ
8280 IF DS(I+DSZ+LO+1)<>0 THEN DB(I)=DS(I+DSZ+LO+1) ELSE DB(I)=DS(I+DSZ+LO)
8290 LO=LO+1
8300 NEXT I
8310 FOR I=1 TO SZ
8320 IF MAK(DB(I)) THEN MAK=DB(I)
8330 NEXT I
8340 BOY=180/MAK
8350 IF BOY>100 THEN BOY=90:GOTO 8430
8360 IF BOY>50 THEN BOY=40:GOTO 8430
8370 IF BOY>30 THEN BOY=20:GOTO 8430
8380 IF BOY>20 THEN BOY=15:GOTO 8430
8390 IF BOY>10 THEN BOY=9:GOTO 8430
8400 IF BOY>5 THEN BOY=4:GOTO 8430
8410 IF BOY>2 THEN BOY=1.5:GOTO 8430
8420 IF BOY>1 THEN BOY=1
8430 FOR I=1 TO SZ
8440 CIZ(I)=DB(I)*BOY
8450 NEXT I
8460 CLS
8470 XK=30
8480 FOR I=1 TO SZ
8490 UZ=200-CIZ(I)
8500 LINE (XK,UZ)-(XK+30,UZ+CIZ(I)),,BF
8510 YC=XK\9+1
8520 LOCATE 16,YC-2
8530 PRINT USING "###.##";DB(I)
8540 XK=XK+80
8550 LOCATE 1,YC+1
8560 PRINT USING "##";I
8570 NEXT I
8580 LINE (30,20)-(30+XK-80,200),,B
8590 LOCATE 1,1:PRINT "AMAC"
8600 LOCATE 17,1:PRINT "Bt=";
8610 PRINT USING "##.###";0
8620 RETURN
8630 FOR I=1 TO SZ

```

```

8640 YQN(I)=QB(I)+YK
8650 NEXT I
8660 GOTO 8990
8670 KEY OFF
8680 CLS
8690 LINE (50,10)-(550,180),,BF
8700 LOCATE 3,15:PRINT ".MODEL..DUZELTME-DEGISTIRME..MENUSU.."
8710 LOCATE 5,15:PRINT "1==>AMAC.MENUSU....."
8720 LOCATE 6,15:PRINT "2==>KISIT.MENUSU....."
8730 LOCATE 7,15:PRINT "3==>DEGISKEN.MENUSU....."
8740 LOCATE 8,15:PRINT "4==>ANA.MENUYE.DONUS....."
8750 LOCATE 10,15:PRINT ".....SECIMINIZ=====>"
8760 W$=INKEY$
8770 IF W$="1" THEN RUN "ADUZ.BAS"
8780 IF W$="2" THEN RUN "KDUZ.BAS"
8790 IF W$="3" THEN RUN "DDUZ.BAS"
8800 IF W$="4" THEN RUN "AMENU.BAS"
8810 GOTO 8760
8820 KEY OFF
8830 DIM A(9,100),AD$(100)
8840 OPEN "I",#1,"kent.bas"
8850 INPUT #1,KUT$
8860 CLOSE #1
8870 KUTA$=KUT$+"1"
8880 OPEN "I",#1,KUTA$
8890 INPUT #1,ASZ,KZZ
8900 FOR I=1 TO ASZ
8910 INPUT #1,TIP(I)
8920 NEXT I
8930 FOR P=1 TO ASZ
8940   FOR P1=1 TO KZZ
8950     INPUT #1,A(P,P1)
8960   NEXT P1
8970 NEXT P
8980 FOR P=1 TO KZZ
8990 INPUT #1,AD$(P)
9000 NEXT P
9010 CLOSE #1
9020 CLS
9030 LINE (50,20)-(550,180),,BF
9040 LOCATE 4,15:PRINT "*****"
9050 LOCATE 5,15:PRINT "###      AMAC      MENUSU      ###"
9060 LOCATE 6,15:PRINT "### 1- TEK AMAC EKLEME      ###"
9070 LOCATE 7,15:PRINT "### 2- TEK AMAC CIKARTMA      ###"
9080 LOCATE 8,15:PRINT "### 3- AMAC DUZELTME MENUSU      ###"
9090 LOCATE 9,15:PRINT "### 4- MODEL DUZ.-DEG. MENUSU      ###"
9100 LOCATE 10,15:PRINT "*****"
9110 LOCATE 12,15:PRINT "***** SECIMINIZ ##[1/2/3/4]*****"
9120 W$=INKEY$
9130 IF W$="1" THEN 9180
9140 IF W$="2" THEN 9210
9150 IF W$="3" THEN 9300
9160 IF W$="4" THEN 9800
9170 GOTO 9120
9180 ASZ=ASZ+1 :J=ASZ

```

```

9190 GOSUB 10240
9200 BOTO 9020
9210 CLS:LOCATE 5,5:INPUT "KACINCI AMAC=";J1
9220 IF J1>ASZ THEN 9210
9230 FOR I=J1 TO ASZ
9240 FOR J=1 TO KZZ
9250 A(I,J)=A(I+1,J)
9260 NEXT J
9270 NEXT I
9280 ASZ=ASZ-1
9290 BOTO 9020
9300 GOSUB 9530
9310 BOTO 9020
9320 FOR J=1 TO ASZ :CLS
9330 LOCATE 2,1:PRINT J;". AMAC :";
9340 IF TIP(J)=1 THEN PRINT "MAX"; ELSE PRINT "MIN";
9350 II=1:III=1
9360 FOR I=1 TO KZZ
9370 LOCATE 2+II,3+III:PRINT "_____";AD$(I)
9380 III=III+16:IF III>64 THEN III=1:II=II+2
9390 NEXT I
9400 II=1:III=1
9410 FOR I=1 TO KZZ
9420 LOCATE 2+II,3+III:INPUT A(J,I)
9430 III=III+16:IF III>64 THEN III=1:II=II+2
9440 NEXT I
9450 LOCATE 22,10 :PRINT "1- DUZELTME - 2- DEVAM"
9460 LINE (60,290)-(570,310),,B
9470 W$=INKEY$
9480 IF W$="1" THEN 9510
9490 IF W$="2" THEN 9520
9500 GOTO 9470
9510 GOSUB 9990
9520 NEXT J
9530 CLS
9540 LINE (70,25)-(550,170),,B
9550 LOCATE 4,15:PRINT "*****"
9560 LOCATE 5,15:PRINT "### AMAC DUZELTME MENUU ###"
9570 LOCATE 6,15:PRINT "### 1- TEK AMACI KONTROL VE DUZELTME ###"
9580 LOCATE 7,15:PRINT "### 2- TEK TEK KONTROL VE DUZELTME ###"
9590 LOCATE 8,15:PRINT "### 3- DEVAM ###"
9600 LOCATE 9,15:PRINT "*****"
9610 LOCATE 11,15:PRINT "***** SECIMINIZ **[1/2/3]**"
9620 W$=INKEY$
9630 IF W$="1" THEN 9670
9640 IF W$="2" THEN 9730
9650 IF W$="3" THEN RETURN
9660 GOTO 9620
9670 CLS
9680 LOCATE 5,5 :PRINT "KACINCI AMAC=";:INPUT J
9690 IF J>ASZ THEN 9680
9700 KD=1
9710 GOSUB 9990
9720 BOTO 9530
9730 FOR J=1 TO ASZ

```

```

9740 KD=1
9750 GOSUB 9990
9760 NEXT J
9770 GOTO 9530
9780 IF MM=2 THEN RETURN
9790 IF KUT#="" THEN CLS:LOCATE 5,10:INPUT "Lutfen problem adini giriniz !!!";KU
T#:GOTO 9790
9800 OPEN "o",#1,"kont.bas"
9810 WRITE #1,KUT#
9820 CLOSE #1
9830 KUTA#=KUT#+1
9840 OPEN "o",#1,KUTA#
9850 WRITE #1,ASZ,KZZ
9860 FOR I=1 TO ASZ
9870 WRITE #1,TIP(I)
9880 NEXT I
9890 FOR P=1 TO ASZ
9900 FOR P1=1 TO KZZ
9910 WRITE #1,A(P,P1)
9920 NEXT P1
9930 NEXT P
9940 FOR P=1 TO KZZ
9950 WRITE #1,AD$(P)
9960 NEXT P
9970 CLOSE #1
9980 RUN "AKDD.BAS"
9990 CLS
10000 LOCATE 2,1:IF TIP(J)=1 THEN PRINT J;". AMAC =MAX"; ELSE PRINT J;". AMAC =M
IN";
10010 II=1:III=1
10020 FOR I=1 TO KZZ
10030 LOCATE 2+II,3+III:PRINT A(J,I);:FF=LEN (STR$(A(J,I))):PRINT STRING$(13-FF,
" ");:LOCATE 2+II,16-FF+III:PRINT AD$(I)
10040 III=III+16:IF III>64 THEN III=1:II=II+2
10050 NEXT I
10060 IF KD=1 THEN KD=0:GOTO 10180
10070 IF W#="" THEN 10180
10080 LOCATE 22,10 :PRINT "!!!!!! sifirlanacak katsayi icin 999 giriniz ! !!!!!"
10090 LINE (60,290)-(550,310),,B
10100 II=1:III=1
10110 FOR I=1 TO KZZ
10120 LOCATE 2+II,2+III:INPUT FD
10130 IF FD=999 THEN A(J,I)=0 :GOTO 10160
10140 IF FD=0 THEN 10160
10150 A(J,I)=FD
10160 III=III+16:IF III>64 THEN III=1:II=II+2
10170 NEXT I
10180 LOCATE 22,10 :PRINT "1- DUZELTME          2- DEVAM
10190 LINE (60,290)-(380,310),,B
10200 W#=INKEY$
10210 IF W#="1" THEN 9990
10220 IF W#="2" THEN RETURN
10230 GOTO 10200
10240 CLS:LOCATE 2,1:PRINT J;". AMAC TIPI :";:INPUT TIP(J):CLS
10250 CLS:LOCATE 2,1:PRINT J;". AMAC :";
10260 IF TIP(J)=1 THEN PRINT "MAX"; ELSE PRINT "MIN";
10270 II=1:III=1
10280 FOR I=1 TO KZZ

```

```

10290 LOCATE 2+II,3+III:PRINT "_____";AD$(I)
10300 III=III+16:IF III>64 THEN III=1:II=II+2
10310 NEXT I
10320 II=1:III=1
10330 FOR I=1 TO KZZ
10340 LOCATE 2+II,3+III:INPUT A(J,I)
10350 III=III+16:IF III>64 THEN III=1:II=II+2
10360 NEXT I
10370 LOCATE 22,10 :PRINT "1- DUZELTME      2- DEVAM"
10380 LINE (60,290)-(550,310),,B
10390 W$=INKEY$
10400 IF W$="1" THEN 10430
10410 IF W$="2" THEN RETURN
10420 GOTO 10390
10430 GOSUB 9990
10440 KEY OFF
10450 DIM D(60,100),AD$(100),ISA$(100),RHS(100)
10460 OPEN "I",#1,"kont.bas"
10470 INPUT #1,KUT$
10480 CLOSE #1
10490 KUTK$=KUT$+"2"
10500 OPEN "I",#1,KUTK$
10510 INPUT #1,KWZ,KZZ
10520 FOR P=1 TO KWZ
10530 FOR P1=1 TO KZZ
10540 INPUT #1,D(P,P1)
10550 NEXT P1
10560 INPUT #1,ISA$(P),RHS(P)
10570 NEXT P
10580 FOR P=1 TO KZZ
10590 INPUT #1,AD$(P)
10600 NEXT P
10610 CLOSE #1
10620 CLS
10630 LINE (70,20)-(550,180),,BF
10640 LOCATE 4,15:PRINT "*****"
10650 LOCATE 5,15:PRINT "###      KISIT  MENUSU      ###"
10660 LOCATE 6,15:PRINT "### 1- TEK KISIT EKLEME      ###"
10670 LOCATE 7,15:PRINT "### 2- TEK KISIT CIKARTMA      ###"
10680 LOCATE 8,15:PRINT "### 3- KISIT DUZELTME MENUSU      ###"
10690 LOCATE 9,15:PRINT "### 4- MODEL DEG.-DUZ. MENUSU      ###"
10700 LOCATE 10,15:PRINT "*****"
10710 LOCATE 12,15:PRINT "***** SECIMINIZ **[1/2/3/4]*****"
10720 W$=INKEY$
10730 IF W$="1" THEN 10780
10740 IF W$="2" THEN 10820
10750 IF W$="3" THEN 10920
10760 IF W$="4" THEN 11250
10770 GOTO 10720
10780 KWZ=KWZ+1 :I=KWZ:NM=2
10790 ISA$(I)="">"
10800 GOSUB 10940:NM=0
10810 GOTO 10620

```



```

10820 CLS:LOCATE 5,5:INPUT "KACINCI KISIT=";I1
10830 IF I1>KWZ THEN 10820
10840 FOR I=I1 TO KWZ
10850   RHS(I)=RHS(I+1)
10860   FOR J=1 TO KZZ
10870     D(I,J)=D(I+1,J)
10880   NEXT J
10890 NEXT I
10900 KWZ=KWZ-1
10910 GOTO 10620
10920 GOSUB 12040
10930 GOTO 10620
10940 CLS
10950 I1=1
10960 LOCATE I1,1:PRINT I1;" KISIT:"
10970 I1=I1+1 :III=1
10980 FOR J=1 TO KZZ
10990   LOCATE 2+I1,3+III:PRINT "_____";AD$(J)
11000   III=III+16:IF III>64 THEN III=1:II=II+1
11010 NEXT J
11020 LOCATE 2+I1,5+III:PRINT ISA$(I1);" _____"
11030 II=2
11040 II=II+1 :III=1
11050 FOR J=1 TO KZZ
11060   LOCATE 1+II,3+III:INPUT D(I,J)
11070   III=III+16:IF III>64 THEN III=1:II=II+1
11080 NEXT J
11090 LOCATE 1+II,3+III:INPUT S$
11100 IF S$="" THEN 11130
11110 IF S$="=" OR S$=">" OR S$="<" THEN 11120 ELSE 11090
11120 ISA$(I)=S$
11130 LOCATE 1+II,7+III:INPUT RHS(I)
11140 II=II+1:III=1
11150 IF II>20 THEN 11160
11160 LOCATE 22,10:PRINT "1- DUZELTME      2- DEVAM"
11170 LINE (60,290)-(380,310),,B
11180 W$=INKEY$
11190 IF W$="1" THEN 11220
11200 IF W$="2" THEN 11230
11210 GOTO 11180
11220 GOSUB 11640
11230 IF W$="2" THEN RETURN
11240 GOSUB 12040
11250 OPEN "a",#1,KUTK$
11260 WRITE #1,KWZ,KZZ
11270 PRINT KWZ,KZZ
11280 FOR P=1 TO KWZ
11290   FOR P1=1 TO KZZ
11300     WRITE #1,D(P,P1)
11310   NEXT P1
11320   WRITE #1,ISA$(P),RHS(P)
11330 NEXT P
11340 FOR P=1 TO KZZ

```

```

11350 SCREEN 0,2
11360 DIM A(50,100),D(50,100),AD$(100),ISA$(100),RHS(100)
11370 KEY OFF
11380 ON ERROR GOTO 12420
11390 OPEN "I",#1,"kont.bas"
11400 INPUT #1,KUT$
11410 CLOSE #1
11420 IF KUT$=" " THEN 12420
11430 KUTA$=KUT$+"1"
11440 OPEN "I",#1,KUTA$
11450 INPUT #1,ASZ,KZZ
11460 FOR I=1 TO ASZ
11470 INPUT #1,TIP(I)
11480 NEXT I
11490 FOR P=1 TO ASZ
11500 FOR P1=1 TO KZZ
11510 INPUT #1,A(P,P1)
11520 NEXT P1
11530 NEXT P
11540 FOR I=1 TO KZZ
11550 INPUT #1,AD$(I)
11560 NEXT I
11570 CLOSE #1
11580 KUTK$=KUT$+"2"
11590 OPEN "I",#1,KUTK$
11600 INPUT #1,KNZ,KZZ
11610 FOR P=1 TO KNZ
11620 FOR P1=1 TO KZZ
11630 INPUT #1,D(P,P1)
11640 NEXT P1
11650 INPUT #1,ISA$(P),RHS(P)
11660 NEXT P
11670 FOR P=1 TO KZZ
11680 INPUT #1,AD$(P)
11690 NEXT P
11700 CLOSE #1
11710 CLS :LOCATE 5,5:INPUT "YAZICIDAN DA ISTIYOR MUSUNUZ (E/H) ?";C$
11720 IF C$<>"E" OR C$<>"e" THEN 11760
11730 CLS:LOCATE 12,28:PRINT "YAZICI ACIK DEBIL!!!!!"
11740 LPRINT
11750 LPRINT CHR$(27);"3";CHR$(15);TAB(10);"ANAC(LAR):"
11760 FOR J=1 TO ASZ
11770 GOSUB 11950
11780 IF C$="E" OR C$="e" THEN GOSUB 12120
11790 IF C$="E" OR C$="e" THEN GOTO 11810
11800 LOCATE 20,5 :INPUT "DEVAM ICIN BIR TUSA BASINIZ";T$
11810 NEXT J
11820 IF C$="E" THEN LPRINT:LPRINT:LPRINT
11830 IF C$="E" THEN LPRINT TAB(10);"KISIT(LAR):" :LPRINT
11840 FOR I=1 TO KNZ
11850 GOSUB 12030
11860 IF C$="E" THEN GOSUB 12250
11870 IF C$="E" THEN GOTO 11890

```

```

1880 LOCATE 20,5 :INPUT "DEVAM ICIN BIR TUSA BASINIZ";T$
1890 NEXT I
1900 CLS:LOCATE 22,5:PRINT "1-GORUNTULEME      2-ANA MENU"
1910 W$=INKEY$
1920 IF W$="1" THEN RUN
1930 IF W$="2" THEN RUN "AMENU"
1940 GOTO 11910
1950 CLS
1960 LOCATE 2,1:IF TIP(J)=1 THEN PRINT J;" . AMAC =MAX"; ELSE PRINT J;" . AMAC =M
"N";
1970 II=1:III=1
1980 FOR I=1 TO KZZ
1990 LOCATE 2+II,3+III:PRINT A(J,I);:FF=LEN (STR$(A(J,I))):PRINT STRING$(13-FF,
"_");:LOCATE 2+II,16-FF+III:PRINT AD$(I)
2000 III=III+16:IF III>64 THEN III=1:II=II+2
2010 NEXT I
2020 RETURN
2030 II=2      :III=1 :CLS:LOCATE 11,III:PRINT I;" . KISIT"
2040 FOR J=1 TO KZZ
2050 LOCATE 2+II,3+III:PRINT D(I,J);:FF=LEN(STR$(D(I,J))):PRINT STRING$(16-FF
,"_");:LOCATE 2+II,16-FF+III:PRINT AD$(J)
2060 III=III+16:IF III>64 THEN III=1:II=II+1
2070 NEXT J
2080 LOCATE 2+II,5+III:PRINT ISA$(I);" -----"
2090 LOCATE 2+II,8+III:PRINT RMS(I)
2100 II=II+1:III=1
2110 RETURN
2120 LPRINT
2130 LPRINT TAB(2);:IF TIP(J)=1 THEN LPRINT "MAX"; ELSE LPRINT "MIN";
2140 II=1:III=6
2150 FOR I=1 TO KZZ
2160 IF I=1 THEN 12180
2170 IF SGN(A(J,I))>0 THEN LPRINT "+";:III=III+1
2180 IF A(J,I)=0 THEN 12230
2190 LPRINT TAB(III);A(J,I);
2200 FF=LEN (STR$(A(J,I))):III=III+FF+1
2210 LPRINT TAB(III);AD$(I);
2220 III=III+3:IF III>64 THEN III=1
2230 NEXT I :LPRINT
2240 RETURN
2250 III=1 :LPRINT :LPRINT TAB(III);1;" . KISIT" ;
2260 III=11
2270 FOR J=1 TO KZZ
2280 IF J=1 THEN 12300
2290 IF SGN(D(I,J))>0 THEN LPRINT "+";:III=III+1
2300 IF D(I,J)=0 THEN 12350
2310 LPRINT TAB(III);D(I,J);
2320 FF=LEN(STR$(D(I,J))):III=III+FF+1
2330 LPRINT TAB(III);AD$(J);
2340 III=III+3:IF III>64 THEN III=1
2350 NEXT J
2360 LPRINT TAB(III);ISA$(I);
2370 PRINT TAB(III);ISA$(I);
2380 LPRINT/(3+III);RMS(I)
2390 PRINT (3+III);RMS(I)
2400 LPRINT

```

```

.2210 KEY OFF
.2220 OPEN "i",#1,"kont.bas"
.2230 INPUT #1,KUT$
.2240 CLOSE #1
.2250 K4$=KUT$+"c"
.2260 CLS
.2270 OPEN "i",#1,K4$
.2280 LOCATE 2,5 :PRINT "## Cozum Vektoru Elemanlari ##"
.2290 PRINT
.2300 IF EOF(1) THEN 12420
.2310 INPUT #1,X$,AA1,DS
.2320 SS=SS+1
.2330 PRINT X$;
.2340 PRINT USING "##";AA1;
.2350 PRINT "=";
.2360 PRINT USING "####.###";DS;
.2370 PRINT " ";
.2380 IF INT(SS/4)*4=SS THEN PRINT
.2390 IF SS>22 THEN LOCATE 24,10 :INPUT "SAYFANIN DEVAMI ICIN BIR TUSA BASINIZ !
!.";T$:CLS :GOTO 12410
.2400 GOTO 12300
.2410 SS=0 :GOTO 12300
.2420 LOCATE 24,10 :INPUT "ANA MENUVE DONMEK ICIN RETURN GIRINIZ!";T$
.2430 RUN "AMENU"

```

ÖZGEÇMİŞ

1958 yılında Kilis'te doğan araştırmacı , ilk ve orta öğrenimini aynı kentte sırasıyla Hürriyet ilkokulu, Kilis Ortaokulu ve Kilis Lisesi'nde tamamladı. Aynı yıl ODTÜ Fen Edebiyat Fakültesi Matematik bölümüne girdi. Bu bölümün "Numerical Analysis and Computational Technics" opsiyonundan 1981 yılında mezun oldu. Ege Üniversitesi Elektronik Hesap Bilimlerinde Master eğitimine başladı. 1982 Martında AİTİA'da asistanlık sınavlarını kazanarak bu kuruma geçti. Gazi Üniversitesi Fen Bilimleri Enstitüsünde yüksek lisans eğitimini 1984 yılında tamamladı. Halen aynı Üniversitenin Fen - Edebiyat fakültesi İstatistik bölümünde araştırma görevlisidir.