KARADENİZ TEKNİK ÜNİVERSİTESİ * SOSYAL BİLİMLER ENSTİTÜSÜ

EKONOMETRİ ANABİLİM DALI

EKONOMETRİ PROGRAMI

DOĞRUSAL HEDEF PROGRAMLAMA VE SANAL BİR TEKSTİL FİRMASINDA UYGULAMA DENEMESİ

YÜKSEK LİSANS TEZİ

Müyesser CANDEĞER ŞİRİN

Danışman : Prof. Dr. Hilmi ZENGİN

Haziran-2010

TRABZON

ONAY

MÜYESSER Candeğer ŞİRİN tarafından hazırlanan Doğrusal Hedef Programlama ve Sanal Bir Tekstil Firmasında Uygulama Denemesi adlı bu çalışma ……………….. tarihinde yapılan savunma sınavı sonucunda …………………………ile başarılı bulunarak jürimiz tarafından Ekonometri Anabilim dalında **yüksek lisans tezi** olarak kabul edilmiştir.

……….

……………………………. (Başkan)

……….

……………………………. (Danışman)

……….

…………………………….

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduklarını onaylarım.

... / ... / ….

……………………………..

Enstitü Müdürü

BİLDİRİM

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını, aksinin ortaya çıkması durumunda her tür yasal sonucu kabul ettiğimi beyan ediyorum.

…………..

Müyesser Candeğer ŞİRİN

…./…./……

Önsöz

Bir karar mekanizması olan Yöneylem Araştırmasının amacı; amaçları belirli koşullar ve kısıtlar altında eniyilemektir. Çok amaçlı karar almanın ana fikri; karar alıcı tarafından her bölüm içindeki problemlerin anlaşılabilir olmasının sağlanması ve bu problemlerin çözüme ulaştırılması seklinde açıklanabilir.

Günümüz işletme yönetimlerinde, yaşanan sorunların çözümünde ve ulaşılmak istenen hedeflere ulaşmada çok amaçlı karar verme teknikleri sıklıkla başvurulan yöntemlerin başında gelir. Artık tek amaca yönelik çözümler hedeflere ulaşmada yetersiz kalmakta bu tür problemlerin çözümünde "Hedef Programlama Yöntemi" kullanılmaktadır.

Bu çalışmada yöneylem araştırmasının doğuşundan başlanarak tarihsel gelişimi, tanımı, türleri, ikinci bölümde Hedef Programlamanın temelini oluşturan Doğrusal Programlama ayrıntılı olarak anlatılarak üçüncü bölümde Hedef Programlamanın tanımı, doğuşu, varsayımları, türleri, matematiksel modeli oluştururken uygulanan adımlardan bahsedilecektir. Uygulama kısmında ise günümüz işletmelerinde çok amaçlı karar verme yöntemlerinin hedeflere ulaşmada ne kadar önemli olduğunu ortaya koymak için Doğrusal Hedef Programlama yöntemi kullanılarak sanal bir tekstil firmasında uygulama denemesi yapılacaktır.

Bu tez K.T.Ü İktisadi ve İdari Bilimler Fakültesi Ekonometri Anabilim Dalı Yüksek Lisans Programında Prof. Dr. Hilmi Zengin yönetiminde gerçekleştirilmiştir. Bu çalışmanın hazırlanmasında benden desteğini ve yardımlarını esirgemeyen danışman hocam Prof. Dr. Hilmi Zengin'e en içten minnettarlığımı belirtmek isterim. Yine bu çalışmanın her aşamasında benden desteklerini esirgemeyen ve hep yanımda olan çok sevdiğim annem, babam ve sevgili eşim Tayfun Şirin'e gönülden teşekkürlerimi sunarım.

Haziran, 2010 Müyesser CANDEĞER ŞİRİN

İÇİNDEKİLER

BİRİNCİ BÖLÜM

İKİNCİ BÖLÜM

ÜÇÜNCÜ BÖLÜM

DÖRDÜNCÜ BÖLÜM

ÖZET

İnsanlar gerek günlük gerekse is yaşamlarında kararlar alarak hayatlarına devam etmektedirler. Bu kararlar kimi zaman önemsiz kararlar olmakla birlikte çoğu kez oldukça önemli sonuçlara neden olmaktadır. Bir kararın verilebilmesi için birden fazla seçeneğin olması gerekmektedir. Böylece karar verecek olan kişi faydası en fazla olan seçeneği seçerek kararını belirleyecektir.

Çalışmanın amacı, çağın ilerlemesiyle birlikte gelişen teknolojinin insanların gereksinimlerinin artması ve karmaşıklaşması sonucu ortaya çıkan problemlerin çözümünde yaygın olarak kullanılan bir yöntem olan Hedef Programlama yönteminin kullanımını ve amacını anlatmaya çalışmaktır.

Çalışmanın teori kısmında Doğrusal Hedef programlama, çözüm yöntemleri ve türleri açıklanmaya çalışılacaktır.

Uygulama kısmında ise, sanal bir işletme olan KAR tekstil firması üzerinde Hedef Programlama yöntemi uygulanmıştır. Yöntem uygulanırken, yönetimin isteklerine göre Ağırlıklı Doğrusal Hedef Programlama ve Eşit Ağırlıklı Doğrusal Hedef Programlama modelleri tercih edilmiştir. Modeller Lingo paket programı ile çözülmeye çalışılıp, işletmenin karar verme modellerinin çözümleri tartışılacaktır.

Anahtar Kelimeler: Yöneylem Araştırması, Doğrusal Programlama, Hedef Programlama, Lingo.

ABSTRACT

People must make decisions in everyday life as well as life is to continue. These decisions, although sometimes insignificant decisions are often quite important consequences. To be given a decision must be more than one option. Thus, people who have to decide which option to choose a maximum of benefit will determine the decision.

The aim of our study, with age developing technology evolves, the needs of the people to rise and become complex as a result of problems arising in the solution is widely used as a method of goal programming method with the use and purpose is to try to explain it.

Part of the study in the theory of linear goal programming, solution methods and types will be tried to explain.

In the second part is a virtual business that KAR textile firms were applied on the target programming methods. Method is applied, according to management's request and Equal-Weighted-Weighted Linear Goal Programming models were preferred Linear Goal Programming. Models have been tried to be solved with Lingo package program and the solution of business decision-making models are discussed.

Key Words: Operation Research, Linear Programming, Goal Programming, Lingo.

TABLOLAR LİSTESİ

GRAFİKLER LİSTESİ

KISALTMALAR LİSTESİ

a	: Hedefin Üstünde Kalan Sapma Değişkenini Gösterir. ($D_İ^+$)
a_i	: Teknoloji Katsayıları (Karar Değişkenlerinin Üretimi İçin Gerekli Kaynak Miktarları)
a_{ij}	: İ. Kıt Kaynağın, J. Madde Üretimi İçin Kullanılması Gereken Miktarı
b	: Sınırlı Kaynak Miktarları (Kaynak Kapasiteleri)
b_i	: İ. Kıt Kaynağın Kullanılabilir Miktarı (Sağ Taraf Sabitleri), Katsayısı
c	: Karar Değişkenlerinin Amaç Fonksiyonuna Katkısı (Gelir Veya Masraf)
C.V.	: Çözüm Vektörü
d_i^-	: Hedeften Negatif Sapma Değeri
d_i^+	: Hedeften Pozitif Sapma Değeri
e	: Hedefin Altında Kalan Sapma Değişkenini ($D_İ^-$)
kg	: Kilogram
K.K.	: Kâr Katsayısı
Lındo	: Linear Interactive And Discrete Optimizer
M	: Yüksek Maliyet Faktörü
Min	: Minimizasyon
Max	: Maksimizasyon
M.K.	: Maliyet Katsayısı
P_i	: İ Hedefine Verilen Öncelik, ($Pi >>> P_{i+1}$)
S	: Aylak Değişken
T.D.V.	: Toplam Değişken Vektör
$\overline{X}$	: Karar Değişkenleri Vektörü
x_{ij}	: Karar Değişkenleri
w_i	: İ. Hedefin Sapma Değişkenlerine Verilmiş Olan Matematiksel Ağırlıklar (Diferansiyel Ağırlık)
Z	: En Büyük Veya En Küçük Yapılacak Olan Amaç Fonksiyonu Değeri

GİRİŞ

Karar alma, yüzyıllardır var olan ve insanın düşünce yapısının değişmesiyle sürekli gelişen bir kavramdır. Başlangıçta tek amaç doğrultusunda alınan kararlar, amaç ve tercih değerlerindeki farklılaşmaların yarattığı bir sonuç olarak birden fazla amacı sağlamaya yönelik sistemlere doğru ilerlemeye başlamıştır.

Bir karar mekanizması olan Yöneylem Araştırmasının gayesi; amaçları belirli koşullar ve kısıtlar altında optimum sonuca ulaşmaktır. Çok amaçlı karar almanın ana fikri; karar alıcı tarafından her bölüm içindeki problemlerin anlaşılabilir olmasının sağlanması ve bu problemlerin çözüme ulaştırılması seklinde açıklanabilir. Bu tanıma göre, çok amaçlı karar alma problemleri, uygulama alanlarına, amaç ve kısıtlarına ve karar alıcı ile ilişki düzeylerine göre sınıflandırılabilir.

Çağın ilerlemesiyle birlikte teknolojinin gelişmesi, insanların gereksinimlerinin çeşitlenmesi ve artmasına neden olmuştur. Böylece verilecek kararlar basit olmaktan çıkıp daha karmaşık bir yapıya sahip olmuştur. İnsanların karşılaştıkları problemler zamanla çok amaçlı bir boyut kazanmıştır. Örneğin; kar amacı güden bir işletmenin karını maksimum yapmak, ya da maliyetini minumuma indirmek gibi genel amaçlarını tatmin etmek her zaman yeterli olmayabilir. Tek amaçlı karar yöntemleri kullanılarak yapılan çalışmalar her zaman en iyi sonucu vermeyebilir. Buradan yola çıkarak çok amaçlı karar verme yöntemleri geliştirilmiştir.

Karar vericilerin çok amaçlı problemleri çözmek için kullandıkları yöntemlerin başında hedef programlama gelir. Hedef programlama doğrusal programlamanın özel bir durumudur. Hedef programlama, birçok amacı aynı anda uzlaşık olarak çözüme kavuşturan bir yöntemdir.

Bu çalışmada, çok amaçlı problemleri çözmek için kullanılan hedef programlama üzerinde durularak, hedef programlamanın yöntemlerinden ağılıklı çok hedefli programlama ve eşit ağırlıklı çok hedefli programlama yöntemleri incelenecektir.

Çalışmanın birinci bölümünde; yöneylem araştırmasının doğuşundan başlanarak tarihsel gelişimi, uygulama alanları, türleri, özellikleri açıklanmaya çalışılarak modelin kurulum aşamaları ayrıntılı olarak açıklanmıştır.

İkinci bölümde, yöneylem araştırmasının temelini oluşturan doğrusal programlamadan ayrıntılı olarak bahsedilecektir. Doğrusal programlamanın çözüm yöntemlerinden olan simpleks çözüm yöntemi ayrıntılı ve adım adım anlatılmıştır. Doğrusal programlama kullanılacak olan hedef programlama yönteminin de temelini oluşturduğundan bu konu üzerinde ayrıntılı olarak durulacaktır.

Çalışmanın üçüncü bölümünde tezin ana konusu olan hedef programlamanın doğuşundan başlanarak kullanılan yöntemler ve daha önce konumuzla ilgili yapılan çalışmalardan bahsedilecektir.

Tezin dördüncü bölümü ise uygulama bölümüdür ve hedef programlama yönteminin türlerinden olan eşit ağırlıklı ve ağırlıklı çok hedefli programlama yöntemin sanal bir işletme olan KAR işletmesi üzerinde LİNGO paket programı kullanılarak uygulaması yapılmıştır.

Tezin son bölümünde ise uygulama sonuçları karşılaştırılmış ve sonuçlar özetlenmiştir.

BİRİNCİ BÖLÜM

1. YÖNEYLEM ARAŞTIRMASI

1.1. Genel Açıklamalar

Yöneylem Araştırması sibernetik, sistem mühendisliği, iletişim bilimleri, çevre bilimleri ve sistem bilimleri gibi İkinci Dünya savaşından sonra gelişen disiplinler arası bir bilimdir (Esin, 1981: 1). Çok sayıda teknik ve bilimsel yaklaşımı içeren Yöneylem Araştırması genellikle kıt kaynakların paylaşımının söz konusu olduğu sistemlerin en iyi şekilde tasarlanması ve işletilmesine yönelik karar problemlerine bilimsel yaklaşımın uygulanmasını amaçlamaktadır. Bilimsel metotlarla problem çözme çalışmalarının başlangıcı çok eskilere dayanmakla birlikte günümüzde yöneylem araştırması olarak adlandırılan bilim dalının temelleri İkinci Dünya Savaşı yıllarında atılmıştır. Savaş yıllarında stratejik ve taktik seviyede çeşitli askeri problemlerin analizi ve muharebelerdeki etkinliğin artırılmasına yönelik olarak çeşitli teknikler geliştirilmiş ve bunlar yoğun olarak uygulanmıştır. Matematiksel modelleme ve bilimsel metotların askeri harekatlara uygulanması ve etkinliğin artırılması amacıyla yapılan optimizasyon çalışmaları sonucunda geliştirilen teknikler Operations Research (Yöneylem araştırması) olarak adlandırılmıştır.

1.2. Yöneylem Araştırmasının Tarihsel Gelişimi

Yöneylem araştırması çalışmalarına ilk olarak II. Dünya Savaşı sırasında başlanmıştır. Savaşın ortalarında İngiltere'yi bombalayan Alman uçaklarının verdiği zarar yüzünden bir önlem almak isteyen İngiliz Genel Kurmayı çareyi bilim adamlarına danışmakta bulmuştur. Çeşitli uzmanlık alanlarından gelen bilim adamları savaş çabalarına katkıda bulunmaları amacı ile bir araya getirildiler. Fizyologlar, genel ve teorik fizikçiler, matematikçiler ve subaylardan oluşan gurubun başkanlığını Nobel ödüllü İngiliz fizikçi P.M.S. Plackett yapmaktaydı.

Grubun temel amaçlarından biri savaşın bir parçası olarak yürütülen askeri harekâta bilimsel yöntemin uygulanması ile radar aygıtları, kontrol birimleri ve havadaki savaş uçaklarından oluşan üçlü sistem arasındaki bilgi alışverişinin sağlanması idi.

Araştırma grubunun başkanı olan Prof. Plackett'de eldeki askeri bilgilere ihtiyaç olacağını ve bu bilgilerin tamamının kendilerine verilmesi durumunda bir çözüm üretebileceklerini bildirmiştir. Bundan sonra araştırma ekibinin Alman hava hücumlarının dağılımını olasılık kurallarına göre hesaplayarak eldeki olanaklarla oluşturduğu hava savunma sistemi beklenenin çok üstünde bir başarıya ulaşmıştır. Uçak sayısı bakımından oldukça üstün durumda bulunan Alman hava kuvvetlerinin hemen her hücumu bu hava savunma sistemi sayesinde İngiliz avcı uçaklarıyla karşılanmış ve böylece Almanlar etkili bir hava baskını yapamamıştır.

Yöneylem araştırmalarının savaş sırasında sağladıkları bu büyük başarı karşısında, savaş sonrasında bu bilimsel yaklaşıma kayıtsız kalınamadı. İngiliz hükümeti büyük bir yıkımla çıktığı savaş sonrasında, ekonomi, sanayi ve yönetimin yeniden gözden geçirilmesine karar vermişti. Büyük bir savaş sonrasında yapılması gereken değişiklikler belirlenmek zorundaydı. Bu çalışmalarda savaş sırasında yetkinliğini ispatlamış olan yöneylem araştırması yönteminden fazlasıyla yararlanıldı. 1950'lerden itibaren Amerika Birleşik Devletlerinde birçok alanda kullanılmaya başlanmıştı (Keys, 1991: 68).

Ülkemizde ilk yöneylem araştırması çalışmaları silahlı kuvvetlerde başlamıştır. İlk yöneylem araştırma ekibi de 1956 yılında daha sonra adı Araştırma ve Geliştirme (ARGE) Başkanlığı olan Genel Kurmay Danışma ve Geliştirme Başkanlığında kurulmuştur. 1 Eylül 1965 tarihinde ordu dışında ilk kez Türkiye Bilimsel ve Teknik Araştırma Kurumunda (TÜBİTAK) altı kişilik bir kadro yöneylem araştırması çalışmalarına başlamıştır (Esin, 1985: 90).

1.3. Yöneylem Araştırmasının Tanımı ve Uygulama Alanları

Yöneylem araştırması, bir sistemde ortaya çıkan problemlere, sistemin denetlenebilir bileşenleri cinsinden bilimsel yöntem, teknik ve araçların uygulanmasıyla en iyi çözümün bulunmasıdır (Churchman, 1957: 18).

Yöneylem araştırması; insan, makine, para ve malzemeden oluşan endüstriyel, ticari, resmi ve askeri sistemlerin yönetiminde karşılaşılan problemlere, bilimin çözüm arayışıdır. Çözüm olarak sistemin risk ölçümünü de içeren ve alternatif karar, strateji ve kontrollerin sonuçların tahmin ve karşılaştırmasını sağlayacak bilimsel bir yöntem geliştirmektir. Amacı, yönetimin politika ve faaliyetlerini bilimsel yöntemlerle belirlemesine yardımcı olmaktır (Doğrusöz, 1973: 9-17).

Yöneylem araştırması örgütün bütünleşik amaçlarına en iyi uyum sağlayacak şekilde, organize (insan - makine) sistemlerin kontrol edilebilir problemlerinin çözümünde disiplinler arası ekiple bilimsel yöntem uygulanmasıdır (Ackoff, 1968: 6).

Bir diğer yöneylem araştırması tanım da, "kısıtlı kaynaklarla en iyi (optimum) çözümün elde edilmesine yarayan bilimsel bir yaklaşım" şeklindedir (Taha, 1987: 1).

Bu tanımlardan yola çıkarak yöneylem araştırmasının;

Konusu; insan-makine sistemlerinin tasarım, kuruluş ve çalışması esnasında karşılaşılan problemleri belirlemek ve bu problemlerde optimum çözümü elde etmektir.

Yaklaşımı; sistemin olduğu gibi tamamını ele alarak, değişik bilim dallarından gelecek olan uzmanlardan (ekonomist, pazarlamacı, mühendis, finans uzmanı, istatistikçi,...) oluşan ekiplerle, bilimsel yöntemler ışığında, gerekli yolu izlemektir. Böylece çözülecek işletme problemine farklı görüş açılarına sahip uzmanların gözüyle problemin çözümüne çalışır.

Amacı; yönetimin politika ve faaliyetlerini bilimsel olarak belirlemesine yardımcı olmak, böylece verilecek kararların tutarlılığını ve uygulanabilirliğini arttırmaktır.

Karar vermek insanın doğasıdır, çünkü karar mevcut pek çok alternatif arasından düşünceye ve analize dayalı bir seçim yapmayı gerektirir. Günümüz koşullarında tüm verileri toplayıp bunlar üzerinden analizler yapıp karar vermek mümkün değildir. Tüm verilere ulaşmak mümkün olmadığından, istatistiki verilere başvurulur ve bunlardan yararlanarak bilimsel yöntemler uygulanır. Diğer alanlarda olduğu gibi yöneylem araştırmasında da istatistiğin katkısı büyüktür.

Optimum koşulların sağlanabilmesi için başarılması gereken amaçlar, uyulması gereken kısıtlar bulunur. Problemlerin yalnız ekonomik veya askeri olması gerekmez. En zıt alanlarda doğan problemler belli çözüm yollarıyla sunulur. Karar verecek olan kişi söz konusu kısıtlar altında, kendi denetimindeki sistemin etkinliğini arttırmayı amaçlar.

Yöneylem araştırması her geçen gün artan bir ivmeyle gelişim göstermektedir. Bu gelişime paralel olarak birçok alanda kullanılmaktadır. Kullanım alanlarını genel başlıklar altında toplamak gerekirse şöyle sıralanabilir;

- Ulusal planlama
- Enerji planlaması ve yönetimi
- Teknoloji planlaması
- Yatırım projeleri yönetimi
- İnsan gücü planlaması
- Sanayide bakım-onarım
- Üretim planlaması
- Stok kontrolü ve malzeme planlaması
- Yönetim bilişim sistemleri
- Stokastik sistemler
- Sağlık planlaması
- Savunma
- Ulaşım planlaması
- Haberleşme sistemleri
- Çevre sağlığı
- Bankacılıkta yöneylem araştırması
- Finansal yönetim ve yatırım planlaması
- Yöneylem araştırmacılarının eğitimi ve yetiştirilmesi

Bu kadar çok alanda problemlere çözüm üretilmeye çalışılan bir konuda alt uzmanlaşma dallarının oluşması kaçınılmazdır. Problemin çözümünde kullanılan modelin yapısına göre isimler verilmiştir.

1.4. Yöneylem Araştırmasının Özellikleri

Yöneylem araştırması bir süreç olup, dikkatlice ilgili verilerin toplanarak problemin formüle edilmesi ve gözlenmesi ile başlar. Sonra gerçek problemin esaslarını özetleyen bilimsel bir model kurulur. Modelin kurulum aşamalarına geçmeden önce Yöneylem Araştırmasının sahip olduğu özellikler aşağıda açıklanmaya çalışılacaktır.

1.4.1. Sistem Yaklaşımı Özelliği

Çözümü aranan sorunlarla ilgili olan ve çözüm sonuçlarını ihmal edilemeyecek biçimde etkileyecek olan, problemin ilişkin olduğu örgütün içindeki veya dışındaki tüm etkenlerin göz önüne alınması sistem yaklaşımının gereğidir (Esin, 1981: 32).

Ele alınan sistemlerin çeşitli bölümlerinin amaçları birbiri ile çelişkili durumda olabilir. Bu nedenle yöneylem araştırması bir sistemle ilgili probleme çözüm ararken, sistemin tümüne en uygun çözümü bulmaya çalışır. Dolayısıyla, yöneylem araştırmasının sistemin belirli bir alt bölümü için bulduğu en uygun çözüm, tüm sistem için en uygun çözüm olmayabilir.

Bu yöntem ilk olarak askeri alanda kullanılmış olmasına rağmen, asıl gelişimi endüstriyel kurumlarda kullanılması ile olmuştur. Endüstri devriminden önce, çok küçük olan işletmeler, genellikle bir kişinin çalışmaları ile yürütülmekteydi. Bu insan genelde işletmenin patronu sıfatını taşırdı ve planlama, satın alma, ürün geliştirme, pazarlama, eleman seçme gibi işletme ile ilgili tüm kararları tek başına almak zorunda kalırdı. Ancak endüstri devriminden sonra büyüyen ve kontrol edilmesi zorlaşan şirketler bu işleri çeşitli bölümlere ayırmış ve daha küçük parçalara bölmüştür.

Yöneylem araştırması problemi çözerken, o problemin ait olduğu organizasyonun bütün unsurlarını, çevresini ve aralarındaki etkileşimi göz önünde bulundurur. Çünkü organizasyonun herhangi bir unsurundaki bir değişiklik, diğer unsurları ayrı ayrı etkiler. Aynı şekilde organizasyonun çevresindeki bir değişiklik organizasyonun faaliyetlerini ve organizasyonun kendi işleyişinde meydana gelecek değişiklikler de çevresini etkileyecektir. Bu yüzden problemi, içinde bulunduğu sistemin bütün unsurlarıyla ve bu unsurlar arasındaki her türlü etkileşimle birlikte incelemek gerekir.

1.4.2. Disiplinler Arası Yaklaşım Özelliği

Yöneylem araştırması, disiplinler arası bir yaklaşımdır. Fizik, kimya, matematik, istatistik gibi farklı disiplinlerden yetişmiş kişilerin her soruna bakış açısı farklıdır. Bu yüzden problemin modellenmesinde ve çözümünde farklı bakış açılarından faydalanabilmek için problemlerin disiplinler arası bir ekip tarafından incelenmesi gerekir.

Herhangi bir sorunu yöneylem araştırması yöntemiyle çözümleyebilmek için bir araştırma ekibinin oluşturulması gerekmektedir. Yöneylem araştırmasının temel özelliklerinden birisi de disiplinler arası ekip çalışması biçiminde olmasıdır. Çünkü problemi her yönüyle görebilmek, dolayısıyla doğru bir çözüme ulaşabilmek için, yöneylem araştırmalarında çeşitli bilim dallarından gelen uzman araştırmacılardan yararlanılır. Bu nedenle, yöneylem araştırması projelerini yürütecek araştırmacı ekiplerin değişik branşlardaki kişilerden oluşması arzulanır (Esin, 1981: 33).

Disiplinler arası yaklaşımın sağladığı iki önemli avantaj vardır. Birincisi bir problemin çözümü için var olduğu düşünülen çözümlerden daha fazlasını ortaya çıkarmasıdır. Bir kişinin tüm çözümleri görebilmesi olanaksız olmasa bile, bir takım çalışmasına göre daha düşük olasılıklıdır. Yani "akıl akıldan üstündür". Diğer avantajı ise, sistem içindeki farklı bilimlerin bir arada olduğu gerçeğinde ortaya çıkmaktadır. Aynı sistem içinde fizik, sosyoloji, psikoloji, ekonomi, mühendislik alanlarının buluşması, sorun çözmede bu alanların temsilcilerinin de katkıda bulunmasını zorunlu kılmıştır (Churcman, 1957: 114).

1.4.3. Bilimsel Yöntemlerle Yaklaşım Özelliği

Yöneylem araştırmasının probleme yaklaşım bakımından en önemli katkısı, sistemin öğelerini ve aralarındaki ilişkileri temsil eden modeller kurabilmesi ve modeldeki parametrelerin veya karar değişkenlerinin bir diğerine olan etkisini kolayca etüt edebilmesidir.

Yöneylem araştırmasının kullandığı bilimsel yöntem beş temel aşamadan oluşur. Bunlar aşağıda açıklanmıştır (Aygüneş ve diğerleri, 2001: 39):

1. Problemin tanımlanması (formüle edilmesi),
2. Modelin kurulması,
3. Modelden çözüm elde edilmesi,
4. Modelin ve çözümün test edilmesi (kanıtlanması),
5. Çözümün uygulanması.

Aşama 1: Problemin Tanımlanması (Formüle Edilmesi):

"Yanlış" ya da eksik problemlerden "doğru" çözüm elde edilemez. Bu ifadeden anlaşılacağı gibi ilgili sistemin detaylı bir şekilde incelenip söz konusu problemin iyi bir şekilde tanımlanması, işin birinci ve en önemli aşamasıdır. Bu aşama, eldeki problemin kantitatif olarak incelenebilecek bir yapıya dönüştürülmesini amaçlar. Bu aşamada problemin çözümüne direk ya da dolaylı olarak etki edebilecek her unsurun özenle ortaya çıkarılması gerekir. Problemin tanımlanması, yöneylem araştırması ekibinin tamamının katılımını gerektiren bir süreç olup yapılacak incelemenin sonunda aşağıdaki hususların belirlenmesi gerekmektedir (Yılmaz, 2005: 12):

1. Amaçların belirlenmesi.
2. Problem alanının, yani organizasyonu ve çevresini kapsayacak şekilde probleme etki edecek olan sistemin belirlenmesi.
3. Problemin çözümüne etki edecek sınırlamaların (kısıtların) belirlenmesi.
4. Varsayımların belirlenmesi.

5. Uygun bir etkinlik ölçüsünün belirlenmesi: Etkinlik ölçüsü çeşitli alternatiflerin amacı ne denli gerçekleştirdiğini değerlendirmede kullanılan bir ölçütü ifade eder.

Aşama 2. Modelin Kurulması

Problem belirlendikten sonra yapılacak iş, problemi en iyi bir biçimde temsil edebilecek bir modelin kurulmasıdır. Model, gerçek yaşamın bir takım varsayımlarla basitleştirilmiş bir biçimidir. Diğer bir ifadeyle, gerçeğin gösterimi veya gerçek durumun bir özeti olarak tanımlanabilir.

Model, bir sistem veya alt sistemin davranış gösterimidir. Araştırmaların çoğunda sistemin özelliklerini taşıyan bir model geliştirilerek model üzerinde, değişmelere karşı sistemin davranışı izlenir. Sistemin davranış gösterimi olan modeller yapılarına, kullanım amaçlarına, zamanla olan ilişkilerine, çözümleme şekillerine, fonksiyonel ilişkilere ve benzeri bakış açılarına göre sınıflandırılmaktadır (Kara, 1979: 84).

Bir sistemin mantıksal akış modelinin simgesel (sembollerle) gösterimine matematiksel model denir. Matematiksel model, çözümleri sistemin durumunu açıklayan bir denklemler kümesi olarak da tanımlanmaktadır. Sistemin mantıksal akış modelinde belirlenen tüm öğe ve ilişkilerin matematiksel yazılımında eşitliklerle birlikte eşitsizliklerin de olması doğaldır (Kara, 1979: 85). Modelin kurulması aşamasında yapılan işleri şunlardır:

- Karar değişkenlerinin belirlenmesi: Karar değişkenleri problemdeki kontrol edilebilir unsurları temsil eden ve çözüm sonunda değerleri elde edilecek olan değişkenlerdir. Örnek olarak toplam n adet ürünün üretileceği bir üretim probleminde üretilecek ürün miktarlarını gösteren n adet karar değişkeni (örneğin x_1, x_2,…, x_n) olarak gösterilir.

- Parametrelerin belirlenmesi: Parametreler ise kontrol edilemeyen ya da çevresel faktörler olarak bilinen unsurları ifade eden sabit değerli katsayılardır. Örneğin bir birim ürünün satışından elde edilecek kar, bir birim ürünün üretimi için gerekli olan hammadde miktarı ve eldeki toplam hammadde kapasitesi gibi unsurlar modelin parametrelerini oluştururlar.

- Amaç fonksiyonun oluşturulması: Ulaşılmak istenen amacı tanımlayan ve karar değişkenlerinin fonksiyonu olarak ifade edilen matematiksel bir fonksiyondur. Yukarıdaki üretim problemi için her bir ürünün bir birimden elde edilecek kar (3,7,…,15)TL/birim olsun. Bu parametreleri amaç fonksiyonu katsayıları olarak kullanmak suretiyle toplam karı ifade eden amaç fonksiyonu (örneğin, $p=3x_1+7x_2+\ldots+15x_n$) olarak yazılır.

- Kısıtların oluşturulması: Karar değişkenlerinin alabilecekleri değerler ile ilgili sınırlamaları belirten kısıtlar da matematiksel olarak ifade edilebilir. Diyelim ki örnek problemde her bir üründen bir birim üretmek için gerekli olan demir miktarı (3,4,…,2) kg/birim ve eldeki toplam demir miktarı ise 20 kg olsun. Bu parametreleri katsayı olarak kullanmak suretiyle demir kapasitesi ile ilgili kısıt (örneğin, $3x_1+4x_2+\ldots+2x_n\leq 20$) olarak ifade edilir (Yılmaz, 2005: 14-15).

Yöneylem araştırmasında kullanılan teknikler Tablo 1'de gösterilmiştir.

Tablo 1: Yöneylem Araştırması Modellerinin Sınıflandırılması

DETERMİNİSTİK MODELLER	OLASILIKLI MODELLER
Doğrusal Programlama	Markov Zincirleri
Tamsayılı programlama	Kuyruk Teorisi
Hedef programlama	Karar Analizi
Ulaştırma ve Atama Modelleri	Simülasyon
Doğrusal Olmayan Programlama	Tahmin Modelleri
Oyun Teorisi	Güvenirlilik Analizi
Deterministik Dinamik Programlama	Olasılıklı Dinamik Programlama
Deterministik Stok Modelleri	Olasılıklı Stok Modelleri
Şebeke (Ağ) Analizi	
CPM – PERT Şebeke Modelleri	

Aşama 3. Modelin Çözümü

Bu aşamada problem içinde kullanılacak değişkenlerin belirlenerek modelin çözümün elde edilmesidir. Çözüm aşamasında değişik yöntemler kullanılmakla birlikte genel olarak iki ayrı çözüm elde edilebilir.

Optimal çözüm, amaç fonksiyonu değerinin maksimum ya da minimum yapılması anlamındadır. Matematiksel modellerin çözülmesinde kullanılan teknik ve yöntemleri analitik teknikler, sayısal teknikler, sezgisel yaklaşımlar olarak değerlendirmek mümkündür. Sezgisel yaklaşımlar, optimizasyon tekniklerinden herhangi birisiyle çözülemeyecek kadar karmaşık yapıdaki modellerde, optimal çözüm yerine yaklaşık bir çözüm elde etmek için geliştirilmiştir. Karmaşık sistemler için kullanılan alternatif bir modelleme yaklaşımı da simülasyondur. Matematiksel modellemedeki gelişmelere rağmen pek çok gerçek olay matematiksel olarak modellenememektedir. Simülasyon teknikleri matematiksel olarak modellenmesi ve analitik tekniklerle çözülmesi mümkün olmayan sistemlerin modellenmesinde ve incelenmesinde kullanılırlar. Simülasyon genel olarak gerçek sistemi küçük parçalara ayırıp bu parçaları, uygun mantıksal bağlantılarla, birbiri ile ilişkilendirmek suretiyle sistemin davranışını taklit etmeye çalışan bir yaklaşım olarak tanımlanabilir (Yılmaz, 2005: 16).

Ayrıca günümüzde yöneylem araştırması çözüm tekniklerini kapsayan ve matematiksel modellerin bilgisayar ile çözümüne olanak sağlayan çok sayıda paket program geliştirilmiştir. Dolayısıyla modelden çözüm elde edilmesi için işin kolay bir aşaması olup, nispeten zor olan bölüm optimal çözüm elde edildikten sonra yapılan analizlerdir. Duyarlılık analizi adı verilen bu süreç, model parametrelerindeki olası değişiklikler (örneğin birim karın ya da eldeki kapasitenin değişmesi) sonucunda optimal çözümün nasıl bir davranış göstereceğinin incelenmesini kapsar.

Aşama 4. Modelin Ve Çözümün Test Edilmesi

Modelden elde edilen çözümü uygulamaya koymadan önce gerçeğe uygunluğunun kanıtlanması gerekir. Eğer çözüm sistemin geçmiş dönem sonuçlarını aynen veya daha olumlu bir şekilde sağlıyorsa, modelin geçerli olduğu kabul edilir. Eğer sistemin geçmiş dönem sonuçları yoksa simülasyondan yararlanılır Model geçerliliğinin kanıtlanmasında bir başka yol olarak da sistemdeki deneyimli kişilerin görüşlerine başvurulabilir.

Aşama 5. Çözümün Uygulanması

Bu aşama, geçerliliği kanıtlanmış bir modelden elde edilen güvenilir bir çözümün gerçek hayattaki probleme uygulanması aşamasıdır. Bu aşamada da asıl yük, yani çözümün anlaşılabilir bir şekilde sistemi işletecek olan personele anlatılması, yine yöneylem araştırması ekibine düşmektedir (Yılmaz, 2005: 17).

Çalışmanın bu kısmında yöneylem araştırmasının ne olduğu açıklanmaya çalışıldı. Yöneylem araştırmasının kullandığı teknikleri ve yaklaşımları model yapılarına göre yukarıda sıralanmıştı. Bundan sonraki aşamada bu modellerden biri olan doğrusal programlamayı ve doğrusal programlamanın çözümünde kullanılan yöntemlerden biri olan simpleks çözüm yöntemi ayrıntılı olarak açıklanmaya çalışılacaktır.

İKİNCİ BÖLÜM

2. DOĞRUSAL PROGRAMLAMA

2.1. Genel Açıklamalar

Günümüzde işletme, ekonomi gibi birçok alanı yakından ilgilendiren doğrusal programlama, yöneylem araştırmasının en temel konularından biridir. Özellikle son yıllarda doğrusal programlamanın iş yaşamına etkisi çok daha fazla olmuştur. Artan rekabet koşullarında her işletme üretimini devam ettirebilmek için en az girdi ile en fazla çıktıyı elde etmek isteyecektir. Bu noktada doğrusal programlama işletmeler için çok fazla önem taşır.

Doğrusal programlama belirli bir amacı gerçekleştirmek için sınırlı kaynakların en etkin kullanımını ve çeşitli alternatifler arasında en optimum dağılımını sağlayan bir matematiksel programlama tekniğidir. Buradaki "doğrusal" terimi modeldeki tüm fonksiyonların doğrusal olduğunu anlatırken; "programlama" terimi ise bir hareket tarzının veya planının seçilmesi anlamına gelmektedir. Doğrusal programlama modellerin karar problemlerine yaklaşım bakımından en önemli katkısı; sistem yaklaşımını benimsemiş olması, faaliyetlerin diğer unsurlarla ilişkilerini de bir bütünlük içerisinde ele alabilmesidir. İşletme karar problemlerinin sayısal verilerle en kolay anlatımı doğrusal programlama modelleriyle olmaktadır (Büyükkeklik, 2007: 30).

Doğrusal programlamanın temel konusu, sınırlı kaynaklar arasında optimum dağılımın sağlanması ile ilgilidir. Bu bağlamda doğrusal programlama, optimizasyon problemlerinin çözümünde kullanılan matematiksel bir programlama tekniğidir.

Maksimum ya da minimum değerlerin belirlenmesi problemiyle genelde matematiğin çeşitli dalları ilgilenmekle birlikte, bu tür problemlere ekonomide de rastlamak mümkündür. Örneğin işletmelerin karlarını maksimize etmek ya da etmek gibi amaçlarına ulaşabilmek için ellerinde bulunan işgücü, hammadde, sermaye, enerji gibi sınırlı kaynaklardan en iyi şekilde yararlanmak isteyecektir. İşte doğrusal programlama problemleri bu türden amaçlara ulaşabilmek için sınırlı kaynakların en iyi şekilde kullanılması veya dağıtımıyla ilgilenir.

2.2. Doğrusal Programlamanın Tanımı, Tarihsel Gelişimi Ve Uygulama Alanları

Doğrusal programlama, belirli doğrusal eşitlik ya da eşitsizliklerin kısıtlayıcı koşulları altında doğrusal bir amaç fonksiyonunun optimum sonucunu bulmak olarak tanımlanabileceği gibi (Alan ve Yeşilyurt, 2004: 152); belirli bir amacı gerçekleştirmek için sınırlı kaynakların en etkin kullanımını ve çeşitli alternatifler arasında en optimum dağılımını sağlayan bir matematiksel programlama tekniği şeklinde de tanımlanabilir. Buradaki "doğrusal" terimi modeldeki tüm unsurların (elemanların) doğrusal olduğunu anlatırken; "programlama" terimi ise bir hareket tarzının veya planının seçilmesi anlamına gelmektedir. Böylece, bir doğrusal programlama modeli, kıt kaynakların rakip faaliyetler arasındaki dağıtımını en uygun bir şekilde gerçekleştiren dağıtım planını bulmada kullanılan doğrusal bir modeldir (Özgüven, 2003: 3).

Doğrusal programlama, kaynakların kullanım şekillerinin ve alternatif karar alma yollarının değerlendirilmesini yaparak, belirli bir amacın gerçekleştirilmesinde bunların en uygunlarını seçmek için geliştirilmiş bir matematiksel optimizasyon tekniğidir (Tanrıkurt, 1979: 67).

Doğrusal programlama, belirli bir amacın gerçekleşmesini etkileyen bazı kısıtlayıcı koşulların ve bu kısıtlayıcı koşulların doğrusal eşitlik ya da eşitsizlik biçiminde verilmesi durumunda amaca en uygun çözümün bulunmasını sağlayan bir matematiksel yönetimdir. Amaç fonksiyonunu en büyük veya en küçük yapacak en iyi çözüme adım adım yaklaşan bir algoritma (hesaplama yöntemi) 'dır.

Doğrusal programlama kavramı ilk olarak Sovyet matematikçisi A.N. Kolmogorol tarafından II. Dünya savaşı yıllarında gerçekleştirilmiştir. II. Dünya Savaşı sırasında büyük askeri ve sivil harekât bu faaliyetlerin sistematik bir şekilde planlanıp, koordine edilmesini gerektiriyordu. Bu sebeple doğrusal programlama konusu araştırılmaya başlandı. Doğrusal programlama konusundaki ilk uygulama 1945 yılında Stigler tarafından gerçekleştirilen ve “Diyet Problemi” olarak bilinen problemdir. 1947 yılında Amerika Birleşik Devletleri’nin askeri faaliyetlerini planlamak amacıyla George B. Dantzing tarafından geliştirilen “Simpleks Metodu” doğrusal programlama konusundaki en büyük gelişmedir. Simpleks çözüm metodunun bulunmasından sonra doğrusal programlama teorisinde ve pratik uygulamasında hızlı bir gelişme olmuştur. Doğrusal programlama başlangıçta askeri gereksinmeler için geliştirilmişse de bugün endüstride geniş bir uygulama alanı bulmuştur.

Matematiksel bir teknik olarak "doğrusal programlama", değişik nitelikteki ekonomik faaliyetlerin karşılaştığı ya da meydana getirdiği birbirinden bağımsız faktörlerin, sınırlı kaynak imkânlarına bağlı olarak en uygun bileşimde biçimlenmesine önemli katkılarda bulunmaktadır. Temel olarak doğrusal programlama, bütün ayrıntıları bilinen şartlar altında uygun bir karar alma aracıdır. Bununla beraber, rekabetçi şartlara bağlı karar alma durumlarında rekabetçi şartlara bağlı karar alma durumlarında, örneğin "oyun teorisinde", doğrusal programlama, gerek stratejiler vektörünün tespit edilmesinde, gerekse oyunun kesin sonucunun tespitinde tam çözüm getiren niteliği ile bu teorinin gelişmesine büyük katkılarda bulunmuştur (Sevüktekin, 2007: 107).

Doğrusal programlama modellerin karar problemlerine yaklaşım bakımından en önemli katkısı, sistemin öğelerini ve aralarındaki ilişkileri temsil eden modeller kurabilmesi ve modeldeki parametre veya karar değişkenlerinin bir diğerine olan etkisinin kolaylıkla etüt edilebilmesidir (Esin, 1988: 4). Problemlerin sayısal verilerle en kolay anlatımı doğrusal programlama modelleriyle olmaktadır.

Bir doğrusal programlama modeli, temelde, amaç fonksiyonu ve sınırlar olmak üzere iki bileşenden oluşur. Amaç fonksiyonu ile minimize edilmek istenen genelde bir işin maliyeti, üretim işlemlerin toplam zamanı, bir tezgâhın boş geçen süresi gibi faktörlerken; maksimize edilmek istenen faaliyetlerdeki toplam kar katkısı, taşınan yük miktarı, üretilen ürün miktarı, satış miktarı gibi faktörlerdir. Sınır şartları ise genelde: zaman, işgücü miktarı, talep miktarı, tezgâh sayısı, kalıp sayısı, tezgâh kapasitesi, depo alanı, talep miktarı, arz miktarı, hammadde ve malzeme miktarı, taşıyıcı araç kapasitesi, işletme sermayesi gibi kıt kaynaklardan oluşur (Büyükkeklik, 2007: 32).

Doğrusal programlama yönteminin matematik için çok önemli olan optimizasyon alanında çok büyük bir rol oynamasının da çeşitli nedenleri vardır. "Yöneylem araştırması" alanında birçok pratik problem doğrusal programlama problemi olarak tanımlanmaktadır. Matematik optimizasyon kavram ve yöntemlerinin geliştirilmesi için yapılan çalışmaların geçmişine bakılırsa, bunların en önemli olanlarının çoğunun ilk defa doğrusal programlama için ortaya atılıp geliştirildiği açık olarak görülmektedir.

Aynı şekilde pratik alanlar olan işletmecilik ve mikro iktisat alanlarında etkin mal ve hizmet üretimi ve arzı için gelirleri maksimum hale getirmek veya maliyet ve masrafları minimum hale getirmek için, doğrusal programlama çok büyük katkılarda bulunmaktadır. Doğrusal programlamanın uygulama alanları arasında birbirinden çok farklı alanlar örnek olarak gösterilebilir.

- Portföy yönetimi problemleri,
- Taşımacılık problemleri,
- Tarımsal planlama problemleri,
- Üretim stok kontrol problemleri,
- Mamul karışım problemleri,
- Makine-İşgücü atama problemleri,
- İşgücü-programlama problemleri,
- Pazarlama problemleri,
- Doğal kaynakların ülke ihtiyaçlarına uygun şekilde tahsisi problemleri,
- Radyoterapi tedavi problemi,
- Diyet problemleri,
- Yatırım ve Üretim planlanması,
- İşletmelerin Kuruluş Yerlerinin Saptanması,
- İşletmelerde görevlerin planlanması

Askeri uygulamalara örnek olarak doğrusal programlama, yük taşıma, üslerin yerleşim merkezlerinin belirlenmesi, yakıt tüketimini minimize edecek sistemlerin geliştirilmesi ve minimum savunma gibi örnekleri sayılabilir (Timor, 2001: 3).

2.3. Doğrusal Programlamanın Varsayımları

Doğrusal programlama iş hayatında doğru kullanıldığında karar vermeye yardımcı olan çok yararlı bir araç olabilir. Doğrusal programlamayı kullananlar bir problemin modelini kurarken bazen belli başlı varsayımları yok sayabilirler. Doğrusal programlamanın varsayımları dikkate alınmazsa, elde edilen sonuç optimal sonucu vermeyebilir. Doğrusal programlamada kullanılmak üzere bir model kurulurken aşağıdaki varsayımları içermesi gerekir.

2.3.1. Doğrusallık Varsayımı

Modelin değişkenleri arasındaki ilişkiler doğrusal olmalıdır. Bu varsayım modelin amaç fonksiyonu ve kısıtlayıcı fonksiyonları ile ilgilidir. Yani, modeldeki tüm eşitlik ya da eşitsizliklerdeki değişkenler birinci dereceden olmalı; aralarındaki ilişkiler birinci dereceden fonksiyonlarla anlatılabilmelidir (Taha, 2000: 13). Şöyle ki, her bir faaliyetin amaç fonksiyonunun değerine katkısı, x_j faaliyetinin düzeyine oranlıdır. Bu, amaç fonksiyonunda c_jx_j terimi ile gösterilebilir.

Bu özellik her bir karar değişkeninin amaç fonksiyonu ve tüm kısıtlara etkisinin, söz konusu değişkenin değeriyle doğru orantılı olması gerektiğini ifade etmektedir. Örneğin, bir üretim işletmesinin hammadde satın alımlarında belli bir miktardan sonra tedarikçisinden iskonto alması, satın alma maliyetleri ile miktar arasındaki doğru orantı ilişkisini bozacaktır. Bu tip durumlarda, bazı uyarlamalar yapmaksızın doğrusal programlama modelleri ile modelleme yapılamamaktadır. Benzer şekilde, her bir faaliyetin sol tarafındaki her fonksiyonel kısıtlayıcıya katkısı, x_j faaliyetinin düzeyine oranlıdır. Bir anlamda işletmenin girdileri ile çıktıları arasında doğrusal bir ilişkinin olduğunu gösterir. Örneğin, iki farklı ürün üretiminde de belirli miktarlarda kullanılan makine saatin toplamı, her bir ürünün tek tek kullandığı miktarın toplamı kadardır.

2.3.2. Bölünebilirlik Varsayımı

Modelin değişkenleri rakamla ifade edilebilmeli ve bölünebilir nitelikte olmalıdır. Buradaki bölünebilirlikten kasıt, karar değişkenlerinin tamsayılı değerler yanında kesirli değerleri de alabilmesi, yani kıt kaynakların kesirli miktarlarda kullanılabilmesidir. Bölünebilirlik varsayımı, her bir karar değişkeninin kesirli değerler almasına izin verilmesini ister. Böylece bu değişkenler, sadece tam sayılı değerler alması için sınırlandırılamaz.

2.3.3. Toplanabilirlik Varsayımı

Doğrusal programlama da her fonksiyon ilişkin olduğu faaliyetlerin bireysel katkılarının toplamıdır. Kıt kaynakların kullanılması çerçevesinde, toplanabilirlik varsayımı ile rakip faaliyetler tarafından birlikte kullanılan toplam kaynak miktarının, bu rakip faaliyetlerin teker teker kullandıkları miktarların toplamına eşit olması kastedilirken; amaç fonksiyonu yönünden de bağımlı değişkenlerinin değerinin tek tek faaliyetlerden kaynaklanan kar katkılarının toplamına eşit olmasıdır (Özgüven, 2003: 8). Ayrıca, toplanabilirlik varsayımı doğrusallık varsayımının da doğal bir sonucudur.

2.3.4. Belirlilik (Kesinlik) Varsayımı

Kesinlik varsayımı, doğrusal programlama modelindeki her bir parametrenin kesin olarak bilinmesidir. Yani, onların bilinen bir sabit olacağı varsayılır ki, bu da modelin deterministik model olduğunu gösterir. Modeldeki rakip faaliyetlerin amaç fonksiyonuna katkılarının (amaç fonksiyonu katsayıları (c_j)), kullandıkları kaynak miktarlarının (teknolojik katsayıların (a_{ij})) ve kıt kaynakların mevcut miktarlarının (sağ taraf sabitlerinin (b_i)) kesinlikle bilindiği varsayılır. Zaten, doğrusal programlama modelleri deterministik yapılı modellerdir. Belirlilik varsayımı, doğrusal programlama modellerinin kullanımını en çok sınırlandıran varsayımıdır.

Bu varsayım iş hayatında nadiren karşılanır. Çünkü doğrusal programlama modelleri, faaliyetlerin gelecekteki durumunu belirlemek için kurulur. Gelecekteki koşulların tahminine dayanarak kullanılan parametre değerlerinde bazı belirsizliklerin olması kaçınılmazdır.

2.3.5. Negatif Olmama Varsayımı

Doğrusal programlamada yer alan temel, aylak ve artık değişkenlerin değeri sıfır ya da sıfırdan büyük olmalıdır. Doğrusal programlama modelleri gerçek problemlere uygulanır, bu nedenle değişkenler negatif değerli olamazlar. Böylece;

$x_1, x_2, x_3, \ldots, x_n \geq 0$ yazılabilir.

Bir karar probleminin çözümünde doğrusal programlama modelinin kullanılabilmesi için yukarıda sayılan varsayımları sağlaması gereklidir. Bu varsayımlardan bazılarının sağlanamadığı durumlar için de, temeli doğrusal programlamaya dayanan ancak modelleme yapısının ve çözüm algoritmalarının farklı olduğu, bulanık doğrusal programlama, olasılıklı modelleme gibi alternatif yöntemler de kullanılabilmektedir.

Örneğin, "belirlilik" varsayımını gerçek yaşam problemlerinde sağlayabilmek zor olduğundan, kıt kaynakların kullanım miktarının veya kar katkılarının tam net sayılarla değil de, belirli bir aralık ile ifade edilebildiği durumlarda bulanık doğrusal programlama yöntemi tercih edilirken; "bölünebilirlik" varsayımının geçerli olmadığı, karar değişkenlerinin tamsayılı değerler alması gereken durumlar için tamsayılı doğrusal programlama yöntemi tercih edilmektedir.

2.4. Doğrusal Programlama Modelinin Yapısı

Doğrusal programlama modelleri yapı olarak üç temel kısımdan oluşmaktadır. Birinci kısım kimi maliyetlerin en küçüklenmesi ya da kimi kazançların en büyüklenmesi gibi, modelin birincil amacını oluşturan amaç fonksiyonu iken; ikinci kısım gerçekleştirilmeye çalışılan en iyilik (optimizasyon) ölçütü üzerindeki sınırlamaları tanımlayan eşitlik ve/veya eşitsizlik kümesinin oluşturduğu kısıtlayıcılardır (Çınar, 1990: 75). Üçüncü kısım ise, karar değişkenlerinin negatif olmama şartının verildiği kısımdır.

Amaç fonksiyonu, yönetimin ulaşmak istediği hedefin matematiksel ifadesidir. Bu ifadede, karar vericinin kontrolü altındaki parametrelerin, yani, karar değişkenlerinin amaç üzerindeki etkilerinin analitik olarak gösterimi sağlanır. Amaç fonksiyonu negatif olmayan karar değişkenlerinden oluşur. Çünkü karar değişkenleri, bir malın üretilmesi, bir aracın yaptığı sefer sayısı, bir işin yapılmasında kullanılacak işgücü, bir reklam aracının kullanılması gibi unsurları temsil etmek için modelde yer alır ve bu unsurların da negatif olması düşünülemez.

Problemi şekillendiren koşullar da sınırlar setini oluşturur. Amaç fonksiyonunda olduğu gibi sınırlar seti de matematiksel denklemler şeklindedir. Bu denklemlerde eşitliklerin bulunması tanımlanan kaynakların tümüyle kullanılacağını gösterirken; eşitsizliklerin bulunması kaynak kullanımlarının koşullu olduğunu gösterir (Çınar, 1990: 75). Eğer tüm sınırlar eşitlik biçiminde ifade edildiyse en iyi çözüm tek bir noktada oluşur (denklemlerin kesim noktası), tam tersi olarak tüm sınırlar eşitsizlik şeklindeyse sonsuz sayıda olası çözüm vardır, ancak, amaç fonksiyonunun tek bir en iyi (optimum) değeri vardır.

Doğrusal programlama, sınırlı kaynakların alternatifler arasında dağıtılması problemi ile ilgilendiğinden negatif faaliyetler veya negatif kaynak kullanımından söz etmek mümkün değildir. Doğrusal programlama modellerinin son kısmı da karar değişkenlerin negatif olmamalarını sağlayan sınırdan oluşmaktadır.

Amaç fonksiyonu ve sınırlar setinin aşağıda ifade edilen durumlarına göre, doğrusal programlama modelleri amaçları bakımından şu şekilde sınıflandırılabilir:

Klasik Maksimizasyon Modelleri: Amaç fonksiyonunun maksimize edildiği ve tüm sınır denklemlerinin $\leq$ şeklinde olduğu modellerdir.

Klasik Olmayan Maksimizasyon Modelleri: Amaç fonksiyonunun maksimize edildiği ve sınır denklemleri içerisinde hem $\leq$ şeklinde eşitsizliklerin, hem de eşitliklerin (=) bulunduğu modellerdir.

Klasik Minimizasyon Modelleri: Amaç fonksiyonunun minimize edildiği ve tüm sınır denklemlerinin $\geq$ şeklinde olduğu modellerdir.

Klasik Olmayan Minimizasyon Modelleri: Amaç fonksiyonunun minimize edildiği ve sınır denklemleri içerisinde hem $\geq$ şeklinde eşitsizliklerin, hem de eşitliklerin (=) bulunduğu modellerdir.

2.5. Doğrusal Programlama Modelinin Kurulum Aşamaları

Sayısal karar verme yöntemlerinin hepsinde olduğu gibi doğrusal programlama modellerin kurulması, problemin tanımlanması ile başlar. Karar vericinin ulaşmak istediği bir amacının olması, bu amaca ulaşmada izlenebilecek alternatif stratejilerin bulunması, bu stratejilerden hangisinin amacı en iyi şekilde gerçekleştireceği hususunda kuşkuların bulunması doğrusal programlama yönteminin kullanılabileceği bir probleminin varlığını gösterir.

Bu problem bulunduğu sistem içerisinde gözlemlenerek, probleme etki eden parametreler belirlenir. Bu parametreler kullanılarak problemin amaç fonksiyonu ve sınırlar setinden oluşan doğrusal programlama modeli kurulmuş olur. Bu modelin çözülmesiyle, tüm sınır şartlarını sağlayan ve amaç fonksiyonunun değerini de optimum yapan karar değişkenlerinin değerleri elde edilir.

Aşama 1. Problemin Belirlenmesi:

Bu aşamada çalışmaya konu olan, karar vericiler, karar vericinin amaçları, karar değişkenleri, parametreler, sınır şartları gibi temel faktörler belirlenir. Çözülmek istenen sorun ortaya konur. Örneğin işletmenin özellikleri, üretilecek alternatif ürünler, üretimde kullanılan girdiler ve miktarları, kullanılan girdilerin kapasiteleri, üretilecek ürünlerden elde edilecek gelirler, vb.

Karar verici, problemi içeren sistemdeki faaliyetleri planlayan, yönelten, denetleyen ve sapmaları düzeltici önlemlerin alınmasını sağlayan kişi veya farklı disiplinlerden temsilcilerden oluşan gruplardır.

Aşama 2. Karar Değişkenlerinin Belirlenmesi:

Bir problemin doğrusal programlama modelinin kurulmasına önce karar değişkenlerinin karar vericinin kontrolü altındaki parametrelerin, yani, karar değişkenlerinin amaç üzerindeki etkilerinin analitik olarak gösterimi sağlanır. Modele girecek olan değişkenler problemi açıklayan kantitatif büyüklüklerdir. Bu değişkenlerin optimum değerleri modelin çözümü ile bulunur. Parametreler ve sabiteler ise, bu değişkenlerin katsayılarını oluştururlar.

Değişkenleri belirlerken üretimde yapılacak herhangi bir değişikliğin modele yeni değişkenleri getireceği ve değişkenler için kabul edilen ölçülerin aynı olmasına dikkat edilmelidir.

Karar değişkenleri, problemi etkileyen kontrol edilebilir değişkenlerdir. Karar verici tarafindan karar değişkenlerinin; kavramsal tanımları (üretilecek ürünler, kullanılacak makineler,... v.b.) yapıldıktan sonra bunlar x_i , i=1,2,3...,n şeklinde simgelerle gösterilir.

Sistemde kontrol edilemeyen değişkenlere ise parametre denir. Parametreler, belirli koşullarda karar vericinin kontrolü dışında belirli değerler alabilirler. Bir doğrusal programlama modelindeki parametreler: makine kapasitesi, kalıp kapasitesi, insan gücü, birim başına maliyet, devletin aldığı kararlar, kanunlara uygun olarak fazla mesainin normal mesaiye oranı, bir hammaddenin ürüne dönüşüm oranı… v.b. olabilir. Parametreler de a_j, j=1,2,3,…,n şeklinde simgelerle gösterilir.

Modele girecek olan değişkenler;

$x_1, x_2, \ldots\ldots, x_n$

Değişkenler arasındaki ilişkileri kuran parametreler;

$a_{11}, a_{12}, \ldots\ldots, a_{nm}$

Verilen sabit değerler (ham madde miktarları veya makine kapasiteleri);

$b_1, b_2, \ldots\ldots, b_m$ ile ifade edilir.

Kurulacak doğrusal programlama modelinin çözüm getirmek istediği problemin içinde bulunduğu sistem veya çevre sistemlerden kaynaklanan sınırlayıcı koşullara kısıtlar (sınırlar) denir. Bir anlamda kısıtlar, problemin dayandığı karar ortamında, karar değişkenlerini ve bunlarla parametreler arasındaki ilişkileri etkileyen koşullardır. Sınırlar kullanılan kaynaklardan (malzeme, para, makine, işgücü), çevre sistemlerden (devlet, toplum, müşteriler, rakipler gibi), karar vericinin amaçlarından ve karar değişkenleri arasındaki zorunlu ilişkilerden (montaj hattı iş sıralaması, iş akışı, işlem zamanı gibi) meydana gelebilir.

Aşama 3. Model Parametrelerinin Belirlenmesi

Üretilecek alternatif ürünlerin gelirleri veya masrafları, DP modelinin amaç fonksiyonunun katsayılarını (c_1, c_2, c_3, ...) oluşturur.

Üretimde kullanılan girdilerin (malların, kaynakların) miktarları, kısıtların a parametrelerini, bu kaynakların kapasiteleri ise kısıtların b parametrelerini oluşturur.

Aşama 4. Matematiksel Modelin Kurulması

Problem belirlendikten sonra problemi en iyi şekilde temsil edecek doğrusal bir matematiksel modelin kurulması gerekir. Model gerçeğin basitleştirilmiş bir gösterimidir. Bir doğrusal programlama modeli oluşturulurken aşağıdaki işlemler yapılır.

Aşama 4a. Amaç Fonksiyonunun Belirlenmesi

Amaç fonksiyonu negatif olmayan karar değişkenlerinden oluşur. Çünkü karar değişkenleri, bir malın üretilmesi, bir aracın yaptığı sefer sayısı, bir işin yapılmasında kullanılacak işgücü, bir reklam aracının kullanılması gibi unsurları temsil etmek için modelde yer alır ve bu unsurların da negatif olması düşünülemez. Herhangi bir doğrusal programlama probleminde, karar verici karar değişkenlerinin bazı fonksiyonunu enbüyüklemek veya enküçüklemek ister. Enbüyüklenen veya enküçüklenen fonksiyona amaç fonksiyonu denir. Doğrusal programlamanın varsayımına uygun olarak doğrusaldır. Genellikle kar maksimizasyonu veya maliyet minimizasyonu amacına uygun şekilde kurulurlar.

Doğrusal programlama modelinden istenen sonucun alınabilmesi için amaç fonksiyonunun açık bir şekilde belirtilmesi ve bilinmesi gerekir.

Modelin amaç fonksiyonunda;

Z: Enbüyük veya enküçük yapılacak olan amaç fonksiyonu değeri

c: Karar değişkenlerinin amaç fonksiyonuna katkısı (gelir veya masraf gibi)

X: Karar değişkenleri

a: Teknoloji katsayıları (karar değişkenlerinin üretimi için gerekli kaynak miktarları)

b: Sınırlı kaynak miktarları (kaynak kapasiteleri) ile gösterilirse amaç fonksiyonu;

Max veya Min (z) = $c_1x_1+c_2x_2+c_3x_3+\ldots+c_jx_j+\ldots c_nx_n$ şeklinde yazılır.

Doğrusal programlama problemleri vektör halinde de ifade edilebilir. Problemin amaç fonksiyonunu vektör olarak gösterilimi ifade edilmiştir.

$$Z_{min/max} = [c_1, c_2\ , c_3, \ldots, c_n] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \text{ olur.}$$

Aşama 4b. Kısıtlayıcıların Belirlenmesi:

Problemi şekillendiren koşullar da sınırlar setini oluşturur. Amaç fonksiyonunda olduğu gibi sınırlar seti de matematiksel denklemler şeklindedir. Bu denklemlerde eşitliklerin bulunması tanımlanan kaynakların tümüyle kullanılacağını gösterirken; eşitsizliklerin bulunması kaynak kullanımlarının koşullu olduğunu gösterir. Eğer tüm sınırlar eşitlik biçiminde ifade edildiyse en iyi çözüm tek bir noktada oluşur (denklemlerin kesim noktası), tam tersi olarak tüm sınırlar eşitsizlik şeklindeyse sonsuz sayıda olası çözüm vardır, ancak, amaç fonksiyonunun tek bir en iyi (optimum) değeri vardır.

Modelde yer alan kıt kaynaklarla ilgili sınır şartları;

b_i : i. kıt kaynağın kullanılabilir miktarı (sağ taraf sabitleri), katsayısı

a_{ij} : i. kıt kaynağın, j. madde üretimi için kullanılması gereken miktarı (teknolojik katsayılar) olursa, bu durumda kısıtlayıcı denklem takımı aşağıdaki şekilde ifade edilir.

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1n}x_n \leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + a_{2n}x_n \leq b_2 \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
a_{i1}x_1 + a_{i2}x_i + \dots + a_{ij}x_j + \dots + a_{in}x_n \leq b_i \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mj}x_j + \dots + a_{mn}x_n \leq b_m
\end{array}
$$

Kısıtlayıcılardaki karar değişkenlerinin katsayıları, farklı ürünlerin üretiminde kullanılan teknolojiyi yansıttığı için teknolojik katsayılar adı verilir. Kısıtlayıcıların sağ tarafındaki b_i parametresi elverişli kaynak miktarını gösterir. Fakat bu kaynak miktarı kısıtlayıcı miktarına göre her zaman sınırlı olmaz. Bazen karar değişkenlerinin istediğimizden fazla veya tam eşitlikte olması da mantıklı olan durumdur.

Bu durumda kısıtlayıcıların doğrusal olması gerektiği gibi diğer iki tür temel sınırlayıcı da bulunmaktadır.

$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq b_i$ bazı i değerleri için;

$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$ bazı i değerleri için olur.

Sınır şatları vektör biçiminde ifade edilirse;

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \begin{matrix} \leq \\ = \\ \geq \end{matrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Aşama 4c. Pozitiflik Şartı

Faaliyetler koordinat ekseninin iki değişkenin de pozitif olduğu, birinci bölgesinde meydana geleceğinden karar değişkenleri mutlaka pozitif olacaktır.

Doğrusal programlama, sınırlı kaynakların alternatifler arasında dağıtılması problemi ile ilgilendiğinden negatif faaliyetler veya negatif kaynak kullanımından söz etmek mümkün değildir. Doğrusal programlama modellerinin son kısmı da karar değişkenlerin negatif olmamalarını sağlayan sınırdan oluşmaktadır.

Bu durum j=1,2,3…,n olmak üzere ; $x_j \geq 0$, şeklinde ifade edilir. Matris şeklinde ifade edersek;

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ olur.}$$

Aşama 5. Modelin Çözümü

Model geliştirildikten sonra modelin çözülerek, bilinmeyen değişkenlerinin hesaplanması gerekir. Doğrusal programlama modellerinin çözümünde grafik yöntemi ve simpleks yöntemi kullanılmaktadır. Ancak günümüzde bilgisayar programlarındaki ilerlemelerle beraber, kurulan çok kompleks modeller bile LINDO (Linear Interactive and

Discrete Optimizer), GINO (General Integer and Non-Linear Optimizer), OSL (Optimization Software Library), WINQSB, Microsoft Excel gibi paket programlar ile kısa sürede çözümlenebilmektedir.

Aşama 6. Modelin Çözüm Sonuçlarının Değerlendirilmesi

Modelin çözümü sonucunda ulaşılan karar değişkenlerinin değerleri ile beklenen değerler karşılaştırılır. Elde edilen optimum çözüm sonuçlarının işletme için (yönetim açısından) uygun olup olmadığı incelenir. Uygun olmayan sonuçlar varsa ilk aşamaya geri dönülerek yeni bir model oluşturulur ve tekrar çözülür, sonuçlar yine değerlendirilir.

Bunun sonucunda modelin sistemi hangi ölçüde temsil ettiği, modeldeki varsayımların geçerliliği, model kapsamına alınmış gereksiz değişkenler, ulaşılmak istenen amaçların tutarlılığı ortaya konulmaya çalışılır. Belirlenen hata veya eksiklikler karar modeli üzerinde düzeltilir ve model tekrar çözümlenir.

Aşama 7. Çözümün Uygulanması

Bu aşama, elde edilen sonuçların karar verici tarafından uygulanmasıdır. Doğrusal programlama da amaç fonksiyonu ve kısıtlayıcı fonksiyonların matematiksel anlatımı aşağıdaki işlemlerin yapılmasıyla başlar.

2.6. Doğrusal Programlama Modelini Çözümünde Kullanılan Yöntemler

Doğrusal programlama problemlerinde karar değişkenlerinin sayısı bir veya iki ise grafik yöntemi ile çözülebilir. Ancak doğrusal programlama problemleri genelde çok değişkenli problemler oldukları için grafik yöntemi yerine simpleks adı verilen algoritma yöntemi kullanılır. Fakat çok karmaşık problemlerin manuel olarak çözümü hem çok zaman alıcı hem de çok zahmetli olması, bazen de imkânsız hale gelmesi nedeniyle bu konuda birçok bilgisayar programları oluşturulmuştur. Modelin amaç fonksiyonu ve kısıtlar girilmek suretiyle programlar gerekli işlemleri yaparak bize optimal sonucu vermektedirler.

2.6.1. Grafik Çözüm Yöntemi

Bir doğrusal programlama problemi sadece iki veya üç değişken ile tanımlanabiliyorsa çözüm için grafik yöntemi kullanılabilir. Üçten daha fazla değişken için grafik yöntemi tercih edilen bir yöntem değildir. Grafik yönteminde iki veya üç değişkenden oluşan sınırlar seti, kartezyen sistemde yatay ve düşey eksenlere yerleştirilerek, öncelikle uygun çözüm alanı bulunmaktadır. Uygun çözüm alanı, karar değişkenlerinin pozitif olması şartı sebebiyle kartezyen sistemde birinci bölgede oluşur.

Tüm sınırları sağlayan, uygun çözüm alanındaki tüm noktalar (tamsayılı ve tamsayılı olmayan sonsuz sayıda nokta), ilgili doğrusal programlama modelinin çözüm kümesinde yer alır. Ancak, optimum çözüm, sadece, amaç fonksiyonunun değerini maksimize veya minimize eden karar değişkenleri değerlerinden oluşmaktadır. Her hangi bir doğrusal programlama modelinin optimum çözümünün uygun çözüm alanının köşe noktalarından birinde yer alacağı matematiksel olarak ispatlandığından (Özgüven, 2003: 16), uygun çözüm alanının köşe noktaları belirlenerek, bu noktalar amaç fonksiyonu üzerinde denenir ve optimum çözüm bulunur. Bir doğrusal karar modelinin grafik çözümünde incelenecek adımlar şunlardır;

Adım 1: Her bir kısıt eşitlik haline getirilerek, karşı gelen doğrunun grafiği çizilerek, kısıtı sağlamayan karar değişkenlerinin olduğu bölge taranır. Düzlemde taranmayan bölge kaldıysa, burası uygun çözüm alanıdır ve ikinci adıma geçilir. Eğer düzlemde taranmayan bölge kalmadıysa, modelin kısıtlarını sağlayan karar değişkenleri yok demektir ve bu adımda durulur. Böyle bir durumda uygun çözüm alanı boş olan, çözümsüz model sözkonusudur.

Adım 2: Amaç fonksiyonuna, bir başlangıç değeri karşı gelen doğrunun grafiği çizilir. Başlangıç değeri sürekli olarak arttırılarak ilk çizilen doğruya paralel doğrular çizilmeye devam edilir. Artan başlangıç değerlerine karşı gelen paralel doğru demetinin uygun çözüm alanına ilk girdiği nokta minimizasyon için, son terk ettiği nokta maksimizasyon için en iyi çözümdür.

Adım 3: Paralel doğru demeti, amaç fonksiyonunun azalan değerlerine göre uygun çözüm alanını terk etmiyorsa minimizasyon için, artan değerlerine göre uygun çözüm alanının dışına çıkmıyorsa maksimizasyon için, sınırsız çözüm olup durulur. Böyle durumlarda, amaç fonksiyonu uygun çözüm alanı üzerinde istenen yönde sonlu olmadığından, karar vericiye en iyi seçenek önerilemez.

Maksimizasyon Problemlerinin Grafik Çözümü

Max $z = 7x_1 + 5x_2$

Kısıtlayıcılar

$4x_1 + 3x_2 \leq 240$

$2x_1 + x_2 \leq 100$

Pozitiflik Kısıtı

$x_1, x_2 \geq 0$

İlk önce modelin sınırlarının kartezyen sistemde eksenlere taşınması gerekmektedir. Bunun için sınırlarda yer alan x_1 ve x_2 değerleri bulunmalı ve bulunan değerler için 1. ve 2. sınırlarda yer alan doğru denklemleri çizilmelidir.

1. Sınır denklemi için: $4x_1 + 3x_2 \leq 240$ şeklindeki eşitsizlik

$4x_1 + 3x_2 = 240$ şeklinde eşitliğe dönüştürülür.

$x_1 = 0$ için; $x_2 = 80$, $x_2 = 0$ için; $x_1 = 60$ değerleri elde edilir.

2. Sınır denklemi için: $2x_1 + x_2 \leq 100$ şeklindeki eşitsizlik

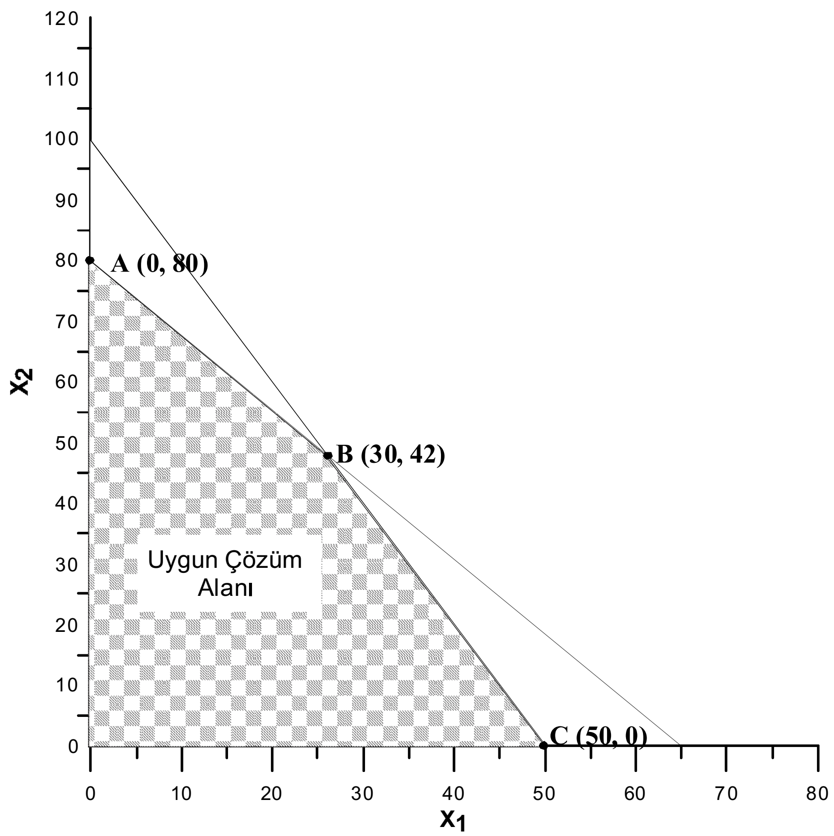
$2x_1 + x_2 = 100$ şeklinde eşitliğe dönüştürülür.

$x_1 = 0$ için; $x_2 =100$, $x_2 = 0$ için; $x_1 = 50$ değerleri elde edilir.

Modelin tüm sınırlarını ve pozitiflik şartını sağlayan bölge, kartezyen sisteminin birinci bölgesidir. Eşitsizlik yönüne göre, doğruların kesiştikleri bölgelerin altında kalan alan uygun çözüm alanı (U.Ç.A) olarak belirlenir. Amaç fonksiyonun maksimize edilmesi istendiğine göre, uygun çözüm alanının köşe noktaları belirlenir ve optimum çözüm bu noktalarda aranır.

Grafik 1’ de görüldüğü gibi, uygun çözüm alanında a,b ve c olmak üzere üç köşe noktası işaretlenmiş, bu noktaların x1 ve x2 değerleri hesaplanmıştır. Köşe noktaları amaç fonksiyonuna yerleştirilerek en yüksek değer 420 olarak b noktasında bulunur ki, bu değer de modelin optimum çözümünü vermektedir.

Grafik 1: Maksimizasyon Probleminin Grafik Çözümü



A(0; 80) noktası için $7x_1 + 5x_2 = 7(0) + 5(80) = 400$

B(30;42) noktası için $7x_1 + 5x_2 = 7(30) + 5(42) =$ **420**

C(50; 0) noktası için $7x_1 + 5x_2 = 7(50) + 5(0) = 350$

Minimizasyon Problemlerinin Grafik Çözümü

Teorik olarak minimizasyon problemlerinin grafik çözümünün maksimizasyon problemlerininkinden bir farkı olmamakla birlikte, minimizasyon problemlerinde ortaya çıkan konveks çokgensel bölgenin kapalı bir alan oluşturmayabileceğine dikkat etmek gereklidir.

Örnek:

Min $z = 2x_1 + 3x_2$

Kısıtlayıcılar

$5x_1 + 10x_2 \geq 90$

$4x_1 + 3x_2 \geq 48$

$0.5x_1 \geq 1,5$

Pozitiflik kısıtı

$x_1, x_2 \geq 0$

Maksimizasyon probleminde olduğu gibi, ilk önce modelin sınırlarının kartezyen sistemde eksenlere taşınması gerekmektedir. Bunun için sınırlarda yer alan x_1 ve x_2 değerleri bulunmalı ve bulunan değerler için 1. ve 2. Sınırlarda yer alan doğru denklemleri çizilmelidir.

1. Sınır denklemi için: $5x_1 + 10x_2 \geq 90$ şeklindeki eşitsizlik

   $5x_1 + 10x_2 = 90$ şeklinde eşitliğe dönüştürülür.

$x_1 = 0$ için; $x_2 = 9$, $x_2 = 0$ için; $x_1 = 18$ olur.

2. Sınır denklemi için: $4x_1 + 3x_2 \geq 48$ şeklindeki eşitsizlik

   $4x_1 + 3x_2 = 48$ şeklinde eşitliğe dönüştürülür.

$x_1 = 0$ için; $x_2 = 16$, $x_2 = 0$ için; $x_1 = 12$

3. Sınır denklemi için: $0.5x_1 \geq 1,5$ şeklindeki eşitsizlik

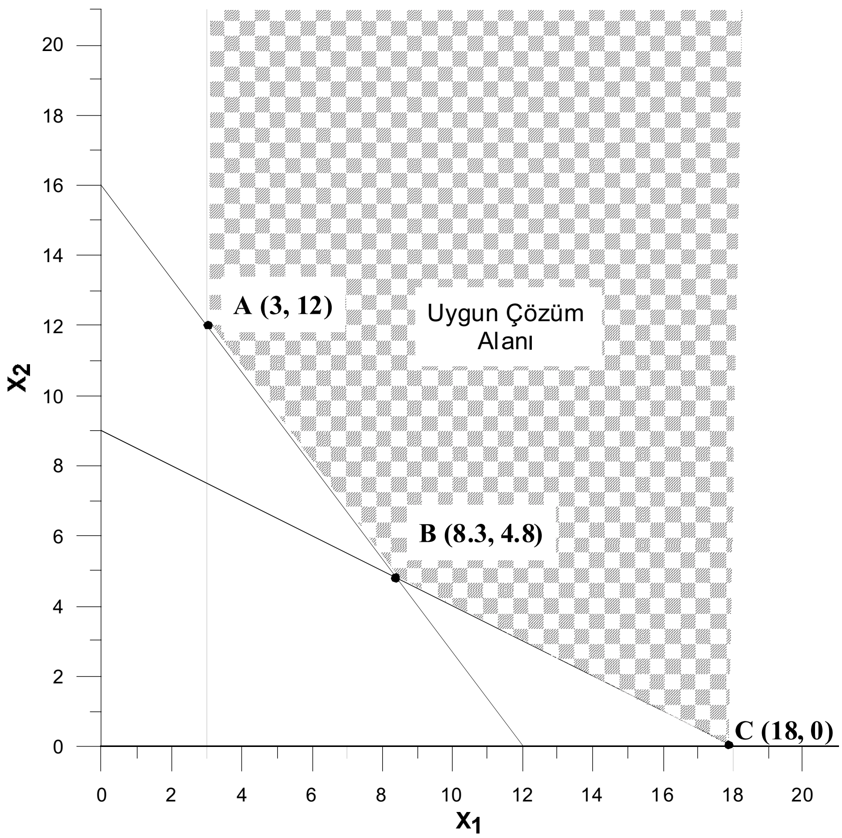
   $0.5x1 = 1,5$ şeklinde eşitliğe dönüştürülür.

$x_1 = 3$ olur.

Modelin tüm sınırlarını ve pozitiflik şartını sağlayan bölge, kartezyen sisteminin birinci bölgesidir. Eşitsizlik yönüne göre, doğruların kesiştikleri bölgelerin üstünde kalan alan uygun çözüm alanı (U.Ç.A) olarak belirlenir. Amaç fonksiyonun minimize edilmesi istendiğine göre, uygun çözüm alanının köşe noktaları belirlenir ve optimum çözüm bu noktalarda aranır.

Grafik 2' de görüldüğü gibi, uygun çözüm alanında a,b ve c olmak üzere üç köşe noktası işaretlenmiş, bu noktaların x_1 ve x_2 değerleri hesaplanmıştır. Köşe noktaları amaç fonksiyonuna yerleştirilerek en yüksek değer 420 olarak b noktasında bulunur ki, bu değer de modelin optimum çözümünü vermektedir.

Grafik 2: Minimizasyon Probleminin Grafik Çözümü



A (3; 12) noktası için; $2x_1 + 3x_2 = 2(3) + 3(12) = 42$

B (42.5; 24.5) noktası için; $2x_1 + 3x_2 = 2(42.5) + 3(24.5) = \mathbf{156.5}$

C (18; 0) noktası için; $0.5x_1 = 2(18) + 3(0) = 36$

2.6.2. Simpleks Çözüm Yöntemi

Doğrusal programlama problemlerini çözmede yaygınca kullanılan simpleks yöntemi ilk kez 1947 yılında G.B. Dantzig tarafından geliştirilerek ABD Hava Kuvvetlerinin planlanmasında kullanılmıştır. Daha sonra Charnes, Cooper ve diğerleri ekonomik ve endüstriyel analizler için uygulamalı öncü çalışmalar yapmışlardır.

Grafik yöntemi en fazla üç değişkenli problemler için uygulanırken, problemin değişken sayısının attığı durumlarda simpleks çözüm yöntemi uygulanır. Simpleks Yöntemi, amaç fonksiyonunu en büyük (maksimum) veya en küçük (minimum) yapacak en iyi çözüme adım adım yaklaşan bir algoritma (hesaplama yöntemi)'dir. Bu nedenle, probleme bir uç noktasından başlayarak optimuma daha yakın bir ikincisine, oradan bir üçüncüsüne... atlayarak en iyi çözümü veren uç noktasına ulaşmamızı sağlar. Simpleks yönteminin temel özelliği çözüme iterasyon (cebirsel tekrar) yoluyla ulaşmasıdır. Simpleks yönteminin kullandığı cebirsel işlemler, Gauss elimizasyon yöntemine benzer. "Gauss eliminasyon" terimi Karl Gauss'a adanmış olup, her adımda bir değişken cari çözümden atılır ve bir diğeri onun yerine konur.

Çözüme başlamadan önce, ilk olarak, modelde bulunan eşitsizlikler halindeki kısıtlar dolgu ve artık değişkenler kullanılarak eşitlik haline dönüştürülerek modelin standart doğrusal programlama haline gelmesini sağlanmalıdır. Bir modelin standart hale gelmesi için; değişkenlerin negatif olmaması koşulunun içeren kısıtların dışındaki tüm kısıtlar, negatif olmayan sağ tarafa sahip eşitlikler haline dönüşmüş olmalı, tüm değişkenler sıfır ya da pozitif olmalı, amaç fonksiyonu maksimum ya da minimum olmalıdır.

Problemin standart modeli kurulduktan sonra, başlangıç simpleks tablosu oluşturulur ve aşamalar halinde belirli bir hesap yöntemi içinde gelişen çözümlere doğru ilerleyerek optimal çözüme ulaşıncaya kadar işlemler sürdürülür. Simpleks yönteminin hedefi, eldeki kaynakların en karlı şekilde nasıl kullanılması gerektiğini belirlemektir.

2.6.2.1. Simpleks Yönteminin Uygulama Alanları

Simpleks yönteminin bütün Doğrusal Programlama problemlerinde uygulanabilir olmasına karşın, bazı tür problemler başka yöntemler kullanılarak daha etkin bir şekilde çözülebilir. Bunun yanında Doğrusal Programlama dışında başka bir yöntemle çözümlenemeyen problemler de vardır. Burada, çözümlerinde Simpleks yöntemin kullanılması gereken bazı problemlere kısaca değinilecektir.

Bunlar:

Karışım problemleri: Burada gıda sanayisinde, petrol sanayisinde, metalürjide maliyetleri de göz önüne alarak yapılacak karışımların bünyelenmesinde en uygun yol araştırılır.

Optimum üretim programının saptanması: Burada ise sınırlı kaynakların (kapasite, hammadde, işgücü, vs.) optimum kullanımı ile karı maksimize veya maliyetleri minimize etme problemleri söz konusudur.

İş ve ücret değerlemesi: Doğrusal Programlama değerleme işleminde açık olarak ilgilenilen faktörler için uygun ağırlıkların saptanması için çoklu korelasyon analizi yerine kullanılabilir.

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

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

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

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

Bütün bunların yanında, bu yöntemin uygulanabileceği daha değişik problemlere de rastlanabilir. Kaldı ki, yöntem herhangi bir Doğrusal Programlama probleminin çözümünde genel olarak uygulanabilmektedir.

2.6.2.2. Simpleks Yönteminin Uygulanması

Simpleks çözüm yönteminin uygulanabilmesi için birinci adım olarak kanonik şekildeki doğrusal programlama probleminin standart hale getirilmesi gerektiğini belirtilmişti. Şimdi problemi standart forma getirmek için kullanılan yöntemler incelenecektir.

Kısıtlamalar

Herhangi bir "≥" eşitsizlik "≤" eşitsizliğine dönüştürülebilir. Bunun için tek yapılması gereken eşitliğin iki yanının da "-1" ile çarpılmasıdır. Standart formda kısıtlamaların eşitlik olması gerekmektedir. Bunu yapabilmek için eşitsizliğin türüne göre yeni değişkenler kullanılmak durumundadır "≤ " Eşitsizliğini, sol tarafa bir atıl değişkeni ekleyerek eşitlik haline getirilebilir. Negatif olmayan atıl değişken kısıtlamada kullanılmayan kapasiteyi gösterir. "≥" Eşitsizliğini ise bu sefer eşitsizliğin sağ tarafına bir yetersiz kapasite değişkeni ekleyerek yaparız. Kısıtlamaların sağ taraflarında sadece değerlerin olmasını istediğimizden bu yeni değişkeni eşitsizliğin sol tarafından çıkartılır.

$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n \leq b_1 \leftrightarrow a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n + s_1 = b_1$ şekline dönüşür.

$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n \geq b_2 \leftrightarrow a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n - s_2 = b_2$ şekline dönüşür.

$s_1, s_2 \geq 0$

Standart formda eşitliğin sağ tarafının pozitif olması gerekmektedir. Bunu sağlamak için eşitlik "-1" ile çarpılır.

$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n - s_2 = -b_1 \leftrightarrow -a_{11}x_1 - a_{12}x_2 - \ldots - a_{1j}x_j + \ldots - a_{1n}x_n + s_2 = b_1$ şekline dönüştürülür.

Değişkenler

Standart formda bütün değişkenlerin sıfır veya sıfırdan büyük, pozitif değerler taşıyacak değişkenler olması gerekmektedir. Eğer problemde sınırlandırılmamış, negatif ya da pozitif değerler alabilecek bir değişken var ise, bu değişkeni negatif olmayan iki ayrı değişken arasındaki fark olarak tanımlanabilir. Bu durumda optimal simpleks çözümünde değişkenlerden sadece bir tanesi pozitif değer alacaktır.

$x_j = x_j^+ - x_j^-$ $\quad\quad$ $x_j^+ - x_j^- \geq 0$ olur.

Amaç Fonksiyonu

Her ne kadar standart form her tür amaç fonksiyonu kullanmaya izin veriyorsa da bazı durumlarda maksimizasyon ve minimizasyon arasında dönüşüm yapmak gerekebilir. Bunu yapmak için amaç fonksiyonu "-1" ile çarpılır. Yani bir $f(x_1, x_2, x_3, \ldots, x_n)$ fonksiyonunun maksimizasyonu, $- f(x_1, x_2, x_3, \ldots, x_n)$ fonksiyonunun minimizasyonuyla eşdeğerdir.

Genel Örnek:

Standart model: Maksimum $Z = 5x_1 + 7x_2$

Kısıtlamalar;

$2x_1 + 4x_2 \leq 15$

$6x_1 + 9x_2 = 10$

$9x_1 + 2x_2 \geq 28$

x_1 sınırlandırılmamış

$x_2 \geq 0$

Kanonik model: Maksimum $Z = 5x_{11} - 5x_{12} + 7x_2$

Kısıtlamalar;

$2x_{11} - 2x_{12} + 4x_2 + s_1 = 15$

$6x_{11} - 6x_{12} + 9x_2 = 10$

$9x_{11} - 9x_{12} + 2x_2 - s_2 = 28$

$x_{11}, x_{12}, x_2, s_1, s_2 \geq 0$ olur.

Temel Çözümün Belirlenmesi

Simpleks metodu uygun bir çözüm ile başlar ve her aşamada daha iyi bir çözüm bularak ilerler. Sistemli bir şekilde ilerlerken daha iyi bir çözüm bulunamayacak duruma gelindiğinde optimal çözüm bulunmuş olur. Simpleks yönteminin çalışabilmesi için ilk olarak uygun temel çözümün bulunması gerekmektedir. İlk uygun temel çözüme ulaşabilmek için kısıtlamalarda ve amaç fonksiyonunda bazı değişiklikler yapmak durumunda kalınabilir.

Standart hale gelmiş bir doğrusal programlama problemi n bilinmeyenli m tane eşanlı doğrusal eşitlik ($m < n$) içerir. Bir eşitlik sisteminde m eşitlik ve n değişken varsa ve değişken sayısı eşitlik sayısından büyükse tam çözüm bulmak matematiksel olarak mümkün değildir. Bu n adet değişkenin $n - m$ adedi sıfır değerini alır, arta kalan m adet değişkenin m tane eşitliğin çözülmesiyle değeri belirlenir. Bu m adet değişken tek bir çözüm veriyorsa bu durumu sağlayan değişkenlere *temel değişken*, kalan $n - m$ adet sıfır değerli değişkene de *temel dışı değişken* denir. Bu durumda tekil çözümle sonuçlanan durum temel çözümü de içermektedir. Tüm değişkenlerin negatif olmayan değerler aldığını varsayarsak, bu durumda temel çözüm uygundur. Aksi halde uygun değildir. Şimdi verilen tanımlar bir örnek yardımıyla açıklanmaya çalışılacaktır.

Genel Örnek:

$x_1 + x_2 + 4x_3 + 2x_4 + 3x_5 = 8$

$4x_1 + 2x_2 + 2x_3 + x_4 + 6x_5 = 4$ şeklinde beş bilinmeyenli iki eşitliği ($m = 2$, $n = 5$) göz önüne alınsın.

Tanıma göre, bir temel çözüm sadece iki temel değişken içerebilir. Bu da sıfır değeri alan temel dışı değişkenlerin üç olabileceği anlamına gelir.

Durum 1. Uygun Temel Çözüm

Sıfır değeri alan (temeldışı) değişkenler: (x_2, x_4, x_5)

Eşitlikler: $x_1 + 4x_3 = 8$

$4x_1 + 2x_3 = 4$

Çözüm: $x_1 = 0$, $x_3 = 2$ ile tekil

Durum: x_1 ve x_3 temel değişkenleri ≥ 0 olduğundan uygun temel çözümdür.

Durum 2. Uygun Olmayan Temel Çözüm

Sıfır değeri alan (temeldışı) değişkenler: (x_3, x_4, x_5)

Eşitlikler: $x_1 + x_2 = 8$

$4x_1 + 2x_2 = 4$

Çözüm: $x_1 = -6$, $x_2 = 14$ ile tekil

Durum: $x_1 \leq 0$ olduğundan uygun olmayan çözüm olur.

Durum 3. Sonsuz Çözüm (Sınırsız Çözüm)

Sıfır değeri alan (temeldışı) değişkenler: (x_1, x_2, x_5)

Eşitlikler: $4x_3 + 2x_4 = 8$

$2x_3 + x_4 = 4$

Çözüm: eşitliklerin birbirine bağımlı olması nedeniyle tekil çözüm yoktur.

Durum: Sonsuz (sınırsız) çözüm.

Durum 4. Çözüm Olmaması

Sıfır değeri alan (temeldışı) değişkenler: (x_1, x_3, x_4)

Eşitlikler: $x_2 + 3x_5 = 8$

$2x_2 + 6x_5 = 4$

Çözüm: Eşitliklerin tutarsızlığı nedeniyle çözüm yoktur.

Durum: Çözüm yoktur (Taha, 2000: 70).

Başlangıç Tablosunun Oluşturulması

İlk aşamada, aylak değişkenler kullanılarak eşitsizlikler eşitlik haline getirilir. s_1, s_2, …, s_n'in aylak değişkenleri ifade etmektedir. Bu değişkenler eşitsizliklerin sol tarafına eklenerek, eşitsizlikler eşitlik haline getirilir. Bu değişkenler amaç fonksiyonundaki katsayılar ise 0 olur. Dolayısı ile amaç fonksiyonu değişmez. Bu işlemlerden sonra başlangıç simpleks tablosu oluşturulur. Tablodaki c_j değerleri, karar değişkenlerinin amaç fonksiyonundaki katsayılarını ifade eder. Karar değişkenlerinin ve aylak değişkenlerin altlarındaki sütunlara yazılan değerler ise, ilgili karar değişkenlerinin veya aylak değişkenin kısıttaki katsayısını ifade etmektedir. Ayrıca başlangıç simpleks tablosu, üretimin henüz başlamadığı bir aşamayı gösterdiğinden Z_j değerleri 0'a eşit olur. Burada

Z_j, amaç fonksiyonunun alabileceği değerleri göstermektedir. Simpleks metodu her adımda en çok kazanç sağlayacak değişkenin temel değişkenler grubuna katılmasını ve en az getiri sağlayanın temel değişken grubundan ayrılması esasına göre çalışmaktadır.

Çözüme geçmeden önce simpleks yönteminin aşamaları kısaca özetlenirse;

- Yukarıda da belirtildiği gibi, birinci aşamada, eşitsizliklerin eşitlik haline dönüştürülerek tablo 2 'de görülen simpleks çizelgesi düzenlenir. Bu eşitlik haline dönüştürme, maksimizasyon için artık değişkenler ve minimizasyon için ise artık ve yapay değişkenlerin ilavesi ile yapılır.

Tablo 2: Simpleks Çizelgesi

<table>
<tr><td>K.K.</td><td>C_j</td><td></td><td rowspan="2">Ç.V.</td></tr>
<tr><td>M.K.</td><td>T.D.V.</td><td></td></tr>
<tr><td></td><td></td><td>Katsayı Matrisi
Birim Matris</td><td></td></tr>
<tr><td colspan="2">Z_j</td><td></td><td></td></tr>
<tr><td colspan="2">C_j - Z_j</td><td></td><td></td></tr>
</table>

K.K. : Kâr Katsayısı, T.D.V. : Toplam Değişken Vektör
C.V. : Çözüm Vektörü, M.K. : Maliyet Katsayısı

- Z_j satırı hesaplanır. Bunun için-kâr katsayıları sütunu ile katsayılar matrisi, birim matris ve çözüm vektörü sütunundaki sayılar çarpılır ve alt alta toplanır.

- (C_j - Z_j) satırı hesaplanır (Z_j satırındaki değerler C_j satırındaki değerlerden çıkarılır.

- Maksimizasyon sorunlarında çizelgedeki (C_j - Z_j) satırındaki pozitif işaretli katsayılar arasından en büyüğü seçilir. Simpleks tablosuna girilen bu kolona

"Anahtar Kolon" denir, ve temel değişken vektörüne hangi değişkenin gireceğini saptar.

Minimizasyon sorunlarında ise simpleks çizelgesine (C_j - Z_j) satırındaki negatif işaretli, temel olmayan değişkenler arasındaki en küçük değerden girilir.

- Çözüm vektöründeki değerlerle seçilen kolondaki değişken katsayılar arasındaki oranlar bulunur. Bu oranlar içinde sıfır ve negatif olanlar dikkate alınmadan en küçüğü seçilir. Bu en küçük değerin bulunduğu satıra da "Anahtar Satır" denir. Ve temel değişken vektöründen hangi değişkenin çıkacağını saptar.

- Anahtar kolonla anahtar satırın kesiştiği yerdeki "Anahtar Sayı" bulunur.

- Anahtar sayı, bulunduğu satırdaki bütün sayılara teker teker bölünerek yeni çözüm vektörü ve diğer öğeler bulunur.

- Diğer satırların ve (C_j - Z_j) satırının yeni öğeleri ise;

(Eski satır öğeleri) - (Satırın anahtar kolondaki sayısı) x (Anahtar satırın yeni öğeleri) = (Yeni satır öğeleri) formülüne göre teker teker saptanır.

- Maksimizasyon sorunlarında (C_j - Z_j) satırında bulunan bütün katsayılar sıfır, ya da negatif işaretli ise optimal çözüme ulaşılmıştır. Ve sonuç çözüm vektöründedir. Aksi takdirde (C_j - Z_j) satırındaki sayılar negatif ya da sıfır olana dek devam edilir.

Minimizasyon sorunlarına ise, (C_j - Z_j) satırındaki ve sütunundaki bütün elemanlar sıfır ya da pozitif işaretli ise optimal çözüme ulaşılmıştır. Değilse, (C_j - Z_j) satırındaki sayıların tümünün sıfır ya da pozitif olana dek işlemlere aynen devam edilir (Yalgın, 1984: 30).

Şimdi örnekler yardımıyla problemlerin simpleks yöntemiyle nasıl çözümlendiği daha iyi anlatılacaktır.

Maksimizasyon Problemlerinin Simpleks Çözümü

Örnek:

Amaç fonksiyonu

$$\text{Max } Z = 10x_1 + 17x_2 \longrightarrow \text{Max } Z = 10x_1 + 17x_2 + 0s_1 + 0s_2$$

Kısıtlayıcılar;

$$3x_1 + 5x_2 \leq 34 \longrightarrow 3x_1 + 5x_2 + s_1 = 34$$

$$2x_1 + 11x_2 \leq 61 \longrightarrow 2x_1 + 11x_2 + s_2 = 61$$

$$x_1, x_2, s_1, s_2 \geq 0$$

Şimdi yukarıdaki adımlar takip edilerek, optimal simpleks tablosuna ulaşılmaya çalışılacaktır. İkinci aşama olarak başlangıç simpleks tablosunu oluşturulsun.

Tablo 3: Başlangıç Simpleks Tablosu (Maksimizasyon)

C_j	10 X_1	17 X_2	0 S_1	0 S_2	b_j	
0 s_1	3	5	1	0	34	34/5 = 6,8
B	2	**11**	0	1	61	61/11= 5,5
Z_j	0	0	0	0		
C_j - Z_j	10	17	0	0		

Amaç fonksiyonu maksimum olduğundan, $C_j - Z_j$ satındaki en büyük değerin bulunduğu sütun, anahtar sütundur. Bundan sonra b_j sütunundaki elemanlardan her biri, kendi satırındaki anahtar sütun elemanına bölünürler. Bu bölme sonucu çıkan değerlerden en küçüğünü sağlayan satır anahtar satırdır. İkinci tabloda bu satır çözümden çıkarılır ve onun yerine birinci tabloda en büyük sütunu sağlamış olan karar değişkeni veya aylak değişken aynı satırda çözüme girer. Bu değişkene ait satır elemanı ise, birinci tablodaki satır elemanlarının pivot sayıya (anahtar satır ile anahtar sütunun birleştiği yerdeki sayı) bölünmesiyle elde edilir. Yukarıda tablo 3'e bakıldığında anahtar sayının 11 anahtar sütun x_2 ve anahtar satır B satırıdır.

Diğer satır elemanları ise şu formülle bulunur;

Yeni satır el.	**=**	**Eski satır el.**	**- (eski satırın**	**anahtar kolondaki el.**	**×**	**anahtar satırın yeni el.**	**)**
23/11	**=**	**3**	**– (**	**5**	×	**2/11**	**)**
0	**=**	**5**	**– (**	**5**	×	**1**	**)**
1	**=**	**1**	**– (**	**5**	×	**0**	**)**
-5/11	**=**	**0**	**– (**	**5**	×	**1/11**	**)**
69/11	**=**	**34**	**– (**	**5**	×	**61/11**	**)**

Tablo 4: Birinci Simpleks Çözüm Tablosu

C_j	10 X_1	17 X_2	0 S_1	0 S_2	b_j	
0 s_1	**23/11**	0	1	-5/11	69/11	69/23 = 3
17 x_2	2/11	1	0	1/11	61/11	61/2 =30,5
Z_j	34/11	17	0	17/11		
C_j - Z_j	76/11	0	0	-17/11		

Yeni S. El.	**=**	**Eski Satır El.**	**- (Eski Satırın**	**Ana. Kolondaki El.**	**×**	**Anahtar Satırın Yeni El.**	**)**
0	**=**	**2/11**	**– (**	**2/11**	×	**1**	**)**
1	**=**	**1**	**– (**	**2/11**	×	**0**	**)**
-2/23	**=**	**0**	**– (**	**2/11**	×	**11/23**	**)**
3/23	**=**	**1/11**	**– (**	**2/11**	×	**-5/23**	**)**
5	**=**	**61/11**	**– (**	**2/11**	×	**3**	**)**

Tablo 5: İkinci (Optimal) Simpleks Çözüm Tablosu

C_j	10 X_1	17 X_2	0 S_1	0 S_2	b_j
10 x_1	1	0	11/23	-5/23	3
17 x_2	0	1	-2/23	3/23	5
Z_j	10	17	76/23	1/23	**115**
C_j - Z_j	0	0	-76/23	-1/23	

Problem maksimizasyon problemi olduğu için, C_j – Z_j çözüm değerlerinin 0 veya negatif olması gerektiği için, problem de ikinci tabloda optimum sonuca ulaşılmış oldu. Yukarıda tablo 3'te oluşturulan yeni satır ve sütun elemanlarını yerleştirildikten sonra, çözüme devam edilir. Tablo 4'te anahtar sayı 23/11'dir. Aynı şekilde yeni satır ve sütun elemanlarını bulunduktan sonra tablo 4'ü elde edilir ki bu optimal çözüm tablosudur.

Elde edilen çözüm değerlerine göre; x_1 = 3, x_2 = 5 olduğunda maksimum kar = 115 olur.

Minimizasyon Problemlerinin Simpleks Çözümü

Tüm kısıtların ≤ olması durumunda artık değişkenler eklenerek problem standart hale getiriliyordu. Ancak ≥ ve = yönündeki kısıtlayıcılar için artık değişken eklemek yerine çıkarmak gerekir ki bu da başlangıç uygun çözümünü bozar. Bu tür koşullarda bu durumu önlemek için yapay değişken kullanılmaktadır. Artık (surplus) değişken, eşitliğin sol tarafında yer alır ve negatif katsayılı bir değişken olarak temsil edilir. Yapay değişken ise, optimal çözümde bulunmaması ve sıfır değer alabilmesi için çok yüksek bir maliyeti (M) ifade eder. Maksimizasyon amaçlı bir problemde, yapay değişkenin katsayısı çok büyük ceza/gider (-M) gösterirken, minumum amaçlı problemde ise çok yüksek bir maliyeti (M) gösterir (Wagner, 1969: 112) .

(=) şeklindeki bir kısıta, sadece bir yapay (A_i) değişken ilave edilerek çözüme geçilir. Yapay değişkenler başlangıç tablosunda taban değişkenleri oluştururlar, daha sonra bu değişkenler tabandan çıkarılmaya çalışılırlar. Artık değişkenlerin amaç fonksiyonundaki katsayıları sıfır olmasına karşın, yapay değişkenlerin amaç fonksiyonundaki katsayıları çok büyük pozitif değerli bir sayıyı gösteren M sayısıdır. M aynı zamanda bir ceza katsayısıdır.

Bu ceza nedeniyle yapay değişkenler iterasyonlar sonunda sıfır değerini almaya çalışacaktır.

2.6.2.3. Büyük M Yöntemi

Büyük M metodu, A.Charnes 2 tarafından bulunmuştur. Problem bir maksimizasyon problemi ise, M>0 çok büyük bir değeri gösterir ve yapay değişkene (-M) fiyatı verilir. Problem bir minimizasyon ise, bu değişkenlere (+M) fiyatı verilir. Bu fiyatlar amaç fonksiyonunun mahiyetine göre ulaşmak istediğimiz en karşıt fiyatlar olduğundan yapay değişkenlerin temel değişkenlerden birer birer çıkmaları beklenir. Bir yapay değişken bir defa temel değişkenlerden çıkınca bir daha çözüme alınmaz (Avralıoğlu,1981: 115).

Büyük M yöntemi minimizasyon problemlerine uygulanırken aşağıdaki adımlara göre hareket edilmelidir.

Adım 1: sağ taraf sabitlerinin negatif olmaması sağlanmalıdır. Orijinal modelde var ise, sağ taraf sabiti negatif olan her bir sınırın her iki tarafını da -1 ile çarpılır. Bu işlem ilgili eşitsizliklerin yönlerini değiştirir.

Adım 2: i'inci sınır (≤) biçiminde bir eşitsizlik ise, bu sınıra Si boş değişkeni eklenir. i'inci sınır (≥) biçiminde bir eşitsizlik ise, bu sınıra V_i artık değişkeni ve A_i yapay değişkeni eklenir. i'inci sınır değişkeni (=) biçiminde ise, bu sınıra A_i yapay değişkeni eklenir.

Adım 3: Her bir yapay değişken için amaç fonksiyonuna $-MA_i$ terimi eklenir.

Adım 4: genişletilmiş sistem üzerinde amaç satırını güncelleştirme işlemi yapılır. Kanonik forma sokulan genişletilmiş sistem başlangıç simpleks tablosuna yerleştirilir. C_j – Z_j satırında negatif sayı kalmayana kadar işleme devam edilir (Özgüven, 2003: 72).

Örnek:

Amaç fonksiyonu

$$\text{Min } Z = 2x_1 + 2x_2 \longrightarrow 2x_1 + 2x_2 + MA_1 + MA_2 + 0V_1 + 0V_2$$

Kısıtlayıcılar;

$$11x_1 + 7x_2 \geq 79 \longrightarrow 11x_1 + 7x_2 + A_1 - V_1 = 79$$

$$5x_1 + 13x_2 \geq 85 \longrightarrow 5x_1 + 13x_2 + A_1 - V_2 = 85$$

$$x_1, x_2, A_1, A_2, V_1, V_2 \geq 0$$

Yapılan bu işlemlerden sonra, aşağıdaki gibi simpleks başlangıç tablosu oluşturulur. Tablo 6'daki C_j değerleri, karar değişkenlerinin amaç fonksiyonundaki katsayıları olup amaç fonksiyonu minimum olduğundan dolayı, iktisadi olarak, birim maliyetleri gösterirler. Karar değişkenlerinin, yapay değişkenlerin ve aylak değişkenlerin altlarındaki sütunlara yazılan değerler ise, ilgili karar değişkenlerinin, yapay değişkenlerin ve aylak değişkenlerin kısıtlardaki katsayılarını ifade eder.

Tablo 6: Başlangıç Simpleks Tablosu (Minimizasyon)

C_j	2 x_1	2 x_2	M A_1	0 V_1	M A_2	0 V_2	b_j	
M A_1	11	7	1	-1	0	0	79	79/7= 11,2
M A_2	5	**13**	0	0	1	-1	85	85/13= 6,5
Z_j	16M	20M	M	-M	M	-M		
C_j - Z_j	2- 16M	2 -20M	0	M	0	M		

Amaç fonksiyonu minimum olduğundan, $C_j - Z_j$ satırındaki en küçük değerin bulunduğu sütun anahtar sütundur. Sonraki adımda b_j sütunundaki her bir sayı, kendi satırlarındaki anahtar sütun elemanına bölünürler. Bu bölüm sonucu çıkan değerlerden en küçüğünün olduğu satır anahtar satır olur. Tablo 6'da, bu satır çözümden çıkarılır ve onun yerine tablo 5'ten en büyük sütunu sağlamış olan karar değişkeni veya aylak değişken aynı satırda çözüme girer. Bu değişkenlere ait satır elemanları ise, tablo 5 teki (eski) satır elemanlarının pivot sayıya bölünmesiyle bulunur.

Diğer satır elemanları ise şu formülle bulunur:

Yeni S. El.	**=**	**Eski Satır El.**	**- (Eski S.**	**Anahtar Kolondaki El.**	**×**	**Anahtar S. Yeni El.)**	
108/13	**=**	**11**	**– (**	**7**	×	**5/13**	**)**
0	=	**7**	**– (**	**7**	×	**1**	**)**
-1	=	**-1**	**– (**	**7**	×	**0**	**)**
-7/13	=	**0**	**– (**	**7**	×	**1/13**	**)**
7/13	=	**0**	**– (**	**7**	×	**-1/13**	**)**
432/13	=	**79**	**– (**	**7**	×	**85/13**	**)**

Tablo 7: Birinci Simpleks Çözüm Tablosu

C_j	2 x_1	2 x_2	M A_1	0 V_1	M A_2	0 V_2	b_j	
M A_1	**108/13**	0	1	-1	-7/13	7/13	432/13	4
2 x_2	5/13	1	0	0	1/13	-1/13	85/13	17
Z_j	$\frac{10+108/M}{13}$	2	M	-M	$\frac{2-7M}{13}$	$\frac{-2+7M}{13}$		
C_j- Z_j	$\frac{11-108M}{13}$	0	0	M	$\frac{-2+20M}{13}$	$\frac{2-7M}{113}$		

Yeni S. El.	=	Eski Satır El.	- (Eski S. Anahtar Kolondaki El.	×	Anahtar S. Yeni El.)
0	=	**5/13**	– (**5/13**	×	**1**)
1	=	**1**	– (**5/13**	×	**0**)
-5/108	=	**0**	– (**5/13**	×	**13/108**)
5/108	=	**0**	– (**5/13**	×	**-13/108**)
11/108	=	**1/13**	– (**5/13**	×	**-7/108**)
-11/108	=	**-1/13**	– (**5/13**	×	**7/108**)
5	=	**85/13**	– (**5/13**	×	**4**)

Tablo 8: İkinci (Optimal) Simpleks Çözüm Tablosu

	2	2	M	0	M	0	
C_j	x_1	x_2	A_1	V_1	A_2	V_2	b_j
2 x_1	1	0	13/108	-13/108	-7/108	7/108	4
2 x_2	0	1	-5/108	5/108	11/108	-11/108	5
Z_j	2	2	16/108	-6/108	8/108	-8/108	**18**
C_j- Z_j	0	0	M-16/108	16/108	M-3/108	8/108	

Tablo 8 de, $C_j - Z_j$ satırındaki değerler 0 veya pozitif işaretli olmuştur. Problem minimizasyon problemi olduğundan, optimal tabloya ulaşılmış oldu. Sonuç olarak $x_1 = 4$, $x_2 = 5$ ve Min Z = 18 olur.

2.6.2.4. İki Aşamalı Simpleks Yöntemi

İki aşamalı yöntem doğrudan süreci tanımlayarak optimal çözümü gerçekleştirmesi açısından, büyük M yönteminin bir alternatifidir (Hillier, 1986: 82). İki aşamalı yöntem, yapay değişkene bağlı olarak kullanılan yüksek maliyet faktörü (-M) ile ilgili geçerli bir kavramı ortaya koyar. El ile yapılan bu matrislerin çözümünde (-M) maliyet faktörlerinin indeks satırında hesaplanması yorucu ve çok zordur (Halaç, 1991: 397).

M tekniğinin bir zorluğu da, M sabitine çok büyük bir değer verilerek çözümlenmesi ve bunun da bir hesaplama hatasına sebep olmasında yatmaktadır. Çok büyük değerler verilen değişkenler amaç katsayısında sıfır değere sahip olması nedeniyle istenmeyen sonuçlara neden olmaktadır. Büyük M yöntemi ile, iki aşamalı yöntem daima, optimal çözümün elde edilmesinde tekrarlanan bir sürece sahip olmaktadırlar. Her ikisi de, hesaplama zorluğuna sahiptirler. İki aşamalı yöntem, büyük M metodundan (şayet M yerine geçecek bir rakam varsa), daha iyi sayısal kararlılığa sahiptir ve genellikle bilgisayar programlarında tercih edilmektedir (Hillier and Lierman, 1986: 83).

Büyük M yöntemi ile, iki aşamalı yöntem daima, optimal çözümün elde edilmesinde tekrarlanan bir sürece sahip olmaktadırlar. Her ikisi de, hesaplama zorluğuna sahiptirler. İki aşamalı yöntem, büyük M metodundan (şayet M yerine geçecek bir rakam varsa), daha iyi sayısal kararlılığa sahiptir ve genellikle bilgisayar programlarında tercih edilmektedir (Hillier and Lierman,1986: 83).

İki aşamalı yöntem, birbirini takip eden iki aşamadan meydana gelmektedir. Bunlar;

Aşama 1: Problemin orijinal Z amaç fonksiyonu görmezlikten gelinir ve bütün yapay değişkenlerin toplamının tanımlandığı yeni bir amaç fonksiyonu Z* kullanılır. Birinci aşamada yapay değişkenler çözümden çıkarılmaya ya da değerleri sıfır yapılmaya çalışılır. Şayet bütün yapay değişkenler temel çözümden uzaklaştırılabilir ve bir optimal çözüm bulunabilirse amaç fonksiyonunun değeri sıfır olacaktır. Z* =0 ise, ikinci adıma geçilir. Aksi durumda Z*sıfırdan daha büyük bir değerse, son hesaplamalarda bütün yapay değişkenlerin temel değişkenlerden uzaklaştırılamadıkları anlamını taşır ve problemin optimal bir çözümü yoktur (Acar, 1989: 126).

- Birinci aşama sonunda üç durum söz konusudur.

- Max Z < 0 ise, problemin bir optimal çözümü yoktur. Temelde pozitif olan bazı yapay vektörler mevcut demektir.

- Max Z = 0 ise, problemin bir optimal çözümü vardır. Temelde yapay değişken yoktur. İkinci aşamaya geçilebilir.

- Max Z = 0 ise, problemin bir optimal çözümü vardır, ayrıca temel değişkenlerde bir veya birden fazla yapay değişken olabilir. Bu nedenle sınırlayıcı koşullarda fazlalık bulunabilir (Tulunay, 1987: 441). Temelde sıfır düzeyli bazı yapay vektörler mevcuttur.

Aşama 2: Problem birinci aşamada b ve c şıkkıyla sonuçlanması halinde ikinci aşamaya geçilir. Bu aşamada, bir önceki aşamanın son tablosu ikinci aşamanın ilk tablosunu oluşturacaktır. Değişiklik sadece amaç fonksiyonunun katsayılarında yapılır. Amaç fonksiyonunun değerleri, gerçek değerler verilerek, çözüme devam edilir.

Şimdi yukarıda büyük M yöntemiyle çözülen örnek bir de iki aşamalı simpleks yöntemiyle çözülüp karşılaştırılsın.

2.6.2.5. Simpleks Çözüm Yönteminin Uygulanmasında Karşılaşılan Özel Durumlar

2.6.2.5.1. Dejenerasyon (Bozulma)

Doğrusal programlama problemleri simpleks çözüm yöntemiyle çözümlenirken bozulma ile aşağıdaki şekillerde karşılaşılır.

1. Anahtar sıranın seçiminde karşılaşılan bozulma; Anahtar sıranın seçiminde, çözüm sütunundaki değerlerin anahtar sütundaki değerlere bölümüyle oluşan oranlara bakılıyordu. Bu oranlar içerisinden pozitif en küçük değerli iki oran varsa bir bozulma durumuyla karşılaşılır. Bunu gidermek için temel olmayan değişkenler matrisindeki elemanlar anahtar sütun elemanlarına bölünür. Bu

bölümden elde edilen oranlar içerisinden en küçük elemana denk gelen sıra anahtar sıra olur.

2. Anahtar sütunun seçiminde karşılaşılan bozulma; Anahtar sütun belirlenirken, C_j-Z_j sıra elemanlarına bakılıyordu. Maksimizasyon amaçlı bir problemde bu elemanlardan en büyük olan, minimizasyon amaçlı problemde ise en küçük olan birden fazla değer varsa bir bozulma durumuyla karşılaşılır. Bunu gidermek için çözüm sütunu elemanlarını her bir anahtar sütun elemanlarına bölerek oluşan oranlardan en küçük pozitif değere karşılık gelen değerin bulunduğu sütun anahtar sütun olarak seçilmektedir.

3. Bir veya daha fazla temel değişkenin çözüm değeri sıfır oluyorsa, bu durumdaki çözüme dejenerasyon çözüm denir. Bu durum bazen geçici olabilir. İşlemler esnasında temel değişkenin sıfır olmasına karşın sonuç tablosunda bu durumla karşılaşılmazsa geçici dejenerasyon denir.

4. İşlemler esnasında bir döngü oluşması sonucu bir türlü sonuca ulaşılmaması nedeniyle bir bozulma olabilir. Buna uygulamada az rastlanır. Sonuca ulaşılamadığı için problemin çözümü yoktur (http://www.iyiodev.com).

Örnek (Öztürk, 2007: 176):

Amaç fonksiyonu

Max $Z = 3x_1 + 9x_2$

Kısıtlayıcılar;

$x_1 + 4x_2 \leq 8$
$x_1 + 2x_2 \leq 4$
$x_1, x_2 \geq 0$

Tablo 9: Başlangıç Simpleks Çözüm Tablosu

	3	9	0	0		
C_j	x_1	x_2	S_1	S_2	b_j	
0 S_1	1	**4**	1	0	8	8/4=2
0 S_2	4	1	2	1	4	4/2=2
Z_j	0	0	0	0		
C_j- Z_j	3	9	0	0		

Tablo 9'daki oranlara bakılırsa 8 / 4 =2 ve 4 / 2 =2 çıkmaktadır. Dolayısıyla S_1 ve S_2' nin her ikisi de çıkan değişken olabilir. Burada rastgele bir seçim söz konusu olacağı için S_1 çıkarılıp iterasyona devam edildiğinde, $x_1 = 0$, $x_2 = 2$ ve Max Z = 18 değerleri elde edilir.

2.6.2.5.2. Sınırsız Çözümler

Bazı DP modellerinde, değişkenlerin değerleri herhangi bir kısıt bozulmaksızın sonsuza kadar artırılabilir veya azaltılabilir. Çözüm uzayı en az bir yönde sınırlandırılmamış olabilir. Bu da modelin iyi kurulmadığına işaret eder. Herhangi bir doğrusal programlama problemini simpleks yöntemi ile çözerken çözümün sınırsız olduğunu söyleyebilmek için maksimizasyon probleminde $C_j - Z_j$ satırındaki işleme girecek pozitif değerli (anahtar) sütundaki elemanların hepsi negatif ya da sıfır değerli olmalıdır. Minimizasyon probleminde ise $C_j - Z_j$ satırındaki işleme girecek mutlak değerce en büyük negatif değerli sayı (anahtar), sütundaki tüm elemanlar negatif veya sıfır değerli olursa çözüme sınırsız çözüm denir (Öztürk, 2007: 180).

Örnek:

Amaç fonksiyonu

Max $Z = 2x_1 + x_2$

Kısıtlayıcılar

$x_1 - x_2 \leq 10$
$2x_1 \leq 40$

Pozitiflik Kısıtı

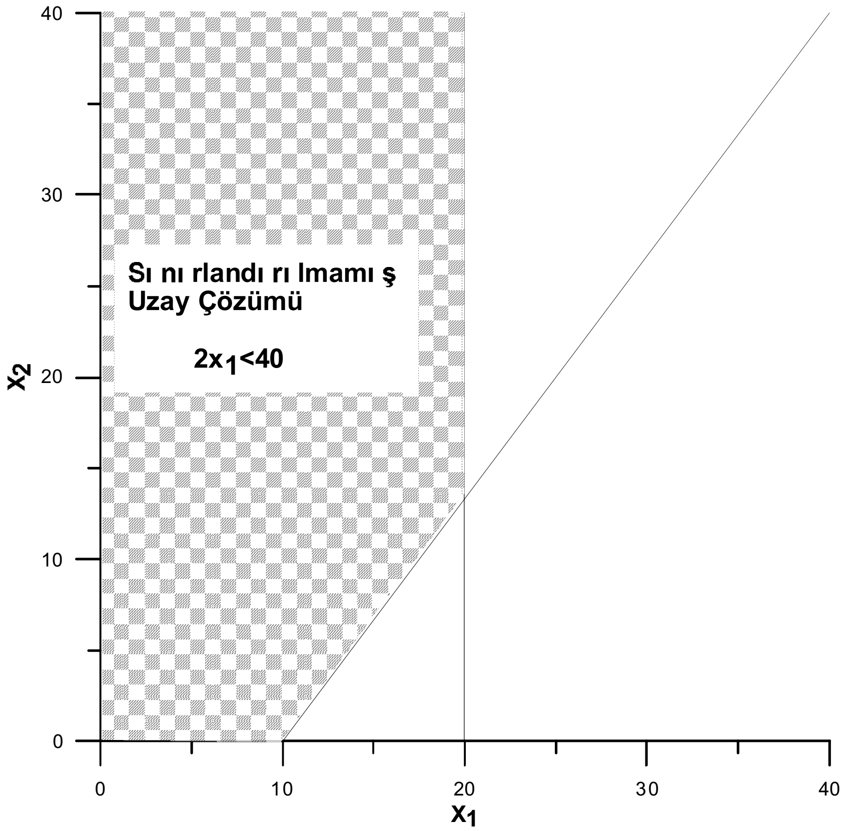
$x_1, x_2 \geq 0$

Tablo 10: Başlangıç Simpleks Çözüm Tablosu

C_j	2 x_1	1 x_2	0 S_1	0 S_2	b_j	
0 S_1	1	-1	1	0	10	8/4=2
0 S_2	2	0	0	1	40	4/2=2
Z_j	0	0	0	0		
C_j- Z_j	2	1	0	0		

x_1 ve x_2 tabana girecek değişkenlerdir. x_1, C_j - Z_j satırında daha büyük katsayıya sahip olduğu için giren değişken olarak seçilir. Ancak x_2'nin altındaki sütunda bulunan tüm katsayılar negatif veya sıfırdır. Bu x_2'nin diğer kısıtları bozmadan sürekli artabileceğini gösterir. x_2'deki bir birimlik artış Z' yi de bir birim artıracaktır. x_2' deki artış sonsuz olduğundan Z deki artış da sonsuz olacaktır. Bu modelin sınırlandırılmış bir çözümü olmadığı söylenebilir. Modele ait grafik çözüm aşağıda verilmektedir (www.ce.yildiz.edu.tr) (grafik 3).

Grafik 3: Sınırlandırılmamış Çözüm



2.6.2.5.3. Seçenekli (Birden Fazla) Optimal Çözüm

Doğrusal programlama probleminin optimal çözümüne yönelik aynı sonucu veren birden fazla değişken olabilir. Diğer değişkenin işleme girmesi optimal çözümü değiştirmiyorsa seçenekli optimal çözüm vardır. Simpleks yöntemde C_j-Z_j sıra elemanlarındaki değerlerden temel olmayan değişkene denk gelen eleman sıfır oluyorsa seçenekli optimal çözüm vardır (http://www.iyiodev.com).

Örnek:

Amaç fonksiyonu

$x_1 + 1.5x_2$

Kısıtlayıcılar

$2x_1 + 3x_2 \leq 12$

$2x_1 + x_2 \leq 8$

Pozitiflik Kısıtı

$x_1, x_2 \geq 0$

Tablo 11: Optimal Simpleks Çözüm Tablosu (Seçenekli Optimal Çözüm)

C_j	1 x_1	1.5 x_2	0 S_1	0 S_2	b_j
1.5 x_2	0.66	1	0.33	0	4
0 S_2	2	0	0	1	6
Z_j	1	1.5	0.5	0	
C_j- Z_j	0	0	-0.5	0	

Temel değişken sütununda bulunmayan değişkenlerden en az bir tanesinin kontrol satır değeri sıfır ise, alternatif optimal bir çözüm daha var demektir. Tablo 11' de görüldüğü gibi seçenekli optimal çözüm var demektir. Bu çözümü bulabilmek için ilgili değişken çözüme alınır. Problemin grafik çözümü grafik 4'te verilmiştir.

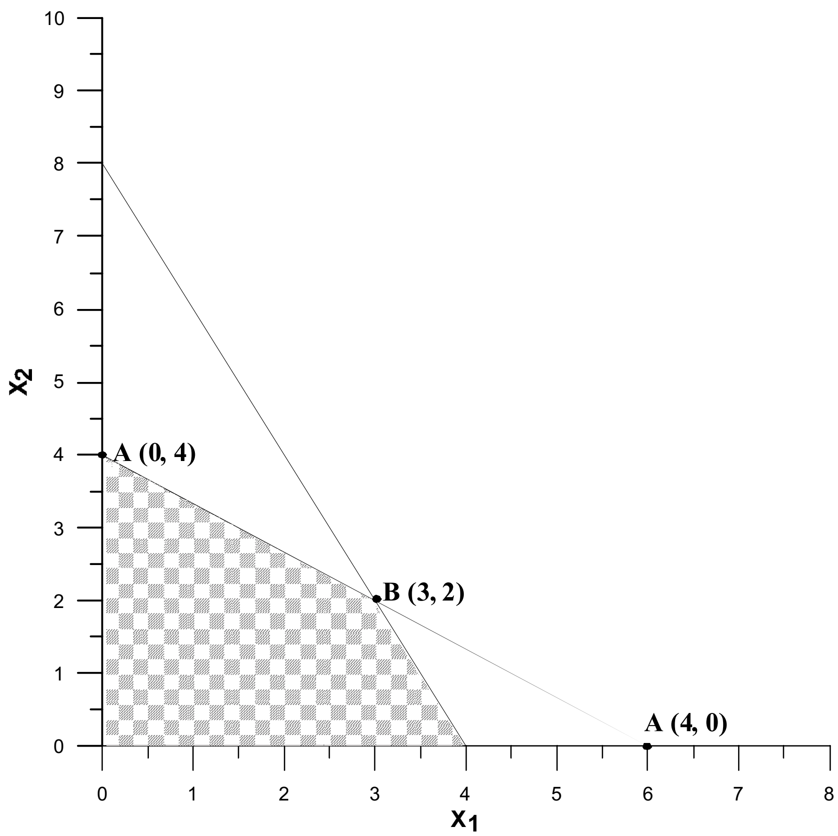
A(0; 4) noktası için; 1(0) + 1.5(4) = **6**

B(3; 2) noktası için; 1(3) + 1.5(2) = **6**

C(4; 0) noktası için; 1(4) + 1.5(0) = 4

Optimal çözüm A ve B noktasıdır (www.business.cu.edu.tr).

Grafik 4: Seçenekli Optimal Çözüm



2.6.2.5.4. Uygun Çözüm Bulunmama

Herhangi bir DP modelinin kısıtları aynı anda sağlanamıyorsa uygun çözüm yok demektir. Böyle bir durum modelin tüm kısıtlarının ≤ yönünde olması durumunda ortaya çıkamaz, zira artık değişkenler uygun çözümü oluşturabilirler. Kısıtların ≥ yönünde olması durumunda ise yapay değişkenler ekleneceği için bunların optimum çözümde sıfır değerini alması beklenir. Modele eklenmiş herhangi bir yapay değişken optimum çözümde negatif değer alıyorsa problem çözümsüzdür. Böyle bir durumla da problem doğru formüle edilmediği zaman karşılaşılır (http://www.iyiodev.com).

Örnek:

Amaç fonksiyonu

Max Z = $5x_1 + 3x_2$

Kısıtlayıcılar

$4x_1 + 2x_2 \leq 8$

$x_1 \geq 4$

$x_2 \geq 6$

Pozitiflik Kısıtı

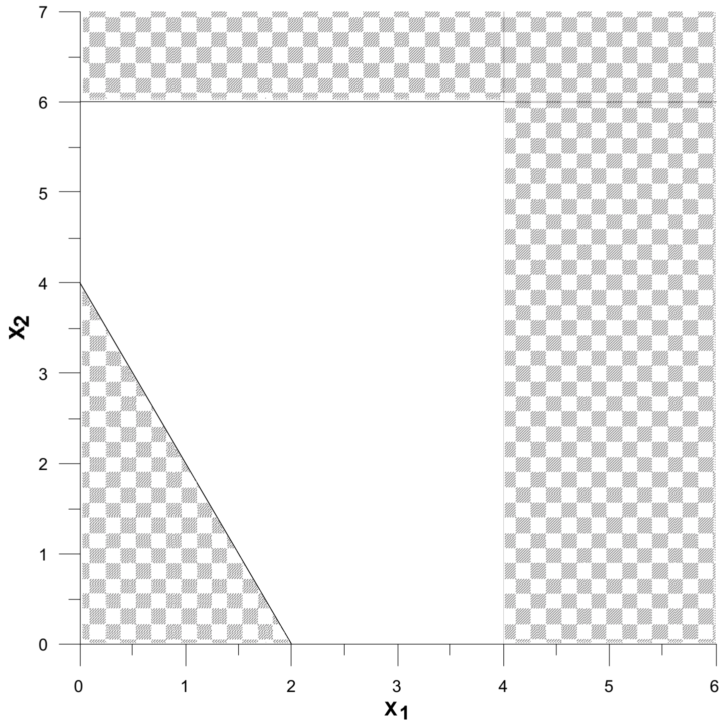
$x_1, x_2 \geq 0$

Tablo 12: Optimal Simpleks Çözüm Tablosu (Uygun Çözüm Bulunmama)

	5	3	0	0	0	-M	-M	
C_j	x_1	x_2	S_1	V_1	V_2	A_1	A_2	b_j
3 x_2	2	1	0.5	0	0	0	0	4
-M A_1	1	0	0	-1	0	1	0	4
-M A_2	-2	0	-0.5	0	-1	0	1	2
Z_j	M+6	3	0.5M+1.5	M	M	-M	-M	12-6M
$C_j - Z_j$	-M-1	0	-0.5M-1.5	0	0	0	0	

Tablo 12'de sonuç optimal gibi gözükmesine rağmen temel değişken sütununda hala A var. Problemin grafik çözümü grafik 5'te verilmiştir (www.business.cu.edu.tr).

Grafik 5: Uygun Çözüm Bulunmama



Grafikte de görüldüğü gibi ortak çözüm alanı olamadığından uygun çözüm yoktur.

2.6.2.5.5. Sınırsız Uygun Bölge Ve Sınırlı Optimal Çözüm

Sınırsız uygun alan olması durumunda, optimal bir çözüm bulunabilir. Bunu anlamak için simpleks tablosundaki herhangi bir sütundaki değerlere bakılır. Bu değerler sıfır ve negatif değerlerden oluşuyorsa sınırsız uygun alan vardır denir. Ancak C_j-Z_j sıra elemanları maksimizasyon ve minimizasyon problemlerine göre istenen değerleri alıyorsa problemin optimal çözümü vardır (http://www.iyiodev.com).

Örnek:

Amaç fonksiyonu

Max $Z = 3x_1 - 2x_2$

Kısıtlayıcılar

$x_1 - x_2 \leq 3$
$x_1 \leq 5$

Pozitiflik Kısıtı

$x_1, x_2 \geq 0$

Tablo 13: Başlangıç Simpleks Çözüm Tablosu (Sınırlı Optimal Çözüm)

C_j	3 x_1	-2 x_2	0 S_1	0 S_2	b_j	
0 S_1	**1**	-1	1	0	3	3
0 S_2	1	0	0	1	5	5
Z_j	0	0	0	0		
C_j- Z_j	3	-2	0	0		

Tablo 14: İkinci (Optimal) Çözüm Tablosu (Sınırlı Optimal Çözüm)

C_j	3 x_1	-2 x_2	0 S_1	0 S_2	b_j
3 x_1	1	0	0	1	5
-2 x_2	0	1	-1	1	2
Z_j	3	-2	2	1	11
C_j- Z_j	0	0	-2	-1	

Başlangıç simpleks çözüm tablosunda olan Tablo 13'teki x_2 sütununa baktığımızda, x_2'nin altında (-1 ve 0) elemanları vardır. Bu da bize uygun alanın sınırsız olduğunu gösterir. Tablo 14'deki çözüm ise sınırlıdır. Yukarıda ele aldığımız ifadenin doğruluğu bir kez daha ortaya çıkmaktadır. Çünkü x_2 sütununun altındaki $C_j - Z_j$ satırı elemanının değeri -2'dir.

2.7. Dualite Ve Duyarlılık Analizi

2.7.1. Dual Problemin Tanımı

Her doğrusal programlama probleminin ilişkili olduğu bir ikiz problemi vardır. Herhangi bir doğrusal programlama problemi veya modeli primal veya asıl problem veya model olarak adlandırılırken diğerine yani ikizine dual veya ikilik adı verilir. Problemlerin birbirine olan yakın ilişkisi göz önüne alındığında hangi problemin primal hangisinin dual olarak adlandırıldığında fazla bir şey fark ettirmez. Çünkü birinin optimal çözümü aynı zamanda ötekinin optimal çözümüdür.

Verilen bir doğrusal programlama probleminin dualini alırken, verilen bu doğrusal programlama problemine primal diyoruz. Eğer primal bir maksimum problem ise dualini minimum problem veya bunun tersi olur. Maksimum problemin değişkenlerini z, x_1, x_2, … x_n ve minimum problemin değişkenlerini de y_0, y_1, y_2, …, y_n olarak tanımlanır (www.özyazilim.com).

Şimdi normal Max problemlerin (kanonik şekildeki) dualini nasıl bulabileceğimizi açıklayalım. Normal Max problemlerin amaç fonksiyonu maksimum, tüm kısıtlayıcıları $\leq$ ve tüm değişkenleri ≥ 0' dır (Öztürk, 2007: 206).

Primal Problem

$$\text{Maksimum } Z = \sum_{j=1}^{n} c_j x_j$$

Kısıtlayıcılar

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1,2,\ldots,m$$

$j= 1,2,\ldots,n$ ve $x_j \geq 0$

Dual Problem

$$\text{Minimum } y_0 = \sum_{i=0}^{m} b_i y_i$$

Kısıtlayıcılar

$$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad j = 1,2, \dots, n$$

$i = 1,2,,..,m$ ve $y_i \geq 0$

Kanonikal şekildeki primal problemin dual problem haline dönüştürüldüğünde ortaya çıkan ilişkileri şu şekilde özetlenebilir:

1. Primal problem bir maksimizasyon problemi ise onun dual problemi minimizasyondur.

2. Kanonikal şekildeki maksimizasyon primal problemin kısıtlayıcılarının yönü ($\leq$) ise minimizasyon dual problemde kısıtlayıcıların yönü ($\geq$) dür.

3. Minimizasyon amaçlı dual problemin dual değişkenlerinin amaç katsayısı olarak primal problem kısıtlayıcı denklemlerinin sağ tarafındaki kaynakları (b_1, b_2.....b_m) kullanır.

4. Dual kısıtlayıcılarının sağ tarafındaki sabitler, primal problemin birim amaç katsayılarıdır.

5. Kısıtlayıcı denklemlerin katsayıları aynı kalırken sadece dönüşüme uğrarlar. Yani primal problemin kısıtlayıcılarının sol tarafındaki satır katsayıları, dual problemin kısıtlayıcılarının sütun katsayıları olur.

6. Dual değişkenlerinin sayısı primal problemin kısıtlayıcı denklem sayısına eşittir.
7. Dual problemin kısıtlayıcı sayısı ise primal problemin değişken sayısına eşittir. Yani eğer maksimizasyon amaçlı primal problemin (n) sayıda değişkeni ve (m) sayıda kısıtlayıcısı varsa, dual problemin (m) değişkeni ve (n) sayıda da kısıtlayıcısı vardır.

8. Her iki problemde yer alan değişkenler negatif değildir (www.özyazilim.com).

2.7.2. Dualitenin Ekonomik Yorumu

Primal probleme kaynak dağıtım modeli olarak bakıldığında, n ekonomik faaliyete ve m adet kaynağa sahip olduğu görülür. C_j 'ler j faaliyetindeki birim karı gösterir. Maksimum kullanımı b_i olan i kaynağı, j faaliyetinin birimi başına a_{ij} birimlik bir hızla kullanılmaktadır. Optimumluk(maksimum getiri) kaynakların tamamının kullanımında gerçekleşir, bu durumda girdi(kaynakların değeri), çıktıya(kar) eşittir. Girdi, çıktıyı aştığında sistem optimum değildir. Denge *girdi=çıktı* olduğunda sağlanır.

w_i dual değişkenlerinin i kaynağının birim başına değerini gösterdiğini söyleyebiliriz. Diğer bir anlamıyla marjinal karları (veya maliyetleri) gösterirler.

2.7.3. Duyarlılık Analizi

2.7.3.1. Duyarlılık Analizinin Tanımı

Doğrusal programlama problemlerinde, problemin bileşenlerinde meydana gelecek değişmelerin optimal çözüme etkilerinin neler olduğu duyarlılık analizi aracılığı ile belirlenebilir.

Duyarlılık analizinde, doğrusal programlama probleminde yer alan katsayılar ve sabit değerlerden birinin veya bir kaçının değişmesi ya da aralarındaki oranın değişikliğe uğraması halinde, optimum planda meydana gelecek değişmeler sistematik bir biçimde araştırılır.

Doğrusal programlama problemlerinde ulaşılan optimal çözüm problemin katsayıları sabit kaldığı sürece geçerlidir. Problemin katsayılarının kısıtlayıcı kaynak miktarlarının değişmesi ve yeni bir faaliyetin eklenmesi halinde daha önce elde edilen çözüm optimallikten çıkacaktır (Öztürk, 2007: 106). Duyarlılık analizleri bu değişkenlerin, optimal çözüm sonucunda bir değişikliğe yok açmadan alt ve üst değişkenleri bulmaya yarar.

İndirgenmiş Maliyet

Herhangi bir temel dışı değişkenin indirgenmiş maliyeti, değişkenin temel değişken olması (DP'nin en iyi çözümüne girmesi) için amaç fonksiyon katsayısında yapılacak iyileştirme miktarıdır.

Eğer bir x_i temel dışı değişkenin amaç fonksiyon amaç fonksiyon katsayısı indirgenmiş maliyet kadar iyileştirilirse, DP'nin bir tek en iyi çözümü olmaz; alternetif çözümler vardır, x_i söz konusu çözümlerden en az birinde temel değişken; ene az birinde ise temel dışı değişken konumundadır.

Eğer x_i temel dışı değişkenin amaç fonksiyon katsayısı indirgenmiş maliyetten daha fazla iyileştirilirse, yeni DP'nin tek bir en iyi çözümüne ulaşılır ve bu çözümde x_i temel değişken olur ($x_i > 0$).

Gölge Fiyat

DP modelinin i. Kısıtının gölge fiyatı, söz konusu kısıtın sağ taraf değerinin 1 birim çoğalması durumunda, en iyi amaç fonksiyon değerinin ne kadar iyileştiğini (en büyükleme sorununda ne kadar arttığını, en küçükleme sorununda ne kadar azaldığını) gösterir.

Bu tanım sadece değişimden önceki çözümün değişimden sonra da aynı kalması durumunda geçerlidir.

Bir $\geq$ kısıtın gölge fiyatı her zaman 0 ya da 0'dan küçük; bir $\leq$ kısıtın gölge fiyatı ise her zaman 0 ya da 0'dan büyük olacaktır.

2.7.3.2. Duyarlılık Analizinin Uygulama Şekilleri

Doğrusal programlama problemi için optimal çözüm belirlendikten sonra aşağıda belirtilen değişimlerin optimal çözümde yaratacağı etkileri yöneticiler duyarlılık analizleri ile belirleyebilirler. Söz konusu değişmeler şunlardır;

1. Amaç fonksiyonu katsayılarının değişimi (C_j)

2. Üretim faktörleri kapasitelerinin değişimi (b_j)

3. Teknoloji katsayılarının değişimi (a_{ij})

4. Yeni bir kısıtlayıcı faktörün modele dahil edilmesi (Altay, 1990: 37).

2.7.3.3. Duyarlılık Analizi ve Gölge Fiyatlar

Duyarlılık analizi bir bakıma hareket halinde bulunan ve üretim kaynaklarının miktar ve birleşimleri değişen modellerde optimal kaynak geliştirilmesinin şartlarını ortaya koyar. Buna karşılık gölge fiyatlar her şeyin sabit bulunduğu statik bir model içinde ve bir defalık çözüme mahsus olmak üzere üretim kaynaklarının sosyal alternatif maliyeti, o kaynağın modelin tümü içindeki önemleri hakkında bir fikir verir. Kaynakların nispi önemleri hakkında etraflı ve geniş bir alanı kaplayan bilgi vermez. Gölge fiyatların, ilk bakışta görülemeyen ve geleneksel metotlarla çözülüp bulunamayan sonuçları verme bakımından büyük bir üstünlüğe sahip bulunduğu şüphesizdir. Buna rağmen kantitatif politika yapıcıları için yeterli bilgiyi verebilme niteliğinden de uzaktır. Zira sadece bir çözüme ve belli kaynak bileşimine göre kaynakların prodüktivitesini göstermektedir.

Duyarlılık analizi, değişik kaynak karışımı karşısında kaynakların birbirine göre sahip olabilecekleri nispi önemlerdeki değişmeler üzerine ışık tutmakta ve hangi kaynaklar demetinin en elverişli, ene ekonomik olacağını göstermektedir.

Başka bir deyimle duyarlılık analizi optimallerin optimalini seçebilme imkanını vererek kantitatif politika yapıcılarına en uygun yolu göstermektedir (Kılıçbay, 1970: 512).

Gölge fiyatlar bir çözümün sonucudur. Duyarlılık analizi ise, çok sayıda ve önceden hazırlanmış kaynaklara göre yapılmış çözümlerden doğmaktadır. Bu sebeple duyarlılık analizinin kavram alanı çok geniştir. Duyarlılık analizi kantitatif iktisat politikasının yapımı için gerekli bilgiyi sağlaması bakımından dikkat çekicidir. Bilhassa gelişmekte bulunan ekonomiler için kaynakların uzunca bir zaman dilimi içinde optimal dağılımında fiyat kontrolünde ve teknolojik değişmelere verilecek yönün tayininde duyarlılık analizi yardımcı olur (Altay, 1990: 41).

ÜÇÜNCÜ BÖLÜM

3. HEDEF PROGRAMLAMA

3.1. Genel Açıklamalar

Günümüz işletmelerinin yatırım, finansman, satış, reklâm ve benzeri özellikteki hedefleri, bir bütün olarak ekonomik kuram içinde değerlendirilmelidir. İşletmenin bu hedeflere ulaşabilmesi, ancak bu hedefler arasındaki uyumun sağlanması ve problemlerin etkin yöntemlerle çözümlenmesiyle mümkündür.

İşletmelerin hedeflerine ulaşabilmesi, gün geçtikçe karmaşıklaşan ve çoğalan problemleri çözümlemelerine bağlıdır. Bu problemlerin çözümünde yöneylem araştırması tekniklerinden biri olan doğrusal programlama tekniği, yalnız başına etkin bir çözüm sunamamaktadır.

Doğrusal programlama, sadece bir tek yönde amacı olan en büyükleme veya en küçükleme problemlerine etkin çözümler üretebilir; ancak çelişen hedefler söz konusu olduğu zaman etkinliğini yitirir. Bunun sonucunda geliştirilen çok amaçlı karar verme teknikleri, birbiriyle çelişen amaçlan göz önüne alarak ideal çözüme en yakın çözümü bulmaya çalışır.

Hedef programlama yaklaşımında her bir amaç için, bir amaç fonksiyonu formüle edilir ve bu amaçlara ulaşamamaktan (kaçırmaktan) doğan toplam cezayı minimum kılan bir çözüm aranır. Bu toplam ceza, amaç fonksiyonlarının her birinin hedeflerinden sapmaların ağırlıklı toplamını ifade eder.

Bir karar verici, matematiksel optimizasyon modellerini kullanırken amaç fonksiyonlarını optimum kılan pek çok seçenekli çözümlerin birini seçmek durumu ile karşılaşabilir. Karar ikinci derecede, üçüncü derecede veya daha yüksek dereceli olabilir.

3.2. Hedef Programlamanın Tanımı ve Tarihi Gelişimi

Hedef programlama, kısıtlı optimizasyon tekniği olan doğrusal programlamanın özel bir durumudur, mümkün olduğunca çok amacı aynı anda sağlayan bir çözüm bulmaktır.

Hedef programlama, çok amaçlı karar verme problemlerini çözmek için karar vericilere doyurucu bir çözüm kümesini bulmayı sağlayan önemli bir tekniktir. Karar vericiler için bu tekniğin en önemli özelliği her bir nitelendirmeye doyurucu bir hedef değerinin atanabilmesidir.

Hedef Programlama, çok alt hedefi bulunan çok hedefli problemler gibi çok alt hedefli tek bir hedefi amaçlayan karar problemlerin çözümünde kullanılan doğrusal programlamanın özel bir genişletilmiş halidir (Wise, 2000: 168). Bir anlamda doğrusal programlamanın amaç fonksiyonu tek boyutlu (karı maksimum kılma veya maliyeti minimum kılma) iken Hedef Programlama ise çok boyut içinde çoklu hedeflere erişmede kullanılabilen bir tekniktir. Burada anlatılması gereken Hedef Programlamanın amaç fonksiyonun boyutsal bir kısıtlamasının olmadığıdır.

Hedef Programlama, verilen kısıtlayıcılar altında amaç kriterini doğrudan maksimum veya minimum kılmaktan ziyade hedeflerin kendi içindeki sapmaları minimum kılmaya odaklanan bir tekniktir.

Hedef programlama, birden çok hedefe aynı anda ulaşılmak istenildiğinde ve bu hedefler birbiriyle çeliştiğinde yararlı olan bir çözüm yöntemidir. Hedef programlamadaki asıl amaç ulaşılmak istenen hedef değerlerinden sapmaların minimum yapılmasıdır.

Hedef Programlamanın önemini ve işlerliğini kavrayabilmek için öncelikle bu programlamanın varsayımlarının ve terimlerin bilinmesi gereklidir. Çünkü varsayımlar bu programlamaya gelebilecek eleştirileri ortadan kaldırabileceği gibi terimlerin açıklanması da okuyuculara bu konunun daha iyi anlaşılmasını sağlayacaktır.

Hedef programlamanın ortaya çıkarılışı, 1955 yılında yapılan çalışmalara dayanır. Charnes ve Cooper görünürde doğrusal programlamayla ilgisi olmayan bir problemle karsılaşmışlardır. Bu problemi çözmek için doğrusal programlamanın değişik bir biçimi olan ve sınırlandırılmış regresyon olarak adlandırdıkları bir yaklaşım ortaya koymuşlardır. Daha sonra 1961'de yaptıkları yayında çok amaçlı doğrusal modelleri de içeren sınırlandırılmış regresyonun daha geniş bir biçimini tanıtmışlardır. Bu yaklaşım hedef programlama olarak adlandırılmış ve günümüz çalışmalarında da çok sık kullanılan bir duruma gelmiştir (Ignizio, 1985: 187).

Sonrasında yeni ufuk açan fikir çalışmaları Lee, Iqnizio ve Cavalier ve Romeo tarafından yapılmıştır. 1965'de Iqnizio, birden fazla durumda amaçların önceliklerinin dikkate alınması gerektiğini ve aynı önceliğe ait sapma değişkenlerinin ağırlıklandırılmasını ortaya atmıştır (\http:\\www.kouemk.com/makale/).

Hedef Programlamanın ilk kullanımı bir mühendislik uygulaması olup, Ignizio'nun 1962'dekinin yanında, uzaya ilk ayak basan Apollo uzay mekiğinin fırlatılmasında kullanılan Satürn V'nin ikinci safhasında kullanılacak antenlerin dizaynı ve yerleştirilmesi üzerine idi (http:\\en.wikipedia.org/wiki/Goal.Programming).

Hedef programlama, 1970'lerin başlarında önem kazanmaya başlamış ve 1980'lerde de bilgisayar sistemlerinin gelişmesine paralel olarak neredeyse tüm çok amaçlı karar verme problemlerinde etkin çözümü bulmakta kullanılmıştır.

Hedef programlama tekniğinin temel düşüncesi, gerçekte çok amaçlı olan bir Problemi tek amaçlı bir probleme dönüştürerek çözüm aramaktır. Karar vericiden her bir amaç için ulaşılmak istenen bir hedef değer belirlemesi istenir. Hedef programlama ile tüm çelişen amaçları optimum kılan tek bir çözüm bulmak imkansız olabilir, amaçların birine ulaşılabilir diğerlerine ulaşılmayabilir. Bu nedenle, bu tür modellerle ilgili problemlerin hedef programlama ile elde edilen çözüm değerlerine "etkin çözüm" adı verilmektedir.

3.3. Hedef Programlama İle İlgili Geçmişte Yapılmış Çalışmalar

Hedef programlamayla ilgili son yıllarda yapılan bazı çalışmalar söyle özetlenebilir.

Dağdeviren, Akay ve Kurt (2004), iş değerlendirme ve hedef programlama yöntemlerinin genel yapısını incelemişlerdir. Çalışmalarında faktör derece puanlarının belirlenmesinde hedef programlama yönteminin kullanılabileceğini göstermişlerdir.

Lee, Chong, Park (2003), proje çizimleri için bir yöntem geliştirmişlerdir. Projeler için yapılan ilk çizimlerle, bilgisayarla elde edilen düzeltilmiş çizimler arasında uyumsuzluk olduğunu belirtmişlerdir. Bu uyumsuzluktan dolayı elde edilen son çizimin, proje için yapılan ilk çizimde kullanılmasında sorun olduğunu göstermişlerdir. Araştırmacılar bu sorunu önlemek için hedef programlamayı kullanarak ilk çizimle son çizim arasında dönüşüm sağlayan basit bir model geliştirmişlerdir.

Ghosh, Sharma ve Mattison (2003), Hindistan'da pirinç üretiminde verimliliği artırmak için gerekli olan maddelerin uygun oranlarda karıştırılmasıyla bulunacak optimal gübrenin elde edilmesi üzerinde çalışmışlardır. Çalışmalarında hedef programlama ve oyun teorisini kullanmışlardır.

Linares ve Romero (2002), gelecekte elde edilebilecek enerji kapasitesini hedef programlamayı kullanılarak tahmin etmeye çalışmışlardır. İspanya'da 2020'deki elektrik üretimi için sosyal gruplar tarafından belirlenen amaçları sağlayan, farklı yakıtlardan ve farklı teknolojilerden yararlanarak elde edilecek enerji kapasitesini belirlemeye çalışmışlardır. Sosyal gruplar tarafından belirlenen amaçlar öncelik sırasına göre sıralanmıştır. Çalışmada en önemli amaç elektrik üretim maliyetinin minimum olmasıdır. Diğer amaçlar ise zararlı gazların atmosfere yayılmasının ve radyoaktif atık miktarlarının minimum yapılması olarak belirlenmiştir.

Karsak, Sözer ve Alptekin (2002), kalite fonksiyonunda müşteri memnuniyetinin artması için müşteriye yönelik ürünlerin geliştirilmesi ve yeni ürünlerin yapılması için bir planlama üzerinde çalışmışlardır. Çalışmalarında analitik ağ süreci ve 0-1 hedef programlama yöntemlerini kullanmışlardır. Karar yaklaşımının amacı müşteri ihtiyaçlarını ve ürün teknik ihtiyaçlarının birbirine bağımlılığını ve bunların iç bağımlılıklarını göstermektir.

Hajidimitriou ve Georgiou (2002), uluslararası ortak girişimlerde ortak seçimini incelemişlerdir. Uluslararası ortak girişimlerin oluşumunda uluslararası büyüme modelleri kullanılır. Uluslararası ortak girişimlerin başarılı olabilmesindeki en önemli faktör uygun ortaklığın kurulmasıdır. Daha önce ortak seçimi için gerekli ölçütler göz önüne alınarak birçok çalışma yapılmıştır. Ancak uygun kantitatif bir model geliştirilmemiştir.

Chang (2002), n terimli parçalı bir doğrusal fonksiyonu çözmek için değişik bir hedef programlama önermiştir. Önerilen yöntemde sadece bir tane toplam kısıtta ihtiyaç vardır. Parçalı doğrusal fonksiyonların çözümü için hedef programlama yöntemine uygun doğrusal kısıtlar eklenmesi gerektiği gösterilmiştir. Bu yöntemin Charnes, Cooper ve Lee tarafından önerilen yöntemlerden daha etkili olduğu belirtilmiştir.

Yamada, Tanino ve Inuiguchi, para piyasalarında portföy optimizasyon problemini çok amaçlı hedef programlama ile çözmüşlerdir. Fon yöneticileri için etkili küme üzerinden tutanak fiyatlarını minimize eden çözümün bulunmasına çalışmışlardır. (Aouni, Kettani, 2001).

Parra, Terol ve Uria, portföy seçimi için yeni bir bulanık hedef programlama formülasyonu önermişlerdir. Modelin aynı anda birbirleriyle çelişen kar, risk ve likidite amaçları vardır. Bu amaçlarla ilişkilendirilen hedeflerin değerleri bulanık olarak düşünülmüştür. Formülasyonda yatırımcının tercihleri hesaba katılmamıştır (Aouni, Kettani, 2001).

Caballero, Galache, Gomez, Molina ve Torrico, Malaga üniversitesinde finansal kaynakların etkili bir şekilde tahsis edilmesi üzerine çalışmışlardır. Karar verme sürecinin amacı en önemli öğrenim ihtiyaçlarını karşılamak, araştırma çıktılarını maksimum yapmak ve personelin kalitesini artırmaktır. Araştırmacılar elde edilen sonuçların ispanya'nın diğer üniversitelerinde de uygulanabilineceğini belirtmişlerdir (Aouni, Kettani, 2001).

Abdelaziz ve Mejri, Tunus'daki bir baraj isletmesi için hedef programlama modeli kurarak Tunus'un kuzeyindeki su kaynakları problemleri üzerinde çalışmışlardır. Sulama ve içme suyu talebinin, tuzluluğun ve pompalama fiyatının minimizasyonu hedeflerine ulaşmak için farklı barajlardan, uygun akışların bulunmasına çalışmışlardır. Problemin çözümü için stokastik hedef programlama modelini kullanmışlardır (Aouni, Kettani, 2001).

Chen ve Tsai (2001), farklı önem düzeylerine ve mutlak önceliklere sahip bulanık hedef programlama üzerinde çalışmışlardır. Bu çalışmada bulanık amaçlı çalışmalardan farklı olarak, bulanık amaçların toplam basarı derecesini maksimum yapan toplam bir model önerilmiştir. Toplam bir model önermelerinin nedeni, her bir bulanık amacın başarı düzeyinin, kendinden önceki hedefe ulaşılması zor olduğunda azalmasıdır. Elde edilen çözüm sonuçlarında, hem mutlak öncelik yapısına hem de toplam maksimum başarı derecesine ulaşıldığı görülmüştür.

Kim ve Emery (2000), Woodward yönetim şirketlerinde uçak kontrol grupları için bir projeyi ele almışlardır. Potansiyel büyüme olarak tanımlanan motor kontrol sistemlerindeki özel bir parça üzerine 0-1 hedef programlama modelini uyarlamışlardır. Woodward için bir pazar araştırması yapılmış ve incelemek için birkaç potansiyel motor programı seçilmiştir. Projenin amacı, sınırlı kaynaklar olan sermaye, personel ve makine üzerinden en avantajlı motor programını seçmektir. Hedef programlama modeli ile maksimum kâra ulatmaya, makine islem planlarını geliştirmeye ve personel ihtiyaçlarını tahmin etmeye çalışmışlardır.

Lee ve Kim (2000), bilgi sistemlerinde proje seçimini incelemişlerdir. Bilgi sistemlerindeki seçim işleminin mevcut yöntemlerdeki aday projelere ve kriterler arasındaki bağımlılığa dikkat etmeden yapıldığını açıklamışlardır. Lee ve Kim, bağımlılığın göz önüne alınmasıyla organizasyonun elde edeceği yararın daha yüksek ve getireceği kârın daha çok olacağını belirtmişlerdir. Bu düşünceyle proje problemlerini değerlendirirken, kriterlere ve projeler arasındaki kaynak, yarar ve teknik bağımlığa dikkat etmişlerdir. Çalışmalarında analitik ağ sürecini ve 0-1 hedef programlama yöntemlerini kullanarak bilgi sistemleri için en iyi proje seçimini yapmışlardır.

Bal (1999), doğrusal hedef programlama modelini kullanarak basit ve etkili bir alternatif üçlü sınıflandırmanın yapılabileceğini göstermiştir. Çalışmada gelişmişlik düzeyi gelişmiş, gelişmekte olan ve gelişmemiş ülke bakımından gruplanan onar ülke rastgele alınmıştır. Üç gruplu sınıflandırma problemi hedef programlamaya göre tek bir optimizasyon modeli ile ifade edilmiş, sınıflandırma fonksiyonu ve çoklu ayırma değerleri bulunarak ülkelerin uygun gruplara atanması sağlanmıştır.

Sueyoshi (1999), veri zarflama analizi ile diskriminant analizi arasındaki benzerlikleri ve farklılıkları hedef programlama yardımıyla incelemiştir. Yeni bir diskriminat analiz modeli önermiştir. Önerilen bu model hem örnek bir veri kümesine hem de bir Japon bankasındaki gerçek verilere uygulanmıştır. Elde edilen sonuçlar ışığında yeni model, diğer diskriminat analizleri ile karşılaştırılarak doğrulanmıştır.

Ali, Blanco ve Buclatin (1998), mutlak öncelikli hedef ağ programlama problemlerinin çözümü için özel bir ağ süreci üzerinde çalışmışlardır. Hedef ağ programlama problemlerinin özel çözümleri için ağ simpleks yönteminin kurallarında değişiklik yapılması gerektiği gösterilmiştir. Uygulama olarak deniz kuvvetlerindeki personelin işlere atanması problemi incelenmistir.

Powell ve Premachandra (1998), doğrusal olmayan hedef programlama modeline göre portföy seçimi üzerinde çalışmışlardır. Çeşitli yardım kurumlarının yatırım kararlarında portföy müdürlerinin karşılaştıkları çelişen yatırım amaçları, risk ve hedef kısıtları, kurum sorumlulukları, fon yöneticilerinin kısıtları, kurumdaki para akışı ve ihtiyaçlar düşünülerek en iyi portföy seçimi yapılmıştır.

3.4. Hedef Programlamanın Varsayımları

Bu çalışmada ele alınacak Hedef programlama, “Doğrusal Hedef Programlama” modelidir. Dolayısıyla Hedef Programlamanın alışılagelmiş doğrusal programlamanın; toplanabilirlik, bölünebilirlik, oransallık, belirlilik varsayımlarına hedeflere iliksin önceliklerin karar verici tarafından belirlenmesi varsayımı da eklenebilir. Ayrıca, programda yer alan tüm değişkenlerin pozitif olma koşulunu da aranması gerektiğini belirtilmelidir (Öztürk, 2005: 36 -38).

3.4.1. Doğrusallık Varsayımı

Bu varsayım girdiler ile çıktılar arasında aynı yönlü bir ilişkinin olduğunu gösterir.

Girdiler artarken/azalırken aynı oranda çıktılarda artar/azalır.

3.4.2. Toplanabilirlik Varsayımı

Çeşitli faaliyetler tarafından kullanılan kaynakların toplam kullanımı ve elde edilen toplam katkı, her bir faaliyet tarafından ayrı ayrı kullanılan kaynakların toplamı ve bunların ayrı ayrı yarattıkları katkıların toplamına eşittir.

3.4.3. Sınırlılık Varsayımı

Amaç, sınırlı kaynakların optimal dağılımını sağlamaktır. Problemin çözümünde kullanılacak olan kaynaklar sonludur. Bu nedenle probleme giren kaynaklar kısıtlanır.

3.4.4. Negatif Olmama Varsayımı

Hedef programlama yöntemi, ancak modelde kullanılan değişkenler pozitif değerler aldığı zaman kullanılabilir. Modeldeki tüm değişkenler yani karar ve sapma değişkenlerinin değerleri sıfır veya sıfırdan büyük olmalıdır. Hedef programlama modelinde yer alan bir değişken negatif değer alırsa bu değişkeni ancak negatif olmayan iki değişkenin farkı olarak yazılır. Çözümde bu farkın oluşturduğu yeni değişken kullanılır. Çözüm sonucunda bulunan değer yerine konularak değişkenin orjinal değeri bulunur.

1. Amaçlara Öncelik Verilmesi Varsayımı: Hedef programlama modelinde her bir amaca veya amaç grubuna belli bir öncelik verilir. İlk amaç P1 ile gösterilir. Daha sonraki amaçlar da sırasıyla P_2 , P_3 , ... , P_K öncelikleri ile tanımlanır (Ignizio, 1976: 5).

Kurulacak olan bir hedef programlaması modeli yukarıda belirtilen bu varsayımları sağlamalıdır.

3.5. Hedef Programlamanın İlkeleri ve Temel Kavramlar

- Hedef programlama her bir amaç bir hedef olarak kabul edilir.

- Hedef programlamada hedeflerin gerçekleştirilmesinde öncelikler dikkate alınır. Önce birinci öncelik düzeyindeki hedefler daha sonra ikinci öncelik düzeyindeki hedefler gerçekleştirilir. Sıra atlamadan bütün hedefler tamamlanana kadar devam edilir.

- d_i^- i. hedefin altında kalınması durumunu, d_i^+ hedefin aşılması durumunu gösterir.

- Hedef düzeyleri dikkate alınarak hedeflerden toplam sapma minimize edilmeye çalışılır. Öncelikle birinci öncelikli hedefler için problemin çözümü belirlenir. Daha sonra bu çözümü ihmal etmeyen ikinci düzey hedeflere ait çözüm belirlenir. Aynı şekilde diğer hedefleri olabildiğince sağlayan ve önceki hedefleri ihmal etmeyen çözümler belirlenir (Aslan – Öztürk, 2009: 55).

Hedef programlamada kullanılan özel terimler ve kavramlar aşağıda açıklanmıştır.

Amaç Fonksiyonu

Karar vericinin kontrolünde, kullanıcı tarafından belirlenen niteliği temsil eder. Karar verici tarafından belirlenen amaç fonksiyonu $Z_j(x)$ terimi ile gösterilir. Ayrıca Hedef Programlama geliştirilen amaç fonksiyonun yapısına göre değişir. Örneğin; karı maksimum yapmak, personel değişimini minimum yapmak bir karar vericinin amaçları olabilir.

Hedef

Karar veren kişinin ortaya koyduğu hedefin sayısal değeri olup genellikle b_j veya g_i ile gösterilir. Ulaşılmak istenen noktanın rakamla gösterilmiş halidir. Örneğin, müşterinin hizmet alma süresinin minimizasyonu bir amaç iken, bu sürenin en fazla 2 dakika olması bir hedeftir. Hedef, amacın gerçekleşmesinde, belirli sınırlamalarla amacı somutlaştırır.

Karar Değişkenleri

Kısıtlayıcılar, karar değişkenlerinin birbirleriyle olan ilişkilerinin matematiksel olarak ifade edilmesini sağlar. Yapısal kısıt (real constraint) ve hedef kısıtı (goal constraint) olmak üzere iki çeşit kısıt vardır. Yapısal kısıt; sistem kısıtı olarak da ifade edilir. Karar değişkenleri üzerindeki kısıtlamalardır. Doğrusal programlamadaki kısıtlara karşılık gelir. Değişmesine izin verilmeyen mutlak kısıtın gerçekleştirilmesi önceliklidir.

Hedef kısıtı ise sağ taraf sabitleriyle açıklanan kısıt kümesidir. Sistem kısıtı sağlandıktan sonra, hedef kısıtının sağlanma süresi baslar. Bu kısıt karar verici tarafindan belirlenen önem derecelerine göre sıralanır.

Sağ Taraf Sabitleri

Modelde yer alan kısıtlayıcılardaki eşitlik veya eşitsizliklerin sağ tarafında bulunan kaynak miktarını tanımlar. Bu değerler karar vericinin ulaşmayı hedeflediği değerlerdir ve b_i ile gösterilir.

Sapma Değişkenleri

Hedeflenen başarı ile gerçekleşen başarı arasındaki fark bize sapmayı gösterir. Bir başka deyişle sapma değişkeni, hedef kısıtındaki sağ taraf sabitinden değişen miktarı gösterir. Sapmanın sıfır olması bize hedefin tam anlamıyla sağlandığı bilgisini verir. Literatürde pozitif ve negatif olmak üzere iki sapma sekli vardır. Pozitif sapma sekli p_i veya d_i^+, negatif sapma p_i veya d_i^- ile gösterilir. Bu değişkenler, doğrusal programlama modelindeki aylak değişkenler ile aynı gibi düşünülebilir.

Ağırlıkların Tanımlanması

Modelde aynı önem derecesinde hedefler var ise, önceliklerin belirlenmesine bağlı olarak ağırlıklar tanımlanır. Bu ağılıklar tamamen karar vericinin isteğine göre tanımlanır. Bu nedenle sübjektiflik özelliğine sahiptir. Diferansiyel ağırlık, k'ıncı seviyede i'inci hedeften oluşan sapmaya ilişkin matematiksel ağırlık olarak ifade edilir. w_{ki} ile gösterilir.

Teknolojik Katsayılar

x_j karar değişkenlerinin oluşumunda b_i kaynak miktarından birim başına kullanım değerini açıklayan sayısal değerdir. Teknolojik katsayılar a_{ij} ile gösterilir.

Öncelik Düzeyi

Öncelik düzeyi kullanıcı tarafından amaç fonksiyonları için belirlenen önem sırasını ifade eder. Hedeflerin önem derecesini yansıtan değere "hedeflerin öncelik düzeyi" adı verilir. Amaç fonksiyonunda k'ıncı öncelik düzeyindeki hedef p_k (k =1,2,3,..., K) ile gösterilir. k, modeldeki hedeflerin sayısıdır, p_k,h yani "öncelikli üstünlük faktörü" hedeflerin düzenli bir şekilde yapılanmasını sağlayan bir sıralama sistemidir. İlk önce karar verici tarafından belirlenen birincil öncelikli hedefin karşılanmasını daha sonra diğer hedeflerin karşılanmasını gerektirir. Bu durum su ilişki ile gösterilebilir.

(En önemli hedef) P1>> P2>>P3 (En az önemli hedef)

Pozitif Kısıtlama

Modelde yer alan tüm değişkenlerin, yani karar ve sapma değişkenlerinin pozitif deler almasını ifade eder ve modelde kesinlikle yer alması gereken kısıtlamadır.

3.6. Hedef Programlamanın Matematiksel Yapısı

Hedef programlamada hedef değerlerinin belirlenebilmesi için problem ilk olarak doğrusal programlama modeli seklinde ifade edilir. Hedef programlamanın matematiksel olarak ifadesinde model, amaç fonksiyonu, hedef kısıtları ve pozitif kısıtlama olarak üç faktör içerir.

Bu faktörlerden ilki olan amaç fonksiyonları, karar vericiye bağlı olarak tek hedefli, eşit öncelikli (öncelikli olmayan), çok hedefli, öncelikli çok hedefli, ağırlıklı çok hedefli ve ağırlıklı öncelikli çok hedefli programlama olarak beş farklı şekilde ifade edilir.

İkinci adıma olan hedef kısıtlarının belirlenmesinde ise, karar vericinin istekleri, eldeki kaynaklar ve karar değişkenleri üzerine konulan kısıtlamalar göz önüne alınır.

Üçüncü adım olan pozitif kısıtlama ise tüm değişkenlerin pozitif değerli olması gerektiğini ifade eder.

3.7. Hedef Programlamanın Kurulum Aşamaları

Hedef Programlama modelinin oluşturulmasında ve bunun matematiksel ifadesinde, aşağıdaki adımlar izlenmektedir (Ignızıo, 1976: 177).

Aşama 1. Karar Değişkenlerinin Belirlenmesi: Bir hedef programlama karar modelinin formülasyonunda öncelikle karar değişkenleri belirlenmelidir. Karar değişkenleri, karar sonucunu belirlemeye hizmet eden ve kontrol edilebilir değişkenlerdir. Bunlar belirlenirken hedef programlama tekniğinin dahil olduğu sınıf ve ele alınan uygulama alanına göre farklılık gösteren problemler sisteminin yapısına uygun değişkenler seçilmelidir. Hedef programlama tekniğinin matematiksel formülasyonunda karar değişkenleri x_j ile gösterilir. Kurulan model ile bu karar değişkenlerinin en uygun değeri belirlenmektedir.

Aşama 2. Amaç ve Kısıt Fonksiyonlarının Belirlenmesi: Hedef programlama modelinin formülasyonunda her amaç bir kısıtlayıcı biçiminde ifade edilir. Bu formülasyonda, modele sapma değişkenleri içermeyen kısıtlayıcıları da dahil etmek mümkündür. Hedef programlama modelinde amaç ve kısıt fonksiyonları belirlenir iken karar vericinin istekleri, kısıtlı kaynaklar ve karar değişkenleri üzerine konulmuş herhangi bir kısıtlama koşulu göz önüne alınır. Buna göre amaç ve kısıt fonksiyonları birkaç grup altında toplanabilir. Bunlar;

Karar Verici İsteklerine Bağlı Amaç ve Kısıtlar: Kâr maksimizasyonu, maliyet minimizasyonu, atıl kapasitenin minimizasyonu, fazla mesai veya üretim zamanın minimizasyonu, risk minimizasyonu, hatalı üretim minimizasyonu ve üretim sırasında kullanılmayan hammaddenin minimizasyonu gibi amaçlardır.

Kısıtlı Kaynaklara Bağlı Amaç ve Kısıtlar: Kısıtlı hammadde, kısıtlı işgücü, kısıtlı bölge ve kısıtlı zamandır. Tüm bu amaçlar eldeki kısıtlı kaynakların optimal şekilde kullanılmasını sağlayacak kısıtlardır.

Karar Değişkenleri ile İlgili Amaç ve Kısıtlar: Değişkenlerin negatif olmaması ve değişkenlerin üzerine konulan bazı kısıtlamalardır.

Amaç ve kısıt fonksiyonları oluşturulduktan sonra, bunlar arasında birleştirilebilmesi mümkün olanlar varsa birleştirilir. Yine modelin çözümüne etkilemeyeceği düşünülen amaç ve kısıtlar elenerek amaç ve kısıt fonksiyonlarının sayısı azaltılır.

Yukarıdaki gruplarda belirtildiği gibi belirlenen amaç fonksiyonlarını modele dahil ederken çözümü etkilemeyeceği düşünülen amaçlar elenebilir ve birleştirilmesi olanaklı olan amaçlar birleştirilebilir.

Hedef programlamasında belirlenen amaçların incelenen probleme uygun olması için bazı özellikleri taşıması gereklidir. Bu özellikler:

1. Amaçlar problemi tüm yönleriyle kapsamalı,
2. Analizi yapılabilmeye uygun olmalı,
3. Problem içerisinde yalnızca bir kez kullanılmalı,

4. Planlama sürecini basitleştirmek amacıyla bölünebilir olmalı,

5. Daha önce açıklanan şekilde düzenlenen amaç fonksiyonların mümkün olduğunca az olmalıdır.

Amaç fonksiyonları, karar değişkenlerinin matematiksel bir fonksiyonu olarak gösterilebilir. Hedef programlama modelinde amaç (hedef) fonksiyonu Z_i şeklinde ifade edilir.

$$Z_i = f_i(\overline{X}) \qquad , \qquad \overline{X} = (X_1, X_2, , X_n)$$

Bu formülasyonda $f_i(\overline{X})$, i'nci amaca ilişkin karar değişkenlerinin bir fonksiyonudur. Modelde bütün amaç fonksiyonlarına ait bir sağ taraf değeri vardır. O halde i'nci amaç fonksiyonunun matematiksel gösterimi;

$$f_i(\overline{X}) \begin{Bmatrix} \leq \\ = \\ \geq \end{Bmatrix} b_i$$

şeklindedir. Bu formülasyonda ;

$\overline{X}$: Karar değişkenleri vektörünü,

b_i : i 'nci amaç fonksiyonuna ait sağ taraf değeri veya $f_i(\overline{X})$' in sağlanması gereken hedef değerini gösterir.

Formülden de görüleceği üzere amaç fonksiyonu, amaç fonksiyonuna ait sağ taraf b_i değerinden küçük eşit ($\leq$), eşit (=) veya büyük eşit ($\geq$) olabilir (Ignizio, 1976: 23 - 24).

Hedef programlama modelinde, herhangi bir amaç fonksiyonu için pozitif ve negatif sapma değişkenleri vardır. Bir başka deyişle erişimüstü ve erişimaltı olarak adlandırılan bu değişkenler sırasıyla P_i ve n_i ile gösterilir. $\overline{X}$ karar değişkenlerinin herhangi bir vektörü için n_i değeri, b_i ' den negatif bir sapmayı temsil ederken, P_i değeri pozitif bir sapmayı temsil eder. Herhangi bir hedef için P_i ya da n_i ' den herhangi biri, ya da her ikisi, sonucun uygun olabilmesi için sıfıra eşit olmak zorundadır.

Hedef programlama modelinde her bir amaç fonksiyonu ,

$f_i\,(\overline{X} + n_i - P_i = b_i)$, $i = (1,2, \ldots, m)$ biçiminde gösterilir.

Aşama 3. Mutlak Amaç Fonksiyonunun (öncelik düzeyinin) Belirlenmesi:

Bütün amaç ve kısıt fonksiyonları oluşturulduktan sonra karar verici açısından en önemli amaç fonksiyonu, mutlak amaç fonksiyonu olarak adlandırılır. Hedeflerin önem derecesini yansıtan değere hedeflerin "öncelik düzeyi" adı verilir. Hedef programlama modelinde, amaç fonksiyonunda k. öncelik düzeyi "p_k" ile gösterilir. Eğer problemdeki hedeflere ilişkin herhangi bir öncelik yoksa yani tüm hedefler eşit önceliğe sahip ise diğer adıma geçilir.

Aşama 4. Ağırlıkların Belirlenmesi: Problemde var olan hedeflerin önemlerine göre ağırlıklandırılması söz konusu ise her bir hedefe ağırlıklar atanır. Amaç fonksiyonunda ağırlıklar "w_k" olarak gösterilir. Problemde hedefler için hem öncelikler hem de ağırlıklar söz konusu olabilir. Öncelikler arasında, Formül aşağıdaki gibi bir ilişki vardır.

$$w_1p_1 > w_2p_2 > w_3p_3 > \ldots\ldots\ldots\ldots > w_kp_k$$

Aşama 5. Amaç Fonksiyonunun Oluşturulması: Amaç fonksiyonu sapma değişkenlerinden oluştuğundan bu aşamada amaç fonksiyonuna doğru sapma değişkenlerinin alınması çok önemlidir. Hangi sapma değişkenlerinin amaç fonksiyonuna alınacağı ise hedeflere göre belirlenir. Hedeflerdeki istenmeyen sapma değişkeni veya değişkenleri en küçüklemek için amaç fonksiyonuna yerleştirilir.

Ayrıca diğer adımlarda belirlenmiş olan öncelikler ve/veya ağırlıklar da her bir hedef için amaç fonksiyonuna eklenir. Çözüm surecinde amaç fonksiyonundaki istenmeyen sapma değişkenleri her bir hedef için atanan öncelikler ve/veya ağırlıklar dikkate alınarak en küçüklenmeye çalışılır. Buna göre hedef programlamada genel olarak amaç fonksiyonu aşağıdaki gibi gösterilir.

$\text{Min } Z = \sum_{i=1}^{m} w_i \, p_i \, (d_i^{+} + d_i^{-})$

$\sum_{j=1}^{n} a_{ij} \, x_{ij} + d_i^{-} - d_i^{+} = b_i$ (hedef kısıtları)

$Cx \leq c$ (sistem kısıtı)

$x_{ij}, d_i^{-}, d_i^{+} \geq 0$;(i=1, 2, 3,…,m; j= 1, 2, 3, …,n)

şeklinde yazılabilir.

x_{ij}: karar değişkenleri

bi: i'inci hedef düzeyi,

d_i^{-}: hedeften negatif sapma değeri,

d_i^{+}: hedeften pozitif sapma değeri,

w_i: i. hedefin sapma değişkenlerine verilmiş olan matematiksel ağırlıklar (diferansiyel ağırlık),

c: eldeki kaynak,

C: sistem kısıtı ile ilgili matris katsayısı,

P_i: i hedefine verilen öncelik, ($Pi >>> P_{i+1}$),

a_{ij}: karar değişkeni katsayısını gösterir.

Burada bi, i=l, 2,..., m karar verici tarafından amaçlar için belirlenmiş hedef değerlerdir, d_i^- ve d_i^+ ’ler i. hedeften eksi ve artı sapmaları göstermektedir, a_{ij}’nin değeri, karar vericinin değer fonksiyonuna bağlıdır. Hedef programlamanın en yaygın kullanılan formülasyon sekli, karar vericinin, amaçlar için bir takım hedef değerler belirlemesine ilave olarak amaçların önem derecelerine göre sıralanması ile ilgili bilgiyi de verebileceğini kabul eder.

Aşama 6. Negatif Değer Almama Kısıtının Oluşturulması: Negatif değer almama kısıtı, modelde yer alan karar değişkenleri ve sapma değişkenlerinin negatif değerler almayacağını gösteren kısıttır. Bu kısıt, modelde aşağıdaki gibi yer alır.

x_j , n_i , $p_i \geq 0$

3.8. Hedef Programlama Türüne Göre Amaç Fonksiyonunun Oluşturulması

Hedef programlamada amaç fonksiyonları karar vericiye göre beş farklı şekilde ifade edilir (Zeleny, 1982, s.300).

Min Z = d_i^+ (p_i) ya da Min Z = d_i^- (n_i)

3.8.1. Eşit Ağırlıklı (Öncelikli) Ya da Öncelikli Olmayan Çok Hedefli Programlamada Amaç Fonksiyonu

Probleme ilişkin hedefler eşit önemli (ağırlıklı) ise modelde yer alan hedef kısıtındaki tüm istenmeyen sapma değişkenlerinin toplamı minimize edilir. Bu biçimdeki amaç fonksiyonunun anlamlılığı, sapma değişkenlerinin aynı biçimde olmasına bağlıdır.

Min Z = $\sum_{i=1}^{m}(d_i^+ + d_i^-)$

3.8.2. Öncelikli Çok Hedefli Programlamada Amaç Fonksiyonu

Hedef programlamanın karar vericinin hedeflerini bireysel isteğine bağlı olarak belirlediği önem derecesine göre ifade edilen bir programlama şeklidir. Karar verici kendi tercihine göre hedeflerini en önemliden daha az önemliye doğru sıralar. m adet hedef p_k öncelik sırasına göre işleme girer.

$$\text{Min } Z = \sum_{i=1}^{m} p_k (d_i^{+} + d_i^{-}) \qquad p_1 >> p_2 \geq p_3 \underline{>>} \ldots \underline{>>} p_k$$

Burada p_1 hedefi gerçekleştirilmeden p_2 hedefinin gerçekleştirilmesi mümkün değildir. Hedefler önem sırasına göre doyurulur.

Hedeflerin öncelik sıralamasının, hedeflere verilen ağırlık ile yapıldığı durumda bile ağırlık çarpanı ne kadar büyük olursa olsun p_1 'in önemi p_2 'den her zaman daha fazla olacaktır (Zeleny, 1982: 300).

3.8.3. Ağırlıklı Çok Hedefli Programlamada Amaç Fonksiyonu

Bu tür problemlerin amaç fonksiyonundaki sapma değişkenlerine ağırlık verilir. Genellikle bu yaklaşım eşit öncelikli çok hedefli problemlerin sapma değişkenlerinin ölçü birimleri farklı olduğunda kullanılır. Burada w_k, i'inci hedeften oluşan sapmaya ilişkin ağırlığı göstermektedir. Amaç fonksiyonunun bu şekilde oluşturulmasıyla kurulan modele Archimedian Hedef Programlama modeli denir.

$$\text{Min } Z = \sum_{i=1}^{m} w_{ik} (d_i^{+} + d_i^{-})$$

3.8.4. Ağırlıklı-Öncelikli Çok Hedefli Programlamada Amaç Fonksiyonu

Bazı hedef programlama problemlerinde aynı hedefe ilişkin iki veya daha fazla sapma değişkeni, aynı öncelik düzeyinde amaç fonksiyonunda yer alabilir. Böyle bir durumda, sapma değişkenlerinin önceliği aynı ise, bu sapma değişkenlerde ağırlıklar kullanılarak hangi sapmanın daha önemli olduğu belirlenir (Öztürk, 2005: 321) bu durumda amaç fonksiyonu aşağıdaki gibi yazılabilir.

$$\text{Min } Z = \sum_{i=1}^{m} p_k (w_{ik}^{+} d_i^{+} + w_{ik}^{-} d_i^{-})$$

Burada;

P_k: Her bir sıralı hedefe atanan öncelikli faktör düzeyini (örneğin, $p_1 > p_2 > p_i$)

w_i: Hedefin altında ve üstündeki sapma değişkenlerine verilen göreceli ağırlık değerlerini gösterir.

3.9. Hedef Kısıtlarının Oluşturulması

Hedef programlamanın ikinci adımı olan hedef kısıtlarının belirlenmesinde üç olasılık söz konusu olur.

Birinci durum için; $f_i(x) \geq b_i$ olduğunda, $d_i^+ \geq 0$ olması istenir yani; d_i^+ kısıtlanmayan sapma değişkenidir. Sağ taraf için istenen durum b_i iken istenmeyen sapma değişkeni d_i^- dir. Bu nedenle amaç fonksiyonunda olması gereken yani istenmeyen sapma değişkeni d_i^- dir.

İkinci durum için; $f_i(x) \leq b_i$ olduğunda, $d_i^- \geq 0$ olması istenir yani; kısıtlanmayan sapma değişkeni d_i^- 'dir. İstenmeyen sapma değişkeni d_i^+ olduğundan, amaç fonksiyonunda yer alması gereken yani istenmeyen sapma değişkeni d_i^+ 'dır.

Üçüncü durum için; $f_i(x) = b_i$ olduğunda ise istenen durum hem d_i^+ hem de d_i^- 'nin sıfıra eşit olmasıdır. Bu durum kaynakların tamamının kullanıldığı durumlarda söz konusu olur. Burada hiçbir sapma değişkeni sınırlandırılmamıştır. Bu durumda istenmeyen sapma değişkeni yani amaç fonksiyonunda yer alması gereken değişkenler d_i^+ ve d_i^- 'dir.

Pozitif Kısıtlama

Modelde yer alan tüm değişkenler sıfır veya sıfırdan büyük değerli olmalıdır. Hedef programlama modelinde bu kısıt aşağıdaki şekilde yer alır.

x_j, d_i^- , $d_i^+ \geq 0$

Burada d_i^-, $d_i^+ = 0$ sabitlik koşulu, herhangi bir aşamada aynı denkleme ait iki sapma değişkeninin aynı anda pozitif değer alamayacağını ifade etmektedir.

3.10. Hedef Programlamanın Türleri

Hedef programlama modelin yapısına, amaç fonksiyonlarının yapısına, karar değişkenlerine ve katsayıların özelliklerine göre sınıflandırılabilir (Lee, 1972: 98).

3.10.1. Modelin Yapısına Göre Hedef Programlama

a) Doğrusal Hedef Programlama

Amaç fonksiyonu ve kısıtlayıcıların doğrusal yapıda olduğu programlama çeşididir.

b) Doğrusal Olmayan Hedef Programlama

Amaç fonksiyonu ve/veya kısıtlayıcıların doğrusal olmadığı programlama çeşididir. Çözümün sağlanması için belirli yöntemlerle doğrusal yapıda dönüştürme işlemleri gerçekleştirilmeli, sonra modelin çözümüne gidilmelidir.

3.10.2. Karar Değişkenlerinin Değerlerine Göre Hedef Programlama

a) Sürekli Değerler Alabilen Hedef Programlama

Karar değişkenlerinin değerlerinin pozitiflik sınırı ile her değer alabildiği programlama çeşididir. Bu tür hedef programlama modellerinin çözümü için White'ın dal sınır metodu, min-minmax yöntemi gibi algoritmalar geliştirilmiştir.

b) Tamsayılı Hedef Programlama

Karar değişkenlerinin değerlerinin sıfırdan büyük ve tam sayı olduğu programlama çeşididir. Bu tür hedef programlama modellerinin çözümü için White'ın dal sınır metodu, minminmax yöntemi gibi algoritmalar geliştirilmiştir.

c) 0-1 Tamsayılı Hedef Programlama

Karar değişkenleri sadece 0 ve 1 delerleri alabilir. Bu tür hedef programlama modellerinin çözümü için White'ın dal sınır metodu, min-minmax yöntemi gibi algoritmalar geliştirilmiştir.

3.10.3. Katsayıların Özelliklerine Göre Hedef Programlama

a) Deterministik Hedef Programlama

Kısıtlayıcılarda karar değişkenlerinin katsayıları olarak tanımlayabileceğimiz teknolojik katsayıların kesin belirlediği durumdur.

b) Stokastik Hedef Programlama

Teknolojik katsayıların olasılık seklinde tanımlandığı programlama şeklidir.

c) Bulanık (Fuzzy) Hedef Programlama

Teknolojik katsayıların kesin olarak ifade edilemediği durumda ortaya çıkan programlama seklidir.

3.10.4. Amaç Fonksiyonunun Yapısına Göre Hedef Programlama

Amaç fonksiyonunu yapısına göre hedef programlama türlerini yukarıda hedef programlamanın matematiksel yapısını anlatırken açıklamıştık. Bunlar; tek hedefli programlama, eşit ağırlıklı çok hedefli programlama, öncelikli çok hedefli programlama, ağırlıklı- öncelikli hedef programlamadır.

3.11. Hedef Programlama ile Doğrusal Programlamanın Karşılaştırılması

Hedef programlama ile doğrusal programlama yöntemlerinin karsılaştırılmasından elde edilen farklar aşağıda verilmiştir.

- Doğrusal programlama modelinde doğrusal bir amaç fonksiyonu optimal yapılmaya çalışılırken, hedef programlama modelinin amaç fonksiyonunda hedeflerden sapmalar minimize edilmeye çalışılır. Hedef programlamada doğrusal programlamadan farklı olarak tek bir amaç yerine, birbiri ile çelişebilen birden çok amaç bulunabilir. Bundan dolayı doğrusal programlamadaki çözüm optimal iken hedef programlamada bulunan çözüm en uygun çözümdür.

- Doğrusal programlamada bütün kısıtlar eşit önemdedir ve hepsi eşanlı olarak sağlanmalıdır. Hedef programlamada ise kısıtlar modelde hedef olarak yer alır ve belirlenen öncelik sıralarına göre sağlanmaya çalışılır. Hedef programlamada, amaç fonksiyonu, aynı zamanda öncelikli hale getirilebilen sapma değişkenlerinin ağırlıklı toplamının minimize edilmesini ister. Öncelikli hale getirme işlemi tercihli bir şekilde yapılır. Burada sapma değişkenleri için ağırlıklar, kendine denk düşen hedef değerlerinden her birim sapma için göreceli bir ceza'yı yansıtır. Ayrıca doğrusal programlamada tek bir amaç mutlak sağlanmaya çalışılırken, hedef programlamada en öncelikli hedeften başlanarak tüm hedefler sağlanılmaya çalışılır ve bütün amaçların sağlanması gerekmez.

- Hedef programlamada bulunan pozitif ve negatif sapma değişkenleri, doğrusal programlamadaki aylak değişkenlere karşılık gelir.

- Doğrusal programlamada amaç fonksiyonu maksimizasyon veya minimizasyon şeklinde olabilirken hedef programlamada amaç fonksiyonu sadece minimizasyon şeklinde olur. Hedef programlamada hedefler birer kısıt olarak modele girer. Kaynaklar üzerindeki sınırlamaları yansıtan kısıtlar modele aynen herhangi bir doğrusal programlama modeline katılacağı gibi dahil edilir.

- Doğrusal programlamada hedef belirlenmez, hedef programlamada ise hedef değerleri gereklidir.

- Hedef programlama karar verici açısından daha esnek yapıya sahip bir yöntemdir.

- Doğrusal programlamada ve hedef programlamada bütün değişkenler sıfır ya da sıfırdan büyük değerler almak zorundadır (Doğan, 1995: 56).

- Hedef programlaması modelinin çözümü esnasında doğrusal programlamada olduğu gibi simpleks algoritması kullanılır. Hedef programlamada çözümde temele girecek değişken doğrusal programlamada olduğu gibi en yüksek önceliğe sahip ve en yüksek gelişmeyi sağlayacak değişken olmalıdır. Bu nedenle hedef programlamada bütün amaçlar eniyilenemeyebilir. Doğrusal programlamada ise bunun aksine bulunan çözüm optimum çözümdür. Yapılan çözümlemeler sonucu doğrusal programlama ile optimum çözüm bulunamaz iken hedef programlamada uygun çözüm bulunabilir.

Hedef programlamasında problemin çözümüne ilk öncelikli, P_1, hedefinden başlayarak tüm amaçların sağlanmaya çalışılarak devam edilir. Hedef programlamasının amaç fonksiyonunda karar değişkenleri, X_j, bulunmaz. Amaç fonksiyonunda sadece minimize edilmesi arzu edilen hedeflerden sapma değişkenleri d_i^- ve d_i^+, yer alır.

3.12. Doğrusal Hedef Programlamanın Çözümünde Kullanılan Yöntemler

Çalışmanın üçüncü bölümünde yani uygulama kısmında doğrusal hedef programlama kullanılacağından çözüm yöntemleri kısmında doğrusal hedef programlamanın çözüm yöntemleri anlatılacaktır.

Doğrusal hedef programlama modelinin çözümünde grafik yöntem, iteraktif çözüm yöntemi ve değiştirilmiş simpleks yöntem kullanılır.

3.12.1. Grafik Yöntemi

Grafik yöntem üç veya daha fazla değişkenden oluşan problemlerin çözümü için uygun değildir. Ancak bu yöntemle, büyük boyutlu problemler için kullanılan çözüm yöntemlerinin anlaşılması sağlanabilir.

Grafik yöntemin adımları aşağıda verilmiştir.

1. Bütün hedef ve sistem kısıtları çizilir.

2. Birinci öncelikli hedefler için çözüm alanı belirlenir.

3. Bir sonraki önceliğe sahip hedefler için çözüm alanı belirlenir. Ancak bu çözüm alanı ilk öncelikli hedefler için belirlenmiş olan çözüm alanına uygun olmalıdır.

4. Eğer işlemin herhangi bir anında çözüm bölgesi tek bir noktaya indirgenebilirse, işlem sonuçlandırılabilir. Çünkü bu durumda yeni bir gelişme sağlanamaz.

5. Üçüncü ve dördüncü adımlar tek bir noktaya indirgenene kadar ya da bütün hedefler sağlanana kadar devam edilir. Böylece en uygun çözüm bulunmuş olur.

Karar verici için tüm hedefler aynı önceliğe sahip ise bu durumda hedeflerle ulaşılmak istenen sonuçlar arasındaki en yakın uzaklığı veren çözüm araştırılır.

Örnek: Masa ve sandalye üreten bir mobilya şirketinde montaj ve son işlem daireleri mevcuttur. Belirlenen üretim devresinde bir masa üretimi için 16 saat montaj ve 8 saat son işlem kullanılmaktadır. Bir sandalye üretimi için 8 saat montaj ve 12 saat son işlem kullanılmaktadır. Şirketin elindeki günlük montaj kapasitesi 192 saat ve toplam son işlem kapasitesi de 240 saattir.

İşletmenin bir masa üretiminden elde ettiği kar 30 TL ve bir sandalye üretiminden elde ettiği kar ise 26 TL dir. İşletmenin tek hedefi günlük karının en az 560 TL olmasıdır.

Çözüm:

Karar değişkenleri;

x_1: Üretilecek günlük masa miktarı
x_2: Üretilecek günlük sandalye miktarı

Sapma değişkenleri;

d_1^-: Hedeflenen 560 TL karın altında kalan kar miktarı
d_1^+: Hedeflenen 560 TL karın üzerinde kalan kar miktarı

Amaç Fonksiyonu;

$30x_1 + 26x_2 + d_1^- - d_1^+ = 560$

Hedef Kısıtlayıcısı;

Min $Z = d_1^-$

Yapısal Kısıtlayıcılar;

$16x_1 + 8x_2 \leq 192$
$8x_1 + 12x_2 \leq 240$

Pozitif Kısıtlayıcı;

$x_1, x_2, d_1^-, d_1^+ \geq 0$

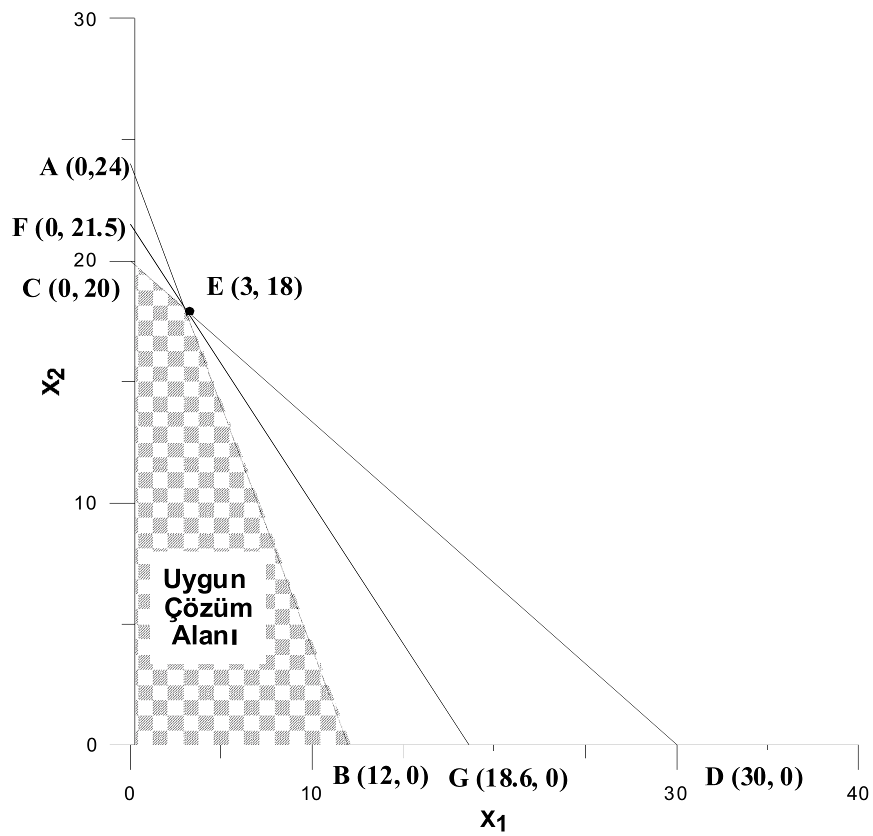
Böylece problemin modeli kurulmuş oldu. Örnekten de anlaşılacağı gibi problem tek hedefli bir problemdir.

Problemi x_1 ve x_2 ye sırasıyla sıfır vererek çözdüğümüzde köşe noktalarına ulaşılabilir. Şöyle ki;

Tablo 15: Tek Hedefli Problemin Köşe Noktaları

Uygun Köşe Noktası	Sapma Değişkenleri		Amaç Fonksiyonu
O (0, 0)	$d_1^- = 560$	$d_1^+ = 0$	z = 560
B (12, 0)	$d_1^- = 200$	$d_1^+ = 0$	z = 200
C (0, 20)	$d_1^- = 40$	$d_1^+ = 0$	z = 40
E (3, 18)	$d_1^- = 2$	$d_1^+ = 0$	z = 2

Grafik 6: Hedef Programlamanın Grafik Çözümü



3.11.2. İteratif Çözüm Yöntemi

Bu yaklaşım hedef programlamanın önceliklerinin belli bir sırada izlendiği özelliğine dayanır. Problemin ilk aşamada sadece en yüksek öncelikli sapma değişkenlerinin minimizasyonunu sağlamak için tanımlanarak çözülür. Elde edilen çözümdeki minimum sapma değeri ve değişkenleri yeni kısıt olarak probleme ilave edilir ve mutlak bir amaç fonksiyonu özelliğini kazanır. Böylece bir sonraki önceliğe sahip sapma değişkenlerinin minimizasyonu sağlamaya çalışılırken, bir önceki aşamada minimize edilmiş sapma değişkenlerinin ideal değerlerinden fedakârlık edilmemiş olunur. Her aşamadaki çözüm grafik veya simpleks çözümlerle elde edilebilir (Kuruüzüm, 1986: 54).

Min $a_1 = h_1$ (n, p)

$\sum_{j=1}^{J} c_{ij} + n_i - p_i = b_i$ $\quad$ iP_1

$x_j, n_i, p_i \geq 0$ $\quad$ $\forall i, j$

Bu modelde yalnızca, birinci öncelik düzeyindeki hedeflere ilişkin sapmalar minimize edilmektedir. a_1 için bulunan optimal çözüm a_1* olarak gösterilir. Daha sonra bir sonraki öncelik düzeyine geçilerek, erişim fonksiyonunun a_2 terimi minimize edilir. Bu modelin kısıtları, birinci öncelik düzeyindeki tüm hedefler, ikinci öncelik düzeyindeki tüm hedefler ve h_1 (n, p) = a* dır. a_1* 'ın kısıt olarak alınmasının nedeni ikinci öncelikli hedefin birinci öncelikli hedefin önüne geçmesini engellemektir. Süreç tüm öncelik düzeyleri ele alınıncaya kadar devam eder. Çözülen son doğrusal programlamanın çözümü, hedef programlamanın çözümünü verir (Sucu, 2003: 86).

İteratif çözüm yönteminin adımları aşağıda sıralanmıştır:

1. k = 1 alınır ve birinci öncelik düzeyinden başlanır.

2. Birinci öncelik düzeyi için doğrusal programlama modeli oluşturulur.

3. İkinci adımda oluşturulan model simpleks yöntemi ile çözülür ve optimal çözüm a_k*olarak belirlenir.

4. k = k +1 yapılır. Eğer k > *K* ise yedinci adıma geçilir.

5. k. düzey için doğrusal programlama modeli kurulur. Önceki adımlarda elde edilen tüm sonuçlar kısıt olarak eklenerek;

$\text{Min } a_k = h_k\,(n, p)$

$f_t\,(\overline{x}) + n_t + p_t = b_t$

$h_s(n, p) = a_s{*}$

$x_j,\ n_i,\ p_i \geq 0$

$s = 1,\ldots, k-1 \qquad \forall i$

$t = 1,2,\ldots,$ *k* öncelik düzeylerindeki hedef ya da kısıtların indis kümesi şeklinde yazılır.

6. Üçüncü adıma dönülür.

7. Son doğrusal programlama modeli çözüldüğünde, elde edilen x* çözümü orijinal doğrusal hedef programlama modelinin çözümüdür (Sucu, 2003: 87).

3.12.3. Değiştirilmiş Simpleks Yöntem

Birinci bölümde simpleks çözüm yöntemi ayrıntılı olarak anlatılmıştı. Doğrusal hedef programlama problemlerinin çözümünde kullanılan simpleks yöntemi, klasik doğrusal programlama modelinde kullanılan simpleks yönteminde bazı değişiklikler yapılarak elde edilmiştir ve bu nedenle de değiştirilmiş simpleks yöntemi adıyla bilinir (Schmidt, 1993: 67).

Tek hedefli doğrusal programlama probleminin çözümü, klasik doğrusal probleminin çözümüne benzer. Tek fark, başlangıç simpleks tablosunda hedef kısıtına karşılık gelen temel değişkenlerin (d_1^-) alınmasıdır. Yapısal kısıta karşılık gelen temel degiskenler doğrusal programlamadaki gibidir. Bilindiği gibi eşitsizlik seklindeki denklemlere eklenen atıl veya artık değişkenler dahil olmak üzere m+n tane bilinmeyen bulunması ve buna karşılık sadece m denklem olduğundan, herhangi n bilinmeyen sıfıra eşit alınıp diğer m bilinmeyen eşitlikler takımından elde edilebilir. Buna temel çözüm denir (Gürel, 1966: 47).

Eşit ağırlıklı çok hedefli programlamanın çözümünde tek hedefli doğrusal programlama probleminin simpleks ile çözümüne benzer, yalnızca hedef birden çok olduğu için d_1^-, d_2^-,d_3^-, ... şeklindeki temel değişkenler sayısında bir artış olur. Ayrıca bu tür programlamada sapmaların ölçü birimleri aynı ise amaç fonksiyonunun anlamlı sonuç vereceği unutulmamalıdır.

Ağırlıklı çok hedefli programlamada amaç fonksiyonunda sapma değişkenlerine ağırlık verilir. Daha öncede belirtildiği gibi eğer sapma değişkenlerinin ölçü birimleri birbirinden farklı ise ağırlıklı çok hedefli programlama tercih edilir.

Öncelikli çok hedefli programlamada ise simpleks yöntemi klasik doğrusal programlamadaki simpleks yöntemden biraz farklıdır. Öncelikli hedef programlamanın simpleks ile çözümünde izlenecek yola aşağıda yer verilmiştir.

1. Başlangıç simpleks tablosunun oluşturulması ile her bir p_i önceliği için n adet $C_j - Z_j$ satırının varlığı anlaşılır. Burada öncelikli olan hedefin başlangıç tablosundan ilk satırda yer alması gerektiği unutulmamalıdır.,

2. p_i öncelikli amaç satırındaki $C_j - Z_j$ değerine bakılır. En küçük negatif değerli değişken çözüme (temel değişken sıfır olarak) girer. Bu işlem amaç fonksiyonunun değerini azaltarak, birinci hedefin sağlanmasına yardımcı olur.

3. Çözümden çıkacak değişkeni belirlerken en küçük $b_i/\ a_{ij}$ değişkeni kullanılır ve her satır için yeni sıralar bulunur. Birinci simpleks tablosu klasik simpleks yönteminde yapılan işlemlerin aynısıdır.

4. Birinci hedef amaç satırının $C_j - Z_j$ satırında eğer negatif elemanlar var ise, Adım 2 ve Adım 3 tekrarlanır ve tüm $C_j - Z_j \geq 0$ oluncaya kadar işleme devam edilir.

5. Tüm p_i ’ler için $C_j - Z_j \geq 0$ olması hedeflerin tatmin edildiği anlamına gelir. Ancak düşük önceliğe ait satırda negatif $C_j - Z_j$ değeri var ve onun altındaki yüksek öncelikli hedef sütununun değeri pozitif ise $c_j - z_j$ negatif olmasına rağmen çözüme girmez ve bu durumda çözüm yine optimaldir. Çünkü çözüme

girmesi, ondan daha öncelikli olan hedefin sapmasını arttıracak, amaç fonksiyonunun ise değerini azaltacaktır.

Ağırlıklı-öncelikli çok hedef programlama çözümünde de yine öncelikli hedef programlama simpleks yöntemi kullanılır. Tek fark, öncelikli programlamada C_j katsayıları 1 iken, bu programlama modelinde C_j katsayıları verilen ağırlığa göre değişir. Simpleks yöntemin uygulanışında bir farklılık yoktur.

Şimdi bu farklılığın daha iyi görülebilmesi için öncelikli hedef programlamayla ilgili bir örnek adım adım anlatılarak simpleks metodu ile çözülmeye çalışılacaktır.

Örnek:

Hedef Kısıtı;

Min Z = $p_1 d_1^- + p_2\, d_2^- + p_3 d_3^-$

Kısıtlayıcılar;

$7x_1+3x_2+d_1^- - d_1^+ = 40$
$10x_1+5x_2+d_2^- - d_2^+ = 60$
$5x_1+4x_2+d_3^- - d_3^+ = 35$

Ve
$x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ \geq 0$

1. Problemin başlangıç simpleks tablosu oluşturulur. Bu tabloda her p_i önceliği için n tane $C_j - Z_j$ satırı yer alırken klasik simpleks tablosunda sadece bir tek $C_j - Z_j$ satırı bulunur. İlk olarak p_1 öncelikli satırdan başlanır ve Adım 2 ye geçilir.

2. Hedef programlama simpleks tablosunda çözüme girecek değişken belirlenirken şu yöntem izlenir. p_i önceliği amaç satırındaki $C_j - Z_j$ değerine bakılır. Enküçük negatif değerli değişken çözüme girer.

3. Çözümden veya temelden çıkacak değişken içinde bilinen ölçüt kullanılır ve sonra da temel sıra işlemleri ile her satır için yeni sıralar bulunur.

4. Birinci öncelikli hedef amaç satırının $C_j - Z_j$ elemanlarına bakılır, eğer negatif elemanlar var ise, Adım 2 ve Adım 3 işlemleri yapılarak tüm $C_j - Z_j \geq 0$ oluncaya kadar devam edilir.

5. Sonra tüm p_i'ler için $C_j - Z_j \geq 0$ oluncaya kadar Adım 2 ve Adım 3'deki işlemler yapılır. Öylece tüm hedefler doyuma ulaştığında optimum çözüme ulaşılmış olur. Ancak düşük düzeyli bir öncelik satırında negatif $C_j - Z_j$ değeri var ve onun altındaki daha yüksek öncelikli hedefin sütunun değeri pozitif ise $C_j - Z_j$ değeri negatif olmasına rağmen çözüme girmez. Bu durumda çözüm yine optimaldir.

Tablo 16: Başlangıç Simpleks Tablosu

A.K.	c_j Temel	0 x_1	0 x_2	0 d_1^+	0 d_2^+	0 d_3^+	1 d_1^-	1 d_2^-	1 d_3^-	Çözüm
1	d_1^-	7	3	-1	0	0	1	0	0	40
1	d_2^-	10	5	0	-1	0	0	1	0	60
1	d_3^-	5	4	0	0	-1	0	0	1	35
p_3	$c_j - z_j$	-5	-4	0	0	1	1	1	0	
p_2	$c_j - z_j$	-10	-5	0	1	0	1	0	1	
p_1	$c_j - z_j$	-7	-3	1	0	0	0	1	1	

Tablo 17: Birinci Simpleks Tablosu

A.K.	c_j Temel	0 x_1	0 x_2	0 d_1^+	0 d_2^+	0 d_3^+	1 d_1^-	1 d_2^-	1 d_3^-	Çözüm
0	x_1	1	3/7	-1/7	0	0	1/7	0	0	40/7
1	d_2^-	0	5/7	10/7	-1	0	-10/7	1	0	20/7
1	d_3^-	0	13/7	5/7	0	-1	-5/7	0	1	45/7
p_3	c_j - z_j	0	-13/7	-5/7	0	1	5/7	0	0	
p_2	c_j - z_j	0	-5/7	-10/7	1	0	10/7	0	0	
p_1	c_j - z_j	0	0	0	0	0	1	0	0	

Yukarıda görüldüğü gibi p_1 satırında hiçbir negatif değerli eleman kalmamıştır yani birinci hedefe ulaşılmıştır. Şimdi ikinci hedef için en iyi çözüm bulunmaya çalışılacaktır. İkinci hedef satırındaki en büyük negatif değerli sayının olduğu sütün d_1^+'dır.

Tablo 18: İkinci Simpleks Tablosu

A.K.	c_j Temel	0 x_1	0 x_2	0 d_1^+	0 d_2^+	0 d_3^+	1 d_1^-	1 d_2^-	1 d_3^-	Çözüm
0	x_1	1	3/5	0	0	0	0	0	0	6
0	d_1^+	0	6/5	1	0	0	-1	0	0	2
1	d_3^-	0	1	0	0	-1	0	0	1	5
p_3	c_j - z_j	0	-1	0	0	1	0	0	0	
p_2	c_j - z_j	0	1	0	1	0	0	0	0	
p_1	c_j - z_j	0	0	0	0	0	1	0	0	

Bu tablo incelenecek olursa, hedef 1 ve hedef 2'nin karşılandığı görülmektedir. Oysaki hedef 3'e karşılık gelen Satır p_3'daki tek negatif katsayılı değişkendir yani hedef 3 karşılanmamıştır. Ancak, x_2 temele giremez, çünkü Satır p_2 'de pozitif katsayıya sahiptir. Sadece, 2.önceliğin karşılanmaması durumunda bu gerçekleşebilirdi. Dolayısıyla, yukarıda görülen bu son elde edilen tablo, verilen Hedef Programlama problemi için en iyi çözümü veren tablodur.

3.13. Hedef Programlamanın Uygulama Alanları

Bu teknikten yararlanılan alanları aşağıdaki gibi sıralamak mümkündür.

- Üretim Planlaması
- Kaynak Planlaması
- Medya Planlaması
- İşgücü Planlaması
- Sağlık Planlaması
- Ulaştırma
- Ekonomik Planlama
- Portföy seçimi
- Finansal Planlama
- Öğrenci başarısının kestirimi
- Bütçe planlaması
- Proje seçimi

3.14. Hedef Programlamanın Avantajları ve Dezavantajları

Avantajları

- Bu yöntemle iki ve daha çok amaca sahip karar problemlerinin çözümü yapılabilir.
- Gevşek kısıtlara izin verir.
- Hedef programlama, doğrusal programlamada "Uygun Çözümü Mevcut olmayan" (infeasible) problemlere uygun bir çözüm geliştirmede yardımcı teknik olarak da kullanılmaktadır.

Dezavantajları

- Başarma fonksiyonu çok sayıda amaç fonksiyonunun birleştirilmesiyle oluşturulur. Bu nedenle karmaşık bir yapıya sahip olabilirler.
- Hedef değerleri karar verici tarafından tespit edilmelidir.
- Karar verici, hedeflerin ağırlıklarını ve öncelik seviyelerini belirlemelidir.
- Bu değerleri bağdaşık hale getirecek bir yol bulunmalıdır.

DÖRDÜNCÜ BÖLÜM

4. HEDEF PROGRAMLAMA MODELİNİN SANAL BİR TEKSTİL FİRMASINDA UYGULANMASI

4.1. Firmanın Üretim Yapısı ve İş Akış Şeması

1995 yılında üretime başlayan KAR tekstil firması, 375.000 m^2 lik bir alanda 4000 çalışanı ile faaliyet göstermektedir. Ele aldığımız KAR tekstil firması entegre bir tesis olup, iplik girdisi ile başlayan süreç örme- boya- konfeksiyon- ihracat olarak devam etmektedir. Ayrıca, 4500 m^2 lik net üretim alanına sahip firma, 110 adet yuvarlak örme, 26 adet düz örme aksesuar makinesi, 4 iplik aktarma ve 3 kumaş kesme makinesiyle 24 saat üretim gerçekleştiren örme işletmesinin günlük kapasitesi 90 ton / gün'dür.

Üretilen kumaşlar 15 adet kalite kontrol makinesinden %100 olarak son kontrole tabii tutulmaktadır. Günlük 150 ton boyama kapasitesine sahip firma bünyesindeki işletmelerde değişik kapasitelerde 120 adet boyama makinesi bulunmaktadır.

Çalışmanın uygulamansında KAR firmasının ürettiği hazır giyim bölüm için yapılacaktır. Bu sebepten sadece bu işin akış şeması anlatılacaktır. Firmanın boyama işletmesinden gelen kumaşlar hazır giyim kumaş deposuna teslim edilir. Kumaşlar buradan kesim bölümüne gönderilir. Burada gelen parçalar dilim için işletmede hazır hale getirilir. Ürünlerin dikiş sürecinde ara kontrol ve dikiş işleri tamamlandıktan sonra da kontrol yapılır. Dikilen ürünler ütülenir ve paketlenir.

4.2. Konfeksiyon İşletmesi için Hedef Modelinin Oluşturulması

Daha önceki Hedef Programlama bölümünde de bahsedildiği gibi hedef programlamanın birçok türü vardır. Bu çalışmanın uygulama kısmında KAR işletmesine uygulanacak modeller Ağırlıklı Hedef Programlama ve Eşit Ağırlıklı Hedef Programlama modelleri olarak belirlenmiştir. Modellerin seçimindeki kriterler firmanın üst yönetiminin hedefleri olan kar, kumaş kesim fireleri, müşteri talepleri, satışlar ve çalışan iş gücü saati yönündedir.

Şimdi sırasıyla modeli kurmak için gerekli aşamalar tamamlamaya çalışılacaktır.

4.2.1. Modelin Karar Değişkenleri

İşletmenin üretim süreci izlendiğinde müşterileri için aylık otalama 30 tür hazır giyim ürünü ürettiğini görmekteyiz. Bu ürünler kullanılan hammadde türdeşliği yanında talebi çok az ürünler birleştirilerek 15 tür ürün elde edilmiştir. Modelimizde karar değişkenleri x simgesiyle ifade edilmiş olup, x_i = (i= 1, 2, 3, …, 15) dir.

x_1 = Aylık üretilecek sıfır yaka uzun kollu t-shirt miktarını
x_2 = Aylık üretilecek cepli t-shirt miktarını
x_3 = Aylık üretilecek kolsuz t-shirt miktarını
x_4 = Aylık üretilecek askılı bayan gecelik miktarını
x_5 = Aylık üretilecek askılı erkek atlet miktarını
x_6 = Aylık üretilecek kısa kollu gecelik miktarını
x_7 = Aylık üretilecek capri erkek pantolon miktarını
x_8 = Aylık üretilecek erkek pijama miktarını
x_9 = Aylık üretilecek polo yaka sweat miktarını
x_{10} = Aylık üretilecek erkek pantolon miktarını
x_{11} = Aylık üretilecek pileli etek miktarını
x_{12} = Aylık üretilecek cepli yelek miktarını
x_{13} = Aylık üretilecek kayık yaka sweat miktarını
x_{14} = Aylık üretilecek bayan atlet miktarını
x_{15} = Aylık üretilecek uzun kollu sweat miktarını gösterir.

Şirket politikası gereğince yönetimin en temel sorumluluklarından biri şirketin karlı büyümesidir. Bunun gerçekleştirilebilmesi için de bu ürünlerin müşterilerin talep ettiği miktarda ve kalitede üretilebilmesi ve zamanında müşteriye sunulmasıdır.

Bu nedenle hangi ürünlerin üretiminin karlı olduğu ve ne miktarda üretilmesi gerektiği bilinmelidir. Burada karar değişkenleri olarak ele aldığımız ürünler işletmenin genelde bahar ve yaz döneminde ürettiği ürünlerdir. Yukarıda ele aldığımız ürünler işletmenin en çok talep edilen ve üretilen ürünleridir.

4.2.2. Modelin Teknolojik Kısıtlarının Belirlenmesi

Modelin teknolojik kısıtlayıcıları üretilecek 15 tür ürünün üretilmesinde kullanılan kumaş miktarı, işçilik süreleri ve makine süreleri verilecektir.

- *Kumaş Kısıtlayıcıları*: İşletmemizde penye, çift iplik, 3 iplik, ribana ve lacost türlerinde örme kumaşlar üretilmektedir. Bu kumaşlara ilişkin kısıtlayıcılar aşağıda verilmiştir. Kısıtların sağ taraf sabitleri, ürünleri için işletmenin elindeki aylık kumaş miktarını göstermektedir.

Penye Kumaş Kısıtlayıcısı;

$0{,}145x_1 + 0{,}216x_2 + 0{,}212x_3 + 0{,}212x_4 + 0{,}179x_5 + 0{,}328x_8 + 0{,}218x_9 + 0{,}147x_{12} + 0{,}116x_{13} \leq 530{,}185$ kg

Çift İplik Kumaş Kısıtlayıcısı;

$0{,}550x_7 \leq 20{,}2145$ kg

3 İplik Kumaş Kısıtlayıcısı;

$0{,}709x_{11} \leq 23{,}355$ kg

Ribana Kumaş Kısıtlayıcısı;

$0{,}149x_{10} + 0{,}145x_{14} + 0{,}210x_{15} \leq 175{,}172$ kg

Lacost Kumaş Kısıtlayıcısı;

$0{,}243x_6 \leq 78{,}880$ kg

- *Makine Kısıtlayıcısı:* Hazır giyim ürünlerin üretiminde 3 overlok, tek iğne, overlok, reçme, karyola, düğme, punterez, ütü olmak üzere 7 tür makine kullanılmaktadır. Kısıtlayıcıların sol tarafındaki rakamlar bir ürün için gerekli makine süresini (dk) sağ taraf sabiti ise makinelerin aylık çalışma kapasite sürelerini (dk) göstermektedir.

3 Overlok Makine Kısıtlayıcısı;

$0{,}035x_1 + 0{,}040x_2 + 0{,}094x_3 + 0{,}011x_4 + 0{,}109x_5 + 0{,}020x_7 + 0{,}005x_8 + 0{,}078x_9 + 0{,}045x_{10} + 0{,}166x_{11} + 0{,}077x_{12} + 0{,}040x_{14} + 0{,}191x_{15} \leq 731{,}203$ dakika

Tek İğne Makine Kısıtlayıcısı;

$0{,}008x_1 + 0{,}055x_2 + 0{,}098x_3 + 0{,}086x_4 + 0{,}032x_5 + 0{,}380x_6 + 0{,}371x_7 + 0{,}142x_8 + 0{,}214x_9 + 0{,}103x_{10} + 0{,}103x_{11} + 0{,}063x_{12} + 0{,}041x_{13} + 0{,}098x_{14} + 0{,}150x_{15} \leq 1{,}077{,}039$ dakika

4 Ovelok Makine Kısıtlayıcısı;

$0{,}042x_1 + 0{,}041x_2 + 0{,}054x_3 + 0{,}028x_4 + 0{,}056x_5 + 0{,}063x_6 + 0{,}085x_7 + 0{,}079x_8 + 0{,}131x_9 + 0{,}071x_{10} + 0{,}137x_{11} + 0{,}050x_{12} + 0{,}021x_{13} + 0{,}054x_{14} + 0{,}023x_{15} \leq 587{,}574$ dakika

Reçme Makine Kısıtlayıcısı;

$0{,}048x1 + 0{,}077x2 + 0{,}099x3 + 0{,}062x4 + 0{,}068x5 + 0{,}076x6 + 0{,}043x7 + 0{,}059x8 + 0{,}171x9 + 0{,}031x10 + 0{,}027x11 + 0{,}035x12 + 0{,}052x13 + 0{,}056x14 + 0{,}158x15 \leq 1{,}044{,}576$ dakika

Karyoka Makine Kısıtlayıcısı;

$0{,}029x_3 + 0{,}009x_4 \leq 452{,}650$ dakika

Düğme Makine Kısıtlayıcısı;

$0{,}009x_4 + 0{,}019x_5 \leq 383{,}011$ dakika

Punterez Makine Kısıtlayıcısı;

$0{,}02x_7 + 0{,}020x_9 + 0{,}014x_{11} + 0{,}021x_{15} \leq 452{,}650$ dakika

Ütü Makine Kısıtlayıcısı;

$0{,}024x_1 + 0{,}029x_2 + 0{,}008x_3 + 0{,}017x_4 + 0{,}023x_6 + 0{,}018x_7 + 0{,}034x_8 + 0{,}054x_9 + 0{,}051x_{11} + 0{,}023x_{12} + 0{,}012x_{13} + 0{,}025x_{14} + 0{,}024x_{15} \leq 522{,}288$ dakika

Ayrıca, ürünlerin üretimi için makine gerektirmeyen düğme, etiket, biye, balika yapılması için harcanan el işi kısıtlayıcısı da bir teknoloji kısıtlayıcısıdır.

El İşi Kısıtlayıcısı;

$0{,}087x_1 + 0{,}089x_2 + 0{,}159x_3 + 0{,}136x_4 + 0{,}149x_5 + 0{,}314x_6 + 0{,}159x_7 + 0{,}089x_8 + 0{,}516x_9 + 0{,}122x_{10} + 0{,}270x_{11} + 0{,}079x_{12} + 0{,}093x_{13} + 0{,}139x_{14} + 0{,}133x_{15} \leq 9{,}627{,}196$ dakika

4.2.3. Modelin Diğer Kısıtlayıcılarının Belirlenmesi

İşletmenin amaçları genellikle müşteri taleplerini, aylık satış cirolarını, aylık karlarını artırmanın yanında aylık kumaş kesi firelerini de azaltmak yönündedir. Dolayısıyla böyle bir hedefin karşılanması için bunlara ilişkin kısıtlayıcıları da belirlemek gerekir.

- *Talep Kısıtlayıcıları*: İşletmenin talep kısıtlayıcılarını belirlerken müşterilerden gelen aylık talep miktarları baz alınarak aşağıdaki gibi yazılmıştır.

$x_1 \geq 45{,}636$	$x_2 \geq 38{,}171$
$x_3 \geq 33{,}229$	$x_4 \geq 30{,}919$
$x_5 \geq 4{,}050$	$x_6 \geq 12{,}627$
$x_7 \geq 14{,}434$	$x_8 \geq 7{,}550$
$x_9 \geq 16{,}837$	$x_{10} \geq 37{,}748$

$x_{11} \geq 17{,}055$ $x_{12} \geq 26{,}024$

$x_{13} \geq 3{,}375$ $x_{14} \geq 2{,}965$

$x_{15} \geq 1{,}011$

- *Satış Kısıtlayıcısı;* İşletmeler aylık satış cirolarının belirli miktarın üzerinde olmasın hedeflerler. Böyle bir hedef kısıtlayıcısını elde etmek için öncelikle ürünlerin satış fiyatlarını ve aylık satış cirosunun bilinmesi gerekir. Satış kısıtlayıcısı aşağıda belirtilmiştir. Kısıtlayıcının sol tarafındaki parametreler ürünlerin fiyatlarını belirtir.

$$4{,}67x_1 + 7{,}07x_2 + 4{,}82x_3 + 3{,}6x_4 + 3{,}6x_5 + 7{,}2x_6 + 10{,}17x_7 + 7{,}2x_8 + 8{,}98x_9 + 7{,}18x_{10} +9x_{11} + 4{,}80x_{12} + 5{,}99x_{13} + 10{,}77x_{14} + 9{,}58x_{15} \leq 3{,}600{,}000$$

- *Kumaş Kesim Fire Kısıtlayıcısı:* Kumaş kesim fire kısıtlayıcısını belirlemek için her bir birim ürün için kesilen kumaşlarda ortaya çıkan fire miktarları kg bazında Üretim – Kesim departmanından temin edilerek belirlenmiştir.

$$0{,}44x_1 + 0{,}65x_2 + 0{,}64x_3 + 0{,}10x_4 + 0{,}60x_5 + 0{,}74x_6 + 0{,}105x_7 + 0{,}192x_8 + 0{,}260x_9 + 0{,}42x_{10} + 0{,}79x_{11} + 0{,}23x_{12} + 0{,}68x_{13} + 0{,}48x_{14} + 0{,}55x15 \leq 90{,}000$$

- *Amaç Fonksiyonu:* Uygulamada bizim işletmemizin amaç fonksiyonu olarak karın belirli bir miktarın üzerinde olmasının yönetimce hedeflendiği düşünülmüştür. Dolayısıyla modelimiz için bir amaç fonksiyonu belirlemek zorundayız. İşletmenin kar fonksiyonunu belirlemek için her bir ürünün satış fiyatlarından birim üretim maliyetleri çıkarılarak ürünlere göre birim kar katsayıları hesaplanmıştır. İşletmenin toplam karını maksimum kılan amaç fonksiyonu aşağıda verilmiştir.

$$1{,}4x_1 + 2{,}2x_2 + 1{,}2x_3 + 1{,}2x_4 + 0{,}8x_5 + 2{,}2x_6 + 2{,}1x_7 + 1{,}9x_8 + 1{,}9x_9 + 2{,}5x_{10} + 2{,}8x_{12} + 2x_{13} + 3x_{14} + 3{,}3x_{15}$$

Bu model, Klasik Doğrusal programlama modeli olup bu modelin çözümü bize bu kısıtlayıcılar altında isletmenin aylık hangi üründen ne miktarda üretileceğini ve bunun sonucunda elde edilecek maksimum karı verir. Bu modelin çözüm sonuçları işletmenin optimal üretim planını veren sonuçlardır.

Bu çalışmanın temel amacı, Doğrusal Hedef Programlama olduğu için üst yönetimden alınan bazı hedeflere göre işletme için *Ağırlıklı Hedef Programlama*, *Eşit Ağırlıklı Hedef Programlama* modelleri kurularak çözümlerinin *LINGO* paket programı ile yapılmasıdır.

4.3. İşletmenin Ağırlıklı Çok Hedefli Programlama Modeli

Bazı durumlarda yöneticiler, amaç fonksiyonundaki sapma değişkenlerine ağırlıklar vererek hedefleri arasındaki önemini ortaya koymak ister. Bu tür karar verici sorununu çözmek için hedefler arasında bir ağırlıklandırmanın yapılması gerekir.

Uygulama yapılan isletmede de üst yönetimin kendince en önemli hedefi, müşteri taleplerini bir anlamda müşteriye satışlarını zamanında ve en az müşterinin istediğinden daha fazlasını göndermektir. Çünkü bu, müşteri memnuniyetinin ön koşuludur. Dolayısıyla müşteri taleplerinin daha üstünde ürünler satıldıkça isletmenin satış cirosu ve karlılığı artacaktır. Bu da isletmeler için karlı büyüme stratejisi için temel bir yol olup küresel rekabette isletmeleri ayakta tutan en önemli stratejidir.

İşletme karının 650,000 $'dan fazla olmasını, kumaş kesim firelerinin 102,000 kg'dan az olmasını ve iş gücü hedefinin de 4,310,000 dakika işçilikten az olması istenmektedir.

Uygulama yapılan isletmede de, üst yönetim bu düşüncesinden hareketle müşteri siparişlerine 10 sayısı verilerek ağırlandırılmaya gidilmiştir. Daha önceki kurduğumuz modeldeki bilgileri de kullanarak isletme için siparişlerin ağırlıklandırıldığı Hedef Doğrusal Programlama modeli bu hedef doğrultusunda kurularak aşağıda verilmiştir.

MODEL

$$\text{Min} = d_1^- + d_2^+ + d_3^- + 10d_4^- + 10d_5^- + 10d_6^- + 10d_7^- + 10d_8^- + 10d_9^- + 10d_{10}^- + 10d_{11}^- + 10d_{12}^- + 10d_{13}^- + 10d_{14}^- + 10d_{15}^- + 10d_{16}^- + 10d_{17}^- + 10d_{18}^- + d_{19}^+$$

Kar Hedefi

$$1{,}4x_1 + 2{,}2x_2 + 1{,}2x_3 + 1{,}2x_4 + 0{,}8x_5 + 2{,}2x_6 + 2{,}1x_7 + 1{,}9x_8 + 1{,}9x_9 + 2{,}5x_{10} + 2{,}8x_{12} + 2x_{13} + 3x_{14} + 3{,}3x_{15} \geq 650{,}000$$

Fire Hedefi

$$0{,}44x_1 + 0{,}65x_2 + 0{,}64x_3 + 0{,}10x_4 + 0{,}60x_5 + 0{,}74x_6 + 0{,}105x_7 + 0{,}192x_8 + 0{,}260x_9 + 0{,}42x_{10} + 0{,}79x_{11} + 0{,}23x_{12} + 0{,}68x_{13} + 0{,}48x_{14} + 0{,}55x15 \leq 90{,}000$$

Satış Hedefi

$$4{,}67x_1 + 7{,}07x_2 + 4{,}82x_3 + 3{,}6x_4 + 3{,}6x_5 + 7{,}2x_6 + 10{,}17x_7 + 7{,}2x_8 + 8{,}98x_9 + 7{,}18x_{10} + 9x_{11} + 4{,}80x_{12} + 5{,}99x_{13} + 10{,}77x_{14} + 9{,}58x_{15} \leq 3{,}600{,}000$$

Talep Hedefi

$x_1 \leq 45636$

$x_2 \leq 38171$

$x_3 \leq 33229$

$x_4 \leq 30919$

$x_5 \leq 4050$

$x_6 \leq 12627$

$x_7 \leq 14434$

$x_8 \leq 7550$

$x_9 \leq 16837$

$x_{10} \leq 37748$

$x_{11} \leq 17055$

$x_{12} \leq 26024$

$x_{13} \leq 3375$

$x_{14} \leq 2965$

$x_{15} \leq 1011$

İşgücü Hedefi

$7x_1 + 15x_2 + 8x_3 + 7x_4 + 6x_5 + 9x_6 + 14x_7 + 11x_8 + 19x_9 + 7x_{10} + 7x_{11} + 8x_{12} + 5x_{13} + 18x_{14} + 11x_{15} \leq 3{,}790{,}800$

Kısıtlayıcılar:

Penye

$0{,}145x_1 + 0{,}216x_2 + 0{,}212x_3 + 0{,}212x_4 + 0{,}179x_5 + 0{,}328x_8 + 0{,}218x_9 + 0{,}147x_{12} + 0{,}116x_{13} \leq 530{,}185$

Çift iplik

$0{,}550x_7 \leq 20{,}2145$ kg

3 iplik

$0{,}709x_{11} \leq 23{,}355$ kg

Ribana

$0{,}149x_{10} + 0{,}145x_{14} + 0{,}210x_{15} \leq 175{,}172$ kg

Lacost

$0{,}243x_6 \leq 78{,}880$ kg

3 Overlok

$0{,}035x_1 + 0{,}040x_2 + 0{,}094x_3 + 0{,}011x_4 + 0{,}109x_5 + 0{,}020x_7 + 0{,}005x_8 + 0{,}078x_9 + 0{,}045x_{10} + 0{,}166x_{11} + 0{,}077x_{12} + 0{,}040x_{14} + 0{,}191x_{15} \leq 731{,}203$ dakika

Tek İğne Makine Kısıtlayıcısı

$0{,}008x_1 + 0{,}055x_2 + 0{,}098x_3 + 0{,}086x_4 + 0{,}032x_5 + 0{,}380x_6 + 0{,}371x_7 + 0{,}142x_8 + 0{,}214x_9 + 0{,}103x_{10} + 0{,}103x_{11} + 0{,}063x_{12} + 0{,}041x_{13} + 0{,}098x_{14} + 0{,}150x_{15} \leq 1{,}077{,}039$ dakika

4 Ovelok

$0{,}042x_1 + 0{,}041x_2 + 0{,}054x_3 + 0{,}028x_4 + 0{,}056x_5 + 0{,}063x_6 + 0{,}085x_7 + 0{,}079x_8 + 0{,}131x_9 + 0{,}071x_{10} + 0{,}137x_{11} + 0{,}050x_{12} + 0{,}021x_{13} + 0{,}054x_{14} + 0{,}023x_{15} \leq 587{,}574$ dakika

Reçme

$0{,}048x_1 + 0{,}077x_2 + 0{,}099x_3 + 0{,}062x_4 + 0{,}068x_5 + 0{,}076x_6 + 0{,}043x_7 + 0{,}059x_8 + 0{,}171x_9 + 0{,}031x_{10} + 0{,}027x_{11} + 0{,}035x_{12} + 0{,}052x_{13} + 0{,}056x_{14} + 0{,}158x_{15} \leq 1{,}044{,}576$ dakika

Karyoka

$0{,}029x_3 + 0{,}009x_4 \leq 452{,}650$ dakika

Düğme

$0{,}009x_4 + 0{,}019x_5 \leq 383{,}011$ dakika

Punterez

$0{,}02x_7 + 0{,}020x_9 + 0{,}014x_{11} + 0{,}021x_{15} \leq 452{,}650$ dakika

Ütü

$0,024x_1 + 0,029x_2 + 0,008x_3 + 0,017x_4 + 0,023x_6 + 0,018x_7 + 0,034x_8 + 0,054x_9 + 0,051x_{11} + 0,023x_{12} + 0,012x_{13} + 0,025x_{14} + 0,024x_{15} \leq 522,288$ dakika

El işi

$0,087x_1 + 0,089x_2 + 0,159x_3 + 0,136x_4 + 0,149x_5 + 0,314x_6 + 0,159x_7 + 0,089x_8 + 0,516x_9 + 0,122x_{10} + 0,270x_{11} + 0,079x_{12} + 0,093x_{13} + 0,139x_{14} + 0,133x_{15} \leq 9,627,196$ dakika

Ve; Modelde yer alan tüm karar değişkenleri ve sapma değişkenleri ≥ 0

Bu model, daha önce ifade edildiği gibi LINGO paket programı ile çözülmüş olup modelin bilgisayar çözümü EK 2'de, Bilgisayar görüntüsü ise Ek 1'de verilmiştir.

Modelin çözümüne 18. adımda optimal çözüme ulaşılmış olup amaç fonksiyonunun değeri 697,708.7'dir. Karar değişkenlerinin optimal değerleri ise;

$$\left.\begin{array}{l} x_1 = 45636 \\ x_2 = 38171 \\ x_3 = 32730 \\ x_4 = 30919 \\ x_5 = 4050 \\ x_6 = 12627 \\ x_7 = 14434 \\ x_8 = 7550 \\ x_9 = 0.00000 \\ x_{10} = 37748 \\ x_{11} = 32940.76 \\ x_{12} = 26024 \\ x_{13} = 301312.1 \\ x_{14} = 2965 \\ x_{15} = 1011 \end{array}\right\} \text{Üretim Miktarları (adet)}$$

Sapma değişkenlerinin değerleri ise aşağıda verilmiştir.

$d_1^- = d_2^- = d_3^- = d_4^- = d_5^- = d_6^- = d_7^- = d_8^- = d_9^- = d_{10}^- = d_{11}^- = d_{13}^- = d_{14}^- = d_{15}^- = d_{16}^- = d_{17}^- = d_{18}^- = d_{19}^- = d_3^+ = d_4^+ = d_5^+ = d_6^+ = d_7^+ = d_8^+ = d_9^+ = d_{10}^+ = d_{11}^+ = d_{12}^+ = d_{13}^+ = d_{15}^+ = d_{17}^+ = d_{18}^+ = 0$

$d_1^+ = 251381.7$

$d_2^+ = 248723$

$d_{12}^- = 16837$

$d_{17}^- = 3406$

$d_{14}^+ = 15885.76$

$d_{16}^+ = 297937.1$

$d_{19}^+ = 280615$

Modelin çözüm değerlerinde görüldüğü üzere bazı değişken değerleri kesirli değerler almıştır. Üretim miktarları tam sayı değerli olmak zorundadır. Bu tür bir sorun, modelin tam sayılı Hedef Programlama modeline dönüştürülerek giderilir.

4.4. İşletmenin Eşit Ağırlıklı Çok Hedefli Programlama Modeli

Bu modelde, dikkat edilmesi gereken iki önemli nokta vardır:

1. Hedeflerin, gerçekleştirilme önceliğinin hedeflere ilişkin ölçü birimlerine bağlı büyüklüklerden kolayca etkilenmemesi.

2. Hedeflerin, ölçü birimi büyüklüğünden doğrudan etkilenen amaç fonksiyonun yorumlanmasıdır (Öztürk, 2007: 305).

Üst yönetim, kar, fire, satış, talep ve işgücü hedeflerinin eşit ağırlıkta gerçekleşmesini istemektedir. Bir anlamda, kar hedefinin aylık 650,000 $’dan az olmamasını, kumaş kesim firelerinin aylık 90,000 kg’ın üstünde olmamasını, satış hedeflerinin aylık 3,600,000 $’ın üstünde olmasını, müşterilerine de aylık taleplerinin üzerinde ürün ihraç etmeyi ve çalıştıracağı aylık işgücü saatini de 3,790,800 dakikanın az olmasını hedeflemektedir.

Üst yönetimin, bu hedeflerini karşılayan Eşit Ağırlıklı Çok Hedefli Doğrusal Programlama modeli kurularak aşağıda verilmiştir.

MODEL:

Min $= d_1^- + d_2^+ + d_3^- + d_4^- + d_5^- + d_6^- + d_7^- + d_8^- + d_9^- + d_{10}^- + d_{11}^- + d_{12}^- + d_{13}^- + d_{14}^- + d_{15}^- + d_{16}^- + d_{17}^- + d_{18}^- + d_{19}^+$

Kar Hedefi ;

$1{,}4x_1 + 2{,}2x_2 + 1{,}2x_3 + 1{,}2x_4 + 0{,}8x_5 + 2{,}2x_6 + 2{,}1x_7 + 1{,}9x_8 + 1{,}9x_9 + 2{,}5x_{10} + 2{,}8x_{12} + 2x_{13} + 3x_{14} + 3{,}3x_{15} \geq 650{,}000$

Fire Hedefi ;

$0{,}44x_1 + 0{,}65x_2 + 0{,}64x_3 + 0{,}10x_4 + 0{,}60x_5 + 0{,}74x_6 + 0{,}105x_7 + 0{,}192x_8 + 0{,}260x_9 + 0{,}42x_{10} + 0{,}79x_{11} + 0{,}23x_{12} + 0{,}68x_{13} + 0{,}48x_{14} + 0{,}55x15 \leq 90{,}000$

Satış Hedefi ;

$4{,}67x_1 + 7{,}07x_2 + 4{,}82x_3 + 3{,}6x_4 + 3{,}6x_5 + 7{,}2x_6 + 10{,}17x_7 + 7{,}2x_8 + 8{,}98x_9 + 7{,}18x_{10} + 9x_{11} + 4{,}80x_{12} + 5{,}99x_{13} + 10{,}77x_{14} + 9{,}58x_{15} \leq 3{,}600{,}000$

Talep Hedefi ;

$x_1 = 45636$

$x_2 = 38171$

$x_3 = 33229$

$x_4 = 30919$

$x_5 = 4050$

$x_6 = 12627$

$x_7 = 14434$

$x_8 = 7550$

$x_9 = 16837$

$x_{10} = 37748$

$x_{11} = 17055$

$x_{12} = 26024$

$x_{13} = 3375$

$x_{14} = 2965$

$x_{15} = 1011$

İşgücü Hedefi ;

$7x_1 + 15x_2 + 8x_3 + 7x_4 + 6x_5 + 9x_6 + 14x_7 + 11x_8 + 19x_9 + 7x_{10} + 7x_{11} + 8x_{12} + 5x_{13} + 18x_{14} + 11x_{15} \leq 3{,}790{,}800$

Penye;

$0{,}145x_1 + 0{,}216x_2 + 0{,}212x_3 + 0{,}212x_4 + 0{,}179x_5 + 0{,}328x_8 + 0{,}218x_9 + 0{,}147x_{12} + 0{,}116x_{13} \leq 530{,}185$

Çift iplik ;

$0{,}550x_7 \leq 20{,}2145$ kg

3 iplik ;

$0{,}709x_{11} \leq 23{,}355$ kg

Ribana ;

$0{,}149x_{10} + 0{,}145x_{14} + 0{,}210x_{15} \leq 175{,}172$ kg

Lacost ;

$0{,}243x_6 \leq 78{,}880$ kg

3 Overlok ;

$0,035x_1 + 0,040x_2 + 0,094x_3 + 0,011x_4 + 0,109x_5 + 0,020x_7 + 0,005x_8 + 0,078x_9 + 0,045x_{10} + 0,166x_{11} + 0,077x_{12} + 0,040x_{14} + 0,191x_{15} \leq 731,203$ dakika

Tek İğne Makine Kısıtlayıcısı;

$00,008x_1 + 0,055x_2 + 0,098x_3 + 0,086x_4 + 0,032x_5 + 0,380x_6 + 0,371x_7 + 0,142x_8 + 0,214x_9 + 0,103x_{10} + 0,103x_{11} + 0,063x_{12} + 0,041x_{13} + 0,098x_{14} + 0,150x_{15} \leq 1,077,039$ dakika

4 Ovelok ;

$0,042x_1 + 0,041x_2 + 0,054x_3 + 0,028x_4 + 0,056x_5 + 0,063x_6 + 0,085x_7 + 0,079x_8 + 0,131x_9 + 0,071x_{10} + 0,137x_{11} + 0,050x_{12} + 0,021x_{13} + 0,054x_{14} + 0,023x_{15} \leq 587,574$ dakika

Reçme ;

$0,048x_1 + 0,077x_2 + 0,099x_3 + 0,062x_4 + 0,068x_5 + 0,076x_6 + 0,043x_7 + 0,059x_8 + 0,171x_9 + 0,031x_{10} + 0,027x_{11} + 0,035x_{12} + 0,052x_{13} + 0,056x_{14} + 0,158x_{15} \leq 1,044,576$ dakika

Düğme;

$0,009x_4 + 0,019x_5 \leq 383,011$ dakika

Punterez ;

$0,02x_7 + 0,020x_9 + 0,014x_{11} + 0,021x_{15} \leq 452,650$ dakika

Ütü ;

$0,024x_1 + 0,029x_2 + 0,008x_3 + 0,017x_4 + 0,023x_6 + 0,018x_7 + 0,034x_8 + 0,054x_9 + 0,051x_{11} + 0,023x_{12} + 0,012x_{13} + 0,025x_{14} + 0,024x_{15} \leq 522,288$ dakika

El işi;

$0,087x_1 + 0,089x_2 + 0,159x_3 + 0,136x_4 + 0,149x_5 + 0,314x_6 + 0,159x_7 + 0,089x_8 + 0,516x_9 + 0,122x_{10} + 0,270x_{11} + 0,079x_{12} + 0,093x_{13} + 0,139x_{14} + 0,133x_{15} \leq 9,627,196$ dakika

Modelde yer alan tüm karar değişkenleri ve sapma değişkenleri ≥ 0

Satışların ağırlıklandırıldığı model, LINGO paket programı ile çözülmüş olup modelin çözümünün sonuçları EK 4'de, ekran görüntüleri ise EK 3'da verilmiştir.

Bilgisayar çıktılarına göre bu modelin optimal çözümüne 20. Adımda ulaşılmıştır.

Amaç fonksiyonun toplam değeri 248,602.9'dir.Amaç fonksiyonun toplam değerinde ürünlerin ölçü birimi (adet), kesim kumaş fireleri (kg), işgücü çalışma saati (dakika), satışlar ve kar ($) içermektedir. Okuyucu tarafından model çözümünün daha iyi anlaşılabilmesi için karar değişkenleri ile sapma değişkenlerinin çözüm değerleri aşağıda verilmiştir.

$x_1 = 45636$, $x_4 = 30919$, $x_6 = 12627$,$x_7 = 14434$, $x_{10} = 370176.5$, $x_{12} = 11172.38$, $x_{13} = 3375$, $x_{15} = 1011$

$x_2 = x_3 = x_5 = x_8 = x_9 = x_{14} = 0$

Üst yönetimin hedefini karşılamak için sıfır yaka uzun kollu t-shirt ürününden aylık 45,636 adet, askılı bayan gecelik ürününden aylık 30,919 adet, kısa kollu gecelikten aylık 12,627 adet, erkek capri pantolondan aylık 14,434 adet, erkek pantolon miktarından aylık 370,176.5 adet, cepli yelek miktarından aylık 11,172.38 adet, kayık yaka sweet miktarından aylık 3,375 adet, uzun kollu sweet miktarından ise 1,011 adet üretilmelidir. Diğer ürünlerden yani ; cepli t-shirt, kolsuz t-shirt, askılı erkek atlet, erkek pijama takımı, polo yaka sweet, bayan atletten ise hiç üretim yapılmayacaktır.

Sapma değişkenlerin değeri ise aşağıda verilmiştir:

$d_1^- = d_2^- = d_3^- = d_4^- = d_7^- = d_9^- = d_{10}^- = d_{13}^- = d_{14}^- = d_{16}^- = d_{18}^- = d_{19}^- = d_3^+ = d_4^+ = d_5^+ = d_6^+ = d_7^+ = d_8^+ = d_9^+ = d_{10}^+ = d_{11}^+ = d_{12}^+ = d_{15}^+ = d_{16}^+ = d_{17}^+ = d_{18}^+ = d_{19}^+ = 0$

$d_2^+ = 130949.3$

$d_5^- = 38171$

$d_6^- = 33229$

$d_8^- = 4050$

$d_{11}^- = 7550$

$d_{12}^- = 16837$

$d_{15}^- = 14851.62$

$d_{17}^- = 2965$

ŞI İ

5 İ

Günümüz rekabet koşullarında işletmelerin karlı bir şekilde büyüyebilmeleri ve diğer işletmelerle rekabet edebilmeleri için üretim ve hizmetlerini sistemli bir şekilde planlamaları gerekir. Ayrıca, işletmeleri başarıya götüren doğru hedeflerinde olması gerekir. Belirlenen hedeflerin optimal bir şekilde gerçekleşebilmesi için ulaşılmak istenen hedeflere ulaşmada nasıl bir yol izlenmesi gerektiği ise muhakkak ki bilinmelidir. İşte çalışmamızın amacı da işletmelerin hedeflerine ulaşmada en önemli Doğrusal Hedef Programlama modelinin karar vericiler için önemini açıklanmaya çalışıldı.

Giriş ve sonuç dışında 4 bölümden oluşan çalışmamızın ilk üç bölümü çalışmamızın teori kısmını dördüncü bölüm ise uygulama kısmını içermektedir.

Yöneylem araştırmasını anlattığımız birici bölümde; yöneylemin tanımları, tarihsel gelişimi, özellikleri, uygulama alanları, avantajları, türleri ayrıntılı olarak ele alınmıştır. Çalışmanın ikinci bölümünde; konumuzun temelini oluşturan Doğrusal Programlamanın tanımı, özellikleri, tarihsel gelişimi, türleri, avantajları, dezavantajları, varsayımları, kullanılan yöntemler örneklerle açıklanmış olup yine bu bölümde Doğrusal Programlamanın en önemli yöntemi olan simpleks çözüm yönteminden ayrıntılı olarak bahsedilerek, çözümü adım adım açıklanmıştır.

Çalışmanın üçüncü bölümü ise, ana konu olan Hedef Programlamanın tanımı, tarihsel gelişimi, varsayımları ve içerdiği terimler ile karar vericilerin hedeflerini belirlemede kullanabileceği Doğrusal Hedef Programlama türleri ele alınmıştır. Ayrıca, bu bölümde seçtiğimiz model Doğrusal Hedef Programlama modeli olduğu için klasik bir Doğrusal Programlama modelini oluştururken izlenen adımlar açıklanmıştır. Yine bu bölümde Hedef Programalama ile yapılan önceki çalışmalardan bahsedilip, Hedef programlamanın ilkeleri ve uygulama alanları ele alınmıştır. İşletme için önce Doğrusal Programlama modeli kurularak, LİNGO paket programı ile çözülmeye çalışılmıştır.

Çalışmanın uygulama kısmını oluşturan dördüncü bölümde, öncelikle işletme tanıtılarak üretim süreci ve iş akışın nasıl olduğu açıklanmıştır. Yine bu bölümde, işletmemizin Doğrusal Hedef Programlama modeli kurularak Ağırlıklı ve Eşit ağırlıklı modelleri kurulmuştur. Bu iki tür için Doğrusal Hedef Programlamanın verdiği sonuçlar açıklanmaya çalışılacaktır.

Siparişlerin ağırlıklandırıldığı modelde fire, satış, aylık üretilecek polo yaka sweet miktarı dışında kalan 16 hedef gerçekleştirilmiştir. Bu hedeflerden kar, 11 nolu talep hedefi aşılmıştır. Bu hedefin sapma değişkenlerinin eksi yönündeki değeri 0 (sıfır), artı yöndeki değeri 15885.76 çıkmış olup talep hedefi aşılmıştır.

Eşit Ağırlıklı model öncelikle yapısal kısıtlayıcıları sağlamakta, daha sonra da amaç fonksiyonunda belirtilen istenmeyen sapma değişken değerlerinin toplamını minimum kılmaktadır. Bu modelin amaç fonksiyonu incelendiğinde, istenmeyen sapma değişkenlerinin farklı birimlerde olduğu görülmektedir. Model farklı birimlerden oluşan bir fonksiyonun toplamını minimize etmeye çalışmakta dolayısıyla birim farklılıkları modelin çözüm sonucunu etkilememektedir. Bu farklılıkları yok etmek amaç fonksiyonu hedeflerinde ağırlıklandırma işlemi ile üstesinden gelinebilir.

Eşit ağırlıklı modelin çözümü ile 19 hedefe ulaşılmak istenmiş, birbiri ile çatışma halindeki bu hedeflerden 8 gerçekleştirilebilmiştir. Bu hedeflerden 7 tanesi tam olarak karşılanmış olur geriye kalan 1 hedefte hedef değerinden sapmalar olmuştur. Ek 5'te her iki yöntem için elde edilen sonuçlar karşılaştırılıp grafiklendirilmiştir.

Sonuç olarak ortaya koyduğumuz bu çalışma, teorik olarak Yöneylem Araştırmasının önemini açıkladığı gibi, çok amaçlı karar vermede kullanılan Hedef Programlamanın işletmeler için değeri de açıklanmaya çalışılmıştır.

YARARLANILAN KAYNAKLAR

Acar, Nesime (1998)**, Üretim Planlaması Yöntem ve Uygulamaları**, Ankara: Milli Prodüktivite Merkezi Yayınları

Alan, Mehmet Ali ve Yeşilyurt, Cavit (2004), **Doğrusal Programlama Problemlerinin Excel İle Çözümü**, Eskişehir: Cumhuriyet Üniversitesi İ.İ.B.F., Cilt 5, Sayı 1.

——— (2003), **Fen Liselerinin 2002 Yılı Göreceli Etkinliğinin Veri Zarflama Analizi (VZA) ile Ölçülmesi**, Sivas: Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, cilt 4, sayı 2, ss.91-104.

Altay, Emel, (1990), **Doğrusal Programa Tekniği ile Ürün Karması Probleminin Çözümü**, Ankara.

Aygüneş, H., Binay,Soner, Çetin,A., Oral,H. (2001). **Yöneylem Araştırması Ders Kitabı,** Ankara: Kara Harp Okulu Matbaası.

Avralıoğlu, Zeki, (1981), **Doğrusal Programlama ve Tarımsal İşletmelerde Bir Uygulama,** İstanbul.

Büyükkeklik, Arzum, (2006), **Sayısal Karar Yöntemlerinde Matematiksel Programlama Modelleri, Yayımlanmamış Seminer Notları**, Niğde Üniversitesi, Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı, Doktora Programı, Niğde.

Charnes, Abraham ve Cooper, William (1961), **Managment Models and Industrial Application of Lineer Programming,** New York.

Churchman C.West, Ackoff, L. Russel, (1957), **İntroduction to Operations Research,** Jhon Willey And Sons., Inc., New York.

Cinemre Nalan, (2003), **Yöneylem Arastırması**, , İstanbul: Beta Basım Yayın Dağıtım.

Çınar, Mehmet, (1990), **Yönetsel Kararlara İlişkin Sayısal Yöntemler**, Erciyes Üniversitesi Yayını No: 8, Kayseri.

Doğrusöz, Halim, (1979), **Yöneylem Araştırması Bildiriler'79**, Ankara.

Doğan, İbrahim, (1995), **Yöneylem Araştırması Teknikleri ve İşletme Uygulamaları**, Bilim, İstanbul: Teknik Kitaı

Esin, Alptekin, (2003), **Yöneylem Araştırmalarında Yararlanılan Karar Yöntemleri**, Ankara: Gazi Kitapevi.

Gülenç, İ., Figen ve Karabulut, Bilge (2005), **Doğrusal Programlama ile Bir Üretim Planlama Probleminin Çözümü**, Kocaeli Üniversitesi, Sosyal Bilimler Enstitüsü Dergisi, Sayı:9, Kocaeli.

Gürel, Okan, (1996), **Lineer Programlama ve Dinamik Programlamaya Giriş**, İstanbul.

Halaç, Osman, (2001), **Kantitatif Karar Verme Teknikleri- Yöneylem Araştırmasına Giriş**, , Bursa: Alfa Kitapevi.

Hedef Programlama http:\\www.ozyazilim.com. (09.10.2008)

Hedef Programlama http:\\www.baskent.edu.tr. (27.11.2008)

Hedef Programlama http:\\www.kouemk.com/makale. (15.12.2008)

Hedef Programlama http:\\en.wikipedia.org/wiki/Goal Programming.(05.09.2009).

Hillier, F. S.Lieberman, G.J., (1986), **İntroduction To Operations Research**, Forth Edtion, Holden, California, America.

Ignizio, J.P. (1976), **Goal Programming and Extensions Lexington Book**, London.

——— (1978), **A Review of Goal Programming A Tool For Multiobjective Analysis**, The Journel of The Operational Research Society, No: 11, London.

Kara, İmdat, (1991), **Doğrusal Programlam**a, Bilim Teknik Yayınevi, Eskişehir.

Kazan, Ayşe, (1997), **Türkiye Ekonomisi için Ekonometrik Model Denemesi ve Ekonometrik Modellerin Hedef Programlama Modelleri ile Eşanlı Kullanımı**, Doktora Tezi, Ankara.

Kılıçbay, Ahmet (1970), **Kantitatif İktisat Teorisi ve Politikası,** , İstanbul: **Sermet Yayınevi**

Kobu, Bülent (1997), **İşletme Matematiği**, 6. Baskı, İstanbul: Avcıol Basınyayın

Kocadağlı, Ozan, (1997), **Doğrusal Hedef Programlama ile Bütçeleme**, İstanbul.

Kuruüzüm, Ayşe, (1998), **Karara Destek Sisteminde Çok Amaçlı Yöntemler**, Antalya: Akdeniz Üniversitesi Basımevi Yayın No:72

Lee, Sang, (1972), **Goal Programming for Decision Analysis**, Philadelphia.

Özgüven, Cemal, (2003**), Programlama ve Uzantıları**, Detay Yayınevi, Ankara.

Öztürk, Ahmet, (2005), **Yöneylem Araştırması, Genişletilmiş 10. Baskı**, Ekin Kitapevi, Bursa.

Schmidt, Eric, G., (1993), **Introductory Management Science**, USA: Prentice- Hall.

Serper, Ö. (1982), **Doğrusal Programlama**, B.İ.T.İ. İşletme Fakültesi Yayını No:15, Bursa.

Sucu, Mehmet, (2003), **Çok Amaçlı Programlama ve Planlama Ders Notları**.

Taha, A. Hamdy, (2000**), Yöneylem Araştırması**, Ş. Alp Boray- Şakir Esnaf (Çevirenler), 6. Basımdan Çeviri, İstanbul: Litaratür Yayıncılık

Tanrıkurt, Mehmet, (1979**), Doğrusal Programlama**, Ankara.

Timor, M., (2001), "Yöneylem Araştırma ve İşletme Uygulamaları", **İ.Ü. İşletme Fakültesi Yayınları, s.1-2**, İstanbul.

Toklu, Bilal, (1985), **Doğrusal Programlamanın Üretim Planlamasına Uygulanması,** Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü.

Tulunay, Yılmaz, (1987), **Matematik Programlama ve İşletme Uygulamaları**, İstanbul: Bayrak Matbaacılık.

Wise, K.P. (2000), **Goal Programming as a Solution Technique for the Acquisitions Allocation Problem**, Library & İnformation Science Research Volume 22, Issue 2.

Yöneylem Araştırması www.business.cu.edu.tr. (11.08.2008)

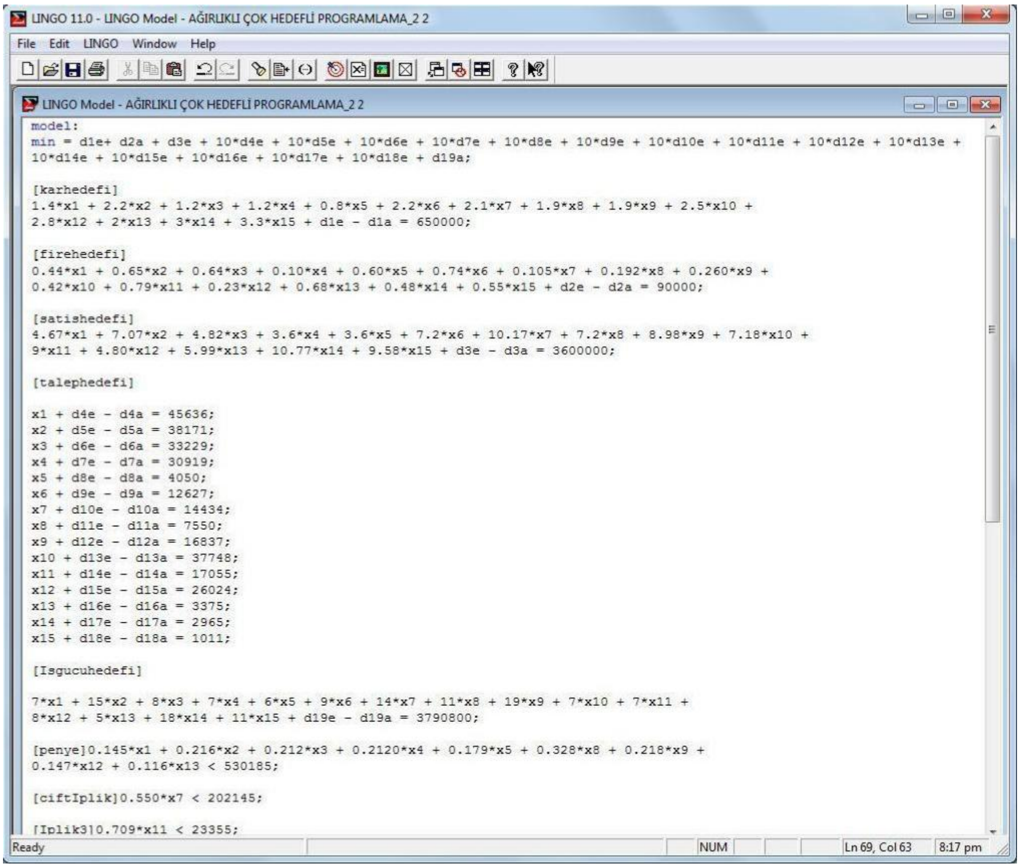
Yöneylem Araştırması www.özyazilim.com. (02.09.2008)

Zengin, Hilmi, (1987), **Tütkiye'de Paketli Çay Dağıtımının Optimizasyonu (Ulaştırma Modeli)**,Trabzon.

Zionts, Stanley, (1989), "**Multiple Criteria Mathematical Programming:An Updated Overview And Several Approaches**", Multiple Criteria Decision Making and Risk Analysis Using Microcomputers,(ed: Birsen Karpak), Springer-Verlag, Berlin.

EK -1

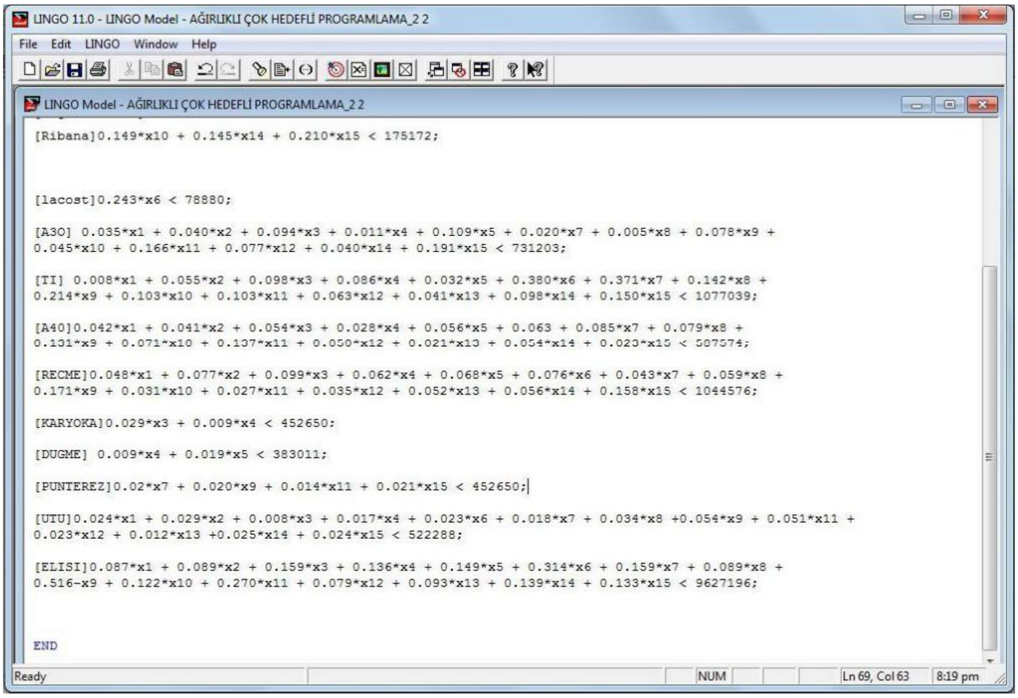
AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA MODELİNİN LİNGO BİLGİSAYAR PAKET PROGRAMININ EKRAN GÖRÜNTÜLERİ



LINGO 11.0 - LINGO Model - AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA_2 2

```
model:
min = d1e+ d2a + d3e + 10*d4e + 10*d5e + 10*d6e + 10*d7e + 10*d8e + 10*d9e + 10*d10e + 10*d11e + 10*d12e + 10*d13e +
10*d14e + 10*d15e + 10*d16e + 10*d17e + 10*d18e + d19a;

[karhedefi]
1.4*x1 + 2.2*x2 + 1.2*x3 + 1.2*x4 + 0.8*x5 + 2.2*x6 + 2.1*x7 + 1.9*x8 + 1.9*x9 + 2.5*x10 +
2.8*x12 + 2*x13 + 3*x14 + 3.3*x15 + d1e - d1a = 650000;

[firehedefi]
0.44*x1 + 0.65*x2 + 0.64*x3 + 0.10*x4 + 0.60*x5 + 0.74*x6 + 0.105*x7 + 0.192*x8 + 0.260*x9 +
0.42*x10 + 0.79*x11 + 0.23*x12 + 0.68*x13 + 0.48*x14 + 0.55*x15 + d2e - d2a = 90000;

[satishedefi]
4.67*x1 + 7.07*x2 + 4.82*x3 + 3.6*x4 + 3.6*x5 + 7.2*x6 + 10.17*x7 + 7.2*x8 + 8.98*x9 + 7.18*x10 +
9*x11 + 4.80*x12 + 5.99*x13 + 10.77*x14 + 9.58*x15 + d3e - d3a = 3600000;

[talephedefi]

x1 + d4e - d4a = 45636;
x2 + d5e - d5a = 38171;
x3 + d6e - d6a = 33229;
x4 + d7e - d7a = 30919;
x5 + d8e - d8a = 4050;
x6 + d9e - d9a = 12627;
x7 + d10e - d10a = 14434;
x8 + d11e - d11a = 7550;
x9 + d12e - d12a = 16837;
x10 + d13e - d13a = 37748;
x11 + d14e - d14a = 17055;
x12 + d15e - d15a = 26024;
x13 + d16e - d16a = 3375;
x14 + d17e - d17a = 2965;
x15 + d18e - d18a = 1011;

[Isgucuhedefi]

7*x1 + 15*x2 + 8*x3 + 7*x4 + 6*x5 + 9*x6 + 14*x7 + 11*x8 + 19*x9 + 7*x10 + 7*x11 +
8*x12 + 5*x13 + 18*x14 + 11*x15 + d19e - d19a = 3790800;

[penye]0.145*x1 + 0.216*x2 + 0.212*x3 + 0.2120*x4 + 0.179*x5 + 0.328*x8 + 0.218*x9 +
0.147*x12 + 0.116*x13 < 530185;

[ciftIplik]0.550*x7 < 202145;

[Iplik3]0.709*x11 < 23355;
```

Ready | NUM | Ln 69, Col 63 | 8:17 pm



LINGO 11.0 - LINGO Model - AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA_2 2

```
[Ribana]0.149*x10 + 0.145*x14 + 0.210*x15 < 175172;



[lacost]0.243*x6 < 78880;

[A30] 0.035*x1 + 0.040*x2 + 0.094*x3 + 0.011*x4 + 0.109*x5 + 0.020*x7 + 0.005*x8 + 0.078*x9 +
0.045*x10 + 0.166*x11 + 0.077*x12 + 0.040*x14 + 0.191*x15 < 731203;

[TI] 0.008*x1 + 0.055*x2 + 0.098*x3 + 0.086*x4 + 0.032*x5 + 0.380*x6 + 0.371*x7 + 0.142*x8 +
0.214*x9 + 0.103*x10 + 0.103*x11 + 0.063*x12 + 0.041*x13 + 0.098*x14 + 0.150*x15 < 1077039;

[A40]0.042*x1 + 0.041*x2 + 0.054*x3 + 0.028*x4 + 0.056*x5 + 0.063 + 0.085*x7 + 0.079*x8 +
0.131*x9 + 0.071*x10 + 0.137*x11 + 0.050*x12 + 0.021*x13 + 0.054*x14 + 0.023*x15 < 507574;

[RECME]0.048*x1 + 0.077*x2 + 0.099*x3 + 0.062*x4 + 0.068*x5 + 0.076*x6 + 0.043*x7 + 0.059*x8 +
0.171*x9 + 0.031*x10 + 0.027*x11 + 0.035*x12 + 0.052*x13 + 0.056*x14 + 0.158*x15 < 1044576;

[KARYOKA]0.029*x3 + 0.009*x4 < 452650;

[DUGME] 0.009*x4 + 0.019*x5 < 383011;

[PUNTEREZ]0.02*x7 + 0.020*x9 + 0.014*x11 + 0.021*x15 < 452650;

[UTU]0.024*x1 + 0.029*x2 + 0.008*x3 + 0.017*x4 + 0.023*x6 + 0.018*x7 + 0.034*x8 +0.054*x9 + 0.051*x11 +
0.023*x12 + 0.012*x13 +0.025*x14 + 0.024*x15 < 522288;

[ELISI]0.087*x1 + 0.089*x2 + 0.159*x3 + 0.136*x4 + 0.149*x5 + 0.314*x6 + 0.159*x7 + 0.089*x8 +
0.516-x9 + 0.122*x10 + 0.270*x11 + 0.079*x12 + 0.093*x13 + 0.139*x14 + 0.133*x15 < 9627196;


END
```

Ready | NUM | Ln 69, Col 63 | 8:19 pm

EK -2

AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA MODELİNİN ÇÖZÜMÜ

```
Global optimal solution found.
  Objective value:                              697708.7
  Infeasibilities:                              0.000000
  Total solver iterations:                            18
```

Variable	Value	Reduced Cost
D1E	0.000000	1.000000
D2A	248723.0	0.000000
D3E	0.000000	0.5175292E-01
D4E	0.000000	6.988314
D5E	0.000000	1.054107
D6E	0.000000	5.930551
D7E	0.000000	6.313689
D8E	0.000000	6.813689
D9E	0.000000	7.087379
D10E	0.000000	5.538673
D11E	0.000000	5.635379
D12E	16837.00	0.000000
D13E	0.000000	9.388414
D14E	0.000000	10.00000
D15E	0.000000	6.321586
D16E	0.000000	10.00000
D17E	0.000000	1.732621
D18E	0.000000	7.534207
D19A	280615.8	0.000000
X1	45636.00	0.000000
X2	38171.00	0.000000
X3	33229.00	0.000000
X4	30919.00	0.000000
X5	4050.000	0.000000
X6	12627.00	0.000000
X7	14434.00	0.000000
X8	7550.000	0.000000
X9	0.000000	0.7447412
X10	37748.00	0.000000
X12	26024.00	0.000000
X13	301312.1	0.000000
X14	2965.000	0.000000
X15	1011.000	0.000000
D1A	432612.7	0.000000
X11	32940.76	0.000000
D2E	0.000000	1.000000
D3A	0.000000	0.9482471
D4A	0.000000	3.011686
D5A	0.000000	8.945893
D6A	0.000000	4.069449
D7A	0.000000	3.686311
D8A	0.000000	3.186311
D9A	0.000000	2.912621
D10A	0.000000	4.461327
D11A	0.000000	4.364621
D12A	0.000000	10.00000
D13A	0.000000	0.6115860
D14A	15885.76	0.000000
D15A	0.000000	3.678414
D16A	297937.1	0.000000

D17A	0.000000	8.267379
D18A	0.000000	2.465793
D19E	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	697708.7	-1.000000
KARHEDEFI	0.000000	0.000000
FIREHEDEFI	0.000000	1.000000
SATISHEDEFI	0.000000	-0.9482471
TALEPHEDEFI	0.000000	-3.011686
6	0.000000	-8.945893
7	0.000000	-4.069449
8	0.000000	-3.686311
9	0.000000	-3.186311
10	0.000000	-2.912621
11	0.000000	-4.461327
12	0.000000	-4.364621
13	0.000000	-10.00000
14	0.000000	-0.6115860
15	0.000000	0.000000
16	0.000000	-3.678414
17	0.000000	0.000000
18	0.000000	-8.267379
19	0.000000	-2.465793
ISGUCUHEDEFI	0.000000	1.000000
PENYE	459744.4	0.000000
CIFTIPLIK	194206.3	0.000000
IPLIK3	0.000000	1.049681
RIBANA	168905.3	0.000000
LACOST	75811.64	0.000000
A30	714365.0	0.000000
TI	1035588.	0.000000
A40	564376.9	0.000000
RECME	1012974.	0.000000
KARYOKA	451408.1	0.000000
DUGME	382655.8	0.000000
PUNTEREZ	451878.9	0.000000
UTU	512494.7	0.000000
ELISI	9558680.	0.000000

EK -3

EŞİT AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA MODELİNİN LİNGO BİLGİSAYAR PAKET PROGRAMININ EKRAN GÖRÜNTÜLERİ

LINGO 11.0 - [LINGO Model - EŞİT AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA_2]

```
model:
min = d1e+ d2a + d3e + 10*d4e + d5e + d6e + d7e + d8e + d9e + d10e + d11e + d12e + d13e +
d14e + d15e + d16e + d17e + d18e + d19a;

[karhedefi]
1.4*x1 + 2.2*x2 + 1.2*x3 + 1.2*x4 + 0.8*x5 + 2.2*x6 + 2.1*x7 + 1.9*x8 + 1.9*x9 + 2.5*x10 +
2.8*x12 + 2*x13 + 3*x14 + 3.3*x15 + d1e - d1a = 650000;

[firehedefi]
0.44*x1 + 0.65*x2 + 0.64*x3 + 0.10*x4 + 0.60*x5 + 0.74*x6 + 0.105*x7 + 0.192*x8 + 0.260*x9 +
0.42*x10 + 0.79*x11 + 0.23*x12 + 0.68*x13 + 0.48*x14 + 0.55*x15 + d2e - d2a = 90000;

[satishedefi]
4.67*x1 + 7.07*x2 + 4.82*x3 + 3.6*x4 + 3.6*x5 + 7.2*x6 + 10.17*x7 + 7.2*x8 + 8.98*x9 + 7.18*x10 +
9*x11 + 4.80*x12 + 5.99*x13 + 10.77*x14 + 9.58*x15 + d3e - d3a = 3600000;

[talephedefi]

x1 + d4e - d4a = 45636;
x2 + d5e - d5a = 38171;
x3 + d6e - d6a = 33229;
x4 + d7e - d7a = 30919;
x5 + d8e - d8a = 4050;
x6 + d9e - d9a = 12627;
x7 + d10e - d10a = 14434;
x8 + d11e - d11a = 7550;
x9 + d12e - d12a = 16837;
x10 + d13e - d13a = 37748;
x11 + d14e - d14a = 17055;
x12 + d15e - d15a = 26024;
x13 + d16e - d16a = 3375;
x14 + d17e - d17a = 2965;
x15 + d18e - d18a = 1011;

[Isgucuhedefi]

7*x1 + 15*x2 + 8*x3 + 7*x4 + 6*x5 + 9*x6 + 14*x7 + 11*x8 + 19*x9 + 7*x10 + 7*x11 +
8*x12 + 5*x13 + 18*x14 + 11*x15 + d19e - d19a = 3790800;

[penye]0.145*x1 + 0.216*x2 + 0.212*x3 + 0.2120*x4 + 0.179*x5 + 0.328*x8 + 0.218*x9 +
0.147*x12 + 0.116*x13 < 530185;

[ciftIplik]0.550*x7 < 202145;

[Iplik3]0.709*x11 < 23355;

[Ribana]0.149*x10 + 0.145*x14 + 0.210*x15 < 175172;
```

Ready | NUM | Ln 24, Col 24 | 8:28 pm

LINGO 11.0 - [LINGO Model - EŞİT AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA_2]

```
[lacost]0.243*x6 < 78880;

[A30] 0.035*x1 + 0.040*x2 + 0.094*x3 + 0.011*x4 + 0.109*x5 + 0.020*x7 + 0.005*x8 + 0.078*x9 +
0.045*x10 + 0.166*x11 + 0.077*x12 + 0.040*x14 + 0.191*x15 < 731203;

[TI] 0.008*x1 + 0.055*x2 + 0.098*x3 + 0.086*x4 + 0.032*x5 + 0.380*x6 + 0.371*x7 + 0.142*x8 +
0.214*x9 + 0.103*x10 + 0.103*x11 + 0.063*x12 + 0.041*x13 + 0.098*x14 + 0.150*x15 < 1077039;

[A40]0.042*x1 + 0.041*x2 + 0.054*x3 + 0.028*x4 + 0.056*x5 + 0.063 + 0.085*x7 + 0.079*x8 +
0.131*x9 + 0.071*x10 + 0.137*x11 + 0.050*x12 + 0.021*x13 + 0.054*x14 + 0.023*x15 < 587574;

[RECME]0.048*x1 + 0.077*x2 + 0.099*x3 + 0.062*x4 + 0.068*x5 + 0.076*x6 + 0.043*x7 + 0.059*x8 +
0.171*x9 + 0.031*x10 + 0.027*x11 + 0.035*x12 + 0.052*x13 + 0.056*x14 + 0.158*x15 < 1044576;

[KARYOKA]0.029*x3 + 0.009*x4 < 452650;

[DUGME] 0.009*x4 + 0.019*x5 < 383011;

[PUNTEREZ]0.02*x7 + 0.020*x9 + 0.014*x11 + 0.021*x15 < 452650;

[UTU]0.024*x1 + 0.029*x2 + 0.008*x3 + 0.017*x4 + 0.023*x6 + 0.018*x7 + 0.034*x8 +0.054*x9 + 0.051*x11 +
0.023*x12 + 0.012*x13 +0.025*x14 + 0.024*x15 < 522288;

[ELISI]0.087*x1 + 0.089*x2 + 0.159*x3 + 0.136*x4 + 0.149*x5 + 0.314*x6 + 0.159*x7 + 0.089*x8 +
0.516-x9 + 0.122*x10 + 0.270*x11 + 0.079*x12 + 0.093*x13 + 0.139*x14 + 0.133*x15 < 9627196;



END
```

Ready | NUM | Ln 46, Col 1 | 8:30 pm

EK -4

EŞİT AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA MODELİNİN ÇÖZÜMÜ

Global optimal solution found.
Objective value: 248602.9
Infeasibilities: 0.000000
Total solver iterations: 20

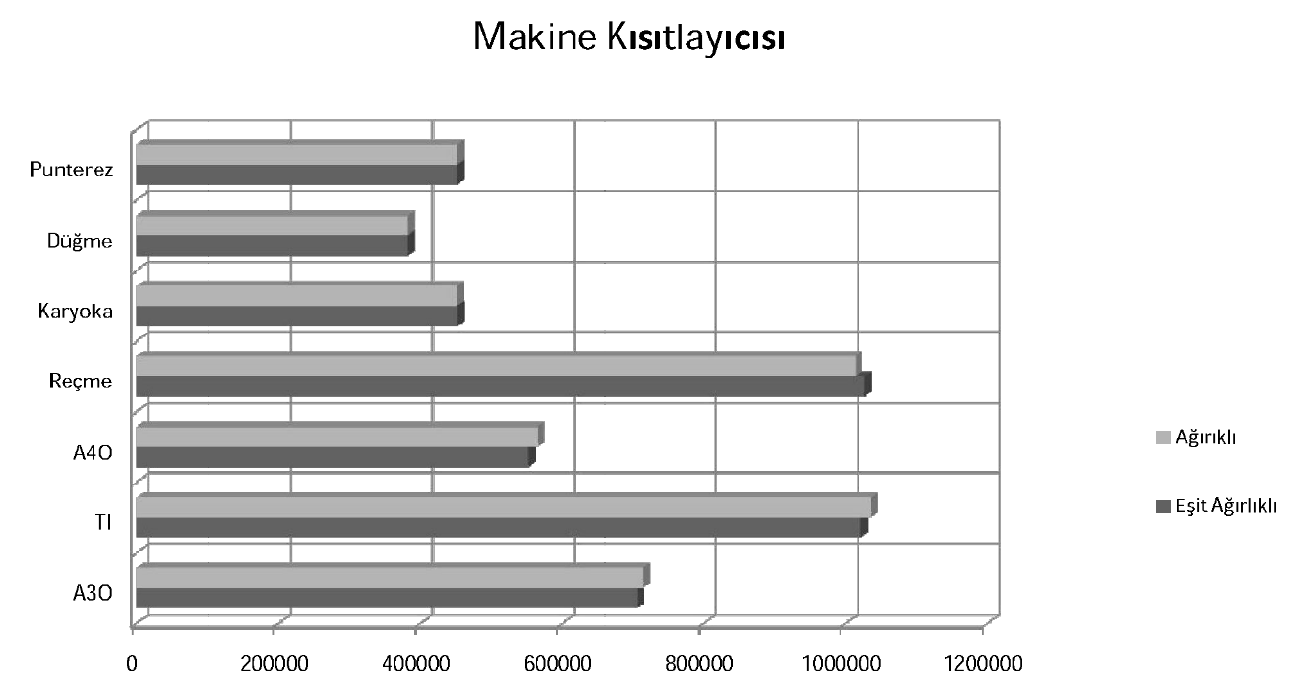
Variable	Value	Reduced Cost
D1E	0.000000	1.000000
D2A	130949.3	0.000000
D3E	0.000000	0.6329698
D4E	0.000000	9.058754
D5E	38171.00	0.000000
D6E	33229.00	0.000000
D7E	0.000000	0.6031879E-02
D8E	4050.000	0.000000
D9E	0.000000	0.5440436E-01
D10E	0.000000	0.1971435
D11E	7550.000	0.000000
D12E	16837.00	0.000000
D13E	0.000000	1.000000
D14E	0.000000	1.000000
D15E	14851.62	0.000000
D16E	0.000000	0.9361703
D17E	2965.000	0.000000
D18E	0.000000	0.4850000
D19A	0.000000	0.6835319
X1	45636.00	0.000000
X2	0.000000	1.802118
X3	0.000000	0.4026594
X4	30919.00	0.000000
X5	0.000000	0.1775000
X6	12627.00	0.000000
X7	14434.00	0.000000
X8	0.000000	0.3053188E-01
X9	0.000000	1.976963
X10	370176.5	0.000000
X12	11172.38	0.000000
X13	3375.000	0.000000
X14	0.000000	1.223511
X15	1011.000	0.000000
D1A	475894.3	0.000000
X11	32940.76	0.000000
D2E	0.000000	1.000000
D3A	0.000000	0.3670302
D4A	0.000000	0.9412458
D5A	0.000000	1.000000
D6A	0.000000	1.000000
D7A	0.000000	0.9939681
D8A	0.000000	1.000000
D9A	0.000000	0.9455956
D10A	0.000000	0.8028565
D11A	0.000000	1.000000
D12A	0.000000	1.000000
D13A	332428.5	0.000000
D14A	15885.76	0.000000
D15A	0.000000	1.000000
D16A	0.000000	0.6382970E-01

D17A	0.000000	1.000000
D18A	0.000000	0.5150000
D19E	0.000000	0.3164681

Row	Slack or Surplus	Dual Price
1	248602.9	-1.000000
KARHEDEFI	0.000000	0.000000
FIREHEDEFI	0.000000	1.000000
SATISHEDEFI	0.000000	-0.3670302
TALEPHEDEFI	0.000000	-0.9412458
6	0.000000	-1.000000
7	0.000000	-1.000000
8	0.000000	-0.9939681
9	0.000000	-1.000000
10	0.000000	-0.9455956
11	0.000000	-0.8028565
12	0.000000	-1.000000
13	0.000000	-1.000000
14	0.000000	0.000000
15	0.000000	0.000000
16	0.000000	-1.000000
17	0.000000	-0.6382970E-01
18	0.000000	-1.000000
19	0.000000	-0.5150000
ISGUCUHEDEFI	0.000000	0.3164681
PENYE	514979.1	0.000000
CIFTIPLIK	194206.3	0.000000
IPLIK3	0.000000	0.4203032
RIBANA	119803.4	0.000000
LACOST	75811.64	0.000000
A30	705797.5	0.000000
TI	1021347.	0.000000
A40	552116.4	0.000000
RECME	1025797.	0.000000
KARYOKA	452371.7	0.000000
DUGME	382732.7	0.000000
PUNTEREZ	451878.9	0.000000
UTU	518115.2	0.000000
ELISI	9557374.	0.000000

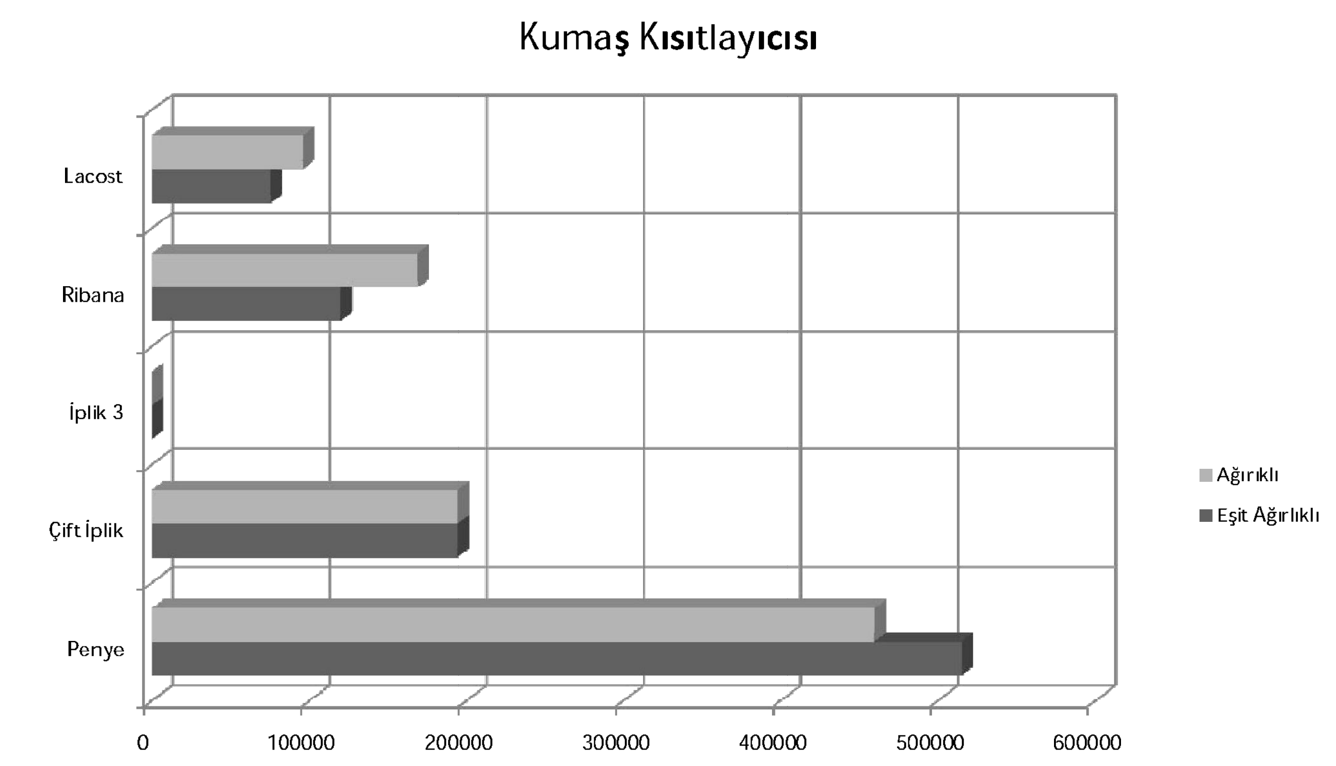
EK -5

EŞİT AĞIRLIKLI VE AĞIRLIKLI ÇOK HEDEFLİ PROGRAMLAMA SONUÇLARININ GRAFİKSEL GÖSTERİMİ



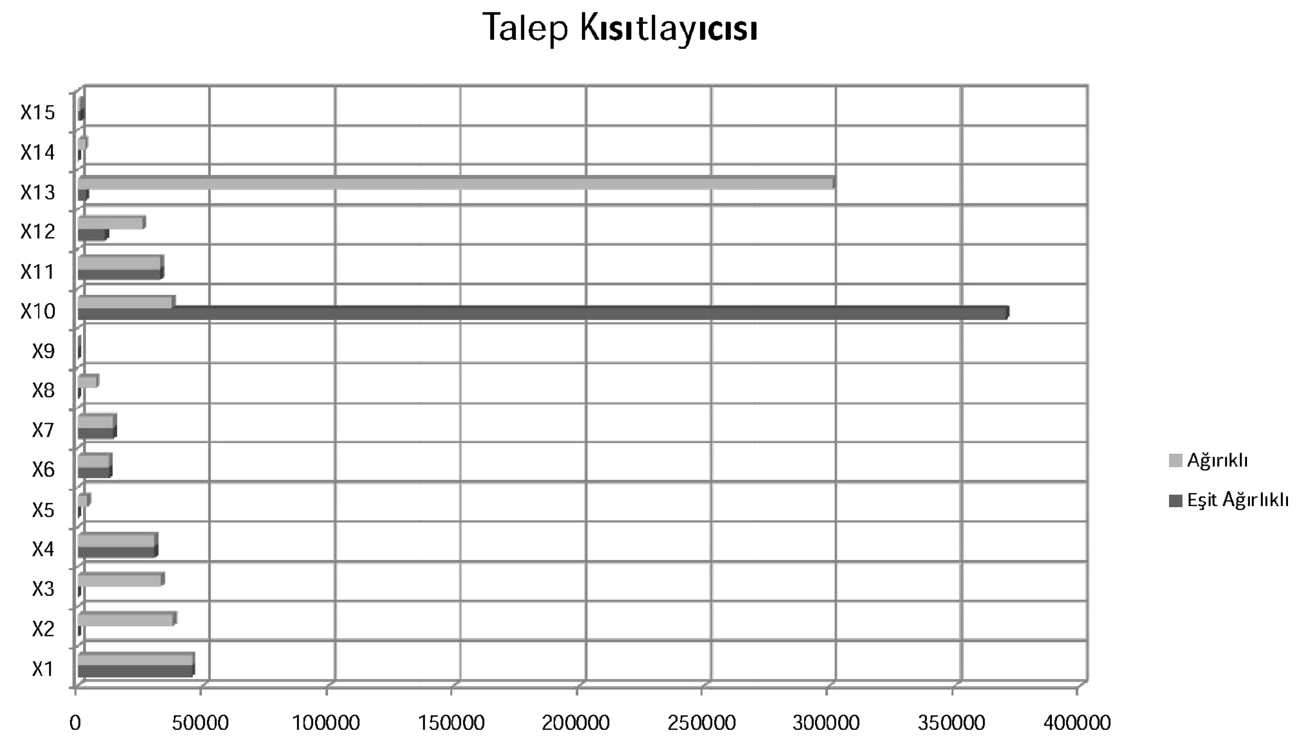
	A3O	TI	A4O	Reçme	Karyoka	Düğme	Punterez
Ağırıklı	714365	1035588	564376,9	1012974	451408,1	382655,8	451878,9
Eşit Ağırlıklı	705797,7	1021347	552116,4	1025797	452371,7	382732,7	451878,9

Ağırlıklı ve Eşit **Ağırlıklı** Makine K**ısıt**lay**ıcı**lar**ını**n Kar**şı**la**ştırı**lması



	Penye	Çift İplik	İplik 3	Ribana	Lacost
Ağırıklı	459744,4	194206,3	0	168905,3	95811,64
Eşit Ağırlıklı	514979,1	194206,3	0	119803,4	75811,64

Ağırlıklı ve Eşit Ağırlıklı Kumaş Kısıtlayıcılarının Karşılaştırılması



	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
Ağırıklı	45636	38171	33229	30919	4050	12627	14434	7550	0	37748	32941	26024	301312	2965	1011
Eşit Ağırlıklı	45636	0	0	30919	0	12627	14434	0	0	370177	32941	11172	3375	0	1011

Ağırlıklı ve Eşit Ağırlıklı Talep Kısıtlayıcılarının Karşılaştırılması

ÖZGEÇMİŞ

Müyesser Candeğer ŞİRİN, 1981 yılında Trabzon'da doğdu. İlk, orta ve lise öğrenimini Trabzon'dae tamamladı. 2000-01 eğitim-öğretim yılında Atatürk Üniversitesi İktisadi ve İdari Bilimler Fakültesi İşletme Bölümü'ne girdi. Temmuz-2005 yılında mezun oldu. 2005 yılında K.T.Ü. Sosyal Bilimleri Enstitüsü Ekonometri Anabilim Dalı'nda yüksek lisans eğitimine başladı. Orta derecede İngilizce bilmektedir.