Karma Tamsayılı Doğrusal Programlama ve Bir Uygulama Denemesi

Fatma TURAN

(Yüksek Lisans Tezi)

Eskişehir, 2002

KARMA TAMSAYILI DOĞRUSAL PROGRAMLAMA VE BİR UYGULAMA DENEMESİ

Fatma TURAN

YÜKSEK LİSANS TEZİ

İşletme Anabilim Dalı

Danışman: Yrd.Doç.Dr.Mahmut ATLAS

Eskişehir

Anadolu Üniversitesi Sosyal Bilimler Enstitüsü

Aralık 2002

YÜKSEK LİSANS TEZ ÖZÜ

KARMA TAMSAYILI DOĞRUSAL PROGRAMLAMA ve BİR UYGULAMA DENEMESİ

Fatma TURAN
İşletme Anabilim Dalı
Anadolu Üniversitesi Sosyal Bilimler Enstitüsü, Aralık 2002
Danışman: Yrd.Doç.Dr.Mahmut ATLAS

Bu çalışmanın ilk bölümünde, doğrusal programlama ve tamsayılı doğrusal programlama tanımlanmıştır. Tamsayılı doğrusal programlama modellerinden bütünüyle tamsayılı doğrusal programlama, sıfır-bir tamsayılı doğrusal programlama ve karma tamsayılı doğrusal programlama teorik olarak incelenmiştir. İkinci bölümde tamsayılı doğrusal programlama çözüm tekniklerinden kesme düzlemi tekniği ve dal-sınır tekniği anlatılmaya çalışılmıştır.

Son bölümde, bir üretim tesisi için karma tamsayılı doğrusal programlama modeli kurularak, QSB+ paket programında dal-sınır tekniğine göre çözüm yapılmıştır. Kağıt ve ambalaj üretiminde kullanılan hammaddelerin optimal bileşimi ile maksimum karı sağlayacak üretim miktarı bulunmaya çalışılmıştır.

ABSTRACT

In the first section of this study, pure linear programming, 0-1 linear programming and mixed integer linear programming is theoretically examined after describing linear programming and integer linear programming. At the following section, the cutting plane methods and branch and bound methods are explained.

At the last section a model of mixed integer linear programming is built for application in a manufacturing plant. Model is solved by QSB+ software that used branch and boud methods. The problem explained the optimal combination of row materials that used in paper-package production and the production quantity that provide the maximum profit.

JÜRİ VE ENSTİTÜ ONAYI

Fatma TURAN'ın "Karma Tamsayılı Doğrusal Programlama ve Bir Uygulama Denemesi" başlıklı tezi **10 Ocak 2003** tarihinde, aşağıdaki jüri tarafından Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca, **İşletme (Sayısal Yöntemler)** Anabilim Dalında, yüksek lisans tezi olarak değerlendirilerek kabul edilmiştir.

İmza

Üye (Tez Danışmanı)	**:**	**Yrd.Doç.Dr.Mahmut ATLAS**	
Üye	**:**	**Prof.Dr.Emel ŞIKLAR**	
Üye	**:**	**Prof.Dr.Yaşar HOŞCAN**	

Prof.Dr.Nurhan AYDIN
Anadolu Üniversitesi
Sosyal Bilimler Enstitüsü Müdürü

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

1. GİRİŞ

Günümüz işletmeleri gerek ulusal, gerekse uluslar arası çok karmaşık ticari ilişki içindedirler. Bu karmaşık ticari ilişkileri yürütecek üretim süreçleri de eskiye göre daha büyük yapıdadır. Bunlara bağlı olarak işletmeler faaliyetlerini sürdürürken karşılaştıkları ticari veya üretim problemleri de büyük ve karmaşık olmaktadır. İşletmelerin böyle problemlerini çözmede Yöneylem Araştırması tekniklerinden doğrusal programlama yardımcı olabilmektedir.

İşletme yöneticileri maksimum kar ya da minimum maliyete ulaşabilmek için ellerinde bulunan işgücü, sermaye, enerji ve hammadde gibi kıt kaynaklardan en iyi şekilde yararlanabilmek için doğrusal programlamadan faydalanmaktadırlar.

Üretim endüstrisinde, çok sayıda ürün aynı üretim ünitelerinden geçerek elde edilir. Bu yüzden hangi ürünlerden ne kadar üretileceğinin bulunması konusu, kapasitenin kullanımı ve maksimum karın sağlanması bakımından önem taşır. Bunun için özellikle üretim sektöründe diğerlerinden çok sayıda doğrusal programlama uygulamaları yapılmıştır. Böyle bir çalışma sonucunda, tezgahların iş yükü, fazla mesai ihtiyacı olup olmadığı, üretilecek ürünler ve miktarları açıklık kazanacaktır. Özellikle çok sayıda standart ürünlerin üretiminde birçok sorunla karşılaşılabilmektedir; bir kısım ürünler fazla üretilirken, bir kısım ürünlerin talepleri karşılanamaz ya da tezgahların bir kısmı fazla mesai yaparken, bir kısmı da iş olmadığından boş bekleyebilmektedir.

Tamsayılı doğrusal programlama ise eldeki kıt kaynakların en iyi dağılımını belirlerken, sonuçların tamsayılı olması için kullanılan matematiksel bir tekniktir. Otomobil, makine v.b üretiminde, üretim sonuçlarının pozitif tamsayılarla ifade edilmesi gerekir. Çünkü bunların tamsayılı olmayan çözümlerinin ekonomik bir anlamı yoktur.

Üretimi karma tamsayılı olan işletmeler için maksimum karlı üretim planının hazırlanmasında yararlanılacak bir yöntem olan karma tamsayılı doğrusal programlama modelinin tanıtımını ve bir kağıt ambalaj işletmesinde uygulamasını yaptığımız bu çalışma üç bölümden oluşmaktadır.

Birinci bölümün ilk kısmında doğrusal programlama ele alınmış, bir doğrusal programlama probleminin nasıl formüle edileceği ve etkin bir hesaplama tekniği olan simpleks çözüm tekniği üzerinde durulmuştur. İkinci kısmında ise tamsayılı doğrusal programlama tanımlanarak matematiksel gösterimleri verilmiştir. Tamsayılı doğrusal programlama karar modelleri anlatılarak uygulama alanlarına değinilmiştir.

Çalışmamızın ikinci bölümünde tamsayılı doğrusal programlama çözüm teknikleri açıklanmaya çalışılmıştır. Tamsayılı modelleme türleri olan bütünüyle tamsayılı, sıfır-bir tamsayılı ve karma tamsayılı doğrusal programlama modelleri için çözüm teknikleri ayrı ayrı teorik olarak incelenmiştir.

Üçüncü bölümde ise bir kağıt ve ambalaj işletmesinde karma tamsayılı doğrusal programlama uygulamasına yer verilmiştir. Bunun için öncelikle işletme hakkında genel bilgi verilerek, üretim çeşitlerinin neler olduğu açıklanmıştır. Daha sonra ise verilerin nasıl elde edildiği anlatılarak modelin değişkenleri tanımlanmış, kısıtlayıcılar ve parametreler belirlenmiş, amaç fonksiyonu oluşturularak model kurulmuştur.

İşletmede 28 değişik ürün, altı üretim ünitesinden geçerek üretilmektedir. Aylak kapasiteyi minimuma indirerek, mevcut şartlar altında karı maksimize edecek aylık üretim planını bulmak için bir karma tamsayılı doğrusal programlama modeli hazırlanmıştır. Modelin çözümü için karma tamsayılı doğrusal programlama tekniklerinden olan dal-sınır tekniği kullanılmıştır. Dal-sınır tekniğine göre çözüm, **QSB+** paket programında 30 iterasyonda gerçekleşmiştir.

Çalışmamız, modelimizin karma tamsayılı doğrusal programlama yöntemi ile çözümünde elde edilen sonuçların özeti ile tamamlanmaktadır.

2. TAMSAYILI PROGRAMLAMA

Günümüzde üretilen birçok mal ve hizmet bölünmezlik özelliği taşımaktadır. Üretilen ürünler bir bütün oluşturmakta, bölündüğünde özelliğini kaybetmektedir. Üretim işletmelerinde üretilen çoğu mal; otomobil, makine parçası, buzdolabı v.s. ancak tamsayı ile ölçülebilmektedir. Ürünleri tamsayı ile ölçülebilen işletmelerin problemlerinin yöneylem araştırması teknikleriyle çözümü araştırılırken bu durum göz önünde tutulması gerekmektedir. Genel olarak yöneylem araştırmasında bu tür problemlerin çözümünde tamsayılı programlama kullanılmaktadır. Yöneylem araştırmasında diğer çözüm tekniklerinde de; dinamik programlama, ulaştırma modelleri v.b. karar değişkenlerinin tamsayılı olması istemine bağlı olarak tamsayılı programlama yaklaşımları kullanılabilmektedir. Ancak tamsayılı programlama denildiğinde daha çok tamsayılı doğrusal programlama akla gelmektedir. Çünkü işletme problemleri çoğunlukla doğrusal yapıdadır. Bu nedenle bu bölümde öncelikle doğrusal programlama kısaca tanıtılarak tamsayılı doğrusal programlama üzerinde durulacaktır.

2.1. Doğrusal Programlama

Para, zaman, makine, insan gücü vb. gibi kaynakların çeşitli koşullar altında maksimum faydayı sağlayacak biçimde kullanılmasını sağlayan tekniğe "Doğrusal Programlama" denir **(Esin, 1983)**.

Doğrusal programlamanın bir başka tanımı ise; "Doğrusal" ve "Programlama" kelimelerin anlamlarından çıkmaktadır: "Doğrusal" kelimesi iki veya daha çok değişken arasındaki aynı yönde ilişkiyi, "Programlama" ise matematiksel tekniklerin kullanılmasıyla eldeki sınırlı kaynakları değerlendiren en iyi çözüme ulaşmasını sağlayan matematiksel bir tekniktir **(Serper ve Gürsakal, 1982)**.

Doğrusal programlama belli bir amacı gerçekleştirmek için sınırlı kaynakların etkin kullanımını ve çeşitli seçenekler arasında en uygun dağılımını sağlayan bir tekniktir **(Sarıaslan, 1986)**.

Doğrusal programlama kimya, kömür, demir çelik, kağıt, reklamcılık ve haberleşme gibi endüstrilerde geniş uygulama alanına sahiptir. Kaynakların optimal dağılımında, ulaştırma sorunlarının çözümünde, beslenme problemlerinin çözümlenmesinde, işletmelerin pazarlama, personel, üretim, yatırım, bütçeleme, sermaye vb. analizlerinde, işletmelerin kuruluş yerlerinin saptanmasında kullanılmaktadır.

Doğrusal programlama, belli doğrusal eşitliklerin veya eşitsizliklerin kısıtlayıcı koşulları altında doğrusal bir amaç fonksiyonunu eniyileme biçiminde tanımlanabilir. Eniyileme, belli bir amaca en az maliyetle ulaşmak veya belli kaynaklarla en büyük ürünü sağlamak anlamına gelir. Bütün ayrıntıları bilinen koşullar altında uygun bir karar alma aracıdır **(Esin, 1983)**.

İkinci dünya savaşı yıllarında askeri problemleri çözmek amacı ile geliştirilen bu teknik, daha sonraları karar verme aracı olarak optimal kaynak dağılım problemlerinin çözümünde yaygın olarak kullanılmıştır. Doğrusal programlama problemi olarak formüle edilebilen optimal kaynak dağılım problemlerinin çözümü için geliştirilen ilk sistematik çalışma 1947 yılında George B. Dantzig ve arkadaşları tarafından yayımlandı. Simpleks tekniği olarak adlandırılan bu çözüm tekniği, günümüzde doğrusal programlama problemlerinin çözümünde kullanılan en etkin tekniktir **(Sarıaslan, 1986)**.

Ayrıca bankacılık, eğitim, petrol işletmesi, sebze meyve satıcılığı gibi farklı endüstrilerde de kullanılan doğrusal programlama 1947 'de G.Danzig 'in etkili bir tekniği olan simpleks algoritmasının gelişmesiyle etkin kullanımı artmıştır **(Winston, 1991)**.

Mal ve hizmet üretimi ile ilgili kararları veren yöneticinin, işletme ile ilgili problemler karşısında akılcı ve uygulanabilir kararlar verebilmesi, problemleri çözmekte kullanacağı yaklaşımlara ve modellere bağlıdır. Çünkü bilimsel yöntemin özünü model kurma oluşturur **(Atlas, 1999)**.

Bilgisayarların yaygın bir kullanım alanına sahip olmasından sonra, endüstri kesimi de karar vermede yararlı bir araç olduğunu gördüğü doğrusal programlama konusuna ilgi duymaya başlamış, makro ve mikro ekonomik faaliyet düzeylerinde başarıyla kullanılabilen bir araç haline getirmiştir **(Doğan, 1995)**.

Doğrusal Programlamanın Varsayımları

Bir karar probleminin doğrusal programlama modeli olarak modellenmesi ve çözülebilmesi için dört ana varsayımı sağlaması gerekmektedir.

- Doğrusallık
- Toplanabilirlik
- Sınırlılık
- Negatif Olmama

Doğrusallık: Amaç fonksiyonu ve kısıtlayıcılar karar değişkenlerinin doğrusal bir bileşeni olarak formüle edilir. Bir anlamda işletmenin girdileri ile çıktıları arasında doğrusal bir ilişki olduğu kabul edilir. Doğrusallık varsayımının sağlanması için, modelde yer alan karar değişkenleri (x_j) 'lerin birinci dereceden ve amaç fonksiyonu katsayıları (c_j) 'lerin sabit olması gerekir **(Atlas, 1988; Öztürk, 1997)**.

Sınırlılık: Üretim sürecinde yer alan kaynakların sonlu olduğu varsayılır.

Toplanabilirlik: Bu varsayım, değişik üretim faaliyetlerine kaynak olan üretim girdileri toplamının, her bir işleme giren üretim girdileri toplamına eşit olduğunu belirtir.

Negatif olmama: Doğrusal programlama problemlerinde yer alan karar değişkenleri, aylak değişkenler ve artık değişkenlerin değerleri sıfır veya sıfırdan büyük olmalıdır. Çünkü ürünlerin sayısı ya da ürünü meydana getiren bileşenler daima pozitif değer alırlar **(Gürdoğan, 1981).**

Doğrusal Programlama Probleminin Matematiksel Yapısı

Matematikte n bilinmeyenli bir doğrusal model en çok n tane birbirinden bağımsız doğrusal denklemlerle çözülebildiği halde, doğrusal programlama ile n tane bilinmeyenli bir doğrusal model n' den daha az denklem yardımıyla çözülebilmektedir **(Esin, 1983).**

Doğrusal programlama probleminde üç temel bileşen vardır;

- Karar Değişkenleri
- Amaç Fonksiyonu
- Kısıtlayıcı Fonksiyonlar

Doğrusal programlamada, amaç fonksiyonu ve kısıtlayıcı fonksiyonların matematiksel ifadesi aşağıdaki işlemlerin yapılmasıyla başlar.

a) Problemin Belirlenmesi

Modelin kurulmasında amaç fonksiyonu belirli sınırlamalar ve şartlar altında sayısal olarak ifade edilebilmelidir. Problemin amacının enküçükleme mi yoksa enbüyükleme mi olacağına karar verilmelidir. Ayrıca hammadde miktarı, talep miktarı, iş günü sayısı, makinelerin kapasiteleri, üretilecek mamullerin birim maliyetleri ya da her bir ürünün satışından firmanın sağlayacağı kar gibi sayısal bilgiler toplanarak, kullanılacak üretim yöntemleri belirlenmelidir **(Gürdoğan, 1981).**

b) Değişkenlerin Belirlenmesi

İşletme problemlerinde genellikle üretim miktarı, makinelerin çalışma süreleri, üretimde kullanılan hammadde miktarları, talep miktarları ve üretim için yapılan masraflar değişken olarak alınır **(Esin, 1983).**

c) Modelin Parametrelerinin Belirlenmesi

Modelde kullanılan parametreler aşağıdaki biçimde belirlenebilir.

- Üretimde kullanılan makinelerin bir birim üretim için çalışması gereken süre ile toplam kullanılabilir makine süresi arasındaki ilişkiden yaralanarak bazı parametreler belirlenebilir.

- Üretim faktörlerinin bileşim oranları olan teknik üretim katsayıları yardımıyla değişkenler arasındaki ilişkiyi kuran parametreler belirlenir.

- Amaç fonksiyonundaki parametreler, problemin türüne göre değişir. Enküçükleme (minimizasyon) problemlerinde değişkenlerin katsayıları birer birimin maliyetini, enbüyükleme (maksimizasyon) problemlerinde, değişkenlerin katsayıları bir birimden elde edilen kar katsayılarıdır **(Esin, 1983).**

Bir doğrusal programlama problemi , açık ve ölçülebilir bir biçimde belirlenen bir doğrusal amaç fonksiyonu ile bu amaç fonksiyonunun gerçekleşme derecesini yani alabileceği sayısal değeri sınırlayan, doğrusal eşitlik ve eşitsizlikler biçiminde ifade edilen kısıtlardan oluşur **(Sarıaslan, 1986)**.

Amaç Fonksiyonu: Doğrusal programlamada amaç, incelenen probleme göre minimizasyon ise üretim ve stok maliyetlerinin minimuma indirmek, maksimizasyon ise eldeki imkanlarla en büyük karı elde etmektir **(Gürdoğan, 1981)**.

Amaç fonksiyonu z, değişkenler x_j ve sabit katsayılar c_j ile gösterilirse;

$$z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

gibi doğrusal bir fonksiyon maksimize ya da minimize edilirse, amaç fonksiyonu genel olarak

$$z_{max/min} = \sum_{j=1}^{n} c_jx_j \qquad (j=1,2,\ldots,n)$$

biçiminde ifade edilir.

Buradan problemin matematiksel modeli;

$$\text{En iyi} \quad z = \sum_{j=1}^{n} c_jx_j$$

kısıtlayıcı,

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

ve $\qquad\qquad (i=1,2,\ldots,m)$

$$x_j \geq 0$$

şeklindedir **(Gaver and Thompson, 1973)**.

Burada;

c_j = amaç fonksiyonunun katsayılarını (kar veya maliyet),

x_j = karar değişkenlerini,

b_i = sağ taraf sabitlerini,

a_{ij} = i inci ürün için j inci kaynaktan gerekli miktarını,

c_j = x_j değişkenlerinin amaç fonksiyonundaki katsayılarını,

m : kısıtların sayısını,

n : değişkenlerin sayısını

göstermektedir.

Simpleks Çözüm Tekniği

Büyük boyutlu problemleri çözüme kavuşturmak için Dantzig tarafından geliştirilen simpleks çözüm tekniğinden faydalanılabilir.

Simpleks çözüm tekniği, amaç fonksiyonunu maksimum ya da minimum yapacak en iyi çözüme adım adım yaklaştıran bir algoritmadır. Yöntem cebirsel yineleme işlemine dayanır. Öncelikle başlangıç simpleks tablosu düzenlenir, sonra tekrarlayan işlemlerle belirli bir hesap yöntemi içinde ilerleyerek optimal çözüme ulaşıncaya kadar işlemler sürdürülür.

Simpleks çözüm tekniği, problemlerin çözümüne uygulanırken, eşitsizlik sistemleri eşitlik haline dönüştürülür. Bunun için de aylak (Slack) değişkenlerin eklenmesi veya çıkarılması gerekir. Aylak değişkenlerin kısıtlayıcı fonksiyonlardaki katsayıları birdir, bu katsayılar birim matrisi oluşturur. Amaç fonksiyonunu etkilememeleri için bu değişkenlerin amaç fonksiyonundaki katsayıları sıfırdır, yani sıfır fiyatlıdır. Diğer değişkenler gibi çözüme girerler, bu değerler kullanılmayan kapasiteyi ve hammaddelerin miktarlarını gösterirler **(Esin, 1983)**.

Verilen doğrusal programlama modelinin kısıtları '≥', '≤' ve '=' şeklinde olabilir. Modelin standart biçiminde yazılması için kısıtlar eşitlik haline dönüştürülür **(Öztürk, 1997)**.

i) Eğer kısıtlayıcı

$$a_{11}x_1+a_{12}x_2 \leq b_1$$

şeklinde ise denklemin sol tarafına aylak değişken (**S**) eklenir.

$$a_{11}x_1+a_{12}x_2+\mathbf{S} = b_1$$

denklemin standart şeklidir.

ii) Eğer kısıtlayıcı

$$a_{11}x_1+a_{12}x_2 \geq b_1$$

şeklinde ise denklemin sol tarafından artık değişken (**V**) çıkarılarak, yapay değişken (**A**) eklenir.

$$a_{11}x_1+a_{12}x_2-\mathbf{V}+\mathbf{A} = b_1$$

denklemin standart şeklidir.

Artık değişkenler fazla kapasiteyi ve fazla üretim faktörlerini veya fazla üretim istem ve sunumu ifade eder. Artık değişkenler başlangıç simpleks tablosunun temel değişkenler sütununda yer almaz, onun yerine ekonomik anlamı olmayan yapay değişkenler yer alır **(Öztürk, 1997)**.

iii) Eğer kısıtlayıcı denklemi tam bir eşitlik halinde ise,

$$a_{11}x_1+a_{12}x_2=b_1$$

denklemin sol tarafına sadece yapay değişken (A) eklenir.

$$a_{11}x_1+a_{12}x_2+\mathbf{A}=b_1$$

Aylak, artık ve yapay değişkenlere bir sabit katsayı değeri verilerek amaç fonksiyonuna konur. Artık ve aylak değişkenlerin katsayıları sıfır değerli olur. Yapay değişkenlerin eklenmesi kısıtların uygunluğunun bozulmasına neden olacağından optimal çözümde yapay değişken, karar değişkeni olarak yer almamalıdır. Bu nedenle yapay değişkenlerin katsayıları keyfi büyük cezayı gösteren **M** ile ifade edilir. Maliyet minimizasyonu problemlerinde bu cezanın değeri **+M**, kar maksimizasyonu problemlerinde ise **-M** olarak gösterilir. **M**, problemde yer alan tüm sabitlerin değerinden daha büyük bir sayıdır. Eşitlik haline dönüştürülen kısıtlar simpleks tablosuna yerleştirilir. Kısıtların simpleks tablosuna genel olarak yerleştirilme şekli Tablo 2.1 ' de görülmektedir **(Öztürk, 1997; Taha, 2000)**.

Tablo 2.1. Simpleks tablosunun genel olarak gösterilmesi

c_j (amaç fonksiyonunun katsayılar satırı)

c_B	x_B	değişkenler ($x_1, x_2, x_3, \ldots, x_n$, S, V, A)	çözümler
çözüme giren değişkenlerin amaç fonksiyonundaki katsayıları	çözüme giren değiş-kenler	katsayılar (a_{ij})	
amaç fonksiyonunun değeri	z_j $c_j - z_j$		

Maksimizasyon problemlerinde, (c_j-z_j) satırında bulunan değerler sıfır veya negatif işaretli iseler en iyi (optimal) çözüme ulaşılmıştır. $c_j - z_j \leq 0$

Minimizasyon problemlerinde, (c_j-z_j) satırında bulunan değerler sıfır veya pozitif işaretli iseler en iyi (optimal) çözüme ulaşılmıştır. $c_j - z_j \geq 0$

2.2. Tamsayılı Doğrusal Programlama

Doğrusal programlama problemlerinin çözüm sonuçları, çoğunlukla tamsayı olmayan pozitif sayılardır. Oysa gerçek yaşamda ürünlerin bölünmezlik özelliği gereği çözüm sonuçlarının tamamının ya da bir kısmının tamsayı olmasını gerektiren problemlerle karşılaşılabilmektedir. Örneğin otomobil, makine, buzdolabı üretiminde, sermaye bütçelemesinde ve işgücü planlaması problemlerinin çözüm sonuçlarının tamsayılarla ifade edilmesi gerekmektedir. Çünkü bu problemlerin tamsayı değerli olmayan çözümlerinin ekonomik anlamı yoktur. Bu tür problemlerin modellenmesi ve çözümünde tamsayılı doğrusal programlama etkin çözümler sağlamaktadır **(Tulunay, 1980)**.

Uygulama alanları; satıcıların satış bölgelerinin ayırımı, sermaye planlaması, depo araştırma geliştirme ve en iyi yerleşimi, gezgin satıcı problemleri, yükleme problemi, atama problemi ve pilot planlamasını içerir. Ayrıca her bir işe sadece ve sadece bir kişinin tayin edilebilmesinin yanında, işlere de uygun kişileri en iyi şekilde yerleştirilmesini sağlar. Tamsayılı değişken içeren modeller genellikle büyük ölçekli planlama modelleridir.

Sonuçların tamsayılı olması gereken problemlerin modellenmesi ve çözümüne ilişkin kavram ve tekniklerin belirlenmesine genel olarak " Tamsayılı Programlama" adı verilmektedir.

Tamsayılı doğrusal programlama modeli; doğrusal programlama modelinin kısıtlayıcılarına, karar değişkenlerinin bir kısmının veya tamamının tamsayı olma koşulu getirilmiş halidir. Bu ise amaç fonksiyonunda bulunan değişkenlerin tamsayılı pozitif değerler almasını ifade eder. Doğrusal programlama sürekli değişkenlerle ilgili iken tamsayılı doğrusal programlama, kesikli değişkenlerle ilgilidir **(Halaç, 1991)**.

Tamsayılı doğrusal programlama modeli; genel olarak aşağıdaki gibi formüle edilir.

$$z_{max/min} = \sum_{j=1}^{n} c_j x_j$$

kısıtlayıcı,

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b_i \qquad (j=1,2,...,n)$$

$$(i=1,2,...,m)$$

ve

$$x_j \geq 0 \text{ ve tamsayı}$$

Yukarıdaki genel modelde görüldüğü gibi doğrusal programlama modeline yalnız "ve tamsayı" koşulu eklenmiştir. Buna ek olarak modele tamsayı olması gereken değişkenler için özel kısıtların eklenmesi de gerekebilmektedir. Tamsayılı değişken ve kısıtlayıcı kullanımı ile ilgili özel durumlar aşağıda sıralandığı gibi olabilir **(Kara, 1984)**.

$x_j = 0$ veya 1 değeri alabilen değişkenler iken, n tane seçenekli bir problemde;

i. En fazla bir seçenek benimsenecek ise; $\sum_{j=1}^{n} x_j \leq 1$

ii. En az bir seçenek benimsenecek ise; $\sum_{j=1}^{n} x_j \geq 1$

iii. Sadece bir seçenek benimsenecek ise; $\sum_{j=1}^{n} x_j = 1$

iv. j benimsendiğinde i de benimsenecek ise; $x_j \leq x_i$

v. En fazla bir seçenekten vazgeçilecek ise; $\sum_{j=1}^{n} x_j \geq n-1$

vi. En az bir seçenekten vazgeçilecek ise; $\sum_{j=1}^{n} x_j \leq n-1$

vii. Sadece bir seçenekten vazgeçilecek ise; $\sum_{j=1}^{n} x_j = n-1$

viii. Sadece k seçenekten vazgeçilecek ise; $\sum_{j=1}^{n} x_j = n-k$

ix. En fazla k seçenek benimsenecek ise; $\sum_{j=1}^{n} x_j \leq k$

x. Yalnız k seçenek benimsenecek ise; $\sum_{j=1}^{n} x_j \geq k$

- *Modeldeki m kısıttan k tanesinin sağlanması*

Modelin kısıtları,

$$f_1(x) \leq b_1$$
$$f_2(x) \leq b_2$$
$$\vdots$$
$$f_m(x) \leq b_m$$

olarak verilsin ve bunların k tanesinin gerçekleşmesi yeterli olsun. y_i 'ler m kadar 0 veya 1 değerini alan değişkenler ve M pozitif bir sayı olmak üzere yukarıdaki kısıtlar,

$$f_1(x) \leq b_1 + My_1$$
$$f_2(x) \leq b_2 + My_2$$
$$\vdots$$
$$f_m(x) \leq b_m + My_m$$

ve

$$\sum_{i=1}^{m} y_j = m-k$$
$$y_j = 0 \text{ veya } 1$$

şeklinde yazılır ve m kısıttan k tanesi sağlanır.

- *Verilen iki grup kısıttan birinin sağlanması*

Bir karar modelinde,

$$\text{ya} \quad f_m(x) \leq b_m \quad \text{ya da} \quad f_n(x) \leq b_n$$

Kısıtının sağlanması istensin. M yeterince büyük pozitif bir tamsayı olmak üzere,

ya

$$f_m(x) \leq b_m$$
$$f_n(x) \leq b_n + M$$

ya da

$$f_m(x) \leq b_m + M$$
$$f_n(x) \leq b_n$$

şeklinde yazılır. $y_i = 0$ veya 1 değerini alması istenen bir değişken olmak üzere kısıtlar,

$$f_m(x) \leq b_m + yM$$
$$f_n(x) \leq b_n + (1-y)M$$

veya

$$f_m(x) - yM \leq b_m$$
$$f_n(x) + yM \leq b_n + M$$

şekline dönüşür. En iyi çözümde ilk kısıt ya da ikinci kısıt gerçekleşir.

- *Belirli Bir Küme İçindeki Kısıtlardan Birinin Sağlanması*

$f(x_1, x_2, \ldots, x_n) = d_1$ veya $d_2, d_3, \ldots, d_m$ olsun. $y_i = 0$ veya 1 değerini alması istenen değişken olmak üzere fonksiyon,

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{m} d_i y_i$$

$$\sum_{i=1}^{m} y_j = 1$$

$$y_i = 0 \text{ veya } 1$$

şekline dönüştürülerek istenilen özellik sağlanır.

Tamsayılı doğrusal programlama problemlerinin çözümü, benzer büyüklükteki doğrusal programlama problemlerinin çözümünden daha zordur. Fakat dikkatlice formüle edilmekle ve uygun bir dallandırma stratejisi uygulamak suretiyle tamsayılı doğrusal programlama da kolay çözümlenebilir **(Camm, 1996)**.

Tamsayılı doğrusal programlama ilk bakışta doğrusal programlamadan daha kolay çözüme ulaşacakmış görünümü verir. Çünkü doğrusal programlamada sonsuz sayıda uygun çözüm varken, tamsayılı doğrusal programlamada sınırlı sayıda uygun çözüm vardır. Ayrıca simpleks çözüm tekniğinde, optimal çözümün bulunmasında birkaç köşe noktasının incelenmesi yeterlidir ve uygun çözüm alanını incelemek gerekmez. Fakat tamsayılı doğrusal programlamada ise bütün köşe ve uygun çözüm alanındaki noktaların incelenmesi gerekebilir **(Tütek ve Gümüşoğlu, 1994)**.

Tamsayılı programlama modelleri genellikle doğrusal yapıdadır. Tamsayılı programlama modeli, ilgilenilen problemin özelliklerine göre farklılık göstermekle birlikte, temel kavram ve teknikler daha çok tamsayılı doğrusal karar modelleri üzerinde yoğunlaşmıştır. Bu nedenle tamsayılı programlama denildiğinde daha çok tamsayılı doğrusal programlama kastedilmektedir.

Tamsayılı doğrusal programlama modelleri, modelde yer alan karar değişkenlerinin yapısına göre sınıflandırılmaktadır. Modelde yer alan karar değişkenlerinin tamamının veya bir kısmının tamsayı olması isteğine bağlı olarak "Bütünüyle Tamsayılı", "Karma Tamsayılı" programlama diye sınıflandırılır. Ek olarak ilgili karar değişkenlerinin 0-1 değer alma koşulu ile "Sıfır-Bir Tamsayılı" programlama söz konusudur. Şimdi bu üç tamsayılı doğrusal programlama çeşitleri sırasıyla açıklanmaya çalışılacaktır.

2.2.1. Bütünüyle Tamsayılı Doğrusal Programlama

Modeldeki tüm karar değişkenlerinin tamsayı olması halinde "Bütünüyle Tamsayılı Doğrusal Programlama" söz konusudur. $\forall$ i için x_i tamsayı koşulu vardır.

Bütünüyle tamsayılı doğrusal programlama modelinin matematiksel olarak ifadesi aşağıdaki şekilde gösterilebilir.

$$z_{max/min} = \sum_{j=1}^{n} c_j x_j$$

kısıtlayıcı,

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b_i \qquad (j=1,2,\ldots,n)$$

$$(i=1,2,\ldots,m)$$

ve

$$x_j \geq 0 \text{ ve tamsayı}$$

2.2.2. Sıfır-Bir Tamsayılı Doğrusal Programlama

Modeldeki tüm karar değişkenleri ya "0" ya da "1" değerini almaları istenmesi halinde " Sıfır-Bir Tamsayılı Doğrusal Programlama" söz konusudur.

Sıfır-bir tamsayılı doğrusal programlama modeli, tamsayılı doğrusal programlama modelinin özel bir durumunu yansıtır. Bu tür bir modelde, mümkün bir faaliyetin yapılması (x =1) ya da yapılmaması (x =0) anlamına gelmektedir. Bir işin yapılması ya da yapılmaması, bir işin bir tezgaha atanması ya da atanmaması gibi durumlarda sıfır-bir tamsayılı doğrusal programlama kullanılır **(Tulunay, 1980)**.

Sıfır-bir tamsayılı doğrusal programlama modelinin çözümünde değişkenler sadece sıfır veya bir değerini alabilir. Değişken sayısı az olduğunda, bu yöntemle çözüm kolaylıkla bulunabilir. Fakat modeldeki değişken sayısı arttıkça bütün değişkenleri taramak güçleşeceğinden, çözüme ulaşmak zaman alıcı olmaktadır **(Atlas, 1988)**.

Sıfır-bir tamsayılı doğrusal programlama modelinin çözümü için geliştirilen Toplamlı Algoritma (Balas) ile ilgili araştırmalar yapılmış ve olumlu sonuçlar elde edilmiştir. Bu çalışmalarda özellikle, amaç fonksiyonu katsayılarının pozitif olması ve kısıtlayıcılarının yönlerinin de aynı yöne olması gerektiği üzerinde durmuşlardır **(Salkın, 1975)**.

Sıfır-bir tamsayılı doğrusal programlama modelinin matematiksel olarak ifade edilişi,

$$Z_{max/min} = \sum_{j=1}^{n} c_j x_j$$

kısıtlayıcı,

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b_i \qquad (j=1,2,\ldots,n)$$

$$(i=1,2,\ldots,m)$$

ve

$$x_j = 0 \text{ veya } 1$$

şeklindedir.

2.2.3. Karma Tamsayılı Doğrusal Programlama

Bir doğrusal programlama modelde karar değişkenlerinin bazılarının tamsayılı olması halinde "Karma Tamsayılı Doğrusal Programlama" söz konusudur. $\forall$ i için $x_i \geq 0$ ve $\exists$ i için $x_i \geq 0$ ve tamsayı koşulu vardır.

Tamsayılı doğrusal programlama genellikle bütün karar değişkenlerinin tamsayı olması olarak anlaşılmaktadır. Oysa birçok planlama modelinde, bütün değişkenlerin değil, temel değişkenlerden sadece birkaçının tamsayılı olması istenebilir. Böylece herhangi bir modelin en az bir karar değişkeni tamsayı (bütünüyle ya da 0-1) olmakla sınırlandırılmış ve en az bir karar değişkeni için de tamsayı koşulu yok ise model "Karma Tamsayı Doğrusal Programlama" modeli adını alır **(Camm, 1996)**.

Değişkenlerden sadece bir kısmının tamsayılı olması istenen problemler, üretim süreçlerinde sıkça rastlanan problemlerdir. Ayrıca, tüm değişkenlerin tamsayılı olması gerektiğinde dahi bütün katsayılar ve sağ taraf sabitleri tamsayı olmadıkça bir karma tamsayılı doğrusal programlama problemi elde edilir **(Gottfried, 1973)**.

Karma tamsayılı doğrusal programlamanın matematiksel olarak ifadesi aşağıdaki gibidir.

$$z_{max/min} = \sum_{j=1}^{n} c_j x_j$$

kısıtlayıcı,

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_j \left\{\begin{matrix} \leq \\ \geq \\ = \end{matrix}\right\} b_i \qquad (j=1,2,...,n)$$

$$(i=1,2,...,m)$$

ve

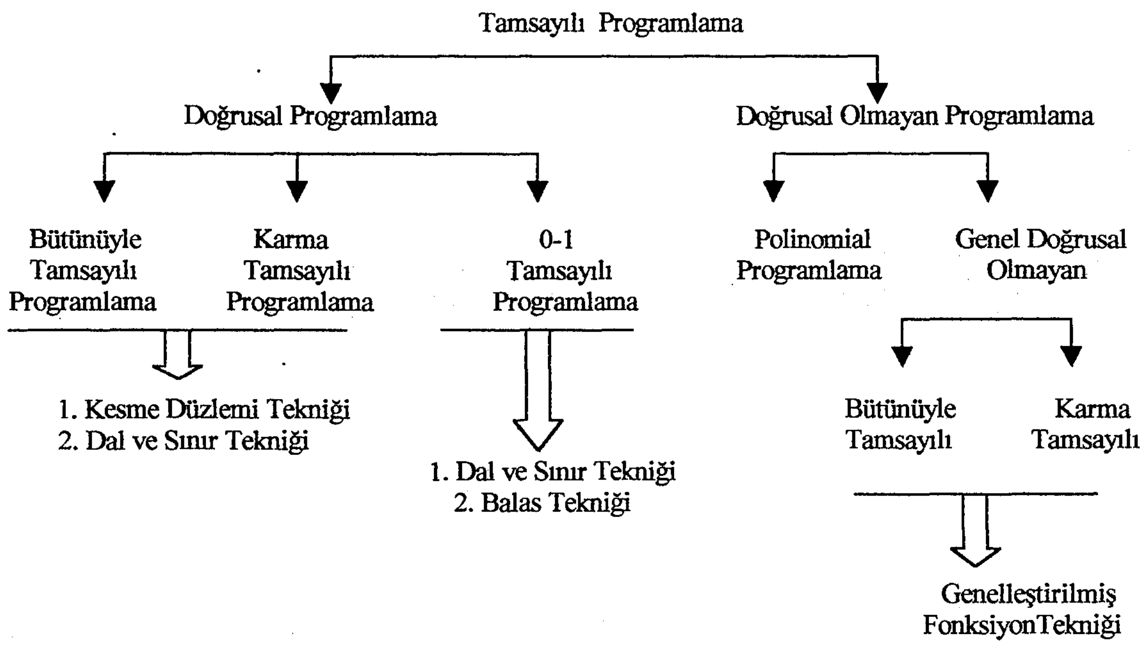
$$x_j \geq 0 \qquad (j=1,2,...,n)$$

$$x_j \text{ tamsayıdır} \qquad (j=1,2,...,I) \quad (I \leq n)$$

Buraya kadar açıklanmaya çalışılan tamsayılı doğrusal programlama modellerinin çözüm teknikleri ise ikinci bölümde açıklanmaya çalışılacaktır.

3. TAMSAYILI DOĞRUSAL PROGRAMLAMA ÇÖZÜM TEKNİKLERİ

Tamsayılı doğrusal programlama modellerinin çözümünde kullanılan çözüm teknikleri bu bölümde açıklanmaya çalışılacaktır. Uygulamada kullanılan tamsayılı programlama çözüm teknikleri birden çok olup, birinin diğerine üstünlüğü amaca ve problemin yapısına göre değişebilmektedir. Bu nedenle araştırmacılar daha çok bu çözüm tekniklerinden kendileri için en uygununu araştırarak uygulamaktadırlar. Teknik tercihinde önem kazanan husus uygulama kolaylığı ve zamandır **(Kara, 1984)**.



Şekil 3.1. Tamsayılı programlama modelleri ve çözüm teknikleri **(Rao, 1979)**

Tamsayılı doğrusal programlama modellerinin çözümünde Geometrik Çözüm Tekniği, Yuvarlama Tekniği, Kesme Düzlemi Tekniği, Dal-Sınır Tekniği kullanılmaktadır. Bu çözüm tekniklerinden Kesme Düzlemi ve Dal-Sınır Tekniği diğerlerine göre daha etkin ve daha yaygın uygulama alanına sahip tekniklerdir.

Grafik Çözüm Tekniği, basit işletme problemlerinin çözümünde uygulama alanı bulabilen bir tekniktir. Tamsayılı doğrusal programlama problemlerinde grafik çözüm tekniği en fazla iki veya üç karar değişkenli modeller için kullanılabilmektedir. Oysa gerçek yaşamda karşılaşılan işletme problemlerinin çoğu çok değişkenli ve çok kısıtlayıcılı karar modelleridir. Bu nedenle grafik çözüm tekniğinin tamsayılı doğrusal programlama problemlerinin çözümünde kullanımı sınırlıdır **(Sarıaslan, 1986).**

Grafik çözüm tekniğinde tamsayılı doğrusal modelin kısıtlayıcılarının sağ taraf sabitlerini oluşturan kaynakların sonuna kadar kullanılması istendiği varsayıldığından, eşitsizlikler önce eşitlik haline dönüştürülerek işlem yapılır **(Esin, 1983).**

Tamsayılı doğrusal programlama problemlerinin grafik çözüm tekniği ile çözülmesinde birtakım özel durumlarla karşılaşabilir. Bu özel durumlar aşağıda sıralanmıştır.

- Eşitsizliklerin tutarsız olması,
- Uygun çözüm alanının bulunmasına rağmen pozitif kısıtlama koşulunun sağlanmaması,
- Uygun çözümün sınırsız olması,
- Tek bir çözümün bulunması,
- Birden fazla optimum çözümün bulunması.

Yuvarlama, öncelikle doğrusal programlama modelinin optimum çözümü simpleks tekniği ile bulunur. Bulunan çözümde tamsayılı olması istenen fakat kesirli değere sahip değişkenin kendisine en yakın tam sayıya yuvarlanarak tamsayılı çözüm elde edilmesine denir.

Yuvarlama tamsayılı programlama problemi çözümünde, zaman ve maliyetten tasarruf sağlaması nedeniyle pratik bir teknik olmasına rağmen sağlıklı bir sonuç vermemektedir. Ulaşılan sonucun gerçek optimal çözümünden farklı olabilmesi ve bazen de uygun çözümün bulunamaması nedeniyle etkin değildir **(Aslan, 1978)**.

Tamsayılı programlamada asıl önemli olan ve üzerinde durulması gereken çözüm teknikleri Kesme Düzlemi ve Dal-Sınır teknikleridir. İzleyen kısımda tamsayılı doğrusal programlama problemi için bu iki teknik ayrıntılı anlatılmaya çalışılacaktır.

3.1. Kesme Düzlemi Tekniği

Doğrusal programlama problemlerinin tamsayılı çözümlerini sağlayacak hesaplama tekniği ilk kez 1958 yılında R.E. Gomory tarafından geliştirilmiştir. Gomory' nin geliştirdiği hesaplama tekniğine Kesme Düzlemi Tekniği adı verilmiştir. Bu teknik sonlu sayıda işlemden sonra optimal bir tamsayı çözümü sağlar. Gomory' nin Kesme Düzlemi tekniği; probleminin simpleks tekniği ile bulunan optimal çözümünden başlar. Eğer bulunan optimal çözüm tamsayılı bir çözüm değilse, doğrusal programlama problemine ek bir kısıt ilave edilir. Bu ek kısıt Gomory' nin geliştirdiği "Kesme Düzlemi" kuralına bulunur. Bulunan yeni ek kısıtlayıcı simpleks çözüm tablosuna yerleştirilerek, istenilen tamsayılı değer elde edilinceye kadar çözüm işlemleri tekrarlanarak devam eder.

3.1.1. Bütünüyle Tamsayılı Programlamada Kesme Düzlemi Tekniği

Bütünüyle tamsayılı doğrusal programlama probleminin çözümünü bulmak için kullanılan Gomory kesme düzlemi tekniğinde takip edilecek aşamalar şunlardır **(Doğan, 1995)**;

i) Bir tamsayılı doğrusal programlama probleminin kısıtlarının katsayıları tamsayılı değilse, ilk aşama katsayıları tam sayıya dönüştürmektir **(Taha, 2000)**.

Örneğin bir problemdeki kısıtlardan biri,

$$(7/3)x_1 + (5/2)x_2 \leq 4$$

formundaysa, o zaman kısıtın her iki tarafı 6 sabit sayısı ile çarpılarak,

$$14x_1 + 15x_2 \leq 24$$

şekline çevrilir. Eğer problemde böyle bir aşama yoksa doğrudan ikinci aşamaya geçilir.

ii) İkinci aşama, doğrusal programlamanın bir çözüm tekniği olan simpleks çözüm tekniğine göre problemin optimum çözümünü bulmaktır. Eğer optimal çözüm tamsayılı değerlere sahipse, tamsayılı doğrusal programlama problemi için çözüm elde edilmiş olur. Eğer değilse bir sonraki aşamaya geçilir **(Doğan, 1995)**.

iii) Üçüncü aşama, kesme düzlemini bulmaktır. Problemin simpleks çözüm tekniği ile bulunan tamsayılı olmayan çözümü ele alınarak doğrusal programlama problemine ek bir kısıt ilave edilir.

Gomory' nin geliştirdiği kurala göre simpleks tekniği ile elde edilen optimal çözüm değerlerinden tamsayı olmadığı varsayılan en büyük kesir değerli karar değişkeni x_k seçilir.

$\mathbf{x_k}$ temelde olan değişken, $\boldsymbol{\beta_k}$ tamsayılı olmayan çözüm değeri ve $\mathbf{w_j}$ temelde olmayan değişken olmak üzere x_k 'nın bulunduğu k 'ıncı eşitlik

$$x_k = \beta_k - \sum_{j=1}^{n} a_{kj} w_j$$

göz önüne alınır.

$[\beta_k]$ tamsayılı kısım ve f_k pozitif bir kesir olmak üzere β_k sabiti tam kısmı ile kesirli kısmı ayrılarak yazılırsa,

$$\beta_k = [\beta_k] + f_k \qquad (0 < f_k < 1)$$

bulunur.

Aynı şekilde $[a_{kj}]$ tamsayılı kısım, f_{kj} negatif olmayan bir kesir olmak üzere a_{kj} sabiti tam kısmı ile kesirli kısmı ayrılarak yazılırsa,

$$a_{kj} = [a_{kj}] + f_{kj} \qquad (0 \leq f_{kj} < 1)$$

olarak bulunur. Bunlar $x_k = \beta_k - \sum_{j=1}^{n} a_{kj} w_j$ eşitliğinde yerine yazılırsa,

$$x_k = [\beta_k] + f_k - \sum_{j=1}^{n} ([a_{kj}] + f_{kj}) w_j$$

elde edilir. Elde edilen bu eşitlik düzenlenerek, tamsayılı değişkenler denklemin sağ tarafında toplanır.

$$f_k - \sum_{j=1}^{n} f_{kj} w_j = x_k - [\beta_k] + \sum_{j=1}^{n} [a_{kj}] w_j$$

Sağ tarafta yer alan tamsayılı değişkenler atılır ve sadece kesirli eleman bırakılır. Tamsayılı değişkenler atıldığı için eşitlik halindeki denklem eşitsizliğe dönüşür ve sol taraftaki kesirli kısım sağ taraftaki elemanın değerinden büyük veya eşit olur.

$$f_k - \sum_{j=1}^{n} f_{kj} w_j \leq 0$$

Böylece ek kısıtlayıcı da elde edilmiş olur. Bu ek kısıtlayıcı denklemi standart hale getirmek için yeni bir aylak değişken eklenir.

$$f_k - \sum_{j=1}^{n} f_{kj} w_j + S_k = 0$$

ve

$$S_k = \sum_{j=1}^{n} f_{kj} w_j - f_k$$

elde edilir. Bu ek kısıta "Kesme Düzlemi" adı verilir. Bulunan yeni ek kısıtlayıcı tamsayılı olmayan optimal çözüm tablosuna yerleştirilerek simpleks çözüm işlemlerine geçilir **(Öztürk, 1986)**.

Yeni kısıt eklenerek kesme düzlemi algoritmasına göre yapılan çözümde istenilen tamsayılı değer elde edilmiş ise işlem bitmiştir. İstenilen tamsayılı değerler bulunamamışsa, son tablodan yeni bir kesme düzlemi elde edilir ve yine simpleks çözüm tekniği uygulanır. Eğer modelin bir tamsayılı çözümü var ise, süreç tamsayılı değer elde edilinceye kadar tekrar edilir. Bununla birlikte simpleks çözüm tekniğinde tamsayılı çözüm elde edilmezse, söz konusu problem uygun tamsayılı çözüme sahip değildir.

Kesirli kısımdaki değişkenlerden negatif olanı veya birden büyük olanı varsa, bu sayıların eşleniği bulunarak onların yerine alınır. Sayı ve eşleniği arasındaki fark tamsayı olmalıdır. Bu durumda bunlara "eşleşik" denir ve x eşleşik y ise $x \equiv y$ şeklinde yazılır **(Wolsey, 1998)**.

Örneğin,

$-3/8 \equiv 5/8$ olduğunda

$5/8 - (-3/8) = 1$ ‘ dir,

ya da

$9/8 \equiv 1/8$ olduğunda

$9/8 - 1/8 = 1$ ’dir.

Tamsayılı çözümü araştırmada kullanılan kesim düzlemi, negatif olmayan tamsayı çözümü veren denklemin katsayısına eşleşik olan en küçük negatif olmayan sayılar kullanılarak oluşturulur. Gomory kesme düzlemi yönteminin dezavantajı hesaplama yapılırken hata olasılığının yüksek olması ve bazen en iyi çözüme ulaşamamasıdır **(Taha, 2000)**.

3.1.2. Karma Tamsayılı Doğrusal Programlamada Kesme Düzlemi Tekniği

Tamsayılı doğrusal programlama problemlerinin çözümünde kullanılan kesme düzlemi tekniğinin, karma tamsayılı doğrusal programlama modelleri için geliştirilen kısmına ”Karma Algoritma” denir.

Algoritmaya göre önce, karma tamsayılı doğrusal model tamsayılı olma şartı önemsenmeden doğrusal programlama problemi olarak simpleks çözüm tekniğine göre çözülerek optimal çözüm tablosu bulunur. Tamsayı olmayan, ama tamsayı olması istenen değişkenin satırı kullanılarak bir kesme düzlemi elde edilir. Bu kesme düzlemi yeni kısıt olarak probleme eklenerek yeniden çözülür ve tamsayılı sonuç bulununcaya kadar işleme devam edilir **(Gençyılmaz, 1991)**.

Tablo 3.1. Kesirli kesme ilave edildikten sonraki simpleks tablosu

temel	x_0	$x_1 \ldots x_2 \ldots x_m$	$w_1 \ldots w_i \ldots w_n$	S_i	çözüm
z	1	$0 \ldots 0 \ldots 0$	$c_1 \ldots c_j \ldots c_n$	0	β_0
x_1	0	$1 \ldots 0 \ldots 0$	$a_{11} \ldots a_{1j} \ldots a_{1n}$	0	β_1
.	.	. . .	. . .	.	.
.	.	. . .	. . .	.	.
x_i	0	$0 \ldots 1 \ldots 0$	$a_{i1} \ldots a_{ij} \ldots a_{in}$	0	β_i
.	.	. . .	. . .	.	.
.	.	. . .	. . .	.	.
x_m	0	$0 \ldots 0 \ldots 1$	$a_{m1} \ldots a_{mj} \ldots a_{mn}$	0	β_m
S_i	0	$0 \ldots 0 \ldots 0$	$-f_{i1} \ldots -f_{ij} \ldots -f_{in}$	1	$-f_i$

Burada,

x_k = temelde olan değişkenler

w_j = temelde olmayan değişkenler

β_k = tamsayılı olmayan çözüm değerleri

olarak ifade edilmiştir.

x_k, karma tamsayılı problemin tamsayı olması istenen değişkeni olsun. Eğer x_k değişkeni temelde ise,

$$x_k = \beta_k - \sum_{j=1}^{n} a_{kj} w_j \tag{1}$$

eşitliği optimal sürekli çözümden elde edilir. x_k değişkeni temelde değilse sıfır değerine sahiptir ve tamsayılı olma şartı sağlanır **(Tulunay, 1980)**.

β_k ‘nın, tamsayı kısmını $[\beta_k]$ ve kesirli kısmını f_k olarak gösterilirse;

$$\beta_k = [\beta_k] + f_k \qquad (0 < f_k < 1)$$

olur. Burada dikkat edilmesi gereken husus kesirli kısmın daima pozitif olmasıdır. Bulunan bu eşitlik (1) ‘de yerine yazılırsa,

$$x_k = [\beta_k] + f_k - \sum_{j=1}^{n} a_{kj} w_j$$

$$f_k - \sum_{j=1}^{n} a_{kj} w_j = x_k - [\beta_k] \tag{2}$$

eşitliği elde edilir. Karma tamsayılı probleme uygun bir çözüm için (2) eşitliğinin sağ tarafı tamsayı olmalıdır. O halde,

$$f_k - \sum_{j=1}^{n} a_{kj} w_j = \text{tamsayı} \tag{3}$$

olmalıdır. a_{kj} ‘yi iki alt kümeye ayırmak faydalıdır. J^+ kümesi tüm $a_{kj} \geq 0$ ‘ı ve J^- kümesi tüm $a_{kj} < 0$ ‘ı kapsasın. Bu iki alt kümeyle ilgili olarak (3) eşitliğini,

$$\sum_{J+} a_{kj} w_j + \sum_{J-} a_{kj} w_j = f_k + t \tag{4}$$

biçiminde yazılabilir. Burada t, pozitif ya da negatif bir tamsayıyı temsil etmektedir.

Şimdi, $\sum_{J} a_{kj} w_j$ ‘nin sıfır ya da pozitif olduğu durumu düşünülürse, bu durumda t’ nin negatif olmayan bir tamsayı olması gerektiğinden,

$$\sum_{J+} a_{kj}w_j + \sum_{J-} a_{kj}w_j \geq f_k \qquad (5)$$

olur. Ayrıca sol taraftaki ikinci terim negatif olduğu için,

$$\sum_{J+} a_{kj}w_j \geq f_k \qquad (6)$$

eşitsizliği elde edilir.

Eğer $\sum_{J} a_{kj}w_j$ 'nin negatif olduğunu farz edilirse, t negatif bir tamsayı olmalıdır. Buradan $t \leq -1$ olur ve yukarıdakiyle benzer mantıkla,

$$\sum_{J-} a_{kj}w_j \leq f_k-1 \qquad (7)$$

yazılabilir. Bu eşitsizliğin her iki tarafı $f_k / (f_k-1)$ ile çarpılırsa,

$$\sum_{J-} [f_k/(f_k-1)]\ a_{kj}w_j \geq f_k \qquad (8)$$

eşitsizliği elde edilir. (6) ve (8) eşitsizliklerinin sağlanması gerektiğinden ve eşitsizliklerin sol tarafları sıfır ya da pozitif olduğundan aşağıdaki eşitsizlik yazılabilir.

$$\sum_{J+} a_{kj}w_j + \sum_{J-} [f_k/(f_k-1)\ ]\ a_{kj}w_j \geq f_k \qquad (9)$$

(9) denklemi, sadece bir değişkenin tamsayılı olmakla sınırlandığı durumda Gomory kesme düzlemini verir.

Birden fazla değişkenin tamsayılı olması koşulu olduğunda Gomory kesme düzlemi aşağıdaki biçimde elde edilir.

Tek değişken için Gomory kesme düzlemi algoritması,

$$\sum_{J} (\lambda_j) w_j \geq f_k \tag{10}$$

olarak yeniden yazılsın. Burada;

$$\lambda_j = \begin{cases} a_{kj} & ; a_{kj} \geq 0 \text{ ve } w_j \text{ tamsayı değil ise} \\ [f_k / (f_k - 1)]\, a_{kj} & ; a_{kj} < 0 \text{ ve } w_j \text{ tamsayı değil ise} \end{cases}$$

olarak tanımlanmıştır.

R : Tamsayılı değişkenler kümesi

R^*: Tamsayılı olmayan değişkenler kümesi

olsun. (2) denklemini,

$$f_k - \sum_{R} a_{kj} w_j - \sum_{R^*} a_{kj} w_j = x_k - [\beta_k] \tag{11}$$

olarak yeniden yazılabilir. Tamsayı sınırlamalı değişkenler için,

$$a_{kj} = [a_{kj}] + f_{kj} \tag{12}$$

yazılsın. Burada;

$[a_{kj}]$, a_{kj} 'den daha küçük ya da eşit olan en büyük tamsayıdır. f_{kj} kesirli kısımdır ve $0 \leq f_{kj} < 1$ 'dir. a_{kj} 'nin tam ve kesirli kısmıyla ilgili olarak (11) denklemi aşağıdaki biçimi alır.

$$f_k - \sum_R f_{kj} w_j - \sum_{R^*} a_{kj} w_j = x_k - [\beta_k] + \sum_R [a_{kj}] w_j = \text{tamsayı} \tag{13}$$

Eğer R^* kümesindeki pozitif ve negatif a_{kj} 'ye karşılık olarak sırasıyla J^{+*} ve J^{-*} alt kümeleri tanımlanırsa,

$$\sum_R f_{kj} w_j + \sum_{J+} a_{kj} w_j + \sum_{J-} [f_k/(f_k-1)]\, a_{kj} w_j \geq f_k \tag{14}$$

biçiminde kısıtlama elde edilir. Şimdi,

$$a_{kj} = [a_{kj}] - (1 - f_{kj}) \tag{15}$$

olarak tanımlansın. $[a_{kj}]$, a_{kj} 'den daha büyük olan en küçük tamsayıdır. Bu tanımlamayla birlikte kısıt olarak

$$\sum_R [f_k/(1-f_k)]\,(1 - f_{kj})\, w_j + \sum_{j+*} a_{kj} w_j + \sum_{j-*} [f_k/(f_k-1)]\, a_{kj} w_j \geq f_k \tag{16}$$

eşitsizliği elde edilir **(Gottfried, 1973)**.

(9), (14) ve (16) eşitsizliklerinin benzerliğinden dolayı bunların tümünü (10) eşitsizliği biçiminde yazılabilir. Böylece birden fazla değişken tamsayılı olmakla sınırlandığı durumda Gomory kesme düzlemi algoritması

$$\sum_{j=1}^{n} (\lambda_j) wj \geq f_k \tag{17}$$

formülüyle elde edilir. Burada;

$$\lambda_j = \begin{cases} a_{kj} & ; a_{kj} \geq 0 \text{ ve } w_j \text{ tamsayı değil ise} \\ [f_k / (f_k - 1)]\, a_{kj} & ; a_{kj} < 0 \text{ ve } w_j \text{ tamsayı değil ise} \\ \\ [f_k / (1 - f_k)]\,(1 - f_{kj}) & ; f_{kj} > f_k \text{ ve } w_j \text{ tamsayı ise} \\ f_{kj} & ; f_{kj} \leq f_k \text{ ve } w_j \text{ tamsayı ise} \end{cases}$$

olarak tanımlanmaktadır **(Tulunay, 1980).**

(17) eşitsizliğinin her iki tarafının (-1) ile çarpılması suretiyle

$$\sum_{j=1}^{n} (-\lambda_j) w_j \leq -f_k \tag{18}$$

elde edilir. Buraya S_k gibi negatif olmayan bir aylak değişken eklenerek (18) eşitsizliği eşitlik haline dönüştürülür.

$$-\sum_{j=1}^{n} (\lambda_j) w_j + S_k = -f_k$$

Bu son eşitlik gerekli kesme düzlemidir ve tamsayı olması istenilen x_k için gerekli bir durumu gösterir **(Gottfried, 1973).**

En iyi kesmenin hangisi olduğu sorunu, çözüm uzayındaki farklı kesmelerin karşılaştırılmasıyla bulunabilir. En etkili olan kesme çözüm uzayını ne derece daralttığı ile ilgili olarak ölçülebilir. Bu sonuç matematiksel olarak ifade edilirse,

$$\sum_{j=1}^{n} f_{kj} w_j \geq f_k \tag{19}$$

ve

$$\sum_{j=1}^{n} f_{ij}\, w_j \geq f_i \qquad (20)$$

eşitsizlikleri elde edilir. Bütün j 'ler için $f_k \geq f_i$ ve $f_{kj} \leq f_{ij}$ ise " (19) kesmesi (20) kesmesinden daha etkilidir" denir. Etkinlik tanımlaması hesaplamalar açısından zordur. Bunun için kaynak satırdan kesmeyi oluşturmak için iki kural vardır. Bunlar;

$$\text{i) enb } \{ f_k \} \qquad (21)$$

$$\text{ii) enb } \{ f_k \,/\, \sum_{j=1}^{n} f_{kj} \} \qquad (22)$$

dir. İkinci kural etkinliğin tanımlamasını daha iyi ifade ettiğinden daha etkilidir **(Taha, 1989)**.

3.2. Dal-Sınır Tekniği

Bu teknik ile ilgili bilinen ilk gelişmeler A.H. LAND ve A.G. DOİG tarafından yapılmıştır. Fakat 1964 yılında DAKİN' in yaptığı değişiklik büyük hesaplama kolaylığı sağlamaktadır. LAND ve DOİG tarafından tanımlanan bu teknik gezgin satıcı problemleri için özelleştirilmek suretiyle Little, Murty, Sweeny ve Karel tarafından 1968' de uygulanmıştır **(Salkın, 1975)**.

Dal-sınır tekniğine göre çözüm yaparken, öncelikle karar değişkenlerinin tamsayılı olmakla sınırlandırılmış kısıtı ihmal edilir. Meydana gelen doğrusal programlama modelinin simpleks çözüm tekniğine göre optimal çözümü bulunur. Eğer optimal çözüm tamsayılı değerlere sahipse, tamsayılı doğrusal programlama problemi çözülmüştür. Eğer değilse, problemin tamsayılı olmayan çözümü ele alınarak, uygun bir dallanma stratejisi uygulanır.

3.2.1. Bütünüyle Tamsayılı Doğrusal Programlamada Dal-Sınır Tekniği

Dal-sınır tekniğinin temel mantığı şu şekilde açıklanabilir. Diyelim ki minimize edilmesi gereken bir amaç denklemi vardır. Ayrıca amaç denkleminin optimum değeri ile ilgili bir "üst sınır" bulunmaktadır ki, bu genellikle amaç denkleminin bulunmuş olan en iyi değeridir. Buna göre ilk adımda, mümkün çözümler çeşitli alt problemlere ayrılacaktır ve bunların her biri için, alt problemlerdeki çözümlerin amaç denklemi değerlerine "bir alt sınır" bulunacaktır. Amaç denkleminin üst sınırını aşan alt sınırlara sahip alt problemler araştırma dışı bırakılacaktır. Kalan alt problemlerden en küçük alt sınıra sahip olanı, yeniden başka alt problemlere ayrılarak bu alt problemlerin de alt problemleri saptanacaktır. Giderek, yukarıda yapıldığı gibi, burada da alt problemlerden bazıları araştırma dışı bırakılacaktır. Bu süreç, ilgili amaç denklemi değeri herhangi bir alt problemin alt sınırından daha büyük olmayan bir mümkün çözüm bulununcaya kadar sürdürülecektir **(Tulunay, 1980)**.

Dal-sınır tekniğinin temel ilkesi "böl ve fethet" mantığıdır. Orijinal problem çok büyük olduğundan daha küçük alt problemlere bölünür ki bunlar fethedilebilsin. Bölme işlemi tüm geçerli çözümlerin küçük alt çözümlere dağıtılmasıdır. Fethetme (göz önünde olma ya da olmama) ise alt kümedeki en iyi çözümü belirler, eğer bu çözüm en iyi olmayacaksa o alt kümeyi olduğu gibi araştırma dışı bırakır **(Hillier, Liberman, 1974)**.

Dal-sınır tekniğinin işlemleri maddelenecek olursa;

i. İşleme bütün çözüm setleri ile başlanır, buna mümkün olmayan çözümler de dahildir ve $S_ü = \infty$ konur. Burada,

$S_ü$: üst sınır

S_a : alt sınır

göstermektedir.

ii. Kalan alt problemler (bunlar henüz ne araştırma dışı bırakılmış ne de alt problemlere ayrılmıştır) bazı dal kurallarından yararlanılarak seçilir ve bunlar tekrar alt problemlerine ayrılır.

iii. Her bir yeni alt problemin mümkün çözümleri için, amaç denklemi değeri ile ilgili bir S_a alt sınırı belirlenir.

iv. Yeni alt problemlerin her biri için aşağıdaki kıyaslamalar yapılır ve;

1. Eğer $S_a \geq S_ü$

veya,

2. Bulunan alt problemde hiçbir mümkün çözüm yok ise,

ya da,

3. $S_a < S_ü$ iken alt problemdeki en iyi mümkün çözüm bulunur. $S_a = S_ü$ kabul edilerek bulunan çözüm için 1. koşul kalan bütün alt problemlere yeniden uygulanırsa,

söz konusu yeni alt problem araştırma dışı bırakılır.

v. İncelenmesi gereken herhangi bir alt problem kalmayınca, o anda elde bulundurulan çözüm problemin optimum çözümü olacaktır. Aksi halde dal kurallarının uygulandığı ikinci adıma dönülmelidir.

Eğer amaç denklemi bir minimizasyon değil de maksimizasyon gerektiriyorsa, süreç aynen uygulanacak, ancak alt ve üst sınırların işlevleri tersine olacaktır. Böylece $S_ü$ yerine S_a ve $+\infty$ yerine $-\infty$ alınacak, eşitsizlikler yön değiştirecektir. Yani $\geq$ ise $\leq$ olacaktır.

3.2.2. Sıfır-Bir Tamsayılı Doğrusal Programlamada Dal-Sınır Tekniği

Sıfır-bir tamsayılı doğrusal modeli, bütünüyle tamsayılı doğrusal model için verilen dal-sınır tekniği ile çözmek mümkündür. Fakat oldukça uzun işlemlerle karşılaşılmaktadır. Artan bu işlem yükünü azaltmak amacıyla modelin yapısından hareketle sıfır-bir tamsayılı modelin çözümü için özel dal-sınır tekniği geliştirilmiştir **(Kara, 1984)**.

Tamsayılı programlama problemlerinin çözümünde kullanılan dal-sınır tekniğinin, sıfır-bir tamsayılı programlama modelleri için geliştirilen kısmına "Balas Algoritması" denir.

1965' te Balas tarafından geliştirilen Balas Algoritması, hesaplamalarda yalnızca toplama ve çıkarma işlemleri kullanılarak uygulandığı için "Toplamlı Algoritma" olarak da bilinmektedir.

Toplamlı algoritmaya göre sıfır-bir tamsayılı programlama problemi aşağıdaki koşulları sağlamak zorundadır **(Taha, 2000)**.

1. Amaç fonksiyonu bütün katsayıları negatif olmayan bir minimizasyondur.

2. Kısıtların tümü ($\leq$) tipinde olmalıdır. Gerekirse sağ taraflar negatif olabilir. Daha sonra bu kısıtlar sol tarafında dolgu değişkeni kullanmak suretiyle eşitlik haline getirilir.

Bir doğrusal programlama modelinde $x = (x_1, x_2, \ldots , x_n)$ gibi n tane karar değişkeninden p tanesinin değeri (p<n) belirlenmiş olsun. Değerleri belli olan p tane değişken kısmi çözüm kümesini oluşturur. Ancak bu kısmi çözüm kümesi uygun olmayabilir. Geriye kalan, n-p tane çözüm değeri araştırılacak olan değişkene serbest

değişken denir. Toplamlı algoritmanın uygulanmasında, p tane kısmi çözüm dallandırma noktasına, serbest değişkenler de düğümlerden yapılacak dallanma değişkenine karşılık gelmektedir **(Kara, 1984)**.

Sıfır-bir tamsayılı programlamada her değişken sadece 0 veya 1 değerini alabileceğinden, n-p tane serbest değişken için uygun dallandırma sayısı en az 2^{n-p} kadardır. O halde bir kısmi çözümden oluşturulacak dallandırmanın gerekli olmadığı ortaya konulabilirse, yalnız o kısmi çözümden 2^{n-p} tane seçenek bütünüyle göz önüne alındığı halde tek tek işleme sokulmayacaktır. Toplamlı algoritmanın temelinde yatan budur **(Kara, 1984)**.

Bir kısmi çözüm, tamamlayanlarıyla birlikte düşünülerek ya yeni dallandırmaya tabi tutulur, ya da işlemin tamam olduğu görülerek işlem dışı bırakılır (2^{n-p} seçenek ile işlem yapmadan). Tüm kısmi çözümler işlem dışı bırakıldığında en iyi çözüme ulaşılma konusunda karar verilebilecektir **(Salkın, 1975)**.

Amaç fonksiyonunun optimum değeri araştırılan sıfır-bir tamsayılı bir karar modelinin, önceden belirlenen bir alt sınır S_a olsun. Herhangi bir çözümün işlem dışı kalması, ya tamamlayan çözümlerin uygun olmaması ya da tamamlayan uygun çözümlere karşı gelen amaç fonksiyonunun değerlerinin S_a 'ya eşit ya da küçük kalmasıyla mümkündür **(Tulunay, 1980)**.

Sıfır-bir tamsayılı programlama probleminin herhangi bir kısmi çözümün işlem dışı olup olmadığına karar vermek için yapılması gereken işlemler şu şekilde açıklanabilir.

Kısmi çözüme giren j 'ler K kümesi ile, serbest değişkenlere karşı gelen j 'ler S kümesiyle gösterilsin. Bu gösterimle verilen modelin kısıtları,

$$\sum_{j\in K} a_{ij}\, x_j + \sum_{j\in S} a_{ij}\, x_j \le b_i\,, \qquad i = 1,2,\dots,m$$

ve amaç fonksiyonu,

$$x_0 = \sum_{j \in K} c_j x_j + \sum_{j \in S} c_j x_j$$

şeklinde yazılabilir.

Amaç fonksiyonunu önceden belirlenen değeri S_a olsun. c_j 'ler tamsayı iken eldeki kismi çözümün işlem dışı olabilmesi için, x_K 'in her x_S tamamlayanı için,

$$\sum_{j \in K} c_j x_j + \sum_{j \in S} c_j x_j < S_a + 1$$

veya en az bir i için,

$$\sum_{j \in K} a_{ij} x_j + \sum_{j \in S} a_{ij} x_j > b_i$$

eşitsizliklerinin sağlanması gerekmektedir.

Bir başka gösterimle, kısmi çözüme karşı gelen c_j x_j 'lerle, a_{ij} x_j 'lerin toplamı bilindiğinden en az bir i için,

$$\sum_{j \in S} c_j x_j < (S_a + 1) - \sum_{j \in K} c_j x_j \qquad (1)$$

$$\sum_{j \in S} a_{ij} x_j > b_i - \sum_{j \in K} a_{ij} x_j \qquad (2)$$

sisteminin gerçekleşmesi, x_K 'nın işlem dışı olması demektir.

x_K için sistemin sağlanıp sağlanmadığını test etmek için (1) nolu eşitsizliğin her iki tarafı (-1) ile çarpılır. $-c_j$ yerine a_{0j} ve $-S_a - 1$ yerine b_0 yazılırsa,

$$-c_j = a_{0j} \quad \text{ve} \quad -S_a - 1 = b_0$$

olur. (1) ve (2) nolu sistemler;

$$\sum_{j \in S} a_{ij} x_j > b_i - \sum_{j \in K} a_{ij} x_j \qquad i = 1,2,\ldots,m \tag{3}$$

şeklinde yazılabilir. Eğer verilen x_K ile ilgili, en az bir i için bu eşitsizlik geçerli ise x_K işlem dışı olur.

x_j 'ler 0 ve 1 değerleri alabildiğine ve (3) nolu sistemin sağ tarafı sabit sayıları olduğuna göre, bu sistemin en az bir i için

$$\sum_{j \in S} \text{Enk}\,\{a_{ij}, 0\} > b_i - \sum_{j \in K} a_{ij} x_j \tag{4}$$

eşitsizliğinin sağlanması demektir. Çünkü x_K 'nın tamamlayanlarında $j \in S$ için $x_j = 0$ veya $x_j = 1$ olacağından (3) nolu eşitsizliğinin sol tarafının toplamını en küçük yapan değer a_{ij} 'nin işaretine göre belirlenecektir.

x_K 'nın tamamlayanları için, (3) nolu sistemin gerçekleşip gerçekleşmediğine bakabilmek için; sol tarafının toplamının alabileceği en küçük değeri hesaplamak gerekir. Bu da, $a_{ij}\, x_j$ 'lerin en küçüklerinin toplamı olur ki;

$$a_{ij}\, x_j = \begin{cases} 0\,, & x_j = 0 \\ a_{ij}\,, & x_j = 1 \end{cases}$$

olduğundan, enk $\{a_{ij}, x_j\}$= enk $\{a_{ij}, 0\}$ olur ve (3) nolu sisteminin sol taraf toplamının alabileceği en küçük değer,

$$\sum_{j \in S} \text{enk } \{a_{ij}, 0\}$$

olarak bulunur.

Toplamlı Algoritmanın Adımları

Maksimizasyon amaçlı bir problem için sıfır-bir tamsayılı doğrusal karar modelinin toplamlı algoritmayla çözümündeki adımlar şöyle sıralanabilir.

Adım 1: Kısmi çözüme girecek değişken kümesini seçerek, bunlardan oluşturulabilir tüm kısmi çözümleri bul. Uygun kısmi çözümlerden amaç fonksiyonunun en büyük değerini alt sınır olarak al. Kısmi çözümlerin hiçbiri uygun çözüm değilse $S_a = -\infty$ olarak al.

Adım 2: Eldeki kısmi çözümler kümesinde tüm kısmi çözümler işlem dışı ise, dördüncü adıma git, değilse işlem dışı olmamış bir kısmi çözümü tüm tamamlayanlarıyla birlikte incele. İşlem dışı ise diğer çözümlere geç, değil ise üçüncü adıma git.

Adım 3: Eldeki kısmi çözümde yer almayan x_K 'yı seçip $x_K = 0$ ve $x_K = 1$ için iki kısmi çözüm bul. Mümkünse alt sınırı yenile. Kısmi çözümde değişken sayısı n 'ye eşit ise, dördüncü adıma geç, değilse ikinci adıma dön.

Adım 4: İncelenmeyen kısmi çözüm varsa ikinci adıma git, yoksa n değişkenli kısmi çözümlerden eldeki S_a 'ya karşı gelen x_0 değeri en büyük olan en iyi çözümdür.

Bu algoritmadan faydalanarak sıfır-bir tamsayılı modelin bilgisayarla daha kısa sürede çözümü için araştırmalar yapılmaktadır. Tüm bu araştırmaların temeli toplamlı algoritmadır. Bu araştırmalarda üzerine durulan nokta, amaç fonksiyonunun pozitif parametrelerle, kısıtların ise aynı yönde oluşturulmasıdır **(Kara, 1984)**.

3.2.3. Karma Tamsayılı Doğrusal Programlamada Dal-Sınır Tekniği

Tamsayılı doğrusal programlama problemlerinin çözümünde kullanılan dal-sınır tekniğinin, karma tamsayılı doğrusal programlama modelleri için geliştirilmiş haline "Ayrıştırma Algoritması" denilmektedir **(Schrijver, 1998)**.

Bir doğrusal programlama probleminde bazı karar değişkenlerin (I kadarı) tamsayılı değerlerle sınırlandırılmış (fakat sadece 0 ve 1 olmadığı) ve diğerlerinin sürekli değişken olduğu karma tamsayılı modeli ele alınsın. İsimlendirme kolaylığı için değişkenler sırayla işleme sokulsun. İlk I değişken tamsayı olma koşulu ile sınırlandırılmış değişken olsun. Ele alınan modelin genel şekli,

$$\max z = \sum_{j=1}^{n} c_j x_j$$

kısıtlayıcı,

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad i = 1,2,\ldots,m \text{ için}$$

ve $\qquad x_j \ge 0 \quad , \; j=1,2,\ldots,n$ için

x_j tamsayıdır. $\qquad j=1,2,\ldots,I \quad (I \le n)$

x_j = karar değişkenleri (dallanma değişkenleri)

a_{ij} = x_j 'nin girdi katsayıları

c_j = x_j 'nin amaç fonksiyonundaki katsayıları

b_i = sınırlı kaynak miktarı

x_j^* = tamsayı olması gereken dallanma değişkeninin tamsayı olmayan değeri

Karma tamsayılı doğrusal programlama için standart bir yaklaşım haline gelen temel dal ve sınır algoritmasının yapısı, A.H. Land ve A.G. Going tarafından geliştirilen dal ve sınır algoritmasına dayanarak ilk defa R.J. Dakin tarafından ortaya konmuştur. Bu algoritma yapı bakımından toplamlı algoritmaya oldukça benzemektedir. Yapısı da dal ve sınır algoritmasına uymakla birlikte, aynı zamanda bölümlemek için gerekli yeni alt problemlerin seçimi için, yeni sınır kuralını kullanmaktadır. Bununla birlikte değişkenler, bu noktadan itibaren birden fazla çözüme sahip olabileceğinden bölümleme bazı değişkenlerin olası değerlerini iki aralığa bölerek yapılabilmektedir **(Hillier and Liebermann, 1974).**

Amaç fonksiyonunun en iyi değerinin sınırları bulunduktan sonra, amaç fonksiyonu değerine karşı gelen ve değeri tamsayı olmayan değişkenlerden biri belirlenerek değişkenin alabileceği iki komşu tamsayı değere göre uygun çözüm alanı alt bölümlere ayrılır **(Kara, 1986).**

R.J. Dakin tarafından yapılan bu düzenleme, değişik hesaplamalarda büyük kolaylıklar sağlamıştır. Dal ve sınır tekniği, kesme düzlemi tekniğinin aksine; hem bütünüyle tamsayılı doğrusal modellere, hem de karma tamsayılı doğrusal modellere doğrudan uygulanabilmektedir. Karma tamsayılı doğrusal modellerde sadece tamsayı olması istenilen değişkenler üzerinde dallandırma yapılır. Tekniğin temeli, öncelikle bir doğrusal model olarak problemi çözmektir **(Paradimitriou, 1987)**.

Doğrusal programlamanın dualite teorisi kullanılarak herhangi bir karma tamsayılı programlama problemi, tamsayılı programlama problemi olarak yazılabilir. Bu tamsayılı programlama problemini çözmek suretiyle karma tamsayılı programlama probleminin çözümü bulunur. Ayrıştırma algoritmasında sırasıyla doğrusal programlama ve tamsayılı programlama çözümleri yapılır. Doğrusal programlama çözümü, tamsayılı programlama için tek bir sınırlama verir. Yine bu optimal çözüm değeri, karma tamsayılı doğrusal programlama problemine optimal çözüm için bir üst sınır verir. Doğrusal programlama çözümünden elde edilen sınırlamalarla birlikte tamsayılı programlama problemi çözüldüğünde, karma tamsayılı doğrusal programlama problemi optimal çözümüne bir alt sınır elde edilir. Alt ve üst sınırlar birbirine eşit olduğunda, optimal karma tamsayılı programlama problemi çözümüne ulaşmış olur. Doğrusal programlamanın dualite teorisi gereğince , dual problem sınırlanmadığı zaman doğrusal programlama probleminin uygun çözümü yoktur ve böylece karma tamsayılı programlama problemi belirlenen x için optimal çözüme sahip değildir **(Salkın, 1975)**.

Bütünüyle tamsayılı doğrusal programlamadan karma tamsayılı doğrusal programlamaya geçiş dört aşamada gerçekleştirilir.

Öncelikle karar değişkenlerinin tamsayılı olmakla sınırlandırılmış kısıtı ihmal edilir, meydana gelen doğrusal programlama modelini simpleks ya da dual simpleks tekniğini kullanarak çözülmesi ile başlar. Ortaya çıkan sonuçta $j = 1,2, ...,I$ için bütün x_j 'ler tamsayı değerler taşırsa, bu arzulanan optimal çözümdür **(Hillier and Liebermann, 1974)**.

İlk aşama, dallanma değişkeninin seçimini kapsar. Önce, doğal sıralamadaki bir sonraki değişken $x_1, x_2, ... , x_n$ seçilir. Burada ele alınacak değişken, doğrusal programlama modelinin optimal çözümünde yer alan ancak tamsayı kısıtlamalı değişkenlerin tamsayı değerli olmayanıdır. Bu değişkenlerin birinin seçilmesinde kural doğal sıralamadaki ilk değişkeni seçmektir.

İkinci aşama, yeni daha küçük alt problemler yaratmak için dallanma değişkenine yeni değerler atanmasını içerir. Bunun için dallanma değişkenine göre dallandırma yapılır ve alt bölümler elde edilir. Bundan sonra yapılması gereken değişkene iki aralık verilerek iki alt problem yaratmaktır.

Sürekli bir değişken hiçbir zaman dallanma değişkeni olarak seçilemez. Kesikli değişkenlerin değerleri tamsayı ise ve amaç fonksiyonunun değeri de mevcut sınıra göre daha iyiyse, alt problem amaç değeri üzerinde yeni bir sınır verir **(Taha, 2000).**

Bunun nasıl yapıldığını görmek için mevcut alt problemin optimal doğrusal programlama modeli için x_j mevcut dallanma değişkeni ve x_j^* da x_j 'nin tamsayı olmayan değeri olduğunu kabul edilsin.

$[x_j^*]$, x_j dan küçük ya da eşit en büyük tamsayı olsun,

$$[x_j^*] \leq x_j \leq [x_j^*] + 1$$

alt problem için sınırlar,

$$x_j \leq [x_j^*] \quad \text{ve} \quad x_j \geq [x_j^*] + 1$$

şeklinde oluşturulabilir. Her eşitsizlik, yeni alt probleme bir kısıtlayıcı eklenmesine neden olmaktadır.

Oluşturulan bu alt problemlerden birinin seçiminde "en iyi sınır" ve "en yeni sınır" kurallarının etkili olduğu bilinir. En iyi sınır kuralına göre, maksimizasyon durumunda, alt sınır değeri en büyük olan alt problemin seçilmesi gerekmektedir. Çünkü böyle bir sette optimal çözümü bulma olasılığı daha yüksektir. En yeni sınır kuralında ise, araştırma dışı bırakılmamış en yeni bulunmuş setin seçilmesi önerilmektedir. Bu seçimin oldukça etkili olduğu söylenebilir **(Tulunay, 1980)**.

Bir doğrusal programlama probleminin optimal çözüm setinde, bir değişken değeri $x_j^* = 3\frac{1}{2}$ olarak bulunmuş ise, dal-sınır tekniği ile ilgili değişkenin tamsayılı değerini bulmak için;

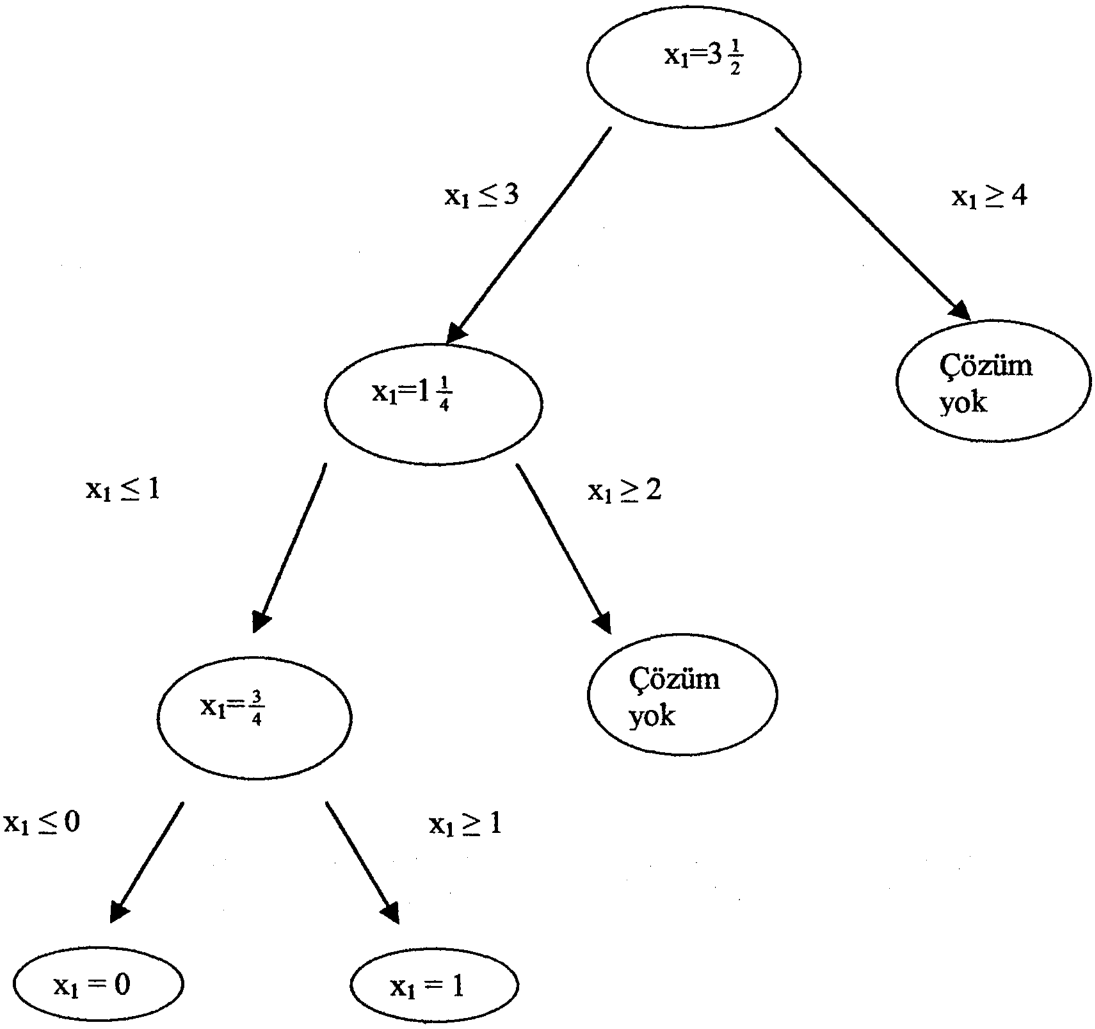
$$[x_j^*] \leq x_j \leq [x_j^*] + 1$$

den yararlanarak

$$3 \leq x_j \leq 4$$

bulunur. Buradan 3 ve 4 için dallandırma yapılır. Her bir dallandırmada modele ayrı ayrı $x_j^* \leq 3$ ve $x_j^* \geq 4$ kısıtları eklenerek optimal çözümleri simpleks çözüm tekniği ile bulunur.

Ayrıştırma algoritmasına göre, doğrusal programlama problemine yukarıda bulunan iki kısıt ilave edildiğinde tekrar dallanma meydana gelebilir. Bu durum Şekil 2 'de açık olarak gösterilmiştir.



Şekil 3.2. Dal-sınır tekniğine göre dallanma şeması

Şekil 3.2' de görüldüğü gibi j=1, dolayısıyla burada $x_1^* = 3\frac{1}{2}$ olsun. $x_1 \leq 3$ olduğundan yeni alt problem göz önüne getirilerek, alt problemin doğrusal programlama modeli simpleks çözüm tekniği ile çözüldüğünde $x_1^*=1\frac{1}{4}$ olarak bulunur. Bundan sonra x_1 dallanan değişken olarak tekrar ortaya çıkar ve oluşturulan iki yeni alt problemde $x_1 \leq 1$ ve $x_1 \geq 2$ ilave kısıtlara sahip olur (daha önceki ilave kısıt olan $x_1 \leq 3$ 'de devam eder). Sonra doğrusal programlama probleminde aynı işlemler tekrar edilerek ve $x_1 \leq 1$ kısıtı da ilave edilerek alt problemi çözüldüğünde $x_1^*= \frac{3}{4}$ olur. Burada x_1

dallanma değişkeni olarak tekrar işleme girer ve iki yeni alt problem $x_1 = 0$ ve $x_1 = 1$ ($x_1 \geq 1$ kısıtı ve $x_1 \leq 1$ kısıtı olduğundan) oluşturulur.

Üçüncü aşama sınırlama adımından oluşur. Sınırlama adımı, amaç fonksiyonunun maksimum değerine bir alt sınır getirir. Önce, bütünüyle tamsayılı doğrusal programlama problemi ve amaç fonksiyonun tamsayı katsayılarıyla sınırı elde etmek için z 'nin değeri optimal çözüm doğrusal programlama modeli için yuvarlanır. Çünkü herhangi bir mümkün çözümde alt problem tamsayı bir z 'ye sahip olmalıdır. Burada bazı değişkenler tamsayı kısıtlı değilse, sınır yuvarlanmamış z değeridir. Sınırlama adımı, her alt gruplama için alt sınırı bulur ve çıkan doğrusal programlama problemini yeni kısıtı da ilave ederek çözer **(Hillier and Liebermann, 1974).**

Bir maksimizasyon probleminde amaç fonksiyonunun en iyi çözümüne karşılık gelen değeri alt sınır olarak kabul edilir. Başlangıçta ise amaç fonksiyonu için altsınır $z = -\infty$ olarak kabul edilmektedir. Minimizasyon probleminde ise en iyi çözümüne karşılık gelen değer üst sınır olarak alınır ve başlangıç üst sınır değeri $z = \infty$ olarak kabul edilir. **(Taha, 2000).**

Karma tamsayılı doğrusal programlama algoritmasını elde etmek için dördüncü ve son adım sağlama aşamasıdır ve 3 optimalite testinin incelenmesini kapsar. Öncelikle, bütünüyle tamsayılı doğrusal programlama modelinde alt problemin doğrusal programlama modelinin çözümü tamsayıdır. Bu da çözümün mümkün olabileceğini gösterir ve bu yüzden alt problem için optimal çözüm olur. Fakat karma tamsayılı doğrusal programlama probleminde doğrusal programlama modelinin alt probleminin optimal çözümü için sadece tamsayı kısıtlamalı değişkeninin tamsayı olması gerekir. Bu da çözümün mümkün olabileceğini göstermek için yeterlidir.

Tamsayılı doğrusal programlama algoritmasında bu dört değişikliğin yapılması halinde, karma tamsayılı doğrusal programlama modeli için yeni karma algoritma elde edilmektedir **(Hillier and Liebermann, 1974).**

Başlangıç adımı : Elimizde bir maksimizasyon problemi olsun ve başlangıç alt sınır değeri $z = -\infty$ olarak kabul edilsin. Bütün modele aşağıda tarif edilen sınırlama, dallanma, sağlama ve optimalite testi uygulanır.

Ayrıştırma algoritmasına göre her iterasyon için adımlar,

1. Adım (Dallanma): Alt problemlerin içinden genellikle en son oluşturulan seçilir. Alt problemin optimal çözümünün tamsayılı olmayan değerini doğal sıralamaya göre seçip dallanma değişkenlerini tayin edin. x_j dallanma değişkeni ve x_j^* bu çözümde onun değeri olur. Alt problemin düğüm noktasından dallandırmaya başlanır. Kısıtlar eklenerek iki yeni alt problem oluşturulur.

$$x_j \leq [x_j^*] \quad \text{ve} \quad x_j \geq [x_j^*] + 1$$

2. Adım (Sınırlama): Her yeni alt problem için doğrusal programlama modeline simpleks tekniğini uygulanır ve optimal çözüm için çıkarılan z değeri kullanılarak alt sınır elde edilir.

Alt problem derinlemesine incelenmiş ve daha iyi bir çözüm bulunmuşsa alt sınır olarak bu yeni değer alınır ve alt problemlerin tümü incelenilinceye kadar devam edilir.

3. Adım (Sağlama): Her yeni alt problem için aşağıda verilen üç optimalite testini uygulanır ve bu testlerden herhangi birinde yeniden değerlenen alt problemi atılır.

Test 1 : Alt problem uygun bir çözüme yol açamazsa,

veya

Test 2 : Alt problem doğrusal programlama modelinin mümkün bir çözümünün olmadığını bulursa,

veya

Test 3: Alt problem doğrusal programlama modelinin bulunan optimal çözümünde tamsayılı olmakla sınırlandırılmış değişkenler için tamsayılı değerler bulursa,

yeni alt grup kontrol edilmiş olur.

Bütün alt setler incelendikten sonra bulunan çözüm arzulanan optimal çözümdür **(Hillier and Liebermann, 1974).**

4. KARMA TAMSAYILI DOĞRUSAL PROGRAMLAMA' NIN BİR KAĞIT AMBALAJ İŞLETMESİNDE UYGULAMA DENEMESİ

İşletmelerin karşılaştıkları mevcut sorunlardan biri; üretim ve pazar şartları altında, kapasiteden maksimum yararlanarak karı maksimum yapan ya da maliyeti minimuma indiren üretim planını tespit etmektir. Çünkü üretim planlaması, aylak kapasitenin en aza indirilmesi bakımından oldukça önemlidir.

Bu bölümde özel bir işletme için üretim planlaması çalışmasının matematiksel modelleme yoluyla nasıl yapıldığı anlatılmaktadır. Öncelikle işletme hakkında genel bilgi alınmıştır. Modelin karar değişkenleri ve parametreleri hakkında bilgi toplanarak, işletmenin üretim şekline göre problem belirlenmiştir. Elimizdeki verilere dayanarak karma tamsayılı bir doğrusal karar modeli kurulmuştur. Hangi üründen ne kadar üretilmeli ki kapasite kullanımı ve kar maksimum olsun problemine çözüm aranmıştır. Modelin çözümü sonunda elde edilen sonuçlar yorumlanarak işletme yöneticisine gerekli önerilerde bulunulmuştur. Böylece tamsayılı doğrusal programlamanın üretim planlaması aracı olarak kullanılmasının sağladığı yararları Turan Kağıt Ambalaj Sanayii 'nde yapılan bir uygulama ile ortaya koymaya çalışılmıştır.

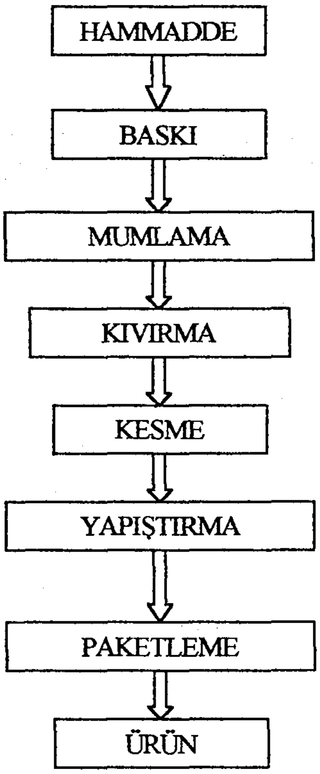
4.1. İşletmenin Tanıtımı

Eskişehir Organize Sanayi Bölgesi'nde bulunan fabrika 4.500 m^2 kapalı, toplam 9.543 m^2 alanda üretim yapmaktadır. Ayrıca firma yine Eskişehir Organize Sanayi Bölgesi'nde 3.000 m^2 alan üzerinde kurulu 2.000 m^2 kapalı alanı olan deposuyla hizmet vermektedir.

Turan Kağıt Ambalaj Sanayi ve Ticaret A.Ş. kağıt sektörüne 1961 yılında İstanbul'dan Eskişehir'e getirilen küçük bir kese kağıdı makinesi ile başlamıştır. 1982 yılında Turan Kağıt Ambalaj Sanayi ve Ticaret A.Ş. 5.000.000 TL sermaye ile Eskişehir'de kurulmuştur.

1987 yılında alüminyum folyo, ziftli kağıt gibi çeşitli endüstriyel atıklardan ve hurda kağıtlardan ürettiği dayanıklı, elastiki ve ısı geçirgenlik katsayısı düşük pres levhaları sanayi ürünü olarak yeniden geliştirilerek, tekrar kullanıma kazandırılması çalışmaları ile Eskişehir Sanayi Odası tarafından Teknoloji Geliştirme Ödülü' ne layık görülmüştür.

Turan Kağıt Ambalaj Sanayi 'nde üretim ana hammadde olan kağıdın ilk form verme makinesine bobin halinde takıldıktan sonra sırasıyla baskı, mumlama, kesme, yapıştırma ve paketleme ünitelerinde geçekleştirilen işlemlerden oluşmaktadır. Bu işlemlerin akış şeması Şekil 4.1 'de görülmektedir.



Şekil 4.1. Ürün oluşum akış şeması

İşletmede paketleme dışındaki bütün üniteler birbirine bağlı olarak çalışmaktadır. Bu nedenle de işletmedeki iş akışı seri üretimdir. Belirli bir model ambalaj üretimi bittikten sonra makine ayarları değiştirilerek başka bir modelin üretimine geçilmektedir. Programlanmış duruşların ve arızaların dışında üretim durmadan devam etmektedir.

Talep edilen başlıca ürünler; baskılı kraft torba, baskısız kraft torba, kraft mumlu ambalaj, baskılı beyaz sülfit ambalaj, baskılı normal sülfit ambalaj, 1.hamur baskılı mumlu, baskılı üzüm poşeti, 3. Hamur baskılı mumlu ambalaj, 1. Hamur özel baskılı mumlu ambalaj, mumlu normal sülfit ambalaj, mumlu beyaz sülfit ambalaj, 3. Hamur ambalaj, beyaz sülfit ambalaj, normal sülfit ambalaj, özel baskılı beyaz sülfit ambalaj, özel baskılı normal sülfit ambalaj, 3. Hamur baskılı ambalaj, 1. Hamur özel baskılı ambalaj, 1. Kalite kese kağıdı, 2. Kalite kese kağıdı, baskılı kese kağıdı, polietilenli beyaz sülfit ambalaj, polietilenli özel baskılı beyaz sülfit ambalaj, polietilenli baskılı normal sülfit ambalaj, baskılı sandık kenarı, 3.hamur kaplık, 1.hamur özel baskılı kaplık ve 3.hamur baskılı kaplık olmak üzere 28 çeşittir.

4.2. Problemin Tanıtımı

Amaç;

Maliyet ve satış fiyatları göz önünde bulundurularak, işletmeden göreli karlar alınmak suretiyle, talebe göre her ay hangi üründen ne kadar üretileceğine dair bir model kurulmasına çalışılmıştır. Uygun amaç fonksiyonu, karı maksimize edecek üretim miktarını bulmaktır.

Model için birim karın bulunması Eylül'02 rakamları kullanılarak yapılmıştır. Amaç fonksiyonunun katsayıları **(c_j)** göreli karlar olarak şirket yöneticilerinden alınmıştır. Her bir ürüne ait göreli karlar Tablo 4.1 'de görülmektedir.

4.2.1 Problemin Kısıtlayıcı Faktörleri

Modelin kısıtlayıcı koşulları talep kısıtı, tezgah zaman kısıtı, hammadde kısıtı ve negatif olmama kısıtıdır.

Talep Kısıtı;

Talep parametreleri **(k_j)** işletmeye gelen aylık talep miktarlarından alınmıştır. Belirli bir talep tahmini olmadığı için modelin testinde işletme yöneticilerinden alınan Eylül'02 taleplerine göre işlem yapılmıştır. Her bir ürüne ait talepler Tablo 4.1 'de verilmektedir.

Belli bir dönemin talebi, o dönem içinde karşılanmak zorundadır. Bu yüzden şirket maliyet yükselse bile müşteri talebini karşılamak istemektedir.

Tablo 4.1. Her bir ürüne ait Eylül'02 talebi ve birim karları

	ÜRÜN ADI	Eylül '02 Talepleri Adet	Kg	Br. Kar (cj)
x1	**baskılı kraft torba**	3000		5,0
x2	**baskısız kraft torba**	3500		6,0
x3	kraft mumlu ambalaj		700	21,2
x4	baskılı beyaz sülfit amb.		2750	19,2
x5	baskılı normal sülfit amb.		221	17,3
x6	1. Hamur baskılı mumlu amb.		2517	16,0
x7	**baskılı üzüm poşeti**	10000		2,0
x8	3. Hamur baskılı mumlu amb.		242	18,3
x9	1. Hamur özel baskılı mumlu		198	16,4
x10	mumlu normal sülfit amb.		210	14,7
x11	mumlu beyaz sülfit amb.		175	14,6
x12	3. Hamur ambalaj		250	17,0
x13	beyaz sülfit amb.		110	14,4
x14	normal sülfit amb.		260	19,5
x15	özel baskılı beyaz sülfit amb.		100	19,0
x16	özel baskılı normal sülfit amb.		175	18,8
x17	3. Hamur baskılı amb.		435	12,6
x18	1. Hamur özel baskılı amb.		250	17,5
x19	**1. Kalite kese kağıdı**	1000		3,0
x20	**2. Kalite kese kağıdı**	2100		2,0
x21	**baskılı kese kağıdı**	1500		5,0
x22	polietilen beyaz sülfit amb.		165	19,6
x23	polietilen öz. baskılı beyaz sülfit amb.		150	19,8
x24	polietilen baskılı normal sülfit amb.		240	17,3
x25	**baskılı sandık kenarı kağıdı**	2500		1,2
x26	3. Hamur kaplık		120	11,2
x27	1. Hamur özel baskılı kaplık		100	10,4
x28	3. Hamur baskılı kaplık		100	10

Tezgah Zaman Kısıtı;

Her tezgahtaki makine çeşidinin farklı olduğu ve her ürünün bu makinelerden geçme sürecinin ürün özelliğine göre değiştiği göz önüne alınarak, her ürünün makinelerden geçme zamanı birim başına saniye olarak hesaplanmıştır. Bir birim ürün için her makinenin ne kadar zaman harcadığı bulunmuştur. Her bir ürünün işlenmesi sırasında tezgahlarda kalma süreleri **(a_{ij})** Tablo 4.2 'de verilmiştir.

Tablo 4.2. Her bir ürünün işlenmesi sırasında tezgahlarda kalma süreleri (sn)

	ÜRÜN ADI	Baskı	Mumlama	Kıvırma	Kesme	Yapıştırma	Paketleme
x1	**baskılı kraft torba**	12,3	0	25	16	20	10
x2	**baskısız kraft torba**	0	0	25	16,4	20	10
x3	kraft mumlu ambalaj	0	16,5	0	30,5	0	8
x4	baskılı beyaz sülfit amb.	9,5	0	0	31,8	0	7
x5	baskılı normal sülfit amb.	8,7	0	0	28,9	0	7
x6	1. Hamur baskılı mumlu	5,6	8,9	0	27,6	0	8
x7	**baskılı üzüm poşeti**	8	0	15,4	10	12	10
x8	3. Hamur baskılı mumlu amb.	9,5	9,7	0	28,4	0	8,5
x9	1. Hamur özel baskılı mumlu	9,6	8,7	0	25,4	0	8,6
x10	mumlu normal sülfit amb.	0	12,8	0	29,75	0	8,5
x11	mumlu beyaz sülfit amb.	0	15,5	0	28,8	0	7
x12	3. Hamur ambalaj	0	0	0	24,92	0	7,4
x13	beyaz sülfit amb.	0	0	0	26,9	0	8
x14	normal sülfit amb.	0	0	0	25,6	0	8,2
x15	özel baskılı beyaz sülfit amb.	15,4	0	0	27,45	0	7,5
x16	özel baskılı normal sülfit amb.	14,7	0	0	25,98	0	7
x17	3. Hamur baskılı amb.	9,2	0	0	27,6	0	6,9
x18	1. Hamur özel baskılı amb.	8,6	0	0	24,5	0	8,3
x19	**1. Kalite kese kağıdı**	0	0	11,2	9,2	12	11
x20	**2. Kalite kese kağıdı**	0	0	10	8,5	15	10
x21	**baskılı kese kağıdı**	12	0	9,5	9	15	12
x22	polietilen beyaz sülfit amb.	0	0	0	32,65	0	7
x23	polietilen öz. baskılı beyaz sülfit amb.	20,2	0	0	33,5	0	7,5
x24	polietilen baskılı normal sülfit amb.	12,5	0	0	33,25	0	7
x25	**baskılı sandık kenarı kağıdı**	20	0	8	24,8	0	12
x26	3. Hamur kaplık	0	0	0	21,6	0	9
x27	1. Hamur özel baskılı kaplık	25	0	0	23	0	8,7
x28	3. Hamur baskılı kaplık	23,5	0	0	22,3	0	11

Tezgah zaman kısıtında, tezgahların aylık normal mesai süreleri (**b_i**) hesaplanırken, bir aylık süre 22 işgünü olarak alınmıştır. Hafta içi ve hafta sonu (cumartesi) 8:30 – 12:00 ve 13:30 – 18:00 olmak üzere 8 saat normal mesai vardır. Günlük mesai saati, aylık toplam işgünü ile çarpılarak tezgahların bir ayda kaç saat çalıştığı bulunur. Bu değer saniyeye çevrilerek makine sayısı ile çarpıldığında tezgahların aylık normal mesai süreleri bulunmuş olur.

22 gün x 8 saat = 176 saat/ay

176 saat/ay x 3600 = 633600 sn/ay

Tezgahların işçi ve makine sayıları ile aylık normal mesai süreleri Tablo 4.3 'de görülmektedir.

Tablo 4.3. Tezgahların işçi ve makine sayıları ile aylık çalışma süreleri

	İşçi Sayısı	**Makine Sayısı**	**Aylık Mesai (sn/ay)**
Baskı	2	1	633600
Mumlama	2	1	633600
Kıvırma	2	1	633600
Kesme	2	1	633600
Yapıştırma	4	1	633600
Paketleme	6	1	633600

Hammadde kısıtı;

Bir birim ürünü işlemek için kullanılan hammadde miktarları her ürün grubu için farklılık göstereceğinden, her ürün grubu için ne kadar hammadde kullanılacağı ayrı ayrı hesaplanmıştır. Baskılı ambalajlarda; kağıt (kraft, sülfit, 1. hamur, 3.hamur v.b.) ve binde 2 oranında boya hammaddesi kullanılmaktadır. Baskılı mumlu ambalajlardaki hammadde kullanım oranı ise kağıt, binde 2 boya ve % 30 parafinden oluşmaktadır. Bir birim ürünü işlemek için kullanılan hammadde **(d_{ij})** miktarları aşağıdaki Tablo 4.4 'de verilmektedir.

Tablo 4.4. Bir birim ürün işlemek için kullanılan kağıt, boya ve parafin miktarı

	ÜRÜN ADI	Kağıt (gr)	Boya (gr)	Parafin (gr)
x1	**baskılı kraft torba**	25	0.5	0
x2	**baskısız kraft torba**	25	0	0
x3	Kraft mumlu ambalaj	25	0	3
x4	baskılı beyaz sülfit amb.	10,1	0,02	0
x5	baskılı normal sülfit amb.	10,3	0,02	0
x6	1. Hamur baskılı mumlu	5	0,02	3,4
x7	**baskılı üzüm poşeti**	18	0,05	0
x8	3. Hamur baskılı mumlu amb.	16	0,02	3,4
x9	1. Hamur özel baskılı mumlu	14,6	0,02	3,2
x10	mumlu normal sülfit amb.	18,2	0	3,3
x11	mumlu beyaz sülfit amb.	12	0	3
x12	3. Hamur ambalaj	11,3	0	0
x13	beyaz sülfit amb.	14	0	0
x14	normal sülfit amb.	12,6	0	0
x15	özel baskılı beyaz sülfit amb.	13	0,02	0
x16	özel baskılı normal sülfit amb.	15	0,02	0
x17	3. Hamur baskılı amb.	15,3	0,02	0
x18	1. Hamur özel baskılı amb.	13,3	0,02	0
x19	**1. Kalite kese kağıdı**	5,5	0	0
x20	**2. Kalite kese kağıdı**	5	0	0
x21	**baskılı kese kağıdı**	5,2	0,03	0
x22	polietilen beyaz sülfit amb.	14,8	0	0
x23	polietilen öz. baskılı beyaz sülfit amb.	15,1	0,02	0
x24	polietilen baskılı normal sülfit amb.	14,3	0,02	0
x25	**baskılı sandık kenarı kağıdı**	3	0,05	0
x26	3. Hamur kaplık	16	0	0
x27	1. Hamur özel baskılı kaplık	17	0,8	0
x28	3. Hamur baskılı kaplık	16,3	0,6	0
MAKSİMUM	KAPASİTE	8000000	50000	120000

Hammadde kısıtında, işletme tarafından bir aylık süre için temin edilebilecek hammadde miktarları **(f_i)** işletme yöneticilerinden alınmıştır.

Negatif Olmama Kısıtı;

Karar değişkenleri üretim miktarları olduğundan negatif değer almaları söz konusu değildir. Çünkü ürünlerin sayısı ya da ürünleri meydana getiren elementler daima pozitif değerler alır. Karar değişkenleri sıfırdan büyük ya da eşit olabilmektedir.

4.2.2. Problemin Karar Değişkenleri

x_j karar değişkenleri, belirlenen planlama döneminde talepler doğrultusunda üretimine karar verilen ürünlerden ne kadar üretileceğini göstermektedir.

$\mathbf{x_1}$ = baskılı kraft torba miktarı **(adet)**

$\mathbf{x_2}$ = baskısız kraft torba miktarı **(adet)**

x_3 = kraft mumlu ambalaj miktarı **(kg)**

x_4 = baskılı beyaz sülfit ambalaj miktarı **(kg)**

x_5 = baskılı normal sülfit ambalaj miktarı **(kg)**

x_6 = 1. Hamur baskılı mumlu ambalaj miktarı **(kg)**

$\mathbf{x_7}$ = baskılı üzüm poşeti miktarı **(adet)**

x_8 = 3. Hamur baskılı mumlu ambalaj miktarı **(kg)**

x_9 = 1. Hamur özel baskılı mumlu ambalaj miktarı **(kg)**

x_{10} = mumlu normal sülfit ambalaj miktarı **(kg)**

x_{11} = mumlu beyaz sülfit ambalaj miktarı **(kg)**

x_{12} = 3. Hamur ambalaj miktarı **(kg)**

x_{13} = beyaz sülfit ambalaj miktarı **(kg)**

x_{14} = normal sülfit ambalaj miktarı **(kg)**

x_{15} = özel baskılı beyaz sülfit ambalaj miktarı **(kg)**

x_{16} = özel baskılı normal sülfit ambalaj miktarı **(kg)**

x_{17} = 3. Hamur baskılı ambalaj miktarı **(kg)**

x_{18} = 1. Hamur özel baskılı ambalaj miktarı **(kg)**

$\mathbf{x_{19}}$ = 1. Kalite kese kağıdı miktarı **(adet)**

$\mathbf{x_{20}}$ = 2. Kalite kese kağıdı miktarı **(adet)**

$\mathbf{x_{21}}$ = baskılı kese kağıdı miktarı **(adet)**

x_{22} = polietilen beyaz sülfit ambalaj miktarı **(kg)**

x_{23} = polietilen öz. baskılı beyaz sülfit ambalaj miktarı **(kg)**

x_{24} = polietilen baskılı normal sülfit ambalaj miktarı **(kg)**

$\mathbf{x_{25}}$ = baskılı sandık kenarı kağıdı miktarı **(adet)**

x_{26} = 3. hamur kaplık miktarı **(kg)**

x_{27} = 1. hamur özel baskılı kaplık miktarı **(kg)**

x_{28} = 3. hamur baskılı kaplık **(kg)**

Çalışmamızın konusu olan üretim planlaması probleminin uygulamasında karar değişkenlerinden bazılarının tamsayılı olması gerektiğinden, problem tamsayılı programlama türlerinden karma tamsayılı programlama modeline uygun düşmektedir.

4.3. Modelin Kurulması

Modelin çözümü için amaç fonksiyonu, karar değişkenleri ve kısıtlayıcılar matematiksel olarak ifade edilirse,

$\max z = \Sigma c_j x_j$ (amaç fonksiyonu)

$x_j \geq k_j$ (talep kısıtı) $(j = 1, 2, ..., 28)$

$\sum_i a_{ij} x_j \leq b_i$ (tezgah zaman kısıtı) $(i = 1, 2, ..., 6)$

$\sum_i d_{ij} x_j \leq f_i$ (hammadde kısıtı) $(i = 1, 2, 3)$

$x_j \geq 0$ $(j = 1, 2, ..., 28)$

ve

$x_1, x_2, x_7, x_{19}, x_{20}, x_{21}, x_{25}$ tamsayı

olarak yazılır.

Burada;

c_j : j inci üründen elde edilen birim başına göreli karlar (j = 1, 2, ..., 28)

x_j : planlama döneminde j inci mamülden üretilecek miktar (j = 1, 2, ..., 28)

k_i : j inci ürün için aylık talep miktarı (j = 1, 2, ..., 28)

a_{ij} : i inci tezgahta j inci ürünün kalma süresi (sn) (j = 1, 2, ..., 28),
(i = 1, 2, ..., 6)

b_i : i inci tezgahın normal mesai süresi (sn) (i = 1, 2, ..., 6)

f_i : planlama döneminde sağlanabilecek i inci hammadde miktarı (i = 1, 2, 3)

d_{ij} : j inci ürünün i inci hammaddeden kullandığı miktar (j = 1, 2, ..., 28),
(i = 1, 2, 3)

olarak ifade edilir. Problemin parametreleri yerine konulduğunda model aşağıdaki şekilde yazılabilir.

$$\text{Max } z = 5\,x_1 + 6\,x_2 + 21{,}2\,x_3 + 19{,}2\,x_4 + 17{,}3\,x_5 + 16\,x_6 + 2\,x_7 + 18{,}3\,x_8 + 16{,}4\,x_9 + 14{,}7\,x_{10} + 14{,}6\,x_{11} + 17\,x_{12} + 14{,}4\,x_{13} + 19{,}5\,x_{14} + 19\,x_{15} + 18{,}8\,x_{16} + 12{,}6\,x_{17} + 17{,}5\,x_{18} + 3\,x_{19} + 2\,x_{20} + 5\,x_{21} + 19{,}6\,x_{22} + 19{,}8\,x_{23} + 17{,}3\,x_{24} + 1{,}2\,x_{25} + 11{,}2\,x_{26} + 10{,}4\,x_{27} + 10\,x_{28}$$

k.a.

$$
\left.
\begin{array}{l}
x_1 \geq 3000 \\
x_2 \geq 3500 \\
x_3 \geq 700 \\
x_4 \geq 2750 \\
x_5 \geq 221 \\
x_6 \geq 2517 \\
x_7 \geq 10000 \\
x_8 \geq 242 \\
x_9 \geq 198 \\
x_{10} \geq 210 \\
x_{11} \geq 175 \\
x_{12} \geq 250 \\
x_{13} \geq 110 \\
x_{14} \geq 260 \\
x_{15} \geq 100 \\
x_{16} \geq 175 \\
x_{17} \geq 435 \\
x_{18} \geq 250 \\
x_{19} \geq 1000 \\
x_{20} \geq 2100 \\
x_{21} \geq 1500 \\
x_{22} \geq 165 \\
x_{23} \geq 150 \\
x_{24} \geq 240 \\
x_{25} \geq 2500 \\
x_{26} \geq 120 \\
x_{27} \geq 100 \\
x_{28} \geq 100
\end{array}
\right\} \text{talep kısıtları}
$$

$$12{,}3\ x_1 + 9{,}5\ x_4 + 8{,}7\ x_5 + 5{,}6\ x_6 + 8\ x_7 + 9{,}5\ x_8 + 9{,}6\ x_9 + 15{,}4\ x_{15} + 14{,}7\ x_{16} + 9{,}2\ x_{17} + 8{,}6\ x_{18} + 12\ x_{21} + 20{,}2\ x_{23} + 12{,}5\ x_{24} + 20\ x_{25} + 25\ x_{27} + 23{,}5\ x_{28} \leq 633600 \qquad \text{(baskı)}$$

$$16{,}5\ x_3 + 8{,}9\ x_6 + 9{,}7\ x_8 + 8{,}7\ x_9 + 12{,}8\ x_{10} + 15{,}5\ x_{11} \leq 633600 \qquad \text{(mumlama)}$$

$$25\ x_1 + 25\ x_2 + 15{,}4\ x_7 + 11{,}2\ x_{19} + 10\ x_{20} + 9{,}5\ x_{21} + 8\ x_{25} \leq 633600 \qquad \text{(kıvırma)}$$

$$16\ x_1 + 16{,}4\ x_2 + 30{,}5\ x_3 + 31{,}8\ x_4 + 28{,}9\ x_5 + 27{,}6\ x_6 + 10\ x_7 + 28{,}4\ x_8 + 25{,}4\ x_9 + 29{,}75\ x_{10} + 28{,}8\ x_{11} + 24{,}92\ x_{12} + 26{,}9\ x_{13} + 25{,}6\ x_{14} + 27{,}45\ x_{15} + 25{,}98\ x_{16} + 27{,}6\ x_{17} + 24{,}5\ x_{18} + 9{,}2\ x_{19} + 8{,}5\ x_{20} + 9\ x_{21} + 32{,}65\ x_{22} + 33{,}5\ x_{23} + 33{,}25\ x_{24} + 24{,}8\ x_{25} + 21{,}6\ x_{26} + 23\ x_{27} + 22{,}3\ x_{28} \leq 633600 \qquad \text{(kesme)}$$

$$20\ x_1 + 20\ x_2 + 12\ x_7 + 12\ x_{19} + 15\ x_{20} + 15\ x_{21} \leq 633600 \qquad \text{(yapıştırma)}$$

$$10\ x_1 + 10\ x_2 + 8\ x_3 + 7\ x_4 + 7\ x_5 + 8\ x_6 + 10\ x_7 + 8{,}5\ x_8 + 8{,}6\ x_9 + 8{,}5\ x_{10} + 7\ x_{11} + 7{,}4\ x_{12} + 8\ x_{13} + 8{,}2\ x_{14} + 7{,}5\ x_{15} + 7\ x_{16} + 6{,}9\ x_{17} + 8{,}3\ x_{18} + 11\ x_{19} + 10\ x_{20} + 12\ x_{21} + 7\ x_{22} + 7{,}5\ x_{23} + 7\ x_{24} + 12\ x_{25} + 9\ x_{26} + 8{,}7\ x_{27} + 11\ x_{28} \leq 633600 \qquad \text{(paketleme)}$$

$$25\ x_1 + 25\ x_2 + 25\ x_3 + 10{,}1\ x_4 + 10{,}3\ x_5 + 5\ x_6 + 18\ x_7 + 16\ x_8 + 14{,}6\ x_9 + 18{,}2\ x_{10} + 12\ x_{11} + 11{,}3\ x_{12} + 14\ x_{13} + 12{,}6\ x_{14} + 13\ x_{15} + 15\ x_{16} + 15{,}3\ x_{17} + 13{,}3\ x_{18} + 5{,}51\ x_{19} + 5\ x_{20} + 5{,}2\ x_{21} + 14{,}8\ x_{22} + 15{,}1\ x_{23} + 14{,}3\ x_{24} + 3\ x_{25} + 16\ x_{26} + 17\ x_{27} + 16{,}3\ x_{28} \leq 600000 \qquad \text{(kağıt)}$$

$$0.5\,x_1 + 0.02\,x_4 + 0.02\,x_5 + 0.02\,x_6 + 0.05\,x_7 + 0.02\,x_8 + 0.02\,x_9 + 0.02\,x_{15} + 0.02\,x_{16} + 0.02\,x_{17} + 0.02\,x_{18} + 0.03\,x_{21} + 0.02\,x_{23} + 0.02\,x_{24} + 0.05\,x_{25} + 0.08\,x_{27} + 0.06\,x_{28} \le 50000 \quad \text{(boya)}$$

$$3\,x_3 + 3{,}4\,x_6 + 3{,}4\,x_8 + 3{,}2\,x_9 + 3{,}3\,x_{10} + 3\,x_{11} \le 120000 \quad \text{(parafin)}$$

ve

$$x_j \ge 0 \quad (j = 1, 2, ..., 28)$$

x_1, x_2, x_7, x_{19}, x_{20}, x_{21} ve x_{25} tamsayı

4.4. Modelin Çözümü ve Yorumlanması

Günümüzde artan rekabet koşullarında firmaların faaliyetlerini sürdürüp ayakta kalabilmeleri için rekabet koşullarına adapte olabilmeleri gerekmektedir. Bunun için firmanın hammadde giderlerini azaltması pek kolay olmadığı için, mevcut kaynaklarla daha fazla ürün elde ederek göreli kar artırılmalıdır.

Bu çalışmanın amacı tamsayılı doğrusal programlama tekniğinin üretim planlaması aracı olarak kullanılmasının sağladığı yararları Turan Kağıt Ambalaj Sanayii' inde yapılan bir uygulama ile ortaya koymaktır. Bunun için öncelikle işletmede araştırmalar yapılmış ve yetkili kişilerle görüşülerek gerekli bilgiler alınmıştır.

İşletmenin amacı bir taraftan talepleri karşılamaya çalışırken, diğer taraftan da karı maksimize edecek üretim planını bulmaktır. Yani belirli koşullar altında eldeki kıt kaynakları en uygun biçimde değerlendirirken, karı da maksimize etmelidir.

Problemin karar değişkenlerinin bir kısmı (baskılı kraft torba, baskısız kraft torba, baskılı üzüm poşeti, 1.kalite kese kağıdı, 2.kalite kese kağıdı, baskılı kese kağıdı, sandık kenarı kağıdı) tamsayılı olduğundan, problem karma tamsayılı doğrusal programlama modeli olarak ele alınıp çözülmüştür.

Bütün kısıtlar göz önüne alınarak amaç fonksiyonu karın en büyüklenmesi olan bir karma tamsayılı doğrusal programlama modeli geliştirilmiştir. Hangi üründen ne kadar üretilmeli ki, kapasite kullanımı ve kar maksimum olsun problemine çözüm aranmıştır. Talepler aylık olarak geldiğinden planlama dönemi, bir aylık süreyi kapsamaktadır.

Model, üretimi karma tamsayılı olan işletme için kar maksimizasyonunu sağlayan ürün karmasını verecek şekilde bilgisayar ortamında QSB+ paket programı kullanılarak çözümlenmiştir. Eylül'02 talepleri göz önüne alınarak elde edilen karma tamsayılı doğrusal programlama problemi 28 karar değişkeni ve 37 kısıttan oluşmaktadır. Üretim planlaması için önerilen karma tamsayılı doğrusal programlama modeli dal-sınır tekniğine göre çözüm yapan QSB+ paket programında 30 iterasyonda çözülmüştür.

Çözüm sonucunda Eylül'02 karı 27.814.840.000 TL olarak bulunmuştur. Modelin çözüm sonuçlarının QSB+ paket programındaki çıktıları EK-1' dedir. Bu sonuçlara göre, toplam karı maksimum yapmak için hangi üründen ne kadar üretilmesi gerektiğini gösteren çözümler aşağıda Tablo 4.5' de verilmektedir.

Tablo 4.5. Üretilen ürün miktarı çözüm sonuçları

	ÜRÜN ADI	Üretilen ürün miktarı
x1	**baskılı kraft torba**	3000
x2	**baskısız kraft torba**	3500
x3	Kraft mumlu ambalaj	700
x4	baskılı beyaz sülfit amb.	2750
x5	baskılı normal sülfit amb.	221.5
x6	1. Hamur baskılı mumlu	2517
x7	**baskılı üzüm poşeti**	10000
x8	3. Hamur baskılı mumlu amb.	242
x9	1. Hamur özel baskılı mumlu	198.00002
x10	mumlu normal sülfit amb.	210.50002
x11	mumlu beyaz sülfit amb.	175
x12	3. Hamur ambalaj	250.75
x13	beyaz sülfit amb.	110
x14	normal sülfit amb.	2251.9275
x15	özel baskılı beyaz sülfit amb.	100
x16	özel baskılı normal sülfit amb.	175.6
x17	3. Hamur baskılı amb.	435
x18	1. Hamur özel baskılı amb.	254.92
x19	**1. Kalite kese kağıdı**	1000
x20	**2. Kalite kese kağıdı**	2100
x21	**baskılı kese kağıdı**	1500
x22	polietilen beyaz sülfit amb.	165
x23	polietilen öz. baskılı beyaz sülfit amb.	187.2
x24	polietilen baskılı normal sülfit amb.	240
x25	**baskılı sandık kenarı kağıdı**	2500
x26	3. Hamur kaplık	120
x27	1. Hamur özel baskılı kaplık	100
x28	3. Hamur baskılı kaplık	100

(x_1, x_2, x_7, x_{19}, x_{14}, x_{20}, x_{21} ve $x_{25} \geq 0$ ve tamsayı)

Max z = 27.814.840.000 TL

Çözüm sonuç tablosunda taleplerin hepsinin karşılandığı, hatta x_5, x_9, x_{10}, x_{12}, x_{14}, x_{16}, x_{18} ve x_{23} karar değişkenlerinin fazla üretildiği görülmektedir. Özellikle normal sülfit ambalajda çok fazla kullanılmayan artık kapasite vardır. Taleplerin ekonomik nedenlere de bağlı olarak düzenli olmadığı ve bazı ürünlere (üzüm poşeti, kaplık gibi) sadece belli mevsimlerde ya da belli dönemlerde talep geldiği için fazla üretilen ürünlerin elde kalma ihtimali vardır.

Tezgahların çalışma sürelerini incelendiğinde, tezgahlarda kullanılmayan aylak kapasite olduğu görülmektedir. Aslında bunların tam kapasite ile çalışmamaları üretimin özelliği gereğidir. Çünkü aynı anda yalnızca bir makine çalışabilmektedir. Tablo 4.6' da tezgahların kullanılmayan aylak süreleri gösterilmektedir.

Tablo 4.6. Aylak tezgah çalışma süreleri

Tezgah Adı	**Aylak Tezgah Süreleri (sn/ay)**
Baskı	381168.47
Mumlama	590171.81
Kıvırma	250650
Kesme	0
Yapıştırma	317600
Paketleme	300032.16

Bu sonuçlar göz önüne alınarak kesme ünitesinin tam kapasite ile çalıştığı görülmektedir. Ani talep değişikliklerinde bu tezgahta oluşabilecek sıkışıklıkları önlemek için duruma göre fazla mesai kullanılması, vardiya konulması ya da makine sayısının artırılması önerilebilir. Buna karşılık diğer tezgahlardaki aylak süreleri önlemek için, bu tezgahlardaki işçi sayıları tekrar gözden geçirilerek azaltılabilir.

x_j' nin indirgenmiş maliyeti = x_j' nin maliyeti - x_j' nin karı

Modelin optimum çözümünde pozitif değer alan karar değişkenlerinin indirgenmiş maliyetlerinin sıfır olduğu görülmektedir. Sıfır değeri almış bir indirgenmiş maliyet ekonomik olarak arzu edilen bir maliyettir. Bu, maliyetin kara eşit olduğu bir denge noktasına ulaşıldığı anlamına gelir. Bir değişkenin indirgenmiş maliyeti pozitifse, bu durumda tüketilen kaynağın birim maliyeti birim kardan daha büyük olacak ve üretim gerçekleştirilemeyecektir.

EK-1' in sonuçlarından çıkarılan bir başka kavram da gölge fiyat (dual price) kavramıdır. **Gölge fiyat**, bir kaynağın kullanımındaki artış ya da azalışın amaç fonksiyonu üzerindeki katkısını ifade eder. Bir faaliyetin gölge fiyatının pozitif bir değer alması, sağ taraf sabitlerinin belirlenen sınırlar içinde artırılması amaç fonksiyonunun değerini yükseltecektir anlamına gelir.

Modelin çözümünde kesme ünitesi ve kağıt hammaddesi kısıtlarının gölge fiyatlarının sırasıyla 0.76 ve 1.99 olmak üzere pozitif değer aldıkları görülmektedir. Bu kısıtların gölge fiyatları amaç fonksiyonunun optimum değerini birim başına 760 TL ve 1990 TL artıracaktır. Bu ikisi dışındaki kısıtlara bakıldığında gölge fiyatlarının sıfır da değer aldıkları da görülmektedir. Buradan anlaşılan, gölge fiyatı sıfır olan kaynaklardaki artışların optimum çözüm üzerinde hiçbir etkisi olmadığıdır.

Baskı, mumlama, kıvırma, yapıştırma, paketleme tezgah kısıtları ile boya, parafin hammadde kısıtlarının gölge fiyatları sıfır ve üst sınırları sonsuz olduğundan bu kapasiteler istenildiği kadar artırılabilir. Bu kısıtların sayılarının artması kazanç sağlamayacaktır, zaten elimizde bir sürü kullanılmayan aylak baskı, mumlama, kıvırma, yapıştırma, paketleme tezgah süreleri ile boya, parafin hammadde miktarları vardır.

Hammadde kısıtlarına bakıldığında, boya ve parafinde kullanılmayan aylak kapasite olduğu görülmektedir. Bunun yanında kağıt hammaddesi tam kapasite ile kullanılmaktadır. Gölge fiyatının en yüksek olmasından dolayı en değerli girdi kağıt hammaddesidir. Bu nedenle işletme elinde yeterli miktarda kağıt hammaddesi bulundurmazsa talepleri yetiştiremediğinden dolayı zarar edebilir. Aşağıdaki Tablo 4.7' de kullanılmayan aylak hammadde miktarları görülmektedir.

Tablo 4.7. Aylak hammadde miktarı

Hammadde	Aylak Hammadde Miktarı
Kağıt	0
Boya	46419.43
Parafin	106666.15

Karma tamsayılı doğrusal programlama modelinin duyarlılık analizi ile ilgili çıktıları EK-2' de verilmektedir. Duyarlılık analizinde karar değişkenlerinin amaç fonksiyonunda alabileceği değerler için sınırlar verilmiştir. Bu sınırlara göre üretim planlamasının yapısını bozmadan planlamalarda değişiklikler yapılabilir. Böylece üretim planlamasında elde edilen değerler kontrol altında tutularak, karın artışı için gerekli düzenlemeler yapılmış olur.

EK-1' deki çözümde indirgenmiş maliyetler (opportunity cost) başlığı altında değerler verilmiştir. **İndirgenmiş maliyet**; bir karar değişkenini kar elde etmek amacıyla üretirken kaynakların tüketildiği bir ekonomik faaliyet olarak tanımlanabilir. Bu ifade aşağıdaki şekilde açık olarak yazılabilir.

Talep kısıtlarının gölge fiyatlarına bakıldığında çoğunun negatif oldukları görülür. Bu kısıtlarda sağ taraf sabitlerinin belirlenen sınırlar içinde artırılması, amaç fonksiyonunun optimum değeri üzerinde ters etki yapacaktır. Gölge fiyatları sıfır olan x_5, x_9, x_{10}, x_{12}, x_{14}, x_{16}, x_{18}, x_{23} ürünlerinin üretimlerinin artırılması amaç fonksiyonunun optimum değerinde herhangi bir değişiklik yapmayacaktır.

Amaç fonksiyonunun katsayılarındaki ve sağ taraf sabitlerindeki değişiklikler belirlenen alt ve üst sınırlar içerisinde yapılabilir. Herhangi bir karar değişkeninin amaç fonksiyonu katsayısındaki değişiklik, karar değişkeninin optimum çözümdeki değerini değiştirmez. Fakat indirgenmiş maliyeti, gölge fiyatı ve amaç fonksiyonunun optimum çözüm değeri değişebilir. Bununla birlikte herhangi bir kısıtın sağ taraf sabitindeki değişiklik kısıtın indirgenmiş maliyetini ve gölge fiyatını değiştirmez, fakat karar değişkeninin değerini ve amaç fonksiyonunun optimum çözümünü değiştirebilir.

SONUÇ

Karma tamsayılı doğrusal programlama çözüm tekniklerinden olan dal-sınır tekniğinin bir işletmede uygulanabilirliğini ortaya koymak amacıyla yapılan bu çalışma için, üretimi karma olan bir kağıt ambalaj işletmesi seçilmiştir. İşletmede toplanan veriler sonucunda kurulan modelde talep kısıtı, tezgah zaman kısıtı ve hammadde kısıtı olmak üzere toplam 38 kısıt bulunmaktadır. Kurulan model bu kısıtları sağlarken, kapasiteyi en uygun bir biçimde değerlendirerek maksimum karı bulmaya çalışmaktadır.

Birçok endüstri problemlerinde yalnızca optimal çözümün bulunması yeterli olmamaktadır. Çünkü uygulamalarda ya bilgi eksikliği ya da bilgi yanlışlığı yüzünden, çözüm sonucunda sapmalar olabilir. Duyarlılık analizi; birim kar, kapasite ve pazar sınırlarındaki değişmeler gibi parametrelerdeki gerçekleşebilir değişikliklerin modelin sonuçlarına etkilerini araştırır.

Modelin çözümünde kesme ünitesinin ve kağıt hammaddesinin gölge fiyatlarının pozitif değer aldıkları görülmektedir. Bu iki kısıtta yapılacak artış amaç fonksiyonunun optimum değerini artıracaktır.

Sonuçlar göz önüne alındığında kesme ünitesinin tam kapasite ile çalıştığı görülmektedir. Ani talep değişikliklerinde kesme ünitesinde oluşabilecek sıkışıklığı önleyebilmek için işletmeye kesme ünitesindeki makine sayısının artırılması önerilmiştir. Ayrıca kağıt hammaddesinin de tam kapasite ile kullanıldığı göz önüne alınırsa; elde bulundurmamadan dolayı talepleri yetiştirememek gibi bir durumla karşılaşmamak için kağıt hammaddesi stoku artırılması önerisinde bulunulmuştur. Diğer kısıtlardaki artış amaç fonksiyonunun optimum değerini artırmayacağından, bu kısıtlardaki sınırlarda herhangi bir değişiklik önerilmemiştir

Önerilen üretim planı, üretim sisteminde değişiklik yapılıp yapılamayacağı, ani talep değişikliklerinden doğan sıkışıklıklarda ne şekilde karar alınacağı konularında işletmeye yardımcı olmaktadır.

EKLER

Ek-1: Modelin QSB+ paket programındaki çözüm sonuçları

Summarized Results for FATMA TURAN   Page : 1

Variable No.	Names	Solution	Opportunity Cost	Variable No.	Names	Solution	Opportunity Cost
1	X1	+3000.0000	0	16	X16	+175.60000	0
2	X2	+3500.0000	0	17	X17	+435.00000	0
3	X3	+700.00000	0	18	X18	+254.92000	0
4	X4	+2750.0000	0	19	X19	+1000.0000	0
5	X5	+221.50000	0	20	X20	+2100.0000	0
6	X6	+2517.0000	0	21	X21	+1500.0000	0
7	X7	+10000.000	0	22	X22	+165.00000	0
8	X8	+242.00000	0	23	X23	+187.20000	0
9	X9	+198.00002	0	24	X24	+240.00000	0
10	X10	+210.50002	0	25	X25	+2500.0000	0
11	X11	+175.00000	0	26	X26	+120.00000	0
12	X12	+250.75000	0	27	X27	+100.00000	0
13	X13	+110.00000	0	28	X28	+100.00000	0
14	X14	+2251.9275	0				
15	X15	+100.00000	0				

Maximized OBJ. function = 278148.4 Iters. = 30

Summarized Results for FATMA TURAN   Page : 2

Variable No.	Names	Slack	Dual Price	Variable No.	Names	Slack	Dual Price
29	S1	0	+7.1874986	44	A8	0	-3.3328090
30	A1	0	-7.1874986	45	S9	+.000020	0
31	S2	0	+6.4921861	46	A9	0	0
32	A2	0	-6.4921861	47	S10	+.500020	0
33	S3	0	+2.0324206	48	A10	0	0
34	A3	0	-2.0324206	49	S11	0	+7.3374977
35	S4	0	+5.0226541	50	A11	0	-7.3374977
36	A4	0	-5.0226541	51	S12	+.750000	0
37	S5	+.500000	0	52	A12	0	0
38	A5	0	0	53	S13	0	+6.0902333
39	S6	0	+5.0234365	54	A13	0	-6.0902333
40	A6	0	-5.0234365	55	S14	+1991.9275	0
41	S7	0	+5.6171870	56	A14	0	0
42	A7	0	-5.6171870	57	S15	0	+1.9091797
43	S8	0	+3.3328090	58	A15	0	-1.9091797

Maximized OBJ. function = 278148.4 Iters. = 30

Summarized Results for FATMA TURAN			Page : 3		
Variable No. Names	Slack	Dual Price	Variable No. Names	Slack	Dual Price
59 S16	+.600000	0	74 A23	0	0
60 A16	0	0	75 S24	0	+8.0271463
61 S17	0	+8.4234362	76 A24	0	-8.0271463
62 A17	0	-8.4234362	77 S25	0	+17.690622
63 S18	+4.92000	0	78 A25	0	-17.690622
64 A18	0	0	79 S26	0	+5.2531247
65 S19	0	+4.0078120	80 A26	0	-5.2531247
66 A19	0	-4.0078120	81 S27	0	+7.1195302
67 S20	0	+4.4746094	82 A27	0	-7.1195302
68 A20	0	-4.4746094	83 S28	0	+6.9863267
69 S21	0	+1.8554683	84 A28	0	-6.9863267
70 A21	0	-1.8554683			
71 S22	0	+5.2701154			
72 A22	0	-5.2701154			
73 S23	+37.2000	0			
Maximized OBJ. function = 278148.4 Iters. = 30					

Summarized Results for FATMA TURAN			Page : 4		
Variable No. Names	Surplus	Dual Price	Variable No. Names	Surplus	Dual Price
85 S29	+381208.47	0	91 S35	0	+1.9937545
86 S30	+590171.81	0	92 S36	+46419.430	0
87 S31	+250650.00	0	93 S37	+106666.15	0
88 S32	0	+.76171869			
89 S33	+317600.00	0			
90 S34	+300032.16	0			
Maximized OBJ. function = 278148.4 Iters. = 30					

Ek-2: Modelin amaç fonksiyonu ve sağ taraf sabitleri için duyarlılık analizi sonuçları

Sensitivity Analysis for Objective Coefficients Page : 1

Variable	Min. C(j)	Original	Max. C(j)	Variable	Min. C(j)	Original	Max. C(j)
X1	-Infinity	+5.00000	+12.1875	X15	-Infinity	+19.0000	+20.9092
X2	-Infinity	+6.00000	+12.4922	X16	+17.3328	+18.7500	+19.7895
X3	-Infinity	+21.2000	+23.2324	X17	-Infinity	+12.6000	+21.0234
X4	-Infinity	+19.2000	+24.2227	X18	+16.7292	+17.5000	+18.6621
X5	+15.2654	+17.3000	+22.0137	X19	-Infinity	+3.00000	+7.00781
X6	-Infinity	+16.0000	+21.0234	X20	-Infinity	+2.00000	+6.47461
X7	-Infinity	+2.00000	+7.61719	X21	-Infinity	+5.00000	+6.85547
X8	-Infinity	+18.3000	+21.6328	X22	-Infinity	+19.6000	+24.8701
X9	+12.7136	+16.4000	+19.3477	X23	+18.2772	+19.8000	+25.5176
X10	+10.6171	+14.7000	+22.6611	X24	-Infinity	+17.3000	+25.3271
X11	-Infinity	+14.6000	+21.9375	X25	-Infinity	+1.20000	+18.8906
X12	+15.9275	+17.0000	+18.9820	X26	-Infinity	+11.2000	+16.4531
X13	-Infinity	+14.4000	+20.4902	X27	-Infinity	+10.4000	+17.5195
X14	+18.4758	+19.5000	+Infinity	X28	-Infinity	+10.0000	+16.9863

Sensitivity Analysis for RHS Page : 1

Constrnt	Min. B(i)	Original	Max. B(i)	Constrnt	Min. B(i)	Original	Max. B(i)
1	0	+3000.00	+6187.08	20	0	+2100.00	+8099.22
2	0	+3500.00	+6609.35	21	0	+1500.00	+7165.93
3	0	+700.000	+2371.91	22	0	+165.000	+1726.82
4	0	+2750.00	+4353.56	23	-Infinity	+150.000	+1672.19
5	-Infinity	+221.500	+1985.98	24	0	+240.000	+1773.63
6	0	+2517.00	+4364.59	25	0	+2500.00	+4556.18
7	0	+10000.0	+15099.3	26	0	+120.000	+2480.80
8	0	+242.000	+2037.54	27	0	+100.000	+2317.10
9	-Infinity	+198.000	+2205.61	28	0	+100.000	+2386.70
10	-Infinity	+210.500	+1924.56	29	+252392	+633600	+Infinity
11	0	+175.000	+1945.60	30	+43428.2	+633600	+Infinity
12	-Infinity	+250.000	+2296.28	31	+382950	+633600	+Infinity
13	0	+110.000	+2005.66	32	+582607	+633600	+1229574
14	-Infinity	+260.000	+2251.93	33	+316000	+633600	+Infinity
15	0	+100.000	+1957.68	34	+333568	+633600	+Infinity
16	-Infinity	+175.000	+2137.79	35	+506669	+800000	+1874986
17	0	+435.000	+2282.59	36	+3580.57	+50000.0	+Infinity
18	-Infinity	+250.000	+2331.36	37	+13333.9	+120000	+Infinity
19	0	+1000.00	+6542.76				

KAYNAKÇA

Aksöz, İbrahim. **İşletmelerde Ekonomik Kararların Alınması ve İşletme Faaliyetlerinin Değerlendirilmesinde Kullanılan Doğrusal Programlama Yöntemi.** 1995.

Aslan, Demir. **Yöneylem Araştırması.** Erzurum: İşletme Fakültesi Araştırma Enstitüsü Ders Notları. No:41. 1978.

Atlas, Mahmut. **Minimum Maliyetli Uygun Beslenme Bileşimlerini Oluşturmada Sıfır-Bir Tamsayılı Doğrusal Programlama ve Anadolu Üniversitesi Yunusemre Kampüsü Öğrenci Kafeteryasında Bir Uygulama Denemesi.** Eskişehir: Yüksek Lisans Tezi. 1988

Atlas, Mahmut. **Out Off Kilter Algoritması ve Sıvılandırılmış Petrol Gazı Dağıtım İşletmesinde Bir Uygulama.** Eskişehir: Anadolu Üniversitesi. 1999.

Camm, Jeffrey D. and James R. Evans. **Management Science: Modeling, Analysis and Interpretation.** South-Western College Publishing. Cincinnati, Ohaio. 1996.

Cooper, L. and D.I. Steinberg. **Methods and Applications of Linear Programing.** W.B. Sounders, Company. 1974.

Doğan, İbrahim. **Yöneylem Araştırması Teknikleri ve İşletme Uygulamaları.** İstanbul: Marmara Üniversitesi İ.İ.B.F. 1995.

Esin, Alptekin. **Yöneylem Araştırmasında Yararlanılan Karar Yöntemleri.** Ankara: Gazi Üniversitesi Fen Edebiyat Fakültesi Yayınları. 1983.

Garfinkel R. S. and G. L. Nemhauser. **Integer Programing.** New York, Wiley, 1972.

Gaver, Donald P. and Gerald L. Thompson. **Programing and Probability Models in Operations Research**. Banks / Cole Pob. Co. Monterey, Californiya. 1973.

Gençyılmaz, G. **Yöneylem Araştırması I – II Ders Notları**. İstanbul Üniversitesi Üretim Yönetimi. 1991.

Gottfried, S.B. and J. Weisman. **Introduction to Optimization Theory**. Prentice-Hall. Inc. Englewood Cliffs. New Jersey. 1973.

Greenberg, H. **Integer Programing**. Academic Prees, New York, London. 1971.

Gürdoğan, Nazif. **Üretim Planlamasında Doğrusal Programlama ve Demir Çelik Endüstrisinde bir Uygulama.** Ankara: Ankara Üniversitesi Siyasal Bilgiler Fakültesi Yayınları. 1981.

Halaç, Osman. **Kantitatif Karar Verme Teknikleri**. 3. Baskı. İstanbul: Evrim Dağıtım. 1991.

Hillier, F.S. and G.J. Liberman. **Introduction to Operations Research**. 19. Ed., Holden-Day, San Fransisco, Californiya. 1974.

Kara, İmdat. **Tamsayılı ve Dinamik Programlamaya Giriş**. Eskişehir: Anadolu Üniversitesi. 1984.

________. **Doğrusal Programlama**. Eskişehir: Bilim Teknik Yayınevi. 1991.

________. **Yöneylem Araştırması**. Doğrusal Olmayan Modeller. Eskişehir: Anadolu Üniversitesi. 1986.

________. **Yöneylem Araştırmasının Yöntem Bilimi**. Eskişehir: Anadolu Üniversitesi. 1985.

Karayalçın, İlhami İ. **Yöneylem "Hareket" Araştırması – Operations Research-Kantitatif Planlama ve Karar Verme Yöntemleri**. İstanbul Teknik Üniversitesi İşletme Fakültesi. 1993.

Öztürk, Ahmet. **Yöneylem Araştırması**. Bursa: Ekin kitapevi yayınları. 1997.

________. **Leontief Modeli ve Doğrusal Programlama**. Bursa: Uludağ Üniversitesi İ.İ.B.F. 1986.

Paradimitriou, C.H. **Combinatorial Optimization: Algorithms and Complexity**. Prentice Hall. New Jersey. 1982.

Rao, F.S. **Optimization Theory and Application**. Willey Easten Limited. Ney Delhi. 1979.

Salkın, H. M. **Integer Programing**. Case Western Reserve Üniv. Cleveland, Ohio. 1975.

Sarıaslan, Halil. **Kaynak Dağılımında Doğrusal Programlama**. Ankara: Ankara Üniversitesi Siyasal Bilgiler Fakültesi Yayınları. 1986 .

Schrijver, Alexander. **Theory of Linear and Integer Programming**. Amsterdam, The Nedherlands. 1998.

Serper, Özer ve Necmi Gürsakal. **Doğrusal Programlama**. Bursa: Bitia İşletme Fakültesi Yayınları. 1982.

Taha, Hamdy A. **Operational Research**. An Introduction: Mac Millan. New York. 1989.

Taha, Hamdy A. **Yöneylem Araştırması**. İngilizceden çeviren: Ş.Alp Baray, Şakir Esnaf. İstanbul: İstanbul Üniversitesi İşletme Fakültesi. 2000.

Tulunay, Yılmaz. **Matematik Programlama ve İşletme Uygulamaları**. İstanbul Üniversitesi İşletme Fakültesi.1980.

Tütek, H.H. ve Ş. Gümüşoğlu. **Sayısal Yöntemler Yönetsel Yaklaşım**. İstanbul. Beta Basım. 1994.

Winston, W.L. **Operations Research Application and Algorithms**. PWS-Kent Pub., Boston. 1991.

Wolsey, Lorence A. **Integer Programming**. John Willey & Sons, Inc. New York. 1998.