İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

STOK KESME PROBLEMİ: ALÜMİNYUM SEKTÖRÜNDE UYGULAMASI

YÜKSEK LİSANS TEZİ

Semih ADAKCI

Anabilim Dalı : Endüstri Mühendisliği

Programı : Endüstri Mühendisliği

EYLÜL 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

STOK KESME PROBLEMİ: ALÜMİNYUM SEKTÖRÜNDE UYGULAMASI

YÜKSEK LİSANS TEZİ

Semih ADAKCI
(507061122)

Tezin Enstitüye Verildiği Tarih : 13 Eylül 2010
Tezin Savunulduğu Tarih : 15 Eylül 2010

Tez Danışmanı : Yrd.Doç.Dr. Cafer Erhan BOZDAĞ(İTÜ)
Diğer Jüri Üyeleri : Yrd.Doç.Dr. Alp ÜSTÜNDAĞ (İTÜ)
Yrd.Doç.Dr. Şule ÖNSEL (DÜ)

EYLÜL 2010

ÖNSÖZ

Günümüz piyasalarında artan rekabet koşulları ile birlikte ürün ve hizmet fiyatları müşteriler tarafından belirlenir hale gelmektedir. Fiyatların belirli olmasıyla kar marjları önemli ölçüde azalarak, işletmelerin her alandaki maliyetlerini azaltması gerekliliği kaçınılmaz olmaktadır. Buna paralel olarak, özellikle alüminyum, kağıt, ağaç, plastik, çelik vb. sektörlerde büyük miktarlarda üretim yapan işletmelerde, nihai ürünlerin üretilmesi belirli aşamadaki yarı mamul stok malzemelerinin dilme, kesme vb. işlemlerinden geçmesiyle üretilir. Bu tür işletmelerde üretim, enerji, hammadde ve fire maliyetlerinin toplam maliyet içerisindeki payı çok fazla olmaktadır. Bundan dolayı, bu tür sektörlerde faaliyet gösteren işletmelerin malzeme kesim kayıplarını minimize etmesi en temel odak noktasını oluşturmaktadır.

Bu çalışmada, çeşitli boyutlardaki stok malzemelerinden üretilmesi istenen belirli ebat ve miktarlardaki siparişlerin minimum malzeme kaybı ve maliyette matematiksel modellerden faydalanarak nasıl karşılanabileceğinin gösterilmesi amaçlanmaktadır.

Yüksek lisans tez çalışmam süresince benden desteğini esirgemeyen danışmanım Yrd. Doç. Dr. Sn. Cafer Erhan BOZDAĞ'a, destekleri ile bana güç veren sevgili aileme ve değerli çalışma arkadaşlarıma ve yüksek lisans süresince eğtimime destek veren TUBİTAK' ye sonsuz teşekkür ederim.

Eylül 2010

Semih Adakcı

Endüstri Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

ARUAY	**:** Açgözlü Rassallaştırılmış Uyarlamalı Arama Yordamı
BT	**:** Benzetilmiş Tavlama
DP	**:** Doğrusal Programlama
GSP	**:** Gezgin Satıcı Problemi
KKA	**:** Karınca Koloni Algoritması
LPP	**:** Lineer Programlama Prosedürü
SAL	**:** Sınırlandırılmış Aday Listesi
SÇP	**:** Sırt Çantası Problemi
SKP	**:** Stok Kesme Problemi
TDP	**:** Tamsayılı Doğrusal Programlama

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

STOK KESME PROBLEMİ: ALÜMİNYUM SEKTÖRÜNDE UYGULAMASI

ÖZET

Sürekli olarak değişen müşteri gereksinimlerine ve müşteri odaklı fiyatlara esnek, hızlı ve sürdürebilir kar marjları ile cevap verebilmek için işletmelerin maliyetlerini ve tedarik sürelerini azaltmaları gerekmektedir. Her işletmenin faaliyet alanına göre maliyetlerini ve tedarik sürelerini azaltmaya yönelik çalışmaları değişkenlik arz edebilmektedir.

Alüminyum, çelik, demir, kağıt, ağaç, cam, plastik vb. sektörlerde üretim yapan işletmeler müşterilerine hızlı cevap verebilmek, tedarik süresini azaltabilmek için nihai ürünlerinden daha büyük boyutlardaki yarı mamul stok malzemelerini üretmektedirler ve gelen müşteri siparişinin boyutlarına göre dilme, kesme vb. işlemler ile siparişler karşılanmaktadır. Stok malzemelerinden nihai müşteri siparişlerin karşılanması sırasında stok malzemesinin her zaman tam verimle kullanılmamasından dolayı hammadde, enerji, tezgah maliyetleri yanısıra kesme kaybı(fire) maliyetleri de oluşmaktadır. Tek bir adet stok malzemesi için kesme kaybının maliyeti çok önemli olmamasına karşın yıllık kesme kayıpları işletmeler için dikkate değer maliyetler oluşturabilmektedir.

Temelde büyük boyutlardaki stok malzemelerinden küçük ebatlardaki müşteri siparişlerin istenilen miktarlarda karşılanması sırasında malzeme kaybını minimize etmeye çalışan probleme stok kesme problemi denmektedir.

Bu çalışmada, stok kesme probleminin işletme maliyetleri açısından önemine bağlı olarak, çeşitli stok kesme problemlerini matematiksel modeller ile nasıl çözülebilirliliği anlatılmaktadır ve alüminyum sektöründe bir uygulamaya yer verilmiştir.

Giriş bölümünde çalışmanın kapsamından genel olarak bahsedilmekte ve ikinci bölümde stok kesme problemi tanımlamaları ve sınıflandırılmaları ele alınmıştır..

Üçüncü bölümde; stok kesme problemlerinin çözümünde kullanılan metodlar başlığı altında literatürde kullanılan matematiksel modeller açıklanmakta, ne tür stok kesme problemlerinde hangi çözüm metodlarının kullanıldığından bahsedilmektedir.

Dördüncü bölümde; alüminyum sektöründe faaliyet gösteren ve çeşitli stok malzemelerinden çeşitli ebat ve miktarlarda müşteri siparişi üretimi yapan firma ile ilgili bir boyutlu stok kesme problemi uygulaması yer almaktadır.

Son bölümde ise; ilk üç bölümdeki teorik kısım ve dördüncü bölümdeki bir boyutlu stok kesme uygulaması değerlendirilmektedir.

CUTTING STOCK PROBLEM: AN APPLICATION IN ALUMINIUM INDUSTRY

SUMMARY

In order to give quick response and provide sustainable profit margin against continuously changing customer requirements and customer oriented price, enterprices should focus on decreasing costs and lead times. However, types of managing costs and lead times can be changed according to business sector.

Enterprices which are established in aluminium, steel, iron, paper, wood, glass, plastic etc. sectors cut and slim large sized semi or end-stock materials into the order specifications. However, in this final process, inefficient utilization of the large-sized end-stock materials can cause trim losses. These losses imply high costs in addition to raw material, energy, and machine costs in the system. Although trim loss of just a stock material can not be taken attention, total trim losses costs in a month or in a year might be so meaningful for these kind of enterprices.

Cutting stock problem is basically try to minimize trim loss while producing the requirement customer orders by cutting, slitting stock materials.

In this study, using mathematical methods for several cutting stock problems solutions are described by taking into importance of enterprice costs and an one-dimensional stock cutting problem application is placed.

In the introduction part, the scope of the study is briefly explained and in the second section the titles about cutting stock problems and its classifications are listed.

In the third section; under solution methods for cuttting stock problem headline, what kinds of methods exist in literature about cutting stock problem and which types of cutting stock problems can be solved by using which types of solution methods.

In the fourth section; an one-dimensional cutting stock problem case study of an aluminum company that is established in Turkey produce many various width and quantity required customers orders from semi or end stock materials will be explained in detail.

In the last section; the theoretical assessment in three sections and the one dimensional cutting stock application in fourth section are discussed.

1. GİRİŞ

Herkesin her şeyi üretebildiği günümüz koşullarında üretmek bir üstünlük sağlamazken, maliyet avantajları sağlayarak kaliteli bir üretim yapmak ve hizmet sunmak pazarda ayakta kalabilmenin bir koşulu haline gelmiştir. Bu yeni ekonomik yapı içinde şirketlerin mâliyetlerini azaltılması her zaman aklından çıkartamayacağı bir unsur olarak önem kazanmıştır. Her işletmenin faaliyet gösterdiği alana göre maliyet avantajı sağlayacağı alanlar değişkenlik gösterebilmektedir. Örneğin hizmet sektöründe iş gücü önemli maliyet kalemi olabilirken, üretim sektöründe hammadde, enerji ve dolayısıyla hurda maliyeti çok önemli olabilmektedir. Özellikle alüminyum, kağıt, ağaç, plastik, çelik vb. sektörlerde faaliyet gösteren işletmelerde hammaddeler(stok malzemeleri) belirli üretim aşamalarından geçtikten sonra dilme, kesme vb. işlemler ile nihai müşteri siparişi ebatlarına getirilmektedir. Bu işlemler sırasındaki her türlü hurda ve kesim kaybı(fire) ürün maliyetinin artmasına sebebiyet vermektedir.

Hurda miktarı, makine, işçi ve diğer üretim faktörlerinden kaynaklı kullanılamayan malzeme miktarını kapsarken, fire miktarı daha önceden bilinen ve stok malzemelerinin tam olarak verimli kullanılmamasıdan kaynaklanan miktarı kapsamaktadır. Hurda miktarı, makine, işçi vb. faktörlerden kaynaklanan hataların toplam kalite yönetimi ana başlığı altında yer alan 5S, kaizen, toplam verimli bakım, altı sigma vb. kalite iyileştirme çalışmaları ile azaltılabilirken, fire miktarı matematiksel modellerden faydalanarak oluşturulan çözüm yöntemleri ile minimize edilebilmektedir.

Fire miktarlarının azaltılmaya çalışıldığı problemler; faaliyet gösterilen sektörel kısıtlar, üretim kısıtlarına, müşteri siparişlerine, stok malzemelerinin özelliklerine göre çeşitlilik gösterebilmektedir ve dolayısıyla bu tür çeşitli problemlere karşı çeşitli çözümler geliştirilmesi gerekmektedir.

Bu çalışmada stok kesme probleminin tanımı, sınıflandırılması ve çözüm metodları ele alınarak alüminyum sektöründe sezgisel yöntem ile tam sayılı programlama yönteminin karşılaştırılmalı bir uygulamasına yer verilecektir.

2. STOK KESME PROBLEMİ TANIMI VE SINIFLANDIRILMASI

2.1 Stok Kesme Problemi

2.1.1 Stok kesme probleminin tanımı

Küçük ebatlardaki parçaların büyük boyutlardaki malzemelerden en uygun ne şekilde karşılanması gerektiğini konu edilen problem, stok kesme problemi olarak adlandırılır.

Küçük parçaların büyük objelerden kesilmesi sırasında ortaya iki problem çıkmaktadır. Birinci problem; büyük objeler için uygun boyutların seçimi konusunu içeren ayırma problemidir. İkinci problem ise, hurdayı minimize edecek şekilde belirli olan büyük objelerden küçük parçaların nasıl kesilmesi gerektiğini içeren kesme kaybı(fire) problemidir(Hinzman, 1979). Pratikte, küçük parçalar sipariş edilen ürünler, büyük objeler stok malzemeleridir. Ayırma problemi ve kesim kaybı probleminin kombinasyonu stok kesme problemi olarak bilinmektedir

Stok malzemelerinden nihai müşteri ürünleri ruloların kesilmesi sırasında herhangi bir matematiksel düşünce önerilmeden kesim yapılması mümkündür, fakat rassal olarak yapılacak bu tür kesimlerde fire verilme ihtimali kaçınılmaz olacaktır.

Stok kesme problemine, geniş stok malzemelerinden daha küçük parçaların elde edilmesini gerektiren bir çok üretim endüstrisinde karşılaşılması muhtemeldir. Genellikle stok malzemeleri kağıt, et, çelik, cam, ağaç, demir, alüminyum, bakır, plastik ve tekstilden oluşmaktadır (Dyckhoff,1990). Ayrıca ambalaj sektöründeki paketleme işlemi sırasında da benzer problem ile karşılaşılmaktadır. Kesme işlemini, stok malzemesi hacmi içerisinde küçük parçaların paketlenmesi olarak; paketleme işlemini de küçük parçalar için stok malzemelerin kesilmesi olarak ele aldığımızda bu benzerlik ortaya açıkça konulabilecektir. Bu yüzden literatürde genellikle kesme ve paketleme problemi(cutting and packing problem) olarak ele alınmaktadır.

Diğer bir tanımda şu şekilde yapılabilir; cam, kağıt, tekstil, alüminyum, çelik, demir ve mobilya benzeri endüstri alanlarında, boyutları bilinen bir malzemeden, çeşitli biçim, miktar ve boyutlara sahip daha küçük parçaların kesilerek kullanılması gerekmektedir. Bu tür problemler, genel olarak, stok kesme problemleri olarak adlandırılmaktadır. Kesme problemlerini karakterize eden temel özellikler; ana temel özellikler; ana malzeme boyutları, istenen parça çesidi, her bir parçanın boyutu, talep miktarı ve kesme teknigi olarak sıralanabilir.

Stok kesme problemleri, küçük bir çalışma uzayı içinde olması durumu hariç, bu problemlere en iyi çözümün üretilmesinini imkansız olduğu NP-tam problemler olarak bilinir. Arama uzayının büyüklüğü nedeniyle kesme problemlerinin çözümü için yönlendirilmemiş, bir arama yapmak oldukça verimsiz olduğundan, probleme ait en iyi çözümün bulunabilmesi için büyük arama uzayı içinde düzenli bir arama yapılması gerekir (Callaghan ve diğ., 1999). Bu sebeple araştırmalar, en iyi çözüme yakın iyi çözümleri verimli bir şekilde bulan olasılıksal yaklaşım teknikleri üzerinde yoğunlaşmaktadır.

NP, belirsiz Turing Makinesi ile çokterimli (polinomsal) zamanda çözülebilen karar problemlerini içeren karmaşıklık sınıfıdır. Bu sınıftaki problemler belirli Turing Makinesi ile çokterimli zamanda doğrulanabilirler ve bu şekilde doğrulanabilen her problem NP sınıfındadır. Bu nedenle NP, (belirli Turing Makinesi ile) çokterimli zamanda doğrulanabilen problemlerin sınıfı olarak da tanımlanabilir. Belirli Turing makinesi aynı zamanda belirsiz Turing makinesi olduğundan, P sınıfındaki bütün problemler aynı zamanda NP'dedir.

NP-Zor; en az herbir NP problem kadar zor olan problemlerin bulunduğu sınıfa NP-Zor (NP-hard) denir. Daha resmi bir şekilde;

NP-Zor $=\{H \mid \forall L \in NP, L \leq_p H\}$ burada , $L \leq_p \mathrm{H}$, L probleminin, H problemine çokterimli zamanda indirgenebildiği anlamına gelir. Bir başka deyişle, NP-Zor sınıfındaki her hangi bir problem çok terimli zamanda çözülebilirse, NP sınıfındaki bütün problemler çok terimli zamanda çözülebilir.NP-Tam , hem NP olup hem NP-Zor olan problemlerin sınıfıdır. Dolayısıyla bu sınıftaki problemler NP sınıfının en zor problemleridir. Yukarıdaki tanımdan yola çıkarak, herhangi biri çokterimli zamanda çözülebilirse, bütün hepsi çok terimli zamanda çözülebilir.

Kesme problemlerinde eniyilemenin amacı, yerleştirmenin yapılacağı stok malzemesinin kullanılabilirliliğini artrmak ve böylelikle kullanılmayan alanı başka bir deyişle fire miktarı en az olan yerleşim planını bulmaktadır (Hopper ve diğ, 2001).

Stok kesme problemi ile ilgili ilk formulü 1939 yılında Kantorovisch tarafından geliştirilmesine rağmen ancak 1960 yılında yayınlanabilmiştir (Kantorovisch, 1960). Stok Kesme Problemi ile ilgili bilimsel araştırmalar yaklaşık olarak elli yıl öncesinde başlamış olup sürekli artan sayıda çeşitli formülasyon ve çözüm metodları literature kazandırılmıştır. Sonraki bölümlerde SKP' nin önemli özelliklerinden bahsedilecektir.

Yerleşim planı(pattern), küçük parçaların stok malzemeleri içindeki yerleşimini gösterir ve küçük parçaların yerleşim sırasını temsil eden permütasyonlar ile temsil edilir. Yerleşim alanında kullanılmayan alan veya yerleşim sırasında olup da yerleşim planı içinde yer almayan parçalar fire olarak isimlendirilir.

Stok kesme problemi Dyckhoff tarafından gelştirilen sınıflandırma şeması ile gruplandırılabilinir (Dyckhoff, 1990). Kesme ve paketleme problemlerinindeki yakın ilişkiden dolayı belirtilecek şema iki türdeki problemlerin gruplandırılmasında kullanılabilir.

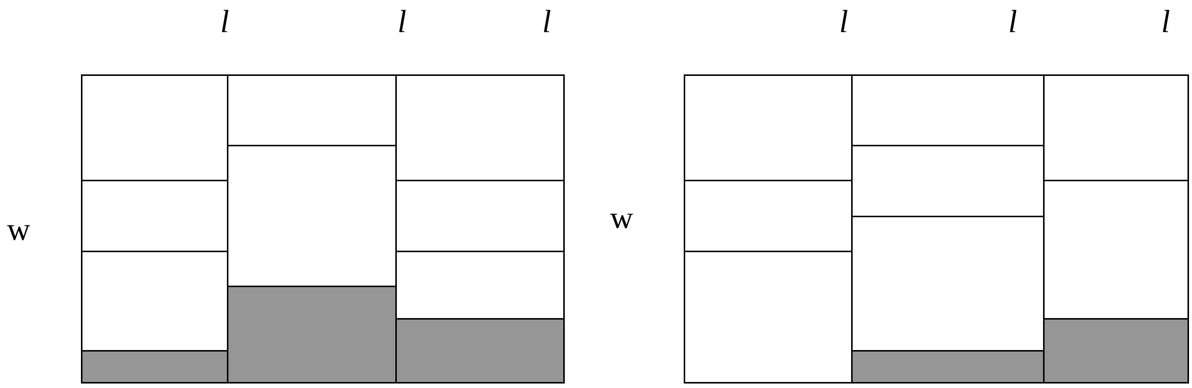
2.1.2 Stok kesme probleminin sınıflandırılması

Stok kesme problemi Dyckhoff tarafından gelştirilen sınıflandırma şeması ile gruplandırılabilinir.(Dyckhoff, 1990). Kesme ve paketleme problemlerinindeki yakın ilişkiden dolayı belirtilecek sınıfladırılma şeması iki türdeki problemlerin gruplandırılmasında kullanılabilir.

Problemlerin sınıflandırması dört temel özellik altında yapılabilir. Aşağıda belirtilecek her bir özellik ileriki paragraflarda detaylı olarak alt maddeleri ile açıklanacaktır.

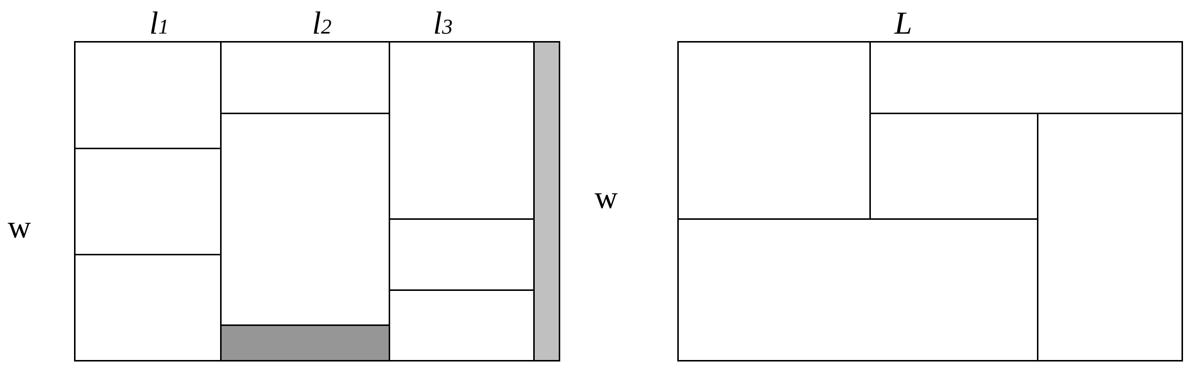
- Boyutluluk
    - Boyut sayısı (N)
- Atama biçimi
    - Tüm stok malzemeler ve küçük parçaların seçimi (B)
    - Sstok malzemelerin seçimi ve tüm küçük parçalar (V)

- Stok mazlemelerin seçimi
    - Bir adet stok malzemesi (O)
    - Birden fazla aynı ebatta stok malzemesi (I)
    - Farklı stok salzemesi (D)
- Küçük Parçaların Ayrımı
    - Farklı boyutlardaki az sayıda küçük parçalar (F)
    - Farklı boyutlardaki çok sayıda küçük parçalar (M)
    - Birbirine yakın benzer boyutlardaki çok sayıdaki küçük parçalar (R)
    - Benzer çok sayıdaki küçük parçalar (C)



Şekil 2.1 : Üst 1 boyutlu ve 1.5 boyutlu kesim biçimleri (Dyckhoff, 1990).

Şekil 2.1 ve şekil 2.2' deki gri renkli kısımlar fireyi göstermekle birlikte şekil 2.1' de soldaki kesme biçimi bir boyutlu kesme problemine örnek teşkil etmektedir ve stok malzemenin eni ve boyu aynıdır. Şekil 2.1' deki sağdaki kesme biçimi ise 1.5 boyutlu kesme problemine örnek olup stok malzemelerinin boyları sabit iken enleri değişkenlik gösterebilmektedir.

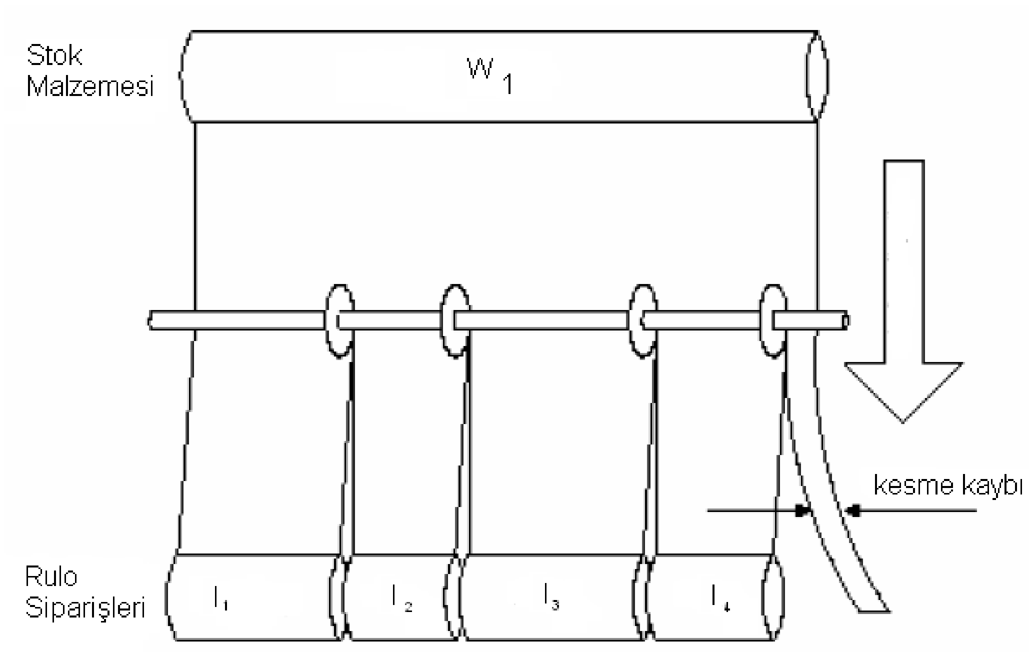


Şekil 2.2 : 2 boyutlu ve giyotinsiz kesim biçimleri (Dyckhoff, 1990).

Dyckhoff' un yaptığı bu sınıflandırmada 1.5 boyutlu stok kesme problemi olarak tanımlama olmasına rağmen literatür incelendiğinde özellikle son zamanlardaki bu tür problemlerin bir boyutlu çeşitli endeki stok malzemeli stok kesme problemi olarak tanımlandığı görülmüştür. Bu çalışmadaki uygulamanın temel konusu olan bu stok kesme problemi bir boyutlu çeşitli endeki stok kesme problemi olarak ele alınacaktır.

Şekil 2.2' de soldaki kesme biçimi iki boyutlu stok kesme problemini figure etmektedir. Stok malzemesi üzerinde sipariş iki boyutta istenildiği şekilde konumlandırılabilir. Şekil 2.2' de sağdaki kesim biçimi giyotinsiz kesim biçimine örnektir.

Kesme problemlerinde kesme işlemi stok malzemesi üzerinde bir uçtan diğer uca kadar yapıldığında giyotinli kesme olarak, bir uçtan diğer uca yapılamadığı takdirde ise giyotinsiz kesme olarak adlandırılır (Leung ve diğ, 2001).



Şekil 2.3 : 1 boyutlu giyotinli kesim örneği.

Şekil 2.3' te çeşitli W_1 eninde stok malzemesinden l_1, l_2, l_3, l_4, enlerindeki siparişlerin giyotinli olarak kesilmesi ve kesme kaybı da belirtilmektedir.

Atama biçimi

Stok kesme probleminin karmaşıklığını etkileyen diğer önemli karakteristiği atama biçimidir. İki tip arama biçimi kabul edilmiştir. Birincisi, tüm stok malzemeleri tüketilecek ve buna karşılık tüm siparişlerin karşılanması zorunlu değildr. İkincisinde ise tüm siparişleri karşılanırken stok malzemelerinin belirli miktarı kullanılacaktır.

Stok malzemelerin seçimi

Stok kesme probleminin diğer önemli özelliği stok malzemelerinin seçimidir ve üç alt kısma ayrılır. Tek stok malzemesinin olma durumu, aynı ebatlarda birden fazla stok malzemesinin olma durumu, farklı boyutlarda ve miktarlarda stok malzemesinin olma durumudur.

Siparişlerin seçimi

Stok kesme probleminin sınıflandırılması kullanılan son özellik müşteri siparişlerin seçimidir. Farklı boyutlardaki az sayıda küçük parçalar, farklı boyutlardaki çok sayıda küçük parçalar , birbirine yakın benzer boyutlardaki çok sayıdaki küçük parçalar , benzer çok sayıdaki küçük parçalar.

3. STOK KESME PROBLEMİ ÇÖZÜM METOTLAROI

3.1 Stok Kesme Problemi Metotları ve Kıyaslanması

Stok kesme problemine ait çözümü metodları üç ana grupta toplanabilir. İlk grupta algoritmik metotlar, problemin çözümü için optimal çözümü garanti ederken dezavantajı, özellikle büyük problemlerdeki hesaplama süresi uzunluğu ve elde edilen çözümdeki sayısal karmaşılıktır. Bundan dolayı, son yıllarda stok kesme problemlerinin çözümü için algoritmik yaklaşımlar nadir görünmektedir. İkinci grupta değerlendirilen sezgisel metotlar seyrek olarak optimal çözümü bulabilirken, genellikle kısa süre zarfında hızlı şekilde kabul edilebilir çözümler sunabilmektedir. Sezgisel bir metodun kabul edilmesi için bilinen optimal çözüme olabildiğince yakın olması gerekmektedir. Sezgisel metotlar ayrıca geliştirilmek üzere yaratılmış olan kısmı problemlere ait bilgileri kullanırlar. Bu yüzden, benzer problemlerin çözümü için aynı metotun kullanılabilirliliği çok sık görünememektedir. (Hinxman, 1979). Metasezgisel metotların, klasik sezgisel metotların aksine lokal uygun çözümlere takılmayan bir özelliği bulunmaktadır. Metasezgisel metotlar, çözüm prosesinde alt seviyedeki sezgisel metotlar tarafından desteklenirler.

Stok kesme problemlerine ait çözüm metotları incelendiğinde özellikle sezgisel ve metasezgisel metotlara ait çözümlerinin probleme göre özelleşebildiği ve ilgili probleme ait çözümün başka bir probleme uyarlanabilirliliğinin pek mümkün olmadığı sadece önemli bir yol gösterici olduğu tespit edilmiştir.

Stok kesme probleminin çözüm metotları, üç başlık altında bu bölümde detaylı olara incelenecektir. İnceleme sırasında özellikle sezgisel metotlar üzerinde yoğunlaşılarak örnek uygulamalar yapılacaktır ve önceliklele sezgisel metotların çözümünde de kullanılan algoritmik yöntemler öncelikle ele alınacaktır.

3.2 Algoritmik Metotlar

Birçok işletme sorunlarının çözümünde kullanılan algoritmik çözümler stok kesme problemlerinin çözümü sırasında da yararlanılan teknikler arasında yer almaktadır.

Kesin optimal sonuçları üretmeyi hedefleyen ve bunu yüksek oranda başaran algoritmik metotlar, stok kesme probleminin büyüklüğüne, detayına bağlı olarak değişmekle birlikte uzun süren uygulama zamanı ihtiyacı gösterebilmektedir. Saatlerce ve günlerce süren uygulama zamanı ihtiyacı bu yöntemin temel eksikliği olmakla birlikte bu tür çalışmalara çoğu kez rastlanabilmektedir

3.2.1 Doğrusal programlama

Doğrusal programlama bir optimizasyon tekniğidir. Yöneylem araştırmasının klasik optimizasyon modellerinden biri olup, üretim sistemlerinin planlanmasında yaygın biçimde kullanılır (Yamak, 1994).

Doğrusal programlama, değişkenlere ve kısıtlayıcılara bağlı kalarak amaç fonksiyonunu en uygun (maksimum veya minimum) kılmaya çalışır. Buna göre DP, değişkenlere ve kısıtlayıcılara bağlı kalarak amaca en iyi ulaşma tekniğidir. Temelde doğrusal programlama, verilen optimallik ölçütüne bağlı kalarak kıt kaynakların optimal dağıtımını içeren matematiksel bir tekniktir (Öztürk, 1997).

Bir başka tanıma göre doğrusal programlama, doğrusal eşitlik veya eşitsizlikler halinde bulunan kısıtlayıcı şartlar altında, belli bir amaç fonksiyonunun en iyilenmesidir (Esin, 1994).

Bir modelin doğrusal programlama tekniği ile çözülebilmesi için, modelde aşağıdaki şartların bulunması gerekir (Şenel, 1974).

a. Modelin unsurları rakamla ifade edilebilmelidir. Bu özellik matematik modellerin en önemli şartıdır. Doğrusal programlama, kalitatif unsurları içine alan modellerin çözümünde kullanılamaz.
b. Değişkenler arasında alternatif seçim mümkün olmalıdır. Objektif fonksiyondaki şartı gerçekleştirebilmek için, üretim faktörleri ve üretim teknikleri arasında bir seçim yapılabilmelidir. Mesela, yalnızca bir makineye veya insan emeğine ihtiyaç gösteren üretim tekniklerinde, seçim yapılması mümkün olmadığından, doğrusal programlama uygulanamaz.
c. Değişkenler arasında kurulan bağlantıların, lineer olması gerekir. Lineerlik denince, modelde bulunan bütün eşitlik ve eşitsizliklerin içindeki değişkenlerin, birinci dereceden olması ve bu ifadelerin grafiklerinin bir düzeyi göstermesi anlaşılır. Bu özellik doğrusal programlamaya uygulanırsa,

her değişkenin başındaki katsayının sabit ve değişkenin birinci dereceden olması gerektiği sonucuna varılır.

d. Doğrusal programlamanın uygulanacağı işletme problemi kısa devreli olmalıdır. Doğrusal programlamanın en önemli şartı olan doğrusallık, ancak kısa devrede gerçekleşebilir. Mesela, kar maksimizasyonu problemlerinde, fiyatlar ancak kısa devrede sabit olabilir. Eğer uzun bir devre ele alınıp, doğrusal programlama tekniği uygulanırsa, doğrusallık şartı gerçekleşmeyeceğinden çıkan sonuç yanlış olur. Çünkü, uzun devrede fiyatlar çeşitli etkenlerle değişebilir. Ayrıca, işletmeler uzun devrede makinelerini yenileyebilirler veya başka bir üretim tekniği ile üretim yapabilirler. Uzun devrede, azalan masraflar kanunu gereğince, belirli bir süre masraflar azalır ve bir noktadan sonra masraflar yükselmeye başlar. Bu gibi hallerde, değişkenler arasındaki ilgi doğrusal değildir. Demek ki, uzun devreyi kapsayan modellerin çözümünde doğrusal olmayan programlama teknikleri kullanılmalıdır.

3.2.1.1 Doğrusal programlama çözüm yöntemleri

Doğrusal programlama problemleri grafik ve simpleks yöntemlerden biri ile çözümlenebilir.

Grafik çözümde, aşağıdaki adımların sırasıyla izlenmesi gerekmektedir:

- Değişkenlerin koordinat sisteminin yatay ve dikey eksenlerine yerleştirilmesi,
- Kısıtlayıcı fonksiyonların grafiğinin çizilmesi,
- Uygun çözüm bölgesinin belirlenmesi,
- En iyi çözümün araştırılması.

Bu çözüm yöntemi bir örnek ile incelenecektir.

Amaç fonksiyonu: $\max z = 6x_1 + 8x_2$ **(3.1)**

Kısıtlayıcı fonksiyonları:

$7x_1 + 3x_2 \leq 21$ **(3.2)**

$6x_1 + 7x_2 \leq 42$ **(3.3)**

$x_1 \leq 3$ **(3.4)**

$x_2 \leq 4$ **(3.5)**

Negatif olmama koşulu:

$x_1, x_2 \geq 0$

x_1 değişkenini yatay, x_2 değişkenini dikey eksen üzerinde olsun. Negatif olmama ($x_1 \geq 0$, $x_2 \geq 0$) koşulundan dolayı uygun çözümler x_1x_2 düzleminin birinci bölgesinde bulunacaktır. Kısıtlayıcı fonksiyonların oluşturduğu sınır, bu bölgeyi ($x_1 \geq 0$, $x_2 \geq 0$) iki kısma ayırır. Bölgelerden biri negatitif olmama koşulu dahil tüm kısıtlayıcıları sağlarken, diğeri yalnızca negatif olmama koşulunu sağlayan noktalardan oluşur.

Çözüm bölgesinin belirlenmesi için kısıtlayıcı fonksiyonları sırasıyla ele alınır ve kendilerine karşılık gelen doğruların x ve y eksenlerini kestikleri noktaların koordinatlarını belirlenir.

Koordinat belirleme ilgili tüm işlemler aşağıda verilmiştir.

Denklem 3.2' deki $7x_1 + 3x_2 = 21$ eşitliğinde,

$x_1 = 0$ için $x_2 = 7$, $x_2 = 0$ için $x_1 = 3$ olur.

Denklem 3.3' deki $6x_1 + 7x_2 = 42$ eşitliğinde,

$x_1 = 0$ için $x_2 = 6$, $x_2 = 0$ için $x_1 = 8$ olur.

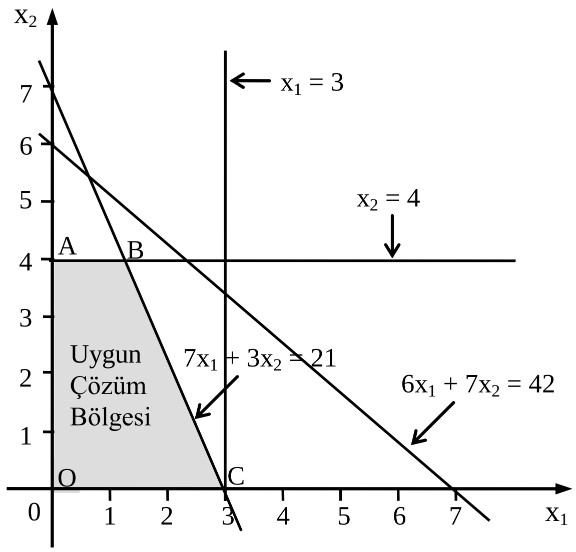
Denklem 3.4' deki $x_1 = 3$ eşitliği, yatay ekseni (3, 0) noktasında kesen ve dikey eksene paralel olan bir doğru tanımlar.

Denklem 3.5' deki $x_2 = 4$ eşitliği, dikey ekseni (0, 4) noktasında kesen ve yatay eksene paralel doğru denklemidir.

Bu belirlemelerden sonra kısıtlayıcı fonksiyonlarla ilgili doğrular çizebilinir. Sayıları dört olan kısıtlayıcı fonksiyonların her biri için bir doğru çizilmesi ve eşitsizliklerin yönlerinin dikkate alınmasıyla uygun çözüm bölgesi Şekil 3.1'deki taralı alan olarak belirir.

Şekil 3.1'deki taralı alanın içindeki (koyu renk çizilmiş sınırları dahil) tüm noktalar kısıtlayıcıları aynı anda sağladığından, OABC dörtgeni uygun çözüm bölgesidir. Bu alan içindeki sınırsız sayıdaki noktaların her biri uygun çözüm olarak nitelendirilir.

Şekil 3.1' den görüldüğü gibi 6x1 + 7x2 = 42 kısıtı olsa da olmasa da uygun çözüm bölgesi OABC alanı olacaktır.



Şekil 3.1 : Doğrusal prograalama grafik çözüm yöntemi örnek grafik

Çözüm bölgesini etkilemeksizin modelden çıkartılabilen bu tür kısıtlayıcılara gereksiz (fazlalık) kısıtlayıcılar denir. x1 ≤ 3 kısıtının da gereksiz olduğu görülebilir.

Taralı alanın içinde ve sınırları üzerindeki tüm noktalar bütün kısıtlayıcı fonksiyonları (negatif olmama koşulu dahil) sağladığından uygun çözüm bölgesi bir konveks (dış bükey) alan olarak ortaya çıkar. Geometrik olarak konveks alan kenarlarında çukurlaşmalar olmayan ve içinde delikler bulunmayan bir alandır. Bu alanın A, B gibi herhangi iki noktası göz önüne alındığında AB doğru parçasının tamamı alan içinde kalır.

Yukarıdaki maksimizasyon örneğine benzer olarak minimizasyon problemleri için de bu çözüm metodu uygulanabilir. Grafik yöntemi en fazla üç değişkenli problemlerin çözümünde elverişlidir. Uygulamada ise problemin değişkenleri çok daha fazla ve dolayısı ile gerçek doğrusal programlama problemlerinin çözümü ise simpleks yöntemi ile sağlanır.

Yöntem cebirsel tekrarlama (iterasyon) işlemine dayanır. Yöntemde önce başlangıç simpleks tablosu düzenlenir sonra tekrarlayıcı işlemler ile belirli bir hesap yöntemi içinde gelişen çözümlere doğru ilerleyerek optimal çözüme ulaşıncaya kadar işlemler sürdürülür. Gelişen çözüm tablolarında amaç fonksiyonunun ve karar değişkenlerinin değişen değerleri gözlenebilir. Simpleks yönteme başlamadan önce problemlerin doğru biçimde ifade edilmesi gerekir.

Genel doğrusal programalama modelini çözerken temel çözümlerin kullanabilmesi için önce problemin standart hale getirilmiş olması gerekir. Doğrusal programlama problemlerinin standart şekilde olması için aşağıdaki özellikleri taşıması gerekmektedir:

- Bütün kısıtlamalar eşitlik olmalıdır.
- Bütün değişkenler pozitif olmalıdır
- Amaç fonksiyonu maksimizasyon ya da minimizasyon olabilir.

Problemi standart forma getirme yöntemleri kısaca aşağıdaki şekilde açıklanabilir.

Kısıtlamalar,

Herhangi bir "$\geq$ " eşitsizlik "$\leq$ " eşitsizliğine dönüştürülebilir. Bunun için tek yapılması gereken eşitliğin iki yanının da "-1" ile çarpılmasıdır.

Örnek: $2x_1 + 3x_2 - 5x_3 \geq 15 \Leftrightarrow -2x_1 - 3x_2 + 5x_3 \leq 15$

Standart formda kısıtlamaların eşitlik olması gerekmektedir. Bunu yapabilmek için eşitsizliğin türüne göre yeni değişkenler kullanılması gerekmektedir.. "$\leq$ " Eşitsizliğini, sol tarafa bir atıl değişkeni ekleyerek eşitlik haline getirilir. Negatif olmayan atıl değişken kısıtlamada kullanılmayan kapasiteyi gösterir. "$\geq$ " Eşitsizliğini ise bu sefer eşitsizliğin sağ tarafına bir yetersiz kapasite değişkeni ekleyerek yapalır. Kısıtlamaların sağ taraflarında sadece değerlerin olması istendiğinden bu yeni değişkeni eşitsizliğin sol tarafından çıkartılır.

Örnek: $x_1 + 2x_2 \leq 6 \Leftrightarrow x_1 + 2x_2 + s_1 = 6 \text{ , } s_1 \geq 0$

$3x_1 + 2x_2 -3x_3 \geq 15 \Leftrightarrow 3x_1 + 2x_2 -3x_3 - s_2 = 15 \text{ , } s_2 \geq 0$

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

Örnek: $3x_1 + 2x_2 -3x_3 - s_2 = -15 \Leftrightarrow -3x_1 - 2x_2 +3x_3 + s_2 = 15$

Değişkenler;

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

Örnek: x_1 değişkeni sınırsız ise, $x_1 = x_{11} - x_{12}$ şeklinde bir tanımlama yaparak iki yeni değişkeni sisteme eklenir.

Amaç fonksiyonu;.

Her ne kadar standart form her tür amaç fonksiyonu kullanmaya izin veriyorsa da bazı durumlarda maksimizasyon ve minimizasyon arasında dönüşüm yapmak gerekebilir. Bunu yapmak için amaç fonksiyonu "-1" ile çarpılır.

Örnek: Maksimum (Z) = $3x_1 + 2x_2 - 3x_3$ matematiksel olarak

Minimum (-Z) = $-3x_1 - 2x_2 + 3x_3$'e eşittir.

Bir eşitlik sisteminde m eşitlik ve n değişken varsa ve değişken sayısı eşitlik sayısından büyükse tam çözüm bulmak matematiksel olarak mümkün değildir. Bu durumda (n – m) tane değişkene herhangi bir değer atanır ve geri kalan n değişkenin değeri bulunur. Simpleks metodunda da bu durumla karşılaşabilinir. Böyle bir durumda n tane değişken temel değişkenler olarak belirlenir ve geri kalan (n – m) tane değişken temel olmayan değişken olur, sıfır değerini alırlar. Bir temel çözümün uygun olabilmesi için bütün temel değişkenlerin değerinin sıfırdan büyük olması gereklidir.

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

Daha önce belirtilen kısıtlamaların standart hale getirilmesine, simpleks metodunun ilk uygun temel çözümünü bulabilmek için bazı eklemeler yapılır. Temel değişkenleri kolayca seçebilmek için her kısıtlama eşitliğinde çarpan faktörü "1" olan bir değişken bulunmalıdır. "≤ " tipindeki eşitsizlikleri eşitlik haline getirirken eklenen "s" değişkenleri bu zorunluluğu yerine getirir, ama "≥ " tipindeki eşitsizliği, eşitliğe dönüştürmek için çıkartılan "s" değişkenlerinin faktörü "-1" olur ve temel değişken seçiminde kullanılamaz. Bütün eşitliği "-1" ile çarpılırsa bu sefer de eşitliğin sağ tarafını negatif yapma tehlikesine girilir ki simpleks metodunun çalışabilmesi için eşitliklerin sağ tarafının pozitif olması gerekmektedir. Bu duruma bulunan çözüm sisteme yapay bir değişken eklemek ve bu değişkeni temel değişkenler listesine almaktır. Bu eklenen değişkenin sistemin genel çözümünü bozmaması için değişkenin işlemler sonunda sıfır değerini almasını sağlamak gerekmektedir. Bunu sağlamak için yapay değişken amaç fonksiyonuna da çok büyük bir katsayı (M) ile eklenir. Böylece optimum çözümde büyük katsayılı değişkenin sıfır olması sağlanır.

Eğer kısıtlama eşitlik olarak verilmişse içinde "1" katsayılı bir değişken olmayabilir. Aynı mantıkla bu durumda da bir yapay değişken sisteme eklenir ve amaç fonksiyonuna büyük bir katsayı (M) ile eklenir. Minimizasyon amaç fonksiyonuna pozitif büyük katsayı eklenir.

Simplex çözümünde kullanılan tabloların oluşturulması bir örnek ile açıklanacaktır.

Süper Boya firması, iç ve dış cephe için boya üretmektedir. Boya üretimi için iki adet hammadde kullanılmaktadır. İç ve Dış cephe boya üretimi için gereken hammadde miktarları ve kullanılabilecek maksimum hammadde miktarları aşağıdaki tabloda verilmiştir.

Çizelge 3.1 : DP probleminin verileri

	1 Ton Boya üretmek için gereken Hammadde tonları		En Çok Kullanılabilecek Miktar
	Dış Boya	İç Boya	
Hammadde A	1	2	6
Hammadde B	2	1	8

Piyasa araştırmaları gösteriyor ki iç boya talebi dış boya talebinden en çok bir ton fazla olabilir. Ayrıca iç boya talebi en fazla 2 ton olabilir. Dış boyanın tonu 3.000$, iç boyanın tonu da 2.000$ dır. Firma kazancını maksimum yapmak için iç ve dış boyadan ne kadar üretmelidir.

Değişkenler,

x_D = Dış Boya üretim miktarı ve

$x_İ$ = İç Boya üretim miktarı olarak tanımladık.

Amaç fonksiyonu (1000 dolar olarak)

$$\text{Max } Z = 3x_D + 2x_İ \quad \textbf{(3.6)}$$

Kısıtlamalar:

$X_D + 2x_İ \leq 6$ (Hammadde A) **(3.7)**

$2x_D + x_İ \leq 8$ (Hammadde B) **(3.8)**

$x_İ - x_D \leq 1$ (İç boya talebi, diğerinden en çok 1 ton fazla) **(3.9)**

$x_İ \leq 2$ (Talep en çok 2 ton) **(3.10)**

$x_D, x_İ \geq 0$

Problemin standart forma getirilmiş hali aşağıdaki gibidir:

$$\text{Max } Z = 3x_D + 2x_İ + 0s_1 + 0s_2 + 0s_3 + 0s_4 \quad \textbf{(3.11)}$$

Kısıtlamalar:

$$X_D + 2x_İ + s_1 = 6 \quad \textbf{(3.12)}$$

$$2x_D + x_İ + s_2 = 8 \quad \textbf{(3.13)}$$

$$-x_D + x_İ + s_3 = 1 \quad \textbf{(3.14)}$$

$$x_İ + + s_4 = 2 \quad \textbf{(3.15)}$$

$x_D, x_İ, s_1, s_2, s_3, s_4 \geq 0$

Simpleks tablosundan amaç fonksiyonunun değeri şöyle hesaplanır:

$$Z - 3x_D - 2x_İ = 0 \quad \textbf{(3.16)}$$

Tablonun hazırlanmasına geçmeden önce temel değişkenler seçelir. Her eşitlikte bir artık değişken olması ve eşitliklerin sağ taraflarının pozitif olması bulanacak temel çözümün uygun (feasible) olacağını belirtir. Değişken sayısının 6, eşitlik sayısının 4 olmasından 2 tane değişkeni temel olmaya değişken olarak seçilip onlara sıfır değerini atanabilir. Temel değişkenleri seçerken eşitliklerde katsayısı 1 olanların seçilmesi kolaylık sağlayacaktır. Bu durumda x_D ve $x_İ$ sıfır kabul edilerek temel çözüm oluşturulur.

Simpleks metodu her adımda en çok kazanç sağlayacak değişkenin temel değişkenler grubuna katılmasını ve en az getiri sağlayanın temel değişken grubundan ayrılması esasına göre çalışmaktadır. Şimdi işlemleri kolaylaştıracak olan çizelge 3.2' deki simpleks tablosu oluştulur:

Çizelge 3.2 : DP problem çözümü için birinci simplex tablosu

Temel	Z	X_D	$X_İ$	s_1	s_2	s_3	s_4	Sonuç
Z	1	-3	-2	0	0	0	0	0
s_1	0	1	2	1	0	0	0	6
s_2	0	2	1	0	1	0	0	8
s_3	0	-1	1	0	0	1	0	1
s_4	0	0	1	0	0	0	1	2

Çizelge 3.2 oluşturulduktan sonra temel değişkenlere girmesi gereken değişkeni seçimi yapılır. Bu işlemi yaparken z amaç fonksiyonundaki katsayısı en büyük olan değişkeni temel gurubuna almak amaçlanır. Bunun için tabloda z satırına bakılır ve katsayısı negatif olan bir değişken aranır. Eğer birden çok negatif katsayılı değişken varsa içlerinden en küçük katsayılı olanını giriş değişkeni olarak seçilir.

Giriş değişkeni belirlendikten sonra sıra temel değişkenlerden çıkacak değişkeni belirlemeye geliyor. Bunu yapmak için de ayrıldığında en az değer azalışına sebep olacak değişkeni aranır. Tabloyu kullanarak bunu yapmak için sonuç sütunundaki değerleri, giren değişken sütunundaki değerlere bölünür, en küçük olanı seçilir. Bu bölme işleminde giren değişkenin sütunundaki negatif ve sıfır değerler işleme katılmaz. s_2 burada çıkan değişken, x_D ise giren değişken olarak bulunmuştur. Böylece temel değişkenlerde bir değişiklik olur. Tablonun tekrar hesaplanması gerekmektedir. Bu hesaplama işlemini matris operasyonlarını kullanarak yapılır. Yapılması gereken temel değişkenlerin giriş kolonundaki bütün değerleri, çıkış değerinin bulunduğu satır hariç, 0 yapmaktır. Çıkış değişkeni ile giriş değişkeninin kesiştiği kutudaki değerin 1 olması gerekmektedir. Çizelge 3.3' te görüldüğü üzere bu kesişimdeki kutucuğa pivot kutucuk adı verilmektedir.

İşlemlere başlarken ilk yapılması gereken pivot elemanının değerini 1 yapmaktır. Bunun için pivot elemanın bulunduğu satırdaki her değeri pivot elemanının değerine bölünür. Burada kullanılan kural, matematikte Gaus-Jordan çıkarma kuralı olarak bilinen, bir matris'in bir satırını aynı değere bölmek ya da çarpmak ile matris'in değerinin değişmeyeceği kuralıdır.

Çizelge 3.3 : DP problem çözümü için ikinci simplex tablosu

Temel	Z	X_D	$X_İ$	s_1	s_2	s_3	s_4	Sonuç	
Z	1	-3	-2	0	0	0	0	0	
s_1	0	1	2	1	0	0	0	6	6/1=6
s_2	0	2	1	0	1	0	0	8	8/2=4
s_3	0	-1	1	0	0	1	0	1	
s_4	0	0	1	0	0	0	1	2	

Ayrıca matris'in bir satırını diğer bir satıra eklediğimiz zaman da matris'in değeri değişmemektedir. Bu kuralları uygulayarak çizelge 3.4 kolayca oluşturabilinir. İlk olarak giriş satırı düzenlenir:

Çizelge 3.4 : DP problem çözümü için üçüncü simplex tablosu

Temel	Z	X_D	$X_İ$	s_1	s_2	s_3	s_4	Sonuç
Z	1	-3	-2	0	0	0	0	0
s_1	0	1	2	1	0	0	0	6
s_2	0	1	½	0	½	0	0	4
s_3	0	-1	1	0	0	1	0	1
s_4	0	0	1	0	0	0	1	2

Daha sonra da giriş sütunundaki pivot eleman dışında kalan bütün değerleri 0 değerine dönüştürmek için gerekli işlemler yapılır. Bu işlemleri yaparken işlemi bütün bir satıra uygulamak zorunda olunduğu unutulmamalıdır. Örneğin, z satırındaki değer –3, bunu 0 yapmak için pivot elemanın değerini –3 ile çarpım z satırına eklenmesi gerekmektedir. Bu işlemi diğer sütunlar için de yapılması önemlidir. İlk işlemlerden sonra durum şu şekildedir.

Eski z satırı: (1 -3 -2 0 0 0 0 0)

+ (3)x pivot: (0 3 3/2 0 3/2 0 0 12)

Yeni z satırı: (1 0 -1/2 0 3/2 0 0 12)

Eski s_1 satırı: (0 1 2 1 0 0 0 6)

+ (-1)x pivot: (0 -1 -1/2 0 -1/2 0 0 -4)

Yeni s_1 satırı: (0 0 3/2 1 -1/2 0 0 2)

İşlemler diğer satırlar için de yapılırsa aşağıdaki çizelge 3.5 oluşur

Çizelge 3.5 : DP problem çözümü için dördüncü simplex tablosu

Temel	Z	X_D	$X_İ$	s_1	s_2	s_3	s_4	Sonuç	
Z	1	0	-1/2	0	3/2	0	0	12	
S_1	0	0	3/2	1	-1/2	0	0	2	4/3
x_D	0	1	1/2	0	1/2	0	0	4	8
s_3	0	0	3/2	0	1/2	1	0	5	10/3
s_4	0	0	1	0	0	0	1	2	2

Çizelge 3.5' e bakıldığında amaç fonksiyonunda hala negatif değerler olduğunu görülür.. Şimdiki giriş değişkeni en küçük negatif değer olan (aslında tek negatif değer olan) $x_İ$ değişkenidir. Çıkış değişkeni için yine sonuç değerleri sütun değerlerine bölünür. Buradan da en küçük değer seçilir, o değişkeni temel değişkenlerden çıkarılır.

Daha önceki işlemlerin çizelge 3.5' e üzerine uygulanması sonrası çizelge 3.6 elde edilir.

Çizelge 3.6 : DP problem çözümü için beşinci simplex tablosu

Temel	Z	X_D	$X_İ$	s_1	s_2	s_3	s_4	Sonuç	
Z	1	0	0	1/3	4/2	0	0	38/3	12,6
$x_İ$	0	0	1	2/3	-1/3	0	0	4/3	
x_D	0	1	0	-1/3	2/3	0	0	10/3	
s_3	0	0	0	-1	1	1	0	3	
s_4	0	0	0	-2/3	1/3	0	1	2/3	

Çizelge 3.6 optimum düzeye ulaşmıştır. Z satırındaki hiçbir değer negatif değildir, yani temel değişkenlere katılması gerekecek bir değişken kalmamıştır.

Firmanın üretmesi gereken İç boya 4/3 ton, Dış Boya ise 10/3 tondur. Bu durumda kazanç 38/3 bin dolar olacaktır.

3.2.2 Tam sayılı doğrusal programlama

Doğrusal programlama modeli hemen her türlü işletme sorunlarını çözmek için uygulanabilecek yapıda olmasına rağmen, uygulamada ortaya çıkan sonuç, ya işletmenin üretim şeklinden dolayı, ya da doğrusal programlama modelinin uygulandığı sorunun yapısından dolayı istenilen durumu ortaya koymayabilir. Çünkü ekonomik yaşamda her zaman girdi ve çıktıların bölünmezlik sorunları ile karşı karşıya kalınmaktadır. Bölünmezlikleri ele alınan problemlerin çözümleri de tam sayı olmalıdır. Modellerin uygulanmasında, değişkenlerin tam sayı olması şartının incelenmesi ve araştırılması durumunda kullanılacak model; tamsayılı doğrusal programlama modelidir (Ahmet, 1984). Tamsayılı doğrusal programlama modeli, değişkenlerin bazısının veya hepsinin pozitif tamsayılı değer alacağını varsayan matematiksel programlama problemlerinin çözümü ile ilgilenir.

Tam sayılı doğrusal programlama, değişkenlerinden bazılarının veya tamamının tam sayılı (ya da kesikli) değerler aldığı bir doğrusal programlama türüdür (Taha, 1992).

Doğrusal programlama modeli ile tamsayılı doğrusal programlama arasındaki fark, doğrusal programlama modelinde karar değişkenlerinin sıfır ve sıfırdan büyük olma koşulu aranırken, tam sayılı doğrusal programlama da değişken değerlerinin sıfıra eşit ve sıfırdan büyük tam sayı almaları şartının istenmesidir.

Genel olarak tam sayılı doğrusal programlama problemi sembolik olarak şöyle gösterilebilir;

$$\text{Maks. } Z = \sum c_p x_p \qquad \textbf{(3.17)}$$

$$\text{Kısıtlayıcılar } \sum a_{ip} x_p \leq d_i \quad i=1, 2, ...m \qquad \textbf{(3.18)}$$

$$x_p = 0, 1, 2, ...\text{tam sayı} \ \ (P = 1, 2, ...n)$$

tam sayılı doğrusal programlama modeli, değişkenlerin alacağı tam sayı değerlerine göre iki kategoride incelenir. Bunlar:

- Karışık(karma)tamsayılıprogramlama;n tane karar değişkeninden k tanesi için tam sayı olma koşulu, n-k tanesi için pozitif olma koşulu vardır.

- Saf(tüm-tamamen)tamsayılıprogramlama: Karar değişkenlerinin tamamının tam sayı değer alması durumudur.

Saf tamsayılı modellerde tam sayılı değişkenlerin alabilecekleri değerleri tibariyle ikiyeayrılırlar.Tamsayılı değişkenler, kısıtların izin verdiği ölçüde her pozitif tam sayı değeri alabiliyorsa pozitif modeller, sadece 0 ve 1 değerlerini alabiliyorlarsa sıfır-bir modeller olarak adlandırılır.

3.2.2.1 Tamsayılı doğrusal programlama çözüm yöntemleri

Tam sayılı doğrusal programlama çözümleri için, dal-sınır ve kesme düzlemi olarak iki yöntem geliştirilmiştir. İki yöntemden hiçbiri tam sayılı doğrusal programlama problemlerinin çözümünde sürekli daha iyi sonuç verememektedir. Bununla birlikte, deneyimler dal-sınır yönteminin kesme düzlemi yöntemine göre çok daha başarılı olduğunu göstermektedir.

Kesme düzlemi yöntemi,

Doğrusal problemlerinin tamsayılı çözümlerini sağlayacak hesaplama yöntemi 1959 yılında R.E. Gomory tarafından geliştirilmiştir.

Gomory'nin geliştirdiği hesaplama yöntemine tamsayılı algoritma veya kesme düzlemi yöntemi adı verilmiştir. Bu yöntem tüm tamsayılı programlamayı ve karışık tamsayılı programlamayı içermektedir. Bu yöntemde takip edilecek aşamalar şunlardır:

1. Bir tam sayılı doğrusal Pprogramlama probleminde ilk aşama, eğer gerekli ise, orijinal sınırlamaları tamsayılaştırmadır. Bu, katsayılar tam olsun diye, tüm sınırların değiştirilmesi anlamına gelir.

2. Doğrusal probleminin optimal çözüm tablosu bulunur. Eğer optimal çözüm değerleri tamsayı ise, tam sayılı doğrusal programlama problemi için çözüm elde edilmiştir. Yoksa sonraki aşamaya geçilir.

3. Bu aşamada kesme bulunur. Bu amaçla optimal çözüm tablosundan tamsayı olmayan değişkenlerin biri seçilir ve yeni bir kısıtlama elde edilir.

- o Değişkenli problemlerde Gomory kısıtlamasının özellikleri şunlardır:
    - Elde edilen bu kısıtlamalar bir önceki uygun alandan genellikle konveks bir alan keserler.
    - Kesme düzeyi uygun olmasa bile en az bir kafes noktasından geçer.
    - Her kesim, bütün uygun kafes noktalarını kapsayacak daha küçük bir alana yaklaşır.

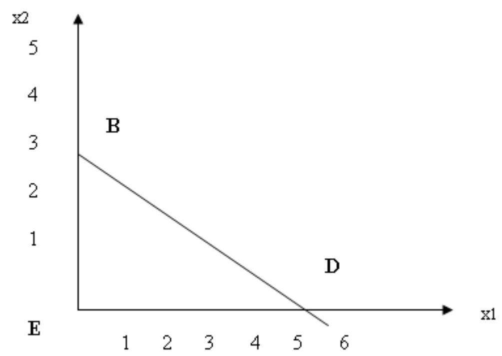
4. Eşitlik sınırlaması (Gomory'nin kesme düzlemi) optimal çözüm tablosuna yeni bir sıra olarak eklenir. Eklenen sınırlamadaki katsayılar tüm tamsayıyı verecek şekildedir. Daha sonra yönteme doğrsal programlama çözüm yöntemleri uygulanarak optimal çözüm tablosu bulunur.

Yöntemi grafik olarak göstermek için bir amaç fonksiyonu ve bir kısıt ele alınsın..

Amaç fonksiyonu: $Z = 2x_1 + 5x_2$ **(3.19)**

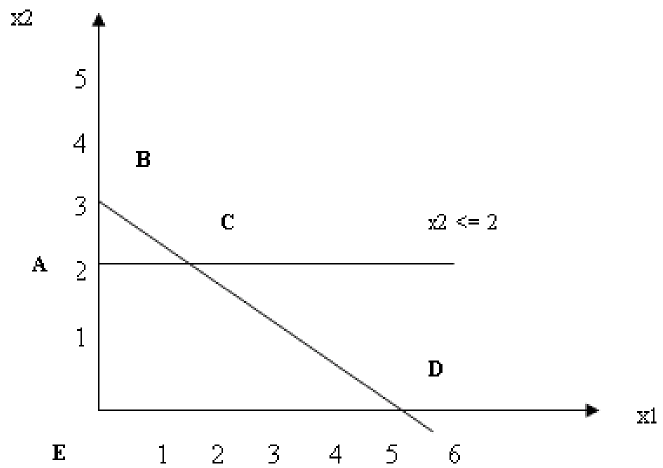
Kısıt : $3x_1 + 6x_2 <= 16$ **(3.20)**

x1 = 16/3 = 5.3 , x2 =16/6 =2.6 doğruları çizildiğinde şekil 3.2' de EDB alanı optimum çözüm bölgesini oluşturmaktadır. Bu bölgedeki B noktası (x1=0, x2=2.6) amaç fonksiyonunu maksimum yapan noktadır. Ancak sonuç tamsayı çıkmadığı için probleme bir kısıt daha ekleyip işleme devam edilmelidir.



Şekil 3.2 : TDP kesme yöntemi örneği birinci grafik

Yeni kısıt : x2 <= 2 kısıtı olsun. Şekil 3.3' de görülen ABC ile gösterilen alanda tamsayılı çözüm olmadığından yeni çözüm bölgesi EACD alanı olup problemin tamsayılı çözümlerini içermektedir. Bu bölgede amaç fonksiyonunu maksimum yapan nokta C (x1 =1, x2=2) noktasıdır.



Şekil 3.3 : TDP kesme yöntemi örneği ikinci grafik

Bundan sonraki kısımda kesme düzlemi algoritmasının simpleks tabloda nasıl uygulandığı açıklanacaktır. Çizelge 3.7' de bir doğrusal programlama problemi ve problemin tamsayı sonuç vermeyen final tablosu görülmektedir.

Amaç fonksiyonu: $\max Z = 3\ x_1 + 5\ x_2$ **(3.21)**

Kısıtlar:

$$x_1 + 4\ x_2 \ <= 9 \quad \textbf{(3.22)}$$

$$2\ x_1 + 3\ x_2 <= 11 \quad \textbf{(3.23)}$$

Çizelge 3.7 : Kesme düzlemi algoritmasının birinci simplex tablosu

amaç fonk.katsa.			3	5	0	0
	taban değişk.	kapasite	x1	x2	S1	S2
5	x2	7/5	0	1	2/5	-1/5
3	x1	17/5	1	0	-3/5	4/5
	Zj	86/5	3	5	1/5	7/5
	Cj - Z j		0	0	-1/5	-7/5

Yeni kısıt eklemek için optimum çözümdeki tamsayı olmayan herhangi bir değişken seçilebilir. Örneğin x_2 değişkenini seçilsin ve bu değişkenin olduğu satır yeniden yazılacaktır. Tam sayı olmayan sayıları ;

Örneğin 4/3 ------→ 1 + 1 / 3

5/4------→ 1 + 1 / 4

2/3------→ 0 + 2 / 3

-2/3 ------→ -1 + 1 / 3

x_2 değişkeninin olduğu satır (tamsayı) + (1 den küçük pozitif kesir) olarak yazılsın.

x_2 (1, 2/5, -1/5, 7/5)

(1+0) x_2+ (0+ 2/5)*S1* + (-1 + 4/5) *S2* = (1+ 2/5)

Tamsayılı katsayıları sağ tarafa alarak yeniden yazıldığında.

2/5 *S1* + 4/5 *S2* = 2/5 +(1- 1 x_2 + 1*S2*)

Tam sayılı kısmı herhangi bir tamsayı olarak düşünüp eşitlikten çıkarılırsa;

2/5 *S1* + 4/5 *S2* > = 2/5 şeklinde yazabilinir.

Probleme yapay(artificial) değişken eklememek için her iki taraf – 1 ile çarparak eşitliğin yönünü değiştirip bir slack(boş) değişken eklenirse kısıt aşağıdaki şekli alacaktır.

-2/5 *S1* -4/5 *S2* + S3 = -2/5

Bu kısıtı aşağıdaki çizelge 3.8' e eklenir.

Çizelge 3.8 : Kesme düzlemi algoritmasının ikinci simplex tablosu

amaç fonk.katsa.			3	5	0	0	0
	taban değişk.	kapasite	x1	x2	S1	S2	S3
5	x2	7/5	0	1	2/5	-1/5	0
3	x1	17/5	1	0	-3/5	4/5	0
0	S3	-2/5	0	0	-2/5	-4/5	1
	Zj	86/5	3	5	1/5	7/5	0
	Cj - Z j		0	0	-1/5	-7/5	0

Bundan sonraki adımda *S3* tabandan çıkacak ve yerine başka bir değişken tabana girecektir. Tabana girecek değişkenin seçimi için *Cj - Z j* satırındaki negatif elemanlar bunlara karşı gelen S3 satırındaki negatif katsayılarla oranlanır. En küçük orana sahip sütundaki değişken tabana girecek değişkendir.

(- 1/5) / (-2/5) = 1 / 2 ve (-7/5) / (-4/5) = 7/4 oranlarına bakılırsa (1/2) en küçük değer olduğu için bu sütundaki *S1* değişkeni, *S3* yerine tabana girecek değişkendir. Bir sonraki simpleks tablo çizelge 3.9 ‘ da verilmektedir.

Çizelge 3.9 : Kesme düzlemi algoritmasının üçüncü simplex tablosu

amaç fonk.katsa.			3	5	0	0	0
	taban değişk.	kapasite	x1	x2	S1	S2	S3
5	x2	1	0	1	0	-1	1
3	x1	4	1	0	0	2	-3/2
0	S1	1	0	0	1	2	-5/2
	Zj	17	3	5	0	1	1/2
	Cj - Z j		0	0	0	-1	-1/2

Çizelge 3.9 ‘ daki tablodaki sonuçlara bakılırsa;

$x_1 = 4$, $x_2 = 1$ ve Z = 17 çıkmaktadır.

Tamsayılı sonuç elde edildiği için iterasyona son verilir. Ele alınan örnekte tek kısıt ilavesi ile optimum sonuca ulaşılmaktadır. Bu her zaman gerçekleşmeyebilir. Eklenen ilk kısıttan sonra elde edilen sonuç hala tamsayı değilse yeniden bir kısıt daha eklenerek tam sayılı sonuç alınıncaya kadar işlemler tekrarlanır.

TDP problemlerinin değişkenlerinin bir kısmı veya hepsi bazen üst veya alt sınırla veya hem alt hem de üst sınırla ayrı ayrı şartlanmaktadır. Bu tipteki optimizasyon problemlerinin çözümü için çok genel bir yaklaşım dal-sınır tekniğidir. Bu yöntem ile minimizasyon tipli bir problem çözülürken temel mantık şöyledir (Anderson ve diğ., 2006):

1. Tam sayılı programlama uygun çözüm alanı çok sayıda alt setlere ayrılır. Her bir alt set için sınır değeri hesaplanır. Alt ve üst sınırlardan, her bir alt set içinde çözüm değeri seçilir.
2. Bir alt sınır (ilk alt setler arasından en küçüğü) ile alt set diğer bölümler için seçilir. Daha önce olduğu gibi alt ve üst sınırların her ikisi de hesaplanır. Bölünmeye optimal çözüm bulunana kadar devam edilir. Optimal çözüm herhangi bir alt set için alt sınırdan daha büyük olamaz. Problem

maksimizasyon tipli ise, çözüm yöntemi alt ve üst sınır seçimi dışında minimizasyon tipli problem gibidir.

Dal sınır yöntemini bir örnek problem üzerinde şu şekilde incelenebilir:

Örnek problem:

$$\max Z = 5x_1 + 4x_2 \tag{3.24}$$

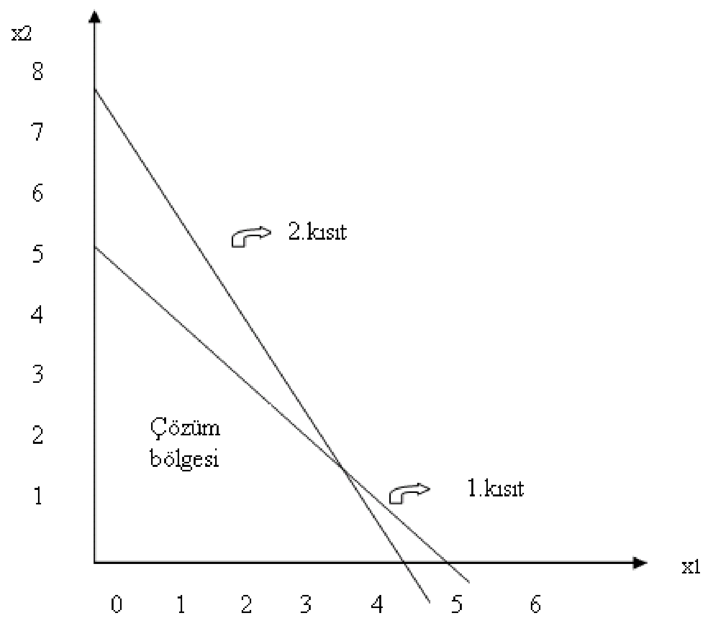
Kısıtlar:

$$x_1 + x_2 <= 5 \tag{3.25}$$

$$10x_1 + 6x_2 <= 45 \tag{3.26}$$

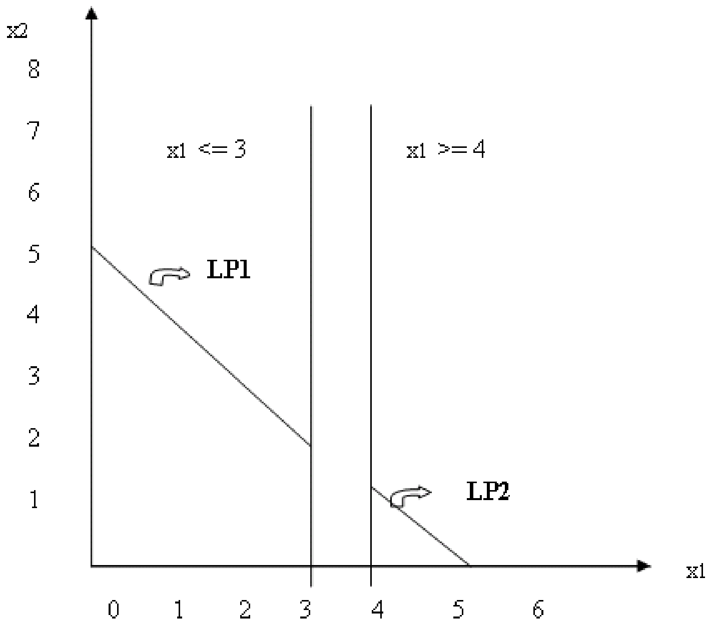
Pozitiflik koşulu: x_1 , x_2 >= 0

Problemin çözüm uzayı şekil 3.4' te görüldüğü gibidir ve grafik çözümden sonuçların tam sayı çıkmadığı görülmektedir.



Şekil 3.4 : TDP dal sınır yöntemi örneği birinci grafik

Problemin çözümünde : Z= 23.75, x_1 = 3.75, x_2 =1.25 çıkmaktadır. Değişkenler tam sayı çıkmadığı için dal-sınır algoritması ile optimum tam sayılı çözümü buluncaya kadar çözüm uzayının düzenlenmesi yapılacaktır. İlk aşamada LP0 çözümünde(DP çözümü) tam sayı değer almayan bir değişken rasgele seçilir. x_1 değişkenini seçilsin.(x1=3.75) LP0 çözüm uzayının 3< x1 <4 bölgesinde tamsayılı değerler olmayacaktır dolayısıyla bu bölge elemine edilebilir.



Şekil 3.5 : TDP dal sınır yöntemi örneği ikinci grafik

LP1 uzayı = LP0 uzayı + (x1 < = 3)

LP2 uzayı = LP0 uzayı + (x1 >= 4)

Optimum çözüm ya LP1 uzayında ya da LP2 uzayında olacaktır. Her iki alt problem ayrı ayrı çözülmesi gerekir. Önce LP1 problemi (x1 <=3) kısıtını ekleyerek ele alınırsa;

$$x1 <= 3 \qquad \textbf{(3.27)}$$

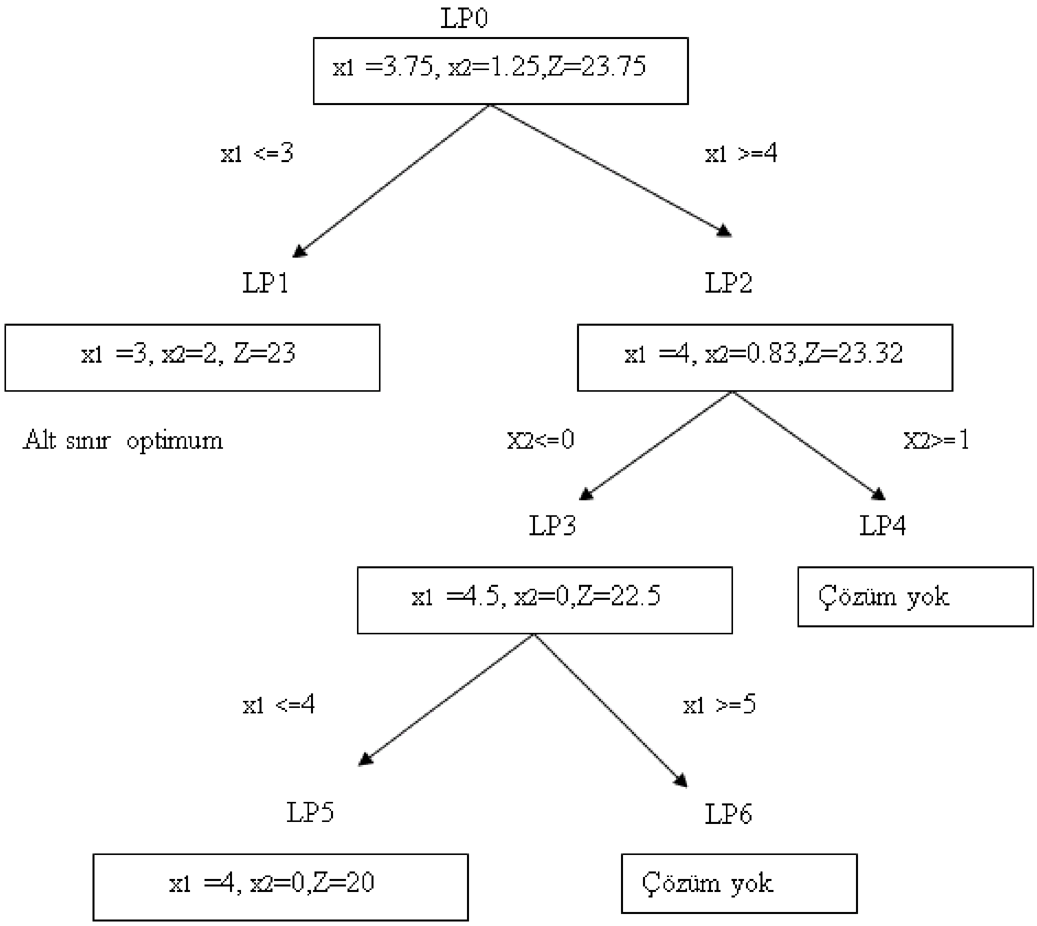
$$x1,x2 >= 0$$

problem çözüldüğünde Z= 23, x_1 = 3, x_2 =2 çıkmaktadır. O halde LP1 optimum değere ulaşmıştır denilebilir. Çözümdeki Z=23.75 değeri LP1 de Z=23 olarak çıktığına göre bu bir alt sınır olarak alınabilir. Bir tamsayılı çözüm elde edildiği için daha fazla ilerletmeye gerek yoktur. LP2 ye ait çözümde de Z= 23.32 , x_1 =4, x_2=0.83 çıkmaktadır. x_2=0.83 olduğundan yeniden bir dallanma yapılarak ;

x2<= 0 ve x2 >=1 kontrolü yapılabilir.

LP3 uzayı = LP2 uzayı + (x2 <= 0)

= LP0 uzayı +(x1 >=4)+ (x2<=0)



Şekil 3.6 : TDP dal sınır yöntemi örneği dal sınır şeması

Şekil 3.6' da görüldüğü üzere LP5 çözümünde de sonuç tamsayılı çıkmaktadır. Ancak LP1 çözümünde Z = 23 alt sınır olarak alınırsa(en büyük alt sınır) bu çözümün optimum olmadığı söylenebilir. Burada hangi dalın seçilip önce çözülmesi konusunda kesin bir kural olmayıp seçim tahmini yapılmaktadır.

3.2.3 Dualite ve duyarlılık analizi

Bölüm 3.2.1' de bahsedilen doğrusal programlama modeli primal problem olarak anılmaktadır. Dual problem ise, primal problemden doğrudan türetilebilen matematiksel tanımla yakından ilgilidir.

Birçok doğrusal programlama işleminde dual, optimizasyon anlamına, kısıtlaın tipine ve değişkenlerin işaretine bağlı olarak primalin çeştli durumları için tanımlanmıştır.

Dual problemin kısıtları ve değişkenleri primal problemden simetrik olarak aşağıdaki gibi oluşturulabilir:

- m tane primal kısıt denkleminin her biri için bir dual değişken tanımlanmıştır.
- N tane primal değişkenin her biri için bir dual kısıt tanımlanmıştır.
- Dual kısıtın sol taraf katsayıları ile ilgili primal değişkenin kısıt katsayılarına eşittir. Sağ taraf sabiti ise aynı primal değişkenin amaç fonksiyonu katsayısına eşittir.
- Dualin amaç fonksiyonu katsayıları primal kısıt denklemlerinin sağ tarafına eşittir.

Normal Enbüyükleme Sorununun Dualini Bulma

PRİMAL

$$
\begin{array}{ll}
\text{maks } z = & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
\text{öyle ki} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\
 & \ldots \quad \ldots \quad \ldots \quad \ldots \\
 & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \\
 & x_j \ge 0 \ (j = 1, 2, \ldots, n)
\end{array}
$$

DUAL

$$
\begin{array}{ll}
\text{min } w = & b_1y_1 + b_2y_2 + \ldots + b_my_m \\
\text{öyle ki} & a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \ge c_1 \\
 & a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \ge c_2 \\
 & \ldots \quad \ldots \quad \ldots \quad \ldots \\
 & a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \ge c_n \\
 & y_i \ge 0 \ (i = 1, 2, \ldots, m)
\end{array}
$$

Normal Enküçükleme Sorununun Dualini Bulma

PRİMAL

$$
\begin{array}{ll}
\text{min } w = & b_1y_1 + b_2y_2 + \ldots + b_my_m \\
\text{öyle ki} & a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \ge c_1 \\
 & a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \ge c_2 \\
 & \ldots \quad \ldots \quad \ldots \quad \ldots \\
 & a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \ge c_n \\
 & y_i \ge 0 \ (i = 1, 2, \ldots, m)
\end{array}
$$

DUAL

$$
\begin{array}{ll}
\text{maks } z = & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
\text{öyle ki} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\
 & \ldots \quad \ldots \quad \ldots \quad \ldots \\
 & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \\
 & x_j \ge 0 \ (j = 1, 2, \ldots, n)
\end{array}
$$

Primal en büyükleme ve en küçükleme sorusunun dualinin nasıl bulunduğunun örneği yukarıdaki gibidir.

3.3 Sezgisel Çözüm Metotları

3.3.1 Lineer programlama prosedürü

Lineer programlama prosedürü, algoritmik ve sezgisel metotların kombinasyonu olup stok kesme probleminin lineer programalama gevşetmesinin çözümü için kullanılan bir metottur.

Bu metot, problemin genişletilmiş versiyonu için optimum çözümü sunmaktadır ve çözüm elde edildikten sonra tam sayılı değerlere ulaşabilmek için özel metotlar kullanılabilir.

Lineer programlama prosedürü anlatımı sırasında bir boyutlu stok kesme problemi formulize edilecektir. Dyckoff' nun yaptığı sınıflandırmaya göre ilgili problemimiz 1/V/I/R olarak adlandırılabilir.

- 1= problemin boyutunun bir olduğunu belirtir
- V= stok malzemelerin hepsinin kullanılmadan tüm siparişlerin karşılandığını belirtir.
- I= aynı özellikte birden fazla stok malzemesinin olduğunu belirtir.
- R= birbirine yakın benzer boyutlardaki çok sayıdaki küçük parçalar ın olduğunu belirtir.

Problemin formülasyonu , notasyonu , değişkenlerin ve parametrelerin tanımları şu şekildedir:

$$\min \sum_{p=1}^{P} x_p \tag{3.28}$$

$$s.t \sum_{p=1}^{P} a_{i,p} x_p \geq d_i \tag{3.29}$$

$x_p \geq 0$

$i =$ siparişlere ait indeks

$p =$ yerleşim planlarına ait indeks

$x_p =$ p yerleşim planının kullanım sayısı

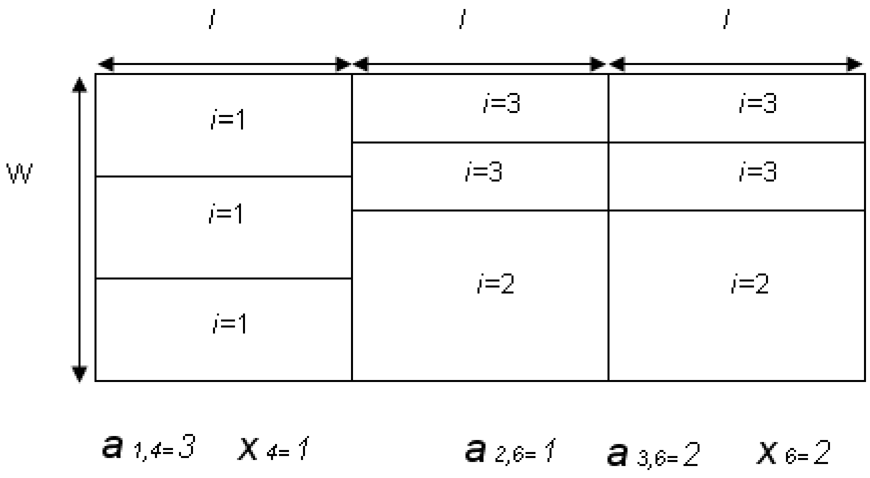
a_{ip} = p yerleşim planında kesilen i siparişinin miktarı

$d_i =$ i siparişinin talep miktarı

$w_i =$ i siparişinin eni

$M =$ stok malzemesinin eni

Denklem 3.28 ' de amaç fonksiyonu belirtilmiştir ve p yerleşim planlarının uygulanmasıyla elde edilen kesme kayıplarını minimize edilmesi amaçlanmaktadır. Ayrıca denklem 3.29' daki denklemde yerleşim planlarında kesilen toplam sipariş miktarlarının, sipariş talep miktarınından eşit veya büyük olması istenmektedir.



Şekil 3.7 : 1 boyutlu stok kesme probleminde tamsayılı değişkenlerin ilişkisi

Şekil 3.7' te a_{ip} ve x_p deişkenlerinin birbiriyle olan ilişkisini göstermektedir. İlk stok malzemesi p=4 numaralı yerleşim planı ile kesilerek 3 adet *i*=1 siparişinden karşılanmaktadır. İkinci ve üçüncü stok malzemeleri p= 6 numaralı yerleşim planı ile kesilirken *i*= 2 siparişinden 1 adet *i*=3 siparşinden 2 adet karşılanmaktadır.

P çözüm uzayında bulunan yerleşim planlarının sayısının çok fazla olması ile problemin çözümünün hesaplanması oldukça zor hale gelebildiği durumlar ile karşılaşabilinmektedir. Bu yüzden bu tür durumlarda probleme ait tüm olası yerleşim planlarını kullanmadan çözüme ulaşma ihtiyacı doğabilmektedir. Bunun için sütun oluşturma yönteminden faydalanılır.

Sütun oluşturma yöntemi uygulaması sırasında yeni eklenecek yerleşim planlarının bulunmasında sırt çantası probleminin algoritması kullanılır. Sütun oluşturma çözümünün ve sırt çantası algoritmasının örneğimize nasıl uygulandığını göstermeden önce bu tekniklerin neler olduğu ve nasıl uygulandığı konusunda açıklamalar aşağıda sırasıyla belirtilmiştir.

3.3.1.1 Sütun(kolon) oluşturma yöntemi

Sütun oluşturma yöntemi özellikle kapsamı geniş olan lineer problemlerin çözümünde verimli olan bir algoritmadır.

Bu yöntemin temelinde, bazı değişkenlerin çözüm uzayında sıfır değeri almakta olduğu için çözüm sırasında bu değişkenlerin olabildiğince dikkate alınmaması düşüncesi yatmaktadır.

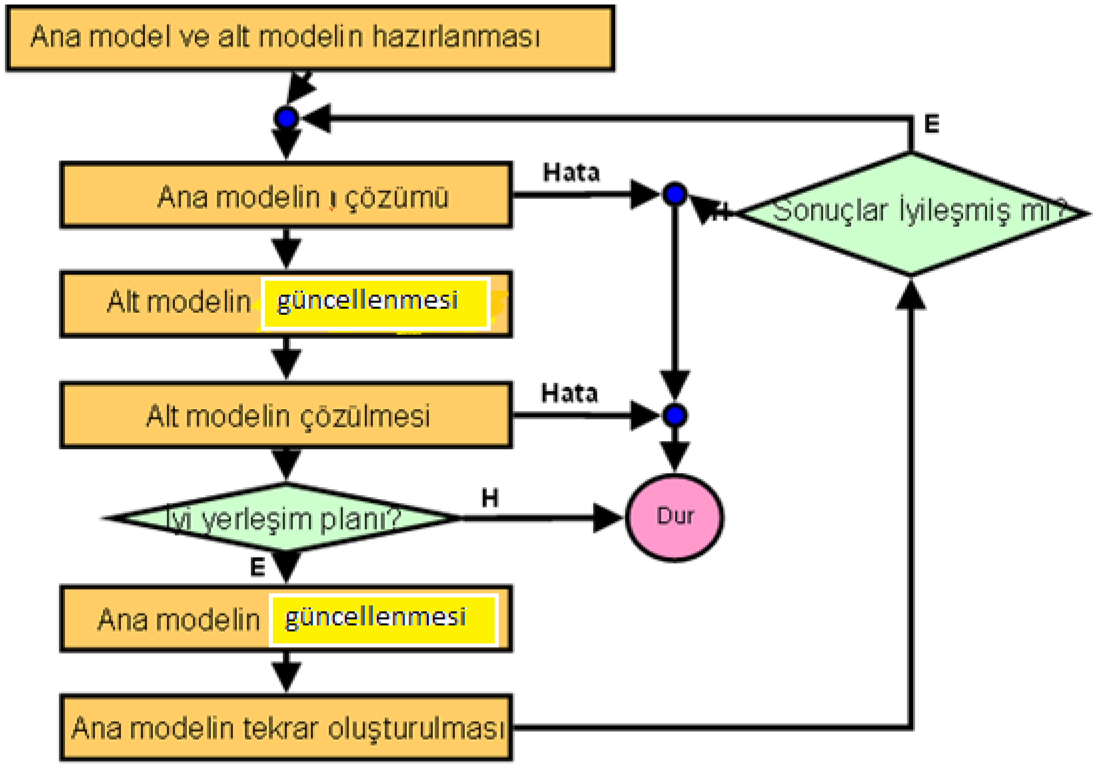
Bu noktadan hareket ile kolon oluşturma yöntemi sadece amaç fonksiyonunu geliştirmeye yönelik olan değişkenlerin oluşturmasını dikkate alır ve problemin minimizasyon problemi olduğu varsayılırsa en büyük negatif indirgenmiş maliyeti olan değişkeni bulmaya çalışır.

Şekil 3.8 : Ana model-alt model arasındaki ilişki

Şekil 3.8' te ana model-alt model ilişkisi görülmektedir. Ana model çözülerek ana modelde yer alan her kısıt için dual maliyetler elde edilir. Dual maliyetler alt modelin amaç fonksiyonunda kullanılarak alt model çözülür. Eğer alt modelin amaç değeri negatif ise, alt model içerisinde negatif değerde indirgenmiş maliyetlerin olduğu belirlenir ve en yüksek değerdeki negatif sütun alt modelden üst modele aktarılarak ana model tekrar çözülür. Ana modelin tekrar çözülmesiyle yeni dual maliyetler elde edilerek alt modelin negatif değerde indirgenmiş maliyet bulundurmamasına kadar döngü devam ettirilir.alt model negatif değerde indirgenmiş maliyet bulundurmadığı zaman ana modelin optimale ulaştığı kabul edilir.

Sütun oluşturma yönteminin en bilinen uygulaması stok kesme problemindedir. Şekil 3.9' da ilgili prosedür şematize edilmiştir. Stok kesme probleminde ana modelde kesim kaybını minimize etmeyi amaçlayan bir model olarak kurulur, alt model ise ana modelin kullanacağı en verimli kesim planlarını bulmasını sağlayacak modeldir. Ana model öncelikle optimal çözüme imkan vermeyecek kesim planları ile çözüme başlar ve daha sonra alt modelin ana modele eklenmesini belirttiği yeni kesim planlarını ile çözüme devam eder. Alt model ana modele yeni kesim planı eklemeyi durdurduğunda ana model optimuma ulaştığı kabul edilir.

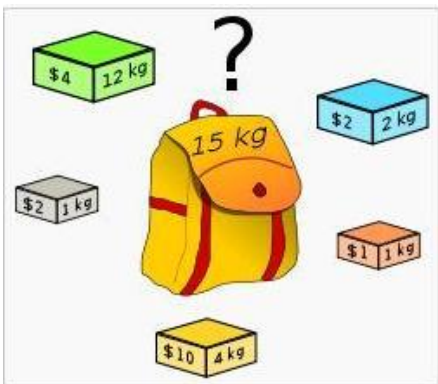
Bir boyutlu stok kesme probleminde yeni kesim planlarının eklenmesinde sırt çantası probleminden (knapsack problem)olarak adlandırılan yardımcı bir optimizasyon probleminden faydalanılır. Problem ana problem ve alt problem olarak iki kısma ayrılarak çözülür, ana problem sadece değişkenlerin alt kümesin dikkate alan orijinal problemdir, alt problem ise yeni değişken belirlemeye çalışan yeni bir problemdir



Şekil 3.9 : Ana model-alt model prosedürü

Sırt Çantası Problemi(Knapsack); Sırt çantası problemi en ünlü NP-zor problemleri arasındadır. Sözde-polinomsal zamanda çözümünü sağlayan algoritmalar bulunmaktadır. Problemin karar problemi uyarlamasının NP-tam olduğu kanıtlanmıştır. Problemin polinomsal zamanda çözümü ispatlanabilirse P = NP savı da ispatlanmış olacaktır.

SÇP basitçe bir çantanın içerisine en fazla eşyanın yerleştirilmesini hedefler. Problem hırsız örneğinden daha iyi anlaşılabilir. Buna göre bir hırsız şekil 3.11' de 15 kg ağırlığındaki çantasına ağırlıkça en az, pâhâca en çok eşyayı doldurmak ister ve bunu nasıl yapmalıdır.



Şekil 3.10 : Sırt çantası problemi örneği

Sırt çantası problemi genel olarak üç başlıkta toplanabilir:

Aşağıdaki her özel problem durumu için eşyaların pahası s_i olsun ve her eşyanın ağırlığı t_i olsun. Çantanın taşıyabileceği azamî kapasite de P olarak tanımlansın.

0&1 sırt çantası problemi

Bu problem tipinde eşyalar ya tamamen alınır ya da tamamen bırakılır. Alınacak olan eşyanın bir kısmını almak mümkün değildir. Dolayısıyla bir eşyanın alınıp alınmamasını x_i ile gösterecek olursak problem aşağıdaki şekilde modellenebilir:

$$\text{Max } \sum_{i=0}^{n} s_i x_i \tag{3.30}$$

$$\sum_{i=0}^{n} t_i x_i \leq P \quad x_i \in \{0,1) \tag{3.31}$$

Sınırlı sırt çantası problemi,

Bu problem tipinde ise her eşyadan alınabilecek miktarda sınır vardır ve bu da c_i olsun. Bu problemin modeli aşağıda verilmiştir.

$$\text{Max } \sum_{i=0}^{n} s_i x_i \tag{3.32}$$

$$\sum_{i=0}^{n} t_i x_i \leq P \quad x_i \in \{0,1,......c_i) \tag{3.33}$$

Sınırsız sırt çntası problemi,

Bir önceki sınırlı sırt çantası probleminden farklı olarak bu problemde alınabilenecek malzemelerin herhangi bir sınırı bulunmamaktadır.

3.3.1.2 Lineer programlama prosedürünün uygulama örneği

Bu kısımda, lineer programlama prosedürünün(LPP) bir boyutlu tek tip stok malzemeli stok kesme problemine ait uygulamalı bir örneğe yer verilecek ve örnek problem üç farklı yöntem ile çözülecektir.

Birinci çözüm, problemin ana model-alt model ilişkisi çerçevesinde herhangi bir paket program kullanmaksızın manuel olarak lineer programlama ile çözümünü kapsamaktadır.

İkinci çözüm, problemin ana model-alt model ilişkisi olmadan GAMS paket programından faydalanarak tüm yerleşim planlarının kullanıldığı lineer programlama ile çözümünü kapsamaktadır.

Son çözüm ise, birinci çözümün tekniğine benzer olarak fakat problemin manuel çözümü yerine GAMS paket programından faydalanarak çözümünü içermektedir. Öncelikle problemden kısaca bahsetmek gerekirse, bu örnekte tek ende sınırsız sayıda stok malzemesi bulunmaktadır. Bu stok malzemelerinden minimum sayıda kullanarak istenen en ve miktarlardaki siparişlerin karşılanması amaçlanmaktadır. Problemin çözümünde kullanılacak siparişlere ait örnek veriler çizelge 3.10' da belirtilmiştir. Ayrıca kesilecek stok malzemesinin eni 2000 mmdir

Çizelge 3.10 : LPP örneği sipariş en&miktar verileri

Sipariş Eni(mm)	Talep Miktarı(adet)
900	511
800	301
700	263
600	383

Problemin modellenmesi için daha önce tanımlanan denklem 3.28, 3.29' daki amaç foksiyonun, parametrelerin, değişkenlerin kullanılması uygundur.

$i =$ siparişlere ait indeks

$p =$ yerleşim planlarına ait indeks

$x_p =$ p yerleşim planının kullanım sayısı

$a_{ip} =$ p yerleşim planında kesilen i siparişinin miktarı

$d_i =$ i siparişinin talep miktarı

$w_i =$ i siparişinin eni

$M =$ stok malzemesinin eni

Problemin çözümün temelinde stok malzemesinin minimum sayıda kullanılmasını sağlayacak yerleşim planlarının tespit edilmesi yatmaktadır. 2000 mm enindeki stok malzemesini, 900 mm, 800 mm, 700 mm ve 600 mm enlerindeki siparişleri ele aldığımızda çizelge 3.11' den görüldüğü üzere toplam 18 adet yerleşim planını alternatifi mevcuttur.

Bunlardan hangileri ne kadar sayıda problemin çözümünde bulunması gerekmektedir.

Çizelge 3.11 : LPP örneğinde tüm yerleşim planları

	900 mm	800 mm	700 mm	600 mm
YerlesimPlanı1	2			
YerlesimPlanı2	1	1		
YerlesimPlanı3	1		1	
YerlesimPlanı4	1			1
YerlesimPlanı5	1			
YerlesimPlanı6		2		
YerlesimPlanı7		1	1	
YerlesimPlanı8		1		1
YerlesimPlanı9		1		2
YerlesimPlanı10		1		
YerlesimPlanı11			2	
YerlesimPlanı12			2	1
YerlesimPlanı13			1	1
YerlesimPlanı14			1	2
YerlesimPlanı15			1	
YerlesimPlanı16				3
YerlesimPlanı17				2
YerlesimPlanı18				1

Tüm yerleşim planı alternatifini denemek yerine işe yarayacak olan yerleşim planlarını tespit etmek amacıyla 3.3.1.1’ de detaylı olarak anlatılan ana model-alt model ilişkisi çerçevesinde sütun oluşturma yönteminden faydalanılacaktır.

Ana model-alt model ilişkisini kurmak için denklem 3.28’ deki amaç fonksiyonunu ve denklem 3.29’ deki eşitsizliğinin duali alınırsa yeni oluşan denklemler şu şekildedir:

$$\max \sum_{i=1}^{I} d_i y_i \qquad \textbf{(3.34)}$$

$$s.t \sum_{p=1}^{P} a_{i,p} y_i \leq 1 \qquad \textbf{(3.35)}$$

y_i sınırlandırılmamış dual değişkendir.

$c_p - y.P_p \leq 0$; $1 - y.P_p \leq 0$; $y.P_p \geq 1$; $y.B = C_B$

Sorunun çözümüne başlamak için en çok sayıda stok malzemesini kullanacak dahi olsa öncelikle ana modelin çözülmesi ve buradan çıkan dual değerlerin alt modelde kullanılması gerekmektedir. Ana modelin çözümüne başlama için 18 adet yerleşim planınından tüm siparişlerin ihtiyaçlarını karşılayacak yerleşim planlarının seçilmesi gerekmektedir. Bunun için 2000 mm enindeki stok malzemesinden sadece 900 mm en siparişi , sadece 800 mm en siparişi, sadece 700 mm en siparişi ve sadece 600 mm en siparişi karşılayacak yerleşim planları başlangıç yerleşim planı kümesinin elemanı olarak belirlenir (modelin manuel çözümü sırasında malzeme eni ve sipariş enlerinin 100 e bölünmüş halleri ile işlemler yapılmıştır). Diğer bir tanımlama ile yerleşimplanı1, yerleşimplanı6, yerleşimplanı11, yerleşimplanı16 aşağıdaki başlangıç matrisini oluşturmaktadır.

$$B = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \tag{3.36}$$

Başlangıç matrisininin kullanılmasıyla kullanılan stok malzemesi miktarı;

255,5+150,5+131,5+127,66=665,16 adettir.

Bu başlangıç çözümünden hareket ile ana modele ait y_i dual değişkenler bulunacaktır.

$$yB = C_B \tag{3.37}$$

C_B, X_p önündeki katsayılar olup malzeme tek tip ve değişken olmayan maliyete sahip olduğu için değeri 1 dir.

$$\begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix} . \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.38}$$

Denklem 3.38’ deki matrisin çözülmesiyle ana modele ait y_i dual değişkenleri hesaplanır ve denklem 3.39’ da değerleri görülmektedir.

.

$$y = [1/2 \quad 1/2 \quad 1/2 \quad 1/3] \qquad \textbf{(3.39)}$$

Ana modele ait dual değişkenlerin denklem 3.39' de elde edilmesi ile alt model çözülerek ana modele dahil olabilecek yeni yerleşim planının olup olmadığı kontrol edilir.

$$9a + 8b + 7c + 6d \leq 20 \qquad \textbf{(3.40)}$$

$$\frac{1}{2}a + \frac{1}{2}b + \frac{1}{2}c + \frac{1}{3}d > 1 \qquad \textbf{(3.41)}$$

$a, b, c, d \geq 0$, tamsayı

Denklem 3.40 ve 3.41' daki kısıtlar sırt çantası problemine uygun olarak tekrar düzenlenirse;

$$\max \quad \frac{1}{2}a + \frac{1}{2}b + \frac{1}{2}c + \frac{1}{3}d \qquad \textbf{(3.42)}$$

$$9a + 8b + 7c + 6d \leq 20 \qquad \textbf{(3.43)}$$

$a, b, c, d \geq 0$, tamsayı

Yukarıdaki problemde tamsayı kısıtından gevşetilirse problem tek kısıtlı basit bir doğrusal probleme dönüşür ve bu tür problemlerde amaç fonksiyonundaki bir değişken katsayısı kısıttaki aynı değişkenin katsayısına bölünmesi ile elde edilen sayının en büyük ola nın optimal çözüm içerisinde olacağı bilinmektedir. Bu noktadan hareketle değişkenlere ait katsayıların oranlanmasıyla problem yeniden yazılır;

$$x_3 = \frac{1}{18} \qquad x_2 = \frac{1}{16} \qquad x_1 = \frac{1}{14} \qquad x_4 = \frac{1}{18} \qquad \textbf{(3.44)}$$

$$\max \quad \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 + \frac{1}{3}x_4 \qquad \textbf{(3.45)}$$

$$7x_1 + 8x_2 + 9x_3 + 6x_4 \leq 20 \qquad \textbf{(3.46)}$$

$X_p \geq 0$

Kesirli yapıdan kurtulmak için amaç fonksiyonu 6 ile çarpılır

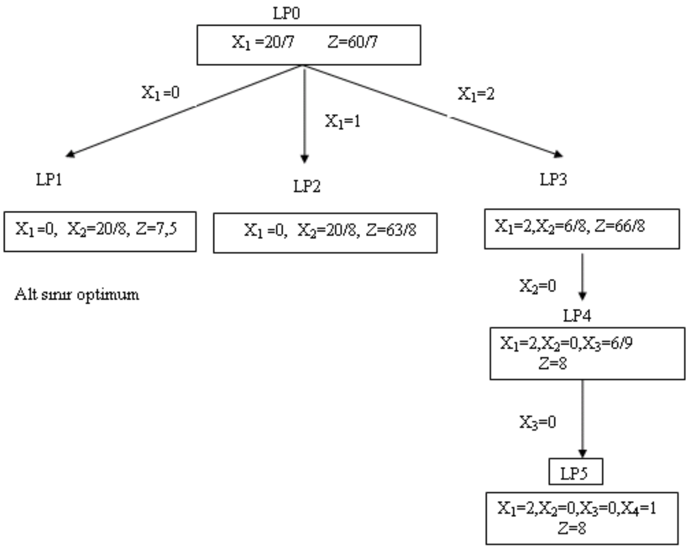
$$\max \quad 3x_1 + 3x_2 + 3x_3 + 2x_4 \tag{3.47}$$

$$7x_1 + 8x_2 + 9x_3 + 6x_4 \leq 20 \tag{3.48}$$

Denklem 3.47 ve 3.48' nin kullanılmasıyla modelin dal-sınır yöntemi ile çözümü şekil 3.11' de görülmektedir

X_1=2, X_2=0, X_3=0, X_4=1 ve Z=8 sonuçlarını veren LP5 çözümünde sonuç tamsayılı çıkmaktadır. Bu sonuca göre 700 mm siparişten 2 adet, 600 mm en siparişten 1 adedi içeren aşağıdaki matris ana modele dahil olacaktır.

$$\begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix} \tag{3.49}$$



Şekil 3.11 : LPP örneğine ait birinci dal sınır çizelgesi

Ana modele dahil olacak matris ile birlikte mevcut yerleşim planlarından bir tanesinin yerleşim planı seçenekleri arasından çıkması gerekmektedir. Çıkması gereken matrisin hesabı şu şekilde yapılır.

$$B.P^{\bar{i}} = P_p \tag{3.50}$$

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix} \qquad \textbf{(3.51)}$$

Aşağıdaki matrise göre 3. veya 4. sütunlarından bir tanesi çıkan matris olmalıdır.

$$P_p^{\bar{7}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1/3 \end{bmatrix} \qquad \textbf{(3.52)}$$

$$255{,}5 - 0.\theta \geq 0 \qquad \textbf{(3.53)}$$

$$150{,}5 - 0.\theta \geq 0 \qquad \textbf{(3.54)}$$

$$131{,}5 - \theta \geq 0 \qquad \textbf{(3.55)}$$

$$127{,}66 - \frac{1}{3}\theta \geq 0 \qquad \textbf{(3.56)}$$

Denklem 3.55' e göre göre θ değerinin hesaplanmasından dolayı tüm yerleşim planlarını içeren matristeki üçüncü sütun çıkan sütun olarak tespit edilir. Bu durumda ana modelin çözümünde kullanılacak tüm yerleşim planlarını gösteren matris şu şekildedir:

$$B = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \qquad \textbf{(3.57)}$$

Bu matris içerinde kullanılan stok malzemesi sayısı;
255,5+150,5+131,5+83,83=621,33 adettir ve görüldüğü üzere 43,43 adet stok malzemesi yeni giren yerleşim planı sayesinde daha az kullanılacaktır.

Ana modele ait yukarıdaki matristen faydalanarak tekrar yeni dual değişlerler *y* hesaplanır.

$$yB = C_B$$

$$\begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \qquad \textbf{(3.58)}$$

Denklem 3.58' deki matrisin çözülmesiyle ana modele ait yeni y_i dual değişkenleri hesaplanır ve denklem 3.59' da değerleri görülmektedir.

$$y = \begin{bmatrix} 1/2 & 1/2 & 1/3 & 1/3 \end{bmatrix} \tag{3.59}$$

Ana modele ait dual değişkenlerin denklem 3.59' da elde edilmesi ile alt model çözülerek ana modele dahil olabilecek yeni yerleşim planının olup olmadığı kontrol edilir.

$$9a + 8b + 7c + 6d \leq 20 \tag{3.60}$$

$$\frac{1}{2}a + \frac{1}{2}b + \frac{1}{3}c + \frac{1}{3}d > 1 \tag{3.61}$$

$$a, b, c, d \geq 0$$

Yukarıdaki kısıtlar sırt çantası problemine uygun olarak tekrar düzenlenirse;

$$\max \quad \frac{1}{2}a + \frac{1}{2}b + \frac{1}{3}c + \frac{1}{3}d \tag{3.62}$$

$$9a + 8b + 7c + 6d \leq 20 \tag{3.63}$$

Yukarıdaki değişkenlere ait katsayıların oranlanmasıyla problem yeniden yazılır;

$$x_2 = \frac{1}{18} \qquad x_3 = \frac{1}{18} \qquad x_1 = \frac{1}{16} \qquad x_4 = \frac{1}{21} \tag{3.64}$$

$$\max \quad \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 + \frac{1}{3}x_4 \tag{3.65}$$

$$7x_1 + 8x_2 + 9x_3 + 6x_4 \leq 20 \tag{3.66}$$

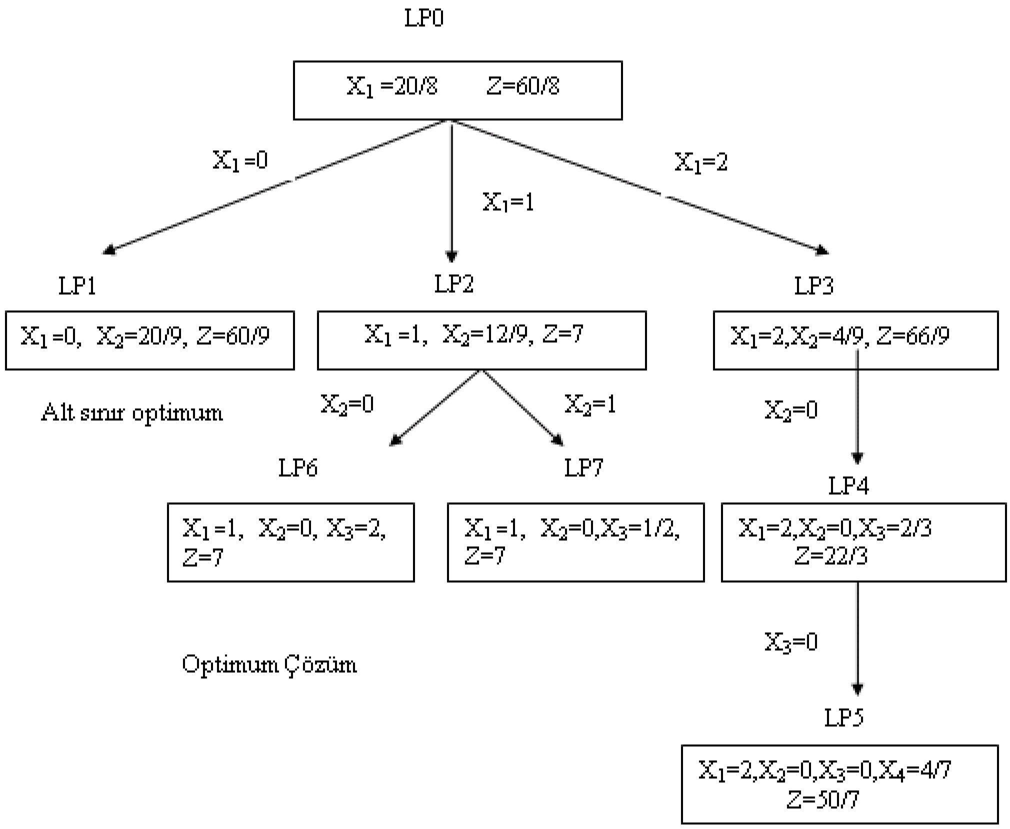
$$X_p \geq 0$$

Kesirli yapıdan kurtulmak için amaç fonksiyonu 6 ile çarpılır

$$\max \quad 3x_1 + 3x_2 + 3x_3 + 2x_4 \tag{3.67}$$

$$7x_1 + 8x_2 + 9x_3 + 6x_4 \leq 20 \tag{3.68}$$

Denklem 3.67 ve 3.68' nin kullanılmasıyla modelin dal-sınır yöntemi ile çözümü şekil 3.12' de görülmektedir



Şekil 3.12 : LPP örneğine ait ikinci dal sınır çizelgesi

Bu sonuca göre 600 mm siparişten 2 adet, 800 mm en siparişten 1 adedi içeren aşağıdaki matris ana modele dahil olacaktır.

Ana modele dahil olacak matris ile birlikte mevcut yerleşim planlarından bir tanesinin yerleşim planı seçenekleri arasından çıkması gerekmektedir.

Çıkması gereken matriksin hesabı şu şekilde yapılır:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

$$B.\bar{P^{i}} = P_p$$

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} . \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} \qquad \textbf{(3.69)}$$

Aşağıdaki matrise göre 2. veya 4. Sütunlarından bir tanesi çıkan matris olmalıdır.

$$P_p^{7} = \begin{bmatrix} 0 \\ 1/2 \\ 0 \\ 2/3 \end{bmatrix} \quad \textbf{(3.70)}$$

$$255{,}5 - 0.\theta \geq 0 \quad \textbf{(3.71)}$$

$$150{,}5 - \frac{1}{2}\theta \geq 0 \quad \textbf{(3.72)}$$

$$131{,}5 - 0.\theta \geq 0 \quad \textbf{(3.73)}$$

$$83{,}83 - \frac{2}{3}\theta \geq 0 \quad \textbf{(3.74)}$$

$$125{,}75 = \theta \quad \textbf{(3.75)}$$

Denklem 3.76' e göre göre θ değerinin hesaplanmasından dolayı tüm yerleşim planlarını içeren matristeki dördüncü sütun çıkan sütun olarak tespit edilir. Bu durumda ana modelin çözümünde kullanılacak tüm yerleşim planlarını gösteren matris şu şekildedir:

$$B = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \quad \textbf{(3.76)}$$

Bu matris yapısında 255,5+87,625+131,5+125,75=600,375 adettir.

Ana modele ait yukarıdaki matristen faydalanarak tekrar yeni dual değişlerler *y* hesaplanır.

$$yB = C_B$$

$$\begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix} . \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \quad \textbf{(3.77)}$$

Denklem 3.77' deki matrisin çözülmesiyle ana modele ait yeni y_i dual değişkenleri hesaplanır ve denklem 3.78' da değerleri görülmektedir.

$$y = \begin{bmatrix} 1/2 & 1/2 & 3/8 & 1/4 \end{bmatrix} \quad \textbf{(3.78)}$$

Ana modele ait dual değişkenlerin denklem 3.78 ile elde edilmesi ile alt model çözülerek ana modele dahil olabilecek yeni yerleşim planının olup olmadığı kontrol edilir.

$$9a + 8b + 7c + 6d \leq 20 \quad \textbf{(3.79)}$$

$$\frac{1}{2}a + \frac{1}{2}b + \frac{3}{8}c + \frac{1}{4}d > 1 \quad \textbf{(3.80)}$$

$$a, b, c, d \geq 0$$

Yukarıdaki denklem 3.79 ve 3.80 nolu denklemlerin sırt çantası problemine uygun olarak tekrar düzenlenmesiyle şu şekildedir ;

$$\max \quad \frac{1}{2}a + \frac{1}{2}b + \frac{3}{8}c + \frac{1}{4}d \qquad 9a + 8b + 7c + 6d \leq 20 \quad \textbf{(3.81)}$$

ukarıdaki değişkenlere ait katsayıların oranlanmasıyla problem yeniden yazılır;

$$x_2 = \frac{1}{18} \qquad x_1 = \frac{1}{16} \qquad x_3 = \frac{3}{56} \qquad x_4 = \frac{1}{24} \quad \textbf{(3.83)}$$

$$\max \quad \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{3}{8}x_3 + \frac{1}{4}x_4 \quad \textbf{(3.84)}$$

$$8x_1 + 9x_2 + 7x_3 + 6x_4 \leq 20 \quad \textbf{(3.85)}$$

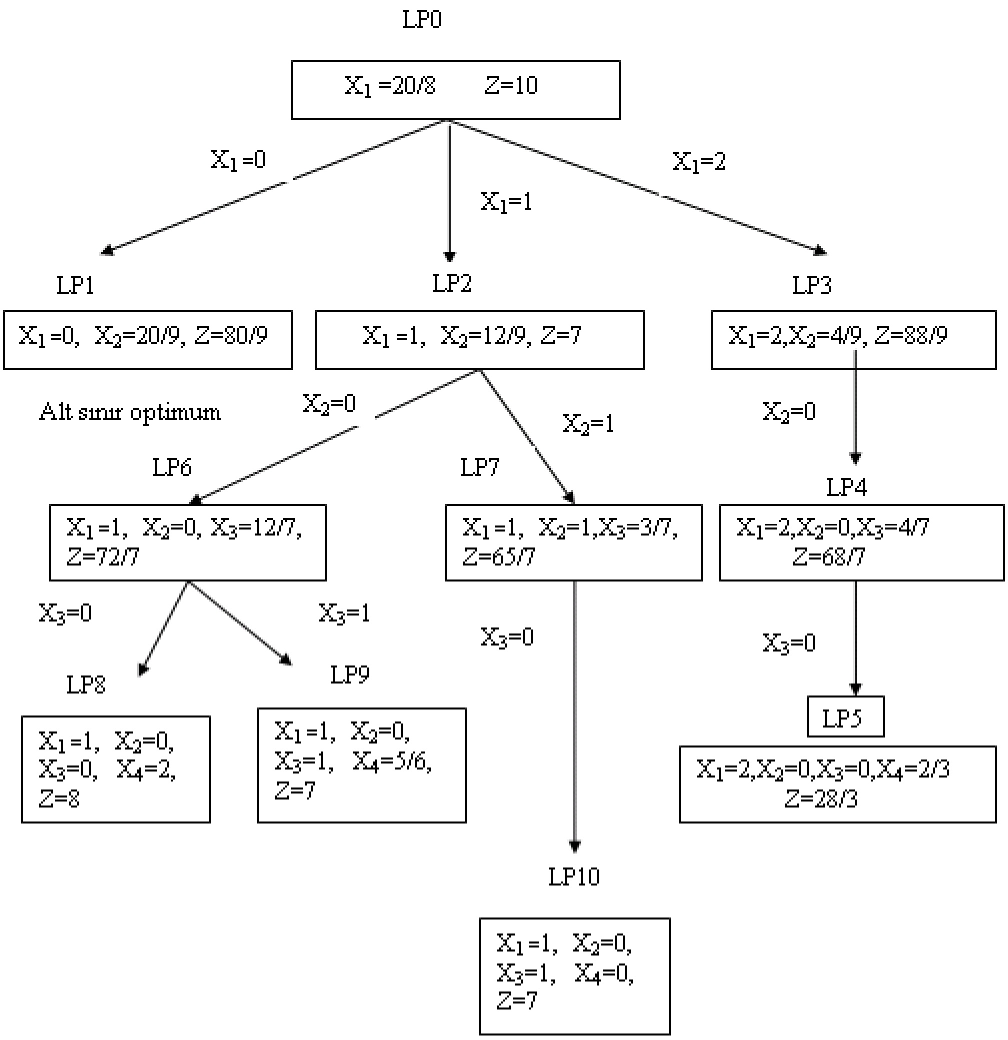
$$X_p \geq 0$$

Kesirli yapıdan kurtulmak için amaç fonksiyonu 6 ile çarpılır

$$\max \quad 4x_1 + 4x_2 + 3x_3 + 2x_4 \quad \textbf{(3.86)}$$

$$8x_1 + 9x_2 + 7x_3 + 6x_4 \leq 20 \quad \textbf{(3.87)}$$

Denklem 3.86 ve 3.87' nin kullanılmasıyla modelin dal-sınır yöntemi ile çözümü şekil 3.13' te görülmektedir



Şekil 3.13 : LPP örneğine ait üçüncü dal sınır çizelgesi

Şekil 3.13' deki dal sınır çözümleri incelendiğinde tam sayılarda elde edilen LP8 ve LP9 çözümlerinin sırt çantası probleminde 1 den daha büyük olmadığı ve yeni yerleşim planının ana modele dahil olması gerekmediği tespit edilmiştir.

Belirlenen 4 adet yerleşim planları, herbir sipariş taleplerini karşılamak ve kullanılan stok malzemesini en küçüklemek koşuluyla , tam sayı da olmayan aşağıdaki miktarlarda kullanılması gerekmektedir.

$$\begin{bmatrix}2\\0\\0\\0\end{bmatrix} = 255,5 \quad \begin{bmatrix}0\\2\\0\\0\end{bmatrix} = 87,625 \qquad \begin{bmatrix}0\\0\\2\\1\end{bmatrix} = 131,5 \quad \begin{bmatrix}0\\1\\0\\2\end{bmatrix} = 125,75$$

Doğrusal programlama çözümüne göre toplam kullanılması gereken malzeme miktarı X_p 255,5+87,625+131,5+125,75=600,375 dir.

Tam sayılı programlama çözümü açısından değerin ne olması gerektiği incelendiğinde;

- 601 adedin tam sayılı programlama çözümünün alt sınırını oluşturduğu,
- Her bir yerleşim planın kullanım sayısının yukarı değere yuvarlanarak ttam sayılı hale getirilmesi ile elde edilen 256+88+132+126= 602 adedin tam sayılı programlama çözümünün üst sınırını oluşturmaktadır.

601 adedi ve sütun oluşturma modelinin oluşturduğu yerleşim planlarını kullanarak sipariş taleplerini eksiksiz karşılamak mümkün değildir, bu noktada ya 602 adetlik çözüm kabul edilmelidir ya da eğer varsa mevcut yerleşim planları haricinde başka bir yerleşim planından faydalanılabilir. Son durum incelendiğinde sütun oluşturma yöntemi ile belirlenen yerleşim planlarının kullanım sayıları aşağı değere yuvarlanarak sipariş taleplerin ne kadar karşılandığı hesaplanır ve kalan sipariş talep miktarları tüm mevcut yerleşim planları değerlendirerek karşılanmaya çalışılır. Bu durumda 255+87+131+125=588 adet öncelikli olarak kullanılır ve 900 mm en siparişten bir adet, 800 mm en siparişten 2 adet, 700 mm en siparişten 1 adet, 600 mm en siparişten adet karşılanılamamış durumdadır.

Çizelge 3.12 : LPP örneğinde ilave olan yerleşim planları

Yerleşim Planı	Yerleşim Planı Kullanım Sayısı	Yerleşim Planındaki Kombinasyon
YerlesimP2	1	900+800
YerlesimP7	1	800+700
YerlesimP17	1	600+600

Çizelge 3.12' den görüleceği üzere yeni 3 adet yerleşim planlarının belirlenmesi ile birlikte toplam 601 adet stok malzemesi kullanılarak toplam sipariş ihtiyacı karşılanmaktadır.

İkinci çözüm,
Daha önce belirtildiği üzere bu çözümde GAMS paket programında yazılan algoritma ile çizelge 3.11' de belirtilen toplam 18 yerleşim planının hepsini birden değerledirerek çözüm elde edilecektir.

Tüm yerleşim planlarını kullanılması sağlayan GAMS programına ait model Ek A' da verilmiştir.:

Ek A daki GAMS modelinin çözümü sonrası elde edilen sonuçların özeti şu şekildedir:

Siparişlerin karşılanması sırasında 18 tane yerleşim planının hepsi kullanılmamıştır. Talep miktarlarını göz önünde bulundururak minimum fireye sebebiyet verebilecek olan yerleşim planları çözümde kullanılmıştır.

Toplam 601 adet 2000 mm eninde stok malzemesinden yararlanılarak çizelge 3.13' de belirtilen miktarlarda siparişler karşılanmıştır

Çizelge 3.13 : LPP örneğinde GAMS model1 sonuçları1

En(mm)	Talep Miktarı(adet)	Karşılanan Miktar(adet)
900	511	511
800	301	301
700	263	264
600	383	384

Ayrıca çizelge 3.14' de görüleceği üzere YerlesimP1, YerlesimP2, YerlesimP6, YerlesimP9, YerlesimP12 yerleşim planları çözümün elde edilmesinde kullanılmıştır.

Çizelge 3.14 : LPP örneğinde GAMS model1 sonuçları2

Yerleşim Planı	Yerleşim Planı Kullanım Sayısı	Yerleşim Planındaki Kombinasyon
YerlesimP1	255	900+900
YerlesimP2	1	900+800
YerlesimP6	87	800+800
YerlesimP9	126	800+600+600
YerlesimP12	132	700+700+600

Üçüncü çözüm,

Bu çözümde sütun oluşturma yöntemini kullanarak GAMS paket programında yazılan model ile en uygun sonuçlar ortaya konulacaktır.

GAMS paket programındaki yazılan model EK B' de modelin çözümünün sonuçları EK C' de verilmiştir.

Üçüncü çözümün sonuçları çizelge 3.15 ve çizelge 3.16' da görülmektedir. GAMS tarafından 602 adet stok malzemesinin kesilmesi gerektiği olarak bulunmuştur.

Çizelge 3.15 : LPP örneğinde GAMS model2 sonuçları1

En(mm)	Talep Miktarı(adet)	Karşılanan Miktar(adet)
900	511	512
800	301	302
700	263	264
600	383	384

Çizelge 3.16 : LPP örneğinde GAMS model2 sonuçları2

Yerleşim Planı	Yerleşim Planı Kullanım Sayısı	Yerleşim Planındaki Kombinasyon
YerlesimP1	256	900+900
YerlesimP6	88	800+800
YerlesimP9	126	800+600+600
YerlesimP12	132	700+700+600

Birinci çözümden farklı olmasının sebebi, sütun oluşturma yöntemi ile belirlenen yerleşim planlarının sadece kullanıldığı ve çıkan miktarların yukarı değere yuvarlanmasıdır.

3.3.2 İlk uygun azalan sezgisel metot

İlk uygun azalan sezgisel metodu şu şekilde uygulanmaktadır. Öncelikle en büyük endeki sipariş herhangi bir yerleşim planında siparişin karşılanması veya yerleşim planında yer kalmamasına kadar, bulunması amaçlanmaktadır. En geniş siparişin yerleşim planından bulunması mümkün veya uygun değilse bir sonraki en geniş sipariş yerleşim planına konulmaya çalışılır. Yerleşim planına herhangi bir siparişin eklenmesi uygun olmadığı durumda yerleşim planı kullanılmak üzere hazırdır ve stok malzemesinin miktarı ve siparişlerin talep miktarlarını geçmemesi göz önünde bulundurularak en fazla sayıda ilgili yerleşim planı tekrarlanır. Daha sonra sipariş miktarları ve stok malzemeleri güncellenerek önceki adımlar tekrar uygulanarak yeni yerleşim planları oluşturulur. İlk uygun azalan sezgisel metodu taleplerin karşılanması ile sonlanır.

İlk uygun azalan sezgisel metodun algoritmasını aşağıdaki şekildedir:

1. k $\longleftarrow$
2. while $I \neq 0$ do
   a. item $\longleftarrow$ *I listesinin ilk elemanı*

b. $\text{pattern}_p \leftarrow \phi; nap_p \leftarrow 0; rest \leftarrow L$

c. While (rest $\geq \min(l_i | i \in I)$) do

i. If (rest $\geq I_{item}$) then

1. $\text{pattern}_p \leftarrow \text{pattern}_p \cup (item)$

2. $rest \leftarrow rest - l_{item}$

ii. *Else*

1. *item* $\leftarrow$ *I listesinin sonraki elemanı*

d. $nap_p \leftarrow \min\{d_i | a_i | \forall i \in I\}$ $a_i p$ *yerleşim planındaki elemanının bulunma sayısı*

e. $I \leftarrow I - \{item\}$

f. $k \leftarrow k+1$

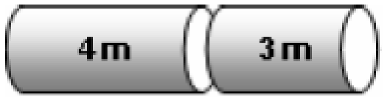
g. *Sipariş elemanlarının ihtiyaçlarını güncelle*

3. Yerleşim planı ve nap_k için çözüm kümelerini oluştur.

Yukarı da modeli bir boyutlu stok kesme probleminde bir örnek şu şekilde açıklanabilir;

- 7 metre boyunda stok malzemesi bulunmaktadır ve siparişler karşılanabilmesi için yeteri kadar stok malzemesinin olduğu varsayılmaktadır.
- Bu stok malzemesinden 4 metre, 3 metre ve 2 metre enlerinde müşteri siparişi karşılanması istenmektedir.
- 4 metre siparişten 89 adet; 3 mm siparişten 59 adet; 2 metre siparişten ise 92 adet talep edilmektedir.

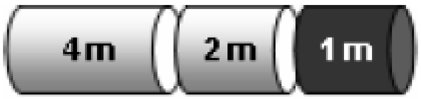
Öncelikle, siparişler büyükten küçüğe doğru sıralanır (4m > 3m > 2m). 7 metrelik stok malzemesine en büyük en olan 4 metre siparişi konulur, geriye kalan 3 metrelik kısma ise 3 metrelik sipariş yerleştirilerek ilk yerleşim planı elde edilir.



Şekil 3.14 : İlk uygun azalan sezgisel metodunda oluşan ilk yerleşim planı

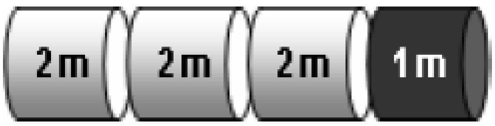
Bulunan yerleşim planı maksimum sayıda siparişlere uygulanmalıdır, 3 metre siparişten 59 adet istendiği için yerleşim planı 59 adetten fazla uygulanmaz. Toplam sipariş talepleri karşılanmadığı için taleplerin son durumu güncellenerek metodun uygulanmasına devam edilir. Son durumdaki taleple şu şekildedir; 30 adet 4 metre, 92 adet 2 metredir.

Bu noktadan hareket ile ikinci yerleşim planı oluşturulur. Bu yerleşim planı balına 1 metrelik toplamda ise 30 metrelik fire kaybı mevcuttur, ayrıca 4 metrenin kalan talebi 30 olduğu için bu yerleşim planının uygulanması sonucu sadece 2 metreden 60 adet kalmaktadır.



Şekil 3.15 : İlk uygun azalan sezgisel metodunda oluşan ikinci yerleşim planı

Üçüncü yerleşim planının uygulanmasıyla 2 metrenin talebi karşılanarak yerleşim planı başına 1 metrelik toplamda 21 metrelik fire kaybı oluşmaktadır.



Şekil 3.16 : İlk uygun azalan sezgisel metodunda oluşan üçüncü yerleşim planı

Sonuçta toplam 51 metrelik fire kaybı oluşarak ihtiyaçlar karşılanmıştır.

3.3.3 Ardışık sezgisel prosedür

Ardışık sezgisel prosedür yaklaşımında, stok malzemelerinden üretilmesi istenen küçük ebatlardaki tüm siparişlerin karşılanıncaya kadar yerleşim planlarının teker teker oluşturulmasıyla çözüm elde edilir. Ardışık sezgisel prosedür ile ilgili ilk çalışma Haessler tarafından gerçekleştirilmiştir (Haessler,1971). Ardışık sezgisel prosedürde seçilen yerleşim planları olabildiğince az fire kaybına sebebiyet vermesi, yüksek kullanım seviyesine sahip olması ve sonraki yerleşim planları için kombine olabilecek ebatlarda siparişleri bırakması beklenir. Bu yaklaşımın prosedürü şu şekildedir:

1. Öncelikle küçük ebatlardaki müşteri siparişleri analiz edilir ve problemin *tanımlayıcıları* belirlenir. Genel tanımlayıcılar, kesilecek stok malzemesi sayısı ve her bir stok malzemesinden karşılanacak ortalama sipariş miktarıdır.

2. İkinci aşamada bir sonraki kesim planının çözüme dahil olmasını belirleyecek *amaçlar* belirlenir. Maksimum kabul edilebilir fire kaybı, yerleşim planı ile kesilmesine izin verilen sipariş sayısı, yerleşim planı kullanımı ardışık sezgisel prosedürü kapsamında kullanılan amaçlardır.

3. Bu aşamada belirlenen amaçları gerçekleştirebilecek bir yerleşim planı araştırılır.

4. Amaçları karşılayan bir yerleşim planı bulunduğunda yerleşim planı en yüksek kullanımla çözüme eklenir. Bu ekleme sırasında karşılanması beklenen siparişlerin talep miktarlarının aşılmaması önem arz eder. İlgili yerleşim planı ile karşılan miktar kadar bu yerleşim planında bulunan siparişlerin talep miktarları azaltılarak birinci adıma geri dönülür.

5. Eğer herhangi bir kesim planı bulunamaz ise amaçların sayısal değerleri düşürülür. Genellikle amaçlardan yerleşim planı kullanımının azaltılması tercih edilerek daha fazla sayıda yerleşim planının çözüm kümesine girmesine izin verilir. Çözüm kümense yeni yerleşim planlarının girmesi sağlandıktan sonra üçüncü adıma geri dönülür.

Ardışık sezgisel prosedürün lineer programlamaya göre en belirgin avantajı, fire kaybının yanı sıra diğer faktörlerin de kontrol edilmesini sağlamasıdır. Örnek olarak, yüksek yerleşim planı kullanımının araştırılmasının yerleşim planlarının değişimin azaltılması verilebilir. Diğer bir yandan ardışık sezgisel prosedür yüksek fire kayıplarına yol açabilmektedir, bu prosedürün en belirgin dezavantajıdır. Ayrıca bu prosedür sadece tam sayılar ile çalışmasından dolayı yuvarlamalardan kaynaklanan problemleri elimine etmiş olur. en önemli avantajı yuvarlamaya gereksinim duyulmamasıdır.

3.3.4 Hibrit çözüm prosedürü

Stok kesme problemlerinin çözümünde kullanılan diğer bir yöntem, yukarıda belirtilen lineer programlama prosedürünün ve ardışık sezgisel prosedürün kombine edilmesidir.Örneğin, öncelikle ardışık sezgisel prosedür ile bir sonuç elde edilir ve bu sonuç lineer programlama prosedüründe başlangıç olarak kullanılır. İlave lineer programlama yinelemeleri(iterasyonları) yapılarak kesme kaybı minimize edilmeye çalışılır. Yinelemeler sonrası en iyi ardışık sezgisel prosedür çözümü ve yuvarlanmış lineer programlama çözümü seçilir.

Diğer bir uygulamada şu şekilde olabilir; optimum gölge maliyetleri elde etmek amacıyla problem ilk önce lineer programlama prosedürü ile çözülür ve daha sonra gölge maliyetler, yerleşim planlarının ardışık sezgisel prosedüründe kabul edilebilirliliğini test etmek için kullanılır. Bu yöntemde, ardışık sezgisel prosedürü sona yaklaştığında yerleşim planı seçimi kararı zor belirlenir hale gelir ve bu aşamada karşılanamayan sipariş miktarları lineer programla prosedürü ile çözülür.

Bu yaklaşımın en önemli avantajı, lineer programlama prosedürünü ve ardışık sezgisel prosedürünü birlikte içerisinde bulundurması ve iki metodun ayrı olarak verdiği çözümlerden daha iyi bir çözümü vermesidir. Bu yaklaşım genellikle yerleşim planlarının değişiminin minimum olmasının istendiği ve kesme kaybının(firenin) minimize edilmesinin istendiği problemler uygulanması literatürde karşılaşılan bir uygulamadır. (Haessler, 1988)

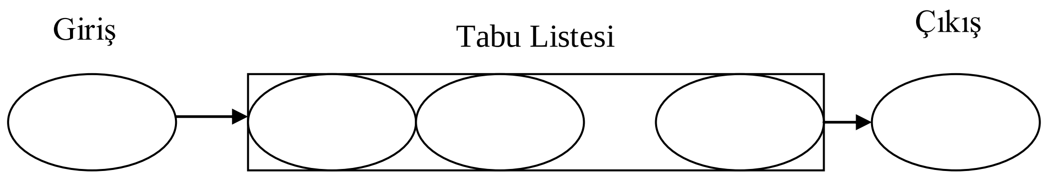
3.4 Metasezgisel Çözüm Metotları

Sezgisel metotların sahip olduğu lokal araştırma ve yoruma dayanan yapısından dolayı lokal çözümler bazı durumlarda yeterli görülmemektedir. Bu noktadan hareket ile temel sezgisel metotları alt yapısı olarak kabul ederek lokal optimal çözümlerden kaçınmayı amaç edinen ve metasezgisel diye adlandırılan yeni metotlar literatüre kazandırılmıştır. Metasezgisel metotlar üzerinde önemli gelişmeler katedilerek çok sayıda algoritmalar sunulmuştur. Metasezgisel metotlarının en önemlileri sırasıyla şu şekildedir; tabu araştırması, benzetilmiş tavlama algoritması, karınca kolonisi algoritması, açgözlü rassallaştırılmış uyarlamalı arama yordamı(ARUAY).

3.4.1 Tabu araştırması

Tabu araştırması temel olarak belirli bir problem üzerine elde edilen ilk çözüm etrafındaki komşuluklar oluşturmaktır. Komşuluk mekanizması ele alınarak birbirini izleyen seçilmiş çözümlerden daha iyi olanı tabu listesi olarak atanır. Tabu arama algoritmasında tabu listesi olarak oluşturulan ilk aday çözüm ve değişken komşu çözüm sayısı, bir tür tabulaştırma görevi yapmaktadır. Kötü sonuç veren bölgelerde daha fazla işlem yapılmaması (bu elemanların komşu sayılarının azaltılması), istenen çözüme daha az hesaplamayla, dolayısıyla daha hızlı ulaşmayla sağlamaktadır. iyi sonuç veren parametrelerin bir sonraki iterasyonda komşu sayıları artmakta, böylece algoritmanın verimliliği de artmış olmaktadır. Tabu listesinin en önemli özelliklerinden birisi, mevcut tabu listesinin aday komşu çözümler ile karşılaştırıldıktan sonra bir sıralama ve karşılaştırma işlemi yaparak kendisini yenileyebilmesidir. Bu amaçla, geliştirilen algoritmada, önceki döngülerde elde edilen çözümlerin listesinin tutulduğu bir tabu listesi oluşturulmuştur. Eğer bir komşu çözüm adayı, tabu listesinde yer alan bir çözümle aynıysa (bu durumda aday çözüm, zaten daha önce denenmiştir), bu çözüm değerlendirme dışı bırakılmaktadır. Tabu listesi oluşturulurken her döngüdeki en iyi çözüm listeye alınmakta, listenin dolduğu durumda listedeki ilk kayıtlar (başlangıçtaki çözümler) listeden atılıp, son döngüler de elde edilen çözümler listeye alınmaktadır.

Tabu Listesi ilk en iyi çözüm kümesinin oluşturularak hafızaya alınma yöntemi ile oluşturulur. Tabu listesi oluşturmanın önemli bir kuralı da giriş değerleri oluşturulurken çeşitli filtrezizasyon işlemlerinden geçirilmesidir (Mori, 1998).



Şekil 3.17 : Tabu listesi oluşturma prensibi

Oluşturulan Tabu listesinin en temel özelliği yeni çözüm adaylarını değerlendirmeye alarak dairesel bir döngü içerisinde yol alarak her yeni döngüden sonra yeni özellikler kazanmasıdır. Şekil 3.17' de genel olarak tabu listesi oluşturmanın genel prensibi görülmektedir. Tabu listesinin yeni elde edilen çözümler ile liste uzunluğu artacak ve çözüm aramada büyük bir etki yaparak kısa çözümlerin daireselliği sayesinde nesnel bir çözüm arama tekniği olduğunu gösterecektir. Burada istenilen durum çözüm adaylarının mevcut bu döngüsü ile Tabu listesinin geliştirilmesidir. Tabu listesine dahil edilen yeni çözümler oluşturulurken eğer elde edilen çözüm tabu listesindeki çözümlerden daha iyi ise elde edilen çözüm tabu listesine eklenebilir. Bunun tersi durumunda, yani daha kötü bir çözüm ise tabu listesine eklenmeyecek ve doğal olarak belirli bir bellek kaplamayacaktır. En iyi çözüm bulunana kadar bu işlemler mevcut döngü ile devam edecektir (Mori, 1998).

3.4.2 Karınca kolonisi algoritması

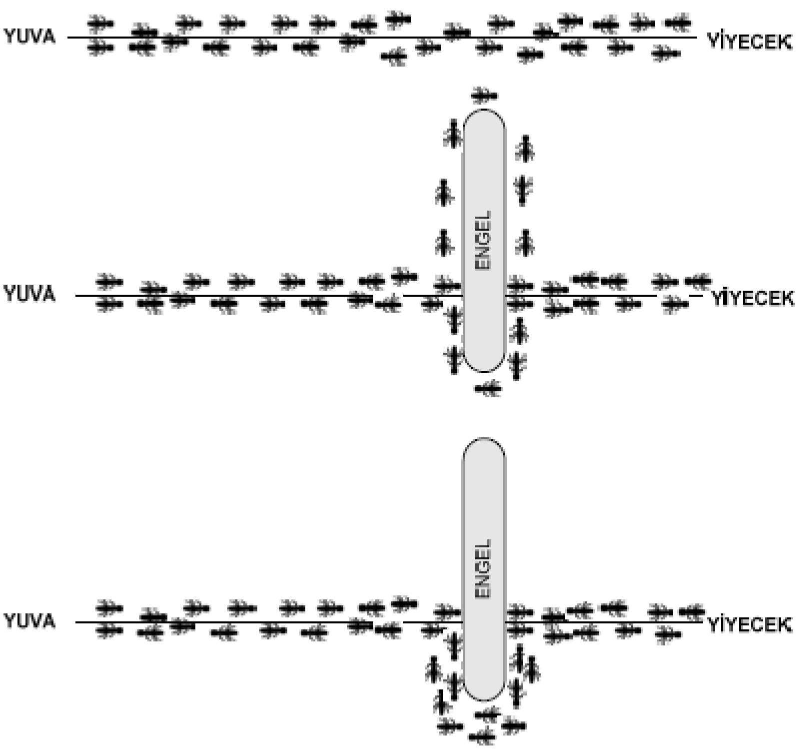
Karınca kolonisi algoritması (KKA) NP-zor optimizasyon problemlerine başarılı bir şekilde uygulanabilen popülasyon temelli bir yaklaşımdır. İlk olarak Gezgin Satıcı Problemi üzerinde uygulanmıştır. Bir noktadan başlayarak ve tüm noktalara bir kez uğrayarak tekrar aynı noktaya en kısa yoldan dönmek diye tanımlanan GSP'nin çözümünde KKA'nın oldukça etkili olduğu görülmüştür. KKA gerçek karınca kolonilerinin davranışlarından esinlenilerek geliştirilmiştir.

Tekniğin en temel unsurlarından biri haberleşme aracı olarak kullanılan ve problemlerde çözümün kalitesini gösteren feromon kimyasalıdır. Feromon, gerçek karıncaların da bir haberleşme ve yön bulma aracı olarak kullandıkları, vücutlarından salgıladıkları kimyasaldır. Feromon izleri, karıncalar tarafından güncellenmekte ve bir bilgiyi temsil etmektedir. Bir yolda feromon izinin yoğun olması, yolun kalitesini

gösterir ve tercih olasılığını arttırır. KKA'da, yapay karıncalar, gerçek mesafeler dikkate alınarak yapılmış olan model üzerinde en kısa yolu araştırmaktadırlar. Yollardaki feromon izleri yine yapay olarak, karıncaların geçiş sıklığıyla orantılı bir şekilde güncellenmektedir.

KKA' sı, gerçek karıncaların yuvaları ile yiyecek noktaları arasındaki en kısa yolu bulma kabiliyetlerinden esinlenilerek geliştirilmiştir. Alternatif yolların söz konusu olduğu durumlarda karıncalar, öncelikle bu yollara eşit olasılıkla dağılırlarken belli bir süre sonra en kısa olan yolda yoğunlaşmaktadır.

Şekil 3.18' de zaman geçtikçe tüm karıncaların en kısa olan yolu kullandıkları görülmektedir. Bunu yaparken önceki geçişlerden yollarda kalan feromon izlerinden faydalanmaktadırlar. Temel kural, feromon miktarının yoğun olduğu yolun tercih edilme olasılığının da yüksek olmasıdır.



Şekil 3.18 : Gerçek karıncaların en kısa yolu bulma aşamaları (Mori, 1998)

Görme duyuları çok gelişmemiş olan karıncalar yol tercihlerini feromon izlerine göre yapmaktadırlar. Kısa olan yolda feromon miktarı uzun yollara nispeten daha fazla birikmektedir. Kısa olan yoldan geçiş daha hızlı gerçekleşeceğinden, birim zamandageçiş yapan karınca sayısı uzun yola göre daha fazla olacaktır. Dolayısıyla herhangi iki düğüm arasındaki yol üzerinde bulunan feromon miktarı, yolun uzunluğuyla ters orantılıdır.

3.4.3 Benzetilmiş tavlama algoritması

Benzetilmiş tavlama(BT) algoritması, fiziksel tavlama kavramının ilk olarak kombinasyonel eniyileme problemlerinde kullanıldığı 1980'li yıllarda, bir eniyileme aracı olarak sunulmuştur. Pek çok yerel en küçük değere sahip doğrusal olmayan fonksiyonların en iyi değerlerinin bulunması için tasarlanmıştır. Bu algoritma, elektronik devre tasarımı, görüntü işleme, yol bulma problemleri, seyahat problemleri, kesme ve paketleme problemleri, akış ve iş çizelgeleme problemleri için başarılı sonuçlar vermiştir (Lai, 1992).BT algoritması, dikdörtgen parçalar içeren iki boyutlu kesme ve paketleme problemlerinin çözümü için çok az sayıda araştırmacı tarafından kullanılmıştır.

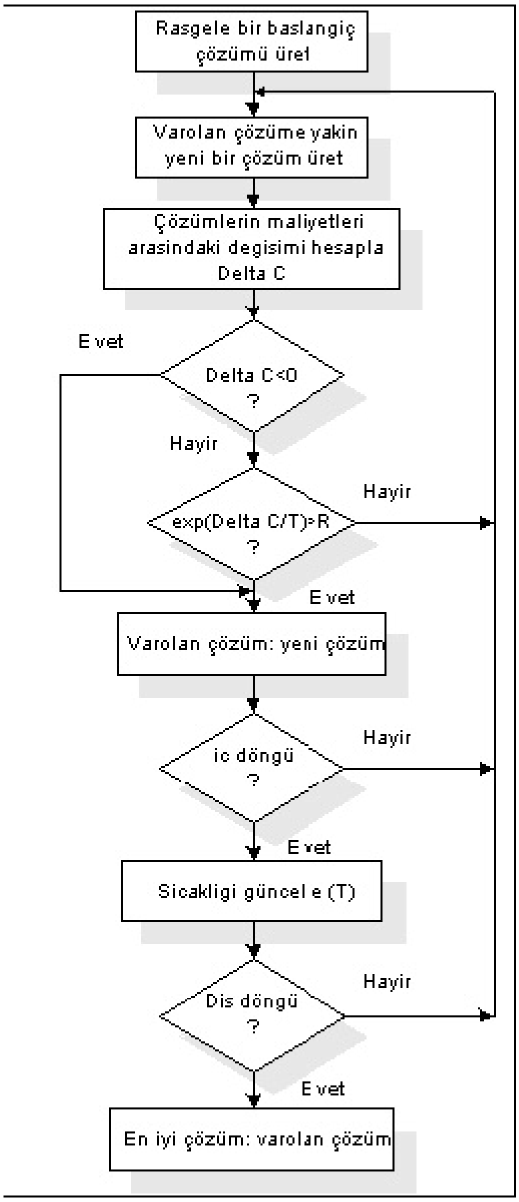
Kämpke (1988), BT algoritmasını kullanarak paketleme problemleri üzerine çalışan ilk araştırmacılardan biridir. BT algoritmasının farklı soğutma çizelgelerinin kullanımıyla bir boyutlu kutu paketleme problemlerine uygulamıştır. BT algoritmasını eş ve eş olmayan kutuları içeren yükleme problemleri üzerinde çalışmıştır (Dowsland, 1993). Dowsland(1993) çalışmasında herhangi bir yerleştirme algoritması kullanmamıştır. Giyotinli ve giyotinsiz kesme problemleri için iki ayrı yerleştirme algoritması kullanarak melez bir BT algoritması geliştirmiştir (Faina, 1999). Giyotinsiz kesme problemleri için kullandığı yerleştirme algoritması, giyotinli kesme problemleri için kullandığı allgoritmadan daha iyi sonuçlar vermiştir. Leung permütasyona dayalı bir BT algoritmasıyla iki boyutlu dikdörtgen biçimindeki parçaların giyotinsiz kesim gerektiren yerleşimleri üzerine çalışmıştır.

BT algoritması, tarafından eniyileme problemlerinin çözümü için geliştirilmiş bir yerel arama algoritmasıdır. Katı bir maddenin enerji durumunu en aza indiren fiziksel sistemlerdeki tavlama süreci ile kombinasyonel eniyileme problemlerindeki çözüm süreci arasındaki benzerlik üzerine kurulmuştur.

Tavlama işlemi iki aşamada gerçekleşmektedir; birinci aşama istenen sıcaklığa kadar ısıtma, ikinci aşama belirli bir sıcaklıkta tutma ve soğutmadır (Masri, 1999). Tavlama işleminin ilk aşaması olan ısıtma işleminde metal istenen sıcaklığa kadar ısıtılır. Metal parçacıklar yüksek sıcaklıklarda oldukça yüksek enerjiye ve serbestliğe sahiptir. Sıcaklığın yavaş yavaş düşürülmesiyle gerçekleştirilen soğutma işlemi sırasında parçacıkların enerjileri azalır. Her yeni enerji duruma uygun olarak yeni bir dengeye girmek üzere kendilerini tekrar düzenlerler. Soğutma işlemi genellikle oda sıcaklığına kadar olur. Bu işlemlerde zaman önemli bir parametredir. Soğutma çok hızlı olursa kristal yapı içerisinde düzensizlikler ve bozulmalar görülür. Metal parçacıkları, enerjinin en düşük olduğu duruma ulaşamaz ve enerjinin en yüksek olduğu kristalimsi durumda soğutma işlemi sona erer. Bu da metalde şekil bozukluğuna hatta çatlamalara neden olur. Gerçek tavlama süresi, gerekli deği,im hareketleri için yeteri derecede uzun olmalıdır. Mükemmel bir kristal yapının elde edilmesi ancak bu süre içinde metalin yavaş soğutulması ile gerçekleştirilir. Ayrıca tavlama sıcaklıüı da bu süreçte önemli bir konudur.

Sıcaklık arttırılarak tavlama işlemi hızlandırılabilir. BT algoritması eniyileme problemlerinin çözümü için son zamanlarda oldukça sık başvurulan bir yerel arama algoritmasıdır.

Fiziksel sistemlerde gerçekleştirilen tavlama işleminin, kombinasyonel eniyileme problemleri için model olarak kullanılmasıyla ortaya çıkmıştır. BT algoritması, pek çok değişkene sahip fonksiyonların en büyük veya en küçük değerlerinin bulunması ve özellikle pek çok yerel en küçük değere sahip doğrusal olmayan fonksiyonların en küçük değerlerinin bulunması için tasarlanmıştır. BT algoritması ile bir eniyileme probleminin çözümünü gösteren akış şekil 3.19' da verilmiştir.



Şekil 3.19 : BT algoritmasının akış diyagramı(Shahookar ve diğ, 1991)

3.4.4 Açgözlü rassallaştırılmış uyarlamalı arama yordamı

Açgözlü rassallaştırılmış uyarlamalı arama yordamı(ARUAY) kombinatoryal eniyileme problemleri için kullanılan bir metasezgisel bir metod olup, her bir tekrarda probleme ilişkin bir uygun çözümün hesaplandığı, arama uzayının tekrarlı rassal örneklenmesine dayalıdır. Her bir ARUAY tekrarlaması (iterasyonu) iki ana aşamadan oluşmaktadır: Rassallaştırılmış bir sezgisel temelindeki oluşturma aşaması ve ilk aşamada olusturulan başlangıç çözümü iyileştiren yerel arama aşaması'dır. Tekrarlama süreci belirtilen bir tekrar sayısına ulaşıncaya kadar sürdürülür ve o zamana kadar elde edilen en iyi çözüm son çözüm olarak değerlendirilir. ARUAY ilk kez tarafından Feo ve Bard tarafından tanıtılmıstır. (Feo ve Bard, 1989) ve uygulamaları için daha detaylı bilgi tarafından sunulmaktadır (Resende ve Riberio, 2002).

Karışık-model sıralama problemi icin bu çalışmada önerilen rota birleştirmeli (path relinking) ARUAY algoritmasının temel adımları asağıda gosterilmektedir.

Rota Birleştirmeli ARUA yordamının adımları:

0: Girdi veriyi oku.

1: j←1, i←1

2: i ≤ EnbTekrar olduğu sürece, adım 3'e git, aksi durumda adım 9'a git

3: Açgözlü Rassallastırılmıs Olusturma

4: Yerel Arama

5: Eğer i=j*TekrarSayısı ise Rota Yenileme gerçeklestir ve j'yi 1 arttır.

6: En iyi çözümü güncelle

7: i'yi 1 arttır

8: Adım 2'ye git

9: Mevcut en iyi çözümü göster ve Dur.

ARUAY, verilen bir en büyük tekrar sayısına (EnbTekrar) erişilinceye kadar tekrar döngüsünü sürdürmektedir. ARUAY' nin her bir tekrarında, açgözlü rassallaştırılmış oluşturma ile adım 3 bir uygun çözüm oluşturulmakta, geçerli çözümün önceden belirlenmiş komşuluğu dikkate alınarak yerel en iyiye erişilinceye kadar yerel arama gerçekleştirilmekte adım 4, önceden belirlenmis sayıdaki tekrar aralıklarında (TekrarSayısı) rota yenileme gerçekleştirilmekte adım 5, ve adım 6 da olası bir çözüm güncellemesi gerçekleştirilmektedir.

3.4.4.1 Oluşturma aşaması

Çözümler, uygunluğu daima sağlayacak şekilde, açgözlü rassallaştırılmış yordam yardımıyla oluşturulmaktadır. Aday işler kümesi, henüz sıralanmamış ve olası bir sırada bulunabilecek tum işleri kapsamaktadır - burada, başlangıçta, sıralanacak olası işlerin toplam sayısının toplam talebe eşit olduğunu not etmek gerekmektedir. Bir çözüm oluşturmak icin ilk iş aday işler icinden rassal olarak seçilir ve bu iş sıraya konur. Bu tek işlik sıra "geçerli sıra" olarak kabul edilir. Sıranın sonraki isleri açgözlü yordam yardımıyla seçilmektedir. Bu yordama gore, aday islerin her biri icin geçerli sıra ile oluşturulabilecek farklı sıralar, uygunluğu da dikkate alarak, ileriye kaydırma metodu yardımıyla türetilmektedir. Bu metod rassal baslangıç sıra oluşturmaya kıyasla daha fazla süre gerektirmekte ancak daha iyi baslangıç çözümler oluşurmaya imkan sağlamaktadır. Kısaca metod şu şekilde çalışmaktadır: Geçerli sıranın 'ABC' olduğunu ve aday işin de D olduğunu varsayalım. D işi ile, ileriye kaydırma metodu yardımıyla, oluşturulabilecek alternatif sıralar şunlardır; "DABC", ADBC", "ABDC" ve "ABCD" - Burada D işinin, geçerli sıra dikkate alındığında, soldan sağa (ileriye) doğru sırasıyla kaydırıldığını not etmek gerekmektedir. Bu sıralardan uygun olmayan -gerekli hazırlık sayısından daha fazla sayıda hazırlık içeren (henüz tüum aday işler sıralanmamışken gereğinden daha az sayıda hazırlık içeren sıralar uygun olarak kabul edilmektedir) - sıralar elenmekte ve her bir uygun sıra için denklem 2 yardımıyla kullanım oranları hesaplanmaktadır. Tüm aday işler icin uygun alternatif sıralar türetildikten sonra, bu aday işler ve ilgili uygun sıralarından "Sınırlandırılmış Aday Listesi (SAL)" (Restricted Candidate List-RCL) oluşturulmaktadır. Amaç, kullanım oranını enkücüklemek olduğundan, tüm alternatif sıralardan en küçük kullanım oranına sahip sıra ile ilişkili iş aslında sıralanacak bir sonraki en iyi iştir, ancak bu sekilde bir seçim yapısı tam bir açgözlü tip sezgiseli doğurmaktadır. Bunun yerine, çalısmada, en iyi kullanım oranlarına sahip sıralarla ilişkili işlerden bir sınırlandırılmış aday listesi oluşturulmaktadır. $Uenk$ ve $Uenb$' nun açgözlü yordam yardımıyla hesaplanan en küçük ve en büyük kullanım oranları olduğu varsayılsın. Sınırlandırılmış aday listesi içinde yer alacak iş sayısı, $\alpha \in [0,1]$ olmak uzere bir α parametresi yardımıyla sınırlandırılabilmektedir: α değerine bağlı olarak, kullanım oranları, $Uenk + \alpha(Uenb - Uenk)$Uenk değerine eşit ya da küçük olan sıralara ilişkin tüm aday işler sınırlandırılmış aday listesi içinde yer almaktadır. α değeri 1'e eşit olduğunda tüm aday işler sınırlandırılmış aday listesine dahil edilir ki bu rassallaştırılmış oluşumla sonuçlanmaktadır. Diğer yandan, α değerinin 0 olması halinde ise, sınırlandırılmış aday listesine içinde en küçük kullanım oranlı sıra ile ilişkili iş bulunur. Bu durum da tam açgözlü oluşumla sonuçlanmaktadır. Sınırlandırılmış aday listesi olusturulduktan sonra, sıralanacak sonraki iş, kullanım oranlarının dikkate alınması ile sınırlandırılmış aday listesi içinden rassal olarak seçilmektedir. Seçim süreci, aynı zamanda, seçilen işle ilişkili sırayı da belirlediğinden, bu sıra gecerli is sırası olarak kabul edilir ve aday isler kümesi güncellenir. Bu yordam aday işler kümesinde hiçbir iş kalmayıncaya yani bir uygun cozumun oluşturulmasına kadar devam ettirilmektedir.

Oluşturma Aşamasının Adımları

0: İlk işi rassal olarak seç ve sıraya koy. Bu sırayı geçerli sıra olarak kabul et.

1: Aday işler kümesini güncelle.

2: Aday işler (henüz sıralanmamış olanlar) için olası uygun sıraları ileriye kaydırma metodu ile türet.

3: Her bir aday iş için türetilen uygun sıraların kullanım oranlarını hesapla

4: Sınırlandırılmıs aday listesini türet

5: Sınırlandırılmıs aday listesinden rassal olarak bir iş ve ilişkili sırasını seç. Bu sırayı geçerli sıra olarak kabul et.

6: Eğer bir çözüm tamamlandı ise, Adım 7'ye git, aksi halde (henüz tüm isler sıralanmadı ise) Adım 1'e git

7. Oluşturulan çözümü geri döndür ve dur.

Oluşum asaması icin α parametresi buyuk onem tasımaktadır. α değerinin 0-1 arasında sabit bir değer olarak belirlenmesinin yerine dinamik olarak değistirilmesinin daha iyi olacağını raporlamaktadırlar. Bu nedenle, α değerleri, bu çalışmada, bir kesikli düzgün olasılık dağılımı yardımıyla seçilmektedir. Olasılık dağılımına gore α'nın alabileceği olası değerler; 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0'dır.

3.4.4.2 Yerel arama aşaması

Arama rassallaştırılmış oluşturmayla, küçük komşuluklar için bile, sıklıkla en iyi çözümlerr türetilememektedir. Yerel arama aşaması, oluşturulan başlangıç çözümde genellikle iyileşme sağlamaktadır. Bir yerel arama algoritması komşulukta bulunan daha iyi bir çözümü geçerli çözümle yer değiştirecek şekilde tekrarlamalı olarak çalışmaktadır. Bu çalışmada uygulanan yerel arama algoritması, değisken komşuluk arama ve değisken komşuluk azalması sezgisellerinin bir varyantı olup (Gupta ve Smith , 2006) tarafından önerilmektedir. İlgili çalışmalarında, üç farklı arama komşuluğu tanımlayarak bunları değişken komşuluk arama ve değişken komşuluk azalması sezgiselleriyle birleştirmişlerdir. Bu arama komşulukları asağıda tariflenmektedir. İkili değisim komşuluğu. Bir geçerli sıradaki tüm olası ikili değişimler gerçekleştirilmektedir. Örneğin, geçerli sıranın "ABC" olduğu varsayımıyla, sırasıyla A ve B, A ve C,…,C ve A, C ve B ikili değişimleri gerceklestirilir. Bu değişimler sonucu türetilen sıralardan uygun olmayanlar elimine edilir. Herhangi bir ikili değişim sonucu bir iyileşme sağlanırsa, geçerli sıra güncellenir ve işlemlere baştan tekrar başlanır. Süreç hiçbir ikili değisim sonucu iyileşme sağlanamadığında durdurulur ve geçerli sıra yerel en iyi olarak kabul edilir. *İleriye kaydırma komşuluğu*. Bir geçerli sıradaki her bir iş için olası tüm ileriye doğru kaydırmalar yerine getirilmektedir. Örnek olarak, geçerli sıranın "ABC" olduğu varsayılsın. Kaydırmalara en soldaki iş, A'dan baslanır ve sırasıyla diğer tüm işler için ileriye doğru tekrarlanır. Boylece, A'nın kaydırılması ile, "BAC" ve "BCA" sıraları, B'nin kaydırılması ile de, "ACB" sırası elde edilir. Herhangi bir kaydırma sonucu bir iyileşme sağlanırsa, geçerli sıra güncellenir ve işlemlere baştan tekrar başlanır. Sureç hiçbir kaydırma sonucu iyileşme sağlanamadığında durdurulur ve geçerli sıra yerel en iyi olarak kabul edilir. Geriye Kaydırma Komsuluğu. Bir geçerli

sıradaki her bir iş için olası tüm geriye doğru kaydırmalar yerine getirilmektedir. Örnek olarak, geçerli sıranın "ABC" olduğu varsayılsın. Kaydırmalara en sağdaki iş, C'den başlanır ve sırasıyla diğer tüm işler icin geriye doğru tekrarlanır. Böylece, C'nin kaydırılması ile, "ACB" ve "CAB" sıraları, B'nin kaydırılması ile de, "BAC" sırası elde edilir. Herhangi bir kaydırma sonucu bir iyileşme sağlanırsa, geçerli sıra güncellenir ve işlemlere baştan tekrar baslanır. Sürec hiçbir kaydırma sonucu iyileşme sağlanamadığında durdurulur ve geçerli sıra yerel en iyi olarak kabul edilir.

Bu komşuluklar temelinde yerel arama asağıda tanımlanmaktadır.

Yerel Arama Aşamasının Adımları

0: Geçerli çözümün kullanım oranı değerini KulORN'a ata. Bir yerel en iyi elde edilene kadar bu çözüm üzerinde *ikili değisim komşuluğu* araması uygula. Yerel en iyiyi gecerli çözümolarak kabul et.

1: Bir yerel en iyi elde edilene kadar gecerli çözümuzerinde *geriye kaydırma komşuluğu* araması uygula. Yerel en iyiyi geçerli çözüm olarak kabul et.

2: Bir yerel en iyi elde edilene kadar geçerli çözümuzerinde *ileriye kaydırma komşuluğu* araması uygula. Yerel en iyiyi geçerli çözüm olarak kabul et.

3: Geçerli çözümün kullanım oranı değeri KulORN'dan daha iyi ise Adım 0'a git, aksi halde dur.

3.4.4.3 Rota birleştirme

Rota birleştirme, temel ARUA yordamının iyileştirilmesi için geliştirilen bir başka yaklaşım olup genellikle çözüm kalitesini arttırmaktadır. Rota birleştirme, bir başlangıç çözüm ve bir klavuz çözümü birbirine bağlayan yörüngelerin araştırılmasını içermektedir. Klavuz çözüm, genellikle, kalitesi yüksek bir çözüm dür. Hareketler başlangıç çözümden başlayarak klavuz çözüme erişilecek şekilde, herhangi bir komşuluk araması (ikili değisim, ileriye veya geriye kaydırma komşulukları) temelinde gerçekleştirilmektedir. Bu hareketler sırasında uygunluk daima korunmak zorundadır. Süreç, başlangıç çözümden klavuz çözüm elde edilmesine kadar sürdürülmektedir.

Klavuz çözüm kümesinden rassal olarak secilen klavuz çözüm"ABCC" ve başlangıç çözüm de "CBAC" olsun. İleriye kaydırma komsuluğu temelinde başlangıç çözümden hareketle elde edilen sıralar şunlardır.

Başlangıç çözüm "CBAC"

C ileriye kaydırılır "BCAC" (Eldeki en iyiden daha iyi mi kontrol et)

C ileriye kaydırılır "BACC" (Eldeki en iyiden daha iyi mi kontrol et)

B ileriye kaydırılır "ABCC" (Klavuz çözüm dur.)

Yukarı kısımlarda ele alınan sezgisel-metasezgisel metotların avantaj ve dezavantajlarının kıyaslanması çizelge 3.17 de yapılmıştır.

Çizelge 3.17 : Sezgisel-metasezgisel metotların karşılaştırılması

Method	**Avantaj**	**Dezavantaj**
Lineer Programala Prosedürü	Hızlı, sadece optimal çözüme ulaştıraca yerleşme planları oluşturulur	Kesirli sonuçlar üretebilir ve tek amacı kesme kaybı minimizasyonudur.
Ardışık Sezgisel Prosedür	Hızlı, tamsayılı çözümler üretir. Birden çok amacın göz önünde bulundurulmasını sağlar	Özellikle zor problemlerde artan kesme kayıplarınayol açan çözümler üretir
Hibrid	Tamsayılı çözümler üretir ve temel sezgisel metotların iyi kısımlarını özümünde ele alır.	Prosedüründeki artan aşama sayısından dolayı temel sezgisel metotlardan daha yavaştır.
Tabu Araştırması	Açgözlü rassallaştırılmış uyarlamalı arama yordamı ile karşılaştırıldığında daha kaliteli sonuçlar üretir.	Açgözlü rassallaştırılmış uyarlamalı arama yordamından daha yavaştır
Açgözlü Rassallaştırılmış Uyarlamalı Arama Yordamı	Tabu Araştırmasına göre çok hızlı olup kabul edilebilir sonuçlar üretir.	Tabu araştırmasına göre daha kesin olmayan sonuçlar üretir.
Genetik Algoritma	Göreceli olarak basittir	Karınca kolonisi algoritmasına göre daha yavaştır.
Karınca Kolonisi Algoritması	Problemlere etkin ve kesin çözümler sunar	Göreceli olarak komplikedir ve kısıtların artmasıyla birlikte verimsiz bir metot haline gelebilir.

4. UYGULAMA

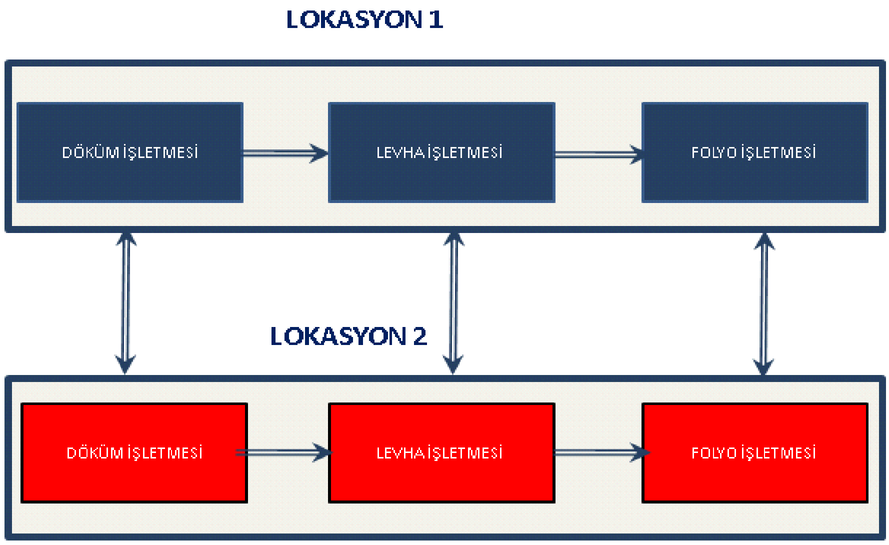
Uygulama kapsamında, önceki bölümlerde detaylı olarak anlatılan stok kesme problemlerinden tek boyutlu değişik endeki stok malzemelerinin kesimini içeren stok kesme problemi ele alınacak ve sezgisel ve sütun oluşturma yöntemleri ile çözümler sunulacaktır. Bu uygulama, ülkemizde yassı alüminyum sektöründe faaliyet gösteren X firmasının verilerinden ve iş yapış şeklinden yola çıkılarak gerçekleştirilmiştir. Şirket politikası gereğince şirket ismi gizli tutulmuştur.

Öncelikli olarak yassı alüminyum sektöründe faaliyet gösteren X firmasından ve sektöre özgü olan üretim tipinden kısaca bahsedilecektir. X firmasının birbirine yakın konumda olan iki farklı lokasyonu bulunmakta olup bazı ürün tipleri hariç birbirine yakın tipte ürünler üretilebilmektedir.

Her iki lokasyonda da şekil 4.1‘ den görüleceği üzere döküm işletmesi, levha işletmesi ve folyo İşletmeleri mevcuttur. Levha ve folyo işletmeleri bulunduğu lokasyonda veya diğer lokasyondaki kendinden önceki işletmenin müşterisi durumundadır ve hammaddesini bu işletmeden karşılar. Dölüm işletmesinin hammaddesi ise yurt dışından tedarik edilen alüminyum ingot, billet ve külçelerdir. Bu durumdan anlaşılabileceği üzere X firması birincil alüminyum üretici olmayıp tedarik ettiği külçe, billet ve ingotları dökme, haddeleme, tavlama, germe, dilme, boy kesme işlemlerine tabi tutarak nihai levha ve folyo ürünleri üretmektedir.

Döküm işletmesi, alüminyum külçe, billet ve ingotları döküm tezgahlarında ergiterek ve magnezyum, silisyum vb. elementlerin katıımıyla istenen alaşımda dökme ruloların üretimi yapılır.

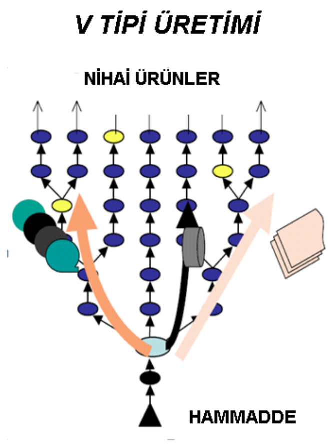
Levha işletmesi, yassı levha ürünlerini ve folyo ürünlerinin hammaddesi olan foilstok ürünlerin üretimini yapar. Döküm işletmesinden gelen dökme rulolar, levha işletmesinde bulunan haddeleme, tavlama tezgahlarının yardımıyla ile nihai yassı levha ürünün istenen kalınlıkta, sertlikte olması sağlanır. Haddeleme ve tavlama işlemlerinden sonra gerdirme, dilme ve boy kesme tezgahlarının yardımıyla nihai ürünün istenen ebatlara gelmesi sağlanır.



Şekil 4.1 : X firmasınına ait işletme çalışma biçimi.

Folyo işletmesi levha işletmene benzer olarak, haddeleme, tavlama, dilme tezgahlarına sahip olup bu tezgahlarda nihai yassı folyo ürünün elde edilmesi için malzemenin iç yapısını değiştiren ve boyutlandıran işlemleri gerçekleştirir.

X firmasının üretim tipi incelendiğinde yassı metal sektöründe yaygın olarak uygulanan şekil de görülen V tipi üretim ile karşılaşılmıştır. Yani iç yapısı özellikleri değiştirilen bir ana malzemeye hiçbir montaj yapmadan bu ana malzemede dilme, bölme, kesme işlemleri ile daha küçük malzemeler oluşturulmaktadır.



Şekil 4.2 : X firmasınının üretimi akış tipi

X firmasının toplam satışlarının %65’ i ihraç ürünlerini kapsamaktadır ve sattığı ürünler incelendiğinde hem levha hemde folyo ürünlerinde alaşım, kalınlık, en, kondüsyon bazında çeşitlilik çok fazladır. Bu çeşitlilik içerisinde bazı ürünlerin alaşım, en kalınlık, kondüsyon bazında standart olduğu bazı ürünlerin ise standart dışı olarak kabul edildiği görülmüştür. Bir ürünün standart olup olmadığını belirleyen en önemli unsurlar, yeni döküm ihtiyacının, standart operasyonlara uygunluğunun ve en kombinasyonuna gereksinim olup olmamasıdır. Iki lokasyondaki döküm tezgahlarına göre verimlilik sağlamak adına standart en ve alaşımlarda dökme rulolar dökülebilmektedir. Bu dökme rulolardan, en verimli şekilde standart üretim akışlarına uygun olarak, karşılanabilen ürünler standart ürünler olarak nitelendirilir. Özel döküm ihtiyacı gerektiren veya mevcut dökme rulolardan üretimi standart üretim akışından geçemeyecek veya sipariş kabulü sırasında kombinasyon yapılması gerektiren ürünlere ise standart dışı ürünler olarak tanımlanmaktadır.

Standart dışı ve standart ürünlerin satış detayları incelendiğinde, toplam satışlarının %30 nun standart dışı ürünler olduğu görülmüştür. Bu aradaki standart dışılığa sebebiyet veren temel unsurun sipariş kabulü sırasında kominasyon gerektiren ürünler olmasıdır. Bu noktadan hareket ile işletmede stok kesme probleminin etkin olarak uygulanabileceği görülmüştür ve sipariş kabulü sırasında kombinasyon için herhangi bir matematiksel modellemenin veya paket programın kullanılmadığı belirlenmiştir. Kombinasyon için MS Excel' de sezgisel ve çeşitli deneme yanılma yöntemleri ile çözümler üretilmeye çalışıldığı tespit edilmiştir.

Sipariş kabulü sırasında yapılan kombinasyonda firmanın dikkate aldığı unsurlar incelendiği sırasıyla şu maddeler önem taşımaktadır.

a. Alaşım farklılığı

b. Kalınlık farklılığı

c. Kondüsyon farklılığı

d. Çeşitli enlerdeki dökme rulolar(stok malzemesi)

e. Çeşitli enlerdeki dökme ruloların stok durumu

f. Çeşitli enlerdeki siparişlere ait miktarların karşılanması

g. Siparişlerin kombinasyonu sırasında boydan bölünmeyeceği

h. Hurdanın minimize edilmesi

Yukarıdaki d, g ve h maddelerinden dolayı problemin çeşitli enlerdeki stok malzemelerinin kullanımını konu edinen bir boyutlu stok kesme problemi olduğu tespit edilmiştir. Problemin temel ihtiyaçlarına bağlı kalarak daha anlamlı bir yapı içermesi için probleme maliyet unsurunun eklenmesiyle ve iki lokasyon arasındaki malzeme transferiyle malzemelerin çeşitliliğinin arttırılmasıyla problem biraz daha genişletilmiştir.

Bu çalışmada, stok kesme probleminin çözümüne ilişkin olarak X firmasının uygulamaya çalıştığı deneme yanılma yöntemi geliştirilerek ilk uygun azalan sezgisel ve ardışık sezgisel prosedüri prensiplerine yakın bir sezgisel metot oluşturulacaktır. Bu sezgisel metot ile sonuçlar elde edilmesinin yanısıra, problemin sütun oluşturma metodunu içeren tam sayılı programlama ile çözümü de sunulup iki yöntemin sonuçları kıyaslanacaktır.

Uygulama için otomotiv sektöründe kullanılan, ihraacat ürünü olup sürekli talep gören, standart dışı ürünler kategorisinde yer alıp standart dökme rulolarından kombinasyon yapılması şartıyla verimli üretilmeye çalışılann A müşterisi siparişleri seçilmiştir. A müşterisinin yıllık talep miktarı kalınlık alaşım bazında çeşitlenerek önemli rakamlarda seyir etmektedir. A müşterisi olabildiğince kısa sürede aylık taleplerini X firması ile paylaşarak ürünlerinin üretilmesini beklemektedir. X firması çeşitli kalınlık ve enlerdeki bu talepleri kendi içerisinde standart ebatlardaki dökme rulo malzemelerinden üretmektedir. Ayrıca A müşterisinin otomotiv sektöründeki piyasa hareketlerinden dolayı siparişlerinin ebatları ani olarak değişebilmektedir. Bu tür durumlarda, X firması A müşterisinin ihtiyaçlarına hızlı cevap verebilmek adına geniş enlerdeki dökme rulo malzemeleri nihai enlere dilmeden nihai kalınlığa kadar üretimde proses görmektedir.

Uygulamaya konu olacak problemin örnek1 verilerini paylaşmak gerekirse; A müşterisinin aylık olarak 0,43 mm kalınlıktaki talep ettiği siparişler miktarları çizelge 4.1' de görülmektedir.

Çizelge 4.1 : A müşterisinin 0,43 mm kalınlıktaki sipariş enleri ve miktarları

Sipari Eni	Minimum Talep Miktarı	Maximum Talep Miktarı
220	28	36
245	13	15
390	42	44
465	80	81

Çizelge 4.1 :(devam) A müşterisinin 0,43 mm kalınlıktaki sipariş enleri ve miktarları

Sipari Eni	Minimum Talep Miktarı	Maximum Talep Miktarı
520	34	36
550	66	66
580	21	23
600	67	69
620	67	69
660	71	73
670	68	70
720	23	24
800	40	43
900	67	69
960	19	20
970	95	100
1060	19	21
1260	39	42
1360	36	38

Ayrıca A müşterisinin aylık olarak 0,74 mm kalınlıktaki talep ettiği siparişler miktarları çizelge 4.2' de görülmektedir.

Çizelge 4.2 : A müşterisinin 0,74 mm kalınlıktaki sipariş enleri ve miktarları

Sipari Eni	Minimum Talep Miktarı	Maximum Talep Miktarı
450	38	42
455	19	21
460	52	55
480	82	85
550	128	133
580	38	41
610	18	20
760	9	10
800	54	56
840	69	73
885	7	8
890	83	88
1010	86	89
1200	62	66

X firmasının, A müşterisinin her iki kalınlıktaki siparişlerini karşılamak için kullanacağı standart dökme rulo enleri, stok miktarları, malzeme ve taşıma maliyetleri çizelge 4.3' te görülmektedir. Dökme ruloların bazıları diğer lokasyondan transfer edilmektedir. Ayrıca dökme rulo enleri tezgah hurda miktarlarının düşülmesi sonrası net olarak kullanılabilecek malzeme enidir.

Çizelge 4.3 : X firmasının standart dökme rulo enleri, miktarları ve maliyetleri

Malzeme Eni	Malzeme miktar	Transfer Maliyeti	Malzeme Maliyeti
1250	100	0	9,2
1300	120	0,3	9,3
1500	123	0,3	9,2
1600	155	0	9,2
2000	190	0	9,1

Bu veriler ışığında X firmasının planlama departmanından stok miktarlarını ve sipariş talep sınırlarına bağlı kalarak toplam minimum maliyetli kombinasyonları yapması beklenmektedir Daha önce belirtildiği üzere X firması bu tür kombinasyonları manuel olarak deneme yanılma yöntemi ile tespit etmeye çalışmaktadır. Bu yöntemin ne kadar verimsiz olduğu ileriki aşamalardaki uygulamalar ile ortaya konulacaktır.

4.1 Sezgisel Yöntemin Açıklanması Ve Çözümü

Bu kısımda önerilecek olan sezgisel metot, ilk uygun azalan sezgisel metodunun temeli üzerine kurulmuş olup ardışık sezgisel prosedüründeki gibi belirli kriterlere göre stok kesme problemine çözüm getirmeye çalışacaktır.Algoritmanın adımları sırasıyla şu şekildedir:

Adım 1: Siparişler listesinden en geniş olan sipaişi ve stok malzeme listesinden en düşük maliyetli malzemeyi seç. Adım 2' ye git.

Adım 2: Adım 1' de seçilen malzemeden kesilebinildiği kadar seçilen siparişi kes. Kesim sonrası kalan en yetersiz ise Adım 3' e git.

Adım 3: Adım 2' de yetersiz kalan en için sipariş listesinden en geniş siparişi ara ve Adım 2' deki gibi kalan malzemeden kesilebilinildiği kadar kes. Bu adımı, kalan enin sıfır olması veya sipariş listesinden başka bir ende sipariş atanamamasına kadar tekrarla. Adım 4' e git.

Adım 4: Adım 3' te oluşturulan yerleşim planı, yerleşim planı havuzuna dahil edilir. Sipariş listesinden en geniş sipariş silinir.

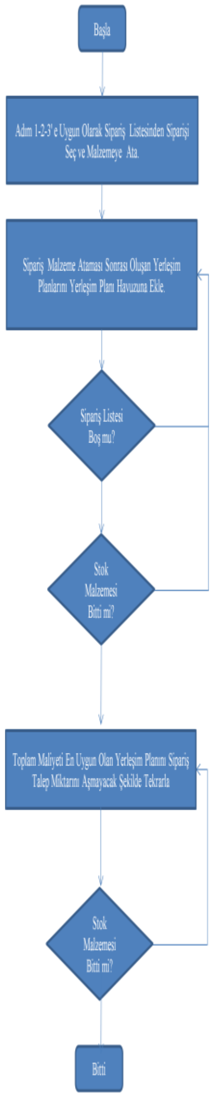
Eğer sipariş listesi boş ise Adım 5’ e gidilir, aksi durumda Adım 1’ e dönülür.

Adım 5: Bu adımda, en düşük maliyetli stok malzemesi malzeme listesinden çıkartılıp tüm siparişler için önceki adımlar tekrarlanır. Eğer stok malzeme listesi boş ise Adım 6’ ya gidilir, aksi halde Adım 1’ e dönülür.

Adım 6: Yerleşim planı havuzunda bulunan her yerleşim planı için toplam maliyet hesaplanır.

Adım 7: En düşük maliyetli yerleşim planı seçilerek, stok malzemesinin miktarı, sipariş talep aralığı aşılmayacak şekilde siparişler karşılanır.Adım 8’ git.

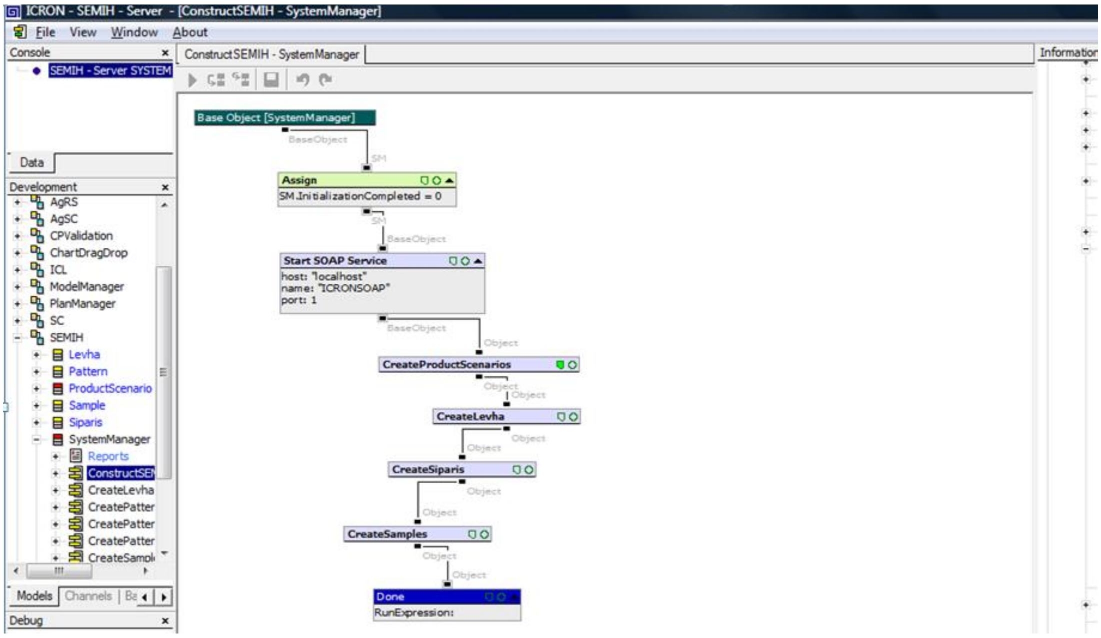
Adım 8: Sipariş talebi karşılanmışsa sonlandır, aksi halde Adım 7’ ye git.



Şekil 4.3 : Sezgisel metodun iş akış şeması

Bu problemin çözümünde kullanılacak algoritmanın yazılması ve sonuçlarının elde edilmesi, ICRON Technologies firmasına ait program tarafından gerçekleştirilmiştir.

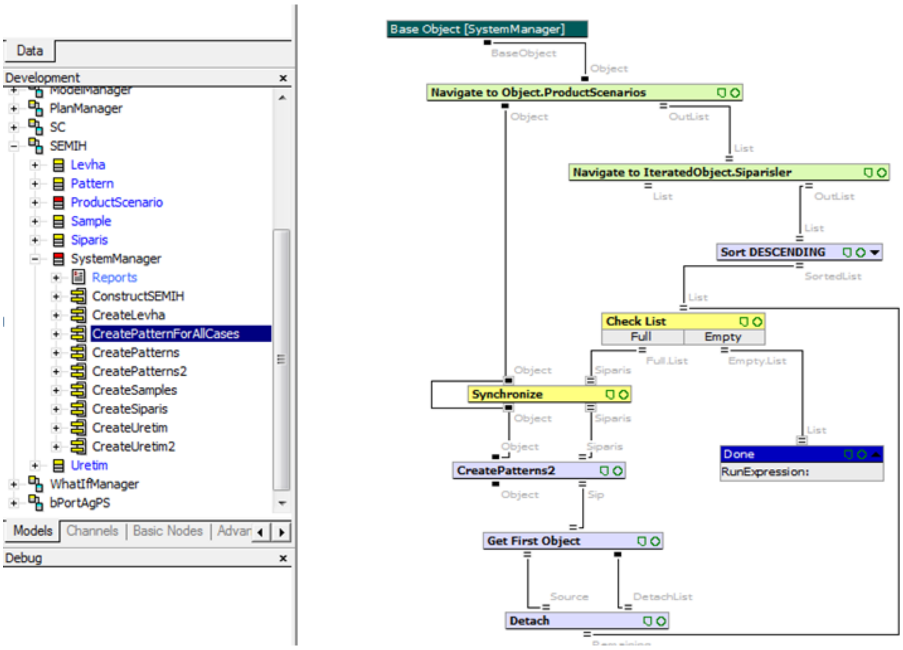
ICRON’ da oluşturulacak algoritmanın veri ihtiyacı MS SQL ile sağlanmıştır. MS SQL’ de oluşturulan levha ve sipariş tabloları ile probleme ait tüm sayısal veriler ICRON’ a aktarılmak üzere hazırlanmıştır. ICRON’ da algoritmanın yazılmasından önce MS SQL deki verilerin tanımlanabilmesi için gerekli tanımlamalar şekil 4.4’ te görüldüğü üzere yapılmıştır.



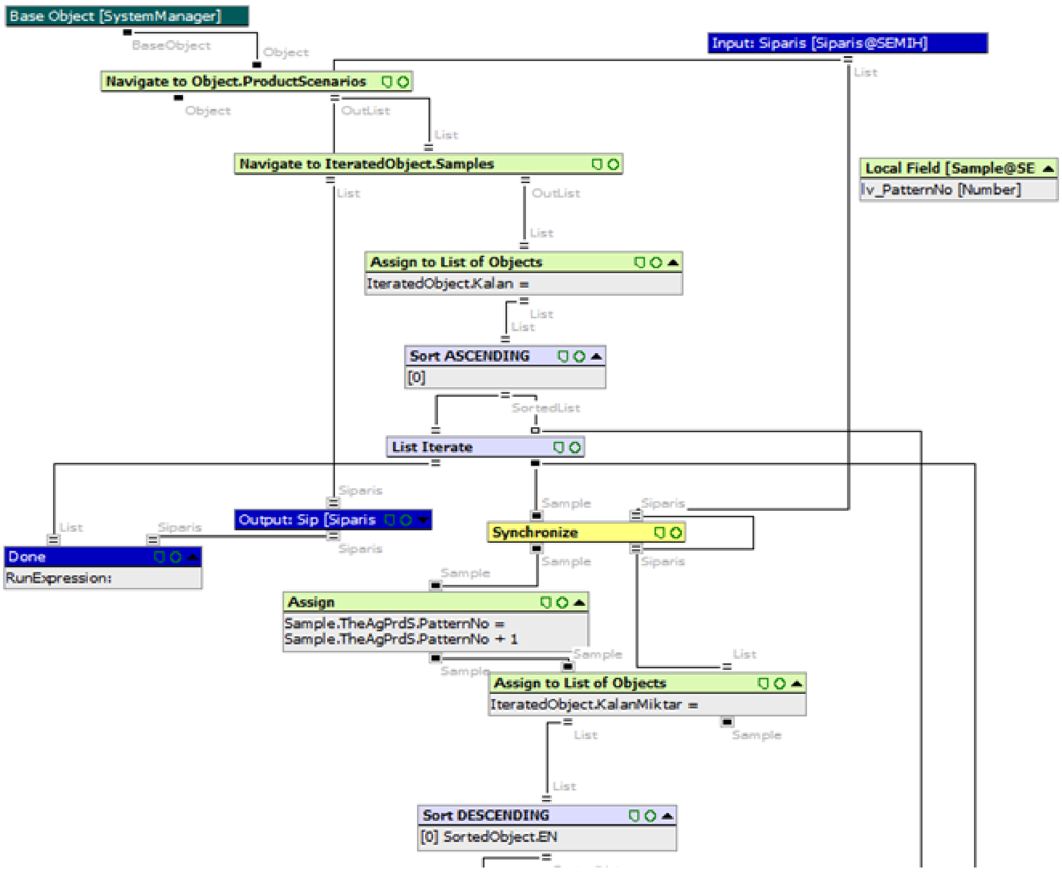
Şekil 4.4 : ICRON&MS SQL ilişkisini kuran yapı

Veri tanımlaması yapıldıktan sonra şekil 4.3’ de belirtilen algoritmaya göre öncelikli olarak en geniş enden en dar ene göre enler sıralanması sağlanır , aynı zamanda malzemeler de toplam maliyete en büyükten en küçüğe doğru sıralanır. Daha sonra bu iki tanımlamanın yardımıyla en düşük toplam maliyete sahip olan malzeme seçilerek en geniş sipariş atanmaya çalışılır. Bu yapının ilk kısmı olan sipariş sıralamanın algoritması şekil 4.5’ de gösterilmektedir.

Benzer şekilde malzemelerin en düşük maliyetli olarak sıralanmasını sağlayan algoritma şekilde 4.6’ daki gibidir. Buarada malzemenin başlangıç eni kalan en diye tanımlanan bir özelliğe eşitlenir. Bu sayede, daha sonraki adımlarda malzemeye atanan siparişlerin enleri malzeme eninden çıkartılarak ilgili andaki maksimum atanabilecek sipariş eni belirlenmiş olur.



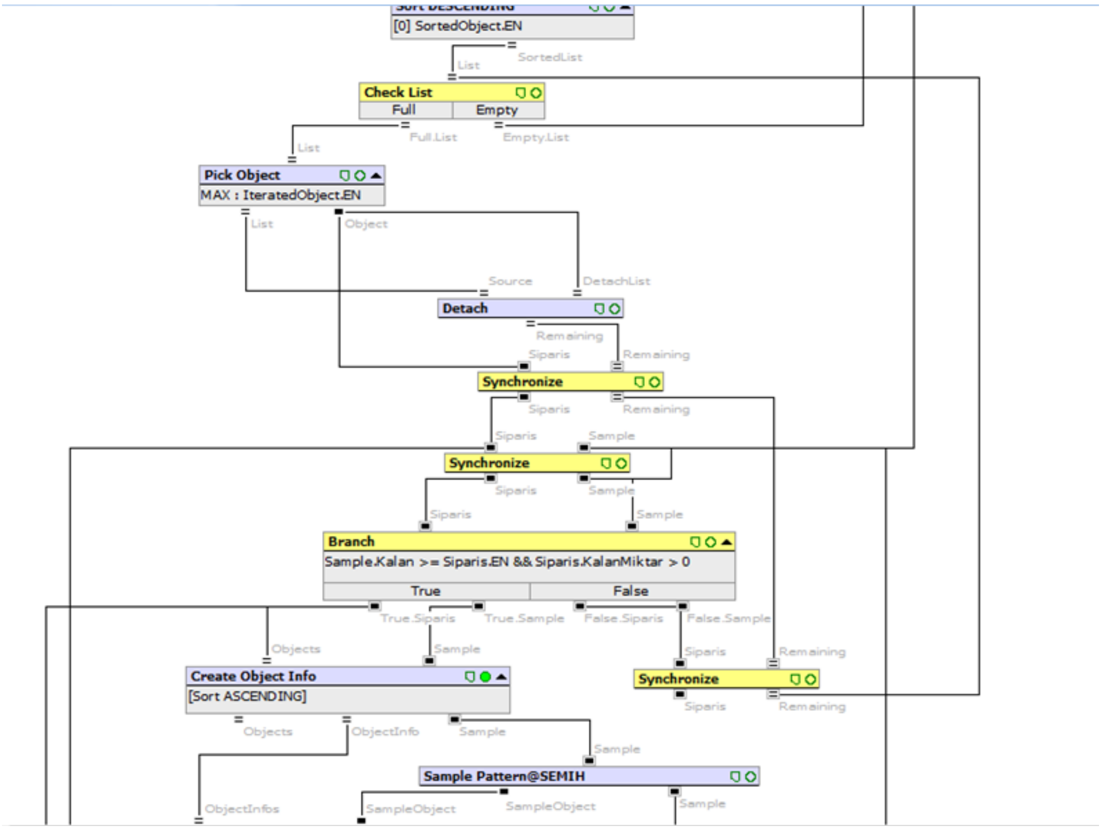
Şekil 4.5 : ICRON' da sipariş sıralamayı gösteren yapı



Şekil 4.6 : ICRON' da malzeme sıralamayı gösteren yapı

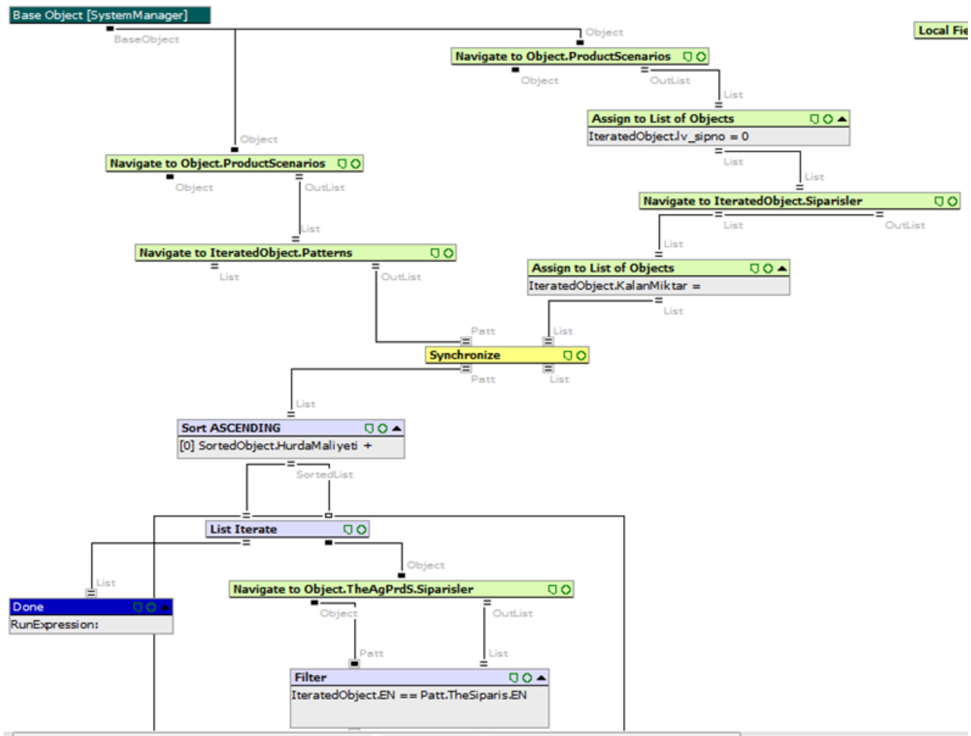
Minimum maliyetli malzemenin seçimi ve maksimum endeki siparişin atanması sonrası, ilgili sipariş için döngü devam eder. Bu sayede malzemeye atanabilecek en çok sayıda aynı sipariş atanması sağlanır. Burada önemli bir nokta siparişin miktarından daha çok sayıdaki siparişin yerleşim planına atanmasını engelleyecek kısıtın konmasıdır. Bu kısıtla ilgili algoritmanın kısmı şekil 4.7' de görülmektedir.

Bu sayede sezgisel yöntemdeki kurala bağlı kalarak tüm siparişler için her malzemeyi deneme yoluyla yerleşim planı alternatifleri oluşturulmuştur.

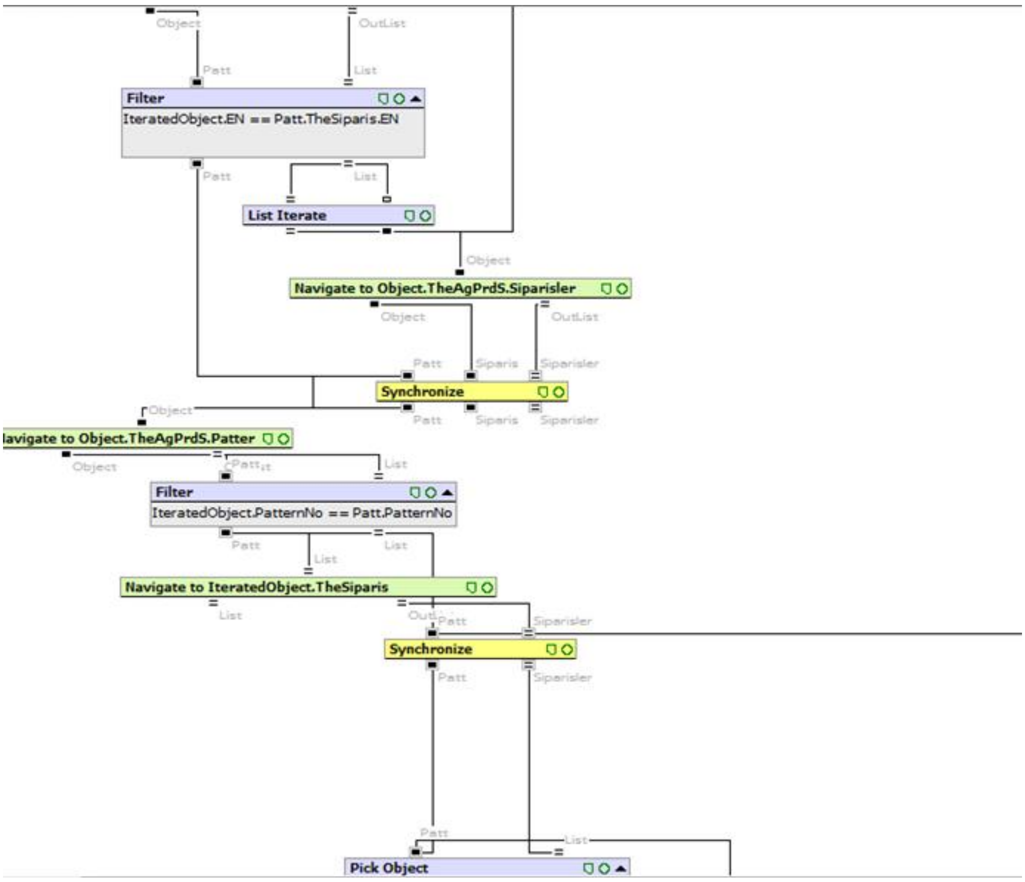


Şekil 4.7 : ICRON' da tüm yerleşim planlarının oluşturulması sırasındamalzeme&sipariş atama yapısı

Bundan sonraki adım ise uygun yerleşim planlarına göre siparişlerin belirli aralıklardaki talep miktarlarını karşılamaktır. Daha önceden belirlenen yerleşim planlarından en düşük toplam maliyetli olan seçilerek, ilgili yerleşim planındaki siparişler algoritmada geri çağırılır. Yerleşim planındaki siparişler ilgili yerleşim planının defalarca uygulanması ile siparişler karşılanmaya başlanır. Şekil 4.8' de bu yapı görülmektedir.

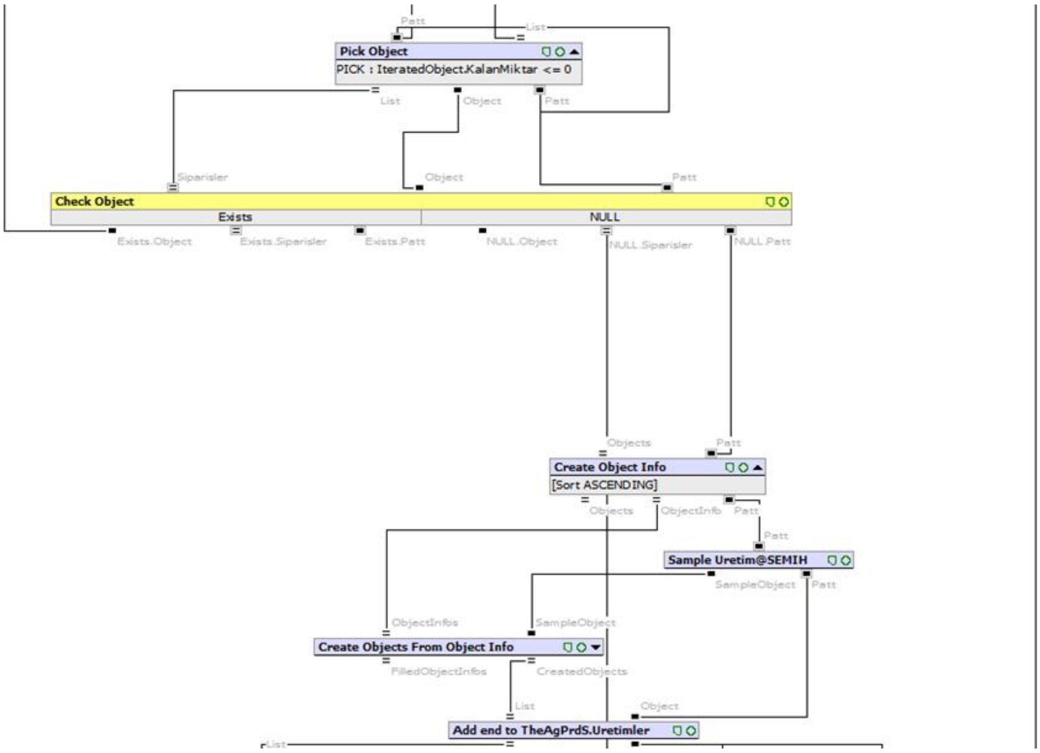


Şekil 4.8 : ICRON' da siparişlerin malzemelerden karşılanma yapısı1

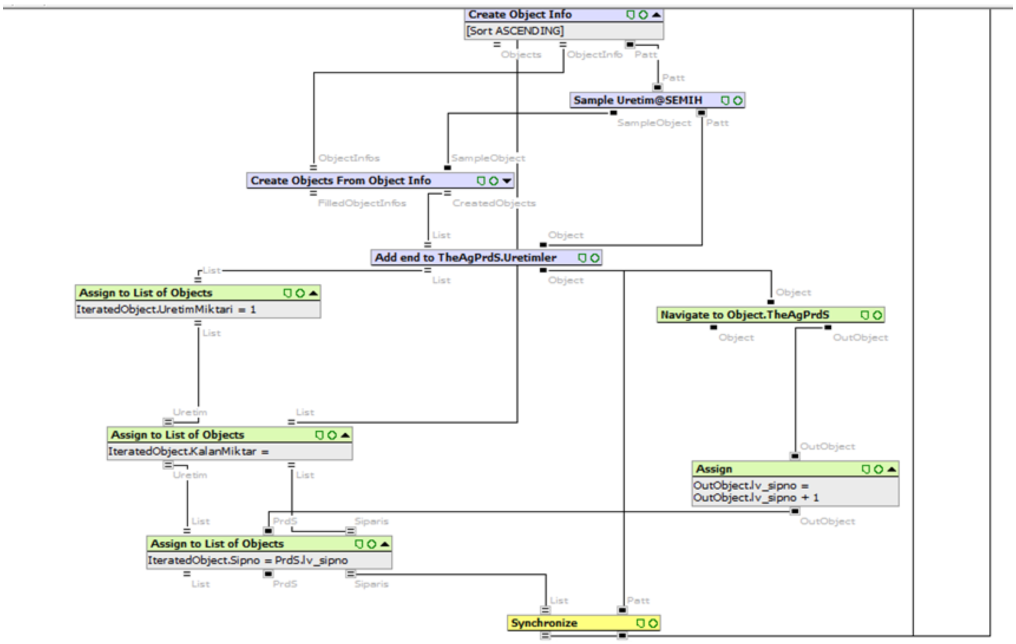


Şekil 4.9 : ICRON' da siparişlerin malzemelerden karşılanma yapısı2

Seçilen yerleşim planında bulunan siparişlerden atama yapılırken gereksiz fazla üretime sebebiyet vermemek için siparişin kalan miktarının sıfırdan küçük olmama kısıtı algoritmaya eklenmiştir.



Şekil 4.10 : ICRON’ da siparişlerin malzemelerden karşılanma yapısı3



Şekil 4.11 : ICRON’ da siparişlerin malzemelerden karşılanma yapısı4

Şekil 4.11’ de görüldü üzere son olarak üretim miktarları, kalan enler gibi algoritma içerisinde sürekli olarak kullanan özelliklerin güncellenmeleri yapılarak model sonlanır.

Bu algoritma iki veri grubu için ayrı ayrı uygulanacaktır ve 0,43 mm ve 0,74 mm siparişlere ait çözümün ayrımı sırasıyla çözüm1, çözüm2 olarak adlandırılmayla yapılacaktır. Öncelikle 0,43 mm kalınlıktaki siparişler ele alınmıştır.

Toplam maliyet açısından bakıldığında çözüm1 in amaç fonksiyonunun 114.493 birim olduğu görülmektedir.

Çizelge 4.4’ da sezgisel yöntemin çözüm1 de kullandığı malzeme miktarı görülmektedir.

Çizelge 4.4 : Çözüm1 de kullanılan malzeme miktarı

Malzeme eni	Kullanılan malzeme miktarı
1250	19
1300	39
1500	123
1600	36
2000	171
	388

Sezgisel yöntemin kapsamı dahilinde çözüm1 için oluşturulan yerleşim planı sayısının çizelge 4.5’ te 95 adet olduğu görülmektedir.

Çizelge 4.5 : Çözüm1 de başlangıçta oluşturulan toplam yerleşim planları

Yerleşim Planı	Kombinasyon	Yerleşim Planı	Kombinasyon
P1	620+1360	48	660+660
P2	1060	49	660+660+245
P3	1360	50	660+620
P4	1360+220	51	620+620+620
P5	1260	52	620+620
P6	1260+720	53	620+620+245
P7	1060	54	620+620+245
P8	1260+220	55	620+620
P9	1260+245	56	600+600+600

Çizelge 4.5 (devam) Çözüm1 de başlangıçta oluşturulan toplam yerleşim planları

Yerleşim Planı	Kombinasyon	Yerleşim Planı	Kombinasyon
P11	1060+900	58	600+600+245
P12	1060	59	600+600+390
P13	1060+390	60	600+600
P14	1060+520	61	580+580+580+245
P15	1060+220	62	580+580
P16	970+970	63	580+580+245
P17	970+245	64	580+580+390
P18	970+520	65	580+580
P19	970+620	66	550+550+550+245
P20	970+245	67	550+550
P21	960+960	68	550+550+390
P22	960+245	69	550+550+465
P23	960+520	70	550+550
P24	960+620	71	520+520+520+390
P25	960+245	72	520+520
P26	900+900	73	520+520+390
P27	900+245	74	520+520+520
P28	900+600	75	520+520+245
P29	900+670	76	465+465+465+465
P30	900+390	77	465+465+245
P31	800+800+390	78	465+465+465
P32	800+390	79	465+465+465
P33	800+670	80	465+465+245
P34	800+800	81	390+390+390+390+390
P35	800+465	82	390+390+390
P36	720+720+550	83	390+390+390+245+
P37	720+520	84	390+390+390+390+
P38	720+720	85	390+390+390
P39	720+720	86	245+245+245+245+245+245+245+245
P40	720+580	87	245+245+245+245+245
P41	670+670+660	88	245+245+245+245+245
P42	670+580	89	245+245+245+245+245
P43	670+670	90	245+245+245+245+245
P44	670+670+245	91	220+220+220+220+220+220+220+220+220
P45	670+620	92	220+220+220+220+220
P46	660+660+660	93	220+220+220+220+220
P47	660+580	94	220+220+220+220+220
		95	220+220+220+220+220

Oluşturulan tüm yerleşim planları içerisinden en küçük maliyetli olanların sırasıyla seçilmesi ve siparişlerin karşılanması işlemi yapıldığında çizelge 4.6’ daki toplam 18 adet yerleşim planlarının çözüm1' in elde edilmesinde kullanıldığı görülmektedir. Yerleşim planlarının sayısı ve kombinasyonları çizelge 4.6 da görülmektedir.

Çizelge 4.6 : Çözüm1 e giren yerleşim planlarının sayısı ve kombinasyonları

Yerleşim Planı No	Yerleşim Planı Sayısı	Kombinasyon
P1	36	620+1360
P2	19	1060
P10	39	1260
P16	15	970+970
P18	34	970+520
P19	31	970+620
P21	10	960+960
P28	67	900+600
P31	20	800+800+390
P36	12	720+720+550
P41	34	670+670+660
P46	13	660+660+660
P61	7	580+580+580+245
P68	22	550+550+390
P69	5	550+550+465
P76	19	465+465+465+465
P86	1	245+245+245+245+245+245+245+245
P91	4	220+220+220+220+220+220+220+220+220
	388	

Çizelge 4.72 ye gore çözüm1 de siparişler minimum ve maksimum sınırları arasında karşılanmıştır.

Çizelge 4.7 : Çözüm1 de siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maximum Talep Miktarı	Karşılanılan Miktar
220	28	36	36
245	13	15	15
390	42	44	42
465	80	81	81
520	34	36	34
550	66	66	66
580	21	23	21
600	67	69	67

Çizelge 4.7: (devam) Çözüm1' deki çözüme göre siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maximum Talep Miktarı	Karşılanılan Miktar
620	67	69	67
660	71	73	73
670	68	70	68
720	23	24	24
800	40	43	40
900	67	69	67
960	19	20	20
970	95	100	95
1060	19	21	19
1260	39	42	39
1360	36	38	36
	895	939	910

Çözüm1' de yerleşim planlarının fire enleri ve toplam fire miktarları çizelge 4.8.' deki gibidir. Toplam fire miktarı 12100 birimdir.

Çizelge 4.8 : Çözüm1 deki fire enleri, sayıları ve toplam miktarları

Yerleşim Planı No	Fire Eni	Yerleşim Planı Sayısı	Fire Miktarı
P1	20	36	720
P2	190	19	3610
P10	40	39	1560
P16	60	15	900
P18	10	34	340
P19	10	31	310
P21	80	10	800
P28	0	67	0
P31	10	20	200
P36	10	12	120
P41	0	34	0
P46	20	13	260
P61	15	7	105
P68	10	22	220
P69	35	5	175
P76	140	19	2660
P86	40	1	40
P91	20	4	80

Çözüm2 incelendiğinde;

Amaç fonksiyonunun değeri 195.135 olarak görünmektedir.

Çizelge 4.9’ de sezgisel yöntemin çözüm2 de kullandığı malzeme miktarı görülmektedir.

Çizelge 4.9 : Çözüm2 de kullanılan malzeme miktarı

Malzeme Eni	Miktar
1250	43
1300	52
1500	66
1600	67
2000	120
Toplam	348

Sezgisel yönteminin kapsamı dahilinde çözüm1 için yerleşim planı sayısı çizelge 4.10’ da 70 adet olduğu görülmektedir.

Çizelge 4.10 : Çözüm2 de başlangıçta oluşturulan toplam yerleşim planları

Yerleşim Planı No	Kombinasyon	Yerleşim Planı No	Kombinasyon
P1	800+1200	36	610*3
P2	1200	37	610*2
P3	1200	38	610*2
P4	1200	39	610*2
P5	890+1010	40	610*2
P6	890+1010	41	580*3
P7	**1010**	42	580*2
P8	480+1010	43	580*2
P9	580+1010	44	580*2
P10	**1010**	45	580*2
P11	890*2	46	550*3
P12	**890**	47	550*2
P13	890+610	48	550*2
P14	890+610	49	480+550*2
P15	890	50	550*2
P16	885*2	51	480*4
P17	885	52	480*2
P18	890+610	53	480*3
P19	890+610	54	480*3
P20	885	55	480*2
P21	840*2	56	460*4
P22	840	57	460*2
P23	840+610	58	460*3

Çizelge 4.10: (devam) Çözüm2' de başlangıçta oluşturulan toplam yerleşim planları

Yerleşim Planı No	Kombinasyon	Yerleşim Planı No	Kombinasyon
P24	840+760	59	460*3
P25	460+840	60	460*2
P26	800*2	61	455*4
P27	760+450	62	455*2
P28	800+610	63	455*3
P29	800*2	64	455*3
P30	800+480	65	455*2
P31	760*2+480	66	450*4
P32	760+480	67	450*2
P33	760+610	68	450*3
P34	760*2	69	450*3
P35	760+480	70	450*2

Oluşturulan tüm yerleşim planları içerisinden en küçük maliyetli olanların sırasıyla seçilmesi ve siparişlerin karşılanması işlemi yapıldığında çizelge 4.11'daki toplam 14 adet yerleşim planlarının sonucun elde edilmesinde kullanıldığı görülmektedir.

Çizelge 4.11 : Çözüm2 ye giren yerleşim planlarının sayısı ve kombinasyonları

Yerleşim Planı No	Yerleşim Planı Sayısı	Kombinasyon
P1	54	800+1200
P2	8	1200
P8	48	480+1010
P9	38	580+1010
P11	33	890*2
P13	18	890+610
P16	4	885*2
P21	9	840*2
P25	52	460+840
P31	5	760*2+480
P47	35	550*2
P49	29	480+550*2
P61	5	455*4
P66	10	450*4

Çözüm 2' de karşılan sipariş miktarları Çizelge 4.12' de görüldüğü üzere çok az miktarlarda maksimum talep miktarını geçebilmektedir.

Çizelge 4.12 : Çözüm2 de siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maksimum Talep Miktarı	Karşılanılan Miktar
450	38	38	40

Çizelge 4.12: (devam) Çözüm2' deki siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maksimum Talep Miktarı	Karşılanılan Miktar
455	19	20	20
460	52	52	52
480	82	82	82
550	128	132	128
580	38	40	38
610	18	20	18
760	9	9	10
800	54	56	54
840	69	69	70
885	7	7	8
890	83	83	84
1010	86	86	86
1200	62	62	62
Toplam	745	756	752

Çözüm 2' de yerleşim planlarının fire enleri ve toplam fire miktarları çizelge 4.13 deki gibidir. Toplam fire miktarı 21050 birimdir.

Çizelge 4.13 : Çözüm2 deki fire enleri, sayıları ve toplam miktarları

Yerleşim Planı No	Fire Eni	Yerleşim Planı Sayısı	Fire Miktarı
P1	0	54	0
P2	50	8	400
P8	10	48	480
P9	10	38	380
P11	220	33	7260
P13	0	18	0
P16	230	4	920
P21	320	9	2880
P25	0	52	0
P31	0	5	0
P47	150	35	5250
P49	20	29	580
P61	180	5	900
P66	200	10	2000
			21050

4.2 Tam Sayılı Modelin Açıklanması ve Çözümü

Bu kısımdaki çözüm metodu tam sayılı programlamayı içermektedir, fakat çözüm yapılırken tüm yerleşim planları göz önünde bulundurularak tamamıyla lineer programlamadan faydanılmayacaktır, aynı zamanda sütun oluşturma yönteminden faydanılarak işe yarayacak yerleşim planların tespit edilmesi ve optimal çözüm bulunmaası amaçlanmaktadır

Problemin notasyonu ve formülasyonu şu şekildedir:

$i =$ siparişlere ait indeks

$k =$ stok malzemelerineait indeks

$p =$ yerleşme planlarına ait indeks

$xp_p =$ p yerleşme planının kullanım sayısı

m_k = k enindeki stok malzemesinden stokta bulunan minimum miktar

$m\max_k$ = k enindeki stok malzemesinden stokta bulunan maksimum miktar

h_k = k enindeki stok malzemesinin malzeme maliyeti

$d\max_i$ = i enindeki siparişe ait maksimum talep miktarı

d_i = i enindeki siparişe ait minimum talep miktarı

t_k = k enindeki stok malzemesinin taşıma maliyeti

θ_{1k} =k malzemesine ait minimum malzeme miktarı kısıtın dual değişkeni

θ_{2k} =k malzemesine ait maximum malzeme miktarı kısıtın dual değişkeni

π_{1i} =i siparişine ait minimum talep miktarı kısıtın dual değişkeni

π_{2i} =i siparişine ait maximum talep miktarı kısıtın dual değişkeni

c_k = k enindeki stok malzemesindeki toplam maliyeti

y_i = alt modelde i siparişin yeni oluşturulan yerleşim planında kullanım sayısı

sk_p = p yerleşme planından doğan fire eni

my_k = k malzemesinin kullanılıp kullanılmadığını gösteren değiştsen

rk_k = k malzemesini genişliği

w_i = i siparişinin genişliği

Ana Model:

$$\text{Min} \sum_{k=1}^{K}\sum_{p=1}^{P} \left(c_k.a_{kp}xp_p\right) + \sum_{k=1}^{K}\sum_{p=1}^{P} h_k.sk_p xp_p \qquad \textbf{(4.1)}$$

Denklem 4.1' de ana modelin amaç fonksiyonunda optimal çözüm için kullanılacak her tip malzemeden oluşacak toplam malzeme maliyetin ve fire maliyetini en küçüklenmek istenmektedir.

Denklem 4.2' de optimal çözümü elde ederken kullanılacak yerleşim planlarında bulunan toplam sipariş miktarının, her bir sipariş için ayrı olmak koşuluyla, minimum ve maksimum talep miktarları arasında olması gerektiğini belirten kısıttır.

$$d_i \le \sum_{p=1}^{P} ai_p.xp_p \le d\max_i \quad \text{her i için,} \qquad \textbf{(4.2)}$$

Optimal çözüme ulaşırken kullanılacak yerleşim planlarının atandığı malzemelerin toplam sayısının, her bir malzeme için ayrı olmak koşuluyla, minimum ve maksimum stok malzeme miktarları arasında olması gerektiğini belirten kısıt Denklem 4.3' dir.

$$m_k \le \sum_{p=1}^{P} ak_p.xp_p \le m\max_k \quad \text{her k için,} \qquad \textbf{(4.3)}$$

Denklem 4.4' te transfer ve malzeme maliyetlerinin toplam maliyete eşit olmasını göstermektedir.

$$c_k = h_k + t_k \qquad \textbf{(4.4)}$$

$$xp_p, ai_p, ak_p \ge 0$$

Alt Model;

Yeni yerleşim planlarının oluşturulmasını sağlayacak model aşağıdaki gibidir.

Denklem 4.5 alt modelin amaç fonksiyonu olup sırt çantası problemine dönüştürebilmek için model -1 ile çarpılabilir.

$$\text{Min} \sum_{k=1}^{K} c_k my_k + \sum_{k=1}^{K} c_k . s_k - \sum_{i=1}^{İ} \pi_{1i} . y_i - \sum_{i=1}^{İ} \pi_{2i} . y_i - \sum_{k=1}^{K} my_k \theta_{1k} . - \sum_{k=1}^{K} my_k \theta_{2k} . \quad \textbf{(4.5)}$$

Denklem 4.6 ilgili k malzemesinin kullanılıp kullanılmadığını gösteren eşitliktir

$$\sum_{k=1}^{K} my_k = 1 \quad \textbf{(4.6)}$$

Denklem 4.7' de alt modelde bulunan her bir yeni yerleşim planındaki siparişlerin toplamının bu siparişlerin atandığı malzemenin toplam eninden küçük olmasını sağlayan kısıttır.

$$\sum_{i=1}^{İ} w_i y_i \leq \sum_{k=1}^{K} rk_k . my_k \quad \textbf{(4.7)}$$

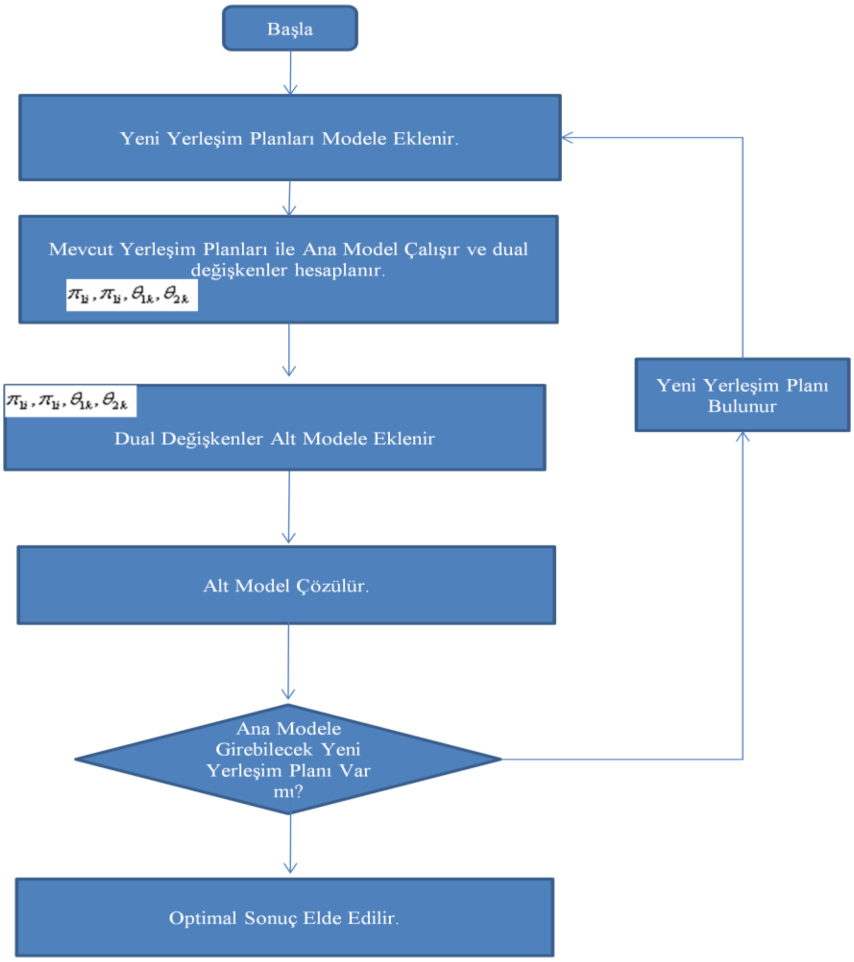
Denklem 4.8' in denklem 4.7' den farkı yeni oluşturulan her bir yerleşim planındaki olası fire eninin belirlenmesi sağlamaktadır

$$\sum_{i=1}^{İ} w_i y_i \leq \sum_{k=1}^{K} rk_k . my_k + sk_p$$
$$my_k \geq 0, \quad \textbf{(4.8)}$$

Şekil 4.12' de iş akışı şematize edilen bu tam sayılı programlama modelinin uygulanması GAMS paket programında gerçekleştirilmiştir.

Algoritmanın GAMS ortamında yazılmış modelinintüm detayını EK D-F ve aynı zamanda tüm sonuçlarını EK E-G kısımlarında sunulmuştur. Burada sadece özet bilgiler sunulacaktır.

Sezgisel yöntemlerin çözümleri, çözüm1 ve çözüm2 olarak belirtilmişti. Tam sayılı programlama ile çözümde de sırasıyla çözüm3 ve çözüm4 olarak ayrım yapılacaktır.



Şekil 4.12 : Tam sayılı programlama sütun oluşturma yöntemi iş akış şeması

Çözüm3 öncelikli olarak incelendiğinde;

Toplam maliyetin 26617 birim olduğu görülmüştür, malzeme kullanım sayısı da en bazında çizelge 4. 14’ daki gibidir.

Çizelge 4.14 : Çözüm3 de kullanılan malzeme miktarı

Malzeme Eni	Malzeme Kullanım Sayısı
2000 mm	190
1600 mm	90
1500 mm	67
1300 mm	0
1250 mm	21
	368

Tam sayılı modelinin çözümü sırasında kullanılan sütun oluşturma metodunun yardımıyla toplam 124 adet yerleşme planı optimal sonucun elde edilmesi sırasında denenmektedir. Optimal sonuç elde edildiğinde toplam 18 adet yerleşim planı kullanılır durumdadır. Her bir yerleşim planının tekrarlanma sayısı ve her bir yerleşim planında oluşan kombinasyon çizelge 4.15' te görülmektedir.

Çizelge 4.15 : Çözüm3 e giren yerleşim planlarının sayısı ve kombinasyonları

Yerleşim Planı No	Yerleşim Planı Sayısı	Kombinasyon
p10	16	660+660+660
p101	67	600+900
p105	10	520+5204960
p106	7	245+390+960
p111	16	220+520+1260
p112	8	245+390+1360
p114	41	620+970
p115	23	720+1260
p117	38	465+550+970
p118	28	620+1360
p119	21	580+670
p120	21	465+465+670
p121	2	220+220+600+960
p123	9	390+550+660
p124	16	220+800+970
p32	12	800+800
p96	19	390+1060+550
p98	14	620+660+660

Çözüm3' te karşınılan siparişlerin miktarları, maksimum ve minimum miktarlarının arasında değiştiği çizelge 4.16' te görülmektedir.

Çizelge 4.16 : Çözüm3 de siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maximum Talep Miktarı	Karşılanılan Miktar
220	28	36	36
245	13	15	15
390	42	44	43
465	80	81	80
520	34	36	36
550	66	66	66
580	21	23	21
600	67	69	69

Çizelge 4.16: (devam) Çözüm3 deki siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maximum Talep Miktarı	Karşılanılan Miktar
620	67	69	69
660	71	73	71
670	68	70	70
720	23	24	23
800	40	43	40
900	67	69	67
960	19	20	19
970	95	100	95
1060	19	21	19
1260	39	42	39
1360	36	38	36
Toplam	895	939	914

Çözüm3' te siparişlerin karşılanmasını sağlayan yerleşim planlarının tekrarlanma sayısı, her bir yerleşim planında oluşan fire eni ve toplam fire miktarı çizelge 4.17' de görülmektedir.

Çizelge 4.17 : Çözüm3 deki fire enleri, sayıları ve toplam miktarları

Yerleşim Planı No	Fire Eni	Yerleşim Planı Sayısı	Fire Miktarı
p10	20	16	320
p32	0	12	0
p96	0	19	0
p98	0	14	0
p101	0	67	0
p105	0	10	0
p106	5	7	35
p111	0	16	0
p112	5	8	40
p114	10	41	410
p115	20	23	460
p117	15	38	570
p118	20	28	560
p119	0	21	0
p120	0	21	0
p121	0	2	0
p123	0	9	0
p124	10	16	160
Toplam	105	368	2555

0,74 mm kalınlığındaki siparişlere ait çözüm4 incelendiğinde amaç fonksiyonundaki toplam maliyetin 24.833 birim olduğu görülmektedir.

Çizelge 4.18' de çözüm4 te toplam 368 adet malzemenin kullanıldığı görünmektedir.

Çizelge 4.18 : Çözüm4 te kullanılan malzeme miktarı

Malzeme Eni	Malzeme Kullanım Sayısı
2000 mm	132
1600 mm	13
1500 mm	121
1300 mm	65
1250 mm	0
	331

Tam sayılı modelinin çözümü sırasında kullanılan sütun oluşturma metodunun yardımıyla toplam 90 adet yerleşme planı optimal sonucun elde edilmesi sırasında denenmektedir. Optimal sonuç elde edildiğinde toplam 15 adet yerleşim planı kullanılır durumdadır. Her bir yerleşim planının tekrarlanma sayısı ve her bir yerleşim planında oluşan kombinasyon çizelge 4.19' da görülmektedir.

Çizelge 4.19 : Çözüm4 e giren yerleşim planlarının sayısı ve kombinasyonları

Yerleşim Planı No	Yerleşim Planı Sayısı	Kombinasyon
p73	1	580*2+840
p74	3	760+840
p75	49	460+840
p76	59	550*2+890
p77	20	610+890
p80	3	580+480*2+460
p81	76	480+1010
p83	10	455*2+580
p84	11	580+450*2
p85	56	1200+800
p86	10	580+1010
p87	7	550*2+885
p88	16	450+840
p89	4	580+890
p90	6	760+1200

Çözüm4’ te karşılanılan siparişlerin miktarları, maksimum ve minimum miktarlarının arasında değiştiği çizelge 4.20’ de görülmektedir.

Çizelge 4.20 : Çözüm4 te siparişlerin karşılanma miktarları

Sipariş Eni	Minimum Talep Miktarı	Maximum Talep Miktarı	Karşılanılan Miktar
450	38	38	38
455	19	20	20
460	52	52	52
480	82	82	82
550	128	132	132
580	38	40	40
610	18	20	20
760	9	9	9
800	54	56	56
840	69	69	69
885	7	7	7
890	83	83	83
1010	86	86	86
1200	62	62	62
Toplam	745	756	756

Çizelge 4.21 : Çözüm4 te fire enleri, sayıları, miktarları

Yerleşim Planı No	Fire Eni	Yerleşim Planı Sayısı	Fire Miktarı
p74	3	0	0
p75	49	0	0
p76	59	10	590
p77	20	0	0
p81	76	10	760
p83	10	10	100
p84	11	20	220
p85	56	0	0
p86	10	10	100
p87	7	15	105
p88	16	10	160
p89	4	30	120
p90	6	40	240
Toplam	328	155	2395

4.3 Uygulama Sonuçlarının Karşılaştırılması

Sezgisel yöntem ile tam sayılı programlama modelin iki ayrı örnek veri üzerindeki sonuçları kıyaslandığında belirgin şekilde metotlar arasında farklılıkların olduğu görülmektedir ve sezgisel metodun daha anlamlı sonuç ürettiği çizelge 4.23’ te görülmektedir.

Çizelge 4.22’ den görüldüğü üzere, benzer sayıdaki siparişin karşılanması için sezgisel yöntemin tam sayılı programlamaya daha fazla malzemeyi kullanarak, daha fazla fireye sebebiyet vererek toplamda dikkate değer maliyet farkı oluşturduğu görülmektedir.

Buradaki temel unsur, kombinasyon veriminin sezgisel yöntemde oldukça düşük olmasıdır. Buna da sebep olarak en geniş siparişlerin öncelikle karşınılmaya çalışılması bazı verimli olabilecek yerleşim planlarının çözümün başında oluşturulamadığını göstermektedir. Ayrıca oluşan yerleşim planlarından en düşük maliyetli olanlarının seçilmeye çalışılması toplam maliyet verimliliği açısından verimsizliğe yol açmaktadır.

Bu sonuçlardan dolayı sütun oluşturma yöntemini içeren tam sayılı programla çözümünün tercih edilmesi kaçınılmaz olup X firmasının bundan sonraki uygulamalarında optimizasyon aracı olarak kullanılması kararlaştırılmıştır. Bu sayede X firması, fire maliyetlerinde önemli derecede tasarruf edebilecektir.

Çizelge 4.22 : Sezgisel ve tam sayılı programlama yöntemlerinin kıyaslanması

	Sezgisel Yöntem	Tam sayılı programlama	Sezgisel Yöntem	Tam sayılı programlama
	Çözüm1	Çözüm3	Çözüm2	Çözüm4
Toplam Maliyet	114493	26617	195135	24833
Kullanılan Malzeme Sayısı	388	368	348	331
Oluşturulan Yerleşme Planı Sayısı	95	124	70	90
Çözümde Kullanılan Yerleşme Planı Sayısı	18	18	14	15
Karşılanılan Sipariş Miktarları	910	914	752	756
Toplam Fire Miktarı	12100	2555	21050	2395

5. SONUÇ VE ÖNERİLER

Günümüzde, müşteri baskın piyasa koşullarının oluşmasıyla birlikte işletmelerin rekabet edebilmesi, kar marjlarını yönetebilmesi için maliyetlerinin azaltılmasına yoğunlaşması kaçınılmaz duruma gelmiştir. İşletmelerin, her alandaki maliyetlerini inceleyerek gerekli önlemleri almaları gerekmektedir. İşletmelerin faaliyet alanlarına göre maliyet kalemlerinin önem derecesi değişebilmektedir. Hizmet sektöründe faaliyet gösteren işletmelerde işgücü önemli maliyet kalemi iken üretim yapan işletmelerde hammadde, enerji , hurda maliyeti, tezgah maliyeti en önemli kalemlerdir.

Alüminyum, çelik, demir, kağıt, plastik, cam vb. alanlarda üretim bazı işletmeler müşteriye hızlı cevap verebilme, ölçek ekonomisi vb. sebeplerden dolayı nihai ürünlerini belirli aşamalara getirilen yarı mamul stok malzemelerinden karşılamak zorundadırlar. Nihai müşteri siparşinden daha büyük boyutlarda olan stok malzemelerinin kesme dilme vb. işlemlerden geçirilmesiyle nihai müşteri siparişi elde edilir. Bu tür işletmelerde kesim işlemleri sırasında kesme kayıplarının(fire) olması kaçınılmazdır. Bu problem literatürde stok kesme problemi olarak karşımıza çıkmaktadır.

1940' lı yılların başlangıcından önce matematiksel modeller kullanmaksızın verimliliği önemsemeden deneme yanılma yöntemleri ile yapılan stok kesme işlemleri zaman ilerledikçe ilgi çekmeye başlayarak stok kesme probleminin tanımlanması ve geliştirilmesi önemli hale gelmiştir. Bu noktadan hareket ile stok kesme problemi üzerine önemli sayıda çalışmalar yapılarak literatür zengin hale gelmiştir ve sürekli çalışmalar yapılmaya devam edilmektedir.

Bu çalışmada, öncelikle stok kesme probleminin tanımı, stok kesme problemlerinin sınıflandırılmasının nasıl yapıldığı ve stok kesme problemlerinin çözümünde kullanılan algoritmik, sezgisel ve metasezgisel çözüm metotlarının alt başlıkları ile birlikte neler olduğu detaylı şekilde incelenmiş ve alüminyum sektöründe bir boyutlu stok kesme problemi ile ilgili bir uygulama yapılmştır. Uygulama çerçevesinde sezgisel ve tam sayılı programlama yöntemleri ile iki ayrı örnek üzerinden çözümler sunulmuştur ve yerleşim planlarını belirli sistematik içerisinde sütun oluşturma tekniğini kullanarak yaratan tam sayılı programlama modelinin oldukça iyi bir sonuç verdiği sonucuna varılmıştır.

KAYNAKLAR

Alptekin, E., 1998 Yöneylem Araştırmasında Yararlanılan Karar Yöntemleri 3.Basım, Gazi Üniversitesi Yayınları, Ankara, s:24.

Anderson ve diğ., 2006, age. s. 340-341; Halaç, age. s. 476; Özgüven, age. s. 197-198. Afyon Kocatepe Üniversitesi, İ.İ.B.F. Dergisi (C.VIII ,S.1, 2006)

Callaghan, A. R., Nair, A. R. and Lewis, K. E., 1999, An extension of the orthogonal packing problem through dimensional flexibility, *ASME Design Engineering Technical Conferences*, 12-15.

David, G. Luenberger. 1984 Linear and Nonlinear Programming, Second Edition, *Addison-Wesley Publishing Company*, London, , s. 14.

Dowsland, K.A., 1993. Some Experiments with Simulated Annealing Techniques For Packing Problems, *European Journal of Operational Research*, Vol. 68, pp. 389-399.

Ducatelle, F., & Levine, J. , 2002. Ant colony optimisation for bin packing and cutting stock problems. http://citeseer.nj.nec.com/506184.html alındığı tarih 03.03.2010..

Dyckhoff, H. , 1990. A typology of cutting and packing problems. European Journal of Operational Research, 44: 145–159.

Faina, L., 1999. Application of Simulated Annealing to The Cutting Stock Problem, *European Journal of Research*, Vol. 114, pp. 542-556.

Feo T.A., Bard J., 1989**.** Flight scheduling and maintenance base planning, Management Science, 35(12), 1415-1432.

Gilmore, P. C., & Gomory, R. E. , 1961. A linear programming approach to the cuttingstock problem. Operations Research, 9: 849–859.

Gilmore, P. C., & Gomory, R. E. , 1963. A linear programming approach to the cuttingstock problem — part II. Operations Research, 11: 863–888.

Gilmore, P. C., & Gomory, R. E. , 1964. Multistage cutting stock problems of two and more dimensions. Operations Research, 13: 94–120.

Gilmore, P. C., & Gomory, R. E. , 1966. The theory and computation of knapsack functions. Operations Research, 14: 1045–1074.

Goulimis, C. , 1990. Optimal solutions for the cutting stock problem, *European Journal of Operational Research*, 44: 197–208.

Gupta S.R., Smith J., 2006**.** Algorithms for single machine total tardiness scheduling with sequence dependent setups, *European Journal of Operational Research*, 175, 722-739.

Abrahart, R. J., and See, L., 2000: Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments, *Hydrolog. Process.*, **14**, 2157–2172.

Haessler, R. W. , 1975. Controlling cutting pattern changes in one-dimensional trim problems, Operations Research, 23: 483–493.

Haessler, R. W. , 1980. A note on computational modifications to the Gilmore-cutting stock algorithm. Operations Research, 28: 1001–1005.

Haessler, R. W. , 1988. Selection and design of heuristic procedures for solving roll trim problems. Management Science, 34: 1460–1471.

Hamdy A. TAHA, 1992, Operations Research An Introduction, Fifth Edition, Macmillan Publishing Company, New York, 1992, s. 303.

Hopper, E. and Turton, B., 2001. A Review of The Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems, *Artificial Intelligence*

Kämpke, T., 1988. Simulated Annealing: Use of A New Tool in Bin-packing, *Annals of Operations Research*, Vol. 16, pp. 327-332.

Lai, K.K. and Chan, J.W.M., 1997. Developing A Simulated Annealing Algorithm for The Cutting Stock Problem, *Computers and Industrial Engineering*, Vol. 32, pp. 115-127.

Leung, T.W., Yung, C.H. and Troutt, M.D., 2001. Applications of Genetic Search and Simulated Annealing to The Two-Dimensional Non-Guillotine Cutting StockProblem, *Computers and Industrial Engineering*, Vol.40, pp. 201-214.

Masri, S.F., Smith, A.W., Chass_akkos, A.G., Nakamura, M. and Cughey T.K., 1999. *Training Neural Networks by Adaptive Random Search Techniques. Journal of Enginnering Mechanics,* Vol 125(2), pp.123-132.

Mori,H., Usami, 1998. 'Unit Commitment Using Tabu search with Restricted and Neighborhood*, Proc. of ISAP'96*, No.0221, pp. 422-427.

Öztürk, AHMET, 2002, Yöneylem Araştırması,, 8. Baskı, Ekin Kitabevi Yayınları, Bursa, , s. 23; s. 167;

Özkan, J., 1986, Kantitatif Karar Verme Teknikleri, A.Ü. .B.F. Araştırma Merkezi, Ders notları:120, Erzurum, s. 1

Resende M.G.C., Riberio C.C., 2002. Greedy randomized adaptive search procedures, *International Series in Operations Research and Management Science*, Vol. 57.

Shahookar K. and Mazumder P., 1991. Vls_ Cell Placement Techniques, *ACM computing Surveys*, Vol. 23(2), pp.144-172.

Şenel M., 1974, Doğrusal Programlama Metodu İle Üretim Planlaması ve bir Tekstil İşletmesinde Uygulama, *Eskişehir İktisadi ve Ticari İlimler Yayınları*, No:110/64, Sevinç Matbaası, Ankara, s:9.

Url-1 <*http://wwwgams.com*>, alındığı tarih 29.07.2010.

Url-2 <*http://enm.blogcu.com* >, alındığı tarih 10.05.2010.

Wongprakornkul, S.,Charnsethikul P., 2010. Solving One-Dimensional cutting stock problem with Descrete Demands and Capaciateted Planning Objective, *International Series in Operations Research and Management Science*, 40.

Yamak O., 1994, Üretim Yönetimi, Sistemler, İlkeler Ve Teknikler, Marmara Üniv., İktisadi ve idari Bilimler Fakültesi, İşletme Bölümü, Üretim Yönetimi ve Pazarlama Ana Bilim Dalı,İstanbul, s:221..

EKLER

EK A:

```
$ontext
Bir Boyutlu Tek Malzemeli Stok Kesme Problemi Örneği
(Yerleşim planlarının manuel olarak modele tanımlanmıştır.)
$offtext

set
i ’en’ /en1*en4/
p ’yerleşim planı’ /YerlesimP1*YerlesimP18/
;
*--------------------------------------------------
* Data
*--------------------------------------------------
scalar M ’stok malzemesi eni’ /2000/;

table SiparisTalepData(i,*)
      sipariseni    talep
  en1    900         511
  en2    800         301
  en3    700         263
  en4    600         383
;

table YerlesimPdata(p,i)
                  En1      En2      En3      En4
 YerlesimP1        2
 YerlesimP2        1        1
 YerlesimP3        1                 1
 YerlesimP4        1                          1
 YerlesimP5        1
 YerlesimP6                 2
 YerlesimP7                 1        1
 YerlesimP8                 1                 1
 YerlesimP9                 1                 2
 YerlesimP10                1
 YerlesimP11                         2
 YerlesimP12                         2        1
 YerlesimP13                         1        1
 YerlesimP14                         1        2
 YerlesimP15                         1
 YerlesimP16                                  3
 YerlesimP17                                  2
 YerlesimP18                                  1
;
```

```
parameter w(i);
w(i) = SiparisTalepData(i,'spariseni');
parameter d(i);
d(i) = SiparisTalepData(i,'talep');
parameter a(i,p);
a(i,p) = YerlesimPdata(p,i);
abort$(sum(p$(sum(i, a(i,p)*w(i)) > M+0.0001), 1)) "Yerlesim Planı eni stok
malzemesinin eninden büyükür";
*----------------------------------------------------
* MIP formulation
*----------------------------------------------------
integer variable x(p) ’Yerlesim Planı Kullanım Sayısı’;
variable z ’objective’;
x.up(p) = sum(i, d(i));
equations
YerlesimPlansayisi ’Yerlesim Planı Kullanım Sayısı (amaç)’
talepmiktari(i) ’Siparis Talep Miktarı’
;
YerlesimPlansayisi.. z =e= sum(p, x(p));
talepmiktari(i).. sum(p, a(i,p)*x(p)) =g= d(i);
model cut1 /YerlesimPlansayisi,talepmiktari/;
option optcr = 0.0;
solve cut1 using mip minimizing z;
display x.l;
```

EK B : Bir Boyutlu Tek Malzemeli Stok Kesme Problemi Örneğinin GAMS modeli

```
GAMS Rev 141  Intel /MS Window                    09/12/10 23:40:43 Page 1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n
   Bir Boyutlu Tek Malzemeli Stok Kesme Problemi Örneği
  (Yerleşim planları sütun oluşturma yöntemi ile oluşturulmuştur.)

Set  i    en    /en1*en4/
     p        Yerlesim Planı Kümesi  /p1*p1000/
     pp(p)    Yerlesim Planı Kümesinin Alt Kümesi
Parameter
     M    Stok Malzemesi Eni /2000/
     w(i) en        /en1  900
                     en2  800
                     en3  700
                     en4  600/
     d(i) talep     /en1  511
                     en2  301
                     en3  263
                     en4  383/

* Sütun Oluşturma Algoritması

```

```
Parameter
    aip(i,p) p Yerlesim Planındaki i Enindeki Sipari Sayısı;
* Ana model
Variable xp(p)     Yerlesim Planı Kullanım Sayısı
       z         Amaç Fonksiyonu Değişkeni
Integer variable xp; xp.up(p) = sum(i, d(i));
Equation master_yerlesimplanisayisi    Yerlesimplanı Kullanım Denklemi
       master_siparistalep(i) Karşılanılan Sipariş Talep Miktarı Denklemi;
master_yerlesimplanisayisi..     z =e= sum(pp, xp(pp));
master_siparistalep(i)..  sum(pp, aip(i,pp)*xp(pp)) =g= d(i);
model master /master_yerlesimplanisayisi, master_siparistalep/;
* Alt model-Knapsack model
Variable  y(i) Yeni Yerleşimplanı;
Integer variable y; y.up(i) = ceil(M/w(i));
Equation defobj
       knapsack Alt Model Kısıtı;
defobj..     z =e= 1 - sum(i, master_siparistalep.m(i)*y(i));
knapsack..   sum(i, w(i)*y(i)) =l= M;
```

```
model pricing /defobj, knapsack/;

* Başlangıç yerleşim planlarının herbiri sipariş eninden sadece bir adet içermektedir..
pp(p) = ord(p)<=card(i);
aip(i,pp(p))$(ord(i)=ord(p)) = floor(M/w(i));
*display aip;

Scalar done  döngü belirleyicisi /0/
Set    pi(p) son yerleşimplanı kümesi; pi(p) = ord(p)=card(pp)+1;

option optcr=0,limrow=0,limcol=0,solprint=off;

 While(not done and card(pp)<card(p),
  solve master using rmip minimizing z;
  solve pricing using mip minimizing z;

* Ana modeli geliştirebilecek yeni yerleşim planı mevcut mudur?
  if(z.l < -0.001,
     aip(i,pi) = round(y.l(i));
     pp(pi) = yes; pi(p) = pi(p-1);
  else
     done = 1;
  );
);
```

```
option solprint=on;
solve master using mip minimizing z;
Parameter patrep Yerleşim planı Sonuc Raporu
          demrep Sipariş Talep Karsilama Raporu;
patrep('# Üretilen',p) = round(xp.l(p));
patrep(i,p)$patrep('# Üretilen',p) = aip(i,p);
patrep(i,'Toplam') = sum(p, patrep(i,p));
patrep('# Üretilen','Toplam') = sum(p, patrep('# Üretilen',p));
demrep(i,'Üretilen') = sum(p,patrep(i,p)*patrep('# Üretilen',p));
demrep(i,'Talep') = d(i);
demrep(i,'Fazla') = demrep(i,'Üretilen') - demrep(i,'Talep');
display patrep, demrep;
```

EK C : Bir Boyutlu Tek Malzemeli Stok Kesme Problemi Sonuçları

COMPILATION TIME = 0.000 SECONDS 3.2 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 2

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Model Statistics SOLVE master Using RMIP From line 63

LOOPS FOR/WHILE 1

MODEL STATISTICS

BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 5

BLOCKS OF VARIABLES 2 SINGLE VARIABLES 5

NON ZERO ELEMENTS 9 DISCRETE VARIABLES 4

GENERATION TIME = 0.062 SECONDS 3.9 Mb WIN216-141 Jan 24, 2005

EXECUTION TIME = 0.078 SECONDS 3.9 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 3

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Solution Report SOLVE master Using RMIP From line 63

L O O P S FOR/WHILE 1

S O L V E S U M M A R Y

MODEL master OBJECTIVE z

TYPE RMIP DIRECTION MINIMIZE

SOLVER CPLEX FROM LINE 63

**** SOLVER STATUS 1 NORMAL COMPLETION

**** MODEL STATUS 1 OPTIMAL

**** OBJECTIVE VALUE 665.1667

RESOURCE USAGE, LIMIT 0.006 1000.000

ITERATION COUNT, LIMIT 0 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0

Cplex 9.0.2, GAMS Link 26

Optimal solution found.

Objective : 665.166667

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 4
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Model Statistics SOLVE pricing Using MIP From line 64

LOOPS FOR/WHILE 1
MODEL STATISTICS
BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 2
BLOCKS OF VARIABLES 2 SINGLE VARIABLES 5
NON ZERO ELEMENTS 9 DISCRETE VARIABLES 4
GENERATION TIME = 0.000 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

EXECUTION TIME = 0.016 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 5
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Solution Report SOLVE pricing Using MIP From line 64

L O O P S FOR/WHILE 1

S O L V E S U M M A R Y
MODEL pricing OBJECTIVE z
TYPE MIP DIRECTION MINIMIZE
SOLVER CPLEX FROM LINE 64

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 1 OPTIMAL
**** OBJECTIVE VALUE -0.3333

RESOURCE USAGE, LIMIT 0.030 1000.000
ITERATION COUNT, LIMIT 1 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0
Cplex 9.0.2, GAMS Link 26
Proven optimal solution.
MIP Solution: -0.333333 (1 iterations, 0 nodes)
Final Solve: -0.333333 (0 iterations)
Best possible: -0.333333
Absolute gap: 0.000000
Relative gap: 0.000000

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED
GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 6
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Model Statistics SOLVE master Using RMIP From line 63

LOOPS FOR/WHILE 2

MODEL STATISTICS
BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 5
BLOCKS OF VARIABLES 2 SINGLE VARIABLES 6
NON ZERO ELEMENTS 12 DISCRETE VARIABLES 5

GENERATION TIME = 0.000 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005
EXECUTION TIME = 0.000 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005
GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 7
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Solution Report SOLVE master Using RMIP From line 63
L O O P S FOR/WHILE 2

S O L V E S U M M A R Y

MODEL master OBJECTIVE z
TYPE RMIP DIRECTION MINIMIZE
SOLVER CPLEX FROM LINE 63

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 1 OPTIMAL
**** OBJECTIVE VALUE 621.3333

RESOURCE USAGE, LIMIT 0.017 1000.000
ITERATION COUNT, LIMIT 1 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0
Cplex 9.0.2, GAMS Link 26

Optimal solution found.
Objective : 621.333333

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 8
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Model Statistics SOLVE pricing Using MIP From line 64

LOOPS FOR/WHILE 2

MODEL STATISTICS

BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 2
BLOCKS OF VARIABLES 2 SINGLE VARIABLES 5
NON ZERO ELEMENTS 9 DISCRETE VARIABLES 4
GENERATION TIME = 0.016 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

EXECUTION TIME = 0.016 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 9

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Solution Report SOLVE pricing Using MIP From line 64

L O O P S FOR/WHILE 2

S O L V E S U M M A R Y

MODEL pricing OBJECTIVE z

TYPE MIP DIRECTION MINIMIZE

SOLVER CPLEX FROM LINE 64

**** SOLVER STATUS 1 NORMAL COMPLETION

**** MODEL STATUS 1 OPTIMAL

**** OBJECTIVE VALUE -0.1667

RESOURCE USAGE, LIMIT 0.059 1000.000

ITERATION COUNT, LIMIT 3 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0

Cplex 9.0.2, GAMS Link 26

Proven optimal solution.

MIP Solution: -0.166667 (3 iterations, 0 nodes)

Final Solve: -0.166667 (0 iterations)

Best possible: -0.166667

Absolute gap: 0.000000

Relative gap: 0.000000

**** REPORT SUMMARY : 0 NONOPT

0 INFEASIBLE

0 UNBOUNDED

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 10

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Model Statistics SOLVE master Using RMIP From line 63

LOOPS FOR/WHILE 3

MODEL STATISTICS

BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 5

BLOCKS OF VARIABLES 2 SINGLE VARIABLES 7

NON ZERO ELEMENTS 15 DISCRETE VARIABLES 6

GENERATION TIME = 0.015 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

EXECUTION TIME = 0.015 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 11

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Solution Report SOLVE master Using RMIP From line 63

L O O P S FOR/WHILE 3

S O L V E S U M M A R Y

MODEL master OBJECTIVE z

TYPE RMIP DIRECTION MINIMIZE

SOLVER CPLEX FROM LINE 63

**** SOLVER STATUS 1 NORMAL COMPLETION

**** MODEL STATUS 1 OPTIMAL

**** OBJECTIVE VALUE 600.3750

RESOURCE USAGE, LIMIT 0.017 1000.000

ITERATION COUNT, LIMIT 1 10000

GAMS/Cplex   Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0
Cplex 9.0.2, GAMS Link 26

Optimal solution found.
Objective :   600.375000

**** REPORT SUMMARY :   0   NONOPT
                             0 INFEASIBLE
                             0 UNBOUNDED
GAMS Rev 141 Intel /MS Window   09/12/10 23:40:43 Page 12
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
Model Statistics   SOLVE pricing Using MIP From line 64

LOOPS   FOR/WHILE   3

MODEL STATISTICS
BLOCKS OF EQUATIONS   2   SINGLE EQUATIONS   2
BLOCKS OF VARIABLES   2   SINGLE VARIABLES   5
NON ZERO ELEMENTS   9   DISCRETE VARIABLES   4

GENERATION TIME   =   0.015 SECONDS   2.9 Mb   WIN216-141 Jan 24, 2005

EXECUTION TIME   =   0.015 SECONDS   2.9 Mb   WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window   09/12/10 23:40:43 Page 13
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
Solution Report   SOLVE pricing Using MIP From line 64

L O O P S   FOR/WHILE 3
S O L V E   S U M M A R Y
MODEL   pricing   OBJECTIVE z
TYPE   MIP   DIRECTION MINIMIZE
SOLVER CPLEX   FROM LINE 64

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 1 OPTIMAL
**** OBJECTIVE VALUE 0.0000

RESOURCE USAGE, LIMIT 0.056 1000.000
ITERATION COUNT, LIMIT 12 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0
Cplex 9.0.2, GAMS Link 26
Proven optimal solution.
MIP Solution: -0.000000 (12 iterations, 6 nodes)
Final Solve: 0.000000 (0 iterations)
Best possible: -0.000000
Absolute gap: 0.000000
Relative gap: 0.000000

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 14
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Model Statistics SOLVE master Using MIP From line 75

MODEL STATISTICS
BLOCKS OF EQUATIONS 2 SINGLE EQUATIONS 5
BLOCKS OF VARIABLES 2 SINGLE VARIABLES 7
NON ZERO ELEMENTS 15 DISCRETE VARIABLES 6

GENERATION TIME = 0.016 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005
EXECUTION TIME = 0.016 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

GAMS Rev 141 Intel /MS Window 09/12/10 23:40:43 Page 15

General Algebraic Modeling System
Solution Report SOLVE master Using MIP From line 75

S O L V E   S U M M A R Y

MODEL master OBJECTIVE z
TYPE MIP DIRECTION MINIMIZE
SOLVER CPLEX FROM LINE 75

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 1 OPTIMAL
**** OBJECTIVE VALUE 602.0000

RESOURCE USAGE, LIMIT 0.080 1000.000
ITERATION COUNT, LIMIT 4 10000

GAMS/Cplex Jan 26, 2005 WIN.CP.CP 21.6 027.030.041.VIS For Cplex 9.0
Cplex 9.0.2, GAMS Link 26

Proven optimal solution.

MIP Solution: 602.000000 (4 iterations, 0 nodes)
Final Solve: 602.000000 (0 iterations)

Best possible: 602.000000
Absolute gap: 0.000000
Relative gap: 0.000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU master_ye~	.	.	.	1.000

master_yerlesimplanisayisi Yerlesimplanı Kullanım Denklemi

---- EQU master_siparistalep Karşılanılan Sipariş Talep Miktarı Denklemi

	LOWER	LEVEL	UPPER	MARGINAL
en1	511.000	512.000	+INF	.
en2	301.000	302.000	+INF	.
en3	263.000	264.000	+INF	.
en4	383.000	384.000	+INF	.

---- VAR xp Yerlesim Planı Kullanım Sayısı

	LOWER	LEVEL	UPPER	MARGINAL
p1	.	256.000	1458.000	1.000
p2	.	88.000	1458.000	1.000
p3	.	.	1458.000	1.000
p4	.	.	1458.000	1.000
p5	.	132.000	1458.000	1.000
p6	.	126.000	1458.000	1.000

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR z	-INF	602.000	+INF	.

z Amaç Fonksiyonu Değişkeni

**** REPORT SUMMARY :   0   NONOPT
0 INFEASIBLE
0 UNBOUNDED

GAMS Rev 141 Intel /MS Window   09/12/10 23:40:43 Page 16
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
E x e c u t i o n

----   89 PARAMETER patrep Yerleşim planı Sonuc Raporu

	p1	p2	p5	p6	Toplam
en1	2.000				2.000
en2		2.000		1.000	3.000
en3			2.000		2.000
en4			1.000	2.000	3.000
# Üretilen	256.000	88.000	132.000	126.000	602.000

---- 89 PARAMETER demrep Sipariş Talep Karsilama Raporu

	Üretilen	Talep	Fazla
en1	512.000	511.000	1.000
en2	302.000	301.000	1.000
en3	264.000	263.000	1.000
en4	384.000	383.000	1.000

EXECUTION TIME = 0.000 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

USER: Centre Versailles - Grigon S021106:1507AB-WIN
I.N.R.A. DC2726
License for teaching and research at degree granting institutions

**** FILE SUMMARY

Input C:\Users\semih\Desktop\Master\colgen\tek boyutlu sütun oluşturma yönt
emi2.gms
Output C:\windows\gamsdir\tek boyutlu sütun oluşturma yöntemi2.lst

EK D: X Firması Çözüm3 için GAMS Uygulama Modeli

```
$ontext
X Firması ait Çeşitli Tipteki Stok Malzemeleriyle Bir Boyutlu Stok Kesme Problemi
(Yerleşim planları sütun oluşturma yöntemi ile oluşturulmaktadır.)
$offtext
set  i 'en' /en1*en19/;
set  k 'malzemeeni'  /malzeme1*malzeme5/;
*----------------------------------------------------
* Data
*----------------------------------------------------
table siparistalepdata(i,*)
          en      talep    maxtalep
en1      220       28       36
en2      245       13       15
en3      390       42       44
en4      465       80       81
en5      520       34       36
en6      550       66       66
en7      580       21       23
en8      600       67       69
en9      620       67       69
en10     660       71       73
en11     670       68       70
en12     720       23       24
en13     800       40       43
en14     900       67       69
en15     960       19       20
en16     970       95      100
en17     1060      19       21
en18     1260      39       42
en19     1360      36       38
 ;
```

```
table malzemestokdata(k,*)
            minmiktar   maxmiktar
  malzeme1         0        190
  malzeme2         0        155
  malzeme3         0        123
  malzeme4         0        120
  malzeme5         0        100
 ;
parameter rk(k)
/ malzeme1 2000, malzeme2 1600 , malzeme3 1500 , malzeme4 1300
malzeme5 1250
/;
parameter h(k)
/ malzeme1 9.1, malzeme2 9.2, malzeme3 8.9 , malzeme4 9.0, malzeme5 9.2
/;
parameter t(k)
/ malzeme1 0, malzeme2 0, malzeme3 0.3, malzeme4 0.3, malzeme5 0
/;
parameter c(k);
c(k) = h(k) + t(k) ;
parameter w(i);
w(i) = siparistalepdata(i,'en');

parameter d(i);
d(i) = siparistalepdata(i,'talep');
parameter maxd(i);
maxd(i) = siparistalepdata(i,'maxtalep');
parameter m(k);
m(k) = malzemestokdata(k,'minmiktar');
parameter maxm(k);
maxm(k) = malzemestokdata(k,'maxmiktar');
set p 'Olası Yerleşim Planı Sayısı' /p1*p10000/;
set iter 'Maksimum Iterasyon Sayısı' /iter1*iter30/;
*---------------------------------------------------
* Ana Model
```

```
*---------------------------------------------------
parameter aip(i,p) 'p boyutunda matriste karşılanan talep miktarı' ;
parameter firedegeri(k,p) 'p boyutunda matriste artan Fire Değeri ';
parameter akp(k,p) 'p boyutunda matriste kullanılan talep miktarı';
integer variable xp(p) 'kullanılan yerleşim planı sayısı';
variable z 'amaç değişkeni';
xp.up(p) = sum(i, d(i));

equations
  master_maliyet   'Toplam maliyet'
  master_talepmin(i) 'sipariş talep alt limit miktarı'
  master_talepmax(i) 'sipariş talep üst limit miktarı'
  master_malzememin(k) 'stok malzemesine ait maximum malzeme adedi'
  master_malzememax(k) 'stok malzemesinin kullanılması gereken minimum adedi'
;
set pp(p) 'dynamic subset';
master_maliyet..     z =e= sum((k,pp),c(k) * akp(k,pp)* xp(pp)) + sum((k,pp), h(k)  *
firedegeri(k,pp)* xp(pp)) ;
master_talepmin(i)..  sum(pp, aip(i,pp)*xp(pp)) =g= d(i);
master_talepmax(i)..sum(pp, aip(i,pp)*xp(pp)) =l= maxd(i);
master_malzememin(k)..  sum(pp, akp(k,pp)*xp(pp)) =g= m(k);
master_malzememax(k)..sum(pp, akp(k,pp)*xp(pp)) =l= maxm(k);
model                                                                master
/master_maliyet,master_talepmin,master_talepmax,master_malzememin,master_mal
zememax/;
master.solprint = 2;
master.limrow = 0;
master.limcol = 0;

*---------------------------------------------------
* Alt Model
*---------------------------------------------------
integer variable
  y(i)  'yeni yerleşim planı'  ;
binary variable
   my(k) 'stok malzemesi kullanılıyor mu kullanılmıyor mu?';
```

```
positive variable
   s(k) 'yeni yerleşim planındaki fire eni ';
equations
   knapsack_obj
   knapsack_constraint
   knapsack_rawmaterial
   knapsack_scrap1
;
knapsack_obj..              z  =e=  sum(k,  c(k)*my(k))+  sum(k,  c(k)*s(k))    -  sum(i, master_talepmin.m(i)*y(i))-sum(i, master_talepmax.m(i)*y(i))
                              -   sum(k,   master_malzememin.m(k)*my(k))-sum(k, master_malzememax.m(k)*my(k))  ;
knapsack_constraint..  sum(i, w(i)*y(i)) =l= sum(k, rk(k)*my(k)) ;
knapsack_rawmaterial..   sum(k, my(k)  )=e= 1;
knapsack_scrap1..     sum(i, w(i)*y(i)) =e= sum(k, rk(k)*my(k) - s(k))  ;

model knapsack /knapsack_obj,knapsack_constraint,knapsack_rawmaterial,knapsack_scrap1/;
knapsack.solprint = 2;
knapsack.optcr = 0;
knapsack.limrow = 0;
knapsack.limcol = 0;
*-----------------------------------------------------
* başlangıç çözümü için pp kümesi ve aip matrisini kullan
*-----------------------------------------------------
set pi(p);
pi('p1') = yes;
loop(k,
  loop(i,
    akp(k,pi) = 1;
    aip(i,pi) = floor(rk(k)/w(i));
    firedegeri(k,pi)=rk(k)- aip(i,pi)* w(i);
    pp(pi) = yes;
    pi(p) = pi(p-1);
    );
);
```

```
display "------------------------------------------------------------",
        "Başlangıç Değeri",
        "------------------------------------------------------------",
        pp,aip,akp,firedegeri;
scalar done /0/;
scalar iteration;
loop(iter$(not done),
*
* Ana Modeli Çöz
*
   solve master using rmip minimizing z;
   display "------------------------------------------------------------",
              "duals",
              "------------------------------------------------------------",
              master_talepmin.m,master_talepmax.m;
*
* Alt Modeli Çöz
*
   solve knapsack using mip minimizing z;

*
* Yeni Yerleşim Planı Var  mı?
*
   if(z.l < -0.001,
      aip(i,pi) = y.l(i);
      akp(k,pi) = my.l(k);
      firedegeri(k,pi) = s.l(k);
      pp(pi) = yes;
      iteration = ord(iter);
      display "------------------------------------------------------------",
              iteration,
              "------------------------------------------------------------",
              pp,aip,akp,firedegeri;
      pi(p) = pi(p-1);
```

```
  else
    done = 1;
  );
);
display y.l,s.l;

abort$(not done) "Too many iterations.";

*
* Final Tam Sayılı Programlamayı Çöz
*
display "-----------------------------------------------------------",
      "Final MIP",
      "-----------------------------------------------------------";
master.optcr=0;
solve master using mip minimizing z;
display z.l,xp.l;
parameter pat(*,*) "yerleşim planı sayısı";
pat(i,p)$(xp.l(p)>0.1) = aip(i,p);
pat('Count',p) = xp.l(p);
pat(i,'Toplam')= sum(p, aip(i,p)*round(xp.l(p)));
pat('Count','Toplam') = sum(p,pat('Count',p));
display pat;
parameter mpat(*,*) "stok malzemesi kullanım sayısı";
mpat(k,p)$(xp.l(p)>0.1) = akp(k,p);
mpat('s',p)$(xp.l(p)>0.1) = sum(k,firedegeri(k,p));
mpat('Count',p) = round(xp.l(p));
mpat(k,'Toplam')= sum(p, akp(k,p)*round(xp.l(p)));
mpat('scrap','Toplam') = sum(p,sum(k,firedegeri(k,p))*pat('Count',p));
mpat('Count','Toplam') = sum(p,pat('Count',p));
display mpat;
```

EK E: X Firması GAMS Uygulama Modelinin Çözüm3 için Sonuçları

GAMS Rev 141 Intel /MS Window 09/13/10 10:35:39 Page 124

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

E x e c u t i o n

---- 231 VARIABLE z.L = 26617.100 amaç değişkeni

---- 231 VARIABLE xp.L kullanılan yerleşim planı sayısı

p10 16.000, p32 12.000, p96 19.000, p98 14.000, p101 67.000
p105 10.000, p106 7.000, p111 16.000, p112 8.000, p114 41.000
p115 23.000, p117 38.000, p118 28.000, p119 21.000, p120 21.000
p121 2.000, p123 9.000, p124 16.000

---- 238 PARAMETER pat yerleşim planı sayısı

	p10	p32	p96	p98	p101	p105
en3			1.000			
en5						2.000
en6			1.000			
en8					1.000	
en10	3.000			1.000		
en11				2.000		
en13		2.000				
en14					1.000	
en15						1.000
en17			1.000			
Count	16.000	12.000	19.000	14.000	67.000	10.000

+	p106	p111	p112	p114	p115	p117
en1		1.000				
en2	1.000		1.000			
en3	1.000		1.000			
en4						1.000

en5		1.000				
en6						1.000
en9				1.000		
en12					1.000	
en15	1.000					
en16				1.000		1.000
en18		1.000			1.000	
en19			1.000			
Count	7.000	16.000	8.000	41.000	23.000	38.000

+	p118	p119	p120	p121	p123	p124
en1				2.000		1.000
en3					1.000	
en4			2.000			
en6					1.000	
en7		1.000				
en8				1.000		
en9	1.000					
en10					1.000	
en11		1.000	1.000			
en13						1.000
en15				1.000		
en16						1.000
en19	1.000					
Count	28.000	21.000	21.000	2.000	9.000	16.000

+	Toplam
en1	36.000
en2	15.000
en3	43.000
en4	80.000
en5	36.000
en6	66.000

en7	21.000
en8	69.000
en9	69.000
en10	71.000
en11	70.000
en12	23.000
en13	40.000
en14	67.000
en15	19.000
en16	95.000
en17	19.000
en18	39.000
en19	36.000
Count	368.000

---- 248 PARAMETER mpat stok malzemesi kullanım sayısı

	p10	p32	p96	p98	p101	p105
malzeme1	1.000		1.000	1.000		1.000
malzeme2		1.000				
malzeme3					1.000	
Count	16.000	12.000	19.000	14.000	67.000	10.000
s	20.000					

+	p106	p111	p112	p114	p115	p117
malzeme1		1.000	1.000		1.000	1.000
malzeme2	1.000			1.000		
Count	7.000	16.000	8.000	41.000	23.000	38.000
s	5.000		5.000	10.000	20.000	15.000

+	p118	p119	p120	p121	p123	p124
malzeme1	1.000			1.000		1.000
malzeme2			1.000		1.000	
malzeme5		1.000				

Count 28.000 21.000 21.000 2.000 9.000 16.000

s 20.000 10.000

+ Toplam

malzeme1	190.000
malzeme2	90.000
malzeme3	67.000
malzeme5	21.000
Count	368.000
scrap	2555.000

EK F: X Firması Çözüm4 için GAMS Uygulama Modeli

```
$ontext
X Firması ait Çeşitli Tipteki Stok Malzemeleriyle Bir Boyutlu Stok Kesme Problemi
(Yerleşim planları sütun oluşturma yöntemi ile oluşturulmaktadır.)

$offtext
set  i 'en' /en1*en14/;
set  k 'malzemeeni'  /malzeme1*malzeme5/;
*---------------------------------------------------
* Data
*---------------------------------------------------
table siparistalepdata(i,*)

         en      talep   maxtalep
en1      450      38      42
en2      455      19      21
en3      460      52      55
en4      480      82      85
en5      550      128     133
en6      580      38      41
en7      610      18      20
en8      760      9       10
en9      800      54      56
en10     840      69      73
en11     885      7       8
en12     890      83      88
en13     1010     86      89
en14     1200     62      66

 ;
table malzemestokdata(k,*)
             minmiktar   maxmiktar
```

```
  malzeme1       0      190
  malzeme2       0      155
  malzeme3       0      123
  malzeme4       0      120
  malzeme5       0      100
 ;
parameter rk(k)
/ malzeme1 2000, malzeme2 1600 , malzeme3 1500 , malzeme4 1300
malzeme5 1250
/;
parameter h(k)
/ malzeme1 9.1, malzeme2 9.2, malzeme3 8.9 , malzeme4 9.0, malzeme5 9.2
/;
parameter t(k)
/ malzeme1 0, malzeme2 0, malzeme3 0.3, malzeme4 0.3, malzeme5 0
/;
parameter c(k);
c(k) = h(k) + t(k) ;
parameter w(i);
w(i) = siparistalepdata(i,'en');
parameter d(i);
d(i) = siparistalepdata(i,'talep');
parameter maxd(i);
maxd(i) = siparistalepdata(i,'maxtalep');
parameter m(k);
m(k) = malzemestokdata(k,'minmiktar');
parameter maxm(k);
maxm(k) = malzemestokdata(k,'maxmiktar');
set p 'Olası Yerleşim Planı Sayısı' /p1*p10000/;
set iter 'Maksimum Iterasyon Sayısı' /iter1*iter30/;

*---------------------------------------------------
* Ana Model
*---------------------------------------------------
```

```
parameter aip(i,p) 'p boyutunda matriste karşılanan talep miktarı' ;
parameter firedegeri(k,p) 'p boyutunda matriste artan Fire Değeri ';
parameter akp(k,p) 'p boyutunda matriste kullanılan talep miktarı';
integer variable xp(p) 'kullanılan yerleşim planı sayısı';
variable z 'amaç değişkeni';
xp.up(p) = sum(i, d(i));
equations
   master_maliyet   'Toplam maliyet'
   master_talepmin(i) 'sipariş talep alt limit miktarı'
   master_talepmax(i) 'sipariş talep üst limit miktarı'
   master_malzememin(k) 'stok malzemesine ait maximum malzeme adedi'
   master_malzememax(k) 'stok malzemesinin kullanılması gereken minimum adedi'
;
set pp(p) 'dynamic subset';
master_maliyet..     z =e= sum((k,pp),c(k) * akp(k,pp)* xp(pp)) + sum((k,pp), h(k)  *
firedegeri(k,pp)* xp(pp)) ;
master_talepmin(i)..  sum(pp, aip(i,pp)*xp(pp)) =g= d(i);
master_talepmax(i)..sum(pp, aip(i,pp)*xp(pp)) =l= maxd(i);
master_malzememin(k)..  sum(pp, akp(k,pp)*xp(pp)) =g= m(k);
master_malzememax(k)..sum(pp, akp(k,pp)*xp(pp)) =l= maxm(k);

model master

/master_maliyet,master_talepmin,master_talepmax,master_malzememin,master_mal
zememax/;
master.solprint = 2;
master.limrow = 0;
master.limcol = 0;
*----------------------------------------------------
* Alt Model
*----------------------------------------------------

integer variable
   y(i)  'yeni yerleşim planı'  ;
binary variable
    my(k) 'stok malzemesi kullanılıyor mu kullanılmıyor mu?';
```

```
positive variable
   s(k) 'yeni yerleşim planındaki fire eni ';
equations
   knapsack_obj
   knapsack_constraint
   knapsack_rawmaterial
   knapsack_scrap1
;

knapsack_obj..              z  =e=  sum(k,  c(k)*my(k))+  sum(k,  c(k)*s(k))    -  sum(i,
master_talepmin.m(i)*y(i))-sum(i, master_talepmax.m(i)*y(i))
                                    -      sum(k,      master_malzememin.m(k)*my(k))-sum(k,
master_malzememax.m(k)*my(k))  ;
knapsack_constraint..  sum(i, w(i)*y(i)) =l= sum(k, rk(k)*my(k)) ;
knapsack_rawmaterial..   sum(k, my(k)  )=e= 1;
knapsack_scrap1..     sum(i, w(i)*y(i)) =e= sum(k, rk(k)*my(k) - s(k))  ;

model                                                                          knapsack
/knapsack_obj,knapsack_constraint,knapsack_rawmaterial,knapsack_scrap1/;

knapsack.solprint = 2;
knapsack.optcr = 0;
knapsack.limrow = 0;
knapsack.limcol = 0;
*---------------------------------------------------
* başlangıç çözümü için pp kümesi ve aip matrisini kullan
*---------------------------------------------------
set pi(p);
pi('p1') = yes;
loop(k,
  loop(i,
    akp(k,pi) = 1;
    aip(i,pi) = floor(rk(k)/w(i));
    firedegeri(k,pi)=rk(k)- aip(i,pi)* w(i);
    pp(pi) = yes;
    pi(p) = pi(p-1);
```

```
    );
);

display "----------------------------------------------------------",
        "Başlangıç Değeri",
        "----------------------------------------------------------",
        pp,aip,akp,firedegeri;
scalar done /0/;
scalar iteration;
loop(iter$(not done),
*
* Ana Modeli Çöz
*
   solve master using rmip minimizing z;
   display "----------------------------------------------------------",
            "duals",
            "----------------------------------------------------------",
            master_talepmin.m,master_talepmax.m;
*
* Alt Modeli Çöz
*
   solve knapsack using mip minimizing z;

*
* Yeni Yerleşim Planı Var  mı?
*
   if(z.l < -0.001,
      aip(i,pi) = y.l(i);
      akp(k,pi) = my.l(k);
      firedegeri(k,pi) = s.l(k);

      pp(pi) = yes;
      iteration = ord(iter);
      display "----------------------------------------------------------",
              iteration,
```

```
            "------------------------------------------------------------",
            pp,aip,akp,firedegeri;
      pi(p) = pi(p-1);

   else
      done = 1;
   );
);
display y.l,s.l;
abort$(not done) "Too many iterations.";

*
* Final Tam Sayılı Programlamayı Çöz
*
display "------------------------------------------------------------",
        "Final MIP",
        "------------------------------------------------------------";
master.optcr=0;
solve master using mip minimizing z;
display z.l,xp.l;
parameter pat(*,*) "yerleşim planı sayısı";
pat(i,p)$(xp.l(p)>0.1) = aip(i,p);
pat('Count',p) = xp.l(p);
pat(i,'Toplam')= sum(p, aip(i,p)*round(xp.l(p)));
pat('Count','Toplam') = sum(p,pat('Count',p));
display pat;
parameter mpat(*,*) "stok malzemesi kullanım sayısı";
mpat(k,p)$(xp.l(p)>0.1) = akp(k,p);
mpat('s',p)$(xp.l(p)>0.1) = sum(k,firedegeri(k,p));
mpat('Count',p) = round(xp.l(p));
mpat(k,'Toplam')= sum(p, akp(k,p)*round(xp.l(p)));
mpat('scrap','Toplam') = sum(p,sum(k,firedegeri(k,p))*pat('Count',p));
mpat('Count','Toplam') = sum(p,pat('Count',p));
display mpat;
```

EK G: X Firması GAMS Uygulama Modelinin Çözüm4 için Sonuçları

```
----    218 ----------------------------------------------------------
        Final MIP
        ----------------------------------------------------------
GAMS Rev 141  Intel /MS Window                09/15/10 02:47:20 Page 88
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
E x e c u t i o n
----    224 VARIABLE z.L              =    24833.000  amaç değişkeni

----    224 VARIABLE xp.L  kullanılan yerleşim planı sayısı

p73  1.000,    p74  3.000,    p75 49.000,    p76 59.000,    p77 20.000
p80  3.000,    p81 76.000,    p83 10.000,    p84 11.000,    p85 56.000
p86 10.000,    p87  7.000,    p88 16.000,    p89  4.000,    p90  6.000

----    231 PARAMETER pat  yerleşim planı sayısı

         p73       p74       p75       p76       p77       p80
en3                        1.000                      1.000
en4                                                 2.000
en5                                1.000
en6       2.000                                       1.000
en7                                        1.000
en8                1.000
en10       1.000     1.000      1.000
en12                                2.000      1.000
Count      1.000      3.000     49.000     59.000     20.000      3.000

   +      p81       p83       p84       p85       p86       p87

en1                        2.000
en2                2.000
en4       1.000
```

```
en5                                                  2.000
en6                1.000      1.000               1.000
en9                                 1.000
en11                                                 1.000
en13       1.000                                1.000
en14                                 1.000
Count      76.000      10.000      11.000      56.000      10.000       7.000

   +       p88       p89        p90     Toplam

en1        1.000                      38.000
en2                                20.000
en3                                52.000
en4                                82.000
en5                               132.000
en6                1.000              40.000
en7                                20.000
en8                        1.000       9.000
en9                                56.000
en10       1.000                       69.000
en11                                 7.000
en12                1.000              83.000
en13                                86.000
en14                        1.000      62.000
Count      16.000       4.000      6.000     331.000

----    241 PARAMETER mpat  stok malzemesi kullanım sayısı
            p73        p74       p75        p76       p77        p80

malzeme1         1.000                       1.000              1.000
malzeme2                 1.000
malzeme3                                          1.000
malzeme4                          1.000
Count         1.000      3.000      49.000      59.000      20.000      3.000
s                                   10.000
```

+	p81	p83	p84	p85	p86	p87
malzeme1			1.000		1.000	
malzeme2				1.000		
malzeme3	1.000	1.000	1.000			
Count	76.000	10.000	11.000	56.000	10.000	7.000
s	10.000	10.000	20.000		10.000	15.000

+	p88	p89	p90	Toplam
malzeme1			1.000	132.000
malzeme2				13.000
malzeme3		1.000		121.000
malzeme4	1.000			65.000
Count	16.000	4.000	6.000	331.000
s	10.000	30.000	40.000	
scrap				2395.000

EXECUTION TIME = 0.016 SECONDS 3.5 Mb WIN216-141 Jan 24, 2005

USER: Centre Versailles - Grigon S021106:1507AB-WIN
I.N.R.A. DC2726
License for teaching and research at degree granting institutions
**** FILE SUMMARY

Input C:\Users\semih\Desktop\tez son olabilir2.gms
Output C:\windows\gamsdir\tez son olabilir2.lst

ÖZGEÇMİŞ

Ad Soyad: Semih Adakcı

Doğum Yeri ve Tarihi: Manisa 12.02.1983

Adres: Klavuzçayır Cad. Polis Mahmut Sok. Sabır Apt. No:18/2 Maltepe/İstanbul

Lisans Üniversite: Yıldız Teknik Üniversitesi