

34805

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİR PAFTA ÜZERİNE İKİ BOYUTLU
PARÇALARIN MINIMUM İSKARTA İLE
BİLGİSAYAR DESTEKLİ YERLEŞTİRİLMESİ**

Mak.Müh. Yunus Emre TOKMAK

**F.B.E. Makina Mühendisliği Anabilim Dalı İmal Usulleri Programında
hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Doç.Dr. Erhan ALTAN

**T.C. YÜKSEKÖĞRETİM BAKANLIĞI
DOKÜMANİZASYON MERKEZİ**

İSTANBUL, 1994

İÇİNDEKİLER

ÖZET	IV
SUMMARY	VI
ÖNSÖZ	VIII
1. GİRİŞ	1
2. BİR PAFTA ÜZERİNE, GEOMETRİK ŞEKLİ BİLİNER PARÇALARIN YERLEŞTİRİLMESİ PROBLEMİNİN TANIMLANMASI.....	4
2.1. Malzeme Yerleştirme Problemi.....	4
2.2. Çok Kullanılan Temel Kavramlar	4
2.2.1. Parça Yerleştirme Planı	4
2.2.2. Kesme Kayıpları	7
2.2.3. Kesme Probleminin Karmaşıklığı	7
2.2.4. Kesme Probleminin Boyutu	7
2.2.5. İhtiyaç Duyulan Teorik Ana Malzeme Miktarı.....	7
2.2.6. Kullanılacak Minimum Ana Malzeme Miktarının Tespiti.....	8
2.2.7. Kayıp Miktarı.....	8
2.2.8. Malzeme Kullanım Etkinliği	9
2.3. Yerleştirme Problemlerinin Boyutlarına Göre Sınıflandırılması.....	10
2.4. Yerleştirme Problemlerinin Kısıtları.....	13
2.4.1. Malzeme Kısıtları.....	14
2.4.2. Teknolojik Kısıtlar	14
2.4.3. Yöntemsel Kısıtlar	15
2.5. Yerleştirme Problemlerinde Benimsenebilir Amaçlar.....	16
2.5.1. Kesme Kaybı Ölçütü.....	16
2.5.2. Malzeme Miktarı Ölçütü	17
2.5.3. Toplam Maliyet Ölçütü	17
2.5.4. Kapasite Kullanım Ölçütü.....	17

3. YERLEŐTİRME PROBLEMLERİNİN MODELLENMESİ.....	19
3.1. Bir Boyutlu Yerleőtirme	
Probleminin Modellenmesi.....	20
3.2. İki Boyutlu Yerleőtirme Probleminin	
Modellenmesi.....	21
3.3. Üç Boyutlu Yerleőtirme Probleminin	
Modellenmesi.....	22
4. YERLEŐTİRME PROBLEMLERİ ÇÖZÜM	
YÖNTEMLERİ.....	24
5. YERLEŐTİRME PROBLEMLERİ İÇİN MEVCUT	
ÇALIŐMALARIN İRDELENMESİ.....	28
5.1. Dikdörtgen Şekilli Parçaların	
Yerleőtirilmesi İle İlgili	
Çalıőmalar.....	29
5.1.1. Gilmore - Gomory Yöntemi.....	29
5.1.1.1. İki Boyutlu Kesme	
Problemlerinde	
Gilmore - Gomory Yöntemi.....	30
5.2. Giyotin Kesim Problemleri İle	
İlgili Çalıőmalar.....	31
5.3. Düzensiz Şekillerin Yerleőtirilmesi	
İle İlgili Çalıőmalar.....	34
6. DÜZENSİZ ŞEKİLLERİN YERLEŐTİRİLMESİ İLE	
İLGİLİ GELİŐTİRİLEN BİLGİSAYAR PROGRAMI.....	44
6.1. Çözüm Yaklaőımı.....	46
6.1.1. Şekillerin Hazırlanması.....	47
6.1.2. Geometrik Şekle Ait	
Bilgilerin Üretilmesi.....	50
6.1.2.1. Geometrik Şeklin	
Alanının Hesaplanması.....	50
6.1.2.2. Geometrik Şeklin Kenar	
Uzunluklarının	
Hesaplanması.....	51

6.1.2.3. Geometrik Şeklin	
Kenarlarının Pozitif X	
Ekseni İle Yaptığı	
Açıların Hesaplanması.....	52
6.1.2.4. Şeklin Kenarları	
Arasındaki İç	
Açıların Hesaplanması.....	54
6.1.2.5. Geometrik Şeklin Konumunun	
Değiştirilmesi	55
6.1.2.5.1 Şeklin Ötelenmesi.....	55
6.1.2.5.2 Şeklin Döndürülmesi	56
6.1.2.5.3 Şeklin Katlanması.....	58
6.1.2.6. Geometrik Şeklin	
Simetrikliğinin	
Belirlenmesi	60
6.1.3 Şeklin Sığıdığı En Küçük	
Dikdörtgene Ait	
Boyutların Alanın ve	
Karmaşıklık Değerinin	
Hesaplanması	62
6.1.3.1. Şeklin Sığıdığı En Küçük	
Dikdörtgene Ait Boyutların	
Ve Alanın Hesaplanması.....	62
6.1.3.2..Karmaşıklık Değerinin	
Hesaplanması	64
6.2. Kullanılan Algoritma	64
7.SONUÇ.....	68
KAYNAKLAR.....	70
EKLER:	
Hazırlanan Bilgisayar Programı.....	74
ÖZGEÇMİŞ	

ÖZET

Nesting veya dilimizdeki karşılığı ile boyutları ve talep adetleri belirli parçaların , yine boyutları belirli paftalar üzerine yerleştirilmesi, endüstrinin bir çok dalında karşılaşılan yaygın bir problemdir. Tanımlanan ,elde mevcut plakalardan istenen malzemenin en ekonomik olarak kesilmesi temel yaklaşımdır.

İki boyutlu yerleştirme problemlerine verilebilecek bazı örnekler bir çok endüstri dalında bulunabilir. İsmen belirtirsek, havacılık ve uzay, gemi yapımı, çelik konstrüksiyon, ayakkabı imalatı, tekstil, düz cam ve mobilya imalatı v.s. gibi.

Bazı uygulamalarda, hasarlı bölgeleri olan büyük plakaların kullanılma mecburiyeti vardır ve parçaların, bu hasarlı bölgeler üzerine yerleştirilmemesi gerekir. Örneğin, takım elbise dikilecek bir kumaş parçasında böyle bir hata bulunmamalıdır. Dikdörtgen parçaların kesilmesinde de benzer problemler geçerlidir. Örneğin, çelik plakaların, düz camın veya tahtanın kesilmesi, kutu yapımı için kağıt profillerin kesilmesi v.b.

Kesme problemlerini aşağıdaki üç özelliğe göre sınıflandırabiliriz :

- a. Boyut (Bir boyutlu, iki boyutlu, üç boyutlu kesme problemleri)
- b. Plaka sayısı (Tek plakanın işlendiği problemler veya birden fazla plakanın işleme sokulduğu problemler)
- c. Paftaların geometrik şekilleri (Dikdörtgen paftalar veya düzensiz şekilli paftalar)

Geçen bir kaç yıl içinde, bazı araştırmacılar düzenli veya düzensiz şekillerin yerleştirilmesi ile ilgili problemler ile ilgilendiler ve hazırladıkları çözüm yaklaşımlarını üç kategoride topladılar.

- a. El ile yapılan yerleştirme
- b. Yarı otomatik yerleştirme
- c. Otomatik yerleştirme

Parçaların yerleştirme işlemi temel olarak el ile yapılır. Her ne kadar bilgisayar kullanılsada, bilgisayar verimliliğin ve doğruluk değerinin artırılması için kullanılır. Yerleştirilecek parçaların tanımlanması bilgisayar ile yapılır. Bir Plotter parçaların kopyalarını çizer. Parçaları örnekleyen modeller kesilir ve paftanın ebatlarına ve şekline uygun bir çerçeve içine el ile yerleştirilir.

Parçaların yerleştirme işlemi kullanılan etkileşimli bir grafik ekran üzerinde tamamlanır. Yerleştirilecek parçalar ve kullanılacak pafta, ekran üzerinde görüntülenir. Kullanıcı, çeşitli giriş ünitelerini (Mouse, Digitizer, Joystic v.b) kullanarak parçaları çevirebilir veya bir noktadan başka bir noktaya taşıyabilir. Yerleştirme işlemi süresince bilgisayar, parçaların üst üste binip binmediğini, fire miktarını ve diğer ortaya çıkması muhtemel hataları kontrol eder.

Otomatik olarak yerleştirme yapan çeşitli metodlar geliştirilmiştir. Fakat, birçok yönden en iyi neticeleri vermelerine rağmen işlem zamanının uzun olması ve hesaplama süresinin maliyetinin çok yüksek olması nedeni ile kullanılmamaktadırlar.

Bu çalışmada düzensiz şekillerin yerleştirilmesi esas olarak alınmıştır. İki aşamalı bir algoritma üzerinde durulmuştur. Birinci aşamada, herhangi bir CAD sisteminde çizilen parçanın geometrisi okunmuş ve bu geometri etrafına bir dikdörtgen zarf tanımlanmıştır. İkinci aşamada ise hesaplanan dikdörtgenlerin doğrusal programlama ile ana pafta üzerine yerleştirilmesine çalışılmıştır.

SUMMARY

Nesting is a common problem that is frequently encountered in a number of industries where raw material has to be cut economically from a given stock sheet.

Some examples of two dimensional allocation problems can be found in many industries, namely, aerospace, shipbuilding, steel construction, shoe manufacturing, clothing, flat glass and furniture.

In some applications, the large sheet to be used contains defective areas, and the allocated pieces must not overlap these defects. Such problems appear, for example, in stamping parts from steel strips, cutting cloth for suits, etc. Other similar cutting problems involve the cutting of rectangular pieces as, for example, the cutting of steel, wood or glass plates, the cutting of paper board for the production of boxes, and several other applications involving cutting operations.

The cutting stock problems can be classified with respect to the following three attributes

- a. Dimension (One-dimensional versus multi-dimensional)
- b. Number of plates (Single-plate versus multi-plate)
- c. Shape of patterns. (Rectangular versus irregular)

In recent years, a number of researchers have investigated the problem of the allocation of irregular shapes and the proposed approaches can be grouped in three categories :

- a. Manual
- b. Semi automatic
- c. Automatic

Nesting is handled with a system which is essentially manual, although a computer is used to improve accuracy and efficiency. Descriptions of the pieces are processed and filed by the computer. A plotter produces copies of the pieces. The pieces are cut out and arranged manually into a frame which corresponds the stock sheet.

Some methods might be classified as semi-automatic. Nesting is carried out using an interactive graphics device. The pieces and the sheet are displayed on the screen. The operator can rotate the pieces and move them around the screen by using some input device. During the nesting the computer checks overlapping and other possible errors and calculates the amount of waste

Methods for automatic nesting have been developed. But in most cases they have been marginally useful since they have proved too expensive in terms of computer time.

In this study, the main subject is to allocate the irregular shapes on to stock sheets. I want to use two stage methods for these operations. First stage, I used a CAD system for to design geometry of shapes. After that, I calculated a rectangular area around of the shape. Second stage, I used an LP based algorithms for allocate the shape on the stock sheet.



ÖNSÖZ

Bir pafta üzerine iki boyutlu parçaların minimum ıskarta ile bilgisayar destekli yerleştirilmesi konusunda çalışmama imkan sağlayan ve araştırmalarım esnasında , bulduğum farklı dökümanların temininde büyük yardımlarını gördüğüm sayın hocam Doç. Dr. Erhan ALTAN beye teşekkürü bir borç bilirim.

Ayrıca tüm çalışmalarına gerekli olan zaman ve teknolojik yardımı benden esirgemeyen ve halen bünyesinde görev yaptığım REİS CNC TAKİM TEAZGAHLARI ve PROGRAMLAMA SİSTEMLERİ LTD. ŞTİ. sahibi sayın Yüksek Elektrik Mühendisi Turgut YÜTMEN beye de teşekkürlerimi sunarım.

Şubat 1994
İSTANBUL

Yunus Emre TOKMAK

1. GİRİŞ

Nesting, boyutları, talep adetleri belirli, düzlem şekilli, basit ve/veya karmaşık profilli, küçük ebatlı, mamül veya yarı mamül parçaların, üretim, ambalaj v.b operasyonlarda kullanılmak üzere, yine boyutları belirli, düzlem şekilli, işlenmeye hazır paftalar üzerine en az fire veya maksimum parça adedi cıde cılcecek şekilde yerleştirilmesi işlemidir.

Nesting, endüstrinin bir çok dalında tasarımcıların yaygın olarak karşılaştıkları bir problemdir. İsmen belirtirsek, kağıt , cam, tekstil, çelik konstrüksiyon, düz cam imalatı, mobilya endüstrisi, uçak, gemi yapım endüstrilerinde, yerleştirme problemlerine örnek olarak gösterilecek bir çok uygulama ile karşılaşmaktadır.

Endüstrinin çeşitli dallarında, farklı tarzlarda karşılaşılan uygulamaların gerektirdiği, işlemlerin tipine özel, bir takım karakteristik kısıtlamalar söz konusudur.

Örneğin;

İki boyutlu metal parçaların, plakalardan kesilmesi probleminde, kullanılacak olan kesme düzeneğine ait kısıtlar.

Bazı uygulamalarda, hasarlı bölgeleri olan büyük paftaların kullanılma mecburiyeti vardır ve parçaların söz konusu hasarlı bölgeler üzerine yerleştirilmemesi gerekir.

Tekstil veya mobilya imalatında kullanılacak kumaş veya ahşap malzemenin kesilmesi esnasında bu kısıt çok önemlidir. Aynı kısıt, çelik plakaların, düz camın, kutu yapımı için kağıt profillerin kesilmesi v.b durumlar içinde geçerlidir.

Bahsi geçen tüm ilave kısıtlara rağmen, Nesting işlemlerinde temel prensip, parçaların en küçük kayıp oranını verecek şekilde veya en fazla kullanılabilir parça elde edilecek şekilde, ana pafta üzerine yerleştirilmesi ve bu şekilde kesme işleminin yapılmasıdır

Yarı mamül, düzlem veya silindirik malzemeler üzerine çeşitli biçimlerde geometrik şekillerin yerleştirilmesi ve kesilmesi ile ilgili problemler, çoğu kez işleme uygun olarak, malzeme kesme (stock cutting) problemleri adıyla bilinmektedir.

Pratikte uygulanan çözüm şekli, deneyimli bir tasarımcının, işlemi başından sonuna kadar kişisel bilgi birikimi, sezgileri ve deneylere bağlı yöntemler yardımıyla veya şablonlar vasıtası ile sınaama yöntemi kullanarak yürütmesi şeklinde görülmektedir. Bu yöntemin sezgisel olarak önerilen ve uygulanan bir yap boz bilmecesinden pek bir farkı yoktur. Şahsın ortaya koyduğu çözümler nadiren optimum değerlerdedir. Bazen de bu çözümler hayret uyandıracak kadar iyi olabilir. Bu şekilde planlama yapmak, hem çok zaman alır hemde pahalıdır. Operasyon ilerledikçe işlem daha da zorlaşmakta, hatta deneyimli tasarımcıları bile uzun süre düşündürmektedir.

Uygulama alanlarında karşılaşılabilen herhangi bir kesme probleminin çözümünü yapabilecek genel amaçlı çözüm yaklaşımları yoktur. Mevcut çözüm yaklaşımlarında, analitik veya sezgisel yaklaşımlar kullanılmaktadır. Sezgisel yaklaşımlar, genel itibarı ile probleme bağımlıdır.

Dikdörtgen şekilli, benzer , düzenli veya düzensiz parçaların yerleştirilmesi için bilgisayar ile çözümler hakkında, son zamanlarda çeşitli makaleler yayınlanmıştır. Bilgisayar destekli yerleşim planlarının kullanılmasındaki ana amaç, hesaplama zamanı müddetince etkili çözümlerin bulunacak olmasıdır. Mümkün olan sonsuz sayıda düzenlemeyi aramak ne ekonomik olarak ne de tatbik edilebilirlik açısından uygundur.

Yerleştirme problemlerinin çözümleri için veya potansiyel uygulamalara uygun, bir kaç ticari amaçlı CAD/CAM paketi piyasada mevcuttur.

Örneğin :

* ComputerVision firmasının hazırladığı

Dynanest programı

* Auto-trol firmasının hazırladığı

Automatic Nesting System programı

* Intergraph firmasının hazırladığı

PINS programı

* Prime firmasının hazırladığı

Autokon programı

bu tür yazılımlara örnek olarak gösterilebilir.



2. BİR PAFTA ÜZERİNE, GEOMETRİK ŞEKLİ BİLİNER PARÇALARIN YERLEŞTİRİLMESİ PROBLEMİNİN TANIMLANMASI

2.1 MALZEME YERLEŞTİRME PROBLEMİ

Ebatları çok büyük olan (örneğin, rulo şeklindeki) veya boyutları belirli (örneğin eni ve boyu belirli dikdörtgen veya düzensiz geometrili plakalar) ana pafta üzerine, daha küçük, basit ve/veya karmaşık şekilli, çeşitli boyutlara sahip, düzlemsel geometrilerin yerleştirilmesi işlemi ve bu amaçla yerleştirme planlarının oluşturulması temel problemdir. Söz konusu yerleştirme işlemi, çeşitli kriterler ve eldeki mevcut kesme düzeneklerinin getirdiği kısıtlamalar dikkate alınarak tamamlanır Böylece parça yerleştirme ve kesme planları oluşturulmuş olur.

2.2 ÇOK KULLANILAN TEMEL KAVRAMLAR

2.2.1 PARÇA YERLEŞTİRME PLANI

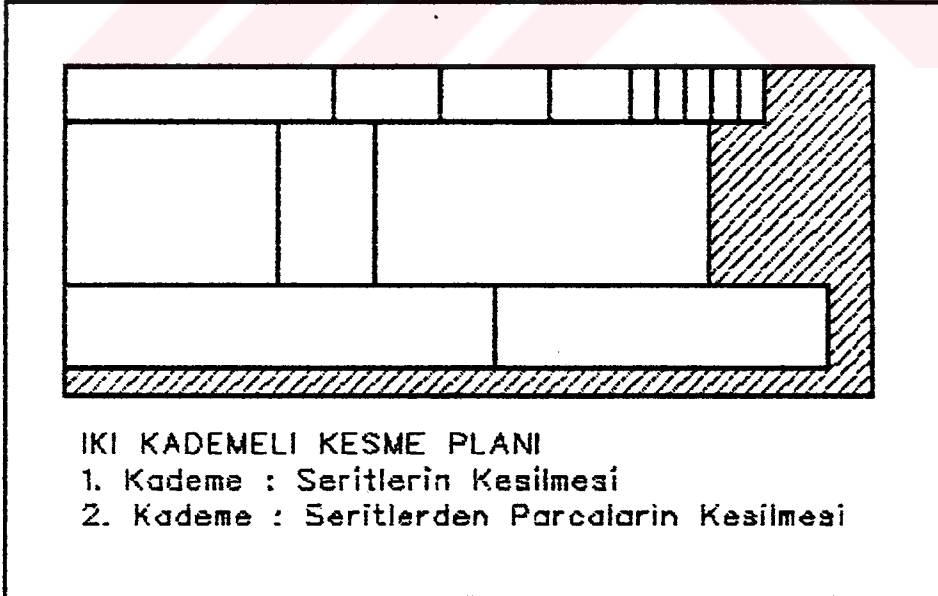
Kesme işlemi için ana malzeme üzerine yapılabilir yerleştirmelerin her birine PARÇA YERLEŞTİRME PLANI denir. Bu planlar istenen parçaların ana malzemeye yerleşim konumlarını ve yerleşim sayılarını gösterirler. Bu planlara kesme işlemini yönlendirmeleri nedeni ile KESME PLANLARI (=pattern) da denir.

İstenen parçaların herbiri belirli bir boyuta ve talep miktarına sahip olduğu için bir parçanın birden çok kesme planında yer alabilmesi olanaklıdır. Ancak bir parçanın tüm kesme planlarında bulunan miktarlarının toplamı o parçanın talep miktarından küçük olmamalıdır.

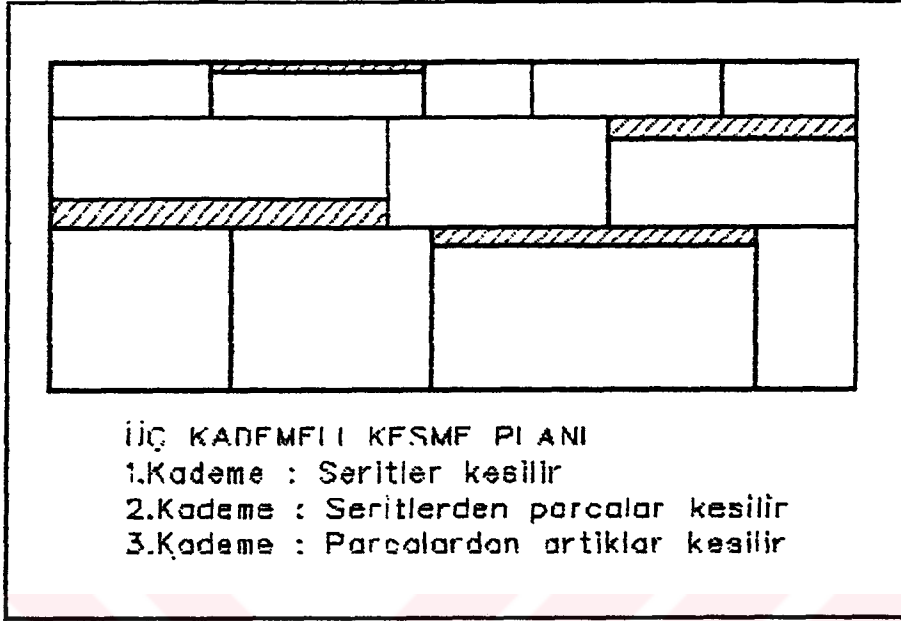
Kesme işlemi için bir planlama yapılabilmesi, işlemin teknolojik boyutu yanında ekonomik boyutunun da göz önüne alınması ile mümkündür.

İki boyutlu yerleştirme problemlerinde, temini istenen, belli talap miktarına sahip geometrisi belirli parçalar, ana pafta üzerine iki, üç hatta çok kademeli kesme operasyonlarının uygulanması ile elde edilebilmektedir. Çok kademeli kesme planları ile daha düşük düzeyde fire elde edilirken, bu planların gerek hazırlanması gerekse uygulanması sırasında kaybedilen zaman çok fazla olduğundan, pratikte pek kullanılmamaktadır. İki kademeli kesme planları, uygulamadaki ihtiyaçları karşılayabilecek düzeydedir. (Farley 1986).

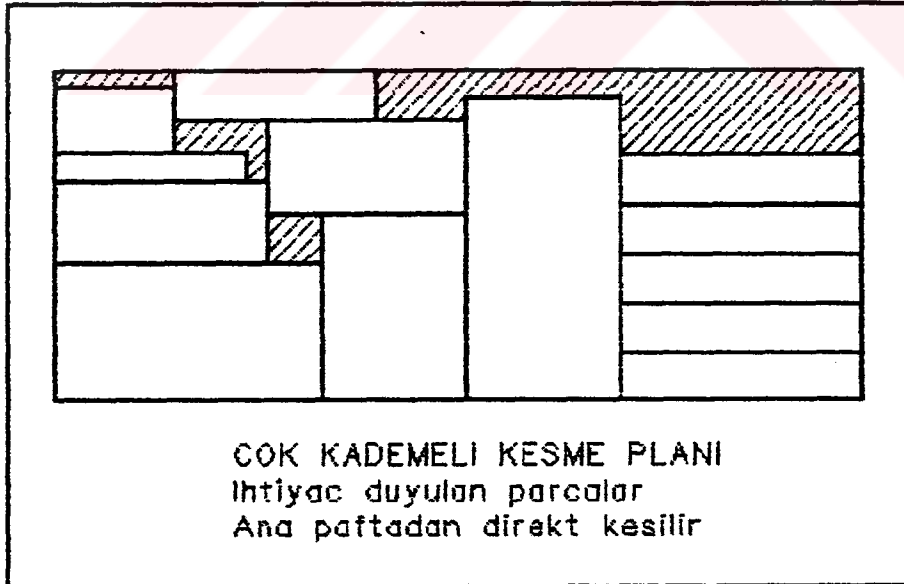
Şekil 2.1'de iki kademeli, Şekil 2.2'de üç kademeli ve Şekil 2.3'de de çok kademeli kesme planlarına örnekler görülmektedir.



Şekil -2.1- İki kademeli kesme planı



Şekil -2.2- Üç kademeli kesme planı



Şekil -2.3- Çok kademeli kesme planı

2.2.2 KESME KAYIPLARI

Herhangi bir malzemenin, hazırlanan kesme planına göre kesilmesinden sonra ortaya çıkan ve boyutları itibarı ile istenen parçaların en küçük boyutlara sahip olanından daha küçük olabilen ve/veya başkaca bir yerleştirmeye olanak tanımayan artık malzemeye kesme kayıpları (Fire) denir.

2.2.3 KESME PROBLEMİNİN KARMAŞIKLIĞI

Ana malzeme olan plakaların boyutları, kesilecek şekillerin geometrik biçimlerinin karmaşıklığı, elde edilmesi istenen parçaların çeşidi, her bir parçanın boyutu ve talep miktarı, kesme işleminde kullanılacak olan kesme tekniği, firmanın uyguladığı talep karşılama politikası ve diğer bazı teknolojik ve/veya üretim kısıtları problemin karmaşıklık derecesini etkilemektedir.

2.2.4 KESME PROBLEMİNİN BOYUTU

Ana malzeme üzerinde kesme işlemi uygulanan doğrultu sayısı, kesme probleminin boyutunu ifade eder.

2.2.5 İHTİYAÇ DUYULAN TEORİK ANA MALZEME MİKTARI

Hazırlanan yerleştirme ve kesme kesme planları neticesinde kullanılacak olan yarı mamül malzeme miktarının boyutları aşağıdaki formül ile hesaplanabilir :

S_{mt} : İhtiyaç duyulan teorik ana malzeme miktarı
 L : Kullanılacak ana paftanın uzun kenarı
 W : Kullanılacak ana paftanın kısa kenarı
 n : Elde edilmek istenen parça çeşit sayısı
 L_i : i . parçanın uzun kenarı
 W_i : i . parçanın kısa kenarı
 T_i : i . parçanın talep miktarı

$1 \leq i \leq n$ olmak üzere :

$$S_{mt} = \frac{\sum_{i=1}^n T_i * L_i * W_i}{L * W} \quad (2.1)$$

eşitliği kullanılır.

2.2.6 KULLANILACAK MİNİMUM ANA MALZEME MİKTARININ TESPİTİ

Elde edilmesi istenen parçaların bazıları, boyutlarından dolayı, kullanılacak ana malzeme içine sadece bir defa yerleştirilebilir. Talep listesinde söz konusu parçaların talep miktarlarının toplamı , teorik ana malzeme miktarından büyükse, hesaplanmış olan değer en az ana malzeme miktarı olarak kabul edilir.

2.2.7 KAYIP MİKTARI

Kayıp miktarı aşağıdaki formül ile hesaplanır.

n_s : Kullanılacak toplam levha miktarı
 S_{mt} : İhtiyaç duyulan teorik ana malzeme miktarı
 n_{min} : En az ana malzeme ihtiyacı

olmak üzere, Kayıp Miktarı :

$$\text{Eğer } n_{min} \geq S_{mt} \text{ ise } \frac{n_s - n_{min}}{(n_s \text{ veya } n_{min})} \quad (2.2)$$

$$\text{Eğer } n_{min} \leq S_{mt} \text{ ise } \frac{n_s - S_{mt}}{(n_s \text{ veya } S_{mt})}$$

2.2.8 MALZEME KULLANIM ETKİNLİĞİ

Yerleştirme planlarını türetme işleminin sonlarına doğru türetilen yerleştirme planlarında, son anda belirebilecek talepleri karşılamaya uygun boyutlarda alanların ortaya çıkabilmesi muhtemeldir. Söz konusu alanlar, ana pafta üzerinde malzeme hataları olan bölgelere denk gelen parçaların yeniden kesilmesi veya olası kesme kusurları sebebi ile ihtiyaç duyulan yeniden kesme işlemleri için uygundur. Dolayısı ile kullanılabilir nitelikli bu artık malzemenin kayıp olarak değerlendirilmesi doğru değildir. Çünkü bu alanlar mevcut talep listesinde yer alan en küçük boyutlara sahip parçadan daha büyük boyutlu olabilmektedir. Bu esaslara göre :

D_i : i. parçanın talep miktarı
 L_i : i. parçanın uzun kenarı

- Wi : i. parçanın kısa kenarı
n : Parça çeşit sayısı
n_s : Kullanılacak toplam levha miktarı
L : Ana malzeme uzun kenarı
W : Ana malzeme kısa kenarı
L_i : Daha sonra kullanılabilir j. alanın uzun kenarı
W_j : Daha sonra kullanılabilir j. alanın kısa kenarı
k : Daha sonra kullanılabilir alan miktarı

olmak üzere :

$$\text{Malzeme kullanım etkinliği} = \frac{\sum_{i=1}^n D_i * L_i * W_i}{n_s * L * W - \sum_{j=1}^k L_j * W_j} \quad (2.3)$$

bağıntısı ile hesaplanır.

2.3 YERLEŞTİRME PROBLEMLERİNİN BOYUTLARINA GÖRE SINIFLANDIRILMASI

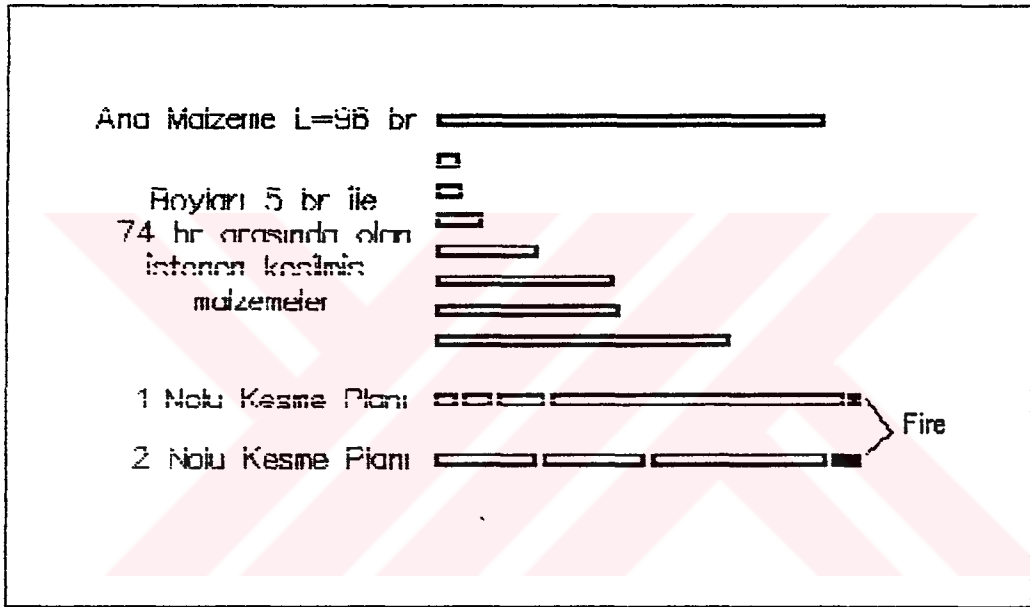
Endüstrinin çeşitli kesimlerinde bu güne kadar karşılaşılan kesme problemleri ,boyutlarına göre ,Farley tarafından şu şekilde sınıflandırılmıştır:

a) Bir boyutlu problemler

- 1.) İnşaat demirlerinin kesilmesi
- 2.) Tesisat borularının kesilmesi

- 3.) Kağıt ruloların kesilmesi
- 4.) Karton ruloların kesilmesi

gibi operasyonlarda, örneklenen malzemelerin her birinden belirli bir boy ve miktarda olmak üzere daha küçük parçaların kesilmesi işlemi bir boyutlu kesme problemlerini açıklamaktadır. (Şekil 2.4)



Şekil -2.4- Bir boyutlu kesme problemi

b) Birbuçuk boyutlu problemler

- 1.) Buzdolabı saçlarının kesilmesi
- 2.) Çamaşır makinası saçlarının kesilmesi

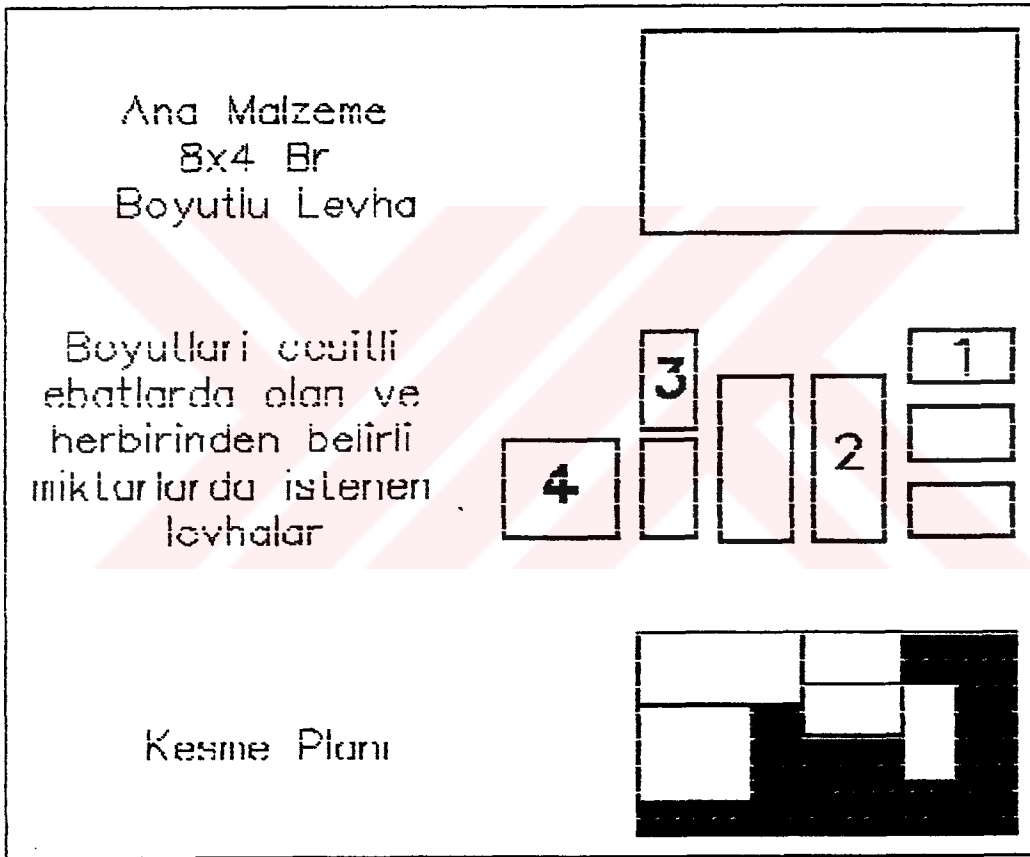
c) İki Boyutlu problemler

- 1.) Pencere camlarının kesilmesi
- 2.) Kumaş rulolarının kesilmesi
- 3.) Çelik saç rulolarının kesilmesi
- 4.) Çelik plakaların kesilmesi

5.) Mobilya suntalarının kesilmesi

6.) Paletlere yükleme yapılması

gibi operasyonlarda, dikdörtgen şekilli plakanın daha küçük dikdörtgen şekillere kesilmesi iki boyutlu kesme problemini açıklamaktadır. (Şekil 2.5)



Şekil -2.5- İki boyutlu kesme problemi

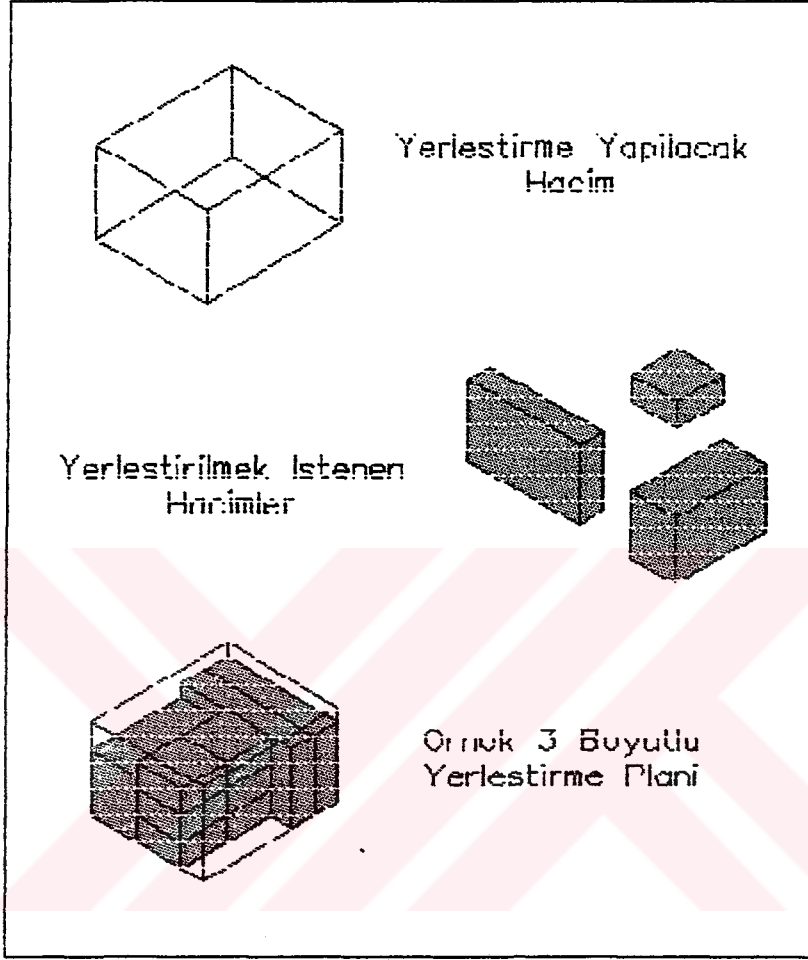
d) Üç boyutlu problemler

1.) Ambalaj kutularına yerleştirme yapılması

2.) Kamyonlara yükleme yapılması

3.) Gemilere yükleme yapılması

4.) Mermer blokların kesilmesi



Şekil -2.6- Üç boyutlu Yerleştirme problemi

(Şekil 2.6), üç boyutlu yerleştirme problemini açıklamaktadır.

2.4 YERLEŞTİRME PROBLEMLERİNİN KISITLARI

Kesme problemleri, bir girdi/çıkıı süreci olduğundan; girişten çıkışa dek tüm süreç kısıtlarının bilinmesinde zorunluluk vardır. Kesme problemlerinin kısıtları;

- a.) Malzeme kısıtları
- b.) Teknolojik kısıtlar
- c.) Yöntemsel kısıtlar

olmak üzere üç başlık altında incelenebilir. İlgilenilen problemin uygulama alanı ve yapısı gereği, daha başka kısıtların da gözönüne alınabileceği unutulmamalıdır.

2.4.1 MALZEME KISITLARI

Kesilmesi istenen malzemenin geometrisi, boyutları, fiziksel özellikleri ve temin edilebilir miktarı, problemin malzeme kısıtlarını oluşturur. Örneğin;

Malzemenin kesiti, yapının lifli olup olmadığı, yoğunluğu, yüzey kusurlarının bulunma yerleri ve bunların bulunma olasılıkları vb. bilgiler malzeme kısıtlarına örnek olarak gösterilebilirler.

2.4.2 TEKNOLOJİK KISITLAR

Kesme olayında esas itibarıyla bir malzemeye, herhangi bir makina veya düzeneğe kesme işlemi uygulanmaktadır. Buna göre kullanılan kesme düzeneğinin kesme kapasitesi, ayar yetenekleri vb. bilgiler teknolojik kısıtları oluşturur.

Kesme işleminin, giyotin makaslarda, destere tezgahlarında veya ergitme etkili bir ekipmanda yapıldığı duruma göre üretilecek kesme planları kesme kısıtına uyumluluk yönüyle farklı olacaktır (İsrani,1984).

2.4.3 YÖNTEMSEL KISITLAR

Kesme problemlerinde, kesme planları, talebin karşılanma durumu ve çözümün tam sayı olması, yöntemsel kısıtlar başlığı altında toplanabilirler.

Kesme problemi çözümünde amaç, yalnızca kesme kayıplarının en aza indirilmesi yerine, denetlenebilir tüm maliyetlerin toplamının en küçüklenmesi ise o vakit elden geçirme ve tezgahların ayar edilme maliyetlerini etkileyen kesme planı toplam sayısının da bir kısıt olarak ele alınması gerekmektedir.

Bir boyutlu kesme problemlerinde, kesme düzeneğinin kapasitesine bağımlı olarak herbir kesme planının içerdiği parça sayısı, bir kesme planı kısıtı olarak nitelenir. Ayrıca herbir kesme planı için öngörülen fire miktarı da bir başka kesme planı kısıtını oluşturmaktadır.

Kesme planları üzerine çok sayıda kısıt koymanın, çözüm süresini artırabileceği, uygulamalarda ortaya çıkmıştır (Christofides, 1977).

İstenilen herbir parçanın talebinin tam olarak karşılanması ideal durum iken; bazen içinde bulunan koşullar ve izlenen işletme politikaları nedeniyle, istenen parçalardan bir kısmının talebinin kısmen karşılanması yahut talep miktarından daha fazla kesim yapılması düşünülebilir (Haessler, 1975). Bu düşünceler, kesme planları oluşturulma aşamasında talep karşılama kısıtı olarak değerlendirilir.

Temin edilebilen boyutlar itibarıyla bir ana malzeme, bazı parçaların taleplerinin bir miktarını, sonradan karşılamayı gündeme getirebilir. Benzer olarak, kesme planlarının üretimi esnasında öngörülen boyutlardan daha büyük boyutlu ana malzeme temin edilmesi durumu ile karşılaşılsa, bu defa herhangi bir parçadan talep miktarının üstündeki değerlerde kesim yapılabilir. Bu durumda çıkan fazla parçalar, daha sonra kullanılmak üzere saklanabilecektir.

Uygulamada karşılaşılan kesme problemlerinin çözüm sonuçları, tamsayıli olmak zorundadır (Litton,1977). Bir kesme probleminin çözüm yöntemi analitik ise, sözgelimi bir doğrusal programlama kullanılıyorsa, buna bir yuvarlatma işlemi eklenmelidir (Haessler,1975). Herbir parçanın herbir kesme planında bulunabilme miktarına bir kısıt konulmasının nedeni, sonuçların tamsayıya yuvarlatılmasıyla ilgilidir. Analitik yöntemlerle, tamsayıya yuvarlatma işlemi sözkonusu olduđu halde yordamsal yaklaşımlarda böyle bir işleme gerek kalmamaktadır (Swccncy,1988).

2.5 YERLEŞTİRME PROBLEMLERİNDE BENİMSENEİLİR AMAÇLAR

Kesme problemini bir karar problemi olarak ele alabilmek için bir yahut birden fazla ölçütün benimsenerek, karşı gelen amaç fonksiyonun tanımlanması gerekmektedir. Kesme problemlerinin çözümünde kesme kaybı, malzeme miktarı, toplam maliyet ve kapasite kullanımı ölçütleri benimsenebilir (Harrison, 1984).

2.5.1 KESME KAYBI ÖLÇÜTÜ

Uygulanan bir yöntemle oluşturulan tüm kesme planları, uygun dilme planları kümesine alınmayabilir. Üretilen bir planın, uygun planlar kümesine girebilmesi için önceden belirlenen bir miktardan daha düşük miktarda veya eşit miktarda fire içermesi istenebilir. Her mümkün plan için bu istek tekrarlanırsa, kesme kayıpları toplamının enküçüklenmesi amaçlanmış demektir. Amaç fonksiyonunda, kesme kaybı ölçütünün benimsenmesi geleneksel ve yaygın bir tutumdur.

2.5.2 MALZEME MİKTARI ÖLÇÜTÜ

Temin edilebilir ana malzemenin, boyutları itibariyle birden fazla olması durumunda; giren malzeme miktarının enküçüklenmesi gerekebilir. Tek çeşit ana malzeme kullanılıyorsa, bu ölçütün benim senmesi, fire ölçütünün benimsenmesi ile aynı anlamda düşünölmelidir.

2.5.3 TOPLAM MALİYET ÖLÇÜTÜ

Bir kesme probleminde toplam maliyeti, kesme planlarını düzenleme ve elden geçirme maliyetleri, kesme süreci maliyeti ile fire maliyetleri oluşturmaktadır. Amaç foksiyonu, bu üç maliyetin toplamı olarak tanımlanabilir ve bu toplamın enküçüklenmesi, karar verici tarafından benimsenebilir. Yüksek maliyetli malzemeler, pahalı düzeneklerde kesilecek ise, toplam maliyetin enküçüklenmek istenmesi önerilebilir. Ancak denetlenebilir tüm maliyetleri içeren bir toplam maliyetin tanımlanabilme olanakları araştırılmalıdır.

2.5.4 KAPASİTE KULLANIM ÖLÇÜTÜ

Taleplerin, kabul edilebilir bir süre içinde karşılanması istenir. Mevcut makina ve kesme düzeneklerinin kapasitelerinin kullanım oranları, talep karşılama sürelerini etkiler. Buna dayanarak, ekonomik sınırları zorlamaksızın kapasite kullanımının enbüyüklenmesi, bazı karar vericilerin benimsenebilecekleri bir ölçüttür (Harrison, 1984).

Ele alınan problemin ilgi alanı ve yapısına bağlı olarak benimsenebilir ölçütlerin farklılık göstermesi doğaldır. Ancak birden fazla ölçütün benimsenebilmesini olanaklı kılan bir çözüm yaklaşımının daha etkin olduğu

söylenebilir (Harrison, 1984).

Mobilya endüstrisinde kullanılacak sunta levhaların kesilmesi probleminde, çok ölçütün benimsendiği belirtilmektedir (Harrison, 1984). Ancak, çelik plakaların iki boyutlu dilinmesi probleminin, çok ölçüt benimsenerek ele alındığı bir çalışmaya, gözden geçirilen kaynaklarda rastlanamamıştır. İncelenen kaynaklarda bu problemin, kesme kaybı (fire) ölçütü ile ele alındığı görülmektedir.

3. YERLEŐTİRME PROBLEMLERİNİN MODELLENMESİ

Arařtırmacılar, kesme problemleri ile 1950 yılından itibaren sistematik biçimde ilgilenmeğe başladılar (Dağlı, 1988). Problemin çözümü için, başlangıçta iki aşamadan oluşan geleneksel yaklaşım kullanıldı (Dyckhoff, 1985). Geleneksel yaklaşımın birinci aşamasında, sayımlama ile uygun kesme planları oluşturulur. ikinci aşamada ise uygulamaya konulacak kesme planlarının herbirisinin miktarı, doğrusal programlama kullanımı ile saptanır. Doğrusal programlama modelinin çözümünden elde edilen sonuçların tamsayıya dönüřtürölmesiyle de beklenen değerlere erişilir. Geleneksel yaklaşımda toplam sayımlama, küçük ölçekli problemlerde bile çok sayıda mümkün kesme planı türeteceğinden; doğrusal programlamada sınırlı sayıda kesme planı kullanılır.

Kesme problemleri, ilk kez 1961 yılından Gilmore ve Gomory tarafından bir matematik model kurularak incelenmiştir (Farley, 1986). Gilmore - Gomory yönteminde, herbir ardıştırma kademesinde sırt çantası yardımcı probleminin çözümü ile kesme planları türetilip, sonra bunlar doğrusal programlama modelinde kullanılarak herbir kesme planından işleme girecek en iyi miktarlar saptanır.

Bu kesimde bir, iki ve üç boyutlu kesme problemlerinin matematik modelleri kurulacak fakat çözüm yaklaşımlarından söz edilmeyecektir. çözüm yaklaşımlarına izleyen kesimde değinilerek. İkinci yaklaşım özetlenecektir.

3.1 BİR BOYUTLU YERLEŞTİRME PROBLEMİNİN MODELLENMESİ

İnşaat demirlerinin ve tesisat borularının kesilmesi, bir boyutlu kesme problemine gösterilebilecek geleneksel uygulamalardır. Anılan bu malzemelerde kesme işlemi başlatılmadan önce, belirli bir uzunlukta demirler veya borular, çapına, kesitine ve benzeri bir özelliğine göre gruplandırılır. Herbir grup kesme işlemine tabi tutularak istenen parçalara erişilir. Kesme işleminde kişisel deneyim ve beceri etkindir. Oysa kesme işleminin, benimsenen bazı ölçütlere göre oluşturulmuş kesme planları ile yürütülmesi gerekmektedir. Kesme planları analitik bir yöntemle üretilmek istenirse, Gilmore - Gomory'nin önerdiği matematik modele başvurulur. Buna göre, bir kaç çeşit ana malzeme kullanımına dayalı bir boyutlu kesme probleminde:

- L : Kesilecek ana malzeme uzunluğu
- m : Uzunluğu ve talep miktarı farklı parça çeşidi
- L_i : i. parçanın boyu
- D_i : i. parçanın talep miktarı
- a_{ij} : j. kesme planındaki i. parça sayısı
- n : Kesme planı sayısı
- C_j : j. planın maliyeti
- X_j : j. plandan gerekli miktar

simgelemek şartı ile , karar modeli :

$$\sum_{i=1}^m a_{ij} * L_i \leq L \quad j= 1, n \quad (\text{Kesme planı kısıtı})$$

$$\sum_{j=1}^n a_{ij} * x_j = D_i \quad i = 1, m \quad (\text{Talep karşılama kısıtları})$$

$$x_j \geq 0$$

ve tamsayı

kısıtları altında :

$$\text{Enk } x_0 = \sum_{j=1}^n C_j x_j \quad \text{şeklinde yazılır}$$

3.2 İKİ BOYUTLU YERLEŞTİRME PROBLEMİNİN MODELLENMESİ

Bu tür problemlerde de, boyutları bilinen dikdörtgen plakalardan, herbirinin boyutları ve talepleri bilinen "m" adet küçük parçanın kesilmesi için gerekli kesme planlarının hazırlanması istenir. Buna göre iki boyutlu problem, bir boyutlu problem için tasarlanan simgesi gösterimlere ek olarak,

- L : Kesilecek ana malzemenin uzun kenar uzunluğu
- W : Kesilecek ana malzemenin kısa kenar uzunluğu
- L_i : i. parçanın uzun kenarını
- W_i : i. parçanın kısa kenarını

gösterdiğinde, model :

$$\sum_{i=1}^m a_{ij} * L_i * W_i \leq L * W_0 \quad (\text{Kesme planı kısıtı})$$

$$\sum_{j=1}^n a_{ij} * x_j = D_i \quad i = 1, m \quad \begin{matrix} (Talep \\ kısıtları) \end{matrix} \quad \begin{matrix} karşılama \end{matrix}$$

$$x_i \geq 0 \quad \text{ve tamsayı}$$

kısıtları altında :

$$\text{Enk } x_0 = \sum_{j=1}^n C_j x_j \quad \text{biçiminde yazılır}$$

3.3 ÜÇ BOYUTLU YERLEŞTİRME PROBLEMİNİN MODELLENMESİ

Üç boyutlu problemler, bir kesme probleminden çok yerleştirme problemi özelliği gösterir (Steudel, 1979). Gözden geçirilebilen kaynaklarda üç boyutlu kesme problemi için kayda değer nitelikte çalışmalara raslanamamıştır. Ancak üç boyutlu kesme problemi, bir ve iki boyutlu problemler için tasarlanan simgesel gösterimlere, şeklinde modellenenebilir.

H : Ana hacmin yüksekliğinin

H_i : i. parçanın yüksekliğinin

eklentisi ile :

$$\sum_{i=1}^m a_{ij} * L_i * W_i * H_i \leq L * W_0 * L \quad (\text{Yerleştirme planı kısıtı})$$

$$\sum_{j=1}^n a_{ij} * x_j = D_{ij} \quad i = 1, \dots, m \quad (\text{Talep karşılama kısıtları})$$

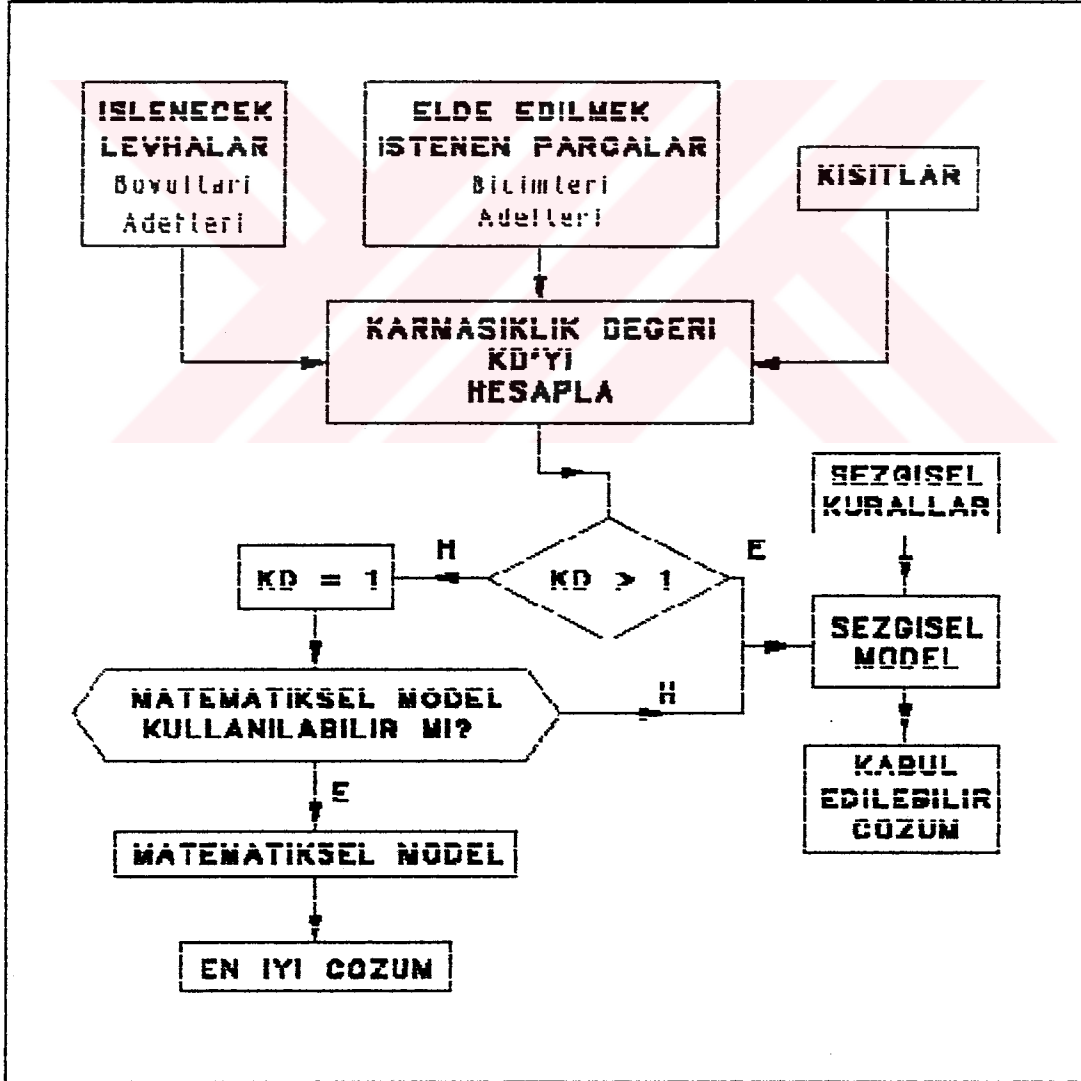
$$x_j \geq 0 \quad \text{ve tamsayı}$$

kısıtları altında :

$$\text{Enk } x_0 = \sum_{j=1}^n C_j x_j \quad \text{şeklinde yazılır}$$

4. YERLEŐTİRME PROBLEMLERİ ÇÖZÜM YÖNTEMLERİ

Bir, birbuçuk, iki veya üç boyutlu bir kesme problemini tüm kısıtları gözönüne alarak çözebilen genel amaçlı bir çözüm yaklaşımı bulunmamaktadır (Dyckhoff,1985). Problemin boyutuna ve kısıtlarına bağılı olarak, benimsenen çözüm yaklaşımları farklılık göstermektedir. Kesme problemlerinde kullanılan çözüm yöntemlerinin akış diyagramı Şekil 4.1'de gösterilmektedir.



Şekil -4.1- Kesme problemlerinde kullanılacak çözüm yönteminin tespiti

Parça yerleştirme ve kesme problemi konusunda mevcut çalışmalarda, fire miktarının minimum olması amaçlanmaktadır. Fire maliyetlerinin yanında denetlenebilir diğer maliyetlerin enküçüklenmesi de istenirse uygulanan çözüm yöntemleri daha da karmaşıklaşacaktır.

İki boyutlu yerleştirme problemlerini :

a.) Dikdörtgen şekillerin yerleştirilmesi:

En çok bilinen gruptur. Kesim işlemi sonunda arta kalan malzemenin en küçükleme hedeftir. Bu amaçla çeşitli matematiksel modeller geliştirilmiştir. Kullanılan ANALİTİK YÖNTEMLER'deki temel problemler , kesme probleminin kısıt sayısı arttıkça karmaşıklık derecesinin artması, çözüm süresinin uzaması ve ayrıca belirlenen çözüm değerlerinin yuvarlatılma zorunluluğudur. Kullanım olanığı bulmuş analitik yöntemler :

- i) Sayımlama ve doğrusal programlama bileşimi,
(geleneksel yöntem),
- ii) Dinamik programlama,
- iii) Tamsayılı programlama ile doğrusal
programlama bileşimi,
- iv) Dal - sınır yöntemi,

başlıklarında toplanmaktadır (Dagli,1983)

b.) Giyotin kesim problemleri:

Dikdörtgen kesim problemlerine benzer özellikler içermektedir. Fakat bu problemlerin çözümünde göz önüne alınması gereken önemli bir kısıt mevcuttur. Yapılacak olan kesme işleminde operasyon, plakanın bir kenarından başlayıp diğer tarafa kadar kesintisiz devam etmek zorundadır. Bu tür problemin çözümünde de, mevcut çalışmalarda ,yine ANALİTİK YÖNTEMLER kullanılmıştır.

c.) Biçimsiz şekillerin yerleştirilmesi

Bu tür problemlerin çözümü için kullanılan ANALİTİK YÖNTEMLER'deki matematiksel programlama modellerinin, kullanılabilirliği, hesaplama yükünün fazlalığı nedeni ile zaman ve uygulanabilirlik açısından kısıtlıdır. Dolayısı ile probleme bağımlı çeşitli SEZGİSEL YÖNTEMLER kullanılmaktadır. Bu yöntemler, ele alınan problemin gereksinimlerine göre tasarlanmış olup en iyiye yakın çözüm verebilmektedir. Analitik yöntemlere göre, bazı kullanım kolaylıklarına sahiptirler. Ayrıca yordamsal yaklaşımlarda çözüm değerlerinin tamsayıya yuvarlatılmasına gereksinim duyulmadığı için, kesme planları üzerine daha fazla kısıt konulmasına gerek kalmamaktadır.

SEZGİSEL YAKLAŞIMLAR'da, en iyiye yakın bir çözüme, durum uzayında araştırma yapılarak veya problem indirgeme yöntemi kullanılarak erişilebilir (Hinxman 1980; Farley 1986).

Mevcut bazı SEZGİSEL YAKLAŞIMLAR'da, iki boyutlu yerleşimi geliştirmek için etkileşimli yöntemler kullanılmıştır. Böylelikle iki boyutlu çeşitli yerleşim seçenekleri üzerinde kullanıcının isteklerine de yer verilmiş olmaktadır. Dolayısı ile dinamik olarak değişen sezgisel kurallarda uygulanabilmektedir.

Biçimsiz şekilli parçaların yerleştirilmesi ile ilgili problemlerin çözümünde, kullanılacak yöntemin belirlenmesi safhasında, ANALİTİK YÖNTEMLER ile SEZGİSEL YÖNTEMLER arasında seçim yapılırken, talep edilen parça çeşit sayısı ile toplam parça sayısının birbirine oranlanması ile elde edilen katsayının önemli olduğu görülmektedir.

m : Talep edilen parçaların çeşit sayısı

n : Talep edilen toplam parça sayısı

olmak üzere,

$$0 < m/n \leq 1$$

aralığında;

(m/n) değeri 0'a yaklaşıyor ise ANALİTİK YÖNTEMLER

(m/n) değeri 1'e yaklaşıyor ise SEZGİSEL YÖNTEMLER

kullanılmalıdır.

Toplam fire miktarını minimum yapacak veya şekillerin toplam yerleşim değerini maximum yapacak yerleşim biçimi aşağıdaki KISITLAR göz önünde tutularak oluşturulur.(Dağlı ve Tatoğlu 1984)

- a.) Parçalar levha sınırları içinde olmalıdır.
- b.) Parçalar üst üste binmemelidir.
- c.) Bazı parçalar, levhanın istenen kısmına yerleştirile bilmelidir
- d.) Bazı parçalar istenen yönlerde yerleştirilmelidir.
- e.) Parçalar arasında kesim payı bırakılmalıdır.
- f.) Diğer olası özel kısıtlar (Giyotin tipi kesme problemleri, şekil tipleri üzerine kısıtlar).

5. YERLEŐTİRME PROBLEMLERİ İÇİN MEVCUT ÇALIŐMALARIN İRDELENMESİ

Kesme problemlerinin çözüm yöntemlerini ve yaklaşımlarını ele alan çalışmaların çoğunda, fire miktarlarının minimum olmasının amaçlandığı görülmektedir. Fire maliyetlerinin yanında denetlenebilir diğer maliyetlerin enküçüklenmesinin istenmesi halinde ise daha çok çaba harcamak gerekir (Smithin ve Harrison 1982; Farley'den 1986).

Çalışmalarda ele alınan kesme problemlerinin çözümünde, eniyi çözüm değerini bulabilen analitik çözüm yöntemleri yanında; sezgisel çözüm yaklaşımlarına da raslanmaktadır. Sezgisel çözüm yaklaşımları, eniyi çözüm değerini garanti edememekle birlikte, en iyiye yakın çözüm değerlerini, uygulamada kabul edilebilecek düzeyde verebilmektedir. Sezgisel çözüm yaklaşımlarının dikkate değer diğer bir özelliği ise probleme bağımlı olmalarıdır.

Çoğu kesme probleminin çözümü için analitik yöntemler ve sezgisel yaklaşımlar ayrımı yapılabildiği halde; gerçek hayat kesme problemlerinin çözümünde geleneksel yaklaşımlar, Gilmore - Gomory yöntemi, sezgisel yaklaşımlar ve diğerleri ayrımda yapılabilir (Dyckhoff,1985).

Geleneksel yaklaşımda, analitik yöntemler kullanılarak mümkün kesme planları, sayımlama ile oluşturulduktan sonra bu kesme planları bir doğrusal karar modelinde kullanılarak çözüme gidilir. Tamsayı olmayan değerler tamsayıya yuvarlatılır (Haessler,1975). Karar değişkenlerinin (Kesme planı sayıları) tamsayılı olması kısıtına uyum sağlamak için, tamsayıya yuvarlatma işlemleri yerine kesikli eniyileme yöntemlerine de başvurulabilir. Fakat bu yöntemin makul sayılamayacak büyüklükteki hesapsal yükünü gözönüne almak gerekecektir. Buna rağmen kesikli eniyileme yönteminin, kesme planlarının eniyi sıralarının belirlenmesinde

kullanılması mümkündür (Dyckhoff,1985)

İki boyutlu yerleştirme problemlerinin çözümünü için yapılan çalışmaları da yine dikdörtgen şekillerin yerleştirilmesi, giyotin kesim ve biçimsiz şekillerin yerleştirilmesi olarak üç ana grupta toplayabiliriz (C.H.Dağlı,M.Y.Tatoğlu,1984)

5.1 DİKDÖRTGEN ŞEKLİ PARÇALARIN YERLEŞTİRİLMESİ İLE İLGİLİ ÇALIŞMALAR

Dikdörtgen şekilli parçaların yerleştirilmesi, en çok bilinen grup olarak dikkat çeker. Bu grupta toplayacağımız çalışmalarda, yerleştirme işlemi sonunda elde edilen fire miktarının minimum olması temel amaç olarak ele alınmıştır. Bu tür problemlerin analitik yöntemler kullanılarak çözülmesi konusunda ortaya konan çalışmalardan ilki, Gilmore - Gomory'ye aittir (Sweeney,1988).

Wang (1983) yaklaşımında ise yalnızca en küçük fireli kesme planları üretilebilmekte ve herhangi maliyet bilgisi kullanılmaktadır.

Chambers ile Dyson ise, ana malzemeler arasından boyutları itibariyle en iyi olanlarını seçebilmek için, istenen parça boyutlarının dağılımını, Gilmore - Gomory yöntemi ile birleştirmişlerdir (Chambers,1976). Ancak bu çalışmada, ana malzemelerin aynı genişlikte olmaları gerektiği bildirilmektedir.

5.1.1 GILMORE - GOMORY YÖNTEMİ

Bu yöntem esas itibariyle bir doğrusal programlama modeli çerçevesinde simpleks algoritmasına dayanmaktadır. Burada doğrusal karar modelinin amaç fonksiyonunda beklenen iyileşmeyi sağlayacak yeni bir sütun, yardımcı

problem çözümlere oluşturulur. Çözümlere yardımcı problem ise, sırt çantası veya yükleme problemi tipinde tamsayılı bir doğrusal programlama problemidir.

5.1.1.1 İKİ BOYUTLU KESME PROBLEMLERİNDE GILMORE - GOMORY YÖNTEMİ

Doğrusal karar modeline eklenen yeni sütun öyle bir yerleşim planı üretebilmelidir ki, o planın içindeki parçalar ana malzemeye yerleştirildiğinde bir taşma veya üst üste binme olmasın.

Bir yerleşim planının kesme planı olabilmesi için kullanılacak kesme düzeneğinde kesilebilir olması gerekir. Buna göre kesme düzeneğinin özellikleri nedeniyle zorunlu olarak konulan kesme kısıtı sonucu oluşturulan kesme planları, yerleşim planlarından fire büyüklükleri yönüyle farklı olacaktır. Ayrıca kesme planları 2,3 veya çok kademeli olarak elde edilebilir. Gilmore ve Gomory, iki kademeli kesme planlarının piratik uygulamalar için yeterli duyarlılıkta olduğunu, çok kademeli planların ise çözüm üretme süresi bakımından makul sayılabileceğini belirtmektedir (Dağlı,1988).

İki kademeli kesme planı üretebilmek için , bir boyutlu yükleme problemini (sırt çantası problemini) çözebilen ve iki aşamada uygulanan bir algoritma kullanılır. Algoritmanın birinci aşamasında en iyi şeritlerin birleştirilmesiyle kesme planları üretilmektedir.

a.) En İyi Şeritlerin Oluşturulması :

Bütün W_i ($i=1,2,...,m$) parça genişlikleri için, W_i genişliğinde ve L uzunluğundaki şerit içine genişlikleri uyan parçalar yerleştirildiğinde elde edilebilir S_i alanlarının en büyükleri belirlenir. Bu aşamada herbir i için tek boyutlu bir

yükleme problemi çözülerek en iyi S_i 'ler belirlenmektedir.

b.) En İyi Şeritlerin Birleştirilmesi :

Bu aşamada, yeniden tek boyutlu yükleme problemi çözülerek en iyi şeritlerin birleşiminden ibaret kesme planları oluşturulur. çözülecek yükleme problemi için karar modeli,

$$\sum_{i=1}^m W_i * b_i \leq W$$

(Genişlik)

$$b_i > 0$$

ve tamsayı

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

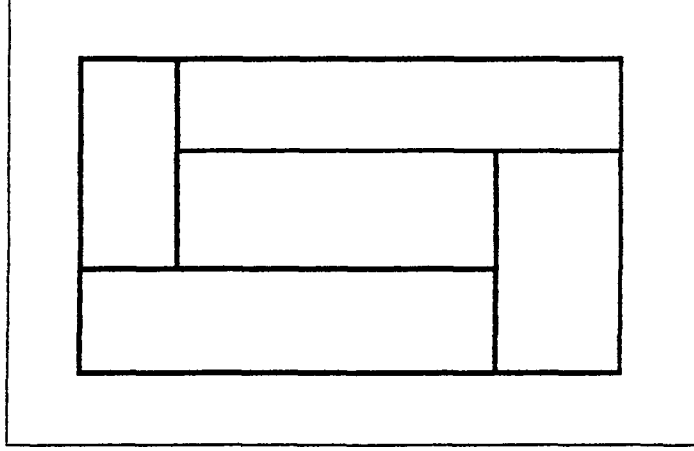
kısıtları altında

$$\text{Enb } Z = \sum_{i=1}^m S_i * b_i$$

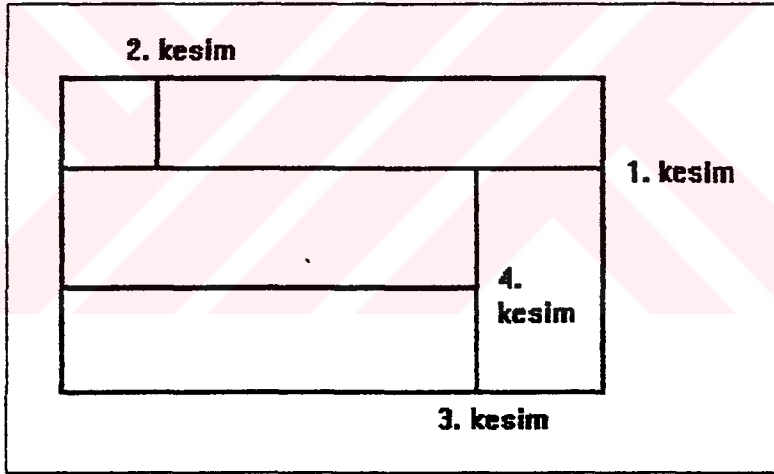
olarak ifade edilebilir

5.2 GİYOTİN KESİM PROBLEMLERİ İLE İLGİLİ ÇALIŞMALAR

Giyotin kesim problemleri, dikdörtgen kesim problemleri ile benzer özellikler taşımaktadır. Yalnız bu tip problemlerde özel bir kısıt vardır. Bu kısıt şudur, yerleştirme sonunda yapılacak olan kesim işlemi, plakanın bir kenarından başlayıp diğer tarafa kadar kesintisiz devam etmek zorundadır.

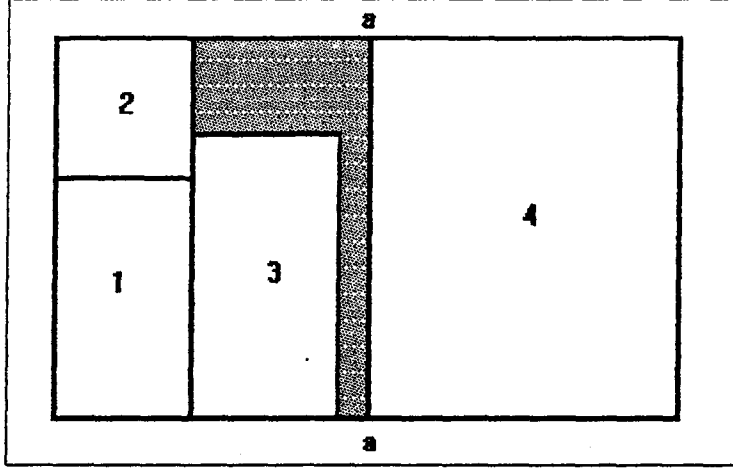


Şekil -5.1- Gilyotin Kesime müsait olmayan bir yerleştirme planı

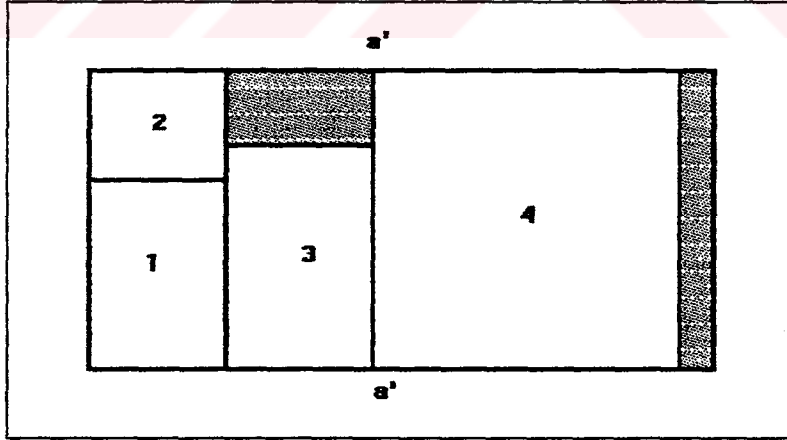


Şekil -5.2- Gilyotin Kesime müsait yerleştirme planı

Gilyotin kesim problemlerinin çözümü konusunda N.Christofides ve C.Whitlock tarafından bir Tree-Search algoritması geliştirilmiştir. Bu algoritma, çözüm taramasını gerçekleştirirken mümkün olan kesme planları için üst değerler üreten bir dinamik programlama modelidir. Algoritma, kesme planlarını ürettikten sonra, gilyotin kesme için göz önüne alınması gereken özel kısıtı gereğince plan üzerinde düzenleme yapmaktadır.



Şekil - 5.3 - Algoritma sonunda oluşturulan kesme planı



Şekil -5.4- Giyotin kesime uygun düzenlenmiş kesme planı

5.3 DÜZENSİZ ŞEKİLLERİN YERLEŞTİRİLMESİ İLE İLGİLİ ÇALIŞMALAR

Bu tür problemlerin çözümünde, matematiksel programlama modellerinin kullanımı, hesaplama yükünün fazlalığı nedeni ile kısıtlıdır. Örneğin :

Dikdörtgen şekilli parçaların bir pafta üzerine yerleştirilmesi söz konusu olsun. Ana paftanın boyutu 200 birim ve istenen 40 farklı parçanın boyutlarının 20 ile 80 birim arasında değiştiğini ve kesilecek parçaların en az bir kenarlarının eşit olduğunu kabul edelim. Bu parçaların ana malzeme üzerine yerleşim olasılıkları kolaylıkla 10 hatta 100 milyonu aşabilmektedir.(C.Dikili,1991).

İşte bu sebepten dolayı çeşitli sezgisel yaklaşımlar bulunmuştur.

Daha önce açıkladığı gibi yordamsal yaklaşımlar, problem bağımlıdır ve eniyiye yakın çözümü, makul bir süre içinde üretebilirler.

Sezgisel yaklaşımlar iki şekilde ele alınabilir :

a) Durum Uzayı Araştırması :

Problemin kısmi potansiyel çözümleri bir serimin düğümleri olarak göz önüne alınır ve bu serimdeki bir yörünge için başlangıç durumdan (toplam olarak çözülmemiş problem), son duruma (komple çözüm) kadar araştırma yapılır.

b) Problem İndirgeme :

Esas problem,alt problemlere ayrıştırılır. Hatta alternatif alt problemlere de yer verilebilir. Alt problemler kümesinin çözümlerinin birleştirilmesiyle de esas problemin çözümü bulunur.

Oluşturulan kesme planları arasında önceden saptanan ölçütleri karşılayabilenleri olabildiğince çok miktarda kullanmak eğilimi, problem indirgeme yaklaşımına örnek olarak gösterilebilir (Haessler,1975). Bir kesme planının mümkün miktarda kullanımına bağlı olarak, planda bulunabilen herbir parçanın karşılanamamış talep miktarları hesaplanarak problem yeniden çözülür. Böylece devam edilerek, istenen parçaların tamamının talep miktarının karşılanabildiği kesme planları kümesi oluşturulur

Litton ve Stainton, yaptıkları bir çalışmada kesilen parçaların, şimdi ve gelecekte kullanılabilmesince olanak tanıyan yordamsal bir yaklaşımdan söz etmektedirler (Litton,1977). Bu yaklaşımda her bir kesme planı, istenen parçaların bir veya ikisinin kendi içinde yerleşebilmesine fırsat tanımaktadır. Ayrıca herbir plan için bazı istekler öngörülerek, kesilecek malzeme sayısı enküçüklenmek istenmektedir. Litton, değinilen çalışmasında geçmiş dönem talep verilerinden hareketle, rassal yaklaşım ile kesilecek çubuk malzeme üzerinde kesilmemesi gereken bölgeleri de belirlemektedir. Bu isteğine neden olarak da, çelik çubukların haddehanelerde üretimi esnasında boylarının saptanmasını göstermektedir.

Herz(1972) tarafından önerilen yordamsal bir yaklaşımın, tekrarlı problem indirgeme ile çözüm ürettiği ve Gilmore Gomory yöntemine göre daha az bilgisayar belleği ile, daha az çalışma süresine gereksinim gösterdiği bildirilmektedir (Dağlı,1988). Bu çalışmada ayrıca, Gilmore ve Gomory'nin ikinci kez açıkladığı çok kademeli algoritmanın yanlış olduğu da iddia edilmektedir.

Adamowicz ile Albano(1976), iki boyutlu kesme probleminde, kesme planlarının türetilmesinde yordamsal yaklaşım kullanmaktadırlar (Dağlı, 1983). İstenen parçaların boyutları ana malzeme boyutlarına oranla küçükse ve herbir parçanın talep miktarı büyükse, yöntemin başarılı olduğu bildirilmektedir. Yaklaşımda, türetilen

herhangi bir kesme planında belirli bir fire miktarından daha büyük fire miktarına izin verilmemektedir.

Albano ılc Sapuppo(1980),Dağlı ılc Nışancı(1981) ve Tatoğlu(1983) da çalışmalarında, dikdörtgen olan ve dikdörtgen olmayan düzlemsel şekillerin bir ana malzemeden kesilmesi probleminde yordamsal çözüm yaklaşımını önermektedirler (Dağlı,1988).

Cihan H.DAĞLI ve M. Yalçın TATOĞLU , geliştirdikleri sezgisel yöntem aşağıda anlatılmaktadır. şekillerin plakalar üzerindeki ilk yerleşimlerini belirler ya da herhangi bir grup şekil için kullanılabilir. Sezgisel bilgisayar yordamının oluşturulmasında modüler modelleme yaklaşımı kullanılmıştır. Problemin çözümüne şekiller yerleştirilerek ulaşılmaktadır. Ancak, uygulanan yöntem, önceden yapılmış herhangi bir yerleşim biçimini geliştirememektedir. Bunun yanısıra, herhangi bir problem için, şekillere atanan öncelikler bazında yalnızca bir tek çözüm bulunmakta, bu da seçilen kurala bağımlı olmaktadır. Ancak, bulunan çözüm kullanıcıyı hiçbir şekilde bağlamayacak ve önerilen çözüm üzerinde değişiklikler yapılabilecektir.

Sezgisel bilgisayar yordamı, şekilleri öncelik kuralına uygun bir sırada yerleştirmektedir. Bu aşamada, problemle atölye tipi üretim çizelgelemesi arasında bir ilişki kurulabilir. Üretim çizelgelemesinde böyle bir düzenlemenin amacı ayrılan zamanın kullanım yüzdesini arttırmaya çalışmak, iki boyutlu yerleştirme problemlerinde ise amaç yeterli alanın kullanımını çoğaltmaktır.

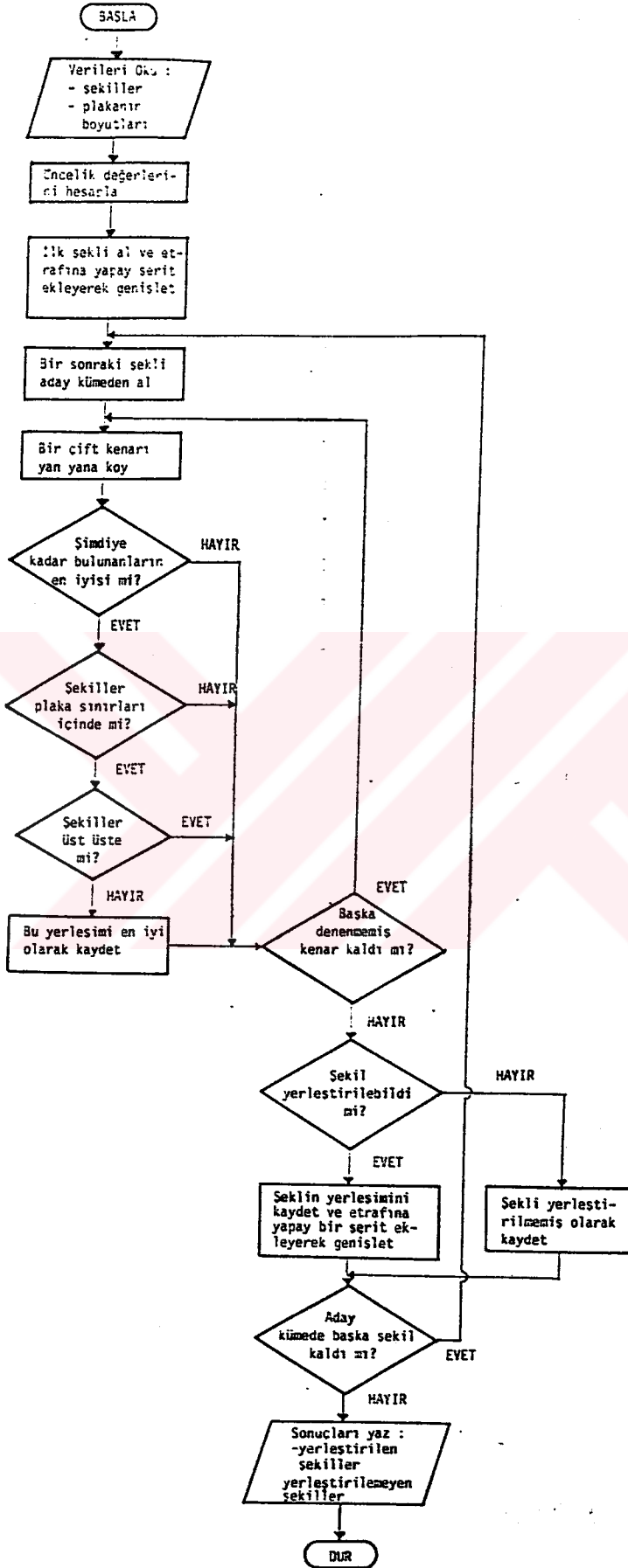
Geliştirilen sezgisel bilgisayar yordamında,aşağıda belirtilen, biri kullanıcının saptayabileceği kural olmak üzere toplam 12 öncelik kuralı bulunmaktadır.

Şekiller için Olası Öncelik Kuralları:

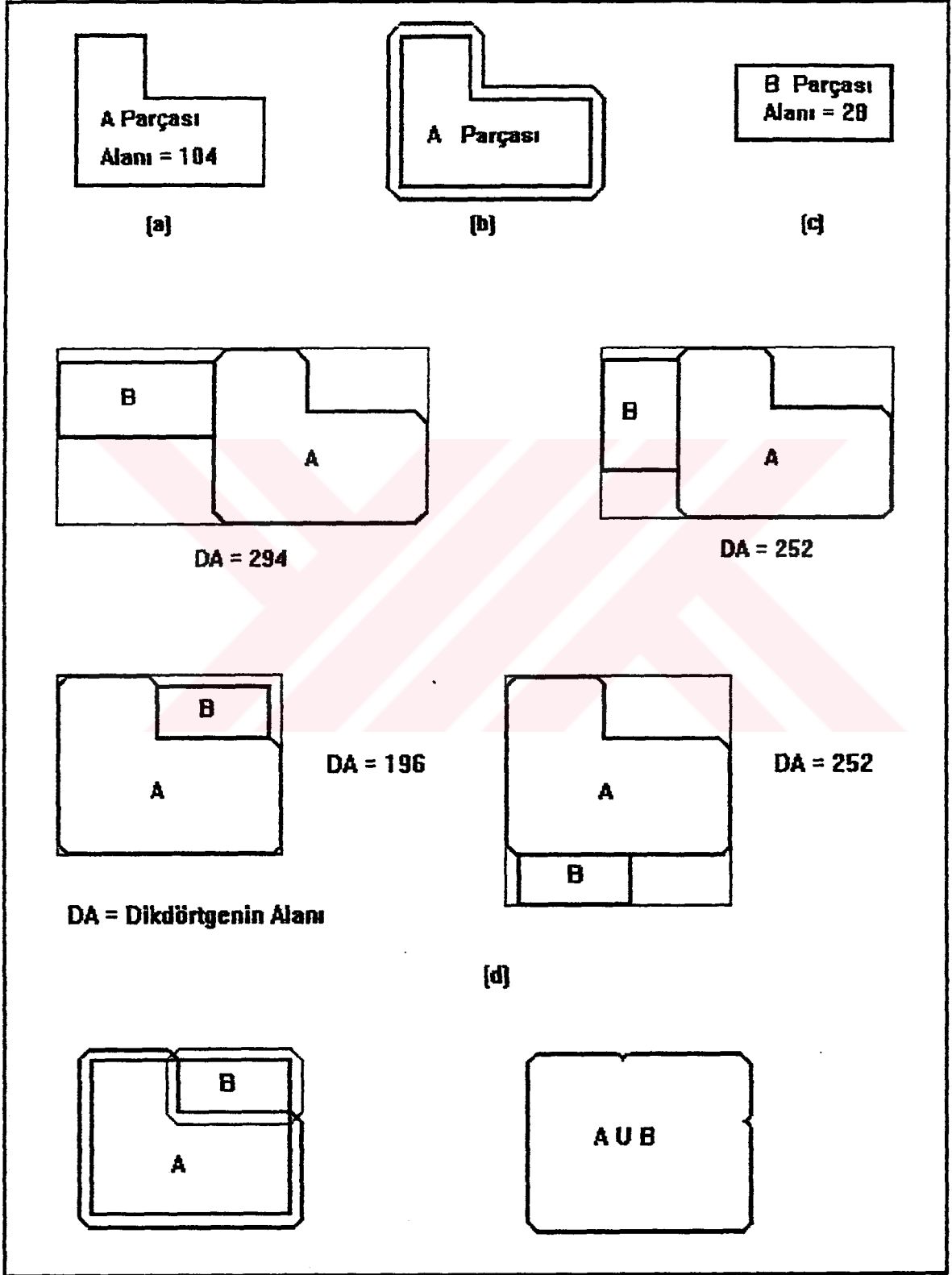
MXAR	Önce enbüyük alan
MNAR	Önce enküçük alan
MXIR	Önce enbüyük karmaşıklık değeri

MNIR	Önce enküçük karmaşıklık değeri
MXSI	Önce enbüyük kenar sayısı
MNSI	Önce enküçük kenar sayısı
MXCU	Önce enbüyük yay sayısı
MNCU	Önce enküçük yay sayısı
MXPE	Önce enbüyük çevre uzunluğu
MNPE	Önce enküçük çevre uzunluğu
FCFS	İlk gelene ilk öncelik
USER	Kullanıcının öncelik kuralı

Algoritmanın başlaması için, en yüksek önceliklere sahip iki şekil aday kümeden seçilmekte ve bunların birer kenarı yanyana gelecek biçimde oluşturulan alternatifler içinden eniyi görelî yerleşimi veren seçilmektedir. Eniyi görelî yerleşim, enküçük alana sahip dikdörtgen zarf içine sığan seçenek olarak tanımlanmaktadır. Daha sonra şekiller yerleştirildikleri gibi saklanmakta ve bir önceki aşamada yerleri belirlenen iki şekil birleştirilerek bir tek şekil durumuna getirilmektedir. Yani bir şeklin kümeden alınmasıyla algoritma tekrarlanmakta ve bu işlem aday kümedeki tüm diğer şekiller için sürdürülmektedir. Doğal olarak, çözüm sürecinde, "şekillerin üst üste binmemesi" ve "plaka sınırları içinde" olmasını gerektiren kısıtlar da sağlanmaktadır. Algoritmanın akış şeması Şekil 5.5 'de gösterilmiştir. Algoritmanın resimsel anlatımı ise Şekil 5.6'da görülmektedir.



Şekil -5.5- Algoritmanın akış diyagramı



Şekil -5.6- Algoritmanın resimsel anlatımı

- 1.Basamak Verilen girdileri oku. Kullanıcı tarafından yapılan tercihler ve hesaplanan öncelik kurallarına göre şekillerin işleme giriş sırasını sapta. Her şeklin alanını hesapla ve 2'inci basamağa geç.
- 2.Basamak Sıralanmış aday kümeden ilk şekli al. Eğer şekil plaka üzerine verildiği biçimde (şekil yönünü değiştirmeden) yerleştirilebiliyorsa 3'üncü basamağa geç;diğer durumda şekil "yerleştirilmemiş" olarak kaydet ve 2'inci basamağı tekrarla
- 3.Basamak Yerleştirilen şeklin koordinat değerlerini kaydet. Bu şeklin etrafına yapay bir şerit ekleyerek genişlet, buna "MAJOR" ismi ver (A şekli), alanını yenile ve 4'üncü basamağa geç.
- 4.Basamak Sıralanmış aday kümeden başka bir şekil al. Eğer bunun alanı plakanın kalan alanından büyükse şekil "yerleştirilmemiş" olarak kaydet ve 4'üncü basamağı tekrarla;en sonunda 10'uncu basamağa git. Diğer durumda,bunu bir başka aday olarak kabul et ve buna "MİNÖR"ismi ver (B şekli). MAJOR ve MİNÖR'ün her kenarı için 5,6,7 ve 8'inci basamakları tekrarla ve sonunda 9'uncu basamağa git.

- 5.Basamak MAJOR ve MİNÖR'ün incelenen kenarlarını, MİNÖR'ü MAJOR'e doğru kaydırarak yanyana getir ve gerekirse MİNÖR'ü bir nokta etrafında döndür. İşlem bitince 6'ıncı basamağa geç.
- 6.Basamak MAJOR ve MİNÖR'ü içine alan enküçük dikdörtgen zarfın alanını hesapla. Eğer bu alan o zamana kadar bulunmuş enküçük alandan daha da küçükse 7'inci basamağa geç. Diğer durumda, şekillerin başka kenarları için 5'inci basamağa geri dön.
- 7.Basamak MİNÖR'ün önceden yerleşmiş şekillerle üst üste binip binmediğini ve plaka sınırları dışına taşıp taşmadığını kontrol et. Eğer yanıt olumluysa 8'inci basamağa geç; diğer durumda 5'inci basamağa geri dön
- 8.Basamak Enküçük dikdörtgen zarfın alanını yenile, MİNÖR'ün yeni yerleşimini kaydet ve 5'inci basamağa geri dön
- 9.Basamak MİNÖR'ü yerleştirmek mümkün oladıysa, "yerleştirilememiş" olarak kaydet ve 4'üncü basamağa git; diğer durumda devam et. MİNÖR'ün plaka üzerindeki eniyi yerleşimini oku ve etrafına yapay şerit ekleyerek genişlet. MAJOR ve MİNÖR'ü birleştir ve bu yeni şekle "MAJOR" ismi ver (f). Aday kümede başka şekil kalmıyorsa 4'üncü basamağa geri dön, kalmıyorsa 10'uncu basamağa geç.

10. Basamak Enküçük x ile y koordinat değerlerini hesaplave bu değerleri yerleştirilmiş şekillerin koordinatlarından çıkartarak onları pozitif bölgeye kaydır. Şekillerin koordinat değerleri, resimsel görünümü ve plakayla ilgili bilgileri yaz. Yerleştirilmemiş şekiller varsa 11'inci basamağa geç, yoksa 13'üncü basamağa git.
11. Basamak İşleme sokulmamış plakalar varsa bir sonraki plakanın boyutlarını oku, yerleştirilmemiş şekilleri aday kümeye aktar ve 2'inci basamağa geri dön. Diğer durumda 12'inci basamağa geç.
12. Basamak Yerleştirilmemiş şekillerle ilgili çıktıyı yaz ve 14'üncü basamağa git
13. Basamak Kullanılmayan plakalarla ilgili çıktıyı yaz ve 14'üncü basamağa geç
14. Basamak Algoritma sonu

Dört yordamsal yaklaşımın karşılaştırılmasının yapıldığı bir araştırmada da istenen parçaların boyutları arasındaki mevcut oranlara göre, yaklaşımların etkinliklerinin değişimi tartışılmaktadır (İsrani, 1985). Bu araştırmada ayrıca, sınırlı miktarda insan müdahalesinin getirebileceği iyileşmelerden söz edilmektedir.

Daha önce açıklanan çalışmalar dışında kalan çalışmalarda, benimsenen çözüm yaklaşımının rassal yaklaşım olduğu veya dinamik programlama ve benzeri bir matematik programlama ile yordamsal yaklaşımın birlikte kullanıldığı görülmektedir.

Bir araştırmacı, yüzey kusurları içeren ruloların dilinmesi problemini, dinamik programlama ile ve rassal bir

yaklaşım ile ele almaktadır(Saker,1988).De Canı(1978), Ana malzeme içinde yer alabilen parçaların kenarlarının ana malzeme kenarlarına paralel veya dik olmasına gerek olmadığını (ortogonal olmayan yerleşim) ve bu durumun daha etkin bir yerleşim sağlayabileceğini savunmakla beraber kendisi herhangi bir çözüm önerisinde bulunmamaktadır. Kesilmesi gereken ana malzeme içinde çatlak ve benzeri diğer istenmeyen kusurların bulunması olasılıklarının gözönüne alınabildiği bir çalışma da, Christofides(1977) tarafından yapılmıştır.

Çalışmaların birinde ise, kesme problemlerinin çözümüne yönelik mevcut çalışmalar özetlendikten sonra, bu tür problemlerin çözümün uzman sistemler ve yapay zekalar yardımı ile mümkün olabilme şartları tartışılmaktadır(Dağlı,1988). Ayrıca bilgi esaslı çizelgelerin kullanılabileceği bir yaklaşım bu çalışmada tanıtılarak; endüşük düzeyli fireler verebilen kesme planlarını oluşturabilmek için, yöneylem araştırmasının mevcut teknikleriyle uzman sistemlerin bütünselleştirilmesinin gereği üzerinde durulmaktadır.

6. **DÜZENSİZ ŞEKİLLERİN YERLEŞTİRİLMESİ İLE İLGİLİ GELİŞTİRİLEN BİLGİSAYAR PROGRAMI**

Geçen bir kaç yıl içinde konu ile ilgilenen araştırmacılar, hazırladıkları çözüm yaklaşımlarını üç katagoride topladılar:

a.) El ile yapılan yerleştirme :

Parçaların yerleştirme işlemi temel olarak el ile yapılır. Her ne kadar bilgisayar kullanılsada, bilgisayar verimliliğın ve doğruluk değerinin arttırılması için kullanılır. Yerleştirilecek parçaların tanımlanması bilgisayar ile yapılır. Bir Plotter parçaların kopyalarını çizer. Parçaları örnekleycen modeller kesilir ve paftanın ebatlarına ve şekline uygun bir çerçeve içine el ile yerleştirilir.

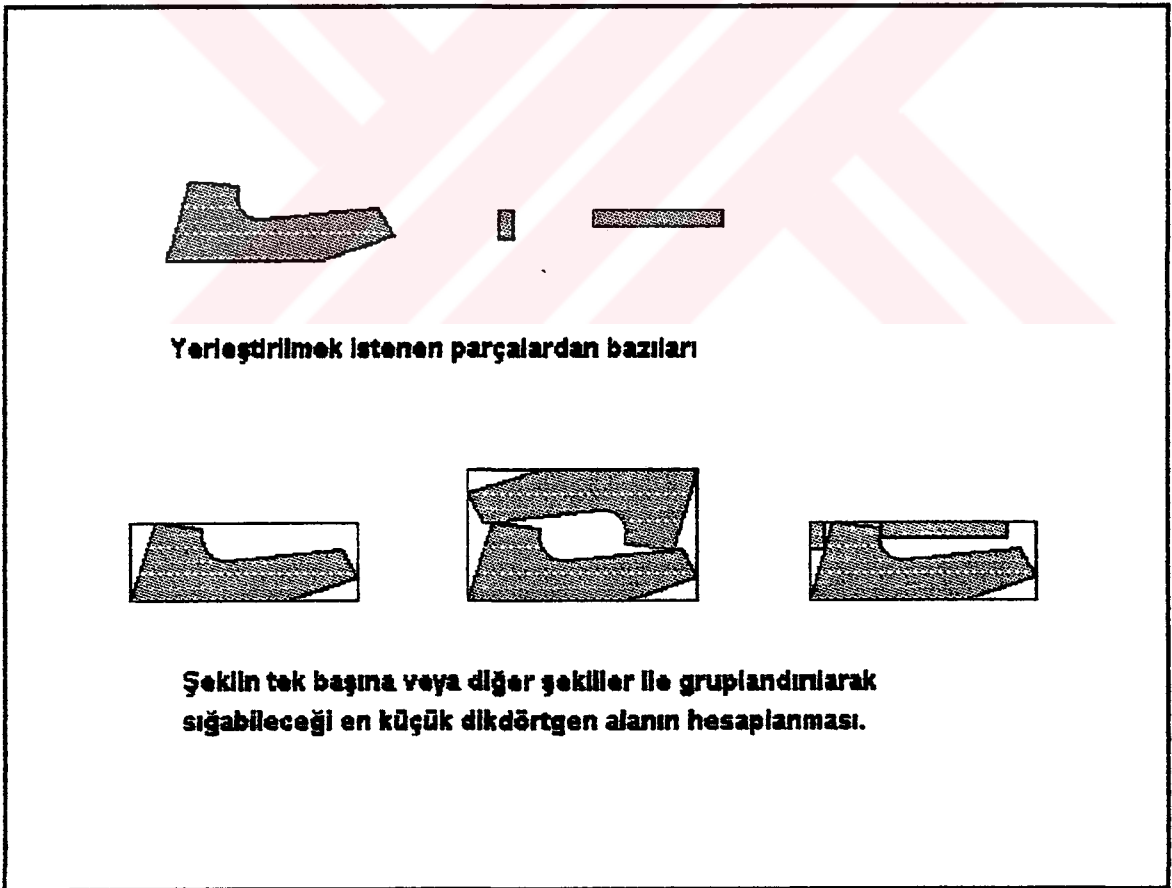
b.) Yarı otomatik yerleştirme :

Parçaların yerleştirme işlemi kullanılan etkileşimli bir grafik ekran üzerinde tamamlanır. Yerleştirilecek parçalar ve kullanılacak pafta, ekran üzerinde görüntülenir. Kullanıcı, çeşitli giriş ünitelerini (Mouse, Digitizer, Joystic v.b) kullanarak parçaları çevirebilir veya bir noktadan başka bir noktaya taşıyabilir. Yerleştirme işlemi süresince bilgisayar, parçaların üst üste binip binmediğini, fire miktarını ve diğer ortaya çıkması muhtemel hataları kontrol eder.

c.) Otomatik yerleştirme :

Otomatik olarak yerleştirme yapan çeşitli metodlar geliştirilmiştir. Bu konudaki çalışmaları da iki alt grupta incelemek mümkündür. Bu grupların birincisi yerleştirme

işlemi için iki basamaklı operasyon içermektedir. İlk basamakta, yerleştirilecek olan düzensiz şekiller tek başlarına veya gruplandırılarak ele alınır ve geometrinin etrafını tam olarak kapatan dikdörtgen zarf hesaplanmaya çalışılır. Bu yaklaşım Şekil 6.1'de anlatılmaya çalışılmıştır. En iyi dikdörtgen zarfın hesaplanmasından sonra ise lineer programlama metodları kullanılarak problem çözümlenir. Bu çözüm yöntemi, talep edilen parçaların tam olarak elde edilmesinin istendiği, parça çeşit sayısının az, parça talep miktarlarının büyük olduğu durumlarda kullanılmaktadır.



Şekil -6.1- En iyi dikdörtgen zarfın hesaplanması

İkinci grupta ise, herhangi bir dikdörtgen tanımlama işlemi yapılmaksızın direkt parça geometrisine ait bilgilerin kullanıldığı ve tanımlanan sezgisel kurallar ile çözümün aranması esasına dayalı bir metoddur. Genellikle yerleştirilecek parçalardan biri kesin bir pozisyonda, pafta üzerinde konumlandırılır. Tüm araştırma boyunca bu parçanın konumu ve yerleştirilme şekli ile ilgili yeni bir araştırma yapılmaz. Fakat bu dezavantaj Adamowicz'in algoritmasında görülmemektedir. Şöyleki, birinci parçanın yerleştirilmesini takiben problem, lineer programlama tekniklerini kullanmakta ve bir sonraki araştırma için yeni sınırlar tesbit etmektedir. Fakat, birçok yönden en iyi neticeleri vermelerine rağmen işlem zamanının uzun olması ve hesaplama süresinin maliyetinin çok yüksek olması nedeni ile bu tür algoritmalar pek kullanılmamaktadırlar.

6.1. ÇÖZÜM YAKLAŞIMI

Ek A'da sunulan bilgisayar programında, düzensiz şekillerin, ebatları belirli, limitsiz sayıdaki, ana pafta üzerine, istenen parça sayısını tam olarak temin edecek şekilde, minimum ana pafta kullanımı ile elde edilmesi temel amaç olarak ele alınmıştır. Bu konuda yeni bir sezgisel metod geliştirmek yerine, yaptığım incelemelerde en sade ve gelişmiş mantığı içerdiğini düşündüğüm A.Y.C.Nee ,K.W.Seow ve S.L.Long'un geliştirdikleri algoritmanın bir bölümünü ve A.Cemil Dikili'nin kullandığı lineer programlama yöntemini temel aldım. Programımda kullandığım algoritma esas itibarı ile yukarıda açıklamaya çalıştığım, düzensiz şekiller etrafında en iyi kapalı dikdörtgenlerin tesbit edilmesi ve lineer programlama yöntemleri ile plakalar üzerine parçaların yerleştirilmesi esasına dayanmaktadır. Şimdi bu çalışmanın detaylarını vermeye çalışacağım.

6.1.1. ŞEKİLLERİN HAZIRLANMASI

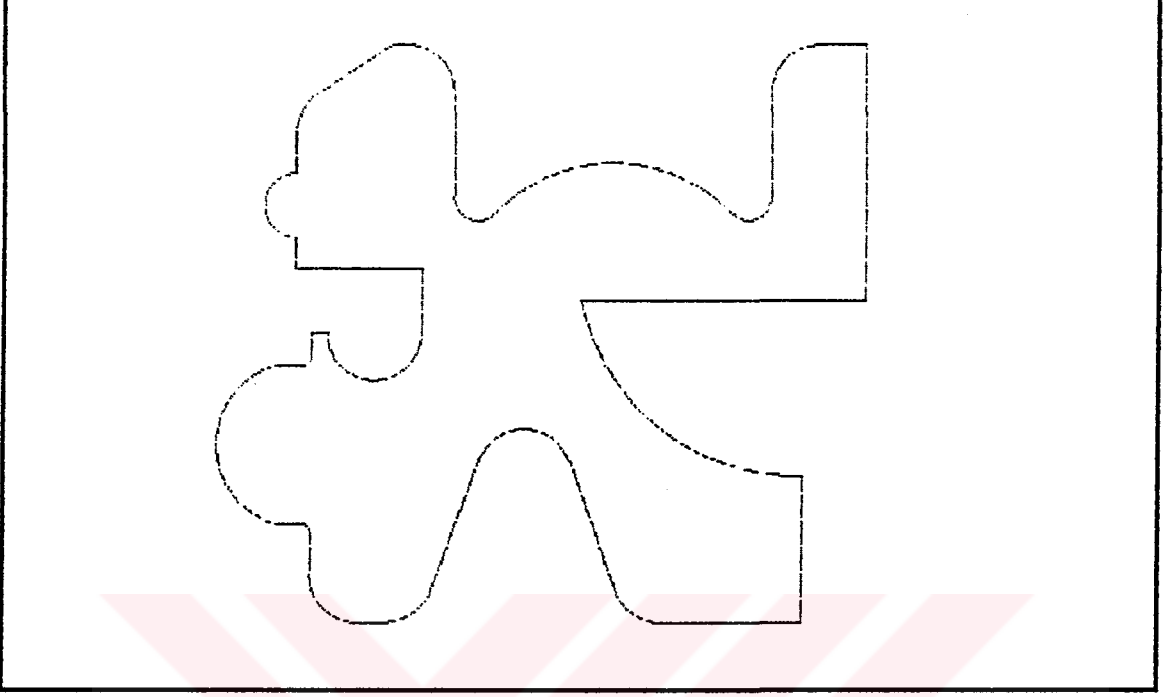
Ebatları belirli ana pafta üzerine yerleştirilecek olan şekiller, herhangi bir CAD programının içerdiği basit çizim komutları (Line, Arc, Circle, Fillet, v.b komutlar) kullanılarak kolaylıkla oluşturulur.

Şekil etrafında, kullanılacak olan kesme düzeneğinin özellikleri göz önüne alınarak tespit edilecek, kesim payı miktarı kadar ek bir şerit alan daha CAD sistemi içinde tanımlanmalıdır.

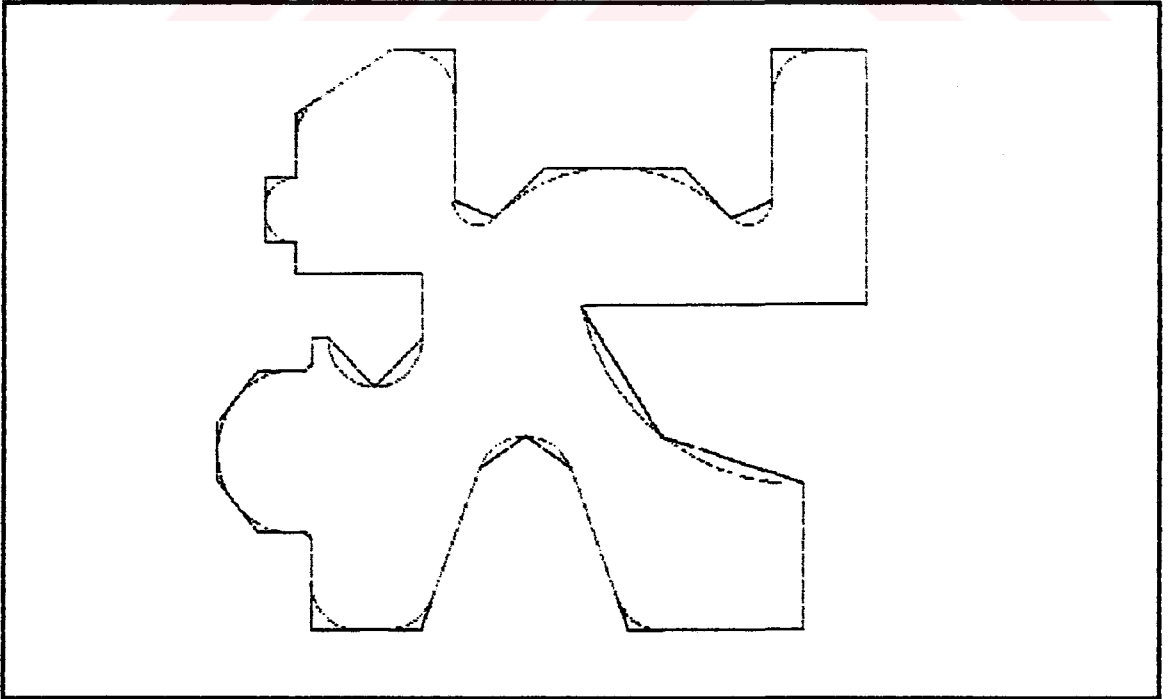
Geometri tanımlandıktan sonra, profilin dış kısmında görülen dairesel elemanlar üzerinde belirli bir giriş uzunluğu göz önüne alınmalıdır. Bu giriş uzunluğunun küçük seçilmesi hesaplama zamanını arttıracaktır. Böylece şeklin geometrisi ne kadar karmaşık olursa olsun tüm profil için çizgilerden oluşan bir polyline tanımlanmış olacaktır. Profilin iç kısmında yer alması muhtemel dairesel profiller ve eğrilerden oluşan geometrik tanımlar göz önüne alınmamıştır.

Açıklanan geometrik yaklaşım Şekil 6.2 ve Şekil 6.3 de açıklanmıştır.

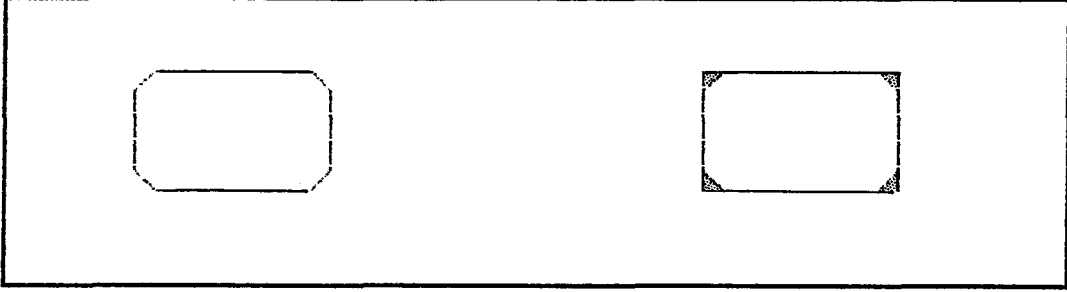
En iyi dikdörtgen alanın hesaplanması sırasında yerleştirilecek şekillerin gruplandırılması söz konusu olursa, bu gruplandırma işlemi, hem optimum yerleştirme zamanını kısaltmakta hemde kayıp malzeme miktarının aaltılmasında önemli rol oynamaktadır. Şöyleki, tanımlanan şekil ve etrafında tanımlanabilecek en iyi dikdörtgen parça Şekil 6.4 deki gibi olsun. Böyle bir durumda ek bir gruplama işlemine girmek gereksiz görülmektedir. Fakat tanımlanacak parça Şekil 6.5 deki gibi ise gruplama işlemi yapmak avantajlı olacaktır. Görüldüğü gibi, bu şekilde oluşturulacak dikdörtgen parçaların yan yana yerleştirilmesi halinde kayıp malzeme miktarı artacaktır. Şekil 6.6 da aynı parça için gruplama işlemi yapılmıştır.



Şekil -6.2- Çizim komutları ile CAD sisteminde oluşturulan resim



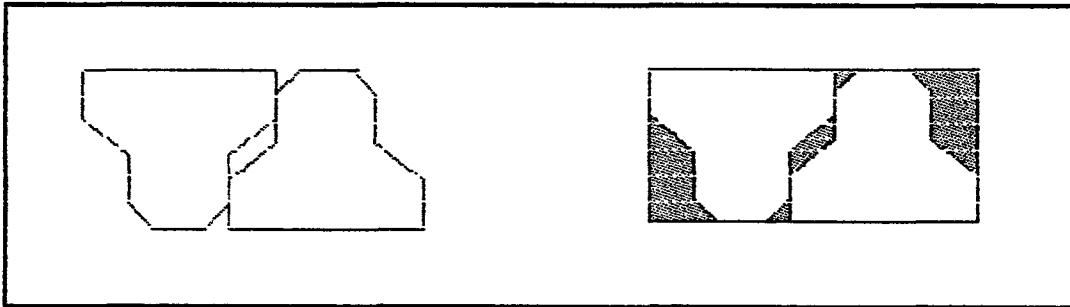
Şekil -6.3- Geometri etrafında tanımlanan polyline



Şekil -6.4- Basit Geometri ve en iyi dikdörtgen zarf



Şekil -6.5- Karmaşık geometri ve en iyi dikdörtgen zarf



Şekil -6.6- Gruplama İşlemi

Talep edilen her bir parçanın geometrisi CAD sisteminde yukarıdaki esaslar altında çizilmelidir. Eğer geometri üzerinde gruplama işlemi yapılacak ise bu işlemlerde CAD sistemi içinde tamamlanmalıdır. Kullanılan CAD sistemi içinde hazırlanan geometrinin DXF çıktısı alınır. Bilindiği gibi DXF dosya yapısı ilk defa AUTODESK firmasının hazırladığı AUTOCAD programının çeşitli versiyonlarında mevcut olan bir bilgi aktarım formatıdır. Çok yaygın olduğundan bir çok program tarafından kolaylıkla yazılıp okunur.

6.1.2 GEOMETRİK ŞEKİLE AİT BİLGİLERİN ÜRETİLMESİ

İki boyutlu yerleştirme probleminde, şekiller özelliklerine göre gruplandırıldıkları ve yerleştirildikleri için, şekillere ait bazı bilgilerin sürekli üretilmesi gerekmektedir.

6.1.2.1 GEOMETRİK ŞEKLİN ALANININ HESAPLANMASI

Sadece çizgilerden oluşan herhangi bir geometrik şeklin n adet köşesinin olduğunu düşünelim. Köşe koordinatları, pozitif yönde (saat yelkovanının dönüş yönünün tersi yönünde), sıra ile verildiğinde, geometrik şeklin alanı aşağıdaki formül yardımıyla bulunur.

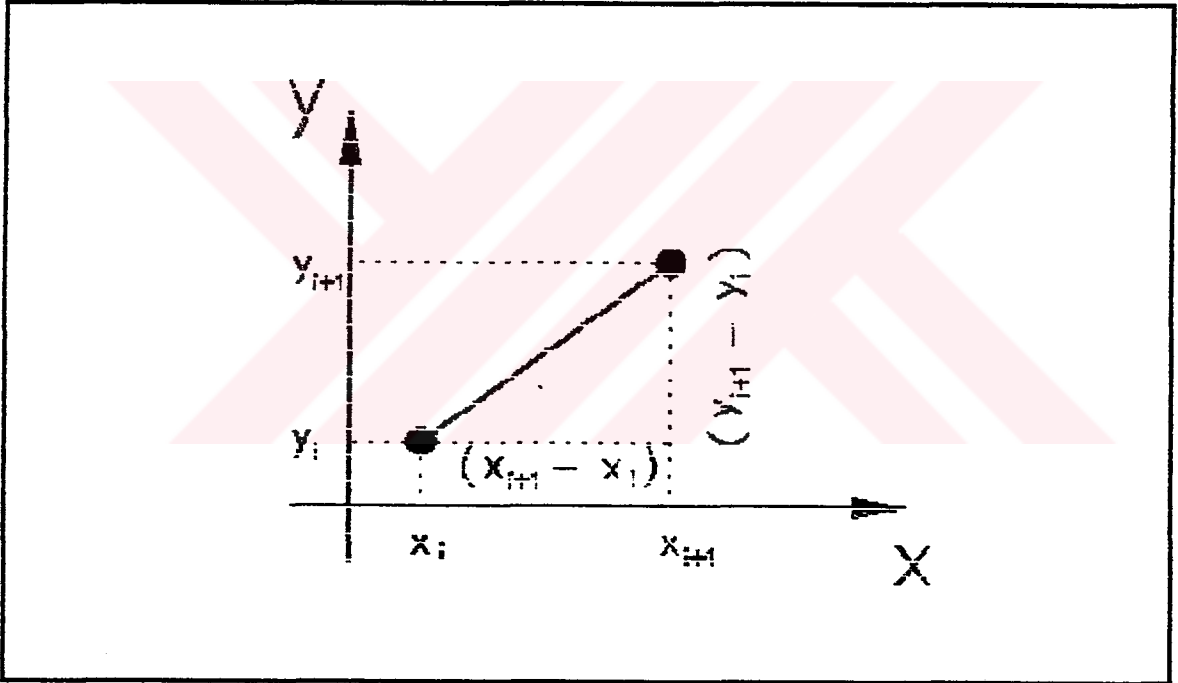
$$S = 0.5 \left\{ y_1(x_n - x_2) + y_n(x_{n-1} - x_1) + \sum_{i=2}^{n-1} y_i(x_{i-1} - x_{i+1}) \right\}$$

Şayet geometrik şeklin köşe koordinatları negatif yönde, sıralı olarak verilmiş ise, yukarıdaki formül ile

hesaplanan alan değeri negatif olacaktır. Hesaplanan alan değerinin mutlak değeri alınacak olursa koordinatların tanımlanma yönü öncemsiz hale gelecektir.

6.1.2.2 GEOMETRİK ŞEKLİN KENAR UZUNLUKLARININ HESAPLANMASI

(x_i, y_i) ve (x_{i+1}, y_{i+1}) noktalarını birleştiren doğrunun uzunluğu, :



Şekil -6.7- Kenar uzunluğunun testipi

$$d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

formülüyle bulunabilir. Bu formül ile şeklin bütün köşe noktaları için kenar uzunlukları hesaplanabilir.

6.1.2.3 GEOMETRİK ŞEKİLİN KENARLARININ POZİTİF X EKSENİ İLE YAPTIĞI AÇILARIN HESAPLANMASI

Geometrik şeklin köşe koordinatları bir referans noktasından başlayarak pozitif yönde sıralı olarak verilmiş olsun..Herhangi bir köşenin referans noktasına göre konumunu belirlemek için, şeklin kenarlarının pozitif x eksenini ile yaptığı θ_i ($i=1,\dots,n$) (n :şekilin köşe sayısı), açısını 0° ile 360° arasında belirlemek gerekir.

Böylece, (x_i, y_i) ile (x_{i+1}, y_{i+1}) noktaları arasındaki kenar için θ_i açısı aşağıdaki bağıntılar yardımıyla bulunabilir.

$$\begin{array}{ll} x_i = x_{i+1} \text{ ise} & y_{i+1} > y_i \text{ iken } \theta_i = 90^\circ \\ & y_{i+1} < y_i \text{ iken } \theta_i = 270^\circ \\ y_i = y_{i+1} \text{ ise} & x_{i+1} > x_i \text{ iken } \theta_i = 0^\circ \\ & x_{i+1} < x_i \text{ iken } \theta_i = 180^\circ \end{array}$$

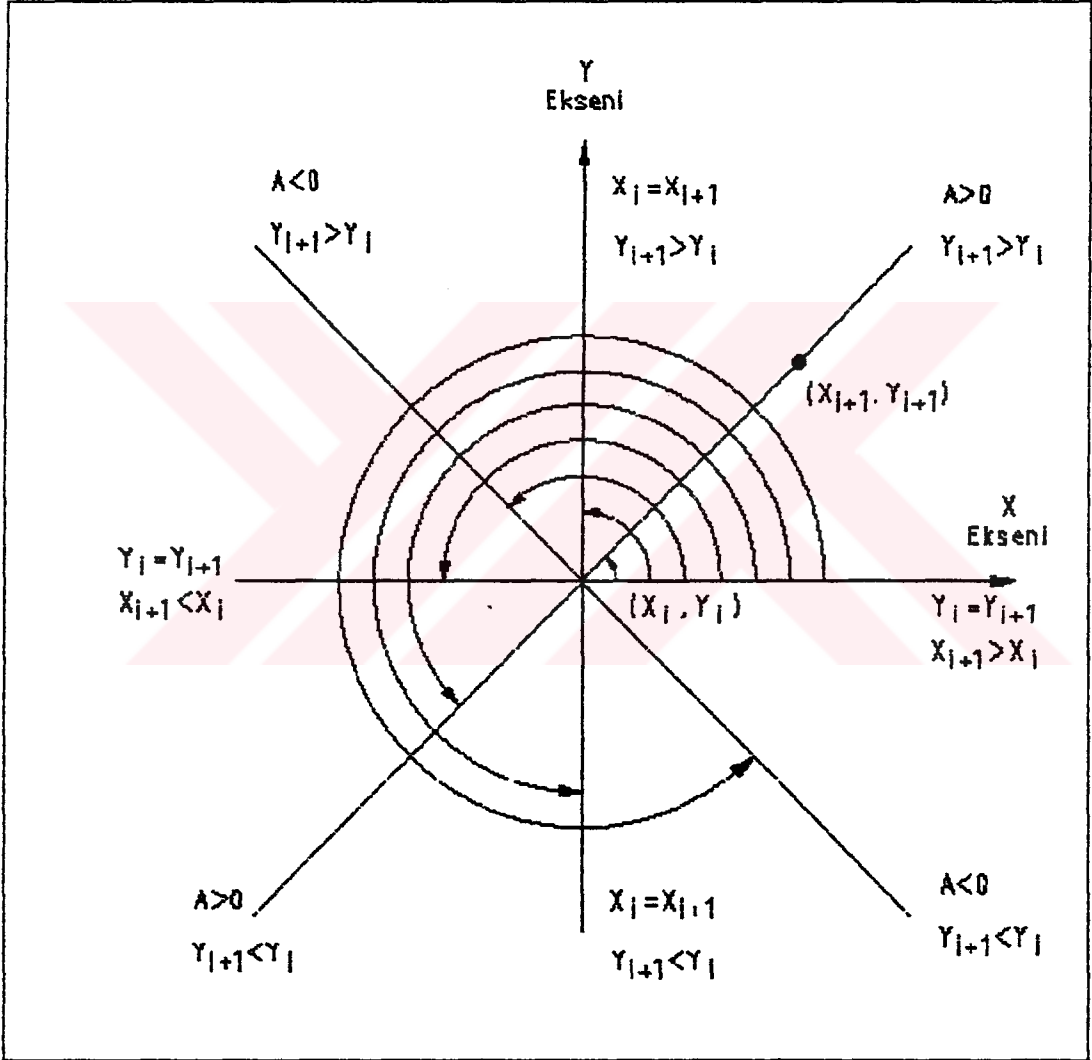
Diğer durumlarda ise

$$A = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

olmak üzere:

$$\begin{array}{ll} A > 0 \text{ ise} & y_{i+1} > y_i \text{ iken } \theta_i = \text{Arctg}(A) \\ & y_{i+1} < y_i \text{ iken } \theta_i = \text{Arctg}(A) + 180^\circ \\ A < 0 \text{ ise} & y_{i+1} > y_i \text{ iken } \theta_i = \text{Arctg}(A) + 180^\circ \\ & y_{i+1} < y_i \text{ iken } \theta_i = \text{Arctg}(A) + 360^\circ \end{array}$$

Kenarların pozitif X eksenini ile yaptığı açılar Şekil 6.7 de görülmektedir.



Şekil -6.7- Geometrisi verilen parçanın kenarlarının X eksenini ile yaptığı açının tespiti

6.1.2.4 ŞEKLİN KENARLARI ARASINDAKİ İÇ AÇILARIN HESAPLANMASI

Şeklin i ve $i+1$ kenarlarının pozitif x eksenine ile yaptığı açılar, θ_i ve θ_{i+1} ise, bu kenarlar arasındaki iç açı, ϕ_i aşağıdaki bağıntılar ile bulunur:

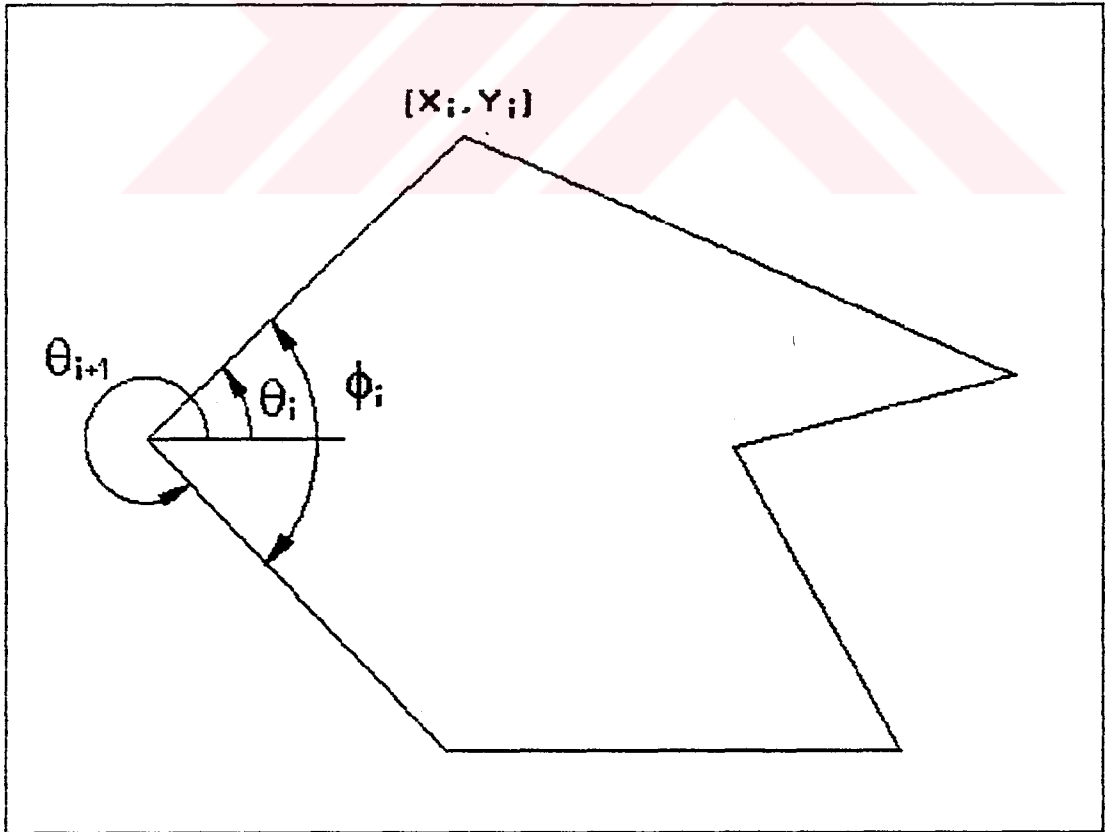
$$i+1-180 > \theta_i \quad \text{ise} \quad i=360-\theta_{(i+1)}+\theta_i+180$$

$$i+1 < \theta_i-180 \quad \text{ise} \quad i=\theta_i-\theta_{(i+1)}-180$$

Aksi takdirde

$$i=180+\theta_i-\theta_{(i+1)}$$

bağıntıları ile bulunur.



Şekil -6.9- Kenarlar arasındaki iç açılarının tespiti

İç açıların değerlerine göre, iki kenar arasındaki köşenin girinti veya çıkıntı teşkil ettiğine karar verilebilir.

ve $i+1$. kenarların iç açısı ϕ_i için:

$i < 180$ ise (x_{i+1}, y_{i+1}) köşesi ÇIKINTI

$i > 180$ ise (x_{i+1}, y_{i+1}) köşesi GİRİNTİ

teşkil eder.

6.1.2.5 GEOMETRİK ŞEKİLİN KONUMUNUN DEĞİŞTİRİLMESİ

İki boyutlu yerleştirme veya diğer bir deyişle stok kesme problemlerinin gerek şekillerin gruplanması safhasında gerekse yerleştirme safhasında şekillerin konumları sıkça değiştirilmektedir. Şekil belli bir miktar ötelenmekte, verildiği konumda yerleştirme şansı yoksa döndürülmekte veya katlanmaktadır. Bir şeklin konumu değiştirildiğinde, tabii ki şeklin kenar uzunlukları ve iç açıları değişmeyecektir. Fakat şeklin koordinatları ve kenarlarının pozitif x eksenini ile yaptığı açılar değişecektir.

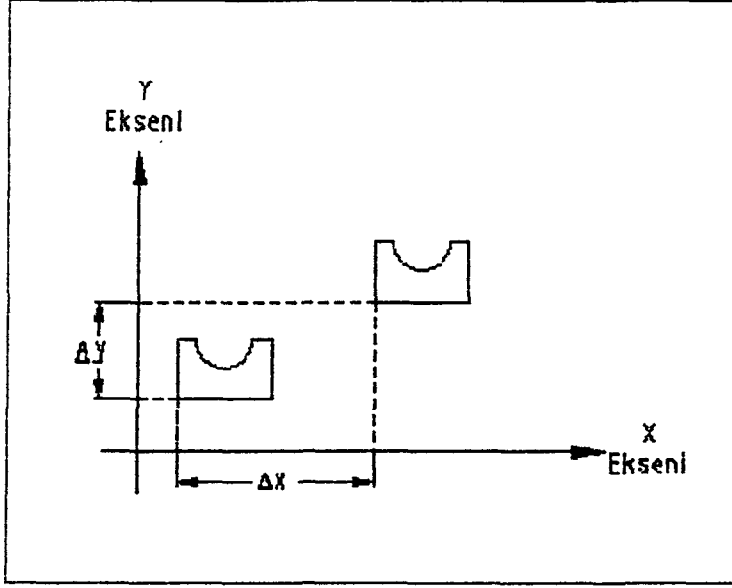
6.1.2.5.1 ŞEKLİN ÖTELENMESİ

n adet köşeye sahip geometrik şeklin mevcut konumundan $\Delta x, \Delta y$ birim kadar uzaktaki yeni konumuna ötelendiğini düşünelim. Yani köşe noktalarının koordinatları (x_i, y_i) $\{i=1, \dots, n\}$ n : Köşe sayısı} olan bir şekil $(\Delta x, \Delta y)$ kadar ötelendiğinde, yeni koordinatları, (X, Y) :

$$X = x_i + \Delta x$$

$$Y = y_i + \Delta y$$

bağıntıları ile hesaplanır. $(\Delta x, \Delta y)$ öteleme miktarları pozitif ise şekil pozitif yönde, negatif ise şekil negatif yönde ötelenir.



Şekil -6.10- Bir şeklin ötelenmesi

Bir şekil ötelendiğinde, kenarlarının pozitif x eksenini ile yaptığı açılar değişmez.

6.1.2.5.2 ŞEKLİN DÖNDÜRÜLMESİ

Bir şeklin dönmeden önceki koordinatları (x_i, y_i) $\{i=1, \dots, n \text{ köşe sayısı}\}$ iken, α kadar döndürüldükten sonraki yeni konumunun koordinatları (x_d, y_d) :

$$\begin{aligned} \alpha > 0 \quad \text{ise} \quad x_d &= x_i \cos(\alpha) - y_i \sin(\alpha) \\ y_d &= x_i \sin(\alpha) + y_i \cos(\alpha) \quad \{i=1, \dots, N\} \end{aligned}$$

$$\begin{aligned} \alpha < 0 \quad \text{ise} \quad x_d &= x_i \cos(\beta) + y_i \sin(\beta) \quad \beta = |\alpha| \\ y_d &= -x_i \sin(\beta) + y_i \cos(\beta) \end{aligned}$$

bağıntıları ile bulunabilir.

$\beta = |\alpha|$ ve $L = \alpha / |\alpha|$ olmak üzere genelleştirirsek;

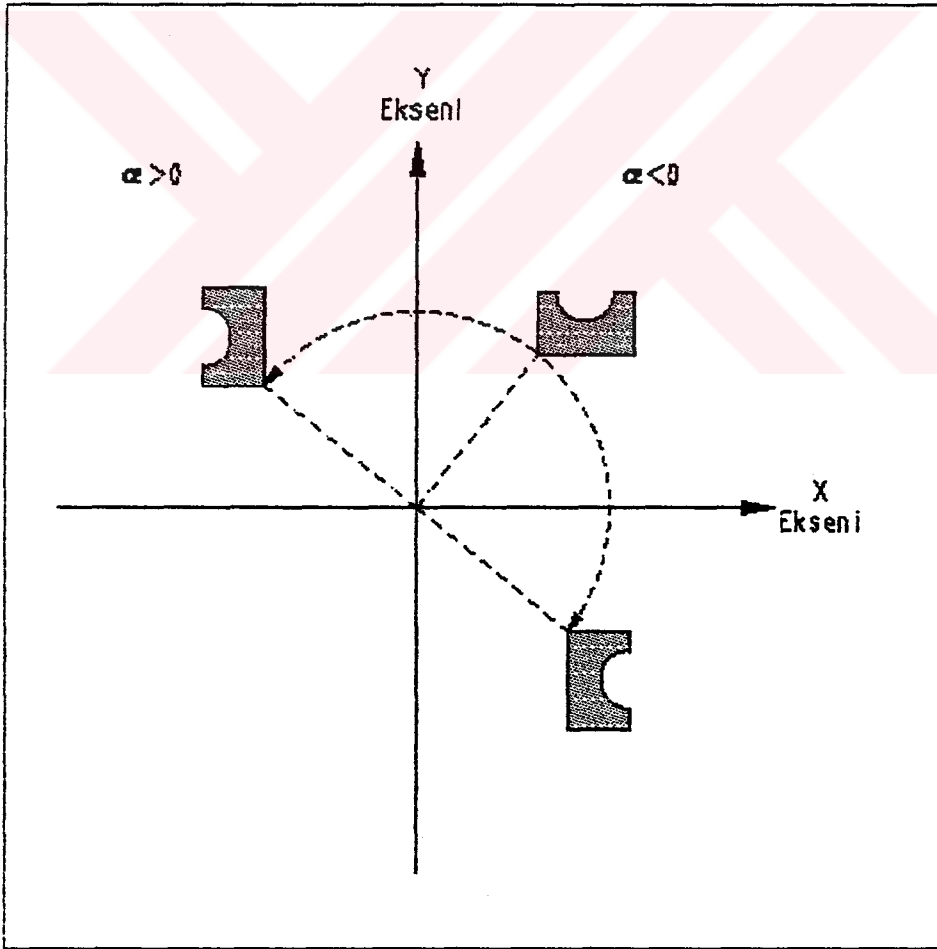
$$x_d = x_i \cos \beta - L y_i \sin \beta$$

$$y_d = L x_i \sin \beta + y_i \cos \beta$$

şeklinde yeni koordinatları elde ederiz.

Şeklin kenarlarının pozitif x eksenine ile yaptığı açılar ise, α kadar artar veya azalır.

$$i' = \theta_i + \alpha \quad \{i=1, \dots, N\}$$



Şekil -6.11- Bir şeklin döndürülmesi

6.1.2.5.3 ŞEKLİN KATLANMASI

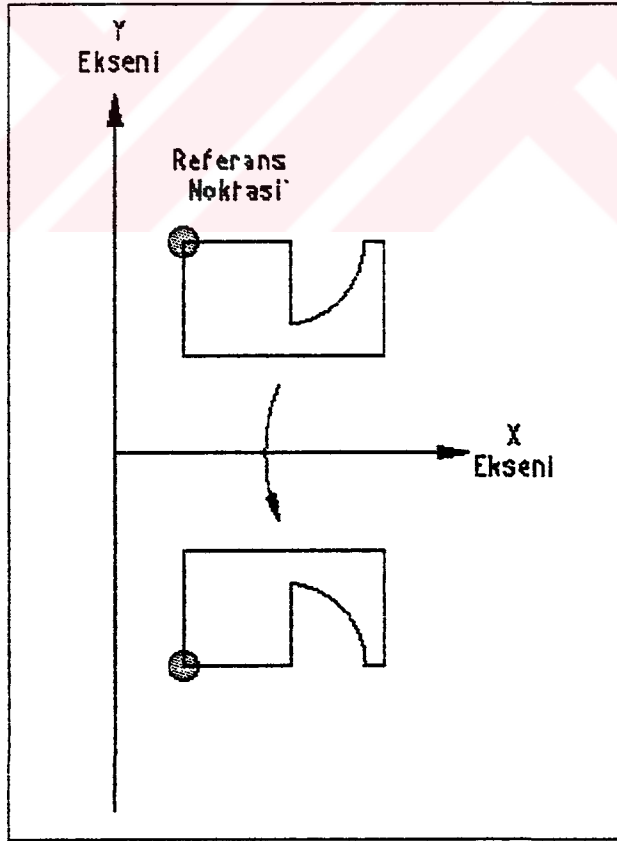
a.) Şeklin X Eksenine Göre Katlanması:

Bir şekil x eksenine göre katlandığında, köşe noktalarının x değerleri sabit kalır, y değerleri ise işaret değiştirir. Bir şeklin katlamadan önceki koordinatları (x_i, y_i) $\{i=1, \dots, n \text{ n:köşe sayısı}\}$ ise x eksenine göre katlamadan sonraki koordinatları (x_k, y_k) :

$$x_k = x_i$$

$$y_k = -y_i$$

şeklinde bulunabilir.



Şekil -6.12- Geometrinin X eksenine göre katlanması

x eksenine göre katlamadan önce bir kenarın pozitif x eksenine ile yaptığı açı $\{i=1, \dots, n\}$ ise, katlamadan sonra aynı kenarın pozitif x eksenine ile yaptığı açı, $\{i=1, \dots, n\}$:

$$c = 180 - \theta_i \text{ olmak üzere}$$

$$c > 0 \text{ ise } \theta_i' = C$$

$$c < 0 \text{ ise } \theta_i' = 360 + C$$

bağıntıları ile bulunabilir.

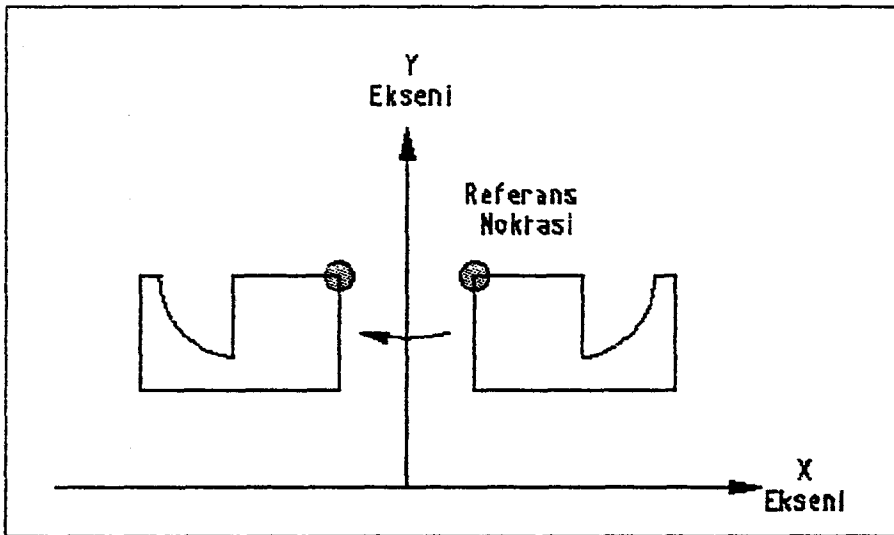
b.) Şeklin Y Eksenine Göre Katlanması:

Bir şekil y eksenine göre katlandığında, köşe noktalarının y değerleri sabit kalır, x değerleri ise işaret değiştirir. Bir şeklin katlamadan önceki koordinatları (x_i, y_i) $\{i=1, \dots, N; N: \text{köşe sayısı}\}$ ise ,y eksenine göre katlamadan sonraki koordinatları, (x_{ky}, y_{ky}) :

$$x_{ky} = -x_i$$

$$y_{ky} = y_i$$

şeklinde bulunabilir. Bir şeklin katlamadan önceki referans noktası (en üst sol köşesi) katlamadan sonra en üst sağ köşe olur. Katlamadan sonraki şeklin yeni konumunun koordinatları negatif (saat yönünde) yöndedir.



Şekil -6.13- Geometrinin Y eksenine göre katlanması

Y eksenine göre katlamadan önceki bir kenarın pozitif x eksenine ile yaptığı açı θ_i ($i=1, \dots, N$) ise, katlamadan sonra aynı kenarın pozitif x eksenine ile yaptığı açı, θ_i' ,

$$\theta_i' = 360 - \theta_i$$

bağıntısı ile bulunabilir.

6.1.2.6 GEOMETRİK ŞEKİLİN SİMETRİKLİĞİNİN BELİRLENMESİ

İki boyutlu yerleştirme problemlerinde şekillerin simetriklik bilgisi, şekilleri gruplama safhasında oldukça önemlidir. Çünkü simetrik bir şekli simetri eksenine göre katlayıp, yerleştirirerek oldukça iyi bir yerleşim elde edebiliriz. Simetrik bir şekil, iki parçanın en iyi bileşiminin belirlenmesi aşamasında işlem sayısını yarıya indirmektedir. Ayrıca simetrik bir şekli herhangi bir eksene göre katlamaya gerek yoktur. Şekli döndürerek istenilen kenara erişebiliriz.

Bir Şeklin X veya Y Eksenine Göre Simetrikliğinin Belirlenmesi:

Bir şeklin herhangi bir eksene göre simetrikliğinin belirlenmesi için aşağıdaki yöntem kullanılmaktadır.:

a.) X eksenine göre simetriklik için :

$$A_1 = \text{Enb}(y_i) \quad 1 \leq i \leq n$$

$$A_2 = \text{Enk}(y_i) \quad 1 \leq i \leq n$$

Y eksenine göre simetriklik için,

$$A_1 = \text{Enb}(x_i) \quad 1 \leq i \leq n$$

$$A_2 = \text{Enk}(x_i) \quad 1 \leq i \leq N$$

b.) Simetri ekseninin yeri

$$AB = \frac{(A_1 + A_2)}{2} \quad \text{ve} \quad B = |AB|$$

olarak hesaplanabilir.

c.) X eksenine göre simetriklik için,

$$i = y_i ; \beta_i = x_i ; 1 \leq i \leq N ; N = \text{Şeklin köşe sayısı}$$

Y eksenine göre simetriklik için,

$$i = y_i ; \beta_i = x_i ; 1 \leq i \leq N ; N = \text{Şeklin köşe sayısı}$$

olmak üzere:

$i < AB$ ise

$YB(J) = i$ $J: AB$ 'den küçük α_i değerine sahip
köşelerin sayısı

$i > AB$ ise

$YE(L) = i$ $L: AB$ den büyük α_i değerine sahip
köşelerin sayısı

olarak YB ve YE dizileri tayin edilir.

d.) $J \neq L$ ise şekil β eksenine göre simetrik değildir.

$J = L$ ise şekil β eksenine göre simetrik olabilir.

e) $M1 = YB(K) \quad K = 1, \dots, J$

$M2 = YE(K1) \quad K1 = J, \dots, 1$

olmak üzere $\beta(M1) = \beta(M2)$ ve

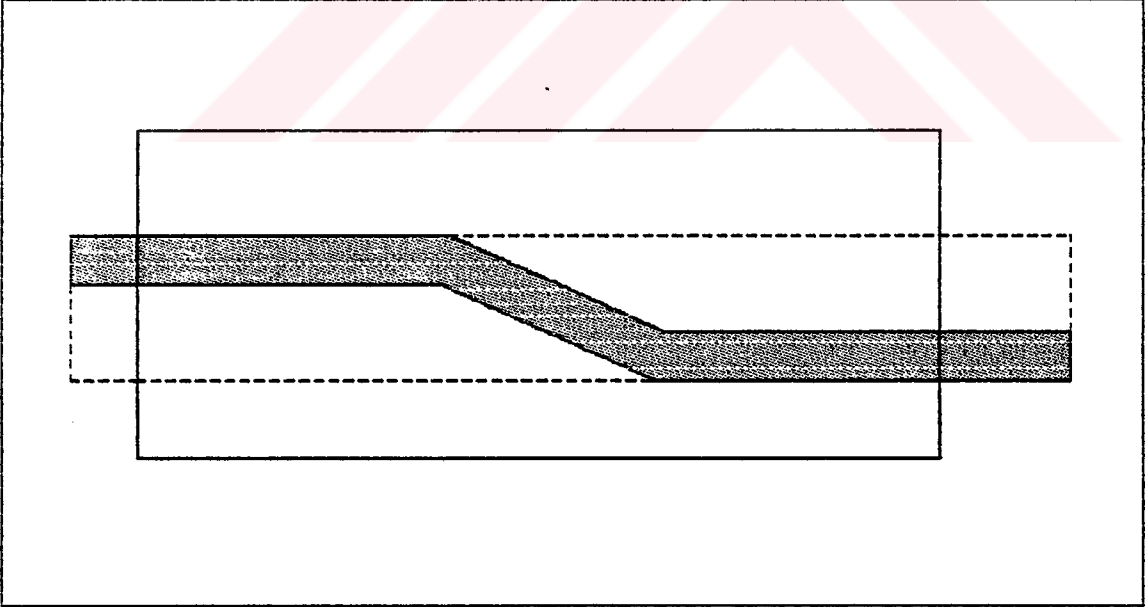
$\alpha(M1) + B = \alpha(M2) + B$ ise

Şekil β eksenine göre simetriktir. Aksi takdirde simetrik değildir.

6.1.3 ŞEKLİN SİĞDİĞİ EN KÜÇÜK DİKDÖRTGENE AİT BOYUTLARIN, ALANIN VE KARMAŞIKLIK DEĞERİNİN HESAPLANMASI

6.1.3.1 ŞEKLİN SİĞDİĞİ EN KÜÇÜK DİKDÖRTGENE AİT BOYUTLARIN VE ALANIN HESAPLANMASI

İki boyutlu yerleştirme problemlerinde, her şeklin sığdığı en küçük zarfın boyutlarını ve alanını belirlemek gereklidir. Çünkü yerleştirilmek istenen şeklin alanı, plaka alanından çok küçük olmasına rağmen şekil plakaya yerleştirilemeyebilir.



Şekil -6.14- Geometrik elemanın alanı, plaka alanından küçüktür.

Fakat parça plakaya sığmaz.

N adet köşesi olan bir şeklin sığdığı, dikdörtgenin boyutları,:

$$X_b = | E_{nb}(x_i) - E_{nk}(x_i) | \quad 1 \leq i \leq N$$

$$Y_b = | E_{nb}(y_i) - E_{nk}(y_i) | \quad 1 \leq i \leq N$$

bağıntıları, alanı, A_k ise :

$$A_k = X_b * Y_b$$

bağıntısı ile bulunabilir.

Bir şeklin sığdığı en küçük dikdörtgene ait boyutlar ve alan ise, şeklin her bir kenarını x veya y eksenine paralel olacak şekilde şekli döndürüp, elde edilen yeni konumlar için şeklin sığdığı dikdörtgenler arasından en küçük alanlısı seçilerek bulunur. En küçük dikdörtgenin alanı, A :

$$A = \min(A_k)$$

bağıntısıyla bulunur.

En küçük dikdörtgenin boyutları (x_e, y_e) ve ,plakanın boyutları (x_h, y_h) iken,

$$x_e \leq x_h \text{ ve } y_e \leq y_h$$

şartları sağlanırsa, şekil plakaya yerleştirilebilir. Eğer,

$$x_e \leq y_h \text{ ve } y_e \leq x_h$$

şartları sağlanırsa, şekil 90° döndürülerek plakaya yerleştirilebilir. Şayet yukarıdaki şartların her ikisinde sağlanmıyorsa, şekil plakaya yerleştirilemez.

6.1.3.2 KARMAŞIKLIK DEĞERİNİN HESAPLANMASI

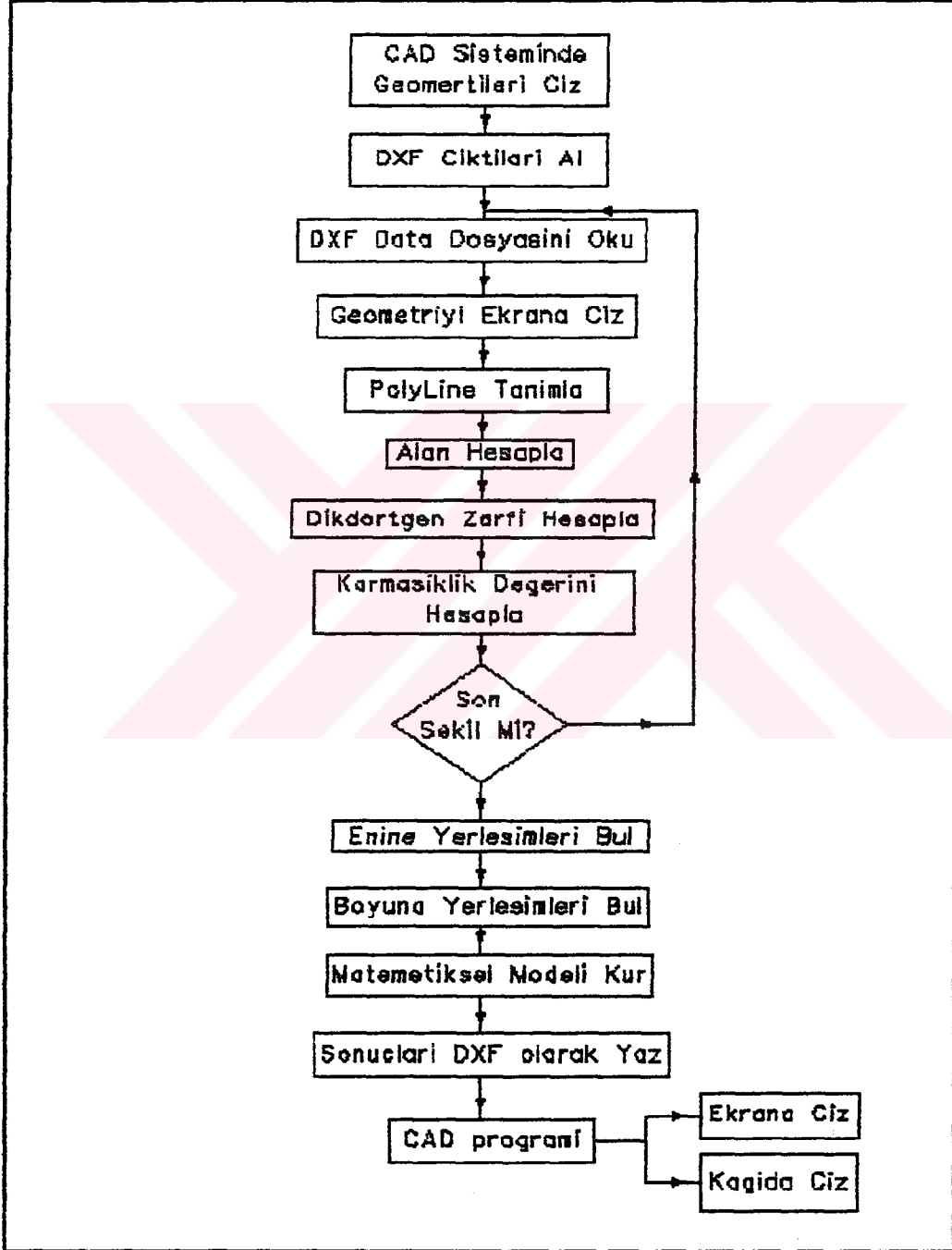
Parçanın karmaşıklık değeri (KD) , şeklin sığdığı en küçük dikdörtgenin alanının ,şeklin alanına oranı olarak tanımlanır. Karmaşıklık değerinin 1 olması ele alınan geometrik şeklin kare yada dikdörtgen olduğunu gösterir. Eşit alanlı iki şeklin karmaşıklık değerleri farklı ise, karmaşıklık değeri daha büyük olan şekil yerleştirildiğinde daha fazla malzemenin ziyanına neden olur.

6.2 KULLANILAN ALGORİTMA

Bu çalışmada oluşturulan bilgisayar programı Şekil 6.15 de verilen algoritmaya göre geliştirilmiştir. Çalışma iki bilgisayar programından oluşmaktadır.

a.) DXF Çevrim Programı:

Bu program ile CAD sisteminde 6.1.1 bölümünde açıklanan esaslar dahilinde çizilmiş olan parça resimlerine ait, DXF bilgi formatında hazırlan geometrik bilgiler okunur ve ekran üzerine çizilir. Ekran üzerine çizilen geometri üzerinde, herhangi bir nokta referans noktası seçilir. Bu noktadan başlayarak saat yelkovanının dönüş yönünün tersi yönde, geometrinin etrafını dolaşacak şekilde parçanın köşe koordinatları kullanılarak bir polyline tanımlanır. Tanımlanan polyline sayesinde geometrinin alanı hesaplanabilmektedir. Daha sonra şeklin içine oturacağı dikdörtgen zarfı otomatik olarak hesaplanmaktadır. Bu iki değer ile geometrik şekil için karmaşıklık değeri hesaplanmaktadır. DXF çevrim programı, yerleştirme işlemini için, kullanılacak olan bilgileri otomatik olarak hazırlamakta ve @ABCDEFG.DAT dosyasına yazmaktadır.



Şekil -6.15- Kullanılan Algoritma

b) Yerleřtirme Programı:

Program, farklı boyut ve miktarlara sahip dikdörtgen parçaların, doğrusal programlama yöntemi kullanarak, ana paftalar üzerine minimum kayıp ve kullanılan toplam ana pafta miktarını en küçükleyecek şekilde yerleřtirilmesini sağlamaktadır. Program řu şekilde çalışmaktadır:

1) DXF çevrim programı tarafından hesaplanarak @ABCDEFG.DAT dosyasına yazılan, Ana paftaya ait en, boy bilgisi, Parça çeřit sayısı, herbir parça veya grubu temsil eden dikdörtgen zarflara ait en boy bilgileri ve parça talep miktarları okunur.

2) Dikdörtgen zarflar, genişlikleri esas alınarak büyükten küçüğe doğru sıralanır.

3) Ana pafta üzerine, tanımlanan dikdörtgenlerin enine yerleřimleri hesaplanır. Burada asıl prensip ilk enine yerleřimin hesaplanmasında en büyük genişliğe sahip dikdörtgenin max. miktarda kullanılması, kalan yerlerin ise mümkünse 2. en büyük genişliğe sahip dikdörtgen ile veya 3. en büyük genişliğe sahip dikdörtgen ile veya n. en büyük genişliğe sahip dikdörtgen ile doldurulmasıdır. 2., 3., ...,n. enine yerleřim hesaplanırken en küçük genişlik değerine sahip olan dikdörtgen dışında, dikdörtgenlerin genişlikleri büyükten küçüğe doğru azalan sırada, her seferinde kullanılan maximum genişlikten bir sonraki genişlik başlangıç değeri olarak kullanılacak şekilde ve geriye kalan ana pafta genişliği ardışık diğer daha küçük değerler ile doldurulur. Herbir enine yerleřimin meydana getirdiği kayıp miktarı hesaplanır.

4) Hesaplanan enine yerleřimler kayıp miktarına göre büyükten küçüğe doğru sıralanır.

5) Her bir enine yerleřim için boyuna yerleřimler hesaplanır. Önce enine yerleřimde belirlenen adette dikdörtgen ana pafta üzerinde birer defa yerleřtirilir.

Kullanılan ile parça alanlarında kayıp alan hesaplanır. Daha sonra en kısa uzunluktaki sıraya, o sıradaki parçalardan birer adet ilave edilir. Aynı uzunlukta birden fazla sıra var ise, boyu daha kısa olan parça sırası öncelik alır. Kayıp alan hesaplanır. Bu işlem ana pafta sonuna gelene kadar devam eder. Kayıp miktarının en küçük olduğu seçenek, üzerinde çalışılan enine yerleşim için hesaplanan boyuna yerleşim olarak kabul edilir.

6) Matematiksel model kurulur. Kurulacak matematiksel model için amaç fonksiyonu, hesaplanana enine ve boyuna yerleşimleri içeren levha sayısı toplamını minimum yapmaktır. Kısıtlar ise; kullanılacak levha adedinin sıfırdan büyük veya sıfır olmasıdır. Ayrıca hesaplanacak levha sayısı değeri tam sayıya yuvarlatılacaktır. Diğer bir kısıt ise, herhangi bir parçanın, hesaplanan yerleşim planlarındaki sayılarının toplamı o parçanın talep miktarına eşit veya büyük olmasıdır.

7) Elde edilen çözümler için DXF çıktıları hazırlanır.

8) Herhangi bir CAD programına aktarılan DXF dosya, bu program tarafından okunur ve neticeler ektana çizilir veya plotter ile kağıda alınır.

7. SONUÇ

Parça yerleştirme problemlerinde, genellikle probleme özgü sezgisel yaklaşımlar kullanılmaktadır. Yapılan çalışmalardan, son yıllarda etkileşimli sistemlerin de kullanım alanına girmeye başladığı görülmektedir. Sezgisel yöntemlerin ağırlık kazanması, kullanım kolaylığının yanı sıra çözüm esnasında istenen her türlü kısıtın probleme entegre edilebilmeside rol oynamaktadır. İki boyutlu çözüm yaklaşımları genel olarak ele alındığında şu sonuçlara ulaşılır:

- 1 Çözüm neticesinde elde edilen yerleştirme planlarında kayıp oranı belirli bir değerin altında olanların çözüm olarak kabul edildiği belirtilmektedir. Fakat kayıp oranlarının nasıl belirlendiği konusunda bir kural veya kıstas ortaya konmamaktadır.
- 2 Yerleştirme planlarının tüm aşamalarından ayrıntılı olarak söz edilmemektedir.
- 3 Her bir parçanın, türetilen yerleştirme planları içindeki miktarına kısıt konmaktadır ve bunun nedenide açıklanmamaktadır..
- 4 Son anda meydana çıkacak olan yeni taleplerin karşılanma yöntemleri ,yarı otomatik yerleştirme proseslerinde yer almaya başlamıştır.

Düzgün olmayan parçaların yerleştirilmesi ile ilgili çalışmalar ele alındığında, oluşturulmak istenen uygulamaların genelde grafik tabanlı çalışmalar olduğu görülmektedir. Bazı çalışmalarda, geliştirilen bilgisayar programları içinde CAD özelliklerinin (Parçayı çizmek taşımak, döndürmekö ölçeklemek v.b işlemler) mevcut olduğu hatta bu amaçla ayrı bir dil geliştirildiği görülmüştür.Örneğin:

Antonio ALBANO'nun COMPUTER AIDED DESIGN V-9 1977 dergisinde yayınlanan A METHOD TO IMPROVE TWO DIMENSIONAL LAYOUT adlı makalesinde CALG isimli bir sistemden ve bu sistemin komut dilinden bahsedilmektedir. Mevcut çalışmamızın grafik özelliklerinin geliştirilmesi gerekmektedir.

Herbir enine yerleşim için işlem sayısı max. 400 olarak sınırlandırılmıştır. Ayrıca herbir enine yerleşim için boyuna yerleşim miktarıda maximum 400 adımla sınırlandırılmıştır. Yöntemin geliştirilmesi ile bu sınırlar iyileştirilebilir.



KAYNAKLAR

1. **Michael ADAMOWICZ - Antonio ALBANO**
A Solution Of The Rectangular Cutting Stock Problem
IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS Vol:SMC-6 NO:4 ... 1976
2. **Antonio ALBANO - Michael ADAMOWICZ**
Nesting Two-Dimensional Shapes In Rectangular Modules
COMPUTER AIDED DESIGN Volume 8 Number 1
Sayfa : 27 - 33 1976
3. **Antonio ALBANO**
A Method To Improve Two-Dimensional Layout
COMPUTER AIDED DESIGN Volume 9 Number 1
Sayfa : 48 - 52 1977
4. **Bengt-Erik BENGTTSSON**
Packing Rectangular Pieces - A Heuristic Approach
THE COMPUTER JOURNAL VOLUME 25 NO:3
Sayfa : 353 - 357 1982
5. **Nicos CHRISTOFIDES - Charles WHITLOC ,**
An Algorithm For Two-Dimensional Cutting Problems OPERATIONS RESEARCH V-25
Sayfa : 30 - 44 1977
6. **Cihan H.DAĞLI - M.Yalçın TATOĞLU**
An Approach To Two-Dimensional Cutting Stock Problems
PRODUCTION RESEARCH V-25, NO:2,
Sayfa : 175 - 190 1987

7. **Robert D.DIETRICH - Sidney YAKOWITZ**
A Rule-Based Approach To The Trim-Loss Problem
INTERNATIONAL JOURNAL OF PRODUCTION
RESEARCH V-29, NO:2,
Sayfa : 401 - 415 1991
8. **A.Cemil DİKİLİ**
Gemi Üretiminde Bilgisayar Destekli Optimum
Malzeme Kullanımı
İSTANBUL TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ
ENSTİTÜSÜ
DOKTORA TEZİ Temmuz,1991
9. **Harald DYCKHOFF**
A New Linear Programming Approach To The
Cutting Stock Problem
OPERATIONS RESEARCH V-29,
Sayfa : 1092 - 1103 1981
10. **Dr. Doğan EROL**
Giyotin İle Kesmede İki Boyutlu Dilme
Probleminin Çözümü İçin Yordamsal Bir Yaklaşım
T.C. ANADOLU ÜNİVERSİTESİ YAYINLARI NO:375 -
Mühendislik Mimarlık Fakültesi Yayınları No:52
1990
11. **J.Sociro FERREIRA - M.Antonio NEVES -
P.Fonseca e CASTRO**
A Two-Phase Roll Cutting Problem
EUROPEAN JOURNAL OF OPERATIONAL
RESEARCH 44
Sayfa : 185 - 196 1990

12. **Constantine GOULIMIS**

Optimal Solutions For The Cutting Stock Problem
EUROPEN JOURNAL OF OPERATIONAL RESEARCH
44,
Sayfa : 197 - 208 1990

13. **Zuhal KUKURT**

STOK KESME PROBLEMİNİN ÇÖZÜMÜ İÇİN
BULUŞSAL BİR YÖNTEM
İ.T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ - Sistem Analizi
Programı
Yüksek Lisans Tezi.. ..1985

14. **A.Y.C. NEE - K.W. SEOW - S.L.LONG**

Designing Algorithm For Nesting Irregular Shapes
With and Without Boundary Constraints
ANNALS OF THE CIRP Vol. 35/1 1986

15. **R.ORSINI - Antonio ALBANO**

A Heuristic Solution Of The Rectangular Cutting
Stock Problem
THE COMPUTER JOURNAL VOLUME 23 NUMBER 4
Sayfa : 338 - 343 1980

16. **S.A.ROBERTS**

Application Of Heuristic Techniques To The
Cutting Stock Problem For Worktops
JOURNAL OF OPERATIONAL RESEARCH Vol.35,
NO:5
Sayfa : 369 - 377 1984

17. **Giuseppe SAPUPPO - Antonio ALBANO**

Optimal Allocation Of Two-Dimensional Irregular
Shapes Using Heuristic Search Methods
IEEE TRANSACTIONS ON SYSTEMS, MAN AND
CYBERNETICS Vol:SMC-10 NO:5 1980

18. *Gerhard WÄSCHER*

An LP-Based Approach To Cutting Stock Problems
With Multiple Objectives

EUROPEN JOURNAL OF OPERATIONAL RESEARCH
44,

Sayfa : 175 - 184

1990



```
*****
** Bu program ile OPTİMİZASYON programı için , 2 Boyutlu parçaların DXF *
** Formatı ile hazırlanmış olan bilgileri incelenecek ve hazırlanacaktır *
*****
```

```
DECLARE SUB GrafikGosterim (OptimizasyonDosyasi$, AnaDosya$, ParcaKoor$)
DECLARE SUB DXFConverter (IslemGorenDosya$)
DECLARE SUB Center (row!, text$)
DECLARE SUB GrafikEkranDuzenleme ()
DECLARE FUNCTION EvetHayirKontrol! ()
DECLARE FUNCTION GetArraySize (Wleft, Wright, WTop, WBottom)
```

```
CONST Siyah = 0, Mavi = 1, Yesil = 2, Turkuaz = 3, Kirmizi = 4, KoyuMor = 5
CONST Kahverengi = 6, Beyaz = 7, Gri = 8, ParlakMavi = 9, ParlakYesil = 10
CONST GokMavisi = 11, ParlakKirmizi = 12, Mor = 13, Sari = 14, ParlakBeyaz = 15
```

```
ErrorKod = 0
```

```
CALL GrafikEkranDuzenleme
```

```
VIEW PRINT 58 TO 60
```

```
COLOR Sari
```

```
LOCATE 59, 1: INPUT "Kullanılacak Ana Paftanın Boyunu Girin : ", AnaPaftaBoy
CLS 2
LOCATE 59, 1: INPUT "Kullanılacak Ana Paftanın Enini Girin : ", AnaPaftaEn
CLS 2
LOCATE 59, 1: INPUT "Parca Cesit Sayisini Girin : ", ParcaCesitSayisi
CLS 2
```

```
VIEW PRINT
```

```
OPEN "c:\@abcdefg.dat" FOR OUTPUT AS #1
PRINT #1, AnaPaftaBoy, ";", AnaPaftaEn
PRINT #1, ParcaCesitSayisi
CLOSE #1
```

```
FOR Cesit = 1 TO ParcaCesitSayisi
```

```
REM ***** DXF DOSYAYA AIT BILGILER *****
```

```
30 VIEW PRINT 58 TO 60
```

```
CLS 2
COLOR Sari
DO
LOCATE 58, 1: INPUT "DXF Dosya Ismi : ", DXFDosyaIsmi$
LOOP WHILE DXFDosyaIsmi$ = ""
```

```
70 REM bu rutinde girilen dosya adinin eklentisi kontrol edilir
REM herhangi bir eklenti verilmemisse dosya adi tamamlanir
```

```
DXFDosyaIsmi$ = UCASE$(DXFDosyaIsmi$)
```

```
IF INSTR(DXFDosyaIsmi$, ".DXF") = 0 THEN DXFDosyaIsmi$ = DXFDosyaIsmi$ + ".DXF"
```

```
ON ERROR GOTO HataTespiti
```

```
OPEN DXFDosyaIsmi$ FOR INPUT AS #1
CLOSE #1
```

```
IF ErrorKod <> 0 THEN 30 ELSE 40
```

- ' Yukarıdaki satırda tüm hatalı bilgi girişleri için uyarı elde edilir
- ' Eger ErrorKod=0 ise işlem normal olarak yürür
- ' ErrorKod<>0 ise sistem girişin yenilenmesini ister

40 VIEW PRINT

```
ToplamKarakterSayisi = LEN(DXFDosyalsmi$)
FOR n = ToplamKarakterSayisi TO 1 STEP -1
    karakter$ = MID$(DXFDosyalsmi$, n, 1)
    IF karakter$ = "." THEN
        DosyaUzantisi = n
    ELSEIF karakter$ = "\" THEN
        DosyaAdiBaslangici = n
        GOTO 10
    END IF
NEXT

10 DosyaAdiUzunlugu = ToplamKarakterSayisi - DosyaAdiBaslangici
   YolAdiUzunlugu = ToplamKarakterSayisi - DosyaAdiUzunlugu

DosyaAdi$ = RIGHT$(DXFDosyalsmi$, DosyaAdiUzunlugu)
YolAdi$ = MID$(DXFDosyalsmi$, 1, YolAdiUzunlugu)
OptimizasyonDosyasi$ = UCASE$(LEFT$(DosyaAdi$, DosyaAdiUzunlugu - 4) + ".NST")
ParcaKoor$ = UCASE$(LEFT$(DosyaAdi$, DosyaAdiUzunlugu - 4) + ".CRD")
ParcaKoor$ = YolAdi$ + ParcaKoor$

VIEW PRINT 58 TO 60
CLS 2
COLOR ParlakKirmizi
Center 59, "DXF DOSYA YUKLENIYOR.... LUTFEN BEKLEYINIZ...."
COLOR Sari
VIEW PRINT

COLOR ParlakYesil
LOCATE 3, 1: PRINT "DXF Data Dosyasi : "; DosyaAdi$
LOCATE 3, 45: PRINT "OPTIMIZASYON Dosyasi : "; OptimizasyonDosyasi$

IslemGorenDosya$ = YolAdi$ + DosyaAdi$

DIM SHARED LineCizim(1, 1), ArcCizim(1, 5)

CALL DXFConverter(IslemGorenDosya$)

Komut1$ = "COPY C:\EMRE1.DAT " + YolAdi$ + OptimizasyonDosyasi$ + ">NUL"
Komut2$ = "DEL C:\EMRE1.DAT >NUL."

SHELL Komut1$: SHELL Komut2$

OptimizasyonDosyasi$ = YolAdi$ + OptimizasyonDosyasi$

CALL J. GrafikGosterim(OptimizasyonDosyasi$, AnaDosya$, ParcaKoor$)

NEXT Cesit

END
```

HataTespiti:

```
SELECT CASE ERR
CASE 58
    KILL OptimizasyonDosyasi$
```

```
CASE 53
  COLOR Mor
  Center 60, "YUKLEMEK ISTEDIGINIZ DXF DOSYA BULUNAMADI."
  DosyaIsmi$ = ""
  ErrorKod = 53
CASE 64
  COLOR Mor
  Center 60, "YUKLEMEK ISTEDIGINIZ DXF DOSYA ISMINDE BIR HATA VAR."
  DosyaIsmi$ = ""
  ErrorKod = 64
END SELECT

RESUME NEXT
```

```
'Center:
' Center text on the given row.
SUB Center (row, text$)
  LOCATE row, 41 - LEN(text$) / 2
  PRINT text$;
END SUB
```

```
SUB DXFConverter (IslemGorenDosya$)
```

```
  DXFKod = 0: ElemanSayisi = 0
```

```
  Gecici$ = "c:\emre.dat"
```

```
  OPEN Gecici$ FOR OUTPUT AS #2
```

```
  OPEN IslemGorenDosya$ FOR INPUT AS #1
```

```
OperasyonBaslangici:
```

```
INPUT #1, GrupKod%
INPUT #1, GrupDegeri$
```

```
  IF GrupKod% = 9 AND GrupDegeri$ = "$EXTMIN" THEN
    GOTO CizimLimitleriMin
  ELSEIF GrupKod% = 9 AND GrupDegeri$ = "$EXTMAX" THEN
    GOTO CizimLimitleriMax
  ELSEIF GrupKod% = 0 AND GrupDegeri$ = "EOF" THEN
    GOTO ProgramSonu
  ELSEIF DXFKod = 1 THEN
    GOTO LineBolumu
  ELSEIF GrupKod% = 0 AND GrupDegeri$ = "LINE" THEN
    DXFKod = 1
    Eleman$ = "LINE"
  ELSEIF DXFKod = 2 THEN
    GOTO ArcBolumu
  ELSEIF GrupKod% = 0 AND GrupDegeri$ = "ARC" THEN
    DXFKod = 2
    Eleman$ = "ARC"
  END IF
```

```
GOTO OperasyonBaslangici
```

```
CizimLimitleriMin:
```

```
INPUT #1, GrupKod%
```

```
IF GrupKod% = 10 THEN
  INPUT #1, SolAltX
```

```
ELSEIF GrupKod% = 20 THEN
    INPUT #1, SolAltY
    GOTO OperasyonBaslangici
END IF
```

GOTO CizimLimitleriMin

CizimLimitleriMax:

```
INPUT #1, GrupKod%

IF GrupKod% = 10 THEN
    INPUT #1, SagUstX
ELSEIF GrupKod% = 20 THEN
    INPUT #1, SagUstY

    DeltaX = ABS(SolAltX)
    DeltaY = ABS(SolAltY)

    SolAltX = 0
    SolAltY = 0
    SagUstX = SagUstX + DeltaX
    SagUstY = SagUstY + DeltaY

    PRINT #2, "CIZIM LIMITLERI"
    PRINT #2, SolAltX
    PRINT #2, SolAltY
    PRINT #2, SagUstX
    PRINT #2, SagUstY

    GOTO OperasyonBaslangici
END IF
```

GOTO CizimLimitleriMax

LineBolumu:

```
IF GrupKod% = 10 THEN
    INPUT #1, LineXBKoordinat
    LineCizim(1, 1) = LineXBKoordinat + DeltaX
ELSEIF GrupKod% = 20 THEN
    INPUT #1, LineYBKoordinat
    LineCizim(1, 2) = LineYBKoordinat + DeltaY
ELSEIF GrupKod% = 11 THEN
    INPUT #1, LineXSKoordinat
    LineCizim(1, 3) = LineXSKoordinat + DeltaX
ELSEIF GrupKod% = 21 THEN
    INPUT #1, LineYSKoordinat
    LineCizim(1, 4) = LineYSKoordinat + DeltaY
    GOTO OptimizasyonDataDosyasiHazirlama
END IF
```

INPUT #1, GrupKod%

GOTO LineBolumu

ArcBolumu:

```
IF GrupKod% = 10 THEN
    INPUT #1, ArcCenterX
    ArcCizim(1, 1) = ArcCenterX + DeltaX
ELSEIF GrupKod% = 20 THEN
```

```
        INPUT #1, ArcCenterY
        ArcCizim(1, 2) = ArcCenterY + Delta Y
ELSEIF GrupKod% = 40 THEN
        INPUT #1, Radius
        ArcCizim(1, 3) = Radius
ELSEIF GrupKod% = 50 THEN
        INPUT #1, StartAngle
        ArcCizim(1, 4) = StartAngle
ELSEIF GrupKod% = 51 THEN
        INPUT #1, EndAngle
        ArcCizim(1, 5) = EndAngle
        GOTO OptimizasyonDataDosyasiHazirlama
END IF
```

```
INPUT #1, GrupKod%
```

```
GOTO ArcBolumu
```

```
OptimizasyonDataDosyasiHazirlama:
```

```
PRINT #2, Eleman$
```

```
ElemanSayisi = ElemanSayisi + 1
```

```
IF Eleman$ = "LINE" THEN
```

```
    FOR Sayici = 1 TO 4
```

```
        PRINT #2, LineCizim(1, Sayici)
```

```
    NEXT
```

```
ELSEIF Eleman$ = "ARC" THEN
```

```
    FOR Sayici = 1 TO 5
```

```
        PRINT #2, ArcCizim(1, Sayici)
```

```
    NEXT
```

```
END IF
```

```
Eleman$ = " ": DXFKod = 0
```

```
GOTO OperasyonBaslangici
```

```
ProgramSonu:
```

```
CLOSE #1, #2
```

```
OPEN "c:\emre1.dat" FOR OUTPUT AS #3
```

```
PRINT #3, "ELEMEN SAYISI"
```

```
PRINT #3, ElemanSayisi
```

```
OPEN Gecici$ FOR INPUT AS #4
```

```
DO WHILE NOT EOF(4)
```

```
    LINE INPUT #4, Satir$
```

```
    PRINT #3, Satir$
```

```
LOOP
```

```
CLOSE #4
```

```
CLOSE #3
```

```
KILL Gecici$
```

END SUB

FUNCTION EvetHayirKontrol

```
cevap$ = ""
KarakterPoz% = 0
DO
    cevap$ = UCASE$(INKEY$)
    IF cevap$ <> "" THEN
        KarakterPoz% = INSTR("EHS", cevap$)
        IF KarakterPoz% = 0 THEN BEEP
    END IF
LOOP UNTIL KarakterPoz% > 0

EvetHayirKontrol = KarakterPoz%
```

END FUNCTION

FUNCTION GetArraySize (Wleft, Wright, WTop, WBottom) STATIC

```
Vleft = PMAP(Wleft, 0)
VRight = PMAP(Wright, 0)
Vtop = PMAP(WTop, 1)
VBottom = PMAP(WBottom, 1)

RectHeight = ABS(VBottom - Vtop) + 1
RectWidth = ABS(VRight - Vleft) + 1

BitPerPixel = 4

ByteSize = 4 + RectHeight * INT((RectWidth * BitPerPixel + 7) / 8)

GetArraySize = ByteSize \ 2 + 1

END FUNCTION
```

SUB GrafikEkranDuzenleme

```
SCREEN 12
WIDTH 80, 60
```

'Grafik Ekranin Basligi

```
COLOR ParlakKirmizi: PRINT STRING$(80, CHR$(178))
COLOR Sari: Center 1, " OPTIMIZASYON PROGRAMI - BILGI GHRISI "
```

'Grafik Ekranin Sinirlari

```
LINE (0, 10)-(640, 10), GokMavisi 'Ust Duz Cizgi
LINE (0, 452)-(640, 452), GokMavisi 'Alt Duz Cizgi
LINE (0, 28)-(416, 444), GokMavisi, B 'Grafik Alan Cercevesi
LINE (420, 28)-(639, 444), GokMavisi, B 'Text Alan Cercevesi
```

```
VIEW (3, 31)-(413, 441) 'Grafik ekran tanimlaniyor
WINDOW (0, 0)-(410, 410) 'Grafik Ekranin (0,0) noktasi grafik pencerenin
                        'sol alt kosesine alindi.
```

END SUB

SUB GrafikGosterim (OptimizasyonDosyasi\$, AnaDosya\$, ParcaKoor\$)

ON ERROR GOTO 0

pi = 4 * ATN(1)

ERASE LineCizim, ArcCizim

VIEW PRINT 58 TO 60

CLS 1

CLS 2

COLOR ParlakKirmizi

Center 59, "GEOMETRI CIZILIYOR.... LUTFEN BEKLEYINIZ...."

COLOR Sari

VIEW PRINT

SatirSayisi = -2: Kayit = 0

OPEN "c:\Parca.Krd" FOR OUTPUT AS #1

CLOSE #1

OPEN OptimizasyonDosyasi\$ FOR INPUT AS #1

DO WHILE NOT EOF(1)

INPUT #1, aciklama\$

SatirSayisi = SatirSayisi + 1

IF aciklama\$ = "ELEMEN SAYISI" THEN

INPUT #1, GeoElmSayisi

DIM GeoElm(GeoElmSayisi, 10)

ELSEIF aciklama\$ = "CIZIM LIMI'LERI" THEN

INPUT #1, SolAltX

INPUT #1, SolAltY

INPUT #1, SagUstX

INPUT #1, SagUstY

ELSEIF aciklama\$ = "LINE" THEN

GeoElm(SatirSayisi, 1) = 1

FOR Sayici = 2 TO 5

INPUT #1, GeoElm(SatirSayisi, Sayici)

NEXT

ELSEIF aciklama\$ = "ARC" THEN

GeoElm(SatirSayisi, 1) = 2

FOR Sayici = 6 TO 10

INPUT #1, GeoElm(SatirSayisi, Sayici)

NEXT

xm = GeoElm(SatirSayisi, 6)

ym = GeoElm(SatirSayisi, 7)

r = GeoElm(SatirSayisi, 8)

SA = (GeoElm(SatirSayisi, 9) * pi) / 180

FA = (GeoElm(SatirSayisi, 10) * pi) / 180

GeoElm(SatirSayisi, 2) = xm + r * COS(SA)

GeoElm(SatirSayisi, 3) = ym + r * SIN(SA)

GeoElm(SatirSayisi, 4) = xm + r * COS(FA)

GeoElm(SatirSayisi, 5) = ym + r * SIN(FA)

```
END IF
LOOP

CLOSE #1

REM ***** GEOMETRIK SEKLIN KOSE KOORDINATLARININ SAAT YONUNDE
SIRALANMASI *****

VIEW PRINT 58 TO 60

CLS 2

PRINT

INPUT "Geometrisi Okunan Parcanin Talep Mikatarini Girin : ", ParcaTalepMiktari

CLS 2

VIEW PRINT

LOCATE 7, 54: PRINT "Eleman Sayisi: "; GeoElmSayisi
LOCATE 9, 54: PRINT "Talep Sayisi: "; ParcaTalepMiktari
LOCATE 12, 54: PRINT "Alan ( mm²) :";

IF SagUstX > SagUstY THEN
    MaxOlcu = SagUstX
ELSEIF SagUstX < SagUstY THEN
    MaxOlcu = SagUstY
ELSEIF SagUstX = SagUstY THEN
    MaxOlcu = SagUstX
END IF

Olcek = (MaxOlcu / 350)

WINDOW (0, 0)-(SagUstX * 410) / MaxOlcu, (SagUstY * 410) / MaxOlcu

REM LINE (0, 0)-(SagUstX * 410) / MaxOlcu, (SagUstY * 410) / MaxOlcu, Sari, B

FOR Sayici = 1 TO GeoElmSayisi

    IF GeoElm(Sayici, 1) = 1 THEN
        xlb = GeoElm(Sayici, 2) / Olcek
        xlb = xlb + 20
        ylb = GeoElm(Sayici, 3) / Olcek
        ylb = ylb + 20
        xls = GeoElm(Sayici, 4) / Olcek
        xls = xls + 20
        yls = GeoElm(Sayici, 5) / Olcek
        yls = yls + 20

        LINE (xlb, ylb)-(xls, yls), ParlakBeyaz

    ELSEIF GeoElm(Sayici, 1) = 2 THEN
        xm = GeoElm(Sayici, 6) / Olcek
        xm = xm + 20
        ym = GeoElm(Sayici, 7) / Olcek
        ym = ym + 20
        r = GeoElm(Sayici, 8) / Olcek
        SA = (GeoElm(Sayici, 9) * pi / 180)
        FA = (GeoElm(Sayici, 10) * pi / 180)

        IF SA > FA THEN
```

```
FOR t= SA TO 2 * pi STEP .005
PSET (xm + r * COS(t), ym + r * SIN(t)), ParlakBeyaz
NEXT
FOR t= 0 TO FA STEP .005
PSET (xm + r * COS(t), ym + r * SIN(t)), ParlakBeyaz
NEXT
ELSE
FOR t= SA TO FA STEP .005
PSET (xm + r * COS(t), ym + r * SIN(t)), ParlakBeyaz
NEXT
END IF
```

```
END IF
NEXT Sayici
```

```
VIEW PRINT 58 TO 60
```

```
COLOR ParlakKirmizi
Center 58, "Geometrinin Koordinatlari Saat Ibresi Donus Yonunde Siralanmalidir."
COLOR Sari
Center 59, "Geometrinin Baslangic Noktasi Kursorun Oldugu Noktadir ?"
```

```
VIEW PRINT
```

```
BasKoor = 1: ElemanSayisi = 1: CCW = 1
```

```
KurKoorX = GeoElm(ElemanSayisi, 2) / Olcek
KurKoorX = KurKoorX + 20
KurKoorY = GeoElm(ElemanSayisi, 3) / Olcek
KurKoorY = KurKoorY + 20
```

```
$DYNAMIC
```

```
DIM Array(1 TO 1) AS INTEGER
```

```
500 GOSUB KursorPozisyon
```

```
LINE (KurKoorX - 10, KurKoorY)-(KurKoorX + 10, KurKoorY), ParlakYesil
LINE (KurKoorX, KurKoorY + 10)-(KurKoorX, KurKoorY - 10), ParlakYesil
```

```
LOCATE 16, 64: PRINT "X : "; (KurKoorX - 20) * Olcek
LOCATE 18, 64: PRINT "Y : "; (KurKoorY - 20) * Olcek
```

```
LOCATE 20, 54: PRINT "Baslangic Koor. :"; BasKoor
```

```
VIEW PRINT 58 TO 60
```

```
Center 60, "[E]vet / [H]ayir / [S]on"
```

```
SELECT CASE EvetHayirKontrol
CASE 1
GOSUB Kaydet
CASE 2
GOSUB BirSonraki
CASE 3
GOTO 700
END SELECT
```

```
VIEW PRINT
```

```
GOTO 500
```

```
Kaydet:
```

```
OPEN "c:\Parca.Krd" FOR APPEND AS #1
PRINT #1, (KurKoorX - 20) * Olcek, ";", (KurKoorY - 20) * Olcek
CLOSE #1
```

```
PUT (KurKoorX - 15, KurKoorY - 15), Array, PSET
```

```
VIEW PRINT 58 TO 60
LOCATE 59, 1: PRINT STRING$(80, " ")
Center 59, "Bir Sonraki Noktayı Secin"
VIEW PRINT
```

```
IF Kayit = 1 THEN
    LINE (ilkX, ilkY)-(KurKoorX, KurKoorY), Sari
END IF
```

```
ilkX = KurKoorX: ilkY = KurKoorY
```

```
Kayit = 1
```

RETURN

BirSonraki:

```
IF BasKoor = 1 THEN
```

```
    PUT (KurKoorX - 15, KurKoorY - 15), Array, PSET
```

```
    KurKoorX = GeoElm(ElemanSayisi, 4) / Olcek
    KurKoorX = KurKoorX + 20
    KurKoorY = GeoElm(ElemanSayisi, 5) / Olcek
    KurKoorY = KurKoorY + 20
```

```
    BasKoor = 0
```

```
ELSEIF BasKoor = 0 THEN
```

```
    PUT (KurKoorX - 15, KurKoorY - 15), Array, PSET
```

```
    ElemanSayisi = ElemanSayisi + 1
```

```
    IF ElemanSayisi = GeoElmSayisi THEN ElemanSayisi = 1
```

```
    KurKoorX = GeoElm(ElemanSayisi, 2) / Olcek
    KurKoorX = KurKoorX + 20
    KurKoorY = GeoElm(ElemanSayisi, 3) / Olcek
    KurKoorY = KurKoorY + 20
```

```
    BasKoor = 1
```

```
END IF
```

RETURN

KursorPozisyon:

```
Wleft = KurKoorX - 15
Wright = KurKoorX + 15
WTop = KurKoorY + 15
WBottom = KurKoorY - 15
```

```
ArraySize% = GetArraySize(Wleft, Wright, WTop, WBottom)
• REDIM Array(1 TO ArraySize%) AS INTEGER
  GET (Wleft, WTop)-(Wright, WBottom), Array
RETURN

700 VIEW PRINT

Komut1$ = "COPY C:\PARCA.KRD " + ParcaKoor$ + " >NUL"
Komut2$ = "DEL C:\parca.krd >NUL"

SHELL Komut1$: SHELL Komut2$

OPEN OptimizasyonDosyasi$ FOR INPUT AS #1

DO WHILE NOT EOF(1)
  INPUT #1, aciklama$
  IF aciklama$ = "ELEMEN SAYISI" THEN
    INPUT #1, n
    DIM X(n + 1), Y(n + 1), Teta(n + 1)
  END IF

LOOP

CLOSE #1

OPEN ParcaKoor$ FOR INPUT AS #1

FOR Sayici = 1 TO n + 1
  INPUT #1, X, Y
  X(Sayici) = X
  Y(Sayici) = Y
NEXT

CLOSE #1

terim1 = 0: terim2 = 0: terim3 = 0

terim1 = Y(1) * (X(n) - X(2))

terim2 = Y(n) * (X(n - 1) - X(1))

FOR i = 2 TO (n - 1)
  terim = Y(i) * (X(i - 1) - X(i + 1))
  terim3 = terim + terim3
NEXT

Alan = (1 / 2) * (terim1 + terim2 + terim3)
```

LOCATE 12, 67: PRINT USING "#####,.###"; Alan

REM ***** Kenarların X(+) kenarı ile yaptığı acıların hesaplanması *****

```
j = 1

351 IF X(j) = X(j + 1) THEN

    IF Y(j + 1) > Y(j) THEN Teta(j) = 90
    IF Y(j + 1) < Y(j) THEN Teta(j) = 270

ELSEIF Y(j) = Y(j + 1) THEN

    IF X(j + 1) > X(j) THEN Teta(j) = 0
    IF X(j + 1) < X(j) THEN Teta(j) = 180

ELSE

    A = (Y(j + 1) - Y(j)) / (X(j + 1) - X(j))

    IF A > 0 THEN
        IF Y(j + 1) > Y(j) THEN Teta(j) = ATN(A): Teta(j) = Teta(j) * 57.2958
        IF Y(j + 1) < Y(j) THEN Teta(j) = ATN(A): Teta(j) = Teta(j) * 57.2958 + 180
    ELSEIF A < 0 THEN
        IF Y(j + 1) > Y(j) THEN Teta(j) = ATN(A): Teta(j) = Teta(j) * 57.2958 + 180
        IF Y(j + 1) < Y(j) THEN Teta(j) = ATN(A): Teta(j) = Teta(j) * 57.2958 + 360
    END IF
END IF

IF j < n THEN
    j = j + 1: GOTO 351
END IF

Komut$ = "DEL " + ParcaKoor$ + ">NUL"

SHELL Komut$

OPEN ParcaKoor$ FOR OUTPUT AS #1

FOR Sayici = 1 TO n + 1

    PRINT #1, X(Sayici); ", "; Y(Sayici); ", "; Teta(Sayici)

NEXT

CLOSE #1

REM ***** En İyi Dikdortgenin Hesaplanması *****

RecXMax = 0: RecXMin = 0: RecYMax = 0: RecYMin = 0

FOR S1 = 1 TO n + 1
    FOR S2 = 1 TO n
        IF X(S1) >= X(S2) THEN 142

        SWAP X(S2), X(S1)
        SWAP Y(S2), Y(S1)
        SWAP Teta(S2), Teta(S1)

142 NEXT S2
    NEXT S1

RecXMax = X(n + 1): RecXMin = X(1)
```

```
FOR S1 = 1 TO n + 1
  FOR S2 = 1 TO n
    IF Y(S1) >= Y(S2) THEN 141

    SWAP X(S2), X(S1)
    SWAP Y(S2), Y(S1)
    SWAP Teta(S2), Teta(S1)

141  NEXT S2
    NEXT S1

RecYMax = Y(n + 1): RecYMin = Y(1)

LINE (RecXMin / Olcek + 20, RecYMin / Olcek + 20)-(RecXMax / Olcek + 20, RecYMax / Olcek +
20), ParlakKirmizi, B

ParcaBoy = RecXMax - RecXMin
ParcaEn = RecYMax - RecYMin

OPEN "c:\@abcdefg.dat" FOR APPEND AS #1
  PRINT #1, ParcaBoy; ";"; ParcaEn; ";"; ParcaTalepMiktari
CLOSE #1

END SUB
```



```
DECLARE SUB SONUCALT (k!, son!(), is1!, ps!, v!(), w!(), sifir!(), sonuc!())
DECLARE SUB MAXIMUM (D!(), q!, r!, m!)
DECLARE SUB ALTERNTF (A!(), kolon!, ikont!, alt1!(), alt2!(), n!)
DECLARE SUB MINIMUM (D!(), q!, r!, m!, kon!)
DECLARE SUB ELE (v!(), k!, ps!)
DECLARE SUB EnineYerles (l!(), ls!, ps!, v!(), k!, nl!())
DECLARE SUB EnineYerlesSiral (v!(), q!, r!)
DECLARE SUB BoyunaYerles (w!(), v!(), ps!, k1!, nl!(), pl!(), pb!(), ll!, lb!, A!())
DECLARE SUB DogrusalProgramlama (n!, m1!, A!(), c!(), b!(), baz!(), kon!(), cik!(), alt1!(), alt2!(),
son!(), is1!, say!)
DECLARE SUB EkranaCiz (ll!, lb!, ps!, pl!(), pb!(), nl!())
```

```
CLS : PRINT : PRINT
```

```
LOCATE 20, 4: PRINT "DATA FILE = "; : LINE INPUT data$
```

```
*****
```

```
OPEN "I", #2, data$
```

```
INPUT #2, BASLIK$
```

```
INPUT #2, ll, lb: ' ll: Plaka Boyu lb: Plaka Eni
```

```
INPUT #2, ps ' ps: Plaka Sayisi
```

```
*****
```

```
DIM pl(ps), pb(ps), ni(ps), v(ps + 2, 400), nl(ps), w(ps + 2, 400), c(420), b(ps)
```

```
DIM baz(ps), kon(ps), cik(420), D(ps), A(ps + 2, 420), alt1(400), alt2(420)
```

```
DIM son(400, 2, ps), sifir(400), sonuc(ps)
```

```
*****
```

```
' pl: Parca Boyu pb: Parca Eni ni: Parca Sayisi
```

```
*****
```

```
FOR i = 1 TO ps: INPUT #2, pl(i), pb(i), ni(i): NEXT i
```

```
FOR j = 1 TO ps - 1
```

```
FOR i = 1 TO ps - 1
```

```
IF pb(i) < pb(i + 1) THEN
```

```
SWAP pb(i), pb(i + 1): SWAP pl(i), pl(i + 1): SWAP ni(i), ni(i + 1)
```

```
END IF
```

```
NEXT i, j
```

```
*****
```

```
CALL EnineYerles(pb(), lb, ps, v(), k, nl())
```

```
IF k > 400 THEN PRINT : PRINT "K > 400 ..... ISLEM DURDURULDU.": STOP
```

```
CALL EnineYerlesSiral(v(), ps + 1, k)
```

```
CALLI. ELE(v(), k, ps)
```

```
LOCATE 21, 16: say = 2 * k - ps + 1: PRINT say
```

```
FOR k1 = 1 TO k
```

```
LOCATE 21, 16: say = say - 1: PRINT say
```

```
CALL BoyunaYerles(w(), v(), ps, k1, nl(), pl(), pb(), ll, lb, A())
```

```
NEXT k1
```

```
*****
```

```
FOR i = 1 TO k: c(i) = 1: NEXT i
```

```
FOR i = 1 TO ps: A(i, 0) = ni(i): NEXT i
```

```
FOR i = 1 TO ps + 2
```

```
FOR j = 1 TO k + 2 * ps
```

```
A(i, j) = v(i, j) * w(i, j)
```

```
NEXT j, i
```

```
*****
```

```
CALL DogrusalProgramlama(k, ps, A(), c(), b(), baz(), kon(), cik(), alt1(), alt2(), son(), is1, say)
```

```
CALLI. SONUCALT(k, son(), is1, ps, v(), w(), sifir(), sonuc())
```

```
CALL EkranaCiz(ll, lb, ps, pl(), pb(), nl())
```

```
END
```

```
SUB ALTERNTF (A(), kolon, ikont, alt1(), alt2(), n) STATIC
  FOR i = 1 TO n
    IF (alt1(i) = 0 AND alt2(i) = 0) THEN ikont = 0: kolon = i: GOTO 71
  NEXT i
  ikont = 1
71 REM
END SUB

SUB BoyunaVerles (w(), v(), ps, k1, nl(), pl(), pb(), ll, lb, A()) STATIC
  FOR il = 1 TO ps
    IF v(il, k1) > 0 THEN A(il, 1) = 1 ELSE A(il, 1) = 0
    nl(il) = A(il, 1) * pl(il): NEXT il
    CALL MAXIMUM(nl(), ps, r, m)
    lk = nl(m): kay = lb * lk
    FOR il = 1 TO ps: kay = kay - (v(il, k1) * A(il, 1) * pl(il) * pb(il)): NEXT il
    A(ps + 1, 1) = kay / (lb * lk) * 100
    count = 1
101 count = count + 1
    CALL MINIMUM(nl(), ps, r, m, 0)
    IF (A(m, count - 1) + 1) * pl(m) > ll THEN 102
    A(m, count) = A(m, count - 1) + 1
    FOR il = 1 TO ps
      IF il <> m THEN A(il, count) = A(il, count - 1)
    NEXT il: nl(m) = A(m, count) * pl(m)
    CALL MAXIMUM(nl(), ps, r, m)
    lk = nl(m): kay = lb * lk
    FOR il = 1 TO ps: kay = kay - (v(il, k1) * A(il, count) * pl(il) * pb(il)): NEXT il
    A(ps + 1, count) = kay / (lb * lk) * 100: GOTO 101
102 kay = 1: min = A(ps + 1, 1)
    FOR il = 2 TO count - 1
      IF min >= A(ps + 1, il) THEN min = A(ps + 1, il): kay = il
    NEXT il
    FOR il = 1 TO ps: w(il, k1) = A(il, kay): NEXT il
END SUB
```

```
SUB DogrusalProgramlama (n, m1, A(), c(), b(), baz(), kon(), cik(), alt1(), alt2(), son(), isl, say) STATIC
  ikont = 0: isl = 0
  w = 1000: m2 = 0: m3 = 0: np = 2 * m1 + m2 + m3 + n: m = m1 + m2 + m3
  IF m1 + m2 > 0 THEN mp = m + 2 ELSE mp = m + 1
  IF m1 = 0 THEN 1
  k = 0
  FOR i = n + 1 TO n + m1
    k = k + 1
    l = i + m1
    c(i) = 0
    c(l) = w
    A(k, i) = -1
    A(k, l) = 1
  NEXT i

1 IF m2 = 0 THEN 2
  k = m1
  FOR i = n + 2 * m1 + 1 TO np - m3
    k = k + 1
    c(i) = w
    A(k, i) = 1
  NEXT i

2 IF m3 = 0 THEN 3
  k = m1 + m2
  FOR i = np - m3 + 1 TO np
```

```
k = k + 1
c(i) = 0
A(k, i) = 1
NEXT i
```

```
3 k = 0
FOR i = np - m + 1 TO np
  k = k + 1
  b(k) = c(i)
  baz(k) = i
NEXT i
```

```
FOR i = 0 TO np: A(m + 1, i) = 0: A(m + 2, i) = 0
FOR j = 1 TO m
  IF b(j) <> w THEN A(m + 1, i) = A(m + 1, i) + b(j) * A(j, i)
  IF b(j) = w THEN A(m + 2, i) = A(m + 2, i) + A(j, i)
NEXT j
```

```
IF c(i) <> w THEN A(m + 1, i) = A(m + 1, i) - c(i)
IF c(i) = w THEN A(m + 2, i) = A(m + 2, i) - 1
NEXT i
```

**** islem ****

```
4 maxx = 0
FOR i = 1 TO np
  IF A(mp, i) > maxx AND cik(i) = 0 THEN maxx = A(mp, i): kolon = i
NEXT i
```

```
IF maxx = 0 AND mp = m + 2 AND A(m + 2, 0) > 0 THEN PRINT "COZUM YOK": STOP
IF maxx = 0 AND mp = m + 2 THEN mp = mp - 1: GOTO 4
IF maxx = 0 AND mp = m + 1 THEN GOTO 7
```

```
77 minn = 20000
FOR i = 1 TO m
  IF A(i, kolon) <> 0 THEN kon(i) = A(i, 0) / A(i, kolon)
  IF A(i, kolon) = 0 THEN kon(i) = 0
  IF kon(i) > 0 AND minn > kon(i) THEN minn = kon(i): satir = i
NEXT i
```

```
IF b(satir) = w THEN cik(baz(satir)) = 1
baz(satir) = kolon
b(satir) = c(kolon)
```

```
payda = A(satir, kolon)
IF payda = 0 THEN 7
FOR i = 0 TO np: A(satir, i) = A(satir, i) / payda: NEXT i
```

```
FOR i = 1 TO mp
  IF i = satir THEN 6
  IF A(i, kolon) = 0 THEN 6
  aa = A(i, kolon)
  FOR j = 0 TO np
    A(i, j) = A(i, j) - A(satir, j) * aa
    IF ABS(A(i, j)) < .00001 THEN A(i, j) = 0
  NEXT j
6 NEXT i
```

GOTO 4

```
7 is1 = is1 + 1
LOCATE 21, 16: say = say - 1: PRINT say
```

```
FOR i = 1 TO n: alt1(i) = A(m + 1, i): NEXT i
FOR i = 1 TO m: alt2(baz(i)) = 1: NEXT i
FOR i = 1 TO m: son(is1, 1, i) = baz(i): son(is1, 2, i) = A(i, 0): NEXT i
CALL ALTERNTE(A(), kolon, ikont, alt1(), alt2(), n)
IF ikont = 0 THEN GOTO 77
END SUB
```

```
SUB EkranaCiz (ll, lb, ps, pl(), pb(), nl()) STATIC
OPEN "I". #1, "c:sonuc1"
SCREEN 2, 0: CLS: WINDOW (0, 0)-(1000, 700): wx1 = 0: wx2 = 1000: WY1 = 0: wy2 = 700
IF (ll / lb) > (wx2 - wx1) / (wy2 - WY1) THEN olc = (wx2 - wx1) / ll ELSE olc = (wy2 - WY1) / lb
lx = olc * ll: ly = olc * lb
LEVHA = 0
WHILE NOT (EOF(1))
LEVHA = LEVHA + 1: x0 = wx1: y0 = WY1: CLS
LINE (wx1, WY1)-(lx, ly), , B
INPUT #1, SS: LOCATE 2, 2: PRINT "LEVHA NR = "; LEVHA
LOCATE 3, 2: PRINT "LEVHA ADEDI = "; SS
LOCATE 2, 40: PRINT "PARCA EN BOY ENINE BOYUNA"
LOCATE 3, 40: PRINT " N O (M) (M) ADET ADET"
nr = 0: nrf = 0: sat = 4
FOR i = 1 TO ps
nr = nr + 1
INPUT #1, ya, xa: IF ya = 0 OR xa = 0 THEN 117
sat = sat + 1
LOCATE sat, 40
PRINT USING "## ###.## ###.## #####": nr, pb(nr), pl(nr), ya, xa
IF nrf = 1 THEN pxx = pxx - px: pyy = pyy - py
py = pb(nr) + olc: px = pl(nr) * olc: IF nrf = 0 THEN pxx = px: pyy = py
IF nrf = 1 THEN pxx = x0: pyy = y0: py
FOR y = 1 TO ya
FOR x = 1 TO xa
LINE (x0, y0)-(pxx, pyy), , B
x0 = x0 + px
pxx = pxx + px
NEXT x
x0 = wx1
y0 = y0 + py
pxx = x0 + px
pyy = y0 + py
NEXT y
nrf = 1
117 NEXT i
118 aa$ = INKEY$
IF aa$ <> " " THEN GOTO 118
WEND
CLOSE #1
CLS
SCREEN 0, 0
END SUB
```

```
SUB ELE (v(), k, ps) STATIC
k2 = 0: FOR il = 1 TO k
IF v(ps + 1, il) <= 20 THEN k2 = k2 + 1
IF k2 = 20 THEN 100
NEXT il
100 k = k2
IF k > 400 THEN k = 400
END SUB
```

```
SUB EnineYeries (l(), ls, ps, v(), k, nl()) STATIC
i = 1: ll = ls: ka = 0
```

```
150 nl(i) = ls \ l(i): ls = ls - l(i) * nl(i): IF i = ps THEN 170
160 i = i + 1: IF ls >= l(i) THEN 150 ELSE IF i = ps THEN nl(i) = 0: GOTO 170 ELSE 160
170 k = k + 1: FOR ii = 1 TO ps: v(ii, k) = nl(ii): ka = ka + l(ii) * nl(ii)
NEXT ii: v(ps + 1, k) = CINT((ll - ka) / ll * 100): ka = 0: IF k = 400 THEN k = 401: GOTO 220
180 ls = ll: i = ps
190 i = i - 1: IF i = 0 THEN 220 ELSE IF nl(i) = 0 THEN 190
200 nl(i) = nl(i) - 1: FOR j = 1 TO i: ls = ls - nl(j) * l(j): NEXT j
210 i = i + 1: GOTO 150
220 END SUB
```

SUB EnineYerlesSirala (v(), q, r) STATIC

```
flag = r
WHILE flag > 0: f = flag - 1: flag = 0: FOR z = 1 TO f
IF v(q, z) > v(q, z + 1) THEN FOR y = 1 TO q: SWAP v(y, z), v(y, z + 1): NEXT y: flag = z
NEXT z: WEND
END SUB
```

SUB MAXIMUM (D(), q, r, m) STATIC

```
r = D(1): m = 1: FOR z = 2 TO q
IF r < D(z) THEN r = D(z): m = z
NEXT z
END SUB
```

SUB MINIMUM (D(), q, r, m, kon) STATIC

```
r = 1000: m = 0: FOR z = 1 TO q
IF r >= D(z) AND D(z) > kon THEN r = D(z): m = z
NEXT z
END SUB
```

SUB SONUCALT (k, son(), isl, ps, v(), w(), sifir(), sonuc()) STATIC

```
FOR i = 1 TO isl
FOR j = 1 TO ps
IF ABS(son(i, 2, j) * 10) > 30000 THEN son(i, 2, j) = -1
son(i, 2, j) = CINT(son(i, 2, j) * 10) / 10
NEXT j, i
FOR i = 1 TO isl
sif = 0
FOR j = 1 TO ps
IF (son(i, 2, j)) = 0 THEN sif = sif + 1
IF (son(i, 2, j)) < 0 THEN sifir(i) = 0: GOTO 90
NEXT j
sifir(i) = sif
90 NEXT i
MAX = -1
FOR i = 1 TO isl
IF MAX < sifir(i) THEN MAX = sifir(i): imax = i
NEXT i
FOR j = 1 TO ps: sonuc(j) = son(imax, 2, j): NEXT j
OPEN "o", #1, "c:sonuc1"
FOR i = 1 TO ps
IF sonuc(i) = 0 THEN 88
PRINT #1, sonuc(i)
FOR j = 1 TO ps
PRINT #1, v(j, son(imax, 1, i)); ", "; w(j, son(imax, 1, i))
NEXT j
88 NEXT i
CLOSE #1
END SUB
```

T.C. YÜREK
T.C. YÜREK

YÜREK
YÜREK

ÖZGEÇMİŞ

Yunus Emre TOKMAK 1967 yılında doğdu. İlk, Orta ve Lise tahsilini İSTANBUL'da tamamladı. 1984 yılında YILDIZ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ MAKİNA MÜHENDİSLİĞİ BÖLÜMÜ'ne girdi. Bu bölümdeki eğitimini 1989 senesinde tamamladı ve Mak. Müh. olarak mezun oldu. 1990 senesinde YILDIZ ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ MAKİNA MÜHENDİSLİĞİ ANA BİLİM DALI İMAL USLLERİ PROGRAMI'nda Yüksek Mühendislik eğitimine başladı.

Halen Makina Mühendisi olarak CAD/CAM/CNC konularında REİS CNC TAKIM TEZGAHLARI PROGRAMLAMA SİSTEMLERİ LTD. ŞTİ. bünyesinde görev yapmaktadır.