

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**GENETİK ALGORİTMA ve BENZETİLMİŞ TAVLAMA ile  
İKİ BOYUTLU GİYOTİNSİZ KESME PROBLEMLERİNE  
OLASILIKSAL YAKLAŞIM**

**YÜKSEK LİSANS TEZİ**

**Alev SÖKE**

**Tezin Enstitüye Verildiği Tarih : 23 Haziran 2003**

**Tezin Savunulduğu Tarih : 16 Temmuz 2003**

**TC. YÜKSEK ÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ**

**Tez Danışmanı**

**Yrd. Doç. Dr. Zafer BİNGÜL**

(.....*Zafer Bingül*.....)

**Üye**

**Prof. Dr. Hüseyin KARABELLİ**

(.....*Hüseyin Karabelli*.....)

**Üye**

**Doç. Dr. Kadir ERKAN**

(.....*Kadir Erkan*.....)

**Temmuz 2003**

# GENETİK ALGORİTMA ve BENZETİLMİŞ TAHLAMA ile İKİ BOYUTLU GİYOTİNSİZ KESME PROBLEMLERİNE OLASILIKSAL YAKLAŞIM

Alev SÖKE

**Anahtar Kelimeler:** Genetik Algoritma, Benzetilmiş Tavlama, Yerleştirme Algoritmaları, İki Boyutlu Kesme ve Paketleme Problemleri, Kıрма Yaklaşımlar.

**ÖZET:** Kesme problemleri tam bir matematiksel modelle ifade edilemezler. Bu yüzden, bu problemlerin çözümünü bulmak için çok boyutlu uzayda kombinasyonel eniyileme yapılır. Kesme problemlerinin amacı, yerleştirmenin yapılacağı ana parçanın kullanılabilirliğini arttırmak ve böylelikle en az fire değerine sahip kesme yerleşim planını elde etmektir.

Bu tez çalışmasında, Matlab ortamında iki boyutlu giyotinsiz kesme problemleri için bir çözüm yaklaşımı geliştirilmiştir. Bu problemlerin çözümü için ilk önce genetik algoritmalar (GA) ve geliştirilmiş aşağı sol (AS) algoritması, daha sonra benzetilmiş tavlama (BT) ve geliştirilmiş AS algoritması kullanıldı. Test problemleri 200x200 birimlik bir ana parça içine yerleştirilecek 17 ve 29 adet düzgün biçimli birbirinden farklı dikdörtgenlerden oluşan farklı parçalar içermektedir.

Çalışmanın birinci bölümünde, kesme problemlerinin çözümü için sıralamaya dayalı bir GA ve geliştirilmiş AS algoritması birleştirildi. Bu çözüm yaklaşımı kırma GA olarak bilinmektedir. İlk önce, kırma GA 17 parçalı test problemlerinin çözümü için kullanıldı. Bu problemlerin çözümü üzerinde farklı nüfus büyüklükleri ve mutasyon oranlarının etkileri incelenmiştir. Bu çalışmaya göre, nüfus büyüklüğü arttıkça fire değerinin azaldığı gözlemlenmiştir. Çözüm üzerinde mutasyon oranının çok küçük bir etkisi vardır. İkinci olarak kırma GA 29 parçalı bir test problemine uygulanmıştır. Bu kesme probleminin çözümü üzerinde farklı çaprazlama tekniklerinin etkileri incelenmiştir. Bu çalışma sonucunda en iyi sonucun sıralamaya dayalı çaprazlama tekniği ile elde edildiği görülmüştür. Çalışmanın ikinci bölümünde, BT ve geliştirilmiş aşağı sol (AS) algoritması aynı kesme problemlerinin çözümü için bir araya getirilmiştir. Bu çözüm yaklaşımı kırma BT olarak bilinmektedir. Kırma BT, ilk önce 17 parçalı test problemlerinin çözümü için uygulanmıştır. Bu kesme problemlerinin çözümü üzerinde farklı sıcaklık değerleri için, farklı soğutma planlarının, komşuluk hareketlerinin ve denge koşulu için farklı değerlerin etkileri incelenmiştir. İkinci olarak 29 parçalı bir test probleminin çözümü bir önceki çalışmada elde edilen en iyi sonuçların parametreleri kullanılarak çözülmüştür.

Bu benzetim çalışmaları sonucunda, kırma GA kullanılarak elde edilen 17 ve 29 parçalı test problemlerinin fire değerleri sırasıyla %2 ve %11 arasında, %5 ve %9 arasında değişmektedir. Benzer olarak kırma BT kullanılarak elde edilen 17 ve 29 parçalı test problemlerinin fire değerleri sırasıyla %4 ve %21 arasında, %10 ve %17 arasında değişmektedir. Sonuç olarak, kesme problemleri için kırma GA'nın kırma BT'den daha iyi olduğu görülmektedir.

**STOCHASTIC APPROACH FOR TWO DIMENSIONAL  
NON-GUILLOTINEABLE CUTTING PROBLEMS  
with GENETIC ALGORITHM and SIMULATED ANNEALING**

**Alev SÖKE**

**Keywords:** Genetic Algorithm, Simulated Annealing, Placement Algorithms, Two Dimensional Cutting and Packing Problems, Hybrid Approaches.

**ABSTRACT:** Cutting problems can not be defined explicitly with a mathematical model. The solution of these problems are found by using combinational optimization in multi dimensional space. The objective of the cutting problems is to increase the usability of main object and thus to obtain the cutting pattern that has minimum trim loss values.

In this thesis, A solution approach was developed for two dimensional non-guillotineable cutting problems in Matlab environment. First of all, genetic algorithms (GA) and improved bottom left (BL) algorithm were used to solve this problems and then simulated annealing (SA) and improved BL algorithm were used. Test problems consist of different pieces: 17 and 29 regular individual rectangles to place in main object with 200x200 unit.

First part of this work, an order-based GA is combined with improved BL algorithm to solve the cutting problems. This solution approach is known as hybrid GA. Firstly, hybrid GA was used to solve 17 pieces test problems. The influences of the different population sizes and mutation rates on the solution of these problems were examined. Base on this examination, it is observed that as the population size increases the trim loss value decreases. There is very small effect of mutation rate on the solution. Secondly hybrid GA was applied to 29 pieces a test problem. The influences of different crossover techniques on the solution of the cutting problem were studied. At the result of this study, it is seen that the best result is obtained with order based crossover technique. Second part of this work, SA and improved BL algorithm are combined to solve the same cutting problems. This solution approach is known as hybrid SA. Firstly, hybrid SA was applied to solve the 17 pieces test problems. The influences of different cooling schemes, neighbourhood moves and values for equilibrium condition on the solution of the cutting problems were examined for the different temperature values. Secondly, the solution for 29 pieces a test problem was solved using parameteres of the best results obtained in previous work.

At the result of these simulations, the trim loss values of the 17 and 29 pieces test problems obtained by using hybrid GA are varied between 2% and 11%, 5% and 9% respectively. Similarly , the trim loss values of the 17 and 29 pieces test problems obtained by using hybrid SA are varied between 4% and 21%, 10% and 17% respectively. Finally, it is seen that hybrid GA gives better results than hybrid SA for cutting problems.

## TEŐEKKÖR

Bu konuda alıőmam iin beni ynlendiren, araőtırma ve geliőtirme safhasında yardımlarını esirgemeyen danıőmanım sayın Yrd. Do. Dr. Zafer BİNGÖL'e (KO.Ö), akademik alıőma hayatına girmem iin beni yreklendiren ve tez alıőmasında fikirleriyle srekli desteęini grdęm sayın Do. Dr. Kadir ERKAN'a (KO.Ö), bu alıőma boyunca yardımlarını esirgemeyen tm alıőma arkadaőlarım ve her daim yanımda olan, sabırla alıőmalarımı destekleyen anne ve babama en iten teőekkrlerimi sunarım.



## İÇİNDEKİLER

ÖZET: .....	ii
ABSTRACT:.....	iii
TEŞEKKÜR.....	iv
İÇİNDEKİLER .....	v
SİMGELER DİZİNİ ve KISALTMALAR.....	ix
ŞEKİLLER DİZİNİ.....	xi
TABLolar DİZİNİ .....	xvi
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. KESME ve PAKETLEME PROBLEMLERİ .....	6
2.1. Paketleme Problemlerinin Sınıflandırılması .....	7
2.2. Ek Sınıflandırmalar .....	8
2.2.1. Düzgün ve düzgün biçimli olmayan parçaların paketlenmesi .....	8
2.2.2. Özel endüstriyel ihtiyaçlar gerektiren parçaların kesilmesi .....	12
2.3. Paketleme Problemlerinin Çeşitleri.....	13
2.3.1. Şerit paketleme.....	13
2.3.2. Kutu paketleme .....	13
2.4. Endüstriyel Uygulamalar .....	15
2.5. Paketleme Problemleri Uygulamaları .....	16
2.5.1. Paketleme problemleri için genetik algoritmaların uygulanışı .....	16
2.5.1.1. Genetik algoritmalarda ele alınan paketleme problemlerinin sınıflandırılması .....	17
2.5.1.2. İki boyutlu düzgün şerit giyotinsiz kesme ve paketleme problemlerinde kırma yaklaşımlar .....	18
2.5.2. Paketleme problemleri için benzetilmiş tavlama algoritmasının uygulanışı .....	23
2.5.2.1. İki boyutlu giyotinli ve giyotinsiz kesme ve paketleme problemlerinde yaklaşımlar.....	23

BÖLÜM 3. YERLEŞTİRME ALGORİTMALARI .....	25
3.1. Aşağı Sol Algoritması .....	25
3.2. Geliştirilmiş Aşağı Sol Algoritması .....	28
3.3. Aşağı Sol Dolgu Algoritması .....	29
BÖLÜM 4. GENETİK ALGORİTMALAR ve BENZETİLMİŞ TAVLAMA.....	31
4.1. Genetik Algoritmalar .....	31
4.2. Genetik Algoritmaların Diğer Arama ve Eniyileme Yöntemlerinden Farkı...	33
4.3. Temel Bir Genetik Algoritma Çalışması.....	34
4.4. Gösterim.....	35
4.4.1. İkili dizi gösterimi .....	36
4.4.2. Tamsayı veya gerçel sayı gösterimi .....	37
4.5. Başlangıç Nüfusunun Oluşturulması .....	37
4.6. Uygunluk Fonksiyonu.....	38
4.7. Eniyileme Kriteri.....	39
4.8. Genetik Algoritma İşlemcileri.....	39
4.8.1. Seçim.....	39
4.8.1.1. Oransal seçim .....	40
4.8.1.2. Derecelemeye dayalı seçimi.....	41
4.8.2. Çaprazlama işlemcisi .....	41
4.8.2.1. Sıralamaya dayalı çaprazlama.....	43
4.8.2.2. Devirli çaprazlama .....	43
4.8.2.3. Sıralı çaprazlama .....	44
4.8.2.4. Kısmi eşleştirmeli çaprazlama .....	45
4.8.2.5. Düzenli çaprazlama.....	45
4.8.2.6. Stefan Jakobs çaprazlama .....	46
4.8.3. Mutasyon işlemcisi .....	47
4.9. Yeni Nüfusun Oluşturulması .....	48
4.10. Benzetilmiş Tavlama.....	49
4.10.1. Metallerde tavlama.....	50
4.10.2. Benzetilmiş tavlama algoritması.....	51
4.10.3. Benzetilmiş tavlama algoritması parametreleri.....	54
4.10.3.1. Çözümlerin gösterimi.....	54

4.10.3.2. Komşuluk hareketi .....	54
4.10.3.2.1. Yer değiştirme .....	55
4.10.3.2.2. Kaydırma .....	56
4.10.3.3. Soğutma planı .....	56
4.10.3.3.1. Oransal soğutma planı .....	56
4.10.3.3.2. Lundy ve Mees soğutma planı .....	57
4.10.3.4. Durdurma kriteri ve denge koşulu .....	57

## BÖLÜM 5. GENETİK ALGORİTMA ve BENZETİLMİŞ TAVLAMA ile KESME PROBLEMLERİNİN ÇÖZÜM YAKLAŞIMI..... 59

5.1. Çözüm Yaklaşımı .....	60
5.2. Geliştirilmiş Aşağı Sol Algoritmasının Uygulanışı .....	61
5.3. Kıрма Genetik Algoritmalar ile Kesme Probleminin Çözümü .....	64
5.4. Kıрма Benzetiilmiş Tavlama ile Kesme Probleminin Çözümü .....	67

## BÖLÜM 6. GENETİK ALGORİTMA ve BENZETİLMİŞ TAVLAMA ile İKİ BOYUTLU GİYOTİNSİZ KESME PROBLEMİNİN ÇÖZÜMÜ..... 70

6.1. Kıрма Genetik Algoritmalar Kullanılarak Elde Edilen Sonuçların Değerlendirilmesi .....	70
6.1.1. 17 adet parçadan oluşan test problemlerine ait değerlendirmeler .....	70
6.1.1.1. Nüfus büyüklüğünün etkileri .....	71
6.1.1.2. Mutasyon oranının etkileri .....	77
6.1.2. 29 adet parçanın yerleşimi için 6 farklı çaprazlama tekniği kullanılarak gerçekleştirilen test problemine ait değerlendirmeler .....	84
6.1.2.1. Farklı çaprazlama tekniklerinin etkileri .....	84
6.1.2.2. Farklı çaprazlama tekniklerinin nesiller boyunca elde edilen uygunluk değerleri üzerindeki değişimleri .....	92
6.2. Kıрма Benzetiilmiş Tavlama Kullanılarak Elde Edilen Sonuçların Değerlendirilmesi .....	94
6.2.1. 17 adet parçadan oluşan test problemlerine ait değerlendirmeler .....	94
6.2.1.1. Komşuluk hareketinin etkileri .....	94
6.2.1.2. Denge koşulu için değişik komşuluk hareketi sayılarının ve soğutma planlarının etkileri .....	101

6.2.2 29 adet parçanın yerleşimi için Lundy ve Mees soğutma planı kullanılarak gerçekleştirilen test problemine ait değerlendirmeler .....	107
6.3. Uygulanan Kıırma Yaklaşımların Değerlendirilmesi .....	110
<b>BÖLÜM 7. SONUÇLAR.....</b>	<b>113</b>
<b>KAYNAKLAR .....</b>	<b>116</b>
<b>EKLER.....</b>	<b>120</b>
<b>ÖZGEÇMİŞ .....</b>	<b>132</b>





## SİMGELER DİZİNİ ve KISALTMALAR

$C_i$	:i. ötelemedeki çözüm
$C_{i-1}$	:i-1. ötelemedeki çözüm
EB	:En büyük
EK	:En küçük
khs	:Denge koşulu için komşuluk hareketi sayısı
M	:Toplam öteleme sayısı
n	:Parça sayısı
O	:Hesapsal yük.
$P_c$	:Çaprazlama olasılığı
$P_m$	:Mutasyon oranı
$P_s$	:Nüfus büyüklüğü
r	:Permütasyona ait herhangi bir parça
R	:Düzgün dağılımlı rasgele bir sayı
$S_i$	:i bireyinin seçilme olasılığı
T	:Sıcaklık
$T_b$	:Başlangıç sıcaklığı
$T_k$	:k. ötelemedeki sıcaklık
$T_{k+1}$	:k+1. ötelemedeki sıcaklık
$T_s$	:Son sıcaklık
$U_i$	:i bireyinin uygunluk değeri
vu	:Vektör uzaklığı
$\alpha$	:Oransal soğutma planı için sıcaklık azaltma oranı
$\beta$	:Lundy ve Mees soğutma planı için sıcaklık azaltma oranı
$\Delta C$	:İki çözüm arasındaki değişim miktarı
$\varepsilon$	: $2.2204 \times 10^{-16}$
$\pi$	:Permütasyon

AS	:Aşağı sol
ASD	:Aşağı sol dolgu
BT	:Benzetilmiş tavlama
CX	:Cycle crossover
FY	:Fark yöntemi
GA	:Genetik algoritmalar
OBX	:Order based crossover
OX	:Order crossover
PMX	:Partially matched crossover
SH	:Shift neighbouring
SJX	:Stefan Jakobs crossover
SW	:Swap neighbouring
UX	:Uniform crossover



## ŞEKİLLER DİZİNİ

Şekil 2.1. Tekstil endüstrisine ait düzgün biçimli olmayan kesme problemi. ....	9
Şekil 2.2. Dikey olmayan ve dikey yerleşim planı.....	10
Şekil 2.3. a) Giyotinli yerleşim planı. b) Giyotinsiz yerleşim planı. ....	11
Şekil 2.4. a) 2 aşamalı giyotinli kırılmasız kesim. b) 2 aşamalı giyotinli kırılmalı kesim. c) 3 aşamalı giyotinli kesim.....	12
Şekil 2.5. a) İki boyutlu şerit paketleme problemine örnek gösterim. b) İki boyutlu kutu paketleme problemine örnek gösterim.....	14
Şekil 2.6. Genetik algoritmalarla ele alınan paketleme problemlerinin sınıflandırılması. ....	18
Şekil 3.1. AS algoritmasının gösterimi. ....	26
Şekil 3.2. Artan parça sayısına göre elde edilen yerleşim planı sayısı. ....	27
Şekil 3.3. İki ayrı permütasyona sahip yerleşim planı. ....	27
Şekil 3.4. Sekiz dikdörtgen parçanın en iyi yerleşim planı.....	28
Şekil 3.5. Geliştirilmiş AS algoritmasının gösterimi. ....	29
Şekil 3.6. ASD algoritmasının gösterimi. ....	30
Şekil 4.1. Temel bir genetik algoritma çalışması.....	34
Şekil 4.2. İkili kodlanmış bir birey. ....	36
Şekil 4.3. Tamsayı ile kodlanmış bir birey. ....	37
Şekil 4.4. Sıralamaya dayalı çaprazlama.....	43
Şekil 4.5. Devirli çaprazlama. ....	44
Şekil 4.6. Sıralı çaprazlama.....	44
Şekil 4.7. Kısmi eşleştirmeli çaprazlama. ....	45
Şekil 4.8. Düzenli çaprazlama.....	46
Şekil 4.9. Stefan Jakobs çaprazlama. ....	46
Şekil 4.10. Yer değiştirme mutasyonu. ....	48
Şekil 4.11. BT algoritmasının akış diyagramı.....	53
Şekil 5.1. Geliştirilmiş AS algoritmasının işleyişine dair akış diyagramı. ....	63

Şekil 5.2. Kısmi bir yerleşim planı içinde geliştirilmiş AS algoritması kullanılarak dikdörtgen bir parçanın yerleştirilmesi. ....	63
Şekil 5.3. Kıрма GA'nın çalışmasına ait akış diyagramı.....	66
Şekil 5.4. Kıрма BT algoritmasının çalışmasına ait akış diyagramı.....	69
Şekil 6.1 1.test problemine ait küçük parçaların yerleşim planı. ....	70
Şekil 6.2. P1 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri. ...	73
Şekil 6.3. P2 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri. ...	74
Şekil 6.4. P3 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri. ...	74
Şekil 6.5. P4 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri. ...	75
Şekil 6.6. P5 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri. ...	75
Şekil 6.7. Farklı nüfus büyüklükleri ve 0.7 mutasyon oranı kullanılarak elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı. ....	76
Şekil 6.8. Farklı nüfus büyüklükleriyle elde edilen normalize edilmiş uygunluk değerleri.....	77
Şekil 6.9. Farklı mutasyon değerleri için elde edilen P1 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.....	80
Şekil 6.10. Farklı mutasyon değerleri için elde edilen P2 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.....	80
Şekil 6.11. Farklı mutasyon değerleri için elde edilen P3 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.....	81
Şekil 6.12. Farklı mutasyon değerleri için elde edilen P4 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.....	81
Şekil 6.13. Farklı mutasyon değerleri için elde edilen P5 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.....	82
Şekil 6.14. 80 nüfus büyüklüğü ve farklı mutasyon oranları kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı.....	82
Şekil 6.15. Farklı mutasyon oranlarıyla elde edilen normalize edilmiş uygunluk değerleri.....	83
Şekil 6.16. a) OBX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. b) CX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri .....	87

Şekil 6. 17 a) OX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. b) PMX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri .....	88
Şekil 6.18.a) UX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. (b) SJX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. ....	88
Şekil 6.19. OBX tekniği kullanılarak yapılmış çalışmaya ait uygunluk değerlerinin aritmetik ortalama ve standart sapmasının güç spektrum analizi.....	89
Şekil 6.20. SJX tekniği kullanılarak yapılmış çalışmaya ait uygunluk değerlerinin aritmetik ortalama ve standart sapmasının güç spektrum analizi.....	90
Şekil 6.21. Her çaprazlama tekniği için hesaplanan normalize uygunluk değerleri. .	91
Şekil 6.22. 80 nüfus büyüklüğü, 0.7 mutasyon oranı ve OBX çaprazlama tekniği kullanılarak yapılmış çalışmaya ait yerleşim planı. ....	91
Şekil 6.23. 80 nüfus büyüklüğü, 0.7 mutasyon oranı ve SJX çaprazlama tekniği kullanılarak ile yapılmış çalışmaya ait yerleşim planı. ....	92
Şekil 6.24. Seçilen nesiller için OBX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları. ....	93
Şekil 6.25. Seçilen nesiller için SJX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları. ....	93
Şekil 6.26. Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P1 problemine ait normalize uygunluk değerleri. ....	97
Şekil 6. 27. Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P2 problemine ait normalize uygunluk değerleri. ....	97
Şekil 6.28. Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P3 problemine ait normalize uygunluk değerleri. ....	98
Şekil 6.29. Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P4 problemine ait normalize uygunluk değerleri. ....	98

Şekil 6.30. Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P5 problemine ait normalize uygunluk değerleri. ....	99
Şekil 6.31. Denge koşulu için sabit bir khs 5, yer değiştirme komşuluk hareketi ve Lundy ve Mees soğutma planı kullanılarak P4 problemi için yapılan çalışmaya ait yerleşim planı. ....	99
Şekil 6.32. Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış çalışmaların ortalama normalize uygunluk değerleri. ....	100
Şekil 6.33. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P1 problemine ait normalize uygunluk değerleri. ....	103
Şekil 6.34. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P2 problemine ait normalize uygunluk değerleri. ....	104
Şekil 6.35. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P3 problemine ait normalize uygunluk değerleri. ....	104
Şekil 6.36. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P4 problemine ait normalize uygunluk değerleri. ....	105
Şekil 6.37. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P5 problemine ait normalize uygunluk değerleri. ....	105
Şekil 6.38. Yer değiştirme komşuluk hareketi, Lundy ve Mees soğutma planı ve khs 3 kullanılarak P4 problemi için yapılan çalışmaya ait yerleşim planı. ....	106
Şekil 6.39. Sekiz başlangıç sıcaklığı ve yer değiştirme komşuluk hareketi kullanılarak, soğutma planlarına ve üç değişik khs (3, 5, 10) göre tüm problemlerin normalize ortalama değerleri. ....	107
Şekil 6.40. 29 adet parçanın yerleşimi sonucu elde edilen normalize uygunluk değerleri.....	109
Şekil 6.41. 0.3 başlangıç sıcaklığı, yer değiştirme komşuluk hareketi, Lundy ve Mees soğutma planı, khs 3 kullanılarak elde edilen çalışmaya ait yerleşim planı. ....	109

Şekil 6.42. GA ve BT yöntemleri kullanılarak 17 parça içeren P4 problemi için elde edilen en iyi uygunluk değerleri.....	111
Şekil 6.43. 29 parça içeren problem için GA ve BT kullanılarak elde edilen en iyi uygunluk değerleri. ....	111



## TABLolar DİZİNİ

Tablo 2. 1. Farklı geometrik biçimler içeren iki boyutlu paketleme problemleri.....	9
Tablo 2.2. İki boyutlu giyotinsiz düzgün biçimli paketleme problemleri için kırma genetik algoritmalar kullanarak literatürde yer alan çalışmalar.....	22
Tablo 2.3.İki boyutlu dikdörtgen biçimindeki paketleme problemleri için benzetilmiş tavlama yaklaşımlarının karşılaştırılması.....	24
Tablo 4.1. Fiziksel sistemler ve eniyileme problemleri arasındaki benzerlik.....	49
Tablo 6.1. Beş test problemi, farklı nüfus büyüklükleri ve sabit 0.7 mutasyon oranıyla yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri.....	71
Tablo 6.2. Nüfus büyüklüklerine göre problemlerde elde edilen en küçük fire değerleri.....	72
Tablo 6.3. Beş test problemi, 80 nüfus büyüklüğü ve farklı mutasyon oranlarıyla yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri..	78
Tablo 6.4. Farklı mutasyon oranlarına göre problemlerde elde edilen en küçük fire değerleri.....	78
Tablo 6.5. 6 farklı çaprazlama tekniği için yapılmış çalışmalar sonucunda elde edilen fire değerleri.....	85
Tablo 6.6. OBX ve SJX teknikleri kullanılarak elde edilen uygunluk değerlerine ait olasılıksal değerler. ....	86
Tablo 6.7. Beş test problemi için, sabit bir khs, iki farklı komşuluk hareketi ve iki soğutma planı kullanılarak yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri. ....	95
Tablo 6.8. Denge koşulu için üç değişik khs ve kaydırma komşuluk hareketi kullanılarak elde edilen fire değerleri. ....	102
Tablo 6.9. 29 adet parçanın yerleşimi sonucunda elde edilen fire değerleri.....	108
Tablo 6.10. Kesme problemleri için elde edilen en küçük ve en büyük fire değerleri. ....	110



## **BÖLÜM 1. GİRİŞ**

Günümüz ekonomik koşullarında işletmelerin, amaçlarına ulaşabilmek için kaynaklarını verimli kullanmak zorunda oldukları açıktır. Bu nedenle üretim maliyetinin azaltılması daima rekabetçi bir ortam içinde bulunan imalat endüstrilerinde en önemli sorunlardan biri haline gelmiştir.

İmalat endüstrilerinde üzerinde çalışılan malzemeler belirli boyutlardadır. Çeşitli boyut ve miktarlardaki üretimler için kullanılan ana malzemenin verimli bir şekilde değerlendirilmesi yerleşim planlarının en iyi şekilde gerçekleştirilmesiyle olur. Bu da üretim maliyetinin azaltılması ve kaynakların verimli kullanımı için gereklidir. Özellikle toplu üretim yapan endüstrilerde bir kaynak kullanımı sorunu olarak karşımıza çıkan yerleşim planlarının en iyi şekilde gerçekleştirilmesi problemiyle çok sık karşılaşılmaktadır.

Pek çok değişik endüstride farklı koşullar ve amaçlarla karşımıza çıkan kesme ve paketleme problemleri birer eniyileme problemleridir. Bu problemler ayrıca matematiksel olarak çok boyutlu uzayda kombinasyonel eniyilemenin yapıldığı ve çözümün belirli bir katsayısal modelle gösterilemediği problemler olarak da bilinir.

Kesme ve paketleme problemleri üzerine son 30 yıldır yapılan çalışmalar literatürde oldukça geniş yer edinmiştir. Problemlerin çeşitliliğine ve uygulama alanlarının farklılığına rağmen literatürde farklı isimler altında görünen kesme ve paketleme problemleri aslında benzerdirler. Pek çoğuyla farklı alanlarda karşılaşılmasına rağmen çözüm yaklaşımı aynı temel mantıksal yapıya sahiptir. Kesme ve paketleme problemlerinin çeşitleri çoğunlukla üzerinde yerleşimin yapılacağı ana parçaya göre değişim göstermektedir.

Problemin karmaşıklığı ve çözüm yaklaşımı, ana parça üzerine yerleştirilecek parçaların geometrisine ve problemin doğasından gelen kısıtlamalara bağlıdır. Ağaç, cam, kağıt ve metal endüstrilerinde düzgün biçimli parçaların kesilmesiyle ilgilenilirken, gemi, tekstil, deri ve metal endüstrileri düzgün biçimli olmayan başka bir deyişle bilinen geometrik şekiller içermeyen parçaların kesilmesi veya paketlenmesiyle ilgilenilir.

Küçük bir arama uzayı içinde olması durumu hariç, kesme ve paketleme problemlerine en iyi çözümün üretilmesi neredeyse imkansızdır. Bu problemler tam bir matematiksel modele sahip olmayan (NP-complete) problemlerdir. Probleme farklı en iyi çözümlerin bulunması için, büyük arama uzayı içinde düzenli bir arama gereklidir. Bu problemlerin çoğu NP-complete olduğundan, yönlendirilmemiş arama oldukça verimsiz olur (Callaghan et al. 1999). Bu nedenle araştırmalar, en iyi çözüme yakın iyi çözümleri verimli bir şekilde bulan yaklaşım teknikleri üzerinde yoğunlaşmaktadır. Kesme ve paketleme problemleri için uygulanabilir bu tekniklerden bazıları yapay sinir ağları, genetik algoritma ve benzetilmiş tavlama içerir. Aşağıda bu teknikleri kullanarak yapılan iki boyutlu giyotinsiz paketleme problemleri için bazı kırma yaklaşımlar kronolojik sırayla verilmektedir.

Smith (1985), paketleme problemlerine çözüm bulmak için genetik algoritmaları kullanan ilk araştırmacıdır. İki boyutlu dikdörtgen paketleme problemleri için iki ayrı kod çözücü yerleştirme algoritmasıyla birlikte genetik algoritmaları kullanmıştır.

Kämpke (1988), benzetilmiş tavlama yöntemini kullanarak paketleme problemleri üzerine çalışan ilk araştırmacılardan biridir. Benzetilmiş tavlama yöntemini farklı soğutma stratejileri kullanarak bir boyutlu kutu paketleme problemlerine uygulamıştır.

Dowland (1993), benzetilmiş tavlama yöntemini kullanarak özdeş ve özdeş olmayan kutuları içeren yükleme problemleri üzerinde çalışmıştır.

Jakobs (1996), kod çözücü olarak aşağı sol algoritmasıyla birlikte sıralamaya dayalı bir genetik algoritmayı kullanan kırma bir yaklaşım gerçekleştirmiştir.

Dagli and Poshyanonda (1997)'nin çalışmalarında kullandıkları kod çözücü algoritma, kaydırma tekniğine dayalı yapay sinir ağlarıyla birleştirilmiş bir yerleştirme algoritmasıdır. Genetik algoritmalar, bu kod çözücü yerleştirme algoritmasında kullanılmak üzere yerleşim sırasını bulmak için kullanılmıştır.

Liu and Teng (1999) tarafından gerçekleştirilen çalışmada Jakobs (1996) tarafından kullanılan kod çözücü yerleştirme algoritmasının geliştirilmesi amaçlanmıştır. Genetik algoritmalar, paketleme problemleri için Jakobs (1996)'un kullandığı kaydırma ilkesine göre çalışan geliştirilmiş aşağı sol algoritmasıyla birlikte daha kolay paketleme sağlayan kırma bir yaklaşım olarak ileri sürülmüştür.

Hopper and Turton (1997, 1999) çalışmalarında yerleşim planı içerisinde aşağı sol koşulunu sağlayan bir yerleştirme algoritması kullanmışlardır. Geliştirilmiş bu aşağı sol algoritma kısmi yerleşim planı içerisinde etrafı parçalarla çevrili boş alanları da değerlendirebilmektedir. Hopper ve Turton çalışmalarında bu yerleştirme algoritmasını genetik algoritmalar ve benzetilmiş tavlama yöntemi ile birleştirmişlerdir.

Faina (1999), giyotinli ve giyotinsiz kesme problemleri için iki ayrı kod çözücü yerleştirme algoritması kullanarak kırma bir benzetilmiş tavlama algoritması geliştirmiştir.

Leung et al. (1999, 2001), fark yöntemi algoritması adıyla bilinen kısmi yerleşim planı içerisinde yerleştirmeler yapabilen bir aşağı sol algoritması geliştirmiştir. Genetik algoritma ve benzetilmiş tavlama yöntemiyle birleştirilen bu yerleştirme algoritması Jakobs (1996) ve Liu and Teng (1999)'un kullandıkları kod çözücü yerleştirme algoritmalarına göre daha iyi yerleşim planları üretmektedir. Ancak hesaplaması daha çok zaman alır.

Kesme ve paketleme problemlerinde yapay sinir ağları nadiren kullanılan bir yöntemdir. Genetik algoritmalar ve benzetilmiş tavlama, doğrusal, doğrusal olmayan, gerçel sayı ve tamsayı eniyileme problemlerinin çözümünde sıklıkla kullanılan eniyileme yöntemleridir. Genetik algoritmanın yerel en küçük değere takılmama,

genetik çeşitliliği sürdürme ve benzetilmiş tavlamanın çözüme daha hassas yakınsama özelliklerinden yararlanılarak yerleşim planı için eniyileme yapılır.

Bu tez çalışmasında, genetik algoritmalar ve benzetilmiş tavlama yöntemleri Matlab yazılımı kullanılarak geliştirilen bir benzetim programı ile (Ek-A), iki boyutlu düzgün dikdörtgen parçalar içeren giyotinsiz bir kesme problemine uygulanmıştır. Genetik algoritmalar ve benzetilmiş tavlama en iyi yerleşim planını veren parça yerleşim sırası permütasyonunu elde etmek için kullanılmıştır. Problemler için elde edilebilecek en iyi yerleşim planı kullanılmayan alanın sıfır başka bir deyişle firenin sıfır olduğu durumda elde edilmektedir. Geliştirilmiş aşağı sol yerleştirme algoritması bu permütasyonu kullanarak düzgün biçimli dikdörtgen parçaların yerleşimini gerçekleştirmektedir.

İkinci bölümde, kesme ve paketleme problemleri hakkında bilgi verilmiştir. Paketleme problemlerinin sınıflandırılması ve çeşitleri incelenmiş, bazı endüstriyel uygulamalar üzerinde durulmuştur. Geçmişten günümüze kadar kesme ve paketleme problemlerinin çözümü için genetik algoritma ve benzetilmiş tavlama yöntemlerinin kullanıldığı literatür çalışması ikinci bölüm içinde incelenmiştir.

Üçüncü bölümde, elde edilen parça yerleşim sırasına göre yerleştirme yapan kod çözücü yerleştirme algoritmaları üzerinde durulmuştur.

Dördüncü bölümde, birer eniyileme yöntemi olarak kesme ve paketleme problemlerinde kullanılacak olan genetik algoritma ve benzetilmiş tavlama yöntemleri yer almaktadır. Temel bir genetik algoritmanın çalışması ve genetik algoritma ile problem çözümünde kullanılan genetik işlemciler incelenmiştir. Ayrıca eniyileme problemlerinin çözümünde sıklıkla başvuru alan benzetilmiş tavlama yöntemi ve parametreleri dördüncü bölümde incelenmiştir.

Beşinci bölümde, kırma genetik algoritma ve kırma benzetilmiş tavlama yöntemi kullanılarak kesme problemlerinin çözümüne ait bir çözüm yaklaşımı üzerinde durulmuştur.

Altıncı bölümde, kırma genetik algoritma ve kırma benzetilmiş tavlama yöntemleri ile gerçekleştirilen eniyileme çalışması yer almaktadır. Ana parçanın 200x200 birimlik bir alan ile sınırlandırıldığı problemler birbirinden farklı düzgün dikdörtgen 17 ve 29 adet parçadan oluşmaktadır. Problemler öncelikle kırma genetik algoritma yöntemiyle ele alınmıştır. Farklı nüfus büyüklükleri, farklı mutasyon oranları ve farklı çaprazlama tekniklerinin kullanılmasıyla problemlerin çözümleri üzerindeki değişimleri incelenmiştir. Çalışmanın ikinci bölümünde aynı kesme problemleri kırma benzetilmiş tavlama yöntemiyle ele alınmıştır. Farklı sıcaklık değerlerinde farklı soğutma planları, farklı komşuluk hareketleri ve denge koşulu için değişik değerler kullanılarak problemlerin çözümleri üzerindeki değişimleri incelenmiştir.

Yedinci bölümde tez çalışmasının sonuçları değerlendirilmiştir.



## **BÖLÜM 2. KESME ve PAKETLEME PROBLEMLERİ**

Kesme ve paketleme problemleri, bir ana parça üzerinde düzgün biçimli veya düzgün biçimli olmayan çok sayıda küçük parçanın en elverişli yerleşim planının bulunmasıyla ilgilenen eniyileme problemleridir (Leung et al. 2001). Ayrıca matematiksel olarak çok boyutlu uzayda kombinasyonel eniyileme yapan ve çözümünün bulunması için belirli bir katsayısal modelle gösterilemeyen problemler olarak da bilinir. Ana parçanın alanı, yerleştirilecek parçaların toplam alanına eşit olabileceği gibi sınırsız da kabul edilebilir. Yerleştirme bir kurallar seti veya kısıtlamalarla tanımlanır.

Paketleme problemlerinde ana parça (konteynır, kutu gibi) küçük parçalar (kutular gibi) ile doldurulması gereken boş bir nesne olarak tanımlanır. Kesme problemlerinde ise ana parça (metal levha, rulo gibi) küçük parçalar ile bölümlere (2 boyutlu şekiller gibi) ayrılacak bir nesne olarak tanımlanır. Yerleşim planında kullanılmayan alan veya yerleşim sırasında olup da yerleşim planı içinde yer almayan parçalar fire olarak isimlendirilir. Bu problemlerin amacı, yerleştirme yapılacak ana parçanın kullanılabilirliğini arttırmak ve böylelikle fireyi en aza indiren “yerleşim planını” bulmaktır (Hopper and Turton 1997, 2001, Leung et al. 2001). Yerleşim veya kesme planı küçük parçaların ana parça üzerindeki yerleşimini göstermektedir.

Parçaların yerleştirileceği malzeme ve dolduracağı boşluk ikilisinden dolayı kesme ve paketleme problemleri arasında yakın bir ilişki vardır. Kesme problemleri ana parça içerisine küçük parçaların doldurulacağı paketleme problemleri gibi düşünülebilir. Paketleme problemleri de ana malzeme olarak kullanılan parçadan küçük parçaların kesileceği kesme problemleri olarak düşünülebilir (Dyckhoff 1990).

Farklı kořullar ve amaçlarla karřımıza çıkan kesme ve paketleme problemleriyle günümüzde pek çok endüstride yaygın olarak karřılařılmaktadır. Problemin karmařıklığı ve çözüm yaklaşımı yerleřtirilecek parçaların geometrisine ve problemin doğasından gelen kısıtlamalara baėlıdır. Aėaç, cam, kaėıt ve metal endüstrisi düzgün biçimli parçaların kesilmesiyle ilgilenirken gemi, tekstil, deri ve metal endüstrisi düzgün biçimli olmayan parçaların kesilmesiyle ilgilenir. Kullanılan ana malzemeden elde edilen önemli derecedeki kazançların üretim maliyetini oldukça düşürmesi, toplu üretim yapan daha pek çok endüstride bu konu üzerine yapılan çalıřmalara ilgiyi arttırmaktadır.

Bu bölümde sırasıyla paketleme problemlerinin sınıflandırılması, çeřitleri ve endüstriyel uygulamaları anlatılmıřtır. Ayrıca geçmiřten günümüze kadar kesme ve paketleme problemlerinin çözümleri için genetik algoritma ve benzetilmiş tavlama yöntemleri ile geliřtirilen çözüm yaklařımları üzerinde durulacaktır.

## **2.1. Paketleme Problemlerinin Sınıflandırılması**

Kesme ve paketleme problemleri üzerine son 30 yıldır yapılan arařtırmalar literatürde geniş çapta yer edinmiřtir. Bu problemler yalnızca imalat sektörüyle sınırlı kalmamıř daha pek çok endüstride kendini göstermiřtir.

Problemlerin ve uygulama alanlarının farklılığına raėmen literatürde farklı isimler altında görünen paketleme problemleri aslında benzerdir ve incelendiklerinde görölmektedir ki pek çoėuyla farklı alanlarda karřılařılmasına raėmen aynı temel yapıya sahiptir. Farklı çalıřma alanlardaki bilgi deėiřimine yardım etmek için bu problemlerin genel karakteristik ve özellikleri belirlenmiř ve bir sınıflandırma sistemi kurulmuřtur (Dyckhoff 1990). Bu çalıřmada boyutsal ve boyutsal olmayan boşluklar içeren paketleme problemleri arasındaki ayırım göz önüne serilmiřtir. Kesme ve paketleme veya yükleme problemlerinden oluřan ilk grup öklit uzayında 3 boyuta kadar olan problemleri tanımlamıřtır (kesme problemi gibi). Diėer grup ise aėırlık, zaman veya mali büyüklükler gibi boyutsal olmayan büyüklüklerin dahil olduėu soyut kesme ve paketleme problemlerinden oluřmaktadır (bellek tahsisi gibi).



Dyckhoff'un sınıflandırma sistemi kesme ve paketleme problemlerinin dört önemli karakteristiğini açıklamaktadır (Dyckhoff 1990, Haessler and Sweeney 1991):

- En önemli karakteristik yerleşim planının geometrisini açıklamak için gereken en az boyut sayısını tanımlayan boyutsallıktır. Boyut sayısı 3'ü aşan problemler (çok boyutlu uzaya sahip problemler), zaman, ağırlık gibi boyutsal olmayan boşluklar söz konusu olduğunda ortaya çıkmaktadır.
- Görev türü, tüm ana parçalar ve küçük parçalarla veya içlerinden sadece birinin belirlenip belirlenmemesiyle tanımlanır.
- Ana parçaların çeşitliliği, özdeş veya farklı biçimli ana parçaya sahip problemler içinde farklılık göstermektedir.
- Küçük parçaların çeşitliliği, parça sayısı ve biçimleriyle ilgilidir. Problemler birkaç küçük parça, özdeş küçük parçalar, pek çok farklı biçime sahip pek çok parça ve nispeten birbirlerinden daha az farklı biçimlere sahip küçük parçalar içerebilir.

## **2.2. Ek Sınıflandırmalar**

### **2.2.1. Düzgün ve düzgün biçimli olmayan parçaların paketlenmesi**

Genel olarak bir paketleme probleminin amacı sınırlandırılmış bir alanda çakışma olmaksızın parçaların verimli bir yerleşimini gerçekleştirmektir. Bu nedenle, paketleme problemlerinin karmaşıklığı paketlenen parçaların geometrik biçimleriyle oldukça ilgilidir. Parçalar geometrik biçimleri açısından ikiye ayrılırlar:

- Dikdörtgen ve daire gibi sadece birkaç parametre ile tanımlanmış düzgün biçimli parçalar,
- Simetrik olmayan ve içbükeylik veya dışbükeylik içeren düzgün biçimli olmayan parçalar.

Düzgün biçimli paketleme problemleri, çoğunlukla dikdörtgen biçimindeki bir parçanın içerisine çok sayıda dikdörtgen parçanın paketlenmesiyle ilgilidir. Düzgün biçimli olmayan paketleme problemleri ise gemi endüstrisinde yerleştirme ve tekstil



endüstrisinde kalıp yerleştirme gibi paketleme problemleri ile ilgilidir. Şekil 2.1’de çoğunlukla tekstil endüstrisinde karşılaşılan düzgün biçimli olmayan kesme problemine ait bir yerleşim görülmektedir Endüstrideki farklı geometrik biçimler içeren paketleme problemlerine örnekler Tablo 2.1’de verilmiştir. (Hopper 2000).



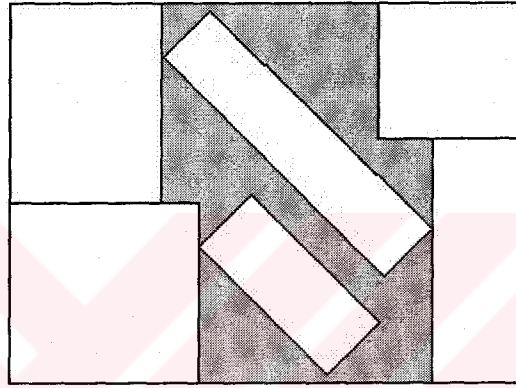
Şekil 2.1. Tekstil endüstrisine ait düzgün biçimli olmayan kesme problemi.

Tablo 2. 1. Farklı geometrik biçimler içeren iki boyutlu paketleme problemleri.

Geometrik biçim	Uygulama örneği
Daire, elips.	Konteynırlara boru yüklenmesi.
Çokgenler: dışbükey veya içbükey.	Metal endüstrisindeki düzgün biçimli olmayan şekillerin paketlenmesi.
Serbest biçimli şekiller: kavisli ve düz hatlı kesimler içeren parçalar.	Tekstil endüstrisindeki kalıp yerleşim problemi.
Tam veya kısmi çevrimli şekiller.	Gemi yükleme.

İki boyutlu dikdörtgen biçimindeki paketleme problemlerinde, paketlenen parçaların geometrisi temel alınmış iki tip yerleşim söz konusudur (Hinxman 1980).

- Dikey olmayan (non-orthogonal) yerleşim, yerleşim planında yerleştirilecek düzgün biçimli parçanın kenarlarının ana parçanın kenarlarına paralel olmama durumudur.
- Dikey (orthogonal) yerleşim, yerleşim planında yerleştirilecek düzgün biçimli parçanın kenarlarının ana parçanın kenarlarına paralel olması durumudur. Şekil 2.2 dikey olmayan ve dikey yerleşim planlarını içeren bir paketleme problemine ait yerleşim planını göstermektedir.

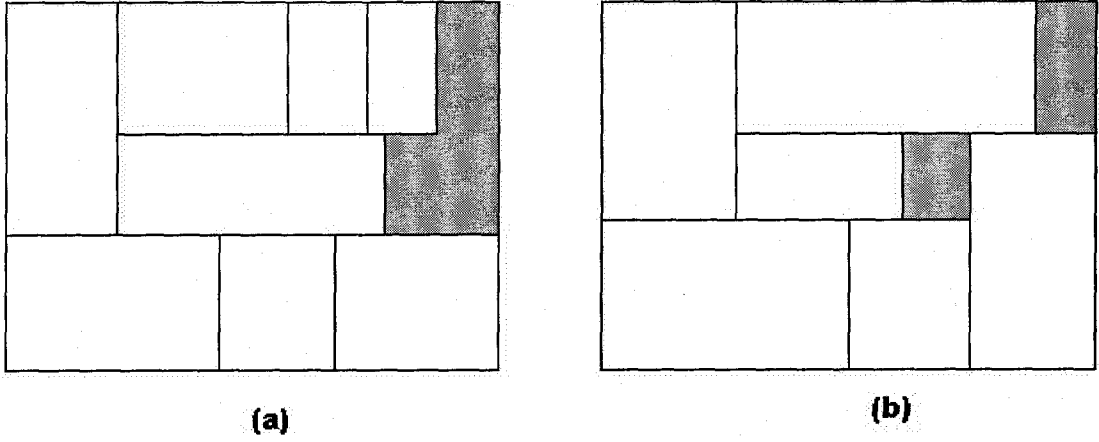


**Şekil 2.2.** Dikey olmayan ve dikey yerleşim planı.

Parçaların geometrileri temel alınarak gruplandırılan bu iki yerleşim tipine ek olarak parçaların kesim biçimine göre dikey yerleşim de iki temel gruba ayrılır (Haessler and Sweeney 1991):

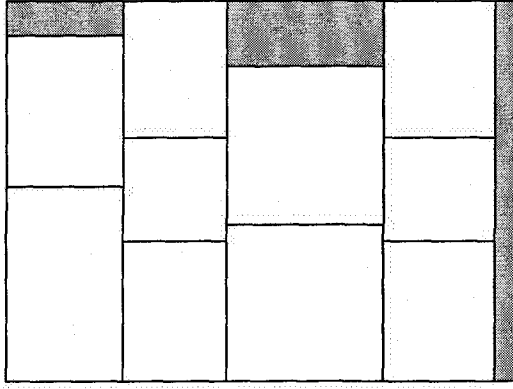
- Giyotinli (guillotineable) yerleşim için giyotinli problemler özel bir kısıtlama içerirler. Bu kısıtlama yerleştirilmiş her parçanın kesilmesinden sonra kesimin kalan parçanın bütün uzunluğu boyunca bir uçundan diğer ucuna kadar yapılmasına izin veren bir yerleşimi gerektirir. Bu tip problemlere çoğunlukla cam ve kağıt endüstrisinde yapılan çalışmalar örnek gösterilebilir. Şekil 2.3 (a)'da giyotinli yerleşim planına örnek bir yerleşim verilmiştir.
- Giyotinsiz (non-guillotineable) yerleşim için giyotinsiz problemler kısıtlamalarla sınırlandırılmazlar. Herhangi bir parça, yerleşim planında çakışmaya imkan

vermeyen mümkün olan her konuma yerleştirilebilir. Şekil 2.3 (b)'de giyotinli yerleşim planına örnek bir yerleşim verilmiştir.

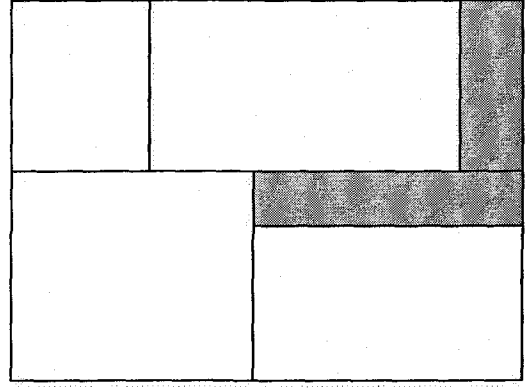


**Şekil 2.3. a)** Giyotinli yerleşim planı. **b)** Giyotinsiz yerleşim planı.

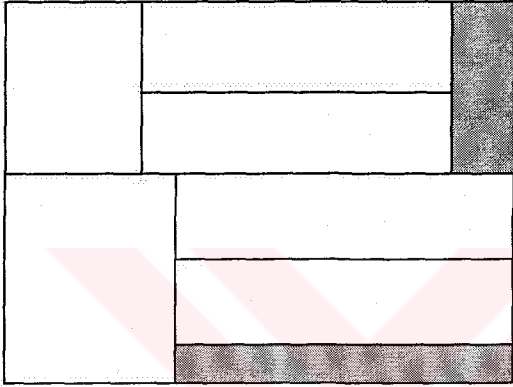
Giyotinli kesim, 2 aşamalı giyotinli kırılmasız kesim, 2 aşamalı giyotinli kırılmalı kesim ve 3 aşamalı giyotinli kesim olarak sınıflandırılmıştır (Gilmore and Gomory 1965). Şekil 2.4 a'da 2 aşamalı giyotinli kırılmasız kesim, Şekil 2.4 b'de 2 aşamalı giyotinli kırılmalı kesim ve Şekil 2.4 c'de 3 aşamalı giyotinli kesime örnekler gösterilmiştir. Sözü edilen aşamalar parçanın yerleşim planından ne şekilde kesileceği ile ilgilidir. Şekil 2.4 a'da gösterilen yerleşim planı için ilk kesimler dikey yönde yapılır. Daha sonra yatay yönde yapılan kesimlerle parçaların son halleri elde edilir. Şekil 2.4 b'de gösterilen yerleşim planı için 3. bir kesim parçayla bitişik fazlalık kısımları çıkarmak için yapılır. Bu 3.kesim işlemi dışında 2 aşamalı giyotinli kırılmasız kesimle benzer bir kesim işlemi gerçekleştirilir. Şekil 2.4 c'de gösterilen yerleşim planı için 3. bir kesim 2 ayrı parçayı oluşturmak için yapılmaktadır.



(a)



(b)



(c)

**Şekil 2.4. a) 2 aşamalı giyotinli kırılmasız kesim. b) 2 aşamalı giyotinli kırılmalı kesim. c) 3 aşamalı giyotinli kesim.**

### 2.2.2. Özel endüstriyel ihtiyaçlar gerektiren parçaların kesilmesi

Dyckhoff tarafından yapılan sınıflandırma içerisinde kesme ve paketleme problemleri ortak bir yapıya sahip olmalarına rağmen, çeşitli endüstriler tarafından getirilen bir takım kısıtlamalarla farklılıklar gösterebilirler (Dyckhoff 1990). Küçük parçaların bütünüyle ana parça içine yerleştirilebilmesi ve çakışma olmaması bütün problemler için kabul edilen iki genel özelliktir. Parçaların olası yerleşimleri hakkındaki detayları içeren ek kısıtlamalar özel endüstriyel ihtiyaçlar olarak belirtilir. Bazı parçalar ana parçanın yalnızca belirli bir alanında (ayakkabı imalatı) veya belirli bir yönünde (tekstil endüstrisi) yerleştirilmeyi gerektirebilirler. Örnek vermek gerekirse metal kesme işleminde parçalar birbirleri arasında en az mesafe kalacak şekilde yerleştirilmelidirler. Ağaç, cam ve kağıt endüstrisinde ise kesme işlemi giyotinli yerleşim ister.

### **2.3. Paketleme Problemlerinin Çeşitleri**

Paketleme problemleri, farklı kısıtlamalar ve amaçlar içeren çeşitli uygulama alanlarında yer almaktadır. Genellikle ana parçanın çeşidine göre değişen şerit paketleme (strip packing) ve kutu paketleme (bin packing) olarak ikiye ayrılır (Hopper and Turton 2001).

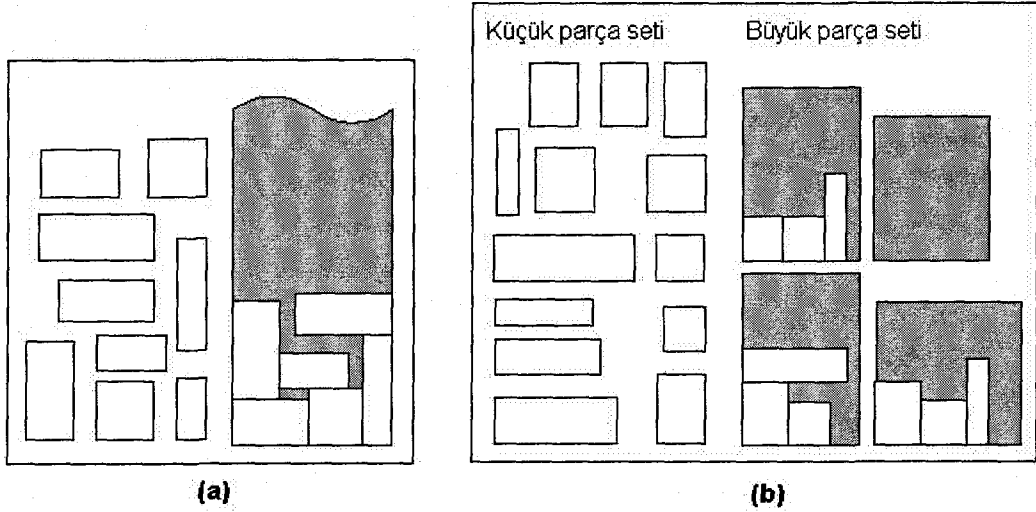
#### **2.3.1. Şerit paketleme**

Kağıt ve tekstil endüstrisinde hammadde rulo halinde mevcuttur. Bu nedenle paketleme işlemi sadece yerleşim planının yüksekliğini azaltmayı amaçlar. Endüstrideki bu paketleme işlemi şerit paketleme problemi olarak bilinir. Şekil 2.5 (a)'da yerleştirilecek dikdörtgen parçalar içeren bir şerit paketleme problemine örnek gösterilmektedir.

#### **2.3.2. Kutu paketleme**

Çok sayıda kutunun paketlenmesiyle ilgilenen kutu paketleme işleminde hammadde levhalar halinde mevcuttur. Kullanılan hammadde sınırsız olduğundan amaç kullanılan toplam hammaddeyi en aza indiren ve ayrıca parça yerleşim sırasındaki bütün parçaları yerleştiren kutu yerleşim sırasını bulmaktır. Uygulamaya bağlı olarak levhalar özdeş veya farklı boyutlarda olabilir.

Bir boyutlu paketleme problemlerinde, paketlenen parçaların genişliğinin hammaddenin genişliği ile aynı olduğu kabul edilir. Bu nedenle bir boyutlu paketleme problemlerinde sadece bir boyut önemlidir. İki boyutlu paketleme problemleri, paketleme problemlerinde farklı genişlikli parçaların kullanımıyla elde edilir (Coffman et al. 1984). Düzgün biçimli kutu paketleme daha çok yükleme işlerinde karşılaşılan 3 boyutlu problemler içinde oldukça yaygın olarak kullanılmaktadır. Şekil 2.5 (b)'de iki boyutlu kutu paketleme problemine bir örnek gösterilmiştir.



**Şekil 2.5. a) İki boyutlu şerit paketleme problemine örnek gösterim. b) İki boyutlu kutu paketleme problemine örnek gösterim.**

Endüstriyel uygulamalarda kutu paketleme problemleriyle çeşitli şekillerde karşılaşılmaktadır. Bu problemler amaçlarına göre değişen çeşitli problemlere ayrılır (Hinxman 1980, Dowsland and Dowsland 1992). Ayrıca geniş uygulama alanlarına sahip çeşitli endüstriyel problemler bu problem tiplerinin iki veya daha fazlasını içerebilirler. Bu problemler aşağıda sıralanmıştır:

- Fire problemi (trim loss problem), verilen ana parça üzerinde yerleşim sırası belirli küçük parçaların yerleştirilmesiyle ilgilidir. Bu problemler uzunluğu sınırsız kabul edilmiş ana parçanın kullanım alanını en aza indirmeyi böylece toplam maliyeti düşürmeyi hedefler.
- Karışım problemi (assortment problem), yerleşim sırasını gerçekleştirmek için gereken ana parçanın boyutunun belirlenmesini içerir. Bu sebeple yerleşim sırası kullanılacak en iyi ana parçanın belirlenmesini sağlar.
- Kesme problemi (cutting problem), verilen parça yerleşim sırasına ait parçaların ana parçadan kesilmesiyle ilgilidir. Problem fire problemi ve karışım problemi olmak üzere 2 ayrı problem olarak ele alınabilir. Fire problemi kesme yerleşim planının belirlenmesiyle, karışım problemi ana parçanın belirlenmesiyle ilgilidir.

- Sırt çantası problemi (knapsack problem) yerleşim sırasındaki her parçaya ait bir değer olduğu problemlerdir. Bu problemlerin amacı ana parça içinde paketlenen küçük parçaların toplam değerinin alabileceği en yüksek değere çıkartılabilmesidir (Martello and Toth 1990).
- Yükleme problemi (loading problem), bir kutu içerisine çok sayıda parçanın en elverişli yerleşiminin yapılmasıyla ilgilidir. Yükleme problemi imalatçı açısından özdeş kutuların bir kutuya yüklenmesi gibi kabul edilebilir.

#### 2.4. Endüstriyel Uygulamalar

Endüstriyel uygulamalarda, diğer faktörlerin çoğu fireyi azaltmaktan ziyade en son yerleşim planını kararlaştırır. Kullanılan malzemenin özellikleri, kesme işlemi, zamanlama, yerleştirme gereksinimleri ile ilgili belirli kısıtlamalar yerleşim sürecinde oldukça etkili faktörlerdir (Hopper and Turton 2001).

Metal endüstrisinde kullanılan metalin homojen olmayan özellikleri, tekstil endüstrisinde kullanılan deri gibi doğal malzemelerin çeşitli nitelikteki kullanım alanları parçaların olası yönlendirme sayısını sınırlamaktadır.

Parçaları elde etmek için kullanılan kesme tekniği yerleşim planı üretiminde önemli bir etkiye sahiptir. Kullanılan kesme teknolojisine bağlı olarak parçalar arasında parçalara da zarar vermemek koşuluyla belirli uzaklıklar gereklidir.

Kesilecek yada paketlenen parçaların sırası bir sonraki imalat işlemi için önemli olabilir. Bu nedenle parçaların ana parçadan kesilmesinde yada parçaların ana parça içine paketlenmesinde zamanlama göz ardı edilemez.

Kısmi yerleşim planı içinde daha küçük parçaların yerleştirilebilmesi için yeterince büyük boş alanlar olabilir. Yerleşim planı içinde oluşan zararı azaltmak için yerleştirme algoritması düzgün biçimli olmayan kısmi yerleşim planlarının boş alanlarına yeni parçaları yerleştirebilir ve bu parçaların nerelere yerleştiklerini yerleşim süreci içinde izleyebilmelidir.



## 2.5. Paketleme Problemleri Uygulamaları

Paketleme problemleri üzerindeki arařtırmaların önemli bir kısmı bu problemlerin çözümü için bir algoritma geliřtirmek adına yürütülmüřtür. Doğrusal programlama teknikleri gibi gerekirci yöntemler kesin çözümleri elde etmek için geliřtirilmiřtir. Kesin çözümler için geliřtirilen algoritmalar sadece belirli bir karmařıklık derecesine kadar etkili çalıřtıđından, daha karmařık kombinyonel problemler için bařka teknikler uygulanır. Belirli paketleme problemleri için ileri sürülmüř çok sayıda probleme özgü yöntem vardır. Bu yöntemlerden bařka genetik algoritma ve benzetilmiř tavlama algoritması gibi genel kullanım amaçlı algoritmalar diđer kombinyonel problemlerde olduđu gibi paketleme görevleri için de kullanılmaktadır.

### 2.5.1. Paketleme problemleri için genetik algoritmaların uygulanıřı

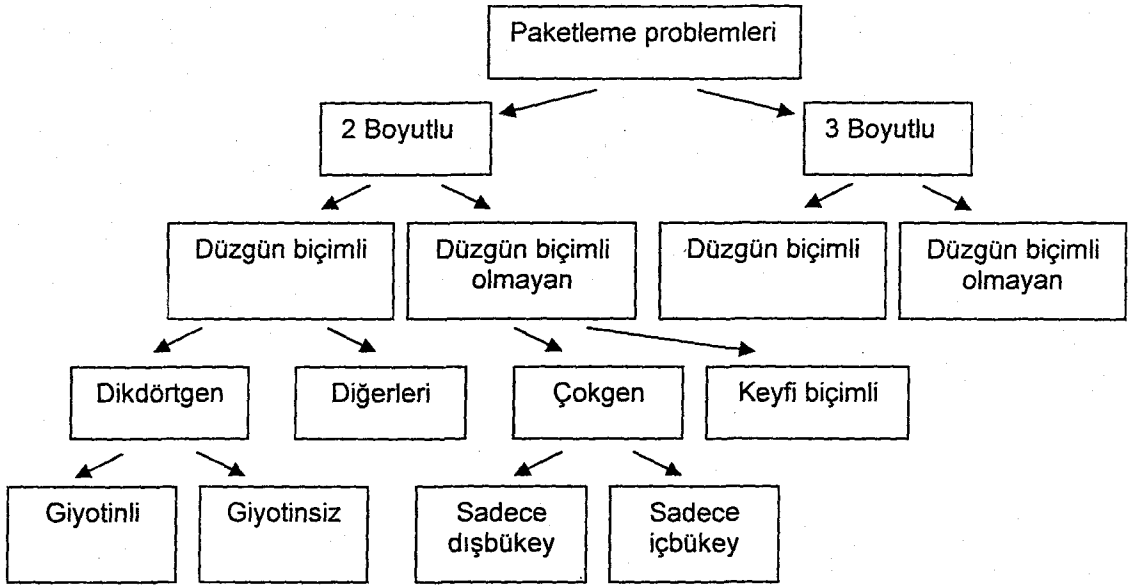
Genetik algoritmalar, 70'li yılların bařında geliřtirilmiř olmalarına rađmen 80'li yılların ortalarına kadar paketleme problemlerine uygulanmamıřtır. Smith (1985) paketleme problemlerine çözüm bulmak üzere genetik algoritmaları ilk kullanan arařtırmacıdır. İki boyutlu dikdörtgen paketleme problemleri için genetik algoritmaları kullanmıřtır. Aynı zamanda Davis (1985) çalıřmasında iki boyutlu bir paketleme problemi örneđi kullanarak genetik algoritmalarda uygulanan teknikleri özetlemiřtir. Son 10 yıl boyunca iki veya daha çok boyutlu, düzgün veya düzgün biçimli olmayan řekiller içeren çeřitli paketleme problemleri ele alınmıřtır. Karmařık problemler çođunlukla kırma genetik algoritma adı verilen 2 ařamalı bir iřlemlerle ele alınır. Öncelikle genetik algoritmalar yerleřim planında en az fire veren, parça yerleřim sırasını bulur. Bunun için genetik algoritma kodlanmıř çözümler kullanır. İkinci olarak bu kodlanmıř çözümler kod çözücü bir yerleřtirme algoritması kullanılarak yerleřim planı oluřturmak üzere deđerlendirilir. Alan bilgisi kod çözücü yerleřtirme algoritması içerisine kurulduđundan arama uzayının boyutu küçültülebilir. Örneđin paketleme stratejisi sadece çakıřma olmayan bir biçimde yerleřim planları üretmek üzerine kurulabilir. Böylelikle arama uzayı sınırlandırılmıř olur.



### 2.5.1.1. Genetik algoritmalarda ele alınan paketleme problemlerinin sınıflandırılması

Günümüze kadar paketleme problemlerinin çözümü üzerine yapılan pek çok çalışmada genetik algoritmalar kullanılarak çeşitli çözüm yaklaşımları araştırılmıştır. Literatürde yer alan problemler çoğunlukla dikdörtgen biçimli şekilleri ve 2 boyutlu şerit paketleme görevlerini içeren problemlerdir. Karmaşıklık ve genetik algoritmanın uygulanışı problem tipine göre değişmektedir. Geometrik özellikler problemin karmaşıklığını ve arama uzayının boyutunu etkilediğinden paketleme görevlerinin çeşitliliği de geometrik özelliklerine göre ayırt edilebilmektedir. Problemin boyutsal büyüklükleri de ayrıca çok önemli bir kriterdir. Bu iki karakteristik temel alınarak paketleme problemlerine ait çözüm yaklaşımlarının kalitesini karşılaştırmak için literatürde genetik algoritmalarla ele alınan paketleme problemleri sınıflandırılmıştır. Şekil 2.6'da bu sınıflandırma gösterilmektedir (Hopper and Turton 1997).

Problemler öncelikle 2 boyutlu ve 3 boyutlu olmak üzere boyutsal büyüklüklerine göre, daha sonra da düzgün biçimli ve düzgün biçimli olmayan paketleme problemleri olmak üzere geometrik özelliklerine göre gruplandırılmıştır. Düzgün biçimli şekiller terimi biçimleri birkaç parametre ile belirlenen dikdörtgen ve daire gibi şekiller için kullanılırken, düzgün biçimli olmayan şekiller terimi nispeten düzensiz, keyfi biçimli dışbükey ve içbükey çokgen gibi şekiller için kullanılır (Dyckhoff 1990).



**Şekil 2.6.** Genetik algoritmalarla ele alınan paketleme problemlerinin sınıflandırılması.

### 2.5.1.2. İki boyutlu düzenli şerit giyotinsiz kesme ve paketleme problemlerinde kırma yaklaşımlar

Düzenli biçimli kesme ve paketleme problemleri çoğunlukla yüksekliği sınırsız kabul edilen dikdörtgen biçimindeki ana parça içerisine çok sayıda küçük dikdörtgen parçanın yerleştirilmesiyle ilgilenen problemlerdir. Genel olarak düzenli biçimli dikdörtgen parçaların yerleştirilmesi, giyotinli ve giyotinsiz yerleşim planlarını içeren kesme ve paketleme problemleri olmak üzere ikiye ayrılır. Bütün bu problemler için amaçlanan tek şey ana parçanın yüksekliğini en aza indiren parçaların yerleşim planını bulmaktır.

Çok sayıda araştırmacı kesme ve paketleme problemlerinin çözümü için yaptıkları çalışmalarda genetik algoritmaları kullanarak, iki boyutlu giyotinsiz düzenli şerit paketleme problemlerini ele almıştır. Bu yöntemlerin pek çoğu bir yerleştirme algoritmasıyla genetik algoritmanın birleşiminden oluşan kırma yaklaşımlardır. Bu iki aşamalı yaklaşımda genetik algoritma, bir kod çözücü yerleştirme algoritmasının yardımıyla paketlenen küçük parçaların yerleşim planı içindeki sırasını bulur.

Smith (1985) iki ayrı kod çözücü yerleştirme algoritması ile deneysel çalışmalarını yürüterek sıralamaya dayalı bir yaklaşım geliştirmiştir. Bu kod çözücü yerleştirme algoritmalarından ilki olan kayma (slide) algoritması, sabit bir konumdan dikey zikzak hareketlerle indiği en uzak köşeye dikdörtgen parçanın yerleştirilmesini sağlar. İkinci kod çözücü yerleştirme algoritması olan ufuk çizgisi (skyline) algoritması kısmi yerleşim planı içerisinde bütün konumları dener. Bu iki algoritma arasında yapılan karşılaştırmalar genetik algoritmanın daha karmaşık bir algoritma ile birleştirilmesinin daha iyi yerleşim planları ürettiğini göstermektedir. Ancak bu çok büyük bir hesap yükü gerektirir. Bu çalışmadaki genetik algoritmaların performansı dinamik programlama üzerine kurulmuş paketleme yöntemleriyle karşılaştırılmış ve genetik algoritmaların aynı paketleme yoğunluğunu daha az zamanda elde ettiği görülmüştür.

Jakobs (1996) sıralamaya dayanan bir genetik algoritmayı aşağı-sol algoritması ile birlikte kullanmıştır. Hesapsal yükün azaltılması için, bu algoritma yerleşim planı içinde mutlaka mümkün olan en düşük konumu bulmak için arama yapmaz. Ancak yerleşim planı içinde her zaman aşağı sol koşulunu sağlamak için çalışır. Her parça ana parçanın sağ üst köşesinden başlayarak kısmi yerleşim planı içinde mümkün olduğunca aşağıya ve sola hareket eder. Genetik algoritmalarda başlangıç nüfusu parçaların yerleşim planı içindeki yerleşim sırasını gösteren sayı dizilerinden oluşturulur. Kıрма genetik algoritmalar değiştirilmiş yerleşim kuralları kullanılarak çokgenler için de kullanılabilir.

Dagli and Poshyanonda (1997)'nin çalışmalarında kullandıkları kod çözücü yerleştirme algoritması bir kaydırma tekniğine dayalı, yapay sinir ağlarıyla birleştirilmiş bir yerleştirme algoritmasıdır. Genetik algoritmaları bu kod çözücü yerleştirme algoritmasında kullanmak üzere yerleşim sırasını bulmak için uygulamışlardır. Burada kullanılan kaydırma tekniği yeni parçayı ana parçanın genişliği boyunca, kısmi yerleşim planını oluşturan diğer parçaların yanına yerleştirir. Eğer kısmi yerleşim planı içindeki boş alan yeterli değilse, yeni parçaları yerleştirmek üzere yerleşim planında yeni bir sıra oluşturulur. Paketleme süresi boyunca yeni oluşan boş alanlar kaydedilir ve daha sonra yapılacak parça yerleşimlerinde kullanılmak üzere saklanır. Ana parça içine yeni bir parça

yerleştirilmeden önce yerleştirilecek yeni parça ve boş alanlar arasında en iyi eşleşmenin seçimini yapmak üzere mevcut boş alanlar bir yapay sinir ağıyla kontrol edilir. Uygun bir eşleşme bulunamazsa yeni parça, kaydırma tekniğine dayanan yerleştirme algoritmasıyla yerleştirilir. Eşleşme işlemi, kısmi yerleşim planı içinde yerleştirilecek yeni parçanın mümkün olan bütün yönlendirmelerini dener.

Liu and Teng (1999) tarafından gerçekleştirilen çalışmada Jakobs (1996) tarafından uygulanan kod çözücü yerleştirme algoritmasının geliştirilmesi amaçlanmıştır. Geliştirilmiş aşağı-sol algoritması Jakobs (1996)'nın kullandığı kaydırma ilkesine göre çalışır. Ancak öncelik her zaman yerleştirilecek parçanın aşağı doğru kaydırılmasına verilmiştir. Liu ve Teng çalışmalarında, Jakobs (1996)'un kullandığı iki paketleme problemini kullanarak yeni aşağı sol yerleştirme algoritmasının performansının daha iyi olduğunu göstermiştir.

Hopper and Turton (1997, 1999) çalışmalarında yerleşim planı içerisinde aşağı sol koşulunu sağlayan bir yerleştirme algoritması kullanmıştır. Üzerinde çalıştıkları geliştirilmiş aşağı sol algoritması kısmi yerleşim planı içerisinde etrafı parçalarla çevrili kullanılmamış boş alanlara ulaşır ve yeterli boş alanın elde edildiği ilk aşağı sol konumuna yeni parçayı yerleştirir. Bu yerleştirme algoritması genetik algoritma ve benzetilmiş tavlama algoritması ile birleştirilmiştir. Benzetilmiş tavlama algoritması çoğunlukla daha yoğun yerleşim planları elde etmiş, fakat daha uzun çalışma zamanları gerektirmiştir.

Leung et al. (1999, 2001), fark yöntemi (FY) algoritması adıyla kısmi yerleşim planı içerisinde hesaplamalar yapabilen bir aşağı sol algoritması geliştirmiştir. Bu algoritma tarafından yerleşim planına her yeni parça eklenirken kısmi yerleşim planının üst ve sağ tarafında boş dikdörtgen biçimde iki uzay oluşturulur. Bununla birlikte bir sonraki parçanın yerleştirilmesi için yerleştirmenin yapılabileceği dikdörtgen biçimindeki bu uzayın aşağı sol köşesine en yakın koordinatlar saklanır ve her yeni parça yerleştirilirken kullanılır. Jakobs (1996)'nın ve Liu ve Teng (1999)'un kaydırma tekniğine dayalı yerleştirme algoritmalarıyla karşılaştırıldığında FY algoritması yerleşim planında etrafı çevrili kullanılmamış boş alanlara da parça yerleştirebildiğinden daha iyi sonuçlar üretmektedir.

Tablo 2.2’de iki boyutlu giyotinsiz düzgün biçimli paketleme problemleri için kırma genetik algoritmalar kullanarak yapılmış literatürde yer alan çalışmalar gösterilmektedir.



**Tablo 2.2.** İki boyutlu giyotinsiz düzgün biçimli paketleme problemleri için kırma genetik algoritmalar kullanarak literatürde yer alan çalışmaları.

	Smith (1985)	Jakobs (1996)	Dagli and Poshyanonda (1997)	Liu and Teng (1999)	Hopper and Turton (1997, 1999)	Leung et al.(1999)
<b>Problem</b>	Kutu paketleme, 90° dönme.	Şerit paketleme, 90° dönme.	Şerit paketleme, 90° dönme	Şerit paketleme, 90° dönme	Şerit paketleme, dönme	Şerit paketleme, dönme yok.
<b>Amaç</b>	Kutuya yerleşen parça sayısını arttırmak.	Ana parçanın yüksekliğini en aza indirmek.	Ana parçanın yüksekliğini en aza indirmek.	Ana parçanın yüksekliğini en aza indirmek.	Ana parçanın yüksekliğini en aza indirmek.	Fire miktarını en aza indirmek.
<b>Gösterim</b>	Permutasyon	Permutasyon	Permutasyon	Permutasyon	Permutasyon	Permutasyon
<b>Uygunluk</b>	Paketlenmemiş alanın paketlenen alana oranı.	Kalan alan ve yükseklik.	Yükseklik ve genişlik.	Kalan alan ve yükseklik.	Kalan alan ve yükseklik.	Fire.
<b>Çaprazlama</b>	OX (2 noktada)	OX (1 noktada)	OX	OX (1 noktada)	PMX (2 noktada)	PMX, CX, OBX,OX (1, 2 noktada)
<b>Mutasyon</b>	Dizinin yeniden rasgele sıralanması, devirli mutasyon.	Tersleme, 2 elemanın yer değişimi, devirli mutasyon.	Tersleme.	Tersleme, 2 elemanın yer değişimi, devirli mutasyon.	2 elemanın yer değişimi.	2 elemanın yer değişimi
<b>Kod çözücü</b>	Slide(kayma) ve Skyline (ufuk çizgisi) algoritması	AS algoritması	Kaydırma algoritması ve parçalar ile boş alanların eşleştirilmesi için yapay sinir ağları.	Geliştirilmiş AS algoritması	ASD algoritması.	FY algoritması

## **2.5.2. Paketleme problemleri için benzetilmiş tavlama algoritmasının uygulanışı**

Benzetilmiş tavlama metallerdeki tavlama sürecinden esinlenerek tasarlanmış ve ilk kez Kirkpatrick (1983) tarafından kombinyonel eniyileme problemlerinin çözümü için geliştirilmiş bir algoritmadır. Benzetilmiş tavlama algoritması düzgün biçimli ve düzgün biçimli olmayan kesme ve paketleme görevlerinin gerçekleştirilmesi için uygulanmıştır.

### **2.5.2.1. İki boyutlu giyotinli ve giyotinsiz kesme ve paketleme problemlerinde yaklaşımlar**

Dikdörtgen biçimindeki paketleme problemlerinin çözümü için çok az sayıda araştırmacı tarafından benzetilmiş tavlama algoritması kullanılmıştır. Bu algoritmayı kullanarak paketleme problemleri üzerine çalışan ilk araştırmacılardan biri Kämpke (1988)'dir. Benzetilmiş tavlama yöntemini farklı soğutma stratejileri kullanarak bir boyutlu kutu paketleme problemlerine uygulamıştır.

Dowsland (1993) benzetilmiş tavlama algoritması kullanarak özdeş ve özdeş olmayan kutuları içeren yükleme problemleri üzerinde çalışmıştır. Özdeş kutular içeren paketleme problemlerinde olası yerleşim konumlarının sayısını problem çözümündeki etkinliği arttırmak için azaltmıştır. Benzetilmiş tavlama algoritmasında komşuluk, bazı sınırlamalar kullanarak arama uzayında her bir parçanın bir konumdan başka bir konuma hareket etmesiyle elde edilen olası çözümler seti olarak tanımlanmıştır. Bu yer değiştirme hareketleri esnasında çakışan yerleşim planlarıyla karşılaşılabileceğinden çakışma kısıtlaması amaç fonksiyonda yer almıştır.

Faina (1999) giyotinli ve giyotinsiz kesme problemleri için bir kırma benzetilmiş tavlama algoritması geliştirmiştir. Çalışmasında paketlenen parçalar, paketleme sırasını temsil eden bir permütasyonla gösterilmiştir. Giyotin kısıtlaması da göz önüne alınarak kod çözücü yerleştirme algoritması küçük parçaları, ana parça içine yerleştirmek için kullanılmıştır. Giyotinsiz yerleşim planları için algoritma önceden yerleştirilmiş dikdörtgen parçanın ya sol üst yada sağ alt köşesini seçerek sıradaki parçayı yerleştirir. Bu seçim rasgele yapılır. Giyotinli yerleşim planlarında ise



algoritma yerleşim planında kullanılmayan boş alanların koordinatlarını saklar. Bir dikdörtgen parçanın yerleştirilmesinden sonra parçanın üstünde ve sağ yanında iki boş alan oluşturulur. Bu boş alanlar bir sonraki paketleme işleminde ana parça gibi ele alınmaktadır.

Leung et al. (1999, 2001) benzetilmiş tavlama için genetik algoritma ile de birlikte kullanılan sıralamaya dayalı bir yaklaşım uygulamıştır. Elde edilen sonuçlara göre genetik algoritmalar benzetilmiş tavlama algoritmasına göre daha iyi bir performans göstermektedir.

Tablo 2.3’de iki boyutlu dikdörtgen biçimindeki paketleme problemleri için benzetilmiş tavlama yaklaşımı kullanarak literatürde yer alan çalışmalar yer almaktadır.

**Tablo 2.3.**İki boyutlu dikdörtgen biçimindeki paketleme problemleri için benzetilmiş tavlama yaklaşımlarının karşılaştırılması.

	Dowland (1993)	Faina (1999)	Leung et al. (1999)
<b>Problem</b>	Özdeş ve özdeş olmayan kutularla yükleme problemi, 90° dönme.	Giyotinli ve giyotinsiz şerit paketleme, dönme yok.	Şerit paketleme, dönme yok.
<b>Amaç</b>	Sabit sayıda kutunun olası yerleşim planının bulunması.	Kullanılan alanı en aza indirme.	Fire miktarını en aza indirme.
<b>Gösterim</b>	Çakışmaya izin veren yerleşim planında parçaların olası konumları.	Permütasyon.	Permütasyon.
<b>Uygunluk</b>	Çakışan kutuların sayısını en aza indirme.	Paketleme yoğunluğu.	Yükseklik.
<b>Komşuluk hareketi</b>	Paketlenmek suretiyle yerleştirilmiş kutuların konum seti.	2 elemanın yer değiştirmesi	2 elemanın yer değiştirmesi.
<b>Soğutma planı</b>	Geometrik.	Geometrik.	Geometrik.
<b>Kod çözücü</b>	Yok.	Giyotin kısıtlaması ile gerçekleştirilen sola dayalı algoritma.	FY algoritması.



## **BÖLÜM 3. YERLEŞTİRME ALGORİTMALARI**

Yerleştirme algoritmaları, en iyi yerleşim planı veya en az fireyi veren, parça yerleşim sırası olarak da bilinen permütasyonu kullanarak, küçük parçaların ana parça içinde yerleşimini sağlar.

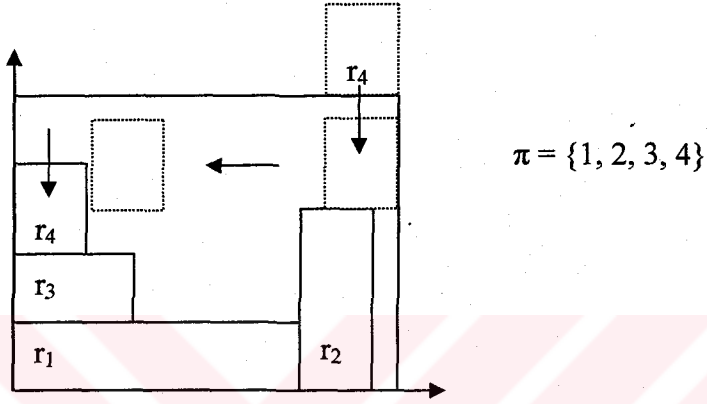
Kesme ve paketleme problemleri için arama uzayı sınırsız olduğundan, yerleşim planı içerisinde bir parçanın olası her hareketi yeni bir yerleşim planı oluşturur. Olası yerleşim planı sayısını verimli bir şekilde azaltmak için aşağı sol (AS) koşulunu gerçekleştiren, AS algoritması kullanılır. AS koşulu, bir yerleşim planı içinde yerleşecek parçaların, ana parça üzerinde daha aşağıya ve sola hareket edemediğinde sağlanmaktadır (Jakobs 1996). Bu bölümde, yerleştirme algoritmaları olarak bilinen 3 AS algoritmasından söz edilmektedir.

Literatürde tanımlı 3 AS algoritması vardır. Bunlardan ilk ikisi kırma yerleştirme algoritmalarında kullanılmıştır. İlk algoritma, kırma GA ile Jakobs (1996) tarafından kullanılan AS algoritmasıdır. İkincisi Liu ve Teng (1999)'in yine GA ile birlikte kullandığı geliştirilmiş AS algoritmasıdır. AS ve geliştirilmiş AS algoritmaları kaydırma tekniği üzerine kurulmuşlardır. Bu çalışmanın tamamında geliştirilmiş AS algoritması kullanılmıştır. Üçüncü AS algoritması, Chazelle (1983) tarafından kullanılmış diğer iki AS algoritmasına göre farklı bir yerleştirme tekniği kullanan aşağı sol dolgu (ASD) algoritmasıdır. Daha karmaşıktır ve bu nedenle hesaplanması daha çok zaman alır.

### **3.1. Aşağı Sol Algoritması**

AS algoritması Jakobs (1996) tarafından kırma GA'da kullanılmıştır. Permütasyonla verilen küçük dikdörtgen parçaların yerleştirme sırasına göre öncelikle ilk parça, ana parçanın en alt sol köşesine yerleştirilir. Daha sonra permütasyondaki sıraya göre

parçalar ana parçanın en üst sağ köşesinden başlayarak mümkün olabildiğince aşağıya ve sonra mümkün olabildiğince sola kaydırılırlar (Jakobs 1996). Bu birbirini izleyen yatay ve dikey kaydırma hareketleri yerleşecek küçük parça sabit bir konumda kalana dek devam eder. Sabit konum, küçük parçanın en alt ve sol kenarı başka bir parça ile çakıştığıında bulunur. Şekil 3.1’de AS algoritması bir örnekle gösterilmiştir.

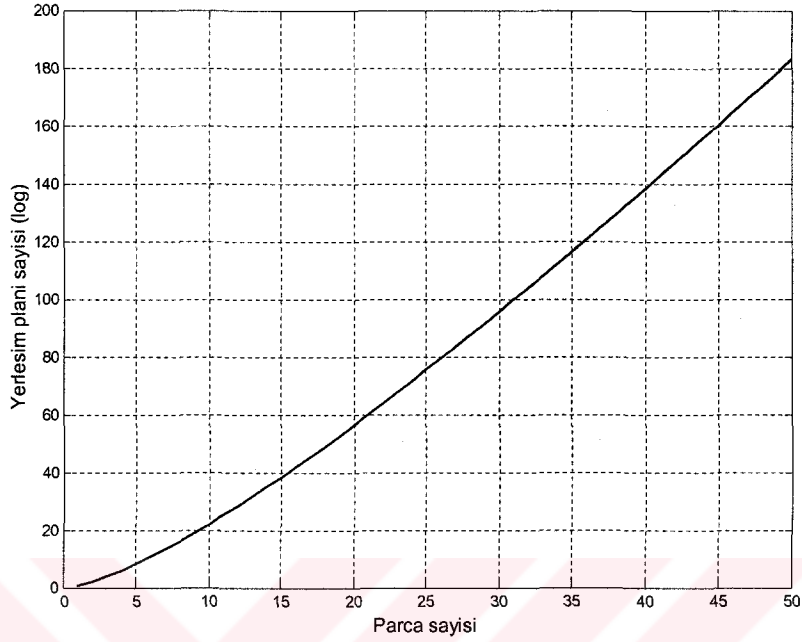


Şekil 3.1. AS algoritmasının gösterimi.

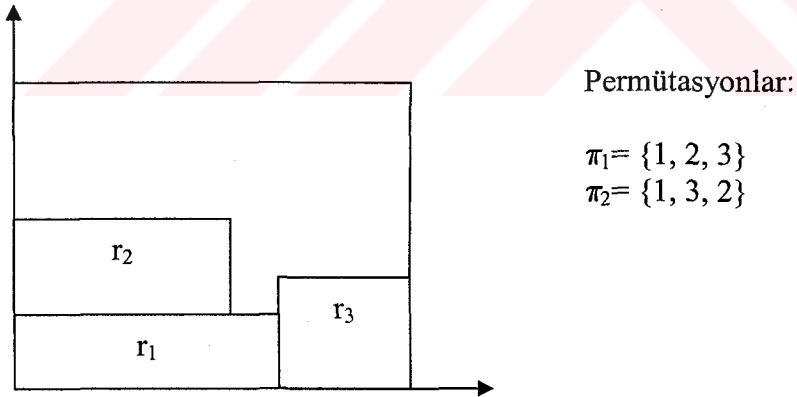
Aşağı sol algoritmasının özellikleri:

1. Yerleştirilecek  $n$  adet parça için AS algoritması ile hesaplanan yerleşim planlarının sayısı en çok  $2^n \cdot n!$ 'dir. Bu da yerleştirme problemlerinin birer permütasyon problemi olduğunu gösterir. Yerleştirilecek parça sayısının en çok 50 adet olduğunu varsaydıığımızda artan parça sayısına göre elde edilen yerleşim planı sayısını gösteren eğilim Şekil 3.2’de görülmektedir. Parça sayısı arttıkça, olası yerleşim planı sayısı da üssel olarak artış göstermektedir. Pratikte, AS algoritması tarafından oluşturulan  $2^n \cdot n!$  yerleşim planından daha az yerleşim planı vardır. Bunun nedeni Şekil 3.3’te de görüldüğü gibi iki ayrı permütasyonun aynı yerleşim planını gösterebilmesidir (Callaghan et al. 1999). Şekil 3.3’de  $\pi_1$  permütasyonu kullanılarak öncelikle  $r_1$  parçası ve permütasyonda onu izleyen  $r_2$  parçası yerleştirilmiştir.  $r_2$  parçasının genişliği en alttaki mevcut boşluktan daha büyük olduğundan sol taraftaki  $r_1$  parçasının üzerine yerleştirilmiştir. Sonrasında sıradaki 3.parça da mevcut boşluğa yerleştirilmiştir. Benzer şekilde  $\pi_2$  permütasyonu kullanılarak önce  $r_1$  parçası ve permütasyonda onu izleyen  $r_3$

parçası yerleştirilmiştir. Bu defa  $r_3$  parçası doğrudan en alta yerleştirilmiştir. Daha sonra  $r_2$  parçası,  $r_1$  parçasının üzerine yerleştirilmiştir.



Şekil 3.2. Artan parça sayısına göre elde edilen yerleşim planı sayısı.



Şekil 3.3. İki ayrı permütasyona sahip yerleşim planı.

2. Arama uzayının büyüklüğü seyahat eden satıcı probleminde kullanılan arama uzayından daha büyüktür. Örneğin, yerleşecek parça sayısı 25 olduğunda,

$$2^{25} \cdot 25! \geq 10^{75}$$

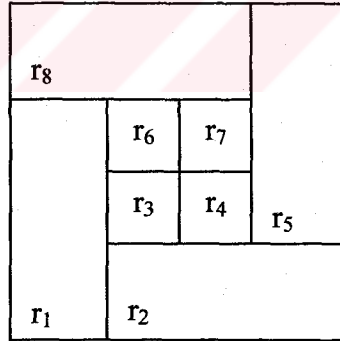
$$(25/e)^{25} \geq 10^{75}$$

$$10^{24} \geq 10^{31}$$

adet yerleşim planı elde edilir (Jakobs 1996).

3. AS algoritmasının hesapsal yükü  $O(n^2)$ 'dir. Buradan yola çıkarak, her bir parça ( $r_i$ ) en çok  $i$  kez yer değiştirebilir. Çünkü her bir yer değiştirme ( $i-1$ .) yerleştirilmiş dikdörtgen parçanın köşeleriyle yada ana parçanın köşeleriyle sınırlandırılmaktadır. Bundan dolayı yerleşen dikdörtgen parçanın ( $r_i$ ) hesapsal yükü  $O(i)$  ve yerleşecek parça sayısı  $n$  adet alındığında tüm hesapsal yük miktarı  $O(n^2)$  olur (Jakobs 1996, Liu and Teng 1999).
4. AS algoritmasının en büyük dezavantajı, yerleşim planında daha önceden yerleşmiş olan ana parçaların bir sonraki yerleşecek parçanın hareketini engellediğinde boş alanlar oluşturmasıdır (Hopper 2000).

AS koşulunu gerçekleştiren  $n$  adet dikdörtgen parçanın en iyi yerleşim planı bilindiği halde AS algoritmasıyla bunun için bir permütasyon yazılamayabilir. Diğer bir deyişle, bazen AS algoritmasıyla en iyi yerleşim planı elde edilemeyebilir. Şekil 3.4'te böyle bir yerleşim planına örnek verilmiştir. Burada AS algoritması kullanmak yerine, hem AS koşulunu gerçekleştiren hem de bilinen en iyi yerleşim planını sağlayan geliştirilmiş AS algoritmasını kullanabiliriz (Liu and Teng 1999).

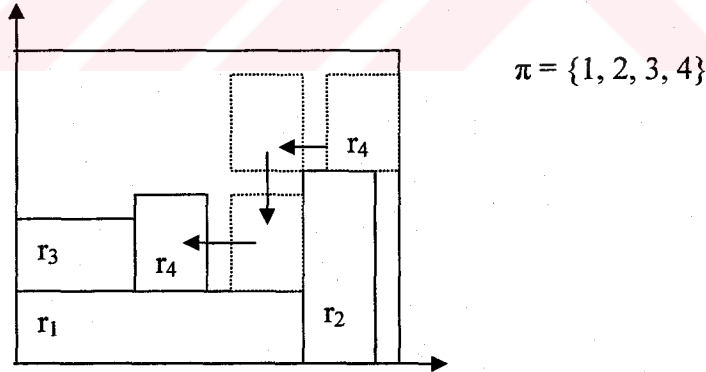


**Şekil 3.4.** Sekiz dikdörtgen parçanın en iyi yerleşim planı.

### 3.2. Geliştirilmiş Aşağı Sol Algoritması

Liu ve Teng giyotinsiz bir yerleştirme probleminin çözümü için kırma bir GA kullanmışlardır (Liu and Teng 1999). Parçaların yerleşim planı içinde yerleştirilmesi için de kaydırma tekniği üzerine kurulu bir AS algoritması geliştirmişlerdir. Bu yeni AS algoritması, Jakobs'un algoritmasının performansını arttırdığını, yine Jakobs'un

kullandığı 2 test problemiyle göstermektedir. Geliştirilmiş AS algoritması da aynı AS algoritmasında olduğu gibi yerleşecek parçanın ana parçanın sağ üst köşesine yerleştirilmesiyle başlar. Öncelikle parça mümkün olduğunca aşağıya doğru sonra da altında kalan parçanın üst kenarları boyunca sola doğru bir köşeyle karşılaşınca kadar hareket ettirilir. Bir köşeyle karşılaşıldığında parça yine aşağıya doğru hareket ettirilir. Sonra eğer mümkünse parça yine sola doğru hareket ettirilir. Bu işlemler aşağıya hareket öncelikli olmak üzere aşağıya ve sola doğru bir parça yada kenarla karşılaşınca kadar devam eder (Liu and Teng 1999). Parça sabit konumuna çakışma olmaksızın aşağıya ve sola doğru hareketi artık gerçekleştiremez konuma geldiğinde ulaşır. Parçanın tamamıyla ana parça içerisine yerleştirilemediği durumlarda, parça çıkartılır diğer bir deyişle yerleştirilmez (Leung et al. 2001). Şekil 3.5’de geliştirilmiş AS algoritmasının parça yerleştirilirken nasıl çalıştığı bir örnek üzerinde gösterilmektedir. Şekil 3.1 ve Şekil 3.5’teki yerleşim planları göz önüne alındığında görülmektedir ki AS koşulunu gerçekleştirdikleri ve yerleştirme sırasını gösteren permütasyonları aynı olduğu halde, AS ve geliştirilmiş AS algoritmaları tarafından elde edilmiş yerleşim planları aynı değildir (Liu and Teng 1999). Burada görüldüğü gibi geliştirilmiş AS algoritmasının kullanılmasıyla daha iyi bir yerleşim planı elde edilmiştir.



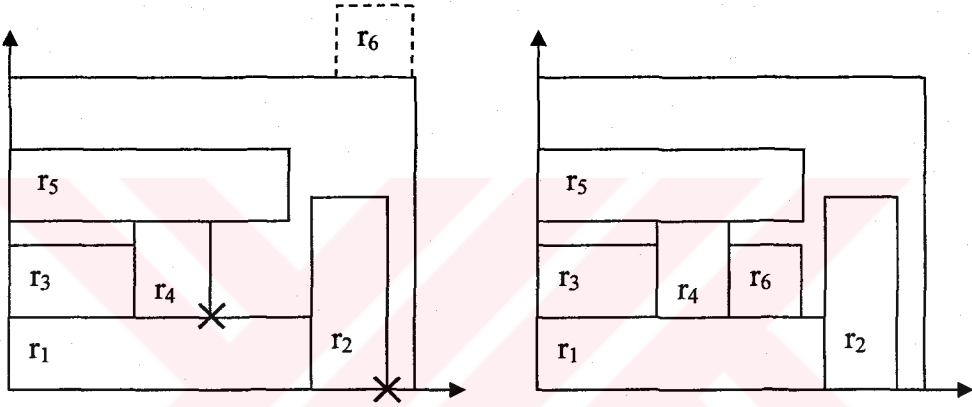
Şekil 3.5. Geliştirilmiş AS algoritmasının gösterimi.

### 3.3. Aşağı Sol Dolgu Algoritması

Bölüm 3.1 ve 3.2’de anlatılan AS ve geliştirilmiş AS algoritmaları kaydırma tekniği kullanarak oldukça geniş boş alanlarda yerleşim planları oluştururlar. Bunların tam tersine ASD algoritması, çoğunlukla yerleşim planında varolan boşlukları

doldurmaya eğilimlidir. ASD algoritması, yerleştirilecek küçük parçanın kısmi yerleşim planı içerisinde mümkün olan en alt seviyedeki boş alana yerleştirilmesiyle gerçekleştirilir (Chazelle 1983). Şekil 3.6’da, geliştirilmiş AS algoritması için verilen örneğe ait permütasyon kullanılarak ASD algoritması ile gerçekleştirilen bir yerleşim planı görülmektedir.

AS ve geliştirilmiş AS algoritmalarıyla karşılaştırıldığında, ASD algoritması daha yoğun bir yerleşim planı oluşturmaktadır (Hopper 2000). Ancak en büyük dezavantajı  $O(n^3)$  olan hesapsal yüküdür (Chazelle 1983).



Şekil 3.6. ASD algoritmasının gösterimi.

## BÖLÜM 4. GENETİK ALGORİTMALAR ve BENZETİLMİŞ TAVLAMA

### 4.1. Genetik Algoritmalar

Çok büyük arama uzayı içinde gerekirci liste yöntemleri kullanımı, hemen hemen imkansızdır. Çünkü bu büyük arama uzayı içinde en iyi çözümlerin bulunması çok zaman alır. Diğer taraftan olasılıksal yaklaşımlar en iyi çözümün en kısa zamanda üretimini sağladıklarından, kesme problemleri gibi matematiksel bir modelle gösterilemeyen kombinasyonel problemlerin eniyileme uygulamalarında tercih edilir. Tüm eniyileme problemleri sürekli ve/veya kesikli arama yüzeyleri üzerinde oluşur. Bu tip problemlerde doğrusal olmayan programlama kullanmak, işlem hacminin artması ve türev ihtiyacının gerekliliği nedeniyle uygun değildir. Genetik algoritmalar, bu tip sorunlara en iyi çözümü veya çözümleri gerçek zamanda en kısa sürede sağlar (Bingül et al. 2000).

Genetik algoritmalar (GA) doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında “en iyinin hayatta kalma” ilkesine göre bütünsel en iyi çözümü arar.

Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesinde John Holland tarafından ortaya atılmıştır. Daha sonra Holland (1975) yaptığı çalışmaları “Adaptation in Natural and Artificial Systems” adlı kitabında bir araya getirmiştir. İlk olarak Holland evrim yasalarını genetik algoritmalar içinde eniyileme problemleri için kullanmıştır (Goldberg 1989). Bu algoritmalar daha sonra, eniyileme problemleri için çok önemli bir araç haline gelmiştir. Günümüzde de genetik algoritmalar pek çok değişik uygulamada kullanılmaktadır. Bunun nedeni, genetik algoritmaların problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretmesidir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir.

Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür (Bingül et al. 1999, 2000).

Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde nüfus adını alır. Nüfuslar vektör, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenirler. Problemin bireyler içindeki gösterimi problemden probleme değişiklik gösterir. Genetik algoritmaların problemin çözümündeki başarısına karar vermedeki en önemli faktör, problemin çözümünü temsil eden bireylerin gösterimidir. Nüfus içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonundan dönen değere göre yüksek değere sahip olan bireylere, nüfustaki diğer bireyler ile çoğalmaları için fırsat verilir. Bu bireyler çaprazlama işlemi sonunda çocuk adı verilen yeni bireyler üretirler. Çocuk kendisini meydana getiren ebeveynlerin (anne, baba) özelliklerini taşır. Yeni bireyler üretilirken düşük uygunluk değerine sahip bireyler daha az seçileceğinden bu bireyler bir süre sonra nüfus dışında bırakılırlar. Yeni nüfus, bir önceki nüfusta yer alan uygunluğu yüksek bireylerin bir araya gelip çoğalmalarıyla oluşur. Aynı zamanda bu nüfus önceki nüfusun uygunluğu yüksek bireylerinin sahip olduğu özelliklerin büyük bir kısmını içerir. Böylelikle, pek çok nesil aracılığıyla iyi özellikler nüfus içersinde yayılırlar ve genetik işlemler aracılığıyla da diğer iyi özelliklerle birleşirler. Uygunluk değeri yüksek olan ne kadar çok birey bir araya gelip, yeni bireyler oluşturursa arama uzayı içerisinde o kadar iyi bir çalışma alanı elde edilir. Probleme ait en iyi çözümün bulunabilmesi için;

- Bireylerin gösterimi doğru bir şekilde yapılmalı,
- Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
- Doğru genetik işlemciler seçilmeli.

Bu durumda çözüm kümesi problem için bir noktada birleşecektir.



Genetik algoritmalar, diğ er eniyileme yöntemleri kullanılırken büyük zorluklarla karşılaş ılan, oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedir. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler. Ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler bulurlar. Genetik algoritmaların asıl amacı, hiçbir çözüm tekniğ i bulunmayan problemlere çözüm aramaktır. Kendilerine has çözüm teknikleri olan özel problemlerin çözümü için mutlak sonucun hız ı ve kesinliğ i açısından genetik algoritmalar kullanılmazlar. Genetik algoritmalar ancak;

- Arama uzayının büyük ve karmaş ık olduğ u,
- Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğ u,
- Problemin belirli bir matematiksel modelle ifade edilemediğ i,
- Geleneksel eniyileme yöntemlerinden istenen sonucun alınmadığ ı

alanlarda etkili ve kullanış ılıdır (Beasley et al. 1993a).

Genetik algoritmalar parametre ve sistem tanı lama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanı ma, mühendislik tasarımları, planlama, yapay zeka uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel eniyileme problemleri ağ tasarımı problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğ er eniyileme yöntemlerinin yanında başarılı sonuçlar vermektedir (Man et al. 1996, Beasley et al. 1993a, Gen et al. 2001, Drezner and Wesolowsky 2003).

#### **4.2. Genetik Algoritmaların Diğ er Arama ve Eniyileme Yöntemlerinden Fark ı**

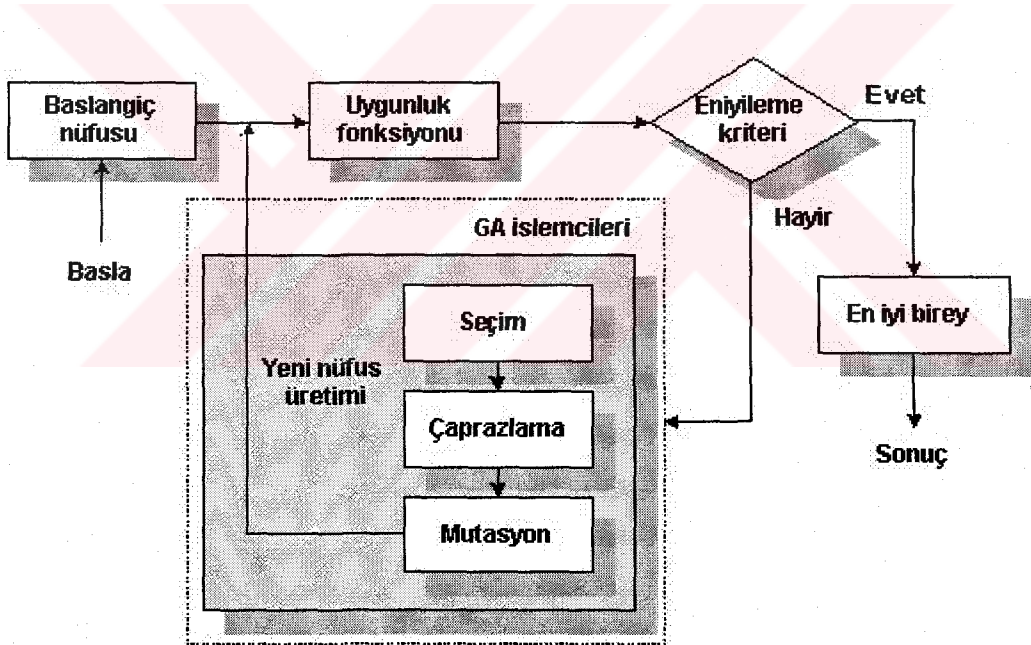
Genetik algoritmaların geleneksel arama ve eniyileme yöntemlerinden dört temel fark ı vardır (Goldberg 1989):

1. Genetik algoritmalar problemlerin çözümünü parametrelerin değ erleriyle değ il, kodlarıyla arar. Parametreler kodlanabildiğ i sürece çözüm üretilebilir. Bu sebeple genetik algoritmalar ne yaptığ ı konusunda bilgi içermez, nasıl yaptığ ını bilir.

2. Genetik algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.
3. Genetik algoritmalar türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değer kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.
4. Genetik algoritmalar gerekirci kuralları değil olasılıksal kuralları kullanır.

#### 4.3. Temel Bir Genetik Algoritma Çalışması

Genetik algoritmalar, çözüm aranan problemin parametreleri kodlanabildiği sürece bir çözüm kümesi, probleme özgü tanımlanmış bir uygunluk fonksiyonu ve genetik algoritma işlemcileriyle problem için verilen eniyileme kriteri sağlanıncaya kadar en iyi çözümü arar. Bu işlemlerin sıralandığı temel bir genetik algoritma çalışması Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Temel bir genetik algoritma çalışması.

Eniyileme işlemi öncelikle problem için olası pek çok çözümü içeren rasgele bireylerden oluşmuş bir başlangıç nüfusunun oluşturulmasıyla başlar. Nüfus büyüklüğü probleme göre değişen bir parametredir. Genetik algoritma için her probleme özgü kurulan uygunluk fonksiyonuyla bireyler problemin parametreleri haline getirilir ve uygunluk değerleri hesaplanır. Bireylerin uygunluk değerlerine

bakılarak herhangi bir bireyin problem için çözüm olup olmayacağına karar verilir. Başlangıç nüfusunun uygunluk değerleri için eniyileme kriteri karşılanmıyorsa genetik algoritma işlemcileri ile yeni bireylerin üretimi yapılır. Eğer eniyileme kriteri karşılanıyorsa problemin çözümü olan birey elde edilmiş demektir. Eniyileme kriteri bir uygunluk değeri olabileceği gibi, önceden belirlenmiş bir döngü sayısı veya genetik algoritmalarındaki adıyla nesil de olabilir. Yeni nüfusun üretimi için kullanılacak bireyler, bir önceki nüfusta bulunan bireyler arasından seçim tekniklerinden biri kullanılarak uygunluk değerlerine göre belirlenir ve üreme havuzuna atılır. Yeni nüfusu üretmek üzere havuzdaki bu bireyler sırasıyla çaprazlama ve mutasyon işlemlerine tabi tutulurlar. Genetik algoritmalarda bu işlemlerin kullanılmasının nedeni, nüfus içindeki çeşitliliği sağlamak ve belirli bir yönde yığılmayı önlemektir. Yeni elde edilen nüfusun uygunluk değerlerinin bir önceki nüfusun uygunluk değerlerinden daha iyi olması beklenir. Bu şekilde eniyileme kriteri karşılanıncaya kadar, diğer bir deyişle çözüme ulaşıncaya kadar nüfus üretimi ve değerlendirme işlemleri devam eder.

#### **4.4. Gösterim**

Bilgisayar biliminin en önemli sorunlarından biri gerçek dünya problemlerini bilgisayar ortamına aktarmaktır. Mutlak çözümün elde edilebilmesi için problemin gösteriminin en doğru şekliyle yapılması gerekir. Bu da her problem için ayrı bir gösterim demektir. Genetik algoritmalarda da çözüm aranan problemin gösterimi problemde değişiklik göstermektedir. Bireylerin gösterimi, genetik algoritmaların performansını etkileyen ve problemin çözümündeki başarısına karar veren en önemli faktörlerden biridir. Farklı gösterimler problemin çalışma zamanı ve mutlak çözümü açısından farklı performanslara neden olabilir. Problemin gösterim biçimi ve kullanılan genetik işlemciler arasında güçlü bir ilişki vardır ve bazı gösterimler kendileri için özelleştirilmiş genetik işlemcilerin kullanımını isterler (Leitch 1995).

Genetik algoritmalar, problemin çözümünün en kısa zamanda gerçekleştirilmesini amaçladığından gösterim mümkün olan en verimli şekliyle yapılmalıdır. Çoğunlukla

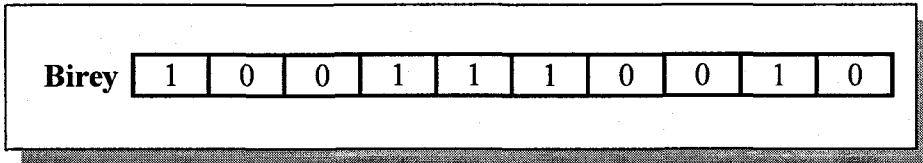
en verimli gösterim şekli, hesaplamalara gerek bırakmayan en kısa gösterimdir (Beasley et al. 1993a).

Eniyileme problemleri için kullanılan iki genel gösterim vardır (Pham and Karaboga 2000).

- İkili dizi gösterimi,
- Tamsayı veya gerçel sayı gösterimi.

#### 4.4.1. İkili dizi gösterimi

İlk genetik algoritma çalışmalarından bu yana ikili dizi gösterimi yaygın olarak kullanılmaktadır. Bu gösterimde her birey  $n$  adet bitle ikili sayıları kullanarak kodlanmaktadır. Şekil 4.2’de verilen sayı dizisi bu gösterime örnek bir bireydir. İkili dizi gösterimi daha çok sayısal eniyileme problemlerini kodlamak için kullanılmaktadır. Genetik algoritmalarda ikili kodlama yapılarak gerçekleştirilen gösterimlerde, gerçel sayıya dönüştürme işlemi sırasında yapılan hesaplamalar zaman kaybına neden olmakta ve bu da ikili kodlamalar için bir dezavantaj olarak gösterilmektedir.



Şekil 4.2. İkili kodlanmış bir birey.

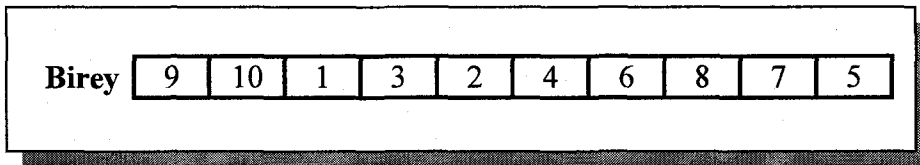
İkili sayıları kullanarak yapılan Gray kodlama literatürdeki alternatif ikili dizi gösterimlerinden biridir. Standart kullanılan ikili kodlu gösterimden tek farkı bir sayının artırılmasının yada azaltılmasının yalnızca tek bir bitin değişimiyle olmasıdır (Leitch 1995). Gray kodlama bu şekilde her bireyde en düşük ağırlıklı bitin değişimiyle devam eder. Bu gösterimde problemi bireyler içinde kodlamaktan daha çok gerçel sayıya dönüştürme işlemi sırasında yapılan hesaplamalar önemlidir. Bu

hesaplamaları yapmak için ayrıca bir algoritma kullanılmalıdır. Bu da gray kodlama için bir dezavantaj olarak gösterilmektedir.

İkili dizi gösterimi kullanılırken parametrelerin kodlanmasında kullanılacak bit sayısının belirlenmesi önemli bir noktadır. Her bir parametre çözüm uzayındaki olası tüm çözümleri karşılayacak en iyi bit sayısı ile kodlanmalıdır. Çok az veya çok fazla kullanılan bit sayısı genetik algoritmanın performansını ters yönde etkileyebilir (Pham and Karaboga 2000).

#### 4.4.2. Tamsayı veya gerçel sayı gösterimi

Tamsayı veya gerçel değerli sayı gösterimi, özellikle ilk kez gerçel parametre problemleriyle ilgilenenler tarafından ortaya atılmıştır. Bu gösterimde birey, tamsayı veya gerçel sayıların ardı arda yazılmasıyla oluşturulur. Şekil 4.3'te verilen sayı dizisi tamsayı gösterime örnek bir bireydir. Tamsayı gösterim çoğunlukla sıralamaya dayalı problemleri (seyahat eden satıcı problemi, yerleştirme problemleri, kesme ve paketleme problemleri vb.) kodlamak için kullanılır. Gen değerleri sabittir, uygunluk genlerin mevcut sıralamasına göre değişmektedir (Beasley et al. 1993b). Gösterimde tamsayı veya gerçel sayıların kullanılması, kodlama ve kod çözme işlemlerine gerek bırakmadığından problem çözümlerini bulmada zaman açısından ikili dizi gösterime göre bir avantajdır.



Şekil 4.3. Tamsayı ile kodlanmış bir birey.

#### 4.5. Başlangıç Nüfusunun Oluşturulması

Genetik algoritmalarla problem çözümüne başlamadan önce problem için olası çözümleri içeren bir başlangıç nüfusu oluşturulur. Başlangıç nüfusu oluşturmada kullanılan iki yöntem vardır (Pham and Karaboga 2000):

- Rasgele bir sayı üretici tarafından rasgele üretilmiş olası çözümler,
- Problem için belirli koşul ve kısıtlamaları karşılayan olası çözümler.

İlk yöntemde nüfus, rasgele bir sayı üretici tarafından rasgele üretilmiş olası çözümlerin kullanılmasıyla oluşturulur. Bu yöntem bir algoritmanın performansının değerlendirilmesi veya hakkında ön bilgi sahibi olunmayan problemlerin çözümü söz konusu olduğunda tercih edilmektedir. İkinci yöntem çözümü istenen probleme ait ön bilgiyi gerektirmektedir. Bu bilgi kullanılarak belirli koşullar ve kısıtlamaları içeren gereksinimler elde edilir. Daha sonra bu gereksinimleri karşılayan çözümler başlangıç nüfusunu oluşturmak üzere bir araya getirilir. Böylelikle genetik algoritma eniyileme işlemine yaklaşık olarak bilinen çözümlerle başlar. Bu sebeple ikinci yöntem ilk yöntemle göre en iyi çözüme daha hızlı bir şekilde yakınsamaktadır.

#### 4.6. Uygunluk Fonksiyonu

Doğadaki uygunluk bireyin yetişmesi ve genlerinin sonraki nesillere iletilmesini engelleyen hastalıklar, düşmanlar ve benzeri zorluklara karşı koyabilme ve hayatta kalabilme kabiliyetidir. Genetik algoritmalarda ise problemlerin çözümü için oluşturulan dizinin sürekliliğine yada elenmesine karar veren uygunluk fonksiyonunun değeridir.

Uygunluk fonksiyonu, genetik algoritmalarda çözümü aranan her problem için mutlaka tanımlanması gereken ve her probleme özgü tanımlanan tek kısımdır. Uygunluk fonksiyonu çözüm aranan probleme bağlı olarak karmaşık veya basit olabilir. Problem ve genetik algoritmalar arasında bir arabirim gibi çalışır.

Uygunluk fonksiyonu bireyleri problemin parametreleri haline getirerek onların bir bakıma şifresini çözer. Yapılan hesaplamalar sonucunda başka hiçbir bilgi kullanılmaksızın uygunluk değerlerine bakılarak içlerinden herhangi birinin problem için çözüm olup olmayacağına karar verilir. Genetik algoritmanın problemin çözümündeki başarısı, çoğunlukla bu fonksiyonun verimli ve hassas tanımlanmasına bağlıdır. Pek çok problem için, uygunluk fonksiyonu önemli bir ölçüttür (Leitch 1995).



Uygunluk fonksiyonu oluşturulurken dikkat edilmesi gereken genel kural, fonksiyonun birey değerini yansıtır yansıtmadığıdır. Bir uygunluk fonksiyonunun ideal olarak tanımlanması, arama uzayında makul uygunluk değerlerine sahip bireylerin kendilerinden biraz daha iyi uygunluk değerlerine sahip bireylere yakın olmasını sağlar. Bu yüzden fonksiyonun ideal olarak tanımlanması istenir ancak pek çok problem için bu mümkün değildir (Beasley et al. 1993a).

#### **4.7. Eniyileme Kriteri**

Genetik algoritmalar öteleme ile işlem yapan bir arama tekniğidir. En iyi çözüm için bir yakınsama garanti edemeyebilir. Bu sebeplerden dolayı arama sürecini sonlandırmak için önceden belirlenmiş bir kritere ihtiyaç vardır. Bu kriter eniyileme veya sonlandırma kriteri olarak bilinir. Bu kriter arama süreci için sabit bir nesil sayısı olabileceği gibi çözümün en iyi değere yakınsamasıyla da elde edilebilir.

#### **4.8. Genetik Algoritma İşlemcileri**

##### **4.8.1. Seçim**

Seçim en temel genetik işlemdir ve ilk kez Darwin tarafından öne sürülen “en iyinin hayatta kalma” ilkesine dayanmaktadır. Seçim işleminin amacı uygunluk değerleri yüksek bireylerin nesiller aracılığıyla daha çok sayıda üretilmelerine imkan sağlamaktır. Seçim işlemi arama uzayında aramanın yönlendirilmesinde ve iyi çözümlerin elde edilmesinde oldukça önemli bir etkinliğe sahiptir. Bununla birlikte nüfus yoğunluğu erken bir yakınsamayı önlemek ve bütünsel en iyi çözüme ulaşmak için korunmalıdır.

Seçim işlemi, bir sonraki nüfusta yeniden üremeleri için o anki nüfusta bulunan en yüksek uygunluk değerine sahip bireylerden birer kopya alınıp, üreme havuzuna atılmasıyla gerçekleştirilir. Üreme havuzunun büyüklüğü nüfus büyüklüğü ile aynıdır. Düşük uygunluk değerine sahip bireylerin üreme havuzuna kopyalarının alınmaları küçük bir olasılıktır. Kopyalama işleminden sonra birey çiftleri üreme havuzundan yeni bireylerin üretimi için alınır. Bu işlem üreme havuzundaki bireyler

bitene kadar tekrarlanır (Lindfield and Penny 1994). Genetik algoritmanın seçim işlemindeki rolü üreme havuzuna gönderilecek bireylerin nasıl seçildiği ile ilgilidir (Leitch 1995).

Genetik algoritmalarda başlıca iki seçim yöntemi vardır (Whitely and Hanson 1989):

- Oransal seçim (Proportional selection),
- Derecelemeye dayalı seçim (Ranking based selection).

#### 4.8.1.1. Oransal seçim

Mekanizması bir rulet tekerleğinin çalışmasını hatırlattığından oransal seçim genellikle “rulet tekerleği seçimi” olarak bilinir. Tekerlek üzerindeki yarıkların genişlikleri bireylerin uygunluk değerlerini temsil etmektedir. Bir sonraki nesilde yer alacak bireyin seçimi için tekerlek rasgele döndürülür. Tekerlek üzerinde büyük genişliklere sahip yüksek uygunluk değerli bireyler seçilmek için daha çok şansa sahiptir.

Rulet tekerleği seçimi kullanılırken izlenen işlem adımları aşağıda sıralanmıştır.

- Nüfustaki bütün bireylerin uygunluk değerleri toplanarak toplam uygunluk değeri Denklem 4.1’de ifade edilen formülle hesaplanmaktadır.

$$\sum_{i=1}^{P_s} U_i \quad (4.1)$$

- Nüfustaki tüm bireylerin uygunluk değerleri, toplam uygunluk değerine bölünerek her bireyin rulet tekerleği üzerinde seçilme olasılığı hesaplanır. Seçilme olasılığı Denklem 4.2’de ifade edilen formülle hesaplanmaktadır.

$$S_i = \frac{U_i}{\sum_{i=1}^{P_s} U_i} \quad (4.2)$$



Denklem 4.2'de  $S_i$  birey  $i$ 'nin seçilme olasılığını,  $U_i$  birey  $i$ 'nin uygunluk değerini göstermektedir. Seçilme olasılığı  $[0, 1]$  aralığında bir değerdir ve toplam seçilme olasılığı 1'e eşittir.

- Bireylerin rulet tekerleği üzerine yerleştirilmesi, her bir bireyin seçilme olasılıklarının birbirine eklenip rulet tekerleği üzerinde gösterilmesiyle gerçekleşir.
- $[0, 1]$  arasında rasgele bir sayı üretilir. Bu sayı rulet tekerleği üzerinde hangi aralığa düşüyorsa o birey seçilmiş olur.

#### **4.8.1.2. Derecelemeye dayalı seçimi**

Dereceleme seçimi, oldukça basit ve etkili bir seçim işlemidir. Nüfus içerisinde bazı bireylerin uygunluk değerleri diğerlerine göre çok büyük olabilir. Uygunluğu büyük bu bireyler yeniden üreme sürecinde etkin olabilirler. Bu da genetik bilginin kaybolmasına neden olur. Bütün bu nedenlerden dolayı genetik yoğunluğun azalmasını ve hızlı yakınsamayı önlemek için dereceleme seçimi kullanılır (Baker 1985). Bireyler öncelikle gerçek uygunluk değerlerine göre sıralanır. Daha sonra yeni elde edilen uygunluk değerleri doğrusal veya üssel olarak derecelerine göre belirlenir. Bir birey mutlak uygunluk değerinden çok derecesine göre orantılı bir olasılıkla seçilir. Sonuç olarak en büyük uygunluk değeri ve ortalama uygunluk değeri arasındaki oran belirli bir değer için normalize edilir. Böylelikle uç değerlerdeki bireylerin etkileri önemsizleşir.

#### **4.8.2. Çaprazlama işlemcisi**

Genetik algoritmaların problem çözümedeki gücü çaprazlama işlemcisinden kaynaklanmaktadır. Olası çözümü temsil eden bireyler arasındaki bilgi değişimi daha iyi çözümler üretilmesini mümkün kılar. Sadece seçim işlemi uygulandığında, nesillerdeki bireyler oldukça birbirine benzer olacaktır. Bunu önlemek için çaprazlama işlemcisi kullanılmaktadır.

Tam (1992)'ye göre çaprazlama işleminin iki amacı vardır:

- Varolan bireylerin genetik özelliklerinin birleşimiyle arama uzayında yeni bireyler başka bir deyişle yeni çözümler elde etmek,
- Küçük uygunluk değerlerine sahip bireylerin “en iyinin hayatta kalma” ilkesinden yola çıkarak nüfus içinden elenmesi.

Çaprazlama işlemi ile, biyolojide iki kromozomun çaprazlanmasına benzer şekilde, sayı dizileriyle gösterilen iki bireyin belirli parçalarının yerleri karşılıklı değiştirilerek iki yeni birey elde edilir. Çaprazlama genellikle, yeniden üretim için seçilmiş birey çiftlerinin tamamına uygulanmaz. 0.6 ve 1.0 arasında rasgele seçilen çaprazlama olasılığına ( $P_c$ ) göre çaprazlama yapılacak bireyler rasgele seçilir. Eğer çaprazlama uygulanmıyorsa çocuklar ebeveynlerinin birer kopyası olarak bir sonraki nüfusta yer alırlar (Beasley et al. 1993a).

Genetik algoritmalarda kullanılan çaprazlama işlemcileri doğrudan problemin bireyler içindeki gösterimiyle ilgilidir ve bazı çaprazlama işlemcileri belirli gösterimler için özelleştirilmiştir.

İkili dizi gösterimleri için kullanılan çaprazlama işlemcileri (Beasley 1993b, Lindfield and Penny 1994):

- Tek noktalı çaprazlama,
- İki veya çok noktalı çaprazlama,
- Düzenli çaprazlama.

Tamsayı gösterimleri için kullanılan çaprazlama işlemcileri (Goldberg 1989, Lindfield and Penny 1994, Houck et al. 1996, Jakobs 1996, Leung et al. 2001).

- Sıralamaya dayalı çaprazlama (Order based crossover - OBX),
- Devirli çaprazlama (Cycle crossover – CX),
- Sıralı çaprazlama (Order crossover- OX),
- Kısmi eşleştirmeli çaprazlama (Partially matched crossover- PMX),
- Düzenli çaprazlama (Uniform crossover – UX),

- Stefan Jakobs çaprazlama (Stefan Jakobs crossover – SJX).

Bu çalışmada ele alınan problemler permütasyona dayalı problemler olduğundan kodlama nüfustaki bireyler içinde tamsayılar kullanılarak yapılmıştır. Bu sebepten dolayı bu bölümde tamsayı gösterimler için kullanılan çaprazlama işlemcileri üzerinde durulmuştur.

#### 4.8.2.1.Sıralamaya dayalı çaprazlama

Sıralamaya dayalı çaprazlamada öncelikle rasgele bir kesim noktası belirlenir. Çaprazlamanın yapılacağı baba olarak isimlendirilen ilk bireyden kesim noktasına kadar olan genler yeni bireyin diğer bir deyişle çocuğun ilk genlerini oluşturur. Çocuğun diğer genleri annenin kullanılmayan genlerinden aynı sırayla alınarak tamamlanır. Bu çaprazlama tekniğine örnek verilen bir Şekil 4.4'te görülmektedir. Kesim noktası 4 alındığında babanın ilk 4 geni (1, 2, 3, 4) çocuğa kopyalanmış ve çocukta babadan alınmayan diğer genler de (9, 10, 6, 8, 7, 5) anneden alınarak çocuk üretilmiştir.

<b>Baba</b>	1	2	3	4	5	6	7	8	9	10
<b>Çocuk</b>	1	2	3	4	9	10	6	8	7	5
<b>Anne</b>	9	10	1	3	2	4	6	8	7	5

Şekil 4.4. Sıralamaya dayalı çaprazlama.

#### 4.8.2.2. Devirli çaprazlama

Devirli çaprazlamada çaprazlamaya babanın ilk geninden başlanır. Çocuğun ilk geni babanın ilk geni olur. Babanın ilk geninin annede karşılık gelen geni babadan alınır. Alınan bu gene karşılık gelen annedeki gen yine babadan seçilir. Böylelikle ilk alınan genle karşılaşılıncaya kadar bu işlem devam ettirilir. Bu işleme bir devir denir. Ve çocuk için bu devir kullanılarak babadan alınacak sabit genler belirlenmiş olur. Daha

sonra çocuğun diğer genleri sırasıyla anneden alınarak çocuk birey oluşturulur. Şekil 4.5’de bu çaprazlama tekniğine ait bir örnek verilmiştir. Babanın ilk geni (1) alınır, babadaki bu gen annedeki (9) genine karşılık gelmektedir. Babadaki, (9) geni de annedeki (7) genine karşılık gelmektedir. Bu devir takip edildiğinde babadan alınacak genler (1, 9, 7, 6, 4, 3) sırasıyla belirlenir. Bu genler çocuk içerisine babadaki konumuna göre yerleştirilir. Diğer eksik kalan genler de anneden tamamlanır.

<b>Baba</b>	1	2	3	4	5	6	7	8	9	10
<b>Çocuk</b>	1	10	3	4	2	6	7	8	9	5
<b>Anne</b>	9	10	1	3	2	4	6	8	7	5

Şekil 4.5. Devirli çaprazlama.

#### 4.8.2.3. Sıralı çaprazlama

Bu teknik aynı zamanda doğrusal çaprazlama olarak da bilinmektedir. Bu çaprazlama tekniğinde ilk önce rasgele iki kesim noktası belirlenir. Bu kesim noktaları arasında kalan genler babadan alınarak çocuk içindeki aynı konuma kopyalanır. Daha sonra babadan alınmayan diğer genler anneden sırasıyla alınarak çocuk oluşturur. Bu teknik Şekil 4.6’da bir örnekle gösterilmektedir. Birinci kesim noktasını 4, ikincisini kesim noktasını 6 seçtiğimizde çocuk için babadan alınan genler (4, 5, 6) olarak belirlenmektedir. Bu genler çocukta aynı konuma yerleştirilir ve çocuğun babadan alınmayan diğer genleri (9, 10, 1, 3, 2, 8, 7) sırasıyla anneden alınarak tamamlanır.

<b>Baba</b>	1	2	3	4	5	6	7	8	9	10
<b>Çocuk</b>	9	10	1	4	5	6	3	2	8	7
<b>Anne</b>	9	10	1	3	2	4	6	8	7	5

Şekil 4.6. Sıralı çaprazlama.

#### 4.8.2.4. Kısmi eşleştirmeli çaprazlama

Kısmi eşleştirmeli çaprazlamada rasgele iki kesim noktası belirlenir. Babadaki bu 2 kesim noktası arasında kalan genler çocuktaki aynı konuma kopyalanır. Çocuktaki diğer genler başlangıçta annenin bu iki kesim noktası dışındaki genlerinden aynı sırayla alınarak tamamlanır. Fakat burada çocuk içinde tekrarlanan genler olabilir ve bu birey içinde istenmeyen bir durumdur. Bu durumda tekrarlanan gen kesim noktaları arasındaki babanın annede karşılık gelen geniyle değiştirilir. Şekil 4.7’de bu çaprazlama tekniği bir örnekle gösterilmektedir. birinci kesim noktasını 3, ikinci kesim noktasını 6 seçtiğimizde çocuk için babadan alınan genler (3, 4, 5, 6) olarak belirlenmektedir. Bu genler çocuktaki aynı konuma yazılır ve çocuğun diğer genleri anneden sırayla tamamlanır (9, 10, 6, 8, 7, 5). Yalnız burada 6 ve 5 genleri tekrar yapmaktadır. Bunu engellemek için tekrar yapan bu genler sırasıyla babanın annede karşılık gelen 1 ve 2 genleriyle yer değiştirerek çocuğun üretilmesi sağlanır.

Baba	1	2	3	4	5	6	7	8	9	10
Çocuk	9	10	3	4	5	6	1	8	7	2
Anne	9	10	1	3	2	4	6	8	7	5

Şekil 4.7. Kısmi eşleştirmeli çaprazlama.

#### 4.8.2.5. Düzenli çaprazlama

Düzenli çaprazlamada öncelikle babanın tüm genleri çocuğa kopyalanır. Boyutu birey boyutunun bir eksiği kadar, değerleri 0 ve 1 olmak üzere rasgele değişen bir vektör oluşturulur. Oluşturulan bu vektörün 1 değerine sahip elemanlarının indisleri babanın bir kopyası olan çocuktaki değiştirilecek genleri göstermek için kullanılır. Babadaki bu indislere sahip genlerin annede karşılık gelenlerinin indis değerleri alınır ve sıralanır. Daha sonra sıralanan bu indis değerlerindeki genler anneden alınarak çocukta önceden tespit edilmiş değiştirilecek genlerin yerine yazılır. Ve böylece çaprazlama gerçekleştirilmiş olur. Şekil 4.8’de bu çaprazlama tekniği bir

örnekle gösterilmektedir. Çocuğun değişecek (1, 4, 7, 8, 9) konumlarındaki genleri yerine anneden bu genlere karşılık gelen (9, 1, 4, 8, 7) genleri çocuğa kopyalanarak yeni bir birey elde edilmiş olur.

<b>Baba</b>	1	2	3	4	5	6	7	8	9	10
<b>Çocuk</b>	9	2	3	1	5	6	4	8	7	10
<b>Anne</b>	9	10	1	3	2	4	6	8	7	5

**Şekil 4.8.** Düzenli çaprazlama.

#### 4.8.2.6. Stefan Jakobs çaprazlama

Stefan Jakobs çaprazlamada öncelikle rasgele 2 sayı (p ve q) belirlenir. Bunlardan biri (p) çaprazlamaya hangi noktadan başlanacağına diğeri de (q) o noktadan itibaren kaç gen alınacağını gösterir. Babanın p.geninden itibaren q adet gen alınır ve bu genler çocuğun ilk genlerini oluşturur. Çocuğun babadan alınmayan diğer genleri de anneden sırayla alınarak çaprazlama tamamlanır. Şekil 4.9'da bu çaprazlama tekniği bir örnekle gösterilmektedir. p ve q sayısı sırasıyla 3 ve 4 olarak alınır. Babanın 3.geninden itibaren 4 gen alınarak çocuğun ilk genleri olarak kopyalanır. Çocuğun eksik kalan diğer genleri de (9, 10,1, 2, 8, 7) anneden sırasıyla alınarak tamamlanır.

<b>Baba</b>	1	2	3	4	5	6	7	8	9	10
<b>Çocuk</b>	3	4	5	6	9	10	1	2	8	7
<b>Anne</b>	9	10	1	3	2	4	6	8	7	5

**Şekil 4.9.** Stefan Jakobs çaprazlama.

### 4.8.3. Mutasyon işlemcisi

Mutasyon işlemi, çoğunlukla çaprazlama işlemcisinden sonra kullanılan ikinci derecede önemli bir genetik işlemcidir. Yanlışlıkla yada kazara kaybolan gen değerlerinin yeniden ortaya çıkarılmasını, genetik yığılmanın önlenmesini ve nüfusun çoğunluğuyla çözüme yakınsadığında bütünsel en iyi çözümü bulmak için nüfus çevresinde rasgele küçük adımlarla arama yapılmasını sağlar. Mutasyon rasgele aramanın küçük bir kısmını gerçekleştirmekte ve daha çok arama uzayında araştırılmamış bir nokta kalmaması için diğer genetik algoritma işlemcilerine yardımcı olmaktadır. Ancak geleneksel görüş arama uzayının hızlı bir şekilde araştırılmasında çaprazlama işleminin mutasyon işleminden daha önemli ve etkin olduğu konusunda birleşmektedir (Beasley et al. 1993b).

Mutasyon işlemi çaprazlama işleminin tersine tek bireyden tek çocuk üretir. Mutasyon işlemi her nesildeki nüfusa daha önceden tespit edilmiş belirli bir mutasyon oranı ( $P_m$ ) ile uygulanır. Herhangi bir bireyin mutasyon işlemine uğrayıp uğramayacağını mutasyon oranı belirler. Mutasyon işlemi bir bireyde herhangi bir genin değerinin değiştirilmesi yada genlerin yerlerinin değiştirilmesi ile gerçekleştirilir. Bireyde genler birbirinden bağımsız olarak mutasyon işlemine tabi tutulurlar ve bir genin değerinin değişmesi diğer genlerin değerlerini etkilemez. Mutasyon oranının çok yüksek olması, bireyler içinde çaprazlama ile elde edilen başarılı yapıların hızla kaybedilmesine yol açabilir (Man et al. 1996):

Genetik algoritmalarda problemlere göre özelleştirilmiş gösterimlerle birlikte kullanılmak üzere özelleştirilmiş mutasyon işlemcileri vardır.

İkili dizi gösterimleri için kullanılan mutasyon işlemcileri (Man et al. 1996):

- İkili mutasyon (Binary mutation),
- Ters mutasyonu.(Inversion mutation).

Tamsayı gösterimleri için kullanılan mutasyon işlemcileri (Houck et al. 1996, Leung et al. 2001):



- Yer deęiřtirme mutasyonu (Swap mutation),
- Üçlü yer deęiřtirme mutasyonu (Three swap mutation),
- Kaydırma mutasyonu (Shift mutation),
- Ters mutasyon (Inversion mutation),
- Bitiřik mutasyon (Adjacent mutation).

Bu çalışmada ele alınan problemler permütasyona dayalı problemler olduğundan nüfusta içindeki bireyler tamsayılar kullanılarak kodlanmıştır. Bu yüzden tamsayı gösterimler için özelleştirilmiş bir mutasyon işlemcisi kullanılır. Mutasyon işlemcisi bireyler üzerinde çaprazlama işlemcisine göre daha az etkili olduğundan burada ele alınan tek mutasyon işlemcisi yer deęiřtirme mutasyonudur.

Yer deęiřtirme mutasyonu: Birey içinde rasgele iki gen seçilir ve yerleri deęiřtirilerek yeni bir birey üretilir. Şekil 4.10'da bu mutasyon işlemine bir örnek verilmiştir. Rasgele seçilen 7. ve 9. genler arasında yer deęiřtirildiğinde çocuk birey elde edilmiş olur.

<b>Birey</b>	9	10	1	3	2	4	6	8	7	5
<b>Çocuk</b>	9	10	1	3	2	4	7	8	6	5

**Şekil 4.10.** Yer deęiřtirme mutasyonu.

#### 4.9. Yeni Nüfusun Oluřturulması

Yeni nüfusun oluşturulmasında ana kriter, nüfus büyüklüğünün eski nüfus büyüklüğüne eşit olmasıdır. Yeni birey sayısı nüfus boyutu kadar olmak zorunda değildir. Ancak başlangıç nüfusu boyutunun korunması esastır. Bu yüzden eski nüfusun bireylerinin seçim, çaprazlama ve mutasyon işlemlerinden geçirilmesinden sonra oluşan yeni bireylerin, uygunluk değerlerine göre yeni nüfusa eklemeleri gerekmektedir (Man et al. 1996).



#### 4.10. Benzetilmiş Tavlama

Yerel arama yöntemleri, arama sürecinde yerel, en küçük çözümde takılıp kalabilir ve daha iyi bir çözüm değerine ulaşılmasına engel olabilirler. Arama algoritmaları için bir dezavantaj sayılan bu durum karşısında daha karmaşık arama yöntemleri geliştirilmiştir. Benzetilmiş tavlama (BT) algoritması, bu yöntemlerden birisidir. Benzetilmiş tavlama fiziksel tavlama kavramının ilk olarak kombinasyonel eniyileme problemlerinde uygulandığı 1980'li yıllarda, bir eniyileme aracı olarak sunulmuştur.

Benzetilmiş tavlama algoritmasının adı, ilk kez Metropolis tarafından ileri sürülen katı maddelerin tavlama sürecinin benzetimi ve kombinasyonel eniyileme problemlerinin çözüm yaklaşımı arasındaki benzerlikten gelmektedir (Kirkpatrick et al. 1983, Thelma and Panos 1997). Benzetilmiş tavlama algoritması, statik fizikte bir katı maddenin enerji durumunu en aza indiren tavlama süreci ve kombinasyonel eniyileme problemlerinin çözüm süreci arasındaki benzerliğe dayanarak Kirkpatrick (1983) tarafından geliştirilmiş bir yerel arama algoritmasıdır. Fiziksel tavlama kombinasyonel problemler için model oluşturmada kullanılır. Buna göre, sistemdeki enerji durumlarının problemin çeşitli olası çözümlerine benzetilmesi ve sistemin enerjisinin de problem için en aza indirilmek istenen maliyet fonksiyonuna benzetilmesiyle tavlama gerçekleştirilir. Eniyileme problemleri ve fiziksel sistemler arasındaki benzerlik Tablo 4.1'de görülmektedir (Lutfiyya et al. 1992).

**Tablo 4.1.** Fiziksel sistemler ve eniyileme problemleri arasındaki benzerlik.

<b>Fiziksel sistemler</b>	<b>Eniyileme problemleri</b>
Durum	Konfigürasyon
Enerji	Maliyet fonksiyonu
Son durum	En iyi çözüm
Hızlı soğutma	Ötelemeye ilerleme
Ölçülü tavlama	Benzetilmiş tavlama

Bir sisteme ait durumlar tamamıyla kombinasyonel bir eniyileme probleminin konfigürasyonuna benzemektedir. Sistemde azaltılmak istenen enerji eniyileme

problemlerindeki en aza indirilmek istenen maliyet fonksiyonuna karşılık gelir. Son durum maliyet fonksiyonunun en küçük olduğu en iyi konfigürasyondur. En iyi konfigürasyonun diğer bir deyişle en iyi çözümün elde edilme süreci, fiziksel sistemlerde son duruma karar verilmesi ile belirlenir. Eniyileme problemlerindeki en iyi çözüm için yapılan ötelemeyle ilerlemeler fiziksel sistemlerdeki soğutma sürecine karşılık gelmektedir. Bu süreç benzetilmiş tavlama olarak bilinir. Kontrol parametresi olarak sıcaklık kullanılmaktadır.

Benzetilmiş tavlama; diğer eniyileme metotlarındaki yerel en küçük değere takılma gibi doğal dezavantajlardan kaçınan, fonksiyonun bütünsel veya bütünsel yakın en küçük değerini bulmaya çalışan bir eniyileme yöntemidir. Bu yöntem, yüzlerce veya daha çok değişkene sahip çeşitli fonksiyonların en iyi değerlerinin bulunması ve özellikle pek çok yerel en küçük değere sahip doğrusal olmayan fonksiyonların en iyi değerlerinin bulunması için tasarlanmıştır (Lutfiyya et al. 1992). Benzetilmiş tavlama yöntemi, elektronik sistemlerin düzeni, parça yerleştirme, seyahat eden satıcı problemi, yol bulma, görüntü işleme, kesme problemleri, akış tipi çizelgeleme ve iş çizelgeleme problemlerinin çözümünde başarılı sonuçlar vermiştir (Kirkpatrick et al. 1983, Lutfiyya et al. 1992, Lai and Chan 1997).

#### **4.10.1. Metallerde tavlama**

Tavlama, belirli bir süre için katı maddenin sıcaklığının artırılıp, daha sonra kristalleşinceye kadar yavaş yavaş soğutulduğu fiziksel bir süreçtir. Kristalleşme düzenli bir yapıyla, sistemin enerjisinin en aza indiği durumda elde edilir (Pham and Karaboga 2000). Çağlar öncesinde demircilerin de fark ettiği gibi daha mükemmel bir kristal yapı daha yavaş soğutma ile gerçekleşmektedir. Bunun nedeni, ısı enerjisi tarafından sağlanan hareket etme ve yer değiştirme kolaylığının atomların yerel en iyi şekillendirmeden kaçınmalarına bunun yerine bütünsel en iyi şekillendirmeyi oluşturmalarına imkan vermesidir (Antonioletti 1996).

Metallerde tavlama işlemi aşağıdaki amaçlarla yapılır:

- Stresi azaltmak
- Yumuşaklık, dayanıklılık ve esnekliği arttırmak,
- Belirli bir mikro yapıyı oluşturmak

Metaldeki atomlar yüksek sıcaklıklarda oldukça yüksek bir enerjiye ve düzenli bir şekillendirme yapabilmek için daha çok serbestliğe sahiptirler. Metal soğumaya başladıkça, atomik enerjisi de azalır. Bu soğutma işlemi genellikle oda sıcaklığına kadar olur. Bu işlemlerde zaman önemli bir parametredir. Eğer soğutma işlemi hızlı bir biçimde gerçekleştirilirse, kristal yapı içerisinde yaygın bir şekilde düzensizlikler ve bozulmalar görülür. Sistem, enerjinin en düşük olduğu duruma ulaşamaz ve enerjinin en yüksek olduğu polikristal durumda soğutma işlemi sona erer (Pham and Karaboga 2000). Bu sebeple sistemde enerjinin en düşük olduğu en iyi kristal yapının elde edilebilmesi için, gerçek tavlama süresi yeteri kadar uzun tutulmalıdır. Ayrıca tavlama sıcaklığı da bu süreçte önemli bir konudur. Sıcaklık artırılarak tavlama işlemi hızlandırılabilir.

#### 4.10.2. Benzetilmiş tavlama algoritması

Benzetilmiş tavlama bir yerel arama algoritması olmakla birlikte eniyileme problemlerinin çözümü için son zamanlarda oldukça sık başvurulan bir arama yöntemidir. Bu yöntemde amaç, problem için uygun çözümler ararken zayıf bir yerel çözümde takılıp kalmayı engellemektir.

Arama süreci çoğunlukla rasgele seçilen bir başlangıç çözümüyle başlar. Sonrasında problemin arama uzayında seçilen başlangıç çözümüne yakın başka bir çözüm üretilir ve bu iki çözümün maliyetleri arasındaki değişim miktarı ( $\Delta C$ ) Denklem 4.3'de ifade edilen formülle hesaplanır. Formüldeki  $C_i$  i.ötelemadaki çözümü başka bir deyişle yeni elde edilen çözümü,  $C_{i-1}$  i-1. ötelemadaki başka bir deyişle o anki çözümü temsil etmektedir.

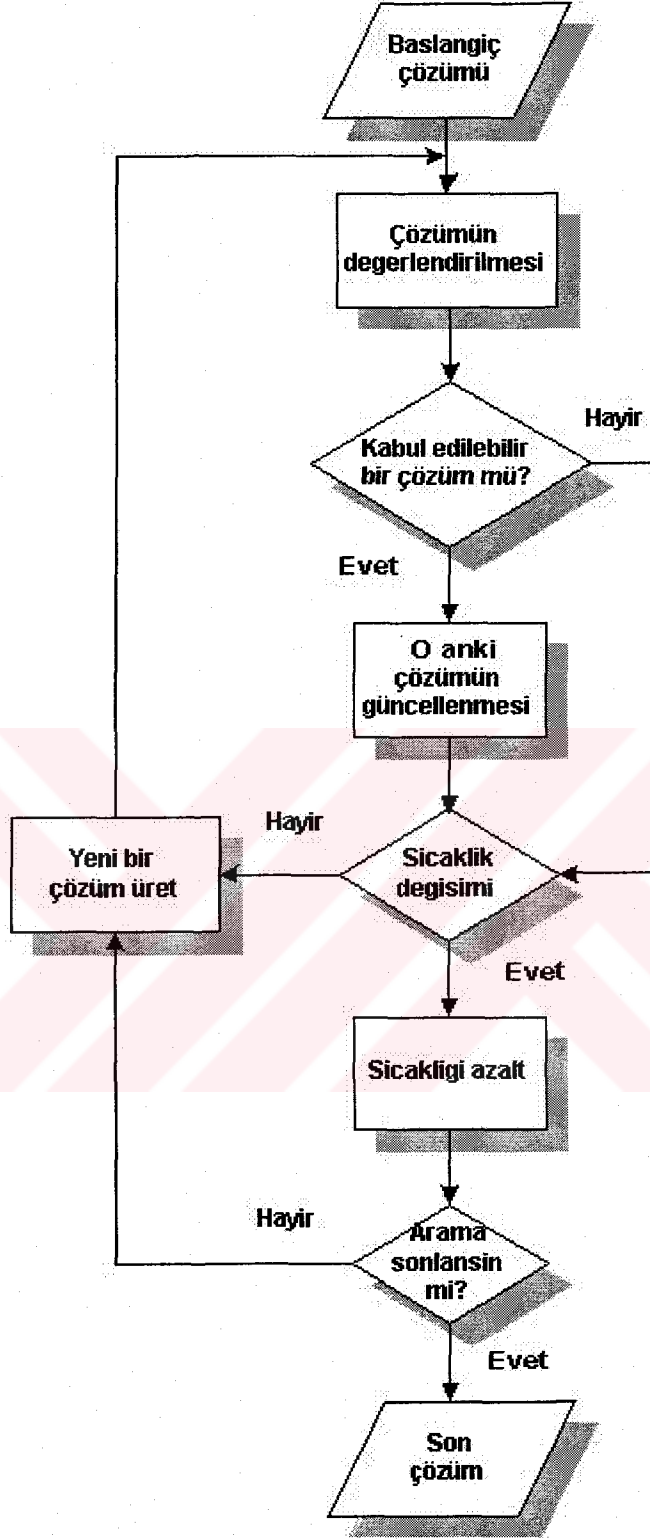
$$\Delta C = C_i - C_{i-1} \quad (4.3)$$

Genel bir yerel arama süreci için maliyette bir azalma gerçekleşiyorsa, o anki çözüm yeni üretilmiş diğer çözümle değiştirilir. Aksi durumda, o anki çözüm Denklem 4.4'te ifade edilen formüle göre belirli bir olasılıkla saklı tutulur (Leung et al. 2001).

$$\exp(-\Delta C / T) > R \quad (4.4)$$

Formüldeki  $\Delta C$  çözümlerin maliyetleri arasındaki değişim miktarıdır.  $R$   $[0,1]$  arasında üretilmiş düzgün dağılımlı rasgele bir sayıdır.  $T$  sıcaklık olarak bilinen kontrol parametresidir. Başlangıçta göreceli olarak yüksek bir sıcaklıktır. Bundan dolayı yeni çözümlerin kabul edilme şansı oldukça yüksektir. Daha sonra tavlama süreci boyunca önceden belirlenmiş bir soğutma planına göre yavaş yavaş azaltılır. Bu azaltmayla, algoritma yeni çözümleri kabul ederken çok daha seçici olur. Tavlama süreci, o anki çözümün yakınlarındaki diğer çözümlerde başka bir değişime rastlanmaya kadar tekrarlanır. Bu durumda yerel arama algoritması, yerel en küçük bir çözümle sonlandırılır. Kombinasyonel eniyileme problemlerinin çözümündeki bu tavlama sürecini gösteren bir benzetilmiş tavlama algoritmasının akış diyagramı Şekil 4.11'de gösterilmektedir (Pham and Karaboga 2000).

Algoritmada, yeni çözümlerin kabul edilme olasılığı ( $\Delta C$ )'nin artan değerleri ve  $T$  sıcaklığının azalan değerleri için azalmaktadır. Sıcaklık değerleri her ötelemede önceden kabul edilmiş bir soğutma planına göre güncellenir. Ayrıca önceden belirlenmiş bir durdurma kriteri de, bu arama sürecini sonlandırmak için kullanılır (Lai and Chan 1997).



Şekil 4.11. BT algoritmasının akış diyagramı.

### 4.10.3. Benzetilmiş tavlama algoritması parametreleri

Benzetilmiş tavlama algoritması bir probleme uygulanmadan önce aşağıda sıralanan parametrelerin belirlenmesi gerekmektedir (Pham and Karaboga 2000).

- Çözümlerin gösterimi,
- Yakın çözümleri üreten mekanizmanın tanımlanması (komşuluk hareketi),
- Soğutma planının seçimi,
- Durdurma kriteri ve denge koşulunun belirlenmesi.

#### 4.10.3.1. Çözümlerin gösterimi

Çözümlerin gösterimi problemden probleme değişiklik göstermektedir. Gösterim yöntemi benzetilmiş tavlamanın performansını etkileyen önemli etkenlerden biridir. Genetik algoritmalarda olduğu gibi çözümlere ait iki genel gösterim vardır. Birinci gösterim ikili dizi kullanılarak yapılır, ikinci gösterim ise tamsayı veya gerçek sayıların bir vektör gibi kullanılması suretiyle yapılır.

#### 4.10.3.2. Komşuluk hareketi

Komşuluk hareketi, benzetilmiş tavlama önemli bir parametredir ve o anki çözüme en yakın yeni çözümü üretmek için kullanılır. Algoritma, önceden kararlaştırılmış denge koşulu sağlanıncaya kadar komşuluk hareketinin yapılmasını ve yeni çözümlerin üretilmesini ister. Yapılan her komşuluk hareketi sonunda arama uzayında yeni bir çözüm elde edilir. Oluşan her yeni çözüm içerisindeki değişikliklerin derecesini kontrol edebilmek için bir vektör uzaklığı tanımlanmıştır (Lai and Chan 1997). Vektör uzaklığı, çözüm olarak elde edilen permütasyonda sondan kaç tane eleman arasında komşuluk hareketinin yapılacağını belirleyen bir faktördür. Vektör uzaklığı arttıkça, bir önceki çözüme göre çözümü oluşturan permütasyonda çeşitlilik artar. Örneğin,  $[r_1, r_2, r_3, r_4, r_5, r_6, r_7]$  olarak verilmiş permütasyonda vektör uzaklığı 5 olsun. Bu, permütasyonda komşuluk hareketi için sadece  $r_3$  ve  $r_7$  arasında rasgele 2 elemanın seçilebileceğini göstermektedir. Vektör

uzaklığı 2 alınır, komşuluk hareketi için sadece  $r_6$  ve  $r_7$  elemanları aralarında yer değiştirebilir.

Vektör uzaklığı, algoritma içerisinde bir sıcaklık değeri ile belirlenir. Algoritmada, her ötelemede vektör uzaklığını belirleyen sıcaklık Denklem 4.5'te ifade edilen formülle hesaplanmaktadır.

$$T = e^{(\log(2/n)/M)} \quad (4.5)$$

Formüldeki  $n$  yerleştirilecek parça sayısını,  $M$  toplam öteleme sayısını temsil etmektedir. Vektör uzaklığı ve sıcaklık arasında üssel bir ilişki vardır. Yüksek sıcaklıklarda vektör uzaklığı daha büyüktür ve büyük komşuluk hareketleri yapılır. Sıcaklık azalmaya başladıkça vektör uzaklığı da üssel olarak azalmaktadır ve daha küçük komşuluk hareketleri yapılır. En yüksek sıcaklıkta vektör uzaklığı en büyüktür, en düşük sıcaklıkta ise vektör uzaklığı 2'dir. Bunun anlamı permütasyonda sadece son iki eleman arasında komşuluk hareketi yapılabileceğidir.

Pek çok araştırmacı yüksek sıcaklıklarda büyük komşuluk hareketlerinin yapılması ve sıcaklık düşüğe küçük hareketlerin yapılması gerektiği düşüncesinde birleşmektedir. Büyük komşuluk hareketleri, maliyet fonksiyonunun değerinde büyük değişimlere neden olurken, küçük komşuluk hareketleri küçük değişimlere neden olmaktadır. Yüksek sıcaklıklarda büyük komşuluk hareketleri algoritmanın, yerel en küçük değerden kaçınmasına yardımcı olur (Lutfiyya 1992).

Bu bölümde yer değiştirme ve kaydırma olmak üzere iki farklı komşuluk hareketi üzerinde durulmuştur (Lai and Chan 1997, Leung et al. 2001).

#### 4.10.3.2.1. Yer değiştirme

Dizide yer değiştirmesi istenen rasgele 2 eleman seçilir ve bunların yer değiştirmeleriyle yeni dizi elde edilir. Örneğin, (1,2,3,4) elemanlarına sahip dizide rasgele olarak seçilmiş 1. ve 4. elemanın yer değiştirmesi yapıldığında (4,2,3,1) elemanlarına sahip yeni dizi elde edilir.

#### 4.10.3.2.2. Kaydırma

Diziden rasgele bir eleman seçilir ve bu eleman yine rasgele seçilmiş olan bir elemanın önüne koyularak yeni dizi elde edilir. Örneğin, (1,2,3,4) elemanlarına sahip dizide rasgele seçilmiş 4.eleman rasgele seçilmiş 2.elemanın önüne koyularak (1,4,2,3) elemanlarına sahip yeni dizi elde edilir.

#### 4.10.3.3. Soğutma planı

Benzetilmiş tavlama algoritmasının performansı her ötelemede sıcaklık değerini güncelleyen soğutma planının seçimine bağlıdır. İyi bir soğutma planıyla pek çok kombinyonel problemin en iyi çözümleri elde edilebilir.

Bu bölümde, oransal soğutma planı ve Lundy ve Mees soğutma planı olmak üzere 2 farklı soğutma planı üzerinde durulmuştur (Lai and Chan 1997, Leung et al. 2001).

##### 4.10.3.3.1. Oransal soğutma planı

Oransal soğutma planında, k. ve k+1. ötelemelerindeki  $T_k$  ve  $T_{k+1}$  sıcaklıkları, aralarında belirli bir  $\alpha$  katsayısıyla ilişkilendirilmiştir. Her bir k ötelemesinde  $T_{k+1}$  sıcaklığı, Denklem 4.6'da ifade edilen formülle yeniden hesaplanarak güncellenmektedir.

$$T_{k+1} = \alpha T_k \quad (4.6)$$

$\alpha$  katsayısı, toplam öteleme sayısı boyunca başlangıç sıcaklığından son sıcaklığa kadar Denklem 4.7'de ifade edilen formülde hesaplanan adımlarla azalmaktadır.

$$\alpha = \exp\left(\frac{\log(T_s / T_b)}{M}\right) \quad (4.7)$$

Denklem 4.7'de  $T_s$  son sıcaklığı,  $T_b$  başlangıç sıcaklığını ve  $M$  toplam öteleme sayısını göstermektedir.  $\alpha$  katsayısı ise, (0, 1) arasında değişmektedir.



#### 4.10.3.3.2. Lundy ve Mees soğutma planı

Bu soğutma planında, k. ve k+1. ötelemelerindeki  $T_k$  ve  $T_{k+1}$  sıcaklıkları, bir  $\beta$  katsayısıyla ilişkilendirilmiştir. Her yeni ötelemede sıcaklık sabit azaltma oranı  $\beta$ 'nın kullanımıyla bir önceki sıcaklığa göre azaltılır. Sıcaklığı güncellemek için kullanılan formül, Denklem 4.8'de görülmektedir.

$$T_{k+1} = \frac{T_k}{(1 + \beta * T_k)} \quad (4.8)$$

$\beta$  katsayısı, toplam öteleme sayısı ile başlangıç ve son sıcaklığa bağlı olarak Denklem 4.9'da ifade edilen formülle hesaplanmaktadır (Lai and Chan 1997).

$$\beta = \frac{T_b - T_s}{M * T_i * T_s} \quad (4.9)$$

Denklem 4.9'da  $T_s$  son sıcaklığı,  $T_b$  başlangıç sıcaklığını ve  $M$  toplam öteleme sayısını göstermektedir.  $\beta$  katsayısı ise, 0 dan büyük bir değerdir.

Bu çalışmanın tamamında kullanılan toplam öteleme sayısı 1000, son sıcaklık 0.01 ve başlangıç sıcaklığı (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8) olmak üzere 8 farklı değer kullanılmıştır.

#### 4.10.3.4. Durdurma kriteri ve denge koşulu

Benzetilmiş tavlama algoritması arama süreci boyunca iki ana döngü kullanır. Dış döngü, durdurma kriterini karşılayıp karşılamadığını, iç döngü denge koşulunun sağlanıp sağlanmadığını kontrol eder. İç döngü tamamlandığında sıcaklık güncellenir ve durdurma kriteri yeniden kontrol edilir.

Bütün bir arama sürecini sonlandırmak için iki tür durdurma kriteri kullanılabilir. Bunlardan biri toplam öteleme sayısı diğeri de önceden belirlenmiş kabul edilebilir

bir fire değeridir. Bu iki kriterin, hesaplama sürelerine ve çözüm kalitelerine bakılarak içlerinden çalışma için en uygun olanı seçilir (Lai and Chan 1997).

Benzetilmiş tavlama algoritması uygulanırken denge koşulunun sağlanıp sağlanmadığını kontrol edebilmek için aşağıda sıralanan yöntemlerden biri kullanılabilir (Leung et al, 2001):

- Olası en büyük komşuluk hareketi sayısı.
- Beş kez tekrarlanan komşuluk hareketi için bir değişim olmaması hali.
- Yeteri kadar komşuluk hareketi gerçekleştirilmesi.

Bu çalışmada durdurma kriteri olarak öteleme sayısı alınmıştır. 1000 öteleme yapıldıktan sonra çalışma tamamıyla sonlandırılmaktadır. Denge koşulu olarak olası en büyük komşuluk hareketleri sayısı (3, 5, 10) olarak alınmıştır. Sistemin dengeye ulaşp ulaşmadığı belirtilen komşuluk hareketleri sayısı kadar, test edilmiştir.

## **BÖLÜM 5. GENETİK ALGORİTMA ve BENZETİLMİŞ TAVLAMA ile KESME PROBLEMLERİNİN ÇÖZÜM YAKLAŞIMI**

Bu bölümde, Bölüm 3'te anlatılan geliştirilmiş AS algoritması ile birlikte Bölüm 4'te anlatılan genetik algoritma ve benzetilmiş tavlama algoritması kullanılarak iki boyutlu giyotinsiz bir kesme probleminin çözümünü gerçekleştirilmiştir.

Kesme problemi önce genetik algoritma yöntemi ile ele alınmıştır. Genetik algoritma kullanarak yapılan çalışmada, her biri 200x200 birimlik bir alan ile sınırlandırılmış bir ana parça ve yerleşecek 17 adet birbirinden farklı düzgün dikdörtgen parçadan oluşan, 5 farklı test problemi üzerinde çalışılmıştır. Bu problemin çözümünde Stefan Jakobs çaprazlama tekniği kullanılarak, farklı nüfus büyüklükleri ve mutasyon oranlarının problemin çözümü üzerindeki değişimleri incelenmiştir. Daha sonra yine aynı özelliklere sahip bu defa 29 adet parçadan oluşan bir test problemi (Hopper 2000) üzerinde nüfus büyüklüğü ve mutasyon oranı sabit tutularak 6 farklı çaprazlama tekniği (Sıralamaya dayalı çaprazlama (OBX), devirli çaprazlama (CX), sıralı çaprazlama (OX), kısmi eşleştirmeli çaprazlama (PMX), düzenli çaprazlama (UX), Stefan Jakobs çaprazlama (SJX)) ile çalışılmıştır. 29 adet parça kullanılarak yapılan bu çalışmada farklı çaprazlama tekniklerinin uygunluk değerleri üzerindeki değişimleri incelenmiştir.

Aynı kesme problemlerine, ikinci çalışmada benzetilmiş tavlama algoritması ile çözüm aranmıştır. 17 parça kullanılarak yapılan çalışmada oransal soğutma planı ve Lundy ve Mees soğutma planı, yer değiştirme ve kaydırma komşuluk hareketleri ve denge koşulu için değişik değerler (3, 5, 10) kullanılarak benzetilmiş tavlama algoritması parametrelerinin problemin çözümü üzerindeki etkileri incelenmiştir. 29 parça kullanılarak yapılan çalışmada da bir önceki çalışmanın çözümünde elde edilen en iyi sonuçların parametreleri kullanılarak farklı sıcaklık değerlerindeki değişimler incelenmiştir.

Bu çalışmada kullanılan 17 ve 29 adet düzgün biçimli dikdörtgen parçadan oluşan giyotinsiz test problemleri Ek-B'de gösterilmektedir (Hopper 2000).

### 5.1. Çözüm Yaklaşımı

Bu çalışmada ele alınan problemler de iki boyutlu giyotinsiz düzgün şerit kesme problemleridir. Bu problemler, özdeş olmayan düzgün küçük parçaların sınırları önceden belirlenmiş levha biçimindeki ana parça içine yerleştirilmesiyle ilgilenir. Kesme süreci boyunca en elverişli yerleşim planını bulmaya çalışır. Bu tür problemler, yerleştirilecek parça sayısı 10 ve 15 arasında olduğunda kolay problem, parça sayısı 25 ve 30 arasında olduğunda zor problem olarak adlandırılır. Fire değerleri kolay problemler için %0'dan başlayan zor problemler için %8'e kadar artan miktarlarda değiştiğinde kabul edilebilir standartlar arasındadır (Leung et al. 2001). Bu çalışmada, özel olarak seçilmiş problemler için bilinen en iyi çözümlere ait fire değerleri sıfırdır.

Yerleşim planının doğal gösterimi, ana parça üzerinde her bir dikdörtgen parçanın yerleştirme koordinatları üzerine kurulmuştur. Bu gösterimin avantajı yerleşim planının yeniden oluşturulmasının kolay olmasıdır. Ancak koordinatlarda küçük değişimler yapıldığında, yeniden oluşturulmuş yerleşim planında çakışmalar olacaktır. Doğal gösterimin bu özelliği genetik algoritma için uygun değildir. Bu sebeple daha kullanışlı bir veri yapısı olan permütasyona gereksinim vardır. Jakobs (1996)'ya göre bir kesme yerleşim planı bir permütasyonla gösterilmelidir. Bu permütasyon belirli kurallara göre ana parça üzerinde yerleştirilecek küçük parçaların sırasını temsil etmektedir. Bu veri yapısının avantajı yerleşim sırasının değişimiyle yeni permütasyonların kolaylıkla elde edilebilmesidir. Bu nedenlerden dolayı çalışmada genetik algoritma ve benzetilmiş tavlama algoritması en az fireyi veren yerleşim sırası permütasyonunu bulmak için kullanılmıştır. Genetik algoritma ve benzetilmiş tavlama algoritması için en iyi çözüm en az fire değerini veren permütasyonu bulunca elde edilir. Bu işlemlere kodlama denir. Yerleşim sırasını veren permütasyon elde edildikten sonra ana parça üzerinde yerleşim geliştirilmiş AS algoritmasının kullanılmasıyla gerçekleştirilir. Bu işleme kod çözme denir. Kesme problemlerinin çözümünde geliştirilmiş AS yerleştirme algoritmasıyla birlikte

kullanılan genetik algoritma ve benzetilmiş tavlama algoritması, kırma GA ve kırma BT algoritması adını alır.

Bu çalışmada gerçekleştirilen yerleştirme işlemi aşağıdaki kabuller göz önünde bulundurularak yapılmıştır (Leung et al. 2001):

1. Bütün parçalar sabit bir yöndedir. Uzunluğu  $l$  genişliği  $w$  olan bir parça eğer parçanın boyutları birbirine eşit değilse uzunluğu  $w$  genişliği  $l$  olan parçadan farklıdır.
2. Yerleştirilecek parçalar, ana parça üzerine dik olarak başka bir deyişle kenarları ana parçanın kenarlarına paralel olarak yerleştirilir. Bu sebeple parçalar ana parça içine yerleştirilirken döndürülemezler.
3. Yerleştirilecek hiçbir parçanın uzunluğu ve genişliği, yerleştirileceği ana parçanın boyutlarını aşamaz.
4. Ana parça ve yerleştirilecek küçük parçaların boyutları tamsayıdır.
5. Ana parça üzerine yerleştirilen küçük parçaların kenar kalınlıkları hiçbir şekilde yer tutmazlar.
6. Her küçük parça, ana parça üzerinde herhangi bir yere yerleştirilebilir. İki parçanın yan yana gelmemesi gibi bir kısıtlama yoktur.

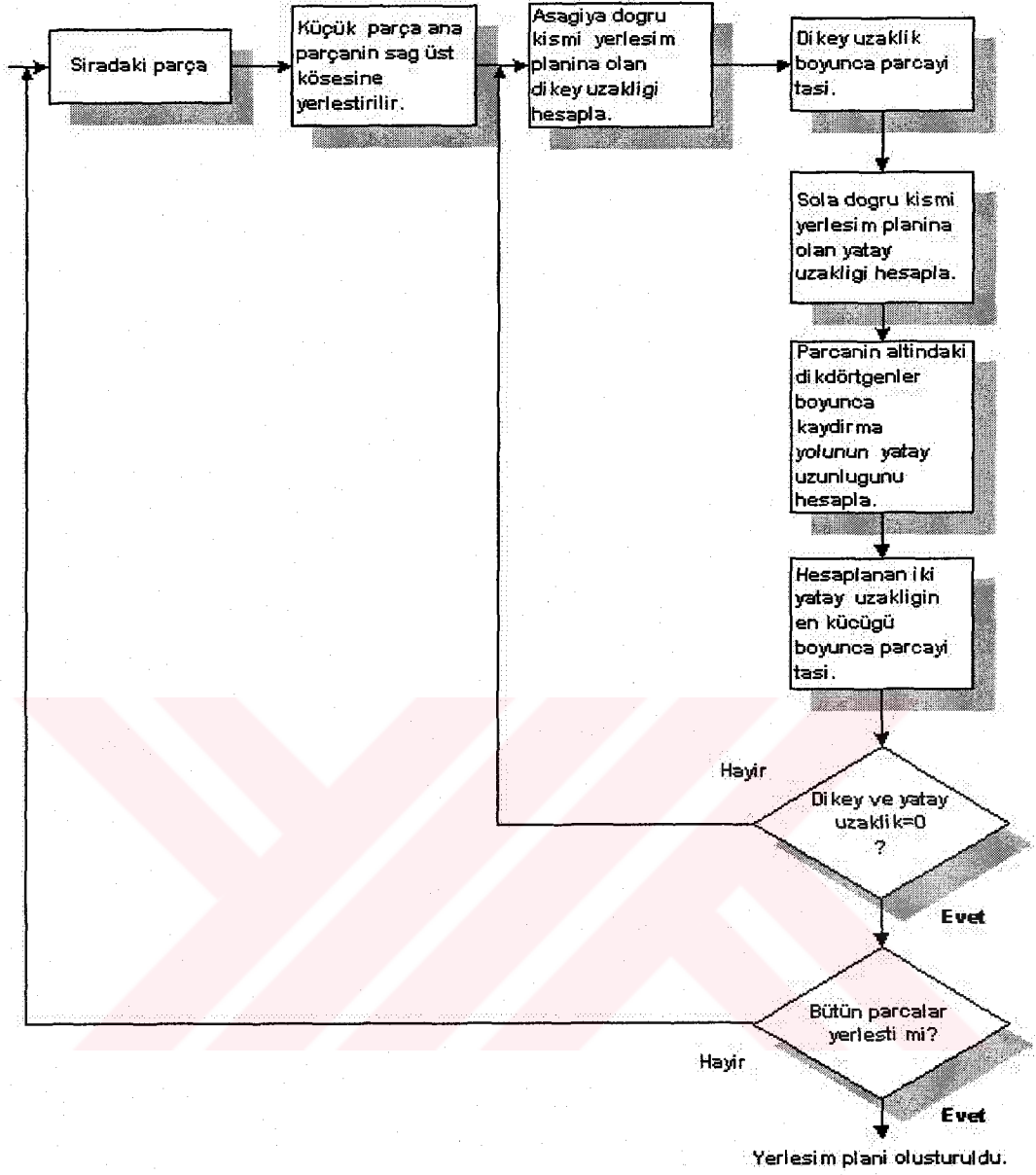
## 5.2. Geliştirilmiş Aşağı Sol Algoritmasının Uygulanışı

Liu ve Teng (1999)'un çalışmalarında kullandığı ve bu çalışmada yerleşim planlarının gerçekleştirilmesi için kullanılan geliştirilmiş AS algoritması önceliği aşağıya kaydırmaya verilmiş bir kaydırma tekniği üzerine kurulmuştur. ASD algoritması, AS ve geliştirilmiş AS algoritmasıyla karşılaştırıldığında daha yoğun bir yerleşim planı oluşturmaktadır. Ancak daha karmaşık ve hesaplanması daha çok zaman almaktadır. Bu nedenlerden dolayı çalışma boyunca geliştirilmiş AS algoritması kullanılmıştır. Şekil 5.1'de akış diyagramıyla gösterilmiş geliştirilmiş AS algoritmasının çalışması aşağıdaki adımlarla sıralanmıştır:

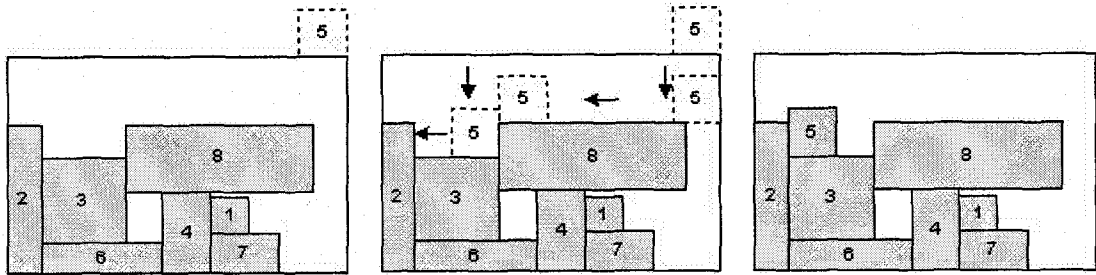
1. Permütasyona ait sıradaki küçük parça yerleştirmenin yapılacağı ana parçanın sağ üst köşesine yerleştirilir.

2. Bu parçanın önceki parçaların yerleştirilmesi suretiyle oluşturulan kısmi yerleşim planına dikey uzaklığı hesaplanır ve bu dikey uzaklık boyunca küçük parça ana parça içinde hareket ettirilir. Böylece küçük parçanın dikey yöndeki hareketi gerçekleştirilmiş olur.
3. Dikey hareketi boyunca mümkün olan en alt konuma yerleştirilen küçük parçanın sol tarafında kalan kısmi yerleşim planına olan yatay uzaklığı ve küçük parçanın altında kalan dikdörtgenler boyunca kaydırma yapabileceği yatay uzaklığı ayrı ayrı hesaplanır.
4. Hesaplanan bu iki yatay uzaklığın en küçüğü boyunca küçük parça hareket ettirilir. Böylece küçük parçanın yatay yöndeki hareketi gerçekleştirilmiş olur.
5. Küçük parçanın bulunduğu konumundan kısmi yerleşim planına dikey ve yatay uzaklıklarının sıfır olup olmadığına bakılır. Eğer herhangi biri sıfır değilse 2. adıma geri dönülerek 2., 3. ve 4. adımların tekrarlanmasıyla dikey ve yatay kaydırma hareketleri yeniden yapılır. Küçük parçanın dikey ve yatay hareketleri, parçanın kısmi yerleşim planına dikey ve yatay uzaklıkları sıfır oluncaya dek tekrarlanır.
6. Parçanın kısmi yerleşim planına dikey ve yatay uzaklığı sıfır ise parça AS koşulunu sağlayan sabit konumuna yerleşmiş demektir. Bu kez permütasyon içindeki bütün parçaların yerleşip yerleşmediği kontrol edilir. Eğer permütasyondaki bütün parçalar için bu yerleştirme işlemi gerçekleştirilmişse yerleşim planı oluşturulmuş demektir. Eğer bütün parçalar yerleştirilmediyse sıradaki parçanın yerleştirilmesi için 1. adıma geri dönülür.

Şekil 5.2’de geliştirilmiş AS algoritması kullanılarak kısmi bir yerleşim planı içinde bir dikdörtgen parçanın yerleştirilmesi adım adım gösterilmektedir.



Şekil 5.1. Geliştirilmiş AS algoritmasının işleyişine dair akış diyagramı.



Şekil 5.2. Kısmi bir yerleşim planı içinde geliştirilmiş AS algoritması kullanılarak dikdörtgen bir parçanın yerleştirilmesi.



### 5.3. Kıırma Genetik Algoritmalar ile Kesme Probleminin Çözümü

En iyi yerleşim planını veren yerleşim sırasını elde etmek için kullandığımız genetik algoritma, geliştirilmiş AS yerleştirme algoritmasıyla birlikte kullanıldığında kırma GA adını alır. Şekil 5.3’de akış diyagramıyla gösterilmiş kırma GA’nın çalışması aşağıdaki adımlarla sıralanmıştır:

1. Öncelikle problem için birden fazla olası çözümü içeren çözüm kümesi, başlangıç nüfusu olarak kullanılmak üzere rasgele oluşturulur. Problem bireyler içinde kodlanırken tamsayı gösterim kullanılır. Nüfustaki her birey, kesme problemleri için yerleşim planında parçaların yerleşim sırasını veren, birer permütasyonla temsil edilir. Her bireyin gen sayısı o kesme problemindeki parça sayısına eşittir.
2. Geliştirilmiş AS yerleştirme algoritması başlangıç nüfusundaki permütasyonları kullanarak Şekil 6.1’de gösterilen akış diyagramındaki adımlarla parçaları ana parça üzerine yerleştirir. Böylece yerleşim planları oluşur. Her birey için oluşturulan yerleşim planı için kullanılmayan alan başka bir deyişle fire Denklem 5.1’de ifade edilen formül kullanılarak elde edilir.

$$\text{Fire} = (\text{Ana parçanın alanı} - \text{Kullanılan alan}) / \text{Ana parçanın alanı} \quad (5.1)$$

Formülde ifade edilen kullanılan alan, sadece yerleşim planının içinde yerleştirilmiş parçaların toplam alanı kadardır. Denklem 5.1’den elde edilen fire değeri ana parçanın kullanılmayan alanının ana parçanın alanına oranını göstermektedir.

3. Her birey için elde edilen fire değeri kullanılarak o nüfusun uygunluk değeri Denklem 5.2’de ifade edilen uygunluk fonksiyonuna ait formül kullanılarak hesaplanır.

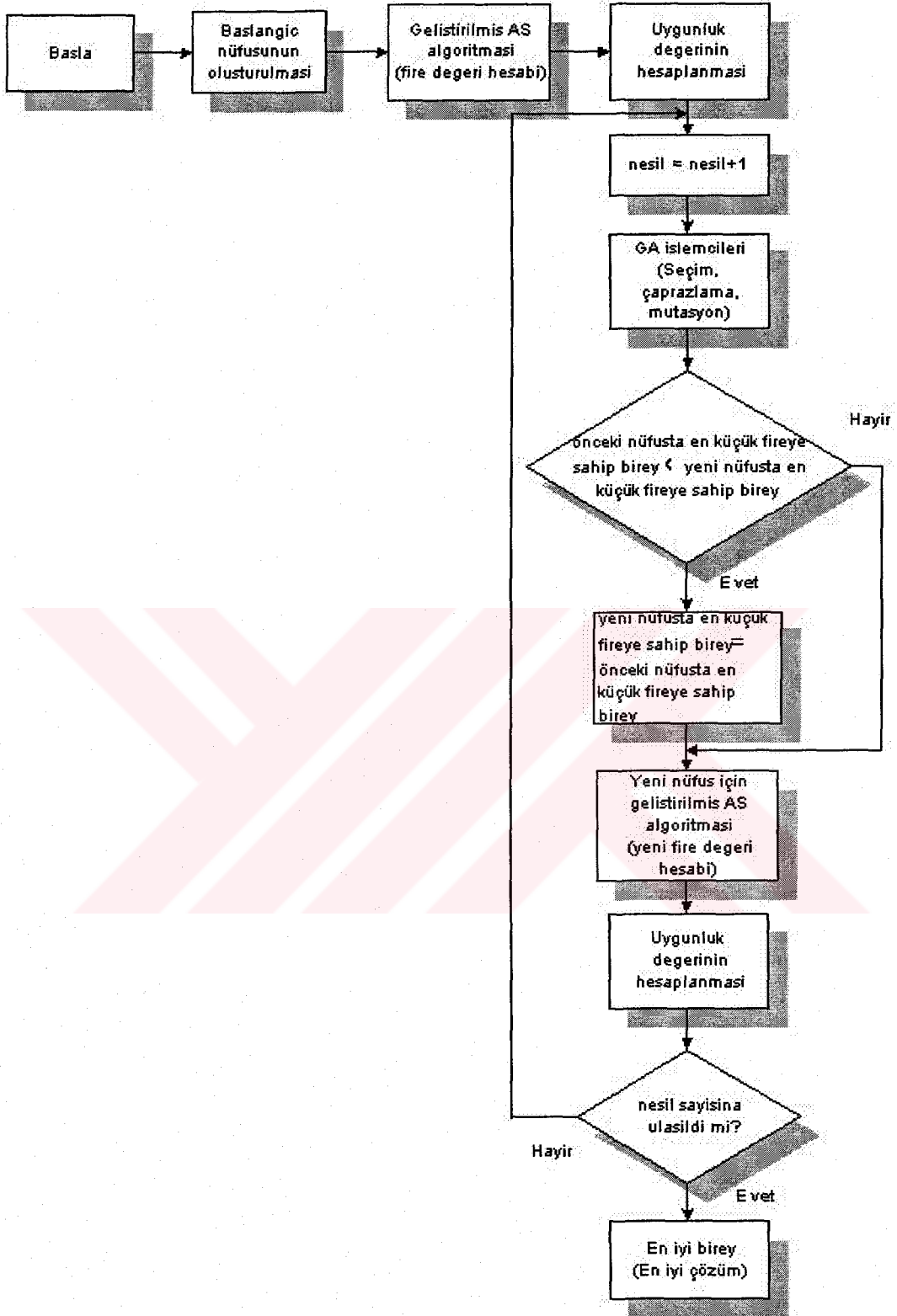
$$\text{Uygunluk} = 1 / (\varepsilon + \text{fire değerleri}) \quad (5.2)$$

Formüldeki uygunluk nüfus içerisindeki bütün bireylerin uygunluk değerlerini içeren bir vektördür. Aynı şekilde fire değerleri nüfus içerisindeki bütün bireylere



ait fire değerlerini içeren bir vektördür. Firenin 0 olarak elde edildiği durumda uygunluk fonksiyonunun bölme hatası vermesini engellemek için bir  $\epsilon$  değeri kullanılmıştır. Bu  $\epsilon$  değeri  $2.2204 \times 10^{-16}$  olarak seçilmiş çok çok küçük bir değerdir.

4. Yeni yerleşim planlarını oluşturacak yeni bireyler elde etmek için nüfus üzerinde seçme, çaprazlama ve mutasyon genetik işlemleri yapılır. Bu çalışmada seçme işlemi için daha çok rulet tekerleği olarak da bilinen oransal seçim kullanılmıştır. Çaprazlama işlemcisi olarak 17 adet parçayı içeren test probleminde SJX çaprazlama tekniği, 29 adet parçayı içeren test probleminde de 6 farklı çaprazlama tekniği (OBX, CX, OX, PMX, UX, SJX) kullanılmıştır. Mutasyon işlemcisi olarak da yer değiştirme mutasyonu kullanılmıştır.
5. Bir önceki nüfus içinde bulunan fire değeri en küçük dolayısıyla uygunluk değeri en büyük birey, yeni nüfus içindeki en küçük fire değerine sahip bireyden daha küçükse nüfus içinde bu bireyler yer değiştirir. Aksi halde bir değişim olmaz.
6. Genetik işlemlerin kullanılmasıyla 4.adımın sonunda yeni bireylerden oluşmuş yeni bir nüfus elde edilir. 2.adımda yapıldığı gibi geliştirilmiş AS yerleştirme algoritması kullanılarak yeni bireyler için yeni yerleşim planları oluşturur. Oluşturulan her yeni yerleşim planı için Denklem 5.1'de ifade edilen formül kullanılarak fire değerleri hesaplanır.
7. Yeni nüfusun elde edilen fire değerleriyle Denklem 5.2'de ifade edilen formül kullanılarak o nüfusa ait uygunluk değerleri hesaplanır.
8. Eniyileme kriteri olarak daha önceden belirlenen nesil sayısına ulaşıp ulaşılmadığı kontrol edilir. Eğer nesil sayısına ulaşıldıysa elde edilen en büyük uygunluk değerine sahip birey problemin çözümü kabul edilir. Eğer nesil sayısına ulaşılmadıysa 4.adıma gidilerek bu adıma kadar olan işlemler tekrarlanır.



Şekil 5.3. Kırma GA'nın çalışmasına ait akış diyagramı.

#### 5.4. Kıırma Benzetiılmiř Tavlama ile Kesme Probleminin Çözümü

Kesme problemi için en iyi yerleřim planını veren yerleřim sırasını elde etmek için kullanılan BT algoritması geliřtirilmiř AS yerleřtirme algoritmasıyla birlikte kullanıldıđında kıırma BT algoritması adını alır. Őekil 5.4'de akıř diyagramıyla gösterilmiř kıırma benzetiilmif tavlama algoritmasının çalıřması ařađıdaki adımlarla sıralanmıřtır:

1. Problem için rasgele olası bir bařlangıç çözümü üretilir. Bu çözüm tamsayı gösterimle kodlanmıřtır. Bařlangıç çözümü kesme problemi için parçaların yerleřim planındaki yerleřim sırasını veren bir permütasyondur, boyutu o kesme problemindeki parça sayısına eřittir.
2. Geliřtirilmiř AS yerleřtirme algoritması kullanılarak bir yerleřim planı oluřturulur. Denklem 5.1'de ifade edilen formül kullanılarak bařlangıç çözümüne ait fire deđeri hesaplanır.
3. Oransal sođutma planı veya Lundy ve Mees sođutma planı kullanılarak sıcaklık güncellemesi yapılır.
4. Permütasyon içinde sondan kaç tane eleman arasında komřuluk hareketinin yapılacađını gösteren vektör uzaklıđı ( $v_u$ ) hesaplanır.  $v_u$  deđeri [2, parça sayısı] arasında deđiřmektedir.
5. Eđer komřuluk hareketi sayısı = 0 ise kaydırma veya yer deđiřtirme komřuluk hareketlerinden biri kullanılarak yeni çözüm üretilir. Bunun için vektör uzaklıđı parça sayısı kadar alınır. Eđer komřuluk hareketi sayısı > 0 ise yeni çözüm aynı Őekilde üretilir ancak vektör uzaklıđı için 4.adımda hesaplanan  $v_u$  deđeri alınır.
6. Üretilen yeni çözüm için geliřtirilmiř AS yerleřtirme algoritması kullanılarak Denklem 5.1'de ifade edilen formülle yeni yerleřim planının fire deđeri hesaplanır.
7. Yeni üretilen çözüm ve bir önceki çözümün fire deđeri arasındaki deđiřim miktarı Denklem 5.3'te ifade edilen formülle hesaplanır.

$$\Delta C = \text{Yeni fire} - \text{Fire} \quad (5.3)$$

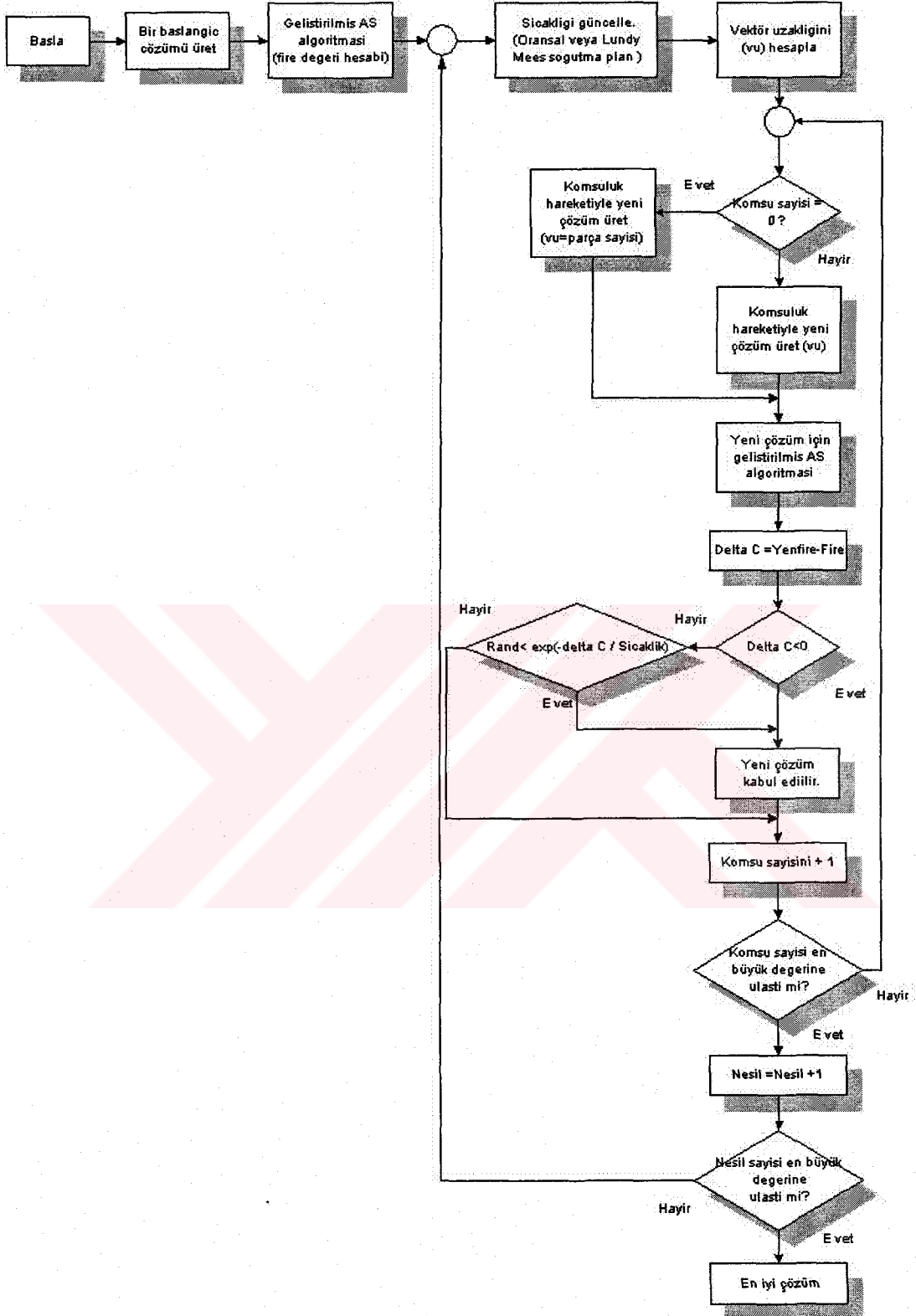
Formülde ifade edilen yeni fire yeni üretilen çözüme, fire bir önceki çözüme ait fire değerleridir.  $\Delta C$  bu iki çözüm arasındaki değişim miktarıdır.

8. Eğer  $\Delta C < 0$  ise yeni çözüm olduğu gibi kabul edilir. Eğer  $\Delta C \geq 0$  ise yeni çözüm Denklem 5.4'te ifade edilen olasılık dahilinde kabul edilir.

$$R < \exp(-\Delta C / \text{sıcaklık}) \quad (5.4)$$

$R$   $[0, 1]$  arasında rasgele bir sayıdır. Eğer Denklem 5.4 sağlanmıyorsa bir sonraki adımda yine bir önceki çözüm kullanılır.

9. Komşuluk hareketi sayısı 1 artırılır ve komşuluk hareketi için kullanılacak sınır değere eşit olup olmadığına bakılır. Eğer eşit değilse 5.adıma gidilir ve yeni bir çözüm üretilerek bu adıma kadar olan işlemler tekrarlanır. Eğer eşitse bir sonraki adımda son öteleme sayısına ulaşıp ulaşılmadığına bakılır. Eğer son öteleme sayısına ulaşıldıysa son çözüm o problem için elde edilen en iyi çözümdür. Eğer ulaşılmadıysa 3. adıma gidilir ve bu adıma kadar olan işlemler tekrarlanır.



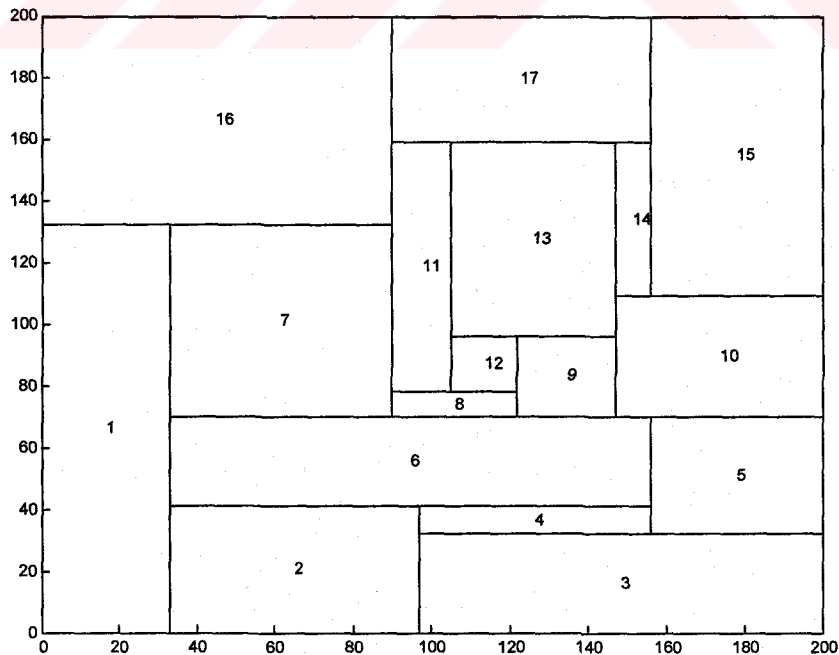
Şekil 5.4. Kırma BT algoritmasının çalışmasına ait akış diyagramı.

## BÖLÜM 6. GENETİK ALGORİTMA ve BENZETİLMİŞ TAVLAMA ile İKİ BOYUTLU GİYOTİNSİZ KESME PROBLEMİNİN ÇÖZÜMÜ

### 6.1. Kıırma Genetik Algorİtmalar Kullanılarak Elde Edilen Sonuların Deęerlendirilmesi

#### 6.1.1. 17 adet paradan oluřan test problemlerine ait deęerlendirmeler

Bu alıřmada, kullanılan 17 adet paradan oluřan test problemlerinin her biri birbirinden farklı dzgn dikdrtgen Őekiller ieren ve giyotinsiz yerleřim planları saęlayan problemlerdir. Bu test problemleri zel olarak seildięinden en iyi özmde sıfır fire elde edilmektedir. Buna rnek olarak Őekil 6.1'de 1.test problemine ait paraların yerleřim planı gsterilmektedir. Kıırma genetik algorİtmalar kullanılarak bu problemlerin özm iin gerekleřtirilen program her problemin olası özm iin farklı olasılıksal kk parametreleriyle 3 kez alıřtırılmıřtır.



Őekil 6.1 1.test problemine ait küçük paraların yerleřim planı.

Bu çalışmada 180 farklı durum için yerleştirme simülasyonu yapılmıştır.

Genetik algortmada kullanılan parametre değerleri:

Nüfus büyüklüğü : 20, 40, 60, 80, 100  
Mutasyon oranı : 0.01, 0.1, 0.3, 0.6, 0.7, 0.9, 1.0  
Çaprazlama işlemcisi : Stefan Jakobs çaprazlama işlemi (permütasyona dayalı)  
Nesil : 1000

#### 6.1.1.1. Nüfus büyüklüğünün etkileri

Beş farklı nüfus büyüklüğü (20, 40, 60, 80, 100) ve 0.7 mutasyon oranı kullanılarak yapılan yerleştirme simülasyonları sonucunda problemlere göre elde edilen en büyük ve en küçük fire değerleri Tablo 6.1'de gösterilmektedir. Tablodan görüldüğü gibi en büyük fire 1.test probleminde, en küçük fire ise 4.test probleminde elde edilmiştir. Firenin küçük olması yerleşim planının en iyi şekilde yapıldığını göstermektedir.

**Tablo 6.1.** Beş test problemi, farklı nüfus büyüklükleri ve sabit 0.7 mutasyon oranıyla yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri.

Problem no	EB fire	EK fire
1	0.1128	0.0579
2	0.0911	0.0589
3	0.1048	0.0658
4	0.0736	0.0256
5	0.0952	0.0561
En kötü/iyi yerleşim planlarına ait fire değerleri	0.1128	0.0256

Tablo 6.2'de farklı nüfus büyüklükleri için her problemde elde edilen en küçük fire değeri ve en büyük uygunluk değeri gösterilmektedir. Farklı nüfus büyüklükleri kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerlerine ait problemlerin en büyük uygunluk ve ortalama uygunluk grafikleri beş problem için sırasıyla Şekil 6.2, Şekil 6.3, Şekil 6.4, Şekil 6.5 ve Şekil 6.6'da gösterilmektedir.



**Tablo 6.2.** Nüfus büyüklüklerine göre problemlerde elde edilen en küçük fire değerleri.

Problem no	Nüfus büyüklüğü	Fire değerleri	EB uygunluk değerleri
1	80	0.0579	17.2786
2	20	0.0589	16.9924
3	80	0.0658	15.1918
4	20	0.0256	38.9864
5	80	0.0561	17.8253

80 nüfus büyüklüğü ve 0.7 mutasyon oranı kullanılarak yapılan çalışmada, P1 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği Şekil 6.2’de gösterilmektedir. Bu yerleştirme simülasyonu 1000 nesil boyunca yapılmıştır. Ancak 650 nesilden sonra değişim görülmemektedir. Tüm nesil boyunca sabit bir eğilim gösteren ortalama uygunluk grafiği nüfus içerisinde birbirinden farklı bireylerin üretilmediğini göstermektedir. Tüm çalışma boyunca elde edilen en büyük uygunluk değeri 17.2786’dır.

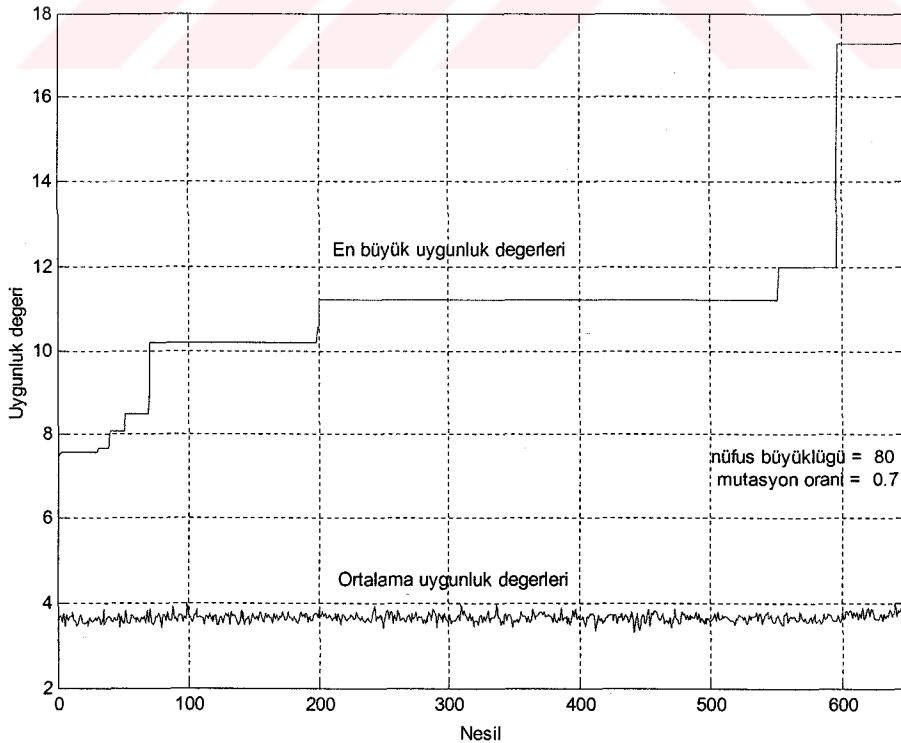
Şekil 6.3’de P2 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği gösterilmektedir. Bu çalışma 20 nüfus büyüklüğü ve 0.7 mutasyon oranı kullanılarak gerçekleştirilmiştir. Yerleştirme simülasyonu 1000 nesil boyunca tekrarlanmıştır. Ortalama uygunluk değerleri yaklaşık olarak en büyük uygunluk değerleri gibi artan bir eğilim göstermektedir. Bu durum nüfus içinde permütasyon hesabının sağlıklı bir şekilde yapıldığını göstermektedir. 1000 nesil sonunda elde edilen en büyük uygunluk değeri 16.9924’dır.

Şekil 6.4’de 80 nüfus büyüklüğü ve 0.7 mutasyon oranı kullanılarak yapılan P3 problemine ait çalışmanın en büyük uygunluk ve ortalama uygunluk grafiği gösterilmektedir. 1000 nesil boyunca tekrarlanan yerleştirme simülasyonu 850 nesilden sonra bir değişim göstermemektedir. Sabit bir eğilim gösteren ortalama uygunluk değerlerine ait grafik nüfus içerisindeki bireylerin yeterince çeşitlilik sağlayamadığını gösterebilir. Bu çalışma sonucunda elde edilen en büyük uygunluk değeri 15.1918’dır.

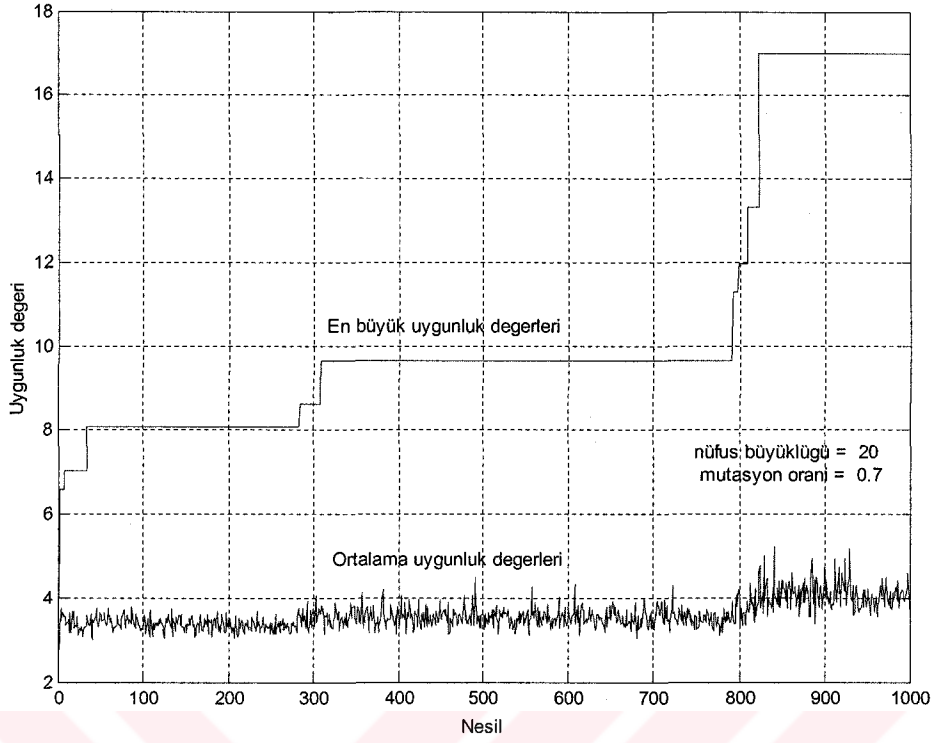
P4 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği Şekil 6.5’te gösterilmektedir. Bu çalışma 20 nüfus büyüklüğü ve 0.7 mutasyon oranı kullanılarak

gerçekleştirilmiştir. Yerleştirme simülasyonu 1000 nesil boyunca tekrarlanmış ancak 450 nesilden sonra uygunluk değerinde bir değişim görülmemiştir. Ortalama uygunluk değeri en büyük uygunluk değeri eğilimiyle yaklaşık olarak aynı eğilimi göstermiştir. Bu durum, permütasyon hesabının sağlıklı bir şekilde yapıldığını ve nüfus içinde birbirinden farklı bireylerin oldukça fazla olduğunu göstermektedir. 1000 nesil boyunca tekrarlanan çalışmada elde edilen en büyük uygunluk değeri 38.9864'dür. Farklı nüfus büyüklükleri ve 0.7 mutasyon oranı kullanılarak yapılan bu yerleştirme simülasyonları sonucunda en büyük uygunluk değeri P4 problemi için yapılan çalışmada elde edilmiştir.

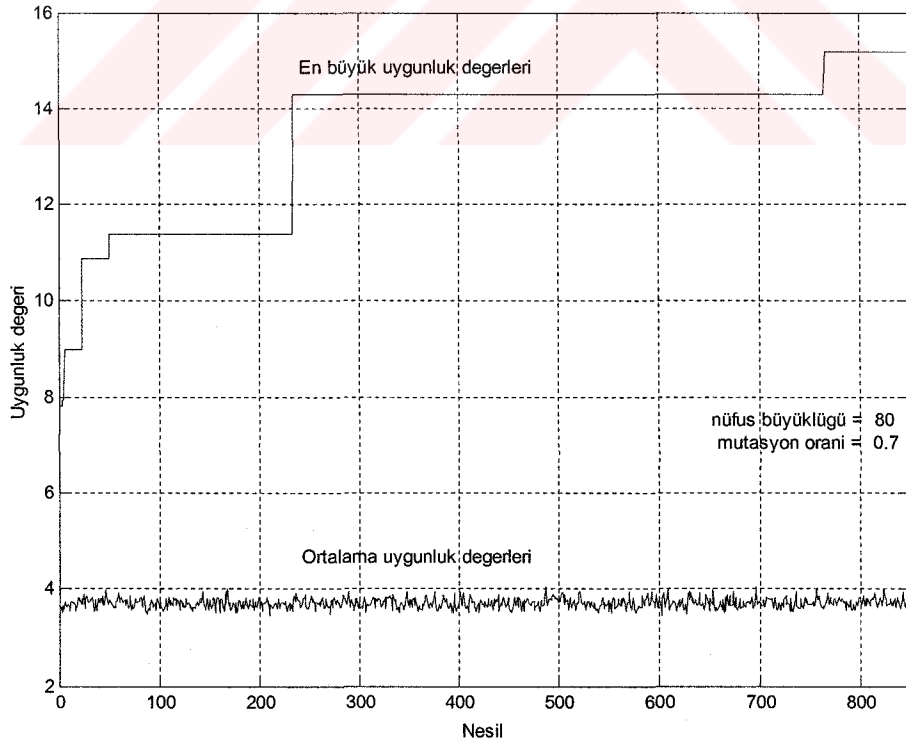
Şekil 6.6'da 80 nüfus büyüklüğü ve 0.7 mutasyon oranı kullanılarak P5 problemi için gerçekleştirilen çalışmaya ait en büyük uygunluk ve ortalama uygunluk grafiği gösterilmektedir. 1000 nesil boyunca tekrarlanan yerleştirme simülasyonu 550 nesilden sonra bir değişim göstermemiştir. Sabit bir eğilim gösteren ortalama uygunluk değerleri, nüfus içinde birbirine benzeyen bireylerin yoğunlukta olduğunu ve dolayısıyla permütasyon hesabının sağlıklı yapılamadığını göstermektedir. 1000 nesil boyunca elde edilen en büyük uygunluk değeri 17.8253'dir.



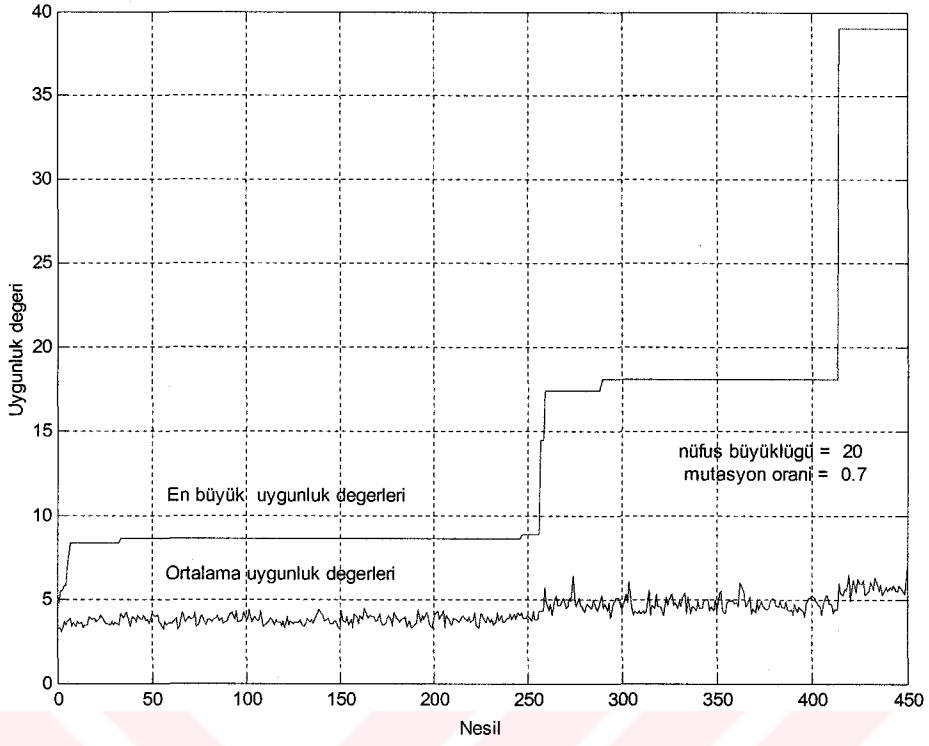
Şekil 6.2. P1 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



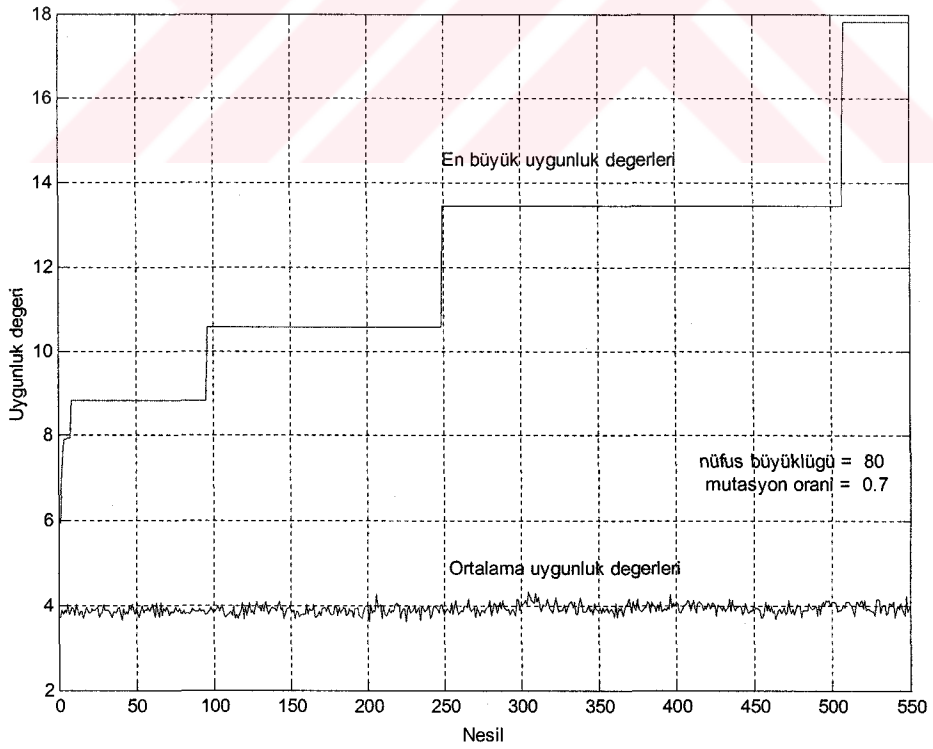
Şekil 6.3. P2 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



Şekil 6.4. P3 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.

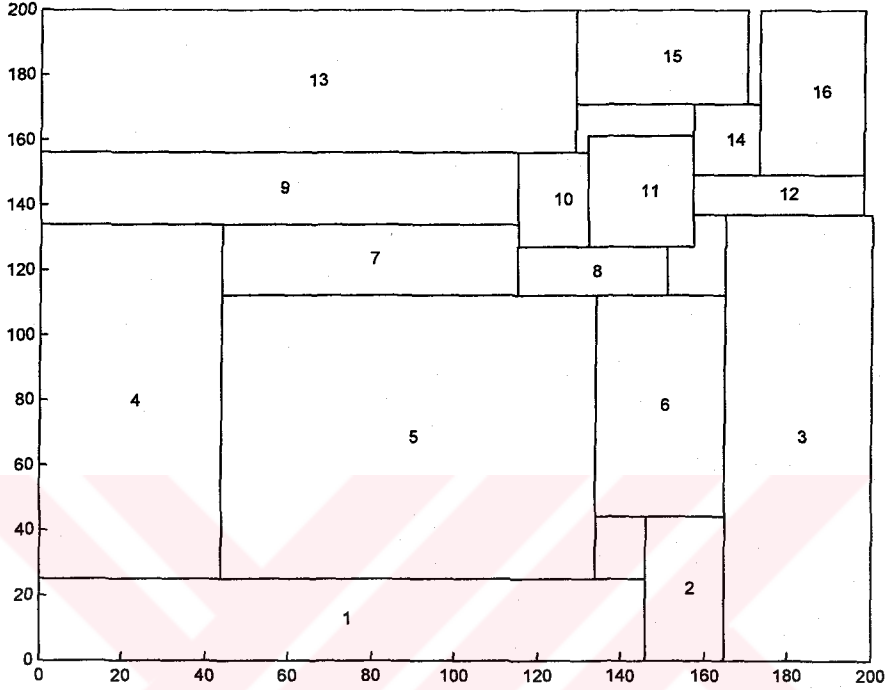


Şekil 6.5. P4 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



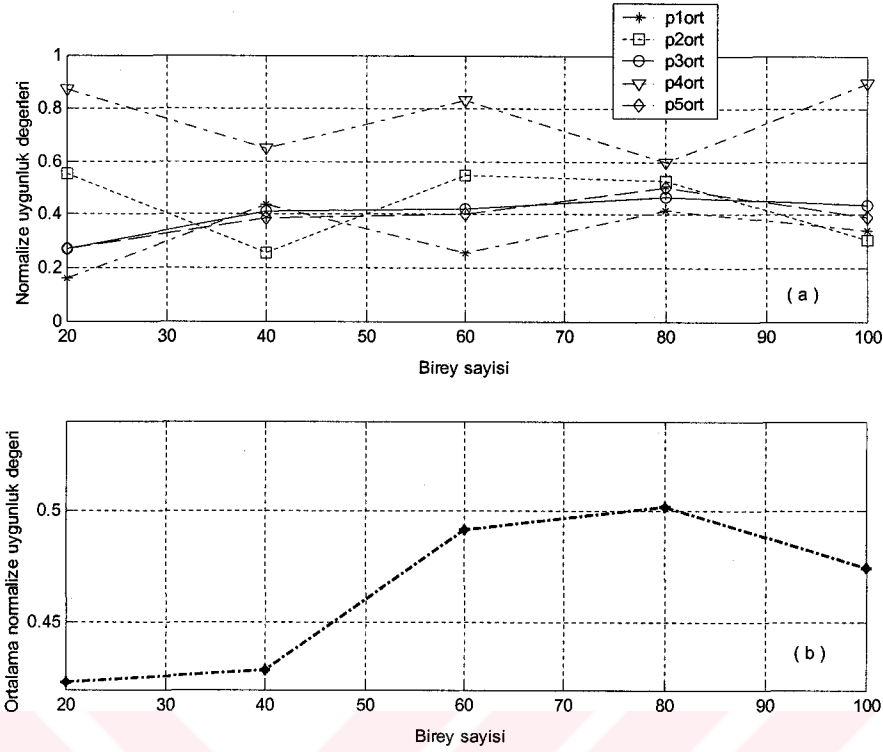
Şekil 6.6. P5 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.

Farklı nüfus büyüklükleri ve 0.7 mutasyon oranı kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı Şekil 6.7'de gösterilmektedir.



Şekil 6.7. Farklı nüfus büyüklükleri ve 0.7 mutasyon oranı kullanılarak elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı.

17 parçanın yerleştirilmesi için yapılan bu çalışmanın sonuçları karşılaştırılarak en büyük ve en küçük fire değerleri bulunmuştur. Bu değerler kullanılarak ortalama fire değerleri normalize edilmiştir. Tablo 6.1'de de görüldüğü gibi en büyük fire değeri 0.1128, en küçük fire değeri 0.0256'dır. En büyük fire değeri (0.1128) sıfır, en küçük fire değeri (0.0256) bir alınarak normalizasyon yapılmıştır. Normalize edilmiş uygunluk değerleriyle fire değerleri arasında ters orantı vardır. Başka bir deyişle, normalize edilmiş uygunluk değeri arttıkça fire azalmakta, azaldıkça fire artmaktadır.



**Şekil 6.8.** Farklı nüfus büyüklükleriyle elde edilen normalize edilmiş uygunluk değerleri.

Farklı nüfus büyüklüklerinde elde edilmiş fire değerlerinin normalize uygunluk değerleri Şekil 6.8 (a)'da beş farklı problem için gösterilmektedir. Şekil 6.8 (b)'de de Şekil 6.8 (a)'daki uygunluk değerlerinin ortalamaları alınarak, ortalama normalize uygunluk değeri elde edilmiştir. Şekil 6.8 (b)'den görüldüğü gibi 20-40 nüfus büyüklükleri arasında problemin çözümü için yetersiz kalan nüfus, 40-60 nüfus büyüklükleri arasında yükselen bir eğilim göstermektedir. Nüfus, 60-80 nüfus büyüklükleri arasında 20 nüfus büyüklüğünde elde edilen uygunluk değerine göre yaklaşık %10'luk bir artış göstermektedir. Bu sonuçlara göre nüfus büyüklüğü arttıkça normalize uygunluk değerinin arttığı, buna karşılık gelen fire değerinin azaldığı görülmektedir.

### 6.1.1.2. Mutasyon oranının etkileri

Yedi farklı mutasyon oranı (0.01, 0.1, 0.3, 0.6, 0.7, 0.9, 1.0) ve 80 nüfus büyüklüğü kullanılarak yapılan yerleştirme simülasyonları sonucunda bel problem için elde edilen en büyük ve en küçük fire değerleri Tablo 6.3'de gösterilmektedir. Tablodan

görüldüğü gibi en büyük fire yine 1.test probleminde, en küçük fire ise 4.test probleminde elde edilmiştir. Bu yerleştirme simülasyonları 1.test probleminin zor bir problem olduğunu, 4.test probleminin kolay bir problem olduğunu göstermektedir.

**Tablo 6.3.** Beş test problemi, 80 nüfus büyüklüğü ve farklı mutasyon oranlarıyla yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri.

Problem no	EB fire	EK fire
1	0.0936	0.0466
2	0.0833	0.0342
3	0.0888	0.0319
4	0.0677	0.0297
5	0.0887	0.0471
<b>En kötü/iyi yerleşim planlarına ait fire değerleri</b>	<b>0.0936</b>	<b>0.0297</b>

Tablo 6.4'de her problemde elde edilen en küçük fire değerleri gösterilmektedir. Farklı mutasyon oranları kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerlerine ait beş problemin en büyük uygunluk ve ortalama uygunluk grafikleri sırasıyla Şekil 6.9, Şekil 6.10, Şekil 6.11, Şekil 6.12 ve Şekil 6.13'de gösterilmektedir.

**Tablo 6.4.** Farklı mutasyon oranlarına göre problemlerde elde edilen en küçük fire değerleri.

Problem no	Mutasyon oranı	Fire değerleri	EB uygunluk değerleri
1	0.9	0.0466	21.4477
2	0.01	0.0342	29.2398
3	0.6	0.0319	31.3725
4	0.9	0.0297	33.6417
5	0.3	0.0471	21.2540

P1 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği Şekil 6.9'da gösterilmektedir. Bu çalışmada 80 nüfus büyüklüğü ve 0.9 mutasyon oranı kullanılmıştır. Yerleştirme simülasyonu 1000 nesil boyunca yapılmış olmasına rağmen 300 nesilden sonra bir değişim görülmemiştir. Tüm çalışma boyunca elde edilen en büyük uygunluk değeri 21.4477'dir.

Şekil 6.10'da 80 nüfus büyüklüğü ve 0.01 mutasyon oranı kullanılarak yapılan P2 problemine ait çalışmanın en büyük uygunluk ve ortalama uygunluk grafiği



gösterilmektedir. Yerleştirme simülasyonu 1000 nesil boyunca tekrarlanmıştır. Ancak 700 nesilden sonra bir değişim görülmemiştir. 1000 nesil sonunda elde edilen en büyük uygunluk değeri 29.2398'dir.

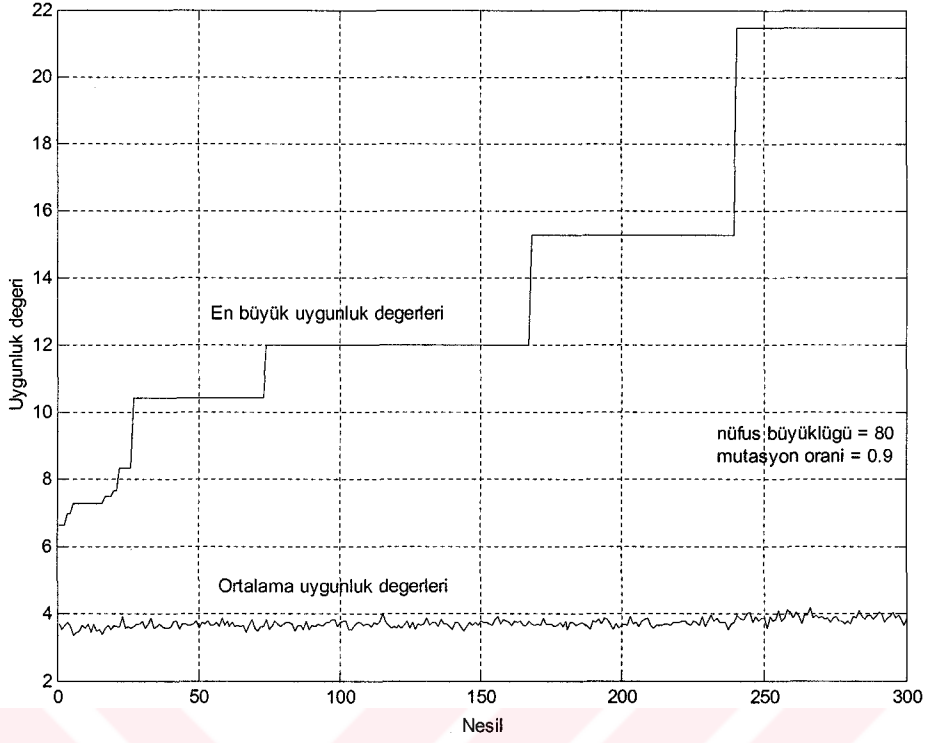
Şekil 6.11'de P3 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği gösterilmektedir. Bu çalışmada 80 nüfus büyüklüğü ve 0.6 mutasyon oranı kullanılmıştır. Yerleştirme simülasyonları 1000 nesil boyunca tekrarlanmıştır. Ancak 900 nesilden sonra bir değişim görülmemiştir. Bu çalışma sonucunda elde edilen en büyük uygunluk değeri 31.3725'dir.

80 nüfus büyüklüğü ve 0.9 mutasyon oranı kullanılarak gerçekleştirilen yerleştirme simülasyonu sonucunda P4 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği Şekil 6.12'de gösterilmektedir. Yerleştirme simülasyonu 1000 nesil boyunca tekrarlanmış ancak 500 nesilden sonra uygunluk değerlerinde bir değişim görülmemiştir. Bu çalışmada elde edilen en büyük uygunluk değeri 33.6417'dir.

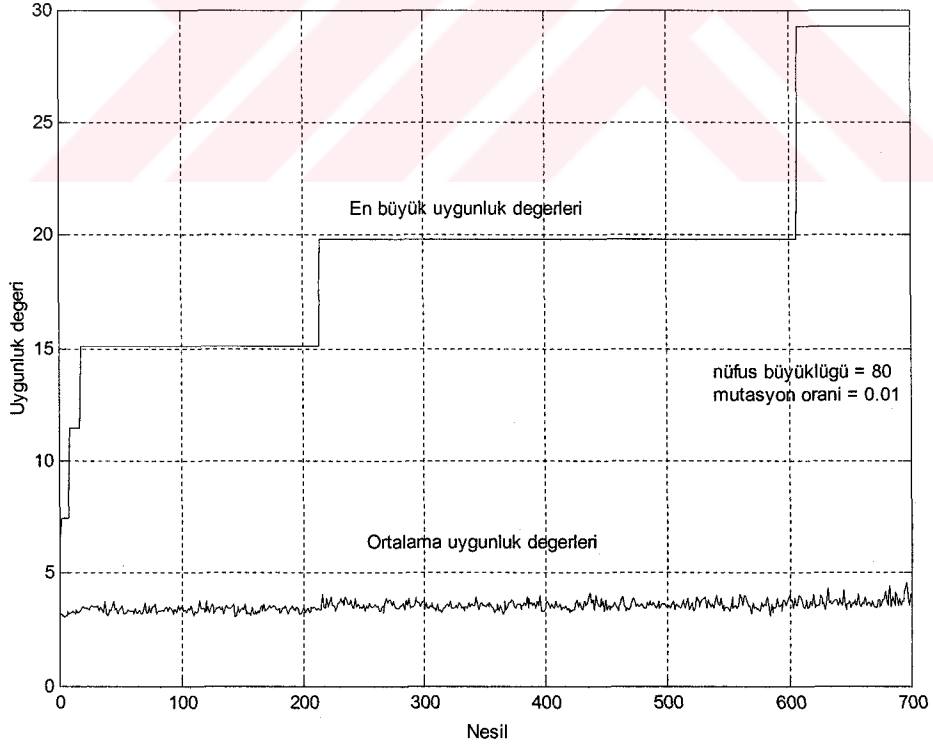
Şekil 6.13'de P5 problemine ait en büyük uygunluk ve ortalama uygunluk grafiği gösterilmektedir. Bu çalışmada 80 nüfus büyüklüğü ve 0.3 mutasyon oranı kullanılmıştır. 1000 nesil boyunca tekrarlanan yerleştirme simülasyonu 900 nesilden sonra değişim göstermemiştir. Bu çalışma sonucunda elde edilen en büyük uygunluk değeri 21.2540'dir.

Farklı mutasyon oranları kullanılarak yapılan bu çalışma içinde genel olarak ortalama uygunluk değerleri sabit bir eğilim göstermektedir. Bu sonuca göre mutasyon işleminin nüfusun genel yapısını değiştirebilecek bir güce sahip olmadığı görülmektedir.

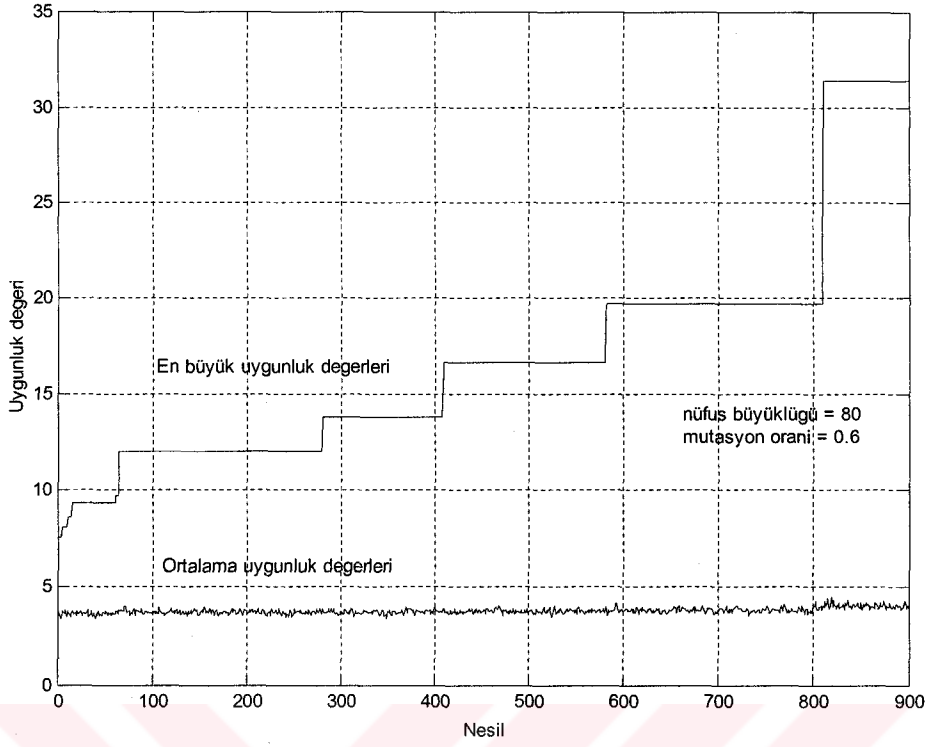
80 nüfus büyüklüğü ve farklı mutasyon oranları kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı Şekil 6.14'de gösterilmektedir.



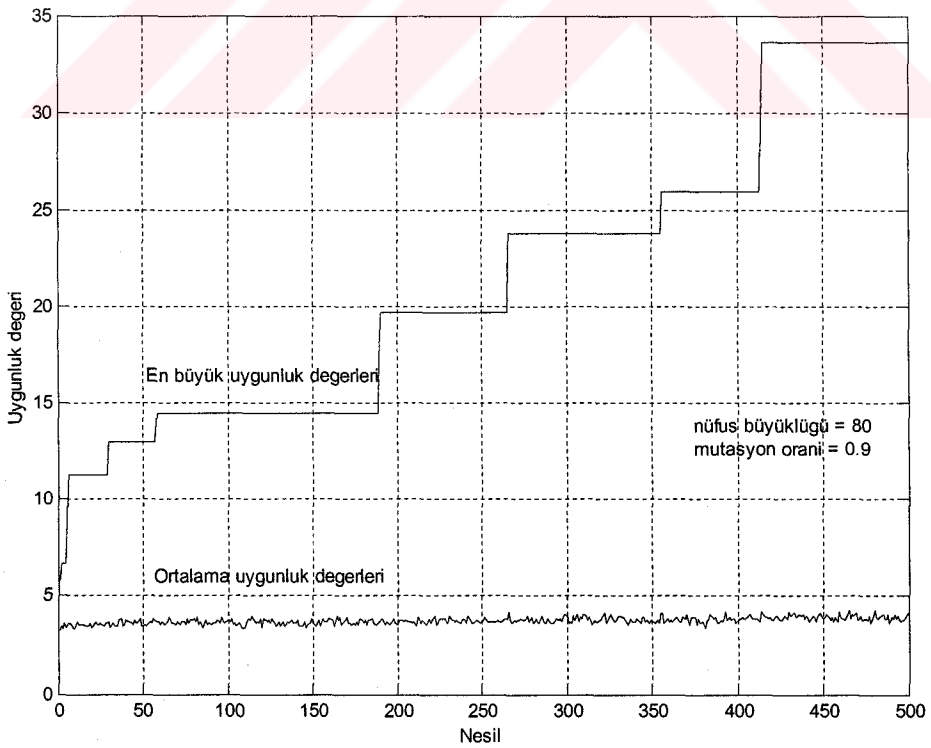
Şekil 6.9. Farklı mutasyon değerleri için elde edilen P1 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



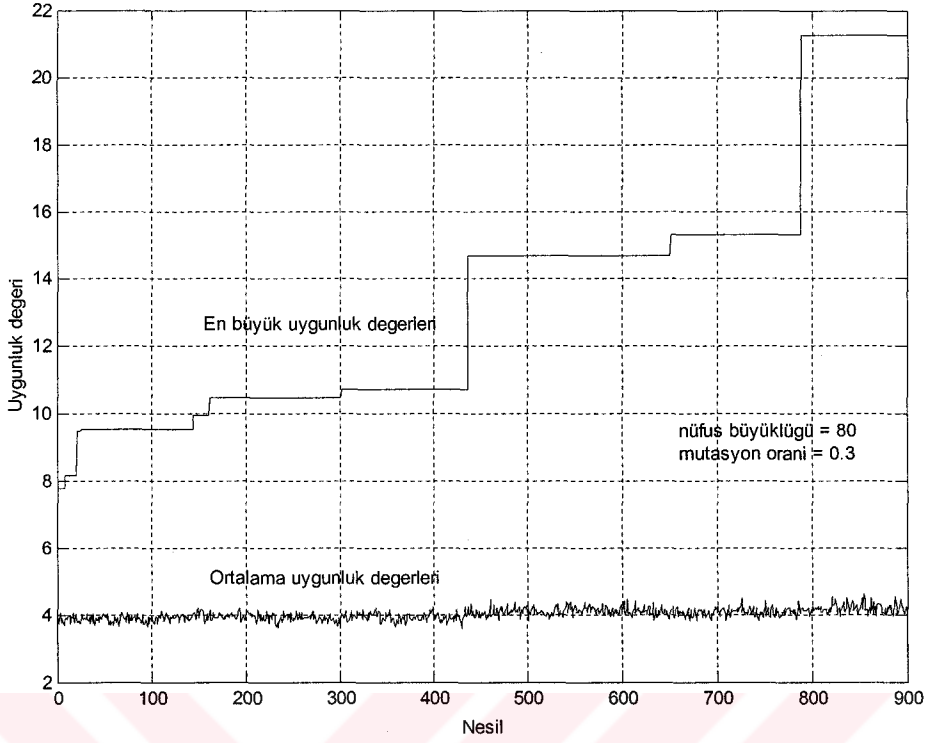
Şekil 6.10. Farklı mutasyon değerleri için elde edilen P2 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



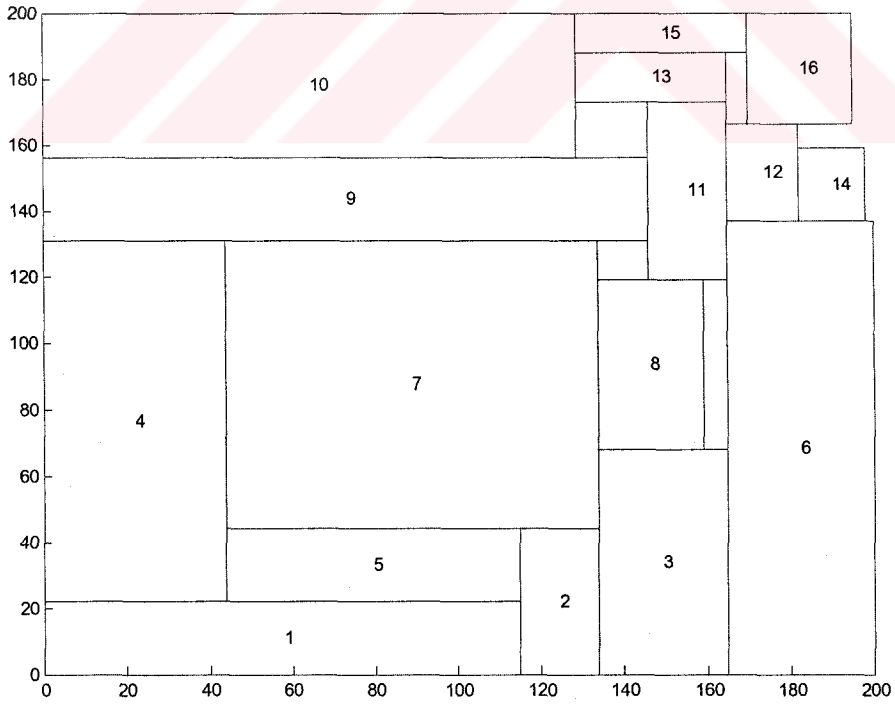
Şekil 6.11. Farklı mutasyon değerleri için elde edilen P3 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



Şekil 6.12. Farklı mutasyon değerleri için elde edilen P4 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



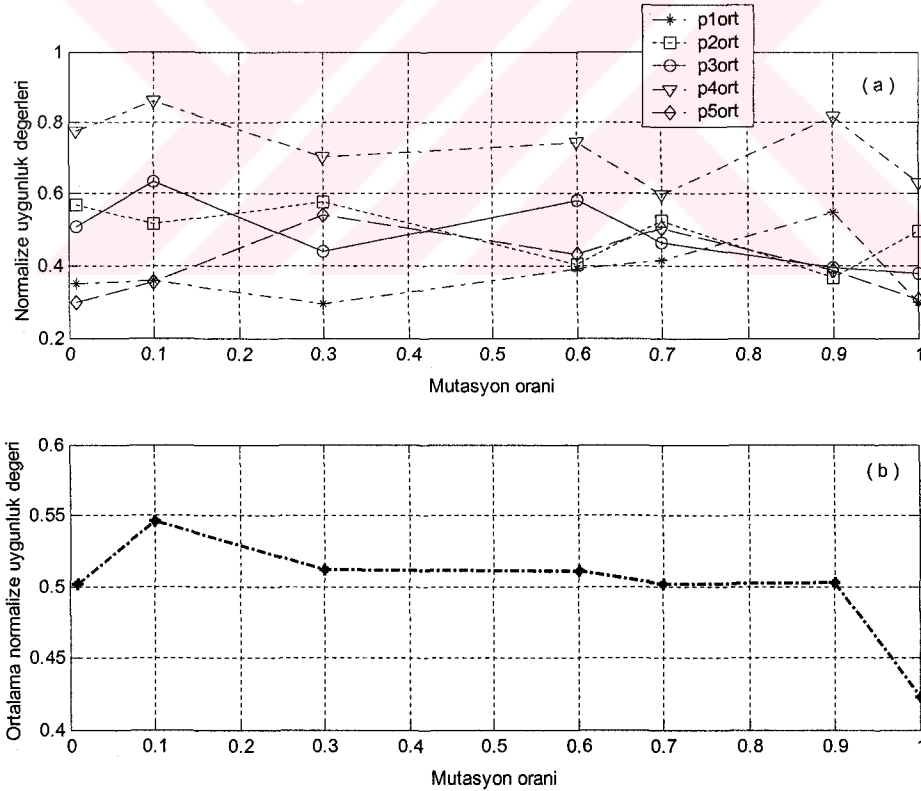
**Şekil 6.13.** Farklı mutasyon değerleri için elde edilen P5 problemine ait en büyük uygunluk ve ortalama uygunluk değerleri.



**Şekil 6.14.** 80 nüfus büyüklüğü ve farklı mutasyon oranları kullanılarak yapılan bu çalışmada elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı

17 adet parçanın yerleştirilmesi için değişik mutasyon oranları kullanılarak yapılan bu çalışmada da ortalama fire değerleri, en büyük fire değeri 0.1128 ve en küçük fire değeri 0.0256 alınarak normalize edilmiştir.

Şekil 6.15 (a)'da beş değişik problem için farklı mutasyon değerlerinde elde edilmiş fire değerlerinin normalize uygunluk değerleri gösterilmektedir. Şekil 6.15 (b)'de de Şekil 6.15 (a)'da elde edilen uygunluk değerlerinin ortalamaları alınarak, ortalama normalize uygunluk değeri elde edilmiştir. Şekil 6.15 (b)'de, en fazla değişim 0.1 mutasyon değerinde gerçekleşmiştir. 0.3–0.9 mutasyon oranları arasında herhangi bir değişim görülmemektedir. Fakat genel eğilim çok az da olsa; mutasyon oranı arttıkça normalize edilmiş uygunluk değerinin azaldığını, buna karşılık gelen fire değerinin arttığını göstermektedir. Dolayısıyla mutasyon işleminin permütasyona dayalı GA'ın ürettiği çözümler üzerinde fazla etkili olmadığı görülmektedir.



Şekil 6.15. Farklı mutasyon oranlarıyla elde edilen normalize edilmiş uygunluk değerleri.

### 6.1.2. 29 adet parçanın yerleşimi için 6 farklı çaprazlama tekniği kullanılarak gerçekleştirilen test problemine ait değerlendirmeler

Bu çalışmada kullanılan test problemi 29 adet birbirinden farklı düzgün dikdörtgen parçalardan oluşmaktadır. Farklı çaprazlama teknikleriyle birlikte kırma GA kullanılarak bu test problemine çözüm aranmıştır. Bu test problemi özel olarak seçildiğinden en iyi çözüm sıfır firenin elde edildiği çözümdür.

Genetik algoritmalarda kullanılan parametre değerleri:

Nüfus büyüklüğü	: 80
Nesil	: 1000
Mutasyon oranı	: 0.7

Çaprazlama işlemcileri: En küçük firenin elde edildiği teknikten başlayarak sırasıyla:

- Sıralamaya dayalı çaprazlama (Order based crossover - OBX),
- Devirli çaprazlama (Cycle crossover – CX),
- Sıralı çaprazlama (Order crossover- OX),
- Kısmi eşleştirmeli çaprazlama (Partially matched crossover- PMX),
- Düzenli çaprazlama (Uniform crossover – UX),
- Stefan Jakobs çaprazlama (Stefan Jakobs crossover – SJX).

Bir önceki çalışmada, mutasyon oranı artışlarının problemin çözümü üzerinde etkili olmadığı görülmüştür. Buna karşılık nüfus büyüklüğü artışlarıyla daha iyi çözümlere ulaşılmaktadır. Bu nedenle bu çalışmada mutasyon oranı 0.7, nüfus büyüklüğü 80 olarak kabul edilmiştir.

#### 6.1.2.1. Farklı çaprazlama tekniklerinin etkileri

Bölüm 6.1.2’de sözü edilen genetik algoritma parametreleri kullanılarak 29 adet parçanın yerleşimi için yapılan çalışmalarda 6 farklı çaprazlama tekniğine göre elde edilmiş fire değerleri Tablo 6.5’de gösterilmektedir. Tablodan da görüldüğü gibi en küçük fire OBX tekniği kullanıldığında, en büyük fire SJX tekniği kullanıldığında,

elde edilmiştir. Firenin en küçük olması yerleşim planının en iyi şekilde yapıldığını göstermektedir. Aynı problem için sadece çaprazlama tekniklerinin değiştirilmesiyle yapılan her çalışmada birbirinden farklı fire değerleri elde edilmiştir. Tablodaki diğer çaprazlama teknikleri için elde edilen fire değerlerine bakıldığında genetik algoritmalarda çaprazlama işlemcisinin çözüm üzerinde önemli derecede etkili olduğu görülmektedir.

**Tablo 6.5.** 6 farklı çaprazlama tekniği için yapılmış çalışmalar sonucunda elde edilen fire değerleri.

Çalışma no	Çaprazlama teknikleri	Fire değerleri	EB uygunluk değerleri
1	OBX	0.0598	16.7224
2	CX	0.0642	15.5824
3	OX	0.0740	13.5227
4	PMX	0.0762	13.1234
5	UX	0.0910	10.9890
6	SJX	0.0992	10.0781

OBX ve SJX teknikleri kullanılarak yapılmış çalışmalardan elde edilen uygunluk değerleri için bazı örnekleme adımlarındaki olasılıksal değerler Tablo 6.6'da gösterilmektedir. Ek-C'de diğer çaprazlama teknikleri için yapılmış çalışmalardan elde edilen uygunluk değerlerine ait olasılıksal değerler bir tablo halinde verilmiştir.



**Tablo 6.6.** OBX ve SJX teknikleri kullanılarak elde edilen uygunluk değerlerine ait olasılıksal değerler.

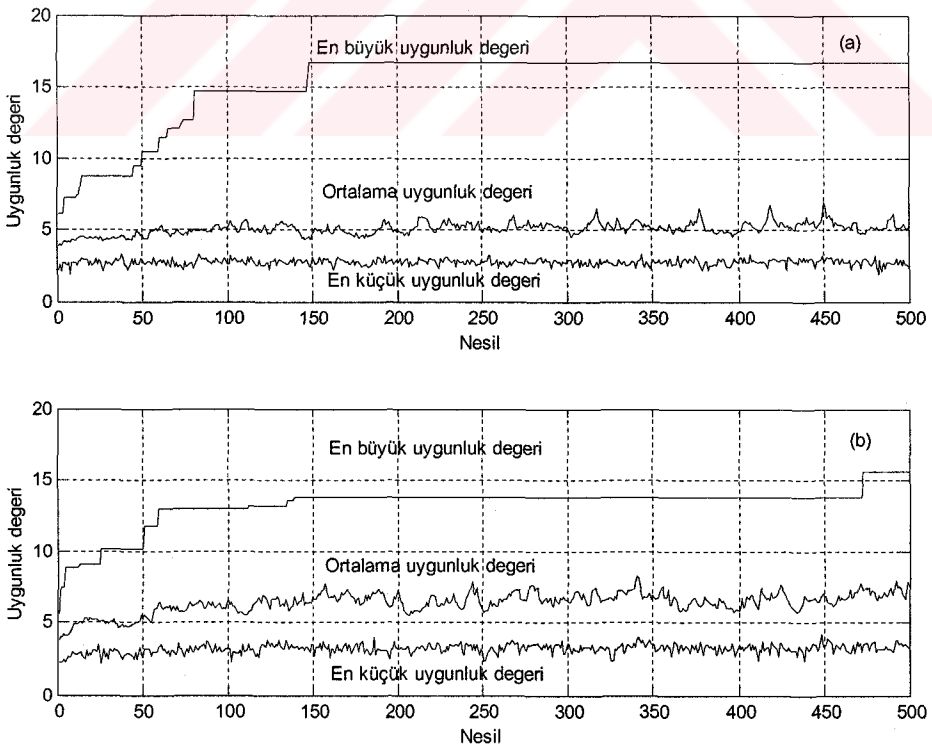
No	Çaprazlama teknikleri	Nesil no	Uygunluk değerlerine ait olasılıksal değerler					
			Aritmetik ortalama	Medyan	Geometrik ortalama	Standart sapma	EB değer	EK değer
1	OBX	1	3,8502	3,6746	3,7784	0,7633	6,0698	2,1674
2	OBX	100	5,1112	4,4472	4,7629	2,3518	14,6843	2,5332
3	OBX	200	4,7759	4,4682	4,5419	1,9611	16,7224	2,8821
4	OBX	300	5,1914	4,9460	4,9306	1,9822	16,7224	2,6508
5	OBX	400	4,8772	4,1258	4,5538	2,3809	16,7224	2,6513
6	OBX	500	5,2050	4,5132	4,8517	2,3501	16,7224	2,3770
7	OBX	600	5,5520	5,1494	5,1161	2,8012	16,7224	2,7012
8	OBX	700	5,4375	5,0697	5,1243	2,1983	16,7224	2,6804
9	OBX	800	7,0810	5,3211	6,1763	4,0552	16,7224	2,8129
10	OBX	900	5,0124	4,4746	4,6806	2,4559	16,7224	2,7461
11	OBX	1000	4,9849	4,7308	4,7653	1,8333	16,7224	2,3369
12	SJX	1	4,0224	3,9926	3,9244	0,9173	7,4724	2,1324
13	SJX	100	4,0420	3,9390	3,9478	0,9444	9,0888	2,1755
14	SJX	200	3,8810	3,6234	3,7944	0,9322	9,0888	2,5259
15	SJX	300	4,0643	3,8560	3,9311	1,1348	9,3262	2,2231
16	SJX	400	3,8250	3,7152	3,7303	0,9926	10,0781	2,3802
17	SJX	500	3,9130	3,7817	3,8114	1,0299	10,0781	2,4480
18	SJX	600	4,0759	3,9313	3,9673	1,0644	10,0781	2,5487
19	SJX	700	4,1131	3,8954	3,9926	1,1183	10,0781	2,4392
20	SJX	800	3,9283	3,7856	3,8087	1,0937	10,0781	2,4066
21	SJX	900	3,9259	3,6839	3,8119	1,0852	10,0781	2,2873
22	SJX	1000	4,0701	3,7880	3,9542	1,0985	10,0781	2,4350

OBX tekniği için hesaplanmış standart sapma değerleri, SJX tekniği için hesaplananlara göre biraz daha büyük değerler arasında değişmektedir. Ayrıca tablodan bakıldığında OBX tekniği için uygunlukların aritmetik ortalama değerlerinin, SJX tekniği için hesaplanan değerlerden daha büyük olduğu görülmektedir. OBX tekniği kullanılarak yapılan çalışmada uygunluk değerlerinin nesil boyunca daha iyiye gittiği görülmektedir. Bu da OBX tekniğinin nüfus içerisinde farklı bireyler üretmede başarılı olduğunu gösterir.

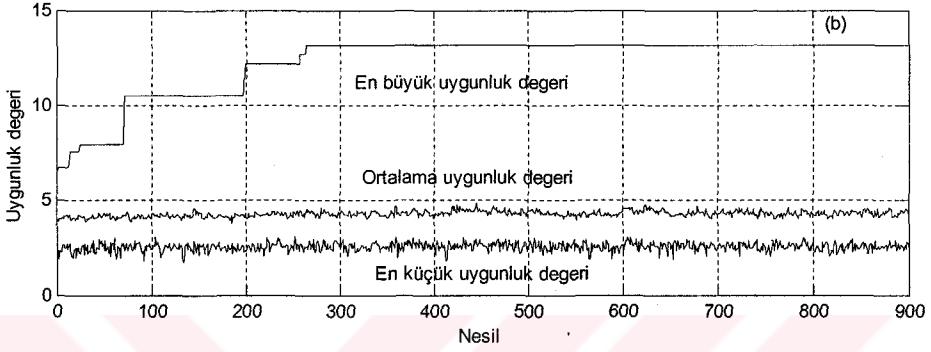
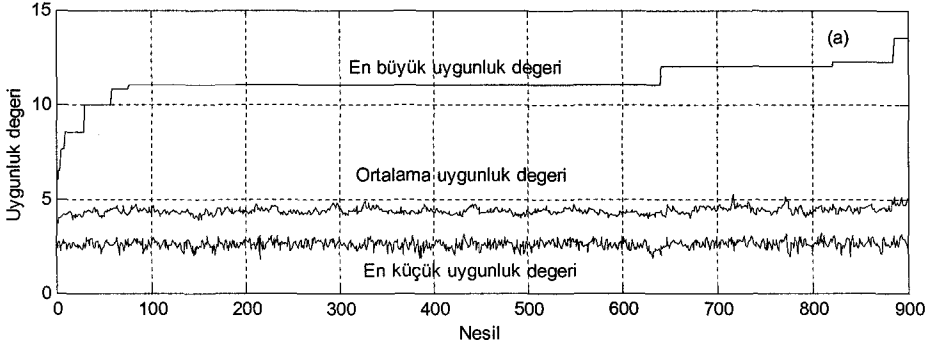
OBX ve CX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk grafiği Şekil 6.16'da, OX ve PMX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk grafiği Şekil 6.17'de, UX ve SJX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk grafiği Şekil 6.18'de gösterilmektedir. Bu yerleştirme simülasyonları 1000 nesil boyunca yapılmıştır. Fakat Şekil 6.16'daki grafikler 500, Şekil 6.17'deki grafikler 900, Şekil 6.18'deki grafikler 700'den sonra değişim göstermemektedirler. Şekil 6.16

(a)'da OBX tekniğinin kullanılmasıyla elde edilen uygunluk değerleri için çizilmiş grafik görülmektedir. Bu problemde genetik algoritma permütasyon hesabı için kullanıldığından nüfus içerisinde birer permütasyonla temsil edilen bireyler birbirinden ne kadar farklı olursa o oranda farklı bireyler üretilir. Nüfus içerisindeki bu çeşitliliğin bir nedeni olarak Şekil 6.16 (a)'dan görüldüğü gibi OBX tekniği için ortalama uygunluk değeri nesil ilerledikçe değişim içerisinde. Şekil 6.18 (b)'de en kötü fire değerine sahip SJX tekniğinin kullanılmasıyla elde edilen uygunluk değerleri için çizilmiş grafik görülmektedir. Bu çalışmadaki ortalama uygunluk değeri OBX tekniği ile yapılan çalışmanın aksine bir değişim sergilememektedir. Bu da nüfus içerisinde farklı bireyler elde edilemediğini dolayısıyla SJX tekniğinin verimli bir çaprazlama olmadığını gösterir.

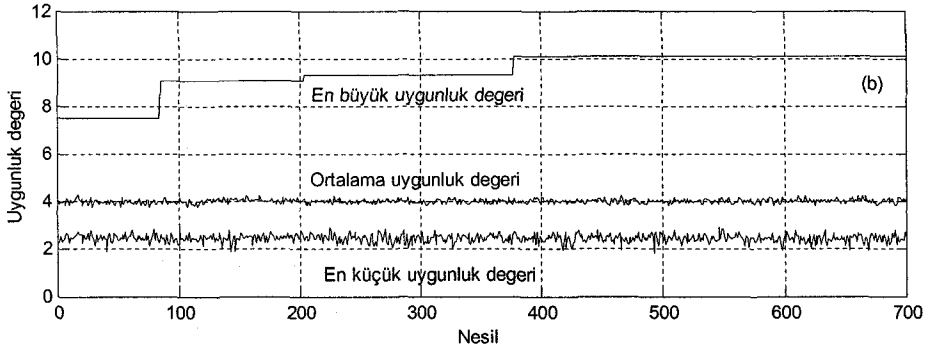
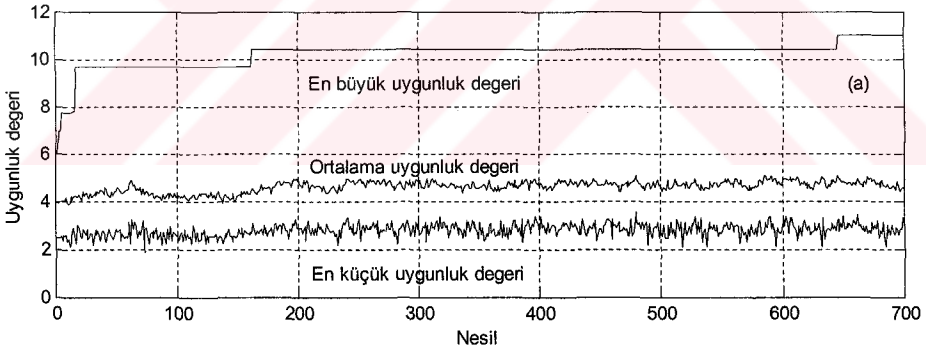
Şekil 6.16 (b)'de görülen ikinci en iyi fire değerine sahip CX tekniğinin kullanılmasıyla elde edilen ortalama uygunluk değerinin de kullanılan diğer çaprazlama tekniklerinin ortalama uygunluk değerlerinden daha iyi bir değişim sergilediği görülmektedir.



**Şekil 6.16. a)** OBX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. **b)** CX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri

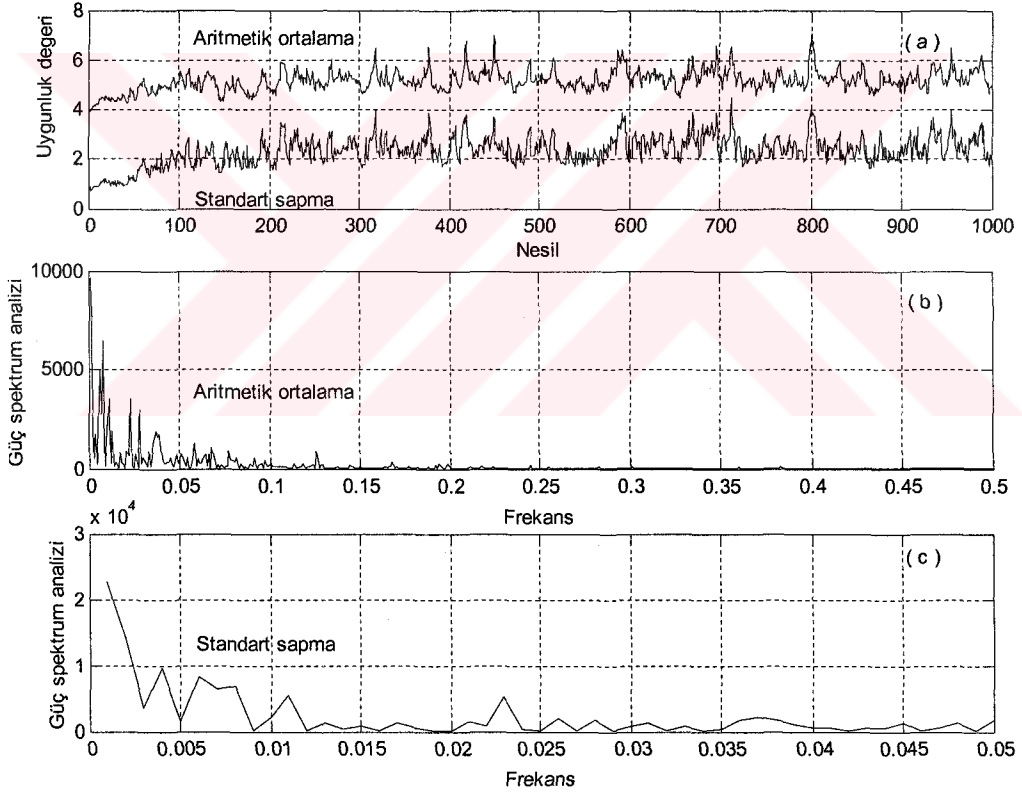


**Şekil 6. 17 a)** OX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. **b)** PMX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri

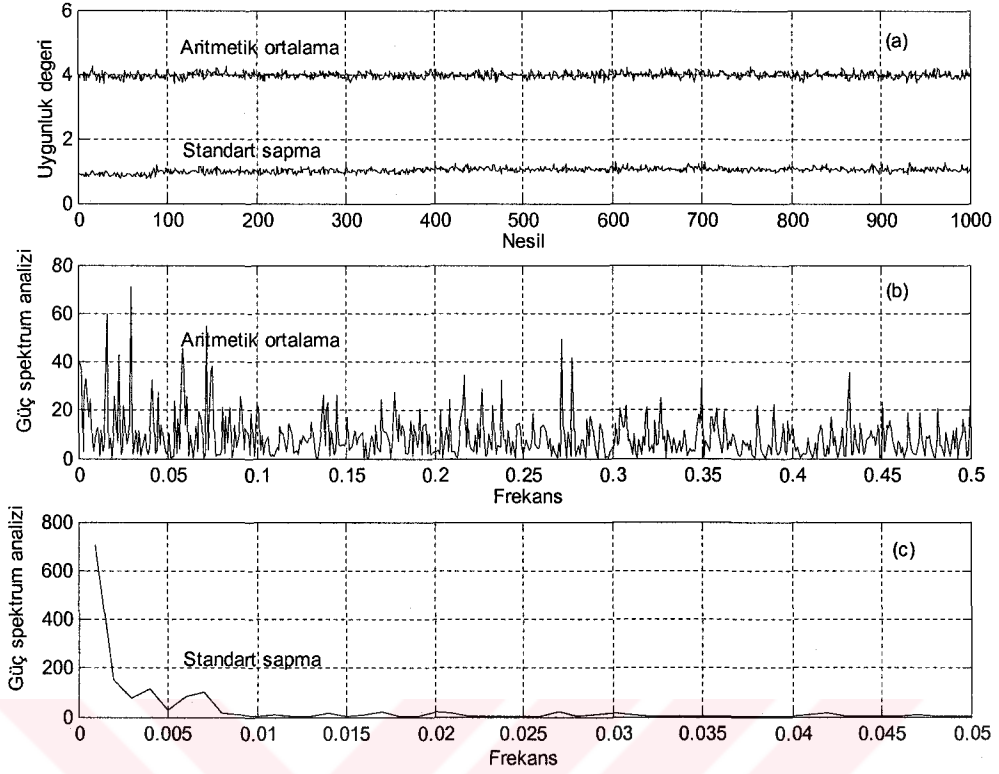


**Şekil 6.18.a)** UX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri. **(b)** SJX tekniğinin kullanılmasıyla elde edilen en büyük, ortalama ve en küçük uygunluk değerleri.

Permütasyon problemlerinde amaç, sonsuz değişimi elde etmektir. Sonsuz değişim, nesiller boyunca her neslin bir önceki nesilden daha iyi bireyler üretmesini ve nüfus içindeki bireylerin birbirinden farklı olmasını gerektirir. Bu açıdan bakıldığında her bir nesilde elde edilen nüfusun aritmetik ortalamasının ve standart sapmasının mümkün olduğu kadar değişim göstermesi istenir. Bu sebeple bu en iyi ve en kötü fire değerlerinin elde edildiği OBX ve SJX teknikler için her bir nesildeki aritmetik ortalama ve standart sapmanın güç spektrum analizi yapılmıştır. Şekil 6.19'da OBX tekniği kullanılarak yapılmış çalışmaya ait Şekil 6.20'de de SJX tekniği kullanılarak yapılmış çalışmaya ait uygunluk değerlerinin aritmetik ortalama ve standart sapmasının güç spektrum analizi gösterilmektedir. Bu analiz nesil boyunca uygunluk değerlerinin nasıl bir değişim gösterdiğini açıklamaktadır.



**Şekil 6.19.** OBX tekniği kullanılarak yapılmış çalışmaya ait uygunluk değerlerinin aritmetik ortalama ve standart sapmasının güç spektrum analizi.

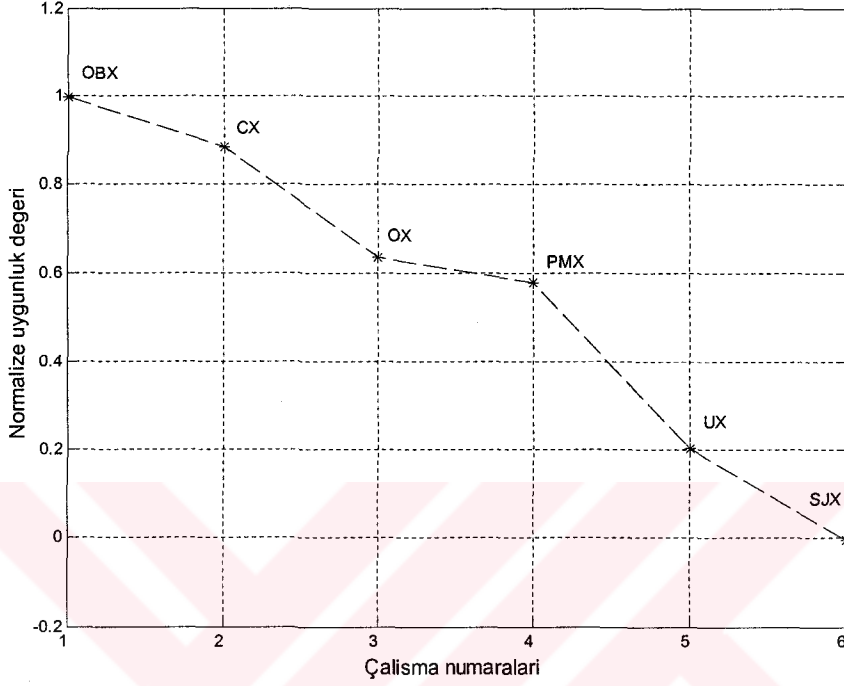


**Şekil 6.20.** SJX tekniği kullanılarak yapılmış çalışmaya ait uygunluk değerlerinin aritmetik ortalama ve standart sapmasının güç spektrum analizi.

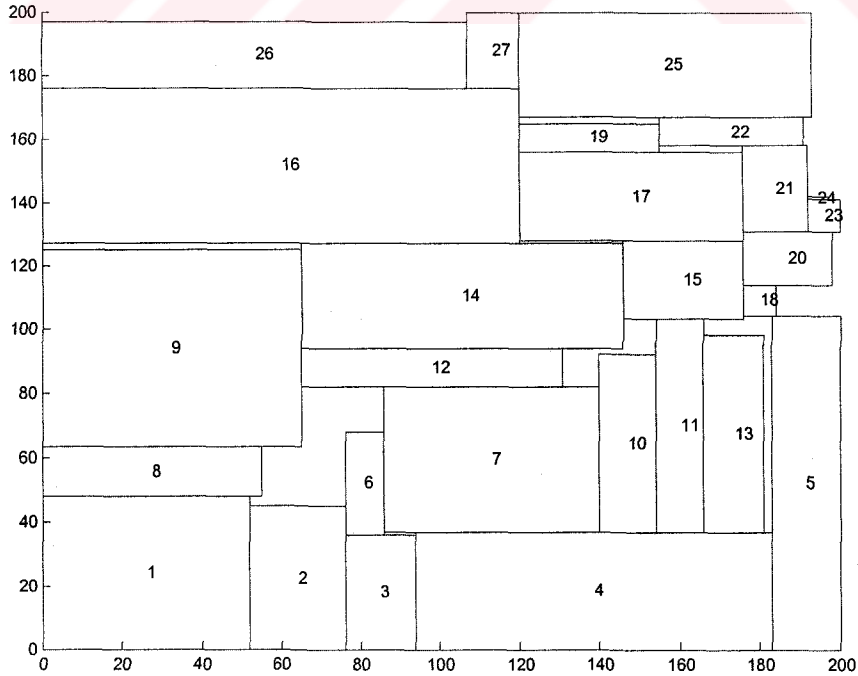
Şekil 6.19 ve Şekil 6.20'deki grafikleri karşılaştırdığımızda OBX tekniği için çizilen uygunluk değerlerinin aritmetik ortalamasının genliği güç spektrum analizinde 0 ile  $1 \times 10^4$  değerleri arasında değişirken, SJX tekniği için bu değer yaklaşık olarak 0 ile 70 arasında değişmektedir. Aynı şekilde uygunluk değerlerinin standart sapmasının genliği güç spektrum analizinde OBX tekniği için 0 ile  $2 \times 10^4$  değerleri arasında değişirken SJX tekniği için yaklaşık olarak 0 ile 700 arasında değişmektedir. Dolayısıyla OBX tekniğinde aritmetik ortalama ve standart sapmalar nesiler boyunca belli frekanslarda değişim göstermektedir. Oysaki SJX tekniğinde elde edilen aritmetik ortalama ve standart sapmalar nesiller boyunca belli frekanslarda fazla değişim göstermemektedir.

Şekil 6.21'de her çaprazlama tekniği için hesaplanan normalize uygunluk değerlerine ait bir grafik gösterilmiştir. 29 adet parçanın yerleştirilmesi için farklı çaprazlama teknikleri kullanılarak yapılan bu çalışmada fire değerleri, en büyük fire değeri 0.0992 ve en küçük fire değeri 0.0598 alınarak normalize edilmiştir.

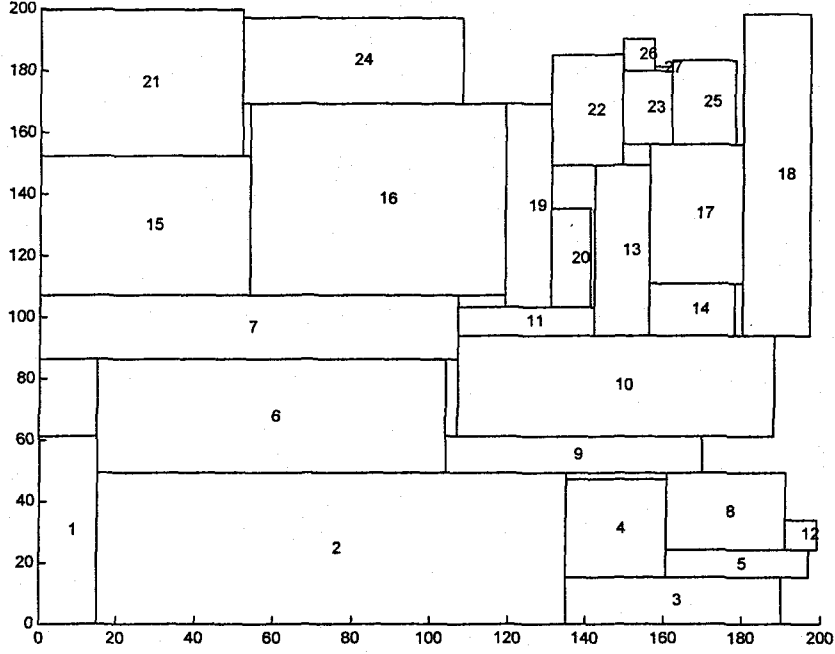
Şekil 6.22’de 80 nüfus büyüklüğü, 0.7 mutasyon oranı ve OBX çaprazlama tekniği kullanılarak yapılan çalışmaya ait yerleşim planı, Şekil 6.23’te de aynı parametreler kullanılarak SJX çaprazlama tekniği ile yapılan çalışmanın yerleşim planı görülmektedir.



Şekil 6.21. Her çaprazlama tekniği için hesaplanan normalize uygunluk değerleri.



Şekil 6.22. 80 nüfus büyüklüğü, 0.7 mutasyon oranı ve OBX çaprazlama tekniği kullanılarak yapılmış çalışmaya ait yerleşim planı.



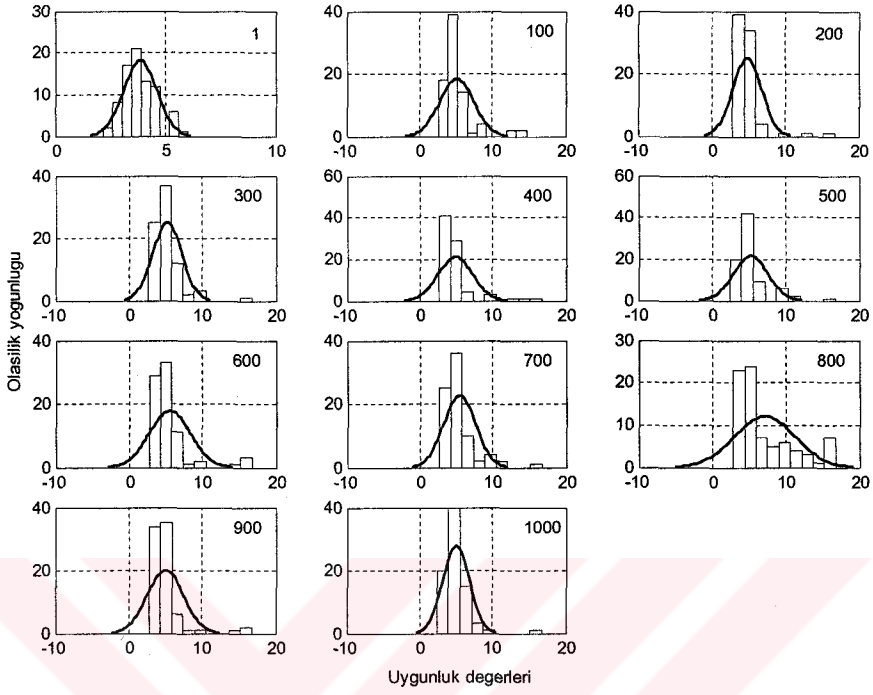
**Şekil 6.23.** 80 nüfus büyüklüğü, 0.7 mutasyon oranı ve SJX çaprazlama tekniği kullanılarak yapılmış çalışmaya ait yerleşim planı.

### 6.1.2.2. Farklı çaprazlama tekniklerinin nesiller boyunca elde edilen uygunluk değerleri üzerindeki değişimleri

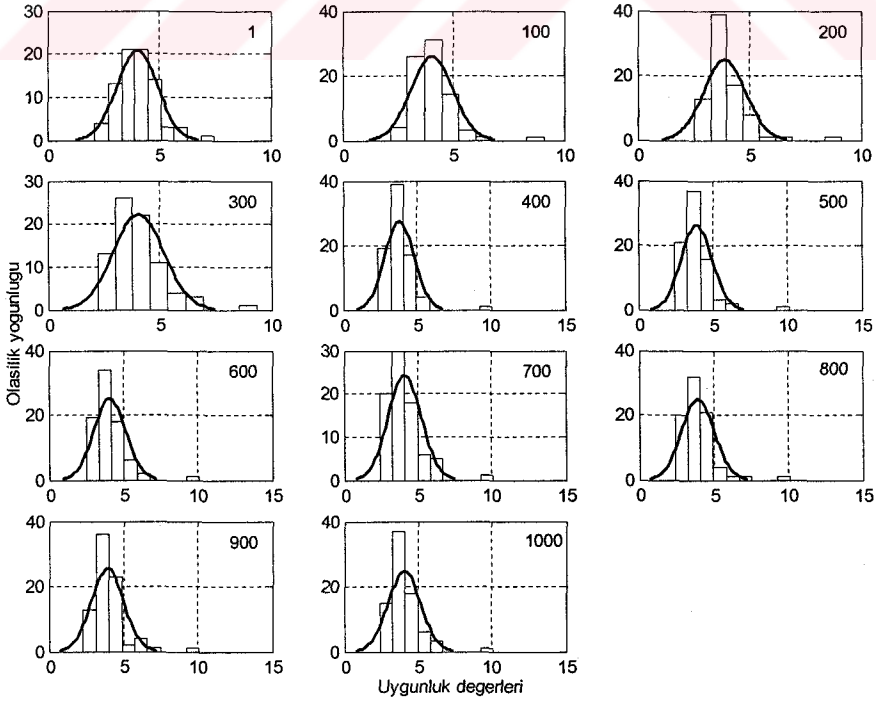
Bu bölümde farklı çaprazlama tekniklerinin nesiller boyunca uygunluk değerleri üzerindeki değişimleri ve olasılıksal bir yapıya sahip olup olmadıkları incelenmiştir. Bundan dolayı, farklı çaprazlama teknikleri için uygunluk değerlerinin olasılık yoğunluk fonksiyonları (probability density functions) incelenmiştir. Şekil 6.24'te en küçük fire değerinin elde edildiği OBX tekniğine ait çalışmanın seçilen nesiller (1., 100., 200., 300., 400., 500., 600., 700., 800., 900., ve 1000.) için uygunluk değerlerine ait olasılık yoğunluk fonksiyonu görülmektedir. Şekillerdeki her bir grafik seçilen nesiller için gösterilmiştir. OBX tekniği için bu fonksiyonlar ve histogramlar incelendiğinde nüfusun nesiller boyunca bir değişim içinde olduğu görülmektedir. Bu değişim ilerleyen nesiller için arama uzayında sürekli olarak farklı alanlarda çalışma yapıldığını göstermektedir. Şekil 6.25'te en büyük fire değerinin elde edildiği SJX tekniğine ait çalışmanın seçilen nesiller içinde uygunluk değerlerine ait olasılık yoğunluk fonksiyonu incelendiğinde bu fonksiyonların ve histogramların birbirine benzediği görülmektedir. İlerleyen nesiller boyunca devam



eden bu benzerlik aramanın arama uzayının belirli bir alanda yapıldığını göstermektedir.



**Şekil 6.24.** Seçilen nesiller için OBX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları.



**Şekil 6.25.** Seçilen nesiller için SJX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları.



Ek-D'de en küçük fire değerinden en büyük fire değerine göre sırasıyla CX, OX, PMX, UX tekniklerinin, nesiller boyunca elde edilen histogramları ve olasılık yoğunluk fonksiyonları görülmektedir.

## **6.2. Kıрма Benzetilmiş Tavlama Kullanılarak Elde Edilen Sonuçların Değerlendirilmesi**

### **6.2.1. 17 adet parçadan oluşan test problemlerine ait değerlendirmeler**

Bu çalışmada kullanılan test problemleri Bölüm 6.1.1'de kıрма GA ile çözümü bulunan 17 adet parçadan oluşan test problemlerinin aynısıdır. Bu çalışmada 320 farklı durum için yerleştirme simülasyonu yapılmıştır.

Benzetilmiş tavlama kullanılan parametre değerleri:

Başlangıç sıcaklıkları	: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
Son sıcaklık	: 0.01
Komşuluk hareketi	: Yer değiştirme (SW) ve Kaydırma (SH)
Soğutma planı	: Oransal soğutma ve Lundy ve Mees soğutma planı
Durdurma kriteri	: Öteleme sayısı (1000)
Denge koşulu için komşuluk hareketi sayısı	: 3, 5, 10

#### **6.2.1.1. Komşuluk hareketinin etkileri**

Denge koşulu için komşuluk hareketi sayısı (5) sabit tutularak sekiz değişik başlangıç sıcaklığı, iki farklı komşuluk hareketi ve iki farklı soğutma planı kullanılarak yerleştirme simülasyonları gerçekleştirilmiştir. Her bir problem için göre elde edilen en büyük ve en küçük fire değerleri Tablo 6.7'de gösterilmektedir. Yer değiştirme ve kaydırma komşuluk hareketleri, oransal soğutma ve Lundy ve Mees soğutma planı kullanılarak yapılan bu çalışmada 8 değişik başlangıç sıcaklığı için elde edilen normalize uygunluk değerlerine ait grafikler beş problem için sırasıyla Şekil 6.26, Şekil 6.27, Şekil 6.28, Şekil 6.29 ve Şekil 6.30'da gösterilmektedir. Her şekildeki (a) grafiği oransal soğutma ile yapılan çalışmayı, (b) grafiği Lundy ve Mees soğutma planı ile yapılan çalışmayı göstermektedir. Grafiklerin yanlarında bulunan

etiketler problem numarasını ve çalışmanın hangi komşuluk hareketiyle (SH ve SW) yapıldığını temsil etmektedir. khs denge koşulu için komşuluk hareketi sayısını temsil etmektedir.

**Tablo 6.7.** Beş test problemi için, sabit bir khs, iki farklı komşuluk hareketi ve iki soğutma planı kullanılarak yapılan yerleştirme simülasyonlarına ait en büyük ve en küçük fire değerleri.

Problem No	Oransal soğutma		Lundy ve Mees soğutma		Oransal soğutma		Lundy ve Mees soğutma	
	SH	SH	SH	SH	SW	SW	SW	SW
	EB fire	EK fire	EB fire	EK fire	EB fire	EK fire	EB fire	EK fire
1	0.1973	0.0892	0.2090	0.1187	0.1756	0.0824	0.1393	0.0649
2	0.2079	0.0836	0.1593	0.0838	0.2159	0.0813	0.1651	0.0833
3	0.1687	0.0826	0.1781	0.1122	0.1771	0.1085	0.1736	0.1030
4	0.1408	0.0479	0.1199	0.0506	0.1726	0.0739	0.1496	0.0469
5	0.1613	0.1073	0.1940	0.0744	0.1780	0.1184	0.1447	0.0896

Şekil 6.26'da P1 problemi için yapılan çalışmaya ait normalize uygunluk değerleri gösterilmektedir. Her iki (a) ve (b) grafiğinde başlangıç sıcaklık değeri arttıkça yer değiştirme komşuluk hareketiyle yapılan çalışmalara ait normalize uygunluk değerleri yükselme eğilimi göstermektedir. Şekil 6.26 (b)'de yer değiştirme ve Lundy ve Mees soğutma planı kullanılarak yapılan çalışmada normalize uygunluk değeri, (a) ve (b) grafiğinde gösterilen diğer çalışmaların uygunluk değerlerine göre daha iyi bir değişim göstermektedir.

Şekil 6.27.'de P2 problemi için yapılan çalışmaya ait normalize uygunluk değerleri gösterilmektedir. P2 problemi için elde edilen en iyi çalışma grafiği değişen başlangıç sıcaklığı değerleri için (b) grafiğinde Lundy ve Mees soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılan çalışmaya aittir.

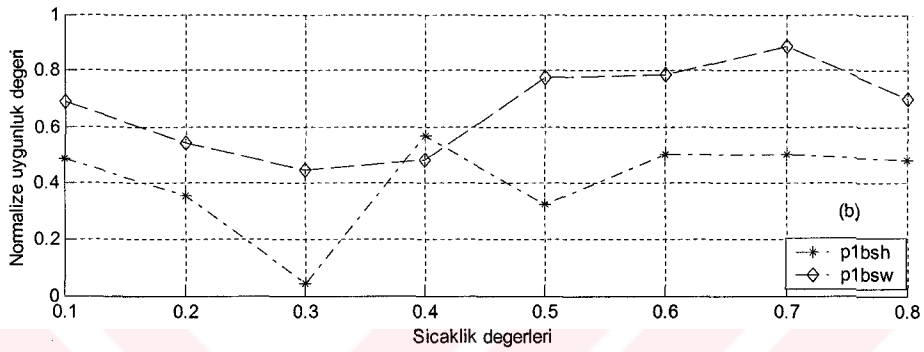
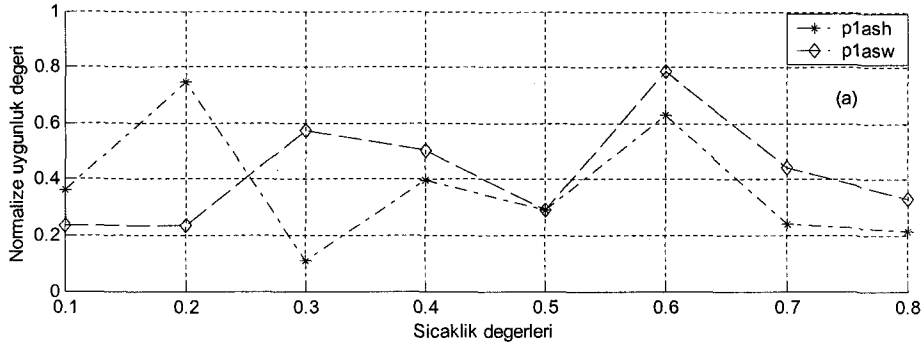
P3 problemi için yapılan çalışmaya ait normalize uygunluk değerleri Şekil 6.28'de gösterilmektedir. (a) grafiğinde oransal soğutma planı kullanılarak yapılan çalışmanın her ikisi de 0.5 başlangıç sıcaklığı değerinden sonra aynı yükselme eğilimini göstermektedir. (b) grafiğinde değişen başlangıç sıcaklık değerleri için Lundy ve Mees soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak

yapılan çalışma kaydırma komşuluk hareketi kullanılarak yapılan çalışmaya göre çoğunlukla daha iyi sonuçlar vermiştir.

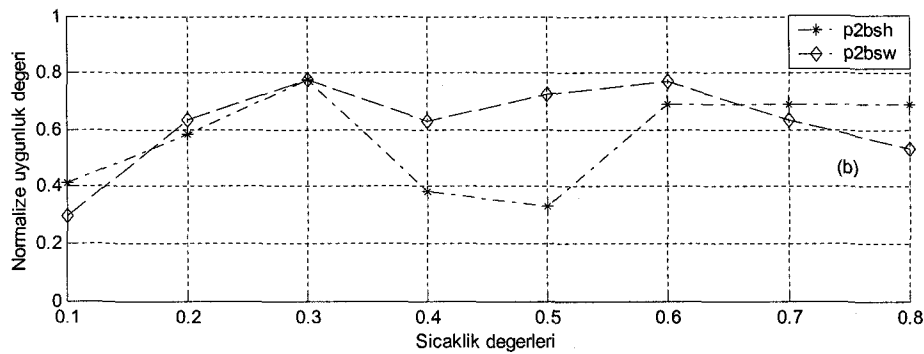
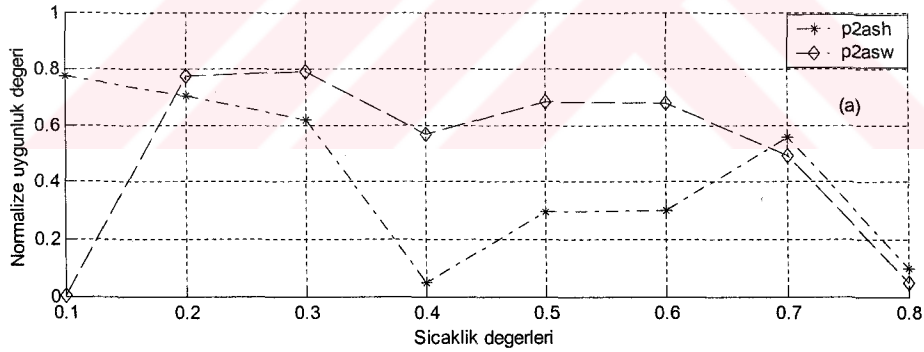
P4 problemi için yapılan çalışmaya ait normalize uygunluk değerleri Şekil 6.29'da gösterilmektedir. (a) grafiğinde oransal soğutma planı kullanılarak yapılan çalışmanın her ikisinde de 0.4 başlangıç sıcaklık değerine kadar sabit normalize uygunluk değerleri elde edilmiştir. Bu çalışmada 0.6 başlangıç sıcaklığı değerinden sonra kaydırma komşuluk hareketi ile yapılan çalışma bir yükselme eğilimiyle en iyi uygunluk değerini göstermiştir. (b) grafiğinde Lundy ve Mees soğutma planı ile birlikte yer değiştirme ve kaydırma komşuluk hareketi kullanılarak yapılan çalışmanın her ikisinde de 0.3 başlangıç sıcaklığı değerinden sonra bir yükselme eğilimi görülmektedir. Yer değiştirme komşuluk hareketi ve Lundy ve Mees soğutma planı ile yapılan çalışmada, 0.3 başlangıç noktasından sonra gerçekleşen yükselme eğilimiyle en iyi uygunluk değeri elde edilmiştir. P4 problemi yapılan tüm çalışmalar içinde en iyi sonuçların elde edildiği problemidir.

Şekil 6.30'da P5 problemine ait normalize uygunluk değerleri gösterilmektedir. (a) grafiğinde oransal soğutma planı ve kaydırma komşuluk hareketinin yapıldığı çalışmada normalize uygunluk değeri artan başlangıç sıcaklığı değerleri için oldukça değişen bir eğilim göstermektedir. (b) grafiğinde Lundy ve Mees soğutma planı ile birlikte kaydırma komşuluk hareketinin kullanıldığı çalışma 0.5 başlangıç sıcaklığı değerine kadar bir yükselme eğilimi göstermektedir. Yer değiştirme komşuluk hareketi kullanılarak yapılan çalışmada ise değişen başlangıç sıcaklığı değerleri için normalize uygunluk değerleri çok fazla bir değişim göstermemiştir.

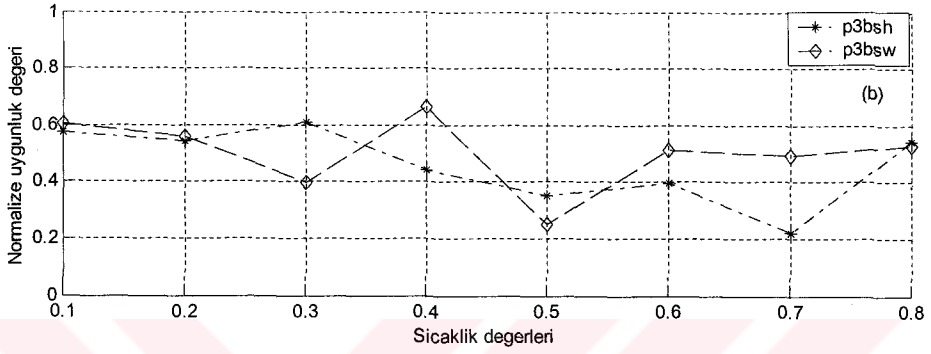
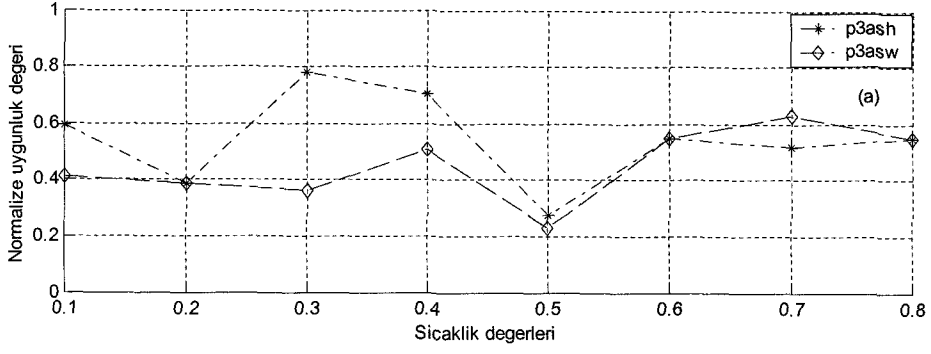
Şekil 6.31'de denge koşulu için komşuluk hareketi sayısı (5) sabit tutularak yer değiştirme komşuluk hareketi ve Lundy ve Mees soğutma planı kullanılarak yapılan çalışma sonucunda elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı gösterilmektedir.



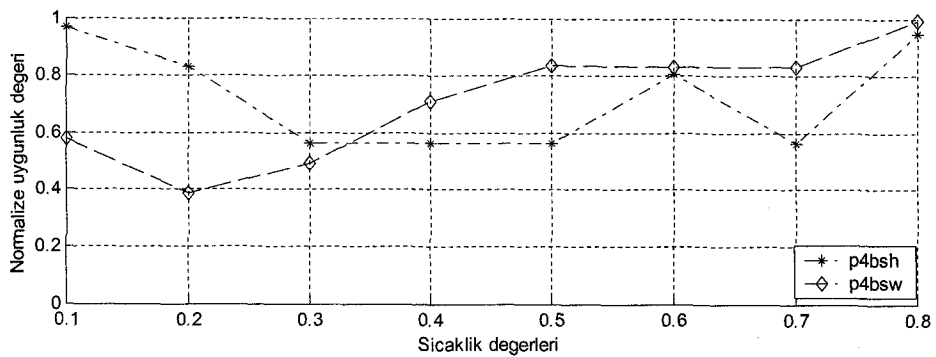
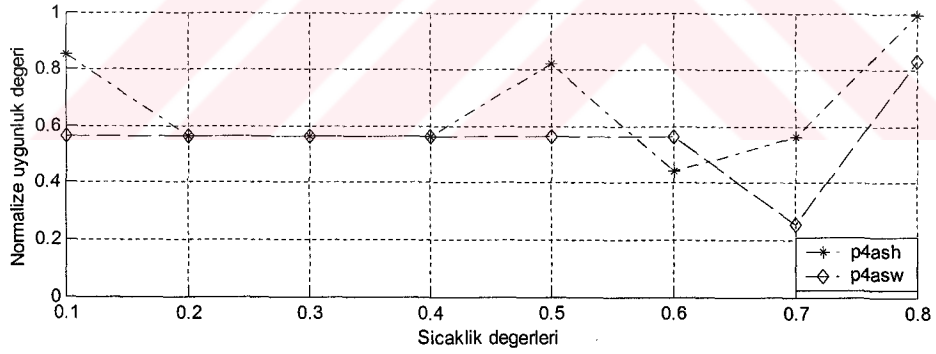
**Şekil 6.26.** Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P1 problemine ait normalize uygunluk değerleri.



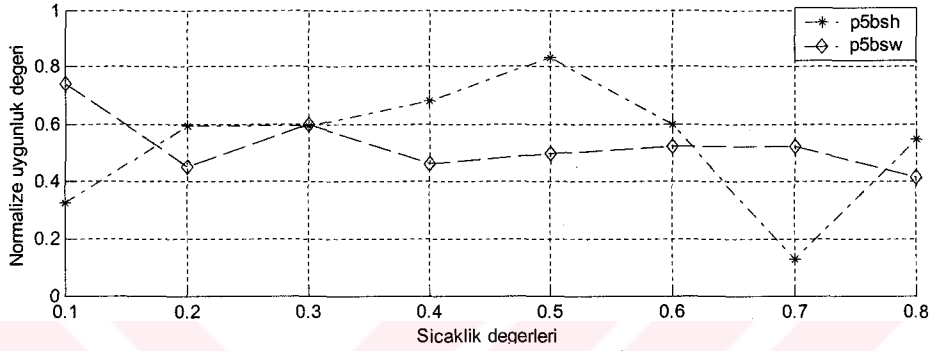
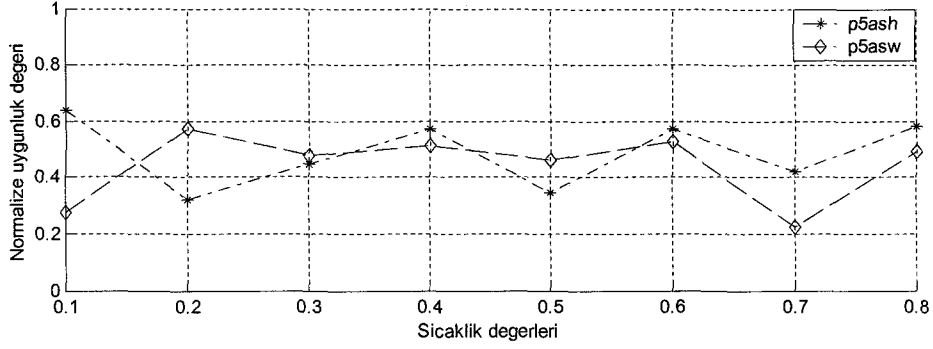
**Şekil 6. 27.** Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P2 problemine ait normalize uygunluk değerleri.



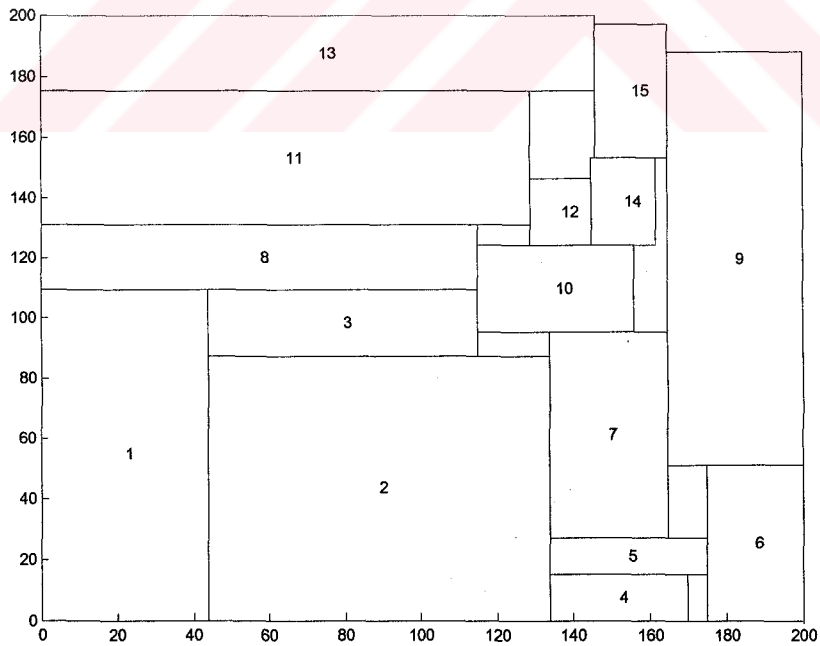
**Şekil 6.28.** Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P3 problemine ait normalize uygunluk değerleri.



**Şekil 6.29.** Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P4 problemine ait normalize uygunluk değerleri.

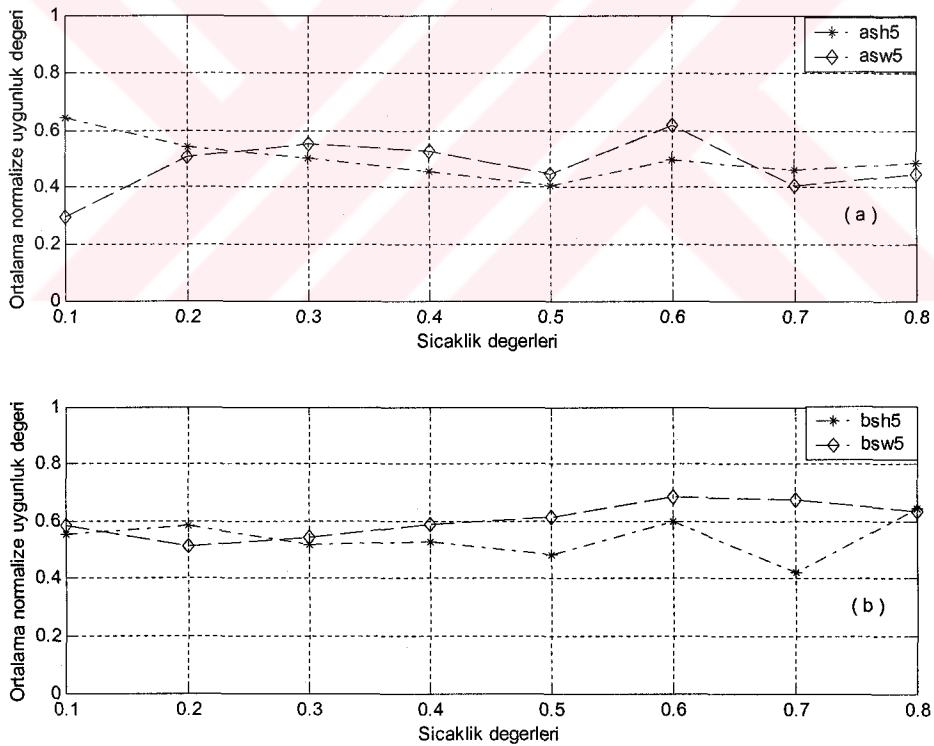


Şekil 6.30. Sekiz değişik başlangıç sıcaklığı, denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış P5 problemine ait normalize uygunluk değerleri.



Şekil 6.31. Denge koşulu için sabit bir khs 5, yer değiştirme komşuluk hareketi ve Lundy ve Mees soğutma planı kullanılarak P4 problemi için yapılan çalışmaya ait yerleşim planı.

Şekil 6.32’de denge koşulu için sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış beş farklı problem için elde edilen ortalama normalize uygunluk değerlerine ait grafikler görülmektedir. Şekildeki (a) grafiği oransal soğutma ile yapılan çalışmayı, (b) grafiği Lundy ve Mees soğutma planı ile yapılan çalışmayı göstermektedir. Grafiklerin etiketlerindeki sh ve sw ise çalışmaların sırasıyla kaydırma ve yer değiştirme komşuluk hareketleriyle yapıldığını göstermektedir. Şekil 6.32 (a) da oransal soğutma ile yapılan çalışmada, yer değiştirme komşuluk hareketi kaydırma komşuluk hareketine göre 0.3-0.6 arasında kalan başlangıç sıcaklığı değerleri için daha iyi sonuçlar vermektedir. Şekil 6.32 (b)’de de Lundy ve Mees soğutma planı ile yapılan çalışmada, yer değiştirme komşuluk hareketi kaydırma komşuluk hareketine göre 0.4-0.8 arasında kalan başlangıç sıcaklığı değerleri için nispeten daha iyi sonuçlar vermektedir. Dolayısıyla devam eden çalışmalarda yer değiştirme komşuluk hareketi kullanılacaktır.



**Şekil 6.32.** Sekiz değişik başlangıç sıcaklığı, sabit bir khs (5), iki farklı soğutma planı ve komşuluk hareketi kullanılarak yapılmış çalışmaların ortalama normalize uygunluk değerleri.

Tablo 6.8’de sadece yer değiştirme komşuluk hareketi kullanılarak oransal soğutma, Lundy ve Mees soğutma planı ve ayrıca denge koşulu için üç değişik komşuluk

hareketi sayısı (3, 5, 10) ile yapılan çalışmalara ait fire değerleri gösterilmektedir. 17 parçanın yerleştirilmesi için benzetilmiş tavlama algoritmasıyla yapılan tüm yerleştirme simülasyonları için elde edilen en büyük fire değeri Tablo 6.8’de görüldüğü gibi 0.2159, en küçük fire değeri 0.0454’tür. Bu değerler kullanılarak fire değerleri normalize edilmiştir. En büyük fire değeri (0.2159) sıfır, en küçük fire değeri (0.0454) bir kabul edilerek, normalizasyon bu değerler arasında yapılmıştır. Normalize edilmiş uygunluk değerleriyle fire değerleri arasında ters orantı vardır. Başka bir deyişle normalize edilmiş uygunluk değeri arttıkça fire azalmakta, azaldıkça fire artmaktadır. Bölüm 6.2.1 içinde yapılan tüm çalışmalar bu oranlar için normalize edilmiş uygunluk değerleriyle gösterilmektedir.

#### **6.2.1.2. Denge koşulu için değişik komşuluk hareketi sayılarının ve soğutma planlarının etkileri**

Bu çalışmada beş problem üzerinde sekiz değişik başlangıç sıcaklığı için yer değiştirme komşuluk hareketi, oransal soğutma ve Lundy ve Mees soğutma planı ve denge koşulu için üç değişik komşuluk hareketi sayısı kullanılarak yerleştirme simülasyonları gerçekleştirilmiştir. Çalışmalara ait grafikler Şekil 6.33, Şekil 6.34, Şekil 6.35, Şekil 6.36 ve Şekil 6.37’de gösterilmektedir. Grafiklerin yanlarında bulunan etiketler problem numarasını, komşuluk hareketi sayısını, ve çalışmanın hangi soğutma planı kullanılarak yapıldığını göstermektedir. Etiketlerde oransal soğutma planı (a), Lundy ve Mees soğutma planı (b) ile temsil edilmiştir. Tablo 6.8’de bu çalışmalar için elde edilen fire değerleri gösterilmektedir.



**Tablo 6. 8** Denge koşulu için üç değişik khs ve kaydırma komşuluk hareketi kullanılarak elde edilen fire değerleri.

	Oransal soğutma					
	SW					
	3		5		10	
Problem No	EB fire	EK fire	EB fire	EK fire	EB fire	EK fire
1	0.1651	0.0656	0.1756	0.0824	0.1913	0.1001
2	0.1593	0.0911	<b>0.2159</b>	0.0813	0.1582	0.1082
3	0.2006	0.1139	0.1771	0.1085	0.1771	0.1137
4	0.1730	0.0556	0.1726	0.0739	0.1726	0.0662
5	0.1519	0.0871	0.1780	0.1184	0.1682	0.0887
	Lundy ve Mees soğutma					
	SW					
	3		5		10	
Problem No	EB fire	EK fire	EB fire	EK fire	EB fire	EK fire
1	0.1663	0.0579	0.1393	0.0649	0.1751	0.1089
2	0.1354	0.0930	0.1651	0.0833	0.1681	0.0658
3	0.1674	0.0884	0.1736	0.1030	0.1523	0.1029
4	0.1419	<b>0.0454</b>	0.1496	0.0469	0.1199	0.0765
5	0.1401	0.1001	0.1447	0.0896	0.1442	0.0954
En kötü/iyi yerleşim planlarına ait fire değerleri	EB fire			EK fire		
	<b>0.2159</b>			<b>0.0454</b>		

Şekil 6.33'de P1 problemi için yapılan çalışmalarda en iyi sonuçlar denge koşulu için komşuluk hareketi sayısı 3 ve 5 alındığında ve Lundy ve Mees soğutma planı kullanıldığında elde edilmiştir. Bu çalışmalarda değişen başlangıç sıcaklığı değerleri için normalize uygunluk değerleri bir artış göstermektedir. En düşük uygunluk değerleri komşuluk hareketi sayısı 10 alındığında ve Lundy ve Mees soğutma planı kullanıldığında elde edilmiştir.

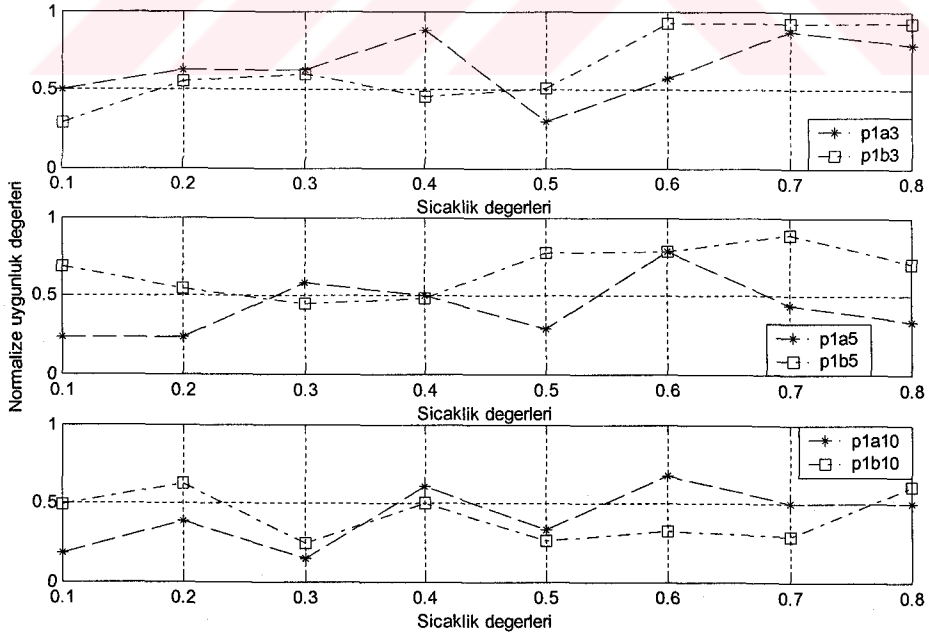
Şekil 6.34'te P2 problemi için yapılan çalışmalara ait grafikler gösterilmektedir. Bu çalışmada, komşuluk hareketi sayısı 3 ve 5 alındığında ve Lundy ve Mees soğutma planı kullanıldığında P2 problemi için yapılan diğer çalışmalara göre daha iyi sonuçlar elde edilmiştir.

Şekil 6.35'de P3 problemi için yapılan çalışmalara ait grafikler gösterilmektedir. Denge koşulu için komşuluk hareketi sayısı 3 alındığında ve Lundy ve Mees soğutma planı kullanıldığında değişen başlangıç sıcaklığı değerleri için sabit bir

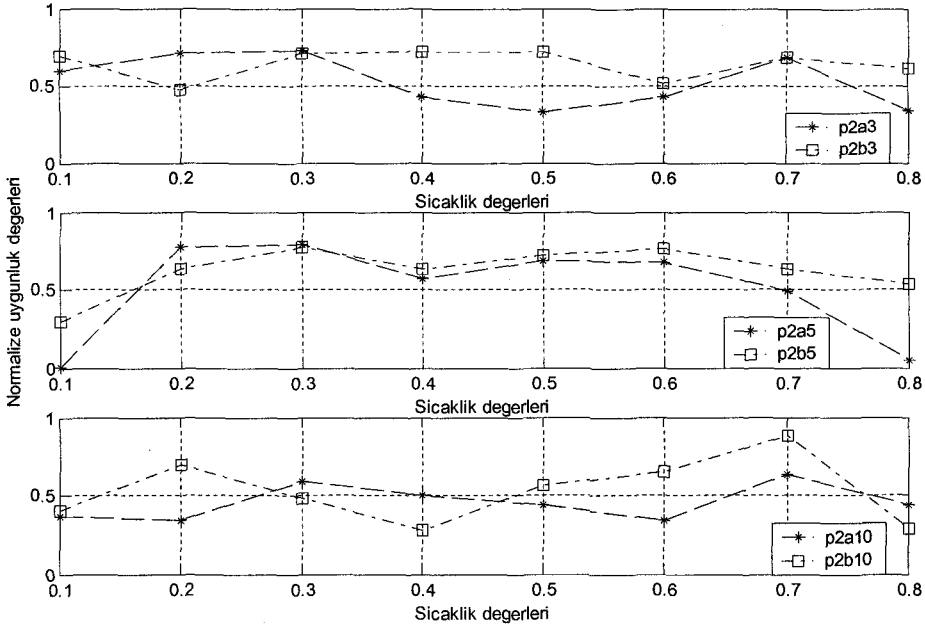
eğilim gösteren uygunluk değerleri elde edilmiştir. Diğer çalışmalarda elde edilen uygunluk değerleri değişen başlangıç sıcaklık değerleri için sürekli değişen ve birbirine benzeyen çalışma grafikleri sergilemektedir.

Şekil 6.36'da P4 problemi kullanılarak yapılan çalışmada denge koşulu için komşuluk hareketi sayısı 5 alındığında ve Lundy ve Mees soğutma planı kullanıldığında değişen sıcaklık değerleri için normalize uygunluk değerleri artan bir eğilim göstermektedir. En iyi normalize uygunluk değeri, P4 probleminin kullanıldığı çalışmada elde edilmiştir. Çalışmanın tamamına genel olarak baktığımızda elde edilen normalize uygunluk değerleri diğer problemler için yapılan çalışmalardan daha iyi sonuçlar vermiştir.

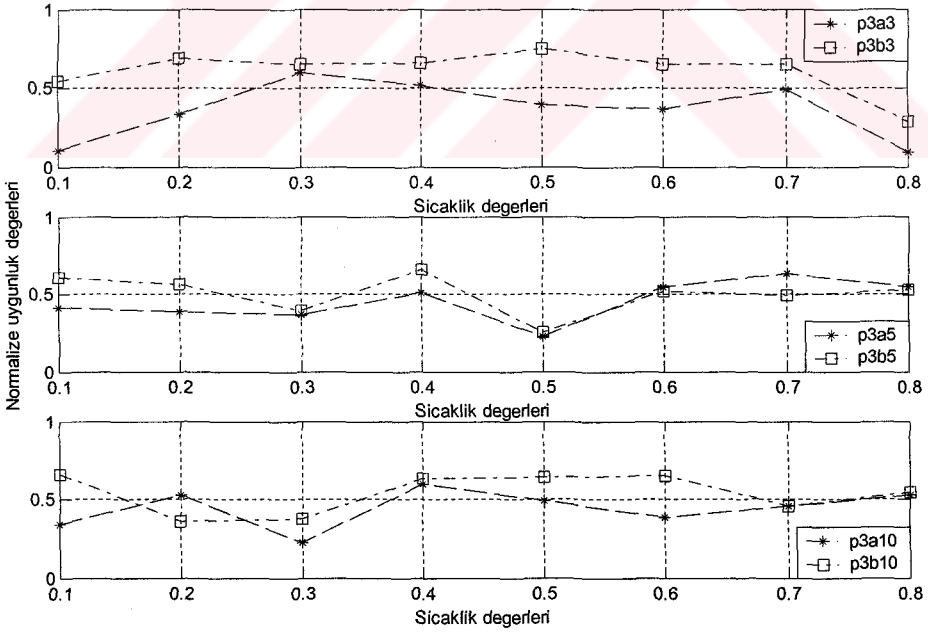
Şekil 6.37'de P5 problemine ait çalışmada denge koşulu için komşuluk hareketi sayısı 5 alınarak her iki soğutma planı için elde edilmiş sonuçlar oldukça birbirine benzer değişimler göstermektedir. Denge koşulu için komşuluk hareketi sayısı 3 ve 10 alınarak Lundy ve Mees soğutma planı kullanıldığında elde edilen sonuçlar diğer çalışmalara göre nispeten daha iyi bir eğilim sergilemektedir.



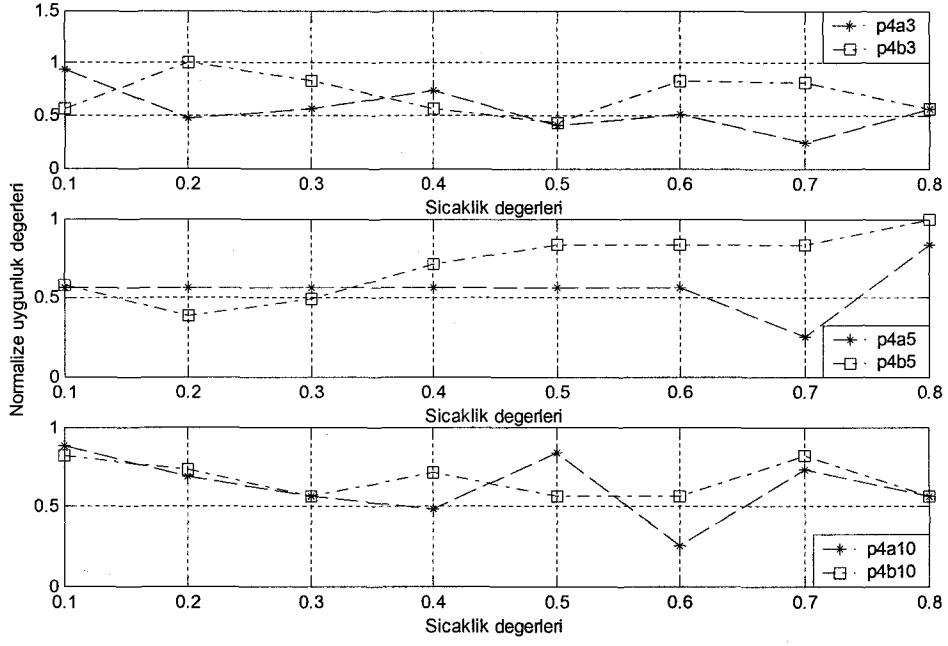
**Şekil 6.33.** Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P1 problemine ait normalize uygunluk değerleri.



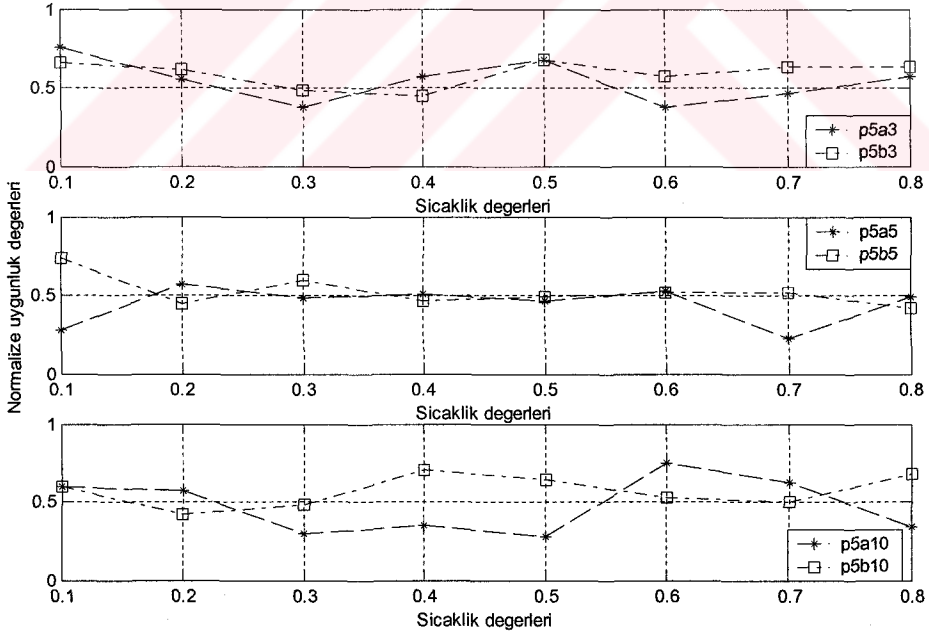
Şekil 6.34. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P2 problemine ait normalize uygunluk değerleri.



Şekil 6.35. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P3 problemine ait normalize uygunluk değerleri.

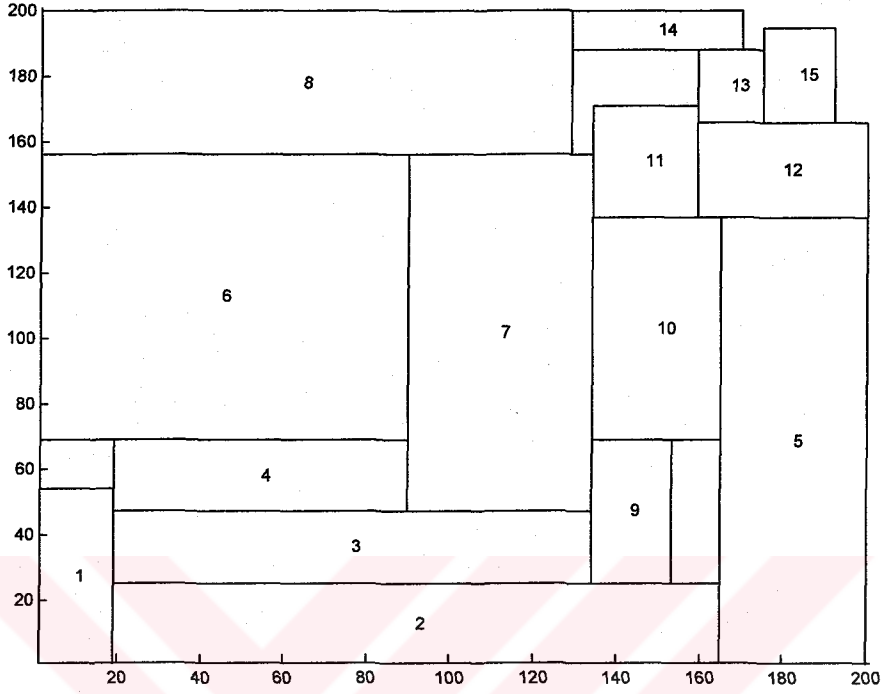


Şekil 6.36. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P4 problemine ait normalize uygunluk değerleri.



Şekil 6.37. Sekiz değişik başlangıç sıcaklığı, üç değişik khs (3, 5, 10) iki farklı soğutma planı ve yer değiştirme komşuluk hareketi kullanılarak yapılmış P5 problemine ait normalize uygunluk değerleri.

Şekil 6.38’de bu çalışmalar için elde edilen en küçük fire değerine sahip P4 problemine ait yerleşim planı gösterilmektedir.

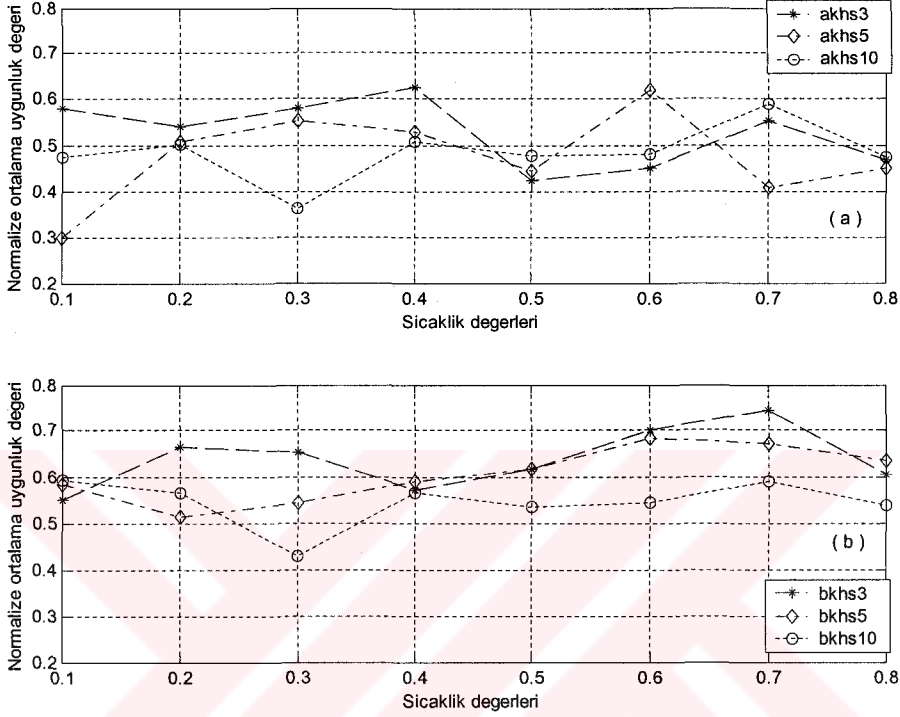


**Şekil 6.38.** Yer deęiřtirme komřuluk hareketi, Lundy ve Mees soęutma planı ve khs 3 kullanılarak P4 problemi için yapılan çalıřmaya ait yerleşim planı.

Sekiz bařlangıç sıcaklıęı ve yer deęiřtirme komřuluk hareketi kullanılarak, soęutma planlarına ve üç farklı komřuluk hareketi sayısına (3, 5, 10) göre tüm problemlerin normalize ortalama deęerleri Şekil 6.39’da gösterilmektedir. Grafiklerin yanlarında bulunan etiketlerde oransal soęutma planı (a), Lundy ve Mees soęutma planı (b) ile temsil edilmektedir. Etiketlerde gösterilen khs denge kořulu için komřuluk hareketi sayısını temsil etmektedir.

Oransal soęutma ve Lundy ve Mees soęutma planının ayrı ayrı deęerlendirildięi Şekil 6.39’daki grafikten normalize ortalama uygunluk deęerlerine bakıldıęında Lundy ve Mees soęutma planının oransal soęutma planına göre daha iyi sonuçlar verdięi görülmektedir. Deęişik komřuluk hareketleri sayısı için oransal soęutma planı kullanılarak yapılan deęerlendirmede (Şekil 6.39 a) elde edilen normalize ortalama uygunluk deęerleri yaklaşık olarak 0.30 ve 0.65 arasında deęişirken, Lundy ve Mees soęutma planı kullanılarak yapılan deęerlendirmede (Şekil 6.39 b) elde

edilen değerler yaklaşık olarak 0.40 ve 0.75 arasında değişmektedir. Her iki soğutma planının da değerlendirildiği bu çalışmada elde edilen en iyi artan eğilimi gösteren normalize ortalama uygunluk değeri komşuluk hareketi sayısı 3, soğutma planı olarak Lundy ve Mees kullanıldığında elde edilmiştir.



**Şekil 6.39.** Sekiz başlangıç sıcaklığı ve yer değiştirme komşuluk hareketi kullanılarak, soğutma planlarına ve üç değişik khs (3, 5, 10) göre tüm problemlerin normalize ortalama değerleri.

### 6.2.2 29 adet parçanın yerleşimi için Lundy ve Mees soğutma planı kullanılarak gerçekleştirilen test problemine ait değerlendirmeler

Bu çalışmada kullanılan test problemi Bölüm 6.1.2’de kullanılan 29 adet parçadan oluşan her biri birbirinden farklı düzgün dikdörtgen şekiller içeren problemin aynısıdır. Bu çalışmada 8 farklı durum için yerleştirme simülasyonu yapılmıştır.

Benzetilmiş tavlama algoritmasında kullanılan parametre değerleri:

Başlangıç sıcaklıkları : 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8

Son sıcaklık : 0.01

Komşuluk hareketi : Yer deęiřtirme (SW)  
Soęutma planı : Lundy ve Mees soęutma planı  
Durdurma kriteri : Öteleme sayısı (1000)  
Denge kořulu için komşuluk hareketi sayısı : 3

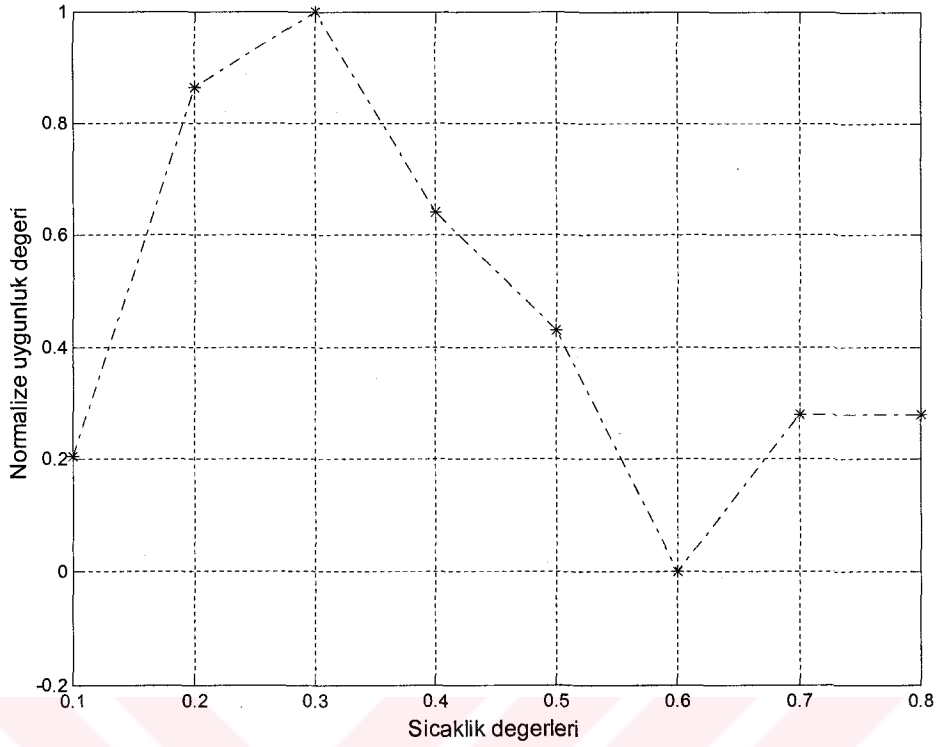
29 adet parçanın yerleřtirilmesi için yapılan bu çalışmada 8 farklı bařlangıç sıcaklıęı olmak üzere, 17 parçaya sahip test problemleri için yapılan çalışmalarda en iyi sonucu veren parametreler kullanılmıřtır. Tablo 6.9'da 29 parçanın yerleřtirme simülasyonu sonucunda 8 deęiřik bařlangıç sıcaklıęı için elde edilen fire deęerleri gösterilmektedir. En küçük fire deęeri 0.3 bařlangıç sıcaklıęı kullanıldıęında elde edilmiřtir.

**Tablo 6.9.** 29 adet parçanın yerleřimi sonucunda elde edilen fire deęerleri.

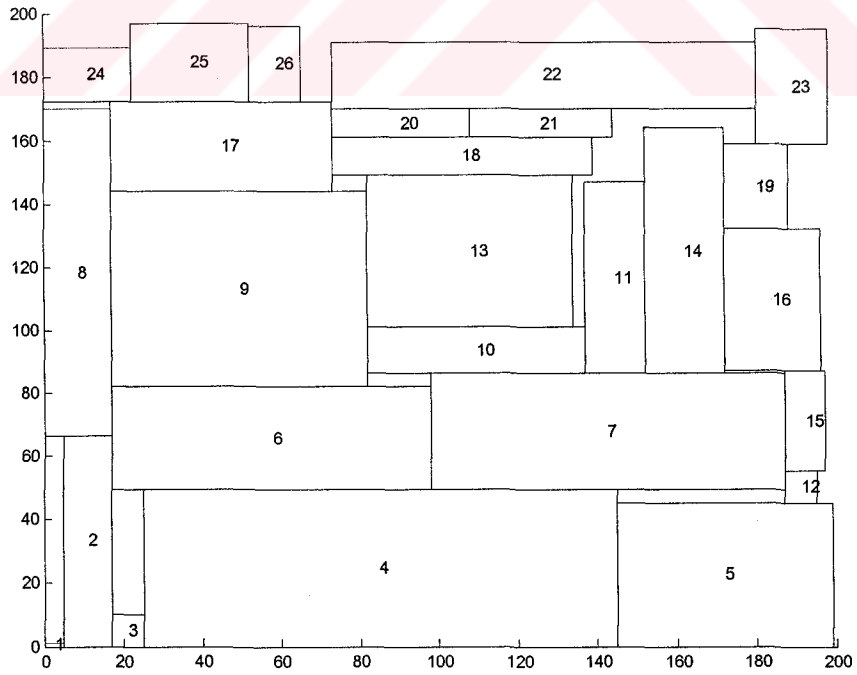
Çalışma no	Başlangıç sıcaklıkları	Fire
1	0.1	0.1627
2	0.2	0.1110
3	0.3	0.1003
4	0.4	0.1284
5	0.5	0.1450
6	0.6	0.1787
7	0.7	0.1568
8	0.8	0.1568

29 adet parçanın yerleřtirilmesi için 8 deęiřik bařlangıç sıcaklıęı kullanılarak yapılan bu çalışmada fire deęerleri, en büyük fire deęeri 0.1787 ve en küçük fire deęeri 0.1003 alınarak normalize edilmiřtir. Buna göre 29 adet parçanın yerleřimi sonucu elde edilen normalize uygunluk deęerleri Őekil 6.40'da gösterilmektedir. 0.1-0.3 bařlangıç sıcaklıkları arasında bir yükseliř gösteren çalışma, 0.4–0.6 bařlangıç sıcaklıkları arasında bir düşüř göstermektedir. Bu çalışma için elde edilen fire deęerleri %10 ve %17 arasında deęiřmektedir.

Őekil 6.41'de 0.3 bařlangıç sıcaklıęı, yer deęiřtirme komşuluk hareketi, Lundy ve Mees soęutma planı , denge kořulu için komşuluk hareketi sayısı 3 kullanılarak elde edilen en küçük fire deęerine sahip çalışmaya ait yerleřim planı görülmektedir.



Şekil 6.40. 29 adet parçanın yerleşimi sonucu elde edilen normalize uygunluk değerleri.



Şekil 6.41. 0.3 başlangıç sıcaklığı, yer değiştirme komşuluk hareketi, Lundy ve Mees soğutma planı, khs 3 kullanılarak elde edilen çalışmaya ait yerleşim planı.



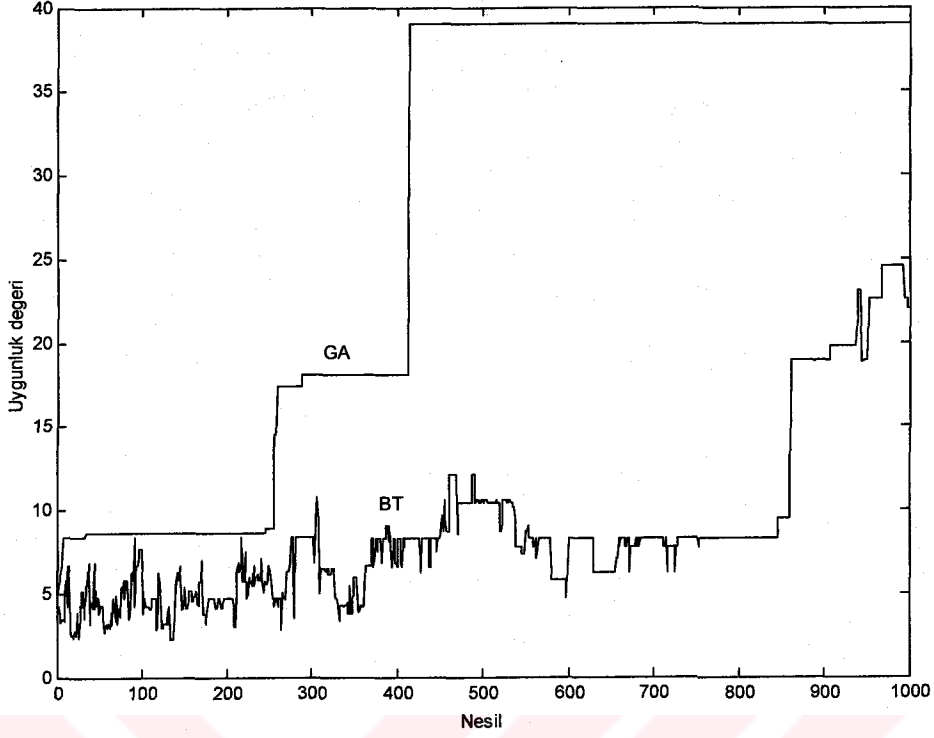
### 6.3. Uygulanan Kıрма Yaklaşımların Değerlendirilmesi

Bu çalışmada GA ve BT yöntemleri kullanılarak 17 ve 29 parçadan oluşan iki kesme probleminin çözümü için kırma yaklaşımlar uygulanmıştır. Her iki yöntem ve her iki kesme problemi kullanılarak problemler için elde edilen en küçük ve en büyük fire değerleri Tablo 6.10'da verilmiştir. Tablodan görüldüğü her iki problem için en küçük fire değerleri GA yöntemi kullanıldığında elde edilmiştir. Bu çalışmalar sonucunda her iki yöntem için de en küçük fire değerleri P4 problemi kullanıldığında elde edilmiştir.

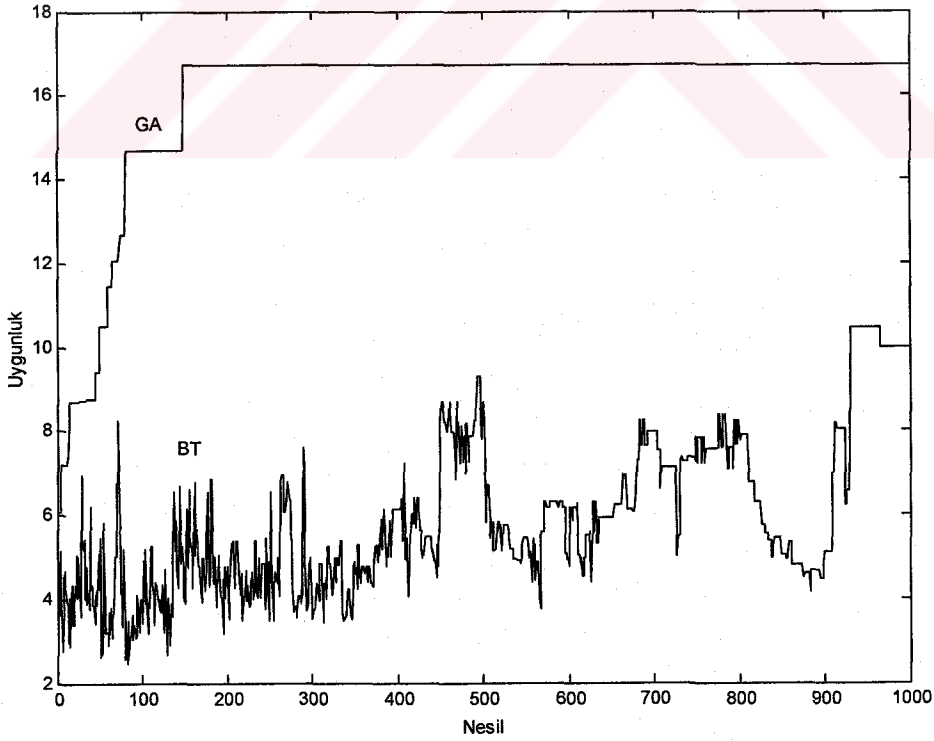
**Tablo 6.10.** Kesme problemleri için elde edilen en küçük ve en büyük fire değerleri.

Problemler	Kırma GA		Kırma BT	
	EK fire	EB fire	EK fire	EB fire
17 parça	%2	%11	%4	%21
29 parça	%5	%9	%10	%17

Şekil 6.42, GA ve BT yöntemleri kullanıldığında 17 parça içeren P4 problemi için elde edilen en iyi uygunluk değerlerine ait grafiği göstermektedir. Şekilden görüldüğü gibi genetik algoritma en iyi uygunluk değerini 450. nesilden sonra elde etmiştir. Benzetilmiş tavlama yöntemi kullanıldığında ise 900. nesle kadar arama uzayında kısır bir araştırma olmuş ve problemin çözümü için bir gelişme olmamıştır. Grafikler incelendiğinde genetik algoritmaların benzetilmiş tavlamaya göre yaklaşık %38 daha iyi sonuç verdiğini söyleyebiliriz. Şekil 6.43, GA ve BT yöntemleri kullanıldığında 29 parça içeren test problemi için elde edilen en iyi uygunluk değerlerine ait grafiği göstermektedir. Şekilden görüldüğü gibi genetik algoritma en iyi uygunluk değerini 200. nesilden sonra elde etmiştir. Benzetilmiş tavlama yöntemi kullanıldığında ise 900. nesle kadar arama uzayında kısır bir araştırma olmuştur. 29 parça için benzetilmiş tavlama ile en iyi uygunluk değeri ancak 900. nesilden sonra elde edilmiştir. Bu çalışma için grafikler incelendiğinde genetik algoritmaların benzetilmiş tavlamaya göre yaklaşık %40 daha iyi sonuç verdiğini söyleyebiliriz.



Şekil 6.42. GA ve BT yöntemleri kullanılarak 17 parça içeren P4 problemi için elde edilen en iyi uygunluk değerleri.



Şekil 6.43. 29 parça içeren problem için GA ve BT kullanılarak elde edilen en iyi uygunluk değerleri.

Genel olarak her iki grafikte incelendiğinde BT yönteminin ilerleyen nesiller için çözümde küçük aşımalar yaparken GA'nın çok büyük aşımalar yaptığı gözlenmektedir. BT yönteminin daha iyi çözümler elde edebilmesi için daha fazla ötelemeye ihtiyaç duyduğu görülmektedir. BT yöntemi için her nesilde sürekli değişen uygunluk değerlerine bakıldığında GA'a göre BT yönteminin daha kararsız bir yapısı vardır.



## BÖLÜM 7. SONUÇLAR

Günümüz ekonomik koşullarında kaynakların verimli kullanımı özellikle imalat yapan işletmelerde çok önemlidir. Daima rekabetçi bir ortamda bulunan işletmelerde üretim maliyetlerinin azaltılması için planlama büyük bir önem taşımaktadır. Planlama yapılabilmesi için eniyileme yapılması gerekmektedir. Genellikle planlama için birden fazla parametrenin eniyilemesi istenmektedir. Bu tür eniyileme yapan en uygun eniyileme teknikleri genetik algoritma ve benzetilmiş tavlama'dır.

Çoğunlukla üretim yapan işletmelerde karşımıza çıkan kesme problemleri kaynakların verimli kullanımına dayanan farklı koşul ve amaçlar içermektedir. Bu problemlerdeki amaç, parçaların kesileceği ana parça üzerinde küçük parçaların en iyi yerleşimini gerçekleştirmektir. Kullanılan ana parça sınırsız bir yüksekliğe sahip olursa amaç yerleştirilen parçaların en az yeri kaplamasıdır. Ana parça eğer sınırları önceden belirlenmiş belirli bir alana sahipse amaç tüm küçük parçaların yerleşimini gerçekleştirmek veya kullanılmayan alanın en aza indirilmesini sağlamaktır. Ancak küçük parçaların ana parça içine yerleştirilirken birbirleriyle çakışmaları istenmeyen bir durumdur. Bu kısıtlama göz önüne alınarak parçalar yerleştirilmek istense bile bu defa çok fazla yerleşim planıyla karşı karşıya kalınacaktır. Bu da arama uzayının büyümesi ve çözüme ulaşmanın zorlaşması demektir. Bu yüzden kesme yerleşim planlarının yerleştirilecek küçük parçaların sırasını temsil eden birer permütasyonla gösterilmesi kaçınılmazdır. Arama uzaylarının büyük ve karmaşık olduğu, matematiksel bir analiz ve modellemenin yapılamadığı kesme problemlerinin çözümünde eniyileme teknikleriyle en iyi çözüme yakın iyi çözümler elde edilebilmektedir.

Bu tez çalışmasında ele alınan kesme problemlerinin çözümü için kullanılan eniyileme teknikleri genetik algoritma ve benzetilmiş tavlama'dır. Genetik

algoritmaların aynı anda pek çok noktada birden arama yapması, genetik çeşitliliği kolaylıkla sağlaması, yerel en küçük değerde takılıp kalmaması ve benzetilmiş tavlamanın çözüme hassas yakınsama özelliklerinden faydalanarak eniyileme gerçekleştirilmiştir. Bu çalışmada çözümü aranan problemler iki boyutlu düzgün şerit kesme problemleridir. Düzgün şeritler biçimindeki parçalar ana parça içine giyotin yapılmasına imkan vermeyecek şekilde yerleştirilmektedir. Yerleştirmenin yapılacağı ana parçanın alanı önceden sınırlandırılmıştır.

Genetik algoritmalarla farklı olasılıksal kök parametreleri, değişik nüfus büyüklükleri ve mutasyon oranları kullanılarak yapılan yerleştirme simülasyonları nüfus büyüklüğü artışlarının çözümü olumlu yönde etkilediğini ancak mutasyon oranı artışlarının çözüm üzerinde çok fazla etkili olmadığını göstermiştir. Değişen nüfus büyüklüğü değerleri için 60-80 nüfus büyüklükleri arasında bir artış gözlenmiştir. Bu da normalize uygunluk değerlerinin arttığını ve buna karşılık gelen fire değerlerinin azaldığını göstermektedir. Değişen mutasyon oranları için 0.3-0.9 mutasyon oranları arasında normalize uygunluk değerleri üzerinde herhangi bir değişim görülmemiştir. Ancak grafiklerdeki genel eğilim çok az da olsa mutasyon oranı arttıkça normalize edilmiş uygunluk değerlerinin azaldığını, buna karşılık gelen fire değerlerinin arttığını göstermektedir. 17 adet parçanın yerleştirilmesi için yapılan bu çalışmada fire değerleri %2 ve %11 arasında bir değişim göstermektedir.

29 adet parçanın yerleştirilmesi için genetik algoritmalar kullanılarak yapılan ikinci çalışmada farklı çaprazlama tekniklerinin çözüm üzerindeki etkileri incelenmiştir. Bir önceki çalışmadan elde edilen sonuçlara göre 0.7 mutasyon oranı ve 80 nüfus büyüklüğü sabit tutularak 6 farklı çaprazlama tekniği ile çalışma gerçekleştirilmiştir. Bu çalışma sonucunda elde edilen fire değerleri, çaprazlama tekniklerinin genetik algoritmanın çözümünde etkili olduğunu göstermiştir. Bu çalışmada en iyi fire değeri OBX tekniği kullanıldığında elde edilmiştir. Elde edilen en iyi fire değerinden en küçük fire değerine göre kullanılan teknikler CX, OX, PMX, UX, SJX olarak sıralanabilir. Nüfus içinde OBX tekniği kullanılarak elde edilen bireyler başka bir deyişle elde edilen olası çözümler diğer çaprazlama tekniklerine göre nüfus içinde ve nesiller boyunca daha çok değişim sergilemiştir. Böylelikle nüfus içerisinde yeterince çeşitlilik sağlanmış ve çözüm için arama uzayında daha geniş bir alan

taranmıştır. 29 adet parçanın yerleştirilmesi için yapılan bu çalışmada fire değerleri %5 ve %9 arasında bir değişim göstermektedir

Benzetilmiş tavlama yöntemiyle denge koşulu için kullanılan değişik komşuluk hareketleri sayısı, iki farklı komşuluk hareketi ve iki farklı soğutma planı kullanılarak aynı 17 parçaya sahip test problemleri üzerinde bir çalışma gerçekleştirilmiştir. Yapılan her çalışma değişik başlangıç koşulları için tekrarlanmıştır. Farklı komşuluk hareketleri ve farklı soğutma planları kullanılarak yapılan çalışmada komşuluk hareketlerinin problemin çözümü üzerindeki etkileri incelenmiştir. Yapılan yerleştirme simülasyonları, değişen başlangıç sıcaklıkları için yer değiştirme komşuluk hareketinin kaydırma komşuluk hareketine göre biraz daha iyi bir eğilime sahip uygunluk değerleri verdiğini göstermektedir. Bunun üzerine sadece yer değiştirme komşuluk hareketi kullanılarak, denge koşulu için değişik komşuluk hareketi sayısı ve farklı soğutma planları kullanılarak bir çalışma gerçekleştirilmiştir. Bu çalışmanın sonucunda denge koşulu için farklı komşuluk hareketi sayısının 3 alınması ve Lundy ve Mees soğutma planının kullanılması bu problemler için artan sıcaklık değerlerinde iyi çözümler vermektedir. BT ile 17 adet parçanın yerleştirilmesi için yapılan bu çalışmada fire değerleri %4 ve %21 arasında bir değişim göstermektedir.

Genetik algoritmalar ile çözümü aranan aynı 29 parçaya sahip test problemi için bu defa da benzetilmiş tavlama yöntemiyle çözüm aranmıştır. Bu çalışmada kullanılan parametre değerleri önceki çalışmalarda elde edilen en iyi değerlere sahip parametre değerleridir. Buna göre değişik başlangıç sıcaklıkları, yer değiştirme komşuluk hareketi ve Lundy ve Mees soğutma planı kullanılarak yapılan çalışmada elde edilen fire değerleri %10 ve %17 arasında değişmektedir.

Problemler için elde edilen fire değerlerinin bir arada verildiği Tablo 6.10 ve her problem ve her yöntem için ayrı ayrı en iyi uygunluk değerlerinin verildiği Şekil 6.42 ve Şekil 6.43 incelendiğinde, GA'nın bu problemler için aşağı sol algoritmasıyla birlikte kullanıldığında daha iyi çözümler verdiği görülmektedir.

## KAYNAKLAR

1. ANTONIOLETTI, M., 1996. Internet, [http://www.epcc.ed.ac.uk/computing/training/document\\_archive/GAs-course/node9.html](http://www.epcc.ed.ac.uk/computing/training/document_archive/GAs-course/node9.html). Erisim Tarihi. 18.02.2003.
2. BEASLEY, D., BULL, D.R., and MARTIN, R.R., 1993a. An Overview of Genetic Algorithms: Part 1, Fundamentals. University Computing, Vol.15(2), pp. 58-69, UK.
3. BEASLEY, D., BULL, D.R., and MARTIN, R.R., 1993b. An Overview of Genetic Algorithms: Part 2, Research Topics .University Computing, Vol. 15(4), pp. 170-181, UK.
4. BINGUL, Z., SEKMEN, A.S. and ZEIN, S., 1999. An Application of Multi-Dimensional Optimization Problems Using Genetic Algorithms. Proceedings of the IASTED International Conference Intelligent Systems and Control, Santa Barbara, CA, USA.
5. BINGUL, Z., SEKMEN, A.S. and ZEIN, S., 2000. Genetic Algorithms Applied to Real Time Multi-objective Optimization Problems. IEEE SoutheastCon 2000 Conference, Nashville, TN, USA.
6. CALLAHGAN, A. R., NAIR A. R. and LEWIS, K. E., 1999. An Extension of The Orthogonal Packing Problem Through Dimensional Flexibility. Proceedings of DETC'99:1999 ASME Design Engineering Technical Conferences, pp. 12-15.
7. CHAZELLE, B., 1983. The Bottom-Left Bin-Packing Heuristic:An Efficient Implementation. IEEE Transactions On Computers, Vol. 32, No. 8, pp. 697-707, USA.
8. COFFMAN, E.G., GAREY, M.R. and JOHNSON D.S., 1984. Approximation Algorithms for Bin-packing—an Updated Survey. Algorithms Design for Computer Systems Design, Springer, Vienna, pp.49-106.
9. DAGLI, C.H. and POSHYANONDA, P., 1997. New Approaches To Nesting Rectangular Patterns. Journal Of Intelligent Manufacturing, Vol. 8, pp. 177-190.
10. DAVIS, L., 1985. Applying Adaptive Search Algorithms to Epistatic Domains. Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp.162-164, Los Angeles.
11. DOWSLAND, K.A., 1993. Some Experiments with Simulated Annealing Techniques For Packing Problems. European Journal of Operational Research, Vol. 68, pp. 389-399.



12. DOWSLAND, K.A. and DOWSLAND, W.B., 1992. Packing Problems. *European Journal of Operational Research*, Vol.56, pp. 2-14.
13. DREZNER, Z. and WESOLOWSKY, G.O., 2003. Network Design: Selection and Design of Links and Facility Location. *Transportation Research Part A*, Vol. 37, pp 241-256.
14. DYCKHOFF, H., 1990. Typology of Cutting and Packing Problems. *European Journal of Operational Research*, Vol. 44, pp. 145-159.
15. FAINA, L., 1999. Application of Simulated Annealing to The Cutting Stock Problem. *European Journal of Research*, Vol. 114, pp. 542-556.
16. GEN, M., CHENG, R. and OREN, S.S., 2001. Network Design Techniques Using Adapted Genetic Algorithms. *Advances in Engineering Software*, Vol. 32, pp. 731-744.
17. GILMORE, P.C. and GOMORY, R.E., 1965. Multistage Cutting Stock Problems of Two and More Dimensions, *Operations Research*, Vol 13, pp.94-120.
18. GOLDBERG, D.E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company Inc.ISBN:0-201-15767-5.
19. HAESSLER, R.W. and SWEENEY, P.E., 1991. Cutting Stock Problems and Solution Procedures. *European Journal of Operational Research*, Vol. 54, pp. 141-150, North-Holland.
20. HINXMAN, A.I., 1980. The Trim Loss and Assortment Problems. *European Journal of Operational Research*, Vol. 5, pp.8-18.
21. HOPPER, E. and TURTON, B., 1997. Application of Genetic Algorithms to Packing Problems-A Review. *Proceedings of The 2nd On-line World Conference on Soft Computing in Engineering Design and Manufacturing*, Springer Verlag, pp. 279-288, London.
22. HOPPER, E. and TURTON, B., 1999. A Genetic Algorithm for A 2D Industrial Packing Problem. *Computers & Industrial Engineering*, Vol. 37, pp. 375-378.
23. HOPPER, E., 2000. *Two Dimensional Packing Utilising Evolutionary Algorithms and Other Meta-Heuristic Methods*. University of Wales, Cardiff School of Engineering, Phd.Thesis, London.
24. HOPPER, E. and TURTON, B., 2001. A Review of The Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems. *Artificial Intelligence Review*, Vol.16, pp. 257-300.



25. HOUCK, C.R., JOINES, J. And KAY, M., 1996. A Genetic Algorithm for Function Optimization: A Matlab Implementation. *ACM Transactions on Mathematical Software*.
26. JAKOBS, S., 1996. On Genetic Algorithms for The Packing of Polygons. *European Journal of Operational Research*, Vol. 88, pp. 165-181.
27. KÄMPKE, T., 1988. Simulated Annealing: Use of A New Tool in Bin-packing. *Annals of Operations Research*, Vol. 16, pp. 327-332.
28. KIRKPATRICK, S., GELATT, C.D. and VECCHI, M.P., 1983. Optimization by Simulated Annealing. *Science, New Series*, Vol. 220, pp. 671-680.
29. LAI, K.K. and CHAN, J.W.M., 1997. Developing A Simulated Annealing Algorithm for The Cutting Stock Problem. *Computers and Industrial Engineering*, Vol. 32, pp. 115-127, Great Britain.
30. LEITCH, D.D., 1995. A New Genetic Algorithm for The Evolution of Fuzzy Systems. PhD Thesis, University of Oxford, UK.
31. LEUNG, T.W., YUNG, C.H. and CHAN, C.K., 1999. Applications of Genetic Algorithm and Simulated Annealing to The Two-Dimensional Non-Guillotine Cutting Stock Problem. Presented at IFORS'99, Beijing China.
32. LEUNG, T.W., YUNG, C.H. and TROUTT, M.D., 2001. Applications of Genetic Search and Simulated Annealing to The Two-Dimensional Non-Guillotine Cutting Stock Problem. *Computers and Industrial Engineering*, Vol.40, pp. 201-214.
33. LINDFIELD G., and PENNY J, 1994. *Numerical Methods Using for Matlab*. University of Aston, Birmingham, ISBN: 0-13-030966-4.
34. LUTFIYYA, H., MCMILLIN, B., POSHYANONDA, P. and DAGLI, C., 1992. Composite Stock Cutting Through Simulated Annealing. *Mathematical Computing Modelling*, Vol. 16, no. 1, pp. 57-74, Great Britain.
35. LIU, D. and TENG, H., 1999. An Improved BL-Algorithm for Genetic Algorithm of The Orthogonal Packing of Rectangles. *European Journal of Operational Research*, Vol. 112, pp. 413-420.
36. MAN, K.F., TANG, K.S. and KWONG, S., 1996. Genetic Algorithms: Concepts and Applications. *IEEE Transactions on Industrial Electronics*, Vol. 43, No. 5, pp. 519-533.
37. MARTELLO, S. and TOTH, P., 1990. *Knapsack Problems: Algorithms and Computer Implementations*. J. Wiley and Sons, Chichester.
38. PHAM, D.T. and KARABOGA, D., 2000. *Intelligent Optimisation Techniques*. Springer - Verlag London, ISBN:1852330287.

39. SMITH, D., 1985. Bin Packing with Adaptive Search. Proceedings of an International Conference on Genetic Algorithms and Their Applications, pp. 202-206, Lawrence Erlbaum.
40. TAM, K.Y., 1992. Genetic Algorithms, Function Optimization and Facility Desing. European Journal of Operational Research. Vol. 63, pp. 322-346.
41. THELMA, D.M. and PANOS, M.P., 1997. Simulated Annealing and Genetic Algorithms for The Facility Layout Problem: A Survey. Computational Optimization and Applications, Vol. 7, pp. 111-126. Boston.
42. WHITELY, D. and HANSON, T., 1989. Optimising Neural Networks Using Faster, More Accurate Genetic Search. Proc. 3rd Intelligence Conference on Genetic Algorithms and Their Applications, pp. 370-374. George Mason University.



## EKLER

### EK-A: Program Listeleri

```
*****
Kırma GA Ana Programı
*****

%*****TANIMLAMALAR*****

clear all
clc
rand('state',0);
parcalar=[89 37; 56 28; 12 66; 13 24; 30 25; 8 10; 5 1; 35 9; 36 9; 20 78; 26 32; 17
104; 52 48; 73 33; 14 55; 24 45; 55 15; 18 36; 107 21; 54 45; 81 33; 10 32;
8 10; 16 27; 22 17; 65 62; 66 12; 15 61; 120 49]; % 29 Parça.

levha=[200 200]; %Yerleştirmenin yapılacağı ana parçanın boyutları.
[n m]=size(parcalar); % n=Parça sayısı.
brs=80; % brs=Birey sayısı.
sonnesil=1000; % sonnesil=Toplam nesil sayısı.
mutoran= 0.9; % mutoran=Mutasyon oranı.
fireler=[];
toplamuygunluk=[ ];
ebuynluk=[ ];
uygunlukdag=zeros(sonnesil,brs);
oncekiuygunluk=0;
toplamalan=[ ];
fire=2;

%*****Kırma GA*****

nufus=uretperm(brs,n); % Başlangıç nüfusu üretiliyor.

% Geliştirilmiş AS algoritmasıyla permütasyon hesabi yapılıyor.
%perm=Parçaların yerleşeceği sırayı gösteriyor.

for i=1:brs
    perm=nufus(i,:);
    [fire,parca,yerlesme,alan]=as(parcalar,levha,perm);
    fireler=[fireler fire];
end

uygunluk=1./(eps+fireler) % Uygunluk hesabi yapılıyor. eps=2.2204e-016
```

eniyiperm=nufus(1,:); % Nüfusun ilk elemanı en iyi permütasyon seçiliyor.

**% Nesil başlıyor.**

for nesil=1:sonnesil

% Seçim işlemi

secnufus=secim(nufus,uygunluk,brs);

% Çaprazlama ve Mutasyon işlemi.

for k=1:2:brs-1

[nufus(k,:) nufus(k+1,:)]=obx(secnufus(k,:),secnufus(k+1,:));

nufus(k ,:)=mutasyon(nufus(k ,:),mutoran);

nufus(k+1,:)=mutasyon(nufus(k+1,:),mutoran);

end

nufus(brs,:)=eniyiperm; % Bir önceki nüfustaki en iyi birey saklanıyor.

fireler=[];

for i=1:brs

perm=nufus(i,:);

[fire,parca,yerlesme,alan]=as(parcalar,levha,perm);

fireler=[fireler fire];

end

uygunluk=1./(eps+fireler); % Uygunluk hesabi yapılıyor.

enbuyukuyg=max(uygunluk);

[a iyi]=min(fireler);

iyiperm=nufus(iyi,:);

% Elde edilen bireyin bir önceki nüfustaki en iyi bireyden iyi olup olmadığına bakılıyor.

if enbuyukuyg > oncekiuygunluk

oncekiuygunluk=enbuyukuyg;

eniyiperm=iyiperm;

end

ebuygunluk=[ebuygunluk max(uygunluk)]; % Her nesildeki en büyük uygunluk

toplamuygunluk=[toplamuygunluk sum(uygunluk)]; % Toplam uygunluk değeri.

ortalamauygunluk=toplamuygunluk/brs; % Ortalama uygunluk değeri.

toplamalan=[toplamalan alan]; % Toplam alan her nesilde hesaplanıyor.

uygunlukdag(nesil,:)=uygunluk; % Tüm nesil boyunca uygunluk değerleri saklanıyor.

end

**% Nesil tamamlandı.**

% En iyi birey için geliştirilmiş AS algoritmasıyla permütasyon hesabi yapılıyor.

[fire,parca,yerlesme,alan]=as(parcalar,levha,eniyiperm);

ciz(parca,yerlesme);

% Yerleşen parçalar çiziliyor.

\*\*\*\*\*

## Geliştirilmiş AS Algoritması Fonksiyonu

\*\*\*\*\*

```
function [fire,parca,yerlesme,alan]=as(parcalar,levha,perm);
```

```
xref=levha(1);          % Ana parçanın yatay eksendeki uzunluğu.
yref=levha(2);          % Ana parçanın dikey eksendeki uzunluğu.
xm=zeros(yref+1,xref); % Ana parça 0'lardan oluşmuş bir matris gibi tanımlı.
[n m]=size(parcalar);
yerlesme=[];
```

```
% Parçaları yerleşeceği permütasyon sırasına göre sıralıyor.
```

```
for i=1:n
    parca(i,1)=parcalar(perm(i),1);
    parca(i,2)=parcalar(perm(i),2);
end
```

```
% İlk parça ana parçanın kösesine yerleştiriliyor.
```

```
yerlesme(1,1)=1;          % Yerleşen parça sayısı.
yerlesme(1,2)=1;          % Yerleşen parçanın sol alt kösesinin x koordinatı..
yerlesme(1,3)=1;          % Yerleşen parçanın sol alt kösesinin y koordinatı.
xm=parcayerles(xm,1,1,parca(1,1),parca(1,2)); %1.parça kösede.

levhaalan=xref*yref;      % Ana parçanın toplam alanı.
alan=parca(1,1)*parca(1,2); % Yerleştirilen ilk parçanın alanı.
yerlesensayi=1;
```

```
for i=2:n
    xp=parca(i,1);
    yp=parca(i,2);
    x0=xref-xp+1; y0=yref+1; % Sıradaki parça ana parçanın dış sağ üst kösesine
                               yerleştiriliyor.
```

```
y=[ ];
otele=1;
hareket=1;
```

```
while hareket
```

```
% ***** Aşağı Hareket*****
```

```
if y0>1
    while (sum(xm(y0-1,x0:x0+xp-1))==0)
        y0=y0-1; %Parça aşağıya doğru iniyor.
        if y0==1
            break
        end % if
    end %while
```

```

end % if

%***** Sola Hareket*****
yt=y0+yp-1;
if yt>yref
    yt=yref;
end
if x0>1
    if sum(xm(y0:yt,x0-1))==0
        yatay=1;
        x0=x0-1;
    else
        yatay=0;
    end
else
    yatay=0;
end
if y0>1
    if (sum(xm(y0-1,x0:x0+xp-1))==0)
        dikey=1;
    else
        dikey=0;
    end
else
    dikey=0;
end
if dikey | yatay          % Parça yatayda veya dikeyde hareket edebiliyor mu?
    hareket=1;
else
    hareket=0;
end
end

if y0+yp-1<=yref
    xm=parcayerles(xm,x0,y0,xp,yp);
    yerlestir=1;
else yerlestir=0;
end
%-----

if yerlestir==1
    yerlesensayi=yerlesensayi+1;
    yerlesme(yerlesensayi,1)=i;
    yerlesme(yerlesensayi,2)=x0;
    yerlesme(yerlesensayi,3)=y0;
    alan=alan+parca(i,1)*parca(i,2); % Her yerleşen parçanın alanı önceki alanların
                                        toplamına ekleniyor.
end %yerlestir

```

```
end % for
fire=(levhaalan-alan)/levhaalan; % Her yerleřtirme simülasyonundan sonra fire
hesaplanıyor.
ciz(parca,yerlesme) % Parçalar verilen sıraya ve koordinatlarına göre
çiziliyor.
```



\*\*\*\*\*

## Kırma BT Ana Programı

\*\*\*\*\*

%\*\*\*\*\*TANIMLAMALAR\*\*\*\*\*

```
clear all
clc
rand('state',0);
parcalar=[89 37; 56 28; 12 66; 13 24; 30 25; 8 10; 5 1; 35 9; 36 9; 20 78; 26 32; 17
104;
52 48; 73 33; 14 55; 24 45; 55 15; 18 36; 107 21; 54 45; 81 33; 10 32; 8 10;
16 27; 22 17; 65 62; 66 12; 15 61; 120 49]; % 29 Parça.
```

```
levha=[200 200]; % Yerleştirmenin yapılacağı ana parçanın boyutları.
[parcasayi z]=size(parcalar); % parcasayi=Parça sayısı.
```

```
nesil=1;
sonnesil=1000; % Toplam öteleme sayısı.
gezmax=5; % Denge koşulu için en büyük komşuluk hareketi sayısı.
itemp=0.80; % Başlangıç sıcaklığı.
ftemp=0.01; % Son sıcaklık.
fireler=[ ];
vu=exp(log(2/parcasayi)/sonnesil); % Başlangıçta vektör uzaklığı=parcasayi.
cik=0;
```

%\*\*\*\*\*Kırma BT\*\*\*\*\*

```
alfa=exp(log(ftemp/itemp)/sonnesil); % Oransal soğutma planı için sıcaklık azaltma
oranı.
```

```
perm=randperm(parcasayi); % Başlangıç çözümü.
```

```
% Geliştirilmiş AS algoritmasıyla permütasyon hesabi yapıyor.
%perm=Parçaların yerleşeceği sırayı gösteriyor.
```

```
[fire,parca,yerlesme,alan]=as(parcalar,levha,perm);
fireler=[fire ];
```

```
temp=itemp; % İlk sıcaklık değeri atanıyor.
tanchor=parcasayi; % Başlangıçta tüm parçalar yer değiştirebiliyor.
```

```
while (cik==0) % Dış döngü için. Durdurma kriteri sağlandı mı?
temp = alfa * temp; % Oransal soğutma planı için 0<alfa<1
tanchor = vu * tanchor; % Vektör uzaklığı için sıcaklık devamlı azalıyor. Yani
vektör uzaklığı kısılıyor.
anchor = round(tanchor);
```

```
komsusayi=0;
```



```

denge=0;          % Komşuluk hareketleri başladı.

while (denge==0) % İç döngüye gelindi mi? Denge koşulu için en büyük
                  komşuluk hareketi sayısı sağlandı mi?
    if (komsusayi == 0)
        yeniper = komsu(perm,parcasayi,parcasayi); % vu=parcasayi
    else
        yeniper = komsu(perm,parcasayi,anchor);
    end

% Fire değeri hesaplanıyor.

[yenifire,parca,yerlesme,alan]=as(parcalar,levha,yeniper);

delta=yenifire-fire;

% Yeni çözümlerin kabul edilmesi.

if delta<0          % Yeni fire daha iyi ise kabul edilir.
    perm=yeniper;
    fire=yenifire;
else
    if (rand < exp(-delta/temp)) % Yeni fire daha kötü ise olasılık dahilinde
        % kabul edilir.
        perm=yeniper;
        fire=yenifire;
    end
end

komsusayi=komsusayi+1;
if ( komsusayi == gezmax ) % Komsusayi=Denge koşulu için en büyük
    % komşuluk hareketi sayısı?
    denge=1; % Denge koşulu sağlandı.
end
end
fireler=[fireler fire];
nesil=nesil+1;

if nesil==sonnesil
    cik=1;
end

end % while (cik==0) Durdurma kriteri sağlandı.

ciz(parca,yerlesme); % En son elde edilen çözüm kabul edilir ve çizilir.

```

**EK-B: Test Problemleri**

<b>TEST PROBLEMLERİ</b>										
<b>No</b>	<b>1. Test problemi</b>		<b>2. Test problemi</b>		<b>3. Test problemi</b>		<b>4. Test problemi</b>		<b>5. Test problemi</b>	
	<b>En</b>	<b>Boy</b>	<b>En</b>	<b>Boy</b>	<b>En</b>	<b>Boy</b>	<b>En</b>	<b>Boy</b>	<b>En</b>	<b>Boy</b>
1	33	132	29	137	35	86	41	12	21	84
2	64	41	11	31	17	54	25	34	56	67
3	103	32	75	22	36	81	19	44	123	86
4	59	9	22	107	70	88	115	22	30	17
5	44	38	63	132	42	117	25	51	26	61
6	123	29	44	9	5	32	16	22	51	44
7	57	62	31	27	12	42	71	22	25	42
8	32	8	55	18	25	15	44	109	79	32
9	25	26	28	74	11	55	41	29	19	66
10	53	39	58	58	40	10	90	87	36	82
11	15	81	80	16	26	78	35	137	43	34
12	17	18	70	14	44	29	31	68	66	27
13	42	63	38	36	77	40	129	44	36	8
14	9	50	24	27	86	49	36	15	23	19
15	44	91	39	68	79	64	17	29	13	64
16	90	68	99	22	9	30	19	54	62	48
17	66	41	161	41	121	34	146	25	89	45

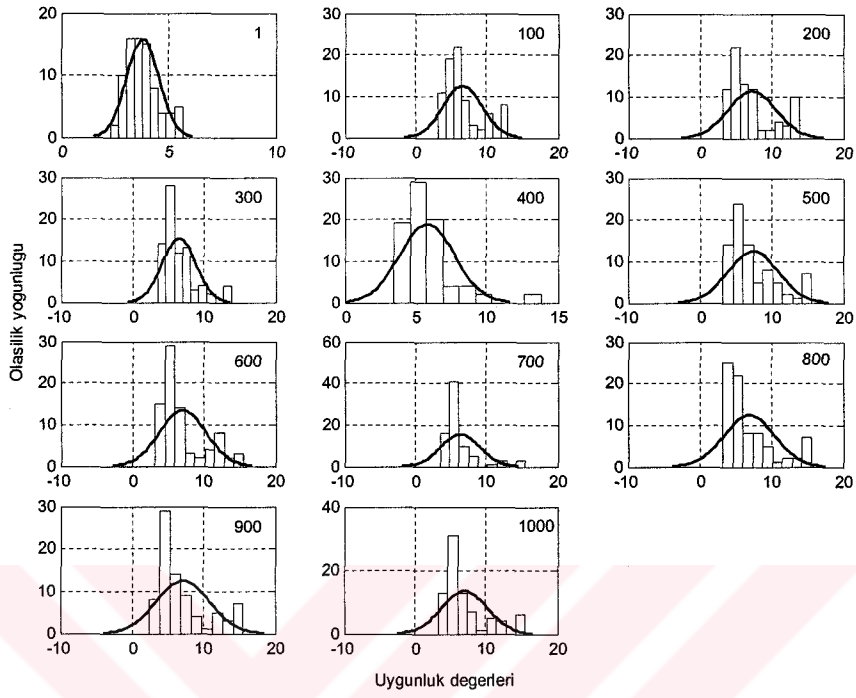
<b>No</b>	<b>Test Problemi</b>		<b>No</b>	<b>Test Problemi</b>	
	<b>En</b>	<b>Boy</b>		<b>En</b>	<b>Boy</b>
1	89	37	16	24	45
2	56	28	17	55	15
3	12	66	18	18	36
4	13	24	19	107	21
5	30	25	20	54	45
6	8	10	21	81	33
7	5	1	22	10	32
8	35	9	23	8	10
9	36	9	24	16	27
10	20	78	25	22	17
11	26	32	26	65	62
12	17	104	27	66	12
13	52	48	28	15	61
14	73	33	29	120	49
15	14	55			

**EK-C: Altı Farklı Çaprazlama Tekniği kullanılarak elde edilen uygunluk değerlerine ait olasılıksal değerler.**

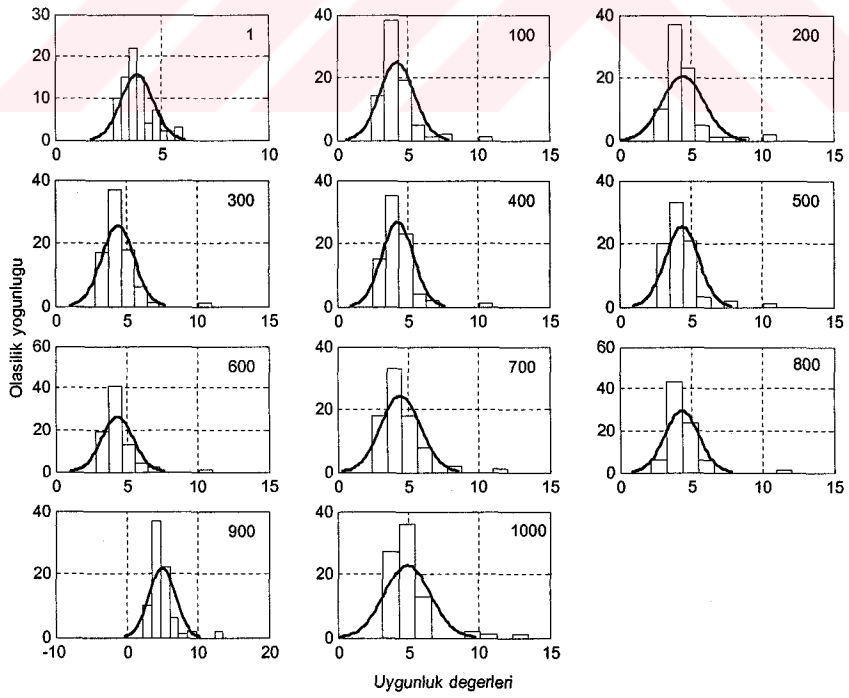
NO	Çaprazlama türleri	Nesil no	Uygunluk değerlerine ait olasılıksal değerler					
			Aritmetik ortalama	Medyan	Geometrik ortalama	Standart sapma	En büyük değer	En küçük değer
1	OBX	1	3,8502	3,6746	3,7784	0,7633	6,0698	2,1674
2	OBX	100	5,1112	4,4472	4,7629	2,3518	14,6843	2,5332
3	OBX	200	4,7759	4,4682	4,5419	1,9611	16,7224	2,8821
4	OBX	300	5,1914	4,9460	4,9306	1,9822	16,7224	2,6508
5	OBX	400	4,8772	4,1258	4,5538	2,3809	16,7224	2,6513
6	OBX	500	5,2050	4,5132	4,8517	2,3501	16,7224	2,3770
7	OBX	600	5,5520	5,1494	5,1161	2,8012	16,7224	2,7012
8	OBX	700	5,4375	5,0697	5,1243	2,1983	16,7224	2,6804
9	OBX	800	7,0810	5,3211	6,1763	4,0552	16,7224	2,8129
10	OBX	900	5,0124	4,4746	4,6806	2,4559	16,7224	2,7461
11	OBX	1000	4,9849	4,7308	4,7653	1,8333	16,7224	2,3369
12	CX	1	3,7965	3,6580	3,7240	0,7629	5,6521	2,2789
13	CX	100	6,6021	5,7594	6,1257	2,7588	12,9912	3,2242
14	CX	200	7,2224	6,1430	6,5764	3,3172	13,8169	3,1611
15	CX	300	6,4927	5,6557	6,1522	2,3746	13,8169	3,5705
16	CX	400	5,8170	5,4811	5,5691	1,9357	13,8169	3,4801
17	CX	500	7,3917	6,1434	6,7163	3,5001	15,5824	3,2229
18	CX	600	7,0250	5,7188	6,4246	3,2616	15,5824	3,1893
19	CX	700	6,3124	5,4340	5,9134	2,7262	15,5824	3,4941
20	CX	800	6,8421	5,5387	6,1490	3,5329	15,5824	3,2281
21	CX	900	7,1400	5,7497	6,3574	3,7437	15,5824	2,4959
22	CX	1000	6,9957	5,6133	6,4397	3,2047	15,5824	3,2887
23	OX	1	3,8483	3,6926	3,7825	0,7502	6,0105	2,7045
24	OX	100	4,2702	4,1439	4,1376	1,2171	10,9890	2,4471
25	OX	200	4,4729	4,1220	4,3059	1,4634	10,9890	2,4805
26	OX	300	4,3910	4,0821	4,2781	1,1331	10,9890	2,8100
27	OX	400	4,2852	4,0831	4,1721	1,1235	10,9890	2,5369
28	OX	500	4,3704	4,2374	4,2524	1,1665	10,9890	2,6415
29	OX	600	4,3265	4,1574	4,2176	1,1130	10,9890	2,8045
30	OX	700	4,4588	4,2211	4,2909	1,3955	12,0192	2,4553
31	OX	800	4,3129	4,1795	4,1946	1,1832	12,0192	2,1881
32	OX	900	4,9417	4,4959	4,7127	1,8245	13,5227	2,2802
33	OX	1000	4,9347	4,6154	4,7503	1,6078	13,5227	3,2003
34	PMX	1	3,9021	3,7849	3,8096	0,8894	6,4819	2,3245
35	PMX	100	3,9806	3,7924	3,8543	1,1369	10,5014	2,3201
36	PMX	200	4,2886	4,0950	4,1599	1,2502	12,1840	2,2603
37	PMX	300	4,2306	4,0467	4,1047	1,3091	13,1234	2,7613
38	PMX	400	4,0025	3,7945	3,8733	1,2975	13,1234	2,4326
39	PMX	500	4,4294	4,2466	4,2496	1,6076	13,1234	2,5575
40	PMX	600	4,4932	3,9519	4,2608	1,7943	13,1234	2,2037
41	PMX	700	4,2389	3,9775	4,0949	1,3594	13,1234	2,6923
42	PMX	800	4,2199	4,0742	4,0804	1,3134	13,1234	2,2873
43	PMX	900	4,2007	4,0112	4,0535	1,3715	13,1234	2,3773
44	PMX	1000	4,6989	4,3550	4,5719	1,3202	13,1234	2,9960
45	UX	1	3,9053	3,6972	3,8286	0,7996	6,0205	2,5165
46	UX	100	4,1949	3,9600	4,1076	0,9666	9,6899	2,5094
47	UX	200	4,8136	4,6469	4,6870	1,1537	10,3923	2,8944

NO	Çaprazlama türleri	Nesil no	Aritmetik ortalama	Medyan	Geometrik ortalama	Standart sapma	En büyük değer	En küçük değer
48	UX	300	4,8476	4,6732	4,7456	1,0845	10,3923	3,3571
49	UX	400	4,8301	4,7377	4,7027	1,1882	10,3923	2,7724
50	UX	500	4,7119	4,4881	4,5917	1,1463	10,3950	3,0243
51	UX	600	4,8201	4,5718	4,6909	1,2207	10,3950	3,0941
52	UX	700	4,5263	4,3690	4,4096	1,1556	10,9890	2,9319
53	UX	800	4,6414	4,4905	4,4978	1,2909	10,9890	2,7770
54	UX	900	4,9116	4,8283	4,7722	1,2953	10,9890	2,8464
55	UX	1000	4,6148	4,4041	4,5069	1,1212	10,9890	3,0130
56	SJX	1	4,0224	3,9926	3,9244	0,9173	7,4724	2,1324
57	SJX	100	4,0420	3,9390	3,9478	0,9444	9,0888	2,1755
58	SJX	200	3,8810	3,6234	3,7944	0,9322	9,0888	2,5259
59	SJX	300	4,0643	3,8560	3,9311	1,1348	9,3262	2,2231
60	SJX	400	3,8250	3,7152	3,7303	0,9926	10,0781	2,3802
61	SJX	500	3,9130	3,7817	3,8114	1,0299	10,0781	2,4480
62	SJX	600	4,0759	3,9313	3,9673	1,0644	10,0781	2,5487
63	SJX	700	4,1131	3,8954	3,9926	1,1183	10,0781	2,4392
64	SJX	800	3,9283	3,7856	3,8087	1,0937	10,0781	2,4066
65	SJX	900	3,9259	3,6839	3,8119	1,0852	10,0781	2,2873
66	SJX	1000	4,0701	3,7880	3,9542	1,0985	10,0781	2,4350

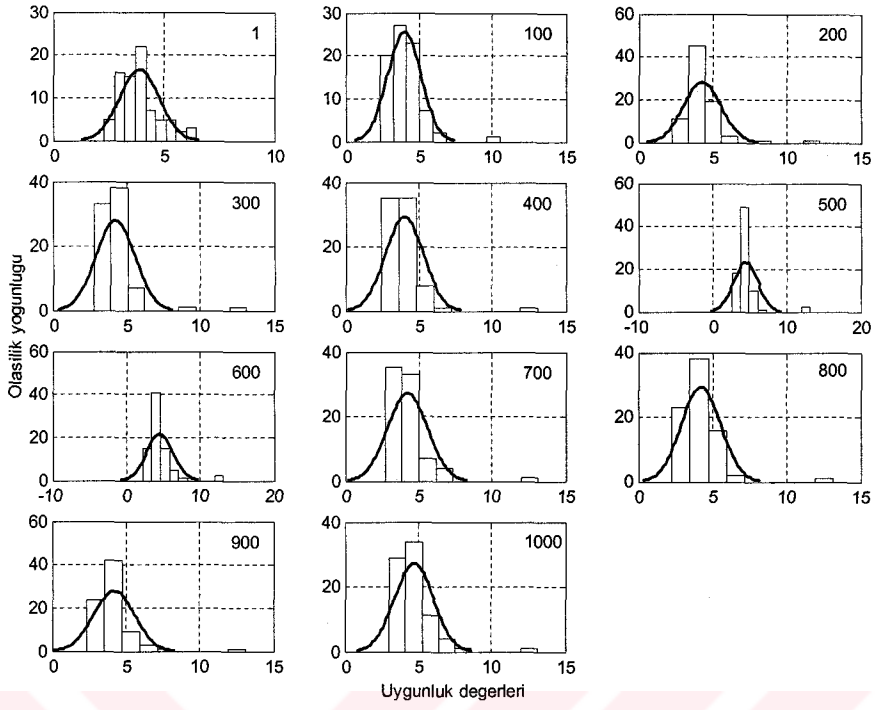
## EK-D: Farklı Çaprazlama Tekniklerinin Histogramları ve Olasılık Yoğunluk Fonksiyonları



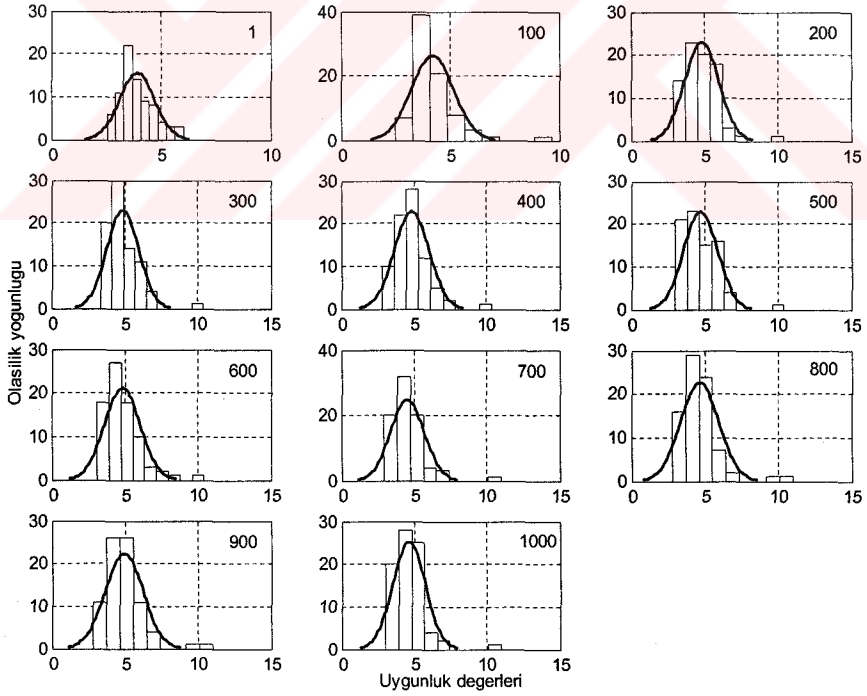
### CX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları



### OX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları.



PMX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları.



UX tekniğine ait histogram ve olasılık yoğunluk fonksiyonları.

## ÖZGEÇMİŞ

1977 yılında Kocaeli, İzmit'te doğdu. İlk ve orta öğrenimini İzmit'te tamamladı. 1995 yılında Kocaeli Teknik Lisesi'nden mezun olarak lise öğrenimini tamamladı. 1995 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümü'nden 1999 yılında mezun oldu. Eylül 2000'de Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı'nda yüksek lisans öğrenimine başladı.

2000 yılında, Kocaeli Üniversitesi Teknik Eğitim Fakültesi, Elektronik Bilgisayar Eğitimi Bölümü'nde Araştırma Görevlisi olarak çalışmaya başlamıştır.