



SAVUNMA YARIÇAPLI İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMLERİ İÇİN
ÇOK AMAÇLI GENETİK ALGORİTMA ÖNERİSİ

Enes ÇALIŞKAN

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

Ocak - 2019

SAVUNMA YARIÇAPLI İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMLERİ İÇİN
ÇOK AMAÇLI GENETİK ALGORİTMA ÖNERİSİ

Enes ÇALIŞKAN

Kütahya Dumlupınar Üniversitesi
Lisansüstü Öğretim ve Sınav Yönetmeliği Uyarınca
Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman: Dr. Öğr. Üyesi İhsan EROZAN

Ocak - 2019

KABUL VE ONAY SAYFASI

Enes ÇALIŞKAN'ın YÜKSEK LİSANS TEZİ olarak hazırladığı SAVUNMA YARIÇAPLI İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMLERİ İÇİN ÇOK AMAÇLI GENETİK ALGORİTMA ÖNERİSİ başlıklı bu çalışma, jürimizce Kütahya Dumlupınar Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

09/01/2019

Prof. Dr. Önder UYSAL
Enstitü Müdürü, Fen Bilimleri Enstitüsü

Prof. Dr. Özden ÜSTÜN
Bölüm Başkanı, Endüstri Mühendisliği Bölümü

Dr. Öğr. Üyesi İhsan EROZAN
Danışman, Endüstri Mühendisliği Bölümü

Sınav Komitesi Üyeleri

Doç. Dr. Feriştah ÖZÇELİK
Endüstri Mühendisliği Bölümü, Osmangazi Üniversitesi

Doç. Dr. Şafak KIRIŞ
Endüstri Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi

Dr. Öğr. Üyesi İhsan EROZAN
Endüstri Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi

ETİK İLKE VE KURALLARA UYGUNLUK BEYANI

Bu tezin hazırlanmasında akademik kurallara riayet ettiğimizi, özgün bir çalışma olduğunu ve yapılan tez çalışmasının bilimsel etik ilke ve kurallara uygun olduğunu, çalışma kapsamında teze ait olmayan veriler için kaynak gösterildiğini ve kaynaklar dizininde belirtildiğini, Yüksek Öğretim Kurulu tarafından kullanılmak üzere önerilen ve Kütahya Dumlupınar Üniversitesi tarafından kullanılan İntihal Programı ile tarandığını ve benzerlik oranının %4 çıktığını beyan ederiz. Aykırı bir durum ortaya çıktığı takdirde tüm hukuki sonuçlara razı olduğumuzu taahhüt ederiz.

Dr. Öğr. Üyesi İhsan EROZAN

Enes ÇALIŞKAN

SAVUNMA YARIÇAPLI İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMLERİ İÇİN ÇOK AMAÇLI GENETİK ALGORİTMA ÖNERİSİ

Enes ÇALIŞKAN

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2019

Tez Danışmanı: Dr. Öğr. Üyesi İhsan EROZAN

ÖZET

Yerleşim problemleri günlük hayatın farklı alanlarında karşılaşılan önemli en iyileme problemlerinden biridir. Tesis yerleşim problemleri, bazı lojistik problemleri ve kesme ve paketleme problemleri yerleşim problemlerine örnek gösterilebilir. Yerleşim problemleri NP-Hard bir yapıya sahiptir. Bu tip problemleri makul zamanda çözmek için son zamanlarda meta-sezgisel algoritmalar tercih edilmektedir. Bu tez çalışmasında yerleşim problemlerinin özel bir çeşidi olan ortogonal yerleşim problemi incelenmiştir. Bu problem için bu tez çalışmasında çok amaçlı bir model önerilmiştir. Önerilen model literatürden farklı olarak iki boyutlu sınırsız bir alanı ve savunma yarıçapı faktörünü kullanmaktadır. Önerilen model iki farklı metotla skalerleştirilmiş ve genetik algoritma ile çözülmüştür. Bir deneysel çalışma modelin tatmin edici sonuçlar elde ettiğini göstermiştir.

Anahtar Kelimeler: Çok amaçlı optimizasyon, Genetik algoritma, Ortogonal yerleşim, Savunma yarıçapı.

A PROPOSAL OF A MULTI OBJECTIVE GENETIC ALGORITHM FOR A TWO-DIMENSIONAL ORTHOGONAL PLACEMENT PROBLEMS WITH DEFENSIVE RADIUS

Enes ÇALIŞKAN

Industrial Engineering, M.S. Thesis, 2019

Thesis Supervisor: Asist. Prof. Dr. İhsan EROZAN

SUMMARY

Placement problems are one of the important optimization problems in the real world. Facility layout problems, some logistic problems, cutting and packaging problems can be given as an example of the placement problems. The placement problems have an NP-Hard structure. Meta-heuristic algorithms are recently preferred to solve such problems in an acceptable time. In this thesis, the orthogonal placement problem which is a special variation of placement problems has been investigated. A multi-objective model has been proposed for the orthogonal placement problem in this thesis. The proposed model, unlike the literature, uses a two-dimensional unlimited area and a defensive radius factor. The proposed model was scalarised by two different methods and solved by a genetic algorithm. An experimental study has shown that the model can obtain satisfactory results.

Keywords: Multi-objective optimization, Genetic algorithm, Orthogonal placement, Defensive radius.

TEŐEKKÜR

Yüksek Lisans tezinin hazırlanması süresince bana her türlü desteęi, kolaylıęı ve sabrı gösteren Sayın danışman hocam Dr. Öğr. Üyesi İhsan EROZAN'a teşekkür ederim.

Yüksek Lisans tezimi hazırlama süresince bana olan güven ve desteklerinden dolayı aileme ve sevgili eşim Elmira ÇALIŐKAN'a teşekkür ederim.

Eęitim hayatım boyunca desteęini ve ilgisini hiç eksik etmeyen ve üzerimde büyük emeęi olan İlker Ozan KOÇ'a ve Mete ALİKALFA'ya teşekkür ederim.



İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ	xi
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMİ.....	14
3.1. Problem	14
3.2. Çözüm Metodu.....	18
3.2.1. Ağırlıklı toplam skalerleştirme yöntemi	20
3.2.2. Epsilon kısıt yöntemi	20
3.2.3. Melez skalerleştirme yöntemi	21
3.2.4. Konik skalerleştirme yöntemi	21
3.3. Problem İçin Önerilen Model.....	26
3.4. Önerilen Çok Amaçlı Genetik Algoritma.....	30
3.4.1. Kromozomun yapısı	32
3.4.2. Başlangıç popülasyonunu oluşturma	32
3.4.3. Uygunluk değeri.....	32
3.4.4. Seçim operatörü	33
3.4.5. Elitizm operatörü	33
3.4.6. Çaprazlama operatörü	33
3.4.7. Mutasyon operatörü	34
3.5. Yerleşim Algoritması.....	36
4. ÖNERİLEN MODELİN DEĞERLENDİRİLMESİ	43
5. SONUÇ VE DEĞERLENDİRMELER	49
KAYNAKLAR DİZİNİ.....	51

İÇİNDEKİLER (devam)**Sayfa****EKLER**

- Ek 1. 30 Bloktan Oluşan Örneğin Genetik Algoritma ve Skalerizasyon Parametreleri
- Ek 2. 30 Bloktan Oluşan Örneğin Yerleşim Görünümleri
- Ek 3. 8 Örnek Problemin Genetik Algoritma Parametreleri, Skalerizasyon Parametreleri ve Çözüm Sonuçları
- Ek 4. 8 Örnek Problemin İlişki Değerleri
- Ek 5. 8 Örnek Problemin Yerleşim Görünümleri

ŞEKİLLER DİZİNİ

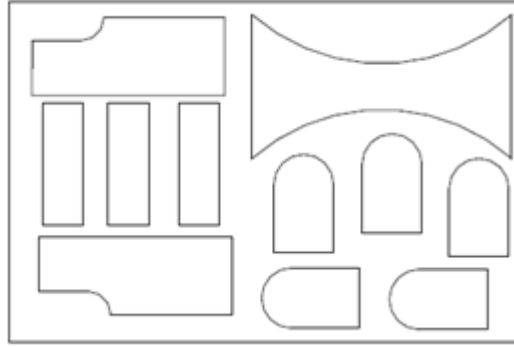
<u>Sekil</u>	<u>Sayfa</u>
1.1. Pastal kesim planı.....	1
3.1. Dikdörtgen bloklar	14
3.2. İki dikdörtgen bloğun merkezleri arasındaki mesafe	15
3.3. Amaç 1 ile Amaç 2'nin karşılaştırılması	16
3.4. İki dikdörtgen blok arasındaki en yakın mesafeler	17
3.5. Savunma yarıçapları dışında kalan alan.....	18
3.6. İki amaçlı bir problem için konveks olmayan görüntü kümesi	23
3.7. Aynı nokta üzerinde bulunan iki blok.....	26
3.8. Genetik algoritmaların akış şeması	31
3.9. 6 bloktan oluşan bir kromozom örneği	32
3.10. Elitizm işlemi	33
3.11. İki noktalı çaprazlama	34
3.12. Birinci kısımda gerçekleşen mutasyon.....	35
3.13. İkinci kısımda gerçekleşen mutasyon	35
3.14. Sınır arama algoritmasının genetik algoritma ile beraber kullanımı.....	36
3.15. Yerleşim algoritması akış şeması	37
3.16. Çözüm alanına yerleştirilmiş bir bloğun yerleştirme sınırları.....	37
3.17. Çözüm alanına yerleştirilmiş bir bloğun sınırının yönleri	38
3.18. Sınırın yönüne göre bir bloğun yerleştirilme konumu	39
3.19. Örtüşme durumu	39
3.20. İki blok arasındaki örtüşme durumları	40
3.21. Kesişmeyen alanın noktalar ile gösterilmiş hali.....	42
4.1. Problem verilerin alındığı ve algoritma parametrelerinin belirlendiği giriş arayüzü	43
4.2. Genetik algoritmanın çalıştırıldığı ve sonuçlarının görsel olarak gösterildiği ikinci arayüz	45

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1. İncelenen literatür arařtırmalarının karşılařtırılması.....	11
3.1. Üç bloktan oluřan bir örnek yakınlık-önem çizelgesi.....	16
3.2. İdeal ve nadir noktalarının hesaplanması.....	25
4.1. En, boy ve savunma yarıçapı verileri için hazırlanmış excel çizelgesi.....	44
4.2. Bloklar arası iliřki (yakınlık-önem deęeri) verileri için hazırlanmış Excel çizelgesi.....	44
4.3. 30 bloktan oluřan veri setindeki blokların en, boy ve savunma yarıçapı deęerleri.....	46
4.4. 30 bloklu veri setindeki blokların birbirleri arasında iliřkiler (yakınlık-önem deęerleri)....	46
4.5. Referans noktaları, ideal ve nadir noktalar.....	47

1. GİRİŞ

Yerleşim problemleri, iş hayatında ve günlük hayatın farklı alanlarında karşılaşılan önemli en iyileme problemlerinden biridir. Yerleşim problemlerinin odak noktası kaynakların veya varlıkların yerleşime uygun bir şekilde atanmasıdır (Özşahin, 2010). Tesis yerleşimi problemleri, bazı lojistik problemleri, kesme ve paketleme problemleri yerleşim problemlerine örnek olarak gösterilebilirler. Bu problemlerden bazıları sınırlı yerleşim alanını göz önüne alırken bazıları da sınırsız yerleşim alanına yerleştirilecek olan varlıkların birbirine göre durumlarını göz önüne alırlar. Örneğin tekstil sektöründe, pastal (kumaş tabakası) kesim planları oluşturulurken kısıtlı alana sahip hammaddenin kullanılmayan kısmı yani fire olacak kısmı en küçüklenmeye çalışılır. Şekil 1.1’deki pastal kesim planında, kalıplar kesim alanı üzerine kumaşın kullanılmayan kısmı en küçük olacak şekilde yerleştirilmeye çalışılmıştır. Tesis yerleşim problemlerinde ise kısıtlı bir alan olan iki boyutlu bir düzleme makineler veya departmanlar malzeme taşıma maliyetlerini ve üretim süresini en küçükleyecek veya bölümler arası etkileşimi en büyükleyecek şekilde yerleştirilmeye çalışılır. Sınırsız alana sahip yerleşim problemlerinde ise eldeki varlıkların sonsuz bir alana amaçları en büyükleyecek şekilde yerleştirilmesi söz konusudur.



Şekil 1.1. Pastal kesim planı (Çelik vd., 2002).

Yerleşim problemleri üretim sistemlerinde maliyetleri düşürme, etkinliği artırma, fireyi azaltma, üretim süresini azaltma ve taşıma mesafesini kısaltma gibi amaçlarla kullanılırken; hizmet sektörlerinde genelde zaman, maliyet ve sinerji avantajı amacıyla kullanılmaktadır. Yerleşim problemleri sadece üretim ve hizmet sistemlerinde değil aynı zamanda yazılım sektöründe de sistem analistlerini sıkça meşgul eden problemlerdendir. Yazılım sektöründe özellikle oyun programlamada yerleşim problemleri önemli bir uygulama alanına sahiptir. Bu uygulama alanlarından biri de strateji oyunlarıdır. Satranç gibi strateji oyunları yerleşim problemi esasına göre programlanabilmektedir. Satrançta, belirli amaçları yerine getirmek için belirli

kısıtlar altında taşların en uygun yerleşimi bir yerleşim problemi örneğidir. Strateji oyunlarının çoğu, genelde oyun unsurlarının oyun alanına uygun bir şekilde yerleştirilmesi esasına dayanır. Bu tip strateji oyunlarına, Clash of Clans ve Boom Beach gibi android tabanlı oyunlar örnek olarak gösterilebilir. Bu oyunlarda yapılar oyun alanına yerleştirilirlerken, yapıların merkezleri arasındaki mesafeleri azaltmak, yapılar arası mesafeye dayalı ilişkileri arttırmak ve en iyi savunmayı sağlamak gibi amaçlar göz önünde bulundurulur. Yani bu oyunlarda duruma ve amaçlara göre yapıların en iyi şekilde konumlandırılması amaçlanır.

Bu tez çalışmasında; mesafe, yapılar arası ilişkiler ve savunma yarıçapı göz önünde bulundurularak bazı strateji oyunlarında ve bazı operasyon planlarında karşılaşılan iki boyutlu dikdörtgen yapıların sınırları olmayan bir alana yerleştirilmesi problemi incelenmiştir. Bu problem özellikle askeri operasyonlar planlanırken askeri unsurların bir alana yerleştirilmesinde ortaya çıkmaktadır. Bu açıdan bu problemin bir Endüstri Mühendisliği problemi olduğu düşünülmektedir. Bu amaçla bu tez çalışmasında önerilen çok amaçlı matematiksel modelin bu tip yerleşim problemlerine çözüm üreteceği düşünülmektedir.

Yerleşim problemlerinde problem boyutu ne kadar artarsa uygun çözümü bulmak da o kadar zorlaştığı için bu tip problemler NP-Hard bir yapıya sahiptir (Yiğit ve Türkbey, 2003). Bu tip problemleri makul zamanda çözmek için son zamanlarda meta-sezgisel teknikler tercih edilmektedir (Özşahin ve Oral, 2010). Bu çalışmada geliştirilen çok amaçlı matematiksel modeli çözmek için bir genetik algoritma önerilmiştir. Genetik algoritmalar NP-Hard yapısındaki optimizasyon problemlerini makul bir sürede çözebilecek meta-sezgisel algoritmalarından biridir. Önerilen çok amaçlı modelin amaçlarını skalerleştirmek için ağırlıklandırılmış toplam skalerleştirme ve Gasimov (2001) tarafından geliştirilen konik skalerleştirme yöntemleri kullanılmıştır. Geliştirilen model, C# dilinde Visual Studio ortamında geliştirilmiş bir yazılımla çeşitli durumlar için simüle edilmiş ve tatmin edici sonuçlar alınmıştır.

Çalışmanın bundan sonraki bölümleri şu şekilde organize edilmiştir: İkinci bölümde literatür araştırması sunulmuştur. Üçüncü bölümde tez konusu ve önerilen genetik algoritma beş başlık altında anlatılmıştır. Dördüncü bölümde önerilen model için geliştirilen bir masaüstü uygulaması ile model test edilip değerlendirilmiştir. Beşinci bölümde ise bu tez çalışması ile ilgili genel bir değerlendirme sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Bu başlıkta bu tez çalışmasında incelenmiş bazı önemli çalışmalara yer verilmiştir. İncelenen çalışmalar üç bölüme ayrılabilir: (1) yerleşim problemleri ile ilgili çalışmalar, (2) çok amaçlı optimizasyon ve skalerleştirme ile ilgili çalışmalar, (3) genetik algoritma ile ilgili çalışmalar.

Bu tez çalışmasının konusu olan yerleşim problemleri ve ortogonal yerleşim problemleri ile ilgili incelenen bazı önemli çalışmalar aşağıda verilmiştir.

Jakobs (1996), çalışmasında çok kenarlı şekilleri dikdörtgen bir alana yerleştirmek için bir genetik algoritma önermiştir. Yazar çalışmasında, sınırsız olan arama uzayının boyutunu kısıtlamak için alt-sol (Bottom-Left, BL) koşulu adında yeni bir yaklaşım da önermiştir. Bu yaklaşıma göre dikdörtgenler en altta sol tarafa daha fazla ilerleyemiyorsa BL koşulunu sağlamış kabul edilir. Yazar algoritmasını çeşitli örneklerle test etmiş ve başarılı sonuçlar almıştır. Ancak bazı durumlarda BL koşulunun iyi çalışmadığını ve daha güçlü algoritmaların geliştirilmesi gerektiğini de belirtmiştir.

Imam ve Mir (1998), çalışmalarında alanları birbirlerinden farklı yapıların sınırları olmayan bir alanda yerleştirilmesi için analitik bir yöntem sunmuşlardır. Her bloğun optimum pozisyonunun, önceden yerleştirilmiş blokların sınırdaki tek boyutlu arama ile belirlendiği bir arama algoritması geliştirilmiştir. Bu çalışmadaki arama süreci dinamiktir, böylece daha fazla iyileştirme sağlanana kadar tüm sınırlar sürekli olarak araştırılır. Elde edilen sonuçlara göre tekniğin optimuma yakın sonuçlar verdiği görülmüştür.

Liu ve Teng (1999), dikdörtgenlerin ortogonal dizilimi için bir geliştirilmiş alt-sol algoritması önermişlerdir. Yazarlar bu çalışmalarında, ortogonal dizilim için genetik algoritmanın uygunluk fonksiyonunda bazı iyileştirmeler yapmışlar ve bu iyileştirmenin etkisini iki örnekle göstermişlerdir. Yazarlar geliştirdikleri algoritmanın Jakobs'un algoritmasından daha etkili olduğu ispatlamışlardır.

Mir ve Al-Saleh (2001), çalışmalarında makro-hücre yerleşim problemini incelemişlerdir. Yazarlar çalışmalarında her harekette bir makro-hücreyi, zaten yerleştirilmiş makro-hücelere göre, optimum pozisyonuna yerleştiren bir yapıcı yerleştirme yöntemi sunmuşlardır. Bu optimum pozisyon, yerleştirilen makro hücrelerin kenarları boyunca tek boyutlu aramalar yaparak belirlenmektedir. Sunulan test sonuçları, elde edilen çözümlerin kalitesinin, benzetilmiş tavlama,

analitik tavlama ve genetik yerleştirme yöntemleri ile elde edilen sonuçlardan daha iyi olduğunu göstermektedir.

Hifi vd. (2003), bu çalışmada sonlu genişlikli sonsuz uzunluklu bir tabaka üzerine hem düzenli hem de düzensiz iki boyutlu şekillerin minimum uzunluklu yerleşiminin bulunması üzerine çalışmışlardır. Bu çalışmadaki problemi çözmek için iki algoritma önerilmiştir. Birincisi özellikle düzensiz şekiller için tasarlanan yeni bir sezgiseldir ama düzenli olanlarda da çalışır. İkincisi genetik algoritma tabanlı melez bir yöntemdir. Sezgisel yöntem ile en iyi yerleştirme sırası, genetik algoritma tabanlı melez yöntem ile de en iyi yerleşim araştırılır. Elde edilen sonuçlar sezgisel yöntemin hızlı bir şekilde çözüm elde ederken melez yöntemin makul bir sürede çözüm geliştirdiği göstermiştir.

Burke vd. (2004), çalışmalarında iki boyutlu dikdörtgen stok kesim problemi için yeni bir best-fit sezgisel önermişler ve elde edilen sonuçları diğer sezgisel algoritmalarla karşılaştırmışlardır. Çalışılan problem ana dikdörtgen üzerine kullanılmayan kısmı en küçükleyecek şekilde küçük dikdörtgenler yerleştirmeyi amaçlamaktadır. Bu çalışmadaki yerleştirme algoritması artan yükseklik veya azalan alan gibi özelliklere göre sıralanmış bir şekil listesi üzerinden çalışmaktadır. Ardından şekillerin her birine sırasıyla bir yerleşim kuralı uygulanır. Önerilen yöntem listeyi dinamik olarak araştırabilir. Önerilen yöntemin literatürdeki diğer sezgisel ve meta-sezgisel yöntemlere göre çözüm kalitesi ve çözüm süresi bakımından daha iyi sonuçlar verdiği gözlemlenmiştir.

Ramin ve arkadaşları (2005), düzensiz şekillerin herhangi bir çakışma olmadan minimum boşluk olacak şekilde yerleştirilmesi için bir popülasyon tabanlı yaklaşım sunmuşlardır. Bu yaklaşımda her bir şekil bir ajan olarak kodlanır ve ajanların yeniden üretilme ve gruplama politikalarıyla nesnelere arasındaki boşluklar azaltılmaya çalışılır. Yaklaşımın kesme problemlerinde uygulanabildiği ve şekil sayısının artmasıyla çözümün bozulmadığı bazı test örnekleriyle gösterilmiştir.

Zhang vd. (2005), çalışmalarında iki boyutlu dikdörtgen paketleme problemi için bir hibrit sezgisel algoritma sunmuşlardır. Bu algoritma "böl ve yönet" tabanlı açgözlü bir algoritmadır. Böl ve yönet tabanlı algoritmalarda problem alt problemlere bölünür. Daha sonra ardışık olarak çözülen alt problemler birleştirilerek orijinal problem çözülür. Dikdörtgen paketleme probleminde böl ve yönet algoritması yerleşim yapılmamış çözüm alanını iki alt alana böler. Alt alanlar dikdörtgenlerle doldurulur. Doldurulan alanlar birleştirilerek paketleme problemi çözülmüş olur. Önerilen yöntem, aynı sınıftaki sınıftaki örnek problemler ile karşılaştırıldığında yarı insan algoritmalarına göre daha iyi performans gösterdiği görülmüştür.

Ahmad vd. (2006), çeşitli yerleşim problemlerinde karşılaşılan NP-Complete zorluk sınıfında olan dikdörtgen paketleme problemini çözmek için yeni ve etkili bir yöntem önermişler ve bazı bilinen algoritmalar ile karşılaştırılmışlardır. Önerilen yerleştirme algoritması çözüm alanını yerleştirilen modüllerin köşeleri ile sınırlayarak olası yerleşim alanını sınırlar. Elde edilen sonuçlar önerilen yöntemin hızlı ve farklı durumlara karşı dirençli olduğunu göstermiştir.

Korf vd. (2008), çalışmalarında NP-Tam karmaşıklık sınıfında olan bir dikdörtgenler kümesinin minimum alanı kaplayacak şekilde dikdörtgen bir alana yerleştirilmesi problemini ele almışlar ve iki farklı probleme iki farklı yaklaşım ile çözüm aramışlardır. Birinci problem kaplama problemi denilen bir dizi dikdörtgenin bir dikdörtgen kutu içine yerleştirilmesi problemi, ikincisi ise minimal sınırlayıcı kutu problemi denilen bir dizi dikdörtgenin üst üste gelmeden minimum alan kaplayacak şekilde yerleştirilmesi problemidir. Her iki çözüm yaklaşımı da kısıt sağlama problemi (constraint problem satisfaction-CSP) gibi formüle edilmiştir. Mutlak yerleştirme yaklaşımında her bir dikdörtgenin sınırlayıcı kutudaki konumunu gösteren bir değişken vardır. Göreceli yerleştirme yaklaşımında ise her iki dikdörtgen çifti için göreceli konumlarını gösteren, soldan, sağdan, üstünden veya altından biri olabilen bir değişken vardır. Küçük boyutlu problemlerde mutlak yerleştirme yaklaşımı göreceli yerleştirme yaklaşımına göre daha iyi sonuçlar vermektedir. Bununla birlikte, problem boyutu arttığında sınırlayıcı kutudaki mutlak yerleşimlerin sayısı arttıkça mutlak yerleştirme yaklaşımının çözüm uzayı da büyür ancak bu durumun göreceli yerleştirme yaklaşımı üzerinde bir etkisi yoktur. Bu çalışmaya göre büyük boyutlu problemlerde göreceli yaklaşım mutlak yaklaşımdan daha iyi sonuçlar verecektir.

McKendall ve Hakobyan (2010), çalışmalarında tesisler arası malzeme taşıma maliyetlerini azaltmak için dinamik tesis yerleştirme problemi üzerine çalışmışlardır. Boyutları birbirinden farklı ve döndürülebilen tesisler sınırları olmayan bir düzlem üzerine yerleştirilmeye çalışılır. Problem karmaşıklığı nedeniyle kesin yöntemlerle sadece küçük boyutlu problemler çözülebilmektedir. Bu çalışmada dinamik tesis yerleşim problemi için yerleştirilecek ona tesisleri yerleştirilmiş olan tesislerin sınırları boyunca yerleştiren bir sınır arama tekniği geliştirilmiştir. Çözüm tabu arama algoritması yöntemi kullanılarak geliştirilmiştir. Geliştirilen teknik literatürdeki bazı statik ve dinamik tesis yerleşim örnekleriyle test edilmiş ve etkin sonuçlar elde edildiği görülmüştür.

Özşahin ve Oral (2010), iki boyutlu dörtgensel şekillerin, iki boyutlu dörtgensel ve çember şeklindeki düzlemlere yerleştirilmesinde en büyük alan kullanımı ile yerleşimin yapılması ve şekillerin önem derecesine göre en büyük oranda yerleştirilmesi amacıyla bir genetik algoritma tasarlamışlardır. Çalışmada özel bir yerleştirme algoritması ile şekillerin genetik algoritmada

kodlanan yerleşim sırasına göre düzlem üzerindeki yerleri saptanmıştır. Önerilen yerleştirme algoritmasına göre sıradaki parça veya şekil seçilmektedir. Seçilen parçanın döndürülmeye uğrayıp uğramayacağı belirlendikten sonra en sol üst köşeden yani sıfır noktasında yerleşim işlemi başlatılır. Parça yukarıdan aşağıya doğru birer kare kaydırılır. En alt noktaya gelindiğinde yerleşim yapılmamışsa bu sırada yine sıfır noktasına dönülür ancak yatayda bulunulan noktadan bir birim sağa hareket edilir. Yerleşim gerçekleşinceye kadar bu süreç sürdürülür, eğer parça yerleşmez ise bir sonraki parçaya geçilir ve bu süreç ya bütün yerleşim sırası uygulanana kadar ya da yerleşecek alan kalmayınca kadar sürdürülür. Elde edilen sonuçlar iki boyutlu şekil yerleştirme problemlerinde genetik algoritmanın iyi sonuçlar verdiğinin göstermektedir.

Hougardy (2012), çalışmasında mükemmel dikdörtgen paketleme problemi için bir kesin çözüm yöntemi önermiştir. Mükemmel dikdörtgen paketleme problemi dikdörtgen paketleme probleminin özel bir halidir. Dikdörtgen paketleme probleminde "n" adet dikdörtgen, bir dikdörtgen çözüm alanına yerleştirilmeye veya dizilmeye çalışılır. Mükemmel paketleme probleminde yerleştirilecek dikdörtgenlerin toplam alanı dikdörtgen çözüm alanının alanına eşittir. Çalışmada önerilen algoritma bir dal sınır algoritmasıdır ve çözüm süresi sadece dikdörtgenlerin sayısına göre değil boyutlarına göre de değişmektedir. Elde edilen sonuçlara göre algoritmanın oldukça iyi çözümler verdiği görülmüştür.

Özcan vd. (2013), dikdörtgen şekillerin sabit genişlikli sonsuz uzunluklu stok üzerine yerleştirilmesini ele alan ortogonal paketleme problemi üzerine çalışmışlardır. Çalışma hurdayı azaltmayı ve malzeme kullanımını arttırmayı amaçlamaktadır. Bu problemi çözmek için iki yönlü best-fit sezgisel algoritmasını (Bidirectional Best-Fit Heuristic-BBF) geliştirerek bir yaklaşım önermişlerdir. Standart BBF algoritması stok şeridi üzerine yerleştirilecek olan dikdörtgenleri tek tek ele almıştır. Önerilen algorithmada yerleştirilecek olan dikdörtgenler çiftler halinde değerlendirilmiştir. Geliştirilen yöntemin daha iyi sonuç verdiği gözlemlenmesine rağmen dikdörtgenlerin çiftler halinde değerlendirilmesinden dolayı çözüm süresinde artış olduğu görülmüştür.

Gonçalves ve Resende (2015), çalışmalarında alanları birbirinden farklı dikdörtgen tesisin dikdörtgen olmayan bir taban alanı üzerinde örtüşmeden yerleştirilmesi için bir genetik algoritma sunmuşlardır. Çalışmanın amacı tesislerin merkezleri arasındaki mesafelerin toplamını en küçükleyecek tesis konumunu ve büyüklüğünü tespit etmektir. Çalışmada, her tesisin yerleştirme sırasını ve boyutlarını belirleyen genetik algoritma ile birleştirilmiş yeni bir yerleştirme stratejisi geliştirilmiştir. Yaklaşımın kalitesi, kapsamlı olarak incelenen 28 veri setinin 19'u için en iyi bilinen çözümlerin geliştirilmesiyle doğrulanmıştır.

Tasadduq vd. (2015), çalışmalarında sabit en boy oranına sahip tesislerin yerleştirilmesi problemine odaklanmışlardır. Problemi çözmek için sınır arama algoritması (Imam ve Mir, 1998) ve dereceli iniş algoritmasının bir kombinasyonunu içeren bir yöntem geliştirmişlerdir. Geliştirilen yöntem iki aşamadan oluşmaktadır. İlk aşamada algoritma daha önce oluşturulmuş bir tesis kümesinin sınırında en uygun konuma yeni bir tesis yerleştirir. İyileştirme aşamasında, algoritma, sınır arama ve dereceli iniş algoritmalarıyla bir yerel optimuma yaklaşıncaya kadar arama yapar. İyi bilinen test problemleri ile yapılan deneyler, önerilen algoritmanın hem yayınlanan sonuçlardan hem de bir yerleşim planlaması ve optimizasyon yazılımı olan VIP-PLANOPT tarafından üretilen çözümlerden daha üstün çözümler ürettiğini göstermektedir.

Al-Saleh vd. (2015), çalışmalarında alanları birbirinden farklı olan tesislerin yerleştirilmesi problemini çözmek için geliştirilmiş sıralama kriteri ile bir geliştirilmiş yapıcı teknik sunmuşlardır. Sunulan teknik, sınır arama algoritmasının (Imam ve Mir, 1998) tesisler arası malzeme taşıma maliyetlerine göre sıralama yapan bir sıralama kriteri ile bütünleştirilmesiyle çalışır. Önerilen tekniğin performansı tabu arama (TS) ve parçacık sürü optimizasyon (PSO) algoritmaları ile kıyaslanmış ve elde edilen sonuçlar önerilen tekniğin TS ve PSO algoritmalarını geride bıraktığı görülmüştür. Ayrıca önerilen tekniğin, büyük boyutlu yerleşim problemleri için TS ve PSO algoritmalarına entegre ederek oluşturulmuş karma tekniklerden daha iyi performans gösterdiği tespit edilmiştir.

Sikaroudi ve Shahanaghi (2016), bir çözüm alanına alanları birbirlerinden farklı tesislerin yerleştirilmesi problemini çözmek için yeni bir yaklaşım sunmuşlardır. Tesisler bir ilişki matrisine dayanarak birbirine kuvvet uygulayan rastgele oluşturulmuş noktalar olarak düşünülür. Her nokta tesisin merkezindedir ve örtüşmeleri önlemek için iki sezgisel yöntem kullanılmıştır. Önerilen algoritma iki aşamadan oluşur. İlk aşamada başlangıç çözümü oluşturulur ve ikinci aşamada ilk aşamanın çıktısı hesaplanan kuvvetlere göre kaydırılarak geliştirilir. Algoritmanın çok kısa sürelerde uygun çözümler elde ettiği test verileriyle gösterilmiştir.

Delorme vd. (2017), çalışmalarında bölünemeyen ama 90 derece döndürülebilen dikdörtgen nesnelere kümesinin sabit genişlikli bir rulonun üzerine örtüşme (çakışma) olmadan en kısa uzunlukta yerleştirilmesi üzerine çalışmışlardır. Yazarlar problemi tamsayılı doğrusal programlama ile formüle etmişlerdir. Problem bir ayrışım algoritmasıyla çözülmüştür. Ayrışım algoritması problemi ana problem ve köle problem olmak üzere iki alt probleme ayırır. Elde edilen deneysel sonuçlar literatürdeki çalışmalar ile kıyaslandığında önerilen yöntemin oldukça etkin olduğu görülmüştür.

Paes vd. (2017), çalışmalarında alanları birbirinden farklı olan tesislerin yerleştirilmesi problemiyle ilgilenmişlerdir. Bu çalışmadaki problem, tesisleri sınırsız düzleme örtüşme olmadan ve tesisler arası malzeme taşıma akışını en küçükleyecek şekilde yerleştirmeyi amaçlamaktadır. Problemi çözmek için iki yaklaşım incelenmiştir: Temel genetik algoritma ve genetik algoritma temelli çözümlerin birleştirilmesi ve ayrılmasına dayalı açgözlü bir ayrıştırma stratejisi. Çalışmada, problemi verimli bir şekilde ayrıştırmak için yerleştirilen tesislerin X veya Y eksenleri üzerinde olmaması sağlanır. Bu kısıtlama, elde edilebilecek en iyi çözümün değerini bozabilirse de, orta ve büyük boyutlu örneklerde yöntemin arama yeteneklerini de büyük ölçüde artırmaktadır. İncelenen her iki yöntem de literatürdeki örneklerle test edilmiş ve yüksek kalitede çözümler elde edilmiştir.

Shiangjen vd. (2018), çalışmalarında iki boyutlu döndürülemeyen dikdörtgen nesnelere içeren kesme problemi üzerine çalışmışlardır. Bu çalışmadaki dikdörtgen nesnelere yerleştirileceği alanın boyutları sınırlı ve sabittir. Çalışmada nesnelere döndürülmeden ve örtüşme olmadan alan kullanımını en büyükleyecek şekilde yerleştirilmeye çalışılmıştır. Bu problemi çözmek için iki yönlü sezgisel bir yerleştirme algoritması önerilmiştir. Sezgisel algoritmada Lourenço vd. (2010) tarafından geliştirilen yinelenen yerel arama algoritmasının çözüme yön verme ve çözüm uzayını arama özellikleri geliştirilmiştir. Elde edilen sonuçlara göre, sunulan yaklaşımın uygun bir başlangıç çözüm ile büyük ölçekli problemler için makul bir hesaplama süresi içinde yüksek kaliteli çözümler elde ettiği görülmüştür.

Bu çalışmada incelenen çok amaçlı optimizasyon ve genetik algoritmalar ile ilgili çalışmalar aşağıdaki gibidir:

Deb ve arkadaşları (2002), çalışmalarında çok amaçlı evrimsel algoritmaların hesaplama karmaşıklığı, elitizm ve parametre paylaşım özellikleri için NSGA-II (Nondominated Sorting Genetic Algorithm II) temelli çok amaçlı bir evrimsel algoritma önermişlerdir. Ayrıca ebeveyn ve yavru kromozomları birleştirerek en iyi çözümleri oluşturan bir eşleme havuzu içeren seçin operatörü önermişlerdir. Test sonuçları NSGA-II temelli algoritmanın pareto cephesine hızlı yakınsadığını göstermiştir.

Gasimov (2004), çalışmasında, farklı enlere sahip stok malzemelerinin seçimini ve seçilen malzemelerin hangi kesme planlarına göre kesileceğinin belirlenmesini içeren 1.5 boyutlu stok malzemesi seçimi problemini incelemiştir. Bu problemin çözümü için, verilen talepleri karşılayacak ve birbiri ile çelişen fire ve depolama maliyetlerinin aynı anda en küçüklendiği doğrusal ve doğrusal olmayan karma tamsayı çok ölçütlü iki model önerilmiştir. Bu çalışmada

Gasimov tarafından önerilmiş olan skalerleştirme yöntemi kullanılmış ve özel olarak türetilmiş olan bir örnek problem için önerilen yaklaşımla pareto en iyi çözümler bulunmuştur.

Sipahioğlu vd. (2010), çalışmalarında konik skalerleştirme yönteminin çok amaçlı sırt çantası probleminde başarıyla kullanılabileceğini, klasik ağırlıklandırma yöntemiyle elde edilmesi mümkün olmayan içbükey pareto etkin çözümlerin bu yöntemle rahatlıkla bulunabileceğini, yazından alınan büyük boyutlu test problemleri üzerinde göstermişlerdir.

Özyön vd. (2011), çalışmalarında çok amaçlı optimizasyon problemlerinden biri olan çevresel ekonomik güç dağıtım problemini incelemişler ve problemi tek amaçlı optimizasyon problemine dönüştürmek için ağırlıklı toplam metodunu, problemin çözümü için de genetik algoritma metodunu kullanılmıştır. Örnek bir sistemin 1 günlük işletim süresi göz önüne alınarak, gaz kısıtsız ve gaz kısıtlı olarak, ağırlık faktörlerinin çeşitli değerleri için çözümler elde edilmiştir. Ağırlık faktörü $w = 1.0$ 'dan başlayarak 0.1 'er aralıklarla azaltılarak 0.0'a kadar hem gaz kısıtlı hem de gaz kısıtsız olmak üzere 11'er kez çalıştırılmıştır. Gaz kısıtlı çözümlerde gaz kısıtı tolerans değerleri içinde kalırken w 'nin azalan değerleri için toplam NOx emisyon miktarının azalmasına karşılık toplam yakıt maliyetinin arttığı görülmüştür.

Çolak (2013), çalışmasında meta sezgisel bir yöntem olan genetik algoritmalar yardımı ile gezgin satıcı problemine çözüm aranmış ve geliştirilen algoritmanın uygulaması Adana ilinde gıda sektöründe faaliyet gösteren bir firma üzerinde gerçekleştirilmiştir. Genetik algoritmanın firmanın iki bölgesinin yeni rota belirlenmesi aşamasında kullanılması ile elde edilen sonuçlar bu algoritmanın etkinliğini göstermiştir. Bu sonuçlar incelendiğinde toplam kat edilen yolun her bir rotada firmanın güncel rotalarına oranla ortalama %20 civarı kısaldığı görülmüştür.

Acar ve Durucasu (2014), çalışmalarında yap işlet devret (YİD) projelerinde proje şirketinin ihaleyi kazanma potansiyeli ve borç bulma potansiyeli amaçlarını içeren finansal optimizasyon modeli sunmuşlardır. Çok amaçlı evrimsel algoritma ile çözümlenen model, pareto çözümler kümesi sunarak, iki çelişen amaç arasındaki değişimi değerlendirmede karar vericilere yardımcı olmaktadır. Modelin uygulaması için Türkiye'deki bir havaalanı YİD projesi verileri kullanılmıştır. Çalışmanın ana sonucu olarak; geliştirilen optimizasyon modeli, ihale hedeflerine en etkili ve verimli bir şekilde ulaşılmasında özel sektör şirketlerinin kullanabileceği yararlı araçlar olarak ortaya konmaktadır.

Ergül (2015), çalışmasında, çok amaçlı genetik algoritma (ÇAGA) yöntemlerinin sıralama yeteneklerinin ölçülmesi için ceza ve ödül ölçütleri önermiştir. Bu iki ölçüt ile bir ÇAGA yöntemi tarafından seçme mekanizmasına ne kadar nitelikli bilgi aktarıldığı sezgisel ve

istatistiksel olarak tespit edilebilmektedir. Bu çalışmada literatürde çok kullanılan SPEA (Pareto Gücü Evrimsel Algoritma, Strength Pareto Evolutionary Algorithm) yöntemi ile yeni önerilmiş DOPGA (Bireyin Baskınlık Gücü Genetik Algoritma, Domination Power of an Individual Genetic Algorithm) yöntemi, dört farklı test fonksiyonu üzerinde çalıştırılmış ve sonuçlar ceza ve ödül ölçütleri kullanılarak değerlendirilmiştir. Sonuç olarak, DOPGA yönteminin SPEA yöntemine göre bireyleri en çok istenilenden en az istenilene doğru daha iyi sıraladığı belirlenmiştir.

Kaya (2016), çalışmasında çok amaçlı optimizasyon problemlerinde karar vericinin sonuçlar içinden tercihini yapabilmesini sağlayan pareto baskınlık kavramını sunmuştur. Örnek problem üzerinde uygulama yaparak bu yöntemin üstün yanları ortaya konulmuştur ve araştırmacılar için çok amaçlı optimizasyon problemlerinin çözümünde pareto optimal kullanmaları önerilmiştir.

Lokman (2017), çalışmasında çok amaçlı tamsayı programlama problemlerinde baskın noktaların sayısı problemin büyüklüğüne bağlı olarak üssel bir büyüme gösterdiğine ve bu nedenle bu problemler için tüm baskın noktaları bulmanın zor ve karar verici için pratik bir yaklaşım olmadığına değinmiştir. Yazar çalışmasında, temsili kümenin değerlendirilmesinde kullanılan kalite ölçülerini ve bu kalite ölçülerine göre tüm baskın nokta kümesini iyi temsil eden noktaları bulan yaklaşımları incelemiştir.

Yiğit ve Aydemir (2018), çalışmalarında konteyner yükleme problemini ele almışlardır. Çalışmada, daha düşük maliyet ile konteyner yükleme probleminin çözümü için sezgisel algoritmalarından biri olan genetik algoritma kullanılmıştır. Genetik algoritma ile konteyner içerisine çeşitli boyutlardaki kutular en uygun şekilde yerleştirilmiştir fakat yerleştirme yaparken kutuların döndürülebilme özellikleri kullanılmamıştır. Böylece, konteyner içi boş alanın en küçüklenmesi ve yüklenen kutu sayısının en küçüklenmesi amaçlanmıştır. Çalışma, literatürdeki diğer sezgisel algoritmalarla karşılaştırılmış ve elde edilen sonuçlar sunulmuştur. Ayrıca problemin çözümünde ağırlıklandırılmış toplam yöntemi ve konik skalerleştirme yöntemi kullanılmıştır. Çalışma literatürdeki diğer sezgisel algoritmalarla karşılaştırılmış ve elde edilen sonuçlar sunulmuştur. Literatürdeki diğer genetik algoritmalar ve diğer sezgisel algoritmalarla karşılaştırıldığında önerilen yöntemin sonuçlarının diğer algoritmalarından daha iyi sonuçlar verdiği gözlemlenmiştir.

Yaşar ve Özyön (2018), çalışmalarında çok amaçlı çevresel ekonomik güç dağıtım problemini konik skalerleştirme metodu (KSM) ile tek amaçlı optimizasyon problemine dönüştürmüşlerdir. Çalışmada skalerleştirilen problemin çözümü için genetik algoritma (GA) metodu kullanılmıştır. Uygulama için ele alınan örnekler, konveks ve konveks olmayan parçalı

yakıt maliyeti fonksiyonlarına sahip üretim birimlerinden oluşan kayıplı güç sistemleridir. Örnek problemlerde farklı ağırlık değerleri için toplam yakıt maliyeti ve toplam NOx emisyon değerlerine ait en iyi çözüm değerleri elde edilmiştir ve sonuçlar tartışılmıştır.

Çizelge 2.1. İncelenen literatür araştırmalarının karşılaştırılması.

Yazar(lar)	Yayın Yılı	Konu	Metot(lar)
1. Jakobs	1996	Çokgen şekilleri dikdörtgen alana yerleştirmek	Genetik algoritma
2. Imam ve Mir	1998	Alanları birbirlerinden farklı yapıların sınırları olmayan bir alanda yerleştirilmesi	Analitik bir arama algoritması
3. Liu ve Teng	1999	Dikdörtgenlerin ortogonal dizilimini sağlamak	Genetik algoritma
4. Mir ve Al-Saleh	2001	Makro-hücre yerleşim problemi	Bir yapıcı yerleştirme yöntemi
5. Hifi vd.	2003	Sonlu genişlikli sonsuz uzunluklu bir tabaka üzerine hem düzenli hem de düzensiz iki boyutlu şekillerin minimum uzunluklu yerleşiminin bulunması	- Sezgisel yöntem - Genetik algoritma tabanlı melez bir yöntem
6. Burke vd.	2004	İki boyutlu dikdörtgen stok kesim problemi	Best-fit sezgisel algoritması
7. Ramin vd.	2005	Düzensiz şekillerin herhangi bir çakışma olmadan minimum boşluk olacak şekilde yerleştirilmesi	Popülasyon tabanlı bir yaklaşım
8. Zhang vd.	2005	İki boyutlu dikdörtgen paketleme problemi	Böl ve yönet tabanlı ağgözlü hibrit sezgisel bir algoritma
9. Ahmad vd.	2006	Dikdörtgen paketleme problemi	Bottom-left (BL), Improved-BL, Bottom-Left Fill, Minimization of Enclosing Rectangle Area (MERA) algorithm ve genetik algoritma tabanlı optimizasyon yöntemi
10. Korf vd.	2008	Bir dikdörtgenler kümesinin minimum alanı kaplayacak şekilde dikdörtgen bir alana yerleştirilmesi	Kısıt esnetme problemi (constraint-satisfaction problem) olarak formüle edilmiş iki farklı yaklaşım
11. Özşahin ve Oral	2010	İki boyutlu dörtgen şekillerin, iki boyutlu dörtgen ve çember şeklindeki düzlemlere en büyük oranda yerleştirilmesi	Genetik algoritma

Çizelge 2.1. İncelenen literatür araştırmalarının karşılaştırılması (devam).

12. McKendall ve Hakobyan	2010	Tesisler arası malzeme taşıma maliyetlerini azaltmak için dinamik tesis yerleştirme problemi	Bir sınır arama tekniği
13. Hougardy	2012	Mükemmel dikdörtgen paketleme problemi	Çalışma zamanı sadece dikdörtgenlerin sayısına göre değil aynı zamanda boyutlarına göre de değişen dal sınır algoritması tabanlı bir kesin çözüm yöntemi
14. Özcan vd.	2013	Dikdörtgen şekillerin sabit genişlikli sonsuz uzunluklu stok üzerine yerleştirilmesi	İki yönlü best-fit sezgisel algoritması
15. Gonçalves ve Resende	2015	Alanları birbirinden farklı dikdörtgen tesisin dikdörtgen olmayan bir taban alanı üzerinde örtüşmeden yerleştirilmesi	Genetik algoritma
16. Tasadduq vd.	2015	Sabit en boy oranına sahip tesislerin yerleştirilmesi	- Sınır arama algoritması - Dereceli iniş algoritmasının bir kombinasyonunu içeren bir yöntem
17. Al-Saleh vd.	2015	Alanları birbirinden farklı olan tesislerin yerleştirilmesi problemi	Sınır arama algoritmasının (Imam ve Mir, 1998) tesisler arası malzeme taşıma maliyetlerine göre sıralama yapan bir sıralama kriteri ile bütünleştirilmiş hali
18. Sikaroudi ve Shahanaghi	2016	Bir çözüm alanına alanları birbirlerinden farklı tesislerin yerleştirilmesi	İki aşamalı bir sezgisel algoritma
19. Delorme vd.	2017	90 derece döndürülebilir dikdörtgen nesnelere kümesinin sabit genişlikli bir rulonun üzerine örtüşme (çakışma) olmadan en kısa uzunlukta yerleştirilmesi	- Problem tamsayılı doğrusal programlama modeli - Bir ayrışım algoritması
20. Paes vd.	2017	Alanları birbirinden farklı olan tesislerin yerleştirilmesi	- Genetik algoritma - Açgözlü bir ayrıştırma stratejisi
21. Shiangjen vd.	2018	İki boyutlu döndürülemeyen dikdörtgen nesnelere içeren kesme problemi	İki yönlü sezgisel bir yerleştirme algoritması

Bu tez çalışmasında iki boyutlu ortogonal yerleşim problemini çözmek için Imam ve Mir (1998) tarafından geliştirilen çözüm alanını büyük ölçüde küçülten bir sınır arama algoritması kullanılmıştır. Imam ve Mir (1998) tarafından geliştirilen algoritma, sıralanmış blokları daha önce yerleştirilmiş olan blokların sınırında en iyi pozisyona yerleştirir. Bununla birlikte Imam ve Mir

(1998) tarafından geliştirilen algoritma genetik algoritma ile birlikte kullanılarak çok amaçlı problemi çözmek için kullanılmıştır.

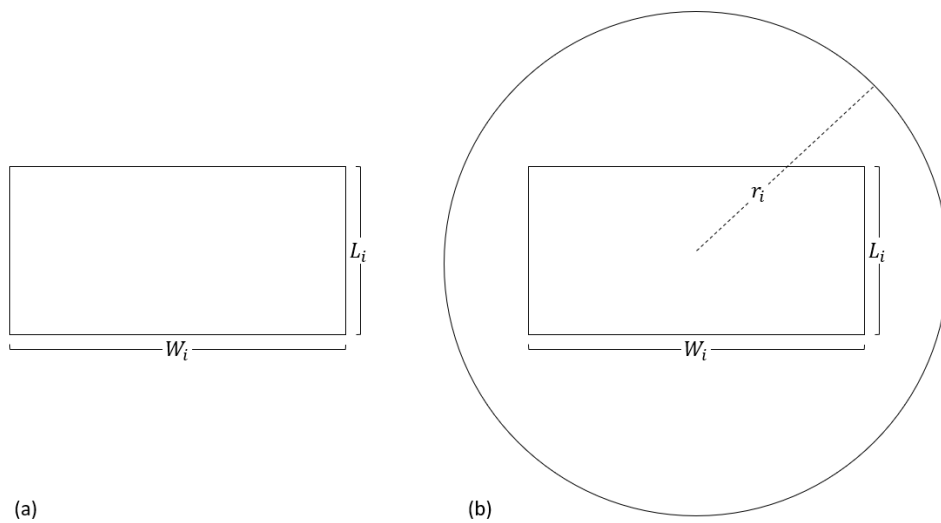
Bu çalışmadaki çok amaçlı olan problemi çözebilmek için amaç fonksiyonları ağırlıklı toplam ve konik skalerleştirme (Gasimov, 2001) yöntemleri ile skaler hale getirilmiştir. Bu çalışmada sınır arama algoritmasında blokların sırasını belirlemek için bir genetik algoritmadan yararlanılmıştır. Geliştirilen genetik algoritmada blokların sırasını ve yönünü belirlemek için kromozom yapısı iki kısım oluşacak şekilde tasarlanmıştır. İki kısımdan oluşan kromozom yapısı sayesinde blokların sırası ve yönü birbirlerinden bağımsız olarak mutasyona uğrayabilmektedir. Bulunan en iyi kromozomun sonraki nesillere değişmeden aktarılmasını sağlamak için elitizm operatörü genetik algoritmaya dahil edilmiştir.

3. İKİ BOYUTLU ORTOGONAL YERLEŞİM PROBLEMİ

Bu başlıkta dikdörtgen blokların yerleşimi için iki boyutlu ortogonal yerleşim problemi (OYP) incelenmiştir. OYP’de iki boyutlu dikdörtgen bloklar çözüm alanına belirli amaçlar doğrultusunda yerleştirilmeye çalışılır. OYP’de yerleştirilecek olan blokların tamamı çözüm alanında bloklar arasında örtüşme olmayacak şekilde istenilen yere yerleştirilebilir. Yine OYP’de W enine ve L boyuna sahip bloklar çözüm alanına ortogonal yani dik açılı olacak şekilde yerleştirilirler ve 90 derece döndürülebilirler. Bu yerleştirmede her bloğun yatay kenarları X -eksenine, dikey kenarları Y -eksenine paralel olmalıdır (Baker vd., 1980).

3.1. Problem

Bu çalışmada, iki boyutlu dikdörtgen blokları belirli amaçlar doğrultusunda bir çözüm alanına yerleştirecek bir probleme odaklanılmıştır. Çalışmadaki bloklar W enine, L boyuna ve r savunma yarıçapına sahiptir. Şekil 3.1a’da savunma yarıçapına sahip olmayan ve Şekil 3.1b’de savunma yarıçapına sahip olan bir örnek blok gösterilmiştir. Çalışmada; en, boy ve savunma yarıçapı parametreleri tam sayı olarak tanımlanmıştır. Çözüm alanı, blokların yerleştirileceği 1×1 birimlik karelere bölünmüş alandır. OYP problemlerinde olduğu gibi bu tez çalışmasında çalışılan problemde de blokların tamamı çözüm alanında bloklar arasında örtüşme olmayacak şekilde istenilen yere yerleştirilebilir. Bloklar çözüm alanına ortogonal yani dik açılı olacak şekilde yerleştirilirler. Bloklar çözüm alanına yerleştirilirken 90 derece döndürülebilirler ancak blokların boyutları değiştirilemez.



Şekil 3.1. Dikdörtgen bloklar.

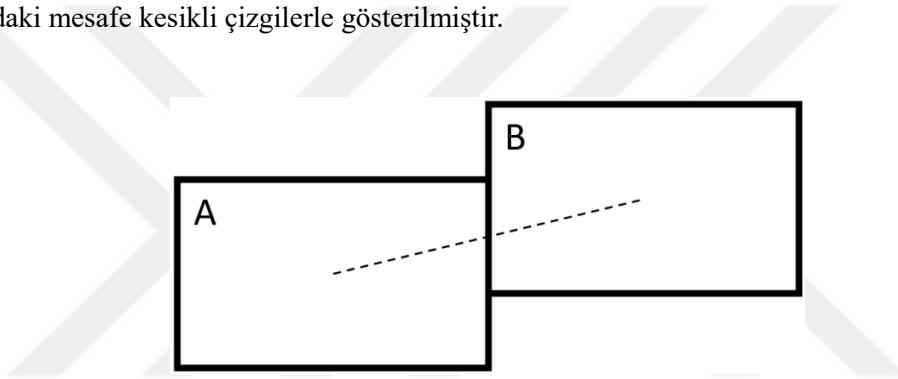
Bu çalışmada incelenen problemin üç temel amacı vardır;

Amaç 1: Blokların merkezleri arasındaki mesafeleri toplamının en küçüklenmesi,

Amaç 2: Birbiriyle ilişkili blokların birbirlerine yakın olması,

Amaç 3: Blokların savunma yarıçapları dışında kalan alanlarının toplamının en küçüklenmesi.

Amaç 1: Blokların merkezleri arasındaki mesafelerin toplamının en küçüklenmesi amacı blokların çözüm alanında dağılmasını engelleyerek birbirlerine yakın bulunmalarını ve çözüm alanında daha az yer kaplamalarını sağlayacaktır. Şekil 3.2’de iki dikdörtgen bloğun merkezleri arasındaki mesafe kesikli çizgilerle gösterilmiştir.



Şekil 3.2. İki dikdörtgen bloğun merkezleri arasındaki mesafe.

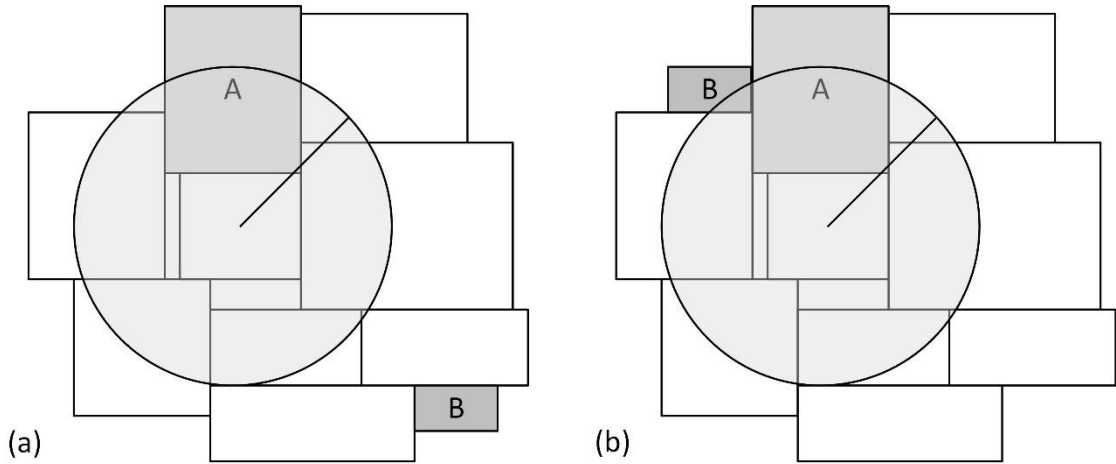
Blokların merkezleri arasındaki mesafelerinin toplamının en küçüklenmesi için aşağıda sunulan (1) numaralı amaç fonksiyonu kullanılmaktadır;

$$\text{Min } F_1 = \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{ij} \quad (1)$$

Yukarıdaki amaç fonksiyonunda N blok sayısını, d_{ij} de i’inci ve j’inci blokların merkezleri arasındaki mesafeyi göstermektedir.

Amaç 2: Birbiriyle ilişkili blokların birbirlerine yakın olması amacı yerleşim problemlerinde sıklıkla karşılaşılan amaçlarından bir tanesidir. Örneğin tesis yerleşim problemlerinde tesisler arası malzeme taşıma maliyetlerinin düşürülmesi için birbirleri arasında malzeme akışı fazla olan tesisler birbirlerine yakın yerleştirilmeye çalışılır. Amaç 1 sadece blokların merkezleri arasındaki mesafeleri dikkate alırken Amaç 2 blokların birbirleriyle olan ilişki değerlerini de dikkate alır. İki amaç arasındaki fark Şekil 3.3’te gösterilmiştir. Şekil 3.3a’da sadece Amaç 1 göz önünde bulundurulmuş, Şekil 3.3b’da ise Amaç 2 de göz önünde

bulundurularak birbiriyle ilişkili olan A ve B blokları birbirine yakın olacak şekilde yerleşim yapılmıştır.



Şekil 3.3. Amaç 1 ile Amaç 2'nin karşılaştırılması.

Bloklar arasındaki ilişkiler değerlendirilirken önceden karar verici tarafından oluşturulmuş bir *yakınlık-önem* matrisinden yararlanır. Üç bloktan oluşan bir örnek *yakınlık-önem* Çizelgesi Çizelge 3.1'de gösterilmiştir. Örneğin *i*'inci ve *j*'inci bloklar arasındaki ilişki değeri r_{ij} olmak üzere Çizelge 3.2'de 1 ve 2 numaralı bloklar arasındaki *yakınlık-önem* değeri $r_{ij} = 10$ olarak gösterilmektedir. Bu durumda 1 numaralı bloğun 2 numaralı bloğa yakın olması 10 kat daha önemlidir.

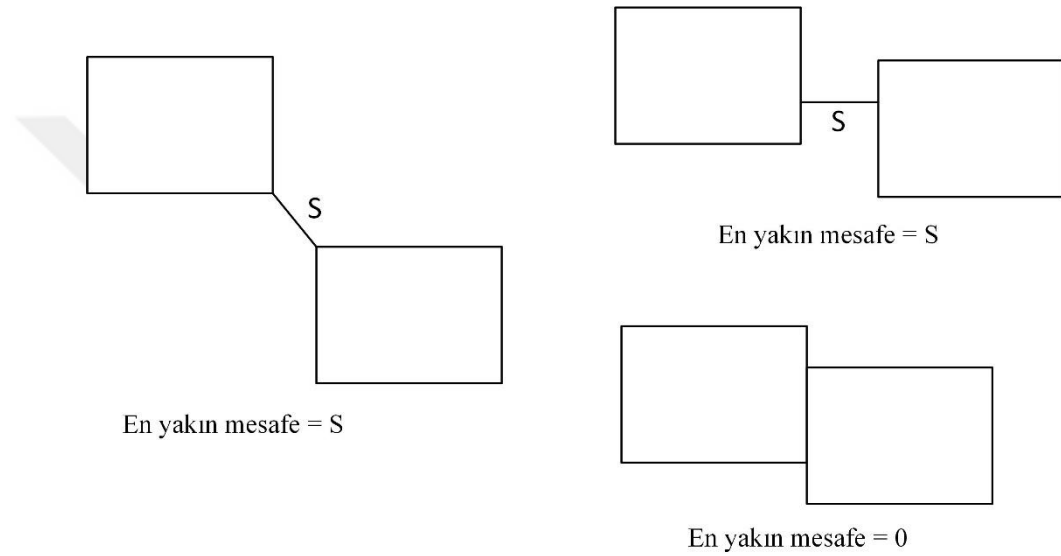
Çizelge 3.1. Üç bloktan oluşan bir örnek *yakınlık-önem* çizelgesi.

r_{ij}	$j = 1$	$j = 2$	$j = 3$
$i = 1$	0	10	5
$i = 2$	10	0	20
$i = 3$	5	20	0

Birbiriyle ilişkili blokların birbirlerine yakın bulunması amacı (2) numaralı fonksiyon ile gösterilebilir:

$$\text{Min } F_2 = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{r_{ij}}{1 + S_{ij}}} \quad (2)$$

Yukarıdaki amaç fonksiyonunda N blok sayısını, r_{ij} yakınlık-önem matrisine göre i'inci ve j'inci bloklar arasındaki ilişkiyi, S_{ij} de i'inci ve j'inci blokların birbirine en yakın noktaları arasındaki mesafeyi göstermektedir. İki blok arasındaki yakınlık-önem değerinin mesafeye oranı bu amaç fonksiyonunun değerini verecektir. İki blok arasındaki mesafe sıfır olduğu durumlarda tanımsız değer oluşmasını engellemek için fonksiyonun paydasına "1" eklenerek tanımsız olma durumu ortadan kaldırılmıştır. Şekil 3.4'te iki blok arasındaki en yakın mesafeler gösterilmiştir.

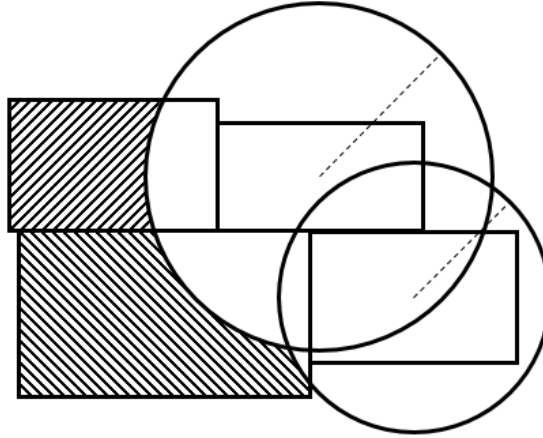


Şekil 3.4. İki dikdörtgen blok arasındaki en yakın mesafeler.

Amaç 3: Bu amaç çoğu askeri operasyon planlarında ve strateji oyunlarında karşılaşılan yerleşim problemi amaçlarından biridir. Strateji oyunlarında ve askeri operasyon planlarında savunma yapabilen yapıların savunma yarıçapı adında bir parametresi vardır. Bu çap, yapının merkezinden itibaren ne kadarlık bir alanda savunma yapabileceğini gösterir. Savunma yarıçapı dışında kalan alan savunmasız alandır ve tüm alanların savunma yarıçapı içinde kalması ve dolayısıyla blokların savunma yarıçapları dışında kalan alanlarının toplamının en küçüklenmesi amaçlanır. Bu amaç fonksiyonu (3) numaralı fonksiyon ile gösterilebilir:

$$\text{Min } F_3 = \sum_{i=1}^N U_i \quad (3)$$

Burada N blok sayısını, U_i i'inci dikdörtgen bloğun savunma yarıçapları dışında kalan alanını göstermektedir. Şekil 3.5'te blokların savunma yarıçapları dışında kalan alanları taralı olarak gösterilmiştir.



Şekil 3.5. Savunma yarıçapları dışında kalan alan.

3.2. Çözüm Metodu

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle günümüzde, sert (hard) optimizasyon teknikleri yerine yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlardan olan genetik algoritmalar bu arayışlar içinde önemli bir yer tutmaktadır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan genetik algoritmalar diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak karma (hibrit) çözümler de geliştirilmeye çalışılmaktadır (Emel, 2002).

Yerleşim problemleri, problem boyutu ne kadar artarsa optimal sonucu bulmak o kadar zorlaştığı için NP-Hard bir yapıya sahiptir (Drira vd., 2007). Genetik algoritmalar, gezgin satıcı problemleri, yerleşim problemleri gibi NP-Hard problemleri çözmek için kullanılan en iyi yöntemlerden bir tanesidir. Bu tez çalışmasında üzerinde çalışılan çok amaçlı problemi çözmek için bir genetik algoritma tasarlanmıştır.

Genetik algoritmalar doğadaki evrim sürecini taklit etmeye dayalı sezgisel tekniklerdir. Doğadaki evrim sürecinde yer alan birçok mekanizma, genetik algoritmalarda probleme ait bir grup çözümün evrimleştirilmesi ve böylece en iyi veya en iyiye yakın çözümlerin elde edilmesi için kullanılmaktadır (Koç, 2007). Genetik algoritmalar ilk defa Bagley'in çalışmalarında kullanılmış, temel ilkeleri Michigan Üniversitesi'nde psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından ortaya atılmıştır. Holland, Darwin'in evrim kuramından etkilenerek canlılarda yaşanan genetik süreci optimizasyon problemlerinde kullanmayı düşünmüştür. Holland'ın çalışmalarının sonucunu açıkladığı kitabının 1975'te yayınlanmasından sonra genetik

algoritmalar birçok bilimsel çalışmada kullanılmıştır. Temel amaçları, deterministik algoritmaların çok maliyetli olduğu problemleri makul zaman ve iyi bir performans ile çözmektir.

Genetik algoritmalarda kullanılan terminoloji genetik biliminin terminolojisi ile bağdaştırılarak açıklanabilir. Genetik algoritmalar kromozomların oluşturduğu bir popülasyon üzerinde çalışmaktadır. Bir kromozom, probleme ait karar değişken değerlerinin genetik gösterim ismi verilen özel bir yapı üzerinde kodlanması ile oluşturulmuş bir vektördür. Dolayısıyla her bir kromozom probleme ilişkin bir çözümü kodlamakta ve temsil etmektedir. Her bir kromozom kodladığı çözümün kalitesini gösteren bir uygunluk değerine (fitness value) sahiptir. Karar değişkenlerinin kodlanması ile oluşan kromozoma genotip, kromozomun kodladığı karar değişkeni değerlerine fenotip denilmektedir. Kromozom üzerinde yer alan çeşitli gen dizileri problemin karar değişkenlerini kodlar ve her bir genin almış olduğu değerlere allel denir (Koç, 2007).

Genetik algoritmalar kromozomlar arasında genetik bilgi aktarımı yapmak için genetik operatör adı verilen prosedürleri kullanırlar. Genetik algoritmaların bilgi aktarımı için yaptığı her iterasyona jenerasyon (nesil) adı verilmektedir. Genetik algoritmalar temel olarak 3 genetik operatör kullanırlar: Seçim, çaprazlama ve mutasyon. Seçim operatörü sonraki jenerasyon için yeni popülasyonu oluşturmak için mevcut popülasyondan kromozomların uygunluk değerlerini göz önünde bulundurarak seçim yapar. Tüm seçim yöntemlerinde uygunluk değeri fazla olan bireylerin seçilme olasılığı daha yüksektir. Çaprazlama operatörü iki kromozom üzerinde çalışır. Seçilen iki ebeveyn kromozom arasında genetik bilgi alışverişi yapılarak iki yeni yavru kromozom oluşturulur. Böylelikle ebeveyn kromozomlardan uygunluk değeri daha yüksek yavru kromozomlar elde edilebilir. Mutasyon operatörü ise genler üzerinde çalışır. Bu operatör kromozomun kendi genleri üzerinde rastgele değişiklik yapmasını sağlayan operatördür. Kromozom üzerinde gerçekleşen mutasyon, aramanın çözüm uzayında farklı noktalara sıçramasını sağlayarak aramaya yön verebilir. Mutasyon operatörü sayesinde aramanın yerel en iyi noktaya takılması önlenir.

Bu tez çalışmasındaki problemde olduğu gibi çoğu gerçek hayat problemi çok amaçlıdır. Örneğin kapasite artırılırken maliyetlerin en küçüklenmesi çok amaçlı bir optimizasyon problemidir. Eğer bir optimizasyon problemi birden fazla amaç ile değerlendiriliyorsa buna çok amaçlı optimizasyon problemi denilmektedir. Çok amaçlı optimizasyon problemleri çözülmek istenildiğinde iki önemli sorunla karşılaşılır. Bunlardan ilki, literatürdeki matematiksel programlama çözüm yöntemlerinin çoğunluğunun tek amaçlı modellere hitap etmesidir. Bu sorunu aşmak ve tek amaçlı modeller için geliştirilen yöntemlerin gücünden yararlanmak için

skalerleştirme yöntemleri önerilmiştir. İkinci zorluk ise skalerleştirilmiş modelin çözümünün bulunmasıyla ilgilidir (Üstün, 2007).

Skalerleştirme çok amaçlı problemlerin çözümünde kullanılan bir yöntemdir. Skalerleştirme ile çok amaçlı problem tek amaçlı probleme dönüştürülebilmektedir. Böylece çok amaçlı problem tek amaçlı çözüm yöntemleriyle çözülebilir hale getirilir. Aşağıda bazı skalerleştirme yöntemlerinden bahsedilmiştir.

3.2.1. Ağırlıklı toplam skalerleştirme yöntemi

Ağırlıklı toplam skalerleştirme yöntemi, en eski ve en yaygın kullanılan skalerleştirme tekniklerinden birisidir. Bu yöntemde amaç fonksiyonları, pozitif ağırlıklarla çarpılıp toplanarak skalerleştirilir (Üstün, 2007). Amaçlara verilen farklı ağırlık değerleri, farklı pareto çözümlerin elde edilmesini sağlar. Birden fazla pareto optimal çözümün bulunabilmesi için çözüm algoritmasının birden fazla kez farklı ağırlıklarla çalıştırılması gerekir (Kaya ve Fırlı, 2017).

w_k , k 'nci amaç fonksiyonunun ağırlığı olmak üzere ağırlıklı toplam yöntemiyle elde edilen skaler amaç fonksiyonu (4) numaralı fonksiyondaki gibidir;

$$en_{k \in X} \sum_{k=1}^P w_k f_k(x) \quad (4)$$

3.2.2. Epsilon kısıt yöntemi

Epsilon (ϵ) kısıt yönteminde amaç fonksiyonlarını birleştirilmek yerine amaç fonksiyonlarından biri hariç diğerleri kısıtlara eklenir. ϵ -kısıt yönteminin en önemli avantajlarından biri çok amaçlı problemin etkin çözümleri garanti etmede dış bükeylik şartlarına ihtiyaç duymaması ve genel problemler için geçerli olmasıdır. Bu nedenle bu yöntem, ağırlıklı toplam yöntemine göre daha üstündür denilebilir (Üstün, 2007). ϵ -kısıt yöntemi önceden belirlenmiş bir ϵ (epsilon) sabiti ile aralık değerlerini değiştirilerek Pareto etkin kümede arama yapmaktadır. Böyle bir değeri seçme zorunluluğu bu yaklaşımın temel dezavantajlarından biridir. Her bir aralıkta sadece bir çözüm bulunabileceğinden, aralıklar herhangi bir pareto-optimal çözümünü kaçırmamak için yeterince iyi olmalıdır. Amaç fonksiyonu vektörleri arasındaki fark için çok küçük bir değer seçilmesi çok fazla gereksiz iterasyona neden olacaktır (Laumanns vd., 2006).

ϵ -kısıt yönteminin matematiksel modeli (5) ve (6) numaralı fonksiyondaki gibi verilmiştir;

$$f_k(x) \leq \epsilon_k, k = 1, \dots, p \text{ ve } k \neq j, \quad (5)$$

Kısıtları altında

$$enk_{x \in X} f_j(x) \quad (6)$$

3.2.3. Melez skalerleştirme yöntemi

Melez skalerleştirme yöntemi, ε -kısıt yöntemi ile ağırlıklı toplam yönteminin bileşiminden oluşan bir yöntemdir. Bu yöntemde bütün amaç fonksiyonlarına konulan kısıtlar altında amaç fonksiyonlarının ağırlıklı toplamları en küçüklenir. x^0 , çok amaçlı problemin herhangi bir uygun çözümü ve $w \in R_+^p$ olmak üzere melez skalerleştirme yönteminin matematiksel modeli (7) ve (8) numaralı fonksiyonlardaki gibidir (Üstün, 2007);

$$f_k(x) \leq f_k(x^0), k = 1, \dots, p \quad (7)$$

Kısıtları altında

$$enk_{x \in X} \sum_{k=1}^p w_k * f_k(x) \quad (8)$$

Melez skalerleştirme yöntemi, ε -kısıt yönteminin çok amaçlı problem üzerinde dış bükeylik şartı gerektirmeme ve karar vericinin amaç fonksiyonlarının değerleriyle ilgili düzey şartını modele yansıtabilme avantajlarının yanı sıra ağırlıklı toplam yönteminin karar vericinin amaç fonksiyonlarına ağırlık vermesine imkân sağlar (Üstün, 2007).

ε -kısıt yönteminde amaç fonksiyonlarının kısıt kümesine dahil edilmesi problemin çözümünü zorlaştırmaktadır. Bu problemi aşmak için ε -kısıtları gevşeterek bu kısıtlarda bir cezaya bağlı olarak sapmalara izin verilir. *Elastik kısıt skalerleştirme yöntemi*, kısıt gevşetmesinden dolayı eniyileme sürecini kolaylaştırıyor gibi gözükse de ceza parametresi, ceza fonksiyonlarıyla ilgili birçok hesaplama zorluğunu da beraberinde getirir (Üstün, 2007).

$$f_k(x) - S_k \leq \varepsilon_k, k = 1, \dots, p \text{ ve } k \neq j, \quad (9)$$

$$S_k \geq 0, k = 1, \dots, k \neq j, \quad (10)$$

Kısıtları altında

$$enk_{x \in X} f_j(x) + \sum_{k \neq j} \mu_x S_k \quad (11)$$

3.2.4. Konik skalerleştirme yöntemi

Konik skalerleştirme yöntemi, Gasimov tarafından genel çok amaçlı eniyileme problemlerinin çözümü için geliştirilmiştir (Gasimov, 2001). Gasimov, amaç fonksiyonu ve kısıtlar üzerine herhangi bir kısıtlayıcı şart koymadan amaç fonksiyonlarını birleştirerek tek

fonksiyona dönüştürmüştür. Gasimov'un skalerleştirme tekniğinin dayandığı temel fikir, pareto etkin değerlerin bulunmasında destek konilerin kullanılmasıdır. Yöntemin en büyük avantajı çok geniş bir problem sınıfına hitap etmesidir. KSY, birçok doğrusal olmayan çok amaçlı programlama problemine uygulanmış ve başarılı sonuçlar elde edilmiştir. Konik skalerleştirme yöntemi, Gasimov (2001) tarafından (12) numaralı fonksiyondaki şekilde ifade edilmiştir;

$$enk[\alpha \sum_{i=1}^k |F_i(x) - B_i| + \sum_{i=1}^k w e_i (F_i(x) - B_i)] \quad (12)$$

Burada α ve $w e_i$ kullanıcı tarafından belirlenmesi gereken parametrelerdir. Parametrelerin $0 \leq \alpha < enk\{w e_1, w e_2\}, w e_1, w e_2 > 0$ koşullarını sağlayacak şekilde seçilmeleri gerekmektedir. Kullanılan α değeri oluşturulacak destek konisinin tepe açısını değiştirmek için kullanılmaktadır. B_i mutlak değer içindeki ifadenin her zaman pozitif değer almasını önlemek amacıyla kullanılması gereken bir sabit değerdir. Aksi halde konik yapı oluşamaz. B_i aynı zamanda çözümlerin civarında aranacağı bir referans noktası görevini de üstlenir (Sipahioğlu, 2010).

Çok amaçlı optimizasyon problemlerinde x karar vektörünü, $z_j(x)$ j'inci amaç fonksiyonunun aldığı değerleri, $X \in \mathbb{Z}^n$ tamsayı karar vektörü kümesini gösterebilir. Eğer herhangi $x_1, x_2 \in X$ çözümü için $z_j(x_1) \leq z_j(x_2)$ $j = 1, \dots, p$ en az bir amaç fonksiyonunda $z_j(x_1) < z_j(x_2)$ koşulları sağlanıyorsa, x_2 çözümünün x_1 çözümünü baskıladığı söylenir. Böyle bir x_2 çözümü yoksa x_1 çözümüne etkin çözüm ve $z(x_1)$ noktasına da baskın nokta denir. Eğer herhangi $x_1, x_2 \in X$ çözümü için $z_j(x_1) < z_j(x_2)$ $j = 1, \dots, p$ koşulu sağlanıyorsa, x_2 çözümünün x_1 çözümünü tam baskıladığı söylenir. Eğer böyle bir x_2 çözümü yoksa x_1 çözümüne zayıf etkin çözüm ve $z(x_1)$ noktasına zayıf baskın nokta denir. Tüm baskın noktalar aynı zamanda zayıf baskın noktalar. Tüm baskın noktaları veya bir kısmını bulmak için önerilen birçok yöntem vardır. Bu yöntemlerden biri, amaç fonksiyonlarını pozitif ağırlıklandırarak çok amaçlı optimizasyon problemini tek amaçlı optimizasyon problemine dönüştürmektir (Lokman, 2017).

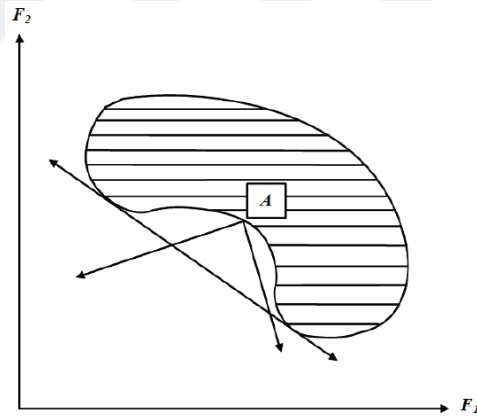
Bu tez çalışmasındaki problemin klasik ağırlıklandırma yöntemine göre skaler amaç fonksiyonu (13) numaralı fonksiyondaki gibidir;

$$enk F = (w_1 F_1 + w_2 F_2 + w_3 F_3) \quad (13)$$

Bu fonksiyonda F değeri skaler amaç fonksiyonunu, F_1, F_2 ve F_3 amaç fonksiyonlarını ve w_1, w_2 ve w_3 amaç fonksiyonlarının ağırlık değerlerini göstermektedir.

Bu tez çalışmasında çalışılan problem, birbirleriyle kıyaslanamaz amaçların aynı anda optimizasyonunu içermektedir. Genellikle bu tür problemlerde tek bir çözüm yoktur, fakat alternatif çözümler kümesi vardır. Bu alternatif çözümler kümesinin elemanları için bütün amaçlar düşünüldüğünde hiçbirinin diğerine göre daha iyi çözüm olduğu söylenemez. Bu tür çözümlere pareto-optimal çözümler denir (Zitzler ve Thiele, 1998).

Çok amaçlı problemi çözmek için bu çalışmada kullanılan diğer yöntem konik skalerleştirme yöntemidir. Konik skalerleştirme yöntemi, çok amaçlı optimizasyon problemlerinin çözümü için Gasimov (2001) tarafından geliştirilmiştir. Gasimov'un skalerleştirme tekniğinin temel fikri, pareto etkin değerleri bulmak için destek konilerini kullanmasıdır. Literatürde konik skalerleştirme yöntemi, birçok konveks olmayan çok amaçlı optimizasyon problemine uygulanmış ve başarılı sonuçlar elde edilmiştir. Çok geniş bir problem sınıfına uygulanabilmesi yöntemin avantajı olarak değerlendirilebilir. Konik skalerleştirme yöntemi, geliştirilmesine temel dayanak olan iki amaçlı bir problem için konveks olmayan görüntü kümesi Şekil 3.6'da gösterilmiştir (Yaşar ve Özyön, 2018).



Şekil 3.6. İki amaçlı bir problem için konveks olmayan görüntü kümesi (Üstün, 2007).

Konik skalerizasyon ile skaler hale getirilecek olan problemin amaçları (14), (15) ve (16) numaralı fonksiyonlardaki gibidir:

$$\text{Min } F_1 = \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{ij} \quad (14)$$

$$\text{Min } F_2 = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{r_{ij}}{1 + S_{ij}}} \quad (15)$$

$$\text{Min } F_3 = \sum_{i=1}^N U_i \quad (16)$$

Konik skalerleştirme yöntemine göre ise skaler amaç fonksiyonu (17) numaralı fonksiyondaki gibidir:

$$\text{enk } F = \alpha \left[|F_1 - B_1| + |F_2 - B_2| + |F_3 - B_3| \right] + \left[w_1 (F_1 - B_1) + w_2 (F_2 - B_2) + w_3 (F_3 - B_3) \right] \quad (17)$$

Yukarıdaki amaç eşitliklerinde d_{ij} i'inci ve j'inci blokların merkezleri arasındaki mesafeyi, r_{ij} i'inci ve j'inci bloklar arasındaki *yakınlık-önem* değerini, S_{ij} i'inci ve j'inci bloklar arasındaki en yakın mesafeyi, U_i i'inci bloğun savunma çemberleri dışında kalan alanını, F_1 , F_2 ve F_3 amaç fonksiyonu değerlerini, α koni açısını belirleyen değeri, w_1 , w_2 ve w_3 amaç fonksiyonları için ağırlık değerini, B_1 , B_2 ve B_3 ise referans değerlerini gösterir. Konik skalerleştirme yönteminde w_1 , w_2 ve w_3 pozitif sayı ve α bu sayıları aşmayacak bir pozitif sayıdır. B_1 , B_2 ve B_3 skaler fonksiyonun bir uygun çözümünden elde edilen değerlerdir.

Problem amaçlarından F_1 fonksiyonu blokların merkezleri arasındaki mesafeleri toplamını, F_2 fonksiyonu blokların *yakınlık-önem* değerinin en yakın mesafelerine oranlarının toplamını ve F_3 fonksiyonu blokların savunma çemberleri dışından kalan alanları toplamını temsil etmektedir. Birimleri farklı olan bu fonksiyonları aynı fonksiyon içinde kullanabilmek ve birbirleriyle karşılaştırabilmek için veya aşırı büyük veya aşırı küçük değerlerin skaler fonksiyonu yanlış yönlendirmesini önlemek için amaç fonksiyonlarının normalize edilmesi gerekmektedir. Normalizasyon çoğunlukla girdi değerlerinin 0 ile 1 arasına ölçeklendirilmesiyle sağlanır (Yavuz ve Deveci, 2012). Ancak her problem için farklı normalizasyon yöntemi kullanılabilir. Literatürde birçok normalizasyon yöntemi vardır. Bunlardan bazıları; Min-Max normalizasyonu, Medyan normalizasyonu, Sigmoid normalizasyonu ve Z-Score normalizasyonu olarak sıralanabilir. Üzerinde çalışılan problemin amaç fonksiyonlarının normalizasyonu için fonksiyonun alabileceği alt ve üst sınırların kullanıldığı bir yöntem kullanılmıştır (Marler ve Arora, 2005). Amaç fonksiyonları problemin ideal ve nadir noktaları arasında normalize edilir. Tez çalışmasında üzerinde durulan 3 amaçlı problemin ideal ve nadir noktaları Çizelge 3'teki gibi hesaplanmaktadır. Çizelge 3'te problem her bir amaç için tek amaçlı olarak çözülmüştür. Elde edilen sonuçlara göre en küçük değer amaç fonksiyonunun ideal noktası en büyük değer nadir noktası olarak belirlenmiştir. Çizelge 3.2'de yıldız ile işaretli değerler ideal noktayı altı çizili değerler nadir noktayı göstermektedir.

Çizelge 3.2. İdeal ve nadir noktalarının hesaplanması.

	F_1	F_2	F_3
$w_1 = 1, w_2 = 0, w_3 = 0$	3112.5404*	0.0112	118
$w_1 = 0, w_2 = 1, w_3 = 0$	3920.3210	0.0072*	116
$w_1 = 0, w_2 = 0, w_3 = 1$	3856.6260	0.0108	48*

Normalize edilmiş amaç fonksiyonu (18) numaralı fonksiyondaki gibi gösterilmiştir.

$$F_i^{Norm} = \frac{F_i - I_i}{N_i - I_i} \quad (18)$$

Burada F_i^{Norm} normalize edilmiş i 'inci amaç fonksiyonunu, I_i i 'inci amaç fonksiyonunun ideal noktasını ve N_i i 'inci amaç fonksiyonunun nadir noktasını göstermektedir.

Konik skalerizasyon yönteminde kullanmak için B_i referans değerlerinin de normalize edilmesi gereklidir. Normalize edilmiş B_i referans değeri (19) numaralı fonksiyondaki gibi hesaplanmaktadır.

$$B_i^{Norm} = \frac{B_i - I_i}{N_i - I_i} \quad (19)$$

Denklemden B_i^{Norm} normalize edilmiş i 'inci referans değerini, B_i i 'inci referans değerini, I_i i 'inci amaç fonksiyonunun ideal noktasını ve N_i i 'inci amaç fonksiyonunun nadir noktasını göstermektedir.

Normalize edilmiş ağırlıklı toplam amaç fonksiyonu;

$$\text{Min } F = \left[w_1 \left(\frac{F_1 - I_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - I_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - I_3}{N_3 - I_3} \right) \right] \quad (20)$$

Normalize edilmiş konik skalerizasyon amaç fonksiyonu;

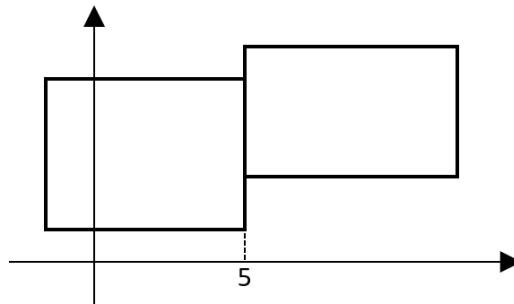
$$\text{Min } F = \left[w_1 \left(\frac{F_1 - B_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - B_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - B_3}{N_3 - I_3} \right) \right] + \alpha \left[\frac{|F_1 - B_1|}{N_1 - I_1} + \frac{|F_2 - B_2|}{N_2 - I_2} + \frac{|F_3 - B_3|}{N_3 - I_3} \right] \quad (21)$$

3.3. Problem İin nerilen Model

Bu tez alıřmasında iki boyutlu yapıların özüm alanına yerleřtirilmesi iin ok amalı bir genetik algorithmadan yararlanılmıřtır. nerilen genetik algoritma ile yerleřtirilecek olan blokların yerleřtirilme sırası ve ynü belirlenecektir. Sırası ve ynü belirlenmiř blokları özüm alanına yerleřtirmek iin ise bir yerleřtirme algoritması kullanılmıřtır. Yerleřtirme algoritması aynı zamanda genetik algorithma kromozomların uygunluk deęerlerinin hesaplanmasında kullanılmıřtır. Problem amalarından birisi olan blokların savunma yarıapları dıřında kalan alanlarının toplamının en küklenmesi amacıyla bir dikdrtgen ile dairelerin keřiřim alanının hesaplaması iin dikdrtgen üzerindeki noktaların daireye gre konumunu kontrol eden bir yaklařım kullanılmıřtır.

Problemle ilgili varsayımlar ařaęıdaki gibidir:

- 1- özüm alanı sonsuz boyuttur. Bloklar özüm alanında herhangi bir yere yerleřtirilebilir.
- 2- Bloklar, eni ve boyu sabit olan dikdrtgen iki boyutlu Őekillerdir.
- 3- Bloklar özüm alanına yerleřtirilirken sadece 90 derece dndürülebilirler.
- 4- Blokların özüm alanına örtüřmeyecek Őekilde yerleřtirileceęi varsayılacaktır.
- 5- Bloklar özüm alanına ortogonal yani dik aılı olacak Őekilde yerleřtirilirler. Her bloęun yatay kenarları X -eksenine, dikey kenarları Y -eksenine paralel olmalıdır.
- 6- Aynı nokta üzerinde birden fazla bloęun sınırı bulunabilir. Aynı nokta üzerinde bulunan 2 blok Őekil 3.7'de rnek olarak gsterilmiřtir.
- 7- Savunma yarıapı deęeri 0 olan blokların savunma yarıapı yoktur.



Şekil 3.7. Aynı nokta üzerinde bulunan iki blok.

Problemin matematiksel modeli aşağıdaki gibidir;

İndisler ve parametreler;

i, j, k : bloklar

L_i : i'inci bloğun boyu

W_i : i'inci bloğun eni,

R_i : i'inci bloğun savunma yarıçapı,

g : bloğun eni üzerinde bir nokta,

h : bloğun boyu üzerinde bir nokta,

T_{igh} : i'inci bloğun üzerindeki (g,h) noktasının savunma yarıçaplarına göre durumu. Eğer bir savunma yarıçapı içinde kalıyorsa 1 diğer durumlarda 0 değerini alır.

T_{ighk} : i'inci bloğun üzerindeki (g,h) noktasının k'inci bloğun savunma yarıçapına göre durumu. Eğer savunma yarıçapı içinde kalıyorsa 1 diğer durumlarda 0 değerini alır.

r_{ij} : i'inci ve j'inci bloklar arasındaki yakınlık önem değeri (ilişki değeri),

d_{ij} : i'inci ve j'inci blokların merkezleri arasındaki öklid mesafesi,

S_{ij} : i'inci ve j'inci blokların en yakın noktaları arasındaki öklid mesafesi,

U_i : i'inci bloğun savunma çemberleri dışında kalan alanı,

N : Blok sayısı,

$\lambda_{x_{ij}}, \lambda_{y_{ij}}$: Örtüşme fonksiyonları

Problemin karar değişkenleri;

M_i^x : i'inci bloğun sol alt köşesinin X koordinatı,

M_i^y : i'inci bloğun sol alt köşesinin Y koordinatı,

M_i^o : i'inci bloğun yerleştirme yönü. Eğer i'inci bloğun kısa kenarı X eksenine paralel olacak şekilde yerleştirilecekse 1, diğer durumlarda 0 değerini alır.

Problemin kısıtı:

Kısıt

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N \beta_{ij} = 0$$

Burada β_{ij} örtüşme durumunu göstermektedir. Eğer iki blok arasında örtüşme varsa 1, yoksa 0 değerini alır. Örtüşme durumu β_{ij} (22) numaralı fonksiyondaki gibi hesaplanmaktadır;

$$\beta_{ij} = \begin{cases} 1, & \lambda_{X_{ij}} > 0 \text{ ve } \lambda_{Y_{ij}} > 0 \\ 0, & \lambda_{X_{ij}} \leq 0 \text{ veya } \lambda_{Y_{ij}} \leq 0 \end{cases} \quad (22)$$

$$\lambda_{X_{ij}} = \frac{1}{2} \left[\left(W_i M_i^o + L_i (1 - M_i^o) \right) + \left(W_j M_j^o + L_j (1 - M_j^o) \right) \right] - \left| M_i^X - M_j^X \right|$$

$$\lambda_{Y_{ij}} = \frac{1}{2} \left[\left(L_i M_i^o + W_i (1 - M_i^o) \right) + \left(L_j M_j^o + W_j (1 - M_j^o) \right) \right] - \left| M_i^Y - M_j^Y \right|$$

$$M_i^o \in \{0, 1\}, i = 1, 2, 3, \dots, N$$

$$M_j^o \in \{0, 1\}, j = 1, 2, 3, \dots, N$$

$$\beta_{ij} \in \{0, 1\}, i = 1, 2, 3, \dots, N \text{ ve } j \neq i$$

Yukarıdaki kısıt altında amaçlar:

Amaç 1

$$\text{Min } F_1 = \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{ij}$$

F_1 amaç fonksiyonuna göre tüm blokların merkezleri arasındaki d_{ij} mesafeleri toplamı minimum olmalıdır. d_{ij} değeri (23) numaralı fonksiyondaki gibi hesaplanmaktadır;

$$d_{ij} = \sqrt{\left[\left(M_i^X + \frac{W_i}{2} \right) - \left(M_j^X + \frac{W_j}{2} \right) \right]^2 - \left[\left(M_i^Y + \frac{L_i}{2} \right) - \left(M_j^Y + \frac{L_j}{2} \right) \right]^2} \quad (23)$$

Amaç 2

$$\text{Min } F_2 = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{r_{ij}}{1+S_{ij}}}$$

F_2 amaç fonksiyonunda N blok sayısını, r_{ij} yakınlık-önem matrisine göre i'inci ve j'inci bloklar arasındaki ilişkiyi, S_{ij} de i'inci ve j'inci blokların birbirine en yakın noktaları arasındaki mesafeyi göstermektedir. İki blok arasındaki yakınlık-önem değerinin mesafeye oranı bu amaç fonksiyonunun değerini verecektir. İki blok arasındaki mesafe sıfır olduğu durumlarda tanımsız değer oluşmasını engellemek için fonksiyonun paydasına 1 eklenerek tanımsız olma durumu ortadan kaldırılmıştır. F_2 amaç fonksiyonuna göre birbiri ile yakınlık-önem ilişkisi olan bloklar yakınlık-önem değerlerine göre birbirine yakın mesafede olmalıdırlar. İki blok arasındaki yakınlık-önem değerinin en yakın mesafelerine oranı en büyük olacak şekilde çözüm alanına yerleştirilmelidir. S_{ij} (24) numaralı fonksiyondaki gibi hesaplanmaktadır. Denklemden, $\lambda_{x_{ij}}$ ve $\lambda_{y_{ij}}$ örtüşme fonksiyonlarını ve M_i^0 , M_j^0 , M_i^x , M_j^x , M_i^y , ve M_j^y karar değişkenlerini göstermektedir.

$$\begin{aligned} \lambda_{x_{ij}} &= \frac{1}{2} \left[\left(W_i M_i^0 + L_i (1 - M_i^0) \right) + \left(W_j M_j^0 + L_j (1 - M_j^0) \right) \right] - \left| M_i^x - M_j^x \right| \\ \lambda_{y_{ij}} &= \frac{1}{2} \left[\left(L_i M_i^0 + W_i (1 - M_i^0) \right) + \left(L_j M_j^0 + W_j (1 - M_j^0) \right) \right] - \left| M_i^y - M_j^y \right| \\ S_{ij} &= \begin{cases} 0 & , \lambda_{x_{ij}} = 0 \text{ veya } \lambda_{y_{ij}} = 0 \\ \left| \left(M_i^x + \frac{W_i}{2} \right) - \left(M_j^x + \frac{W_j}{2} \right) \right| - \left(\frac{W_i + W_j}{2} \right) & , \lambda_{x_{ij}} < 0 \text{ ve } \lambda_{y_{ij}} > 0 \\ \left| \left(M_i^y + \frac{L_i}{2} \right) - \left(M_j^y + \frac{L_j}{2} \right) \right| - \left(\frac{L_i + L_j}{2} \right) & , \lambda_{x_{ij}} > 0 \text{ ve } \lambda_{y_{ij}} < 0 \\ \sqrt{\left(\left| \left(M_i^x + \frac{W_i}{2} \right) - \left(M_j^x + \frac{W_j}{2} \right) \right| - \left(\frac{W_i + W_j}{2} \right) \right)^2 + \left(\left| \left(M_i^y + \frac{L_i}{2} \right) - \left(M_j^y + \frac{L_j}{2} \right) \right| - \left(\frac{L_i + L_j}{2} \right) \right)^2} & , \lambda_{x_{ij}} < 0 \text{ ve } \lambda_{y_{ij}} < 0 \end{cases} \quad (24) \end{aligned}$$

Amaç 3

$$\text{Min } F_3 = \sum_{i=1}^N U_i$$

F_3 amaç fonksiyonuna göre savunma yarıçapı olmayan blokların savunma yarıçapları dışında kalan alanları toplamı en küçük olmalıdır. U_i i'inci bloğun savunma yarıçapları dışında kalan alanını göstermektedir. U_i (25) numaralı fonksiyondaki gibi hesaplanmaktadır.

$U_i = (\text{i'inci bloğun alanı}) - (\text{i'inci bloğun savunma yarıçapları içinde kalan alanı})$

$$U_i = (W_i L_i) - \sum_{g=0}^{W_i-1} \sum_{h=0}^{L_i-1} T_{igh} \quad (25)$$

$$T_{igh} = \begin{cases} 1, & \sum_{g=0}^{W_i-1} \sum_{h=0}^{L_i-1} \sum_{k=1}^N T_{ighk} > 0 \\ 0, & \text{Diğer durumlarda} \end{cases}$$

$$T_{ighk} = \begin{cases} 1, & \sqrt{\left((M_i^X + 0,5 + g) - \left(M_k^X + \frac{W_k}{2} \right) \right)^2 - \left((M_i^Y + 0,5 + h) - \left(M_k^Y + \frac{L_k}{2} \right) \right)^2} \leq R_k \\ 0, & \text{Diğer durumlarda} \end{cases}$$

Üç amaç fonksiyonunun ağırlıklı toplam ve konik skalerizasyon yöntemleriyle normalize edilmiş skaler amaç fonksiyonları (26) ve (27) numaralı fonksiyonlardaki gibidir;

Normalize edilmiş ağırlıklı toplam amaç fonksiyonu;

$$\text{Min } F = \left[w_1 \left(\frac{F_1 - I_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - I_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - I_3}{N_3 - I_3} \right) \right] \quad (26)$$

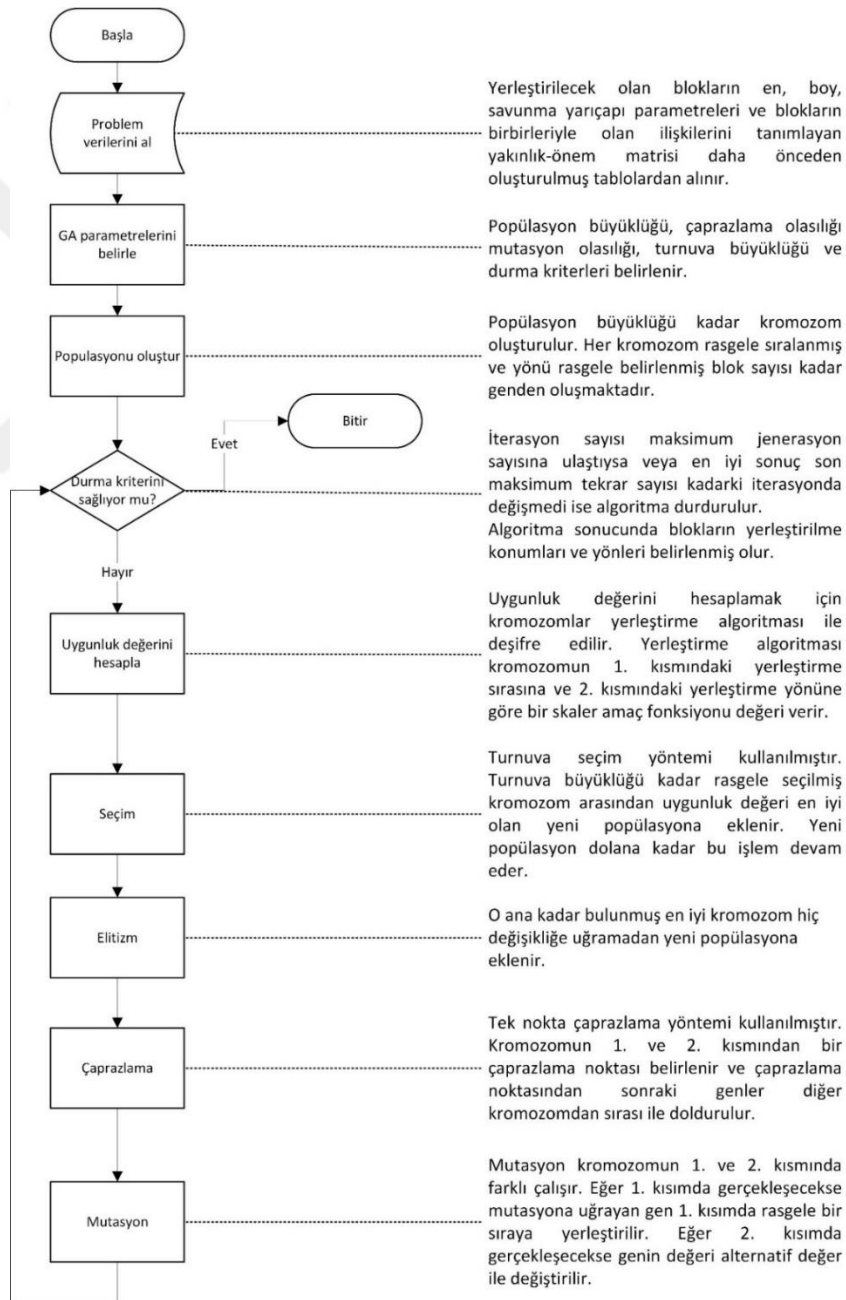
Normalize edilmiş konik skalerizasyon amaç fonksiyonu;

$$\text{Min } F = \left[w_1 \left(\frac{F_1 - B_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - B_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - B_3}{N_3 - I_3} \right) \right] + \alpha \left[\frac{|F_1 - B_1|}{N_1 - I_1} + \frac{|F_2 - B_2|}{N_2 - I_2} + \frac{|F_3 - B_3|}{N_3 - I_3} \right] \quad (27)$$

3.4. Önerilen Çok Amaçlı Genetik Algoritma

Geleneksel bir genetik algoritmanın çalışma adımları Şekil 9'daki akış şemasında gösterilmiştir. Genetik algoritmaya karar verici tarafından; popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, turnuva büyüklüğü ve durma kriteri parametreleri belirlenerek

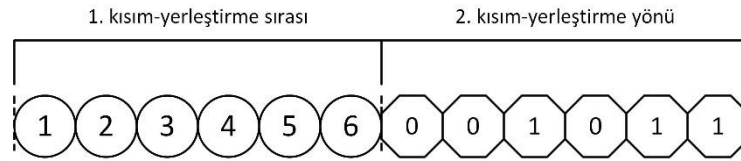
başlanır. Algoritmanın adımları durma kriteri sağlanana kadar devam eder. Durma kriteri iki parametreyi kontrol etmektedir. Birincisi karar verici tarafından belirlenen maksimum jenerasyon sayısı, ikincisi ise yine karar verici tarafından belirlenen maksimum aynı sonucu tekrar sayısıdır. Eğer maksimum tekrar sayısı kadar jenerasyonlardaki en iyi uygunluk değeri değişmedi ise algoritma durdurulur aksi taktirde maksimum jenerasyon sayısına kadar devam edilir. Bazı genetik algoritmalarda çalışma süresi de durdurma kriteri olarak kullanılmaktadır. Bu durumda verilen süre sonunda elde edilen sonuç, algoritmanın sonucu olarak kabul edilir.



Şekil 3.8. Genetik algoritmaların akış şeması (Koç, 2007).

3.4.1. Kromozomun yapısı

Kromozomun yapısı: Bu tez çalışmasında çalışılan iki boyutlu ortogonal yerleşim probleminde blokların yerleştirme sırası ve yönü önemli olduğu için kromozom iki kısımdan oluşacak şekilde tasarlanmıştır. Kromozomun birinci ve ikinci kısmı, yerleştirilecek olan blok sayısı kadar genden oluşur. 6 bloktan oluşan bir kromozom örneği Şekil 3.9'da gösterilmiştir.



Şekil 3.9. 6 bloktan oluşan bir kromozom örneği.

Geliştirilen kromozomun birinci kısmında blokların yerleştirme sırası belirlenir. Bu kısımda, permütasyon kodlama kullanılmış olup, genlerin değerleri blokların numaralarını göstermektedir. İkinci kısımda ise blokların yerleştirme yönü belirlenir. İkinci kısımda genler 0 veya 1 değerini alır. Eğer genin değeri 0 ise blok kısa kenarı X eksenine paralel olacak şekilde yerleştirilir. Eğer genin değeri 1 ise blok uzun kenarı X eksenine paralel olacak şekilde yerleştirilir.

3.4.2. Başlangıç popülasyonunu oluşturma

Başlangıç popülasyonunu oluşturma: Başlangıç popülasyonu karar verici tarafından belirlenmiş olan popülasyon büyüklüğü kadar kromozomdan oluşur. Kromozomun birinci kısmı rastgele sıralanmış bloklardan ikinci kısmı ise rastgele belirlenmiş 0 veya 1 değerlerinden oluşur. Bu sayede popülasyonda yönü ve sırası rastgele belirlenmiş popülasyon büyüklüğü kadar kromozom oluşturulur.

3.4.3. Uygunluk değeri

Uygunluk değeri: Her bir kromozom kodladığı çözümün kalitesini gösteren bir uygunluk değerine sahiptir. Genetik algoritmaların temel çalışma mantığı, tıpkı evrim sürecinde olduğu gibi, uygunluk değerleri daha iyi olan kromozomların çoğalarak artması üzerine kuruludur (Koç, 2007). Uygunluk değeri hesaplanırken kromozomun birinci kısmındaki blok sırasına ve ikinci kısmındaki blok yerleştirme yönüne göre çözüm alanına yerleştiren bir yerleştirme algoritmasından yararlanır. Yerleştirme algoritması Bölüm 3.5'te detaylı olarak anlatılmıştır. Yerleştirme algoritması kromozomun birinci kısmındaki yerleştirme sırası ve ikinci kısmındaki

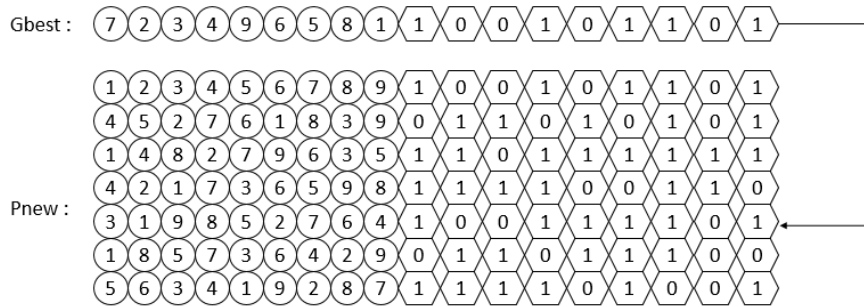
yerleştirme yönüne göre blokları skaler amaç fonksiyonuna göre çözüm alanına yerleştirir. Skaler amaç fonksiyonu değeri uygunluk değerini verir.

3.4.4. Seçim operatörü

Seçim operatörü: Seçim operatörü olarak turnuva seçim operatörü kullanılmıştır. Turnuva seçim operatörü, popülasyondan turnuva büyüklüğü kadar seçilmiş rastgele kromozom arasından en iyi uygunluk değerine sahip olanı yeni popülasyona ekler. Bu işlem yeni popülasyon dolana kadar devam eder. Turnuva büyüklüğünün doğru belirlenmesi yeni oluşturulacak olan popülasyonda genetik çeşitliliğin sağlanması açısından önemlidir. Turnuva büyüklüğünün popülasyon büyüklüğüne eşit olması durumunda yeni popülasyondaki tüm kromozomlar önceki popülasyondaki en iyi uygunluk değerine sahip kromozomdan oluşacaktır. Turnuva büyüklüğü 1 olarak seçilirse seçim operatörü tamamen rastgele seçim yapacaktır.

3.4.5. Elitizm operatörü

Elitizm operatörü: Geliştirilen genetik algorithmada klasik genetik algoritma operatörlerinin dışında elitizm operatörü de kullanılmıştır. Elitizm operatörü o jenerasyona kadar bulunmuş en iyi kromozomun bir sonraki jenerasyona ait popülasyona değişmeden geçmesini sağlamaktadır. Elitizm ile yeni popülasyondan rastgele bir kromozom seçilir ve o jenerasyona kadar bulunmuş en iyi kromozomla değiştirilir. Elitizm işlemi Şekil 3.10'da gösterilmiştir.

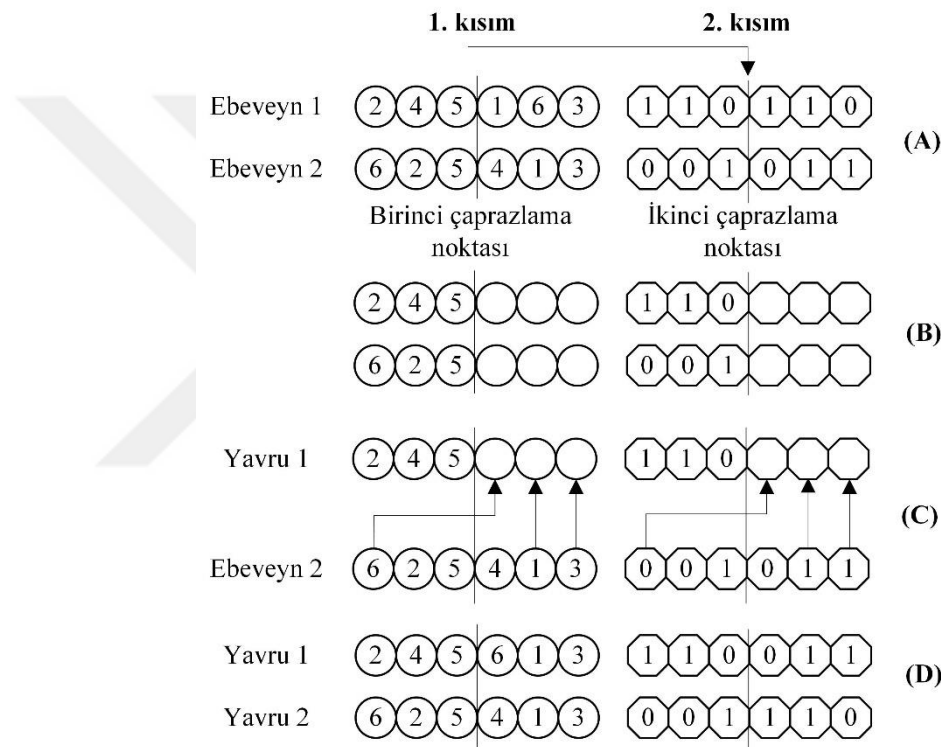


Şekil 3.10. Elitizm işlemi.

3.4.6. Çaprazlama operatörü

Çaprazlama operatörü: Çaprazlama operatörü olarak iki noktalı çaprazlama operatörü kullanılmıştır. Problemin çözümü için blokların sırası ve yerleştirme yönü önemli olduğu için çaprazlama yapılırken kromozomun birinci ve ikinci kısmı beraber hareket ettirilir. Popülasyondan rastgele iki kromozom eşleştirilir ve 0 ile 1 arasında rastgele bir sayı belirlenir.

Eğer bu rastgele sayı karar verici tarafından belirlenmiş olan çaprazlama olasılığından küçükse bu iki kromozom çaprazlanır. Çaprazlama yapılırken kromozomun birinci kısmından rastgele bir nokta belirlenir. Her iki kromozomun birinci ve ikinci kısımlarından çaprazlama noktasından sonraki genleri silinir. Birinci yavru kromozomun çaprazlama noktasından sonraki boş genleri ikinci ebeveynden sırayla doldurulur. Birinci yavru kromozomun birinci kısmı için yapılan işlemler ikinci kısmı için de uygulanır. Aynı işlem ikinci yavru kromozom için de tekrarlanarak iki kromozom çaprazlanır.

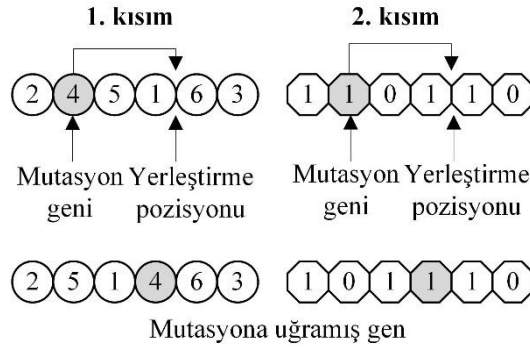


Şekil 3.11. İki noktalı çaprazlama.

3.4.7. Mutasyon operatörü

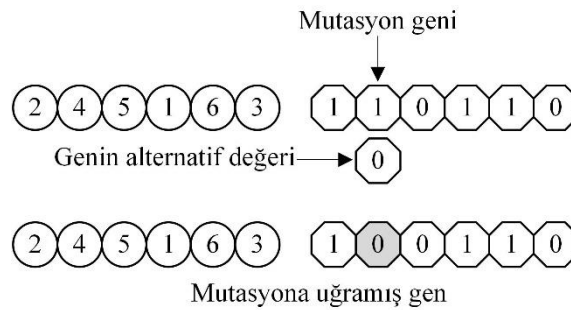
Mutasyon operatörü: Kromozom üzerindeki her gen için 0 ile 1 arasında rastgele bir sayı belirlenir. Eğer bu rastgele sayı karar verici tarafından belirlenmiş olan mutasyon olasılığından küçükse gen mutasyona uğrar. Bu tez çalışmasında geliştirilen mutasyon operatörü kromozomun birinci ve ikinci kısımlarında farklı şekillerde çalışmaktadır. Eğer mutasyon birinci kısımdaki bir gen üzerinde gerçekleşecekse bu gen kromozomun birinci kısmında rastgele bir konuma yerleştirilir. Mutasyonun birinci kısımda gerçekleştiği bu durumda mutasyona uğrayan gen ikinci kısımdaki karşılığı ile beraber hareket ettirilir. Sonuç olarak birinci kısımda gerçekleşen mutasyon

ile mutasyona uğrayan genin yönü değişmeden sadece sırası değiştirilmiş olur. Birinci kısımda gerçekleşen mutasyon Şekil 3.12’de gösterilmiştir.



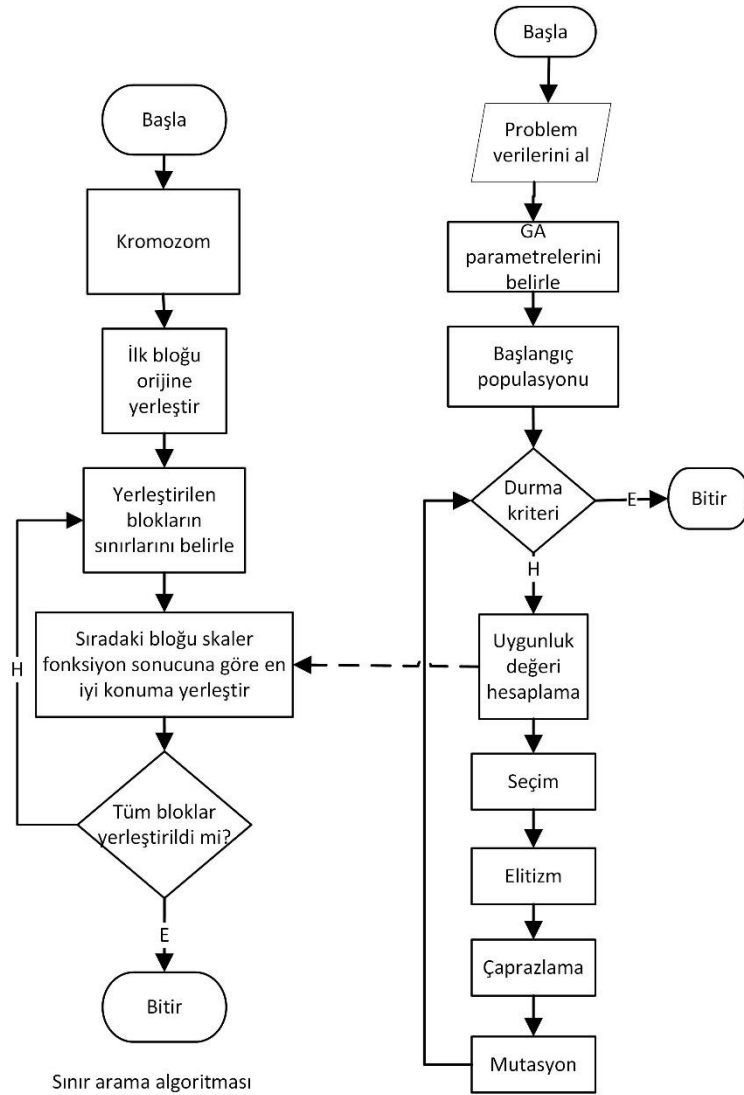
Şekil 3.12. Birinci kısımda gerçekleşen mutasyon.

Eğer mutasyon kromozomun ikinci kısmındaki bir gen üzerinde gerçekleşecekse mutasyona uğrayacak olan genin değeri alternatif değeri ile değiştirilir. Sonuç olarak ikinci kısımda gerçekleşen mutasyon ile mutasyona uğrayan genin sırası değiştirilmeden sadece yönü değiştirilmiş olur. İkinci kısımda gerçekleşen mutasyon Şekil 3.13’te gösterilmiştir.



Şekil 3.13. İkinci kısımda gerçekleşen mutasyon.

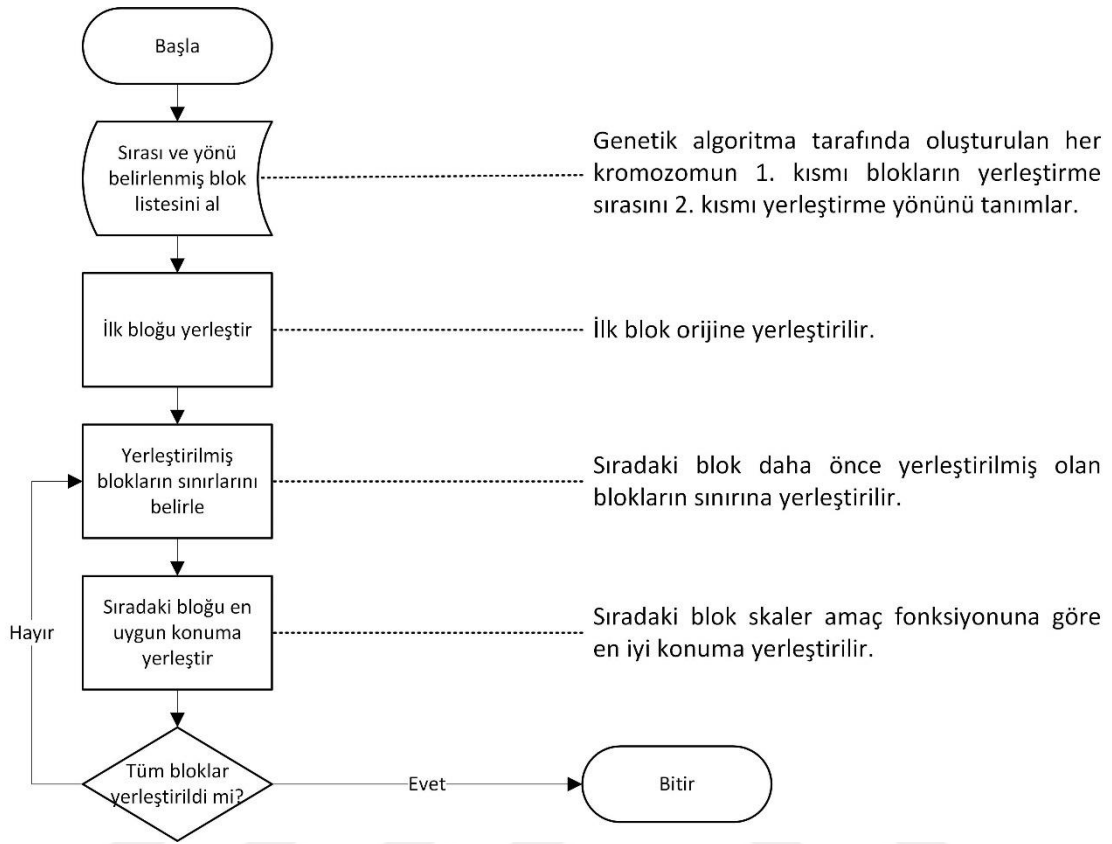
Imam ve Mir (1998) tarafından geliştirilen sınır arama algoritması genetik algortmada kromozomların uygunluk değeri hesaplanırken yerleşim algoritması olarak Şekil 3.14’teki gibi kullanılmıştır. Genetik algortmanın her iterasyonunda uygunluk değeri hesaplanırken her bir kromozom sınır arama algoritması ile deşifre edilir. Yerleşim algoritması (sınır arama algoritması) bir sonraki bölümde detaylı olarak anlatılmıştır.



Şekil 3.14. Sınır arama algoritmasının genetik algoritma ile beraber kullanımı.

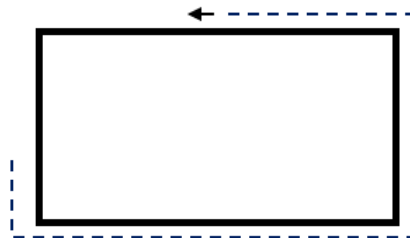
3.5. Yerleşim Algoritması

Geliştirilen genetik algoritma tarafından yerleşim yönü ve sırası belirlenmiş blokların çözüm alanına yerleştirilmesi için Imam ve Mir (1998) tarafından geliştirilen analitik bir yöntem kullanılmıştır. Imam ve Mir (1998) tarafından geliştirilen yerleştirme algoritması her bir bloğu zaten yerleştirilmiş olan blokların sınırlarına yerleştirme esasına dayanır. Bu yerleşim algoritması hangi sırayla çözüm alanına yerleştirileceği belirlenmiş bir blok listesi kullanır. Yerleşim algoritması bu tez çalışmasında blokları çözüm alanına yerleştirirken skaler amaç fonksiyonuna göre daha önce yerleştirilmiş olan blokların sınırında en iyi konuma yerleştirir. Yerleşim algoritmasının akış şeması Şekil 3.15'te gösterilmiştir.



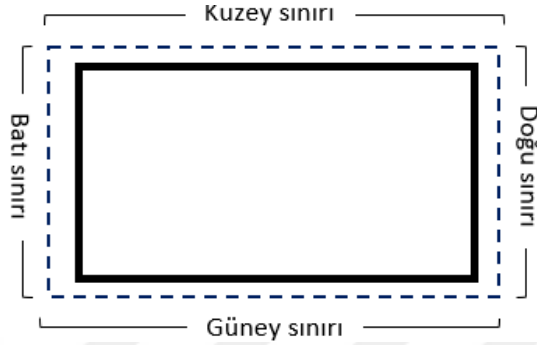
Şekil 3.15. Yerleşim algoritması akış şeması (Imam ve Mir, 1998).

Yerleşim algoritmasına göre sıralanmış blok listesindeki ilk blok, çözüm alanındaki herhangi bir konuma yerleştirilerek çözüme başlanır. Çözüm alanının sınırları olmadığı için diğer blokların yerleştirilebileceği çok sayıda alternatif nokta vardır. Sınırları olmayan çözüm alanını kısıtlamak için diğer bloklar yerleştirilmiş olan blokların sınırlarına yerleştirilir. Şekil 3.16'da yerleştirilmiş olan bir bloğun sınırları gösterilmektedir.



Şekil 3.16. Çözüm alanına yerleştirilmiş bir bloğun yerleştirme sınırları.

Diğer blokları yerleştirilmiş blokların sınırlarına yerleştirebilmek için bir sınır listesi oluşturulur. İlk blok yerleştirildiğinde yerleştirilen bloğun sınırlarındaki her noktanın koordinatı sınır listesine eklenir. Sınır listesine eklenecek olan noktanın bloğun hangi kenarında olduğu belirtilir. Şekil 3.17 çözüm alanına yerleştirilmiş bir bloğun sınırlarının yönlerini göstermektedir.

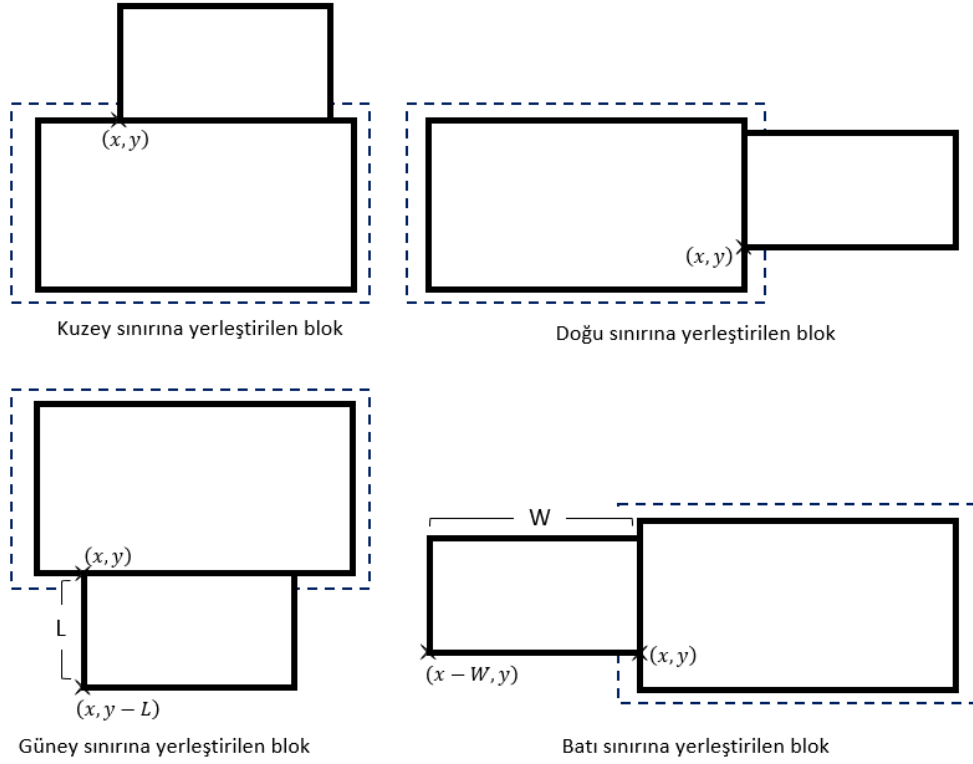


Şekil 3.17. Çözüm alanına yerleştirilmiş bir bloğun sınırlarının yönleri.

Sıradaki blok sınır listesindeki bir noktaya aşağıdaki şekillerde yerleştirilir;

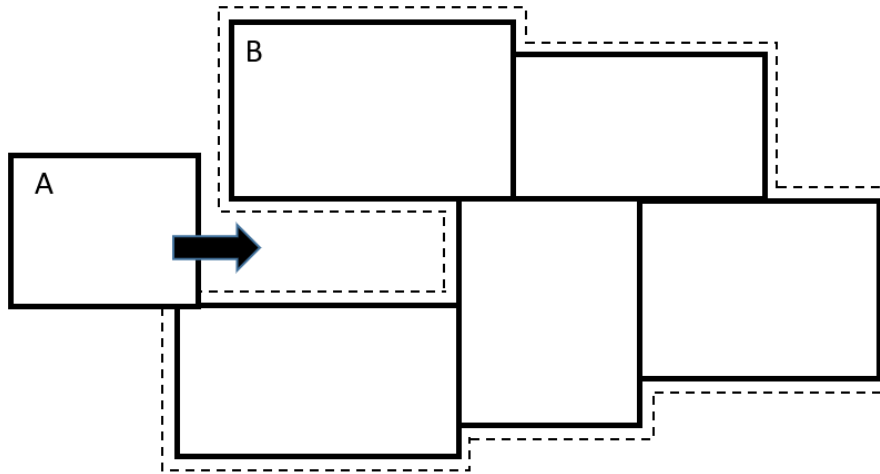
- Eğer blok kuzey sınırındaki bir noktaya yerleştirilecekse yerleştirilecek olan bloğun sol alt köşesi o sınır noktasına yerleştirilir.
- Eğer blok doğu sınırındaki bir noktaya yerleştirilecekse yerleştirilecek olan bloğun sol alt köşesi o sınır noktasına yerleştirilir.
- Eğer güney sınırındaki bir noktaya yerleştirilecekse yerleştirilecek olan bloğun sol alt köşesi o sınır noktasının Y koordinatından bloğun boyu çıkartılarak yerleştirilir.
- Eğer batı sınırındaki bir noktaya yerleştirilecekse yerleştirilecek olan bloğun sol alt köşesi o sınır noktasının X koordinatından bloğun eni çıkartılarak yerleştirilir.

Sınırın yönüne göre bir bloğun yerleştirme konumu Şekil 3.18'de gösterilmiştir.



Şekil 3.18. Sınırın yönüne göre bir bloğun yerleştirilme konumu.

Bir blok çözüm alanına yerleştirilirken dikkat edilmesi gereken diğer bir durum da blokların birbiri ile örtüşme durumudur. Bu durum Şekil 3.19'daki gibi gösterilebilir. Eğer Şekil 3.19'daki A bloğu ok yönünde hareket ettirilirse B bloğu ile örtüşecektir.



Şekil 3.19. Örtüşme durumu.

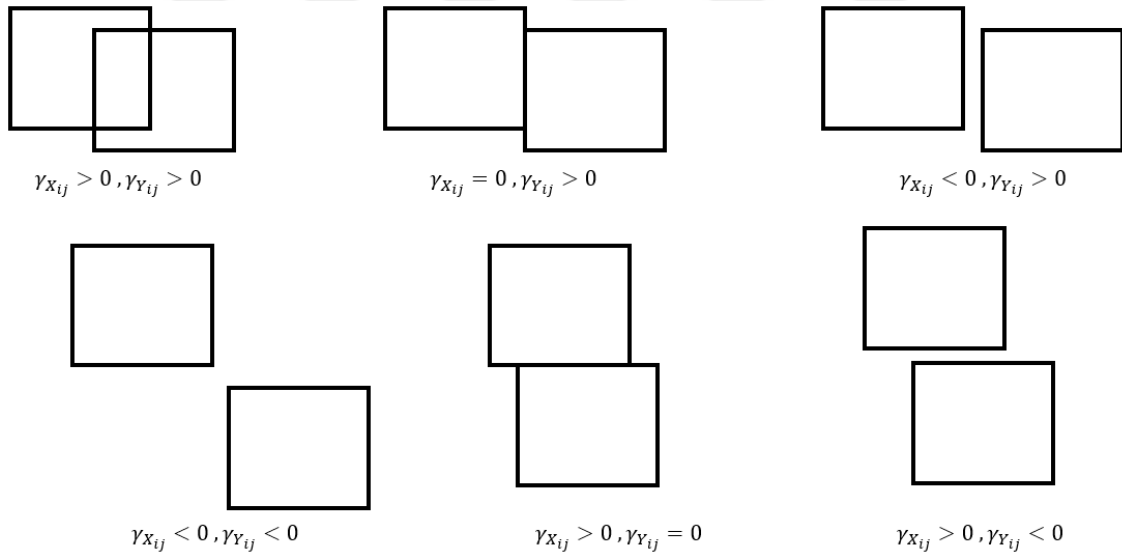
İki blok arasındaki örtüşme kontrolü için (28), (29) ve (30) numaralı fonksiyonlardan faydalanılır (Imam ve Mir, 1998);

$$\beta_{ij} = \begin{cases} 1, \gamma_{X_{ij}} > 0 \text{ ve } \gamma_{Y_{ij}} > 0 \\ 0, \gamma_{X_{ij}} \leq 0 \text{ veya } \gamma_{Y_{ij}} \leq 0 \end{cases} \quad (28)$$

$$\gamma_{X_{ij}} = \frac{1}{2} * (W_i + W_j) - |X_i - X_j| \quad (29)$$

$$\gamma_{Y_{ij}} = \frac{1}{2} * (L_i + L_j) - |Y_i - Y_j| \quad (30)$$

Burada L_i i'inci bloğun boyunu, L_j j'inci bloğun boyunu, W_i i'inci bloğun enini, W_j j'inci bloğun enini, X_i i'inci bloğun merkezinin X koordinatını, X_j j'inci bloğun merkezinin X koordinatını, Y_i i'inci bloğun merkezinin Y koordinatını ve Y_j j'inci bloğun merkezinin Y koordinatını gösterir. Eğer $\beta_{ij} = 0$ ise i'inci ve j'inci bloklar arasında örtüşme yoktur. Eğer $\beta_{ij} = 1$ ise i'inci ve j'inci bloklar arasında örtüşme vardır. Şekil 3.20'de iki blok arasındaki örtüşme durumları gösterilmiştir.



Şekil 3.20. İki blok arasındaki örtüşme durumları.

Yerleşim algoritması her bir bloğu normalize edilmiş skaler amaç fonksiyonunun değerine göre en iyi yere yerleştirir. Tez çalışmasındaki üç amaç fonksiyonunun ağırlıklı toplam ve konik skalerizasyon yöntemleriyle normalize edilmiş skaler amaç fonksiyonları (31) ve (32) numaralı fonksiyonlardaki gibidir.

Normalize edilmiş ağırlıklı toplam amaç fonksiyonu;

$$\text{Min } F = \left[w_1 \left(\frac{F_1 - I_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - I_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - I_3}{N_3 - I_3} \right) \right] \quad (31)$$

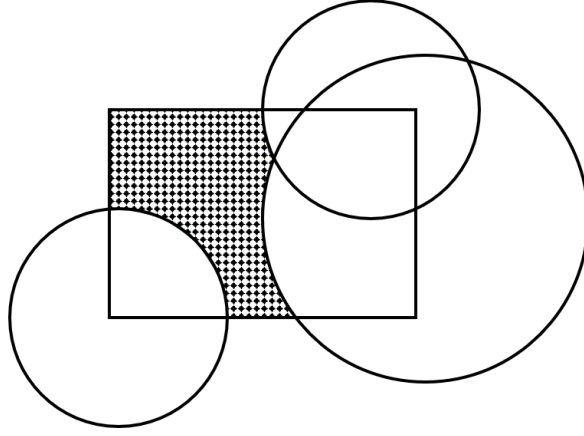
Normalize edilmiş konik skalerizasyon amaç fonksiyonu;

$$\text{Min } F = \left[w_1 \left(\frac{F_1 - B_1}{N_1 - I_1} \right) + w_2 \left(\frac{F_2 - B_2}{N_2 - I_2} \right) + w_3 \left(\frac{F_3 - B_3}{N_3 - I_3} \right) \right] + \alpha \left[\frac{|F_1 - B_1|}{N_1 - I_1} + \frac{|F_2 - B_2|}{N_2 - I_2} + \frac{|F_3 - B_3|}{N_3 - I_3} \right] \quad (32)$$

Blokların savunma yarıçapları dışında kalan alanlarının toplamının en küçüklenmesi için kullanılan F_3 amaç fonksiyonu (33) numaralı fonksiyondaki gibidir;

$$\text{Min } F_3 = \sum_{i=1}^N U_i \quad (33)$$

Burada N blok sayısını, U_i i'inci dikdörtgen bloğun savunma yarıçapları dışında kalan alanını göstermektedir. Bir daire ile bir dikdörtgen arasındaki kesişme alanının hesaplanması kolay olmayan bir problemdir. Bu problemin çözümü için birçok yöntem vardır. Dairenin içinde yer alan dikdörtgenin köşelerini aramak ve daha sonra üçgen ve dairesel kesimlere dayalı alan hesaplamasını kullanmak en hızlı ve en doğru yaklaşım olabilirken, uygulanması çok zordur. Bu yüzden bir dikdörtgen ile daire arasındaki kesişim alanını hesaplamak için kesişen alanı birçok dar dikdörtgene ayırıp bu dikdörtgenlerin alanlarını toplayarak hesaplayan bir yöntem kullanılabilir. Dolayısıyla eğer dikdörtgen ve daire arasındaki kesişim alanı hesaplanabilirse kesişmeyen alan da kolayca hesaplanabilecektir. Ancak böyle bir yöntem genetik algoritmayı büyük ölçüde yavaşlatacaktır. Tez çalışmasında daire ile dikdörtgenin kesişen alanının hesaplanmasında dikdörtgen üzerindeki her noktanın dairelerle kesişip kesişmediğini kontrol eden bir yaklaşım kullanılmıştır. Bu yaklaşımda dikdörtgen içindeki tüm noktalardan dairelerle kesişmeyen noktalar toplanarak kesişmeyen alan hesaplanır. Bu durumun bir örneği Şekil 3.21'deki gibi gösterilebilir.



Şekil 3.21. Kesişmeyen alanın noktalar ile gösterilmiş hali.

4. ÖNERİLEN MODELİN DEĞERLENDİRİLMESİ

Bu çalışmada iki boyutlu dikdörtgen blokların çözüm alanına yerleştirilmesi için tasarlanan genetik algoritma ile ilgili kodlamalar C# programlama dilinde Microsoft Visual Studio 2017 geliştirme ortamında tamamlanmış ve oluşturulan örnek veriler Intel Core™ i7-7700HQ işlemciye ve 8 GB belleğe sahip bir bilgisayar üzerinde test edilmiştir.

Geliştirilen yazılım iki adet arayüzden oluşmaktadır. İlk arayüzde problem verisi alınır, genetik algoritma parametreleri belirlenir, skalerizasyon türü seçilir ve skalerizasyon parametreleri belirlenir. Geliştirilen ilk arayüz Şekil 4.1'de gösterilmiştir.

Şekil 4.1. Problem verilerin alındığı ve algoritma parametrelerinin belirlendiği giriş arayüzü.

Problem verilerin alındığı ve algoritma parametrelerinin belirlendiği giriş arayüzü dört bölümden oluşmaktadır. İlk bölümde problem için genetik algoritma parametreleri belirlenir. Belirlenen genetik algoritma parametreleri; popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, turnuva büyüklüğü ve durma kriterleridir. İkinci bölümde skalerizasyon parametreleri ve normalizasyon değerleri belirlenir. Aynı zamanda ikinci bölümde çok amaçlı problemin amaç fonksiyonlarını skaler hale getirecek olan iki skalerizasyon yönteminden birisi seçilebilir. Üçüncü bölüm problem amaçlarının gösterildiği bölümdür. Dördüncü bölümde dış kaynaktan alınan problem verileri gösterilir. Alınan problem verileri her blok için rasgele bir renk ataması yapılarak listelenir.

Problem verileri ilk arayüzün sol üst köşesinde bulunan “Veri al” butonu ile önceden hazırlanmış bir Microsoft Excel Çizelgesinden alınır. Excel Çizelgesinden blokların en, boy, savunma yarıçapı ve bloklar arası *yakınlık-önem* değeri verileri alınır. En, boy ve savunma yarıçapı verileri Çizelge 4.1’deki gibi hazırlanır. Çizelge 4.1’de “No” sütunu blok numarasını, “En” sütunu blok enini, “Boy” sütunu blok boyunu ve “Yarıçap” sütunu bloğun savunma yarıçapını göstermektedir.

Çizelge 4.1. En, boy ve savunma yarıçapı verileri için hazırlanmış excel çizelgesi.

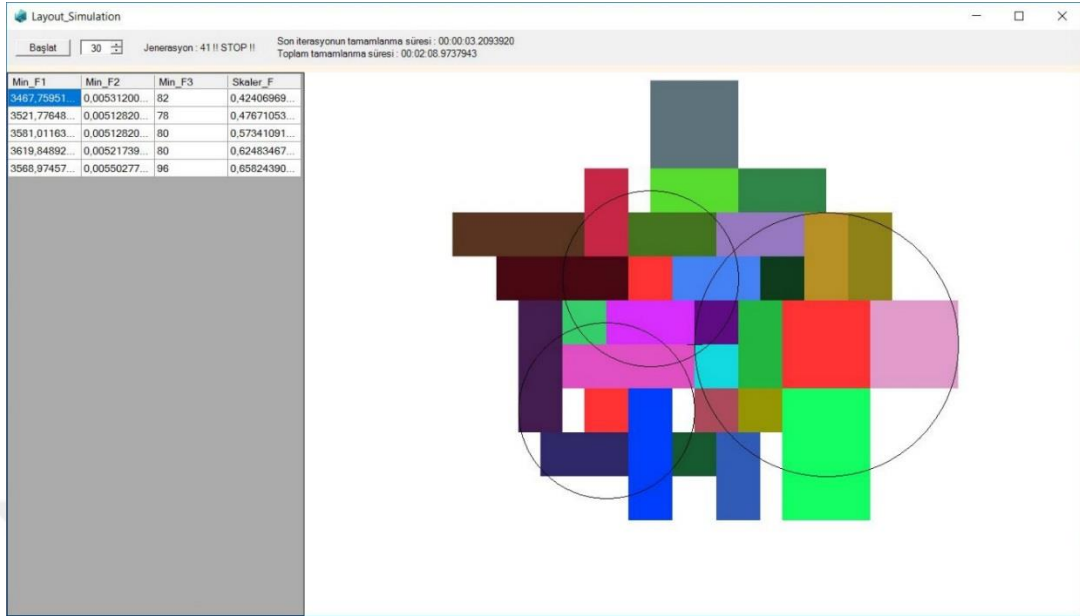
No	En	Boy	Yarıçap
1	4	6	0
2	4	6	6
3	4	4	0
4	4	4	0
5	6	2	0

Bloklar arası *yakınlık-önem* değeri verileri **Çizelge 4.2**’deki gibi hazırlanır. **Çizelge 4.2**’de “No1” ilk blok numarasını, “No2” ikinci blok numarasını ve “Değer” ilk blok ile ikinci blok arasındaki *yakınlık-önem* değerini gösterir.

Çizelge 4.2. Bloklar arası ilişki (yakınlık-önem değeri) verileri için hazırlanmış Excel çizelgesi.

No1	No2	Değer
1	2	0
1	3	0
1	4	10
1	5	10
1	6	0

İkinci arayüzde genetik algoritma çalıştırılır ve çözüm sonuçları görsel olarak gösterilir. İkinci arayüz Şekil 4.2’de gösterilmiştir.



Şekil 4.2. Genetik algoritmanın çalıştırıldığı ve sonuçlarının görsel olarak gösterildiği ikinci arayüz.

Genetik algoritmanın çalıştırıldığı ve sonuçlarının gösterildiği ikinci arayüzün sol kısmındaki Çizelgede o ana kadar bulunmuş en iyi skaler fonksiyon değerleri gösterilir. Çizelgede “Min F1” sütunu blokların merkezleri arasındaki mesafeleri toplamını gösteren amaç fonksiyonu değerini, “Max F2” sütunu blokların birbirleri arasındaki *yakınlık-önem* değerlerinin en yakın mesafelerine oranları toplamını gösteren amaç fonksiyonu değerini, “Min F3” sütunu blokların savunma çemberleri dışında kalan alanları toplamını gösteren amaç fonksiyonu değerini ve “Skaler F” sütunu skaler fonksiyon sonucunu gösterir. Çizelgede sonuçlar en son bulunan sonuç başta olacak şekilde gösterilir.

Bu bölümün bundan sonraki kısmında problemin çözümü için geliştirilen yaklaşım farklı parametreler ile test edilmiştir. 30 bloktan oluşan test veri setinin blok boyutları Çizelge 4.3’te ve bloklar arası *yakınlık-önem* değerleri Çizelge 4.4’te gösterilmiştir.

Çizelge 4.3. 30 bloktan oluşan veri setindeki blokların en, boy ve savunma yarıçapı değerleri.

No	Width	Length	Radius	No	Width	Length	Radius
1	4	6	0	16	4	2	0
2	4	4	6	17	4	2	0
3	4	4	0	18	2	2	4
4	4	4	0	19	2	2	4
5	6	2	0	20	2	2	0
6	6	2	0	21	2	2	0
7	6	2	0	22	2	2	0
8	6	2	0	23	2	2	0
9	6	2	0	24	2	2	0
10	4	2	0	25	2	2	0
11	2	4	0	26	2	2	0
12	4	2	0	27	4	2	0
13	4	2	0	28	4	2	0
14	4	2	0	29	4	2	0
15	4	2	0	30	4	2	0

Çizelge 4.4. 30 bloklı veri setindeki blokların birbirleri arasında ilişkiler (yakınlık-önem değerleri).

	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	18	19	22	23	24	27	30
1				10															20	20		
2						7										15	10					
3																			10			
4	10				20											9						
5				20																		
6		7											5									
7																	15	8				
8																					2	
9														5								
10										3									10			
11										3												
12																		6				
14						5																
15									5									4				
16																						5
18		15		9													10					
19		10					15									10						
22							8					6		4								
23	20		10							10												
24	20																					
27								2														
30															5							

30 bloktan oluşan test veri setinde 1 ve 4 numaralı bloklar arasındaki *yakınlık-önem* değeri 10, 1 ve 23 arasındaki ilişki değeri 20'dir. Benzer şekilde diğer ilişkili blokların *yakınlık-önem* değerleri Çizelge 4.4'deki çizelgede gösterilmiştir. Bunların dışındaki bloklar birbirleriyle ilişkili olmadığı için çizelgede gösterilmemiştir ve *yakınlık-önem* değerleri 0'dır. 30 bloktan oluşan örneğin referans noktaları, ideal noktaları ve nadir noktaları Çizelge 4.5'te gösterilmiştir.

Çizelge 4.5. Referans noktaları, ideal ve nadir noktalar.

Referans noktaları	B_1	B_2	B_3
	3271.458163	0.005752	62.6
İdeal nokta	I_1	I_2	I_3
	3214.786439	0.005128	50
Nadir nokta	N_1	N_2	N_3
	3781.503681	0.011369	176

30 bloktan oluşan örneğin genetik algoritma parametreleri, skalerizasyon parametreleri ve elde edilen sonuçlar Ek-1'de gösterilmiştir. Ek-1'de 30 bloktan oluşan örnek problemde 1 ile 8 arasındaki test verilerinde popülasyon büyüklüğü her iki yöntem için de 20'şer artırılarak, 9 ile 16 arasında çaprazlama olasılığı her iki yöntem için de 0.1 artırılarak, 17 ile 24 arasında turnuva büyüklüğü her iki yöntem için de %10 artırılarak, 25 ile 30 arasında maksimum tekrar sayısı her iki yöntem için de %10 artırılarak, 31 ile 40 arasında konik skalerizasyon yöntemi için α değeri 0.025 artırılarak, 41 ile 60 arasında her iki yöntem farklı ağırlık değerleriyle çözülmüş sonuçlar elde edilmiştir.

Ek-1'de "GA'nın Durduğu İterasyon Sayısı" sütunu genetik algoritmanın durma kriteri sağlandığında hangi iterasyonda algoritmanın sonlandırıldığını, "Toplam Sıçrama Sayısı" sütunu genetik algoritma durana kadar kaç kez en iyi sonuç bulunduğunu, "Tamamlanma Süresi" sütunu genetik algoritma sonlanana kadar ne kadar süre geçtiğini "Min F1", "Min F2" ve "Min F3" sütunları amaç fonksiyonu değerlerini, "N-F1", "N-F2" ve "N-F3" sütunları normalize edilmiş fonksiyon sonuçlarını, "Skaler Sonuç" sütunu Çizelge 10'daki skalerizasyon yöntemine göre elde edilmiş skaler amaç fonksiyonu değerini ve "Orijine Uzaklık" sütunu normalize edilmiş fonksiyon sonuçlarının (0,0,0) noktasına olan öklid uzaklığını, "Referans Değerine Uzaklık" sütunu normalize edilmiş fonksiyon sonuçlarının referans değerine yani (0.1,0.1,0.1) noktasına olan Öklid uzaklığını verir.

Amaç fonksiyonu değerleri normalizasyon ile 0 ile 1 arasında bir değer aldığı ve amaç fonksiyonları en küçükleme olduğu için test sonuçları normalize edilmiş amaç fonksiyonu değerlerinin (0,0,0) noktasına ve referans noktasına (0.1,0.1,0.1) olan uzaklığına göre karşılaştırılabilir. Elde edilen Çizelge 4.11'deki sonuçlara göre;

- Popülasyon büyüklüğü arttıkça her iki yöntem de daha iyi sonuçlar vermektedir.
- Çaprazlama olasılığının 0.7 ile 0.8 arasında seçilmesi genetik algoritmanın çözüm alanında daha fazla arama yapmasını ve daha iyi sonuçlar elde etmesini sağlamıştır.
- Turnuva büyüklüğünün artması büyük ölçüde bir önceki iterasyondaki iyi uygunluk değerine sahip kromozomların yeni jenerasyona geçmesini ve dolayısıyla çeşitliliğin azalmasına neden olmuştur. Bunun sonucu olarak daha düşük turnuva büyüklüklerinin daha iyi sonuçlar verdiği söylenebilir.
- Maksimum tekrar sayısının artması genetik algoritmanın daha uzun süre çalışmasına neden olmaktadır.
- Konik skalerizasyon yönteminde α değerinin artması ile en iyi çözüme daha yakın sonuçlar elde edilmektedir.
- Genetik algoritmanın aynı parametrelerle birden çok çalıştırıldığı durumda genetik algoritmanın rassal tabanlı bir algoritma olması nedeniyle farklı sonuçlar elde edilmiştir.
- Konik skalerizasyon ve ağırlık toplam yöntemleri karşılaştırıldığında (0,0,0) noktasına göre ağırlıklı toplam yönteminin 15 örnekte konik skalerizasyon yönteminin 4 örnekte daha iyi sonuçlar verdiği görülmektedir.
- Ayrıca konik skalerizasyondaki referans değerlerine göre de bir karşılaştırma yapılırsa ağırlıklı toplam yönteminin 15 örnekte konik skalerizasyon yönteminin 4 örnekte daha iyi sonuçlar verdiği görülmektedir.

Elde edilen sonuçlara ait yerleşim görünümleri Ek-2'deki gibi listelenmiştir.

Problem için önerilen model Ek-3'teki 8 farklı problemde popülasyon büyüklüğü 80, çaprazlama olasılığı % 90, mutasyon olasılığı % 0.3, turnuva büyüklüğü % 5, en büyük jenerasyon sayısı 100, en büyük tekrar sayısı % 30 olacak şekilde test edilmiştir. Ek-3'teki her problemin ilişki tabloları Ek-4'de gösterilmiş ve sonuçların yerleşim görünümleri Ek-5'de gösterilmiştir.

5. SONUÇ VE DEĞERLENDİRMELER

Bu tez çalışmasında yerleşim problemlerinin bir çeşidi olan mesafe, yapılar arası ilişkiler ve savunma yarıçapı göz önünde bulundurularak bazı strateji oyunlarında ve bazı operasyon planlarında karşılaşılan iki boyutlu dikdörtgen yapıların sınırları olmayan bir alan yerleştirilmesini inceleyen özel bir problem üzerinde çalışılmıştır. Çalışılan özel problemin çözümünde birçok eniyileme problemlerinde kullanılan sezgisel algoritmalarından biri olan genetik algoritma ile çözüm önerilmiştir. Önerilen genetik algoritmada seçim operatöründe turnuva seçim yöntemi, çaprazlama operatöründe tek nokta çaprazlama yöntemi kullanılmıştır. Uygunluk değerini hesaplamak için yerleştirilmiş yapıların sınırında arama yapan bir yerleştirme algoritması kullanılmıştır. Bulunan en iyi kromozomun sonraki nesillere değişmeden aktarılmasını sağlamak için elitizm operatörü genetik algoritmaya dahil edilmiştir. Geliştirilen genetik algoritmada yapıların yerleşim alanına hangi sırayla ve hangi yönde yerleştirileceği belirlenmiş, sırası ve yönü belirlenen yapılar yerleştirme algoritması ile skaler fonksiyon değerini eniyileyecek şekilde yerleşim alanına yerleştirilmiştir. Çok amaçlı problemin amaç fonksiyonu değerlerini skaler hale getirmek için ağırlıklı toplam yöntemi ve konik skalerizasyon yöntemi kullanılmış ve sonuçlar karşılaştırılmıştır.

Elde edilen sonuçlar, genetik algoritmaların tezde çalışılan özel yerleştirme probleminde başarı ile uygulanabileceğini göstermiştir. Uygunluk değerini hesaplamak için kullanılan yerleştirme algoritması çözüm uzayını büyük ölçüde küçülttüğü için bulunan en kötü skaler fonksiyon sonucunda dahi yapıların merkezleri arasındaki mesafelerin en küçüklenmesini ve birbiriyle ilişkili yapıların birbirlerine yakın olmasını ifade eden amaç fonksiyonu değerlerinin çok iyi sonuçlar verdiği görülmüştür. Bu nedenle çok amaçlı problem amaçlarını skaler hale getirmek için ağırlıklı toplam veya konik skalerizasyon yöntemlerinden herhangi birinin kullanılmasında sakınca yoktur. Dördüncü bölümde 30 bloktan oluşan test verisinin farklı parametrelerle elde edilen sonuçlarına göre bu problemde ağırlıklı toplam yönteminin konik skalerizasyon yöntemine göre daha iyi sonuçlar verdiği görülmüştür. Ayrıca referans değerine göre yapılan karşılaştırmada (Referans değerinin normalize edilmiş fonksiyonların en iyi değeri olan (0,0,0) noktasına çok yakın olmasına rağmen) ağırlıklı toplam yönteminin daha iyi sonuçlar verdiği görülmüştür.

Geliştirilen genetik algoritma çalışma süresi bakımında değerlendirildiğinde aynı popülasyon büyüklüğüne sahip örneklerde her bir iterasyonun tamamlanma süresinin her iki skalerizasyon yönteminde de çok yaklaşık değerler aldığı görülmüştür. Çalışma süresinin skalerizasyon yöntemine göre değil durma kriterine göre değiştiği söylenebilir.

Uygunluk deęerini hesaplamak için kullanılan yerleřtirme algoritmasında yerleřtirilmiř yapıların uygun olmayan sınırlarının arama algoritmasından çıkarılmıř olmasına raęmen tüm uygun sınırların taranması genetik algoritmayı bir miktar yavařlatmaktadır. Tüm uygun sınırları taramadan en iyi skaler fonksiyon sonucunu arayacak bir yaklařım ile genetik algoritmanın daha hızlı çalıřması saęlanabilir.

Önerilen sistemin unsurlarının oyun alanına yerleřtirilmesinin önemli olduęu strateji oyunlarında ve bazı operasyon planlarında karřılařılan iki boyutlu dikdörtgen yapıların sınırları olmayan bir alana yerleřtirilmesinde iyi sonuçlar verebileceęinde uygulama alanının gelecekte artabileceęi öngörülmektedir.



KAYNAKLAR DİZİNİ

- Acar, E. ve Durucasu, H. (2014). Özel sektör şirketleri için yid ihale probleminin çok amaçlı optimizasyonu. *Alphanumeric Journal*, 2(2), 36-49.
- Ahmad, A. R., Basir, O. A., Imam, M. H., ve Hassanein, K. (2006). An efficient, effective, and robust decoding heuristic for metaheuristics-based layout optimization. *International journal of production research*, 44(8), 1545-1567.
- Al-Saleh, M. A., Mir, M., ve Hassanin, A. (2015). Comparison of enhanced constructive layout optimization technique with Tabu-search and Particle Swarm Optimization methodologies. In *Industrial Engineering and Operations Management (IEOM), 2015 International Conference on* (s.1-7). IEEE.
- Baker, B. S., Coffman, Jr, E. G., ve Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on computing*, 9(4), 846-855.
- Burke, E. K., Kendall, G., ve Whitwell, G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4), 655-671.
- Costa, G., Delorme, M., Iori, M., Malaguti, E., ve Martello, S. (2017). Training software for orthogonal packing problems. *Computers & Industrial Engineering*, 111, 139-147.
- Çelik, Ş. A., Kayacan, M. C., Aydoğan, T., ve Çakır, A. (2002). Bilgisayar kontrollü kumaş kesme makinası tasarımı ve imalatı. *Politeknik Dergisi*, 5(2).
- Çolak, S. (2010). Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 19(3).
- Deb, K., Pratap, A., Agarwal, S., ve Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.
- Delorme, M., Iori, M., ve Martello, S. (2017). Logic based Benders' decomposition for orthogonal stock cutting problems. *Computers & Operations Research*, 78, 290-298.
- Drira, A., Pierreval, H., ve Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual reviews in control*, 31(2), 255-267.
- Ergül, E. U. (2015). Çok amaçlı genetik algoritma yöntemlerinin başarımının belirlenmesi için iki yeni ölçüt önerisi. *İleri teknoloji bilimleri dergisi*, 4(1).
- Gasimov, R. N., Sipahioğlu, A., ve Saraç, T. (2004). 1.5 boyutlu stok malzemesi seçimi problemi için çok ölçütlü bir karar modeli ve çözüm yaklaşımı.
- Gasimov, R. N. (2001). Characterization of the Benson proper efficiency and scalarization in nonconvex vector optimization. In *Multiple criteria decision making in the new millennium* (s. 189-198). Springer, Berlin, Heidelberg.

KAYNAKLAR DİZİNİ (devam)

Gonçalves, J. F., ve Resende, M. G. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*, 246(1), 86-107.

Güzel, N. (2013). A proposal to the solution of multiobjective linear fractional programming problem. In *Abstract and applied analysis*. Hindawi.

Halavati, R., Shouraki, S. B., Noroozian, M., ve Zadeh, S. H. (2005). Optimizing allocation of two dimensional irregular shapes using an agent based approach. *World Academy of Science, Engineering and Technology*, 11.

Hifi, M., ve M'Hallah, R. (2003). A hybrid algorithm for the two-dimensional layout problem: the cases of regular and irregular shapes. *International Transactions in Operational Research*, 10(3), 195-216.

Hougary, S. (2012). A scale invariant algorithm for packing rectangles perfectly. In *Proceedings of the fourth International Workshop on Bin Packing and Placement Constraints*, Nantes, France.

Imam, M. H., ve Mir, M. (1998). Cluster boundary search algorithm for building-block layout optimization. *Advances in Engineering Software*, 29(2), 165-173.

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European journal of operational research*, 88(1), 165-181.

Koç, I. O. (2007). Gezin satıcı problemi için çok popülasyonlu paralel bir genetik algoritma tasarımı, geliştirilmesi ve analizi. Doktora Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Eskişehir.

Korf, R. E., Moffitt, M. D., ve Pollack, M. E. (2010). Optimal rectangle packing. *Annals of Operations Research*, 179(1), 261-295.

Laumanns, M., Thiele, L., ve Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3), 932-942.

Liu, D., ve Teng, H. (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112(2), 413-420.

Lokman, B. (2017). Çok amaçlı tamsayı programlama problemleri için temsili çözüm üreten yaklaşımların ve kalite ölçülerinin incelenmesi. *Journal of Industrial Engineering (Turkish Chamber of Mechanical Engineers)*, 28(1).

Lourenço, H. R., Martin, O. C., ve Stützle, T. (2010). Iterated local search: Framework and applications. In *Handbook of metaheuristics* s.363-397. Springer, Boston, MA.,

KAYNAKLAR DİZİNİ (devam)

- Marler, R. T., ve Arora, J. S. (2005). Function-transformation methods for multi-objective optimization. *Engineering Optimization*, 37(6), 551-570.
- McKendall Jr. A. R., ve Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201(1), 171-182.
- Mir, M., ve Al-Saleh, M. A. (2001). A constructive procedure for optimizing the placement of macrocells. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on (S. 5, s. 7-60)*. IEEE.
- Nikbay, M., Yanangönül, A., ve Öncü, L. (2008). Bir uçak kanadının çok amaçlı genetik algoritmalar ile yapısal optimizasyonu.
- Özcan, E., Kai, Z., ve Drake, J. H. (2013). Bidirectional best-fit heuristic considering compound placement for two dimensional orthogonal rectangular strip packing. *Expert Systems with Applications*, 40(10), 4035-4043.
- Özşahin, M., ve Oral, M. (2010). Genetik Algoritma ile İki Boyutlu Şekil Yerleştirme. *Bilimde Modern Yöntemler Sempozyumu*, 415-432.
- Paes, F. G., Pessoa, A. A., ve Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research*, 256(3), 742-756.
- Serkan, K., ve Fırlı, N. (2017). Çok Amaçlı Optimizasyon Problemlerinde Pareto Optimal Kullanımı. *Sosyal Bilimler Araştırma Dergisi*, 5(2), 9-18.
- Shiangjen, K., Chaijaruwanch, J., Srisujalertwaja, W., Unachak, P., ve Somhom, S. (2018). An iterative bidirectional heuristic placement algorithm for solving the two-dimensional knapsack packing problem. *Engineering Optimization*, 50(2), 347-365.
- Sikaroudi, A. M. E., ve Shahanaghi, K. (2016). Facility layout by collision detection and force exertion heuristics. *Journal of Manufacturing Systems*, 41, 21-30.
- Sipahioğlu, A., ve Saraç, T. (2010). Çok amaçlı sırt çantası probleminin çözümüne yeni bir yaklaşım: konik skalerleştirme.
- Tasadduq, I. A., Imam, M. H., ve Ahmad, A. (2015). A hybrid algorithm for optimising facility layout. *South African Journal of Industrial Engineering*, 26(1), 120-134.
- Türkölmez, G. B. (2016). Yazılım projesi seçim probleminde çok amaçlı optimizasyon uygulaması. *Dokuz Eylül Üniversitesi İşletme Fakültesi Dergisi*, 17(1), 47-63.
- Üstün, Ö. (2007). Çok amaçlı portföy optimizasyon problemi ve çözüm yaklaşımları, Doktora Tezi, Eskişehir Osmangazi Üniversitesi. Fen Bilimleri Enstitüsü, Eskişehir.

KAYNAKLAR DİZİNİ (devam)

Yasar, C., ve Ozyon, S. (2018). A new approach in the solution of the environmental economic power dispatch problems with piecewise quadratic fuel cost function: Conic scalarization method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(2), 527-541.

Yaşar, C. (2018). Parçalı yakıt maliyeti fonksiyonlarına sahip çevresel ekonomik güç dağıtım problemlerinin çözümüne yeni bir yaklaşım. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*.

Yavuz, S., ve Deveci, M. (2012) İstatiksel normalizasyon tekniklerinin yapay sinir ağı performansına etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (40), 167-187.

Yiğit, T., ve Aydemir, M. (2018). Paket döndürmeksizin genetik algoritma kullanarak konteyner yükleme problemi optimizasyonu. *Mühendislik Bilimleri ve Tasarım Dergisi*, 6(1), 21-28.

Yiğit, V., ve Türkbey, O. (2003). Tesis yerleşim problemlerine sezgisel metotlarla yaklaşım. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 18(4).

Zhang, D., Deng, A., ve Kang, Y. (2005). A hybrid heuristic algorithm for the rectangular packing problem. In *International Conference on Computational Science* (pp. 783-791). Springer, Berlin, Heidelberg. , May.

Zitzler, E., ve Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature* (s. 292-301). Springer, Berlin, Heidelberg. September.

EKLER

Ek 1. 30 Bloktan Oluşan Örneğin Genetik Algoritma ve Skalerizasyon Parametreleri

No	Populasyon Büyüklüğü	Çaprazlama Olasılığı	Mutasyon Olasılığı	Turnuva Büyüklüğü %	Max Jenerasyon	Max Tekrar %	Skalarizasyon Yöntemi	α	w1	w2	w3	GA'nın Durdduğu İterasyon Sayısı	Toplam Sıçrama Sayısı	Tamamlanma Süresi	Min F1	Min F2	Min F3	N-F1	N-F2	N-F3	Skaler Sonuç	Orijine uzaklık	Referans noktarsına uzaklık
1	60	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	48	6	5 dk. 35 sn.	3409,7843	0,005339	67	0,3441	0,0339	0,1349	0,0639	0,3711	0,3282
2	60	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	32	2	3 dk. 45 sn.	3482,2166	0,005341	76	0,4719	0,0341	0,2063	0,2815	0,5162	0,4862
3	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	63	8	9 dk. 39 sn.	3408,7934	0,005128	76	0,3423	0,0000	0,2063	0,0746	0,3997	0,3602
4	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	44	3	6 dk. 36 sn.	3475,0272	0,005792	80	0,4592	0,1064	0,2381	0,2972	0,5281	0,4989
5	100	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	88	12	16 dk. 42 sn.	3377,6115	0,005128	75	0,2873	0,0000	0,1984	0,0557	0,3492	0,3032

6	100	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	62	6	11 dk. 45 sn.	3477,6418	0,005353	67	0,4638	0,0360	0,1349	0,2346	0,4844	0,4524
7	120	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	49	4	11 dk. 5 sn.	3388,6020	0,005353	73	0,3067	0,0360	0,1825	0,0676	0,3587	0,3141
8	120	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	48	5	10 dk. 53 sn.	3396,1688	0,005579	78	0,3201	0,0722	0,2222	0,2017	0,3963	0,3564
9	80	0,6	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	100	11	15 dk. 15 sn.	3366,8675	0,005128	89	0,2684	0,0000	0,3095	0,0834	0,4097	0,3712
10	80	0,6	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	33	3	5 dk. 11 sn.	3446,8320	0,005298	82	0,4095	0,0273	0,2540	0,2727	0,4826	0,4504
11	80	0,7	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	53	5	8 dk. 7 sn.	3402,9298	0,005374	71	0,3320	0,0394	0,1667	0,0714	0,3736	0,3310
12	80	0,7	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	79	12	12 dk. 3 sn.	3444,5904	0,005216	78	0,4055	0,0140	0,2222	0,2515	0,4626	0,4290
13	80	0,8	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	85	11	12 dk. 56 sn.	3432,9247	0,005128	79	0,3849	0,0000	0,2302	0,0945	0,4485	0,4137

14	80	0,8	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	100	13	15 dk. 12 sn.	3395,1785	0,005128	73	0,3183	0,0000	0,1825	0,1765	0,3669	0,3235
15	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	51	8	7 dk. 51 sn.	3444,8554	0,005128	72	0,4060	0,0000	0,1746	0,0842	0,4419	0,4066
16	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	62	7	9 dk. 19 sn.	3419,5002	0,005348	81	0,3612	0,0352	0,2460	0,2396	0,4385	0,4028
17	80	0,9	0,003	10	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	56	7	8 dk. 26 sn.	3414,9188	0,005549	71	0,3531	0,0674	0,1667	0,0862	0,3963	0,3564
18	80	0,9	0,003	10	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	100	13	15 dk. 15 sn.	3357,4387	0,005587	82	0,2517	0,0735	0,2540	0,1801	0,3650	0,3213
19	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	75	5	11 dk. 25 sn.	3417,6621	0,005128	76	0,3580	0,0000	0,2063	0,0793	0,4132	0,3751
20	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	41	4	6 dk. 20 sn.	3525,6797	0,005278	69	0,5486	0,0240	0,1508	0,2939	0,5694	0,5425
21	80	0,9	0,003	30	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	64	7	9 dk. 48 sn.	3393,4266	0,005128	77	0,3152	0,0000	0,2143	0,0689	0,3812	0,3395

22	80	0,9	0,003	30	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	80	12	12 dk. 1 sn.	3389,0926	0,005128	78	0,3076	0,0000	0,2222	0,1936	0,3795	0,3376
23	80	0,9	0,003	40	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	59	7	8 dk. 54 sn.	3347,0857	0,005128	82	0,2334	0,0000	0,2540	0,0562	0,3450	0,2983
24	80	0,9	0,003	40	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	56	5	8 dk. 28 sn.	3499,5960	0,005353	78	0,5026	0,0360	0,2222	0,3090	0,5507	0,5227
25	80	0,9	0,003	20	100	10	Ağırlıklı Toplam	-	0,3	0,3	0,3	20	3	3 dk. 3 sn.	3437,6669	0,005128	82	0,3933	0,0000	0,2540	0,1042	0,4682	0,4349
26	80	0,9	0,003	20	100	10	Konik Skalerizasyon	0,290	0,3	0,3	0,3	16	4	2 dk. 31 sn.	3412,0316	0,005505	87	0,2602	0,0603	0,2937	0,2602	0,3970	0,3572
27	80	0,9	0,003	20	100	20	Ağırlıklı Toplam	-	0,3	0,3	0,3	57	7	8 dk. 41 sn.	3417,8955	0,005128	82	0,3584	0,0000	0,2540	0,0937	0,4393	0,4037
28	80	0,9	0,003	20	100	20	Konik Skalerizasyon	0,290	0,3	0,3	0,3	38	4	5 dk. 47 sn.	3478,2291	0,005317	89	0,4649	0,0302	0,3095	0,3382	0,5593	0,5318
29	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,3	0,3	0,3	39	5	6 dk. 0 sn.	3378,7584	0,005128	83	0,2893	0,0000	0,2619	0,0754	0,3903	0,3497

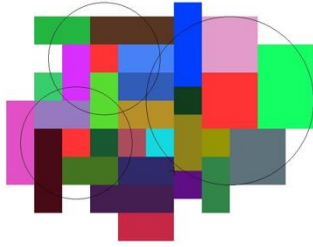
30	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,290	0,3	0,3	0,3	32	3	4 dk. 58 sn.	3427,6383	0,005128	78	0,3756	0,0000	0,2222	0,2337	0,4364	0,4006
31	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,050	0,3	0,3	0,3	57	3	8 dk. 37 sn.	3508,6758	0,005219	75	0,5186	0,0146	0,1984	0,1596	0,5554	0,5277
32	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,075	0,3	0,3	0,3	34	3	5 dk. 11 sn.	3379,6851	0,005313	87	0,2910	0,0297	0,2937	0,1284	0,4145	0,3765
33	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,100	0,3	0,3	0,3	46	7	7 dk. 7 sn.	3441,8462	0,005128	74	0,4007	0,0000	0,1905	0,1365	0,4436	0,4084
34	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,125	0,3	0,3	0,3	52	5	7 dk. 52 sn.	3429,3698	0,005128	76	0,3786	0,0000	0,2063	0,1461	0,4312	0,3949
35	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,150	0,3	0,3	0,3	45	5	6 dk. 53 sn.	3461,6212	0,005278	67	0,4356	0,0240	0,1349	0,1553	0,4566	0,4225
36	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,175	0,3	0,3	0,3	61	11	9 dk. 20 sn.	3411,2428	0,005597	77	0,3467	0,0752	0,2143	0,1683	0,4144	0,3765

37	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,200	0,3	0,3	0,3	36	4	5 dk. 25 sn.	3442,7806	0,005128	74	0,4023	0,0000	0,1905	0,1864	0,4451	0,4100
38	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,225	0,3	0,3	0,3	70	9	10 dk. 31 sn.	3469,2896	0,005128	71	0,4491	0,0000	0,1667	0,2108	0,4790	0,4466
39	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,250	0,3	0,3	0,3	92	10	13 dk. 52 sn.	3330,2049	0,005325	74	0,2037	0,0315	0,1905	0,1034	0,2806	0,2208
40	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,275	0,3	0,3	0,3	37	4	5 dk. 40 sn.	3400,4512	0,005330	101	0,3276	0,0324	0,4048	0,3044	0,5217	0,4922
41	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,45	0,15	0,4	58	5	8 dk. 53 sn.	3418,9862	0,006122	71	0,3603	0,1592	0,1667	0,1527	0,4277	0,3911
42	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,140	0,45	0,15	0,4	34	3	5 dk. 18 sn.	3502,8344	0,005330	65	0,5083	0,0324	0,1190	0,2505	0,5230	0,4935
43	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,4	0,15	0,45	41	7	6 dk. 12 sn.	3350,1003	0,005622	73	0,2388	0,0792	0,1825	0,0895	0,3108	0,2581
44	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,140	0,4	0,15	0,45	39	6	6 dk. 6 sn.	3470,7445	0,005535	65	0,4517	0,0652	0,1190	0,2008	0,4716	0,4387

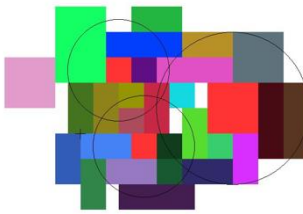
45	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,15	0,45	0,4	100	12	15 dk. 5 sn.	3405,0824	0,005128	73	0,3358	0,0000	0,1825	0,0234	0,3822	0,3407
46	80	0,9	0,003	20	100	30	Konik Skalerezasyon	0,140	0,15	0,45	0,4	80	13	12 dk. 13 sn.	3479,3898	0,005128	68	0,4669	0,0000	0,1429	0,0986	0,4883	0,4565
47	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,45	0,4	0,15	44	6	6 dk. 45 sn.	3331,8116	0,005128	102	0,2065	0,0000	0,4127	0,0548	0,4615	0,4277
48	80	0,9	0,003	20	100	30	Konik Skalerezasyon	0,140	0,45	0,4	0,15	67	13	10 dk. 16 sn.	3269,9756	0,005128	106	0,0974	0,0000	0,4444	0,0731	0,4550	0,4207
49	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	32	2	5 dk. 1 sn.	3414,6790	0,005322	87	0,3527	0,0311	0,2937	0,1969	0,4600	0,4262
50	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	70	7	10 dk. 43 sn.	3362,8940	0,005128	90	0,2613	0,0000	0,3175	0,1577	0,4112	0,3729
51	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	37	5	5 dk. 47 sn.	3350,1389	0,005764	86	0,2388	0,1019	0,2857	0,1439	0,3861	0,3450
52	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	40	6	6 dk. 9 sn.	3376,2433	0,005832	73	0,2849	0,1127	0,1825	0,1267	0,3567	0,3118

53	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	32	2	5 dk. 2 sn.	3399,2067	0,005353	66	0,3254	0,0360	0,1270	0,1171	0,3512	0,3055
54	80	0,9	0,003	20	100	30	Ağırlıklı Toplam	-	0,5	0,1	0,4	41	6	6 dk. 18 sn.	3409,7978	0,005362	82	0,3441	0,0375	0,2540	0,1774	0,4293	0,3928
55	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	35	4	5 dk. 21 sn.	3401,4852	0,005349	77	0,3294	0,0354	0,2143	0,1907	0,3946	0,3545
56	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	100	8	15 dk. 1 sn.	3345,7211	0,005128	79	0,2310	0,0000	0,2302	0,1401	0,3261	0,2763
57	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	63	15	9 dk. 35 sn.	3337,8800	0,005597	77	0,2172	0,0752	0,2143	0,1249	0,3142	0,2622
58	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	60	6	9 dk. 5 sn.	3378,2079	0,005329	85	0,2884	0,0322	0,2778	0,1976	0,4017	0,3624
59	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	43	5	6 dk. 37 sn.	3376,1050	0,005627	88	0,2847	0,0800	0,3016	0,2075	0,4224	0,3852
60	80	0,9	0,003	20	100	30	Konik Skalerizasyon	0,090	0,5	0,1	0,4	33	3	5 dk. 4 sn.	3335,0403	0,005128	104	0,2122	0,0000	0,4286	0,2262	0,4782	0,4458

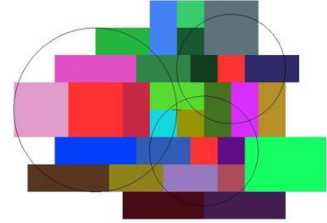
Ek 2. 30 Bloktan Oluşan Örneğin Yerleşim Görünümleri



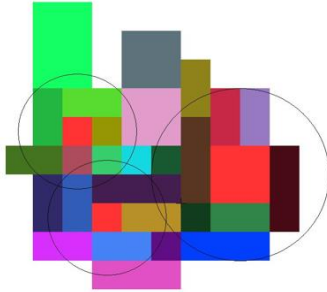
(1)



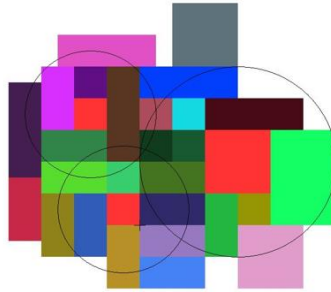
(2)



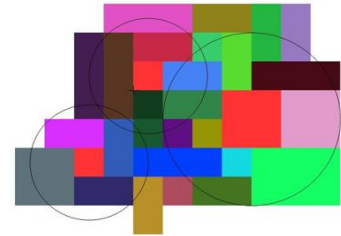
(3)



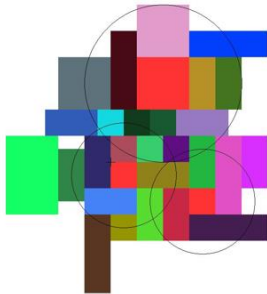
(4)



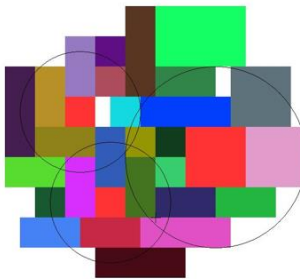
(5)



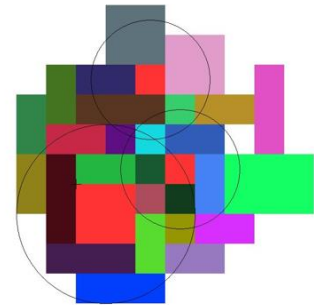
(6)



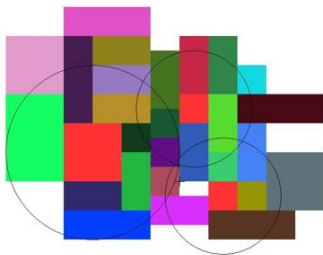
(7)



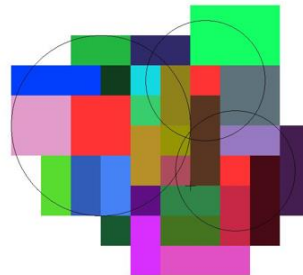
(8)



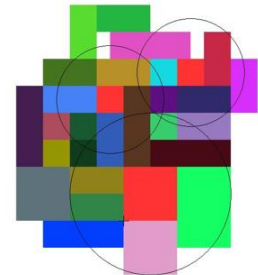
(9)



(10)

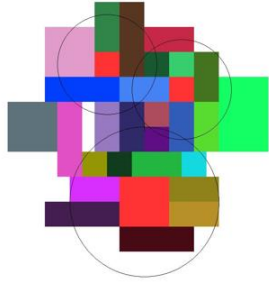


(11)

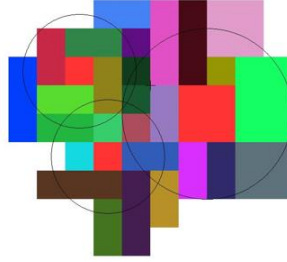


(12)

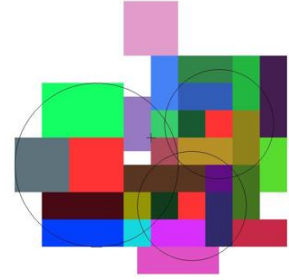
Yerleşim görünümleri (I).



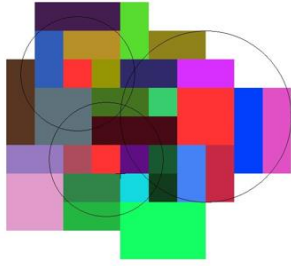
(13)



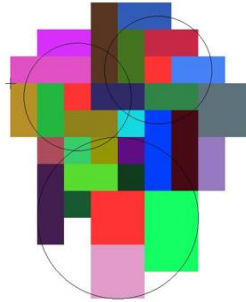
(14)



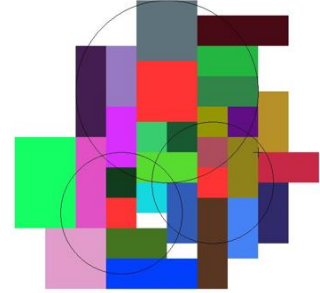
(15)



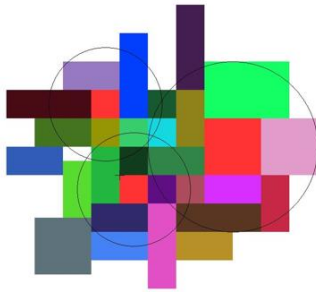
(16)



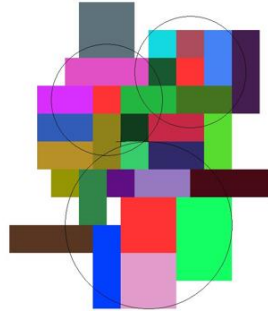
(17)



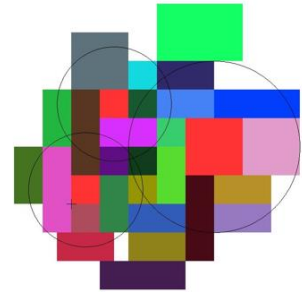
(18)



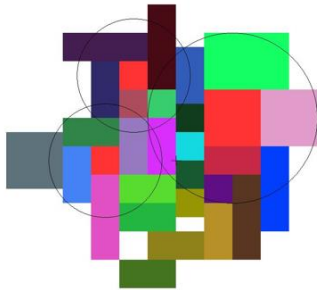
(19)



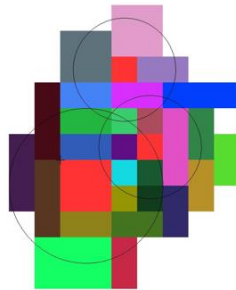
(20)



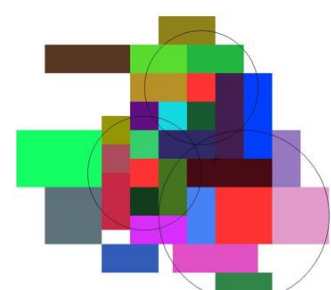
(21)



(22)

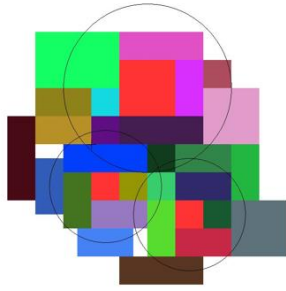


(23)

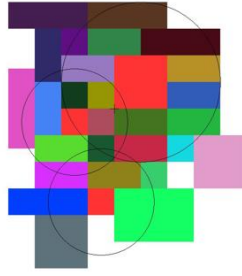


(24)

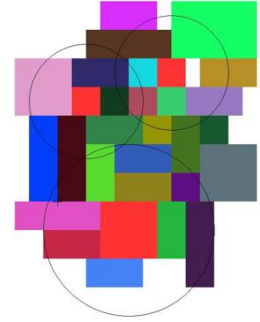
Yerleşim görünümleri (II).



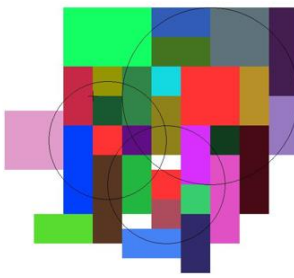
(25)



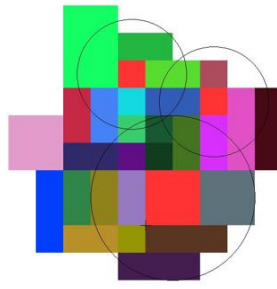
(26)



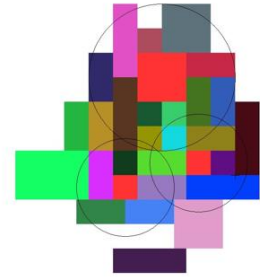
(27)



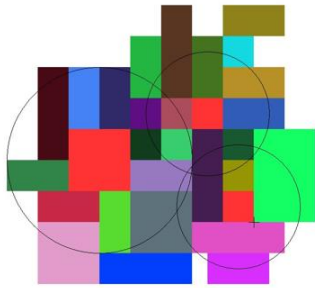
(28)



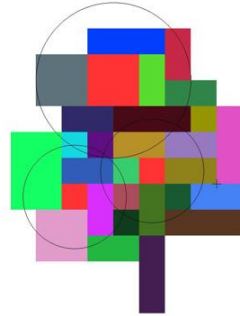
(29)



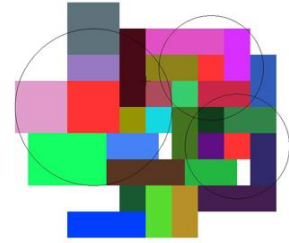
(30)



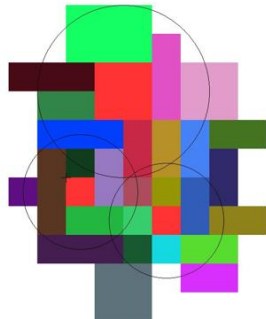
(31)



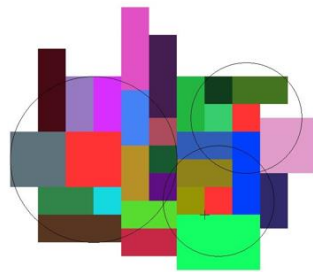
(32)



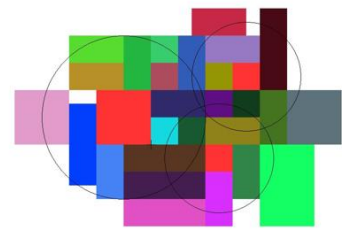
(33)



(34)

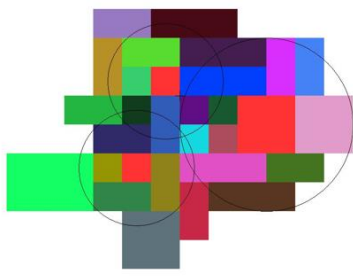


(35)

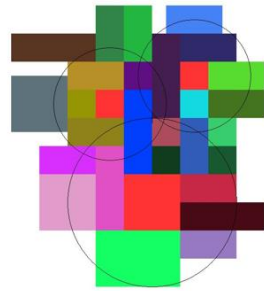


(36)

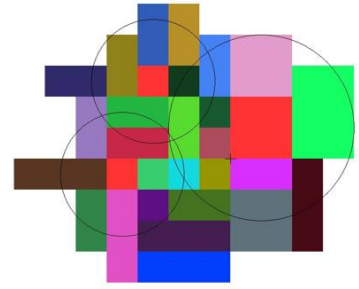
Yerleşim görüntüleri (III).



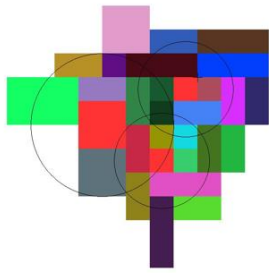
(37)



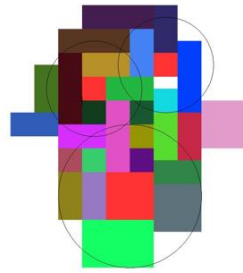
(38)



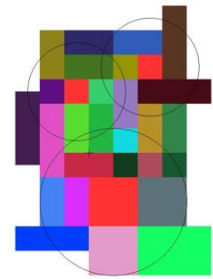
(39)



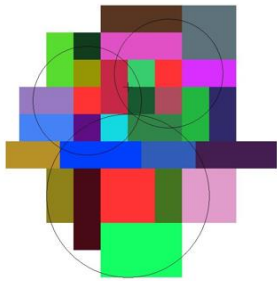
(40)



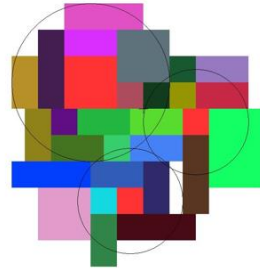
(41)



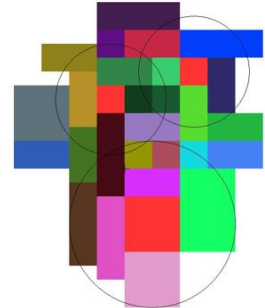
(42)



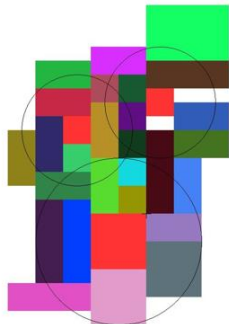
(43)



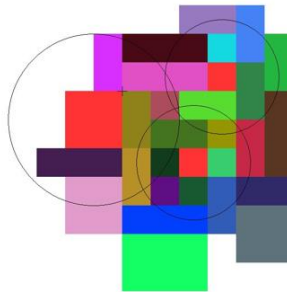
(44)



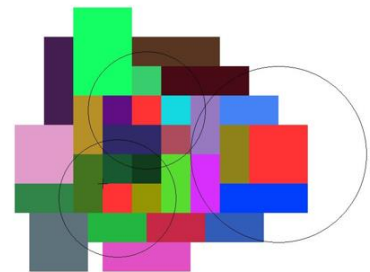
(45)



(46)

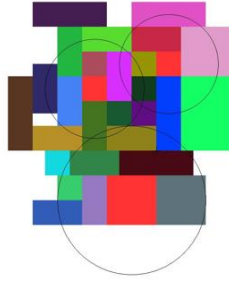


(47)

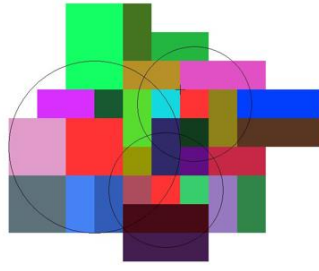


(48)

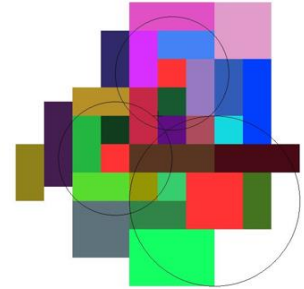
Yerleşim görünümleri (IV).



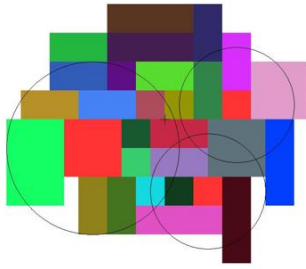
(49)



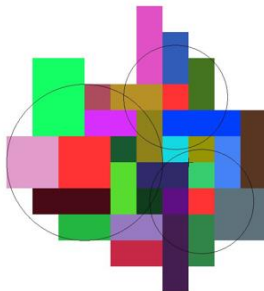
(50)



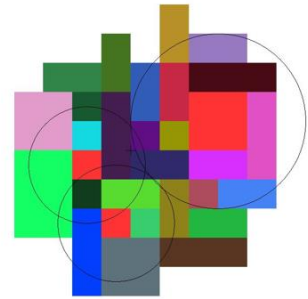
(51)



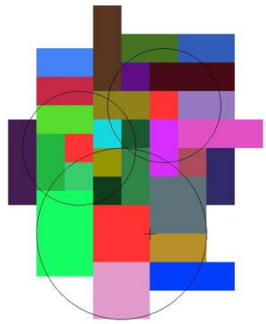
(52)



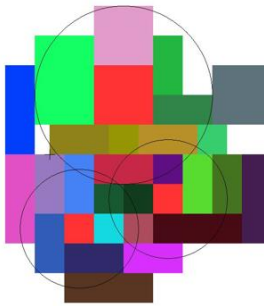
(53)



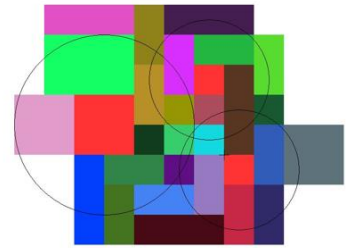
(54)



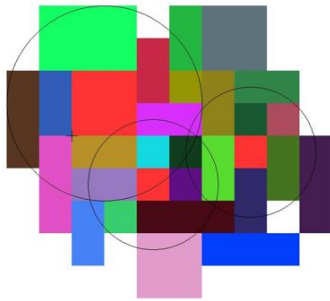
(55)



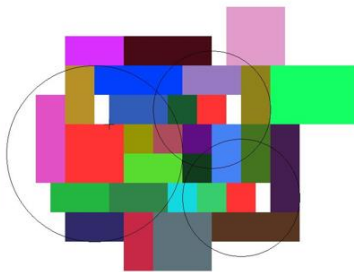
(56)



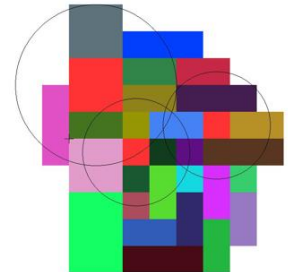
(57)



(58)



(59)



(60)

Yerleşim görünümleri (V).

Ek 3. 8 Örnek Problemin Genetik Algoritma Parametreleri, Skalerizasyon Parametreleri ve Çözüm Sonuçları

No	I1	I2	I3	N1	N2	N3	B1	B2	B3	Skalerizasyon Yöntemi	α	w1	w2	w3	GA'nın Durdduğu İterasyon Sayısı	Toplam Sıçrama Sayısı	Tamamlanma Süresi	Min F1	Min F2	Min F3	Z-F1	Z-F2	Z-F3	Skaler Sonuç	Orijine uzaklık	Referans noktarsma uzaklık
1.1	197,01760	0,000960	56	247,68506	0,001352	66	202,08434	0,000999	57	Ağırlıklı Toplam	-	0,3	0,3	0,3	35	3	15 sn.	205,3146	0,000971	56	0,1638	0,0281	0,0000	0,057555	0,1661	0,0872
1.2	197,01760	0,000960	56	247,68506	0,001352	66	202,08434	0,000999	57	Konik Skalerizasyon	0,290	0,3	0,3	0,3	33	3	14 ns.	207,2315	0,000970	57	0,2016	0,0253	0,1000	0,059190	0,2264	0,1459
1.3	197,01760	0,000960	56	247,68506	0,001352	66	202,08434	0,000999	57	Ağırlıklı Toplam	-	0,25	0,35	0,4	32	2	13 sn.	212,5878	0,000970	56	0,3073	0,0253	0,0000	0,085689	0,3083	0,2551
1.4	197,01760	0,000960	56	247,68506	0,001352	66	202,08434	0,000999	57	Konik Skalerizasyon	0,240	0,25	0,35	0,4	42	5	20 sn.	203,2217	0,000999	56	0,1224	0,0994	0,0000	-0,005062	0,1577	0,0699
2.1	1029,87407	0,000220	60	1322,08216	0,000403	104	1059,09488	0,000238	64	Ağırlıklı Toplam	-	0,3	0,3	0,3	49	7	4 dk. 7 sn.	1140,7681	0,000241	70	0,3795	0,1160	0,2273	0,216833	0,4573	0,4232

3.3	5481,32818	0,000136	124	7963,15845	0,000240	162	5729,51121	0,000146	128	Ağırlıklı Toplam	-	0,25	0,35	0,4	29	4	22 dk. 25 sn.	5905,0066	0,000155	126	0,1707	0,1846	0,0526	0,128344	0,2569	0,1897
3.2	5481,32818	0,000136	124	7963,15845	0,000240	162	5729,51121	0,000146	128	Konik Skalarizasyon	0,290	0,3	0,3	0,3	25	5	20 dk. 12 sn.	6526,8180	0,000146	129	0,4213	0,0934	0,1316	0,208108	0,4511	0,4165
3.1	5481,32818	0,000136	124	7963,15845	0,000240	162	5729,51121	0,000146	128	Ağırlıklı Toplam	-	0,3	0,3	0,3	48	6	29 dk. 41 sn.	5988,9413	0,000146	131	0,2045	0,0944	0,1842	0,144936	0,2910	0,2338
2.4	1029,87407	0,000220	60	1322,08216	0,000403	104	1059,09488	0,000238	64	Konik Skalarizasyon	0,240	0,25	0,35	0,4	37	4	4 dk. 7 sn.	1107,4315	0,000239	71	0,2654	0,1059	0,2500	0,180526	0,3797	0,3379
2.3	1029,87407	0,000220	60	1322,08216	0,000403	104	1059,09488	0,000238	64	Ağırlıklı Toplam	-	0,25	0,35	0,4	38	2	3 dk. 6 sn.	1141,7788	0,000235	64	0,3830	0,0829	0,0909	0,161106	0,4022	0,3630
2.2	1029,87407	0,000220	60	1322,08216	0,000403	104	1059,09488	0,000238	64	Konik Skalarizasyon	0,290	0,3	0,3	0,3	50	5	4 dk. 30 sn.	1124,9083	0,000252	66	0,3252	0,1739	0,1364	0,197933	0,3932	0,3530

5.1	16021,15667	0,000139	184	18650,41478	0,000307	273	16284,08248	0,000156	193	Ağırlıklı Toplam	-	0,3	0,3	0,3	47	12	1 sa. 15 dk. 27 sn.	16445,3521	0,000147	187	0,1613	0,0435	0,0337	0,071578	0,1705	0,1705
4.4	9528,97954	0,000137	84	12073,76929	0,000269	140	9783,45851	0,000150	90	Konik Skalarizasyon	0,240	0,25	0,35	0,4	47	5	25 dk. 14 sn.	10112,5229	0,000158	115	0,2293	0,1599	0,5536	0,388959	0,6201	0,5955
4.3	9528,97954	0,000137	84	12073,76929	0,000269	140	9783,45851	0,000150	90	Ağırlıklı Toplam	-	0,25	0,35	0,4	45	9	23 dk. 25 sn.	10487,2397	0,000151	104	0,3766	0,1047	0,3571	0,273637	0,5294	0,5003
4.2	9528,97954	0,000137	84	12073,76929	0,000269	140	9783,45851	0,000150	90	Konik Skalarizasyon	0,290	0,3	0,3	0,3	42	7	22 dk. 12 sn.	10691,3411	0,000148	110	0,4568	0,0817	0,4643	0,425235	0,6564	0,6331
4.1	9528,97954	0,000137	84	12073,76929	0,000269	140	9783,45851	0,000150	90	Ağırlıklı Toplam	-	0,3	0,3	0,3	78	8	2 sa. 24 dk. 6 sn.	10044,5476	0,000149	97	0,2026	0,0913	0,2321	0,157821	0,3214	0,2707
3.4	5481,32818	0,000136	124	7963,15845	0,000240	162	5729,51121	0,000146	128	Konik Skalarizasyon	0,240	0,25	0,35	0,4	29	6	23 dk. 25 sn.	6560,9414	0,000149	131	0,4350	0,1220	0,1842	0,230999	0,4879	0,4561

6.3	117,71821	117,71821	117,71821	5.2	16021,15667	16021,15667	16021,15667
0,012346	0,012346	0,012346	0,000139	0,000139	0,000139	0,000139	0,000139
4	4	4	184	184	184	184	184
132,75000	132,75000	132,75000	18650,41478	18650,41478	18650,41478	18650,41478	18650,41478
0,013807	0,013807	0,013807	0,000307	0,000307	0,000307	0,000307	0,000307
16	16	16	273	273	273	273	273
119,22139	119,22139	119,22139	16284,08248	16284,08248	16284,08248	16284,08248	16284,08248
0,012492	0,012492	0,012492	0,000156	0,000156	0,000156	0,000156	0,000156
5,2	5,2	5,2	193	193	193	193	193
Ağırlıklı Toplam	Konik Skalarizasyon	Ağırlıklı Toplam	Konik Skalarizasyon	Ağırlıklı Toplam	Konik Skalarizasyon	Konik Skalarizasyon	Konik Skalarizasyon
-	0,290	-	0,240	-	0,290	0,290	0,290
0,25	0,3	0,3	0,25	0,25	0,3	0,3	0,3
0,35	0,3	0,3	0,35	0,35	0,3	0,3	0,3
0,4	0,3	0,3	0,4	0,4	0,3	0,3	0,3
32	32	32	45	55	46	46	46
1	1	1	5	14	11	11	11
7 sn.	7 sn.	7 sn.	1 sa. 22 dk. 12 sn.	1 sa. 32 dk. 16 sn.	1 sa. 17 dk. 02 sn.	1 sa. 17 dk. 02 sn.	1 sa. 17 dk. 02 sn.
119,6588	119,6588	119,6588	18002,7669	16166,8553	16268,4936	16268,4936	16268,4936
0,012346	0,012346	0,012346	0,000150	0,000142	0,000150	0,000150	0,000150
5	5	5	205	187	195	195	195
0,1291	0,1291	0,1291	0,7537	0,0554	0,0941	0,0941	0,0941
0,0000	0,0000	0,0000	0,0605	0,0183	0,0612	0,0612	0,0612
0,0833	0,0833	0,0833	0,2360	0,0337	0,1236	0,1236	0,1236
0,065608	0,016001	0,063729	0,402973	0,033725	0,013474	0,013474	0,013474
0,1537	0,1537	0,1537	0,7921	0,0674	0,1669	0,1669	0,1669
0,0601	0,0601	0,0601	0,7729	0,0674	0,1669	0,1669	0,1669

8.1	1028,55581	1110,24282	1110,24282	1110,24282	1110,24282	1171,84410	1171,84410	1171,84410	1171,84410	1171,84410	1171,84410	119,22139	117,71821	6.4
	0,000225	0,000225	0,000225	0,000225	0,000225	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,013807	0,012346	
	144	244	244	244	244	295	295	295	295	295	295	16	4	
	1291,90718	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	1717,84410	132,75000	132,75000	
	0,000343	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,000427	0,013807	0,013807	
	144	295	295	295	295	295	295	295	295	295	295	16	16	
	1054,89095	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	1171,00295	119,22139	119,22139	
	0,000237	0,000245	0,000245	0,000245	0,000245	0,000245	0,000245	0,000245	0,000245	0,000245	0,000245	0,012492	0,012492	
	144	249,1	249,1	249,1	249,1	249,1	249,1	249,1	249,1	249,1	249,1	5,2	5,2	
	Ağırlıklı Toplam	Konik Skalarizasyon	Konik Skalarizasyon	Ağırlıklı Toplam	Konik Skalarizasyon	Konik Skalarizasyon	Ağırlıklı Toplam	Konik Skalarizasyon	Konik Skalarizasyon	Ağırlıklı Toplam	Konik Skalarizasyon	Konik Skalarizasyon	Konik Skalarizasyon	
	-	0,240	0,25	0,35	0,4	-	0,290	0,3	0,3	0,3	0,3	0,240	0,25	
	0,5	0,25	0,35	0,4	0,4	0,25	0,3	0,3	0,3	0,3	0,3	0,25	0,35	
	0,5	0,35	0,4	0,4	0,4	0,35	0,3	0,3	0,3	0,3	0,3	0,35	0,4	
	0	0,4	0,4	0,4	0,4	0,4	0,3	0,3	0,3	0,3	0,3	0,4	0,4	
	41	39	39	39	39	54	48	48	48	48	48	32	32	
	5	8	8	13	13	13	7	7	7	7	7	1	1	
	3 dk. 5 sn.	4 dk. 3 sn.	7 dk. 3 sn.	7 dk. 3 sn.	7 dk. 3 sn.	7 dk. 3 sn.	7 dk. 46 sn.	7 dk. 46 sn.	7 dk. 46 sn.	7 dk. 46 sn.	7 dk. 46 sn.	7 sn.	7 sn.	
	1091,0620	1446,0605	1510,5903	1510,5903	1510,5903	1510,5903	1528,1145	1528,1145	1528,1145	1528,1145	1528,1145	119,6588	119,6588	
	0,000245	0,000242	0,000255	0,000255	0,000255	0,000255	0,000245	0,000245	0,000245	0,000245	0,000245	0,012346	0,012346	
	144	272	246	246	246	246	248	248	248	248	248	5	5	
	0,2373	0,5527	0,6589	0,6589	0,6589	0,6589	0,6877	0,6877	0,6877	0,6877	0,5013	0,1291	0,1291	
	0,1643	0,0864	0,1486	0,1486	0,1486	0,1486	0,1018	0,1018	0,1018	0,1018	0,1256	0,0000	0,0000	
	0,0000	0,5490	0,0392	0,0392	0,0392	0,0392	0,0784	0,0784	0,0784	0,0784	0,3333	0,0833	0,0833	
	0,200817	0,507698	0,232415	0,232415	0,232415	0,232415	0,347634	0,347634	0,347634	0,347634	0,288071	0,000592	0,000592	
	0,2887	0,7838	0,6766	0,6766	0,6766	0,6766	0,6996	0,6996	0,6996	0,6996	0,6149	0,1537	0,1537	
	0,2309	0,7644	0,6540	0,6540	0,6540	0,6540	0,6779	0,6779	0,6779	0,6779	0,5901	0,0601	0,0601	

8.1	1028,55581	0,000225	144	1291,90718	0,000343	144	1054,89095	0,000237	144	Konik Skalarizasyon	0,490	0,5	0,5	0	41	5	3 dk. 13 sn.	1073,5886	0,000245	144	0,1710	0,1636	0,0000	0,000000	0,2366	0,1613
8.3	1028,55581	0,000225	144	1291,90718	0,000343	144	1054,89095	0,000237	144	Ağırlıklı Toplam	-	0,45	0,55	0	46	7	3 dk. 31 sn.	1080,8698	0,000243	144	0,1986	0,1524	0,0000	0,173199	0,2504	0,1808
8.4	1028,55581	0,000225	144	1291,90718	0,000343	144	1054,89095	0,000237	144	Konik Skalarizasyon	0,440	0,45	0,55	0	65	5	6 dk. 5 sn.	1117,0351	0,000228	144	0,3360	0,0196	0,0000	0,000000	0,3365	0,2886

Ek 4. 8 Örnek Problemin İlişki Değerleri

1 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	75	74	25	51	91	0	0
2	0	0	0	0	20	68	0	0	0	10
3	0	0	0	49	0	0	0	86	0	0
4	75	0	49	0	0	17	45	92	0	0
5	74	20	0	0	0	0	60	0	77	0
6	25	68	0	17	0	0	0	16	0	0
7	51	0	0	45	60	0	0	0	68	0
8	91	0	86	92	0	16	0	0	69	48
9	0	0	0	0	77	0	68	69	0	0
10	0	10	0	0	0	0	0	48	0	0

2 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	46	19	58	31	35	95	0	0	44	0	0	94	0	0	0	13	0	31
2	0	0	0	47	74	26	46	0	25	0	33	33	19	84	0	0	54	0	0	31
3	46	0	0	0	0	15	33	32	12	30	95	0	0	0	45	15	47	0	52	
4	19	47	0	0	21	0	0	85	0	0	0	57	43	29	0	0	44	0	0	
5	58	74	0	21	0	0	0	0	0	0	17	66	0	53	0	69	0	0	75	
6	31	26	0	0	0	15	0	63	0	44	0	84	0	0	0	64	0	86	18	
7	35	46	15	0	0	15	0	35	65	0	0	83	35	74	0	71	0	0	70	42
8	95	0	33	85	0	0	35	0	0	21	0	51	0	98	35	80	0	0	30	0
9	0	25	32	0	0	63	65	0	0	94	0	0	12	86	59	12	0	20	38	51
10	0	0	12	0	0	0	0	21	94	0	0	13	84	90	0	96	0	0	0	0
11	44	33	30	0	0	44	0	0	0	0	0	0	33	0	29	0	72	21	77	0
12	0	33	95	57	17	0	83	51	0	13	0	0	0	48	75	0	73	0	0	0
13	0	19	0	43	66	84	35	0	12	84	33	0	0	36	49	0	51	85	0	0
14	94	84	0	29	0	0	74	98	86	90	0	48	36	0	12	0	80	0	0	50
15	0	0	0	0	53	0	0	35	59	0	29	75	49	12	0	0	0	0	10	64
16	0	0	45	0	0	0	71	80	12	96	0	0	0	0	0	0	0	0	0	0
17	0	54	15	0	69	64	0	0	0	0	72	73	51	80	0	0	0	0	69	0
18	13	0	47	44	0	0	0	0	20	0	21	0	85	0	0	0	0	0	0	65
19	0	0	0	0	0	86	70	30	38	0	77	0	0	0	10	0	69	0	0	88
20	31	31	52	0	75	18	42	0	51	0	0	0	0	50	64	0	0	65	88	0

3 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	0	0	87	19	54	0	0	0	0	0	45	0	0	76	97	0	0	0	0	0	47	95	0	65	2	0	0	0	58	0	0	0	0	0	0	10	0	32	0	0
2	0	0	0	0	0	87	0	0	0	0	0	0	43	0	17	0	0	0	0	0	64	0	0	66	0	0	94	0	0	0	0	0	0	0	78	0	0	0	0	
3	87	0	0	0	0	0	66	0	1	0	0	0	0	7	59	0	9	0	0	0	62	0	62	0	0	0	0	0	0	0	0	62	0	0	0	0	0	44	61	0
4	19	0	0	0	0	91	0	18	0	0	0	0	0	0	0	41	0	0	0	0	74	0	0	0	96	0	91	0	0	0	0	0	0	44	76	34	0	12		
5	54	0	0	0	0	91	0	0	0	0	66	0	0	84	0	0	0	0	30	0	0	0	0	25	0	0	0	0	2	0	0	0	0	11	0	0	0	0	26	
6	0	87	0	91	91	0	0	0	77	0	0	0	0	0	54	83	0	0	0	74	0	0	0	84	0	0	0	0	0	0	0	46	0	0	65	0	0	90		
7	0	0	66	0	0	0	0	0	0	57	0	63	0	0	53	0	0	0	0	0	69	55	0	0	0	0	0	46	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	18	0	0	0	0	18	8	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	31	0	0	17	0	0	0	0	0	0	0	0	60	0	0
9	0	0	1	0	0	77	0	18	0	92	0	0	0	0	33	0	0	0	0	0	76	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	28	
10	0	0	0	0	0	57	8	92	0	0	0	54	0	0	0	0	48	5	0	69	96	0	0	0	0	0	0	61	0	71	0	0	0	0	66	0	90	0	0	
11	45	0	0	0	66	0	0	0	0	0	0	0	0	0	28	0	0	66	0	13	0	0	0	0	0	0	0	0	0	0	96	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	63	0	0	0	0	0	0	57	0	63	0	23	0	61	0	0	0	0	49	0	0	2	0	15	0	0	0	0	0	0	0	0	0	0	
13	0	43	0	0	0	0	0	0	0	54	0	0	0	0	3	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	0	0	
14	76	0	7	0	84	0	0	0	0	0	57	0	0	0	0	0	42	0	0	0	61	0	0	25	0	30	0	16	0	0	0	32	0	0	0	0	0	80	0	0
15	97	17	59	0	0	0	53	0	0	0	0	0	0	0	99	0	0	44	0	23	0	0	0	0	0	10	0	3	0	6	50	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	54	0	0	33	0	28	63	3	0	99	0	46	0	0	47	66	0	0	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	9	41	0	83	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	2	0	32	0	74	0	0	0	0	0	0	0	0	0	0	70	0	
18	0	0	0	0	0	0	0	0	0	48	0	23	83	42	0	0	0	0	0	0	0	0	0	71	0	0	28	0	0	0	84	0	0	2	0	0	0	0	0	
19	0	0	0	0	30	0	0	0	0	5	66	0	0	0	44	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	18	16	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	61	0	0	0	47	0	0	0	0	0	59	0	0	0	0	0	35	0	27	0	78	0	0	0	0	0	87	0	
21	47	0	62	0	0	74	69	0	0	69	13	0	0	0	23	66	0	0	0	0	0	0	30	58	0	0	9	23	0	0	0	0	19	0	0	44	7	0	0	
22	95	64	0	0	0	0	55	0	0	96	0	0	0	61	0	0	0	0	0	59	0	0	0	0	0	0	0	83	0	22	0	0	32	0	0	12	0	49	0	

23	0	0	62	74	0	0	0	0	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	0	0	0									
24	65	0	0	0	0	0	0	91	0	0	0	0	0	0	85	2	0	35	0	30	0	0	0	0	0	94	33	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
25	2	66	0	0	25	0	0	0	0	0	0	0	0	25	0	0	0	71	0	0	58	0	37	0	0	52	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0							
26	0	0	0	0	0	84	0	0	0	0	0	0	0	0	32	0	0	0	0	0	31	0	52	0	94	0	0	22	0	0	0	0	0	0	0	67	0	0	0	0	33	10	0	0	0						
27	0	0	0	96	0	0	0	31	0	0	0	0	0	30	10	0	0	0	0	0	0	0	0	94	0	94	0	61	0	0	0	0	9	0	0	0	0	0	0	0	0	42	0	0	0	0					
28	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	28	0	0	9	0	0	33	0	0	61	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
29	58	0	0	91	0	0	46	0	0	61	0	2	0	16	3	0	0	0	0	35	23	83	0	0	0	0	0	3	0	0	31	0	0	0	0	0	0	0	0	29	0	0	0	0							
30	0	0	0	0	2	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	0	22	0	0	0	0	0	40	30	0	0	0	0	0	0	0	32	74	8	0	0	0					
31	0	0	0	0	0	0	0	0	0	71	0	15	0	0	6	0	0	0	0	27	0	22	0	0	0	0	0	0	31	40	0	0	0	0	57	0	33	87	0	0	0	0	0	0	0	0					
32	0	0	0	0	0	0	0	0	0	96	0	0	0	0	50	0	0	84	0	0	0	0	0	0	3	0	0	0	0	30	0	0	0	0	49	0	0	0	0	0	0	0	0	0	0	0	0				
33	0	0	62	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	18	78	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	12	0	0	0				
34	0	0	0	0	0	46	0	0	1	0	0	0	0	0	0	0	0	16	0	19	32	0	0	0	0	0	0	0	0	0	57	49	0	0	0	0	1	0	0	3	0	0	0	0	0						
35	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	67	0	0	0	0	0	0	0	0	0	0	0	0	0	44	3	0	74	0	0	0	0	0				
36	10	78	0	44	0	0	0	0	0	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	40	1	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
37	0	0	0	76	0	65	0	0	0	0	0	0	87	0	0	0	0	0	0	44	12	65	0	0	0	42	0	29	32	87	0	0	0	0	3	0	0	0	0	0	95	51	0	0	0	0	0				
38	32	0	44	34	0	0	0	60	0	90	0	0	0	80	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
39	0	0	61	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	87	0	49	0	0	0	33	0	0	0	8	0	0	0	3	74	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	
40	0	0	0	12	26	90	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	51	5	0	0	0	0	0	0	0	0	

5 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	87	19	54	0	0	0	0	0	45	0	0	76	97	0	0	0	0	0	47	95	0	65	2	0	0	0	58	0
2	0	0	0	0	0	87	0	0	0	0	0	0	43	0	17	0	0	0	0	0	0	64	0	0	66	0	0	94	0	0
3	87	0	0	0	0	0	66	0	1	0	0	0	0	7	59	0	9	0	0	0	62	0	62	0	0	0	0	0	0	0
4	19	0	0	0	0	91	0	18	0	0	0	0	0	0	0	41	0	0	0	0	0	74	0	0	0	96	0	91	0	0
5	54	0	0	0	0	91	0	0	0	0	66	0	0	84	0	0	0	0	30	0	0	0	0	0	25	0	0	0	0	2
6	0	87	0	91	91	0	0	0	77	0	0	0	0	0	0	54	83	0	0	0	74	0	0	0	0	84	0	0	0	0
7	0	0	66	0	0	0	0	0	0	57	0	63	0	0	53	0	0	0	0	0	69	55	0	0	0	0	0	0	46	0
8	0	0	0	18	0	0	0	0	18	8	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	31	0	0	17
9	0	0	1	0	0	77	0	18	0	92	0	0	0	0	0	33	0	0	0	0	0	0	76	0	0	0	0	0	0	0
10	0	0	0	0	0	0	57	8	92	0	0	0	54	0	0	0	0	48	5	0	69	96	0	0	0	0	0	0	61	0
11	45	0	0	0	66	0	0	0	0	0	0	0	0	0	0	28	0	0	66	0	13	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	63	0	0	0	0	0	0	57	0	63	0	23	0	61	0	0	0	0	0	49	0	0	2	0
13	0	43	0	0	0	0	0	0	0	54	0	0	0	0	0	3	0	83	0	0	0	0	0	0	0	0	0	0	0	0
14	76	0	7	0	84	0	0	0	0	0	0	57	0	0	0	0	42	0	0	0	61	0	0	25	0	30	0	16	0	0
15	97	17	59	0	0	0	53	0	0	0	0	0	0	0	0	99	0	0	44	0	23	0	0	0	0	0	10	0	3	0
16	0	0	0	0	0	54	0	0	33	0	28	63	3	0	99	0	46	0	0	47	66	0	0	85	0	0	0	0	0	0
17	0	0	9	41	0	83	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	2	0	32	0	74	0	0
18	0	0	0	0	0	0	0	0	0	48	0	23	83	42	0	0	0	0	0	0	0	0	0	0	71	0	0	28	0	0
19	0	0	0	0	30	0	0	0	0	5	66	0	0	0	44	0	0	0	0	0	0	0	0	35	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	61	0	0	0	47	0	0	0	0	0	0	59	0	0	0	0	0	35	0
21	47	0	62	0	0	74	69	0	0	69	13	0	0	0	23	66	0	0	0	0	0	0	0	30	58	0	0	9	23	0
22	95	64	0	0	0	0	55	0	0	96	0	0	0	61	0	0	0	0	0	59	0	0	0	0	0	0	0	0	83	0
23	0	0	62	74	0	0	0	0	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	31	0	0	0	0
24	65	0	0	0	0	0	0	91	0	0	0	0	0	0	0	85	2	0	35	0	30	0	0	0	0	0	94	33	0	99
25	2	66	0	0	25	0	0	0	0	0	0	0	0	25	0	0	0	71	0	0	58	0	37	0	0	52	0	0	0	0
26	0	0	0	0	0	84	0	0	0	0	0	49	0	0	0	32	0	0	0	0	0	31	0	52	0	94	0	0	22	0
27	0	0	0	96	0	0	0	31	0	0	0	0	0	30	10	0	0	0	0	0	0	0	0	94	0	94	0	61	0	0
28	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	74	28	0	0	9	0	0	33	0	0	61	0	3	0	0
29	58	0	0	91	0	0	46	0	0	61	0	2	0	16	3	0	0	0	0	35	23	83	0	0	0	0	0	3	0	0
30	0	0	0	0	2	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	0	22	0	0	0	0
31	0	0	0	0	0	0	0	0	0	71	0	15	0	0	6	0	0	0	0	27	0	22	0	0	0	0	0	31	40	0
32	0	0	0	0	0	0	0	0	0	0	96	0	0	0	50	0	0	84	0	0	0	0	0	3	0	0	0	0	30	0
33	0	0	62	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	18	78	0	0	0	0	0	9	0	0	0	0
34	0	0	0	0	0	46	0	0	1	0	0	0	0	0	0	0	0	0	16	0	19	32	0	0	0	0	0	0	0	0
35	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	67	0	0	0	0

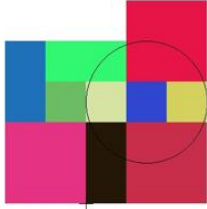
7 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	46	19	58	31	35	95	0	0	44	0	0	94	0	0	0	13	0	31
2	0	0	0	47	74	26	46	0	25	0	33	33	19	84	0	0	54	0	0	31
3	46	0	0	0	0	0	15	33	32	12	30	95	0	0	0	45	15	47	0	52
4	19	47	0	0	21	0	0	85	0	0	0	57	43	29	0	0	0	44	0	0
5	58	74	0	21	0	0	0	0	0	0	0	17	66	0	53	0	69	0	0	75
6	31	26	0	0	0	0	15	0	63	0	44	0	84	0	0	0	64	0	86	18
7	35	46	15	0	0	15	0	35	65	0	0	83	35	74	0	71	0	0	70	42
8	95	0	33	85	0	0	35	0	0	21	0	51	0	98	35	80	0	0	100	0
9	0	25	32	0	0	63	65	0	0	94	0	0	12	86	59	12	0	20	38	51
10	0	0	12	0	0	0	0	21	94	0	0	13	84	90	0	96	0	0	0	0
11	44	33	30	0	0	44	0	0	0	0	0	0	33	0	29	0	72	21	77	0
12	0	33	95	57	17	0	83	51	0	13	0	0	0	48	75	0	73	0	0	0
13	0	19	0	43	66	84	35	0	12	84	33	0	0	36	49	0	51	85	0	0
14	94	84	0	29	0	0	74	98	86	90	0	48	36	0	12	0	80	0	0	50
15	0	0	0	0	53	0	0	35	59	0	29	75	49	12	0	0	0	0	10	64
16	0	0	45	0	0	0	71	80	12	96	0	0	0	0	0	0	0	0	0	0
17	0	54	15	0	69	64	0	0	0	0	72	73	51	80	0	0	0	0	69	0
18	13	0	47	44	0	0	0	0	20	0	21	0	85	0	0	0	0	0	0	65
19	0	0	0	0	0	86	70	100	38	0	77	0	0	0	10	0	69	0	0	88
20	31	31	52	0	75	18	42	0	51	0	0	0	0	50	64	0	0	65	88	0

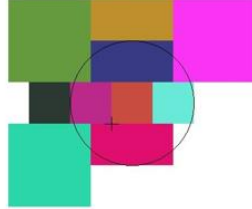
8 numaralı örneğin ilişki değerleri.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	46	19	58	31	35	95	0	0	44	0	0	94	0	0	0	13	0	31
2	0	0	0	47	74	26	46	0	25	0	33	33	19	84	0	0	54	0	0	31
3	46	0	0	0	0	0	15	33	32	12	30	95	0	0	0	45	15	47	0	52
4	19	47	0	0	21	0	0	85	0	0	0	57	43	29	0	0	0	44	0	0
5	58	74	0	21	0	0	0	0	0	0	0	17	66	0	53	0	69	0	0	75
6	31	26	0	0	0	0	15	0	63	0	44	0	84	0	0	0	64	0	86	18
7	35	46	15	0	0	15	0	35	65	0	0	83	35	74	0	71	0	0	70	42
8	95	0	33	85	0	0	35	0	0	21	0	51	0	98	35	80	0	0	100	0
9	0	25	32	0	0	63	65	0	0	94	0	0	12	86	59	12	0	20	38	51
10	0	0	12	0	0	0	0	21	94	0	0	13	84	90	0	96	0	0	0	0
11	44	33	30	0	0	44	0	0	0	0	0	0	33	0	29	0	72	21	77	0
12	0	33	95	57	17	0	83	51	0	13	0	0	0	48	75	0	73	0	0	0
13	0	19	0	43	66	84	35	0	12	84	33	0	0	36	49	0	51	85	0	0
14	94	84	0	29	0	0	74	98	86	90	0	48	36	0	12	0	80	0	0	50
15	0	0	0	0	53	0	0	35	59	0	29	75	49	12	0	0	0	0	10	64
16	0	0	45	0	0	0	71	80	12	96	0	0	0	0	0	0	0	0	0	0
17	0	54	15	0	69	64	0	0	0	0	72	73	51	80	0	0	0	0	69	0
18	13	0	47	44	0	0	0	0	20	0	21	0	85	0	0	0	0	0	0	65
19	0	0	0	0	0	86	70	100	38	0	77	0	0	0	10	0	69	0	0	88
20	31	31	52	0	75	18	42	0	51	0	0	0	0	50	64	0	0	65	88	0

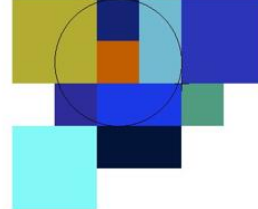
Ek 5. 8 Örnek Problemin Yerleşim Görünümleri



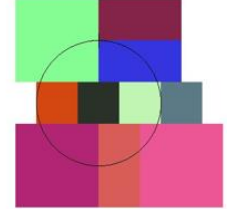
(1.1)



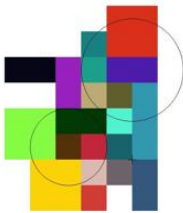
(1.2)



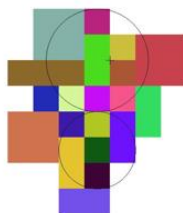
(1.3)



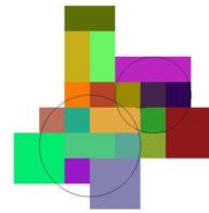
(1.4)



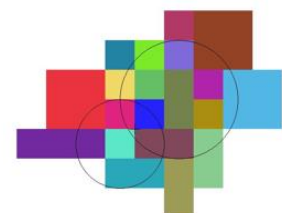
(2.1)



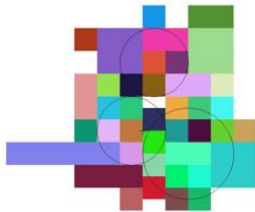
(2.2)



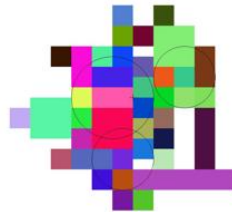
(2.3)



(2.4)



(3.1)



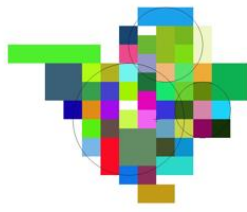
(3.2)



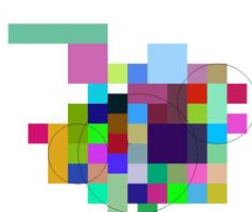
(3.3)



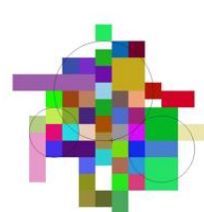
(3.4)



(4.1)



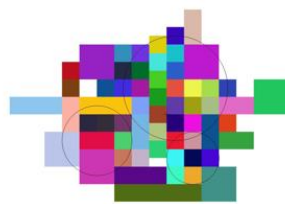
(4.2)



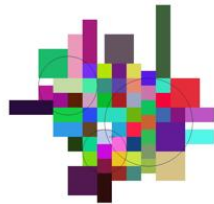
(4.3)



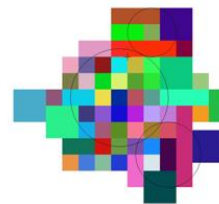
(4.4)



(5.1)



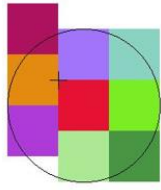
(5.2)



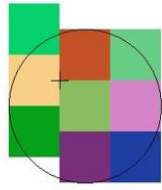
(5.3)



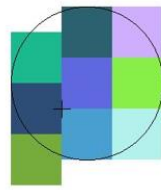
(5.4)



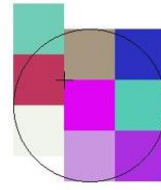
(6.1)



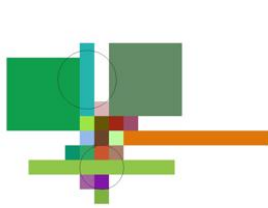
(6.2)



(6.3)



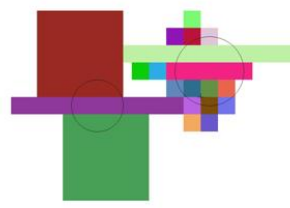
(6.4)



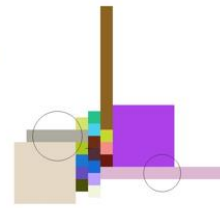
(7.1)



(7.2)



(7.3)



(7.4)



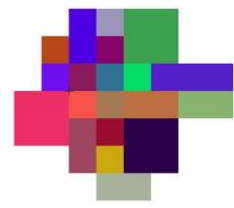
(8.1)



(8.2)



(8.3)



(8.4)