

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GENETİK ALGORİTMA İLE TESİS YERLEŞİMİ
TASARIMI VE BİR UYGULAMA**

End. Müh. Cem DUMAN

**FBE Endüstri Mühendisliği Anabilim Dalı Sistem Mühendisliği Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Bahadır GÜLSÜN

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
2. TESİS YERLEŞİMİ	2
2.1 Giriş	2
2.2 Tesis Yerleşim Düzenlemenin Temelleri	2
2.2.1 Tesis yerleşim tipleri	3
2.2.2 Tesis yerleşimi düzenlemede amaçlar	5
2.2.3 Tesis yerleşimi düzenlemede ilkeler	7
2.2.4 Tesis yerleşimi düzenleme çalışması aşamaları	8
2.2.5 Tesis yerleşim sorunlarının sınıflandırılması	10
2.2.6 Tesis yerleşim etmenleri	10
2.3 Tesis Yerleşimi Yöntemleri	11
2.3.1 Nicel çözümleme modelleri ve teknikler	12
2.3.1.1 Grafik metodu	12
2.3.1.2 Gezi şeması	12
2.3.1.3 Atama yöntemi	12
2.3.1.4 Wimmert yöntemi	13
2.3.1.5 Spiral analiz yöntemi	13
2.3.1.6 Doğrusal hat yerleştirme yöntemi	13
2.3.1.7 Hat dengeleme	13
2.3.1.8 Noy metodu	13
2.3.1.9 Yükleme yolu metodu (Load path method)	14
2.3.1.10 Dal-sınır algoritması (Branch and bound algorithm)	14
2.3.2 Bilgisayar destekli algoritmalar	14
2.3.2.1 Kuruluş veya yapı algoritmaları	15
2.3.2.2 Geliştirme algoritmaları	17
2.3.3 Sezgisel metodlar	19
2.3.3.1 Tepe tırmanma metodu (Hill climbing)	19
2.3.3.2 Tavlama benzetimi metodu (Simulated annealing)	20
2.3.3.3 Tabu araştırması (Tabu search)	20

2.3.3.4	Yapay sinir ağları (Artificial neural network)	21
2.3.3.5	Genetik algoritmalar (GA-Genetic algorithms).....	21
2.3.3.6	Karınca algoritmaları (Ant colony algortihms)	21
2.3.3.7	Melez yöntemler	22
2.3.4	Kareli atama problemi (KAP).....	23
3.	GENETİK ALGORİTMA (GA).....	26
3.1	Genetik Algoritma Nedir?	26
3.2	Genetik Algoritma'nın Tarihsel Gelişimi	29
3.3	Neden GA ? Neden Değil?	30
3.4	Genetik Algoritmanın Çalışma Prensipleri.....	31
3.4.1	Temel kavramlar	32
3.4.1.1	Gen.....	32
3.4.1.2	Kromozom	33
3.4.1.3	Popülasyon (Yığın, kütle).....	33
3.4.1.4	Yeniden üretim işlemi.....	33
3.4.1.5	Başlangıç yığınının oluşturulması	33
3.4.1.6	Uygunluk değeri	34
3.4.1.7	Genetik operatörlerin uygulanacağı dizilerin seçilmesi	35
3.4.2	Kodlama (Dizi gösterimi).....	35
3.4.2.1	İkili kodlama (Binary encoding).....	36
3.4.2.2	Permütasyonlu kodlama (Permutation encoding).....	36
3.4.3	Başlangıç popülasyonunun oluşturulması	36
3.4.4	Uyum fonksiyonu	37
3.4.5	Yeniden üretim (Seçim).....	38
3.4.5.1	Orantılı yeniden üretim mekanizmaları	40
3.4.5.2	Sıralı seçim mekanizması	42
3.4.5.3	Turnuva seçim mekanizması	42
3.4.5.4	Denge durumu seçim mekanizması	42
3.4.6	Genetik operatörler	43
3.4.6.1	Çaprazlama operatörü	43
3.4.6.2	Mutasyon operatörü	44
3.5	GA'da Şema Teoremi	46
3.6	Kontrol Parametreleri	47
3.7	Genetik Algoritmaların Performansını Etkileyen Etmenler	48
3.8	Genetik Algoritma Uygulamaları	49
4.	TESİS YERLEŞİMİ PROBLEMLERİNDE GENETİK ALGORİTMALAR	52
4.1	Problemin Tanımlanması ve Tesis Yerleşiminin Gösterimi.....	52
4.2	Yerleşim Planının Kodlanması	56
4.3	Tesislerin Geometrik Kısıtları	59
4.4	Genetik Parametre Değerlerinin Hesaplanması.....	60
4.4.1	Başlangıç popülasyonu ve popülasyon büyüklüğü.....	60
4.4.2	Çaprazlama ve mutasyon.....	61
4.5	Değerlendirme ve Seçim.....	65
5.	UYGULAMA: GENETİK ALGORİTMA KULLANARAK YAPISAL ELEKTRİK MALZEMELERİ ÜRETEN BİR FİRMA İÇİN YERLEŞİM TASARIMI YAPILMASI	66
5.1	Tesis Yerleşimi İçin Kareli Atama Problemine Uygun Bir Genetik Algoritma	

	Tasarımı	66
5.1.1	Problemin tanımı ve varsayımlar	66
5.1.2	Temel yapı ve işleyiş	67
5.1.2.1	Algoritma akış diyagramı	68
5.1.2.2	Kodlama.....	69
5.1.2.3	Uyum fonksiyonu	69
5.1.2.4	Başlangıç popülasyonu	69
5.1.2.5	Elitizm.....	69
5.1.2.6	Seçim mekanizması	70
5.1.2.7	Çaprazlama yöntemi	70
5.1.2.8	Mutasyon	71
5.1.2.9	Nesil sayısı.....	71
5.2	Programın Tanıtılması	71
5.2.1	Problemin programa girilmesi	72
5.2.2	Parametreler	73
5.2.3	Programın çalıştırılması.....	75
5.2.4	Rapor formatı ve yorumlanması	75
5.2.5	Doğrulama	76
5.3	Literatür Problemleri ile Programın Test Edilmesi	77
5.3.1	Belirlenen literatür problemleri	77
5.3.2	Bulunan parametre ve yerleşimler	77
5.4	Yapısal Elektrik Sektöründe Programın Kullanılması	79
5.4.1	Sektör ve firma tanıtımı	79
5.4.2	Firmada yaşanan problem ve modelleme varsayımları	80
5.4.3	Modelleme aşamaları.....	81
5.4.4	Mevcut durum analizi	82
5.4.5	Oluşturulan yeni durum	83
6.	SONUÇLAR ve ÖNERİLER	86
	KAYNAKLAR	88
	EKLER	91
	Ek 1 Literatür problemleri için bulunan en iyi değerler ve parametreler	92
	Ek 2 Her DM için kullanılan bölümler	94
	Ek 3 Uygulamada kullanılan bölümlerin mevcut ilk yerleşimleri.....	95
	Ek 4 Her DM için bölümler arası iş akışı matrisleri.....	98
	Ek 5 Her DM için yerleşimler arası uzaklık matrisleri.....	103
	ÖZGEÇMİŞ.....	108

SİMGE LİSTESİ

a_i	i tesisinin görünüş (aspect) oranı
a_{ij}	i tesisini j yerleşimine atamanın sabit maliyeti
$a_{\min}^i$	a_i için alt sınır değeri
$a_{\max}^i$	a_i için üst sınır değeri
C^*	Yığındaki en iyi birey
c_{jl}	j yerleşiminden l yerleşimine birim yük taşıma maliyeti
d_{ij}	i ve j tesislerine atanmış alanların merkezleri arasındaki dikdörtgensel uzaklığı
F	Amaç fonksiyonu
f_i	Yığındaki i . kromozomun uygunluğu
f_{ik}	i tesisinden k tesisine olan iş akışı
$f = \sum_{i=1}^n f_i$	Yığının toplam uygunluk değeri
$f_{ort} = \frac{f}{n}$	Yığının ortalama uygunluk değeri
G	Nesil aralığı
gen	Bir kromozomdaki her bir hücre
G_{max}	Maksimum iterasyon sayısı
l	Yığındaki bir kromozomun uzunluğu
Np	Popülasyon genişliği
N	Tesislerin ve yerleşimlerin toplam sayısı
n	Bir yığındaki toplam kromozom sayısı.
o_i	i tesisine atanmış alanın ölü alan oranını
$o_{\max}^i$	o_i için üst sınır değeri
Pc	Çaprazlama oranı
Pe	Elitizasyon oranı
Pm	Mutasyon oranı
U_{max}	Verilen bir parametrenin üst sınırı
U_{min}	Verilen bir parametrenin alt sınırı
x_{ij}	i tesisi j yerleşimine atandıysa 1, aksi halde 0

KISALTMA LİSTESİ

ALDEP	Automated Layout Design Programme
AnTabu	Karınca Algoritması-Tabu Arama
COFAD	Computerized Facility Design
COL	Computerized Model for Official Layout
CORELAP	Computerized Relationship Layout Planning
CRAFT	Computerized Relative Allocating of Facilities Technique
CX	Dizi Çaprazlama
DM	Dış Montaj
GA	Genetik Algoritma (Genetic Algorithms-GA)
GATB	Genetik Algoritma-Tavlatma Benzetimi
GP	Genetic Programming
GSP	Gezgin Satıcı Problemi
KA	Karınca Algoritması (Ant Colony Algorithms-ACA)
KAP	Kareli Atama Problemi
LAYOPT	Layout Optimizing Programme
MKKS	Melez Karınca Kolonisi Sistemi
NIST	National Institute of Standards and Technology
OX	Sıralı Çaprazlama
PMX	Kısmi Eşleşmeli Çaprazlama
PLANET	Computerized Plant Layout Analysis
TA	Tabu araştırması (Tabu Search-TS)
TB	Tavlama Benzetimi (Simulated Annealing-SA)
T-T	Tepe-Tırmanma (Hill Climbing-HC)
TYD	Toplam Yakınlık Derecesi
TYP	Tesis Yerleşim Problemleri
YSA	Yapay Sinir Ağları (Artificial Neural Network- ANN)

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Sürece göre yerleşim, grup yerleşim tipleri (Benjafaar, 2000)	4
Şekil 2.2 Sürece göre, parçalı dağıtılmış ve maksimum dağıtılmış yerleşim tipleri (Benjafaar, 2000)	4
Şekil 3.1 Genetik algoritmaların genel akış şeması	28
Şekil 3.2 Kod ve çözüm uzayı (Cheng vd., 1996)	35
Şekil 3.3 Ters çevirme mutasyonu	45
Şekil 3.4 Ekleme mutasyonu.....	45
Şekil 3.5 Yer değişikliği mutasyonu	46
Şekil 3.6 Karşılıklı değişim mutasyonu.....	46
Şekil 4.1 Örnek genetik kodlama gösterimi	53
Şekil 4.2 Eşit alanlı ve farklı alanlı departmanlara sahip yerleşim	54
Şekil 4.3.a Dilimlenmiş yapı	57
Şekil 4.3.b Dilimlenmiş ağaç.....	57
Şekil 4.4.a Altı tesisli bir dilimlenmiş yapı	59
Şekil 4.4.b Dilimlenmiş ağaç.....	59
Şekil 4.5 PMX (a) değişim (b) planlama ilişkisi (c) uyumlu hale getirme.....	62
Şekil 4.6 Komşu yerleşim.....	64
Şekil 5.1 Oluşturulan genetik algoritmanın temel yapısı ve işleyişi	67
Şekil 5.2 Oluşturulan genetik algoritmanın genel akış diyagramı	68
Şekil 5.3 SCLO’da problemin tanıtılması	72
Şekil 5.4 Elle girişte akış matrisi	73
Şekil 5.5 Akış matrisinin txt dosyasından okunması.....	73
Şekil 5.6 Akış matrisinin rastgele oluşturulması	74
Şekil 5.7 SCLO kullanıcı arayüzü	74
Şekil 5.8 SCLO sonuç raporu	75
Şekil 5.9 SCLO test girişi ile doğrulama	76
Şekil 5.10 Autocad ve Excel ortamında örnek modelleme	82

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 Yeniden üretim mekanizmaları	39
Çizelge 3.2 Tek noktalı çaprazlama	44
Çizelge 3.3 İki noktalı çaprazlama	44
Çizelge 4.1 Altı tesisin geometrik kısıtlan	57
Çizelge 5.1 SCLO programının seçilen literatür problemleri ile test edilmesi	78
Çizelge 5.2 DM'ler ve alanları	81
Çizelge 5.3 Mevcut durumda taşıma maliyetleri.....	83
Çizelge 5.4 SCLO kullanılarak bulunan yeni durumda taşıma maliyetleri.....	83

ÖNSÖZ

Fiyatların piyasada belirlendiği günümüz ticari rekabet ortamındaki işletmeler, her türlü araç ve yöntemi kullanarak kaliteden ödün vermeden maliyetlerini minimize etmeye çalışmaktadırlar.

Ürün imalat süreçlerindeki maliyet kalemlerinden biri olan ve son yıllarda, maliyet yaratsada imalata direk katkısı bulunmadığı çeşitli modern yönetim teknikleri ile de özellikle belirtilen taşıma maliyetlerinin minimize edilmesi sorunu işletmelerde maliyet düşürme çalışmalarında önemli bir yer tutmaktadır.

En küçük atölyeden uluslar arası büyük sanayi kuruluşlarına kadar başarılı işletmelere bakıldığında; tamamının, yapacakları işin büyüklüğü ve niteliğine göre bir tesis kurma; bu tesise en uygun şekilde yerleşim sağlama; bu yerleşimi değişen şartlara göre sürekli yenileme konusunda özel bir çaba içinde olduğu gözlenmektedir.

Üretim ve hizmet işletmelerinde maliyet ve verimlilik üzerine etkisi bulunan önemli unsurlardan biri olan tesis yerleşimi konusunda geçmişten günümüze uygulama ve deneysel nitelikte pek çok çalışma yapılmış ve yapılmakta olup, farklı uygulama alanları ve sektörlerle göre çeşitli metotlar önerilmektedir.

Tesis yerleşimi problemi, bazı durumlarda tüm çözüm alternatiflerinin denenemeyeceği sayıda alternatif çözüme sahip olduğundan dolayı, son yıllarda bu alanda tüm alternatif çözümleri değil, olası en uygun çözümleri arayan sezgisel yöntemler öne çıkmaktadır.

Bu sezgisel yöntemler içerisinde Genetik Algoritmalar önemli bir yer tutmakta olup; temel olarak, ilk bulunan en iyi çözümün çeşitli evrimsel yöntemlerle iyileştirildiği ve optimum zaman içerisinde bulunabilecek en iyi çözüme ulaşıldığı iteratif bir arama yöntemidir.

İşletmelerde taşıma maliyetlerinin minimize edilmesi amacıyla “Genetik Algoritma ile Tesis Yerleşimi Tasarımını” incelediğim bu çalışmamda destek olan; başta tez danışmanım Yrd. Doç. Dr. Bahadır GÜLSÜN; konu hakkındaki bilgi birikimini paylaştan Ar. Gör. Gülfem TUZKAYA ve oluşturulan algoritmanın C++ ortamında yazılımını gerçekleştiren arkadaşım Elektronik ve Haberleşme Mühendisi Hüseyin BULUT olmak üzere; iş hayatım devam ederken akademik hayatımın başlamasına olanak tanıyan Güner Kuyumculuk Makine San. Tic. Ltd. Şti. firma sahipleri Ahmet ve Nurettin GÜNER’e; ara verdiğim yükseköğrenimime geri dönmemde büyük desteğini gördüğüm, kendisi de bir eğitimci olan Baymak Sanayi ve Tic. A.Ş. Yönetim Kurulu Başkanı Dr. Murat AKDOĞAN’a; yüksek lisansa devamım konusunda desteğini esirgemeyen Legrand Elektrik A.Ş. İnsan Kaynakları Direktörü Kartal TOLGA ile uygulama çalışmasındaki desteklerinden dolayı yöneticilerim Çağatay DOMAÇ ve Emmanuel CALVEZ’e teşekkür ederim.

İş hayatının hız ve stresine yükseköğrenimin yoğunluğunun da eklendiği günler boyunca sürekli yanımda olan; özellikle manevi destelerini her an arkamda hissettiğim eşim Özlem ve kardeşim Derya olmak üzere, onlarsız bu çalışmayı asla gerçekleştiremeyeceğim aileme çok teşekkür ederim.

ÖZET

Tesis yerleşimi problemleri zor problemler olup; en iyi çözümün aranmasında sezgisel yöntemler tercih edilmektedir. Kareli Atama Problemleri (KAP) için tesis yerleşimi tasarımı NP-Zor sınıfına girmekte olup, bilinen hiçbir metot en iyi çözümü kesin olarak verememektedir. Bu tip problemler farklı çeşitte sezgisel metotlarla çözülür. Genetik algoritmalar bu tip sezgisel metotlardan olup KAP için iyi sonuçlar verebilmektedir. Genetik algoritmalar evrimsel yapıya sahiptirler ve optimum sürede, optimum uygunlukta çözümü bulmak için tasarlanmışlardır.

C++ ile bir bilgisayar programı oluşturuldu, SCLO (Subcontractor Layout Optimizer – Tedarikçi Yerleşim En İyileyici) olarak adlandırıldı. Program tasarımı aşamasında pek çok kod Genetik Algoritma kütüphanesinden alındı, bunlara çeşitli eklemeler yapıldı.

Oluşturulan program, KAP kütüphanesinden alınan literatür problemleriyle test edildi. Her problem için bilinen en iyi çözüme %99'dan daha fazla bir oranda yaklaşıldı.

Gerçek hayatta, yapısal elektrik malzemeleri imalat sektöründe programın uygulaması gerçekleştirildi. Teorik zeminde %41 taşıma maliyeti düşüşü sağlandı. Bu fazla kazanca sadece programın minimizasyon başarısı değil, bunun yanı sıra modelleme de yaşanan kısıtlarda sebep oldu.

Anahtar Kelimeler: Tesis yerleşimi problemi, genetik algoritma, kareli atama problemi

ABSTRACT

Facility layout problems're difficult and heuristic methods're chosen to solve when search the best solutions. Quadratic Assignment Problem (QAP) is a NP-hard problem, there is no method known giving the best solutions exactly. This kind of problems are solved by different kind of heuristics methods. Genetic algorithms are that kind of heuristic method, and they can provide a good solutions for QAP. Genetic algorithms has evolutionary structure; and they're designed to find optimum solution in optimum duration.

There is a computer program that is coding with C++ 's called SCLO (Subcontractor Layout Optimizer). In the designing steps of program many codes're taken from GA library and some additions're made on these.

Created program was tested by QAP library literature problems. Program reach the each problem known best results more than 99%.

In real life, program're implemented the firm that manufacture building elektrical materials. Theoretically transportation cost was decrease 41%. Not only the minimization success of program, but also some constraints problems caused by this too much gain.

Keywords: Facility layout problem, genetic algorithm, quadratic assignment problem

1. GİRİŞ

Donanımların ve iş alanlarının düzenlenmesi, endüstride kaçınılmazı olası olmayan bir sorundur. Donanımlar binanın duvarları arasına koyulmuş olsada çok değişik türde yerleşim alternatifleri söz konusudur. Bu sebeple "Tesis yerleştirebilecek miyiz?" değil, "Elimizdeki yerleşim ne kadar iyidir?" sorusu öne çıkmaktadır. Tesis düzenleme, endüstrinin temelidir ve bir işletmenin etkinliğini ve bazı durumlarda doğrudan doğruya yaşamını belirler. Pahalı donanımlar, karmaşık takımlar, iyi giden satışlar ve iyi tasarlanmış ürünlerin tümü, zayıf bir düzenleme nedeniyle feda edilebilir.

Darvin'in "en iyi olan yaşar (survival of the fittest)" prensibine dayalı olarak bir popülasyonu oluşturan bireylerin rekabet etmelerini ve rekabet sonucu elenmelerini sağlayan evrimsel süreci simüle eden genetik algoritmalar, ilk olarak John Holland, meslektaşları ve Michigan üniversitesindeki öğrencileri tarafından ortaya atılmıştır. Bunlar güçlü arama algoritmaları olarak kendilerini ispatlamış ve çeşitli alanlardaki zor problemlerle başa çıkmak için kullanılmışlardır. Genetik algoritmaların asıl etkisi, çok çeşitli problemlerde başarıyla uygulanabilmesinden kaynaklanmaktadır.

Bu çalışmada bir genetik algoritma geliştirilerek kodlanmış ve bu genetik algoritmayı kullanarak Kareli Atama Problemlerini çözebilecek bir bilgisayar programı yazılmıştır.

Birinci bölümde genel bir giriş yapıldıktan sonra ikinci bölümde tesis yerleşimi problemi incelenerek; tesis yerleşimi düzenlemenin temelleri, yerleşim tipleri ve bu konudaki yeni eğilimler, çözüm yöntemleri ile kareli atama probleminden bahsedilmiştir.

Üçüncü bölümde genetik algoritmaların temel yapısı anlatılmış; avantaj ve dezavantajlarının ne olduğu, çalışma prensiplerinin ne olduğu ve nerelerde kullanılabileceği literatürden alıntılarla verilmiştir.

Dördüncü bölümde literatürde bulunan genetik algoritma ile tesis yerleşimi çalışmaları incelenmiş, tüm aşamalar adım adım belirtilmiştir.

Beşinci bölümde sırasıyla algoritma-program yazımında kullanılan varsayımlar, algoritma işleyiş ve akışı özetlenmiştir. Programın doğru çalıştığı, literatürden seçilen problemler ile test edilmiştir. Gerçek hayattan bir problem tanımlanarak program yardımı ile alternatif bir çözüm oluşturulması aşamaları üzerinde detaylı bir şekilde durulmuştur.

Altıncı ve son bölümde çalışmadan çıkarılan sonuçlar ve öneriler aktarılmıştır.

2. TESİS YERLEŞİMİ

2.1 Giriş

Tesis yerleşim düzeni tasarımı, bir üretim ya da hizmet organizasyonunda, kuruluşunda; kuruluşun en etkin şekilde işletilmesini sağlayacak ve verimliliği artıracak şekilde bölümlerin, makinelerin, yardımcı hizmet ve tesislerin yerlerinin belirlenmesi ve düzenlenmesidir (Serpil, 1994).

Tesis yerleşimi, işletmedeki olanakların fiziksel olarak düzenlenmesini içermektedir. Yerleşmiş veya planlanmış bu düzenleme; malzeme hareketi, depolama, dolaylı işçilikler ve tüm diğer destek eylemler ve hizmetler için ve de işlem gören donanım ve işgücü için gereksinim duyulan alanları içermektedir.

Tesis yerleşimi kavramı bazen var olan yerleşim, bazen önerilen yeni yerleşim planı ve sıklıkla da, çalışma alanı veya tesis yerleşimi yapma işini belirtmek için kullanılır.

İş alanlarının sorunu, insanlık kadar eski bir sorundur. İlk yerleşim çalışmaları, işi yürüten adamın veya binayı tasarlayan mimarın bir ürünü idi. Arşivlerde bu çalışmalara ilişkin çok sayıda araştırma yer almaktadır. Ancak bunlardan çok azı, çizili bir örnek bırakmışlardır. Ancak bu, o zamanlarda üretimin olmadığını göstermemektedir. Çoğu durumda o insan, malzeme ve makinenin becerilerinin izin verdiği ölçüde etkin olabiliyordu. Nitekim bilindiği kadarıyla Venedikliler tarafından kullanılan ve kaydedilen bazı gemi üretim yöntemleri, 2. Dünya Savaşı'nın sonlarına kadar kullanılmıştır (Erkut ve Baskak, 1996).

Endüstri devriminin yüz elli yıl önce gerçekleşmesiyle sermaye sahipleri için fabrikaların düzenlenmesi ekonomik bir anlam kazandı. İlk gelişmeler, donanımın mekanizasyonu yönünde oldu. Yüzyılın sonlarına doğru işgücünün uzmanlaşması o denli arttı ki, malzemelerin işlemler sırasında taşınması büyük bir önem kazanmaya başladı. Bu arada sermaye sahipleri, işyeri düzenleme sorunlarını incelemek için uzmanlardan yararlanmaya başlamışlardı. İşte bu çalışmalarla birlikte teknikler gelişmeye başladı.

2.2 Tesis Yerleşim Düzenlemenin Temelleri

Yerleşim tasarımı genellikle üretim yada hizmet endüstrisi sistemlerinin performansı üzerinde önemli bir etkiye sahiptir ve onlarca yıldır aktif bir araştırma sahası olmuştur (Meller ve Gau,1996).

Tesis yerleşim problemlerinde optimum çözümü bulmak önemlidir. Bunun iki nedeni vardır (Kulluk,2003);

- Toplam maliyetlerin, % 30-70'ini malzeme taşıma maliyetleri oluşturduğu için malzeme taşıma maliyetlerini en aza indirmek gerekir.
- Tesis yerleşimi uzun dönemli ve maliyetli bir iştir. Mevcut tesis üzerinde değişiklik yapmak veya tesisi yeniden düzenlemek büyük yatırım ister.

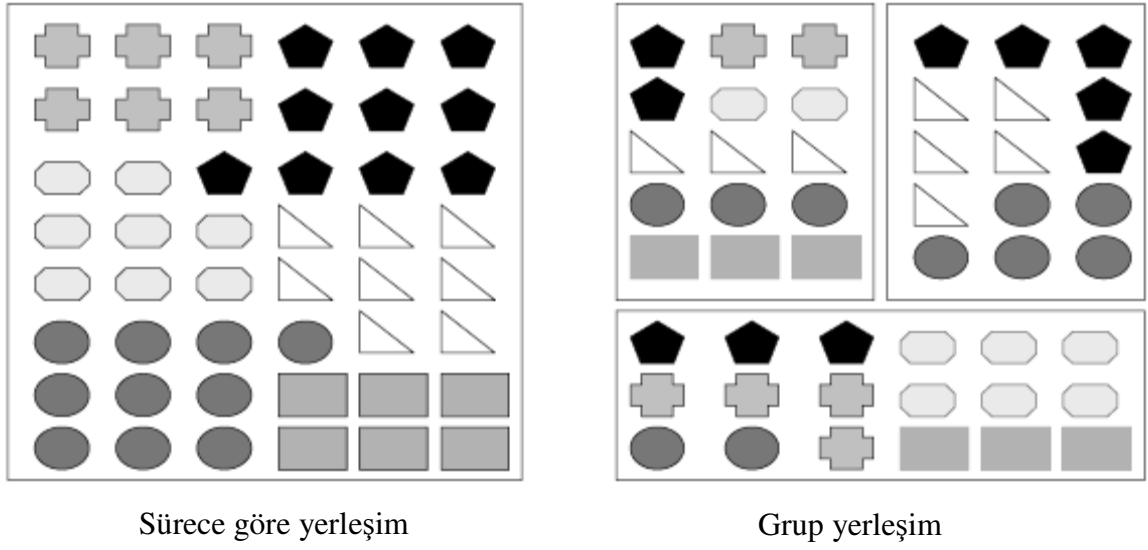
2.2.1 Tesis yerleşim tipleri

Tesis yerleşim tipleri temel olarak dört grupta toplanabilir (Serpil, 1994):

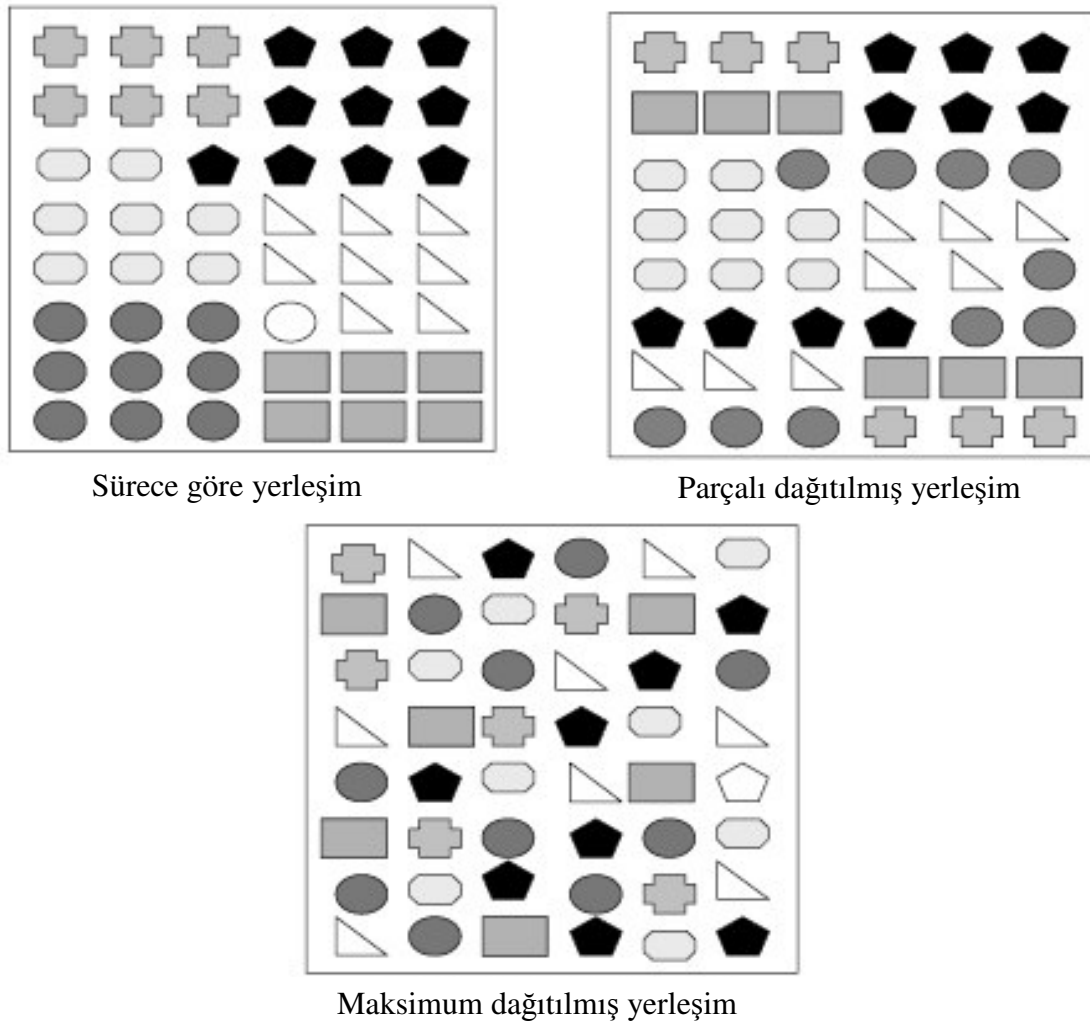
- Sabit konum yerleştirme düzeni
- Ürüne göre yerleştirme düzeni
- Sürece göre yerleştirme düzeni
- Grup yerleştirme düzeni

Sabit konum yerleştirme düzeninde üretilen ürün değişmez bir konumda durmakta, gerekli makine ve servisler ürüne getirilmektedir. Bu yerleşim düzeni daha çok gemi ve uçak gibi çok büyük yer kaplayan ürünler için kullanılır.

Ürüne göre yerleştirme düzeninde ürünün üretiminde kullanılan makine ve yardımcı servislerin ürün işlem sırasına göre yerleştirilmesi yapılır. Ürüne göre yerleştirmede makineler değişik ürünler tarafından paylaşılmadığından makinelerden yararlanma oranlarının yüksek olmasını sağlamak için ancak üretim hacmi çok yüksek olan ürünlerde bu yerleştirme şekli uygun olmaktadır.



Şekil 2.1 Sürece göre yerleşim, grup yerleşim tipleri (Benjafaar, 2000)



Şekil 2.2 Sürece göre, parçalı dağıtılmış ve maksimum dağıtılmış yerleşim tipleri (Benjafaar, 2000)

Sürece göre yerleştirme düzeninde ise aynı ya da benzer tip makine ve servisler birbirlerine yakın yerleştirilirler. Bu tip yerleştirme düzeni genellikle çok sayıda ürünün, göreceli olarak az hacimde üretildiği işletmelerde kullanılır.

Grup yerleşim düzeni; ürüne göre yerleştirme düzeni ile sürece göre yerleştirme düzeninin bir ara noktası olarak görülebilir. Bu yerleşim düzeni, ürünlerin üretim hacimleri ürüne göre yerleştirme düzenini uygun kılacak büyüklükte olmadığı durumlarda, bazı ürünleri kendi içlerinde gruplandırıp, yerleştirme düzeni yapılmasıyla oluşturulmuştur. Ürünlerin kendi aralarında gruplandırılmaları, Grup Teknolojisi adlı çalışmaların sonucunda yapılır.

Bununla birlikte, günümüzün gelişen rekabet koşulları ve buna paralel gelişen ürün çeşitliliği, talep esnekliği ortamında artık bu temel yaklaşımlarda yeterli olamamakta; esnek, modüler, daha kolay konfigüre edilebilir yerleşim tiplerine ihtiyaç duyulmaktadır. Benjafaar (2000)'in yeni nesil yerleşim tipleri olarak adlandırdığı bu yerleşim tiplerinde, ürün çeşitliliğini verim düşürmeden sağlama, talepte yaşanan dalgalanmalara hızlı tepki verebilme ve hızla değişen iş ortamına veya yeni ürün gelişine hızla ayak uydurulabilmektedir. Böylece daha esnek ve tepkisel üretim sistemleri kurabilmek olanaklı hale gelmektedir (Benjafaar, 2000). Şekil 2.2'de bu yerleşim tipine ait örnekler görülmektedir.

2.2.2 Tesis yerleşimi düzenlemede amaçlar

Genel olarak tesis düzenlemenin amacı, çalışma alanının ve donanımın en işlevsel; işgücü için en güvenli ve doyurucu bir şekilde düzenlenmesinin yapılmasıdır. İstenilen; üretken insan, malzeme, makine ve destek eylemlerin düzenlenmesidir. Daha özel alınırsa, iyi bir işyeri düzenlemenin avantajları, işletme maliyetlerinden kazanç ile ortaya çıkar.

İşyeri düzenlemede amaçlar şöyle sıralanabilir (Erkut ve Baskak,1996):

- İnsan, malzeme, makine ve sermaye gibi üretim öğelerinden en büyük verimi sağlamak,
 - Bina ve tesislerin rasyonel kullanımı daha düşük yatırım maliyeti sağlanması,
 - Malzeme taşıma maliyetlerinin azaltılması,
 - İşin yapılması için işçinin harcayacağı çabanın en aza indirilmesi,
 - Üretim etkinliklerinin daha verimli bir şekilde denetiminin sağlanması,
 - Süreçteki işlemlerin, tasarlanan sırada ve sürede yapılması.

- Düzgün bir yönde yerleştirme yapmak,
- Düzgün ve engelsiz bir taşıma yolu sağlamak,
- Geri yönde taşımadan kaçınmak,
- İlgili işlemleri bir araya toplamak,
- Değişiklikleri içerecek esnekliği sağlamak,
- Genişleme için olanak sağlamak,
- El ile taşımadan kaçınmak,
- Planlı depolama yapmak, daha az ara stok ve daha az karışıklık olmasını sağlamak, işlem gören malzeme stoğunu azaltmak,
- Donanımın bakımını daha iyi yapmak,
- Yoğun işte daha fazla işgücü çalıştırmak,
- Malzemeye veya kalitesine daha az zarar verilmesini sağlamak,
- Bürokratik işleri ve dolaylı işgücünü azaltmak. Morali ve işgücü doyumunu arttırmak,
- İşgücünün sağlığı ve güvenliği üzerindeki riski azaltmak,
- İşlevsel (yardımcı) tesislerin verimli ve ekonomik dağılımını sağlamak,
- Toplam üretim süresinin en aza indirgenmesi.

Hiç kuşkusuz tüm bu amaçların aynı anda gerçekleştirilmesi olanaksızdır. Bununla birlikte bu amaçlar, yerleşim mühendisleri tarafından elde edilmiş ve sağlanmış amaçlardır.

Stevenson (1999) ise tesis yeniden düzenlemesine neden olan etmenleri şöyle sıralamıştır:

- Etkin olmayan operasyonlar (Örn; yüksek maliyet, darboğazlar),
- Kazalar ve İş güvenliği riskleri,
- Ürün ve servis tasarımında değişiklikler,
- Yeni ürünler veya hizmetler,
- Çıktı miktarlarındaki değişimler veya karışıklıklar,

- Metot ve ekipmanlardaki değişiklikler,
- Çevresel veya yasal ihtiyaçlarda oluşan değişiklikler,
- Moral Problemleri (Örn; yüz yüze kontak halinde çalışma).

2.2.3 Tesis yerleşimi düzenlemede ilkeler

Tesis yerleşiminin ilkeleri aşağıdaki gibi sayılabilir (Erkut ve Baskak, 1996):

1. Bütünsel integrasyon ilkesi

İşyeri düzenleme; işgücü, malzeme, makine, destek eylemler ve herhangi bir diğer etmeni en iyi bir biçimde uzlaştırarak bir araya getiren yoldur. Tüm olanakları bir araya getirerek büyük bir birim oluşturmaktır. Bir anlamda tüm tesisler, tek bir makine oluşturur.

2. En küçük hareket ilkesi

Tüm diğer etmenler eşdeğer olduğunda, malzemeyi işlemler arasında en az hareket ettiren yerleşim, en iyidir.

3. Akış ilkesi

Tüm diğer etmenler eşdeğer olduğunda, iş alanını her bir işlem veya süreç için, işin akış sırası yönünde düzenleyen yerleşim en iyidir. Bu, en küçük hareket ilkesinin tamamlayıcısıdır. İşlerin bitime doğru ilerlemesini ve iş akışlarında geri dönmelerin ve kesişmelerin olmamasını vurgular.

Malzeme, makine parkında hiçbir kesintiye uğramadan, “kayar gibi” ilerleyecektir. İyi bir akış daima, doğrusal akış demek değildir. Değişik türden akışlar (U, L,...) söz konusu olabilir.

4. Üç boyutluluk ilkesi

Ekonomiklik ile hem yatay hem de dikey alanın etkin bir biçimde kullanılabilmesi sağlanır.

Temelde işyeri düzenleme, mekana biçim verilmesidir. Yani işgücü, malzeme, makine ve destek eylemleri tarafından doldurulan değişik boşlukların düzenlenmesidir. İnsanın, malzemenin veya makinenin hareketi, üç yönden herhangi birisine doğru olabilir.

5. İş doyumu ve iş güvenliği ilkesi

Tüm diğer etmenler eşdeğer olduğunda işi, işgücü için doyumlu ve güvenli yapan yerleşim en iyidir.

İşgücünün doyumu çok önemlidir. En temel amaçtır. Bazı işyeri düzenleme uzmanları için tek amaçtır.

6. Esneklik ilkesi

Tüm diğer etmenler eşdeğer olduğunda, en küçük maliyetle, en az sakınca ile ayarlanıp düzenlenebilen yerleşim en iyidir.

2.2.4 Tesis yerleşimi düzenleme çalışması aşamaları

Tesis yerleşim düzeni tasarımı, her tasarım gibi sistem yaklaşımı ile çözümlenmelidir. Sistem yaklaşımının temel felsefesi sistemi oluşturan unsurların tümünün birden ele alınması ve yapılacak çalışmaların bir bütünsellik içinde yürütülmesidir. Sistem yaklaşımını oluşturan evreleri şöyle sıralamak mümkündür (Serpil C., 1994):

- Problemin tanımlanması,
- Bilgilerin toplanması,
- Çeşitli çözüm önerilerinin çıkarılması,
- Önerilerin karşılaştırılıp en uygununun seçilmesi.

Problemin tanımlanması

Bu aşamada tasarımda kullanılan amaçlar tanımlanır. Sonraki aşamalarda geliştirilen çözüm önerilerinin karşılaştırılmalarında kullanılacak olan amaçlar ve etkinlik ölçüleri saptanır. Etkinlik ölçüleri, önerilerin amacı ne denli gerçekleştirdiklerini saptamada kullanılan değerlendirme ölçüleridir.

Bir yerleştirme düzeni tasarımında amaçların tümünü sağlayan seçeneklerin üzerinde durulması beklenir. Bunun mümkün olmadığı durumlarda, öncelikler belirlenerek hareket edilmelidir.

Kuruluş yeri seçimi probleminde geçerli olan faktörlerde olduğu gibi bu amaçların bir kısmı için nicel etkinlik ölçüleri tanımlamak mümkündür. Etkinlik ölçülerinden en çok kullanılanı malzeme taşıma maliyetleri ya da malzeme taşımaları sırasında kat edilen uzaklıktır.

Bilgilerin toplanması

Yukarıda belirtilen ve yerleşim düzeni tasarımını gerektiren durumlar göz önüne alındığında, bu tasarımın diğer bazı tasarımlarla, özellikle ürün tasarımı, süreç tasarımı ve üretim planlaması ile iç içe olduğu ve bunların birbirleri ile etkileşim içinde oldukları görülür.

Bilgi toplama aşamasında amaç; ürünler, üretim süreçleri ve üretim planları hakkında veriler

toplayıp, bunları yerleşim düzeni tasarımı için gerekli bilgilere dönüştürmektir.

Ürün tasarımı, yerleşim düzeni tasarımını doğrudan etkilemektedir. Ürünün hangi hammaddeleden yapıldığı, üzerinde ne gibi işlemlerin yapıldığı yerleşim düzenini doğrudan etkiler. Ürün tasarımı hakkında gerekli bilgiler ürün resimlerinden, çizimlerden, prototiplerden, parça listelerinden, ürün ağaçlarından ve montaj çizelgelerinden elde edilebilir. Ürün ağaçlarının parça listelerinden farkı; bir ürünün, içindeki parçalarla olan ilişkisini hiyerarşik bir şekilde göstermesidir. Montaj çizelgeleri, bir ürünün üzerinde yapılacak tüm işlem ve montajların sırasını gösteren grafiksel bir gösterimdir.

Süreç tasarımı, bir üründe kullanılan parçaların kuruluş dışından mı alınacağı yoksa kuruluş içinde mi üretileceği; parçaların üretiminin nasıl yapılacağı; hangi makine araç ve gereçlerin kullanılacağı; operasyonların ne kadar zamanda yapılacağı gibi bilgiler elde edilebilir. Bu veriler operasyon çizelgeleri ve rota çizelgelerinden elde edilebilir. Operasyon çizelgesinde, üretim için gerekli olan işlemler, işlemler için ihtiyaç duyulan işgücü, kullanılması gereken makine araç ve gereçler ile parça başına makine hazırlama ve işleme zamanları gösterilir.

Süreç tasarımı ile ilgili veriler toplanırken, var olan süreç tasarımının en iyi süreci tanımlamasına özen göstermek gerekmektedir. Etkili bir yerleşim düzeni tasarımı için, var olan ancak düzeltilmesi gereken süreçler değil, en uygun sürecin tanımlanması gerekir.

Üretim planlarından ise ürünlerin üretim zamanları ve miktarları ile ilgili bilgiler toplanır. Üretim için gerekli makine, araç ve gereç sayısı, iş gücü gereksinimleri üretim planlarından elde edilir.

Bunların dışında üretim faaliyetlerinin sağlıklı bir şekilde yürütülmesini sağlamak için gerekli olan yardımcı hizmet veren bölümler ile ilgili bilgiler de toplanmalıdır. Bakım, onarım, takım odası, soyunma odaları, yükleme ve teslim alma alanları, depo alanları, kafeterya ve sağlık odaları gibi hizmet alanları hakkındaki bilgiler bunlara örnektir.

Çözüm önerilerinin geliştirilmesi aşamasında; problemin tanımına göre belirlenmiş olan yöntemlerden uygun olanlar tanımlanır. Yöntemlerin avantaj ve dezavantajları, güçlü ve zayıf yönleri, tahmini maliyetleri, uygulanabilme süreleri gibi özellikleri ortaya konarak karar alıcılara bilgi sunulur.

Önerilerin karşılaştırılıp en uygununun seçilmesi aşaması; geliştirilen öneriler içerisinden işletme ve işletmede tanımlanan probleme maliyet, uygulanabilirlik, bitiş süresi, kaynaklar vb.

açısından en uygun çözümün seçilmesi aşaması olup yönetimin onayı alınarak gerçekleştirilmesi gereklidir.

2.2.5 Tesis yerleşim sorunlarının sınıflandırılması

Tesis yerleşim sorunları; ürün, hizmet; miktar, tasarım yada prosesinde değişiklik yapılması, yeni bir makine-teçhizatın devreye alınması gibi durumlarda ortaya çıkabildiği gibi yeni bir tesise taşınması veya yeni bir işletmenin kurulması şeklinde de ortaya çıkabilmektedir.

Uzun zaman boyunca biriken problemlerin çözümü için de tesis yerleşim düzeninin yeniden gözden geçirilmesi gereği duyulabilir. Malzeme aktarma zamanlarının toplam üretim zamanına göre yüksek olması, kalabalık çalışma ortamları, yetersiz işlemler, makinelerde darboğazlar, açıklanamayan gecikmeler ve boş zamanlar ve kötü çalışma ortamı böyle durumlara örnek olarak gösterilebilir.

Sebepleri dikkate almaksızın temel olarak tesis yerleşim problemleri şu şekilde sınıflandırılabilir (Erkut ve Baskak,1996) :

- Var olan düzenlemede küçük değişikliklerin yapılması,
- Var olan düzenlemenin yeniden ele alınması,
- Var olan tesislere taşınma,
- Yeni bir işletmenin/fabrikanın kurulması

2.2.6 Tesis yerleşim etmenleri

Bir tesisin veya işyerinin düzenlenmesinde etkili olan çok sayıda etmen vardır. Bu etmenleri sekiz ayrı grupta toplamak olasıdır (Erkut ve Baskak, 1996):

- **Malzeme:** Tasarım, değişiklik, miktar, gerekli işlemler ve bunların sırası.
- **Makine:** Üretim araçları, takımlar ve bunların kullanımı.
- **İnsan:** Gözetim, denetim ve hizmet.
- **Hareket:** Bölüm içi ve bölümler arası taşıma, değişik işlemler, depolamalar ve muayeneler.
- **Bekleme:** Sürekli ve geçici depolamalar ve gecikmeler.

- **Hizmet:** Bakım, muayene vb. hizmetler.
- **Bina:** Binanın iç ve dış özellikleri, kullanım özellikleri ve donanım.
- **Değişim:** Genişleme, esneklik.

2.3 Tesis Yerleşimi Yöntemleri

Pek çok literatür kaynağı tesis yerleşim yöntemlerini yapısal açıdan algoritmik ve prosedürel olmak üzere iki büyük kategoriye ayırırlar (Yang ve Kuo, 2003).

Algoritmik yaklaşımlar; genellikle çözümü elde edilebilen amaç fonksiyonunun yerini tutan bir fonksiyona ulaşmak için tasarım kısıtlarını ve amaçlarını basitleştirirler. Mevcut literatürün çoğunluğu algoritmik yaklaşımları anlatır (Heragu,1997). Algoritmik yaklaşımlar, özellikle, ticari yazılım, örneğin Spiral® (Goetschalckx,1992) ve LayOPT® (Bozer ve diğerleri,1994), verimli bir şekilde yerleşim alternatifleri üretebilirler. Buna rağmen, nicel hedeflere ulaşılması çoğu zaman tasarım amaçlarının hepsini gerçekleştirmez.

Muther (1973)'e göre prosedürel yaklaşımlar tasarım proseslerindeki nitel ve nicel amaçları birleştirebilirler (Yang ve Kuo, 2003). Bu yaklaşımlar için, tasarım prosesi ardışık olarak çözölen çeşitli adımlara bölünmüştür. Bir prosedürel yaklaşım uygulamasının başarısı genellikle deneyimli bir tasarımcı tarafından sağlanmış nitel tasarım alternatiflerinin ortaya çıkmasına bağlıdır. Bu nedenle, böyle bir yaklaşım kişisel (öznel) olabilir ve sağlam bilimsel temel eksikliği yüzünden iyi olamayan bir çözüm üretebilir. Dolayısıyla, kişiselleşme (öznellik) ve etkinsizlik ihtimalleri yerleşim tasarımı problemini çözmeye yönelik prosedürel yaklaşımın adaptasyonunu engeller.

Yerleşim oluşturma ve değerlendirme, içsel çok amaçlı doğası ve bilgi toplama prosesi yüzünden mücadele gerektirir ve zaman tüketir (Lin ve Sharp,1999a,b).Geçmişteki ve yeni ortaya çıkan araştırmalar bu ihtiyaçların karşılanması için çözüm metodolojisi geliştirmeyi amaçlamıştır. Buna rağmen, algoritmik yaklaşımlar esas olarak malzeme taşıma maliyetlerini minimize etmek için akış uzaklıklarını minimize etmeye odaklanmışlardır. Diğer yandan, prosedürel yaklaşımlar uzmanların tecrübelerine güvenmişlerdir. Ne algoritmik ne de prosedürel yerleşim tasarım metodolojisi, uygulanabilir tasarım problemlerinin çözümünde zorunlu bir etkinlik değildir (Yang vd,2000).

2.3.1 Nicel çözümleme modelleri ve teknikler

Sayısal metot ve teknikler olukça fazla olmakla birlikte; problemin özel durumu ve karmaşıklığı dikkate alınarak oluşturulan yöntemler arasında genel olarak kullanıma sahip olanlar şunlardır (Erkut ve Baskak, 1996; Tompkins vd., 2003).

2.3.1.1 Grafik metodu

Fabrikada üretilen çeşitli ürünler, bu işlem merkezlerinin birinden ötekine kümeler veya yığınlar halinde taşınmakta ve işlemlere tabi tutulmaktadır. Her bir ürün mutlaka aynı işlem sırasına veya aynı sayıda işleme tabi değildir.

Malzeme dolaşımını minimize edebilmek için, üretim çevrimi esnasında yarı ürünün hangi işleme merkezleri arasında hareket ettiği kaydedilir. Hareketleri tespit edebilmek için ürün kümelerini oluşturan malzemenin (sayı, hacim veya ağırlık olarak) eşit miktara ayrılmış gruplar halinde bir merkezden diğerine taşındığı farz edilir.

Bu metotta bütün mesele, işlem merkezlerini, çeşitli malzeme kümelerinin minimum masrafla dolaşmalarını temin etmektir.

2.3.1.2 Gezi şeması

Bu metot ürüne göre düzenlemede uygulanmaz. Buna karşın prosese göre düzenlemede sistemli bir yerleşmeyi incelemeyi mümkün kılan bir analiz aracıdır.

Gezi şeması, çok sayıda verilerin matris halinde ve topluca sunumunu sağlar. Bu şema, yol haritalarının kilometre grafiğine benzer.

Gezi şeması iki işlem merkezi arasındaki (genellikle metre cinsinden) uzaklıkları veya taşınan malzeme miktarını gösterir. Bazen de bu sayılar nakliyat frekansını veya hem frekansı hem de uzaklıkları, miktarları veya ağırlıkları da gösterir.

2.3.1.3 Atama yöntemi

Doğrusal ulaştırma yönteminin özel bir durumudur ve çok yalın bir yapıya sahip olması nedeniyle kendisine özgü çözümleme yolları getiren özel bir yaklaşımdır.

Yerleşim düzeni sorunları kesikli veya sürekli konumlar kullanılarak ele alınabilir. Çoğunlukla yeni makineler için konumların kesikli olduğu kabul edilir.

2.3.1.4 Wimmert yöntemi

İlk sayısal teknikleri önerenlerden biri olan R.J.Wimmert ölçek olarak makine veya bölümler arasındaki uzaklık-hacim-malzeme taşımalarının en küçük olmasını kullanmaktadır.

Üç temel varsayımı vardır:

- Bölüm yerleri birbiriyle yer değiştirebilir,
- İki yer arasındaki uzaklık hareketin yönüne bağlı değildir,
- Maliyet ile eşdeğer uzaklık arasında doğru orantı vardır.

2.3.1.5 Spiral analiz yöntemi

Spiral analiz yönteminin amacı, imalat bölümlerinin sıralamasında birbirini izleyen işlemler arasında en dolaysız malzeme akışını sağlamaktır. İlk yapılan yerleştirmenin rasyonel olduğu varsayımı altında fabrika yerleştirilmesinde değişiklikler yapılması durumunda, bölümlerin gereksinmesi açısından çok küçük değişiklikler söz konusu olacaktır. Bundan başka bölümler veya atölyelerin kare veya dikdörtgen binalardan oluşması nedeniyle binaların çevresel biçiminde yapılacak değişiklikler sonucunda imalat bölümlerinin kapladığı alanlar açısından ortaya çıkacak etkiler çok küçük olacaktır. Bu iki varsayım altında temel amaç, eldeki toplam alan içinde, var olan birim alanların konum ve biçim açısından ilişkilerini belirlemektir.

2.3.1.6 Doğrusal hat yerleştirme yöntemi

Yöntem; makinelerin veya bölümlerin düz doğrular üzerinde yer aldığı ve malzeme akışlarının da bu doğrular üzerinde gerçekleştiği varsayımına dayanır. Eğer çaprazlama atlayışlar olabilecekse yöntem bu tip yerleştirme problemlerinde kullanılamaz.

2.3.1.7 Hat dengeleme

Ürün oluşumu sırasında yapılması gereken işlerin, montaj istasyonlarına, kayıp süreleri en aza indirecek şekilde atanması işlemine hat dengeleme denir. Hat dengeleme kavramı yerleşimler arasında kıyaslama yapılmasını sağlayacak denge kaybı vb. sayısal sonuçlar üretebildiğinden yerleşim tasarımında kullanılacak yöntemlerden biri olmaktadır.

2.3.1.8 Noy metodu

Noy tarafından geliştirilen bu metot, prosese göre düzenlemede basit ve pratik bir çözüm yolu

oluşturmaktadır. Bu metot, çeşitli ürün partilerinin ait oldukları işlemlerin takip edecekleri en uygun rotanın araştırmasıyla ilgilidir. Noy'a göre, bir imal prosesini teşkil eden çeşitli imal merkezleri veya üniteleri, her bir ürün için belli bir yörünge üzerinde bulunur. Bu kabul üretim merkezlerinin fiziksel olarak bir doğru üzerinde yerleşmelerinden çok, ürün içeren tezgahlar arasında çaprazvari dolaşamayacakları anlamına gelmektedir. Diğer deyimle, Noy'a göre ürün akımının, son işleme varıncaya kadar belli bir hattı takip etmesi gerekir.

2.3.1.9 Yükleme yolu metodu (Load path method)

Üretim prosesine giren malzemenin işlem merkezleri arasında en kısa yoldan hareketini sağlamak esasına dayanır.

2.3.1.10 Dal-sınır algoritması (Branch and bound algorithm)

Amaç işlevinin en aza indirme olduğu kabul edildiğinde amaç işlevinin en iyi değeri için bir üst sınır bulunduğu varsayılır. Yapılacak ilk şey tüm olurlu çözümler kümesinin alt kümelere bölünmesi ve her birinde o alt kümedeki çözümlerin amaç işlevinin değeri için alt sınır bulunmasıdır. Daha sonraki alt sınırları amaç işlevinin son durumdaki üst sınırını geçen alt kümeler işlem dışı bırakılır. Geriye kalan alt kümelerden biri daha sonra yeniden birçok alt kümelere bölünür onlarında alt sınırları bulunur ve daha öncede yapıldığı gibi kimi alt kümeler yine işlem dışı bırakılır. Geriye kalan alt kümelerden biri seçilerek yeniden parçalara ayrılır. Bu işlem amaç işlevinin değeri herhangi bir alt kümenin sınırından küçük oluncaya ve olurlu bir çözüm buluncaya değin yinelenir.

2.3.2 Bilgisayar destekli algoritmalar

Tesis yerleşimi çalışmasında bölüm veya iş merkezi sayısının artması ile mümkün seçeneklerin sayısı çok daha fazla bir oranda artmaktadır. Örneğin hiçbir kısıt konulmadığı takdirde 10 iş merkezinin yerleştirilmesi için $10! = 3.628.800$ adet seçenek söz konusudur. Çok iş merkezli problemlerin grafik yoldan çözümü oldukça zor olduğundan, bilgisayar destekli tesis yeri düzenleme yöntemleri geliştirilmiştir. Bu yöntemler üretim ve hizmet tesislerine uygunluk ile kurucu ve geliştirici olma özelliklerine göre sınıflandırılabilir. Kurucu yöntemler boş bir alan ile çözüme başlarken, geliştirici yöntemler bir başlangıç tesis yeri düzeni (örneğin: mevcut düzen) ile çözüme başlamaktadır.

Söz konusu yöntemler; yerleştirdiği bölüm veya iş merkezi sayısı, yerleştirdiği bina katı sayısı, bilgisayar işlem süresi ve sonuç kalitesi açısından farklı özelliklere sahiptir.

Başka bir sınıflandırmaya göre tesis yerleşim yöntemleri kullanılış amaçlarına göre iki şekilde sınıflandırılır (Aleisa ve Lin, 2005):

- Kuruluş veya Yapı Algoritmaları
- Geliştirme Algoritmaları

2.3.2.1 Kuruluş veya yapı algoritmaları

Bu algoritmalar, işyeri düzenini taslak bir çözüm bularak kurarlar. Bu; işyeri düzeni tasarımına ulaşıncaya değin bölümlerin birbirini izleyen seçimini ve yerleştirilmesini içerir. Kuruluş algoritmaları, belirli bir amaç ölçütüne göre, sıfır çözümden başlayarak bölümleri tek tek ele alıp yerleştirirler ve sonuçta, amacı gerçekleyen biçimde yerleşim düzeni planı (kalıp plan) oluştururlar.

Kurucu yöntemler olarak şunlar sayılabilir (Tanyaş, 1995) :

- CORELAP (Computerized Relationship Layout Planning - Programlanmış işyeri düzeni tasarımı programı)
- ALDEP (Automated Layout Design Programme - Makineleştirilmiş işyeri düzeni tasarımı programı)
- PLANET (Computerized Plant Layout Analysis and Evaluation Technique - Programlanmış işyeri düzeni çözümlemesi ve değerlendirme tekniği)
- LAYOPT (Layout Optimizing Programme - İşyeri düzenini eniyileştirme programı)
- CASS
- COLO2
- COMP1
- COMP2
- COMSBUL
- DOMINO
- GENOPT
- IMAGE

- KONUVER
- LAYADAPT
- LSP
- MUSTLAP2
- PLAN
- RMA
- SISTLAP
- SUMI

Kurcu yöntemlerin en çok tanınanı olan CORELAP ilk kez 1967 yılında Robert C. Lee ve James M. Moore tarafından önerilmiş ve daha sonra R. Sepponen tarafından geliştirilmiştir. CORELAP yönteminde başlangıçta aşağıdaki veriler bilgisayara okutulmaktadır (Lee ve Moore, 1967):

- Bölümler (veya iş merkezleri) arası yakınlık dereceleri,
- Bölüm sayısı,
- Bölümlerin alanları,
- Yakınlık derecesi ağırlıkları.

Bu temel bilgilerin yanı sıra zorunlu olmayan aşağıdaki bilgiler de bilgisayara okutulabilir:

- Sonuç tesis yeri düzeninin ölçeği,
- Binanın veya arsanın uzunluk ve genişliği,
- Bazı bölümler için ön atamalar.

CORELAP yöntemi öncelikle her bölüm için Toplam Yakınlık Derecesini (TYD) hesaplar. Bölümler TYD'lerine göre büyükten küçüğe doğru sıralanır. En yüksek TYD'ne sahip bölüm yerleştirme yapılacak alanın merkezi noktasına yerleştirilir. Yöntem bundan sonra iki sorunun cevabını arar:

Bundan sonra hangi bölüm yerleştirilecektir? Bu bölüm “Başaran (Victor)” olarak

adlandırılmaktadır.

Başaran bölüm yerleştirme yapılacak alana nasıl yerleştirilecektir? Yerleştirilen bölüm ise “Kazanan (Winner)” olarak adlandırılmaktadır.

Yöntem bir kazanan bölüm elde ettikten sonra bu bölüm ile A ilişkisi içinde olduğu Aday (Candidate) bölümler aramaktadır. Aday bölümler arasında en yüksek TYD'ne sahip bölüm Başaran bölüm olmakta ve düzene yerleştirme hakkı elde etmektedir. İlk kazanan ile A yakınlık derecesine sahip diğer aday bölümler de yerleştirildikten sonra yöntem ilk Başaran bölümü (ikinci olarak düzene yerleştirilen bölüm) Kazanan bölüm olarak nitelendirmekte bu bölüm ile A ilişkisi içinde olan aday bölümler TYD büyüklüğüne göre yerleştirilmektedir. Daha sonra sırasıyla ikinci, üçüncü ve diğer başaran bölümler kazanan bölüm olarak nitelendirilmekte yine A ilişkisi aranmaktadır. A yakınlık derecesi kalmadıktan sonra yöntem başa dönerek sırasıyla B, C, D, E ve F ilişkileri için aynı işlemleri tekrarlamaktadır. Sonuçta tek bir tesis yerleşimi elde edilmektedir. Seçenek tesis yerleşimi elde etmek için başlangıç verilerinin değiştirilmesi gerekmektedir. Kuruluş algoritmaları, yeni bir işyerinin yerleştirilmesi sorununa çözüm getirebilirler.

2.3.2.2 Geliştirme algoritmaları

İkinci tip algoritmalar "Geliştirme Algoritmaları" olarak tanımlanırlar. Bu tür algoritmalarda ilkin, işyerinin var olan düzeni tümüyle incelenir ve bölümlerin yerleşimleri, tasarımı geliştirecek biçimde birbirleriyle değiştirilir. Daha doğrusu geliştirme algoritmaları; var olan yerleşim düzeninin planını girdi olarak ele alır ve amaç ölçütünün gerçekleştirilmesi doğrultusunda, dış sınırları aynı kalmak koşuluyla, plan üzerinde bölümlerin birbirleriyle yerlerini değiştirerek iyileştirilmiş bir yerleştirme düzeni planı elde ederler.

Geliştirici yöntemler olarak şunlar sayılabilir (Tanyaş, 1995) :

- CRAFT (Computerized Relative Allocating of Facilities Technique - Tesislerin Programlanmış Görelî Yerleştirilmesi Tekniği)
- COFAD (Computerized Facility Design - Programlanmış Tesis Tasarımı)
- COL (Computerized Model for Official Layout)
- COSFAD
- GRASP

- OFFICE
- PREP
- SET

CRAFT yöntemi ilk kez 1963 yılında G. C. Armour ve E. S. Buffa tarafından önerilmiş ve daha sonra E. S. Buffa, G. C. Armour ve T. E. Vollman tarafından geliştirilmiştir. CRAFT yöntemi daha ziyade fiziksel malzeme akışının söz konusu olduğu üretim tesisleri için uygundur. Bu nedenle Geliş-Gidiş Şemasını kullanır. Amaç malzeme akışındaki maliyet giderlerinin en aza indirilmesidir. CRAFT yönteminde başlangıçta aşağıdaki veriler bilgisayara girilir (Tanyaş, 1995):

- Geliş-gidiş şeması verileri,
- Malzeme hareketi maliyet verileri (Birim yük ve uzaklık taşıma maliyetleri),
- Başlangıç tesis yeri düzeni.
- Yerleri değişmeyecek bölümler (bu bölümler yöntemin işlemesi boyunca başlangıçta belirlenen yerlerinde kalacaklardır)

CRAFT yönteminde bölümler arası toplam malzeme taşıma maliyetini en aza indirecek şekilde ikili bölüm yer değiştirmeleri gerçekleştirilir. Önce başlangıç tesis yeri düzeninin toplam malzeme maliyeti yük, uzaklık ve taşıma maliyeti verileri dikkate alınarak belirlenir. Daha sonra olası tüm bölüm yer değiştirmeleri saptanarak, en fazla malzeme taşıma maliyeti indirimi sağlayan değiştirme gerçekleştirilir. Bu şekilde hiçbir malzeme taşıma maliyeti indirimi sağlanamama haline kadar yöntem işlemeye devam eder. Her değişiklikte bölümlerin merkez noktası uzaklıkları yeni düzene göre yeniden belirlenir, seçilen malzeme taşıma teçhizatı özelliğine göre (söz konusu teçhizat farklı bölümler arasında farklı özelliklerde olabilir) malzeme taşıma maliyetleri hesaplanır ve yeni bir değişiklik ile toplam malzeme taşıma maliyetinin azalıp azalmadığına bakılır. En düşük toplam malzeme taşıma maliyeti değerine sahip tesis yeri düzeni yöntemin sonucudur. Bölümler arası yer değiştirmeler ikili veya üçlü bir şekilde yapılabilmektedir.

Geliştirme algoritmaları çok sayıda olmalarına karşın hiçbir algoritma, işyeri düzeni tasarımı CRAFT yöntemine üstünlük sağlayamamıştır. Bu yöntem, bölümler arası hareketin doğrusal işlevinin en aza indirilmesi ile ilgilidir.

2.3.3 Sezgisel metodlar

Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (NIST)'nün tanımına göre NP-Zor (NP-Hard) problemler; belirsiz Turing Makinesi ile çok terimli (polinomsal) zamanda çözülebilen karar problemlerini içeren karmaşıklık sınıfıdır. (<http://www.nist.gov/dads/HTML/nphard.html>)

Tesis Yerleşim Problemleri (TYP), kareli optimizasyon problemleri içerisinde önemli bir yere sahip olup, çözüm zamanının, problem boyutuna bağlı olarak üstel artış gösterdiği çözülmesi zor NP-Zor problemlerdendir (Mirchandani ve Francis,1990).

Bu tür problemler, problem boyutunun polinom fonksiyonu ile sınırlı olan hesaplama zamanı içinde çözülemezler. Bu nedenle bu tür problemlerin çözümünde, optimumu bulmayı garanti etmeyen ancak, çözüm zamanı polinom sınırlar içinde kalan sezgisel metotlardan yararlanılır. Sezgisellerin çoğu probleme özgü özelliğe sahiptirler ve bir problem için kullanılan bir sezgisel, başka bir problem için kullanılamamaktadır (Reeves, 1993). Bununla beraber, problemlere uygulanmaları açısından esnek ve daha genel özelliğe sahip olan tekniklere ilgi son yıllarda giderek artmıştır. Bu teknikler çok sayıda problemlere uygulanmışlar ve oldukça güçlü bulunmuşlardır.

NP-Zor problemlerin çözümünde kullanılan sezgisel yaklaşımlar şunlardır:

- Tepe-Tırmanma (T-T) Metodu (Hill Climbing-HC)
- Tavlama Benzetimi (TB) Metodu (Simulated Annealing-SA)
- Tabu araştırması (TA) (Tabu Search-TS)
- Yapay Sinir Ağları (YSA) (Artificial Neural Network- ANN)
- Genetik Algoritmalar (GA) (Genetic Algorithms-GA)
- Karınca Algoritmaları (KA) (Ant Colony Algorithms-ACA)
- Melez Yöntemler

2.3.3.1 Tepe tırmanma metodu (Hill climbing)

Tepe tırmanma (T-T) metodu bir iteratif iyileştirme (yerel arama) yöntemidir. T-T, iniş ya da adım adım iniş stratejisi olarak da adlandırılmaktadır. Bu metodun temelinde, tanımlanan bazı kurallara göre bir çözümünden bir diğer komşu çözüme ulaşma vardır. T-T metodunun

etkinliğinde iyi bir komşuluk yapısı seçiminin önemi büyüktür. İyi, mutlak açıdan en iyi olmak zorunda değildir. T-T'nin zayıf yanı yerel ve genel en iyi arasındaki ayrımı yapamaması sonucu yerel optimumdan kaçamamasıdır. Yerel optimumdan genel optimuma geçebilmek için modern sezgiseller geliştirilmiştir (Yiğit ve Türkbey, 2003)

Kısaca T-T Algoritmasının adımları aşağıdaki gibi gösterilebilir (Yiğit ve Türkbey, 2003):

- Başlangıç çözümü üret; x_0 : mevcut çözüm ve $x_0 \in R$
- Aşağıdaki adımları tekrarla:
 - $N(x_0)$ komşu seti içinden en iyi x' komşusunu seç.
 - $f(x') \leq f(x_0)$ ise x' çözümünü yeni mevcut çözüm olarak ata: $x_{0+1} = x'$
 - Aksi halde dur.

2.3.3.2 Tavlama benzetimi metodu (Simulated annealing)

Tavlama Benzetimi (TB), kareli optimizasyon problemleri için iyi çözümler veren olasılıklı karar verme yöntemi olarak görülebilir. Kontrol parametresi “sıcaklık”tır ve minimizasyon problemleri için çözümün daha iyi bir çözüme ulaşmasının kabulü olasılığını değerlendirir. “Tavlama Benzetimi” ismi, katıların fiziksel tavlama süreci ile olan benzerlikten ileri gelmektedir. TB algoritması, birbirlerinden bağımsız olarak, Kirkpatrick, Gelatt ve Vecchi (1983) ve Cerny (1985) tarafından ortaya konmuştur. Günümüze kadar, bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme gibi farklı alanlardaki bir çok optimizasyon problemine uygulanmıştır (Balakrishnana vd., 2003).

2.3.3.3 Tabu araştırması (Tabu search)

Glover tarafından geliştirilen Tabu araştırması (TA) algoritması önceleri ayrık optimizasyon problemlerine uygulanmıştır (Glover, 1989, 1990). TA'nın zeki bir yapıda olmasının arkasında, belleğinde arama uzayının geçmiş bilgilerini tutma özelliği vardır. Bu sayede arama sınırlandırılmıştır ve diğer algoritmalarda karşılaşılan yerel optimumdan kurtulmaktadır. Brucker'e göre, TA, yerel optimuma düşüşten kaçınmak için bir bellek fonksiyonu kullanıp, küresel optimumu hızlı bir şekilde aramada bir veya daha çok yerel arama prosedürünü hiyerarşik olarak yönlendiren zeki bir tekniktir (Brucker, 1998). Algoritmanın oluşturulmasında temel alınan iki önemli unsur, daha önce denenmiş çözümleri

yeniden işleme almaması ve yerel optimum noktadan uzaklaşıp küresel optimum çözüme ulaşabilmesidir (Hao vd., 1998).

2.3.3.4 Yapay sinir ağları (Artificial neural network)

İnsanlığın doğayı araştırma ve taklit etme çabalarının en son ürünlerinden bir tanesi, yapay sinir ağları (YSA) teknolojisidir. YSA, basit biyolojik sinir sisteminin çalışma şekli simüle edilerek tasarlanan programlama yaklaşımıdır. Simüle edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. YSA'ların bir insanın, düşünme ve gözlemleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğine sahip olmasıdır.

2.3.3.5 Genetik algoritmalar (GA-Genetic algorithms)

Çeşitli kombinatoriyal optimizasyon problemlerine uygulanabilen Genetik algoritmalar (GA), doğal genetik ve doğal arama mekaniğine dayanan stokastik bir arama yöntemidir. Genetik algoritma düşüncesi, doğadaki biyolojik organizmaların evrimsel prosesine dayanmaktadır (Beasley ve Chu, 1995). Evrim alanında, doğal popülasyonlar, doğal arama ve en iyi olanın ayakta kalması prensiplerini içerir. Çevrelerine daha çok uyum gösteren bireyler, hayatta kalmak için daha çok şansa sahip olurken, az uyum gösteren bireyler elenirler. Bu, bireylerle en çok uyum gösteren genlerin sayısının her başarılı nesilde artacağı anlamına gelir. Yüksek uyumlu atalardan iyi özellikler bileşimi, daha uyumlu çocuklar üretebilir. Bu yolla türler, çevrelerine çok daha uyumlu hale getirilmek için geliştirilir.

2.3.3.6 Karınca algoritmaları (Ant colony algorithms)

Karınca Kolonisi Optimizasyonu (Ant Colony Optimization - ACO) birçok NP-Zor kombinatoriyal optimizasyon problem için başarı ile uygulanabilen popülasyon tabanlı bir algoritmadır.

KA ismini, yapısının gerçek karıncaların yemlerini arama davranışlarından esinlenerek oluşturulmasından almaktadır. Algoritmanın ana fikirlerinden bir tanesi koloninin bireyleri (karıncalar) arasında direkt bir iletişimin olmamasıdır. İletişim için gerçek karıncalar kimyasal

bir madde olan fenomen izlerini kullanmaktadır.

Karıncalar gibi neredeyse kör hayvanların kolonisinin beslenme kaynaklarına ulaşmada en kısa yolu bulmayı nasıl yönettikleri, günümüzde biyoloji bilimi tarafından da çalışılan bir problemdir. Koloniler yönle ilişkin bireyler arasında bilgi iletişimini sağlamada ve nereye gideceklerine karar vermede fenomen izlerini bir araç olarak kullanır. Hareket halindeki bir karınca yerin üzerine değişik miktarlarda fenomenler yayar, ve gittiği yolu bu madde vasıtasıyla işaretler. Aslında bir tek karınca rastgele hareket ederken önceden başka karıncalar tarafından yayılan izle karşılaşabilir ve büyük olasılıkla onu takip eder, böylece kendi fenomenini de yola yayarak yol üzerindeki izi güçlendirir. Ortak davranış, izi takip eden ne kadar çok karınca varsa izlenilen yolun istenebilirlik oranının o kadar artmasına neden olur. Böyle bir davranış kendisini güçlendiren bir proses olan otokatalitik davranışın bir çeşidini ortaya koyar. Süreç, bir karıncanın önceden başka karıncalar tarafından seçilen yolu seçmesi olasılığının yüksek olduğu yerde pozitif bir geri besleme mekanizması ile tanımlanmıştır.

Bir çok kombinatorial optimizasyon probleminin çözümü için kullanılabilen, KA ilk kez Dorigo (1992) tarafından ortaya atılmıştır. Dorigo, "Optimization, Learning and Natural Algorithms" isimli doktora tezinde bir karınca kolonisinin iki nokta arasındaki en kısa yolu bulmada kullandığı haberleşme mekanizmasını kombinatorial optimizasyon problemlerine çözüm üretmekte kullanmıştır (Toksarı, 2004).

Algoritmanın ilk uygulamasını Dorigo vd. (1992) GSP'ye uygulamışlardır. Algoritmanın doğruluğunu ispatlayan bu çalışmada yapılan deneysel çalışmalarda fenomen izinin güncellenmesi ve atama olasılıklarında kullanılan parametrelerin en iyi kombinasyonu saptanmıştır (Toksarı, 2004).

2.3.3.7 Melez yöntemler

Son yıllarda literatürde, MKKS (Melez Karınca Kolonisi Sistemi) ve AnTabu (Karınca Algoritması-Tabu Arama), GATB (Genetik Algoritma-Tavlatma Benzetimi) gibi melez algoritmalara sıkça rastlanmaktadır. Melez algoritmaların yapısı genellikle iki zeki sezgisel metottan oluşmaktadır. Bu iki zeki sezgisel metottan bir tanesi yerel çözümleri araştırırken ikincisi bu çözümden optimum çözüme ulaşacak ilerlemeyi sağlar. Melez algoritmaların geliştirilmesindeki amaç kullanılan metotların birbirlerinin eksikliklerini kapatmasıdır. Melez algoritmalar, yapısını oluşturan zeki sezgisellere göre daha iyi sonuçlar vermektedir. Lee vd. (2002) melez genetik algoritmalar üzerine çalışmışlardır. Çalışmalarında genetik

algoritmaların zayıflıklarını, yerel aramada hızlı çözümler bulan TA ve TB algoritmalarının kuvvetli noktalarını kullanarak ortadan kaldırmışlardır (Toksarı, 2004).

2.3.4 Kareli atama problemi (KAP)

Tüm bölümler eşit alanlıysa veya bölümlerin fiziksel yerleri, diğer kalan bölümlerle arasındaki yakınlık veya uzaklık ilişkilerini değiştirmeden değiştirilebiliyorsa bölümlerin işgal edeceği potansiyel yerleri belirlemek kolaydır. Böyle bir durumda, yerleşim problemi, maliyetin akış miktarı ve kat edilen uzaklık bakımından doğrusal olduğu farz edilen Kareli Atama Problemi (KAP) olarak modellenebilir (Tate and Smith, 1995).

Kareli Atama Problemi, 1957 yılında ilk kez Koopmans ve Beckman tarafından ortaya atıldıktan sonra, tesis yerleşimi konusunda çalışan araştırmacıların ilgi odağı olmuştur (Brown vd., 1989). KAP, çok bilinen bir kombinatoriyal optimizasyon problemidir (Tate and Smith, 1995).

Kareli atama problemlerinde amaç fonksiyonu, değişkenlerin ikinci dereceden polinom bir fonksiyonudur. Kareli atama probleminde kısıtlar, atama probleminin kısıtlarına benzer. KAP'de amaç, iş akışı ve seyahat edilen uzaklığın çarpımı olarak ifade edilen malzeme taşıma maliyetini en küçükleyecek şekilde, n tesisin n adet yere optimum atamasını belirlemektir. KAP için çözüm zamanlarının, yerleştirilecek tesis sayısı arttıkça üstel olarak arttığı ispatlanmıştır (Tavakkoli-Moghaddain ve Shayan, 1998). Kareli atama problemi, kesme düzlemi, dal-sınır veya diğer optimizasyon teknikleriyle belli bir sınıra kadar çözülebilmektedir.

TYP, kareli atama problemi olarak ilk defa Armour ve Buffa tarafından 1963 yılında formüle edilmiştir. Bir üretim fabrikasının yerleşimini, minimize edilecek ölçütü bölümler arası ürün akış maliyeti olarak ele alarak incelemişlerdir. Çalışmalarını CRAFT (Computerized Relative Allocation of Facilities) adını verdikleri bir programla sonuçlandırmışlardır (Liggett, 2000).

Kareli atama probleminde her bölümün yeri özel olarak belirlenir. Son çözümde, bölümler arasındaki toplam ağırlıklı uzaklık minimize edilir ve her bölüm sadece bir pozisyona atanarak yerleşim elde edilir. Bölümler ile ilgili ağırlıklar, komşuluk ilişkileri (ilişki çizelgesi -relationship chart-) veya bölümler arasındaki iş akış hacmi (nereden nereye çizelgesi -from to chart-) ile belirlenir. KAP şu şekilde formüle edilir (Brown vd., 1989):

Ele alınan TYP'nin KAP ile çözümü için tesislerin eşit büyüklükte olması zorunlu değildir. Zaten bu pratikte pek mümkün de değildir. Farklı büyüklükteki birimler uygun yerlerden

kesilerek birbirine eşit alanlara bölünür ve KAP uygulama aşamasında bir bölümü oluşturulan parçaların birbirinden ayrılmasına izin verilemeyerek bölümün çözüm esnasında dağılması engellenmiş olur. Bu açıdan bakıldığında KAP'nin uygulama alanı oldukça genişlemektedir (Konak vd., 2006).

$$MinZ = \sum_{i=1}^N \sum_{j=1}^N a_{ij} x_{ij} + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N f_{ik} c_{jl} x_{ij} x_{kl} \quad (2.1)$$

S.T.

$$\sum_{i=1}^N x_{ij} = 1 \quad i = 1, 2, \dots, N$$

$$\sum_{j=1}^N x_{ij} = 1 \quad j = 1, 2, \dots, N$$

$$x_{ij} \in \{0,1\} \quad i, j = 1, 2, \dots, N$$

Notasyonlar:

N = Tesislerin ve yerleşimlerin toplam sayısı,

a_{ij} = i tesisini j yerleşimine atamanın sabit maliyeti,

f_{ik} = i tesisinden k tesisine olan iş akışı,

c_{jl} = j yerleşiminden l yerleşimine birim yük taşıma maliyeti,

x_{ij} = i tesisi j yerleşimine atandıysa 1, aksi halde 0.

Bu modelin (2.1) ilk kısmı bölümleri ilgili pozisyonlara atama maliyetlerini, ikinci kısım ise bölümlerin işlem gördükleri sürede aralarındaki iş akışına bağlı olarak değişken maliyetleri göstermektedir. Bu problemde amaç fonksiyonu karelidir ve konveks olmayan bir yapıya sahiptir. Kareli atama problemi için birden çok yerel optimum çözüm vardır ve çözüm uzayında $n!$ kadar çözüm noktası bulunmaktadır. KAP, NP-tam bir problemdir ve 15-20 tesis için mümkün hesaplama zamanda bilinen hiçbir metot optimum çözümü bulma kabiliyetine sahip değildir (Sha and Chen, 2001). Kareli atama problemi ismi, amaç fonksiyonunun ikinci dereceden polinom bir fonksiyon olması ve sınırlarının atama probleminin sınırlarına benzer olmasından dolayı kullanılmaktadır (Chiang ve Chiang, 1998).

TYP'nin en temel ve sık kullanılan anaetik formülasyonu olan KAP; farklı yerleşim konfigürasyonları buluncaya kadar devam eden klasik uzaklık maliyetleri optimizasayonu modelidir.

Kareli atama problemi, tesis yerleşimi problemlerinin çözümü ile ilgili uygulamaların geliştirilmesinde önemli bir araçtır. KAP'nin tesis yerleşimi problemi gibi NP-tam yapıda olmasından dolayı çözümü için sezgisel yöntemler geliştirilmiştir. Sezgisel çözüm yöntemleri genellikle kareli atama probleminde iyi sonuçlar vermektedir (Rosenblatt, 1986).

KAP çözümü çok zor olduğundan dolayı yeni algoritmaların test edilmesi için uygun bir problemdir. Sadece KAP için elde edilen yeni çözümlerin sorulması ve bu zor alanda yeni daha uygun bir çözüm bulunması standart bir kıyaslamaya olanak sağlamaktadır. Bu geniş alanda sezgisel yöntemler kadar sayısal olmayan stratejiler de çözüm üretebilmektedir (Hahn ve Krarup, 2000).

3. GENETİK ALGORİTMA (GA)

3.1 Genetik Algoritma Nedir?

Genetik algoritma (GA) rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir (Goldberg, 1989).

Genetik algoritmalar, doğadaki canlıların geçirdiği süreci örnek alır ve iyi nesillerin kendi yaşamlarını muhafaza edip, kötü nesillerin yok olması prensibine dayanır. Matematiksel modellemenin yapılamadığı veya kesin çözümün olmadığı problemlerde genetik algoritmadan yararlanılır. Bu algoritma, anne ve baba bireyden (bir önceki nesil) doğan yeni çocuk bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesine dayanır. Yeni bireyler, anne ve babasından gelen iyi genleri bünyelerinde muhafaza edebileceği gibi kötü genleri de almış olabilir. Bu durumda kötü genlere sahip çocuk bireyler varlıklarını sürdüremeyecektir. Genetik algoritmalar doğadaki en iyinin yaşamasını gerektirmekte ve bunu belirleyen uygunluk (fitness) işlevi, yeni çözümler üretmek için çaprazlama, kopyalama ve değiştirme gibi operatörleri kullanır. Genetik algoritmanın önemli özelliklerinden birisi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyiyi seçmesidir (Biroğul, 2005).

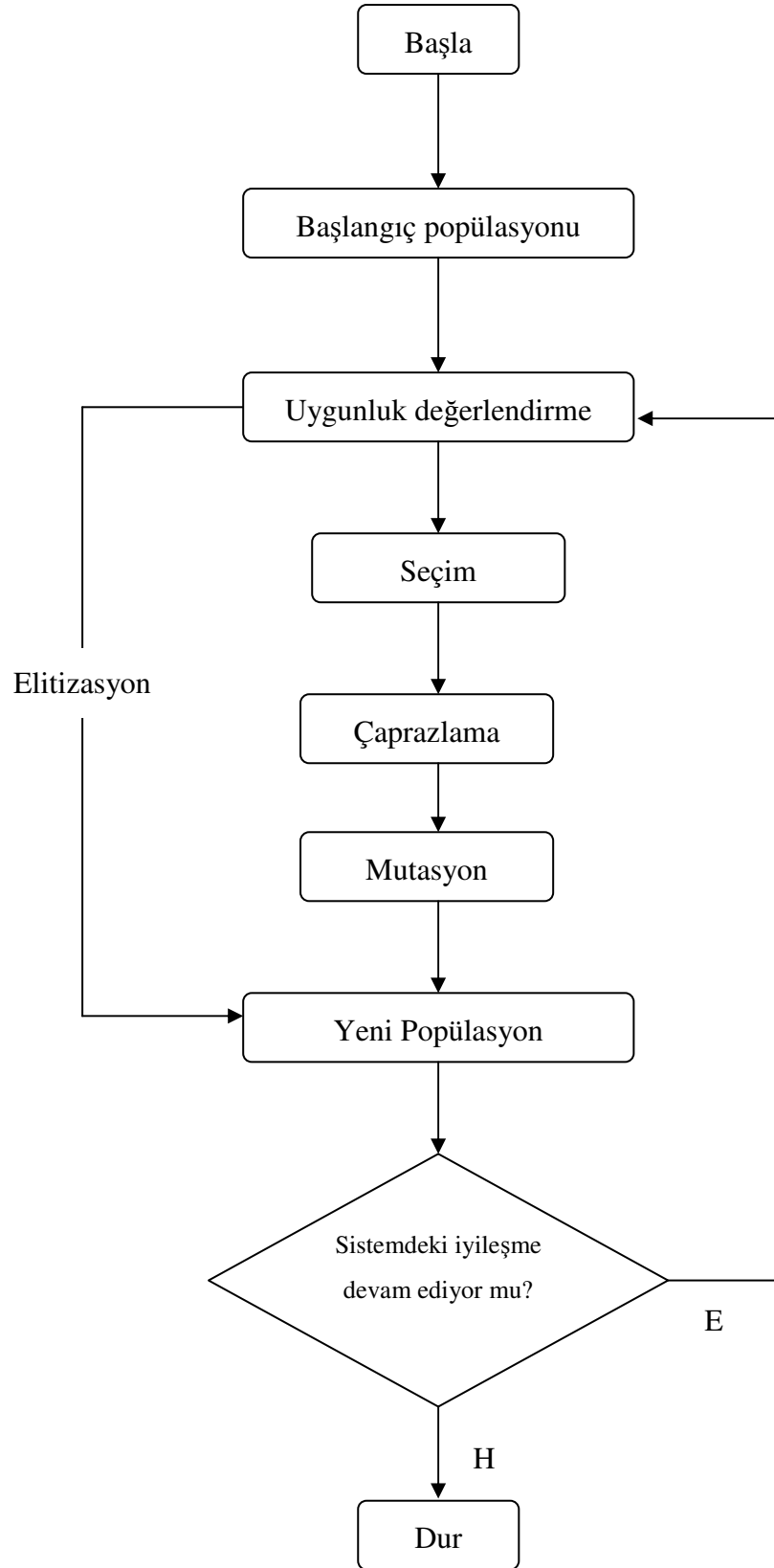
Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılmaktadır. Çok genel anlamda genetik algoritmanın üç uygulama alanı bulunmaktadır. Bunlar deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleridir (Gathercole ve Ross, 1997).

GA'lar, birden fazla (popülasyon büyüklüğü kadar) çözüm üzerinde çalışırlar (Şerifoğlu vd. 2000) . Çözümler, problemin yapısına uygun bir şekilde kodlanmış kromozomlar tarafından temsil edilirler. Genetik algoritma, bireylerin bir başlangıç popülasyonunu alarak ve her yeniden üretimde genetik operatörleri kullanarak bu prosesleri simüle eder. Optimizasyon açısından, popülasyondaki her birey, verilen problemde olası çözümü gösteren bir dizi veya kromozoma kodlanır. Bir bireyin uyumu, verilen amaç fonksiyonuna göre değerlendirilir. Uyum fonksiyonu, kromozomların ifade ettikleri çözümlerin ne derece iyi olduğunu ifade eder. Yüksek uyum gösteren bireyler veya çözümler, çaprazlama prosedüründe diğer yüksek uyumlu bireylerle, genetik bilgilerindeki parçaları değiştirerek yeniden üretilme fırsatını bulurlar. Bu da ebeveynlerin her ikisinden de alınan bazı karakterleri paylaşan yeni çocuk çözümler üretir. Mutasyon çoğu zaman dizilerdeki bazı genleri değiştirerek çaprazlamadan

sonra uygulanır. Çocuk, bütün popülasyonu değiştirmekle elde edilebileceği gibi, az uyumlu özelliklerin değişmesiyle de oluşabilir. Bu, çevrim-seçim-yeniden üretim döngüsü, istenen sayıda kuşak yaratılana veya başka bir durma kriteri doğrulanana kadar tekrarlanır ve yaratılan en iyi çözüm problemin çözümü olarak alınır. Genetik algoritmaların genel akış şeması Şekil 3.1’de gösterilmiştir.

Genetik algoritmayı diğer arama tekniklerinden farklı kılan özellikleri şöyledir (Goldberg, 1989):

- Genetik algoritma, parametrelerin kendileri ile doğrudan ilgilenmez, parametre setlerinin kodları ile ilgilenir,
- Genetik algoritmanın arama alanı, yığının veya popülasyonun tamamıdır; tek nokta veya noktalarda arama yapmaz,
- Genetik algoritmalarda amaç fonksiyonu kullanılır, sapma değerleri veya diğer hata faktörleri kullanılmaz,
- Genetik algoritmaların uygulanmasında kullanılan operatörler stokastik yöntemlere dayanır, deterministik yöntemler kullanılmaz.



Şekil 3.1 Genetik algoritmaların genel akış şeması

3.2 Genetik Algoritma'nın Tarihsel Gelişimi

Evrimsel hesaplama 1960'larda I.Rechenberg'in Evrim Stratejileri ("Evolutionssstrategie") adlı çalışmasında tanıtılmıştır. (Quagliarella vd., 1997)

Rechenberg'in çalışması daha sonra başka araştırmacıların da ilgisini çeker ve araştırmalar bilgisayarla bütünleşik çözümler bulma üzerine yoğunlaşır. Genetik algoritma böylece John Holland tarafından bulunur. Onun öğrencileri ve arkadaşları tarafından da geliştirilir. Holland'ın kitabı "Doğal ve Yapay Sistemlerde Adaptasyon (Adaption in Natural and Artificial Systems)" 1975 yılında yayınlanır. (Roggen, 2005)

Holland ve arkadaşları çalışmalarını iki amaçta toplamışlardır:

- Doğal sistemlerin uygun proseslerini açıklamak,
- Doğal sistemlerin mekaniğini elinde tutan yapay sistemler yazılımını tasarlamak.

Luis (1993)'in belirttiğine göre ilk defa 1960'larda John Holland tarafından ortaya atılan; kendisi, öğrencileri ve meslektaşları tarafından geliştirilen. GA'lar, mühendislik dünyasında birçok alanda genişçe işlenmiş, deneyleri yapılmış ve uygulanmıştır (Çetin, 2002). Holland'ın asıl amacı, adaptasyon olgusunu tabiatta meydana geldiği şekliyle resmi olarak incelemek ve özel problemleri çözmek için algoritmalar düzenlemek yerine, doğal adaptasyon mekanizmalarının bilgisayar sistemlerine aktarılabilceği yollar geliştirmektir (Tanrıseven, 2000),

Lois (1993)'ün belirttiğine göre 1992'de, John Koza genetik algoritma'yı, verilen bazı görevleri gerçekleştirecek programlar geliştirmek için kullandı. Koza, bu yöntemle genetik programlama ("Genetic Programming" - GP) adını vermiştir (Çetin, 2002).

Sonraki yıllarda araştırmalar devam etmiş ve genetik algoritmalar çözüm alternatifleri üstel olarak artan problemlerde başarı ile uygulanmıştır. Günümüzde genetik algoritmalar güçlü arama algoritmaları olarak kendilerini ispatlamıştır. Genetik algoritmaların asıl etkisi, çok çeşitli problemlerde başarıyla uygulanabilmesinden kaynaklanmaktadır.

3.3 Neden GA ? Neden Değil?

Genetik algoritma yaklaşımı, kendi arama prosesini oluşturması bakımından bir çok avantaja sahiptir. Bu özellik GA'yı tipik optimizasyon ve arama tekniklerinden ayırır (Goldberg, 1989). Genetik algoritmaların önemli bir özelliği, aramanın tek bir nokta yerine noktaların kümesi tarafından yönetilmesidir ki bu da genetik algoritmaların aramadaki yeteneğini ortaya koyar. Buna karşın, fonksiyonun farklılaştırılması gibi ek bilgi gerektiren eğimli arama tekniklerine göre GA, amaç fonksiyonunu veya cezayı direk olarak kullanır. Böylece aday ürün profilleri belirtilen amaca göre değerlendirilir. Genetik algoritmaların diğer bir özelliği, parametrelerin kendileri yerine, parametre kümesinin direk kodlanması ile çalışmasıdır. Bu da, eğimli veya tesadüfi arama tekniklerinin tersine, sürekli olmayan, büyük-boyutlu, çok modellen optimizasyon problemleri alanına özellikle genetik algoritmaların uyması anlamına gelmektedir.

Shaffer (1998)'a göre GA'nın avantajları şöyle sıralanabilir (Kulluk, 2003):

- Genetik algoritmalar, çözüm uzayı hakkında herhangi bir bilgiye ihtiyaç duymazlar,
- Çözüm uzayını oluşturan bölümler arasındaki devamsızlık ve kopuklukların genel optimizasyon performansı üzerinde çok az etkisi vardır,
- Yerel optimum noktada takılı kalmaya direnç gösterirler,
- Büyük ölçekli optimizasyon problemlerinin çözümünde iyi sonuç verirler,
- Çeşitli optimizasyon probleminin çözümünde kullanılabilirler.

Shaffer (1998) bütün bu avantajlarına karşılık GA 'nın şu dezavantajlara sahip olduğunu da belirtmiştir (Kulluk, 2003):

- Çok fazla uyum fonksiyonu hesaplaması gerektirir,
- Global olarak tam optimum noktanın saptanmasında problem yaşanmaktadır,
- Kullanılan konfigürasyon genel değil, probleme özgü bir yapı taşımaktadır,
- Nesiller arasında iyi özelliklerin kaybolma ihtimali vardır.

Genetik algoritmalar, optimizasyon problemlerinde sonlu bir alfabe ile sonlu sayıda diziler şeklinde kodlanmış parametre kümelerine ihtiyaç duyarlar. GA'lar, kodlama şekillerinden

yararlanmaktadır. Bu da diğer metotlardaki kısıtları (süreklilik, türevlerin varlığı v.b.) elimine etmektedir.

Bir çok optimizasyon metodunda, karar kümesindeki bir noktadan hareketle bazı kurallar kullanılarak, başka bir noktaya ulaşmak şeklinde bir yöntem kullanılır. Bu metot birden fazla ekstramum noktası bulunan problemlerde, yanlış noktaya yönelme gibi tehlikeler taşır. GA bir dizi popülasyonu ile çalıştığından birden çok ekstramum noktaya paralel olarak ulaşabilmekte, bunun sonucu olarak da yanlış bir noktaya yönelme ihtimali, bir noktadan diğerine hareket eden diğer yöntemlere nazaran çok daha düşük olmaktadır.

3.4 Genetik Algoritmanın Çalışma Prensibi

Ana hatlarıyla genetik algoritma şöyle özetlenebilir (Çetin, 2002):

1. **[Başlat]** Problem için rastgele n kromozomlu popülasyon oluşturulur.

Problem için n tane çözüm önerilir.

2. **[Fitness]** Popülasyondaki her bir x kromozomu için $f(x)$ (fitness) fonksiyonu hesaplanır.
3. **[Yeni popülasyon]** Yeni bir popülasyon oluşuncaya kadar aşağıdaki adımlar tekrar edilir:

- a. **[Seleksiyon]** Fitness durumuna göre popülasyondan iki tane kromozom çaprazlanmak (crossover) amacıyla seçilir. Fitness derecesi yüksek olanın seçilme şansı yüksektir.
- b. **[Çaprazlama]** Seçilmiş olan ebeveyn kromozomlar, çaprazlama oranına (crossover probability) göre yeni yavrular oluşturmak üzere çaprazlanırlar. Eğer çaprazlama uygulanmazsa bireyler atalarının tamamen kopyası olacaklardır
- c. **[Mutasyon]** Kromozom üzerindeki bazı stringlerin (DNA dizilerinin) yerleri ile oynanarak belirli mutasyon oranına göre değişiklikler yapılır.
- d. **[Kabul]** Oluşturulan yeni bireyler yardımıyla yeniden bir popülasyon oluşturulur

4. **[Değiştirme]** Oluşturulan yeni popülasyon eskileriyle yer değiştirilir.

5. **[Test]** Programı bitirme şartı gerçekleşiyorsa program durdurulur ve popülasyondaki en iyi çözüm olarak alınır.
6. **[Döngü]** Aksi takdirde ikinci adım tekrarlanır.

Croce'a (1995) göre GA ile çözüme gidilirken takip edilmesi gereken adımlar aşağıda belirtilmiştir (Biroğul, 2005):

- Arama uzayındaki bütün muhtemel çözümler dizi olarak kodlanır. Bu diziyi (kromozomu) oluşturan her bir elemana gen denir. Her bir dizi, arama uzayında belirli bir bölgeyi temsil eder.
- Genellikle rastgele bir çözüm seti belirlenir ve bu başlangıç yığını olarak seçilir.
- Her bir dizi için uygunluk değeri hesaplanır. Bulunan bu uygunluk değeri dizinin (çözümün) kalitesini gösterir.
- Bir grup dizi belirli bir olasılık değerine göre rastgele seçilip üreme işlemi gerçekleştirilir.
- Üreme işlemlerinde, çeşitli genetik operatörler kullanılır.
- Yeni bireylerin uygunluk değeri hesaplanır, ve belirlenen durumlara göre çaprazlama ve/veya değişim işlemlerine tabi tutulur.
- Önceden belirlenen nesil sayısı boyunca veya durdurma kriterleri sağlanıncaya kadar yukarıdaki işlemlere devam edilir.
- Belirlenen döngü veya durma kriteri sağlandığında, oluşan son nesilden en iyi uygunluk değerine sahip birey çözüm olarak seçilir.

3.4.1 Temel kavramlar

3.4.1.1 Gen

Kromozom yapısında kendi başına birer genetik bilgi taşıyan en ufak yapı birimine gen denir. Kısmi bilgiler taşıyan bu ufak yapıların bir araya gelmesiyle bütün bir çözüm kümesini oluşturan kromozom (dizi) meydana gelir. GA'nın kullanıldığı programlama yapısında bu gen yapıları programcının tanımlamasına bağlıdır. Bir genin içerdiği bilgi sadece ikili tabandaki sayıları içerebileceği gibi onluk taban ve onaltılık tabandaki sayı değerlerini de içerebilir. Dolayısıyla yazılan programa göre gen içeriği çok önem kazanmaktadır.

3.4.1.2 Kromozom

Bir yada birden fazla gen yapısının bir araya gelerek oluşturduğu, problemin çözümüne ait tüm bilgiyi içeren dizilere kromozom denir. Kromozomların bir araya gelmesiyle popülasyon (yığın) oluşturulur. Yığındaki her bir bireye kromozom, kromozomdaki her bir bilgiye gen denir. Kromozomlar, üzerinde durulan problemin olası çözüm bilgilerini içermektedir.

Kromozomlar, GA yaklaşımında üzerinde durulan en önemli birim olduğu için bilgisayar ortamında iyi ifade edilmesi gerekir. Kromozomun hangi kısmının ne anlam taşıyacağı, ne tür bilgi içereceği kullanıcının olaya bakışını değiştirir.

3.4.1.3 Popülasyon (Yığın, kütle)

Popülasyon, çözüm bilgilerini içeren kromozomların bir araya gelmesiyle oluşan olası çözüm yığına denir. Yığındaki kromozom sayısı sabit olup problemin özelliğine göre programlayıcı tarafından belirlenir. GA'nın işleyişi esnasında bu yığın kümesinden bir takım kromozomlar yok olmakta ve yerlerine yeni kromozom yapıları eklenerek popülasyon büyüklüğü sabitlenmektedir.

Yığın büyüklüğü, problemin çözüm süresini etkilemektedir. Fazla sayıdaki kromozom yığını problemin çözüm süresini uzatırken, az sayıdaki yığın çözüm değerlerine ulaşamamasına ya da sistemin belirli çözüm uzayında takılıp iyileşememesine neden olabilir. Problemin özelliğine göre seçilecek olan yığın sayısı programcı tarafından iyi belirlenmelidir.

3.4.1.4 Yeniden üretim işlemi

Mevcut yığından gelecek yığına aktarılacak olan dizilerin seçilme işlemidir. Taşınan diziler, genetik olarak mevcut yığında en uygun yapıya (değere) sahip olan dizilerdir.

Bu işlem belirlenen uygunluk değerlerine sahip iyi bireylerin bir sonraki nesile aktarılmasını sağlar.

3.4.1.5 Başlangıç yığınının oluşturulması

GA'yı diğer sezgisel arama metotlarından ayıran bir özellik; çözümü noktadan noktaya değil, noktaların oluşturduğu yığın içinde aramasıdır. Bu nedenle GA'nın ilk adımı başlangıç yığınının oluşturulmasıdır. Genelde başlangıç yığını rastgele oluşturulur. Ancak bu olay kısıtlı en iyileme problemlerinde yığının uygun olmayan çözümlere doğru yönelmesine sebep olabilir. Bu durumu ortadan kaldırmak için probleme özgün çeşitli sezgisel yöntemler

geliştirilebilir.

3.4.1.6 Uygunluk değeri

Kromozomların, çözümde gösterdikleri başarı derecesini belirleyen bir değerlendirme işlevidir. Hangi kromozomların (dizi) bir sonraki nesile taşınacağı ve hangi kromozomların yok olacağı uygunluk değerlerinin büyüklüğüne göre karar verilir. Uygunluk değeri, yığındaki dizilerin bir değerlendirme işlevi yardımıyla hesaplanır. GA'da kullanılan değerlendirme işlevi problemin amaç işlevini oluşturur (Fang, 1992).

GA'nın işleyişi anlatılırken literatürde kullanılan semboller ve anlamları aşağıda belirtilmiştir:

n : Bir yığındaki toplam kromozom sayısıdır.

l : Yığındaki bir kromozomun uzunluğudur.

gen : Bir kromozomdaki her bir hücre

f_i : Yığındaki i . kromozomun uygunluğu

$f = \sum_{i=1}^n f_i$: Yığının toplam uygunluk değeri

$f_{ort} = \frac{f}{n}$: Yığının ortalama uygunluk değeri

C^* : Yığındaki en iyi birey

U_{max} : Verilen bir parametrenin üst sınırı

U_{min} : Verilen bir parametrenin alt sınırı

G_{max} : Maksimum iterasyon sayısı

F : Amaç fonksiyonu

Çoklama problemi için başarı ölçütü genellikle uygunluk işlevi olarak kullanılabilir. Ancak çizelgeleme problemleri genellikle azaltma problemleri olduğundan uygunluk işlevi farklı bir metodla bulunabilir. C_{max} en son işin tamamlandığı sürenin azaltılması probleminde uygunluk değeri şu şekilde hesaplanabilir.

$$f(s_i(t)) = \max\{C(s_i(t))\} - C(s_i(t)) \quad i = 1, 2, \dots, n \quad (3.1)$$

(3.1) incelendiğinde, en önce yığındaki bütün diziler için C_{max} değeri bulunur. Daha sonra en büyük C_{max} değeri saptanır ve her dizinin C_{max} değerinin bu en büyük C_{max} değerinden sapması o dizinin uygunluk değeri olarak hesaplanır.

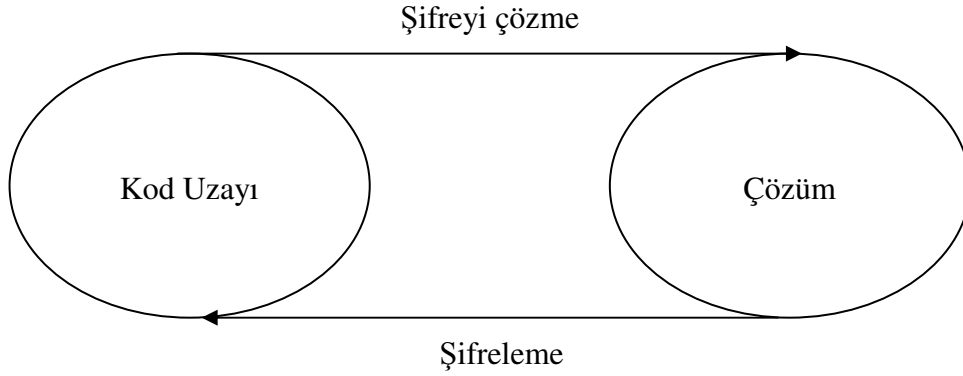
3.4.1.7 Genetik operatörlerin uygulanacağı dizilerin seçilmesi

(3.2) eşitliği kullanılarak hesaplanan uygunluk değerlerine göre daha iyi uygunluk değerine sahip dizilerin seçilme ihtimalleri daha fazladır. t . nesildeki i . dizinin seçilme olasılığı $P(s_i(t))$ olarak gösterilirse eşitlik aşağıdaki gibi ifade edilir:

$$P(s_i(t)) = f(s_i(t))t / \sum f \quad (3.2)$$

3.4.2 Kodlama (Dizi gösterimi)

Genetik algoritmaların en temel özelliği, çözüm uzayı ve kod uzayında çalışmasıdır. Kod ve çözüm uzayı arasındaki etkileşim aşağıda, Şekil 3.2'de gösterilmiştir.



Şekil 3.2 Kod ve çözüm uzayı (Cheng vd., 1996)

Çözümün kromozomlara nasıl kodlanacağı genetik algoritmada anahtar niteliği taşır. Son on yıl boyunca değişik kodlama teknikleri geliştirilmiştir. Kromozomların kodlanması aşamasında dikkat edilmesi gereken üç önemli nokta vardır (Cheng vd., 1996). Bunlar;

- Kromozomun uygunluğu (Feasibility) kontrol edilir,
- Kromozomun belirtilen şartları (Constraints) sağlayıp sağlamadığı (Legality) kontrol edilir.

- Kodlama haritasında, kromozomun tek olduğu kontrol edilir.

Kromozomun uygunluğu ile şartları sağlayıp sağlamadığı işlemleri Şekil 3.2'de belirtilmiştir (Cheng vd., 1996).

3.4.2.1 İkili kodlama (Binary encoding)

İkili düzende kodlamanın önemi, ikili dizilere genetik operatörler uygulandığında uygun çözümünün garanti olmasıdır. Uygun olmayan çözümlerle ilgilenmekte iki yol vardır. Birinci yol, uyumun görünüşünü değiştirmeden uygun olmayan çözümlerin uyumunu cezalandırmak için ceza fonksiyonu oluşturmaktır. İkinci yol ise, uygun olmayan çözümü, uygun çözüme dönüştüren sezgisel operatörler tasarlamaktır. Genelde ceza fonksiyonunu belirlemek zor bir iş olduğu için ikinci yol tercih edilmektedir.

En yaygın kullanılan kodlama yöntemidir. Sayıların ikili sistemde gösterimine dayanır. Bu yöntemde kromozomlar sıfır ve birlerden oluşan dizilerdir. Her parametre için n-bit kullanılır, n, her parametre için farklı olabilir.

İkili düzende kodlama, çok sık kullanılan bir kodlama tipi olmasına rağmen bazı sakıncaları vardır. Örneğin, çok değişkenli fonksiyon eniyilemesi için değişkenlerin alt ve üst sınırlarına bağlı olarak elde edilen dizi çok uzun olabilir. Aynı zamanda gezgin satıcı, çizelgeleme, kareli atama gibi kombinatoriyal eniyileme problemlerinde ikili düzende kodlama, araştırma uzayını tam olarak temsil edememektedir. Bu nedenle literatürde, permütasyonlu kodlama (sayısal kodlama) daha sık kullanılmaktadır.

3.4.2.2 Permütasyonlu kodlama (Permutation encoding)

Sıralama problemlerinde permütasyonlu kodlama kullanılır. Bu kodlamada, her kromozom, dizideki bir sırayı temsil eden sayılardan oluşur.

Örneğin, gezici satıcı probleminde (travelling salesman problem) "verilen şehirler arasındaki uzaklıklar bilinirken, bütün şehirleri birleştiren minimum yol uzunluğu ne olmalıdır?" sorusuna cevap aranmaktadır. Burada kromozomlar, şehirlerin hangi sırada ziyaret edilebileceğini gösteren permütasyonlar şeklinde kurulurlar.

3.4.3 Başlangıç popülasyonunun oluşturulması

GA'nın uygulamasında ilk adım, kromozomların bir başlangıç popülasyonunun oluşturulmasıdır. Başlangıç popülasyonu, çoğunlukla rastgele olarak oluşturulur. Ancak,

özellikle kısıtlı optimizasyon problemlerinde, rastgelelik, uygun olmayan (infeasible) çözümlere neden olabilir. Bu durumdan kaçınmak için, genellikle incelenen probleme özgü sezgisel yöntemlerden yararlanılır. Örneğin, Grefenstette 1987'de yaptığı çalışmada, gezgin satıcı problemi için greedy sezgisellerinden; Kapsalis ve arkadaşları 1993'de yaptığı çalışmada, Steiner ağaç problemi için minimum ağaç yaklaşımından; Thiel ve Vass 1994'de yaptığı çalışmada, 0-1 sırt çantası problemi için ekleme çıkarma sezgiselinden; Chen ve arkadaşları ise 1995'de çizelgeleme problemi için Campbell-Dudek-Smith ve Dannenbrig sezgisellerinden yararlanarak başlangıç popülasyonunu oluşturmuşlardır (Çetin, 2002).

GA, bir popülasyonu değerlendirir ve uygulanan bir süreç sonunda yeni bir popülasyon elde eder. Birbirini izleyen her popülasyona, kuşak denir.

Yeniay (1999)'ın ifadesine göre GA'nın herhangi bir uygulamasında, popülasyon genişliğinin (N) belirlenmesi gerekir. Büyük popülasyonlarda, çözüm uzayı iyi örneklendiği için aramanın etkinliği artar, ancak makul bir sürede yüksek kalitede çözüme yeterince ulaşmadan ağır bir hesaplama külfetiyle karşılaşılabilir. Küçük popülasyonlar ise çözüm uzayını yeterli bir biçimde örnekleyememe riskini taşır ve zamansız yakınsama (premature convergence) görülebilir. Popülasyon genişliği için, 50-100 arasında kalan değerler yaygın olarak kullanılmaktadır. Ancak 30 diziye sahip popülasyonlar ile tatmin edici sonuçlara ulaşan birçok uygulama da vardır (Çetin, 2002).

Başlangıç popülasyonunun kalitesi, son popülasyonun kalitesini etkileyen başlıca faktörlerden biridir. Literatürde başlangıç popülasyonu genel olarak rastgele şekilde oluşturulmaktadır. Ancak, kısıtlı optimizasyon problemlerinde başlangıç popülasyonun rastgele oluşturulması sonucunda uygun olmayan çözümler ortaya çıkabilmektedir. Bu durumu yok etmek için sezgisel metotlardan yararlanılmaktadır.

3.4.4 Uyum fonksiyonu

Uygunluk değeri, yeni yığına taşınacak dizilerin belirlenmesinde kullanılan bir araçtır. Bu nedenle, algoritmanın her çevriminde, yığındaki dizilerin bir değerlendirme fonksiyonu yardımıyla uygunluk değerleri hesaplanır. GA'da kullanılan değerlendirme fonksiyonu problemin amaç fonksiyonudur (Dengiz ve Altıparmak, 1998).

Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunmasına ise hesaplama (evaluation) adı verilir. Bu fonksiyon, genetik algoritmanın beynini oluşturmaktadır. Genetik algorithmada

probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte (decoding), sonra bu parametrelere göre hesaplama yaparak kromozomların uygunluğunu bulmaktadır. Çoğu zaman genetik algoritmanın başarısı, bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır.

GA ile çözülecek her problem için uyum fonksiyonu (fitness function) gerekir. Belirli bir kromozom verildiğinde, uyum fonksiyonu, o kromozom ile temsil edilen bireyin yararı (utility) ya da gücüyle (ability) orantılı olan bir sayısal değer verir. Uyum fonksiyonu, çevre rolünü oynamaktadır. GA, bir kromozomun uyum fonksiyonunu, o kromozomun çevreye uyum derecesini incelemeye ve istenmeyen kromozomların yaşama ve üreme olasılığını azaltmada kullanır. Böylece algoritma, başarısız olan aday çözümlerin elenmesini sağlar. Bu nedenle, uyum fonksiyonunun seçimi, algoritmanın etkinliği üzerinde büyük öneme sahiptir. Çoğu problemde uyum fonksiyonuna karar vermek kolaydır.

3.4.5 Yeniden üretim (Seçim)

Kodlamaya karar verilip başlangıç popülasyonu oluşturulduktan sonra, yeni popülasyona geçebilmek için seçimin nasıl yapılacağı belirlenmelidir. Seçme, bireylerin üreme için seçildiği bir işlemdir. İlke olarak, yüksek uyuma sahip bireyler daha büyük bir olasılıkla seçilmelidir. Seçim operatörünün, yüksek uyuma sahip bireylerin sonraki kuşağa kopyalanma olasılığını arttırması istenir. Böylece seçim, arama uzayının başarılı olacağı umulan bölgelerinde yoğunlaşır. Sonraki kuşak için, yavru meydana getirecek bireylerin popülasyondan nasıl seçileceği ve her bir bireyin kaçar tane yavruya sahip olacağı belirlenmelidir. Sonraki kuşak için yavru meydana getirecek bireyler ile üreme havuzu (mating pool) adı verilen bir ara popülasyon oluşturulur. Daha sonra, üreme için bu havuzdan rastgele bireyler seçilir. Seçimde iki önemli konu vardır: Popülasyon çeşitliliği (population diversity) ve seçicilik baskısı (selective pressure). Bu faktörler birbirleriyle ilişkilidir. Seçicilik baskısındaki bir artış, popülasyonun çeşitliliğini azaltır. Bunun aksi de doğrudur. Yani güçlü seçicilik baskısı, GA'nın zamansız yakınsamasını destekler, zayıf seçici baskı ise aramanın etkinliğini azaltabilir. Bu nedenle bu iki faktör arasında bir denge kurulmalıdır. Örneklem mekanizmaları bu hedefi gerçekleştirmeye çalışır.

Çizelge 3.1 Yeniden üretim mekanizmaları

Mekanizmalar	Açıklama
Orantılı yeniden üretim mekanizması	Yığındaki her bireyin seçilme olasılıkları belirlenir ve bu olasılıklar kullanılarak bir sonraki yığın oluşturulur. Bu grupta bulunan yeniden üretim mekanizmaları: rulet çemberi, stokastik artan ve stokastik universal metodudur.
Sıralı yeniden üretim mekanizması	Yığındaki bireyler, uygunluk değerlerine göre küçükten büyüğe doğru sıralanır. En iyiden başlayarak bir azalan fonksiyon yardımı ile dizilere kopya sayısı atanır ve orantılı seçim mekanizmalarından birisi kullanılarak yeni yığın elde edilir.
Turnuva yeniden üretim mekanizması	Yığından rastgele olarak bir grup dizi (yerine koyarak/yerine koymadan) seçilir ve grup içindeki en iyi uygunluk değerine sahip dizi, yeni yığına kopyalanır. Bu işleme, yığın genişliğine ulaşınca kadar devam edilir. Grup genişliği en az ikidir.
Denge durum yeniden üretim mekanizması	Doğrusal sıralı seçim mekanizması kullanılarak seçilen bir ya da iki bireye genetik operatörler uygulanır. Elde edilen yeni diziler, mevcut yığındaki uygunluk değeri en küçük diziler ile yer değiştirerek yeni yığını oluştururlar.

GA literatüründe çeşitli seçim planları önerilmiştir. Holland'ın orjinal GA'sında popülasyon seçimi, en iyi uyum yapanın yaşaması ilkesine dayanır. GA'da uygun bir dizi çok sayıda yavru elde eder ve bunun sonucunda, sonraki kuşakta daha yüksek bir yaşama şansına sahip olur. Mevcut popülasyondan bireyleri uygunluk değerlerine göre seçen yöntemler, uyuma orantılı seçim (fitness proportionate selection) yöntemleri grubu olarak adlandırılırlar.

Goldberg ve Deb (1991) literatürde mevcut ve çok sık kullanılan seçim mekanizmalarını Çizelge 3.1'de görüldüğü gibi, orantılı yeniden üretim (proportionate reproduction) mekanizması, sıralı (ranking), turnuva (tournament), denge durumu (steady state, diğer adıyla Genitor) yeniden üretim mekanizmaları olmak üzere 4 ana sınıfta toplamışlardır (Çetin, 2002).

Literatürde, orantılı, sıralı ve turnuva yeniden üretim mekanizmaları ile birlikte elitist stratejisinin de çok sık kullanıldığı görülmektedir. Elitist stratejisi ile mevcut yığındaki eniyi bir ya da birkaç dizi bir sonraki yığına taşınır. Amaç, elde edilen eniyi uygunluk değerine sahip dizinin (veya dizilerin) örnekleme hatası ya da genetik operatörler kullanımı sonucunda kaybolmasını önlemektir (Dengiz ve Altıparmak, 1998).

3.4.5.1 Orantılı yeniden üretim mekanizmaları

Yerine koyarak stokastik örnekleme (Rulet çemberi seçimi)

Yerine koyarak stokastik örnekleme, rulet çemberi (roulette wheel) seçimi olarak da adlandırılmaktadır. Bu metot ilk defa Holland tarafından ortaya çıkarıldı ve muhtemelen genetik algorithmada kullanılan en eski ihtimale dayanan seçim metodudur. Muhtemel seçim metodları, hayatta kalma ihtimali üzerine kurulan her bir kromozom kopyasının gerçek sayısını belirler. Böylece seçim bölümü her bir kromozomun beklenen değerinin kararlaştırılması ve bu değer döl miktarına dönüştürülmesinden oluşur. Ana düşünce, her bir kromozomun seçilme ihtimalinin uygunluk değeriyle orantılı olduğudur. f_k uygunluk (fitness) değerindeki her bir kromozomun seçilme olasılığı (ihtimali) P_k dır. Öncelikle

$$P_k = \frac{f_k}{\sum_{k=1}^N f_k}$$

eşitliği yardımıyla her bir bireyin P_k seçilme olasılığı hesaplanır. Bu sayılar bir

tabloda tutulur. Tablodaki olasılık değerleri birbirine eklenerek rastgele bir sayıya kadar ilerlenir. Bu rastgele sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu birey seçilmiş olur. Bu işlem N kez tekrar edilir. Bu yönteme rulet çemberi seçimi ismi, bir daireyi, bireylerin uyum değerlerine orantılı olarak dilimleyip çevirdiğimizde olacakların benzeşimi olduğu için verilmiştir.

Rulet tekerleği seçimini adım adım aşağıdaki gibi yazabiliriz:

- Tüm bireylerin uygunluk değerleri bir tabloya yazılır,
- Bu değerler toplanır,
- Tüm bireylerin uygunluk değerleri toplama bölünerek [0,1] aralığında sayılar elde edilir. Bu sayılar bireylerin seçilme olasılıklarıdır. Sayıların hepsi bir tabloda tutulur,
- Seçilme olasılıklarını tuttuğumuz tablodaki sayılar birbirine eklenerek rastgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu

çözüm seçilmiş olur.

Rulet çemberinde stokastik hatalar nedeniyle bir bireye pay edilen yavru sayısı, beklenen yavru sayısından önemli ölçüde farklı olabilir. Bu nedenle rulet çemberi seçimi büyük örnekleme hataları doğurabilmektedir. İterasyon sayısı arttıkça, örnekleme hatası da artmaktadır. Pay edilen yavru sayısı, sadece çok büyük popülasyonlar için beklenen yavru sayısına yaklaşmaktadır. Rulet çemberi seçimindeki bu örnekleme hatası nedeniyle, arama tahmin edilenden farklı yönlerde devam etmektedir. Bu ise algoritmanın muhtemelen yerel optimuma zamansız yakınsamasına neden olmaktadır. Bu sorunu ortadan kaldırmak için çeşitli seçim yöntemleri önerilmiştir.

Yerine koymadan stokastik örnekleme

Yerine koymadan stokastik örnekleme, De Jong'un beklenen değer modelinin öteki adıdır (Çetin, 2002). Bu yöntem, rulet çemberi seçiminin stokastik hatalarını azaltmaktadır. f_k , k.

bireyin uyum değerini; $\sum_{k=1}^N f_k$, popülasyonun toplam uyumunu ve $\bar{f} = \frac{\sum_{k=1}^N f_k}{N}$, popülasyonun

ortalama uyum değerini göstermektedir. Her birey için $\frac{f_k}{\bar{f}}$ beklenen yavru sayısı hesaplanır.

Bu sayı, birey çaprazlamaya her seçildiğinde 0.5, çaprazlama olmadan üremeye seçildiğinde ise 1 azaltılır.

Kalanı stokastik örnekleme

Kalanı stokastik örnekleme yöntemi, yerine koyarak ve koymadan uygulanabilir. Yöntem, beklenen birey sayılarını bilinen biçimde hesaplar ve tamsayı kısmını bireye atar. Popülasyon genişliğine ulaşılmadıysa, beklenen değerlerin kesirli kısımlarından yararlanır. Yerine koyarak kalanı stokastik örneklemede, beklenen değerlerin kesirli kısımları, rulet çemberi seçim yönteminde bireylere pay edilen ağırlıkları hesaplamak için kullanılır. Yerine koymadan kalanı stokastik örneklemede, beklenen değerlerin kesirli kısımları kullanılarak birer birer ağırlıklı para atışları (Bernoulli denemeleri) yapılır. Örneğin, beklenen kopya sayısı 1,36 olan bir birey, bir kopyayı kesin olarak, diğer kopyayı ise 0,36 olasılıkla elde eder. Popülasyon tamamlanana dek bu süreç devam eder.

Stokastik evrensel örnekleme

James Baker, bireye atanan gerçek yavru sayısı ile beklenen yavru sayısı arasındaki farkı

minimize eden stokastik evrensel örnekleme adında bir yöntem önermiştir (Çetin, 2002). Temel düşünce tek bir seçimle tüm N bireyi örnekleme. Bu yöntemi gerçekleştirmek için, rulet çemberine seçim çemberi (selection wheel) denilen bir ek parça eklenir. Bu parça eşit aralıklı N göstergeye sahiptir. Çember bir kez çevrilir ve durduğunda göstergelerin bulunduğu yerler bireyleri gösterir. Böylece N birey, bir adımda seçilir. Göstergeler eşit aralıklı olduğu için, bir bireyin küçük popülasyonlarda bile tüm popülasyona hakim olma tehlikesi yoktur. Bu yöntemle, her bireyin beklenen yavru sayısı kadar (daha çok değil) üremesi garanti edilir. Bu algoritma ile seçim planının belirli bir işleyişinin sonucu beklenen davranışa mümkün olduğunca yakındır, yani ortalama değişim minimumdur.

3.4.5.2 Sıralı seçim mekanizması

Sıralama seçimi, ilk olarak Baker tarafından uyuma orantılı seçimin dezavantajlarını yok etmek üzere önerilmiştir. Sıralama seçiminde bireyler (kromozomlar) uyum değerlerine göre (fitness durumlarına göre) sıralanır. Her bireyin beklenen değeri, sahip olduğu sıraya bağlıdır.

Bu sıra dikkate alınarak seçme yapılır. Bu seçme işleminde tüm kromozomların seçilme ihtimali mevcuttur. Fakat bu yöntem çok farklı olmayan bireylerin seçilmesine daha çok imkan tanıyacağından istenen tarama bölgesini çabuk taramayabilir.

3.4.5.3 Turnuva seçim mekanizması

Turnuva seçim yöntemi, seçim baskısı yönünden sıralama seçimlerine benzer, ancak işlemsel olarak daha verimli olup, paralel uygulamaya daha yatkındır. Turnuva seçiminde, popülasyondan yerine koyarak ya da yerine koymadan rastgele t birey seçilir, t büyüklüğüne turnuva genişliği adı verilir. Bu gruptaki en iyi birey, yeni popülasyona kopyalanır. Bu işlem N kez yinelenir. Büyük t değeri, yöntemin seçicilik baskısını artırır. Turnuvalarda çoğunlukla $t=2$ alınır ve ikili turnuva olarak adlandırılır. Bu yöntemde zamansız yakınsama, durağanlaşma ve açık uyuma gereksinim yoktur.

Stokastik turnuva seçimi, Wetzel tarafından önerilmiştir (Çetin, 2002). Bu yöntemde seçim olasılıkları hesaplanır ve rulet çemberi yardımıyla ard arda birey çiftleri belirlenir. Bir çift belirlendikten sonra, yüksek uyuma sahip olanı alınarak yeni popülasyona dahil edilir ve diğer çift seçilir. Süreç, popülasyon tamamlanana dek sürer.

3.4.5.4 Denge durumu seçim mekanizması

GA'nın her iterasyonunda mevcut popülasyondan bütünüyle yeni bir popülasyon

yaratılmaktadır. Bütün popülasyonu değiştiren GA'lara kuşaksal (generational) GA'lar denir. Bazı seçim planlarında (elitist plan gibi) birbirlerini izleyen kuşaklar bir derece örtüşmekte, yani önceki kuşağın bir bölümü yeni popülasyonda korunmaktadır. Her kuşakta popülasyonun değiştirilen oranına kuşak aralığı (generation gap) denir ve G ile gösterilir.

Böylece t . kuşağın $N(1-G)$ adet bireyi, $(t+1)$. kuşakta aynen yaşamaya devam eder. Kuşaksal eenetik algoritmelerde $G=1$ 'dir.

Denge durumu (steady state) seçiminde, her kuşakta sadece birkaç birey değiştirilir. Bu seçimde, ata olarak iki bireyin nasıl seçileceğine ve gelecek yavrulara yer açmak için hangi iki bireyin silineceğine karar verilir. Örneğin; Whitley'in GENITOR algoritması ataları sıralanan uyum skorlarına göre seçmekte ve yavrular en kötü iki bireyin yerini almaktadır (Çetin, 2002).

3.4.6 Genetik operatörler

GA'da çözüm yığını incelenirken belirli noktalardan sonra nesil çeşitliliği olmadığı için çözüme gidilememektedir. Nesil çeşitliliğini sağlayarak, çözüm uzayında algoritma istenen kısıtları sağlayacak olan çözüm yığına ulaşabilir. Bunun için dizilere çaprazlama (crossing over) ve değişim (mutation) operatörleri belirli yüzdelik oranlarıyla uygulanarak nesil çeşitliliği sağlanır. Böylelikle sistemin belirli noktalara gelip takılması önlenmiş olur. Aşağıda bu genetik operatörlerin çeşitleri ve uygulamaları anlatılmıştır.

3.4.6.1 Çaprazlama operatörü

İki dizinin bir araya gelerek karşılıklı gen yapılarının değişimi ile yeni dizilerin oluşumunu sağlayan operatördür. Çaprazlanarak gen (bilgi) değişiminin yapılmasından önce dizilerin çaprazlamaya tutulma olasılığı belirlenmelidir.

Literatürde bu oranın %50 - %95 oranında uygulandığı görülmüştür. Çaprazlamada bir diğer önemli unsur ise ne tür bir çaprazlamanın yapılacağıdır. Örneğin, eş kromozom seçiminde ilk olarak en yüksek uygunluk değerine sahip kromozom seçilirken, ikinci kromozom rastgele olarak seçilebilir. Bir yığına çaprazlama operatörü P_c olasılığı ile uygulanır.

Tek noktalı çaprazlama operatörü

Bu operatörde çaprazlama noktası 1 ile N arasında rastgele seçilir. Eşlenen iki dizide, bu çaprazlama noktasından sonraki bölümler yer değiştirilerek iki tane yeni birey elde edilir.

Çizelge 3.2 Tek noktalı çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A_1 :	1	0	0	1	1	1	0	0	1	1	0	1
A_2 :	1	1	0	0	0	0	1	1	1	0	0	0
Çaprazlama sonucunda oluşan yeni bireyler												
A'_1 :	1	0	0	1	1	1	0	0	1	0	0	0
A'_2 :	1	1	0	0	0	0	1	1	1	1	0	1

İki noktalı çaprazlama operatörü

Bu operatörde çaprazlama noktası 1 ile N arasında rastgele iki bölge seçilir. Eşlenen iki dizide, bu çaprazlama noktaları arasında kalan bölümler yer değiştirilerek iki tane yeni birey elde edilir.

Çizelge 3.3 İki noktalı çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A_1 :	1	0	0	1	1	1	0	0	1	1	0	1
A_2 :	1	1	0	0	0	0	1	1	1	0	0	0
Çaprazlama sonucunda oluşan yeni bireyler												
A'_1 :	1	0	0	1	0	0	1	1	1	1	0	1
A'_2 :	1	1	0	0	1	1	0	0	1	0	0	0

3.4.6.2 Mutasyon operatörü

GA'larda diğer önemli bir operatör, mutasyondur. Mutasyon doğada olduğu gibi GA

uygulamalarında da geri planda kalan bir operatördür. GA'nın çalışmasında ikinci dereceden rol oynar. GA'da mutasyon operatörü, küçük bir olasılıkla bir dizi içindeki bir veya birkaç değeri rastgele olarak değiştirerek, yığında yeni dizilerin (yani arama uzayında yeni çözüm noktalarının) elde edilmesini sağlar (Goldberg, 1989). Her yığına mutasyon operatörü, Pm olasılığı ile uygulanır. İkili düzende kodlamanın kullanıldığı bir dizide, mutasyon operatörü ile rastgele olarak seçilen eleman değeri 1 ise 0, 0 ise 1 olarak değiştirilerek yeni bir dizi elde edilir.

Özellikle GA'nın son çevrimlerinde mutasyonun etkinliği artmaktadır. Çünkü son çevrimlerde yığın iyi çözümlere yakınsadığından diziler birbirine çok benzemektedir. Bu durum, çaprazlama operatörünün farklı yeni diziler oluşturmalarını engelleyerek aramayı kısıtlar. Bu aşamada mutasyon, yığındaki değişkenliği gerçekleştirerek arama uzayında yeni çözüm noktalarının elde edilmesini sağlamaktadır (Dengiz ve Altıparmak, 1998).

Ters Çevirme

Ters çevirme mutasyonu bir alt dizi oluşturmak için bir kromozomda rastgele iki pozisyon seçer. Daha sonra bu alt dizi Şekil 3.3'de olduğu gibi iki ucu arasında ters çevrilir.

Anne-baba	7	3	2	9	10	1	6	4	5	8
-----------	---	---	---	---	----	---	---	---	---	---

Yavru	7	3	1	10	9	2	6	4	5	8
-------	---	---	---	----	---	---	---	---	---	---

Şekil 3.3 Ters çevirme mutasyonu

Ekleme

Ekleme rastgele bir parça seçer ve onu rastgele bir yere yerleştirir.

Anne-baba	7	3	2	9	10	1	6	4	5	8
-----------	---	---	---	---	----	---	---	---	---	---

Yavru	7	3	4	2	9	10	1	6	5	8
-------	---	---	---	---	---	----	---	---	---	---

Şekil 3.4 Ekleme mutasyonu

Yer deęişiklięi mutasyonu

Yer deęişiklięi mutasyonu rastgele bir alt dizi seęer ve onu Şekil 3.5'de görüldüęü gibi rastgele bir yere yerleřtirir.

Anne-baba	7	3	2	9	10	1	6	4	5	8
-----------	---	---	---	---	----	---	---	---	---	---

Yavru	7	3	6	4	5	2	9	10	1	8
-------	---	---	---	---	---	---	---	----	---	---

Şekil 3.5 Yer deęişiklięi mutasyonu

Karşılıklı deęişim mutasyonu

Karşılıklı deęişim mutasyonu, rastgele iki gen seęer ve yerlerini deęiřtirir.

Anne-baba	7	3	2	9	10	1	6	4	5	8
-----------	---	---	---	---	----	---	---	---	---	---

Yavru	7	3	5	2	9	10	1	4	2	8
-------	---	---	---	---	---	----	---	---	---	---

Şekil 3.6 Karşılıklı deęişim mutasyonu

3.5 GA'da Şema Teoremi

Genetik algoritmanın performansını anlamak için, herhangi bir arama prosedüründe mevcut ilk bilgilere bakmak gerekir ve bu da kodlamadaki benzerlikleri işleterek daha etkili bir arama yapılmasını sağlar. Bu genetik algoritma yaklaşımının kilit taşı olan, dizi oluřturma hipotezine ulaşımı sağlayan benzerlik şablonları veya şemadır (Kulluk, 2003).

Şema, belirli dizi pozisyonlarındaki benzerliklerle dizilerin alt kümelerini tanımlayan benzerlik şablonudur. Şema { 0,1} ikilik düzene "#" işaretini ekleyerek çalışır. Bu sembol, bulunduęu pozisyona 0 veya 1 deęerlerinden herhangi birisinin gelebileceęini ifade eder. Geniřletilmiş bu alfabeyi kullanarak, {0,1,#} üçlü alfabede şemalar oluřturulabilir.

Örnek olarak 5 uzunluęundaki dizileri ve şemayı ele alalım, "#0000" şeması {10000, 00000} olmak üzere iki diziyeye karşılık gelir. Aynı şekilde "#111#" dört elemanlı {01110,

01111,11110,11111} alt kümeyi ifade eder. Son bir örnek olarak, 0#l## şeması 0 ile başlayan ve üçüncü pozisyonunda 1 olan 5 uzunluğunda 8 diziyi ifade eder. Böylece aralarında benzerlik bulunan dizileri göstermenin etkili bir yolu elde edilmiş olur.

Şema, 0 ve l'in normal anlamını koruduğu ve "#"ın 0 veya 1 değerlerinden herhangi birini alabileceği {0,1,#} genişletilmiş alfabeden oluşan bir dizidir (Goldberg, 1989). Bu notasyon yöntemi genetik algoritma yönteminin analizini basitleştirir. Çünkü dizilerden oluşan bir popülasyondaki olası bütün benzerlikleri açıkça görür. Mutasyonun şema oluşturmada çok az bir etkisi vardır ve sadece önemli potansiyel genetik bilgilerin düzeltilemez kayıplarını engellemek için çalışırlar.

Stephens vd. (1995), şema oluşturmada büyüklüğün önemi konusunda çalışma yapmışlardır (Kulluk, 2003).

3.6 Kontrol Parametreleri

Genetik algoritmaların performansını etkileyen 6 farklı kontrol parametresi vardır. Bunlar :

- Popülasyon genişliği,
- Çaprazlama oranı,
- Mutasyon oranı,
- Nesil aralığı (Generation gap),
- Seçim stratejisi,
- Ölçeklendirme fonksiyonu.

Popülasyon genişliği (N_p): Popülasyon genişliği algoritmanın yakınsamasını etkileyen bir faktördür. Popülasyon genişliğinin küçük olması, küçük popülasyonların arama uzayını örneklemede yetersiz kalmasına ve zamansız yakınsamaya neden olmasından dolayı Genetik algoritmanın performansını azaltır. Popülasyon genişliğinin büyük olması ise, çözüm uzayının çok iyi örneklenmesini sağladığı için aramanın etkinliğini artırır ve zamansız yakınsamayı önler. Ancak popülasyon genişliği büyük olduğunda, her iterasyonda dizilerin değerlendirilmesi çalışma zamanını artırır ve yakınsama çok yavaş gerçekleşir.

Çaprazlama oranı (P_c): Çaprazlama operatörünün kullanım sıklığını kontrol eder. Her yeni yığında $P_c \times N$ kadar diziyi çaprazlama uygulanır. Düşük çaprazlama oranı, aramanın çok

yavaş gerçekleşmesine neden olurken; yüksek çaprazlama oranı ise yığın değişkenliğini hızlı bir şekilde gerçekleştirir.

Mutasyon oranı (P_m): Seçim işlemi ile elde edilen yeni yığındaki her dizinin her elemanı. P_m mutasyon oranına eşit bir olasılıkla rastgele değişime uğrar. Sonuçta $P_m \times N \times L$ adet mutasyon gerçekleşir. Yüksek mutasyon oranı genetik algoritmadaki aramanın bir rastgele aramaya eşdeğer olmasına neden olduğundan, seçilen mutasyon oranının düşük olması tercih edilir.

Seçim stratejileri: Genetik algoritmalarda kullanılan seçim stratejileri daha öncede anlatıldığı gibi, orantılı yeniden seçim mekanizması, sıralı seçim mekanizması, turnuva seçim mekanizması ve denge durumu seçim mekanizması olarak dörde ayrılmaktadır. Seçim stratejileri de genetik algoritmaların performansını etkileyen bir faktördür.

Ölçeklendirme faktörü: Genetik algoritmaların çalışmaları süresince yığındaki değişkenliğin korunması önemlidir. Yığındaki uyum değerlerinin kullanılacak bir metot yardımı ile ölçeklendirilmesi gerekir.

Bitirme koşulu: GA 'da üretim- değerlendirme- seçim çevrimi, önceden belirlenen çevrim sayısına ulaşıncaya ya da yığının ortalama uygunluk değeri yığındaki en iyi dizinin uygunluk değerine önceden belirlenen oranda yaklaşıncaya kadar devam eder (Goldberg, 1989).

3.7 Genetik Algoritmaların Performansını Etkileyen Etmenler

Genetik algoritmaların performansını etkileyen etmenler kısaca şöyle özetlenebilir (Kulluk, 2003):

Kromozom sayısı: Kromozom sayısını artırmak çalışma zamanını arttırırken, azaltmak da kromozom çeşitliliğini yok eder.

Mutasyon oranı: Kromozomlar birbirlerine benzemeye başladığında, çözüm hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi GA'nın sıkıştığı yerden kurtulması için tek yoldur. Ancak yüksek bir değer vermek GA'yı kararlı bir noktaya ulaşmaktan alıkoyacaktır.

Kaç noktalı çaprazlama yapılacağı: Normal olarak çaprazlama tek noktada gerçekleştirilmekle beraber yapılan çalışmalar bazı problemlerde çok noktalı çaprazlamanın çok yararlı olduğunu göstermiştir. Çaprazlama sonucu elde edilen bireylerin nasıl değerlendirileceği, elde edilen iki bireyin birden kullanılıp kullanılamayacağı bazen önemli

olmaktadır.

Nesillerin birbirinden ayrık olup olmadığı: Normal olarak her nesil tümüyle bir önceki nesle bağlı olarak yaratılır. Bazı durumlarda yeni nesli, eski nesille birlikte; yeni neslin o ana kadar elde edilen bireyleri ile yaratmak yararlı olabilir.

Parametre kodlamasının nasıl yapıldığı: Kodlamanın nasıl yapıldığı en önemli noktalardan biridir. Örnek vermek gerekirse kimi zaman bir parametrenin doğrusal yada logaritmik kodlanması GA'nın performansında önemli bir farka yol açabilir.

Kodlama gösteriminin nasıl yapıldığı: Bu da nasıl olduğu yeterince açık olmamakla beraber genetik algoritmaların performansını etkileyen bir noktadır. İkili düzen, kayan nokta aritmetiği ya da Gray kodu ile gösterim en yaygın yöntemlerdir.

Başarı değerlendirmesinin nasıl yapıldığı: Akıllıca yazılmamış bir değerlendirme işlevi çalışma zamanını uzatabileceği gibi çözüme hiçbir zaman ulaşılmamasına da neden olabilir.

Shaffer (1998), Genetik algoritmanın performansını artırmak için şunları önermiştir (Kulluk, 2003):

- Farklı mutasyon oranları, farklı popülasyon büyüklükleri vs. kullanmak,
- Optimizasyonu daha stokastik hale getirmek,
- Genetik gösterimi değiştirmek (Gray kodlama kullanmak vb.).

3.8 Genetik Algoritma Uygulamaları

Genetik algoritmaların karmaşık uzaylarda, sağlam araştırmalara imkan tanıdığı teorik olarak ispatlanmıştır. Problemlerin ihtiyaç duyduğu verimli ve etkili araştırmalarda geçerli bir yaklaşım olarak yerleşen genetik algoritmalar; mühendislik, iş/ticaret ve bilimsel çalışmalarda daha geniş uygulama imkanları bulmaktadır. Genetik algoritmaların uygulama alanlarının genişlemesinin sebebi, hesaplamalardaki kolaylığı ve arama uzayı ile ilgili varsayımlarda önemli sınırlayıcılarla kısıtlandırılmamasıdır.

GA, sezgisel bir metot olduğundan dolayı verilen bir problem için optimum sonucu bulamayabilir, ancak; geleneksel metotlarla çözülemeyen veya çözüm zamanı problemin büyüklüğü ile üstel olarak artan problemlerde optimele çok yakın sonuçlar vermektedir. Başlangıçta doğrusal olmayan (non-linear) optimizasyon problemlerine uygulanan GA, sonraları gezgin satıcı, kareli atama, yerleşim, atölye çizelgeleme, ders/sınav programı

hazırlanması gibi kombinatoriyal optimizasyon problemlerine de başarıyla uygulanmıştır (Liepins vd., 1990). Genetik algoritmaların kombinatoriyal problemlerin çözümünde oldukça başarılı olduğu ispatlanmıştır (Drezner, 2003).

GA, çeşitli alanlarda kullanılmıştır (Balakrishnan ve Jacob, 1996): Goldberg (1989) genetik algoritmaları boru hattı sistemlerinin optimizasyonunda; Cleveland ve Smith (1989) seri üretim çizelgelemesinde; Liepins ve Potter (1991) çoklu hata tanımında; Holsapple ve arkadaşları esnek imalat sistemlerinin çizelgelenmesinde ve uyumlu karar destek sistemlerinde kullanmışlardır.

GA'lar problem çözmede alternatif metotlar sağlamakla kalmaz, aynı zamanda problemlerin çoğunda diğer geleneksel yöntemleri de tutarlıca gerçekleştirir. Örneğin, optimal parametreler bulmada yer alan gerçek dünya problemleri geleneksel metotlar için zor olabilir; fakat GA'lar için idealdir. GA'lar, optimizasyondaki etkili performanslarından dolayı yanlış bir kanıyla işlem en iyileyici olarak görülmektedirler. Aslında genetik algoritmalara çok fazla bakış açısı vardır. Birçok kullanıcı, GA'ya problem çözücü olarak bakmaktadır. Fakat bu kısıtlayıcı bir bakış açısıdır. GA'lar,

- Araştırma alanı geniş, kompleks ve anlaşılması zayıfsa,
- Konudaki bilgi azsa ya da eldeki uzman bilgi, araştırma alanını daraltmada zorlanıyorsa,
- Matematiksel analiz elde edilemiyorsa,
- Geleneksel araştırma metotları başarısız olmuşsa faydalı ve etkilidirler (Çetin, 2002).

GA yaklaşımının avantajı, zorluklarla ve hedeflerle başa çıkmadaki rahatlığıdır.

Genetik algoritmaların genel uygulama alanları aşağıdaki gibi verilebilir (Emel ve Taşkın, 2002):

- Genel uygulama alanları
 - Optimizasyon
 - Otomatik programlama ve bilgi sistemleri
 - Mekanik öğrenme
 - Ekonomik ve sosyal sistem modelleri
- İşletmelerdeki uygulama alanları

- Finans
- Pazarlama
- Üretim/İşlemler
- Montaj hattı dengeleme problemi
- Çizelgeleme problemi
- Tesis yerleşim problemi
- Atama problemi
- Hücresel üretim problemi
- Sistem güvenilirliği problemi
- Taşıma problemi
- Gezgin satıcı problemi
- Araç rotalama problemi
- Minimum yayılan ağaç problemi

4. TESİS YERLEŞİMİ PROBLEMLERİNDE GENETİK ALGORİTMALAR

4.1 Problemin Tanımlanması ve Tesis Yerleşiminin Gösterimi

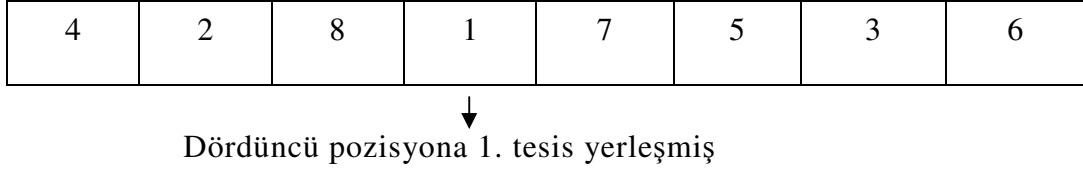
Tesis yerleşimi problemlerini genetik algoritmalar yardımıyla çözmekte ilk adım amaç fonksiyonunun belirlenmesidir. Amaç fonksiyonu genetik algoritmalarda, daha önceki bölümlerde de belirtildiği gibi uyum fonksiyonu olarak kullanılır. Dolayısıyla bireylerin uyum değerlerinin doğru hesaplanmasında büyük öneme sahiptir. Amaç fonksiyonu yapısı, problem özelliklerine ve iyileştirilmek istenen kriterlere göre farklılık gösterir.

Tam (1992), tesis yerleşimi problemlerinde amacın, tesislerin çeşitli alan ve geometrik kısıtlarını sağlamak ve aynı zamanda tesisler arasındaki ağırlıklı iş akışını minimize etmek olduğunu belirtmiştir. Genetik algoritmaların kullanımı çözüm uzayının farklı bölgelerinin eş zamanlı olarak incelenmesine izin vermesi yönüyle etkin bir araştırmaya imkan sağlar (Kulluk, 2003).

Tesis yerleşimi problemlerinde ilk kısıt olarak, tesisler üst üste binmemelidir. Buna ek olarak, özel bölümler ve toplam zemin alanı ilk belirlenen limitlerin içinde olmalıdır. Sonuçta oluşan çözüm, hem bölümlerin hem de bütün yerleşimin uygun şekillerini vermelidir.

Tesis yerleşimi için geliştirilen genetik algoritmalarda genellikle tesis alanlarının eşit olduğu varsayımı yapılır (Tavakkoli-Moghaddain ve Shayan, 1998). Bu durumda çözüm 15-20 tesisli küçük problemler için uygun zamanda elde edilebilir. Ancak gerçek hayattaki uygulamalarda bütün tesisler eşit alana sahip değildir. Bu durumda farklı alanlara sahip tesisler, eşit birim karelerle (bloklarla) temsil edilir.

Tesis yerleşimi probleminin yapısı, genetik algoritmaların uygulanması için çok elverişlidir. Genetik algoritmaların kullanımı için gerekli dizilerin (bireylerin) uzunluğu verilen bir problemdeki uygun yerlere atanacak tesis sayılarına bağlıdır. Her birey için bir uyum değeri çözümün kalitesini göstermek üzere hesaplanır. Çözüm dizileri, genetik operatörlerin doğrudan uygulanabileceği özelliktedir. Yani tesislerin pozisyonlarını göstermek üzere oluşturulan çözüm dizileri hiçbir dönüşüme tabi tutulmadan genetik algoritmalara birey olarak tanıtılabilirler. Bu bakımdan tesis yerleşimi problemlerinde, genetik algoritmaların çözüm dizilerinin oluşturulmasında ikili düzende kodlamanın kullanılmasına gerek yoktur. Tesis yerleşiminde çözümler, düzgün permütasyonel bir kodlamayla gösterilir ve çözüm dizilerinde sayılar tesisleri, sayıların dizide bulundukları yerler ise tesislerin yerleşimdeki pozisyonlarını göstermektedir (Şekil 4.1).



Şekil 4.1 Örnek genetik kodlama gösterimi

Suresh (1995), arkadaşlarıyla yaptığı çalışmada, bu gösterim şeklini kullanmıştır. Geliştirdikleri genetik algorithmada, satır ve sütun değerleri önceden belirlenmiştir ve sıfırdan başlayarak ileriye doğru değerler almaktadırlar. Kullanılmayacak pozisyonlara (eğer varsa), kukla değerler atanır. Çözüm tamsayılardan oluşan bir dizi ile gösterilir, ikili düzende bir gösterim kullanılmaz. Tamsayılar tesisleri ve tesislerin dizideki pozisyonları da tesislerin yerleşimdeki pozisyonlarını gösterir (Şekil 4.1) (Kulluk, 2003).

GA'da bir bireyin uyumu ise, $(\mu: S \rightarrow R^+)$ olarak μ fonksiyonuyla ölçülmektedir. Burada S bütün bireylerin kümesi (V^k) ve R^+ ise negatif olmayan gerçek sayıların kümesini ifade etmektedir. Eğer amaç fonksiyonunun (f) bütün sonuçlarının pozitif olduğu biliniyorsa f fonksiyonu, μ uyum fonksiyonu olarak aynen alınabilir. Aksi halde $\mu; f$ fonksiyonunun bir dönüşümüdür. Tam (1992)'ye göre bu dönüşümün şekli bazı faktörlere bağlıdır. Bunlar (Kulluk, 2003):

- $\max f$ ya da $\min f$ olarak kullanılması,
- Belirlenen seçim mekanizması,
- Kullanılan ölçekleme fonksiyonudur.

Kochhar vd. (1998), farklı alanlı tesis yerleşimi problemlerine HOPE (heuristically operated placement evolution) adını verdikleri bir algoritma geliştirmişlerdir (Kulluk, 2003). Yaptıkları çalışmada, tesis yerleşimini göstermek için tesis zemininin birim kare alanlardan meydana geldiğini düşünmüşlerdir. Her bölüm, kendi alanına eşit büyüklükteki kareler setine atanır. Her dizi, N kare sayısını göstermek üzere l 'den N 'e kadar tamsayılar kümesinden meydana gelir. Tamsayılar, ilk bölümden başlamak üzere artan şekilde her bölümün karelerine atanır. Boş kerelerin olması durumunda buralara kukla bölümler atanır. Eşit alanlı yerleşim problemlerinde, her bölüm atanabileceği tek bir kareye sahiptir. Şekil 4.2'de sol taraf 12 eşit alanlı bölümden oluşan tesisi 3×4 olarak göstermektedir. Bu şeklin ilgili dizi gösterimi aşağıdaki gibidir;

03 02 10 01 11 12 09 04 05 06 08 07

3	2	10	3(1)	2(1)	10(6)
1	11	12	1(1)	11(6)	12(6)
9	4	5	9(5)	4(4)	5(2)
6	8	7	6(3)	8(4)	7(4)

Eşit alanlı yerleşim

Farklı alanlı yerleşim

Şekil 4.2 Eşit alanlı ve farklı alanlı bölümlere sahip yerleşim

Şekil 4.2’de sağ kısım ise farklı alanlı bir tesis yerleşimini göstermektedir. Her bölüm alan ihtiyacına bağlı olarak bir veya daha çok kareye atanır. Birinci bölüm 1,2 ve 3 kareleriyle; ikinci bölüm 4 ve 5 kareleriyle gibi gösterim şekli uygulanmaktadır. Şekilde parantez içindeki sayılar o karenin hangi bölüme ait olduğunu göstermektedir. İlgili dizi gösterimi eşit alanlı gösterimle aynı şekilde olacaktır,

03 02 10 01 11 12 09 04 05 06 08 07

Ancak bu dizideki tamsayılar bölümleri değil, doğrudan karelerin sırasını göstermektedir.

Tam (1992), geliştirdiği Genetik algoritmada, tesislerin alan ve şekil kısıtlarını sağlarken iş akışı yoğun olan tesislerin birbirine yakın yerleştirildiği alan yerleşimini amaç olarak belirlemiştir (Kulluk, 2003). “S, s” çözüm kümesinde bir çözümü göstermek üzere amaç fonksiyonunu şu şekilde belirlemiştir:

$$\min_{s \in S} F = \sum_{i=1}^n \sum_{j=1}^n v_{ij} d_{ij} \quad (4.1)$$

$$\text{kısıtlar} \quad a_{\min}^i \leq a_i \leq a_{\max}^i, \quad i, j = 1, 2, \dots, n$$

$$0 \leq o_i \leq o_{\max}^i, \quad ,$$

Burada (4.1);

v_{ij} : i ve j tesisleri arasındaki iş akışını,

d_{ij} : i ve j tesislerine atanmış alanların merkezleri arasındaki dikdörtgensel uzaklığı,

a_i : i tesisinin görünüş (aspect) oranını,

o_i : i tesisine atanmış alanın ölü alan oranını,

$a_{\min}^i$: a_i için alt sınır değerini,

$a_{\max}^i$: a_i için üst sınır değerini,

$o_{\max}^i$: o_i için üst sınır değerini ifade etmektedir.

Genetik algoritmalar, kısıtlan olan optimizasyon problemlerim çözmekte sorunlarla karşılaşmaktadır. Bu nedenle Tam (1992), mevcut kısıtlan amaç fonksiyonuna ceza değeri şeklinde sokarak, problemi kısıtsız bir optimizasyon problemi haline dönüştürmüştür (Kulluk, 2003). Şöyle ki;

$$\min F = \sum_{i=1}^n \sum_{j=1}^n v_{ij} d_{ij} + \sum_{k=1}^n (w_k^\alpha \alpha_k + w_k^\beta \beta_k) \quad (4.2)$$

Burada (4.2);

$$\alpha_k = \max\{0, \max\{(a_k - \max\{a_{\max}^k, 1/a_{\min}^k\}), (\min\{a_{\min}^k, 1/a_{\max}^k\} - a_k)\}\}$$

$$\beta_k = \max\{0, o_k - o_k^{\max}\}$$

$$w_k^\alpha, w_k^\beta \geq 0, \quad i, j, k = 1, 2, \dots, n$$

F fonksiyonundaki ilk ifade, tesisler arası ağırlıklı yük akışını ölçmektedir ve tesisler arası uzaklık (d_{ij}) ile tesisler arası yük akış hacminin (v_{ij}) çarpılmasıyla belirlenmektedir. İkinci ifade ise, geometrik kısıtların sağlanması ile ilgili ceza fonksiyonudur. Ceza fonksiyonunun ilk ifadesi, görünüm (aspect) oranının ne kadar ihlal edildiğini belirlemektedir. Bu oran, tesisin kendine ait faaliyetleri gerçekleştirirken atandığı alanı ne kadar geçebileceğini (palet

kullanımı, güvenlik faktörü vb.) göstermektedir. Her tesisin belli seviyede bu kısıtı ihlal etme özelliği olduğundan, α_k değeri bir ağırlık faktörü olan $w_k^\alpha \geq 0$ ile çarpılmaktadır. Benzer şekilde ikinci ifade, β_k tesisin atandığı pozisyonda daha önceden başka bir tesis olmasını değerlendirmektedir. Burada her tesise düzgün olmayan alanların atanması durumunda, sahip olduğu tolerans seviyesini gösteren bağımsız $w_k^\beta \geq 0$ değerleri atanmaktadır.

Genetik algoritmaların uygulanmasında karşılaşılan diğer bir sorun, çözümleri oluşturan bireylerin sınırlı uzunlukta diziler olarak tanımlanmasıdır. Bu kodlama yapılırken orijinal problemin ifade edilmesine ve özelliklerinin korunmasına dikkat edilmelidir. Bu aşamada Goldberg'in (1989) belirlediği ilkeler temel alınarak yapılan bir kodlama, tesis yerleşimine etkin bir çözüm sağlamak için yeterli olmaktadır.

4.2 Yerleşim Planının Kodlanması

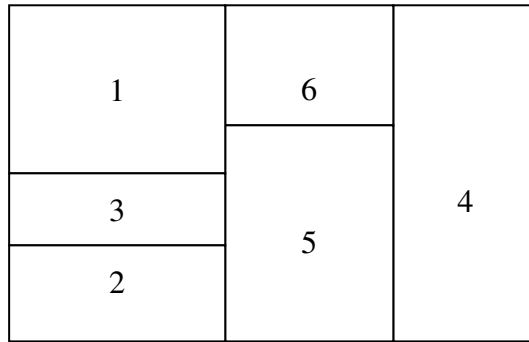
Genetik algoritmaların uygulanmasında çözümlerin sınırlı uzunlukta dizilerle kodlanması önemli bir problemdir (Goldberg, 1989). Genetik algoritmaların kodlanmasında iki prensip belirlemiştir. Bunlar:

- **Anlamlı bloklar ilkesi:** Kullanıcılar, problemi ifade etmek için problemle ilgili kısa, düşük seviyeli şemalar (diziler) kullanmalı ve problemle ilgisiz sabit pozisyonların ifadesinde ise, ifadesiz semboller kullanmalıdır.
- **En az harf ilkesi:** Kullanıcılar, problemi doğal şekilde ifade edebilecek en az harf kümesini seçmelidirler.

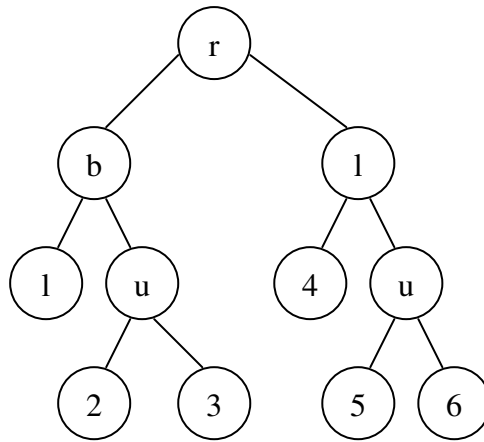
Bir tesis yerleşim planı, bir dörtgen bloğun dilimlenmiş yapısı olarak ifade edilir. Böylece dilimlenmiş yapıdaki her dörtgen parça, bir tesise ayrılan alanı gösterir. Çizelge 4.1, 6 tesisin geometri kısıtlarını göstermektedir. Çizelgedeki 2. sütun her tesisin alan ihtiyacını ifade etmektedir. Bu 6 tesisin yerleşimi, Şekil 4.3.a'daki gibi dilimlenmiş yapı ile gösterilir. Alan ihtiyacı, eğer her tesise ayrılan alan en az istenilen alan kadar büyükse karşılanır. Şekil 4.3.a'da bütün alan ihtiyaçlarının karşılandığı bir dilimlenmiş yapı gösterilmektedir.

Çizelge 4.1 Altı tesisin geometrik kısıtları (Kulluk, 2003)

Tesis Numarası	Alan	Görünüş Oranı		Ölü Alan Oranı
		Alt Sınır	Üst Sınır	Üst Sınır
1	100	0,5	1,2	0
2	80	0,4	1,1	0
3	50	0,3	3,0	0
4	60	0,1	5,5	0
5	120	0,9	1,9	0
6	40	0,4	1,2	0



Şekil 4.3.a Dilimlenmiş yapı (Kulluk, 2003)



Şekil 4.3.b Dilimlenmiş ağaç (Kulluk, 2003)

Bu dilimlenmiş yapıya alternatif olarak dilimlenmiş ağaç da kullanılabilir. Dilimlenmiş ağaç, dilimlenmiş yapının oluşturulmasını sağlayan süreci gösteren ikili düzende bir ağaçtır. Şekil 4.3.b'de örneğe ilişkin dilimlenmiş ağaç gösterilmektedir. Bu ağaç yapısında her iç düğüm dörtgen yapının nasıl kesildiğini göstermektedir. Kesme tipi, her iç düğüme atanan bir harf ile gösterilmektedir ('u', 'b' gibi). Düğümleri oluşturan yapraklara ise sadece atandığı tesisi belirten belirli bir sayı verilmektedir. Tam (1992)'a göre burada eğer ağacın yapısı sabit tutulup sadece bir iç düğümün kesimi değiştirilirse, farklı bir yerleşim elde edilir (Kulluk, 2003).

Bütün yerleşimleri içeren S çözüm kümesi, ağaçta bulunan kesimler tekrar düzenlenerek elde edilebilir. Verilen bir ağaç yapısında, dilimlenmiş ağacın daha kullanışlı bir gösterim şekli düğümlerin sıralanmasıdır. Örneğin, Şekil 4.3.b'deki dilimlenmiş ağaç: 65u4132ulbr dizisi ile gösterilebilir. Şekilde kullanılan l soldan, r sağdan, u yukarıdan ve b aşağıdan kesme işlemlerini göstermektedir. Bir ağaç yapısı verildiğinde, aşağıdaki eşit gösterimler gözlenebilir;

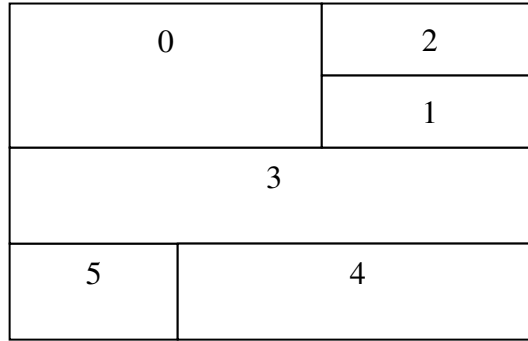
Dilimlenmiş ağaç $\leftarrow\rightarrow$ İşlem Sırası $\leftarrow\rightarrow$ Genetik aloritmada Yapı ve Yerleşim Planı $\leftarrow\rightarrow$
 $\rightarrow$ Dilimlenmiş Yapı

Genetik algoritmaların terminolojisi kullanılarak, bir yapı bir işlem sırası ile aynı olarak ifade edilebilir. Bütün yapılar, n tesis sayışım göstermek üzere $\{b, u, r, /\}^{n-1}$ harf kümesinden meydana getirilmektedir. Bu nedenle çözüm uzayının büyüklüğü 4^{n-1} 'dir. Kullanılan her harf, bir bölüm yaratmak için kullanılmaktadır ve diğer pozisyonlardaki harflerden bağımsızdır. Ayrıca kullanılan bu küme, problemi tanımlayan en küçük yapıdır. Böylece, Goldberg'in iki ilkesi sağlanmaktadır.

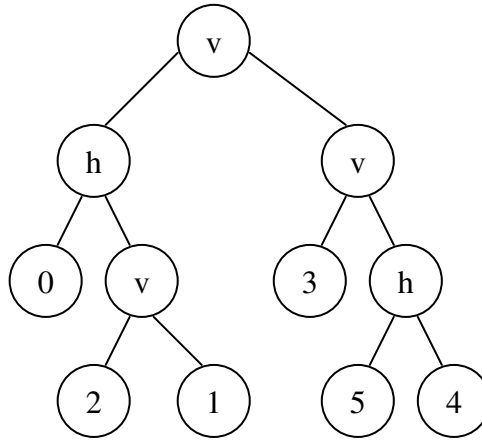
Tam ve Chan (1998), Tam'in 1992'de yaptığı çalışmadan yola çıkarak bir paralel genetik algoritma geliştirmişlerdir. Tam'in kullandığı dilimlenmiş ağaç gösterimini kullanmışlardır ancak sağdan, soldan, yukarıdan ve aşağıdan kesme yerine dikey ve yatay kesme kullanmışlar ve bunları da v: dikey, h: yatay olarak göstermişlerdir. Çalışmalarındaki dilimlenmiş yapı ve dilimlenmiş ağaç Şekil 4.4.a ve 4.4.b'de görülebilir.

Tam'in 1992'de yaptığı çalışmasında, kodlama şeklinin basitleştirilmesiyle sadece bütün çözüm uzayının bir alt kümesi elde edilir. Eğer çözüm kalitesi, dilimlenmiş ağaç yapısına bağlı ise, bazı iyi çözüm adaylarına dikkat edilmeyebilir. Tam ve Chan (1998), bütün çözüm kümesini içine almak için arama uzayını genişleten kodlama sistemi geliştirmişlerdir.

Al-Hakim (2000), Tam ve Chan'in yaklaşımlarında bir çok zorluk olduğunu öne sürmüş ve bu konuda yeni bir yaklaşım önermiştir (Kulluk, 2003).



Şekil 4.4.a Altı tesisli bir dilimlenmiş yapı (Kulluk, 2003)



Şekil 4.4.b Dilimlenmiş ağaç (Kulluk, 2003)

4.3 Tesislerin Geometrik Kısıtları

Önceden de belirtildiği gibi tesis yerleşimi problemlerinde ilk kısıt, tesislerin üst üste binmemesi yani her tesisin sadece bir pozisyona atanması ve bir pozisyona birden çok tesisin atanamamasıdır. Buna ek olarak tesislerin, geometrik şekillerinin karşılanacağı alanlara atanması gerekir. Gerçek hayatta tesisler genellikle farklı alanlara ve düzgün olmayan şekillere sahiptirler. Bu bakımdan tesislerin geometrik kısıtlarının sağlanması büyük bir problem oluşturmaktadır.

Tesislerin farklı alanlara sahip oldukları durumlarda genetik algoritmaların kullanılmasında Tam (1992), Suresh (1995), Kochhar (1998) ve Li (2000) değişik yaklaşımlar geliştirmişlerdir. Suresh (1995) ve Kochhar (1997) farklı alanlara sahip tesisler için, yerleşim alanının belli birim karelere bölünmesini ve tesislere yeterli birim alanın atanmasını önermektedir (Kulluk, 2003). Bu durumda tesis ihtiyaçlarını karşılayacak şekilde belli sayıda

birim karelerden oluşan grupların bir arada tutulması ile ilgili bir kısıt gerekmektedir.

Tam (1992), Tam ve Chan (1998) ise yaklaşımlarında iki geometrik kısıt üzerinde durmuşlardır (Kulluk, 2003). Bunlardan ilki alan ihtiyacıdır. Bu ihtiyaç toplam yerleşim alanının, bir araya getirilen tesislerin alanları toplamına eşit veya büyük olmasını şart koşar. Ayrıca, zemin planında her tesise ayrılan alan, tesislerin alan ihtiyaçlarını karşılamalıdır. İkinci geometrik kısıt ise, tesislerin şeklidir.

Genellikle, tesislerin yerleştirilmesi için bütün alan kullanılabilir. Her zaman yerleşim planında; asansör boşlukları, sütunlar gibi tesis olarak nitelendirilen kukla alanlar (dummy areas) vardır. Bu tesislerin kapladığı kukla alanlar, problemde söz konusu olan tesislerin atanacağı alandan çıkarılmalıdır. Bu şekilde zorunlu olarak kaplanan alanlar, bütün yerleşim planının şekillendirilmesinde kullanılabilir. Dörtgensel bir yüzeyin köşelerinde, bu şekilde kullanılamayan alanların tanımlanması, yerleşimin yapılacağı kullanılabilir alanın belirlenmesini sağlayacaktır.

4.4 Genetik Parametre Değerlerinin Hesaplanması

4.4.1 Başlangıç popülasyonu ve popülasyon büyüklüğü

Genetik algoritmaların kullanımı, genetik operatörlerde temel olarak kullanılacak başlangıç popülasyonunun oluşturulmasını gerektirir. Başlangıç popülasyonunun oluşturulması, tesis yerleşimi problemlerine genetik algoritmaların uygulanmasında önemli bir faktördür. Yapılan bir çok çalışmada araştırmacılar, başlangıç popülasyonunun oluşturulmasında rastgele yöntemleri kullanmışlardır. Suresh (1995) arkadaşlarıyla yaptığı çalışmada başlangıç popülasyonunu oluşturmak için şöyle bir yaklaşım önermektedir (Kulluk, 2003); n/m yapısındaki bir problem için $0 \leq m \leq 2m$, bir dizide son sıradaki tesisin başa getirilmesi şeklinde bir işlem uygulanmaktadır. Bu şekilde elde edilen çözüme ek olarak kalan $(Np-m)$ çözüm rastgele olarak üretilmektedir. İlk üretilen m çözüm en az bir tesisin istenen belli bir pozisyonda olmasını garanti etmektedir. Li ve Love (2000), başlangıç popülasyonunun oluşturulmasına ilişkin, bazı kısıtlarla rastgele sayı üretimini çalışmalarında kullanmışlardır. Kochhar vd. (1997), geliştirdikleri HOPE algoritması için, başlangıç popülasyonunun oluşturulmasında iki yaklaşımı test etmişlerdir (Kulluk, 2003). Bu yaklaşımlardan birincisinde (INIT_RANDOM), kareler rastgele üretilip sıralanır, ikincisinde (INIT_SFC) ise bölümler üretilip sıralanır. Birinci yaklaşım eşit alanlı tesisler için uygundur ama farklı alanlı tesislerde uygun olmayan sonuçlar vermektedir; ikinci yaklaşım ise farklı alanlı tesisler için uygun

sonuçlar vermektedir. Bu sebepten dolayı Kochhar ve arkadaşları geliştirdikleri algorithmada başlangıç popülasyonunu oluştururken bu ikinci yaklaşımı kullanmışlardır.

Popülasyon büyüklüğü en az popülasyonun oluşturulma yöntemi kadar genetik algorithmaların performansını etkileyen bir faktördür. Popülasyon büyüklüğü popülasyondaki kromozom (birey) sayısını belirleyen bir değerdir. Suresh vd. (1995), yaptıkları çalışmada en uygun popülasyon büyüklüğünün belirlenmesinde 20 aralıkla 40 ile 160 arasında popülasyon büyüklükleri üzerinde çalışmışlardır. Artan popülasyon büyüklüğünün 15 tesise kadar olan problemlerde, çözüm kalitesi üzerinde dikkate değer bir gelişmeye sebep olmadığını belirtmişlerdir. Ancak, problemin boyutunun 15 tesisten fazla olması halinde çözüm kalitesinin artan popülasyon büyüklüğüne paralel olarak iyileştiğini gözlemlemişlerdir. Buna ek olarak 100 bireyden fazla popülasyon büyüklükleri de çözüm kalitesinde belirgin farklar yaratmamıştır. Bu sebepten dolayı Suresh ve arkadaşları çalışmalarında popülasyon büyüklüğü olarak 100 bireyden oluşan bir popülasyon ($N_p = 100$) kullanmaktadırlar. Shaffer (1998), yaptığı çalışmada daha önce yapılan çalışmaları da göz önünde bulundurarak, genetik algorithmaların uygulanmasında, popülasyon büyüklüğünün 25 ile 100 birey arasında olmasının en uygun sonuçları verdiğini belirtmiştir. Tam (1992), Kochher (1997), İşlier (1998), ve Li (2000) çalışmalarında bu aralıktaki popülasyon büyüklüklerini kullanmışlardır (Kulluk, 2003).

4.4.2 Çaprazlama ve mutasyon

Daha etkili bir çaprazlama ve mutasyon işleminin gerekli rastgeleliği sağlayacak şekilde mümkün oldukça düşük seviyede tutulması, daha verimli genetik algorithmaların oluşturulmasını sağlamaktadır.

Son birkaç yıldır permütasyon gösterim için kısmi eşleşmeli çaprazlama (PMX), sıralı çaprazlama (OX) ve dizi çaprazlama (CX) gibi çeşitli çaprazlama operatörleri geliştirilmiştir. Cheng ve Tasawa (1996), çalışmalarında aşağıdaki üç adımdan meydana gelen kısmi eşleşmeli çaprazlamayı kullanmışlardır (Kulluk, 2003).

- Rastgele iki kesme noktası seç ve ebeveynlerin iki kesme noktaları arasındaki genlerini değiştir.
- Değişen genlerden planlama ilişkisini belirle.
- Uyuşmazlığı plan ilişkisiyle yeniden çöz.

Bunu bir örnekle açıklayacak olursak: P_1 ve P_2 ebeveynleri gösterebilirsin. İki kesme noktası 4 ve 6 olsun. Buradan altgen 1 [4,5,6] ve altgen 2 [5,1,9] olarak oluşturulur. Bir çocuk oluşturmak için altgen 1 ve altgen 2 yer değiştirilir (Şekil: 4.5.a). Bu işlem sonucu oluşan çocuk uygun değildir. Bu durumda planlama ilişkisini belirlemek için altgen 1 ve altgen 2 karşılaştırılır (Şekil 4.5.b). Son olarak uyumsuzluk plan ilişkisiyle yeniden çözülerek uygun bir çocuk elde edilir (Şekil 4.5.c).

Değişim için seçilen genler	Planlama ilişkileri	Uyumsuzluk yeniden çözümü
$P_1 : 1\ 2\ 3\ \underline{4\ 5\ 6}\ 7\ 8\ 9$	$1 \leftrightarrow 5 \leftrightarrow 4$	$o : \underline{1\ 2\ 3\ 5\ 1\ 9\ 7\ 8\ 9}$
↓ ↓		↓ ↓
$O : 1\ 2\ 3\ 5\ 1\ 9\ 7\ 8\ 9$	$6 \leftrightarrow 9$	$o : \underline{4\ 2\ 3\ 5\ 1\ 9\ 7\ 8\ 6}$
$P_2 : 2\ 7\ 8\ \underline{5\ 1\ 9}\ 6\ 3\ 4$		
(a)	(b)	(c)

Şekil 4.5 PMX (a) değişim (b) planlama ilişkisi (c) uyumlu hale getirme

Suresh vd. (1995) çalışmalarında; KAP için geliştirilen bir GA'nın, popülasyonda bulunan bireylerin uygulanabilir oldukları garanti altına alınmadıkça, yeterli etkinlikte sonuçlar vermeyeceğini belirtmektedirler. Standart çaprazlama işlemlerini kullanarak her ne kadar iyi sonuçlar elde etmiş olsalar da, özel geliştirilen çaprazlama işlemleri daha kaliteli sonuçlar vermektedir. Bunun sebebi küçük boyutlu problemler için PMX, CX ve OX çaprazlamalarının iyi sonuçlar vermesine rağmen, büyük ölçekli problemlerde farklı sonuçlarla karşılaşıldığından, hangi metodun kullanılacağı belirlenememesidir. Bu sebeple çalışmalarında uygun ana-baba genlerinden, uygun çocuklar üretilmesini sağlayacak yeni bir çaprazlama işlemini geliştirmişlerdir. Geliştirdikleri çaprazlama işlemi ile; çocuklarla aile arasındaki yapısal benzerliği korurken, aynı zamanda çeşitliliğin sağlanmasını garanti altına almışlardır. Aşağıda Suresh vd.nin geliştirdiği çaprazlama için bir örnek verilmektedir. Bu çaprazlama işleminde önce rastgele olarak bir kesme noktası belirlenmektedir (kesme noktası 3. pozisyonadadır).

$$P1 = 1\ 2\ 3\ |4\ 5\ 6\ 7\ 8$$

$$P2 = 471152836$$

Çaprazlama sonrasında şu diziler elde edilmektedir:

$$A1 = 123163825$$

$$A2 = 876541471$$

Görüldüğü gibi, ilk kaynak bireyin kuyruk kısmı, ters sırada ikinci çocuğun baş kısmına ve ikinci kaynak bireyin baş kısmı ikinci çocuğun kuyruk kısmına getirilmektedir. Alt dizilerin sırasının değiştirilmesi, aile bireylerinden çok farklı çocuklar üretilmesini sağlamaktadır. Uygulanabilirliğin sağlanması için, aynı dizide tekrar eden tesisler eksik tesislerle değiştirilmelidir. Bunun gerçekleştirilmesi için ilk çocukta tekrar eden ilk tesis, ikinci çocukta tekrar eden ilk tesisle yer değiştirilir. Bu işlem, çocuklar uygun hale gelene kadar tekrar edilir. Buna göre yukarıdaki örnekte üretilen çocuklar sonuç olarak aşağıdaki hale gelmektedir:

$$C1 = 17463825$$

$$C2 = 82653471$$

Burada; uygun olmayan ailelerden, uygun çocuklar üretilmesini sağlayan çaprazlamaların geliştirilebileceğini belirtmekte fayda vardır. Fakat bu şekilde bir çaprazlama; rastgele olarak uygun çocuklar üretecek, sonuç olarak hesaplama zamanı ve maliyeti çok yüksek olacaktır.

Çaprazlama olasılığı (P_c), seçilen ebeveyn dizilere çaprazlama işleminin uygulanıp uygulanmayacağını belirler. Literatürde farklı araştırmacılar çaprazlama olasılığını çeşitli değerlerde almışlardır. Li ve Love (2000), yaptıkları çalışmada çaprazlama olasılığı olarak 0.6 değerini kullanmışlardır (Kulluk, 2003).

Mutasyon, çözüm dizisini oluşturan sayıların rastgele olarak değiştirilerek çözüm uzayında yeni bölgelerin araştırılmasına olanak sağlamaktadır. Cheng ve Tasawa (1996), çalışmalarında şu şekilde bir mutasyon işlemi kullanmışlardır. Mutasyon operatörü, daha iyi çocuklar elde etmek için komşu arama teknikleriyle tasarlanır (Kulluk, 2003).

Burada, yerleşimin komşusu olarak en fazla λ sayıda gen değiştirilerek oluşturulan bir yerleşimde, yerleşimler kümesi değişebilir haldedir. Eğer bir yerleşim herhangi başka bir komşudan daha iyi ise λ -optimal olarak adlandırılır. Bunu bir örnekle açıklayacak olursak; Şekil 4.6'daki gibi 2,6 ve 8 olarak 3 gen rastgele seçilmiştir. Ebeveynin seçilen bu genlere

göre komşu kromozomları şekildeki gibidir. Sonuçta bu komşu yerleşimlerin hepsi değerlendirilerek bunlardan en iyi olanı mutasyonda çocuk olarak kullanılır.

Ebeveyn ve seçilen genler:

1 2 3 4 5 6 7 5 9

Komşu kromozomlar ebeveyn ve seçilen genler:

1 2 3 4 5 8 7 6 9

1 8 3 4 5 2 7 6 9

1 8 3 4 5 6 7 2 9

1 6 3 4 5 8 7 2 9

1 6 3 4 5 2 7 8 9

Şekil 4.6 Komşu yerleşim

Mutasyon oranı popülasyonun büyüklüğüne bağlıdır ve uygulanan mutasyon oranının çok yüksek tutulması, oluşturulan başarılı yapıların kolaylıkla yok edilmesine yol açmaktadır. Bu sebepten dolayı araştırmacılar yaptıkları çalışmalarda, düşük mutasyon oranlarını ($P_m = 0,1$ veya daha düşük) kullanmayı tercih etmektedirler. Ayrıca Li ve Love (2000), 30-50 popülasyon büyüklükleri için, mutasyon oranı olarak 0.01 değerini önermektedirler (Kulluk, 2003).

Buna karşılık mutasyon oranları, kullanılan çaprazlama işlemine bağlı olarak farklı sonuçlar verebilmektedir. Örneğin, Suresh vd. (1995) çalışmalarında mutasyon işlemini, söz konusu çözümde rastgele olarak seçilen iki tesisin yer değiştirmesi şeklinde uygulamaktadırlar. Algoritmanın uygulama aşamasında, değişen çaprazlama oranlarına bağlı olarak, 0.25 aralıklarla 0.25-1.00 arasında mutasyon oranları kullanılmıştır. Sonuç olarak, Suresh vd. bazı durumlarda düşük mutasyon oranlarıyla iyi sonuçlar alınmasına rağmen, genellikle en iyi sonuçlara mutasyon oranı olarak 1.00 değerinin kullanıldığı denemelerde ulaşıldığını belirtmektedirler.

4.5 Değerlendirme ve Seçim

Her yeni nesilde, bazı uyum ölçümleri kullanılarak kromozomlar (bireyler) değerlendirilir. Çoğu optimizasyon uygulamasında, uyum orijinal amaç fonksiyonuna göre hesaplanır. Minimizasyon problemleri ele alındığında, her kromozom için amaç fonksiyonu uyum değerine dönüştürülmelidir; böylece daha uyumlu bir kromozom daha yüksek bir uyum değerine sahip olur. Böyle bir işlem Cheng vd.'nin (1996) algoritmalarında basitçe kendi değerinin tersi olarak şu şekilde hesaplanmaktadır

$$eval(v_k) = \frac{1}{f(v_k)} , \quad k= 1,2,..., \text{popülasyon büyüklüğü}$$

Burada $eval(v_k)$ k. kromozomun uyum fonksiyonunu ve $f(v_k)$ bütün parçalar için toplam taşınan miktarı veya parça aileleri arasında maksimum taşınan miktar sayısını göstermektedir.

Daha sonra, uyumlu fonksiyonun sonraki nesilde yeniden üretim için daha yüksek şansa sahip olacağı, mevcut genişletilmiş popülasyona göre gelecek nesli üretmek için temel seçim mekanizması olarak rulet çemberi kullanılır.

5. UYGULAMA: GENETİK ALGORİTMA KULLANARAK YAPISAL ELEKTRİK MALZEMELERİ ÜRETEN BİR FİRMA İÇİN YERLEŞİM TASARIMI YAPILMASI

Bu bölümde, yapısal elektrik malzemeleri imal eden bir firmaya bağlı dış montaj firmalarında kullanılmak üzere geliştirilen ve genetik algoritma kullanılarak oluşturulan bir bilgisayar programının oluşturulma ve gelişim süreci anlatılacaktır.

Oluşturulan algoritma yapısının anlatılmasının ardından, program tanıtılacak; literatür problemleri ile programın test edilmesi gerçekleştirildikten sonra gerçek hayatta yaşanan bir yerleşim problemine program yardımıyla çözüm geliştirilecektir.

Esasen ilgili parametreler değiştirilerek farklı sektör ve uygulama alanlarında da kullanılabilecek olan bu programın kullanımı ile taşıma maliyetlerinin minimizasyonu hedeflenmektedir.

5.1 Tesis Yerleşimi İçin Kareli Atama Problemine Uygun Bir Genetik Algoritma Tasarımı

5.1.1 Problemin tanımı ve varsayımlar

Bütün bölümlerin eşit alanlara sahip olduğu problem, daha önceki bölümlerde de değinildiği gibi Kareli Atama Problemi (KAP) olarak adlandırılır. Tesis yerleşimi problemi ve kareli atama problemlerinin her ikisinin de NP-tam yapıda olmasından dolayı bu problemler için optimum çözümü elde etmek büyük boyutlu problemler için çok zordur.

Bu çalışmada, bütün bölümlerin eşit alana sahip olduğu tesis yerleşimi problemi incelenmiştir. Amaç, iş akışı ve seyahat edilen uzaklığın çarpımı olarak ifade edilen materyal taşıma maliyetini en küçükleyecek şekilde n adet bölümün n adet yerleşime atanmasıdır.

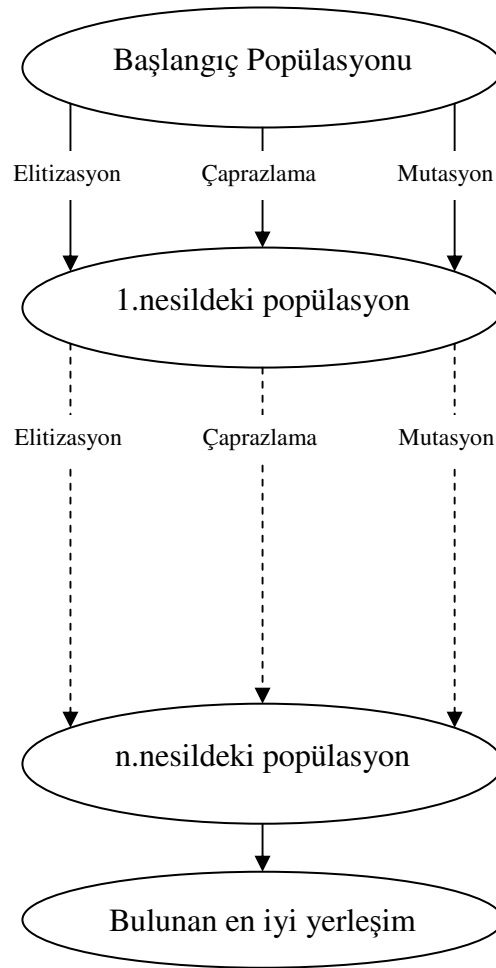
Bu problemde dikkate alınan varsayımlar şunlardır:

- Mevcut çalışma alanlarının hepsi eşit alanlı ve kare biçimindedir.
- Bütün c_{ij} birim maliyetleri "1" değerine sahiptir.
- Her bölüm her alana yerleştirilebilir.
- Yerleşim sayısı bölüm sayısına eşittir.

5.1.2 Temel yapı ve işleyiş

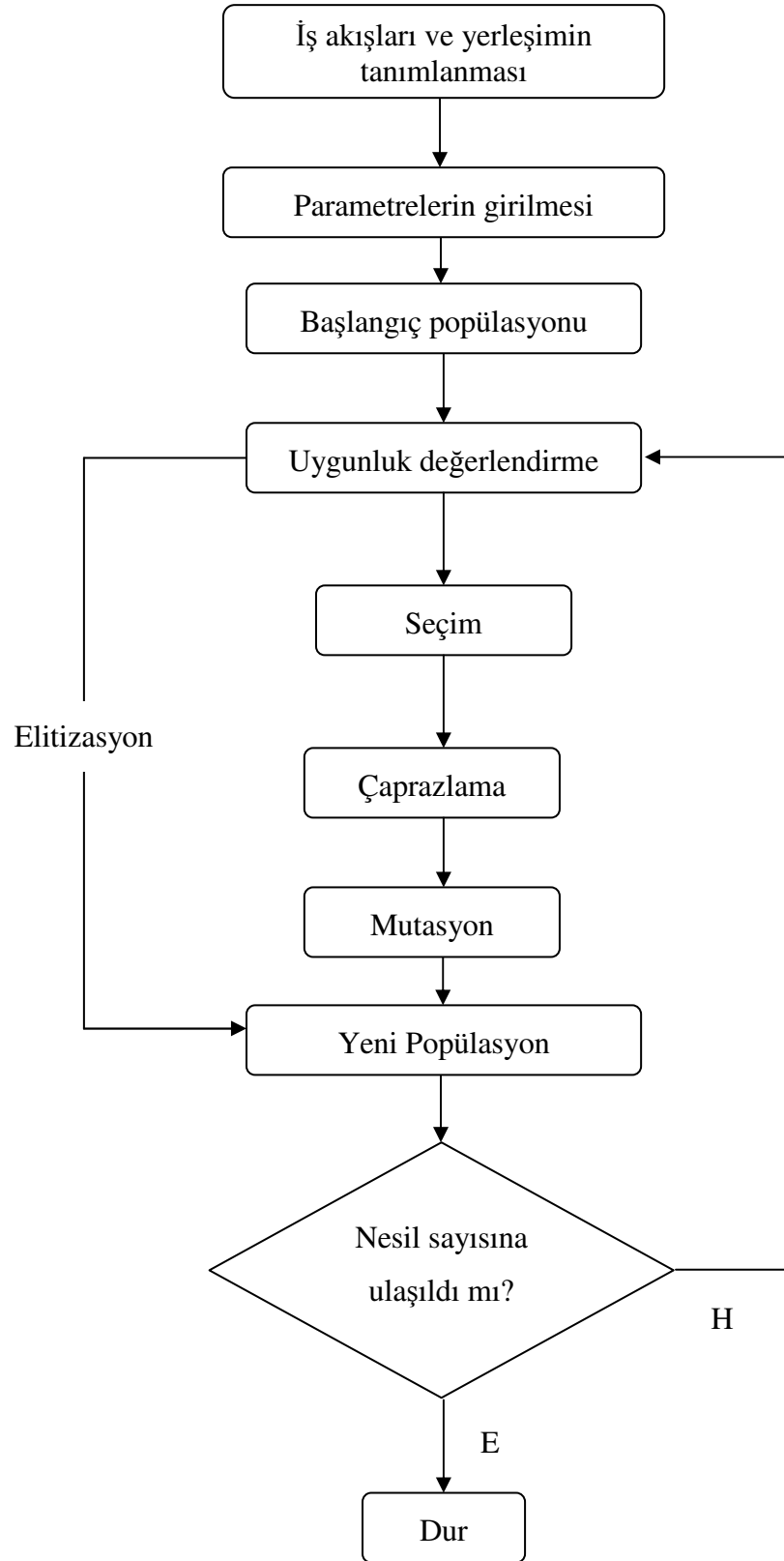
Oluşturulan genetik algoritmanın amacı KAP problemi için minimum malzeme akışı sağlayan yerleşim alternatifinin optimum süre ve uygunlukta bulunmasını sağlamaktır.

Şekil 5.1.'de algoritmanın temel yapısı ve işleyişi gösterilmektedir:



Şekil 5.1 Oluşturulan genetik algoritmanın temel yapısı ve işleyişi

5.1.2.1 Algoritma akış diyagramı



Şekil 5.2 Oluşturulan genetik algoritmanın genel akış diyagramı

5.1.2.2 Kodlama

Tesis yerleşimi probleminde permütasyon dizi gösterimi seçilmiştir. Bu gösterimde hangi bölümün hangi yerleşimde olacağı kolayca görülebilmekte; mutasyon, çaprazlama gibi işlemler yapılarak çözüm alternatifleri aramak kolayca gerçekleştirilebilmektedir.

Tam sayıların kullanıldığı bu gösterimde N bölümlü bir tesis için N adet genin kullanıldığı bir diziliş kullanılmıştır:

1.Dizi: 15 12 1 2 3 6 7 9 4 13 10 5 8 11 14

2.Dizi: 13 1 5 7 4 15 11 9 8 14 12 2 3 6 10

Burada 1.Dizi için 15 rakamı 15.bölümün, 1. tesise; 2.Dizi için 1 rakamı, 1.bölümün 2.tesise yerleştirildiğini göstermektedir. Kısacası rakamlar bölümleri, kromozom içindeki sıraları da kaçınıcı tesiste bulunduklarını göstermektedir.

5.1.2.3 Uyum fonksiyonu

Bu durumda uyum (amaç) fonksiyonu aşağıdaki gibi olmaktadır (5.1):

$$\text{Min} \sum_i \sum_j (f_{ij} c_{ij}) d_{ij} \quad (5.1)$$

f_{ij} i bölümünden j bölümüne iş akışı

c_{ij} i bölümünden j bölümüne birim taşıma maliyeti

d_{ij} i yerleşiminden j yerleşimine uzaklık

5.1.2.4 Başlangıç popülasyonu

Başlangıç popülasyonu ve popülasyon büyüklüğü GA'ların verimliliğini etkileyen iki önemli etmen olup bu iki etmen dikkate alınarak algoritmanın performansı arttırmak mümkündür.

Bu çalışmada literatürde yaygın olduğu üzere başlangıç popülasyonu rastgele olarak oluşturulmuştur.

5.1.2.5 Elitizm

Popülasyonda o ana kadar bulunan en iyi çözümün diğer nesillerde de bulunmasını ihtimale bırakmamak için elitizm fonksiyonu kullanılmıştır. Bu fonksiyonda popülasyondaki en büyük uyuma sahip elitizasyon oranındaki birey hiçbir genetik operasyona tabi tutulmadan

yeni popülasyona aktarılır

5.1.2.6 Seçim mekanizması

Oluşturulan popülasyon içinden bireyler Goldberg (1989)'un belirttiği rulet tekerleği seçim yöntemine göre uygunlukları ile ihtimallendirilerek rastgele olarak seçilmiştir.

5.1.2.7 Çaprazlama yöntemi

Geliştirilen genetik algorithmada iki noktalı çaprazlama kullanılmıştır. Çaprazlama şu şekilde yapılmaktadır:

- Seçim mekanizması ile iki birey ebeveyn olarak seçilir.
- Çaprazlama noktası olarak rastgele iki sayı türetilir ve bu sayılara göre iki ebeveyn ve boş çocuk diziler üç parçaya ayrılır,
- Birinci çocuk dizinin 1. ve 3. parçasındaki genler, doğrudan birinci ebeveynin 1. ve 3. parçasından kopyalanır. İkinci çocuk için de benzer işlem 2. ebeveyninden gerçekleştirilir.,
- Birinci çocuk dizinin orta parçası ikinci ebeveyn diziden atanır. Ancak atanacak elemanların daha önce çocuk dizide atanmamış elemanlar olması şarttır, ikinci çocuk dizi için ise aynı işlem birinci ebeveyn diziden atanarak sağlanır,
- Birinci çocuk dizi için orta parçada kalan boş pozisyonlar ikinci ebeveynin 1. ve 3. parçasındaki atanmayan genlerin boş genlere sıra ile atanmasıyla elde edilir. İkinci çocuk dizi için ise boş pozisyonlar aynı şekilde birinci ebeveynin 1.ve 3. parçasındaki atanmayan genlerin boş genlere sıra ile atanmasıyla elde edilir.

Örneğin ebeveyn olarak seçilen iki dizi P_1 : 8-3-4-5-6-7-1-2 ve P_2 : 3-4-7-6-8-5-2-1 olsun. Rastgele üretilen çaprazlama noktaları ise 2 ve 5 olsun. Bu durumda çaprazlama ile çocuk diziler şu şekilde elde edilir:

1.Aşama: Çaprazlama noktalarının tespiti

P_1 :	8	3	4		5	6	7		1	2
P_2 :	3	4	7		6	8	5		2	1
$\Ç_1$:	-	-	-		-	-	-		-	-
$\Ç_2$:	-	-	-		-	-	-		-	-

2.Aşama: Ebeveynlerin 1. ve 3. kısımlarının çocuk dizinlere kopyalanması

Ç₁: 8 3 4 || - - - || 1 2

Ç₂: 3 4 7 || - - - || 2 1

2.Aşama: Ebeveynlerin 1. ve 3. kısımlarının çocuk dizinlere kopyalanması

Ç₁: 8 3 4 || - - - || 1 2

Ç₂: 3 4 7 || - - - || 2 1

3.Aşama: Ebeveyn dizilerden 2. parçadaki atanmamış genlerin kopyalanması

Ç₁: 8 3 4 || 6 - 5 || 1 2

Ç₂: 3 4 7 || 5 6 - || 2 1

4.Aşama: Boş pozisyonların doldurulması

Ç₁: 8 3 4 || 6 7 5 || 1 2

Ç₂: 3 4 7 || 5 6 8 || 2 1

5.1.2.8 Mutasyon

Geliştirilen genetik algorithmada mutasyon rastgele olarak seçilen bir bireyin başka bir rastgele olarak seçilen bireyle yer değiştirilmesi ile sağlanır. "P_m" parametresi mutasyon olma ihtimalini belirler

5.1.2.9 Nesil sayısı

Programda bir parametre olup kullanıcı tarafından girilir. Problemin uygun çözümü için araştırılarak bulunur.

5.2 Programın Tanıtılması

Hazırlanan bilgisayar programı C++ yapısal programlama dili ile oluşturulmuş olup "<http://lancet.mit.edu/ga/>" adresindeki genetik algoritma kütüphanesinden temin edilen kodlara ihtiyaçlara göre eklemeler yapılarak oluşturulmuştur.

Program uygulama yapılan ana firma ile çalışan küçük montaj firmalarında kullanılacağından SCLO (Subcontractor Layout Optimizer -Tedarikçi Yerleşim Optimize Edici) olarak

isimlendirilmiştir.

5.2.1 Problemin programa girilmesi

Şekil 5.3 SCLO’da problemin tanıtılması

Problem SCLO’ya tanıtılırken birkaç farklı yol izlemek mümkündür. Bunlar:

- **Elle giriş:** Şekil 5.3’deki kullanıcı ara yüzünde Tesis Sayısı (N) bölümüne istenen rakam girilir. Girilebilecek en büyük değer ihtiyaç gereği 100 olarak tanımlanmıştır.
- **Txt dosyası girişi:** Girişler→ Dosya bölümüne ilgili yol ve önceden hazırlanmış txt dosyası adı yazılarak problemin tanıtılması sağlanır (Şekil 5.3).

Kullanılan matrisler şunlardır:

- **İş akış matrisi:** i bölümünden j bölümüne ne kadar bir iş akışı olduğu burada belirlenir. Bu problemde, matris diyagonala göre simetrik olmayabilir (i’den j’ye iş akışı, j’den i’ye iş akışına eşit olmayabilir). Ancak diyagonal üzeri (i.tesisten i. tesise iş akışı 0’dır)
- **Uzaklık matrisi:** i yerleşimden j.yerleşime olan uzaklık burada tanımlanır. Bu problemde matris diyagonala göre simetrik olmak zorundadır. (i. yerleşimden j.yerleşime olan uzaklık ile; j.yerleşimden i. yerleşime olan uzaklık aynıdır). Ancak diyagonal üzeri (i. yerleşimden i. yerleşime olan uzaklık 0’dır)

İş akışı ve uzaklık matrisleri üç şekilde belirlenebilir:

- Elle girişte tesis sayısı girildikten sonra akış matrisi açılarak ilgili kutulara akışlar girilir. Şekil 5.4’te 5. tesisten 7.tesise iş akışı 0 olarak görünmektedir.

0	5	8	12	15	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	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	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	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	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	0	0	0	0	0	0

Şekil 5.4 Elle girişte akış matrisi

- İş akış matrisi oluşturulan txt dosyasından okunabilir. Şekil 5.5’de literatürdeki nug12 problemi akış matrisinin dosyadan okunmuş hali görülmektedir.

0	5	2	4	1	0	0	6	2	1	1	1
5	0	3	0	2	2	2	0	4	5	0	0
2	3	0	0	0	0	0	5	5	2	2	2
4	0	0	0	5	2	2	10	0	0	5	5
1	2	0	5	0	10	0	0	0	5	1	1
0	2	0	2	10	0	5	1	1	5	4	0
0	2	0	2	0	5	0	10	5	2	3	3
6	0	5	10	0	1	10	0	0	0	5	0
2	4	5	0	0	1	5	0	0	0	10	10
1	5	2	0	5	5	2	0	0	0	5	0
1	0	2	5	1	4	3	5	10	5	0	2
1	0	2	5	1	0	3	0	10	0	2	0

Şekil 5.5 Akış matrisinin txt dosyasından okunması

- Rastgele oluşturma; programın eğitim aşamasında kullanıcının hızlı problem tasarlayabilmesi amacıyla oluşturulmuştur. Örneğin tesis sayısı olarak 12 girildikten sonra akış butonu hizasındaki rastgele kutucuğu işaretlenir ve akışların hangi değer aralığında istendiği girilir. 0-20 aralığı için rastgele oluşturulan bir akış matrisi Şekil 5.6’da gösterilmiştir.

5.2.2 Parametreler

Şekil 5.7’deki kullanıcı ara yüzünden de görüleceği üzere kullanıcı tarafından belirlenebilen parametreler şunlardır:

0	13	4	18	5	11	6	9	6	4	12	19
14	0	4	7	18	9	4	7	15	12	9	15
9	9	0	2	11	4	5	12	11	10	12	18
20	4	14	0	3	20	0	15	10	0	1	8
0	9	9	14	0	9	7	16	12	16	0	14
10	8	3	6	14	0	14	0	4	18	3	6
16	17	10	18	9	10	0	7	6	16	4	17
9	13	8	16	9	17	11	0	2	5	2	5
10	10	1	6	16	14	1	18	0	17	1	9
17	2	17	13	9	7	0	1	3	0	7	20
3	8	0	9	7	5	5	10	1	0	0	19
19	14	4	20	14	4	17	17	17	4	11	0

Şekil 5.6 Akış matrisinin rastgele oluşturulması

- **Çaprazlama Oranı (Pc):** %0-100 arasında girilebilir. %80-90 arası uygun görünmektedir.
- **Mutasyon Oranı (Pm):** %0-100 arasında girilebilir. %5-15 arası uygun görünmektedir.
- **Elitizm Oranı (Pe):** %0-100 arasında girilebilir. %5-15 arası uygun görünmektedir.
- **Nesil Sayısı (N):** İstenen sayı girilebilir. Tüm diğer parametreler sabit olduğunda belirli bir sayıdan sonrası nesilde iyileşmeye yol açmamakta işlem süresinin uzamasına yol açmaktadır. En fazla 100.000 uygun görünmektedir.

Şekil 5.7 SCLO kullanıcı arayüzü

- **Popülasyon Genişliği (Np):** İstenen popülasyon büyüklüğü buradan girilir. Tahmin edilenin aksine küçük popülasyonlar (20-50 arası) daha verimli sonuçlar verebilmektedir.

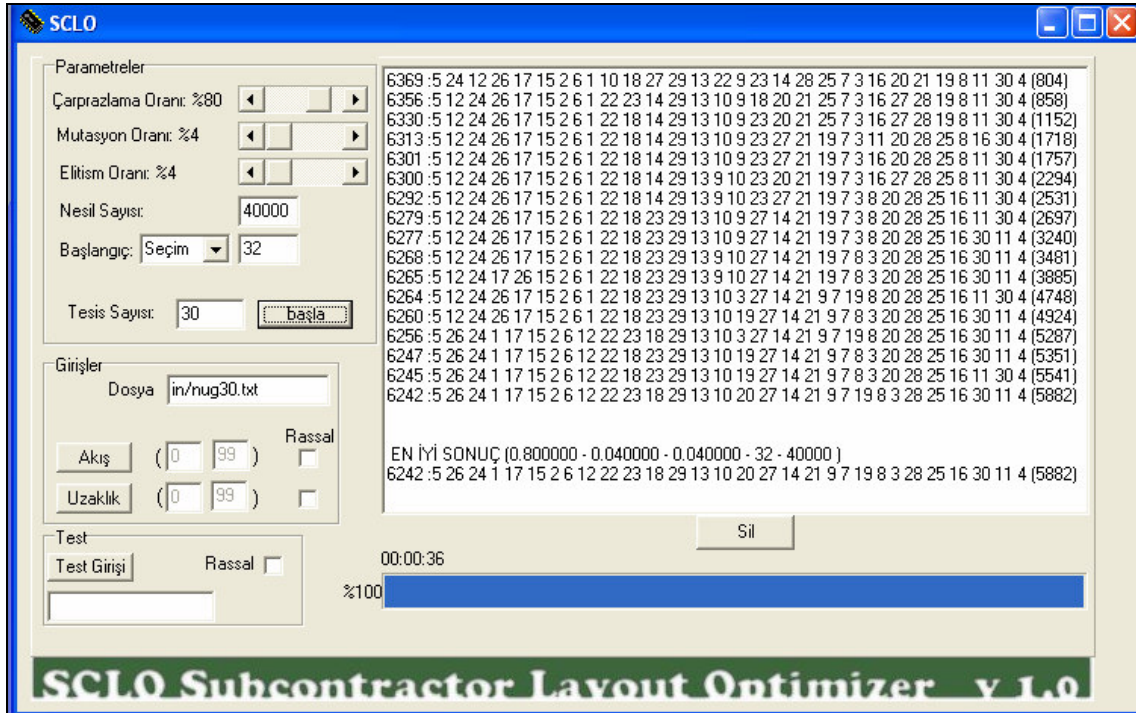
5.2.3 Programın çalıştırılması

Problem SCLO'ya tanıtıldıktan ve ilgili parametreler belirlendikten sonra “Başla” butonuna basılarak program çalıştırılır.

5.2.4 Rapor formatı ve yorumlanması

Sonuç olarak çıkan raporda SCLO daha iyi çözüm bulduğu her nesili raporlar. Burada sırasıyla bulunan çözümün uygunluk değeri, bulunan yerleşim planı ve bunun kaçınıcı nesilde bulunduğu ekrana raporlanır. Nesil sayısının raporlanması programa büyük bir yük getirmemekle birlikte nesil sayısı parametresinin belirlenmesinde avantaj yaratır. Şekil 5.8'deki raporda bulunan aşağıdaki çözüm alternatifinde 6245 uygunluk değerini; 5 ile başlayıp 4 ile biten sıralama 1. yerleşimden 30. yerleşime kadar hangi bölümün hangi yerleşime yerleştirileceğini belirtir. Örneğin 5. bölüm 1. yerleşime yerleştirilecektir. Sondaki 5541 sayısı bu sonuca 5541. nesilde ulaşıldığını göstermektedir.

6245 :5 26 24 1 17 15 2 6 12 22 18 23 29 13 10 19 27 14 21 9 7 8 3 20 28 25 16 11 30 4
(5541)



Şekil 5.8 SCLO sonuç raporu

Bulunan eni iyi sonuç şu şekilde raporlanır:

EN İYİ SONUÇ (0.800000 - 0.040000 - 0.040000 - 32 - 40000)

6242 :5 26 24 1 17 15 2 6 12 22 23 18 29 13 10 20 27 14 21 9 7 19 8 3 28 25 16 30 11 4
(5882)

Burada parametre değerleri bulunan en iyi çözümleri kayıt altına almak amacıyla ekrana yazdırılmakta olup aşağıdaki değere sahiptir:

Çaprazlama Oranı (Pc): %80

Mutasyon Oranı (Pm): %4

Elitizm Oranı (Pe): %4

Popülasyon Genişliği (Np): 32

Nesil Sayısı(N): 40.000

Burada kaçınıcı nesilde nasıl bir değişim olduğuna bakılarak parametreler yenilenebilir. Örneğin pek çok nesil boyunca iyileşme olmuyor ve yerel optimumda takılıp kalınıyorsa mutasyon oranı arttırılabilir, ya da sondaki nesillerde iyileşmenin devam ettiği görülüyorsa nesil sayısı arttırılabilir.

5.2.5 Doğrulama

Programaya yazılan küçük bir eklenti ile bulunan alternatifin uyumluluk değerini tekrar hesaplamak mümkündür. Burada ilk başta tesisler 1'den n'e kadar sıralanmış olarak gelir. İlgili kutucuğa yol ve dosya adı yazılarak txt dosyasından çözüm yüklemek de mümkündür. Şekil 5.9'da n=30 için sıralı olarak çözüm alternatifinin uygunluğu görünmektedir. Sıralama değiştirilerek elle uygunluk hesaplamak amacıyla kullanılabilir.

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
8060																													

Tamam

Şekil 5.9 SCLO test girişi ile doğrulama

5.3 Literatür Problemleri ile Programın Test Edilmesi

Programın düzgün çalıştığı literatür problemleri ile test edilmiştir. Testin başarılı olabilmesi için bugüne kadar bulunan en iyi çözümlere %99 oranında yaklaşılması yeterli görülmüştür.

Problemler QAP library'den (KAP kütüphanesi) “<http://www.seas.upenn.edu/qaplib/>” adresinden alınmıştır.

5.3.1 Belirlenen literatür problemleri

Yapılan incelemede iş akış ve uzaklık matrisleri açısından NUG, SKO, WIL, SCR ve THO problemleri programın denenmesi için uygun görünmektedir.

Bunlar içerisinde NUG27, NUG28, NUG30, SKO42, SKO49, SKO56, WIL50, SCR20, THO30, THO40 denenmiştir.

Yapılan pek çok deneme sonucu elde edilen veriler ve bugüne kadar bulunan en iyi çözüme yaklaşma yüzdeleri Çizelge 5.1’de gösterilmiştir.

Çizelge 5.1’den de görüleceği üzere tüm problemlere %99’un üzerinde yanaşılmıştır. Burada pek çok deneme ile sonuca ulaşılmıştır. Çözüm verimliliğini arttırmak daha fazla deneme yapılması ile mümkündür. Programın çalıştığı doğrulandığından daha fazla deneme yapılmaya gerek görülmemiştir.

5.3.2 Bulunan parametre ve yerleşimler

Literatür problemleri için bulunan parametre ve yerleşim alternatifleri Ekler’de verilmiştir (Ek-1).

Çizelge 5.1 SCLO programının seçilen literatür problemleri ile test edilmesi

Literatür Problemi	Bilinen En İyi Çözüm	%99 ile Yakalanması Gereken Hedef Maliyet	SCLO ile Yakalanan Maliyet-	Yakalama Yüzdesi	Doğrulama
NUG27	5.234	5.287	5.272	99,27%	OK
NUG28	5.166	5.218	5.180	99,73%	OK
NUG30	6.124	6.186	6.133	99,85%	OK
SKO42	15.812	15.972	15.936	99,22%	OK
SKO49	23.386	23.622	23.604	99,07%	OK
SKO56	34.458	34.806	34.785	99,05%	OK
WIL50	48.816	49.309	49.197	99,22%	OK
SCR20	110.030	111.141	110.676	99,41%	OK
THO30	149.936	151.451	151.312	99,08%	OK
THO40	240.516	242.945	242.130	99,33%	OK

5.4 Yapısal Elektrik Sektöründe Programın Kullanılması

5.4.1 Sektör ve firma tanıtımı

Yapısal elektrik malzemeleri sektörü; binalarda kullanılan her türlü elektrikli malzemenin imalat ve satışı ile ilgilenmektedir. Kablo, elektrik saati, telefon prizinden kablo kanallarına binalarda kullanılan her tip elektrik malzemesi bu sektör bileşenleri tarafından üretilir, üretilir ve satılır.

Çalışmanın gerçekleştirildiği firma; alçak gerilim elektrik malzemeleri konusunda faaliyet gösteren dünya çapında bir grubun Türkiye imalat ve satış operasyonlarını yürütmektedir. Bağlı bulunan uluslararası grup; stratejisi gereği orta ve yüksek gerilim ürünlerinin imal ve satışı ile ilgilenmemektedir.

Firmanın bağlı bulunduğu grup, 2007 rakamlarına göre; dünya çapında 70 ülkede üretim merkezi, 160'dan fazla ülkede 3 milyar Euro'luk iş hacmi, yaklaşık 26.000 çalışan ve 130.000 ürün referansı ile alanındaki en büyük uluslararası firmalardan biridir. Dünya pazarında anahtar ve priz grubunda %18'lik, kablo taşıma sistemlerinde %18'lik paya sahiptir.

Firma Türkiye'de, 40 dönüm arazi üzerine kurulu 15 dönüm kapalı alanı bulunan fabrikası, 5 bölge satış müdürlüğü, 7 bölge satış temsilciliği ve 400'ü aşkın çalışanıyla imalat yapmakta; satış ve satış sonrası servis hizmetleri vermektedir.

Ürün gamı içerisinde, evler, ofisler ve endüstriyel alanlar için değişik tip ve özelliklerde, koruma üniteleri, kablolu sistemleri, anahtar, fiş ve prizler, şalter ve kabinler bulunmaktadır.

Araştırmanın gerçekleştirildiği fabrikada, çeşitli montaj birimleri bulunmakla birlikte teknik detay gerektirmeyen ve düşük risk taşıyan bir kısım işler, maliyetleri düşürmek amacıyla dış montaj firmalarına verilmektedir.

Bu dış montajcılar sadece firma ile çalışmakta olup başka firmalara yönelik imalatları yoktur. Gerekli tüm teknik destek (kalite, metot, bakım vb.) ana firma tarafından karşılanmakta olup üretimde optimum düzeyde kalite ve maliyet karşılanmaya çalışılmaktadır.

Dış montajda üretilen ürünler arasında çeşitli tip ve değerlerde sigorta, kaçak akım koruma, devre kesici, anahtar-priz grupları, buton, elektrik kutusu, duvar ve şalterler bulunmaktadır.

5.4.2 Firmada yaşanan problem ve modelleme varsayımları

Bahsedilen bu büyük yapı içerisinde, yeni imalatına başlanacak veya grubun faaliyet gösterdiği yüksek işçilik maliyetleri bulunan ülkelerden transfer olacak olan ürünlerin grup bünyesindeki üretim merkezlerinden hangisi veya hangilerinde imal ettirileceği konusunda büyük bir rekabet yaşanmakta; her ülke yöneticileri işleri ve yatırımları kendi ülkesine çekmeye çalışmaktadır. Pek çok üründe malzemelerin büyük çoğunluğunun (tahminen %80'i) grup bünyesindeki merkezi satınalma tarafından sağlandığı veya grup bünyesindeki üretim düzeyi yüksek merkezlerde imal ettirilerek, merkezi bir lojistik organizasyonu ile dünya çapındaki ilgili üretim merkezlerine dağıtıldığı düşünüldüğünde; rekabet edebilme konusunda ülkesel düzeyde işçilik ve verimlilik faktörü öne çıkmaktadır.

Bu çalışmanın amacı dış montaj (DM) firmalarındaki yerleşimi yeniden düzenleyerek taşıma maliyetlerinden kazanç sağlamaktır.

5 adet DM vardır. Uygulamanın devamında bunlar: H51, L52, M54, O67, R69 şeklinde belirtilecektir. DM'lerdeki toplam ve kullanılabilir alanlar Çizelge 5.2'de verilmiştir.

Kullanılabilir alan kavramı işlerin yerleştirileceği alanları temsil etmektedir. Toplam alanlardan ofis, yemekhane, merdiven boşlukları, soyunma odası, tuvaletler gibi imalatla kullanılamayan alanlar çıkarılarak bulunmuştur.

Bölümler ihtiyaç duydukları ara koridorlarla birlikte düşünülmüş olup; KAP'ye göre her biri 10 m² 'lik kare alanlardan oluşturulmuştur. Buna göre bölümler 3,16 m x 3,16 m = 10 m² 'lik büyüklüklere sahiptir.

Her birim karede bir bölüm ve onun ihtiyaç duyacağı koridor gereksinimi mevcuttur. Malzeme ve bitmiş ürün stok alanları da ayrı bir bölüm olarak tanımlanmıştır. Sütunlar ve basınçlı hava, su, elektrik tesisatı vb alt yapısal kısıtlar ihmal edilmiştir. DM'lere göre oluşturulan bölüm (iş birimi -tesis) sayısı Çizelge 5.2'de gösterilmiştir.

Çizelge 5.2'de alan hesaplamaları yapılırken iki ayrı katta (H51, O67, R69) veya iki ayrı binada (L52) parçalı olarak sürdürülen işler hesaplamalarda özellikle ayrıntılı verilmiş olup uzaklık matrisleri buna göre oluşturulmuştur. Sadece M54 tek katta bütün bir yerleşime sahiptir.

Çizelge 5.2'den görüleceği üzere DM'lerin kullanılabilir alanları 290 - 500 m² arasında değişmekte; Her DM için 29-50 arası bölüm söz konusu olmaktadır.

Çizelge 5.2 DM'ler ve alanları

Dış Montaj Birimi	Toplam Alan (m ²)	Kullanılabilir Alan (m ²)	Birim Sayısı
H51	150 + 270 = 420	100 + 190 = 290	10 + 19 = 29
L52	330 + 110 = 440	260 + 100 = 360	26 + 10 = 31
M54	560	460	46
O67	295 + 365 = 660	210 + 290 = 500	21 + 29 = 50
R69	340 + 170 = 510	300 + 70 = 370	30 + 7 = 37

H51, O67, R69 DM'lerinde 2 vardiya uygulanmasına rağmen iş dağılımında bu durum ihmal edilmiştir.

5.4.3 Modelleme aşamaları

İlk olarak her DM için birim sayısı kadar bölüm seçilmiştir. Bunlar aşağıdaki gibi kodlanmıştır:

DM kodu_x

x= 1'den n'e kadar bölümler

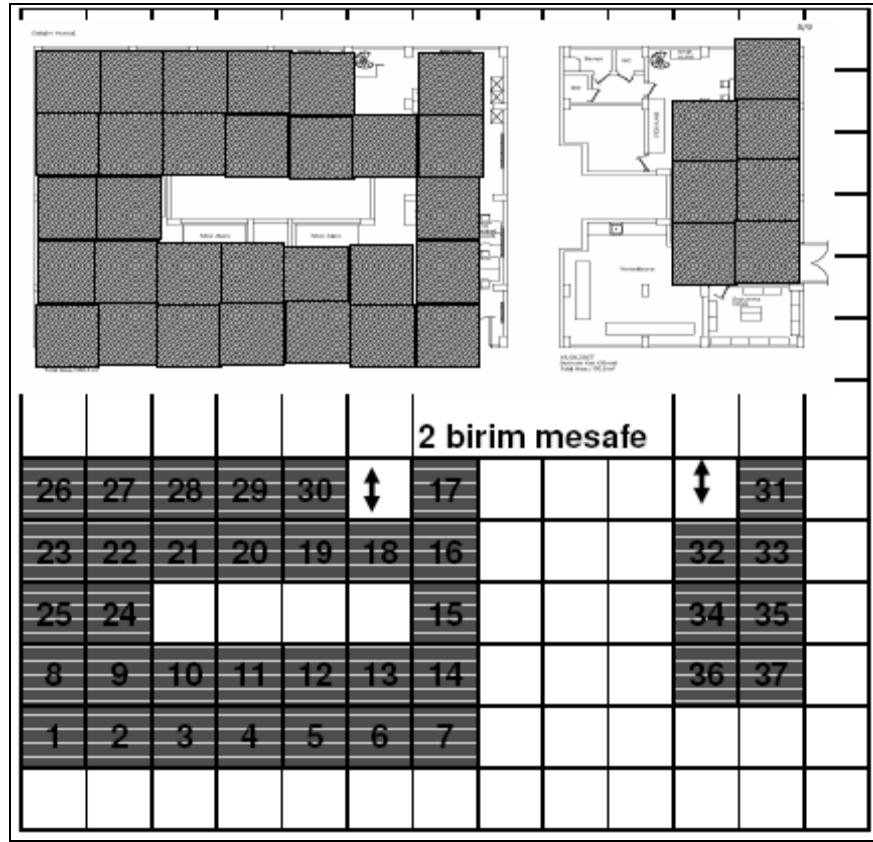
Bölümler (Ek-2) ve bölümler arası iş akış matrisleri (Ek-4) her DM için ekler bölümünde mevcuttur.

Sonraki aşamada autocad ortamında çizilmiş olan layoutlarda kullanılabilir alanlar yine autocad ortamında önceden anlatılan ilkelere göre 10 m²'lik n parçalara bölünmüştür (Şekil 5.10). Tüm DM'ler için oluşturulan ilk yerleşimler ekler bölümünde mevcuttur (Ek-3).

Oluşan yerleşimin bir benzeri excel ortamında oluşturulmuş ve numaralandırılmıştır. Kat ve bina geçişlerinde kaç birim uzaklık varsa bunlar uzaklık matrisi oluşturulurken dikkate alınmıştır (Şekil 5.10).

Bölümler mevcut durumda i. yerde i. bölüm olacak şekilde yerleştirilmiştir. Yani ilk olarak 1.

yerde 1. bölüm 15. yerde 15. bölüm vardır (Şekil 5.10)



Şekil 5.10 Autocad ve Excel ortamında örnek modelleme

Yerleşimler arası uzaklık matrisi oluşturulurken Monte-carlo uzaklık hesaplama yöntemi kullanılmıştır. Örneğin; 1 ile 2 yerleşimleri arasında 1 birim uzaklık varken 1 ile 3 yerleşimleri arasında 2 birim, 1 ile 9 yerleşimleri arasında 2 birim, 1 ile 24 arasında 3 birim uzaklık bulunmaktadır. Şekilde ok (↑↓) ile gösterilen 18 ve 32 numaralı birimler katlar arası geçişte kullanılan bölümler olup uzaklık hesaplanırken sayılan karelere buradaki örnekte 2 birim eklenmiştir.

Her DM için oluşturulan yerleşimler arası uzaklık matrisleri ekler bölümünde mevcuttur (Ek-5).

5.4.4 Mevcut durum analizi

Her DM için programın doğrulama bölümünde anlatıldığı şekilde mevcut yerleşime göre bulunan taşıma maliyetleri ve tüm DM'ler için oluşan toplam taşıma maliyetleri Çizelge 5.3'te gösterilmiştir.

Tüm DM'ler için toplam taşıma maliyeti 26.750 olarak bulunmuştur.

Çizelge 5.3 Mevcut durumda taşıma maliyetleri

Dış Montaj Birimi	Birim Sayısı	Mevcut Taşıma Maliyeti	Toplam Taşıma Maliyeti
H51	29	3220	26.750
L52	31	5545	
M54	46	8550	
O67	50	5675	
R69	37	3760	

5.4.5 Oluşturulan yeni durum

Program, her DM için pek çok kez çalıştırılarak en uygun parametreler ve bunlara bağlı olarak minimum maliyetler bulunmaya çalışılmıştır. Bulunan çözümler ile maliyetlerde yaşanan düşüş Çizelge 5.4'te gösterilmiştir.

Çizelge 5.4 SCLO kullanılarak bulunan yeni durumda taşıma maliyetleri

Dış Montaj Birimi	Mevcut Taşıma Maliyeti	Yeni Oluşturulan Yerleşimde Taşıma maliyeti	İyileşme Yüzdesi	Toplam Yeni Maliyet	Toplam İyileşme Yüzdesi
H51	3220	1630	%49	15.743	%41
L52	5545	3119	%44		
M54	8550	5630	%34		
O67	5675	3215	%43		
R69	3760	2149	%43		

Çizelge 5.4'te görüleceği üzere DM'lerde %34-49 arası maliyet düşüşü yaşanmıştır. Toplam taşıma maliyeti 26.750'den 15.743'e düşürülerek %41 iyileşme sağlanmıştır. Daha fazla uğraşarak maliyetlerde düşüş arttırılabilir.

Sonuç olarak başlangıçta 1'den n'e kadar sırasıyla yerleşen DM'lere göre bulunan yeni bölüm yerleşimleri ve uygulanan parametreler (Pc, Pm, Pe, Np, N) şöyle olmuştur:

- H51 için (n= 29)

Uygunluk değeri: 1630

Parametreler: 0.850000 - 0.060000 - 0.050000 - 32 - 10000

Sıralama: 26 27 24 25 28 3 2 21 6 5 19 29 18 10 17 12 13 22 11 14 23 20 4 16 1
7 15 8 9

- L52 için (n= 36)

Uygunluk değeri: 3119

Parametreler: 0.850000 - 0.060000 - 0.050000 - 32 - 10000

Sıralama: 2 24 23 21 20 7 25 26 35 22 3 11 19 34 36 33 28 30 8 6 4 31 29 27 9
32 1 10 13 14 16 18 12 15 17 5

- M54 için (n= 46)

Uygunluk değeri: 5630

Parametreler: 0.830000 - 0.040000 - 0.040000 - 20 - 20000)

Sıralama: 24 25 28 29 20 9 8 7 27 10 26 34 12 14 3 2 19 4 30 31 21 22 5 6 45 11
41 16 40 46 35 39 37 32 13 1 44 42 36 15 17 38 43 33 23 18

- O67 için (n= 50)

Uygunluk değeri: 3215

Parametreler: 0.830000 - 0.040000 - 0.040000 - 20 - 90000

Sıralama: 27 28 26 25 29 23 21 20 19 6 4 3 24 16 13 5 1 2 15 14 7 8 22 10 9 11
18 17 12 47 31 30 38 39 32 48 50 33 34 49 36 43 46 42 37 44 35 40

41 45

- R69 için (n= 37)

Uygunluk değeri: 2149

Parametreler: 0.850000 - 0.040000 - 0.050000 - 500 - 10000

Sıralama: 9 24 10 2 3 4 14 8 12 11 1 6 5 19 30 16 17 15 18 20 23 7 21 13 25 22
26 27 28 29 31 32 33 37 35 34 36 (4159)

6. SONUÇLAR ve ÖNERİLER

Uygulamanın sonucuna bakıldığında iyileşme yüzdesi %41 ile oldukça fazladır. Bu durum yerleşimde iyileştirme sağlanmasının yanı sıra; mümkün olduğunca aslına uygun tasarlanmaya çalışılan problemin, modelleme aşamasında gerçek yaşamda dikkate alınan bazı kısıtları içermemesinden de kaynaklanmaktadır. Çeşitli sebeplerle her tesis her yere yerleşemeyebilmektedir. Örneğin kapılara yakın alınan depo alanları problem çözümünde ortaya kaymış olup, bu durum dış malzeme akışı dikkate alınmayıp ihmal edildiği için yaşanmıştır. KAP'nin yapısı gereği birbiri ile iş akışı olan bölümler sabit kalamayacağından depoların ortaya kaymasının önüne geçilememiştir. Kısacası dışarı ile malzeme alışverişi (malzemenin DM'ye ilk girişi veya bitmiş malzemenin yüklenmek üzere dışarı çıkarılması gibi) dikkate alındığında oluşturulan yeni çözümler daha maliyetli olabilir.

Algoritma ve programın gelişebilecek yönlerini şöyle özetlemek mümkündür:

Bu çalışmada literatürde yaygın olduğu üzere başlangıç popülasyonu rastgele olarak oluşturulmuştur. Bu durumun olumsuzluğu, yerel optimuma takılma ve istenen verimlilikte bir çözüme ulaşılamamasıdır. Son yıllarda yapılan çalışmalar literatürdeki mevcut sezgisellerden yararlanarak oluşturulan başlangıç popülasyonları ile daha verimli çözümlerin yakalanmasının mümkün olduğunu göstermektedir. Burada da çeşitli sezgiseller oluşturularak çözümün uygunluğu arttırılabilir.

Nesil sayısı, programda bir parametre olup mevcut durumda kullanıcı tarafından girilmektedir. Problemin uygun çözümü için araştırılarak bulunur. Bu durumun özellikle büyük nesillerde ($n=32$ için $N>40.000$ gibi ve üzeri durumlarda) iyileşme devam etmese bile programın işlem yapmaya devam etmesine yol açtığı görülmüştür. Bu durumun, belirlenen miktar veya yüzde kadar iyileşme yoksa programın durdurulması yönünde bir ekleme yapılarak ortadan kaldırılması mümkün olabilir. Böylece nesil sayısı kullanıcı tarafından girilmeyecek program iyileşme yoksa duracaktır. Bu durum nesil sayısının elle girilmesi istenen durumlarda iyileşme gerçekleşmiyorsa manuel olarak programı durdurabilecek bir buton eklenmesi ile de sağlanabilir.

Paralel yapı, gelişmiş teknikler (Goldberg, 1989) bu probleme adapte edilebilecek nitelikte özelliklerdir. Paralel yapı; birbirine paralel çalışan birden fazla algoritmanın aynı sonucu farklı kollardan araması ve birbirileri ile birey alışverişinde bulunmasını sağlayarak çözüme ulaşma konusunda avantaj yaratmaktadır. Yerel optimumlara takılmanın önüne geçmek için önemli bir araçtır.

İşletmenin taşıma maliyeti en aza indirme probleminin çözümünde şunlara dikkat edilebilir:

İşlerin dış montajlara dağıtılması dış montajın özellikleri (önceki ve şu anki tecrübeleri, hangi tip işlerde verimli ve eğitimli oldukları, kalite oranları, risk faktörü vb.) dikkate alınarak tecrübe ile dağıtılması ve problemin çözümüne buradan başlanması esasen yetersiz bir durum olup, global olarak tüm dış montajcılar ve tüm işler arasında bir çözüm üretmek daha faydalı sonuçlar üretebilir. Örneğin işlerin direk DM'ler için seçilmeden önce, ilgili kısıtlar oluşturularak hangi işlerin hangi dış montajlara gideceğinin belirleneceği bir çalışma oldukça faydalı olabilir.

Şüphesiz gerçek hayatta yaşanan kısıtların çoğunu modellemek mümkün olamamakta ve varsayımlarla sınırlar çizilmektedir. Duruma göre bu varsayımlar sebebiyle bulunan çözümlerin gerçek hayatta uygulanabilirliği riske girmektedir. Gerekli kabuller gerektiği kadar yapıldığında bu tip çalışmalar %100 uygulanabilir sonuçlar üretmeseler bile büyük oranda uygulanma şansına sahiptirler.

H51, O67, R69 bölümlerinde 2 vardiya uygulanmasına rağmen iş dağılımında bu durum ihmal edilmiştir. Bu durum da ilerleyen aşamalarda hesaplamalara dahil edilebilir.

Bunlara ek olarak: KAP yaklaşımının yanı sıra farklı büyüklükteki alanlarla da çözüm oluşturulabilir. Burada bu durumun önüne farklı alan ihtiyaçlarına sahip bölümler eşit karelere bölünerek ve iş akışı buna göre tarifleterek çözüm bulunmaya çalışılmıştır.

Ayrıca n sayıda çalışma alanına sahip her işletme için k ($k \leq n$) adet de iş tanımlayıp. Boş alan yüzdeleri de incelenebilir.

Günümüz bilgisayar donanımları ne kadar güçlü olursa olsunlar hala belli bir tesis sayısının üzerinde çözümü tüm alternatiflerin denenmesi ile makul bir sürede bulmaya yetecek kadar hızlı değildirler. Mevcut özel amaçlı (askeri, ticari, vb.) bilgisayarlarla bile 30 tesisli bir kareli atama probleminin tüm alternatiflerini ($30!$) denemek onyıllarca sürebilir.

Bu sebeple sezgisel bir metot olan genetik algoritmalar tesis yerleşimi problemlerine çözüm bulmada, uzun yıllar etkilerini sürdüreceğe benzemektedir.

KAYNAKLAR

- Aleisa, E. E., Lin, L., (2005), "For Effective Facilities Planning: Layout Optimization Then Simulation, Or Vice Versa? 2005 Winter Simulation Conference, Orlando, FL, USA December 4-7
- Balakrishnana, J., Cheng, C., Wong, K., (2003), "FACOPT: A User Friendly Facility Layout Optimization System", *Computers & Operations Research*, 30:1625–1641.
- Balakrishnan, P. V., Jacob, S. V., (1996), "Genetic Algorithms for Product Design", *Management Science*, 42 (8):1105-1117.
- Beasley, J. E., Chu, P.C., (1995), "A Genetic Algorithm for The Set Covering Problem", *European Journal of Operational Research* 94: 392-404.
- Benjafaar, S., Heragu, S. S., Irani, S. A. (2000), "Next Generation Factory Layouts: Research Challenges And Recent Progress (Araştırma)", December, National Science Foundation, USA
- Biroğul, S.(2005), *Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme*, Yüksek Lisans Tezi Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara
- Bozer, Y.A., Meller, R.D., Erlebacher, S.J., (1994), "An improved-type layout algorithm for single and multiple-floor facilities". *Management Science* 40 (7): 918–932.
- Brown, D. E., Huntley, C. L., Spillane, A. R., (1989), "A Parallel Genetic Heuristic for The Quadratic Assignment Problem", in *Proc of the 3rd International Conference on Genetic Algorithms*, 406-415.
- Brucker, P. (1998), "An Efficient Algorithm for the Job-shop Problem With Two Jobs", *Computing*, vol. 40, 353-359,
- Cheng, R., Gen, M., Tsujimura, Y., (1996), "A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms-I, Representation", *Computers and Industrial Engineering*, 30 (4): 983-997.
- Chiang, Wen-Chyuan, Chiang, Chi, (1998), "Intelligent Local Search Strategies for Solving Facility Layout Problems with The Quadratic Assignmet Problem Formulation", *European Journal of Operation Research*, 106, 457-488.
- Çetin, N., (2002), "Genetik Algoritma", Yüksek Lisans Tezi, YTÜ FBE, İstanbul
- Dengiz, B. ve Altıparmak F., (1998), "Genetik Algoritmalar Genel Bakış", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Dergisi*, 9 (3), 3-10
- Drezner, Z., (2003), "A Hybrid Genetic Algorithm for The Quadratic Assignment Problem", *INFORMS*, 15.
- Emel, G. G., Taşkın, Ç., (2002), "Genetik Algoritmalar ve Uygulama Alanları", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (1):129-152.
- Erkut, H. ve Baskak, M., (1996) , *Stratejiden Uygulamaya Tesis Tasarımı*, İrfan Yayıncılık, İstanbul
- Fang, H., (1992), "Genetic Algorithms for Scheduling", *MSc Thesis, Department of Artificial Intelligence University of Edinburg*, 4: 7-11, 21-27.

- Gathercole, C., Ross, P (1997), "Tackling the Boolean Even N Parity Problem with Genetic Programming and Limited-Error Fitness", Genetic Programming 1997: Proceedings of the Second Annual Conference, Stanford University, CA, USA
- Glover, F. (1989), "Tabu Search - Part I", ORSA Journal on Computing, 1(3): 190-206.
- Glover, F. (1990), "Tabu Search - Part II", ORSA Journal on Computing, 2 (1): 4-32.
- Goetschalckx, M., (1992) "An Interactive Layout Heuristic Based on Hexagonal Adjacency Graphs", European Journal of Operational Research 63: 304–321.
- Goldberg, D. E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Hahn, P.M., Krarup, J. (2000), "A Hospital Facility Layout Problem Finally Solved", <http://www.seas.upenn.edu/~hahn/krarup30.pdf>
- Hao, J.K, Dorne. R ve Galinier, P. (1998), "Tabu Search For Frequency Assignment In Mobile Radio Networks", Journal Of Heuristics, 4, 1, 47-62
- Heragu, S., (1997), Facilities Design, PWS Publishing, Boston.
- Konak, A., Kulturel-Konak, S., Norman, B.A., Smith, A.E., (2006) "A new mixed Integer Programming Formulation for Facility Layout Design Using Flexible Bays", Operations Research Letters, 34: 660 – 672.
- Kulluk, S. (2003), "Tesis Yerleşimi Problemlerinde Genetik Algoritmalar ve Bir Genetik Algoritma Uygulaması", Yüksek Lisans Tezi, G.Ü. Fen Bilimleri Enstitüsü, Ankara.
- Lee, R.C., Moore, J.M., (1967), "CORELAP (Computerized Relationship Layout Planning)", Journal of Industrial Engineering, 8 (3)
- Liepins, G. E., Potter, W. D., Tonn, B. E., Hillard, M. R., Goeltz, R. T. Purucker, S. L., (1990), "Diagnosis, Parsimony and Genetic Algorithms", IEA/AIE, 1 : 1-8.
- Ligeett, Robin S., (2000), "Automated Facilities Layout: Past, Present and Future", Automation in Construction, 9: 197-215.
- Lin, L.C., Sharp, G.P., (1999a), "Quantitative and Qualitative Indices for the Plant Layout Evaluation Problem", European Journal of Operational Research, 116: 100–117.
- Lin, L.C., Sharp, G.P., (1999b) "Application of the Integrated Framework for the Plant Layout Evaluation Problem", European Journal of Operational Research 116: 118–138.
- Mirchandani P.B. and Francis R.L., (1990), Discrete Location Theory, John Wiley & Sons, New York
- Meller, R.D., Gau, K.-Y., (1996), "The Facility Layout Problem: Recent and Emerging Trends and Perspectives", Journal of Manufacturing Systems, 15 (5): 351–366.
- Quagliarella, D., Periauz, J., Poloni, C. and Winter, G. (Der.) (1997), Genetic Algorithms in Engineering and Computer Science, John Wiley and Sons, NY
- Reeves, C. R., (1993), Modern Heuristic Techniques for Combinatorial Problems, John Wiley & Sons, New York
- Roggen, D., (2005), "Multi-Cellular Reconfigurable Circuits: Evolution Morphogenesis and Learning", Ecole Polytechnique Fédérale De Lausanne, PhD thesis

- Rosenblatt, M. J., (1986), “The Dynamics of Plant Layout”, *Management Science*, 32 (1): 76-86.
- Sha, D. Y., Chen, C., (2001), “A New Approach to The Multiple Objective Facility Layout Problem”, *Integrated Manufacturing Systems*, 12(1): 59-66.
- Serpil, C., (1994), “Tesis Planlaması”, *Makine Mühendisliği El Kitabı Cilt-1: Üretim ve Tasarım*, A.M. Cerit (Der.), 1 :12-01, 12-23, MMO Yayın No: 169, Ankara
- Stevenson, W.J., (1999), *Production Operations Management*, McGraw Hill, NewYork
- Şerifoğlu, F.S., Ulusoy, G., Şahin, Ş. (2000), “Kaynak Kısıtlı Proje Çizelgelemede İndirgenmiş Nakit Akışı Maksimizasyonu İçin Bir Genetik Algoritma Yaklaşımı”, *Endüstri Mühendisliği Dergisi*, 2 : 2-12., Ankara
- Tanrıseven, D., (2000), *Genetik Algoritmalar Kullanarak Peptidlerde Motif Bulma*, , Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi
- Tanyaş, M., (1995), *Endüstri Mühendisliğine Giriş*, İstanbul, İrfan Yayıncılık
- Tate, D. M., Smith, A. E., A (1995), “Genetic Approach to The Quadratic Assignment Problem”, *Computers and Operations Research*, 22 (1): 73—83.
- Tavakkoli-Moghaddain, R.ve Shayan, E., (1998), “Facilities Layout Design by Genetic Algorithms”, *Computers and Industrial Engineering*, 35(3-4): 527-530.
- Toksarı, M.D., (2004), "Karıncalar Algoritması ve Tesis Yerleşimi Problemlerine Uygulanması", Yüksek Lisans Tezi, Y.T.Ü. Fen Bilimleri Enstitüsü, İstanbul
- Tompkins, J.A., White, J.A., Bozer, Y.A. ve T. J. M., (2003), *Facilities Planning*, 3rd. New York, JohnWiley & Sons Inc.
- Yang, T., Su, C.T., Hsu, Y.R., (2000), “Systematic Layout Planning: A study on Semiconductor Wafer Fabrication Facilities”, *International Journal of Operations and Production Management*, 20 (11): 1360–1372.
- Yang, T., Kuo, C. (2003), “A Hierarchical AHP/DEA Methodology for the Facilities Layout Design Problem”, *European Journal of Operational Research* , 147: 128-136.
- Yiğit, V., Türkbey, O., (2003), “Tesis Yerleşim Problemlerine Sezgisel Metotlarla Yaklaşım”, *Gazi Üniv. Müh. Mim. Fak. Der.*, 18(4): 45-56.

EKLER

- Ek 1 Literatür problemleri için bulunan en iyi değerler ve parametreler
- Ek 2 Her DM için kullanılan bölümler
- Ek 3 Uygulamada kullanılan bölümlerin mevcut ilk yerleşimleri
- Ek 4 Her DM için bölümler arası iş akışı matrisleri
- Ek 5 Her DM için yerleşimler arası uzaklık matrisleri

Ek 1 Literatür problemleri için bulunan en iyi değerler ve parametreler

NUG27 EN İYİ SONUÇ (0.800000 - 0.050000 - 0.100000 - 50 - 10000)

5272 :22 10 25 8 1 2 14 16 12 13 7 20 11 26 9 17 4 24 23 27 19 15 6 21 5 3 18 (**6225**)

NUG28, EN İYİ SONUÇ (0.810000 - 0.040000 - 0.040000 - 56 - 10000)

5180 :25 15 20 6 22 16 11 5 9 10 12 17 8 21 28 13 27 19 4 7 1 18 26 24 3 23 14 2 (**9535**)

NUG30, EN İYİ SONUÇ (0.800000 - 0.040000 - 0.400000 - 32 - 40000)

6133 :5 21 13 28 6 12 2 9 10 25 26 24 29 3 19 7 22 1 20 27 16 8 18 17 14 4 30 11 23 15
(**31584**)

SKO42, EN İYİ SONUÇ (0.850000 - 0.030000 - 0.030000 - 56 - 10000)

15936 :23 22 2 28 33 27 13 36 24 30 26 35 4 20 16 37 39 19 14 10 31 3 1 21 18 12 9 32 41
38 5 40 7 15 17 6 8 25 29 11 42 34 (**9857**)

SKO49, EN İYİ SONUÇ (0.850000 - 0.040000 - 0.040000 - 44 - 10000)

23604 :47 41 14 18 45 20 46 13 42 35 29 38 27 28 1 9 10 34 37 43 2 7 15 25 11 6 12 3 19 36
8 30 33 31 23 4 24 26 16 5 40 48 17 32 21 44 39 22 49 (**9745**)

SKO56, EN İYİ SONUÇ (0.830000 - 0.030000 - 0.050000 - 32 - 10000)

34785 :39 35 18 47 33 19 31 43 40 41 13 48 12 34 17 15 10 44 32 5 27 20 45 46 51 36 29 1
37 4 53 26 52 14 7 28 55 50 24 6 3 9 30 23 2 22 38 54 49 42 11 56 8 16 25 21 (**9302**)

WIL50, EN İYİ SONUÇ (0.830000 - 0.030000 - 0.050000 - 32 - 10000)

49197 :20 12 19 34 38 33 37 36 4 46 27 8 24 39 3 45 43 6 28 17 7 2 47 49 11 30 44 21 5 42
25 13 40 50 41 32 16 10 29 35 26 9 15 18 23 31 1 14 48 22 (**9930**)

SCR20, EN İYİ SONUÇ (0.800000 - 0.100000 - 0.100000 - 100 - 5000)

110676 :20 3 10 2 12 1 7 8 14 11 18 9 19 15 16 17 13 5 6 4 (**917**)

THO30, EN İYİ SONUÇ (0.800000 - 0.040000 - 0.060000 - 32 - 10000)

151312 :29 3 5 16 21 17 4 27 13 18 20 11 7 24 22 28 26 25 6 15 19 14 1 23 8 30 12 2 10 9
(**8861**)

THO40 EN İYİ SONUÇ (0.750000 - 0.040000 - 0.040000 - 80 - 10000)

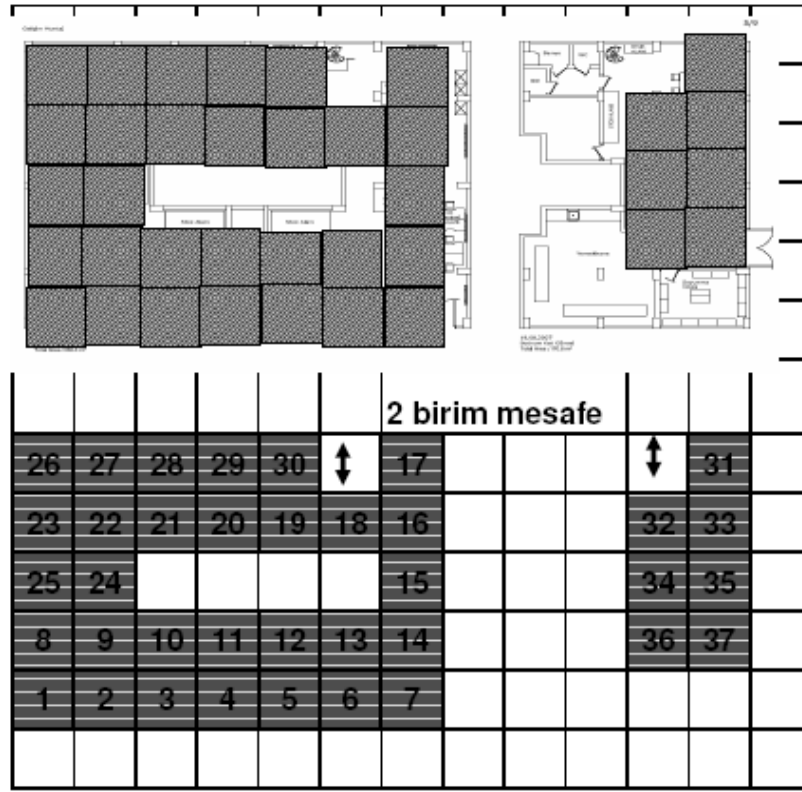
242130 :3 26 13 21 7 24 23 19 36 27 5 22 1 30 2 18 25 8 32 6 16 33 28 37 34 38 35 12 15 39
4 9 17 31 29 11 20 10 14 40 (**9371**)

Ek 2 Her DM için kullanılan bölümler

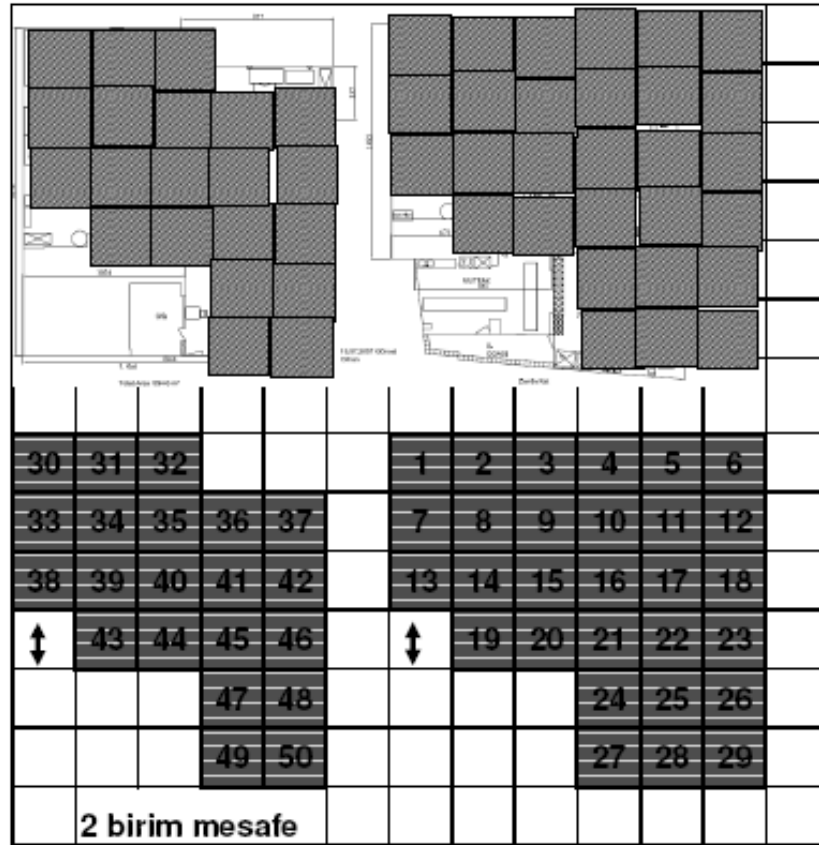
29			36			46			50			57		
H51	1	H51_1	L52	1	L52_1	M54	1	M54_1	O67	1	O67_1	R69	1	R69_1
H51	2	H51_2	L52	2	L52_2	M54	2	M54_2	O67	2	O67_2	R69	2	R69_2
H51	3	H51_3	L52	3	L52_3	M54	3	M54_3	O67	3	O67_3	R69	3	R69_3
H51	4	H51_4	L52	4	L52_4	M54	4	M54_4	O67	4	O67_4	R69	4	R69_4
H51	5	H51_5	L52	5	L52_5	M54	5	M54_5	O67	5	O67_5	R69	5	R69_5
H51	6	H51_6	L52	6	L52_6	M54	6	M54_6	O67	6	O67_6	R69	6	R69_6
H51	7	H51_7	L52	7	L52_7	M54	7	M54_7	O67	7	O67_7	R69	7	R69_7
H51	8	H51_8	L52	8	L52_8	M54	8	M54_8	O67	8	O67_8	R69	8	R69_8
H51	9	H51_9	L52	9	L52_9	M54	9	M54_9	O67	9	O67_9	R69	9	R69_9
H51	10	H51_10	L52	10	L52_10	M54	10	M54_10	O67	10	O67_10	R69	10	R69_10
H51	11	H51_11	L52	11	L52_11	M54	11	M54_11	O67	11	O67_11	R69	11	R69_11
H51	12	H51_12	L52	12	L52_12	M54	12	M54_12	O67	12	O67_12	R69	12	R69_12
H51	13	H51_13	L52	13	L52_13	M54	13	M54_13	O67	13	O67_13	R69	13	R69_13
H51	14	H51_14	L52	14	L52_14	M54	14	M54_14	O67	14	O67_14	R69	14	R69_14
H51	15	H51_15	L52	15	L52_15	M54	15	M54_15	O67	15	O67_15	R69	15	R69_15
H51	16	H51_16	L52	16	L52_16	M54	16	M54_16	O67	16	O67_16	R69	16	R69_16
H51	17	H51_17	L52	17	L52_17	M54	17	M54_17	O67	17	O67_17	R69	17	R69_17
H51	18	H51_18	L52	18	L52_18	M54	18	M54_18	O67	18	O67_18	R69	18	R69_18
H51	19	H51_19	L52	19	L52_19	M54	19	M54_19	O67	19	O67_19	R69	19	R69_19
H51	20	H51_20	L52	20	L52_20	M54	20	M54_20	O67	20	O67_20	R69	20	R69_20
H51	21	H51_21	L52	21	L52_21	M54	21	M54_21	O67	21	O67_21	R69	21	R69_21
H51	22	H51_22	L52	22	L52_22	M54	22	M54_22	O67	22	O67_22	R69	22	R69_22
H51	23	H51_23	L52	23	L52_23	M54	23	M54_23	O67	23	O67_23	R69	23	R69_23
H51	24	H51_24	L52	24	L52_24	M54	24	M54_24	O67	24	O67_24	R69	24	R69_24
H51	25	H51_25	L52	25	L52_25	M54	25	M54_25	O67	25	O67_25	R69	25	R69_25
H51	26	H51_26	L52	26	L52_26	M54	26	M54_26	O67	26	O67_26	R69	26	R69_26
H51	27	H51_27	L52	27	L52_27	M54	27	M54_27	O67	27	O67_27	R69	27	R69_27
H51	28	H51_28	L52	28	L52_28	M54	28	M54_28	O67	28	O67_28	R69	28	R69_28
H51	29	H51_29	L52	29	L52_29	M54	29	M54_29	O67	29	O67_29	R69	29	R69_29
			L52	30	L52_30	M54	30	M54_30	O67	30	O67_30	R69	30	R69_30
			L52	31	L52_31	M54	31	M54_31	O67	31	O67_31	R69	31	R69_31
			L52	32	L52_32	M54	32	M54_32	O67	32	O67_32	R69	32	R69_32
			L52	33	L52_33	M54	33	M54_33	O67	33	O67_33	R69	33	R69_33
			L52	34	L52_34	M54	34	M54_34	O67	34	O67_34	R69	34	R69_34
			L52	35	L52_35	M54	35	M54_35	O67	35	O67_35	R69	35	R69_35
			L52	36	L52_36	M54	36	M54_36	O67	36	O67_36	R69	36	R69_36
						M54	37	M54_37	O67	37	O67_37	R69	37	R69_37
						M54	38	M54_38	O67	38	O67_38			
						M54	39	M54_39	O67	39	O67_39			
						M54	40	M54_40	O67	40	O67_40			
						M54	41	M54_41	O67	41	O67_41			
						M54	42	M54_42	O67	42	O67_42			
						M54	43	M54_43	O67	43	O67_43			
						M54	44	M54_44	O67	44	O67_44			
						M54	45	M54_45	O67	45	O67_45			
						M54	46	M54_46	O67	46	O67_46			
									O67	47	O67_47			
									O67	48	O67_48			
									O67	49	O67_49			
									O67	50	O67_50			

Şekil Ek 2.1 Her DM için i'den n'e kadar bölümler

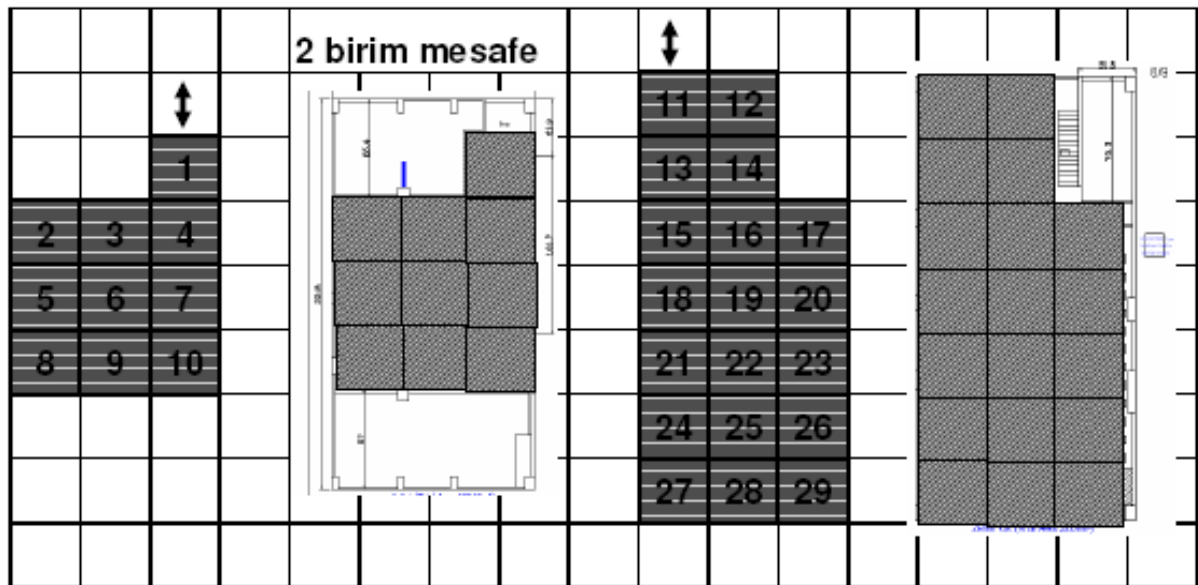
Ek 3 Uygulamada kullanılan bölümlerin mevcut ilk yerleşimleri



Şekil Ek 3.1 R69 (n=37 için) mevcut ilk yerleşim



Şekil Ek 3.2 O67 (n=50 için) mevcut ilk yerleşim



Şekil Ek 3.5 H51 (n=29 için) mevcut ilk yerleşim

Ek 4 Her DM için bölümler arası iş akışı matrisleri

	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	
1	0	30	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	0	0	0	0	0	0	
2	0	0	35	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	0	0	0	0	0	
3	0	0	0	40	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	0	0	0	0	
4	0	0	0	0	40	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	0	0	0	
5	0	0	0	0	0	40	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	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	35	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	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	35	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	25	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
10	0	0	0	0	0	0	0	0	0	0	10	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
11	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	30	5	5	0	0	0	0	10	5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	20	5	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	##	0	0	10	5	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
14	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	35	0	0	0	0	0	0	0	5	5	10	0	10	0	0	0	0	0	0	0	0	0	0	0	5	0	10	0	0	0	0	
19	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	10	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	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	30	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	5	5	0	0	0	0	5	10	15	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	20	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	
25	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	60	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	0	
27	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	30	0	0	0	0	0	0	0	0	0	0	0	0
28	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	25	0	0	0	0	0	0	0	0	0	0	0	0
29	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	15	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
31	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	0	0	0	20	0	0	0	0	
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	5	5	5	10	35	
33	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	0	0	0	0	25	0	0	0	
34	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	0	0	0	0	0	0	15	0	
35	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	0	0	0	0	0	0	30	0	
36	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	0	0	10	0	0	0	0		
37	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	0	0	35	0	0	0	0		

Şekil Ek 4.1 R69 (n=37 için) iş akış matrisi

Şekil Ek 4.2 O67 (n=50 için) iş akış matrisi

Şekil Ek 4.3 M54 (n=46 için) iş akış matrisi

	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	
1	0	5	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	0	0	0	0	0	
2	0	0	0	0	0	15	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	0	
3	5	10	0	0	0	0	15	10	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	
4	0	0	0	0	5	0	0	0	15	5	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	
5	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	0	0	0	0	0	0	0	
6	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	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	15	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	
8	0	0	0	0	0	25	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	0	
9	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	0	0	0	0	0	0	0	
10	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	0	0	0	0	0	0	0	
11	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	10	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	25	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	140	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	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	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	10	0	10	5	15	0	0	0	0	0	0	0	0	0	0
23	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	0	0	0	0	0	0	0	
24	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	0	0	0	0	0	0	0	
25	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	0	0	0	0	0	0	0	
26	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	0	0	0	0	0	0	0	
27	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	0	0	0	0	0	0	0	
28	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	0	0	0	0	0	0	0	
29	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	0	0	0	70	0	0	0	
30	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	0	0	0	0	0	0	0	
31	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	0	0	0	0	80	0	0	
32	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	0	0	0	0	0	0	0	
33	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	0	0	0	0	0	0	0	
34	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	0	0	0	0	60	0	80	
35	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	0	0	0	0	80	0	0	
36	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	0	0	0	0	0	0	0	

Şekil Ek 4.4 L52 (n=36 için) iş akış matrisi

	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
1	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
2	0	0	10	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	20	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
6	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	15	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
8	20	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
9	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	20	0	10	0	0	5	10	5	0	0	45	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	45
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	30	0	5	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	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
14	0	0	0	0	0	0	0	0	0	0	15	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	5	10	10	0	10	0	5	10	15	0	0	0	0	0	0	0
19	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
20	90	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
21	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
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	5	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	5	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	10	10	5	5	5	0	0

Şekil Ek 4.5 H51 (n=29 için) iş akış matrisi

Ek 5 Her DM için yerleşimler arası uzaklık matrisleri

	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
1	0	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	8	7	6	5	4	3	3	2	4	5	6	7	8	13	11	12	12	13	13	14
2	1	0	1	2	3	4	5	2	1	2	3	4	5	6	7	8	9	7	6	5	4	3	4	2	3	5	4	5	6	7	12	10	11	11	12	12	13
3	2	1	0	1	2	3	4	3	2	1	2	3	4	5	6	7	8	8	7	6	5	4	5	3	4	6	5	6	7	8	13	11	12	12	13	13	14
4	3	2	1	0	1	2	3	4	3	2	1	2	3	4	5	6	7	7	8	7	6	5	6	4	5	7	6	7	8	9	12	10	11	11	12	12	13
5	4	3	2	1	0	1	2	5	4	3	2	1	2	3	4	5	6	6	7	8	7	6	7	5	6	8	7	8	9	8	11	9	10	10	11	11	12
6	5	4	3	2	1	0	1	6	5	4	3	2	1	2	3	4	5	5	6	7	8	7	8	6	7	9	8	9	8	7	10	8	9	9	10	10	11
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	2	3	4	4	5	6	7	8	9	7	8	10	9	8	7	6	9	7	8	8	9	9	10
8	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	8	9	7	6	5	4	3	2	2	1	3	4	5	6	7	12	10	11	11	12	12	13
9	2	1	2	3	4	5	6	1	0	1	2	3	4	5	6	7	8	6	5	4	3	2	3	1	2	4	3	4	5	6	11	9	10	10	11	11	12
10	3	2	1	2	3	4	5	2	1	0	1	2	3	4	5	6	7	7	6	5	4	3	4	2	3	5	4	5	6	7	12	10	11	11	12	12	13
11	4	3	2	1	2	3	4	3	2	1	0	1	2	3	4	5	6	6	7	6	5	4	5	3	4	6	5	6	7	8	11	9	10	10	11	11	12
12	5	4	3	2	1	2	3	4	3	2	1	0	1	2	3	4	5	5	6	7	6	5	6	4	5	7	6	7	8	7	10	8	9	9	10	10	11
13	6	5	4	3	2	1	2	5	4	3	2	1	0	1	2	3	4	4	5	6	7	6	7	5	6	8	7	8	7	6	9	7	8	8	9	9	10
14	7	6	5	4	3	2	1	6	5	4	3	2	1	0	1	2	3	3	4	5	6	7	8	6	7	9	8	7	6	5	8	6	7	7	8	8	9
15	8	7	6	5	4	3	2	7	6	5	4	3	2	1	0	1	2	2	3	4	5	6	7	7	8	8	7	6	5	4	7	5	6	6	7	7	8
16	9	8	7	6	5	4	3	8	7	6	5	4	3	2	1	0	1	1	2	3	4	5	6	6	7	7	6	5	4	3	6	4	5	5	6	6	7
17	10	9	8	7	6	5	4	9	8	7	6	5	4	3	2	1	0	2	3	4	5	6	7	7	8	8	7	6	5	4	7	5	6	6	7	7	8
18	8	7	8	7	6	5	4	7	6	7	6	5	4	3	2	1	2	0	1	2	3	4	5	5	6	6	5	4	3	2	5	3	4	4	5	5	6
19	7	6	7	8	7	6	5	6	5	6	7	6	5	4	3	2	3	1	0	1	2	3	4	4	5	5	4	3	2	1	6	4	5	5	6	6	7
20	6	5	6	7	8	7	6	5	4	5	6	7	6	5	4	3	4	2	1	0	1	2	3	3	4	4	3	2	1	2	7	5	6	6	7	7	8
21	5	4	5	6	7	8	7	4	3	4	5	6	7	6	5	4	5	3	2	1	0	1	2	2	3	3	2	1	2	3	8	6	7	7	8	8	9
22	4	3	4	5	6	7	8	3	2	3	4	5	6	7	6	5	6	4	3	2	1	0	1	1	2	2	1	2	3	4	9	7	8	8	9	9	10
23	3	4	5	6	7	8	9	2	3	4	5	6	7	8	7	6	7	5	4	3	2	1	0	2	1	1	2	3	4	5	10	8	9	9	10	10	11
24	3	2	3	4	5	6	7	2	1	2	3	4	5	6	7	6	7	5	4	3	2	1	2	0	1	3	2	3	4	5	10	8	9	9	10	10	11
25	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	7	8	6	5	4	3	2	1	1	0	2	3	4	5	6	11	9	10	10	11	11	12
26	4	5	6	7	8	9	10	3	4	5	6	7	8	9	8	7	8	6	5	4	3	2	1	3	2	0	1	2	3	4	11	9	10	10	11	11	12
27	5	4	5	6	7	8	9	4	3	4	5	6	7	8	7	6	7	5	4	3	2	1	2	2	3	1	0	1	2	3	10	8	9	9	10	10	11
28	6	5	6	7	8	9	8	5	4	5	6	7	8	7	6	5	6	4	3	2	1	2	3	3	4	2	1	0	1	2	9	7	8	8	9	9	10
29	7	6	7	8	9	8	7	6	5	6	7	8	7	6	5	4	5	3	2	1	2	3	4	4	5	3	2	1	0	1	8	6	7	7	8	8	9
30	8	7	8	9	8	7	6	7	6	7	8	7	6	5	4	3	4	2	1	2	3	4	5	5	6	4	3	2	1	0	7	5	6	6	7	7	8
31	13	12	13	12	11	10	9	12	11	12	11	10	9	8	7	6	7	5	6	7	8	9	10	10	11	11	10	9	8	7	0	2	1	3	2	4	3
32	11	10	11	10	9	8	7	10	9	10	9	8	7	6	5	4	5	3	4	5	6	7	8	8	9	9	8	7	6	5	2	0	1	1	2	2	3
33	12	11	12	11	10	9	8	11	10	11	10	9	8	7	6	5	6	4	5	6	7	8	9	9	10	10	9	8	7	6	1	1	0	2	1	3	2
34	12	11	12	11	10	9	8	11	10	11	10	9	8	7	6	5	6	4	5	6	7	8	9	9	10	10	9	8	7	6	3	1	2	0	1	1	2
35	13	12	13	12	11	10	9	12	11	12	11	10	9	8	7	6	7	5	6	7	8	9	10	10	11	11	10	9	8	7	2	2	1	1	0	2	1
36	13	12	13	12	11	10	9	12	11	12	11	10	9	8	7	6	7	5	6	7	8	9	10	10	11	11	10	9	8	7	4	2	3	1	2	0	1
37	14	13	14	13	12	11	10	13	12	13	12	11	10	9	8	7	8	6	7	8	9	10	11	11	12	12	11	10	9	8	3	3	2	2	1	1	0

Şekil Ek 5.1 R69 (n=37 için) yerleşimler arası uzaklık matrisi

	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	41	42	43	44	45	46	47	48	49	50	
1	0	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	41	42	43	44	45	46	47	48	49	50
2	0	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	41	42	43	44	45	46	47	48	49	50
3	2	1	0	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5	4	3	4	5	6	5	6	7	6	7	8	9	10	11	8	9	10	11	12	7	8	9	10	11	9	10	11	12	13	14	15	16	
4	3	2	1	0	1	2	4	3	2	1	2	3	5	4	3	2	3	4	5	4	3	4	5	6	5	6	7	10	11	12	9	10	11	12	13	8	9	10	11	12	10	11	12	13	14	15	16	17	18	19	20
5	4	3	2	1	0	1	5	4	3	2	1	2	6	5	4	3	2	3	6	5	4	3	4	5	4	5	6	5	6	11	12	13	10	11	12	13	14	9	10	11	12	13	11	12	13	14	15	16	17	18	
6	5	4	3	2	1	0	6	5	4	3	2	1	7	6	5	4	3	2	7	6	5	4	3	6	5	4	7	6	5	12	13	14	11	12	13	14	15	10	11	12	13	14	12	13	14	15	16	17	18	19	
7	1	2	3	4	5	6	0	1	2	3	4	5	1	2	3	4	5	6	3	4	5	6	7	6	7	8	7	8	9	6	7	8	5	6	7	8	9	4	5	6	7	8	6	7	8	9	10	10	11	12	
8	2	1	2	3	4	5	1	0	1	2	3	4	2	1	2	3	4	5	2	3	4	5	6	5	6	7	6	7	8	7	8	9	6	7	8	9	10	5	6	7	8	9	7	8	9	10	11	11	12	13	
9	3	2	1	2	3	4	2	1	0	1	2	3	3	2	1	2	3	4	3	2	3	4	5	4	5	6	5	6	7	8	9	10	7	8	9	10	11	6	7	8	9	10	8	9	10	11	11	12	12	13	
10	4	3	2	1	2	3	3	2	1	0	1	2	4	3	2	1	2	3	4	3	2	3	4	5	4	5	6	5	6	9	10	11	8	9	10	11	12	7	8	9	10	11	9	10	11	12	13	13	14	15	
11	5	4	3	2	1	2	4	3	2	1	0	1	5	4	3	2	1	2	5	4	3	2	3	4	3	4	5	4	5	10	11	12	9	10	11	12	13	8	9	10	11	12	10	11	12	13	13	14	15	16	17
12	6	5	4	3	2	1	5	4	3	2	1	0	6	5	4	3	2	1	6	5	4	3	2	5	4	3	6	5	4	11	12	13	10	11	12	13	14	9	10	11	12	13	11	12	13	14	15	16	17	18	19
13	2	3	4	5	6	7	1	2	3	4	5	6	0	1	2	3	4	5	2	3	4	5	6	5	6	7	6	7	8	5	6	7	4	5	6	7	8	3	4	5	6	7	5	6	7	8	8	9	9	10	11
14	3	2	3	4	5	6	2	1	2	3	4	5	1	0	1	2	3	4	1	2	3	4	5	6	5	6	7	6	7	8	5	6	7	8	9	4	5	6	7	8	6	7	8	9	10	10	11	12	13	14	15
15	4	3	2	3	4	5	3	2	1	2	3	4	2	1	0	1	2	3	2	1	2	3	4	3	4	5	4	5	6	7	8	9	6	7	8	9	10	5	6	7	8	9	7	8	9	10	11	11	12	13	14
16	5	4	3	2	3	4	4	3	2	1	2	3	3	2	1	0	1	2	3	2	1	2	3	2	3	4	3	4	5	8	9	10	7	8	9	10	11	6	7	8	9	10	8	9	10	11	11	12	12	13	14
17	6	5	4	3	2	3	5	4	3	2	1	2	4	3	2	1	0	1	4	3	2	1	2	3	2	3	4	3	4	9	10	11	8	9	10	11	12	7	8	9	10	11	9	10	11	12	13	13	14	15	16
18	7	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0	5	4	3	2	1	4	3	2	5	4	3	10	11	12	9	10	11	12	13	8	9	10	11	12	10	11	12	13	13	14	15	16	17
19	4	3	4	5	6	7	3	2	3	4	5	6	2	1	2	3	4	5	0	1	2	3	4	3	4	5	4	5	6	7	8	9	6	7	8	9	10	5	6	7	8	9	7	8	9	10	11	11	12	13	14
20	5	4	3	4	5	6	4	3	2	3	4	5	3	2	1	2	3	4	1	0	1	2	3	2	3	4	3	4	5	8	9	10	7	8	9	10	11	6	7	8	9	10	8	9	10	11	11	12	12	13	14
21	6	5	4	3	4	5	5	4	3	2	3	4	4	3	2	1	2	3	2	1	0	1	2	1	2	3	2	3	4	9	10	11	8	9	10	11	12	7	8	9	10	11	9	10	11	12	13	13	14	15	
22	7	6	5	4	3	4	6	5	4	3	2	3	5	4	3	2	1	2	3	2	1	0	1	2	1	2	3	3	10	11	12	9	10	11	12	13	8	9	10	11	12	10	11	12	13	13	14	15	16	17	
23	8	7	6	5	4	3	7	6	5	4	3	2	6	5	4	3	2	1	4	3	2	1	0	3	2	1	4	3	2	11	12	13	10	11	12	13	14	9	10	11	12	13	11	12	13	14	15	16	17	18	19
24	7	6	5	4	5	6	6	5	4	3	4	5	5	4	3	2	3	4	3	2	1	2	3	0	1	2	1	2	3	10	11	12	9	10	11	12	13	8	9	10	11	12	10	11	12	13	14	15	16	17	
25	8	7	6	5	4	5	7	6	5	4	3	4	6	5	4	3	2	3	4	3	2	1	2	1	0	1	2	1	2	11	12	13	10	11	12	13	14	9	10	11	12	13	11	12	13	14	15	16	17	18	19
26	9	8	7	6	5	4	8	7	6	5	4	3	7	6	5	4	3	2	5	4	3	2	1	2	1	0	3	2	1	12	13	14	11	12	13	14	15	10	11	12	13	14	12	13	14	15	16	17	18	19	20
27	8	7	6	5	6	7	7	6	5	4	5	6	6	5	4	3	4	5	4	3	2	3	4	1	2	3	0	1	2	11	12	13	10	11	12	13	14	9	10	11	12	13	11	12	13	14	15	16	17	18	19
28	9	8	7	6	5	6	8	7	6	5	4	5	7	6	5	4	3	4	5	4	3	2	3	2	1	2	1	0	1	12	13	14	11	12	13	14	15	10	11	12	13	14	12	13	14	15	16	17	18	19	20
29	10	9	8	7	6	5	9	8	7	6	5	4	8	7	6	5	4	3	6	5	4	3	2	3	2	1	2	1	0	13	14	15	12	13	14	15	16	11	12	13	14	15	13	14	15	16	17	18	19	20	
30	7	8	9	10	11	12	6	7	8	9	10	11	5	6	7	8	9	10	7	8	9	10	11	10	11	12	11	12	13	0	1	2	1	2	3	4	5	2	3	4	5	6	4	5	6	7	7	8	8	9	
31	8	9	10	11	12	13	7	8	9	10	11	12	6	7	8	9	10	11	8	9	10	11	12	11	12	13	12	13	14	1	0	1	2	1	2	3	4	3	2	3	4	5	3	4	5	6	6	7	7	8	
32	9	10	11	12	13	14	8	9	10	11	12	13	7	8	9	10	11	12	9	10	11	12	13	12	13	14	13	14	15	2	1	0	3	2	1	2	3	4	3	2	3	4	5	4	5	6	6	7	8		
33	6	7	8	9	10	11	5	6	7	8	9	10	4	5	6	7	8	9	6	7	8	9	10	9	10	11	10	11	12	1	2	3	0	1	2	3	4	1	2	3	4	5	3	4	5	6	6	7	7	8	
34	7	8	9	10	11	12	6	7	8	9	10	11	5	6	7	8	9	10	7	8	9	10	11	10	11	12	11	12	13	2	1	2	1	0	1	2	3	2	1	2	3	4	2	3	4	5	5	6	6	7	
35	8	9	10	11	12	13	7	8	9	10	11	12	6	7	8	9	10	11	8	9	10	11	12	11	12	13	12	13	14	3	2	1	2	1	0	1	2	3	2	1	2	3	3	2	3	4	4	5	5	6	
36	9	10	11	12	13	14	8	9	10	11	12	13	7	8	9	10	11	12	9	10	11	12	13	12	13	14	13	14	15	4	3	2	3	2	1	0	1	4	3	2	1	2	4	3	2	3	3	4	4	5	
37	10	11	12	13																																															

	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	41	42	43	44	45	46	
1	0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	11	8	9	10	11	12	10	11	12	13	2	3	4	5	6	3	4	5	6	7	4	5	6	7	8	
2	1	0	1	2	3	4	5	6	7	8	9	2	1	2	3	4	5	6	7	8	9	10	7	8	9	10	11	9	10	11	12	3	2	3	4	5	4	3	4	5	6	5	4	5	6	7	
3	2	1	0	1	2	3	4	5	6	7	8	3	2	1	2	3	4	5	6	7	8	9	6	7	8	9	10	8	9	10	11	4	3	2	3	4	5	4	3	4	5	6	5	4	5	6	
4	3	2	1	0	1	2	3	4	5	6	7	4	3	2	1	2	3	4	5	6	7	8	5	6	7	8	9	7	8	9	10	5	4	3	2	3	6	5	4	3	4	7	6	5	4	5	
5	4	3	2	1	0	1	2	3	4	5	6	5	4	3	2	1	2	3	4	5	6	7	4	5	6	7	8	6	7	8	9	6	5	4	3	2	7	6	5	4	3	8	7	6	5	4	
6	5	4	3	2	1	0	1	2	3	4	5	6	5	4	3	2	1	2	3	4	5	6	3	4	5	6	7	5	6	7	8	7	6	5	4	3	8	7	6	5	4	9	8	7	6	5	
7	6	5	4	3	2	1	0	1	2	3	4	7	6	5	4	3	2	1	2	3	4	5	2	3	4	5	6	4	5	6	7	8	7	6	5	4	9	8	7	6	5	10	9	8	7	6	
8	7	6	5	4	3	2	1	0	1	2	3	8	7	6	5	4	3	2	1	2	3	4	3	2	3	4	5	3	4	5	6	9	8	7	6	5	10	9	8	7	6	11	10	9	8	7	
9	8	7	6	5	4	3	2	1	0	1	2	9	8	7	6	5	4	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	10	9	8	7	6	11	10	9	8	7	12	11	10	9	8
10	9	8	7	6	5	4	3	2	1	0	1	10	9	8	7	6	5	4	3	2	1	2	5	4	3	2	3	5	4	3	4	11	10	9	8	7	12	11	10	9	8	13	12	11	10	9	
11	10	9	8	7	6	5	4	3	2	1	0	11	10	9	8	7	6	5	4	3	2	1	6	5	4	3	2	6	5	4	3	12	11	10	9	8	13	12	11	10	9	14	13	12	11	10	
12	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	8	9	10	7	8	9	10	11	9	10	11	12	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7	
13	2	1	2	3	4	5	6	7	8	9	10	1	0	1	2	3	4	5	6	7	8	9	6	7	8	9	10	8	9	10	11	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6	
14	3	2	1	2	3	4	5	6	7	8	9	2	1	0	1	2	3	4	5	6	7	8	5	6	7	8	9	7	8	9	10	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	
15	4	3	2	1	2	3	4	5	6	7	8	3	2	1	0	1	2	3	4	5	6	7	4	5	6	7	8	6	7	8	9	4	3	2	1	2	5	4	3	2	3	6	5	4	3	4	
16	5	4	3	2	1	2	3	4	5	6	7	4	3	2	1	0	1	2	3	4	5	6	3	4	5	6	7	5	6	7	8	5	4	3	2	1	6	5	4	3	2	7	6	5	4	3	
17	6	5	4	3	2	1	2	3	4	5	6	5	4	3	2	1	0	1	2	3	4	5	2	3	4	5	6	4	5	6	7	6	5	4	3	2	7	6	5	4	3	8	7	6	5	4	
18	7	6	5	4	3	2	1	2	3	4	5	6	5	4	3	2	1	0	1	2	3	4	1	2	3	4	5	3	4	5	6	7	6	5	4	3	8	7	6	5	4	9	8	7	6	5	
19	8	7	6	5	4	3	2	1	2	3	4	7	6	5	4	3	2	1	0	1	2	3	2	1	2	3	4	2	3	4	5	8	7	6	5	4	9	8	7	6	5	10	9	8	7	6	
20	9	8	7	6	5	4	3	2	1	2	3	8	7	6	5	4	3	2	1	0	1	2	3	2	1	2	3	3	2	3	4	9	8	7	6	5	10	9	8	7	6	11	10	9	8	7	
21	10	9	8	7	6	5	4	3	2	1	2	9	8	7	6	5	4	3	2	1	0	1	4	3	2	1	2	4	3	2	3	10	9	8	7	6	11	10	9	8	7	12	11	10	9	8	
22	11	10	9	8	7	6	5	4	3	2	1	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	5	4	3	2	11	10	9	8	7	12	11	10	9	8	13	12	11	10	9	
23	8	7	6	5	4	3	2	3	4	5	6	7	6	5	4	3	2	1	2	3	4	5	0	1	2	3	4	2	3	4	5	8	7	6	5	4	9	8	7	6	5	10	9	8	7	6	
24	9	8	7	6	5	4	3	2	3	4	5	8	7	6	5	4	3	2	1	2	3	4	1	0	1	2	3	1	2	3	4	9	8	7	6	5	10	9	8	7	6	11	10	9	8	7	
25	10	9	8	7	6	5	4	3	2	3	4	9	8	7	6	5	4	3	2	1	2	3	2	1	0	1	2	2	1	2	3	10	9	8	7	6	11	10	9	8	7	12	11	10	9	8	
26	11	10	9	8	7	6	5	4	3	2	3	10	9	8	7	6	5	4	3	2	1	2	3	2	1	0	1	3	2	1	2	11	10	9	8	7	12	11	10	9	8	13	12	11	10	9	
27	12	11	10	9	8	7	6	5	4	3	2	11	10	9	8	7	6	5	4	3	2	1	4	3	2	1	0	4	3	2	1	12	11	10	9	8	13	12	11	10	9	14	13	12	11	10	
28	10	9	8	7	6	5	4	3	4	5	6	9	8	7	6	5	4	3	2	3	4	5	2	1	2	3	4	0	1	2	3	10	9	8	7	6	11	10	9	8	7	12	11	10	9	8	
29	11	10	9	8	7	6	5	4	3	4	5	10	9	8	7	6	5	4	3	2	3	4	3	2	1	2	3	1	0	1	2	11	10	9	8	7	12	11	10	9	8	13	12	11	10	9	
30	12	11	10	9	8	7	6	5	4	3	4	11	10	9	8	7	6	5	4	3	2	3	4	3	2	1	2	2	1	0	1	12	11	10	9	8	13	12	11	10	9	14	13	12	11	10	
31	13	12	11	10	9	8	7	6	5	4	3	12	11	10	9	8	7	6	5	4	3	2	5	4	3	2	1	3	2	1	0	13	12	11	10	9	14	13	12	11	10	15	14	13	12	11	
32	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	8	9	10	11	12	10	11	12	13	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	
33	3	2	3	4	5	6	7	8	9	10	11	2	1	2	3	4	5	6	7	8	9	10	7	8	9	10	11	9	10	11	12	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	
34	4	3	2	3	4	5	6	7	8	9	10	3	2	1	2	3	4	5	6	7	8	9	6	7	8	9	10	8	9	10	11	2	1	0	1	2	3	2	1	2	3	4	3	2	3	4	
35	5	4	3	2	3	4	5	6	7	8	9	4	3	2	1	2	3	4	5	6	7	8	5	6	7	8	9	7	8	9	10	3	2	1	0	1	4	3	2	1	2	5	4	3	2	3	
36	6	5	4	3	2	3	4	5	6	7	8	5	4	3	2	1	2	3	4	5	6	7	4	5	6	7	8	6	7	8	9	4	3	2	1	0	5	4	3	2	1	6	5	4	3	2	
37	3	4	5	6	7	8	9	10	11	12	13	2	3	4	5	6	7	8	9	10	11	12	9	10	11	12	13	11	12	13	14	1	2	3	4	5	0	1	2	3	4	1	2	3	4	5	
38	4	3	4	5	6	7	8	9	10	11	12	3	2	3	4	5	6	7	8	9	10	11	8	9	10	11	12	10	11	12	13	2	1	2	3	4	1	0	1	2	3	2	1	2	3	4	
39	5	4	3	4	5	6	7	8	9	10	11	4	3	2	3	4	5	6	7	8	9	10	7	8	9	10	11	9	10	11	12	3	2	1	2	3	2	1	0	1	2	3	2	1	2	3	
40	6	5	4	3	4	5	6	7	8	9	10	5	4	3	2	3	4	5	6	7	8	9	6	7	8	9	10	8	9	10	11	4	3	2	1	2	3	2	1	0	1	4	3	2	1	2	
41	7	6	5	4	3	4	5	6	7	8	9	6	5	4	3	2	3	4	5	6	7	8	5	6	7	8	9	7	8	9	10	5	4	3	2	1	4	3	2	1	0	5	4	3	2	1	
42	4	5	6	7	8	9																																									

Şekil Ek 5.3 M54 (n=46 için) yerleşimler arası uzaklık matrisi

	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
1	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	7	8	9	3	4	5	6	7	8	9	10	13	14	12	13	11	12	13	10	11	12
2	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	6	7	8	4	3	4	5	6	7	8	9	14	15	13	14	12	13	14	11	12	13
3	2	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	6	7	5	4	3	4	5	6	7	8	15	16	14	15	13	14	15	12	13	14
4	3	2	1	0	1	4	3	2	1	2	5	4	3	2	3	4	5	6	6	5	4	3	4	5	6	7	16	17	15	16	14	15	16	13	14	15
5	4	3	2	1	0	5	4	3	2	1	6	5	4	3	2	3	4	5	7	6	5	4	3	4	5	6	17	18	16	17	15	16	17	14	15	16
6	1	2	3	4	5	0	1	2	3	4	1	2	3	4	5	6	7	8	2	3	4	5	6	7	8	9	12	13	11	12	10	11	12	9	10	11
7	2	1	2	3	4	1	0	1	2	3	2	1	2	3	4	5	6	7	3	2	3	4	5	6	7	8	13	14	12	13	11	12	13	10	11	12
8	3	2	1	2	3	2	1	0	1	2	3	2	1	2	3	4	5	6	4	3	2	3	4	5	6	7	14	15	13	14	12	13	14	11	12	13
9	4	3	2	1	2	3	2	1	0	1	4	3	2	1	2	3	4	5	5	4	3	2	3	4	5	6	15	16	14	15	13	14	15	12	13	14
10	5	4	3	2	1	4	3	2	1	0	5	4	3	2	1	2	3	4	6	5	4	3	2	3	4	5	16	17	15	16	14	15	16	13	14	15
11	2	3	4	5	6	1	2	3	4	5	0	1	2	3	4	5	6	7	1	2	3	4	5	6	7	8	11	12	10	11	9	10	11	8	9	10
12	3	2	3	4	5	2	1	2	3	4	1	0	1	2	3	4	5	6	2	1	2	3	4	5	6	7	12	13	11	12	10	11	12	9	10	11
13	4	3	2	3	4	3	2	1	2	3	2	1	0	1	2	3	4	5	3	2	1	2	3	4	5	6	13	14	12	13	11	12	13	10	11	12
14	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	2	3	4	4	3	2	1	2	3	4	5	14	15	13	14	12	13	14	11	12	13
15	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	1	2	3	5	4	3	2	1	2	3	4	15	16	14	15	13	14	15	12	13	14
16	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	0	1	2	6	5	4	3	2	1	2	3	16	17	15	16	14	15	16	13	14	15
17	8	7	6	5	4	7	6	5	4	3	6	5	4	3	2	1	0	1	7	6	5	4	3	2	1	2	17	18	16	17	15	16	17	14	15	16
18	9	8	7	6	5	8	7	6	5	4	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	18	19	17	18	16	17	18	15	16	17
19	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	10	11	9	10	8	9	10	7	8	9
20	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	5	6	7	1	0	1	2	3	4	5	6	11	12	10	11	9	10	11	8	9	10
21	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	4	5	6	2	1	0	1	2	3	4	5	12	13	11	12	10	11	12	9	10	11
22	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	4	5	3	2	1	0	1	2	3	4	13	14	12	13	11	12	13	10	11	12
23	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	2	3	4	4	3	2	1	0	1	2	3	14	15	13	14	12	13	14	11	12	13
24	8	7	6	5	4	7	6	5	4	3	6	5	4	3	2	1	2	3	5	4	3	2	1	0	1	2	15	16	14	15	13	14	15	12	13	14
25	9	8	7	6	5	8	7	6	5	4	7	6	5	4	3	2	1	2	6	5	4	3	2	1	0	1	16	17	15	16	14	15	16	13	14	15
26	10	9	8	7	6	9	8	7	6	5	8	7	6	5	4	3	2	1	7	6	5	4	3	2	1	0	17	18	16	17	15	16	17	14	15	16
27	13	14	15	16	17	12	13	14	15	16	11	12	13	14	15	16	17	18	10	11	12	13	14	15	16	17	0	1	1	2	2	3	4	3	4	5
28	14	15	16	17	18	13	14	15	16	17	12	13	14	15	16	17	18	19	11	12	13	14	15	16	17	18	1	0	2	1	3	2	3	4	3	4
29	12	13	14	15	16	11	12	13	14	15	10	11	12	13	14	15	16	17	9	10	11	12	13	14	15	16	1	2	0	1	1	2	3	2	3	4
30	13	14	15	16	17	12	13	14	15	16	11	12	13	14	15	16	17	18	10	11	12	13	14	15	16	17	2	1	1	0	2	1	2	3	2	3
31	11	12	13	14	15	10	11	12	13	14	9	10	11	12	13	14	15	16	8	9	10	11	12	13	14	15	2	3	1	2	0	1	2	1	2	3
32	12	13	14	15	16	11	12	13	14	15	10	11	12	13	14	15	16	17	9	10	11	12	13	14	15	16	3	2	2	1	1	0	1	2	1	2
33	13	14	15	16	17	12	13	14	15	16	11	12	13	14	15	16	17	18	10	11	12	13	14	15	16	17	4	3	3	2	2	1	0	3	2	1
34	10	11	12	13	14	9	10	11	12	13	8	9	10	11	12	13	14	15	7	8	9	10	11	12	13	14	3	4	2	3	1	2	3	0	1	2
35	11	12	13	14	15	10	11	12	13	14	9	10	11	12	13	14	15	16	8	9	10	11	12	13	14	15	4	3	3	2	2	1	2	1	0	1
36	12	13	14	15	16	11	12	13	14	15	10	11	12	13	14	15	16	17	9	10	11	12	13	14	15	16	5	4	4	3	3	2	1	2	1	0

Şekil Ek 5.4 L52 (n=36 için) yerleşimler arası uzaklık matrisi

	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
1	0	3	2	1	4	3	3	6	5	4	4	3	4	4	6	5	6	7	6	7	8	7	8	9	8	9	10	9	10
2	3	0	1	2	1	2	3	2	3	4	7	6	7	7	9	8	9	10	9	10	11	10	11	12	11	12	13	12	13
3	2	1	0	1	2	1	2	3	2	3	6	5	6	6	8	7	8	9	8	9	10	9	10	11	10	11	12	11	12
4	1	2	1	0	3	2	1	4	3	2	5	4	5	5	7	6	7	8	7	8	9	8	9	10	9	10	11	10	11
5	4	1	2	3	0	1	2	1	2	3	8	7	8	8	10	9	10	11	10	11	12	11	12	13	12	13	14	13	14
6	3	2	1	2	1	0	1	2	1	2	7	6	7	7	9	8	9	10	9	10	11	10	11	12	11	12	13	12	13
7	3	3	2	1	2	1	0	3	2	1	6	5	6	6	8	7	8	9	8	9	10	9	10	11	10	11	12	11	12
8	6	2	3	4	1	2	3	0	1	2	9	8	10	9	11	10	11	12	11	12	13	12	13	14	13	14	15	14	15
9	5	3	2	3	2	1	2	1	0	1	8	7	9	8	10	9	10	11	10	11	12	11	12	13	12	13	14	13	14
10	4	4	3	2	3	2	1	2	1	0	7	6	8	7	9	8	9	10	9	10	11	10	11	12	11	12	13	12	13
11	4	7	6	5	8	7	6	9	8	7	0	1	1	2	2	3	4	3	4	5	4	5	6	5	6	7	6	7	8
12	3	6	5	4	7	6	5	8	7	6	1	0	2	1	3	2	3	4	3	4	5	4	5	6	5	6	7	6	7
13	4	7	6	5	8	7	6	10	9	8	1	2	0	1	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
14	4	7	6	5	8	7	6	9	8	7	2	1	1	0	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6
15	6	9	8	7	10	9	8	11	10	9	2	3	1	2	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6
16	5	8	7	6	9	8	7	10	9	8	3	2	2	1	1	0	1	2	1	2	3	2	3	4	3	4	5	4	5
17	6	9	8	7	10	9	8	11	10	9	4	3	3	2	2	1	0	3	2	1	4	3	2	5	4	3	6	5	4
18	7	10	9	8	11	10	9	12	11	10	3	4	2	3	1	2	3	0	1	2	1	2	3	2	3	4	3	4	5
19	6	9	8	7	10	9	8	11	10	9	4	3	3	2	2	1	2	1	0	1	2	1	2	3	2	3	4	3	4
20	7	10	9	8	11	10	9	12	11	10	5	4	4	3	3	2	1	2	1	0	3	2	1	4	3	2	5	4	3
21	8	11	10	9	12	11	10	13	12	11	4	5	3	4	2	3	4	1	2	3	0	1	2	1	2	3	2	3	4
22	7	10	9	8	11	10	9	12	11	10	5	4	4	3	3	2	3	2	1	2	1	0	1	2	1	2	3	2	3
23	8	11	10	9	12	11	10	13	12	11	6	5	5	4	4	3	2	3	2	1	2	1	0	3	2	1	4	3	2
24	9	12	11	10	13	12	11	14	13	12	5	6	4	5	3	4	5	2	3	4	1	2	3	0	1	2	1	2	3
25	8	11	10	9	12	11	10	13	12	11	6	5	5	4	4	3	4	3	2	3	2	1	2	1	0	1	2	1	2
26	9	12	11	10	13	12	11	14	13	12	7	6	6	5	5	4	3	4	3	2	3	2	1	2	1	0	3	2	1
27	10	13	12	11	14	13	12	15	14	13	6	7	5	6	4	5	6	3	4	5	2	3	4	1	2	3	0	1	2
28	9	12	11	10	13	12	11	14	13	12	7	6	6	5	5	4	5	4	3	4	3	2	3	2	1	2	1	0	1
29	10	13	12	11	14	13	12	15	14	13	8	7	7	6	6	5	4	5	4	3	4	3	2	3	2	1	2	1	0

Şekil Ek 5.5 H51 (n=29 için) yerleşimler arası uzaklık matrisi

ÖZGEÇMİŞ

Doğum tarihi	19.10.1974	
Doğum yeri	İstanbul	
Lise	1989-1991	Bakırköy İbrahim Turhan Lisesi
Lisans	1993-2001	İstanbul Teknik Üniversitesi İşletme Fak. Endüstri Mühendisliği Bölümü
Yüksek Lisans	2002-2007	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Müh. Anabilim Dalı, Sistem Müh..Programı

Çalıştığı kurumlar

2002-2004	Güner Kuyumculuk Makine San.Tic.Ltd.Şti.
2004-2006	Baymak Sanayi ve Tic.A.Ş.
2006-2007	Legrand Elektrik A.Ş.
2007-Devam Ediyor	Baymak Sanayi ve Tic.A.Ş.