

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMA VE BİR UYGULAMA

YÜKSEK LİSANS TEZİ
Müh. Ayça ALTAY

Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Programı : ENDÜSTRİ MÜHENDİSLİĞİ

HAZİRAN 2007

GENETİK ALGORİTMA VE BİR UYGULAMA

YÜKSEK LİSANS TEZİ
Endüstri Müh. Ayça ALTAY
507051105

Tezin Enstitüye Verildiği Tarih : 7 Mayıs 2007
Tezin Savunulduğu Tarih : 11 Haziran 2007

Tez Danışmanı : Doç. Dr. Y. İlker TOPÇU (İ.T.Ü.)

Yard. Doç. Dr. Gülgün KAYAKUTLU (İ.T.Ü.)

Diğer Jüri Üyeleri Prof. Dr. Ramazan EVREN (İ.T.Ü.)

Doç.Dr. Yasemin Claire ERENSAL (Doğuş Üniv.)

Prof. Dr. H. Ziya ULUKAN (Galatasaray Üniv.)

HAZİRAN 2007

ÖNSÖZ

Bu tez çalışmasında, İstanbul Sanayi Odası'na bağlı şirketlere atanmak üzere İstanbul Teknik Üniversitesi son sınıf öğrencilerinin Genetik Algoritma (GA) ile belirlenmesi yapılmıştır. Uygunluk fonksiyonu olarak iki amaç fonksiyonu alınmıştır. Bunlardan biri tercihlerin arasındaki sapmayı minimize eden Minimum Kareler Yöntemi, bir diğeri de Analitik Hiyerarşi Süreci'nde kullanılan Saaty Uygunluk İndeksi'dir. Bu iki amaç fonksiyonu kullanılarak, Microsoft Excel Visual Basic Editor'de yazılan makro ile çözümler listelenmiş ve hem gelecek çalışmalar hem de gelecek seneki eşleşmeler için önerilerde bulunulmuştur.

Tez çalışması boyunca gösterdiğim çabada, tez hocalarım sayın Doç. Dr.Y. İlker TOPÇU' nun ve Öğr. Gör. Dr. Gülgün KAYAKUTLU' nun her aşamada büyük yardımı oldu. Kendilerine desteklerinden dolayı teşekkür ediyorum. Yüksek lisans dönemim boyunca bana burs ile katkıda bulunan TÜBİTAK BİDEB'e ve tez çalışmam sırasında beni yüreklendiren, destekleyen aileme de teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÖZET	viii
SUMMARY	ix
1. GİRİŞ	1
2. EN İYİLEMeye GİRİŞ	3
2.1. En İyileme	3
2.2. En İyileme Yöntemleri	3
2.2.1. Sayma Yöntemleri	3
2.2.2. Deterministik Yöntemler	3
2.2.2.1. Açgözlülük Algoritması (Greedy Algorithm)	3
2.2.2.2. Tepe Tırmanma (Hill Climbing)	3
2.2.2.3. Dal Sınır Algoritması	4
2.2.2.4. Matematiksel Programlama	4
2.2.3. Stokastik Yönyemler	4
2.2.3.1. Rassal Arama	4
2.2.3.2. Benzetim Tavlama	4
2.2.3.3. Monte Carlo Simülasyonu	4
2.2.3.4. Tabu Araştırması	4
2.2.3.5. Yapay Sinir Ağları	4
2.2.3.6. Genetik Algoritma	5
3. GENETİK ALGORİTMA	6
3.1. Genetiğe Giriş	6
3.2. Genetik Algoritma	7
3.2.1. Genetik Algoritmanın Tarihçesi	7
3.2.2. Genetik Algoritma'nın Tanımı	8
3.2.3. Genetik Algoritma İle İlgili Tanımlar	9
3.2.4. Genetik Algoritmanın Adımları	10
3.2.4.1. Genetik Algoritma Kodlama Yöntemleri	11
3.2.4.2. Yeniden Üreme Yöntemleri	13
3.2.4.3. GA'da Kullanılan Çaprazlama Yön.	15
3.2.4.4. GA'da Kullanılan Dönüşüm Yöntemleri	22
3.2.5. Genetik Algoritma ile İlgili Bir Örnek	24
3.2.6. Plan (Katar), Plan Teoremi ve Yapı Taşı Teoremi	28

3.2.7. GA ve Diğer En İyileme Yöntemlerinin Karşılaştırılması	29
4. ATAMA PROBLEMLERİ VE GENETİK ALGORİTMA	30
5. İSO – İTÜ ŞİRKET/ÖĞRENCİ ATAMA PROBLEMİ	33
5.1. İSO-İTÜ Şirket/Öğrenci Eşleştirme Problemi	33
5.2. Problemin Genetik Algoritma ile Çözümü	37
5.2.1. Genetik Kodlama	37
5.2.2. Amaç Fonksiyonun Tanımlanması	38
5.2.3. Başlangıç Populasyonunun Oluşturulması	41
5.2.4. Yeniden Üreme (Seçim) Operatörü	41
5.2.5. Çaprazlama Operatörü	41
5.3. Uygulama	42
5.4. Uygulama Sonuçları	42
5.4.1. Minimum Kareler Amaç Fonksiyonunun Çözümü	42
5.4.2. Saaty Uygunluk İndeksi Amaç Fonksiyonu Çözümü	44
6. SONUÇLAR VE ÖNERİLER	46
KAYNAKLAR	47
EK A – ŞİRKET LİSTESİ	50
EK B – ÖĞRENCİ TERCİHLERİ VE NUMARALARI	51
EK C – ŞİRKETLERİN KROMOZOMDAKİ KONUMLARI	53
EK D – MİNİMUM KARELER EXCEL VBA KODLARI	56
EK E – SAATY UYGUNLUK İNDEKSİ EXCEL VBA KODLARI	67
EK F – MİNİMUM KARELER YÖNTEMİ SONUÇLARI	78
EK G - SAATY UYGUNLUK İNDEKSİ SONUÇLARI	80
ÖZGEÇMİŞ	82

KISALTMALAR

GA :Genetik Algoritma

İTÜ :İstanbul Teknik Üniversitesi

İSO :İstanbul Sanayi Odası

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1 Genetik Algoritma ile İlgili Tanımlar.....	9
Tablo 3.2 Hamming Uzaklığı.....	11
Tablo 3.3 Örnek soru – Uygunluğun hesaplanması	25
Tablo 3.4 Rulet Çarkı Değerleri ve Yeni Tablo.....	26
Tablo 3.5 Yeni Populasyon.....	28
Tablo 5.1 Minimum Kareler Yöntemi 100 İterasyon Çözümü.....	43
Tablo 5.2 Minimum Kareler Yöntemi 100 İterasyon Çözümü.....	43
Tablo 5.3 Saaty Uygunluk İndeksi 100/500 İterasyon Çözümü 1.....	44
Tablo 5.4 Saaty Uygunluk İndeksi 100/500 İterasyon Çözümü 2.....	45
Tablo A.1 Şirket Listesi	51
Tablo B.1 Öğrenci Tercih ve Numaraları.....	52
Tablo C.1 Şirketlerin Kromozomdaki Konumları ve Tercihleri.....	54
Tablo F.1 Şirketlere Atanan Öğrenciler (Minimum Kareler).....	78
Tablo G.1 Şirketlere Atanan Öğrenciler (Saty Uygunluk İndeksi).....	80

ŞEKİL LİSTESİ

	Sayfa No
Şekil 3.1 :DNA Yapısı.....	7
Şekil 3.2 :Çaprazlama.....	7
Şekil 3.3 :Dönüşüm.....	7
Şekil 3.4 :Genetik Algoritmanın Adımları.....	10
Şekil 3.5 :Bir Noktalı Çaprazlama.....	16
Şekil 3.6 :İki Noktalı Çaprazlama.....	16
Şekil 3.7 :Kes – Ekle Çaprazlama.....	17
Şekil 3.8 :Ebeveynler.....	18
Şekil 3.9 :Çaprazlamanın Gerçekleştirilmesi.....	18
Şekil 3.10 :Eşleşme İlişkilerinin Belirlenmesi.....	18
Şekil 3.11 :Çocuklar.....	19
Şekil 3.12 :Sıralı Çaprazlama.....	19
Şekil 3.13 :Pozisyon Tabanlı Çaprazlama.....	20
Şekil 3.14 :Sıra Tabanlı Çaprazlama.....	20
Şekil 3.15 :Dairesel Çaprazlamada Döngünün Kurulması.....	21
Şekil 3.16 :Yarı-çocuğun Oluşturulması.....	21
Şekil 3.17 :Ebeveyn 2’den Genlerin Silinmesi.....	21
Şekil 3.18 :Ebeveyn 2’den Çocuğun Oluşturulması.....	21
Şekil 3.19 :Basit Dönüşüm.....	22
Şekil 3.20 :Değiştirme Dönüşümü.....	22
Şekil 3.21 :Ekleme Dönüşümü.....	23
Şekil 3.22 :Karşılıklı Değişim Dönüşümü.....	23
Şekil 3.23 :Yer Değiştirme Dönüşümü.....	24
Şekil 3.24 :Rulet Çarkı.....	26
Şekil 3.25 :Çaprazlama Operatörü.....	27
Şekil 5.1 :Şirket A’nın Tercih Vektörü.....	35
Şekil 5.2 :Şirket B’nin Tercih Vektörü.....	35
Şekil 5.3 :Şirket C’nin Tercih Vektörü.....	35
Şekil 5.4 :Öğrenci A’nın Tercih Vektörü.....	36
Şekil 5.5 :Öğrenci B’nin Tercih Vektörü.....	36
Şekil 5.6 :Öğrenci C’nin Tercih Vektörü.....	36
Şekil 5.7 :Olurlu Bir Eşleşme.....	36
Şekil 5.8 :Kromozom Yapısı.....	38

ÖZET

İstanbul Teknik Üniversitesi Endüstri Mühendisliği Bölümü son sınıf öğrencileri, bitirme çalışmalarına hazırlık olması ve ön bilgi toplanması amacıyla 7. dönemlerinde Endüstri Mühendisliği Projesi adlı 1 kredilik dersi almaktadırlar. 8. dönemde ise 1 kredilik Bitirme Çalışması dersinde, Endüstri Mühendisliği Projesi dersinde teorik bilgilerini işledikleri konuları uygulamaya dökmektedirler. Ne var ki, sanayide uygulama alanı bulmak öğrenciler için büyük bir sorun oluşturmaktadır. Bu nedenle, 2006-2007 dönemi öğrencileri ile birlikte yeni bir uygulama başlatılmıştır. İstanbul Sanayi Odası'na bağlı bazı şirketler ile anlaşılmış ve öğrenciler bu şirketlere atanarak uygulamalarını gerçekleştirme şansı bulmuşlardır. Ancak, şirketler bu öğrencilerin belirli çalışma konularında çalışmalarını istemektedir. Her şirketin istediği çalışma konusu diğerlerinden farklıdır. İşte bu noktada, hangi öğrencinin hangi şirkete atanacağı sorunu ortaya çıkmaktadır.

Bu çalışmanın amacı, öğrenciler ve şirketlerin arasındaki en uygun eşleşmeleri listelemektir. Bunun içinse Genetik Algoritma (GA) kullanılacaktır. GA ile çeşitli amaç fonksiyonlarının karşılaştırılması yapılacak, hangi amaç fonksiyonunun daha verimli işlediği belirlenecektir.

Bunun için, birinci adımda öncelikle şirketlere 9 çalışma başlığının bulunduğu bir liste gönderilir. Şirketler atanmasını istedikleri öğrencilerin çalışmaları içinden tercih yaparlar. Tercih konusunda kısıtları bulunmamaktadır.

İkinci adımda, öğrencilerin Endüstri Mühendisliği Projeleri'ne bakılarak çalışma konuları belirlenir. Üçüncü adım ise eşleştirmenin yapıldığı adımdır. Bu çalışmada, Microsoft Excel Visual Basic Editor'de yazılan bir makro ile iki amaç fonksiyonu kullanılarak GA ile öğrenciler eşleştirilmiştir. Bir sonraki adımda, elde edilen sonuçlar listelenmiştir. Buna göre amaç fonksiyonlarının etkinlikleri belirlenmiş ve sonraki çalışmalarda yapılabilecekler belirtilmiştir. Ayrıca eşleşmenin diğer senelerde daha verimli yapılabilmesi için öneriler bulunmaktadır.

SUMMARY

Junior grades students at İstanbul Technical University Industrial Engineering department have a compulsory “Industrial Engineering Project” course which is 1 credit and a preparation step for their Graduation Theses in their 7th terms. Finally, on the 8th term, they have their “Graduation Thesis” courses which is also 1 credit. In this “Graduation Thesis” course, a junior student is expected to apply his subject on which he previously studied information in the “Industrial Engineering Project” course. However, for students it is difficult to find a company to apply their studies in. For this reason, a new application has been started beginning from 2006-2007 junior students. Agreements have been made with Istanbul Chamber of Industry, that enables students to apply their subjects to companies that are members of Istanbul Chamber of Industry. On the other hand, companies demand students according to their subjects. Each company determines and selects different lists and selections of subjects. As a result, the problem of assigning a student to a company has risen.

The aim of this study is to make lists of most appropriate matchings between students and companies. For this, a Genetic Algorithm (GA) has been applied. Using GA, different fitness functions and their effectivenesses will be compared.

For this, in the first step a list of 9 subjects has been sent to companies. They select the subjects on which they desire the students to work on without any constraints.

In the second step, subjects of each student are determined by considering their “Industrial Engineering Project” studies. The third step is to match students and companies by a Genetic Algorithm (GA) macro coded in Microsoft Excel Visual Basic Editor. In the next step, results are listed. According to these results, effectivenesses of functions are determined and propositions for further studies have been made. Additionally, propositions on matching have been made for the following years.

1. GİRİŞ

Genetik Algoritma (GA), bir mühendis ve aynı zamanda bir biyolog olan John Holland tarafından 1965'te ortaya atılmış bir yöntemdir[1]. Holland, Darwin'in Evrim Teorisi'ni ve "Güçlü olan hayatta kalır" prensibini en iyileme sürecine uyarlamıştır. Evrim sürecinin en iyileme sürecinde de başarılı sonuçlar verdiği görülmüş ve geliştirilmeye başlanmıştır. GA'da aynı evrim sürecinde olduğu gibi bir çok bireyden çözüme başlanmaktadır. Bu özelliği sonucu zorlukla bulunabilen problemlerde kullanılmaya başlanmış ve başarılı sonuçlar vermiştir. Bu konuyla ilgili sayısız çalışma yapılmış ve çözüm alanı Holland'ın başladığından çok daha geniş bir alana yayılmıştır.

GA en çok sıralama, çizelgeleme, atama, dağıtım, planlama problemlerinin çözümünde kullanılmaktadır[2]. Bilgisayar çağına girilmesiyle birlikte GA uygulamaları bilgisayar programları ile daha hızlı ve kolay hale getirilmiştir. Bilgisayarlardaki bu gelişme, sadece sonuca ulaşmayı hızlandırmakla kalmamış, daha ileri GA'ların geliştirilebilmesinde çok büyük bir yardımcı olmuştur. Bugün bir çok yazılımda GA yazılabilmektedir ve uygulama programları içerisinde GA eklentileri oluşturulabilir durumdadır.

Günümüzde çeşitli GA, teknikleri çok sayıda alanda uygulanmaktadır. Bu çalışmada ise GA, İSO – İTÜ şirket/öğrenci eşleştirme probleminde kullanılacaktır. İSO – İTÜ şirket/öğrenci eşleştirme problemi, son sınıftaki tez yapan İTÜ öğrencilerini son sınıftaki öğrencileri kendi bünyeleri için hazırlamak isteyen İSO'ya bağlı şirketler arasında tercihlere bağlı bir eşleştirme yapma problemidir. İlk olarak, bu sene çalışmaya başlanmış ve eşleştirmeler sezgisel yapılmıştır. Çalışmanın bir amacı da, hem bundan sonraki seneler için eşleştirmenin kolay yapılmasını hem de eşleşmenin

olabildiğince uygun olmasını sağlamaktır. Tercihler bu eşleşmedeki en önemli ölçütlerdir.

Çalışmanın ikinci bölümünde, en iyileme yöntemlerine bir giriş yapılmış, kısaca en iyileme yöntemleri tanıtılmıştır. üçüncü bölümde ise, en iyileme yöntemlerinden biri olan GA tanıtılmış ve diğer en iyileme yöntemlerinden farkları gösterilmiştir. Bu bölümde ayrıca GA, ayrıntılarla açıklanmış ve gelişim süreci incelenmiştir.

Dördüncü bölümde ise, atama problemleri, proje atamaları ve GA ile ilgili yapılmış çalışmalar genel bir başlıkla özetlenmiş ve bu çalışmaların nerede kesiştikleri belirlenmiştir. Bu bölümde ayrıca İSO-İTÜ şirket/öğrenci atama probleminin neden GA ile çözülmesi gerektiğinden bahsedilerek, problemin çözümüne geçiş yapılmıştır.

Beşinci bölüm olan uygulama bölümünde iki değişik uygunluk (amaç) fonksiyonu ile bu seneki İSO-İTÜ şirket/öğrenci eşleştirme problemi Microsoft Excel Visual Basic Editor'de yazılan makro ile çözülmüştür. Bu uygunluk fonksiyonları Minimum Kareler Yöntemi ve Saaty Uygunluk İndeksi'dir. Bu uygunluk fonksiyonlarının getirdikleri çözümler 100 ve 500 iterasyon için belirlenerek, karşılaştırılmıştır. Aynı zamanda bu seneki eşleşmeler için olası çözümler sunulmuştur.

Son bölümde ise uygulama sonuçları yorumlamış ve sonraki çalışmalarda yapılabilecekler listelenmiştir.

2. EN İYİLEMEYE GİRİŞ

2.1 En İyileme

En iyileme bir sorunun çözüm kümesindeki en iyi sonucu ortaya çıkarma, problem için en iyi çözümü bulma işlemidir. Bu en iyi çözüm iki şekilde bulunur: yerel en iyi çözüm ve global en iyi çözüm. Global en iyi çözüm birbirine yakın tüm çözüm kümeleri içinde en iyi çözümü bulmayı amaçlar. Bölgesel veya yerel en iyi çözüm ise başlangıç noktasına göre değişir ve tüm çözümler içinde değil belirli bir komşuluk içindeki en iyi noktadır. En iyileme problemlerinde amaç eldeki amaç fonksiyonu ve kısıtlar çerçevesinde en küçük ya da en büyük değer veren noktayı bulabilmektir[1]. En iyileme için çeşitli yöntemler geliştirilmiştir.

2.2. En İyileme Yöntemleri

2.2.1. Sayma Yöntemleri

Diğer yöntemlerden üstünlüğü kolaylık sağlamasıdır. Olası her noktanın değerini hesaplayarak en iyi noktayı bulmayı hedefler. Ancak bu yöntemler yavaş çalışmakta ve problemin çözümünde verimsiz kalmaktadır. Büyük arama uzaylarında ise uygulanması çok zor olan yöntemlerdir[2].

2.2.2. Deterministik Yöntemler:

2.2.2.1. Açgözlülük Algoritması (Greedy Algorithm)

Yerel en iyi çözümler bulunarak bu en iyi çözümlerden birinin global en iyi çözüm olduğu düşünülür. En iyi çözümü bulurken o an eldeki veriden elde edilebilecek sonuçları düşünür[3].

2.2.2.2. Tepe Tırmanma (Hill Climbing)

Gradyen değeri hesaplanarak bu değerin 0'a eşit olduğu noktalarda en iyi çözümü arama yöntemidir. Bu yöntemin uygulanabilmesi için arama yapılacak düzlemin sürekli olması gerekir ve bölgesel en iyi çözümlere takılma olasılığı yüksektir[2].

2.2.2.3. Dal-Sınır Algoritması

Her adımda bir sınır değeri hesaplanarak diğer adımlarda elde edilen sonuçlarla karşılaştırma yöntemidir [3].

2.2.2.4. Matematiksel Programlama

Doğrusal veya doğrusal olmayan matematiksel modellerin kurulmasıyla en iyi çözüm aranır.

2.2.3. Stokastik Yöntemler

2.2.3.1. Rassal Arama

Stokastik bazı sonuçlar elde edilerek karşılaştırılır. En iyi çözüm kabul edilir. En kolay stokastik yöntemdir [3].

2.2.3.2. Benzetim Tavlama

Katı cisimlerin özellikle metallerin soğurkenki dizilişlerini örnek alan bir yapay zeka en iyileme yöntemidir. Özellikle metaller ısıtıldığında, atomlar ısı kaynaklı hareketlenme gösterir. Benzetim tavlama bu özelliği taklit etmektedir. Madde soğuk olduğunda atomlar yerel en iyi noktalardadır. Madde ısıtıldığında genel en iyi noktalar aranır. Rassallık oranı yüksektir. Global en iyi çözümlere ulaşma amacı vardır [4].

2.2.3.3. Monte Carlo Simülasyonu

Parametreler değiştirilerek en iyi çözüm bulunur. Genellikle en iyi çözümü bulmak için kullanılmaz. Durum incelemesi için daha uygundur.

2.2.3.4. Tabu Arařtırması

Maksimum/minimum için çözümler üretilir. Üretilen çözümler bir yasak listesinde tutulur. Yeni çözümler üretilerek tabu listesindekiler ile karşılaştırılır. Kısa sürede en iyi çözüme gitmeyi amaçlar.

2.2.3.5 Yapay Sinir Ağları

Beyindeki sinir ağlarını simüle eden bir yöntemdir. Öğrenme ve test verileri kullanılarak hücrelere fonksiyonları öğretilir.Çok boyutlu problemlerde kullanılması uygundur [4].

3. GENETİK ALGORİTMA

3.1. Genetiğe Giriş

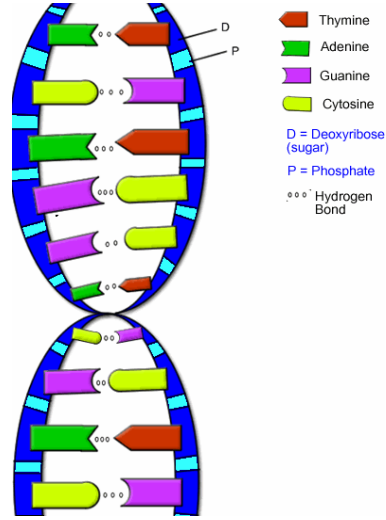
İnsan hayatı genler tarafından yönetilir. Genler, canlılarda görünüş, kişilik, davranışlar gibi özellikleri belirleyen yapıtaşları ve bu canlılar hakkında bilgi elde etmek için kullanılan bir birimdir. Ayrıca, bir canlının herhangi bir özelliğini denetleyen bir kromozom parçasıdır. [5]

Genetik Algoritma'nın kökeni Charles Darwin'e dayanır. Darwin, 1859 yılında "Türlerin Kökeni" eserinde şu tezleri öne sürdü [6 - 7]:

1. Türler sürekli bir gelişme içindedir.
2. İnsanların ataları maymunlar olabilir.
3. Organizmaların üreme potansiyeli yüksektir ama hayatta kalan birey oranı çok düşüktür.
4. Bir türün bireyleri arasındaki değişimler çok yüksektir.
5. Bireyler arasındaki bu çeşitlilik kalıtsal olarak ebeveynlerden çocuklara aktarılır.

Buna göre bir canlının doğada sağ kalabilmesi, genlerinin doğaya ne kadar uygun olduğuyla doğru orantılıdır [6]. Doğada, güçlü olan canlının hayatta kalabilmesine doğal seçim denir[6]. Bu çalışmaların üstüne Hugo de Varis bitkilerde varyasyonlar üzerine deneyler yaptı ve bazı bitkilerin ebeveyn özelliklerinin dışında genlerde bazı ani değişiklikler olduğunu ortaya koydu. Bu değişikliklere dönüşüm (mutasyon) denildi [7].

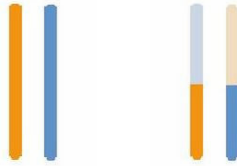
Canlıların gen özelliklerini belirleyen yapıtaşları DNA (Deoksiribo Nükleik Asit)'lardır. DNA üçlü fosfat bağı, deoksiriboz ve 4 çeşit aminoasitten oluşur. Örnek bir DNA yapısı aşağıda verilmiştir [5].



Şekil 3.1: DNA Yapısı

Genlerin güçlü olabilmesi için, genlerde gerçekleşen 2 işlem vardır. Bunlardan bir çaprazlama, bir diğeri ise dönüşümdür.

Çaprazlama iki kromozom arasındaki gen alışverişi olarak tanımlanır. Dönüşüm ise bir kromozomdaki herhangi bir ya da birkaç genin değişmesiyle gerçekleşir. Çaprazlama ve dönüşüm örnekleri aşağıda verilmiştir.



Şekil 3.2: Çaprazlama



Şekil 3.3: Dönüşüm

3.2. Genetik Algoritma

Genetik Algoritma (GA) ilk defa Holland tarafından 1975te kullanılmıştır ve canlı sistemlerdeki genetik şifre mantığı kullanılarak sezgisel olarak en iyi çözümü veya en iyi çözüme yakın olabilecek bir sonuç bulmayı hedefler. Bu mantık doğal seçim yani güçlü bireyin hayatta kalma olasılığının yüksek olmasıdır. Bu yöntemle evrim sonucu hayatta kalan birey en iyi sonuç olarak alınır [5].

3.2.1. Genetik Algoritmanın Tarihçesi

Genetik Algoritma (GA), ise Darwin ve onu izleyen bilim adamlarının çalışmalarının en iyileme problemleri üzerine uygulanmasıdır. “Genetik Algoritma” kalıbı, ilk olarak 1967 yılında Bagley tarafından bir oyun programının yenmek üzere tasarlanmasında kullanılmıştır [7]. Bagley’in kullandığı GA yöntemleri bugünkünden çok farklı olmakla birlikte günümüz GA’ları için temel olarak kullanılmıştır. Rosenberg de aynı dönemde bu algoritmaya biyolojik ve benzetimsel etmenleri eklemiştir [8]. Genetik Algoritmanın bir çok uygulama alanı bulunmaktadır. De Jong bu çalışmaları matematiksel boyuta taşımış ve fonksiyon enküçüklenmesi için kullanmıştır [8]. Bagley ,Rosenberg, De Jong gibi öncülerinin bulunmasına rağmen, GA’nın babası Holland olarak kabul edilmektedir çünkü Holland ‘Genetik Algoritma’ kavramını “Cellular Automata” çalışmaları ve “Doğal ve Yapay Sistemlerde Uyarılma” kitabının yayınlanması ile literatüre kazandırmıştır. Holland’ın bu kitabında ana düşünce şudur: “Verilen bir popülasyonun genetik havuzu potansiyel olarak istenen en iyi çözümü içerir veya uyarılan probleme ilişkin iyi bir çözüm vardır.” [7] Holland GA’nın temel ilkeleri olan yeniden üreme ve çaprazlama operatörlerini tanımlamıştır. Holland’dan sonra Genetik Algoritma’ların gelişimi doktora öğrencisi olan Goldberg ile devam etmiştir. Goldberg, GA’nın teoriden pratiğe dönüştürülemez tabusunu yıkmış ve 1989’da yayınlanan “Genetic Algorithms in Search, Optimization and Machine Learning” adlı kitabında GA içeren 83 uygulama göstermiştir. Goldberg’i takiben GA çalışmaları geniş kullanım alanları bulmaya başlamış ve GA kodlaması yapılan bilgisayar programları ortaya çıkmıştır. Bunların en önemlisi ve ünlüsü 1992’de John Koza tarafından LISP ile yazılan bilgisayar programıdır [7 - 8].

3.2.2. Genetik Algoritma'nın Tanımı

Genetik Algoritma, canlı sistemlerdeki genetik şifre mantığı kullanılarak sezgisel olarak en iyi çözüm veya en iyi çözüme yakın olabilecek bir sonuç bulmayı hedefler. Bu mantık, doğal seçime yani güçlü bireyin hayatta kalma olasılığının yüksek olmasına dayanır. Bu yöntemle evrim sonucu hayatta kalan birey en iyi sonuç olarak alınır [5]. GA sezgisel bir yöntemdir.

Doğadaki bireylerin birleşerek çocukları oluşturması ve bu çocuklardan güçlü olanın hayatta kalması GA'nın temelini oluşturur [7]. GA, bir bireyler topluluğuyla başlayarak bu bireylerden çocuklar elde eder. Bu bireylerin özellikleri ve sayısı rassal olarak belirlenir. Bu bireyler de rassal olarak eşleştirilerek ebeveyn olurlar. Bu ebeveynlerden elde edilen çocuklar seleksiyon, çaprazlama ve dönüşüm işlemlerine maruz kalırlar. Bu işlemlerin tekrarlanması ile evrim gerçekleştirilir. Genetik algoritmanın bitirilmesi bitiş ölçütlerinin gerçekleşip gerçekleşmediğine bağlıdır.

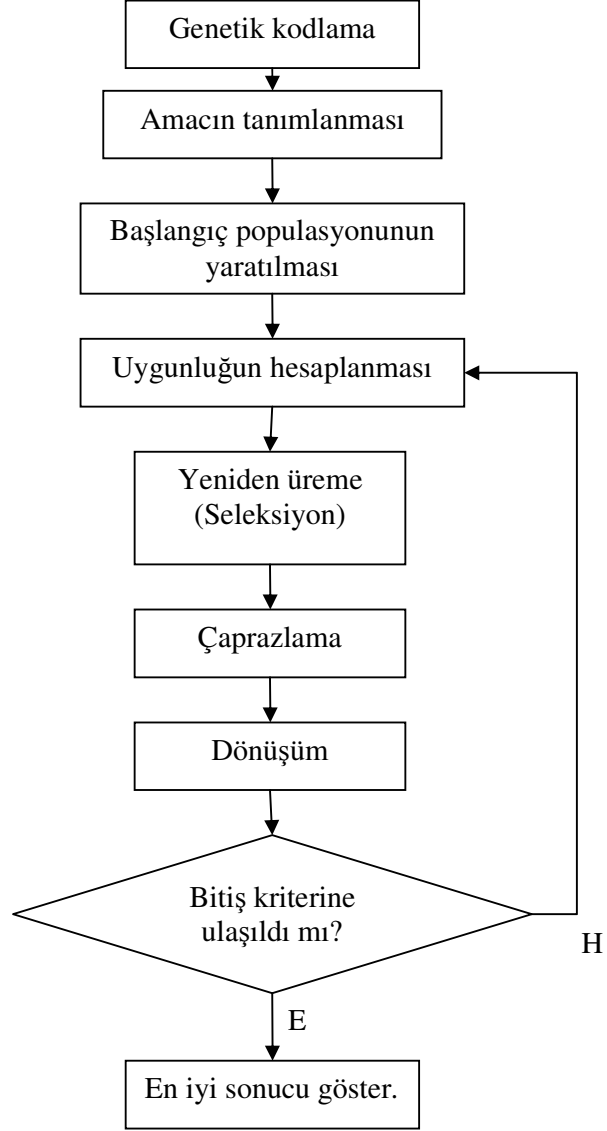
3.2.3. Genetik Algoritma İle İlgili Tanımlar [7-9]

Tablo 3.1: Genetik Algoritma ile İlgili Tanımlar

Terim	GA Karşılığı	Tanım
Gen	Karakter özelliği	Bit olarak da tanımlanır. Bireyin karakterini dolayısıyla de değerini belirleyen öğedir.
Kromozom	Birey	Dizi olarak da tanımlanır. Genlerin birleşerek oluşturduğu dizidir. Kromozomlar belli bir kodlama sistemiyle oluşturulurlar. Aday çözümleri gösterirler.
Genotip	Bireyin gen yapısı	Aday çözüm olan kromozomların içindeki belirli gen gruplarına genotip denir.
Fenotip	Deşifre edilmiş yapı	Alternatif çözüm kümesidir. Genotipin deşifre edilerek asıl değerini belirlenmiş halidir.
Allel	Özellik değeri	Genlerin kaç değişik değer alabiliyorsa, bu değerlerin kümesidir.
Lokus (Locus)	Karakterin yeri	Genin kromozomdaki pozisyonu (yeri)dir.
Popülasyon	Aday çözümler topluluğu	Kromozomların bir araya gelerek oluşturduğu çözüm topluluğudur. Genellikle algoritma boyunca sabit tutulur. Bu sayı çoğunlukla 10 ila 100 arasında değişir.
Uygunluk Fonksiyonu	Amaç fonksiyonu	Bireyin hayatta kalma durumunu yani uyumunu gösteren değerdir. Yüksek değerler bireyin hayatta kalma olasılığının daha yüksek olduğunu belirtir.
Seleksiyon	Seçim	Popülasyon içinden hayatta kalabilen canlıların seçilmesidir. Seçim için uygulanan yöntemler açıklanacaktır.

3.2.4. Genetik Algoritmanın Adımları

GA'nın işleyişi aşağıdaki şekilde verilmiştir [4].



Şekil 3.4: Genetik Algoritmanın Adımları

Adım 1. Genetik Kodlama: Genetik kodlama işlemi fenotipin genotipe dönüştürülme işlemidir. Bir popülasyon kromozom topluluğu, kromozom ise genler topluluğu olarak açıklanır. Kromozomun yapısı ise aşağıda verilmiştir.

$$S = S_1 S_2 \dots S_j \dots S_n$$

Burada s_i , $i = 1, 2, \dots, n$ genleri göstermektedir. i değeri genin lokusudur. Herhangi bir s_i geninin alabileceği değişik değerler allel olarak adlandırılır. Bir kromozomun uzunluğu (n) genellikle sabit tutulur [10].

Herhangi bir sayı değerini genetik kodlamaya çevirme işlemi bir fonksiyon olarak ele alınabilir. Bu fonksiyonda fenotipler çözüm uzayında, genotipler kod uzayında yer alırlar. Bir kodlama fonksiyonunun başarılı olabilmesi için çözüm uzayı ve kod uzayı birebir eşlenebilir olmalıdır [11].

Gerçek sayılar, tamsayılar, harfler ya da bazı semboller kodlama için uygundur ancak kodlama için en çok kullanılan yöntem Holland tarafından geliştirilen 0-1 ikili kodlama sistemidir. Kodlamanın doğru yapılması GA'nın işleyişi için çok önemlidir. Kullanılan kodlama yöntemleri aşağıda verilmiştir [10].

3.2.4.1. Genetik Algoritma Kodlama Yöntemleri

- **İkili Kodlama:**Çözüm uzayındaki değerlerin ikilik sayma sistemine dönüştürülmesiyle gerçekleşir. İkili kodlama tekniğinin avantajı ileride anlatılacak olan çaprazlama ve dönüşüm kolaylaştırmasıdır. Örneğin 7 kod uzayında 0111 ile gösterilirken, 8 kod uzayında 1000 ile gösterilir. Bu kodlama sisteminde karşılaşılan sorun şudur: 2'lik sayma sisteminde 0111 ve 1000 ardışık sayılar değildir. Bu sorunu gidermek için gri kodlama tekniği geliştirilmiştir [9].
- **Gri Kodlama:** 0-1 ikili kodlamadan farkı Hamming Uzaklığı kavramını geliştirmesidir. Sayıları ikili sistemde yazmaya başladığında ilk olarak Hamming Uzaklığı ölçülür. Hamming Uzaklığı, kromozomdaki değişen genlerin sayısını gösterir. Oysa ikili kodlamada aradaki Hamming Uzaklığı'nın 1 olması istenir. 0-7 arası sayıların gri kodlamada gösterimi aşağıda verilmiştir [8].

Tablo 3.2: Hamming Uzaklığı ve Gri Kodlama

Sayı	İkili Kodlama Gösterimi	Hamming Uzaklığı	Gri Kodlama Gösterimi	Hamming Uzaklığı
0	000		000	
1	001	1	001	1
2	010	2	011	1
3	011	1	010	1

Sayı	İkili Kodlama Gösterimi	Hamming Uzaklığı	Gri Kodlama Gösterimi	Hamming Uzaklığı
4	100	3	110	1
5	101	1	111	1
6	110	2	101	1
7	111	1	100	1

- Harf Kodlama: 0-1 ikili kodlamanın kullanılmak istenmediği durumlarda kullanılır. Örneğin [0-15] aralığındaki değerler, ikili kodlama dışında İngiliz Alfabeti'ndeki [A-O] harfleri ile kodlanabilir [8].

Çözüm uzayı bu yöntemler dışında, problem tipleri için özel olarak geliştirilen yöntemlerle kodlanabilirler [9].

Eğer çözüm uzayında birden çok değişken için değer aranıyorsa, bu durumda kromozomlar bu değişkenlerin birleştirilmesi ile gösterilir.

$$\underbrace{0010011011}_{X_1} \dots \dots \dots \underbrace{10101}_{X_n}$$

Adım 2. Amacın Tanımlanması: Elimizde çözüm kümesi olduğunda bu çözümlerin ne kadar iyi olduğunu belirlemek gerekir. Bunun için amaç fonksiyonu tanımlanmalıdır. Genetik Algoritma'nın bir avantajı da amaç fonksiyonunu formüle etmenin zor olduğu problemlere kullanım kolaylığı getirmesidir.

Adım 3. Başlangıç Populasyonun Yaratılması: Bu adım ile algorithmada kullanılacak bireyler (ebeveynler) oluşturulur. Burada yapılacak seçim başlangıç populasyonundaki birey sayısıdır. Az sayıda birey algorithmada kullanılabilecek bir çok değer işlenmeden atlanmasına neden olur. Bunun nedeni çaprazlamanın az sayıda birey arasında gerçekleşmesi ve çeşitliliğin daha az olmasıdır. Çok sayıda birey ise algoritmanın yavaşlamasına neden olur. Burada her döngüde (nesil) daha çok birey incelenir. Populasyon sayısı belli bir değeri geçtikten sonra daha fazla arttırmak algoritmanın verimliliği azalmaktadır. Her problem tipi için en iyi çözümün başlangıç populasyonu sayısı farklıdır. Nesiller boyunca populasyon sayısı genellikle korunur [7 - 9].

Adım 4. Yeniden Üreme (Seleksiyon): Yeniden üreme operatörü, doğadaki doğal seleksiyon işleminin GA tarafından uygulanmasıdır. Bu işlemle güçlü bireylerin hayatta kalması ve güçlü çocuklar oluşturması hedeflenir [10]. Burada öncelikle çocukları oluşturacak bireylerin eşleşmesi gerekmektedir. Bunun için 2 temel model vardır: Göç Modeli ve Komşuluk Modeli [9]. Göç Modeli'nde popülasyondaki kromozomlar belli sayıdaki alt gruplara ayrılır. GA il her alt popülasyon ayrı ayrı değerlendirilir. Değerlendirme işleminden sonra alt popülasyonlar arası göç yolları oluşturularak gen aktarımı yapılır [12]. Komşuluk Modeli'nde ise daha önce belirlenmiş n boyutlu uzayda her kromozomun yeri belirlenir. Bu kromozomlar için belirli bir komşuluk seçilir ve bu noktadaki bireyler belirlenir. Seçilen komşuluk bir uzaklık fonksiyonunun temsil ettiği kromozomlar bütünüdür. Komşuluk içinde kalan bireylerle seçilen ana kromozom çiftleştirilerek tüm sonuçlar birbiriyle karşılaştırılır. Böylelikle çiftleşme için en iyi birey seçilir [13]. Üreme için bir seçilmesinin en basit yolu ise rassal seçimdir [5].

Bireyler seçildikten sonra yeniden üreme işlemine geçilir. Üreme için bir çok yöntem bulunmaktadır.

3.2.4.2.Yeniden Üreme Yöntemleri

- **Rassal Üreme:** Varolan ebeveynlerden popülasyon sayısı kadar rassal seçim yapılır.
- **Rulet Çarkı Yöntemi:** En sık kullanılan yöntemlerden biridir. Bu yöntemle uygunluk değeri yüksek olan güçlü bireyin doğma olasılığının yüksek olması sağlanır. Bu yöntemin adımları şu şekildedir [14].

Adım 1: Tüm fonksiyonlar için uygunluk fonksiyonun aldığı değer diğer bir deyişle uygunluk değeri (f_i) hesaplanır.

Adım 2: Uygunluk değerinin toplamı ($\sum f$) bulunur.

Adım 3: Bireylerin seçilme olasılığı $p_i = f_i / \sum f$ hesaplanır.

Adım 4: Her birey için bu olasılık değerleri sınır alınarak rassal sayı aralıkları belirlenir.

Adım 5: Birey sayısı kadar rassal sayı atılır.

Adım 6: Rassal sayılara karşılık gelen kromozomlar yeni bireyler olarak alınır.

- Beklenen Değer Yöntemi: Rulet çarkından daha hassas bir yöntem olarak kabul edilir. Bu yöntemde daha önce geliş olasılıkları hesaplanmış olan bireylerin beklenen değerleri bulunur. Beklenen değer, olasılıkla birey sayısının çarpımına eşittir. Bu da aşağıdaki eşitlikte verildiği üzere her kromozomun uygunluk değerinin ortalama uygunluk değerine bölünmesiyle elde edilir.

$$E_i = \frac{f_i}{\sum f} \cdot n = \frac{f_i}{f_{avg}} \quad (3.1)$$

Buradan diğer nesle geçmek gerekmektedir. Bir sonraki nesil için kromozomdan beklenen değerinin tamsayısı kadar alınır [10].

- Boltzmann Yöntemi: Bu yöntem bir bireyin bir sonraki nesle aktarılma olasılığını şu şekilde hesaplar:

$$p_i = e^{\beta \cdot \frac{f_i}{\sigma}} \quad (3.2)$$

Burada σ uygunluk değerlerinin standart sapmasıyken β karar verici tarafından seçilen bir sıcaklık parametresidir. Bu yöntem olasılıklar arasında orantı sağlamak için kullanılmaktadır ve bu formül sonucu çıkan olasılıklar doğrusal olarak değil ivmeli olarak artmaktadır. Bu olasılık değerleri arasındaki oran β değerinin ayarlanmasıyla belirlenir [15].

- Sıralı Seçim Yöntemi: Rulet çarkı yönteminde çok yüksek olasılıklı bir kromozoma karşı çeşitlilik yaratmak için kullanılır. Bu yöntemde kromozomlar uygunluk değerine göre azalacak şekilde sıralanır. Bu sıralamaya göre olasılıklar belirlenir ve rassal sayı atışı yapılarak yeni nesilde hangi bireylerin olacağı belirlenir [9].

- Turnuva Yöntemi: Bu yöntemde kromozomlar bir eşleşme havuzuna atarak eşleştirilirler. Bu eşleşmelerden galip çıkan birey güçlü birey olarak tanımlanır ve bir sonraki nesle geçmesi istenir. Bunun için bir p olasılığı belirlenir. p değeri [0.5 – 1] aralığında seçilir. Turnuva yönteminde en iyi

1. birey diğer nesle p olasılığıyla geçer.
2. birey diğer nesle p(1-p) olasılığıyla geçer.
3. birey diğer nesle p(1-p)² olasılığıyla geçer.

n. birey diğer nesle $p(1-p)^{n-1}$ olasılığıyla geçer.[16] Ayrıca, 2 turnuvadan fazlasını kazanan birey diğer nesle aktarılır [9].

- **Elitizm Yöntemi:**Bu yöntem en güçlü bireyin bir sonraki nesilde yer almasını garantilemek ve nesilleri güçlendirmek için uygulanır. Çaprazlama ve dönüşüm sonrasında en güçlü birey yeni nesil arasında yer almamışsa, yeni nesildeki en güçsüz birey öldürülerek yerine bir önceki neslin en güçlü bireyi yerleştirilir [9].
- **Denge Durumu Seçim Yöntemi:** Bu yöntemde populasyonlar arasından belirlenen birey(ler) diğer tüm nesillere aktarılır [14].

Adım 5. Çaprazlama: Çaprazlama genetik algoritma süresince kullanılan en önemli operatördür ve amacı popülasyonda olmayan bireyleri yaratarak bireyler arasındaki çeşitliliğin artırılmasıdır [9 - 10]. Bu işlem ebeveynlerden çocuk oluşturma sürecinde yer alır. Problem çözümleri için bir çok çaprazlama yöntemi üretilmiştir [10]. Genelde tüm çaprazlama operatörlerinde, iki birey bir çiftleşme havuzundan alınarak karşılıklı gen değişimiyle gerçekleşir [14].

Çaprazlama operatörünün en önemli özelliği çaprazlama olasılığı (p_c)dır. Bu olasılık popülasyon içindeki herhangi bir bireyin çaprazlanma olasılığıdır. Bir popülasyon içindeki bireylerin beklenen çaprazlanma sayısı ile çaprazlama olasılığı ile popülasyondaki birey sayısının çarpılmasıyla elde edilir. Çaprazlama olasılığı genellikle 0.6 ile 1.0 arasında belirlenir. Her nesil için çaprazlama olasılığı değişebilir. Çaprazlama olasılığının küçük olması sonuca yavaş ulaşılmasına, büyük olması en iyi çözüm noktasının kaçırılmasına neden olabilir [9].

3.2.4.3.Genetik Algoritmada Kullanılan Çaprazlama Yöntemleri

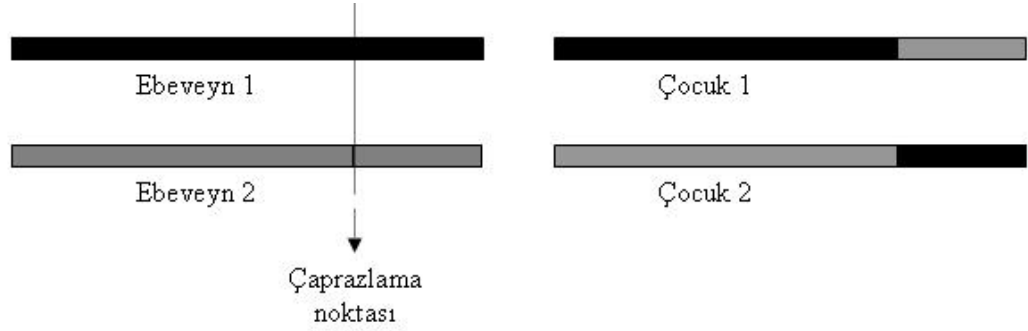
En çok kullanılan çaprazlama yöntemleri aşağıda verilmiştir:

- **Basit Çaprazlama:**

n noktalı çaprazlama da denir. ($n = 1,2,\dots$) En çok kullanılanları bir ve iki noktalı çaprazlamadır.

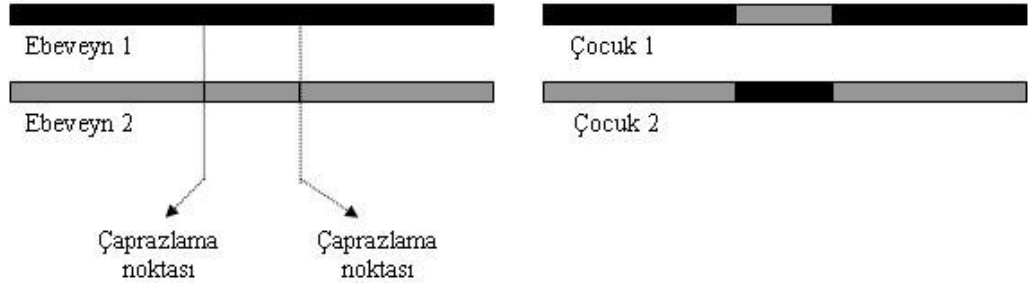
Bir noktalı çaprazlamada iki gen üzerinde sabit bir nokta rassal olarak seçilir ve bu noktanın sağ tarafında kalan genler değiştirilir [14]. Örneğin 01100 ve 11011 genleri

eşleşen ebeveyn kromozomlar olsun. Eğer rassal olarak seçilen sabit nokta 3 ise son 3 basamaktan sağda kalan genler değiştirilir. Elde edilen yeni bireyler 01111 ve 11000 bireyleridir. Bir noktalı çaprazlama aşağıdaki gibi şematize edilebilir [8].



Şekil 3.5: Bir noktalı çaprazlama

Benzer şekilde 2 noktalı çaprazlama’da ise 2 sabit nokta belirlenir ve bu iki nokta arasındaki genler yer değiştirir. Bir önceki örnekte (ebeveynler: 01100 ve 11011) sabit noktalar 2 ve 4 olsun. Bu durumda sondan 2. ve 4. basamaklar arasındaki genler yer değiştirecektir. Yeni oluşan çocuklar 01010 ve 11101’dir. 2 noktalı çaprazlama aşağıdaki gibi şematize edilir [8].



Şekil 3.6: İki noktalı çaprazlama

Basit çaprazlama sırasında ilk gen değişmemekte her durumda sabit kalmaktadır. İlk genin değişmesini sağlamak için n sayısı artırılır. Böylelikle çeşitlilik artırılarak çeşitliliğin artması sağlanır. Bu çaprazlama yöntemi 0-1 ikili kodlanmış bireylerde kullanılırken diğer çaprazlama yöntemleri diğer yöntemlerle kodlanmış bireylerde kullanılır [9].

- Uniform Çaprazlama:

n noktalı çaprazlamanın özel bir uygulamasıdır. Bu yöntemde n genlik kromozom için yine n genlik bir çaprazlama maskesi oluşturulur. Çocuk 1’in genleri eğer

maskede o genin lokusundaki deęer 1 ise Ebeveyn 1'den, 0 ise Ebeveyn 2'den gelir. Çocuk 2'nin genleri ise eęer maskede o genin lokusundaki deęer 0 ise Ebeveyn 1'den, 1 ise Ebeveyn 2'den gelir. Örnek bir uniform çaprazlama ařaęıda verilmiřtir [10].

Ebeveyn 1	110000001
Ebeveyn 2	101110100
Maske	101101101
Çocuk 1	100010001
Çocuk 2	111000100

- Kes – Ekle Çaprazlama (Cut and Splice): Özel bir tek noktalı çaprazlama teknięidir. Adımları řu řekildedir [17]:

Adım 1: Ebeveyn 1'de rassal bir çaprazlama noktası seęilir.

Adım 2: Ebeveyn 2'de rassal bir çaprazlama noktası seęilir.

Adım 3: Ebeveyn 1 ve 2'de çaprazlama noktasının saęında kalan genler deęiřtirilerek çocuklar oluřturulur.



řekil 3.7: Kes – Ekle Çaprazlama

Bu yöntem kullanıldıęında kromozomun uzunluęunun deęiřme olasılıęı yüksektir.

- Kısmi Planlı Çaprazlama (Partially Matched Crossover – PMX): Bu model ilk kez Goldberg tarafından önerilmiřtir. İki noktalı çaprazlanmasının 0-1 ikili kodlama hariç dięer kodlama tekniklerine uygulanması olarak dūřünülebilir. Algoritması řu řekildedir [18]:

Adım 1: Kromozomlar için iki çaprazlama noktası seęilir.

Adım 2: Bu iki nokta arasında kalan genler değiştirilir. (Bu adıma kadar iki noktalı çaprazlama ile aynıdır.) Böylelikle yarı-çocuklar oluşturulur.

Adım 3: Değiştirilen genler arasındaki ilişki (eşleşme ilişkisi) belirlenir.

Adım 4: Değiştirilmeyen bireylere de bu eşleşme ilişkisi uygulanır.

Bir kısmi planlı çaprazlama örneği aşağıda verilmiştir [18].

Adım 1: Çaprazlama noktalarının ve değişecek genlerin belirlenmesi

1	2	3	4	5	6	7	8	9	Ebeveyn 1
5	4	6	9	2	1	7	8	3	Ebeveyn 2

Şekil 3.8: Ebeveynler

Adım 2: Çaprazlamanın gerçekleştirilmesi

1	2	6	9	2	1	7	8	9	Yarı-çocuk 1
5	4	3	4	5	6	7	8	3	Yarı-çocuk 2

Şekil 3.9: Çaprazlamanın gerçekleştirilmesi

Adım 3: Eşleşme ilişkilerinin belirlenmesi

6	9	2	1	$1 \leftrightarrow 6 \leftrightarrow 3$
3	4	5	6	$9 \leftrightarrow 4$
				$2 \leftrightarrow 5$

Şekil 3.10: Eşleşme İlişkilerinin Belirlenmesi

Adım 4: Eşleşme ilişkisinin değişmeyen bireylere uygulanarak çocukların elde edilmesi

3	5	6	9	2	1	7	8	4	Çocuk 1
2	9	3	4	5	6	7	8	1	Çocuk 2

Şekil 3.11: Çocuklar

- Sıralı Çaprazlama (Ordered Crossover - OX): Sıralı çaprazlama ilk olarak Davis tarafından önerilmiştir. Kısmi Sıralı Çaprazlama'nın bir varyasyonudur. Adımları şu şekildedir [18]:

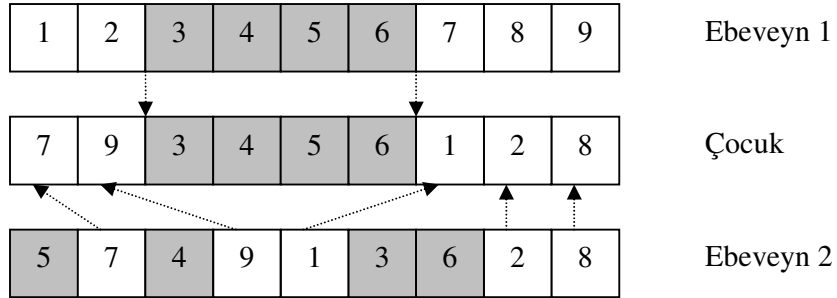
Adım 1: İki çaprazlama noktası seçilir.

Adım 2: Yarı-çocuğa Ebeveyn 1'deki çaprazlama noktaları arasındaki genler olduğu gibi aynı lokusa aktarılır.

Adım 3: Yarı-çocuğa alınan genler Ebeveyn 2'den silinir.

Adım 4: Ebeveyn 2'deki kalan genler çocuğa sırayla aktarılır.

Sıralı çaprazlama ile ilgili bir örnek aşağıda verilmiştir [18]:



Şekil 3.12: Sıralı Çaprazlama

- Pozisyon Tabanlı Çaprazlama (Position – Based Crossover - PBX): İlk olarak Syswerda tarafından önerilmiştir. Hem kısmi sıralı çaprazlamaya hem de sıralı çaprazlamaya benzetilmektedir. Adımları şu şekildedir [18]:

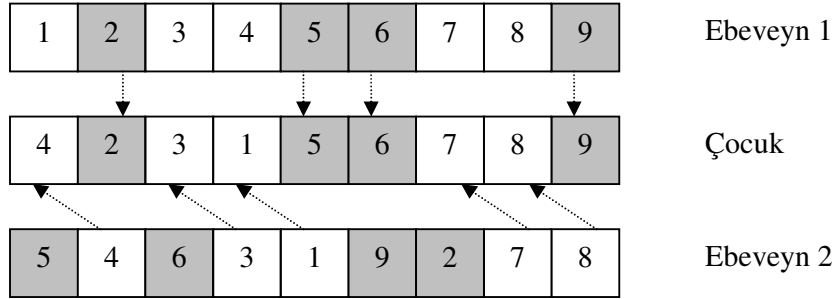
Adım 1: Ebeveyn 1'den rassal sayıda rassal pozisyonda genler belirlenir.

Adım 2: Yarı-çocuğa bu genler aynı lokusta yerleştirilir.

Adım 3: Yarı-çocuğa yerleştirilen genler Ebeveyn 2'den silinir.

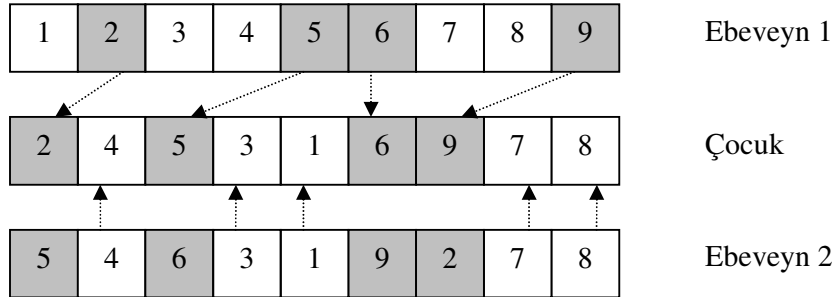
Adım 4: Ebeveyn 2'de kalan genler çocuğa sırayla yerleştirilir.

Pozisyon Tabanlı Çaprazlama ile ilgili bir örnek aşağıda verilmiştir [18]:



Şekil 3.13: Pozisyon Tabanlı Çaprazlama

- Sıra Tabanlı Çaprazlama (Order-based Crossover - OBX): Bu yöntem de ilk olarak Syswerda tarafından önerilmiştir. Pozisyon tabanlı çaprazlamaya benzemektedir ancak burada 2. Ebeveyn'deki silinmemiş genlerin lokusu önem taşır. Bir önceki örnek sıra tabanlı çaprazlama ile çaprazlanacak olursa [18]:



Şekil 3.14: Sıra Tabanlı Çaprazlamazlama

- Dairesel Çaprazlama (Cycle Crossover - CX): İlk kez Oliver, Smith ve Holland tarafından önerilmiştir. İlk ebeveynden başlar ve bu ebeveyndeki genlerin diğer ebeveyndeki karşılığını bularak bir döngü yaratır. Adımları şu şekildedir [18]:

Adım 1: İlk ebeveynin ilk geninden başlayarak genler arasındaki ilişkinin döngüsü kurulur.

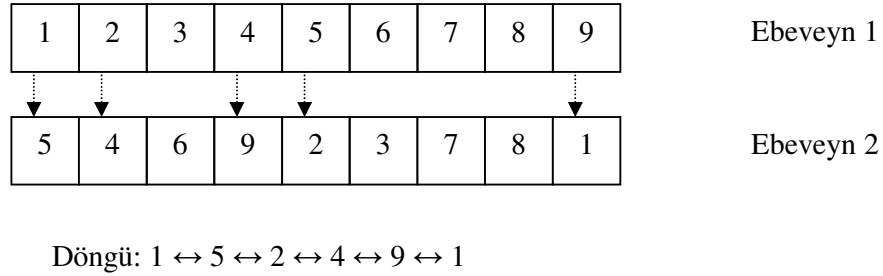
Adım 2: Döngü içindeki genler Ebeveyn 1'den alınarak yarı-çocukta aynı lokusa yerleştirilir.

Adım 3: Döngüdeki genler Ebeveyn 2'den silinir.

Adım 4: Ebeveyn 2'deki kalan genler sırayla çocuğa yerleştirilir.

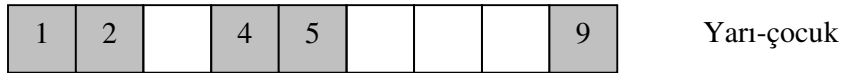
Dairesel çaprazlama ile ilgili bir örnek aşağıda yer almaktadır [18]:

Adım 1: Döngünün kurulması



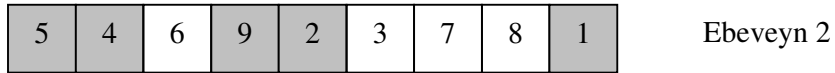
Şekil 3.15: Dairesel Çaprazlamada Döngünün Kurulması

Adım 2: Yarı-çocuğa döngüdeki genlerin yerleştirilmesi



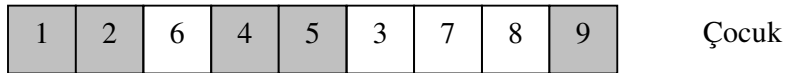
Şekil 3.16: Yarı-çocuğun oluşturulması

Adım 3: Ebeveyn 2'den döngüdeki genlerin silinmesi



Şekil 3.17: Ebeveyn 2'den genlerin silinmesi

Adım 4: Ebeveyn 2'deki genlerin çocuğa aktarılması



Şekil 3.18: Ebeveyn 2'den çocuğun oluşturulması

- Sezgisel Çaprazlama: İkili kodlanmış populasyonlarda, genlerin bir kısmını bir ebeveynden, kalan kısmını ikinci bireyden almayı gerektirir. Formülü şu şekildedir

$$\text{Çocuk} = \text{Ebeveyn 1} * \alpha + (1 - \alpha) * \text{Ebeveyn 2}$$

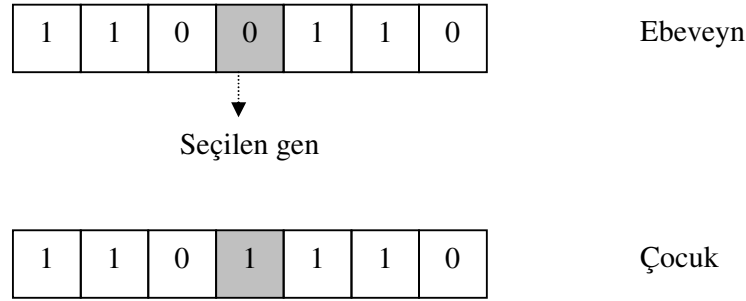
α değeri [0,1] aralığında alınır [5].

Adım 6. Dönüşüm (Mutasyon): Dönüşüm genetik algoritma operatörlerinden biridir. Bu operatör nesiller boyunca kaybolan bir allel varsa onun geri getirilmesinde

kullanılır. Dönüşüm operatörü bir geni alarak allellerinden biriyle değiştirir [14]. Dönüşüm operatörünün en önemli özelliği dönüşüm olasılığı (p_m)dır. Dönüşüm olasılığı bir genin dönüşüme uğrama olasılığıdır. Bir populasyon içinde dönüşümün beklenen değeri dönüşüm olasılığının o populasyondaki gen sayısı ile çarpımına eşittir [9]. Dönüşüm olasılığı genellikle 0.001 ile 0.01 arasındadır. Bu hız genlerin kaybolma hızına bağlıdır [5].

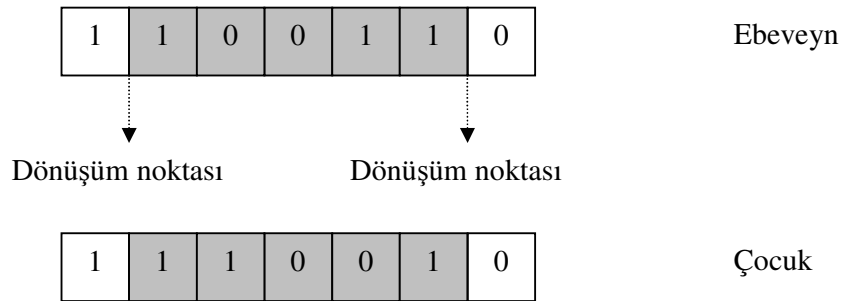
3.2.4.4. Genetik Algoritmada Kullanılan Dönüşüm Yöntemleri

- Basit Dönüşüm: Kromozomdan herhangi bir gen seçilerek alleliyle değiştirilir [5].



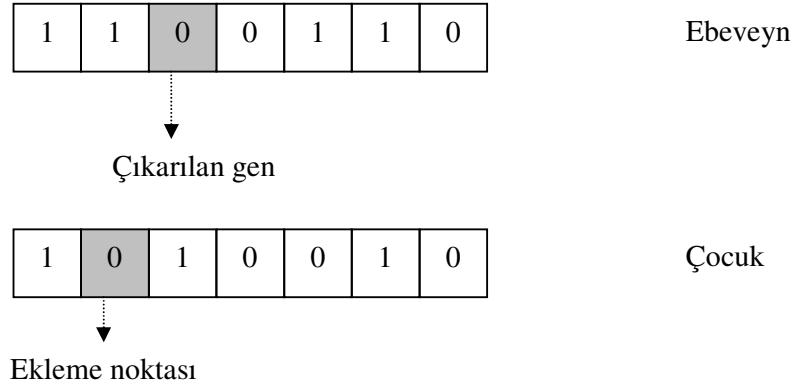
Şekil 3.19: Basit Dönüşüm

- Değiştirme Dönüşümü: Kromozom içinde iki dönüşüm noktası seçilerek, no noktalar arasında kalan genler ters çevrilir [10].



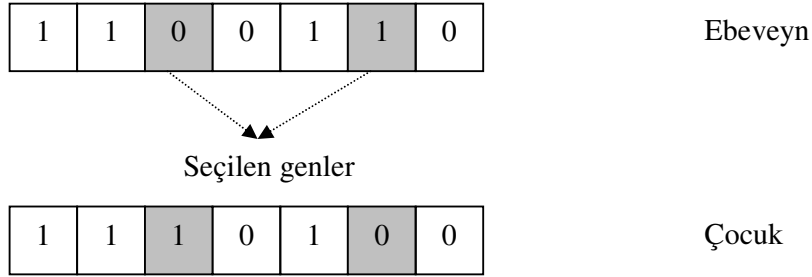
Şekil 3.20: Değiştirme Dönüşümü

- Ekleme Dönüşümü: Kromozomdan rassal bir gen seçilir. Seçilen gen rassal bir noktaya eklenir [10].



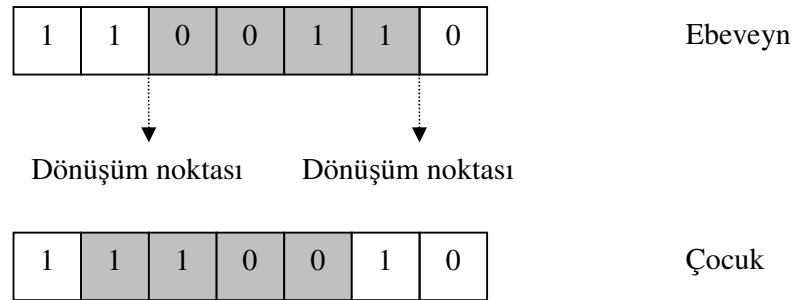
Şekil 3.21: Ekleme Dönüşümü

- Karşılıklı Değişim Dönüşümü: Kromozomdan rassal iki gen seçilerek yerleri değiştirilir [18].



Şekil 3.22: Karşılıklı Değişim Dönüşümü

- Yer Değiştirme Dönüşümü: Değiştirme Dönüşümünün özel bir durumudur. değiştirme dönüşümünde olduğu gibi iki nokta arasında kalan genleri ters çevirir. Yalnız ters çevrilen genleri rassal bir pozisyona ekler [18].



Şekil 3.23: Yer Değiştirme Dönüşümü

Adım 7. Bitiş Ölçütlerine Ulaşılması: Genetik Algoritma'nın döngülerinin bitirilmesi (nesillerin sonlandırılması) için gerekli ölçütler farklı tanımlanabilir. Bazı ölçütler şunlardır:

1. Belli bir nesil sayısı bitirildikten sonra döngüler durdurulur ve nesiller boyunca kimin en çok uygunluk gösterdiğine bakılır [5].
2. Evrim sonucunda uygunluk fonksiyonundaki iyileşme belli bir değerin altına düştüğünde döngüler bitirilir [5].
3. Evrim için belirli bir süre verilerek, bu süre bittiğinde evrim durdurulabilir [9].
4. Uygunluk fonksiyonu için belirli bir değer verilerek, bu değeri geçen ilk döngüde evrim bitirilebilir [5].

Adım 8. En İyi Sonucun Gösterilmesi: Bitiş ölçütlerine ulaşıldıktan sonra, en iyi değer en iyi çözüm kabul edilerek bu sonuç problemin çözümü olarak sunulur.

3.2.5. Genetik Algoritma ile İlgili Bir Örnek

[0,31] aralığında $f(x) = x^2$ fonksiyonunu GA kullanarak enbüyüklemek isteyelim.

Adım 1. Genetik Kodlama: GA kodları 0-1 (ikilik) düzeninde yazılan bireylerdir. Bu durumda 0 değeri 00000 ile 31 değeri ise 11111 ile gösterilir.

Adım 2. Amacın Tanımlanması: Amaç $f(x) = x^2$ fonksiyonunun en büyüklenmesidir.

Adım 3. Başlangıç Populasyonunun Yaratılması: GA'da başlangıç populasyonu sayısı ve başlangıç populasyon değerleri rassal olarak belirlenir. Bu örnek için populasyondaki bireylerin sayısı $n = 4$ olarak, bireylerin değerleri ise yazı-tura atışları ile belirlensin. Yazı 0 değerini, tura 1 değerini versin. Her bireydeki 5 basamak için 1'er yazı-tura atışı atılacağından toplamda $5 * 4 = 20$ atış yapılacaktır. 20 atışın ardından başlangıç populasyonunun aşağıda verilen şekilde oluştuğunu varsayalım.

01101

11000

01000

10011

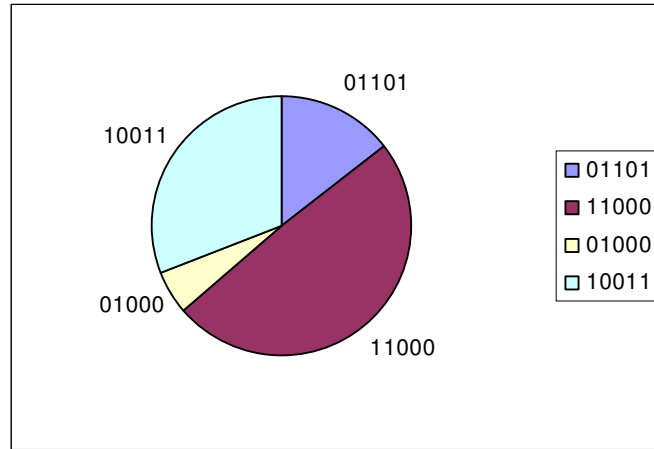
Adım 4. Uygunluğun Hesaplanması: Uygunluk değeri bireylerin amaç fonksiyonundaki değeridir. Aşağıdaki tabloda başlangıç populasyonunun uygunluk değerleri verilmiştir.

Tablo 3.3: Örnek soru – Uygunluğun Hesaplanması

Birey No.	Birey	Değeri	Uygunluk	Toplam İçindeki %
1	01101	13	169	14.4
2	11000	24	576	49.2
3	01000	8	64	5.5
4	10011	19	361	30.9
Toplam			1170	100.0

Adım 5. Seçim /Çaprazlama / Dönüşüm

- Seçim: Seleksiyon işlemi her aşamada gerçekleştirilen bir işlemdir. Bu işlem için öncelikle uygunluk değerleri ele alınır. Bu değerleri için bir rulet çarkı oluşturulur. Rulet çarkında bireyler toplam içindeki yüzdeleri ile orantılı olarak bölünür ve çark populasyon sayısı kadar çevrilir. Belli bir noktada gelen değerler yeni populasyonları diğer bir deyişle ebeveynleri oluşturur. Bu örnekteki değerler için rulet çarkı aşağıda verilmiştir.



Şekil 3.24: Rulet Çarkı

Rulet arkında gelen deęerler ve yeni populasyon iin oluřturulan tablo ařaęıdaki gibidir.

Tablo 3.4: Rulet arkı Deęerleri ve Yeni Tablo

Birey No	Bařlangı Populasyonu	Deęer $f(x)$	Uygunluk	Yüzde $\frac{f_i}{\sum f}$	Beklenen Geliř Sayısı $\frac{f_i}{f}$	Rulette Geliř Sayısı
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
Toplam			1170	1.00	4.00	4.0
Ortalama			293	0.25	1.00	1.0
Maksimum			576	0.49	1.97	2.0

Tabloda uygunluk deęerleri ve uygunluk yüzdelerinin yanı sıra deęerlerin beklenen deęerleri ve rulet arkında ka kez geldikleri aıklanmıřtır. Buna göre ortalama uygunluk deęeri 293'tür ve gelen en büyük deęer 576'dır. Bu deęerler algoritmayı bitirme ölçütü olarak da kullanılabilir. Örneęin, nesiller arası ortalama uygunluk deęeri %5'in altına düřüldüğünde algoritma bitirilip, en büyük deęer döndürülebilir. Buna benzer olarak, en büyük deęerler arasında belirli bir artış oranı da belirlenebilir. Rulet arkı ile yapılan seim ařamasından sonra aprazlama ve dönüşüm işlemlerine geilir.

- aprazlama ve Dönüşüm (Mutasyon): aprazlama ve dönüşüm işlemlerini yapabilmek iin öncelikle bu işlemlerin gerekleřme olasılıklarının belirlenmiř olması gereklidir. Bu olasılıkların aprazlama iin 0.6 ile 1.0, dönüşüm iin 0.001 ile 0.01 arasında deęiřtięi belirtilmiřti. Bu örnek iin aprazlama olasılığı 1.0 ve dönüşüm olasılığı 0.001 olsun. Bu demektir ki her döngüde bir aprazlama ve 1000 döngüde bir dönüşüm gerekleřecektir. Bu döngüde aprazlama gerekleřecek ve dönüşüm gerekleřmeyecek olsun. aprazlama iin eřlerin belirlenmesi ve bu eřlerin

hangi genden itibaren aprazlanacađı belirlenmelidir. Bunun iin rassal sayısı atışı yapılacaktır.

İlk olarak eřlerin belirlenmesi gerekmektedir. 1 numaralı bireye eř bulmak iin 2-3-4 rakamlarından rassal olarak bir seim yapalım. Gelen rakam 2 olduđunda 1 no’lu bireyin eři 2 no’lu birey olacaktır. Bu durumda 3 ve 4 no’lu bireyler de eřlerdir. řimdi de aprazlama noktası belirlenmelidir. Bu durumda 1 ve 2 no’lu bireylerin aprazlanması iin 4 nokta bulunmaktadır. Bu noktalar ařađıdaki řekilde izgilerle gsterilmiřtir.

1. No’lu Birey	0 1 1 1 0 1
2. No’lu Birey	1 1 0 0 0 0

řekil 3.25: aprazlama Operatr

rnekte grldđ gibi 1-2 No’lu bireyler ve 3-4 No’lu bireyler iin 1-4 arası rassal sayısı atışı yapılması gerekmektedir. aprazlama iřlemi ve sonucu bir sonraki sayfadaki tabloda verilmiřtir:

Tablo 3.5: Yeni populasyon

Seleksiyon Sonrası Oluşan Bireyler	Eşler	Çaprazlama Noktası	Yeni Populasyon	Değeri	Uygunluk $f(x) = x^2$
011011	2	4	01100	12	144
110010	1	4	11001	25	625
111000	4	2	11011	27	729
101011	3	2	10000	16	256
Toplam					1754
Ortalama					439
Maksimum					729

Görüldüğü gibi ilk döngüde çaprazlama sonrası toplam uygunluk değeri 1170'ten 1754'e, ortalama uygunluk değeri 293'ten 439'a ve maksimum değer 576'dan 729'a çıkmıştır.

Bu adımlara belirlenen ölçütlerin sağlanmasına kadar devam edilir ve en iyi sonuç gösterilir [4].

3.2.6. Plan (Katar - Schemata), Plan Teoremi ve Yapı Taşı Teoremi

Bireyleri ikili kodlama yoluyla oluşturulmuş bir GA problemi olsun. Bu durumda çözüm uzayı 3 boyutlu bir uzaydır, bu uzay aşağıda verilen bir köşesi başlangıç noktasına (0,0,0) oturtulmuş bir birim küptür ve çözümler sadece köşe noktalarında aranır.

Arama uzayı daraltılmak, örneğin bu küpün sadece ön yüzünde arama yapılmak istensin. Bu köşeleri 000, 001, 010 ve 011 noktalarında bulunan küp oluşur. Bu köşe noktalarının ortak noktası 0 ile başlıyor olmalarıdır. GA'da bu noktalar 0** şeklinde gösterilir. Buradaki * işareti o genlerin 0 ya da 1 olabileceğini gösterir. İçinde * bulunan ifadelere plan (katar) denir. Yukarıda belirtilen çözüm uzayı için 3 gen de 0, 1 ya da * değerlerinden birini alabilir. Bu yüzden çözüm ararken $3^3 = 27$ ifadeyi taramamız gerekir. Buna benzer olarak L uzunluğundaki bir kromozomun arama uzayı 3^L tanedir denebilir [5].

Farklı kodlama tiplerinde ise bu formül $(k + 1)^L$ olarak verilir. k kodlamadaki karakter sayısını belirtir [11].

Bir s planının düzeni $o(s)$ ile gösterilir ve planda ‘*’ bulunmayan gen sayısını gösterir. Planın tanımlanmış uzunluğu ise ilk ve son ‘*’ içermeyen genlerin lokusları arasındaki farktır ($\delta(s)$). Planın uygunluğu $f(s)$, o plandaki tüm kromozomların ortalama uygunluk değeridir.

Plan Teoremi, yüksek uygunluk değerine sahip olan planların nesiller boyunca üstel olarak artacağını, düşük uygunluk değerine sahip olanların sayısının ise nesiller boyunca üstel olarak azalacağını belirtir. Yüksek uygunluk değerine sahip bir plan için şu eşitlik geçerlidir [8]. Burada $\bar{f}$ planın uygunluğudur.

$$M(s, t + 1) \geq \frac{M(s, t) \cdot f(s)}{\bar{f}} \cdot \left[1 - p_c \cdot \frac{\delta(s)}{l - 1} - p_m \cdot o(s) \right] \quad (3.3)$$

Burada $M(s, t + 1)$ t+1. nesilde s planının sayısıdır.

Bu teorem Goldberg tarafından öne sürülmüştür. GA’nın, yapı taşı adı verilen kısa, düzeni az, performansı yüksek planların en iyi çözüme yakın performans gösteren noktalarında sonuç arayacağını belirtir.

3.2.7. Genetik Algoritma ile Diğer En iyileme Yöntemlerinin Karşılaştırılması

Goldberg Genetik Algoritma’nın diğer en iyileme yöntemlerinden aşağıdaki gibi ayrıldığını belirtmiştir[8]:

1. GA parametrelerin aşıyla değil, kodlanmış parametre kümeleriyle çalışır.
2. GA birden çok noktayı aynı anda araştırır, noktaları tek tek araştırmaz. Bir grup çözümü aynı anda inceler.
3. Sadece amaç fonksiyonu değerini kullanır, türev ya da benzer başka bilgiye ihtiyaç duymaz.
4. Olasılıkları hesaba katar, deterministik değildir.

Tüm bunlara ek olarak

1. GA problemin çözüm yöntemiyle ilgili bir bilgi taşımaz. Bu nedenle kör arama yöntemi olarak tanımlanır [7].

2. GA tek nokta inceleme yapmadığı gibi, noktalardan farklı yönleere de gidebilir. Buna paralelizim adı verilir.
3. Paralelizmi kullanarak uygun olmayan ya da sonucu kötüleştiren noktaları eleme yeteneği kazandırır.
4. GA yerel en iyi çözümlere takılmaz. Algoritma sonlandırılmadığı sürece daha iyi noktaları ararlar.
5. GA çok amaçlı olarak da kullanılabilir. Birden çok parametreyi formalize etmek mümkündür. Parametrelerde aramalar yine paralelizm ile yapılır.
6. GA stokastik nedeniyle rassal bir yöntemdir. Parametrelerle ilgili seçim yapma aşamaları ve kodlama aşamasında dikkatli olunmalıdır. Bu özellik GA'nın dezavantajlı yanıdır [9].

4. ATAMA PROBLEMLERİ VE GENETİK ALGORİTMA

Atama problemleri literatürde geniş yer tutmaktadır. Bu problemler, matematiksel programlama gibi diğer tipteki problemlerin çözümü için kullanılabilecek yöntemler dışında, dengeli evlilik problemi gibi çeşitli özel geliştirilmiş algoritma ve sezgisellerle de çözülmektedir.

Atama problemleri için genel olarak kullanılan yöntem matematiksel programlamadır. Sonia ve Puri, doğrusal programlamayı birincil ve ikincil işlere kaynak atamak için kullanmışlardır [19]. Bunun yanı sıra programlama ile birlikte çalışan sezgisel yöntemler geliştirenler de bulunmaktadır. French ve Wilson [20] ise, genel atama problemi için doğrusal programlama ile birlikte uygulanan bir sezgisel geliştirmişlerdir. Benzer bir çalışma Przybylski ve diğ. tarafından yapılmıştır[21]. Przybylski ve diğ. iki amaçlı doğrusal olmayan programlama modelinin üstüne bir sezgisel geliştirmişlerdir. Yine doğrusal programlamaya sezgisel eklenerek yapılan bir çalışma Zhang ve Ong'a aittir [22].Zhang ve Ong, iki amaçlı doğrusal programlama modeli üzerine üzerine sezgisel bir model geliştirerek, bu sezgiselin özellikle sırt çantası probleminde yararlı olduğunu vurgulamışlardır.

Ek olarak, tam sayılı programlama modelleri de çoğunlukta olduğu görülmektedir: Bu tür çalışmalardan birini yapan Yakoob ve Sherali [23], saf bir tamsayılı programlama modeli ile Kuveyt Üniversitesi'nde sınıfların, derslere ve hocalara atanması problemini çözerken, Zaghal ve Minoux [24], Tunus Havayollarında personelinin uçaklara atanması konusundaki çalışmalarını hem tamsayı programlama modelleri ile çözebilmiş hem de kendi geliştirdikleri sezgiselle problemi çözerek karşılaştırmıştır. Tamsayılı programlama modellerinin üstüne çeşitli sezgisel ve algoritmalarla geliştirilmiştir ya da başka modellere geçiş yapılmıştır. Bu algoritmalarından bazıları minimum maliyet akışı [25], kısıt programlama [26], dal-sınır algoritmasıdır [27].

Genetik Algoritma ve benzeri meta-sezgiseller atama problemlerinde sıkça kullanılmaktadır: Toroslu ve Arslanoğlu [28], işlerin personele atanmasında Genetik

Algoritma kullanmıştır ki bu çalışma İSO-İTÜ şirket/öğrenci atama problemine yakın bir problemdir. Farklı olarak, Toroslu ve Arslanoğlu çalışmalarında hiyerarşik bir atama yaklaşımını çelişen 3 farklı amaç fonksiyonunda kullanmış, çalışmayı grafiksel metotlarla desteklemiştir. Literatürde sıkça kaynak atama çalışmalarına rastlanmaktadır [25, 26, 29, 30]. Bu çalışmaların birinde tamsayı programlama ve minimum maliyet akışı problemi kullanılırken [25], bir diğerinde kısıt programlama kullanılmıştır. Wang ve Lin [29] sipariş üstüne çalışan bir fabrikada kaynakların atanmasında sıralama ve düzenlemeyi Genetik Algoritma ile kaynakların siparişe göre dağılımında Bulanık Mantık kullanmışlardır. Genetik Algoritma tüm atama problemlerinde verimli kullanılabilen yöntemlerden biridir. Purnaprajna ve diğ.[30], yazılım ve donanımı en hızlı ve en az zamanda işleyebilecek bir şekilde sisteme atamak üzere Genetik Algoritma kullanmışlardır. Lin ve Mitsuo [31] insan kaynakları atama problemini Genetik Algoritma ile çözmüşlerdir. Wang tarafından yapılan bir çalışmada [32], Uzak Doğu Üniversitesi'nde derslere öğretmenlerin atanması, ders saatleri, öğretim üyelerinin uzmanlık alanları gibi kısıtlarla birlikte Bilgi Tabanlı Genetik Algoritma ile çözülmüştür. Genetik algoritma ile bilgisayarlara görev atama problemini inceleyen 2 çalışma da Lin ve diğ. [33], Salgado-Sanchez ve diğ. [34], tarafından yapılmıştır. Lin ve diğ. [33], Genetik Algoritmalar, ağ modellerinde kullanarak atama ve çalışma maliyetini en küçüklerken, Salgado-Sanchez ve diğ.[34], bu maliyet en küçüklemesinde GA'nın yanı sıra Yapay Sinir Ağları'nı da kullanmaktadır. Bunlardan farklı olarak, Majumdar ve Bhunia [35], atama problemlerinde Elitist Genetik Algoritma yöntemini kullanmışlardır.

Atama problemleri için özel geliştirilmiş algoritmalar da vardır. Macar Yöntemi bu algoritmalarından biridir. Macar yöntemi, atama matrisinin oluşturularak, atamamanın maliyetini en küçüklemeyi hedefler[36]. Bu algoritmalarından diğeri Bozulmayan Evlilik Problemi (Gale-Shapley Algoritması / Stable Marriage Problem) vb. algoritmalar. Bozulmayan Evlilik Problemi, n adet iki tür bireyin (n kadın ve n erkek birey) tercih sıralarına göre eşleştirilmesinde kullanılır. Bu eşleşme öyle bir eşleşme olmalıdır ki, erkek birey için tercih düzeyi daha yüksek bir kadın birey olmamalı, kadın birey için de tercih düzeyi daha yüksek bir erkek birey olmamalıdır[37-39]. Bunun yanı sıra Bozulmayan Evlilik Problemi'nin birkaç özel türü bulunmaktadır. Bunlardan biri Ağırlıklı Eşleşme Problemidir. Burada amaç bir bireye, birey üzerinde maksimum ağırlığı olacak bireyin atanmasıdır [40]. Diğer bir

özelleştirilmiş algoritma Uygun Ev Arkadaşı Problemi'dir. Bu problem Uygun Evlilik Problemine benzer, farkı ise iki tip birey yerine bir seçim havuzunda tek tip bireylerin tercihlerini bulundurmasıdır [39,41]. Yine özel bir tip problem de Hastane – Yatılı Hasta Problemidir (Hospitals/Residents Problem). Bu problemin Uygun Evlilik Probleminden farkı birinci tip bireyin ikinci tip bireyden birden fazla tercih edebilme özelliğinin olmasıdır [42].

Atama problemlerinin özel bir türü olan proje atamaları da literatürde geniş yer tutmaktadır. Bu çalışmalardan biri Ar-Ge çalışmalarını değerlendirmek üzere projelere müfettiş atama için yapılmıştır[43]. Bu çalışma karar destek sistemleri ile yapılmışsa da proje atama problemlerin çoğu tabu araştırması [44] ve GA ile çözülmüştür Homg ve Cochran [45], JIT hücrelerinde işçileri birden çok işe sahip olacak şekilde atarken, Bassett [46], Ar-Ge projelerinde çalışacak olan çalışanların atamak üzerine çalışmaktadır. Benzer olarak çalışma gruplarında çeşitliliği artırıcı bir eleman atama algoritması Bhadury ve diğ. [47], tarafından küme parçalama algoritması ile geliştirilmiştir. Benzer bir çalışma Patanakul ve Milosevic [48] tarafından projelerin proje müdürlerine atanmasını işlemiştir. Bu çalışmalar, GA dışındaki metotları kullanmış olsa da GA ile çözülen problemlerin de çoğunlukta işe insan gücü atama problemleridir ki bu problemler, tezin konusu olan İSO-İTÜ şirket/öğrenci atama problemine benzemektedir. Örneğin, Harper ve diğ. [49], öğrencilere proje dağıtımını Genetik Algoritma ile yaparken, Garrett ve diğ. [38], Amerikan Donanması'ndaki gemilere denizci atama problemlerini GA ile çözmüştür. Tüm bu problemler genellikle kısıtsız problemlerken, Garrett ve diğ. çalışmasında bir denizciye en fazla bir gemi atanması kısıdı bulunmaktadır. İSO-İTÜ şirket/öğrenci atama probleminin diğer çalışmalardan farkı olurlu olmayan eşleşmelerin olmasıdır.

Bir sonraki bölümde İSO-İTÜ şirket/öğrenci eşleştirme problemi GA ile çözülecektir. Bunun nedenlerinden biri çözüm uzayının çok geniş olmasıdır. GA çözüme birden fazla çözümle başladığı ve genel bir arama yaptığından ötürü bu problemin çözümü için uygundur. Başka bir neden ise, olası en iyi eşleşmelerden bir kaçının sunulmak istenmesidir. Bu da iki uygunluk fonksiyonun karşılaştırılması ile sağlanmıştır.

5. İSO – İTÜ ŞİRKET/ÖĞRENCİ ATAMA PROBLEMİ

5.1. İSO-İTÜ Şirket/Öğrenci Eşleştirme Problemi

İTÜ Endüstri Mühendisliği Bölümü son sınıf öğrencileri ilk dönem Endüstri Mühendisliği Projesi dersi için bitirme çalışmaları için ön bir araştırma yapmaktadırlar. İkinci dönem ise, ilk dönem yaptıkları çalışma ile ilgili bir uygulama alanı bulmakta ve yöntemlerini bu alandaki sorunları çözmek için kullanmaktadırlar. Bunun için de uygulama yapabilmek için şirketler aramaktadırlar. Ayrıca uygulama yaptıkları bu şirketlerde geleceğe yönelik kariyer planları da yapabilmektedirler.

İSO' ya bağlı şirketler ise kendilerine yardımcı olmaları ve şirket bünyelerine yetiştirmeleri için son sınıf öğrencileri ile birlikte çalışmak istemektedirler. Hem öğrencilerin hem de şirketlerin bu talepleri doğrultusunda Üniversite-Sanayi işbirliği kapsamında, İTÜ son sınıf öğrencilerinin, İSO'ya bağlı şirketlerde bitirme çalışmaları uygulamaları yapmaları konusunda anlaşmaya varılmıştır. Problem şirkete atanacak öğrencilerin belirlenmesi üzerinedir.

Eşleşmenin yapılabilmesi için öncelikle 9 ana çalışma konusu belirlenmiştir. Bu çalışma konuları aşağıdaki gibidir:

- Stok ve Malzeme Yönetimi
- Süreç Yönetimi
- Kalite Yönetimi
- İnsan Kaynakları Yönetimi
- Ergonomi
- Muhasebe / Yatırım Analizleri
- Tedarik Zinciri Yönetimi / Müşteri İlişkileri Yönetimi
- Bilgi Yönetimi
- Stratejik Yönetim

Şirketler bu çalışma başlıklarından ihtiyaç duydukları alanları tercih sıralarına göre seçmektedirler. Şirketlerin, tercih sıraları 9 boyutlu bir vektörde gösterilir. Bu gösterimde tercih numaraları, ilgili çalışma başlığının altında yer alır. Örnek gösterimler aşağıdaki gibidir:

1. Şirket A, sadece Kalite Yönetimi çalışan bir öğrenci arıyor olsun. Kalite Yönetimi, üçüncü çalışma konusu olduğundan ve şirketin de birinci ve tek tercihi olduğundan şu şekilde gösterilir:

0	0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Şekil 5.1: Şirket A'nın tercih vektörü

2. Şirket B, sırasıyla Stok Yönetimi, Bilgi Yönetimi, Süreç Yönetimi çalışan bir öğrenci istemiş olsun. Bu durumda birinci konu başlığı olan Stok Yönetimi'nin hanesine 1, sekizinci konu başlığı olan Bilgi Yönetimi'nin hanesine 2, ikinci konu başlığı olan Süreç Yönetimi'nin hanesine de 3 rakamı girilecektir.

1	3	0	0	0	0	0	2	0
---	---	---	---	---	---	---	---	---

Şekil 5.2: Şirket B'nin tercih vektörü

3. Şirket C için tercih sırası önemli olmasın. Bu durumda, Şirket C'nin tüm tercihlerine birinci tercih gözüyle bakılır.

1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

Şekil 5.3: Şirket C'nin tercih vektörü

Öğrenci tercihleri ise seçim yoluyla yapılmamaktadır. Tercihler oluşturulurken, ilk dönem aldıkları 1 kredili “Endüstri Mühendisliği Projesi” dersi çalışmaları göz önünde bulundurulmaktadır. 9 boyutlu tercih vektörü, öğrencilerin yaptıkları çalışmalar tespit edilerek, hangi konu başlığına kaçınıcı dereceden girdiğinin tespiti ile yapılmaktadır. Örnek öğrenci tercihi vektörleri aşağıdaki gibidir:

1. Öğrenci A'nın Endüstri Mühendisliği Projesi, “Tedarik Zinciri Yönetimi” konusunda olsun. Bu durumda, öğrenci birinci dereceden dokuzuncu konu başlığını çalışıyor demektir. Öğrencinin tercih vektörü aşağıdaki gibidir:

0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---

Şekil 5.4: Öğrenci A'nın Tercih Vektörü

2. Öğrenci B'nin Endüstri Mühendisliği Projesi, “Stok Yönetiminde Veri Madenciliği” üzerine olsun. Bu durumda, öğrenci hem Stok Yönetimi hem de Bilgi Yönetimi konusuna giren Veri Madenciliği'ni birinci dereceden bilmek durumundadır. Öğrencinin tercih vektörü aşağıdaki gibidir:

1	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---

Şekil 5.5: Öğrenci B'nin Tercih Vektörü

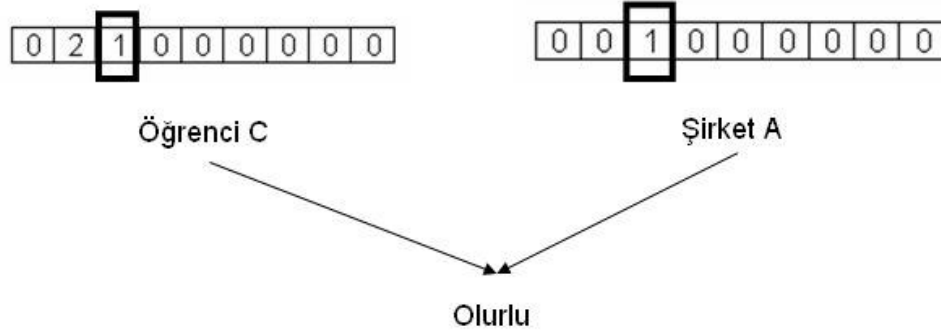
3. Öğrenci C'nin Endüstri Mühendisliği Projesi “Kalite Yönetiminde Süreç İyileştirme” üzerine olsun. Bu durumda, öğrenci öncelikli olarak Kalite Yönetimi'ni bilmek, yardımcı konu olarak ikinci derecede Süreç Yönetimi'ni bilmek zorundadır. Bu öğrencinin tercih vektörü aşağıda verilmiştir:

0	2	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Şekil 5.6: Öğrenci C'nin Tercih Vektörü

Çalışmada kullanılan şirket listesi Ek A'da, öğrenci tercihleri tablosu ise Ek B'de ve şirketlerin tercih tablosu Ek C'de verilmektedir.

Problemde bir öğrencinin bir şirkete atanabilmesi için olurlu bir eşleşme olması gerekmektedir. Olurlu eşleşme ise ancak, şirketin herhangi bir tercihi öğrencinin herhangi bir derecedeki çalışma alanına uyuyorsa gerçekleşir. Örnek bir olurlu eşleşme aşağıda verilmiştir.



Şekil 5.7: Olurlu Bir Eşleşme

Eşleşme için çalışma alanlarının derecesi önemli değildir, tercih alanında boş olmayan iki hücrenin çakışması olurlu bir eşleşme için yeterlidir. Problem Ek C'da verilen İSO'ya bağlı 54 şirketlerle, Ek B'de tercihleri verilen öğrencileri 84 öğrenciyi

eşleştirme sorunudur. Çözümün olması için her bir şirketin olurlu bir öğrenci eşleşmesi yapması gerekmektedir. Başka bir deyişle, herhangi bir şirkete, o şirketin tercihlerinde çalışmak istemeyen bir öğrenci atanırsa, o çözüm olurlu olmayan bir çözümdür.

İSO-İTÜ şirket – öğrenci eşleştirme problemi 45 şirket ve 84 öğrenciyi eşleştirme problemidir. Problem matematiksel programlama ile çözülmesi zor bir problemidir. Bunun nedeni amaç fonksiyonundaki belirsizliktir. Amaç fonksiyonu, tercihler bazında yapılacaktır ancak herhangi bir amaç fonksiyonu değişik bir sonuca yönelebilir. Ayrıca, problemin matematiksel programlama ile çözülememesinin diğer bir nedeni de değişken sayısının çokluğudur. Çok sayıda değişken ile problemin çözümünde yazılımlar da yetersiz kalmakta, ayrıca işlem süresi çok uzun sürmektedir. Problemin çözümünde Genetik Algoritma kullanılacaktır. Problem çok sayıda çözümünden başlayıp yine çok sayıda çözüm üretecek ve nesiller boyunca elde edilen en iyi sonuç alınacaktır.

5.2. Problemin Genetik Algoritma İle Çözümü

Problemin Genetik Algoritma ile çözülmesi, daha önceden anlatılan algoritma adımlarının probleme uyarlanması ile yapılacaktır.

5.2.1. Genetik Kodlama

Problem için en uygun kodlama, kromozomdaki genlerin lokusunun şirket numarasını ve genlerin de bu şirketlere atanacak öğrenci sayısını göstermesidir. i . şirkete atanacak öğrenci numarası, i . lokustaki gendir. i . şirkete atanacak öğrenci kromozomda i . lokustadır. i . şirkete atanan öğrenci, ilgili gendeki öğrenci numarasıdır. Şirkete atanacak öğrenci 1 – 83 arasında rassal sayı atılmasıyla belirlenir. Kromozomları oluşturan 2 önemli kısıt bulunmaktadır:

1. Bir kromozomda aynı sayının tekrar etmemelidir. Bu durum, bir öğrencinin birden fazla şirkete atandığını belirtir. Bu olurlu bir çözüm değildir.
2. Şirket ve o şirkete atanacak öğrencinin ortak çalışma konusu olmasıdır. Bu eşleşmenin olurlu olması anlamına gelmektedir. Ancak, her şirket için olurlu eşleşme olursa çözüm olurlu bir çözüm olarak kabul edilebilir.

Örnek bir kromozom yapısı aşağıda verilmiştir:

1	2	3		53	54
25	16	74		67	8

Şekil 5.8: Kromozom Yapısı

Kromozomdaki gen sayısının şirket sayısından fazla olmasının nedeni, birden çok öğrenci isteyen şirketlerin tekrar etmesidir. Şirket ne kadar öğrenci istiyorsa, kromozomda da o kadar gene sahip olacaktır. n öğrenci isteyen bir firma, kodlama sırasında aynı tercih vektörüne sahip n adet gene dönüşür. Şirketlerin kromozomda belirttiği gen numaraları Ek C’de verilmiştir.

5.2.2. Amaç Fonksiyonunun Tanımlanması

Problemde amaç fonksiyonları belirsizdir. Bu nedenle karşılaştırma çeşitli amaç fonksiyonlarının karşılaştırılması üzerine yapılacaktır. 2 tipte amaç fonksiyonu karşılaştırılıp hangisinin daha iyi sonuçlar verdiği incelenecektir. Bu 2 amaç fonksiyonu aşağıda verilmiştir:

1. Minimum Kareler Yöntemi: Şirket – öğrenci tercihleri arasındaki sapmayı minimize etmeyi planlayan bir amaç fonksiyonudur. Öğrenci ile şirket tercihleri arasında bir eşleşme olduğunda, bu çakışan tercihin arasındaki farkın karelerini minimize etmeyi amaçlar. Örneğin Şekil 5.7’deki eşleşmeyi ele aldığımızda çakışma Kalite Yönetimi’ndedir ve öğrenci – şirket arasındaki tercih farkı (Şirket tercihi – öğrenci tercihi)² = (1-1)² = 0’dır.

2. Saaty Uygunluk İndeksi: Saaty Uygunluk İndeksi, asıl olarak Analitik Hiyerarşi Süreci / Analitik Ağ Süreci yöntemlerinde öncelik vektörlerinin birbirlerine yakınlığının ölçülmesinde kullanılır [50]. İki vektör arasındaki oran yakınlığı belirler. İki vektör aynı ise bu oran 1’dir [51].

İSO – İTÜ şirket/öğrenci eşleştirme probleminde Saaty Uygunluk İndeksi şu şekilde hesaplanır: İlk olarak şirket için olurlu olarak belirlenmiş öğrencinin tercih vektörü alınır. Bu vektörden $W_{ij} = W_i / W_j$ matrisi oluşturulur. Bu matris oluşturulurken 0 olan değerler belirsizlik yaratmaktadır. Bu belirsizliğin ortadan kaldırılabilmesi için 0 olan değerler 10 ile değiştirilmiştir. Bunun anlamı, seçilmek istenmeyen fazladan bir yapay değişken yaratılmasıdır. Örneğin, bir A öğrencisinin tercih vektörü $\vec{v} = [9 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0]^T$ olsun. Matrise geçişteki belirsizliğin ortadan

kaldırılması için öncelikle 0 olarak belirtilen tercihler 10 ile değiştirilir ve vektör $\vec{v}' = [9 \ 1 \ 10 \ 10 \ 10 \ 10 \ 10 \ 2 \ 10]^T$ durumunu alır. Vektörün matris formu aşağıda verilmiştir:

$$W_1 = \begin{bmatrix} 9/9 & 9/1 & 9/10 & 9/10 & 9/10 & 9/10 & 9/10 & 9/2 & 9/10 \\ 1/9 & 1/1 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 & 1/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \\ 2/9 & 2/1 & 2/10 & 2/10 & 2/10 & 2/10 & 2/10 & 2/2 & 2/10 \\ 10/9 & 10/1 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/2 & 10/10 \end{bmatrix}$$

$$W_1 = \begin{bmatrix} 1 & 9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 4.5 & 0.9 \\ 0.111 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.5 & 0.1 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \\ 0.222 & 2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0.2 \\ 1.111 & 10 & 1 & 1 & 1 & 1 & 1 & 5 & 1 \end{bmatrix}$$

İkinci aşamada öğrencinin eşleştiği şirket tercihi vektörü ele alınır. Bu tercih vektörü $W_{ij} = W_j / W_i$ matrisi oluşturulur. Bu matris oluşturulurken de 0 olan değerler belirsizlik yaratmaktadır. Bu belirsizliğin ortadan kaldırılabilmesi için 0 olan değerler 10 ile değiştirilmiştir. Örneğin A öğrencisinin eşleştiği bir A şirketinin tercih vektörü $\vec{v} = [1 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3 \ 0]^T$ olsun. Bu durumda 0 değerleri 10 no'lu tercih olarak değiştirilir. Yeni vektör $\vec{v} = [1 \ 2 \ 10 \ 10 \ 10 \ 10 \ 10 \ 3 \ 10]^T$ 'dür. Bu vektörün matris formu ise

$$W_2 = \begin{bmatrix} 1/1 & 2/1 & 10/1 & 10/1 & 10/1 & 10/1 & 10/1 & 3/1 & 10/1 \\ 1/2 & 2/2 & 10/2 & 10/2 & 10/2 & 10/2 & 10/2 & 3/2 & 10/2 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \\ 1/3 & 2/3 & 10/3 & 10/3 & 10/3 & 10/3 & 10/3 & 3/3 & 10/3 \\ 1/10 & 2/10 & 10/10 & 10/10 & 10/10 & 10/10 & 10/10 & 3/10 & 10/10 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 1 & 2 & 10 & 10 & 10 & 10 & 10 & 3 & 10 \\ 0.5 & 1 & 5 & 5 & 5 & 5 & 5 & 1.5 & 5 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \\ 0.333 & 0.667 & 3.333 & 3.333 & 3.333 & 3.333 & 3.333 & 1 & 0.333 \\ 0.1 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.3 & 1 \end{bmatrix}$$

İki matris elde edilince bu matrislere Hadamard Çarpımı uygulanır. Hadamard çarpımı $W_{ij} = W_{1ij} \cdot W_{2ij}$ ile elde edilir. Bu durumda W matrisi

$$W = \begin{bmatrix} 1 & 18 & 9 & 9 & 9 & 9 & 9 & 13.5 & 9 \\ 0.056 & 1 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.75 & 0.5 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \\ 0.074 & 1.333 & 0.667 & 0.667 & 0.667 & 0.667 & 0.667 & 1 & 0.667 \\ 0.111 & 2 & 1 & 1 & 1 & 1 & 1 & 1.5 & 1 \end{bmatrix}$$

Bu matrisin her satırı toplanır.

$$\sum W = [86.5 \quad 4.806 \quad 9.611 \quad 9.611 \quad 9.611 \quad 9.611 \quad 9.611 \quad 4.607 \quad 9.611]$$

Bu vektörün de tüm sayıları toplanır ve n^2 olan matris boyutuna bölünür.

$$\sum \sum W = 86.5 + 4.806 + 9.611 + 9.611 + 9.611 + 9.611 + 9.611 + 4.607 + 9.611 = 155.379$$

$$\frac{\sum \sum W}{n^2} = \frac{155.379}{81} = 1.918$$

Saatly Uygunluk İndeksi tüm şirketlerle eşleşen öğrenciler için hesaplandıktan sonra bütün uygunluk değerlerinin toplamı, çözümün amaç fonksiyonundaki uygunluk değerini verir.

5.2.3. Başlangıç Populasyonunun Oluşturulması

Problemin doğası gereği çözüm uzayı sınırlıdır. Tipik bir GA uygulamasında başlangıç populasyonu 100 – 1000 gibi değerlerle ifade edilir ancak bu uygulamada bu değerler kullanıldığında çözümlerin tekrar edeceği görülmektedir. Başlangıç populasyonu 20 olarak belirlenmiştir, evrim süreci boyunca tüm çözümleri kapsayabileceği düşünülmektedir.

5.2.4. Yeniden Üreme (Seçim) Operatörü

Yeniden üreme (seçim) operatörü olarak, en yaygın kullanılan rulet çarkı yöntemi seçilmiştir. Bu yöntem literatürde en çok rastlanan yöntemdir [8, 18].

5.2.5. Çaprazlama Operatörü

Literatürde, atama problemleri için özel bir çaprazlama yöntemi önerilmektedir [18, 38]. Bu yöntemle iki bireyden tek bir çocuk oluşturulmaktadır. Bu sonuç, populasyon sayısını azaltacağından ve 3 - 4 iterasyondan sonra bireyler tükeneceğinden kullanılmamıştır. Çaprazlama için, olurluluk sağlayan tek çaprazlama yöntemi olan, bir noktalı çaprazlama kullanılmıştır. Bu yöntemin ise bazı bireylerde öğrenci numaraları tekrarlamasına neden olarak, problemi olurlu olmayan çözüme götürdüğü tespit edilmiştir. Denemeler yapılarak bir noktalı çaprazlama üzerinde modifikasyonlar yapılmıştır ancak bu denemelerde olurlu olmayan çözümlerin çoğalmasıyla sonuçlanmıştır. Bu sorunun giderilebilmesi için olurlu olmayan çözüm düzeltilmediği durumlarda, bireyin öldürülmesine karar verilmiştir. Öldürülen bireyin yerine başlangıç populasyonundan rassal olarak başka bir birey seçilir. Böylece olurlu olmayan çözümler engellenerek, bir yandan da populasyondaki birey sayısının azalması önlenmiştir.

5.2.6. Dönüşüm (Mutasyon) Operatörü

Mutasyon için basit mutasyon yöntemi seçilmiştir. Bunun nedeni, diğer mutasyon yöntemlerinin olurlu olmayan çözüme yol açmasıdır.

Problemin çözümü için Excel Makroları kullanılmıştır. Her bir amaç fonksiyonu için program kodları Ek D ve Ek E’de verilmiştir.

5.3. Uygulama

Uygulama sırasında iki amaç fonksiyonu algoritma öncelikle 100 sonrasında 500 iterasyon çalıştırılmıştır. Ancak İTÜ öğrencileri Stratejik Yönetim ağırlıklı çalışırken, İSO’ya bağlı şirketler Stok/ Malzeme ve Süreç Yönetimi ağırlıklı çalışan öğrenciler istemektedirler. Süreç Yönetimi talebindeki açık azdır ancak Stok Yönetimindeki açık oldukça fazladır. Bu nedenle özgün verilerle olurlu bir çözüm bulunamamaktadır. Bu sorun ancak şu işlemle giderilmiştir: Öğrencilerin hepsi tercihlerine bakılmaksızın 9. tercih olarak Stok Yönetimi eklenmiştir. Bu durumda çözüm uzayı çok az sayıda çözüm içermektedir. Çözüm uzayını genişletmek için çaprazlama ve dönüşümde çeşitliliği artırarak daha iyi sonuçlar elde etmek için öğrencilerden sanayiye fazla gelen konularda çalışan öğrencilerin diğer konuları 9. tercih olarak seçme değişimine gidilmiştir. Bu şekilde uygunluk fonksiyonunun daha uygun sonuçlar verdiği gözlenmiştir.

Problemin çözümünde çıkan diğer bir sorun da Saaty Uygunluk İndeksi’nin kullanımı sırasında 0 tercihlerinin sorun yaratmasıdır. Bunun nedeni oluşturulan matrislerde sayı/0 belirsizliğine rastlanmasıdır. Bunun giderilmesi için 0 no’lu tercihler 10 no’lu istenilmeyen tercihlere dönüştürülmüştür.

Program her amaç fonksiyonunun 100 ve 500 iterasyonu için 5 defa çalıştırılmıştır. Uygulama sonuçları aşağıda verilmiştir.

5.4. Uygulama Sonuçları

5.4.1. Minimum Kareler Amaç Fonksiyonu Çözümü

Minimum kareler yöntemi için 100 iterasyonun verdiği sonuçlar aşağıdaki tabloda verilmiştir. Uygunluk değeri 108 olarak hesaplanmıştır.

Tablo 5.1: Minimum Kareler Yöntemi 100 iterasyon çözümü

§1	§2	§3	§4	§5	§6	§7	§8	§9
78	20	24	27	37	13	43	73	56
§10	§11	§12	§13	§14	§15	§16	§17	§18
4	26	40	14	44	12	34	62	2
§19	§20	§21	§22	§23	§24	§25	§26	§27
22	39	41	15	64	23	35	52	30
§28	§29	§30	§31	§32	§33	§34	§35	§36
6	36	58	31	59	46	51	79	68
§37	§38	§39	§40	§41	§42	§43	§44	§45
50	66	60	38	57	54	5	25	63
§46	§47	§48	§49	§50	§51	§52	§53	§54
67	81	32	53	77	84	71	82	19

Minimum kareler yöntemi için 500 iterasyonun verdiği sonuçlar aşağıdaki tabloda verilmiştir. Uygunluk değeri 43 olarak hesaplanmıştır.

Tablo 5.2: Minimum Kareler Yöntemi 500 İterasyon Tablosu

§1	§2	§3	§4	§5	§6	§7	§8	§9
78	2	24	18	37	13	43	29	56
§10	§11	§12	§13	§14	§15	§16	§17	§18
4	26	40	14	44	12	34	62	20
§19	§20	§21	§22	§23	§24	§25	§26	§27
22	39	41	15	64	23	35	52	30
§28	§29	§30	§31	§32	§33	§34	§35	§36
6	36	83	31	59	46	51	79	68
§37	§38	§39	§40	§41	§42	§43	§44	§45
50	66	60	38	57	54	5	25	63
§46	§47	§48	§49	§50	§51	§52	§53	§54
67	81	32	53	77	84	71	82	19

5.4.2. Saaty Uygunluk İndeksi Amaç Fonksiyonu Çözümü

Saaty Uygunluk İndeksi için 100 iterasyonda da 500 iterasyonda da sonuçlar uygunluk değerini veren 2 çözümü önermektedir. Uygunluk değeri her iki amaç fonksiyonu için de 155'tir. Şirketlere düşen öğrenci listesi Ek F'de verilmiştir.

Çözüm 1:

Tablo 5.3: Saaty Uygunluk İndeksi 100/500 İterasyon Çözümü 1

Ş1	Ş2	Ş3	Ş4	Ş5	Ş6	Ş7	Ş8	Ş9
76	59	47	77	4	21	20	12	65
Ş10	Ş11	Ş12	Ş13	Ş14	Ş15	Ş16	Ş17	Ş18
1	27	7	60	71	10	82	48	14
Ş19	Ş20	Ş21	Ş22	Ş23	Ş24	Ş25	Ş26	Ş27
32	53	52	29	37	56	15	50	68
Ş28	Ş29	Ş30	Ş31	Ş32	Ş33	Ş34	Ş35	Ş36
79	78	45	80	57	38	35	42	24
Ş37	Ş38	Ş39	Ş40	Ş41	Ş42	Ş43	Ş44	Ş45
63	18	69	8	64	75	6	54	84
Ş46	Ş47	Ş48	Ş49	Ş50	Ş51	Ş52	Ş53	Ş54
17	55	74	25	28	62	49	83	66

Tablo 5.4: Saaty Uygunluk İndeksi 100/500 İterasyon Çözümü 2

Ş1	Ş2	Ş3	Ş4	Ş5	Ş6	Ş7	Ş8	Ş9
40	20	80	30	78	46	52	71	22
Ş10	Ş11	Ş12	Ş13	Ş14	Ş15	Ş16	Ş17	Ş18
37	45	44	21	32	31	63	25	43
Ş19	Ş20	Ş21	Ş22	Ş23	Ş24	Ş25	Ş26	Ş27
36	68	77	6	48	70	49	65	59
Ş28	Ş29	Ş30	Ş31	Ş32	Ş33	Ş34	Ş35	Ş36
9	53	61	58	54	65	35	7	24
Ş37	Ş38	Ş39	Ş40	Ş41	Ş42	Ş43	Ş44	Ş45
33	15	5	8	83	17	72	27	67
Ş46	Ş47	Ş48	Ş49	Ş50	Ş51	Ş52	Ş53	Ş54
82	39	51	3	28	81	42	79	38

6. SONUÇLAR VE ÖNERİLER

Bu çalışmada İSO-İTÜ şirket/öğrenci eşleştirme problemi için GA kullanılmıştır. Bunun nedeni çözüm uzayının diğer yöntemlerden daha geniş olmasıdır. Bu çalışma ile, hem problem için olurlu ve en iyiye yakın çözümler listelenmiştir hem de minimum kareler ve Saaty Uygunluk İndeksi uygunluk fonksiyonları karşılaştırılmış, hangisinin daha etken olduğu belirlenmiştir.

Minimum kareler amaç fonksiyonu incelendiğinde 100 iterasyondan 500 iterasyona yavaş da olsa en iyiye yakınsamanın olduğunu görmekteyiz. Ancak Saaty Uygunluk İndeksi'ni amaç fonksiyonu olarak kullandığımızda 100 ve 500 iterasyonda aynı değeri ve çözümü vermesi, bu amaç fonksiyonunun diğerine göre daha çabuk yakınsadığı anlamına gelmektedir. Çözümün en iyi çözüm olup olmadığı tahmin edilemez ancak iterasyonlar arası yakınsama diğer uygunluk fonksiyonunda devam ederken, bu uygunluk fonksiyonunda durmuştur, daha fazla yakınsamasına gerek kalmamıştır. Bu nedenle Saaty Uygunluk İndeksi'nin daha hızlı yakınsayan bir fonksiyon olduğunu söyleyebiliriz.

Bundan sonraki çalışmalarda değişik parametreler test edilebilir. Bu parametreler şunlardır:

- Daha farklı amaç fonksiyonları denenerek onların yakınsama hızı tesbit edilebilir.
- Çeşitli seçim, çaprazlama ve dönüşüm operatörleri karşılaştırılarak optimuma daha yakın sonuç veren operatörler seçilebilir.
- Başlangıç popülasyonundaki birey sayısı değiştirilerek, yeni çözümlerin olup olmadığı araştırılabilir.
- Yeni, güncel algoritmalarla işlem verimliliği artırılabilir.

Ayrıca olurlu olmayan çözüm probleminin ortadan kaldırılabilmesi için diğer senelerde öğrencilerde tercihlerde bulunabilir ve bu tercihlere göre yerleştirmeler yapılabilir. Bunun yanı sıra, öğrenciler de konuları listeden tercih ederek buna göre bir bitirme projesi hazırlayabilirler.

KAYNAKLAR

- [1] **Erentürk, M.**, 2004. Performance Analysis of Meta-Heuristic Approaches for Traveling Salesman Problem. *Yüksek Lisans Tezi*, Yeditepe Üniversitesi, İstanbul.
- [2] **Yavuz, A.S.**, 2004. Genetic Algorithms Convergence To Traditional Optimization Methods, <http://www3.itu.edu.tr/~okerol/Optimization%20Methods.pdf>.
- [3] **Kusiak, A.**, 2004. Evolutionary Computation: Method Categorization, internetten http://ppc.iem.yzu.edu.tw/course/POM94/DataMining/EC_Methods.pdf.
- [4] **Zhau, Y.**, 2006. Study on Genetic Algorithm Improvement and Application, *Yüksek Lisans Tezi*, Worcester Polytechnic Institute, New England.
- [5] **Man, K.F., Tang, K.S. ve Kwong, S.**, 1999. *Genetic Algorithms*, Springer Publishing.
- [6] **Beasley, D., Bull, D.R. ve Martin, R.R.**, 1993. *An Overview of Genetic Algorithms: Part 1, Fundamentals*, http://ralph.cs.cf.ac.uk/papers/GAs/ga_overview1.pdf.
- [7] **Özkan R.**, 2003. Tek Modelli Deterministik Montaj Hattı Dengelem Problemlerine Genetik Algoritma ile Çözüm Yaklaşımı. *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [8] **Goldberg, D.E.**, 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company.
- [9] **Vural, M.**, 2005. Genetik Algoritma Yöntemi İle Toplu Üretim Planlama, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [10] **Sakawa, M.**, 2002. *Genetic Algorithms and Fuzzy Multiobjective Optimization*, Kluwer Academic Publishers.
- [11] **Eren, H.**, 2002. Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma (GA) ile Çözüm Performansının Arttırılmasında Deney Tasarımı Uygulaması. *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [12] **Gehring, H. ve Bortfeldt, A.**, 2004. A Parallel Genetic Algorithm for Solving The Container Loading Problem, *2nd International Technology and Strategy Forum, "Logistics and Supply Management"*, Berkeley, 7-8 Ekim 2004.
- [13] **Rocha, M., Pereira, F., Afonso, S. & Neves, J.**, 2001. A Genetic and Evolutionary Programming Environment with Spatially Structured Populations and Built-in Parallelism, *14th International Conference on Industrial and Engineering Applications of Artificial Intelligence*

and Expert Systems: Engineering of Intelligent Systems, Budapeşte, Macaristan, 4 – 7 Temmuz, s. 383 – 392.

- [14]**Deb, K.**, 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons Ltd., England.
- [15]**Elsayed, T.**, 2006. Genetic Algorithms Tutorial, *Ders Notları*, California Institute of Technology, <http://www.its.caltech.edu/~tamer/GATutorial.pdf>.
- [16]**Miller, B.L. ve Goldberg, D.E.**, 1995. Genetic Algorithms, Tournament Selection and the Effect of Noise, Illinois, ABD.
- [17]**To, T.K.W. & Ho, J.K.L.**, 2002. *A Genetic Algorithm for Configuring Reconfigurable Conveyor-Components in a Flexible Assembly Line System*, http://www.ifm.eng.cam.ac.uk/mcn/pdf_files/part6_8.pdf.
- [18]**Gen, M. ve Cheng, R.**, 1997. Genetic Algorithms and Engineering Design, John Wiley & Sons Inc., ABD.
- [19]**Sonia ve Puri, M.C.**, 2006. Two Stage Minimizing Assignment Problem, *Omega*, Basimda.
- [20]**French, A.P. ve Wilson, J.M.**, 2006. An LP Based Heuristic Procedure for the Generalized Assignment Problem with Special Ordered Sets, *Computers & Operations Research*, **34**, 2359 – 2369.
- [21]**Przybylski, A., Gandibleux, X. ve Ehrgott, M.**, 2006. Two Phase Algorithms for the Bi-Objective Assignment Problem, *European Journal of Operations Research*, Basimda.
- [22]**Zhang, C.W., ve Ong, H.L.**, 2007. An Efficient Solution to Biobjective Generalized Assignment Problem, *Advances in Engineering Software*, **38(1)**, 50-58.
- [23]**Yakoob, S.M.A. & Sherali, H.D.**, 2006. Mathematical programming models and algorithms for a class-faculty assignment problem, *European Journal of Operations Research*, **173(2)**, 488-507.
- [24]**Zaghal, F.M. ve Minoux, M.**, 2005. Modeling and Solving a Crew Assignment Problem in Air Transportation, *European Journal of Operations Research*, **175(1)**, 187-209.
- [25]**Felici, G. ve Mecoli, M.**, 2007. Resource Assignment with Preference Conditions, *European Journal of Operations Research*, **180(2)**, 519-531.
- [26]**Liu, S.S. & Wang C.J.**, 2007. Optimization Model for Resource Assignment Problems of Linear Construction Projects, *Automation in Construction*, **16(4)**, 460-473.

- [27]**Lieshout P.M.D. ve Volgenant, A.**, 2007. A Branch and Bound Algorithm for the Singly Constrained Assignment Problem, *European Journal of Operations Research*, **176(1)**, 151-164.
- [28]**Toroslu, İ.H. ve Arslanoğlu, Y.**, 2007. Genetic Algorithm for the Personnel Assignment Problem with Multiple Objectives, *Information Sciences*, **177(3)**, 787-803.
- [29]**Wang, K.J. ve Lin, Y.S.**, 2007. Resource Allocation by Genetic Algorithm with Fuzzy Inference, *Expert Systems with Applications*, **33(4)**, 1025-1035.
- [30]**Purnaprajna, R., Reformat, M., ve Pedryez, W.**, 2006. Genetic Algorithms for Hardware–Software Partitioning and Optimal Resource Allocation, *Journal of Systems Architecture*, Basımda.
- [31]**Lin, C.M. & Mitsuo, G.**, 2007. Multi-criteria Human Resource Allocation for solving Multistage Combinatorial Optimization Problems using Multiobjective Hybrid Genetic Algorithm, *Expert Systems with Applications*, Basımda.
- [32]**Wang, Y.Z.**, 2002. An application of Genetic Algorithm Methods for Teacher Assignment Problems, *Expert Systems with Applications*, **22(4)**, 295-302.
- [33]**Lin, X.H., Kwok, Y.K. ve Lau, V.K.N.**, 2003. A Genetic Algorithm Based Approach to Route Selection and Capacity Flow Assignment, *Computer Communications*, **26(9)**, 961-974.
- [34]**Salcedo-Sanchez S., Xu, Y ve Yao X.**, 2006. Hybrid Meta-heuristics Algorithms for Task Assignment in Heterogenous Computing Systems, *Computers and Operations Research*, **33(3)**, 820-835.
- [35]**Majumdar, J. ve Bhunia, A.K.**, 2007. Elitist Genetic Algorithm for Assignment Problem with Imprecise Goal, *European Journal of Operations Research*, **177(2)**, 684-692.
- [36]**Hochstattler, W., Jin, H. ve Nickel, R.**, 2005. The Hungarian Method in a Mixed Matching Market, Technical Report, Hagen, Almanya.
- [37]**Iwama, K., Miyazaki, S. ve Yamauchi, N.**, 2004. Approximation Algorithm for the Stable Marriage Problem, *The 9th Scandinavian Workshop on Algorithm Theory, SWAT*, Danimarka, 8-10 Temmuz.
- [38]**Garrett, D., Vanucci, J., Silva, R., Dasgupta, D. ve Simien, J.**, 2005. Genetic Algorithms for the Sailor Assignment Problem, *Genetic and Evolutionary Computation Conference GECCO 2005*, Washington, ABD, 25-29 Haziran, s. 1921 – 1928.
- [39]**Fleiner, T., Irwing, R.W. ve Manlove, D. F.**, 2007. Efficient algorithms for generalised stable marriage and roommates problems, *Theoretical Computer Science*, basımda.

- [40]**Drake, D.E. ve Hougardy, S.**, 2003. A simple approximation algorithm for the weighted matching problem, *Information Processing Letters*, **85(4)**, 211-213.
- [41]**Irwing, R.W. ve Manlove D.F.**, 2002. The Stable Roommates Problem with Ties, *Journal of Algorithms*, **43**, 85-105.
- [42]**Abraham, D.J., Irwing, R.W. ve Manlove, D.F.**, 2007. Two algorithms for the Student-Project Allocation Problem, *Journal of Discrete Algorithms*, **5**, 73-90.
- [43]**Sun, Y.H., Ma, J., Fan, Z.P., Wang, J.**, 2006. A Hybrid Knowledge and Model Approach for Reviewer Assignment, *Expert Systems with Applications*, Basimda.
- [44]**Higgins, A.J.**, 2001, A Dynamic Tabu Large-Scale Generalized Assignment Problems, *Computers and Operations Research*, **28(10)**, 1039-1048.
- [45]**Homg, H.C. ve Cochran, J.K.**, 2001. Project Surface Regions: A Decision Support Methodology Multitasking Workers Assignment in JIT Systems, *Computers and Industrial Engineering*, **39**, 159-171.
- [46]**Bassett, M.**, 2000. Assigning Projects to Optimize the Utilization of Employees' Time and Expertise, *Computers and Chemical Engineering*, **24**, 1013-1021.
- [47]**Bhadury, J., Mighty E.J. & Damar H.**, 2000. Maximizing Workforce Diversity in Project Teams: A Network Flow Approach, *Omega*, **28(2)**, 143-153.
- [48]**Patanakul, P., ve Milosevic, D.**, 2006. Assigning New Product Projects to Multiple Project Managers: What market leaders do?, *The Journal of HighTechnology Management Research*, **17(1)**, 53-69.
- [49]**Harper, P.R., Senna, V.D., Vieira, I.T., ve Shahani, A.K.**, 2005. A Genetic Algorithm for the Project Assignment, *Computers and Operations Research*, **32(5)**, 1255-1265.
- [50]**Saaty, T.L.**, 2005. *Theory and Applications of the Analytic Network Process: Decision Making with Benefits, Opportunities, Costs, and Risks*, RWS Publications, Pittsburgh.
- [51]**Yoon, M.S. ve Jung, H.W.**, 2003. A Study on the Utilization of Compatibility Metric in the AHP: Applying to Software Process Assesments, *International Symposium on the Analytical Hierarchy Process (ISAHP 2005)*, Hawai, Honolulu, 8-10 Temmuz.

EK A - ŞİRKET LİSTESİ

Tablo A.1: Şirket Listesi

Şirketler	
1	Akçalı Boya Sanayi
2	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri
3	Betek Boya ve Kimya
4	Bilgi Elektronik
5	Çağrı End. A.Ş.
6	Demo Dekorasyon Mobilya Ağaç Pazarlama
7	Denet Cıvata
8	Doğan Ofset A.Ş.
9	Ekol Ofset Matbaacılık
10	Erge Elektrik Elektronik Sanayi
11	Etem Triko Konfeksiyon A.Ş.
12	Finansbank
13	Fita Teknik LTD.
14	Ford Otomotiv Sanayi A.Ş.
15	Gemak Gemi İnşaat
16	Gerboy Tekstil ve Tic. Sanayi A.Ş.
17	GES Elektronik
18	Gimsan Bağlantı Elemanları
19	Halkalı Kağıt Karton San. A.Ş.
20	Isı Sanayi A.Ş.
21	İkiler Reklam Aydınlatma
22	İkizler Aydınlatma Armatürleri
23	İntema İnşaat ve Tesisat Malzemeleri
24	İsfalt A.Ş.
25	Kalealtınay Robotik ve Otomasyon A.Ş.
26	Koleksiyon Mobilya
27	Marshall A.Ş.
28	Marshall Boya ve Verniksan A.Ş.
29	Nova Reklamcılık ve Sanayi A.Ş.
30	Önaysan Metal Sanayi
31	Öz Yüksel Deri ve Sanayi A.Ş.
32	Pharmavision
33	Rekor Kauçuk
34	Sarkuysan Elektronik
35	Sem Metal Plastik
36	Ses Plastik
37	Siemens San. Ve Tic. A.Ş.
38	Sözer Kuyumculuk A.Ş.
39	Tat Konserve Sanayi
40	Torgem Gemi İnşaat
41	Trakya Cam ve Sanayi A.Ş.
42	Travertine Bros
43	Yıldız Cam A.Ş.
44	Yılteks Yıkama Sanayi
45	Zorlu Tekstil

EK B - ÖĞRENCİ TERCİHLERİ TABLOSU

Tablo B.1: Öğrenci Tercihleri ve Numaraları

No	Öğrenci	1	2	3	4	5	6	7	8	9
1	Nazmi Ziya Özgür	9	0	0	0	1	0	0	0	0
2	Eren Sevim	9	0	0	1	0	0	0	0	0
3	Metin Balcı	9	0	0	0	0	1	0	0	0
4	Ahmet Boğ	9	0	0	0	0	1	0	0	0
5	Hikmet Çakırtaş	9	1	0	0	0	0	0	2	0
6	Nejdet Yılmaz	9	0	0	0	0	0	1	0	0
7	Sibel Yılmaz	0	1	0	0	0	0	0	0	2
8	Eren Yeşil	9	0	0	0	0	0	0	0	1
9	Yavuz Selim Torbaoğlu	9	0	0	0	0	0	0	0	1
10	Emre İstanbullu	9	0	0	0	0	0	0	0	1
11	Derya Üstün	9	9	9	9	9	9	1	9	9
12	Adem Kazaz	9	2	0	0	0	1	0	0	0
13	Tolga Maden	9	1	0	0	0	0	0	0	0
14	Kerem Ödev	9	1	0	0	0	0	0	0	0
15	Cahit Sinan Hamarat	9	1	0	0	0	0	0	0	0
16	Yusuf Aksoy	9	0	0	0	0	0	0	0	1
17	Canan Herdem	3	2	0	0	0	4	5	0	1
18	Özkan Işıl	9	9	9	2	9	9	9	9	1
19	Engin Güzel	9	0	0	0	0	1	0	0	2
20	Çiğdem Zaman	9	9	9	9	9	9	1	9	9
21	Mustafa Bayram	9	1	0	0	0	0	0	0	0
22	Alper Hasboyacı	9	2	0	0	0	0	0	1	0
23	Kerim Karakuş	9	0	0	0	0	0	0	0	1
24	Aybike Eskin	2	9	9	9	9	9	1	9	9
25	Gülver Karamemiş	9	3	0	0	0	0	0	1	2
26	Deniz Bahar	9	2	0	0	0	1	0	0	0
27	Hanife Selen Akyol	2	9	9	9	9	9	9	9	1
28	Çiğdem Kadaifçi	9	0	0	0	0	0	1	0	2
29	Hüseyin Emre Yüksel	9	0	0	0	0	0	0	1	0
30	Özgür Aykul	3	2	0	0	0	4	5	0	1
31	Gökmen Çalışan	9	1	0	0	0	0	0	2	0
32	Umut Albayrak	9	1	0	9	0	0	0	2	0
33	Ayşegül Tizer	9	2	0	0	0	0	1	0	0
34	Gizem Karlılar	9	0	0	0	0	0	0	1	2
35	Zafer Eren	9	1	0	0	0	0	0	0	2
36	Ayşe Yücel	1	9	9	9	9	9	9	9	9
37	Yasin Akyıldız	1	9	9	9	9	9	9	9	9
38	Sadık Dadaloğlu	9	1	0	0	9	0	0	0	0
39	Murat Kurt	9	9	9	9	9	9	1	9	9
40	Onur Akça	1	3	9	9	9	2	9	9	9
41	Melih Nakaş	9	9	9	1	9	9	9	9	9
42	Muhammed Erdoğan	9	2	0	0	0	0	0	0	1
43	Ali Erdem	9	2	0	0	0	0	0	1	0
44	Kanarya İsmail Daud	9	1	9	9	9	9	9	9	9
45	Neslihan Gül Mençi	9	1	0	0	0	0	0	0	0
46	Erdal Aydın	9	9	1	9	9	9	9	9	9
47	Faruk Edik	9	9	1	9	9	9	9	9	9

48	Furkan Aşkın	9	0	0	1	0	0	0	0	2
49	Ali Güncan	9	9	1	9	9	9	9	9	9
50	Gökhan Eğilmez	9	0	0	0	0	0	0	0	1
51	Hüseyin Cengiz	9	9	1	9	9	9	9	9	9
52	Cengiz Günaydın	9	0	0	0	0	0	0	0	1
53	Tezcan Mutlu	9	1	0	0	0	0	0	0	0
54	Hakan Bozoğlu	9	9	9	1	9	9	9	9	9
55	Salih Çiçek	9	9	9	1	9	9	9	9	9
56	Gamze Akıncı	9	0	0	0	0	0	0	0	1
57	Çiğdem Atazüla	9	1	0	0	0	0	0	0	0
58	Mehmet Emin İşbilen	9	0	1	0	0	0	0	0	0
59	Onur Kurtçu	9	0	0	0	0	0	1	0	0
60	Tarık Mete	9	1	0	0	2	0	0	0	0
61	Çağdaş Atalay	9	0	0	0	0	1	0	0	2
62	Deniz Aslan	9	0	0	0	0	0	0	0	1
63	Mustafa Caferoğlu	9	9	9	9	9	9	9	9	1
64	Serhat Yalçınkaya	9	0	0	0	0	0	0	0	1
65	Onur Özdemir	9	0	0	0	0	0	0	0	1
66	İbrahim Taştekin	9	0	0	0	0	0	0	0	1
67	Aslı Çağla Yıldırım	9	0	0	0	0	0	0	0	1
68	Aytekin Bal	9	2	0	0	0	0	0	1	0
69	Armağan Bayram	1	9	9	9	9	9	9	9	9
70	Myrat Amangeldiyew	9	0	0	0	2	0	0	1	0
71	Duygu Demir	9	1	2	0	0	3	0	5	4
72	Yalçın Kayacık	9	1	0	0	0	0	0	2	0
73	Dinçer Kara	1	9	9	9	9	9	9	9	9
74	Muhammet Kaplan	9	1	0	0	0	0	0	2	0
75	Hande Alaca	9	0	0	0	0	0	0	1	0
76	Gökhan Zengin	9	9	9	9	1	9	9	9	9
77	Figen Şam	9	9	9	9	9	1	9	9	9
78	Belgin Koç	9	9	9	1	9	9	9	9	9
79	Övgü Tunç	9	0	0	0	0	0	0	0	1
80	Müge Gök	9	9	1	9	9	9	9	9	9
81	Burak Öç	1	0	0	0	0	0	0	0	2
82	Bahadır Duman	9	2	0	0	0	0	0	1	0
83	Sermin Berberoğlu	9	9	9	9	9	9	9	1	9
84	Hasan Berk Aymete	9	0	0	0	0	0	1	2	0

EK C – ŞİRKETLERİN KROMOZOMDAKİ KONUMLARI

Tablo C.1: Şirketlerin Kromozomdaki Konumları ve Tercihleri

No	Şirket	1	2	3	4	5	6	7	8	9
1	Ekol Ofset Matbaacılık	0	0	0	0	1	0	0	0	0
2	Pharmavision	0	0	0	0	1	0	0	0	0
3	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	0	0	0	1	0	0	0	0	0
4	Marshall Boya ve Verniksan A.Ş.	0	0	0	0	0	1	0	0	0
5	Marshall Boya ve Verniksan A.Ş.	0	0	0	0	0	1	0	0	0
6	Sem Metal Plastik	1	2	0	0	0	0	0	0	0
7	Torgem Gemi İnşaat	3	0	4	2	0	0	0	0	1
8	Denet Cıvata	1	3	0	0	0	0	0	0	2
9	Erge Elektrik Elektronik Sanayi	1	1	1	1	1	1	1	1	1
10	Marshall	1	0	0	0	2	0	3	0	0
11	Marshall	1	0	0	0	2	0	3	0	0
12	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	0	1	0	0	0	0	0	0	0
13	Gimsan Bağlantı Elemanları	0	1	2	0	0	0	0	0	0
14	Yıldız Cam A.Ş.	0	3	1	2	0	0	0	0	0
15	Yılteks Yıkama Sanayi	0	4	3	0	2	0	1	0	5
16	Demo Dekorasyon Mobilya Ağaç Pazarlama	1	3	4	0	5	0	0	0	2
17	İsfalt A.Ş.	2	1	0	0	0	3	0	5	4
18	Etem Triko Konfeksiyon A.Ş.	1	3	5	4	6	0	2	0	4
19	Isı Sanayi A.Ş.	0	1	0	0	0	0	0	0	0
20	Önaysan Metal Sanayi	6	5	4	3	2	0	0	1	0
21	Sarkuysan Elektronik	2	0	0	0	1	0	0	0	0
22	Finansbank	5	2	3	0	0	0	0	1	4
23	Fita Teknik LTD.	2	1	3	0	4	0	5	0	0
24	Çağrı End. A.Ş.	1	1	1	1	1	1	1	1	1
25	Marshall Boya	1	1	0	0	0	1	0	0	0
26	Betek Boya ve Kimya	4	1	6	5	0	8	7	2	3
27	Bilgi Elektronik	1	1	1	1	1	1	1	1	1
28	Siemens San. Ve Tic. A.Ş.	0	0	0	0	0	0	1	0	2
29	Kalealtınay Robotik ve Otomasyon A.Ş.	0	2	0	1	0	0	0	0	0
30	GES Elektronik	2	1	0	0	0	0	3	4	5

31	Zorlu Tekstil	2	3	0	4	0	0	0	1	0
32	Doğan Ofset A.Ş.	1	2	0	0	0	0	0	0	0
33	İntema İnşaat ve Tesisat Malzemeleri	1	0	0	0	0	0	0	0	0
34	Rekor Kauçuk	1	0	0	0	0	0	0	0	0
35	Fita Teknik LTD.	2	1	3	0	4	0	5	0	0
36	İkiler Reklam Aydınlatma	1	2	0	0	0	0	3	4	5
37	Marshall Boya ve Verniksan A.Ş.	1	0	0	0	1	0	1	0	0
38	Trakya Cam ve Sanayi A.Ş.	1	0	0	0	2	0	0	0	0
39	Nova Reklamcılık ve Sanayi A.Ş.	2	1	0	3	0	0	0	0	0
40	Öz Yüksel Deri ve Sanayi A.Ş.	2	1	0	0	0	0	4	0	3
41	Halkalı Kağıt Karton San. A.Ş.	2	5	4	0	0	3	0	1	0
42	Rekor Kauçuk	1	0	0	0	0	0	0	0	0
43	Koleksiyon Mobilya	3	2	1	0	0	4	0	0	0
44	Gerboy Tekstil ve Tic. Sanayi A.Ş.	0	2	1	0	0	0	0	0	0
45	Travertine Bros	1	0	2	0	0	0	0	0	0
46	İkizler Aydınlatma Armatürleri	0	1	2	0	0	0	0	0	3
47	Tat Konserve Sanayi	1	0	0	1	0	0	0	1	0
48	Ses Plastik	1	2	4	6	0	3	5	0	0
49	Gemak Gemi İnşaat	1	0	0	3	0	0	0	2	0
50	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	1	2	0	0	3	4	0	0	0
51	Sözer Kuyumculuk A.Ş.	3	1	2	0	0	0	0	0	4
52	Ford Otomotiv Sanayi A.Ş.	0	4	3	0	2	0	1	0	5
53	Kalealtınay Robotik ve Otomasyon A.Ş.	2	0	0	0	0	0	1	0	0
54	Akçalı Boya Sanayi	1	2	0	0	0	0	0	3	0

EK D – MİNİMUM KARELER EXCEL VBA KODLARI

```
Private Function generation(sayac2 As Integer)
Dim sayac As Integer
Dim sayac3 As Integer
    For sayac = 1 To 54
        Sheets("eşleşme").Cells(sayac2, sayac) = Int(84 * Rnd + 1)
        For sayac3 = 1 To sayac - 1
            If Sheets("eşleşme").Cells(sayac2, sayac) = Sheets("eşleşme").Cells(sayac2, sayac3) Then
                Sheets("eşleşme").Cells(sayac2, sayac) = Int(84 * Rnd + 1)
                sayac3 = 1
            End If
        Next sayac3
    Next sayac
End Function
```

```
Public Function comparison(sirketNo As Integer, ogrenciNo As Integer)
Dim durum As Boolean
Dim tercihogr(9) As Integer
Dim tercihsir(9) As Integer
Dim sutun As Integer
For sutun = 1 To 9
    tercihogr(sutun) = Sheets("öğrenciler").Cells(ogrenciNo + 1, sutun + 1).Value
    tercihsir(sutun) = Sheets("şirketler").Cells(sirketNo + 1, sutun + 1).Value
Next sutun
durum = False
For sutun = 1 To 9
    If ((tercihogr(sutun) > 0) And (tercihsir(sutun) > 0)) Then
        durum = True
    End If
Next sutun
comparison = durum
End Function
```

```
Public Function feasibility(satirNo As Integer)
Dim durum As Boolean
Dim sutun As Integer
Dim ogrenci As Integer
durum = True
For sutun = 1 To 54
    ogrenci = Sheets("eşleşme").Cells(satirNo, sutun).Value
    If (comparison(sutun, ogrenci) = False) Then
        durum = False
    End If
Next sutun
feasibility = durum
End Function
```

```

Public Function getval(sirketNo As Integer, ogrenciNo As Integer)
Dim tercihogr(9) As Integer
Dim tercihsir(9) As Integer
Dim uyum As Integer
Dim sutun As Integer
For sutun = 1 To 9
uyum = 0
tercihogr(sutun) = Sheets("öğrenciler").Cells(ogrenciNo + 1, sutun + 1).Value
tercihsir(sutun) = Sheets("şirketler").Cells(sirketNo + 1, sutun + 1).Value
If ((tercihogr(sutun) > 0) And (tercihsir(sutun) > 0)) Then
uyum = (tercihogr(sutun) - tercihsir(sutun)) ^ 2
End If
Next sutun
getval = uyum
End Function

```

```

Public Function fitness(satirNo As Integer)
Dim sutun As Integer
Dim toplam As Integer
Dim ogrenci As Integer

```

```

toplam = 0
For sutun = 1 To 54
ogrenci = Sheets("eşleşme").Cells(satirNo, sutun).Value
toplam = toplam + getval(sutun, ogrenci)
Next sutun
fitness = toplam
End Function

```

```

Private Function reproduction()
Dim satir As Integer
Dim rassal As Double
Dim sutun As Integer
For satir = 1 To 20
rassal = Rnd
For sutun = 1 To 55
If (rassal <= Sheets("eşleşme").Range("BF2").Value) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(2,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF2").Value) And (rassal <=
Sheets("eşleşme").Range("BF3").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(3,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF3").Value) And (rassal <=
Sheets("eşleşme").Range("BF4").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(4,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF4").Value) And (rassal <=
Sheets("eşleşme").Range("BF5").Value)) Then

```

```

Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(5,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF5").Value) And (rassal <=
Sheets("eşleşme").Range("BF6").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(6,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF6").Value) And (rassal <=
Sheets("eşleşme").Range("BF7").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(7,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF7").Value) And (rassal <=
Sheets("eşleşme").Range("BF8").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(8,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF8").Value) And (rassal <=
Sheets("eşleşme").Range("BF9").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(9,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF9").Value) And (rassal <=
Sheets("eşleşme").Range("BF10").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(10,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF10").Value) And (rassal <=
Sheets("eşleşme").Range("BF11").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(11,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF11").Value) And (rassal <=
Sheets("eşleşme").Range("BF12").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(12,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF12").Value) And (rassal <=
Sheets("eşleşme").Range("BF13").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(13,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF13").Value) And (rassal <=
Sheets("eşleşme").Range("BF14").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(14,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF14").Value) And (rassal <=
Sheets("eşleşme").Range("BF15").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(15,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF15").Value) And (rassal <=
Sheets("eşleşme").Range("BF16").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(16,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF16").Value) And (rassal <=
Sheets("eşleşme").Range("BF17").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(17,
sutun).Value

```

```

ElseIf ((rassal > Sheets("eşleşme").Range("BF17").Value) And (rassal <=
Sheets("eşleşme").Range("BF18").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(18,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF18").Value) And (rassal <=
Sheets("eşleşme").Range("BF19").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(19,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF19").Value) And (rassal <=
Sheets("eşleşme").Range("BF20").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(20,
sutun).Value
Else
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(21,
sutun).Value
End If
Next sutun
Next satir
End Function

```

```

Public Function pairs()
Dim satir(20) As Integer
Dim sayac As Integer
Dim sayac2 As Integer
For sayac = 1 To 20
satir(sayac) = Int(20 * Rnd + 25)
For sayac2 = 1 To sayac - 1
Do While (satir(sayac) = satir(sayac2))
satir(sayac) = Int(20 * Rnd + 25)
sayac2 = 1
Loop
Next sayac2
Sheets("eşleşme").Cells(sayac + 24, 57).Value = satir(sayac)
Next sayac
End Function
Public Function mutation(satir As Integer, sutun As Integer)
Sheets("eşleşme").Cells(satir, sutun).Value = Int(83 * Rnd + 1)
End Function

```

```

Public Function crossover()
Dim i As Integer
Dim sutun As Integer
Dim cift1 As Integer
Dim cift2 As Integer
Dim satir1 As Integer
Dim satir2 As Integer
Dim pair1(54) As Integer
Dim pair2(54) As Integer
Dim rassal As Double
Dim cpoint As Integer

```

```

Dim offspring1(54) As Integer
Dim offspring2(54) As Integer
Dim sayac As Integer
Dim rassal2 As Double 'crossover probability
satir1 = 25
satir2 = 26
For i = 1 To 10
cift1 = Sheets("eşleşme").Cells(satir1, 57).Value
cift2 = Sheets("eşleşme").Cells(satir2, 57).Value
For sutun = 1 To 54
pair1(sutun) = Sheets("eşleşme").Cells(cift1, sutun).Value
pair2(sutun) = Sheets("eşleşme").Cells(cift2, sutun).Value
Next sutun
rassal2 = Rnd
If (rassal2 <= 0.9) Then
    rassal = Rnd
    If (rassal <= 1 / 53) Then
        cpoint = 1
    ElseIf ((rassal > 1 / 53) And (rassal <= 2 / 53)) Then
        cpoint = 2
    ElseIf ((rassal > 2 / 53) And (rassal <= 3 / 53)) Then
        cpoint = 3
    ElseIf ((rassal > 3 / 53) And (rassal <= 4 / 53)) Then
        cpoint = 4
    ElseIf ((rassal > 4 / 53) And (rassal <= 5 / 53)) Then
        cpoint = 5
    ElseIf ((rassal > 5 / 53) And (rassal <= 6 / 53)) Then
        cpoint = 6
    ElseIf ((rassal > 6 / 53) And (rassal <= 7 / 53)) Then
        cpoint = 7
    ElseIf ((rassal > 7 / 53) And (rassal <= 8 / 53)) Then
        cpoint = 8
    ElseIf ((rassal > 8 / 53) And (rassal <= 9 / 53)) Then
        cpoint = 9
    ElseIf ((rassal > 9 / 53) And (rassal <= 10 / 53)) Then
        cpoint = 10
    ElseIf ((rassal > 10 / 53) And (rassal <= 11 / 53)) Then
        cpoint = 11
    ElseIf ((rassal > 11 / 53) And (rassal <= 12 / 53)) Then
        cpoint = 12
    ElseIf ((rassal > 12 / 53) And (rassal <= 13 / 53)) Then
        cpoint = 13
    ElseIf ((rassal > 13 / 53) And (rassal <= 14 / 53)) Then
        cpoint = 14
    ElseIf ((rassal > 14 / 53) And (rassal <= 15 / 53)) Then
        cpoint = 15
    ElseIf ((rassal > 15 / 53) And (rassal <= 16 / 53)) Then
        cpoint = 16
    ElseIf ((rassal > 16 / 53) And (rassal <= 17 / 53)) Then
        cpoint = 17

```

```

ElseIf ((rassal > 17 / 53) And (rassal <= 18 / 53)) Then
cpoint = 18
ElseIf ((rassal > 18 / 53) And (rassal <= 19 / 53)) Then
cpoint = 19
ElseIf ((rassal > 19 / 53) And (rassal <= 20 / 53)) Then
cpoint = 20
ElseIf ((rassal > 20 / 53) And (rassal <= 21 / 53)) Then
cpoint = 21
ElseIf ((rassal > 21 / 53) And (rassal <= 22 / 53)) Then
cpoint = 22
ElseIf ((rassal > 22 / 53) And (rassal <= 23 / 53)) Then
cpoint = 23
ElseIf ((rassal > 23 / 53) And (rassal <= 24 / 53)) Then
cpoint = 24
ElseIf ((rassal > 24 / 53) And (rassal <= 25 / 53)) Then
cpoint = 25
ElseIf ((rassal > 25 / 53) And (rassal <= 26 / 53)) Then
cpoint = 26
ElseIf ((rassal > 26 / 53) And (rassal <= 27 / 53)) Then
cpoint = 27
ElseIf ((rassal > 27 / 53) And (rassal <= 28 / 53)) Then
cpoint = 28
ElseIf ((rassal > 28 / 53) And (rassal <= 29 / 53)) Then
cpoint = 29
ElseIf ((rassal > 29 / 53) And (rassal <= 30 / 53)) Then
cpoint = 30
ElseIf ((rassal > 30 / 53) And (rassal <= 31 / 53)) Then
cpoint = 31
ElseIf ((rassal > 31 / 53) And (rassal <= 32 / 53)) Then
cpoint = 32
ElseIf ((rassal > 32 / 53) And (rassal <= 33 / 53)) Then
cpoint = 33
ElseIf ((rassal > 33 / 53) And (rassal <= 34 / 53)) Then
cpoint = 34
ElseIf ((rassal > 34 / 53) And (rassal <= 35 / 53)) Then
cpoint = 35
ElseIf ((rassal > 35 / 53) And (rassal <= 36 / 53)) Then
cpoint = 36
ElseIf ((rassal > 36 / 53) And (rassal <= 37 / 53)) Then
cpoint = 37
ElseIf ((rassal > 37 / 53) And (rassal <= 38 / 53)) Then
cpoint = 38
ElseIf ((rassal > 38 / 53) And (rassal <= 39 / 53)) Then
cpoint = 39
ElseIf ((rassal > 39 / 53) And (rassal <= 40 / 53)) Then
cpoint = 40
ElseIf ((rassal > 40 / 53) And (rassal <= 41 / 53)) Then
cpoint = 41
ElseIf ((rassal > 41 / 53) And (rassal <= 42 / 53)) Then
cpoint = 42

```

```

ElseIf ((rassal > 42 / 53) And (rassal <= 43 / 53)) Then
cpoint = 43
ElseIf ((rassal > 43 / 53) And (rassal <= 44 / 53)) Then
cpoint = 44
ElseIf ((rassal > 44 / 53) And (rassal <= 45 / 53)) Then
cpoint = 45
ElseIf ((rassal > 45 / 53) And (rassal <= 46 / 53)) Then
cpoint = 46
ElseIf ((rassal > 46 / 53) And (rassal <= 47 / 53)) Then
cpoint = 47
ElseIf ((rassal > 47 / 53) And (rassal <= 48 / 53)) Then
cpoint = 48
ElseIf ((rassal > 48 / 53) And (rassal <= 49 / 53)) Then
cpoint = 49
ElseIf ((rassal > 49 / 53) And (rassal <= 50 / 53)) Then
cpoint = 50
ElseIf ((rassal > 50 / 53) And (rassal <= 51 / 53)) Then
cpoint = 51
ElseIf ((rassal > 51 / 53) And (rassal <= 52 / 53)) Then
cpoint = 52
Else
cpoint = 53
End If
End If
For sayac = 1 To cpoint
offspring1(sayac) = pair1(sayac)
offspring2(sayac) = pair2(sayac)
Next sayac
For sayac = cpoint + 1 To 54
offspring1(sayac) = pair2(sayac)
offspring2(sayac) = pair1(sayac)
Next sayac
For sutun = 1 To 54
Sheets("eşleşme").Cells(satir1 + 23, sutun).Value = offspring1(sutun)
Sheets("eşleşme").Cells(satir2 + 23, sutun).Value = offspring2(sutun)
Next sutun
satir1 = satir1 + 2
satir2 = satir2 + 2
Next i
End Function

```

```

Public Function duplicate(satirNo As Integer)
Dim sutun As Integer
Dim sayac As Integer
Dim rndm As Integer
Dim sutun2 As Integer
For sutun = 1 To 54
For sayac = 1 To sutun - 1
If (Sheets("eşleşme").Cells(satirNo, sutun).Value = Sheets("eşleşme").Cells(satirNo,
sayac).Value) Then

```

```

rndm = Int(19 * Rnd + 2)
For sutun2 = 1 To 55
Sheets("eşleşme").Cells(satirNo, sutun2).Value = Sheets("eşleşme").Cells(rndm,
sutun2).Value
Next sutun2
sayac = 1
End If
Next sayac
Next sutun
End Function

```

```

Public Function minresult()
Dim satir1 As Integer
Dim satir2 As Integer
Dim sutun As Integer
Dim result As Long
Dim result1 As Long
Dim result2 As Long
Dim sonuc1(54) As Integer
Dim sonuc2(54) As Integer
Dim sonuc(54) As Integer
result1 = 50000
result2 = 50000
For satir1 = 2 To 21
If (result1 > Sheets("eşleşme").Cells(satir1, 55).Value) Then
result1 = Sheets("eşleşme").Cells(satir1, 55).Value
For sutun = 1 To 54
sonuc1(sutun) = Sheets("eşleşme").Cells(satir1, sutun).Value
Next sutun
End If
Next satir1
For satir2 = 48 To 67
If (result2 > Sheets("eşleşme").Cells(satir2, 55).Value) Then
result2 = Sheets("eşleşme").Cells(satir2, 55).Value
For sutun = 1 To 54
sonuc2(sutun) = Sheets("eşleşme").Cells(satir2, sutun).Value
Next sutun
End If
Next satir2
If (result1 > result2) Then
result = result2
For sutun = 1 To 54
sonuc(sutun) = sonuc2(sutun)
Next sutun
Else
result = result1
For sutun = 1 To 54
sonuc(sutun) = sonuc1(sutun)
Next sutun
End If

```

```

Sheets("eşleşme").Cells(71, 55).Value = result
For sutun = 1 To 54
Sheets("eşleşme").Cells(71, sutun).Value = sonuc(sutun)
Next sutun
End Function

```

```

Public Function kopyala()
Dim satir As Integer
Dim sutun As Integer
For satir = 48 To 67
For sutun = 1 To 55
Sheets("eşleşme").Cells(satir - 46, sutun).Value = Sheets("eşleşme").Cells(satir,
sutun).Value
Next sutun
Next satir
End Function

```

```

Private Sub CommandButton1_Click()
Dim satir As Integer
Dim sutun As Integer
Dim sayac2 As Integer
Dim mutprob As Double
Dim rassal3 As Double
Dim iteration As Integer
Dim sonuc As Long
Dim result(54) As Integer
mutprob = 0.001
For sayac2 = 2 To 21
Call generation(sayac2)
Next sayac2
For satir = 2 To 21
If (feasibility(satir) = True) Then
Sheets("eşleşme").Cells(satir, 55) = "ok"
Else
Do While (feasibility(satir) = False)
Call generation(satir)
Loop
End If
Sheets("eşleşme").Cells(satir, 55) = "ok"
Next satir
For satir = 2 To 21
Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call reproduction
Call pairs
Call crossover
For satir = 48 To 67
For sutun = 1 To 54
rassal3 = Rnd

```

```

If (rassal3 <= mutprob) Then
Call mutation(satir, sutun)
End If
Next sutun
Next satir
For satir = 48 To 67
Call duplicate(satir)
Next satir
For satir = 48 To 67
If (feasibility(satir) = True) Then
  Sheets("eşleşme").Cells(satir, 56) = "ok"
Else
  rassal3 = Int(19 * Rnd + 2)
  For sutun = 1 To 54
    Sheets("Eşleşme").Cells(satir, sutun).Value = Sheets("eşleşme").Cells(rassal3,
sutun).Value
  Next sutun
End If
Next satir

For satir = 48 To 67
Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call minresult
sonuc = Sheets("eşleşme").Cells(71, 55).Value
For sutun = 1 To 54
result(sutun) = Sheets("eşleşme").Cells(71, sutun).Value
Next sutun
Sheets("eşleşme").Cells(74, 55).Value = sonuc
For sutun = 1 To 54
Sheets("eşleşme").Cells(74, sutun).Value = result(sutun)
Next sutun
For iteration = 1 To 100
Call kopyala
Call reproduction
Call pairs
Call crossover
For satir = 48 To 67
For sutun = 1 To 54
rassal3 = Rnd
If (rassal3 <= mutprob) Then
Call mutation(satir, sutun)
End If
Next sutun
Next satir
For satir = 48 To 67
Call duplicate(satir)
Next satir
For satir = 48 To 67
If (feasibility(satir) = True) Then

```

```

Sheets("eşleşme").Cells(satir, 56) = "ok"
Else
rassal3 = Int(19 * Rnd + 2)
For sutun = 1 To 54
    Sheets("Eşleşme").Cells(satir, sutun).Value = Sheets("eşleşme").Cells(rassal3,
sutun).Value
Next sutun
End If
Next satir
For satir = 48 To 67
    Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call minresult
If (Sheets("eşleşme").Cells(71, 55).Value < Sheets("eşleşme").Cells(74, 55).Value)
Then
    Sheets("eşleşme").Cells(74, 55).Value = Sheets("eşleşme").Cells(71, 55).Value
For sutun = 1 To 54
    Sheets("eşleşme").Cells(74, sutun).Value = Sheets("eşleşme").Cells(71, sutun).Value
Next sutun
End If
Next iteration
End Sub

```

EK E – SAATY UYGUNLUK İNDEKSİ EXCEL VBA KODLARI

```
Private Function generation(sayac2 As Integer)
Dim sayac As Integer
Dim sayac3 As Integer
For sayac = 1 To 54
    Sheets("eşleşme").Cells(sayac2, sayac) = Int(84 * Rnd + 1)
    For sayac3 = 1 To sayac - 1
        If Sheets("eşleşme").Cells(sayac2, sayac) = Sheets("eşleşme").Cells(sayac2,
sayac3) Then
            Sheets("eşleşme").Cells(sayac2, sayac) = Int(84 * Rnd + 1)
            sayac3 = 1
        End If
    Next sayac3
Next sayac
End Function
```

```
Public Function comparison(sirketNo As Integer, ogrenciNo As Integer)
Dim durum As Boolean
Dim tercihogr(9) As Integer
Dim tercihsir(9) As Integer
Dim sutun As Integer
For sutun = 1 To 9
    tercihogr(sutun) = Sheets("öğrenciler").Cells(ogrenciNo + 1, sutun + 1).Value
    tercihsir(sutun) = Sheets("şirketler").Cells(sirketNo + 1, sutun + 1).Value
Next sutun
durum = False
For sutun = 1 To 9
    If ((tercihogr(sutun) > 0) And (tercihsir(sutun) > 0)) Then
        durum = True
    End If
Next sutun
comparison = durum
End Function
```

```
Public Function feasibility(satirNo As Integer)
Dim durum As Boolean
Dim sutun As Integer
Dim ogrenci As Integer
durum = True
For sutun = 1 To 54
    ogrenci = Sheets("eşleşme").Cells(satirNo, sutun).Value
    If (comparison(sutun, ogrenci) = False) Then
        durum = False
    End If
Next sutun
feasibility = durum
End Function
```

```
Public Function getval(sirketNo As Integer, ogrenciNo As Integer)
```

```

Dim tercihogr(9) As Integer
Dim tercihsir(9) As Integer
Dim uyum As Integer
Dim sutun As Integer
uyum = 0
For sutun = 1 To 9
tercihogr(sutun) = Sheets("öğrenciler").Cells(ogrenciNo + 1, sutun + 1).Value
tercihsir(sutun) = Sheets("şirketler").Cells(sirketNo + 1, sutun + 1).Value
Sheets("hadamard").Cells(sutun, 1).Value = tercihogr(sutun)
If (tercihogr(sutun) = 0) Then
Sheets("hadamard").Cells(sutun, 1).Value = 10
End If
Sheets("hadamard").Cells(sutun + 10, 1).Value = tercihsir(sutun)
If (tercihsir(sutun) = 0) Then
Sheets("hadamard").Cells(sutun + 10, 1).Value = 10
End If
Next sutun
uyum = Sheets("hadamard").Cells(31, 3).Value
getval = uyum
End Function

```

```

Public Function fitness(satirNo As Integer)
Dim sutun As Integer
Dim toplam As Integer
Dim ogrenci As Integer
toplam = 0
For sutun = 1 To 54
ogrenci = Sheets("eşleşme").Cells(satirNo, sutun).Value
toplam = toplam + getval(sutun, ogrenci)
Next sutun
fitness = toplam
End Function

```

```

Private Function reproduction()
Dim satir As Integer
Dim rassal As Double
Dim sutun As Integer
For satir = 1 To 20
rassal = Rnd
For sutun = 1 To 55
If (rassal <= Sheets("eşleşme").Range("BF2").Value) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(2,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF2").Value) And (rassal <=
Sheets("eşleşme").Range("BF3").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(3,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF3").Value) And (rassal <=
Sheets("eşleşme").Range("BF4").Value)) Then

```

```

Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(4,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF4").Value) And (rassal <=
Sheets("eşleşme").Range("BF5").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(5,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF5").Value) And (rassal <=
Sheets("eşleşme").Range("BF6").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(6,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF6").Value) And (rassal <=
Sheets("eşleşme").Range("BF7").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(7,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF7").Value) And (rassal <=
Sheets("eşleşme").Range("BF8").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(8,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF8").Value) And (rassal <=
Sheets("eşleşme").Range("BF9").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(9,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF9").Value) And (rassal <=
Sheets("eşleşme").Range("BF10").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(10,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF10").Value) And (rassal <=
Sheets("eşleşme").Range("BF11").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(11,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF11").Value) And (rassal <=
Sheets("eşleşme").Range("BF12").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(12,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF12").Value) And (rassal <=
Sheets("eşleşme").Range("BF13").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(13,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF13").Value) And (rassal <=
Sheets("eşleşme").Range("BF14").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(14,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF14").Value) And (rassal <=
Sheets("eşleşme").Range("BF15").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(15,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF15").Value) And (rassal <=
Sheets("eşleşme").Range("BF16").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(16,
sutun).Value

```

```

ElseIf ((rassal > Sheets("eşleşme").Range("BF16").Value) And (rassal <=
Sheets("eşleşme").Range("BF17").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(17,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF17").Value) And (rassal <=
Sheets("eşleşme").Range("BF18").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(18,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF18").Value) And (rassal <=
Sheets("eşleşme").Range("BF19").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(19,
sutun).Value
ElseIf ((rassal > Sheets("eşleşme").Range("BF19").Value) And (rassal <=
Sheets("eşleşme").Range("BF20").Value)) Then
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(20,
sutun).Value
Else
Sheets("eşleşme").Cells(satir + 24, sutun).Value = Sheets("eşleşme").Cells(21,
sutun).Value
End If
Next sutun
Next satir
End Function

```

```

-----
Public Function pairs()
Dim satir(20) As Integer
Dim sayac As Integer
Dim sayac2 As Integer
For sayac = 1 To 20
satir(sayac) = Int(20 * Rnd + 25)
For sayac2 = 1 To sayac - 1
Do While (satir(sayac) = satir(sayac2))
satir(sayac) = Int(20 * Rnd + 25)
sayac2 = 1
Loop
Next sayac2
Sheets("eşleşme").Cells(sayac + 24, 57).Value = satir(sayac)
Next sayac
End Function
Public Function mutation(satir As Integer, sutun As Integer)
Sheets("eşleşme").Cells(satir, sutun).Value = Int(83 * Rnd + 1)
End Function

```

```

-----
Public Function crossover()
Dim i As Integer
Dim sutun As Integer
Dim cift1 As Integer
Dim cift2 As Integer
Dim satir1 As Integer
Dim satir2 As Integer

```

```

Dim pair1(54) As Integer
Dim pair2(54) As Integer
Dim rassal As Double
Dim cpoint As Integer
Dim offspring1(54) As Integer
Dim offspring2(54) As Integer
Dim sayac As Integer
Dim rassal2 As Double 'crossover probability
satir1 = 25
satir2 = 26
For i = 1 To 10
cift1 = Sheets("eşleşme").Cells(satir1, 57).Value
cift2 = Sheets("eşleşme").Cells(satir2, 57).Value
For sutun = 1 To 54
pair1(sutun) = Sheets("eşleşme").Cells(cift1, sutun).Value
pair2(sutun) = Sheets("eşleşme").Cells(cift2, sutun).Value
Next sutun
rassal2 = Rnd
If (rassal2 <= 0.9) Then
    rassal = Rnd
    If (rassal <= 1 / 53) Then
        cpoint = 1
    ElseIf ((rassal > 1 / 53) And (rassal <= 2 / 53)) Then
        cpoint = 2
    ElseIf ((rassal > 2 / 53) And (rassal <= 3 / 53)) Then
        cpoint = 3
    ElseIf ((rassal > 3 / 53) And (rassal <= 4 / 53)) Then
        cpoint = 4
    ElseIf ((rassal > 4 / 53) And (rassal <= 5 / 53)) Then
        cpoint = 5
    ElseIf ((rassal > 5 / 53) And (rassal <= 6 / 53)) Then
        cpoint = 6
    ElseIf ((rassal > 6 / 53) And (rassal <= 7 / 53)) Then
        cpoint = 7
    ElseIf ((rassal > 7 / 53) And (rassal <= 8 / 53)) Then
        cpoint = 8
    ElseIf ((rassal > 8 / 53) And (rassal <= 9 / 53)) Then
        cpoint = 9
    ElseIf ((rassal > 9 / 53) And (rassal <= 10 / 53)) Then
        cpoint = 10
    ElseIf ((rassal > 10 / 53) And (rassal <= 11 / 53)) Then
        cpoint = 11
    ElseIf ((rassal > 11 / 53) And (rassal <= 12 / 53)) Then
        cpoint = 12
    ElseIf ((rassal > 12 / 53) And (rassal <= 13 / 53)) Then
        cpoint = 13
    ElseIf ((rassal > 13 / 53) And (rassal <= 14 / 53)) Then
        cpoint = 14
    ElseIf ((rassal > 14 / 53) And (rassal <= 15 / 53)) Then
        cpoint = 15

```

```

ElseIf ((rassal > 15 / 53) And (rassal <= 16 / 53)) Then
cpoint = 16
ElseIf ((rassal > 16 / 53) And (rassal <= 17 / 53)) Then
cpoint = 17
ElseIf ((rassal > 17 / 53) And (rassal <= 18 / 53)) Then
cpoint = 18
ElseIf ((rassal > 18 / 53) And (rassal <= 19 / 53)) Then
cpoint = 19
ElseIf ((rassal > 19 / 53) And (rassal <= 20 / 53)) Then
cpoint = 20
ElseIf ((rassal > 20 / 53) And (rassal <= 21 / 53)) Then
cpoint = 21
ElseIf ((rassal > 21 / 53) And (rassal <= 22 / 53)) Then
cpoint = 22
ElseIf ((rassal > 22 / 53) And (rassal <= 23 / 53)) Then
cpoint = 23
ElseIf ((rassal > 23 / 53) And (rassal <= 24 / 53)) Then
cpoint = 24
ElseIf ((rassal > 24 / 53) And (rassal <= 25 / 53)) Then
cpoint = 25
ElseIf ((rassal > 25 / 53) And (rassal <= 26 / 53)) Then
cpoint = 26
ElseIf ((rassal > 26 / 53) And (rassal <= 27 / 53)) Then
cpoint = 27
ElseIf ((rassal > 27 / 53) And (rassal <= 28 / 53)) Then
cpoint = 28
ElseIf ((rassal > 28 / 53) And (rassal <= 29 / 53)) Then
cpoint = 29
ElseIf ((rassal > 29 / 53) And (rassal <= 30 / 53)) Then
cpoint = 30
ElseIf ((rassal > 30 / 53) And (rassal <= 31 / 53)) Then
cpoint = 31
ElseIf ((rassal > 31 / 53) And (rassal <= 32 / 53)) Then
cpoint = 32
ElseIf ((rassal > 32 / 53) And (rassal <= 33 / 53)) Then
cpoint = 33
ElseIf ((rassal > 33 / 53) And (rassal <= 34 / 53)) Then
cpoint = 34
ElseIf ((rassal > 34 / 53) And (rassal <= 35 / 53)) Then
cpoint = 35
ElseIf ((rassal > 35 / 53) And (rassal <= 36 / 53)) Then
cpoint = 36
ElseIf ((rassal > 36 / 53) And (rassal <= 37 / 53)) Then
cpoint = 37
ElseIf ((rassal > 37 / 53) And (rassal <= 38 / 53)) Then
cpoint = 38
ElseIf ((rassal > 38 / 53) And (rassal <= 39 / 53)) Then
cpoint = 39
ElseIf ((rassal > 39 / 53) And (rassal <= 40 / 53)) Then
cpoint = 40

```

```

ElseIf ((rassal > 40 / 53) And (rassal <= 41 / 53)) Then
cpoint = 41
ElseIf ((rassal > 41 / 53) And (rassal <= 42 / 53)) Then
cpoint = 42
ElseIf ((rassal > 42 / 53) And (rassal <= 43 / 53)) Then
cpoint = 43
ElseIf ((rassal > 43 / 53) And (rassal <= 44 / 53)) Then
cpoint = 44
ElseIf ((rassal > 44 / 53) And (rassal <= 45 / 53)) Then
cpoint = 45
ElseIf ((rassal > 45 / 53) And (rassal <= 46 / 53)) Then
cpoint = 46
ElseIf ((rassal > 46 / 53) And (rassal <= 47 / 53)) Then
cpoint = 47
ElseIf ((rassal > 47 / 53) And (rassal <= 48 / 53)) Then
cpoint = 48
ElseIf ((rassal > 48 / 53) And (rassal <= 49 / 53)) Then
cpoint = 49
ElseIf ((rassal > 49 / 53) And (rassal <= 50 / 53)) Then
cpoint = 50
ElseIf ((rassal > 50 / 53) And (rassal <= 51 / 53)) Then
cpoint = 51
ElseIf ((rassal > 51 / 53) And (rassal <= 52 / 53)) Then
cpoint = 52
Else
cpoint = 53
End If
End If
For sayac = 1 To cpoint
offspring1(sayac) = pair1(sayac)
offspring2(sayac) = pair2(sayac)
Next sayac
For sayac = cpoint + 1 To 54
offspring1(sayac) = pair2(sayac)
offspring2(sayac) = pair1(sayac)
Next sayac
For sutun = 1 To 54
Sheets("eşleşme").Cells(satir1 + 23, sutun).Value = offspring1(sutun)
Sheets("eşleşme").Cells(satir2 + 23, sutun).Value = offspring2(sutun)
Next sutun
satir1 = satir1 + 2
satir2 = satir2 + 2
Next i
End Function

```

```

Public Function duplicate(satirNo As Integer)
Dim sutun As Integer
Dim sayac As Integer

```

```

For sutun = 1 To 54
For sayac = 1 To sutun - 1
If (Sheets("eşleşme").Cells(satirNo, sutun).Value = Sheets("eşleşme").Cells(satirNo,
sayac).Value) Then
Sheets("eşleşme").Cells(satirNo, sutun).Value = Int(83 * Rnd + 1)
sayac = 1
End If
Next sayac
Next sutun
End Function
Public Function minresult()
Dim satir1 As Integer
Dim satir2 As Integer
Dim sutun As Integer
Dim result As Long
Dim result1 As Long
Dim result2 As Long
Dim sonuc1(54) As Integer
Dim sonuc2(54) As Integer
Dim sonuc(54) As Integer
result1 = Sheets("eşleşme").Cells(2, 55).Value
For sutun = 1 To 54
sonuc1(sutun) = Sheets("eşleşme").Cells(2, sutun).Value
Next sutun
For satir1 = 3 To 21
If (result1 > Sheets("eşleşme").Cells(satir1, 55).Value) Then
result1 = Sheets("eşleşme").Cells(satir1, 55).Value
For sutun = 1 To 54
sonuc1(sutun) = Sheets("eşleşme").Cells(satir1, sutun).Value
Next sutun
End If
Next satir1
result2 = Sheets("eşleşme").Cells(48, 55).Value
For sutun = 1 To 54
sonuc2(sutun) = Sheets("eşleşme").Cells(48, sutun).Value
Next sutun
For satir2 = 49 To 67
If (result2 > Sheets("eşleşme").Cells(satir2, 55).Value) Then
result2 = Sheets("eşleşme").Cells(satir2, 55).Value
For sutun = 1 To 54
sonuc2(sutun) = Sheets("eşleşme").Cells(satir2, sutun).Value
Next sutun
End If
Next satir2
If (result1 > result2) Then
result = result2
For sutun = 1 To 54
sonuc(sutun) = sonuc2(sutun)
Next sutun
Else

```

```

result = result1
For sutun = 1 To 54
sonuc(sutun) = sonuc1(sutun)
Next sutun
End If
Sheets("eşleşme").Cells(71, 55).Value = result
For sutun = 1 To 54
Sheets("eşleşme").Cells(71, sutun).Value = sonuc(sutun)
Next sutun
End Function
-----
Public Function kopyala()
Dim satir As Integer
Dim sutun As Integer
For satir = 48 To 67
For sutun = 1 To 55
Sheets("eşleşme").Cells(satir - 46, sutun).Value = Sheets("eşleşme").Cells(satir,
sutun).Value
Next sutun
Next satir
End Function
-----
Private Sub CommandButton1_Click()
Dim satir As Integer
Dim sutun As Integer
Dim sayac2 As Integer
Dim mutprob As Double
Dim rassal3 As Double
Dim iteration As Integer
Dim sonuc As Long
Dim result(54) As Integer
mutprob = 0.001
For sayac2 = 2 To 21
Call generation(sayac2)
Next sayac2
For satir = 2 To 21
If (feasibility(satir) = True) Then
Sheets("eşleşme").Cells(satir, 55) = "ok"
Else
Do While (feasibility(satir) = False)
Call generation(satir)
Loop
End If
Sheets("eşleşme").Cells(satir, 55) = "ok"
Next satir
For satir = 2 To 21
Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call reproduction
Call pairs

```

```

Call crossover
For satir = 48 To 67
For sutun = 1 To 54
rassal3 = Rnd
If (rassal3 <= mutprob) Then
Call mutation(satir, sutun)
End If
Next sutun
Next satir
For satir = 48 To 67
Call duplicate(satir)
Next satir
For satir = 48 To 67
If (feasibility(satir) = True) Then
  Sheets("eşleşme").Cells(satir, 56) = "ok"
Else
  rassal3 = Int(19 * Rnd + 2)
  For sutun = 1 To 10
    Sheets("Eşleşme").Cells(satir, sutun).Value = Sheets("eşleşme").Cells(rassal3,
sutun).Value
  Next sutun
End If
Next satir
For satir = 48 To 67
Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call minresult
sonuc = Sheets("eşleşme").Cells(71, 55).Value
For sutun = 1 To 54
result(sutun) = Sheets("eşleşme").Cells(71, sutun).Value
Next sutun
Sheets("eşleşme").Cells(74, 55).Value = sonuc
For sutun = 1 To 54
Sheets("eşleşme").Cells(74, sutun).Value = result(sutun)
Next sutun
For iteration = 1 To 100 'iterations
Call kopyala
Call reproduction
Call pairs
Call crossover
For satir = 48 To 67
For sutun = 1 To 54
rassal3 = Rnd
If (rassal3 <= mutprob) Then
Call mutation(satir, sutun)
End If
Next sutun
Next satir
For satir = 48 To 67
Call duplicate(satir)

```

```

Next satir
For satir = 48 To 67
If (feasibility(satir) = True) Then
    Sheets("eşleşme").Cells(satir, 56) = "ok"
Else
    rassal3 = Int(19 * Rnd + 2)
    For sutun = 1 To 54
        Sheets("Eşleşme").Cells(satir, sutun).Value = Sheets("eşleşme").Cells(rassal3,
sutun).Value
    Next sutun
End If
Next satir
For satir = 48 To 67
    Sheets("eşleşme").Cells(satir, 55).Value = fitness(satir)
Next satir
Call minresult
If (Sheets("eşleşme").Cells(71, 55).Value < Sheets("eşleşme").Cells(74, 55).Value)
Then
    Sheets("eşleşme").Cells(74, 55).Value = Sheets("eşleşme").Cells(71, 55).Value
    For sutun = 1 To 54
        Sheets("eşleşme").Cells(74, sutun).Value = Sheets("eşleşme").Cells(71, sutun).Value
    Next sutun
End If
Next iteration
End Sub

```

EK F – MİNİMUM KARELER YÖNTEMİ SONUÇLARI**Tablo F.1: Şirketlere Atanan Öğrenciler(Minimum Kareler)**

	Şirket	100 İterasyon	500 İterasyon
1	Ekol Ofset Matbaacılık	Belgin Koç	Belgin Koç
2	Pharmavision	Çiğdem Zaman	Çiğdem Zaman
3	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Aybike Eskin	Aybike Eskin
4	Marshall Boya ve Verniksan A.Ş.	Hanife Selen Akyol	Özkan Işıl
5	Marshall Boya ve Verniksan A.Ş.	Yasin Akyıldız	Yasin Akyıldız
6	Sem Metal Plastik	Tolga Maden	Tolga Maden
7	Torgem Gemi İnşaat	Ali Erdem	Ali Erdem
8	Denet Cıvata	Dinçer Kara	Eren Yeşil
9	Erge Elektrik Elektronik Sanayi	Gamze Akıncı	Gamze Akıncı
10	Marshall	Ahmet Boğ	Ahmet Boğ
11	Marshall	Deniz Bahar	Deniz Bahar
12	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Onur Akça	Onur Akça
13	Gimsan Bağlantı Elemanları	Kerem Ödev	Kerem Ödev
14	Yıldız Cam A.Ş.	Kanarya İsmail Daud	Kanarya İsmail Daud
15	Yılteks Yıkama Sanayi	Adem Kazaz	Adem Kazaz
16	Demo Dekorasyon Mobilya Ağaç Pazarlama	Gizem Karlılar	Gizem Karlılar
17	İsfalt A.Ş.	Deniz Aslan	Deniz Aslan
18	Etem Triko Konfeksiyon A.Ş.	Eren Sevim	Eren Sevim
19	Isı Sanayi A.Ş.	Alper Hasboyacı	Alper Hasboyacı
20	Önaysan Metal Sanayi	Murat Kurt	Murat Kurt
21	Sarkuysan Elektronik	Melih Nakaş	Melih Nakaş
22	Finansbank	Cahit Sinan Hamarat	Cahit Sinan Hamarat
23	Fita Teknik LTD.	Serhat Yalçinkaya	Serhat Yalçinkaya
24	Çağrı End. A.Ş.	Kerim Karakuş	Kerim Karakuş
25	Marshall Boya	Zafer Eren	Zafer Eren
26	Betek Boya ve Kimya	Cengiz Günaydın	Cengiz Günaydın
27	Bilgi Elektronik	Özgür Aykul	Özgür Aykul
28	Siemens San. Ve Tic. A.Ş.	Nejdet Yılmaz	Nejdet Yılmaz
29	Kalealtınay Robotik ve Otomasyon A.Ş.	Ayşe Yücel	Ayşe Yücel
30	GES Elektronik	Mehmet Emin İşbilen	Sermin Berberoğlu
31	Zorlu Tekstil	Gökmen Çalışan	Gökmen Çalışan
32	Doğan Ofset A.Ş.	Onur Kurtçu	Onur Kurtçu
33	İntema İnşaat ve Tesisat Malzemeleri	Erdal Aydın	Erdal Aydın
34	Rekor Kauçuk	Hüseyin Cengiz	Hüseyin Cengiz
35	Fita Teknik LTD.	Övgü Tunç	Övgü Tunç
36	İkiler Reklam Aydınlatma	Aytekin Bal	Aytekin Bal
37	Marshall Boya ve Verniksan A.Ş.	Gökhan Eğilmez	Gökhan Eğilmez
38	Trakya Cam ve Sanayi A.Ş.	İbrahim Taştekin	İbrahim Taştekin
39	Nova Reklamcılık ve Sanayi A.Ş.	Tarık Mete	Tarık Mete
40	Öz Yüksel Deri ve Sanayi A.Ş.	Sadık Dadaloğlu	Sadık Dadaloğlu
41	Halkalı Kağıt Karton San. A.Ş.	Çiğdem Atazülal	Çiğdem Atazülal
42	Rekor Kauçuk	Hakan Bozoğlu	Hakan Bozoğlu
43	Koleksiyon Mobilya	Hikmet Çakırtaş	Hikmet Çakırtaş

44	Gerboy Tekstil ve Tic. Sanayi A.Ş.	Gülver Karamemiş	Gülver Karamemiş
45	Travertine Bros	Mustafa Caferoğlu	Mustafa Caferoğlu
46	İkizler Aydınlatma Armatürleri	Aslı Çağla Yıldırım	Aslı Çağla Yıldırım
47	Tat Konserve Sanayi	Burak Öç	Burak Öç
48	Ses Plastik	Umut Albayrak	Umut Albayrak
49	Gemak Gemi İnşaat	Tezcan Mutlu	Tezcan Mutlu
50	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Figen Şam	Figen Şam
51	Sözer Kuyumculuk A.Ş.	Hasan Ber Aymete	Hasan Ber Aymete
52	Ford Otomotiv Sanayi A.Ş.	Duygu Demir	Duygu Demir
53	Kalealtınay Robotik ve Otomasyon A.Ş.	Bahadır Duman	Bahadır Duman
54	Akçalı Boya Sanayi	Engin Güzel	Engin Güzel

EK G - SAATY UYGUNLUK İNDEKSİ SONUÇLARI**Tablo G.1: Şirketlere Atanan Öğrenciler (Saaty Uygunluk İndeksi)**

	Şirket	Çözüm 1	Çözüm 2
1	Ekol Ofset Matbaacılık	Gökhan Zengin	Onur Akça
2	Pharmavision	Onur Kurtçu	Çiğdem Zaman
3	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Faruk Edik	Müge Gök
4	Marshall Boya ve Verniksan A.Ş.	Figen Şam	Özgür Aykul
5	Marshall Boya ve Verniksan A.Ş.	Ahmet Boğ	Belgin Koç
6	Sem Metal Plastik	Mustafa Bayram	Erdal Aydın
7	Torgem Gemi İnşaat	Çiğdem Zaman	Cengiz Günaydın
8	Denet Cıvata	Adem Kazaz	Duygu Demir
9	Erge Elektrik Elektronik Sanayi	Onur Özdemir	Alper Hasboyacı
10	Marshall	Nazmi Ziya Özgür	Yasin Akyıldız
11	Marshall	Hanife Selen Akyol	Neslihan Gül Mençi
12	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Sibel Yılmaz	Kanarya İsmail Daud
13	Gimsan Bağlantı Elemanları	Tarık Mete	Mustafa Bayram
14	Yıldız Cam A.Ş.	Duygu Demir	Umut Albayrak
15	Yılteks Yıkama Sanayi	Emre İstanbullu	Gökmen Çalışan
16	Demo Dekorasyon Mobilya Ağaç Pazarlama	Bahadır Duman	Mustafa Caferoğlu
17	İsfalt A.Ş.	Furkan Aşkın	Gülver Karamemiş
18	Etem Triko Konfeksiyon A.Ş.	Kerem Ödev	Ali Erdem
19	Isı Sanayi A.Ş.	Umut Albayrak	Ayşe Yücel
20	Önaysan Metal Sanayi	Tezcan Mutlu	Aytekin Bal
21	Sarkuysan Elektronik	Mustafa Bayram	Figen Şam
22	Finansbank	Hüseyin Emre Yüksel	Nejdet Yılmaz
23	Fita Teknik LTD.	Yasin Akyıldız	Furkan Aşkın
24	Çağrı End. A.Ş.	Gamze Akıncı	Myrat Amangeldiyew
25	Marshall Boya	Cahit Sinan Hamarat	Ali Güncan
26	Betek Boya ve Kimya	Övgü Tunç	Onur Özdemir
27	Bilgi Elektronik	Aytekin Bal	Salih Çiçek
28	Siemens San. Ve Tic. A.Ş.	Övgü Tunç	Yavuz Selim Torbaoğlu
29	Kalealtınay Robotik ve Otomasyon A.Ş.	Belgin Koç	Tezcan Mutlu
30	GES Elektronik	Neslihan Gül Mençi	Çağdaş Atalay
31	Zorlu Tekstil	Müge Gök	Mehmet Emin İşbilen
32	Doğan Ofset A.Ş.	Çiğdem Atazülal	Hakan Bozoğlu
33	İntema İnşaat ve Tesisat Malzemeleri	Sadık Dadaloğlu	Onur Özdemir
34	Rekor Kauçuk	Zafer Eren	Zafer Eren
35	Fita Teknik LTD.	Muhammed Erdoğan	Sibel Yılmaz
36	İkiler Reklam Aydınlatma	Aybike Eksin	Aybike Eksin
37	Marshall Boya ve Verniksan A.Ş.	Mustafa Caferoğlu	Kerim Karakuş
38	Trakya Cam ve Sanayi A.Ş.	Özkan Işıl	Cahit Sinan

			Hamarat
39	Nova Reklamcılık ve Sanayi A.Ş.	Armağan Bayram	Hikmet Çakırtaş
40	Öz Yüksel Deri ve Sanayi A.Ş.	Eren Yeşil	Eren Yeşil
41	Halkalı Kağıt Karton San. A.Ş.	Serhat Yalçinkaya	Sermin Berberoğlu
42	Rekor Kauçuk	Hande Alaca	Canan Herdem
43	Koleksiyon Mobilya	Nejdet Yılmaz	Yalçın Kayacık
44	Gerboy Tekstil ve Tic. Sanayi A.Ş.	Hakan Bozoğlu	Hanife Selen Akyol
45	Travertine Bros	Hasan Berk Aymete	Aslı Çağla Yıldırım
46	İkizler Aydınlatma Armatürleri	Canan Herdem	Bahadır Duman
47	Tat Konserve Sanayi	Salih Çiçek	Murat Kurt
48	Ses Plastik	Muhammet Kaplan	Hüseyin Cengiz
49	Gemak Gemi İnşaat	Gülver Karamemiş	Metin Balcı
50	Arçelik A.Ş. Çamaşır Makinesi İşletmeleri	Çiğdem Kadaifçi	Çiğdem Kadaifçi
51	Sözer Kuyumculuk A.Ş.	Deniz Aslan	Burak Öç
52	Ford Otomotiv Sanayi A.Ş.	Ali Güncan	Muhammed Erdoğan
53	Kalealtınay Robotik ve Otomasyon A.Ş.	Sermin Berberoğlu	Övgü Tunç
54	Akçalı Boya Sanayi	İbrahim Taştekin	Sadık Dadaloğlu

ÖZGEÇMİŞ

Ayça ALTAY, 18 Ağustos 1983 tarihinde Bergama’da doğdu. 2000 yılında İzmir Atakent Anadolu Lisesi’nden mezun oldu. Aynı yıl İTÜ İşletme Fakültesi Endüstri Mühendisliği Bölümü’ne girdi. Lisans eğitimini Haziran 2005’te tamamladıktan sonra İTÜ Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Yüksek Lisans programında eğitim görmeye başladı. Aynı yılın Aralık ayında halen görev yapmakta olduğu Yöneylem Araştırması Anabilim Dalı’nda araştırma görevlisi olarak çalışmaya başladı. İleri derecede İngilizce ve başlangıç derecesinde Almanca bilmektedir.