T. C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SU DAĞITIM ŞEBEKESİNİN OPTİMİZASYONUNDA GENETİK ALGORİTMANIN KULLANILMASI

Tezi Hazırlayan
Ahmet LOĞOĞLU

Tezi Yöneten
Yrd. Doç. Dr. Nurhan KARABOĞA

Elektronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Ağustos 2006
KAYSERİ

Yrd. Doç. Dr. Nurhan KARABOĞA danışmanlığında **Ahmet LOĞOĞLU** tarafından hazırlanan **"Su Dağıtım Şebekesinin Optimizasyonunda Genetik Algoritmanın Kullanılması"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

04 / 08 / 2006

JÜRİ:

Başkan: Prof. Dr. Mustafa ALÇI

Üye : Doç. Dr. Mehmet Emin YÜKSEL

Üye : Yrd. Doç. Dr. Nurhan KARABOĞA

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 15/08/2006 tarih ve 2006/21-20 sayılı kararı ile onaylanmıştır.

29/08./2006.

Prof. Dr. Nusret AYYILDIZ

Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım sırasında yardımlarını esirgemeyen ve tez danışmanlığımı yürüten hocam Sayın Yrd. Doç. Dr. Nurhan Karaboğa' ya; Sayın Prof. Dr. Derviş Karaboğa' ya lisans ve yüksek lisans eğitimim sırasında emeği geçen hocalarıma; özellikle tezin bilgisayarla desteklenmesi aşamasında yardımda bulunan hocam Sayın Yrd. Doç. Dr. Aytekin Bağış, Sayın Yrd. Doç. Dr. İbrahim Develi'ye ve tüm hayatım boyunca hiçbir konuda desteğini esirgemeyen aileme teşekkürlerimi sunarım.

SU DAĞITIM ŞEBEKESİNİN OPTİMİZASYONUNDA GENETİK ALGORİTMANIN KULLANILMASI

Ahmet LOĞOĞLU
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ağustos 2006
Tez Danışmanı: Yrd. Doç. Dr. Nurhan KARABOĞA

ÖZET

Algoritmalar, matematik ifadeler veya belli kurallarla tanımlanabilen problemleri çözmek için kullanılan yöntemlerdir. Ayrık optimizasyon problemlerinin klasik algoritmalarla çözülmesi oldukça zordur. Bunun için, bu tip problemlerde optimal çözüm üretebilen sezgisel algoritmalara ihtiyaç vardır. Genetik bilimi ve tabii seçme üzerine dayalı olan genetik algoritmalar bu tipten sezgisel algoritmalar olup, farklı ayrık optimizasyon problemlerinde iyi performansa sahiptirler.

Tez çalışması beş bölümden oluşmaktadır: Birinci bölümde; tezin amacı ve literatür çalışması hakkında temel bilgiler verilmiştir.

İkinci bölüm optimizasyon konusu ve genetik algoritmaların optimizasyon problemlerine uygulanmasını içermektedir. Ayrıca seçme, çaprazlama, mutasyon gibi temel genetik algoritma operatörleri ve bunların algoritmayı nasıl etkiledikleri üzerinde durulmuştur.

Üçüncü bölümde; Genetik Algoritma konusu ile ilgili iki farklı örnek incelenmiş ve bilgisayarla analizi gerçekleştirilmiştir.

Dördüncü bölümde; genetik algoritma kullanılarak su dağıtım şebekesinin optimum kontrolü yapılmıştır. Sistemin kontrolü PLC (Programmable Logic Controller) ile gerçekleştirilmiştir.

Son bölümde, sonuçlar verilmiştir.

Anahtar Kelimeler: Genetik Algoritmalar, Optimizasyon, Su Dağıtım Sistemleri.

USING GENETIC ALGORITHM IN THE OPTIMISATION OF WATER DISTRIBUTION NETWORK

Ahmet LOĞOĞLU
Erciyes University, Graduate School of Natural and Applied Sciences
M. Sc. Thesis, August 2006
Thesis Supervisor: Assist. Prof. Dr. Nurhan KARABOĞA

ABSTRACT

The algorithms are the methods used for solving the problems which can be defined by mathematical expressions or certain rules. It is difficult to solve the discrete optimisation problems by using classical algorithms. Hence, heuristic algorithms which are able to produce optimal solutions are needed in solving these kind of problems. Genetic Algorithms (GA) based on genetic science and natural selection are one of these heuristic algorithms and have good performances in different discrete optimisation problems.

This thesis contain five chapters. In the first chapter, basic information are given about the aim of the thesis and literature reviews. Second chapter contains the optimisation subject and application of Genetic Algorithms to the optimisation problems. Also, in this chapter basic genetic operators, such as selection, cross-over and mutation, effects on the performance of the algorithm are investigated.

In the third chapter, two different problems about GA are investigated and analysed by computer simulation.

In chapter four, optimum control of the water distribution network is realized. The control of the system is realized by PLC.

In the last chapter, conclusions are given.

Keywords: Genetic Algorithms, Optimisation, Water Distribution Systems.

İÇİNDEKİLER

KABUL VE ONAY II
TEŞEKKÜR III
ÖZET IV
ABSTRACT V
ŞEKİLLER LİSTESİ IX

1. BÖLÜM
SUNUŞ
1.1. Giriş 1
1.2. Literatür Taraması 1
1.3. Tezin Amacı 4

2. BÖLÜM
GENETİK ALGORİTMALAR (GA)
2.1. Giriş 7
2.2. Doğal Evrim 12
2.3. Genetik Algoritmalar 13
2.4. GA'ların Temel Prensipleri 15
2.4.1. Çözümlerin Kromozom Olarak Tanımlanması 15
2.4.2. Uygunluk Fonksiyonu 16
2.4.3. Genetik Operatörler 16
2.4.3.1. Seçme Operatörü 17
2.4.3.2. Çaprazlama Operatörü 18
2.4.3.2.1. Optimizasyon Problemlerinde Kullanılan Çaprazlama Operatörleri 19
2.4.3.2.2. Yeniden Dizilim Yapısına Sahip Çaprazlama Operatörleri 20
2.4.3.2.3. Alternatif Çaprazlama Operatörleri 22
2.4.3.3. Terslendirme 23
2.4.3.4. Mutasyon Operatörü 24
2.4.4. GA Performansının İyileştirilmesi 25

2.4.4.1. Uygunluk Ölçeklemesi 25
2.4.4.2. Kararlı Şekilde Tekrar Üreme 25
2.4.4.3. Operatör Parametreleri 26
2.4.4.4. En İyinin Saklanması 26

3. BÖLÜM
GA UYGULAMALARI

3.1. Giriş 27
3.2. Bir Dizideki 1'lerin Sayısının GA İle Maksimize Edilmesi 27
3.2.1. Rulet Tekerleği Seçme Metodu 28
3.2.2. Her Bir Bireyin Seçilme Olasılığı 28
3.2.3. Mutasyon 30
3.2.4. Bitirme Kriteri 30
3.3. İki Değişkenli Fonksiyonun GA Kullanılarak Optimize Edilmesi 31
3.3.1. Değişkenlerin Tanımlanması 32
3.3.2. Başlangıç Popülasyonunun Oluşturulması 33
3.3.3. İki Değişkenli Fonksiyonda Çaprazlama 36
3.3.4. İki Değişkenli Fonksiyonda Mutasyon 38
3.3.5. Çalışmanın Değerlendirilmesi 41
3.4. İki Değişkenli Fonksiyonun A ve B Katsayılarının GA Kullanılarak Optimize Edilmesi 42
3.4.1. Probleme Genetik Algoritmanın Uyarlanması 42
3.4.2. Problemin Kodlanması 42

4.BÖLÜM
SU DAĞITIM ŞEBEKESİNİN OPTİMİZASYONUNDA GENETİK ALGORİTMANIN KULLANILMASI

4.1. Giriş 45
4.2. Su Dağıtım Şebekelerinin Genetik Algoritma İle Optimizasyonu 45
4.3. Problemin Tanımlanması 46

5.BÖLÜM
SONUÇLAR 58
KAYNAKLAR 60
ÖZGEÇMİŞ 64
EKLER 65

ŞEKİLLER LİSTESİ

Şekil 2.1. Bir Fonksiyonun Bölgesel ve Küresel Minimum Noktaları……..…9
Şekil 2.2. Gradyent Vektörü…….……. 9
Şekil 2.3. Temel Bir GA'nın Akış Diyagramı……..… 14
Şekil 2.4. Rulet Tekerleği Seçme Operatörü ……...…17
Şekil 2.5. Tek Noktalı Çaprazlama…….…….18
Şekil 2.6. Kısmen Uyumlu Çaprazlama……...20
Şekil 2.7. Sıralı Çaprazlama…… 21
Şekil 2.8. Periyodik Çaprazlama……..21
Şekil 2.9. Sıraya Dayalı Çaprazlama……... 22
Şekil 2.10. Konuma Dayalı Çaprazlama……22
Şekil 2.11. İki Noktalı Çaprazlama……23
Şekil 2.12. Düzgün Çaprazlama…….. 23
Şekil 2.13. Kromozomdaki Terslendirme…… 23
Şekil 2.14. Kromozomdaki Mutasyon……. 24
Şekil 3.1. f(x_1 x_2) Fonksiyonunun Grafiği……... 32
Şekil 3.2. Oluşturulan Başlangıç Popülasyonu……33
Şekil 3.3. Popülasyondaki Kromozomların Seçim Olasılığı……... 34
Şekil 3.4. Kromozomlar İçin Kümülatif Olasılıklar…… 34
Şekil 3.5. Kromozomların Uygunluk Değerlerinin Hesaplanması……..35
Şekil 3.6. [0,1] Aralığında Rasgele Üretilen 20 Adet Sayı……..35
Şekil 3.7. Rulet Tekerleği Seçiminden Sonra Oluşan Yeni Popülasyon……..36
Şekil 3.8. Çaprazlama İşlemi İçin [0,1] Aralığında Üretilen Rasgele Sayılar….36
Şekil 3.9. Çaprazlama İşleminden Sonra Oluşan Yeni Popülasyon…… 38
Şekil 3.10. Mutasyona Uğrayan Bitlerden Sonra Oluşan Yeni Popülasyon.....39
Şekil 3.11. 1000 Jenerasyondan Sonra Popülasyonun Durumu…….. 39
Şekil 3.12. Kromozomların Uygunluk Değerlerinin Hesaplanması…….40
Şekil 3.13. Jenerasyon Sonrası Oluşan Popülasyondaki Kromozomların Uygunluk Değerleri……... 41
Şekil 3.14. Probleme Uygulanan Genetik Algoritmanın Akış Diyagramı…….. 43
Şekil 4.1. Su Talebi ve Maliyet Arasındaki İlişki……..…. 47

Şekil 4.2. Sistemdeki Tanka ait Başlangıç Değerleri ve Sınırlamalar……48

Şekil 4.3. Su Şebekesi Dağıtım Sistemi Modeli…….……48

Şekil 4.4. 10 Bitlik Dizinin İki Tane Tetikleme Seviyesine Dönüştürülmesi…. 51

Şekil 4.5. Osmaniye Fakıuşağı Mevkii Su Kaynakları……51

Şekil 4.6. Osmaniye Fakıuşağı Mevkii Kuyu Suları Dağıtım Ünitesi…….52

Şekil 4.7. Su Ünitesinin Kontrol Sistemi…….53

Şekil 4.8. Şamandıranın İç Yapısı, Alt Ve Üst Tetikleme Noktaları……...53

Şekil 4.9. Optimizasyonda Kullanılan Epanet Programı……54

Şekil 4.10. Osmaniye İli Fakıuşağı Mevkii Su Şebekesinin Programa Tanıtılması……... 55

Şekil 4.11. Temiz Su Depolama Tankında Var Olan Pompalama Tetikleme Seviyeleri…….…..... 56

Şekil 4.12. Şebekede Kullanılan Elemanların Özeti…..……56

Şekil 4.13. Üretilen ve Tüketilen Su Miktarı..……. 57

1. BÖLÜM

SUNUŞ

1.1. Giriş

Bu tezin ilk bölümünde optimizasyon konusu ele alınmış, ikinci ve üçüncü bölümünde sırasıyla Genetik Algoritma (GA)'lar anlatılarak iki örnek fonksiyon üzerinde, GA optimizasyonu ile ilgili temel adımlar detaylı bir şekilde açıklanmıştır. Tezin dördüncü bölümünde, GA ile Osmaniye İli Fakıuşağı Mevkii su kaynaklarının dağıtımının simülasyonu (modelleme) gerçekleştirilmiştir. Çalışılan sistemin uygulandığı bölge, iki kuyuya sahip bir alanda, bir adet ana depo ve bir adet de yedek depodan oluşmaktadır. Birçok su filtreleme işleminde temiz su depolarından (CWS-Clear Water Storage) pompalar yardımıyla su şebekeye dağıtılır. Başlama ve bitiş durumlarındaki pompaların kontrolü, genellikle yüksek ve düşük su seviyelerine bağlı olan tetikleme seviyeleri yardımıyla başarılır. Depodaki temiz su PLC (Programmable Logic Controller-Programlanabilen Mantık Kontrolcüleri) ile su tüketimine göre şebekeye dağıtılır [1,2].

1.2. Konu İle İlgili Literatürdeki Çalışmalar

Algoritmalar, matematik ifadeler veya belli kurallarla tanımlanan problemleri çözmek için kullanılan yöntemlerdir. Algoritmaların tamamı optimum çözümü bulmayı garanti edemezler. Bunun yerine optimuma yakın olan çözümü bulmayı garanti eden algoritmalar sezgisel (heuristic) algoritmalar olarak adlandırılırlar.

Bu tezde ele alınan algoritma, oldukça zor ve karmaşık sayılabilecek bir probleme çözüm aramaktadır. Programlama, problemlerin doğası, uygun ve kabul edilebilir çözümlerin bulunması için dikkate alınması gereken birçok zorluğu içermektedir. Bu

zorlukların ve bunların etkilerinin matematik yolla ifadesi her zaman mümkün olmamaktadır. Böyle problemlerin çözümü kesin bir algoritma kullanılarak yapılamamaktadır. Buna ek olarak, zorluklar genelde özeldir ve bu yüzden genel teknikler çözüm için oldukça yetersizdir. Herhangi bir gerçek problemdeki olası çözümlerin sayısı çok fazladır. Araştırma uzayının geniş sınırları ve karmaşıklığı programlamadaki problemlerin önemli bir nedenidir. Programlama problemlerinin esası, klasik metotların etkin olmadığı önerisine dayanır. Tepeye Tırmanma (Hill-Climbing) algoritması gibi hesaplamaya dayalı popüler araştırma teknikleri programlama için yeterli değildir. Tepeye Tırmanma algoritması hesaplamaya dayalı bölgesel bir araştırma tekniğidir. Araştırma uzayı genişlediğinde programlama sırasında böyle metodlar kolaylıkla bölgesel optimali yakalar. Bunun sonucunda bunlar diğer daha iyi çözümleri kaçırırlar. Yani bu tür algoritmalar, karmaşık bir sistemin sadece bir kısmını optimize etmek için uygundurlar. Bu nedenle böyle metodlar geniş araştırma uzayında kullanım için uygun değildirler. Çünkü tüm çözümlerin değerlendirilmesi oldukça uzun sürecektir. Dolayısıyla optimum çözümün bulunması oldukça uzun bir süre gerektirecektir. Rastgele (random) araştırma algoritmaları da benzer problemleri sergiler. Bunlar iyi çözümlere doğru bir yol izlemediklerinden çözümün bulunmasında problem uzayının çoğu araştırılmalıdır. Sonuç olarak, bunlar aynı zamanda verimsiz tekniklerdir [3].

İçme suyu şebekelerinin boyutlandırılması ve işletilmesinde 1960'lı yılların ortalarından beri yapılan optimizasyon çalışmaları, yöntemlerin karmaşık olması ve gerçek sistemlerde uygulanmasının teknik açıdan zor olması sebebi ile teori olmaktan öteye gidilememiştir. Giderek büyüyen su dağıtım şebekelerinin ilk yatırım ve işletme maliyetlerinin artması optimizasyon ihtiyacını da önemli ölçüde artırmıştır. Çok gözlü ağ şebekelerinin çözümü doğrusal olmayan denklem takımlarının çözümü ile sağlanabilmektedir. Bu çözüm, enerji ve süreklilik denklemleri ile yük kaybı fonksiyonlarını kapsamaktadır. Uzun yıllar boyunca sadece bir kararlı haldeki hidrolik çözümü sağlayan yöntemler geliştirilmiştir [4]. Şebeke hesaplarının güçlüğü nedeniyle su dağıtım şebekelerinin yalnızca ilk yatırım maliyetlerinin ekonomik olması sağlanmaya çalışılmış, gelecekteki işletme maliyetleri ve şebekenin genişlemesi durumları göz önüne alınmamıştır. Birçok araştırmacı boru çaplarının gerekli basınç ve

debiyi sağlayacak biçimde seçilmesi ile tüm sistemin maliyetinin minimum olacağını düşünmüştür [5].

Genetik algoritmadaki seçme, çaprazlama, mutasyon, elitizm gibi operatörler bir başlangıç popülasyonuna uygulanarak her jenerasyonda daha iyi çözüme doğru gidilmektedir. Bu teknikle optimizasyon konusunda birçok deneyimli mühendisin kullandığı sınırlı sayıda deneme yanılmalar ile şebekenin iyileştirilmesi yöntemi elimine edilmiştir. GA tekniğinin birçok uygulamada deneyimli mühendislerden maliyet açısından daha ucuz bir çözüme ulaştığı görülmüştür [6]. Avustralya, Kanada ve Amerika'nın çeşitli eyaletlerinde GA tekniği gerçek su dağıtım şebekelerinin tasarımında kullanılmış ve bu tekniğin optimum çözümü bulmada etkili olduğu kanıtlanmıştır [7]. Bütün bu avantajların yanında genetik algoritmalar diğer optimizasyon tekniklerine göre daha etkili ve uygulanışı bakımından daha kolaydır.

Su dağıtım şebekelerinin optimum tasarımında daha önceden birçok araştırmacının çözüm yapmış olduğu 8 borulu (iki gözlü) bir şebeke ile 34 borulu (üç gözlü) Hanoi kenti su dağıtım şebekesinin çözümünde GA kullanılmıştır [8]. Genetik algoritma parametre değerlerinin belirlenmesi için kesin bir yol olmamakla birlikte bu parametrelerin belirlenmesi deneme-yanılma yöntemleri ile oluşan tecrübeye bağlıdır. Çözüme ulaşmayı etkilediğinden ve problemin özelliklerine göre değişken olacağından bu değerlerin belirlenmesi literatürdeki birçok programda kullanıcıya bırakılmıştır. Bu çalışmada genetik kodlama için kodlama tiplerinden tamsayı kodlama tipi seçilmiştir. Popülasyon sayısı literatürde 100-300 değerleri arasında alınmaktadır. Örneklerde karar değişkenlerinin ve çözüm uzayının çok az olması sonucu tecrübe ile elde edinilen 30 ve 70 kişilik popülasyonlar bu problemler için yeterli olmaktadır. Bu sayılar küçüldükçe programın çalışma süresi azalmaktadır. Literatüre baktığımızda çaprazlama olasılığı için %100'e yakın değerler alınmış; yani popülasyondaki bütün kromozomlar çaprazlama işlemine tabi tutulmuş ve mutasyon olasılığı için %1 civarı değerler alınmıştır [8]. Bu çalışmada da benzer yaklaşımlar uygulanmıştır. Programda sözü geçen genetik çalışma bir popülasyonun kaç tane jenerasyonla sonlandırılacağını, diğer bir deyişle jenerasyon sayısını, genetik çalışma sayısı ise bu popülasyonların kaç kez oluşturulacağını ifade etmektedir. Genetik çalışma sayısı arttırılarak rastgele sayı üreticisinin ürettiği farklı sayılarla

değişik başlangıç popülasyonları üretilmiş olur. Bu sayede çözüm uzayının farklı bölümleri taranarak küresel optimum çözümü bulma olasılığı artmış olur. Sonlandırma fonksiyonu şebekede istenilen basınç ve hız değerlerinin sağlanabilmesi amacıyla kullanılmaktadır. Bunun için her boru için minimum hız ve maksimum hız, her düğüm noktası için minimum basınç ve maksimum basınç sınır değerleri girilir. Genetik parametreler uygun olmazsa iyi bir sonuca ulaşılamayabilir. GA parametrelerinin farklı değerleri için yine optimum şebeke çözümüne ulaşmak da mümkündür.

Alperovits ve Shamir tarafından optimizasyon yapmak amacıyla oluşturulan 8 borudan oluşan hayali şebeke, daha sonraları bir çok araştırmacı tarafından defalarca optimizasyon amaçlı olarak test edilmiştir. Sistem 8 boru, 6 düğüm noktası ve bir su haznesinden oluşmaktadır. Her bir boru 1000 m. uzunluğunda olup, Hazen-Williams katsayısı C=130 olarak alınmıştır [9,10].

Vietnam'daki Hanoi kenti su dağıtım şebekesi 32 düğüm noktası, 34 boru, 3 gözden oluşmaktadır. Fujiwara ve Khang bu şebekenin ilk yatırım maliyetini minimum yapan çözümleri vermişlerdir. Daha sonra çeşitli araştırmacılar (Savic ve Walters, 1997; Abebe ve Solomatine, 1998; Cunha ve Sousa, 1999; Eusuff ve Lansey, 2003; Liong ve Atiquzzaman, 2004) Hanoi kenti su dağıtım şebekesi üzerinde çalışmışlardır [11–15].

1.3. Tezin Amacı

Bu tezin amacı evrimsel gelişim algoritmalarından olan GA'ların su filtreleme işlemindeki pompalama işleminin optimizasyonunda uygulanabilirliğinin gösterilmesidir. Temiz su, kuyulardan depoya, dalgıç pompalar vasıtasıyla çeşitli çaptaki borularla getirilir. Su genellikle, kuyudan yükseklik oranı fazla olan başka bir depoya alınır. Birçok su filtreleme işleminde temiz su depolarından pompalar yardımıyla su şebekeye dağıtılır. Çalışılan sistemin uygulandığı bölge, iki kuyuya sahip bir alanda, bir adet ana depo ve bir adet de yedek depodan oluşmaktadır. Pompalama işleminde esas alınan nokta tetikleme noktalarıdır. Önemli olan bölgedeki su tüketimine göre tetikleme noktalarının belirlenmesidir. Tetikleme noktalarının klasik olarak belirlenmesi tamamen mekanik olarak yapılmakta; yani depodaki su seviyesinin değişmeyen minimum ve maksimum noktalarında şamandıra yardımıyla depolanmış

temiz su şebekeye dağıtılmaktadır. Bu işlem verimli değildir. Genetik Algoritmaların problemin doğası ile ilgili bilgilere ihtiyaç duymaması sebebi ile seviyeleri tespit ederken GA kullanılmıştır. Rasgele oluşturulan bir çözüm popülasyonu ile genetik süreç başlatılır. Genetik Algoritmanın performansı ve kullanılan operatörlerle daha iyi çözümlerden en iyi çözüme ulaşılır. Başlama ve bitiş durumlarındaki pompaların kontrolü, genellikle yüksek ve düşük su seviyelerine bağlı olan tetikleme seviyeleri yardımıyla başarılır. GA kullanılarak yapılan çalışmalar sonucunda alt ve üst tetikleme noktaları optimize edilerek maliyet minimuma indirgenmiştir. İçme suyu şebekelerinin projelendirilmesinde, önce şebekenin geometrisi ve dağıtacağı debiler belirlenir, daha sonra çaplar seçilerek hidrolik benzeşim modeli ile çözüm yapan bir program yardımıyla hız ve basınçlar hesaplanır. Çaplar değiştirilerek şebekenin kritik noktalardaki basınç ve hız değerlerini sağlayacak minimum maliyetli çözüme ulaşıncaya kadar çap değiştirme işlemi yapılır ve şebekedeki pompa çalışma zamanı değiştirilir. Bu tür klasik çözüm yaklaşımlarında optimum çözüm için uzun ve yorucu çalışmalar yapılması gerekmektedir. Bu aşamaların başarı düzeyi büyük ölçüde proje mühendisinin deneyimine bağlıdır. Ayrıca bu konuda en deneyimli mühendis bile global optimum çözümü elde edebileceğini garanti edemez. Genetik algoritmalar ise deneme yanılma aşamasını evrim teorisine dayandırarak, yine bir hidrolik benzeşim modeli yardımı ile aylar sürecek bir çalışmayı saatler mertebesine indirebilmektedir. Üstelik genetik algoritma ile bulunan çözümler proje mühendislerinin bulduğu klasik çözümlere kıyasla çok daha ekonomik ve hidrolik açıdan da daha uygun şebekeler olmaktadır. Bu çalışmada pompalamanın optimizasyonunda klasik teknikler yerine sezgisel tekniklerden Genetik Algoritma kullanılmış, debi sabit kabul edilmiş ve gerçekleştirilen sistemde maliyet azaltılmıştır. Aynı zamanda uygulanan GA modeli ve bilgisayar programının, Fakıuşağı bölgesindeki su şebekesi üzerinde uygulanması sonucu mevcut stratejinin geliştirilebileceği görülmüştür. İçme suyu şebekelerinde tetikleme noktalarının iyileştirilmesi ekonomiye de kazanç sağlayacaktır.

Günümüzde yaygın olarak, otomatik kontrol düzenlerinde kullanılmakta olan mikroişlemci tabanlı endüstriyel otomasyon cihazları olan PLC'ler ikili giriş sinyallerini işleyerek, teknik işlemleri, çalışmaların adımlarını direkt olarak etkileyecek çıkış işaretlerini oluşturur. Çoğunlukla programlanabilir denetleyicilerin yapabileceği işlerde bir sınır yoktur. PLC, bir iş akışındaki bütün adımların doğru zaman ve doğru sıradaki

bir hareket içerisinde olmasını sağladığından bu çalışmada PLC kullanılması tercih edilmiştir.

Su dağıtım şebekesi tasarımını önemli ölçüde etkileyebilen; kişi başına günlük su tüketimi, terfi tesisleri, hazneler, projelendirme debisinin (pik katsayısı) belirlenmesi gibi gerçek işletme koşulları (aşırı su kaçakları, basınç yetersizlikleri)' nın ileride yapılacak optimizasyon çalışmalarında dikkate alınması uygun olacaktır.

2. BÖLÜM

GENETİK ALGORİTMALAR

2.1. Giriş

Matematik bir terim olarak optimizasyon, en iyinin tanımlanması ve çözülmesi anlamına gelir. Optimizasyon teorisi, optimalin, yani en iyinin, sayısal olarak elde edilmesi için kullanılan tüm yöntemleri kapsamaktadır. Yani optimizasyon, bilinen bir fonksiyonun veya amacın belirli kısıtlamalar altında ya da kısıtlamalar olmaksızın optimumunun bulunması demektir. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir. Optimizasyon işleminde ilk adım olarak karar parametreleri veya karar değişkenleri ya da tasarım parametreleri olarak da adlandırılan parametreler setinin tanımlanması gerekir. Sonra, bu parametrelere bağlı olarak enküçük yapılacak (minimize edilecek) bir maliyet fonksiyonu veya enbüyük yapılacak (maksimize edilecek) bir kâr fonksiyonu ve problem ile ilgili sınırlama fonksiyonları tanımlanmalıdır. Maliyet fonksiyonu, daha iyi çözümü temsil eden parametre değerlerinin kullanılması durumunda daha düşük bir sayısal değer üretirken; kâr fonksiyonu ise daha yüksek bir sayısal değer üretmektedir. Sınırlamalar, parametrelerin alamayacağı değerleri tanımlamakta ve karar parametrelerine bağlı olarak ifade edilmektedir. Bazı sınırlamalar eşitsizlikler, bazıları ise eşitlikler biçiminde olabilir.

Maliyet fonksiyonu ve sınırlamalar matematik formda aşağıdaki gibi tanımlanabilir: n değişkenli $x = (x_1, x_2,\ldots x_i,...x_n)$ vektörünü tanımlayalım. Burada x_i, i. parametrenin değerini göstermektedir. Maliyet fonksiyonu f(x),

$$f(x)=f\{ x_1,x_2,\ldots,x_n) \quad (2.1)$$

aşağıdaki gibi tanımlanan p tane eşitlik sınırlamalarına

$$h_j(x)=h_j(x_1,x_2,\ldots,x_n)=0, \quad 1\leq j\leq p \quad (2.2)$$

ve yine aşağıdaki gibi tanımlanan m tane eşitsizlik sınırlamalarına

$$g_i(x)=g_i(x_1,x_2,\ldots,x_n)\leq 0, \quad 1\leq i\leq m \quad (2.3)$$

sahip olabilir. Bazı problemlerde birden fazla maliyet fonksiyonu bulunabilir. Yani, problem birden fazla enküçük yapılacak fonksiyon içerebilir. Bu tür problem, çok amaçlı optimizasyon problemi olarak adlandırılır. Problem için sınırlamaları sağlayan mümkün tüm çözümlerin oluşturduğu bölge, araştırma yapılabilecek uygun (feasible) çözüm bölgesi olarak adlandırılır. Optimum (eniyi) çözüm, enküçük yapılacak problem durumunda uygun bölgede en düşük maliyet değerine sahip çözümken, enbüyük yapılacak problem durumunda ise enbüyük amaç fonksiyon değerine sahip çözümdür. f(x) maliyet fonksiyonu, $x^* = (x_1^*, x_2^*, ..., x_n^*)$ çözümü için Denklem (2.4a) ile verilen şartı uygun bölge (S) içindeki tüm muhtemel çözümler (x) için sağlıyor ise f(x) fonksiyonu x*'de *küresel (global) minimuma* sahiptir denir.

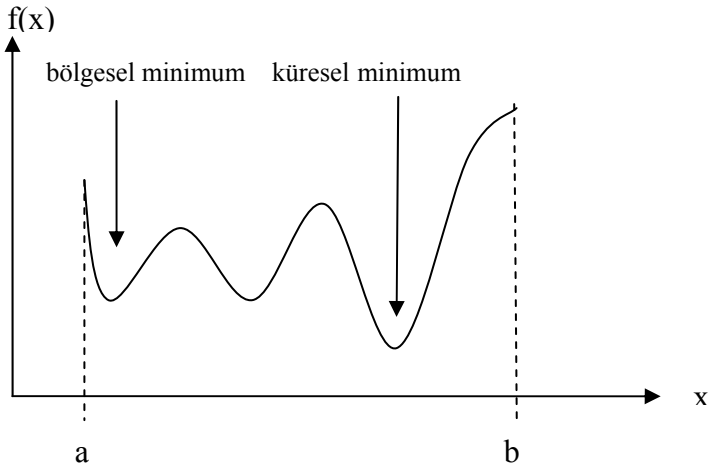
$$f(x^*)\leq f(x) \quad (2.4a)$$

Eğer Denklem (2.4a) uygun bölge içerisinde, x*'in dar bir komşuluğu (N) içindeki tüm muhtemel çözümler için sağlanıyorsa f(x) fonksiyonu x*'de *bölgesel (local) minimuma* sahiptir denir. x*'in komşuluğu (N), küçük bir değere sahip olan δ parametresine bağlı olarak

$$N = \{x|\varepsilon S \text{ ve } \| x-x^* \| <\delta\} \quad (2.4b)$$

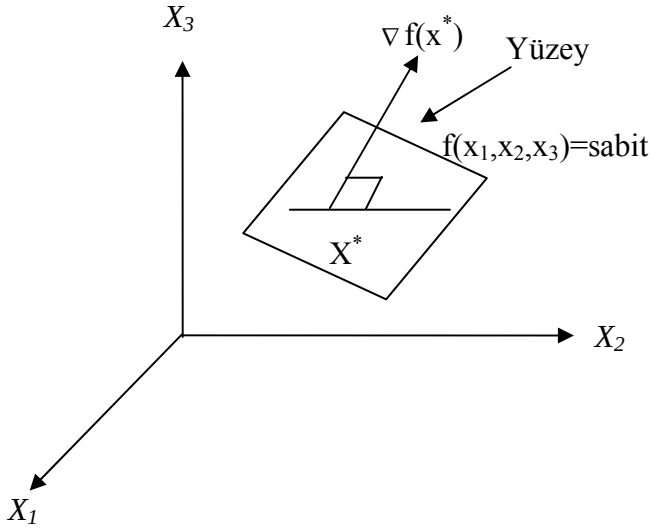
şeklinde tanımlanabilir. Bu, geometrik olarak x* noktasını ihtiva eden küçük uygun bir bölgeyi tanımlamaktadır. Şekil 2.1.'de tek değişkenli f(x) fonksiyonu için bölgesel ve küresel optimum (minimum) noktaları gösterilmektedir.

Optimum noktada sağlanması gereken şartlar, gerek şartlar olarak adlandırılır. Gerek şartları sağlayamayan noktalar optimum olamaz. Ancak, gerek şartların sağlanması tek başına o noktanın optimum olmasını garanti edemez. Bu yüzden gerek şartları sağlayan optimum ve optimum olmayan noktaları ayırt etmek için yeter şartlar kullanılır. Gerek şartlar sağlandıktan sonra yeter şartlar da sağlanıyorsa o noktanın optimum nokta olduğu söylenir.



Şekil 2.1. Bir Fonksiyonun Bölgesel ve Küresel Minimum Noktaları

Üç değişkenli fonksiyon için bu durum Şekil 2.2.'de gösterilmiştir.



Şekil 2.2. Gradyent Vektörü, x* noktasına teğet olan düzleme dik konumdadır ve yönü de fonksiyonun en hızlı artış sağladığı doğrultuyu göstermektedir.

f(x) fonksiyonunun x* noktasındaki x_1 'e göre kısmi türevi $\partial f(x)/\partial x_1 |_{x=x^*}$ şeklinde tanımlanır. Şayet tüm kısmi türevler bir sütun vektörü formunda düzenlenirse, vektör gradyent vektörü olarak adlandırılır ve ∇f veya grad f ile gösterilir. Geometrik olarak gradyent vektörü, x* noktasına teğet olan f(x) = *sabit* düzlemine diktir ve f(x) fonksiyonunun en hızlı artış gösterdiği yönü belirtmektedir. Gradyent vektörünün bu özelliği, klasik gradyente dayalı optimizasyon algoritmalarının temelini teşkil etmektedir.

Gradyent vektörün ikinci kez türevini almak suretiyle klasik optimizasyonda önemli bir yeri olan Hessian matris H elde edilir. Dolayısıyla bu matris, f(x) fonksiyonun ikinci dereceden kısmi türevlerinden oluşmaktadır. Hessian matris, simetrik özelliğe sahiptir ve optimallik için yeterli şartların tayininde önemli rol oynar.

x^*, f(x) fonksiyonunun *bölgesel minimumu* olan bir nokta ve x, x^* 'in yakın civarında diğer bir nokta olsun. Bu durumda, çözümlerin x* civarındaki değişimi d ve bundan dolayı amaç fonksiyonunda ortaya çıkan f(x*) civarındaki değişimi Δf sırasıyla,

$$d = x-x^* \text{ ve } \Delta f=f(x)-f(x^*) \qquad (2.5)$$

olarak tanımlanabilir. x^* *bölgesel minimum* olduğu için yakın civarında, yani küçük d değerleri için aşağıdaki eşitsizliğin sağlanması gerekir:

$$\Delta f=f(x)-f(x^*)\geq 0 \qquad (2.6)$$

Bu eşitsizlik, *bölgesel minimum* için gerek ve yeter şartların türetilmesi amacıyla kullanılır. d oldukça küçük olduğundan Δf, x* noktasında Taylor serisine açılarak optimallik şartları türetilebilir. Tek değişkenli fonksiyon için birinci dereceden gerek şart

$$f'(x^*) =0 \qquad (2.7a)$$

dır. Ancak bu şartı hem maksimumlar hem de minimumlar, yani değişimin sıfır olduğu durağan noktalar sağlamaktadır. Minimumluk için yeter şart

$$f''(x^*)>0 \qquad (2.7b)$$

dır. Eğer, f"(x*)= 0 ise, x*'in bir minimum nokta olmadığı sonucuna varılamaz. Bu durumda daha yüksek dereceden türevlerin hesaplanması gerekir. İkinci dereceden gerek şart ise

$$f''(x^*)\geq 0 \qquad (2.7c)$$

ile tanımlıdır ve f"(x*)<0 şartını sağlayan herhangi bir nokta *bölgesel minimum* olamaz, f"(x*)=0 ise, bu noktanın *bölgesel minimum* olabilmesi için

$$f''' (x^*)=0 \qquad (2.7d)$$

gerek şartı ile

$$f^{iv}(x^*)>0 \qquad (2.7e)$$

yeter şartını sağlaması gerekir. Bu türev sıfıra eşitse yine şüpheli durum söz konusudur ve bir üst dereceden türeve bakılmalıdır. Kısacası çift-derece türev sıfırdan küçükse aday nokta kesinlikle *bölgesel minimum* şartını sağlamıyor demektir ama sıfıra eşitse bir üst tek-dereceli türevin sıfıra eşit ve bir üst çift-derece türevin ise sıfırdan büyük olup olmadığı kontrol edilmelidir [3,16].

Genetik Algoritma (GA) evrim ve genetiğin doğal sürecine dayalı stokastik bir araştırma tekniğidir [17]. Bir problemin iyi çözümleri, olası çözümlerin jenerasyonlarının seçkin üremeleri ile geliştirilir. GA'lar, daha iyi çözümlere doğru yavaş yavaş bir yaklaşımı sağlayan geniş bir problem uzayı boyunca yönlendirilmiş rastgele bir araştırmaya imkân sağlar. Bu algoritma canlılarda bulunan genetik gelişimini simüle etmektedir. Goldberg GA'ların temel avantajının kararlılığı olduğunu kanıtlamıştır [17–18]. Bu, De Jong ve Spears tarafından da doğrulanmıştır [19–20]. GA'lar oldukça geniş ve karmaşık araştırma uzayları ve hatta yüksek boyut, çok modluluk, süreksizlik ve gürültü gibi zorlukların varlığında bile, araştırmada yüksek etkinlik sağlamıştır.

GA'ların doğası, aynı anda bir problem uzayının birçok alanının araştırılmasına yani paralelliğe imkân sağlar. Kullanılan operatörlerin kalıtımsal stokastik doğası araştırma uzayının genel bir incelemesine olanak verir. Bu yüzden bir genetik araştırma algoritmasının bir tepeye tırmanma algoritmasına göre bölgesel bir optimuma geçmiş

olması oldukça düşük bir ihtimaldir. GA'ların temel avantajlarından biri, optimize etmeye çalıştıkları problemin doğası ile ilgili herhangi bir bilgiye ihtiyaç göstermemeleridir. Bunlarla özel bilgi gerekmeksizin bir kısım zor problemlerin çözümü mümkündür. Problemin parametre değerlerinin ve bunların uygunluğunun değerlendirilmesinde sadece bir kodlamaya ihtiyaç duyan GA'lar sezgisel bir fonksiyon kullanılarak çözülür. Bu, araştırma prosedürünün karmaşıklığını azaltır ve diğer araştırma metodlarına göre daha etkin bir yol sağlar. GA'lar birçok problem için optimale yakın çözümlerin bulunmasında başarılı olmuşlardır [21].

Bilgisayar biliminin bir dalı olarak kabul edilen yapay zeka, bilgisayarlara problem çözmede insanlar gibi öğrenme ve düşünme becerisini kazandırmaya çalışır. Yapay zeka teknikleri, deneyerek öğrenme ve öğrendiklerini daha sonraki adımlarda kullanabilme yetenekleri sayesinde optimum çözüme büyük bir hızla yaklaşabilmektedir.

Yapay sinir ağları, canlılarda bulunan sinir ağlarından esinlenilerek bilgisayar üzerinde benzeşim yoluyla gerçekleştirilen öğrenme yeteneğine sahip yapılar olarak tanımlanır. Doğanın insanlar için ne kadar sonsuz bir esin kaynağı olabileceğini bir kez de Michigan Üniversitesi'nden John Holland'ın çalışmaları kanıtlamıştır. Mekanik öğrenme (machine learning) konusunda çalışmalar yapan Holland, canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Doğada çevreye daha iyi uyum sağlayan bireylerin soylarını sürdürebilmesi ve üstün nitelikli ana babadan yine üstün nitelikli kuşakların yetişmesi, hatta yeni kuşakların öncekilerden bile üstün olabilmesi; eldeki problemin bilinen bazı çözümlerinin melezlenmesiyle daha iyi çözümler üretilebileceği fikrini doğurmuştur. Biyolojideki bu doğal seçim kuramının ayrık problemlerin çözümüne de uygulanabileceği düşüncesine dayanan genetik algoritma kavramı 1960 yılında John Holland tarafından geliştirilmiştir. Holland tarafından yazılan "Adaptation in Natural and Artificial Systems (Doğal ve Yapay Sistemlerde Adaptasyon)" adlı kitap GA'lar için teorik bir zemin sağlamıştır [22].

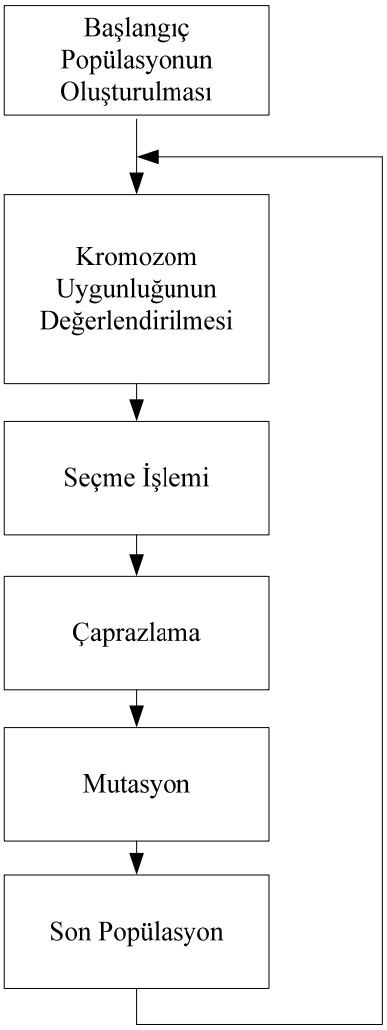
2.2. Doğal Evrim

GA'ların kullanımı incelendiğinde, doğanın temel prensiplerinin burada da olduğu görülebilir. Kalıtım; türlerin yaşadıkları çevrede hayatta kalmaları için uyum

sağlamalarına yol açan gelişime izin verir. Doğada türlerin tabiatı hakkındaki bilgi, onların kromozomlarında kodlanmıştır. Bunlar göz rengi ve ağırlık gibi bireysel özellikler hakkında bilgi içeren bir gen dizisinden oluşur. Bir türün en uygun (fittest) üyeleri soy için oldukça kuvvetli seçilme olasılığına sahiptir. Bu yüzden gelecek jenerasyona bunların kromozomlarda temsil edilen, genetik özelliği geçer. Türün en zayıf üyeleri yaşamak için daha az şansa sahiptir. Bu yüzden sonraki jenerasyona bunların genetik tabiatının ulaşması şansı oldukça düşüktür. Çocuk (offspring) genelde genetik özelliği iki ebeveynden, bunların tekil genetik yapılarının birleşimi olarak, alır. Bu durum, bir popülasyonda varolan ve aktarılan genlerin sayısız kombinasyonunu mümkün kılar. Burada aynı zamanda mutasyonun meydana getirdiği çok küçük bir olasılık da vardır. Mutasyon, çocuk kromozoma ebeveynlerin hiçbirinden bir gen değeri (allele) geçmediği zaman meydana gelir. Bu olay, kromozom yapısında küçük bir farklılığa neden olan rastgele bir değişimdir. Genlerdeki farklılık ve üyelerin özelliklerinin yer aldığı popülasyonun yapısında, üyeler arasında sıkça yeniden yapılanmalar meydana gelir ve yaşam ortamına adapte olabilen türlerin üretiminde başarı sağlanır. Doğal evrimin hızı ve bunun daha iyi türlerin üretimindeki başarısı genetik araştırmanın gücüne bağlıdır.

2.3. Genetik Algoritmalar

Yapay GA'ların genel prensibi, yapay kromozomu ifade eden dizi (string)'ler içine karmaşık problemlerin olası çözümlerinin kodlanması esasına dayanmaktadır. Her bir dizi, özel uygulamaya uygun bir değerlendirme fonksiyonu kullanılarak hesaplanan, bir uygunluk değerine (fitness value) sahiptir. Bu popülasyon seçme (selection), çaprazlama (crossover) ve mutasyon (mutation) gibi bir dizi operasyondan geçer. Bir GA'nın çalışması için ilk adım, başlangıçta kullanılacak olan topluluğun oluşturulmasıdır. Bu noktada topluluk bireyleri rastgele olarak oluşturulur. Ama göz ardı edilmemesi gereken bir seçenek de, eğer biliniyorsa başlangıç bireylerini çözüme yakın olduğu bilinen parametre değerleri ile oluşturarak çözümü hızlandırmaktır. Evrim sürecine girmeden önce dikkat edilmesi gereken bir başka nokta da, başlangıç bireylerini değerlendirmektir. Bu, evrim süreci içerisinde bir sonraki nesilde geliştirilecek olan yeni çözümler için gereklidir. Daha sonra istenilen düzeyde başarılı bir birey bulunana, topluluk ortalama başarıda artış sağlayamaz duruma gelene ya da önceden belirlenen evrim sayısı tamamlanana değin sürecek olan evrim başlayacaktır.



Şekil 2.3. Temel Bir GA'nın Akış Diyagramı

Şekil 2.3.'e göre temel bir GA'nın adımları aşağıdaki gibi verilmektedir.

1. Kromozomlardaki bilgiyi kodla, parametreleri belirle,
2. İlk popülasyonu oluştur,
3. Yeni bireylerin uygunluk (başarı) değerini hesapla,
4. Bu değere bağlı olarak bir çift birey seç (üreme),
5. Bunları, rastgele bir çaprazlama noktasına göre çaprazla,
6. Rastgele bir noktadan mutasyon uygula,
7. Yeni ve daha iyi nüfusla eski kromozomu yer değiştir,
8. Yeterli sayıda çözüm yetiştiyse ya da uygunluk değeri belirlenen değere ulaştı ise dur, yoksa 2'ye dön [23].

Her evrim sürecinde tekrarlanacak olan işlemler de şöyle sıralanacaktır: Daha sonraki çözümleri geliştirecek olan bireyler daha önce hesaplanmış olan başarı değerlerine bağlı olarak seçilir. Daha sonra seçilen bireyler istenilen bir yöntemle çaprazlanırlar. Bu işlem sonucu meydana gelen bireyler belirli bir oranda mutasyona uğratılır. Sonunda meydana gelen yeni topluluğun başarı değeri hesaplanır. Yukarıda söz edilen mutasyon olayına deterministikten çok stokastik olarak karar verilir. Deneysel çalışmalar klasik operatörler üzerinde sayısız değişkenin kullanımını sağlar. Bu operatörler algoritma performansının iyileştirilmesi ya da bunların farklı uygulamalara uyumunun sağlanması için çalışır [24]. Özel bir problem için bir GA'lar, aşağıdaki faktörlere sahip olmalıdır:

- Problemin mevcut çözümleri için genetik bir tanımlama,
- Bu mevcut çözümler için bir başlangıç popülasyonunun oluşturulması,
- Bunların uygunluğuna bağlı olarak çözüm artışını sağlayan ve gelişmede rol oynayan bir değerlendirme fonksiyonu,
- Yeniden üreme esnasında kromozomların dizilimini değiştiren genetik operatörler,

GA' nın kullandığı değişik parametre (popülasyon büyüklüğü, uygulanan genetik operatörlerin olasılıkları vb.) değerleri.

2.4. GA'ların Temel Prensipleri

GA uygulamaları bir dizi parametre ile karakterize edilmektedir. Bunların başlıcalarını; kodlama stratejisi, başlangıç popülasyonu, bireylerin seçimi, çaprazlama mekanizması, mutasyon, popülasyon büyüklüğü, popülasyon yineleme oranı ve elde edilen yeni çözümler için uygulanan strateji oluşturur.

2.4.1. Çözümlerin Kromozom Olarak Tanımlanması

Uygulanan teknikler kromozomların değişik şekilde kodlanmasında kullanılır. Holland 0 ve 1 alfabesinden yararlanarak tek bit dizi kodlamasını ilk olarak kullanmıştır. Holland aynı zamanda Şema Teorisi olarak bilinen ve bir şemanın sürekli kullanımını ve değerlendirilmesini içeren matematik bir analiz de gerçekleştirmiştir [22]. Bu şema, kromozomları oluşturan bitlerin bir örneği olarak tanımlanabilir. Burada genler 0,1 ya da $*$ (önemsiz) sembollerinden biri ile belirtilir. Örneğin ($*1*0****$) dizilimi, bir

şemadır. Sırasıyla 2 ve 4 konumlarında 1 ve 0 değerlerine sahip olan bu şema sürekli kullanılır ve şemadaki her bir yeni oluşum GA'ya yüksek verimli çözümler için güç verir. Diğer kodlama biçimlerinden daha anlaşılır olduğundan bit dizi halâ oldukça popüler bir kodlama metodudur. Bunun yanı sıra kodlama amaçlı olarak sayısal değerler de kullanılabilir. Bu tamamen problemin niteliğine bağlıdır. Örneğin, yeniden sıralanma (re-ordering) problemlerinde sayısal değer kullanımı oldukça yararlı olabilir. Kodlama tipinin seçiminde farklı tasarımcıların yaklaşımı da değişik olacaktır. Doğal terminoloji ile GA karşılaştırıldığında, doğal dünyadaki kromozom, gen ve allele olarak ifade edilen kavramlar, GA'da sırasıyla ikili dizi(binary string), ikili dijit (binary digit) ve ikili değere(binary value) karşılık gelmektedir [17].

Herbir jenerasyondaki kromozom sayısı önem taşımaktadır. Geniş popülasyonlar işlem süresini arttırmasına rağmen bunlarla küçük popülasyonlardan daha iyi sonuçlar elde etmek mümkün olmaktadır. Genellikle 50-100 arasındaki popülasyon büyüklüğü yaygın olarak kullanılmaktadır. Tabii ki, bu büyüklük problemin tipine göre daha az ya da daha fazla olabilir. Bir popülasyonun ilk olarak elde edilmesi rastgele şekilde yapılabilir. Çözüme yönelik herhangi bir bilgi mevcutsa, oluşturma esnasında bu bilgi de kullanılabilir.

2.4.2. Uygunluk Fonksiyonu (Fitness Function)

GA'nın en önemli fonksiyonlarından birisidir. Bir problem için her bir kromozom bir çözümü ifade eder. Değerlendirme fonksiyonu, çözümün kalitesini değerlendirir. Bunun için kromozomun uygunluğunu belirten sayısal bir değer üretir. Bir kromozom topluluğu için bu değerler gelecek jenerasyonun elde edilmesinde yani çözümlerin seçimi açısından önem taşır. Bu fonksiyonun yüksek değerlere sahip olması en uygun dizilerin seçilmiş olması ihtimalini kuvvetlendirir.

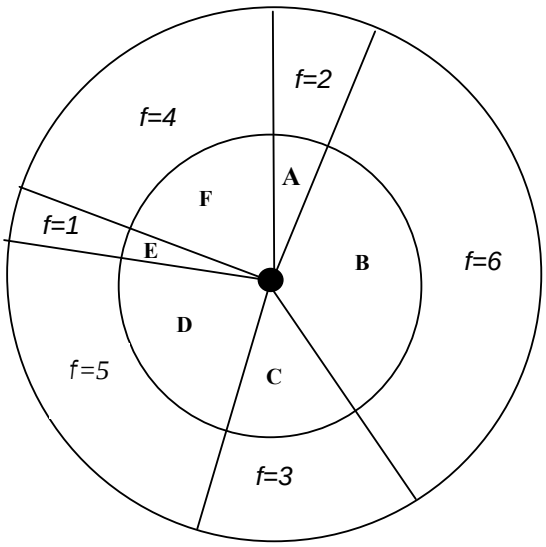
2.4.3. Genetik Operatörler

İkili diziler üzerinde işlem yapmaya yarayan bu operatörler ile ilgili olarak oldukça fazla sayıda çalışmalar yapılmış ve çok sayıda değişik genetik operatörler geliştirilmiştir [21]. Buna rağmen GA'ların çoğunluğu tekrar üreme (reproduction), çaprazlama ve mutasyon operatörlerinin değişik formlarını ihtiva ederler. Bu operatörlerin uygun şekilde seçimi GA performansı için çok önemlidir.

2.4.3.1. Seçme Operatörü

Seçme operatörü, bir popülasyonda çaprazlamaya uğrayan dizilerin seçimi görevini yerine getirir. Bu çözümlerin belirlenmesi için gereklidir. Fakat aynı zamanda çok sayıda tür içinden en iyi dizinin belirlenmesi oldukça zordur. Tekrar üreme operatörü, tabii seçme işlemi olarak adlandırılan ve kaliteli olanların hayatta kalmaları prensibine göre çalışan bir elemandır. Tabiatta bu seçme işlemi çevre tarafından kontrol edilirken yapay sistemlerde ise değerlendirme fonksiyonu ile diğer kalite yani uygunluk değerlendirme işlemleri tarafından kontrol edilir. Bireysel yapılar, bir kuşaktan diğerine geçerken uygunluk değerlerine göre kopya edilerek çoğaltılırlar. Uygunluğu daha yüksek olan çözümlerin diğerlerine göre daha fazla kopya edilme şansları vardır. Bu yüzden daha uygun çözümlerin yoğunluk olarak baskın olmaları, yeterli uygunluğu taşımayanların yani kalitesiz olanların ise yavaş yavaş elenmeleri yani ölmeleri sağlanır.

Bu operatörün uygulanması için kullanılan en basit metod, taraflı rulet tekerleği (biased roulette wheel) tekniğidir [17,25,26]. Rulet tekerleği ebeveyn seçme operatörü, iyi bilinen ebeveyn belirleme metodlarından biridir.



Şekil 2.4. Rulet Tekerleği Seçme Operatörü

Bir sonraki jenerasyon için ebeveynlerin A kromozomlarından F'ye belirlendiği bir örnek Şekil 2.4.'de verilmiştir. İlk olarak her bir kromozomun uygunlukları toplanarak popülasyonun toplam uygunluğu bulunur. Seçilen her ebeveyn için bu prosedür

kullanılır. Sonra her bir kromozom uygunluğunun toplam uygunluğun yüzde kaçını temsil ettiği bulunur. Böylece her biri bir çözümü ifade eden kromozomların uygunlukları oranında bunlara tekerlek üzerinde bir parça verilir. Şekil 2.4.'de tekerlek toplam uygunluk değeri olan 21 eşit parçaya ayrılmış ve her kromozom bu tekerlek üzerinde ağırlıkları (uygunlukları) oranında yerleşmişlerdir. Tekrar üreme için rulet tekerleğinin döndürülmesi gerekir. Bunun için sıfırla toplam uygunluk değeri arasında rastgele bir sayı üretilerek bu sayının tekerleğin hangi parçasına karşılık geldiğine ve dolayısıyla hangi kromozomun seçildiğine bakılır. Artık tekerlek bir kere dönmüştür ve ilk kromozom belirlenmiştir. Benzer şekilde diğer kromozomların da belirlenmesinden sonra uygunluk değerleri en başarılı olan adaylar eşleştirme havuzu (mating pool) olarak adlandırılan bir havuzda toplanarak, diğer genetik operatörlerin uygulanması için hazırlanırlar.

Şekil 2.4.'e göre A-F kromozomları için kromozom uygunluğu f ve toplam uygunluk ise 20'dir. Yani şekildeki daire 21 eşit parçaya ayrılmış ve f=2 uygunluğuna sahip olan A kromozomu bu parçalardan ikisini almıştır. Benzer olarak B kromozomu bu parçalardan altısına, C kromozomu üçüne, D kromozomu beşine, E kromozomu birine ve F kromozomu ise dördüne sahiptir. Şekle göre A'nın 1 ve 2, B'nin 3-8, C'nin 9-11, D'nin 12-16, E'nin 17 ve F'nin 18-21 no' lu parçalara sahip olduğu düşünülürse, 8, 17, 3, 12, 9 ve 19 değerlerine sahip rastgele sayıların üretilmesi durumunda seçilen ebeveynler sırasıyla B, E, B, D, C ve F olacaktır. Kolaylıkla görülebileceği gibi f=6 ile en yüksek uygunluk değerine sahip olan B kromozomunun diğerlerine göre seçilme şansı daha fazla olacaktır.

2.4.3.2. Çaprazlama Operatörü

Çaprazlama operatörü GA' nın çok önemli bir elemanıdır. Bu eleman değişik kromozom dizilimlerinin test edilmesine ve hızlı şekilde tekrar yapılanmasına olanak sağlar. Genelde tek noktalı çaprazlama kullanılmaktadır. Şekil 2.5.'de böyle bir tek noktalı çaprazlama işlemi gösterilmiştir.

1. Kromozom 1 0 1 ¦ 0 1 0 1 0 ⇒ 1. Yeni Kromozom 1 0 1 ¦ 1 0 0 0 0
2. Kromozom 1 1 1 ¦ 1 0 0 0 0 ⇒ 2. Yeni Kromozom 1 1 1 ¦ 0 1 0 1 0

Şekil 2.5. Tek Noktalı Çaprazlama

Bunun için eşleştirme havuzunda bulunan yapıların birer çifti rastgele olarak seçilir ve çaprazlama operatörü bu iki yapıdan yeni iki yapı meydana getirmek için kullanılır. Çaprazlamaya tabi tutulacak iki kromozom üzerinde bir çaprazlama noktası belirlenir. Bu çaprazlama bölgesinde yer alan tüm bitler karşılıklı yer değiştirilerek iki yeni çözüm kromozomu elde edilir. Oluşturulan çözümler eski kromozomlardan tamamen farklı olabilir. Bu yüzden her bir jenerasyonda oldukça farklı çözümler incelenebilir.

Çaprazlama operatörünün amacı, iyi çözümlerden daha iyi çözümler üretmek için yapıların elemanlarını birleştirmektir [17,22].

2.4.3.2.1. Optimizasyon Problemlerinde Kullanılan Çaprazlama Operatörleri

Optimizasyon problemlerinin bazı tiplerinde standart GA kuralları kullanılarak çözüm ve uygulama oldukça zor olabilir. Böyle problemlerde parametre değerlerinin yanısıra kullanılan niteliklerin konumu ve düzeni de ayrı bir önem taşır. GSP (Gezginci Satıcı Problemi- Travelling Salesman Problem) ile Çizelge Problemi bu özellikteki problemlerdir. Standart GA'larla bu problemlere karşı başarı sağlanamamasına rağmen yeniden dizilim sağlayan (re-ordering) GA'ların bazı formlarında ayrık optimizasyon problemlerine çözüm sağlanabilmektedir. Yeniden dizilim sağlayan GA'lar gen dizilerinin farklı şekilde yerleşimlerine ve değerlendirilmesine imkân sağlar. İlk kez 1985'de sunulan uygun yeniden dizilim operatörleri bazı çalışmalarda oldukça umut verici sonuçlara ulaşılmasını sağlamıştır. Bunlar Davis(1985), Cleveland ve Smith(1989), Syswerda ve Palmucci(1991) ve Syswerda(1991) tarafından yapılan çizelge problemleri üzerinedir [28-31]. Yeniden dizilim sağlayan GA'ların diğer uygulamaları ise Davis(1991) tarafından yapılan grafik renklendirme, Whitley(1991) tarafından yapılan GSP ve Cohooni(1987-1988) tarafından yapılan zamanlama ve sıralama tasarımıdır [26,32-34].

Yeniden dizilim sağlayabilen GA'lar klasik tiplerine göre bazı farklar içerirler. Yeniden dizilim probleminin doğası, kromozomlardaki farklı bir tanımlama ile çaprazlama ve mutasyon operatörlerinin yine farklı şekildeki kullanımlarını gerektirir. Kromozomların kodlanması probleme bağlıdır. İkili bir kodlama yerine bireysel etiketlere sahip genlerin kullanımı daha yararlı olabilir. Örneğin genlerin 1-8 değerlerini aldığı 8 genden oluşan

bir kromozom kullanılabilir. Farklı çaprazlama ve mutasyon operatörleri, kromozomdaki genlerin kaybolmasını ve kopyalanmasını önlemek için gereklidir.

2.4.3.2.2. Yeniden Dizilim Yapısına Sahip Çaprazlama Operatörleri

Yeniden dizilebilen çaprazlama operatörleri, iki ana kromozomdan bir ya da iki yeni çözüm üretmek için bir dizilim kombinasyonuna ihtiyaç duyarlar. Araştırmalarla farklı birkaç çaprazlama operatörü geliştirilmiştir. Goldberg (1989) üç farklı yeniden dizilim operatörü tanımlar: Kısmen Uyumlu Çaprazlama (Partially Matched Crossover), Sıralı Çaprazlama (Order Crossover) ve Periyodik Çaprazlama (Cycle Crossover). Bunun yanında Syswerda (1991) Sıraya Dayalı (Order-Based) ve Konuma Dayalı (Position-Based) çaprazlama olarak bilinen diğer çaprazlama operatörlerini de tanımlamıştır [17].

1-Kısmen Uyumlu Çaprazlama

İlk kez Goldberg ve Lingle (1985) tarafından GSP için uygulanmıştır. Bunlar arasındaki genlerin bulunduğu bölge uygunlama bölgesi olarak adlandırılır. Yeni çözümleri üretmek için bir dizinin uygunlama bölgesindeki genler diğerlerinin konumlarına uygun olarak transfer edilir. Transfer edilen genler, ilk kromozomdaki orjinal konumlarına yerleşecek şekilde yer değiştirir [18]. Şekil 2.6.'da bu durum görülmektedir.

P1	1 2 3 ¦ 4 5 6 ¦ 7 8	⇒	C1	1 2 3 ¦ 8 7 4 ¦ 5 6
P2	3 6 1 ¦ 8 7 4 ¦ 5 2	⇒	C2	3 8 1 ¦ 4 5 6 ¦ 7 2

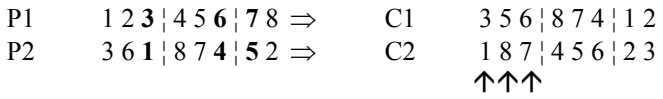
Şekil 2.6. Kısmen Uyumlu Çaprazlama

Buna göre P2'nin uygunluk bölgesindeki genler 8, 7 ve 4 şeklindedir. Bu genler aynı konumdaki 4, 5 ve 6 değerlerine sahip P1 içine yerleştirilir. Böylece 4 ile 8, 5 ile 7 ve 6 ile 4 değişerek P1'den C1 üretilmiş olur. C2 de yine benzer yolla elde edilir.

2-Sıralı Çaprazlama

Şekil 2.7.'de bu çaprazlamaya ait bir uygulama gösterilmektedir. Burada da yine bir çaprazlama bloğu mevcuttur. Bu blok içinde her iki kromozoma ait uygunlama bölgelerindeki genler karşılıklı yer değiştirir. Bu sırada transfer edilen genlerin orjinal konumlarında doldurulmak üzere boşluklar oluşur. Bundan sonra kromozomda kalan bu boşluklar kullanılmayan genlerle doldurulur. Şekil 2.7.'ye bakılırsa, C2'yi oluşturmak için P2 P1'den 4, 5 ve 6 değerlerini alır. Bu sırada bu değerlerin P2'deki orjinal

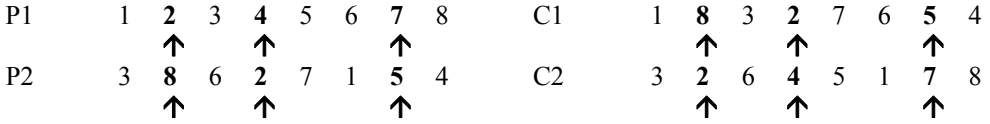
konumlarında boşluklar oluşmuştur. Önce bu boşluklar doldurulur. Dizideki diğer genler daha sonra bulunur. 2 ve 3 değerli genler çaprazlama bloğundan sonra yer alır. İlk çaprazlama bölgesine ise geriye kalan 1, 8 ve 7 değerli genler yerleştirilir.



Şekil 2.7. Sıralı Çaprazlama

3-Periyodik Çaprazlama

Bu çaprazlama tipinde çaprazlama işlemi belirli bir periyoda göre yapılır. Şekil 2.8.'de bu durum açıkça gösterilmiştir. Bunun için bir dizideki belirli genler sabit konumlarını korurken diğer genler ikinci dizideki aynı tür genlerle yer değiştirir.



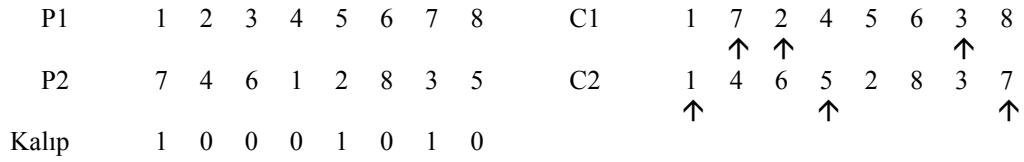
Şekil 2.8. Periyodik Çaprazlama

Şekil 2.8.'e göre P1'de ve P2'de işaretli olan gen değerleri sabit noktaları oluşturur. P1'de 1, 3 ve 6 konumlarındaki yine 1, 3 ve 6 değerleri ile P2'de aynı konumda yer alan 3, 6 ve 1 değerleri sabit genlerdir. O halde bunlar dışında kalan genler sırasıyla birbirleriyle yer değiştirir.

Yukarıda değinilen üç çaprazlama operatörü genelde plân optimizasyonunda kullanılmıştır. Yapılan çalışmalar bu çaprazlama operatörleri içinde en verimli olan elemanın periyodik çaprazlama olduğunu ortaya çıkarmıştır [20].

4-Sıraya Dayalı Çaprazlama

Sıraya Dayalı Çaprazlama kullanıldığında bir kromozomdan seçilen konumlardaki gen dizisi diğer kromozomdaki aynı değerli genlerin üzerine konulur. Bunun için rastgele olarak belirlenen bir kalıp (template) gereklidir. Şekil 2.9.'da yer alan kalıp üzerindeki 1 değerli genler önem taşımaktadır.

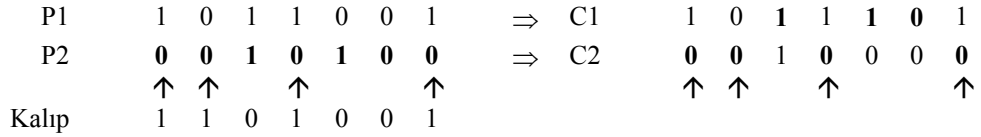


Şekil 2.9. Sıraya Dayalı Çaprazlama

Bu genlerin konumlarına bağlı olarak P2'den sırasıyla 7, 2 ve 3 değerleri çaprazlanacak genler olarak seçilmiştir. P2'de belirlenen bu 7, 2 ve 3 değerleri, P1 üzerinde bunların sıraya dizilmiş şekli olan 2, 3 ve 7'nin bulunduğu konumlara aynı sırada yerleştirilir. Böylece C1 oluşturulmuş olur. Aynı kalıba göre P1 üzerinde 1, 5 ve 7 değerleri seçilmiştir. P2'de bu değerlerin bulunduğu konumlara (Şekilde önce 7 ve 1, sonra 5 değeri yerleşmiş) 1, 5 ve 7 genleri sırayla transfer edilir. Bu şekilde de C2 oluşturulur.

5-Konuma Dayalı Çaprazlama

Bu çaprazlamada da yine ikili formdaki bir kalıp sabit konumları belirlemede kullanılır. P1'deki bu konumlarda mevcut olan genler C1 içine kopyalanırken bu genler dışında kalan genler P2'de aynı kalıbın seçtiği genlerce doldurulur. Benzer işlem C2 için de uygulanır [31]. Şekil 2.10.'da bu çaprazlama açıkça gösterilmiştir.



Şekil 2.10. Konuma Dayalı Çaprazlama

2.4.3.2.3. Alternatif Çaprazlama Operatörleri

GA uygulamalarında genelde tek noktalı çaprazlama yöntemi uygulanmasına rağmen, problemin tipine göre daha değişik çaprazlama yöntemleri de kullanılmaktadır. Aşağıda iki noktalı (two-point) ve düzgün (uniform) çaprazlama konuları anlatılmıştır.

- İki Noktalı Çaprazlama

İki noktalı çaprazlama, tek noktalı çaprazlama kullanıldığında yaşanan zorlukları aşma amacı taşımaktadır. Bunlardan en önemlisi de, çaprazlama esnasında ortaya çıkan uzun şemanın bozulmasıdır. Bu durum, klasik tek noktalı çaprazlama kullanıldığında

kolaylıkla birleşemeyen şemanın kombinasyonunda görülür. Şekil 2.11.'de bu çaprazlama örneği görülmektedir.

P1	1 0 **0** ¦ 1 0 1 1 0 1 ¦ **1 0**	⇒	C1	1 0 **0** ¦ 1 **1** 0 1 1 **0** ¦ **1 0**
P2	0 0 1 ¦ 1 **1** 0 1 1 **0** ¦ 1 1	⇒	C2	0 0 1 ¦ 1 0 1 1 0 1 ¦ 1 1

Şekil 2.11. İki Noktalı Çaprazlama

İki noktalı çaprazlama performansı önemli ölçüde artırabilir.

- Düzgün Çaprazlama

Düzgün Çaprazlama, Syswerda (1989) tarafından tanımlanmıştır. Bu şekilde, birleşemeyen kromozomlar daha kolay birleşme imkânı bulabilmektedir. Şekil 2.12.'de bu çaprazlama tipi görülmektedir. Bunun için rastgele olarak üretilen bir bit dizisine ihtiyaç duyulur. Bu bit dizisi çaprazlama işlemi için kalıp niteliği taşır. İlk çözüm için, kalıp üzerinde bit değeri 0 iken çaprazlama için gerekli gen değeri ilk kromozomdan ve 1 iken çaprazlama için gerekli gen değeri ikinci kromozomdan alınır. 2. çözüm için bu durumun tersi geçerlidir [31,35].

P1	1 0 0 1 0 1 0 0	⇒	C1	1 **1** 0 1 **1** 1 **0 1**
P2	0 1 0 0 1 1 0 1	⇒	C2	**0** 0 **0 0** 0 **1** 0 0
Kalıp	0 1 0 0 1 0 1 1			

Şekil 2.12. Düzgün Çaprazlama

2.4.3.3. Terslendirme (Inversion)

Terslendirme operatörü ile kromozom içindeki gen değerleri yeniden düzenlenir. Bit dizisi şeklinde kodlanan bilgi, problem için en iyi çözümü ifade eden gen dizisi olmayabilir. Genlere bağlı olan bu durum, bunları içeren şemanın süreklilik olasılığını etkiler. Yani gen dizisi algoritmanın etkinliğini etkiler. Terslendirme işlemi, rastgele olarak seçilerek belirlenen noktalar arasındaki bit dizisi değişimiyle sağlanır. Bu durum Şekil 2.13.'de gösterilmiştir.

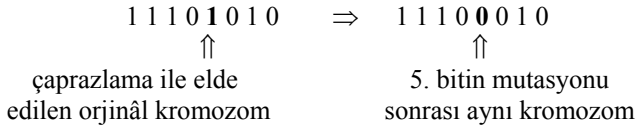
	terslendirme bölgeli orjinal kromozom		terslendirme sonunda kromozomun durumu
	0 1 ¦ 1 0 1 0 ¦ 0 1	⇒	0 1 ¦ 0 1 0 ¦ 1 0 1
Gen Etiketi	1 2 3 4 5 6 7 8		1 2 6 5 4 3 7 8

Şekil 2.13. Kromozomdaki Terslendirme

Terslendirme işleminden sonra da her bir gen yine önceki bilgiyi ifade eder. Bu fonksiyon değişik optimizasyon tiplerinde, GA'lar için simülasyon aşamasında oldukça verimli olabilmektedir [22,26].

2.4.3.4. Mutasyon Operatörü

Mutasyonun amacı, popülasyonda sürdürülen çeşitliliği sağlamaktır. Bu işlem, arasıra meydana gelerek bir kromozom içindeki genetik materyalin rastgele değişimini içerir. Mutasyon, bit dizileri olarak ifade edilen kromozomlarda, herhangi bir bitin rastgele tersinin alınmasıdır. Şekil 2.14.'de bu durum gösterilmiştir.



Şekil 2.14. Kromozomdaki Mutasyon

Mutasyon GA'nın başarısında temel bir rol oynar. Tekrar üreme ve çaprazlama etkin bir araştırma sağlamasına karşın bazen bunlar aşırı kullanılmış ve mevcut bazı genetik materyaller yani 1 ya da 0'lar ve dolayısıyla iyi çözümler kaybedilmiş olabilir. Yapay genetik sistemlerde mutasyon operatörü böyle kayıplara karşı koruma görevi yapar. Temel bir GA'da mutasyon, bir dizinin herhangi bir konumundaki bit değerinin değişimini ifade eder. Yani 1 ve 0 değerleri karşılıklı şekilde değişir. Böylece mutasyon, dizi uzayı boyunca gelişigüzel bir ilerlemeyi ifade eder. Tekrar üreme ve çaprazlamayla dikkatli kullanıldığında, önem taşıyan bilgilerin erken kaybına karşı bir sigorta görevi yapmaktadır [26,35]. Mutasyon; yeni, görülmemiş ve araştırılmamış çözüm elemanlarının bulunması için kullanılır. Bu operatör aynı zamanda algoritmanın alt optimal (sub-optimal) bir çözümü bulmasını engeller ve çalışmanın bölgesel bir noktada sıkışıp kalmamasını sağlar.

2.4.4. GA Performansının İyileştirilmesi

Özel bir probleme uygulanan GA'nın performansının iyi olması, algoritmada kullanılan parametrelerin uygun seçimine bağlıdır. Dikkatli ve uygun parametre kullanımının yanı

sıra probleme özel bazı kalite arttırıcı yöntemlere de başvurulabilir. Böylece optimum çözümler daha kısa sürede bulunabilir. Bu yöntemlerden bazıları aşağıda kısaca incelenecektir:

2.4.4.1. Uygunluk Ölçeklemesi (Fitness Scaling)

Bu birim, değerlendirme birimi tarafından değerlendirilen çözümlerin uygunluk değerlerini normalize eder. GA'nın çalışması incelendiğinde bunun açık olması istenir. Çalışma esnasında birbirinden farklı bir kromozom popülasyonu ve bu yüzden uygunluk değerlerinde geniş bir sınır mevcuttur. Tüm problem uzayı tamamen incelenmeden önce bu sınır içinde tekrar üreme için uygunluk değeri en iyi olan bireylere doğru bir eğilim vardır. Çalışmanın sonuna doğru benzer uygunluklara sahip birçok birey mevcut olacaktır. Tekrar üreme için uygunluğu daha iyi olan bireyler diğerlerine göre daha fazla şansa sahip olacaktır. Uygunluk ölçekleme ile, üretilen çözümler arasında yakın olan sonuçlar birbirlerine göre fark ifade eder hale gelir. Bu tekniklerin başarısı geniş ölçüde gösterilmiştir [36].

2.4.4.2. Kararlı Şekilde Tekrar Üreme(Steady Stade Reproduction)

Her bir jenerasyonda GA tarafından üretilen tüm bir popülasyon, yeni bir popülasyon ile yer değiştirir. Kararlı olarak tekrar üreme kavramı Davis (1991) tarafından ele alınmıştır. Bunun temel prensibi, herbir tekrar üreme periyodundaki bir jenerasyonda kromozomların sadece bir kısmının yer değiştirmesidir. Böyle bir anda popülasyon içine belirli sayıda (genelde bir ya da iki) çözüm girer ve bunlar en kötü popülasyon üyelerinin yerini alır [25,26,28]. Kararlı şekilde tekrar üremenin avantajı, jenerasyonlar arasında meydana gelebilecek, popülasyondan iyi bireylerin kaybını önlemesidir. Böylece popülasyon içine daha iyi yeni çözümler de katılmış olur. Bir algoritmada üretilen çözümlerin çoğu diğerinin aynısı olabilir. Bu durumda gelecek jenerasyonlarda aynı çözümlerin yerleşimini önlemek performansta bir iyileşme sağlayacaktır.

2.4.4.3. Operatör Parametreleri

Bir GA'nın performansı parametre değerlerine bağlıdır. Algoritma çalışmadan önce çaprazlama ve mutasyon olasılıkları karar verilen değerde sabit kalır. Buna karşın optimum olasılıklar, operatörlere ve kullanılan tekrar üreme programına bağlı olacaktır.

Bu olasılıklar, algoritmanın hızı ve algoritmadan sağlanan bütün en iyi çözümlerin performansı üzerinde önemli bir etkiye sahiptir. Farklı operatörlerce kullanılan parametre oranları, performansı büyük ölçüde etkiler ve optimal oranlar, bir çalışma boyunca değişir. Bir çalışma esnasında değişken olasılıkların kullanımı verim artışı sağlayabilir.

Bir GA'nın işleyişi sırasında çaprazlama önemli bir operatördür. Çünkü popülasyonu tamamen değiştirir ve problemin geniş bir araştırma uzayında incelenmesini sağlar. Bu esnada mutasyon operatörü daha az önem taşır. Buna karşın algoritmaya bağlı olarak kromozomlar benzer olduğunda ve çözüme giden yolda bir tıkanma olduğunda mutasyon operatörü daha önemli olur.

Algoritmanın çalışması sırasında çaprazlama olasılığı azalmış ve mutasyon artmış olabilir. Algoritmaya uygulanan teknikler, parametrelerin algoritmada ne kadar etkili olduğunu bulmayı amaçlar. Parametre değerlerine göre değişen algoritma performansı, optimum değerlerin bulunmasında etkin rol oynar [37].

2.4.4.4. En İyinin Saklanması

Her bir jenerasyonda mevcut en iyi çözümün kopyalanarak bir sonraki jenerasyon için saklanmasıdır. En iyinin saklanması, jenerasyonda yüksek kaliteli genetik bilgilere ulaşmasını sağlamak için kullanılır. Bu, her bir jenerasyondaki en uygun kromozomun daha ilerisi için otomatik olarak kopyalanması yoluyla yapılan basit bir işlemdir. Buna rağmen bulunan en iyi çözümün son jenerasyona kadar korunmasını sağlayan oldukça önemli bir yöntemdir.

3. BÖLÜM

GA UYGULAMALARI

3.1. Giriş

Bu bölümde, GA'nın çalışmasının daha iyi anlaşılması için iki farklı örnek üzerinde GA optimizasyon uygulaması gerçekleştirilmiştir. Dolayısıyla ilk olarak belli uzunluktaki bir dizideki 1'lerin sayısının maksimize edilmesinde ve ikinci olarak iki değişkenli bir fonksiyonun katsayılarının optimizasyonu işleminde GA'nın nasıl kullanıldığına ilişkin gerçekleştirilen uygulamalar verilmiştir.

3.2. Bir Dizideki 1'lerin Sayısının GA İle Maksimize Edilmesi

Amaç: 5 Rakamdan oluşan bir dizide bulunan 1'leri (1 sayılarını) maksimize etmek.

Gen Sayısı=Kromozom Uzunluğu = 5

Popülasyon Büyüklüğü = 4

Uygunluk Fonksiyonu = 1 değerine sahip genlerin sayısı

Bu popülasyon Tablo 3.1.'de gösterilmiştir. Sırasıyla yapılacak işlemler aşağıdaki gibi verilebilir:

- Bireylerin Seçimi,
- Seçilen bireylerin eşleşme havuzuna atılması yani kopyalanması,
- Havuzdaki bireylerin eşleştirilmesi,
- Uygunluk için kullanılan nispi seçim metodu; rulet tekerleği seçim metodu.

Tablo 3.1. Popülasyondaki Bireylerin Gen Değerleri Ve Bunların Uygunluğu

Bireyler Ve Gen Değerleri						Uygunluk
Birey 1	1	1	0	0	1	3
Birey 2	0	0	0	0	1	1
Birey 3	1	1	1	1	1	5
Birey 4	0	1	1	1	0	3

3.2.1. Rulet Tekerleği Seçme Metodu

Eşitlik (3.1) popülasyondaki bireylerin seçilme olasılığını ifade etmektedir. Tablo 3.2.'de başlangıç popülasyonu görülmektedir.

$$prob_i = \frac{fitness_i}{\sum_j fitness_j} \text{ , (j=1, ..., popülasyon büyüklüğü), (i=1,...,4)} \quad (3.1)$$

Tablo 3.2. Başlangıç Popülasyonu

Geçerli Popülasyon						Uygunluk
i1	1	1	0	0	1	3
i2	0	0	0	0	1	1
i3	1	1	1	1	1	5
i4	0	1	1	1	0	3
Toplam Uygunluk						12

Uygunluk Fonksiyonu = 1 değerine sahip genlerin sayısı

Uygunluk (Birey 1) = 1+1+1=3

Uygunluk (Birey 2) = 1

Uygunluk (Birey 3) = 1+1+1+1+1=5

Uygunluk (Birey 4) = 1+1+1=3

Toplam Uygunluk = 3+1+5+3 =12

3.2.2. Her Bir Bireyin Seçilme Olasılığı

Prob (i1) =3/12=0.25

Prob (i2) =1/12=0.08

Prob (i3) =5/12=0.42

Prob (i4) =3/12=0.25

Havuzun Büyüklüğü= Popülasyon Büyüklüğü=4

Eşleşme havuzunda her bir bireyin kaç defa temsil edileceğinin beklenen sayısı,

i1 için prob (i1) * Popülasyon Büyüklüğü = (3/12*4)=1 Temsilci

i2 için prob (i2) * Popülasyon Büyüklüğü = (1/12*4)=0 Temsilci

i3 için prob (i3) * Popülasyon Büyüklüğü = (5/12*4)=2 Temsilci

i4 için prob (i4) * Popülasyon Büyüklüğü = (3/12*4)=1 Temsilci

Farz edelim ki; tekerlek 4 defa döndürülmüş olsun ve i1' in 1 kopyası, i3'ün 2 kopyası ve i4'ün 1 kopyası eşleşme havuzuna kopyalanmış, yani atılmış olsun. Bu durumda geçerli eşleşme havuzundaki yeni bireyler; Tablo 3.3.'te verilmiştir.

Tablo 3.3. Rulet Tekerleği Seçim Metodundan Sonra Havuzdaki Durum

Geçerli Eşleşme Havuzu						Uygunluk
Eş 1	1	1	0	0	1	i1
Eş 2	1	1	1	1	1	i3
Eş 3	1	1	1	1	1	i3
Eş 4	0	1	1	1	0	i4

Rasgele eşleştirmenin bir sonucu olarak (eş1, eş3) ile ve (eş2, eş4) ile yeniden çoğalma için Tablo 3.4.'deki gibi eşleşmiş olsun [38].

Tablo 3.4. (eş1, eş3) ile ve (eş2, eş4)' ün Yeniden Çoğalma İçin Eşleştirilmesi

Çiftler	
Çift 1	Çift 2
11001	11111
11111	01110

Seçilen çiftler yeni bireyler üretirler. Bu örnekte tek noktalı çaprazlama kullanılmaktadır. Çaprazlama olma ihtimali P_c ile gösterilir. P_c tipik olarak 1'e yakın seçilmelidir. Tek noktalı çaprazlamada seçilen yerden birey bölünmeli ve çiftler arasında parçalar takas edilmelidir. Sonuçta iki yeni bireye dönüşmelidir. P_c =1 için Tablo 3.5.'teki sonuçlar elde edilmiştir.

Tablo 3.5. Üretilen Yeni Bireyler Ve Uygunluk Değerleri

Çift 1 için çaprazlama yeri 3. bitten sonra				Yeni Bireyler					
110 | 01	→	11011	i1	1	1	0	1	1
111 | 11	→	11101	i2	1	1	1	0	1
			i3	0	1	1	1	1
Çift 2 için çaprazlama yeri 1. bitten sonra			i4	1	1	1	1	0
1 | 1111	→	01111						
0 | 1110	→	11110	Toplam uygunluk=16					

3.2.3. Mutasyon

Herhangi bir birey mutasyona uğrayabilir. Bir bireyin her bir geni mutasyona uğrayabilir. Mutasyon olması ihtimali P_m ile gösterilir. P_m tipik olarak çok küçük örneğin 0,001 civarında seçilmelidir. Mutasyon operatörü ikili (binary) gösterimde uygulandığı genin tümleyenini alır. Mutasyon çeşitliliği arttırır. Farz edelim ki rasgele çekiliş sonucunda i1'in 1. geni ve i3'ün 4. geni Tablo 3.6.'daki gibi mutasyona uğramış olsun.

Tablo 3.6. Mutasyona Uğrayan Bitlerin Gösterimi

i1	**1**	1	0	1	1	→	0	1	0	1	1
i3	1	1	1	**1**	0	→	1	1	1	0	0

Mutasyon işleminden sonra yeni popülasyon Tablo 3.7.'de gösterildiği gibi olmaktadır.

Tablo 3.7. Mutasyon İşleminden Sonra Popülasyonun Durumu

Geçerli Popülasyon						Uygunluk
i1	0	1	0	1	1	3
i2	1	1	1	0	1	4
i3	1	1	1	0	1	4
i4	0	1	1	1	1	4
Toplam Uygunluk						15

3.2.4. Bitirme Kriteri

Algoritmanın ana döngüsü bitirme kriterini buluncaya kadar devam eder. Örneğin aşağıdaki durumlarda ana döngü çevrimine devam eder.

- Jenerasyonunun önceden belirlenmiş sayısı (Ör. Gen=100),
- Hedefi buluncaya kadar,
- Popülasyon yakınsayıncaya kadar.

1. Durum:

Jenerasyon Sayısı =250

- Döngü 250 kez tekrar edecek,
- Her bir jenerasyonda en iyi bireyler bulunacak,
- Bulunan bireylerin en iyisi çözümü temsil edecek.

2. Durum

Amaç: 1'lerin sayısını maksimize etmek.

- Hedef: Bütün Genlerin değerini 1 yapmak.
- Döngü sonsuza kadar tekrarlanacak,
- Her bir jenerasyondaki en iyi bireyleri bulacak,
- Algoritma hedef bireyleri bulduğu zaman sonlandırılacak.

3. Durum

- 4 bireyin %95 i aynı gen değerlerine sahip olmalıdır.
- Gen yakınsaması için 4 bireyinde aynı yerlerde aynı değerlere sahip olması gerekir
- Popülasyon yakınsaması için 5 genin değerleri yakınsamalıdır.

Tablo 3.8. Belirtilen Durumlar Altında Yakınsamış Popülasyon Örneği

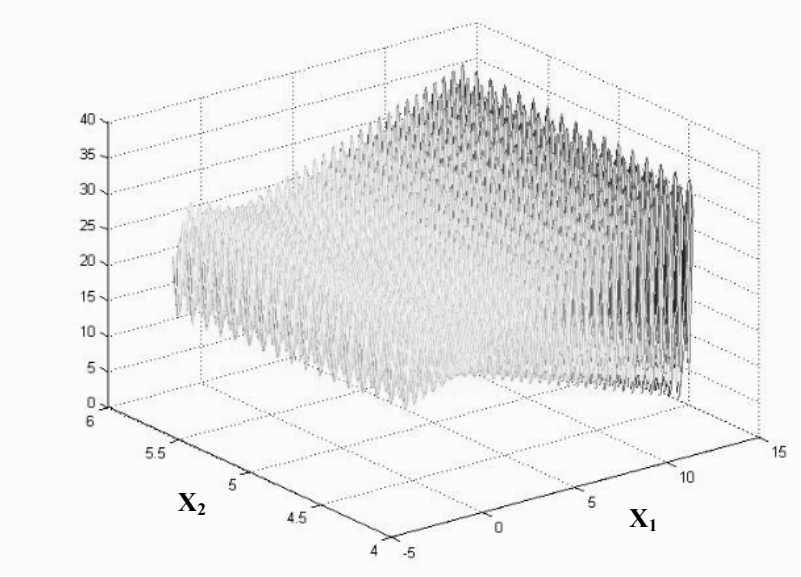
Yakınsamış Popülasyon Örneği		
Örnek 1	Örnek 2	Örnek 3
i1: 11010	i1: 00000	i1: 11111
i2: 11010	i2: 00000	i2: 11111
i3: 11010	i3: 00000	i3: 11111
i4: 11010	i4: 00000	i4: 11111

3.3. İki Değişkenli Fonksiyonun GA Kullanılarak Optimize Edilmesi

Bu örnekte fonksiyon optimizasyonu için, GA'nın kullanımı incelenmiştir. Bunun için, başlangıçta popülasyon büyüklüğü pop_size = 20 ve genetik operatörlerin olasılıkları Pc=0.25 ve Pm=0.01 alınmıştır. Eşitlik (3.2)'de verilen fonksiyon maksimize edilmiştir.

$$F(x_1, x_2) = 21.5 + x_1 .\sin(4\pi x_1) + x_2 .\sin(20\pi x_2) \qquad (3.2)$$

Burada, $-3.0 \leq x_1 \leq 12.1$ ve $4.1 \leq x_2 \leq 5.8$' dir. F fonksiyonunun grafiği Şekil 3.1.'de verilmiştir.



Şekil 3.1. f(x_1 x_2) fonksiyonunun grafiği

3.3.1. Değişkenlerin Tanımlanması

İhtiyaç duyulan doğruluk, her bir değişken için 4 desimal boşluk ayrılmıştır. X_1 değişkeni domeni 15.1 uzunluğa sahiptir. Dolayısıyla [-3.0 , 12.1] aralığı 15.1.10000 değerine eşit büyüklükte alanlara ayrılmalıdır. Bu, kromozomun ilk parçası olarak 18 bite gerek duyulduğu anlamına gelir($2^{17} < 151000 \leq 2^{18}$). x_2 değişken domeni ise 1.7 uzunluğa sahiptir. O halde [4.1 , 5.8] aralığı 1.7.10000 değerine eşit alanlara ayrılmalıdır. Bu, kromozomun ikinci kısmı için 15 bite gerek duyulduğu anlamına gelir($2^{14} < 17000 \leq 2^{15}$). Çözüm vektörü olan bir kromozomun toplam uzunluğu m=18+15=33 bit olduğundan, ilk 18 bit ise (19-33) x_2'nin kodudur.

$(010001001011010000111110010100010)_2$

Yukarıdaki örnek kromozom için ilk 18 bit bu şekildedir.

$(010001001011010000)_2$

Bunun tam sayı değeri,

$$X_1 = -3.0 + \text{decimal}\,(a)_2 . \quad - \quad \frac{12.1 - (-3.0)}{2^{18} - 1}$$

$$= 3.0 + 70352 \ \frac{15.1}{262143}$$

$$= -3.0 + 4.052426 = 1.052426$$

Kalan 15 bit ise, $(111110010100010)_2$ olarak yazılır. Bunun tam sayı değeri,

$$X_2=4.1 + \text{decimal } (b)_2 . \frac{5.8-4.1}{2^{15}-1}$$

$$=4.1 + 31906 . \frac{1.7}{32767}$$

=4.1 + 1.655330 = 5.755330 olarak bulunur.

Bu durumda $(01000100101101000011111100101000 10)_2$ kromozomu aşağıdaki gibi temsil edilir.

$(x_1 , x_2)=(1.052426 , 5.755330)$

Bu kromozomun uygunluk değeri ise,

F=(1.052426 , 5.755330)=20.25640 olarak bulunmuştur.

3.3.2. Başlangıç Popülasyonunun Oluşturulması

Genetik algoritma kullanılarak f fonksiyonunu optimize etmek için ilk önce pop_size=20 kromozomdan oluşan bir popülasyon oluşturulmalıdır. Kromozomlardaki tüm 33 bitler rasgele olarak Şekil 3.2.' deki gibi oluşturulur [39].

$V_1 = (10011010000000111111101001101111 1)$
$V_2 = (11100010010011011100101010001101 0)$
$V_3 = (00001000001100100000101011011101)$
$V_4 = (10001100010110100111100000111001 0)$
$V_5 = (00011101100101001101011111100010 1)$
$V_6 = (00010100001001010100101011111101 1)$
$V_7 = (00100010000011010111101101111101 1)$
$V_8 = (10000110000111010001011010110011 1)$
$V_9 = (01000000010110001011000000111110 0)$
$V_{10} = (00000111100011000001101000011101 1)$
$V_{11} = (01100111111011010110000110111100 0)$
$V_{12} = (11010001011110110100010101000000 0)$
$V_{13} = (11101111101000100011000000100011 0)$
$V_{14} = (01001001100000101010011110010100 1)$
$V_{15} = (11101110110111000010001111101111 0)$
$V_{16} = (11001111000001111110000110100101 1)$
$V_{17} = (01101011111100111101000110111110 1)$
$V_{18} = (01110100000000111010011111010110 1)$
$V_{19} = (00010101001111111111000011000110 0)$
$V_{20} = (10111001011001111001100010111111 0)$

Şekil 3.2. Oluşturulan Başlangıç Popülasyonu

Buradan açıkça görülebileceği gibi, v_2 kromozomu en kötü v_{15} kromozomu ise en iyi uygunluk değerine sahiptir. Seçim için sistem bir rulet tekerleğine uyarlanır. Popülasyonun toplam uygunluğu

$$F=\sum_{i=1}^{20} \text{eval}(v_i) = 387.776922\text{' dir.}$$

Her bir v_i(i= 1,.....,20) kromozomu için P_i seçim olasılığı Şekil 3.3.'de gösterilmiştir.

P_1 = eval(V_1)/F =0.067099	P_2 = eval(V_2)/F =0.019547
P_3 = eval(V_3)/F =0.050355	P_4 = eval(V_4)/F =0.044867
P_5 = eval(V_5)/F =0.065350	P_6 = eval(V_6)/F =0.046677
P_7 = eval(V_7)/F =0.041315	P_8 = eval(V_8)/F =0.046315
P_9 = eval(V_9)/F =0.041590	P_{10} = eval(V_{10})/F =0.054873
P_{11} = eval(V_{11})/F =0.660372	P_{12} = eval(V_{12})/F =0.038712
P_{13} = eval(V_{13})/F =0.070444	P_{14} = eval(V_{14})/F =0.051257
P_{15} = eval(V_{15})/F =0.077519	P_{16} = eval(V_{16})/F =0.061549
P_{17} = eval(V_{17})/F =0.035320	P_{18} = eval(V_{18})/F =0.039750
P_{19} = eval(V_{19})/F =0.051823	P_{20} = eval(V_{20})/F =0.035244

Şekil 3.3. Popülasyondaki Kromozomların Seçim Olasılığı

v_i (i=1,.....,20) kromozomu için q_i kümülatif (gittikçe artan) olasılıkları ise Şekil 3.4.'de gösterilen biçimde oluşacaktır.

Q_1 =0.067099	Q_2 =0.086647	Q_3 =0.137001	Q_4 =0.181890
Q_5 =0.247240	Q_6 =0.293917	Q_7 =0.335232	Q_8 =0.361546
Q_9 =0.423137	Q_{10} =0.478000	Q_{11} =0.538781	Q_{12} =0.577093
Q_{13} =0.647537	Q_{14} =0.698794	Q_{15} =0.776314	Q_{16} =0.837863
Q_{17} =0.873182	Q_{18} =0.912932	Q_{19} =0.964756	Q_{20} =1.000000

Şekil 3.4. Kromozomlar İçin Kümülatif Olasılıklar

Değerlendirme esnasında her bir kromozomun kodu çözülür ve çözülen bu kodların (x_1,x_2)'den uygunluk değerleri Şekil 3.5.'deki gibi hesaplanır.

$$\text{eval}(v_1) = \int (6.084492, 5.652242) = 26.019600$$
$$\text{eval}(v_2) = \int (10.348434, 4.380264) = 7.580015$$
$$\text{eval}(v_3) = \int (-2.516603, 4.390381) = 19.526329$$
$$\text{eval}(v_4) = \int (5.278638, 5.3593460) = 17.406725$$
$$\text{eval}(v_5) = \int (-1.255173, 4.734458) = 25.341160$$
$$\text{eval}(v_6) = \int (-1.811725, 4.391937) = 18.100417$$
$$\text{eval}(v_7) = \int (-0.991471, 5.680258) = 16.020812$$
$$\text{eval}(v_8) = \int (4.910618, 4.703018) = 17.959701$$
$$\text{eval}(v_9) = \int (0.795406, 5.381472) = 16.127799$$
$$\text{eval}(v_{10}) = \int (-2.554851, 4.793707) = 21.278135$$
$$\text{eval}(v_{11}) = \int (3.130078, 4.996097) = 23.410669$$
$$\text{eval}(v_{12}) = \int (9.356179, 4.239457) = 15.011619$$
$$\text{eval}(v_{13}) = \int (11.134646, 5.378671) = 27.316702$$
$$\text{eval}(v_{14}) = \int (1.335944, 5.151378) = 19.876294$$
$$\text{eval}(v_{15}) = \int (11.089025, 5.054515) = 30.060205$$
$$\text{eval}(v_{16}) = \int (9.211598, 4.993762) = 23.867227$$
$$\text{eval}(v_{17}) = \int (3.367514, 4.571343) = 13.696165$$
$$\text{eval}(v_{18}) = \int (3.843020, 5.158226) = 15.414128$$
$$\text{eval}(v_{19}) = \int (-1.746635, 5.395584) = 20.095903$$
$$\text{eval}(v_{20}) = \int (7.935998, 4.757328) = 13.666916$$

Şekil 3.5. Kromozomların Uygunluk Değerlerinin Hesaplanması

Bundan sonra rulet tekerleği 20 defa döndürülerek, her bir defada yeni popülasyon için tekbir kromozom seçilir. [0,1] aralığında 20 sayıdan oluşan rasgele bir sayı dizisi Şekil 3.6.'daki gibi oluşmuş olsun.

0.513870	0.175741	0.308652	0.534534	0.947628
0.171736	0.702231	0.226431	0.494773	0.424720
0.703899	0.389647	0.277226	0.368071	0.983437
0.005396	0.765682	0.646473	0.767139	0.780237

Şekil 3.6. [0,1] Aralığında Rasgele Üretilen 20 Adet Sayı

İlk sayı r= 0.513870 y_{10}'dan daha büyük ve q_{11}'den daha küçüktür. Bu yeni popülasyon için v_{11} kromozomun seçildiği anlamına gelir. İkinci sayı r= 0.175741 ise q_3'den büyük

ve q_4'den daha küçüktür. Bu da, yeni popülasyon için v_4 kromozomun seçildiği anlamına gelmektedir. Bu durum benzer şekilde sürdürülür. Sonuçta yeni popülasyon Şekil 3.7.'deki kromozomlardan meydana gelecektir.

$V'_1 = (0110011111110110101100001101111000)\ (V_{11})$
$V'_2 = (1000110001011010011110000011100 10)\ (V_4)$
$V'_3 = (0010001000001101011110110111111011)\ (V_7)$
$V'_4 = (0110011111110110101100001101111000)\ (V_{11})$
$V'_5 = (0001010100111111111110000110001100)\ (V_{19})$
$V'_6 = (1000110001011010011110000011100 10)\ (V_4)$
$V'_7 = (1110111011011100001000111111011110)\ (V_{15})$
$V'_8 = (0001110110010100110101111111000101)\ (V_8)$
$V'_9 = (0110011111110110101100001101111000)\ (V_{11})$
$V'_{10} = (0000100000110010000010101110111101)\ (V_3)$
$V'_{11} = (1110111011011100001000111111011110)\ (V_{18})$
$V'_{12} = (0100000001011000101100000011111100)\ (V_{11})$
$V'_{13} = (0001010000100101010010101111111011)\ (V_6)$
$V'_{14} = (1000011000011101000101101011001111)\ (V_8)$
$V'_{15} = (1011100101100111100110001011111110)\ (V_{20})$
$V'_{16} = (1001101000000011111111010011011111)\ (V_1)$
$V'_{17} = (0000011110001100000110100001110 11)\ (V_{10})$
$V'_{18} = (1110111110100010001100000010001 10)\ (V_{13})$
$V'_{19} = (1110111011011100001000111111011110)\ (V_{16})$
$V'_{20} = (1100111100000111111000011010010 11)\ (V_{16})$

Şekil 3.7. Rulet Tekerleği Seçiminden Sonra Oluşan Yeni Popülasyon

Bundan sonra çaprazlama operatörü yeni popülasyonun bireylerine (v_i vektörlerine) uygulanır. Pc=0.25 olan çaprazlama olasılığı, ortalama olarak kromozomlardan %25'inin (5/20 sinin) çaprazlamaya uğrayacağını ifade eder. Bu işlem şöyle yapılır: Yeni popülasyondaki her bir kromozom için [0,1] aralığında rasgele bir r sayısı üretilir. Eğer r<0.25 ise, çaprazlama için, verilen kromozom seçilir. Söz konusu rasgele sayılar Şekil 3.8.'deki gibi üretilmiş olsun.

0.822951	0.151932	0.625477	0.314685	0.346901
0.917201	0.519760	0.401154	0.606758	0.785402
0.031523	0.869921	0.166525	0.674520	0.758400
0.581893	0.389248	0.200232	0.355635	0.826927

Şekil 3.8. Çaprazlama İşlemi İçin [0,1] Aralığında Üretilen Rasgele Sayılar

3.3.3. İki Değişkenli Fonksiyonda Çaprazlama

Bu v_2', v_{11}' ,v_{13}' ve v_{18}' 'nün çaprazlama için seçilmiş olsun. Seçilen kromozomlar çift sayıda olduğundan bunlar kolaylıkla eşitlenebilir. Eğer seçilen kromozom sayısı tek olsaydı, ya ekstra bir kromozom daha seçilecekti ya da seçilen kromozomlardan biri bırakılırdı. Seçilen bu kromozomların eşleştirilmesi gerekmektedir. v_2' ve v_{11}'

kromozomları ve v_{13}' ve v_{18}' kromozomlarını birbiriyle eşleştirelim. Bu iki kromozomdan oluşan kısımlarının her biri için [1,32] aralığında (bir kromozomdaki bitlerin toplam uzunluğu 33 olduğundan) bir p sayısı, çaprazlama noktasını belirler. İlk kromozom ikilisi,

$V_2'=(100011000 \mid 101110000100011111011110)_2$
$V_{11}'=(111011101 \mid 101101001111000001110010)_2$

için üretilen sayı p=9'dur. Bu kromozomlar 9. bitten sonra çaprazlamaya tabi tutulurlar. Böylece,

$V_2'' = (100011000 \mid 101110000100011111011110)_2$
$V_{11}''= (111011101 \mid 101101001111000001110010)_2$

Yeni çözüm kromozomları elde edilir. Diğer kromozom ikilisi,

$V_{13}'= (000101000010010101 00 \mid 1010111111011)_2$
$V_{18}'= (111011111010001000 11 \mid 0000001000110)_2$

için üretilen sayı p=20 olduğundan çaprazlama sonrası elde edilecek yeni çözüm kromozomları aşağıdaki gibi olacaktır.

$V_{11}'= (000101000010010101 00 \mid 0000001000110)_2$
$V_{18}'= (111011111010001000 11 \mid 1010111111011)_2$

Böyle bir örnek çalışmada bu sayılardan 5 tanesi 0.01'den küçük bulunmuş olsun. Tablo 3.9.'da kromozom içindeki bit sayısı ve kromozom sayısı içindeki bitin konumu görülmektedir.

Tablo 3.9. Kromozom İçindeki Bit Sayısı ve Kromozom Sayısı İçindeki Bitin Konumu

Bit Pozisyonu	Kromozom Sayısı	Kromozomdaki Bit Sayısı
112	4	13
319	11	19
418	13	22
429	13	33
602	19	8

Böylece popülasyonun yeni şekli Şekil 3.9.'daki gibi olacaktır.

V'_1 = (01100111111011010110000110111 1000)
V'_2 = (10001100010111000010001111101111 0)
V'_3 = (00100010000011010111101101111 1011)
V'_4 = (01100111111011010110000110111 1000)
V'_5 = (00010101001111111111100001100 01100)
V'_6 = (10001100010111010011110000011 10010)
V'_7 = (11101110110111000010001111101 11110)
V'_8 = (00011101100101001101011111100 0101)
V'_9 = (01100111111011010110000110111 1000)
V'_{10}= (00001000001100100000101011101 1101)
V'_{11}= (11101110110111000010001111101 11110)
V'_{12}= (01000000010110001011000000111 1100)
V'_{13}= (00010100001001010100101011111 11011)
V'_{14}= (10000110000111010001011010110 0111)
V'_{15}= (10111001011001111001100010111 11110)
V'_{16}= (10011010000000011111111010011 011111)
V'_{17}= (00000111100011000001101000011 10111)
V'_{18}= (11101111101000100011000000100 0110)
V'_{19}= (11101110110111000010001111101 11110)
V'_{20}= (11001111000001111110000110100 1011)

Şekil 3.9. Çaprazlama İşleminden Sonra Oluşan Yeni Popülasyon

3.3.4. İki Değişkenli Fonksiyonda Mutasyon

Bundan sonraki mutasyon operatörü, bir bitteki değişimi ifade eder. P_m=0.01 olan mutasyon olasılığı ortalama olarak bitlerin %1'inin mutasyona uğrayacağını ifade eder. Her bir popülasyon toplam max. pop_size =33.20=660 bit olduğundan jenerasyon başına ortalama 6.6 bit mutasyona uğrayacaktır. Her bitin mutasyon için eşit şansı bulunmaktadır. Böylece popülasyondaki her bir bit için [0,1] aralığında değişen bir r sayısı üretilir. Eğer r<0.01 ise bu bit mutasyona uğrar. Bu ifade, 660 adet rasgele sayı üretileceği anlamını taşır. Bit konumu ve rasgele sayılar Tablo 3.10.'da listelenmiştir.

Tablo 3.10. Mutasyon İşlemi İçin Üretilen Rasgele Sayılar

Bit Pozisyonu	Üretilen Rasgele Sayı
112	0.000213
319	0.009945
448	0.008809
429	0.005425
602	0.002836

Buna göre mutasyon operatöründen 4 kromozom etkilenmiştir. Kromozomlardan birinde (13.) iki tane bit değişime uğramıştır. Mutasyona uğrayan bitlerden oluşan popülasyon Şekil 3.10.' da listelenmiştir.

V_1 = (01100111111011010110000110111 1000)
V_2 = (10001100010111000010001111011110)
V_3 = (00100010000011010111101101111 1011)
V_4 = (01100111111001010110000110111 1000)
V_5 = (00010101001111111111100001100 01100)
V_6 = (10001100010110100111100000111 0010)
V_7 = (11101110110111000010001111011110)
V_8 = (00011101100101001101011111100 0101)
V_9 = (01100111111011010110000110111 1000)
V_{10}= (00001000001100100000010101110 11101)
V_{11}= (11101110110110100101100000111 0010)
V_{12}= (01000000010110001011000000111 1100)
V_{13}= (00010100001001010100010000 1000111)
V_{14}= (10000110000111010001011010110 0111)
V_{15}= (10111001011001111001100010111 1110)
V_{16}= (10011010000000011111110100110 11111)
V_{17}= (00000111100011000001101000001 11011)
V_{18}= (11101111101000100011101011111 11011)
V_{19}= (11101110010111000010001111011110)
V_{20}= (11001111000001111110000110100 1011)

Şekil 3.10. Mutasyona Uğrayan Bitlerden Sonra Oluşan Yeni Popülasyon

Bundan sonra seçim işlemi yeniden çalıştırılır ve genetik operatörler uygulanarak sonraki jenerasyonlar değerlendirilir. 1000 jenerasyon sonra popülasyon Şekil 3.11.'deki gibidir.

V_1 = (11101111011001101110010101011 1011)
V_2 = (11100110011000010001010101011 1000)
V_3 = (11101111011101101110010101011 1011)
V_4 = (11100110001000011000010101011 1001)
V_5 = (11101111011101101110010101011 1011)
V_6 = (11100110011000010000010001010 0001)
V_7 = (11010110001001001000110001011 0000)
V_8 = (11110110001000101000110101001 0001)
V_9 = (11100110001001001000110001011 0001)
V_{10}= (11101111011101101110010101011 1011)
V_{11}= (11010110000001001000110001011 0000)
V_{12}= (11010110001001001000110001011 0001)
V_{13}= (11101111011101101110010101011 1011)
V_{14}= (11101111011101101110010101011 1011)
V_{15}= (11100110011000010000010101011 1011)
V_{16}= (11100110101011100101010011 0110001)
V_{17}= (11100110011000010000010101011 1011)
V_{18}= (11100110011000010000010101011 1001)
V_{19}= (11110110001000101000111000001 0001)
V_{20}= (11100110011000010000010101011 1001)

Şekil 3.11. 1000 Jenerasyondan Sonra Popülasyonun Durumu

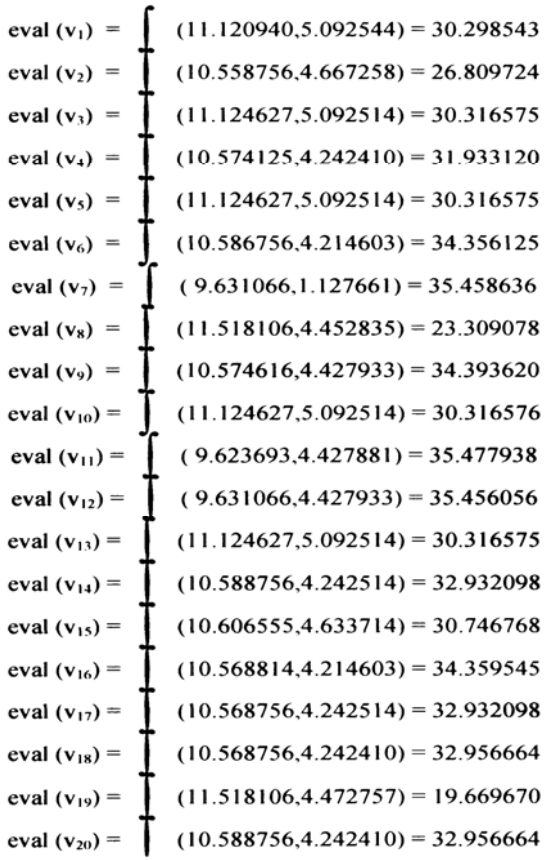
Yeni popülasyonun toplam uygunluğu F= 447.049688'dir. Bu değer önceki popülasyonun toplam uygunluğu olan F= 387.776822'den daha büyüktür. Önceki popülasyonda v_{15} iken, şimdi 33.351874 uygunluğa sahip v_{11} kromozomudur.

Böylelikle genetik prosedürdeki tüm döngünün bir iterasyonu ya da bir jenerasyonu tamamlanmış olur. Yeni popülasyondaki değerlendirme aşamasında herbir kromozomun kodu çözülür ve bunların (x_1,x_2) değerlerine bağlı olarak uygunluk fonksiyonları değerleri hesaplanır. Böylece Şekil 3.12.' deki liste elde edilir.

$$
\begin{aligned}
\text{eval}(v_1) &= \int (3.130078, 1.996097) = 23.410669 \\
\text{eval}(v_2) &= \int (5.279012, 5.054515) = 18.201083 \\
\text{eval}(v_3) &= \int (-0.991471, 5.680258) = 16.020812 \\
\text{eval}(v_4) &= \int (3.128235, 4.996097) = 23.412613 \\
\text{eval}(v_5) &= \int (-1.746635, 5.395584) = 20.095903 \\
\text{eval}(v_6) &= \int (5.278638, 5.593460) = 17.406725 \\
\text{eval}(v_7) &= \int (11.080025, 5.054515) = 30.060205 \\
\text{eval}(v_8) &= \int (-1.255173, 4.734458) = 25.341160 \\
\text{eval}(v_9) &= \int (3.430078, 4.996097) = 23.410669 \\
\text{eval}(v_{10}) &= \int (-2.516603, 4.390381) = 19.526329 \\
\text{eval}(v_{11}) &= \int (11.088621, 4.743434) = 33.351874 \\
\text{eval}(v_{12}) &= \int (0.795106, 5.381472) = 16.127799 \\
\text{eval}(v_{13}) &= \int (-1.811725, 4.209937) = 22.692462 \\
\text{eval}(v_{14}) &= \int (4.910618, 4.703018) = 17.959701 \\
\text{eval}(v_{15}) &= \int (7.935998, 4.757338) = 13.666916 \\
\text{eval}(v_{16}) &= \int (6.084492, 5.652242) = 26.019660 \\
\text{eval}(v_{17}) &= \int (-2.554851, 4.793707) = 21.276435 \\
\text{eval}(v_{18}) &= \int (11.134616, 5.666976) = 27.591064 \\
\text{eval}(v_{19}) &= \int (11.059532, 5.054515) = 27.608441 \\
\text{eval}(v_{20}) &= \int (9.211598, 4.993762) = 23.867227
\end{aligned}
$$

Şekil 3.12. Kromozomların Uygunluk Değerlerinin Hesaplanması

Bunların uygunluk değerleri ise Şekil 3.13.'de gösterilmiştir.



Şekil 3.13. Jenerasyon Sonrası Oluşan Popülasyondaki Kromozomların Uygunluk Değerleri

3.3.5. Çalışmanın Değerlendirilmesi

Programın çalışmasına bakıldığında önceki jenerasyonlardaki bazı kromozomların uygunluk değerleri, 1000 jenerasyon sonra en iyi kromozomun uygunluk değeri olan 35.477938'den daha iyi olmaktadır. Örneğin 396. jenerasyonda en iyi kromozom 38.827553 değerini almaktadır. Bu, örneklerin stokastik hatalarına bağlıdır. Bir algoritmada üretilen bireylerin çoğu diğerinin aynısı olabilir. Bu durumda gelecek jenerasyonlarda aynı bireylerin yerleşimini önlemek performansta bir iyileşme sağlayacaktır.

Değerlendirme projesindeki en iyi bireyin korunması nispeten kolaydır. Genetik algoritma uygulamalarında dağınık yerleşimli en iyi bireyin kaydedilmesi uygulaması oldukça yaygındır. Bu yolla algoritma bütün süreç esnasında bulunan en iyi değerin sergilenmesini sağlar.

3.4. İki Değişkenli Fonksiyonun A ve B katsayılarının GA Kullanılarak Optimize Edilmesi

Problem Y(i) = A*Y(i-1) + B*u(i-1) +prbs (i)

Y(i) : Biliniyor

Y(i-1) : Biliniyor

U(i-1) : Biliniyor

Prbs(i) : Biliniyor

A : Bilinmiyor

B : Bilinmiyor

3.4.1. Probleme Genetik Algoritmanın Uygulanması

Adım 1: Popülasyonun Oluşturulması=Kromozom Sayısı X Gen Sayısı,

Adım 2: Popülasyondaki her bir bireyin uygunluğunun bulunması,

Adım 3: Uygunluğa göre seçim yapılması,

Adım 4: Seçilen kromozomların çaprazlanması,

Adım 5: Oluşan yeni kromozomların mutasyona tabi tutulması,

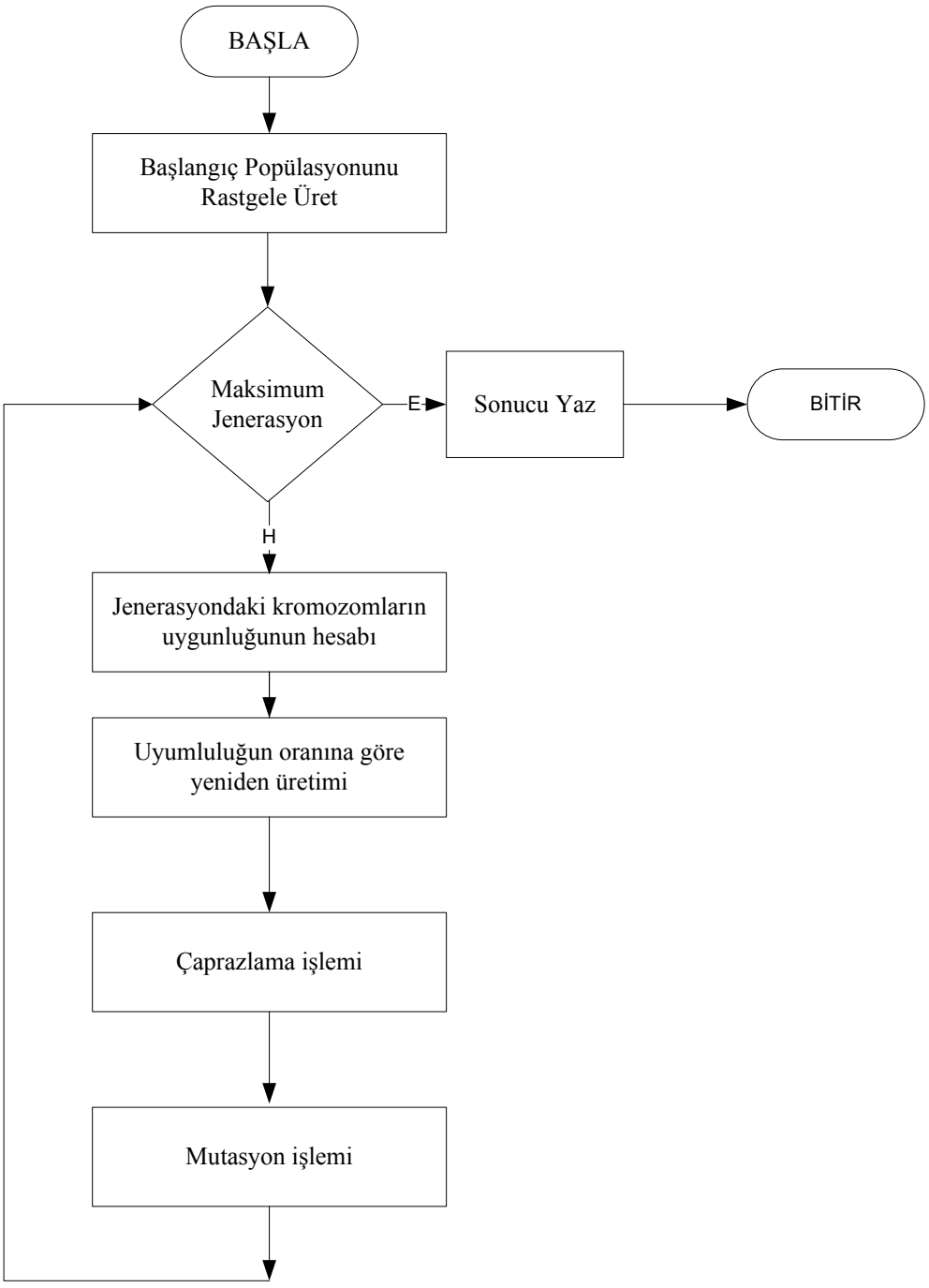
Adım 6: Yeni popülasyonun oluşturulması,

Adım 7: Adım 2 'ye geri dön (=Jenerasyon Sayısı).

3.4.2. Problemin Kodlanması

Problemde A katsayısı 18 bitle, B katsayısı 20 bitle ifade edilmiştir. Böylelikle her bir kromozomdaki gen sayısı 38 olacaktır. Kromozom sayısı 100, çaprazlama oranı P_c= 0,8, mutasyon oranı P_m= 0,5 alınmıştır. Jenerasyon Sayısı başlangıçta 20 olarak alınmıştır. Programın içeriği Ek-[1]'de sunulmuştur. Programın çalışmasını bitiren iki etken vardır. İlk etken en fazla tekrar sayısına ulaşılmasıdır. İkincisi ise en fazla başarısız tekrar sayısına ulaşılmasıdır.

Şekil 3.14.'de iki değişkenli fonksiyonun A ve B katsayılarının GA ile optimize edilmesi problemine uygulanan akış diyagramı gösterilmiştir.



Şekil 3.14. Probleme Uygulanan Genetik Algoritmanın Akış Diyagramı

Aramanın yapılmasında izlenen sıra yaklaşık değerlere bağlıdır. Genetik algoritma çalışırken mutasyon işlemcisi sayesinde her aşamada değişik çözümler oluşur. Bu çözümlerin bir kısmı bir süre daha popülasyonda kalır ve daha iyi çözümlerin popülasyonda çoğalmasını yavaşlatır. Bu etki en anlamsız bitlerde en fazladır; çünkü bu bitlere doğru ilerlendikçe çözümün kesinliği azalır. Çözümün arama aralığının solunda mı sağında mı olduğunu ortaya koymak genetik algoritma için oldukça kolaydır. Seçilen

çözümler yaklaşık olarak iki bölgeye de eşit dağılmıştır. Zaman ilerledikçe bir tarafta kalan parçaların sayısı artar. Genetik algoritma böylece çözümün yerini kabaca tahmin etmiş olur. Çözüm aralığı küçüldükçe iş güçleşir çünkü bu küçük aralıklarda eşit değerli noktaların sayısı daha fazladır. Çözüm aralığının küçülmesinin bir diğer sonucu da hata payını arttırmasıdır. Arama derinliği arttıkça hata payı da paralel olarak artar [40].

4. BÖLÜM

SU DAĞITIM ŞEBEKESİNİN OPTİMİZASYONUNDA GENETİK ALGORİTMANIN KULLANILMASI

4.1. Giriş

Günümüzde su kaynaklarının temini ve dağıtımı önemli bir problemdir. Elde edilen su depolardan her tüketim noktasına iletilebilmeli ve bu iletim sürekli olup maliyeti de düşük olmalıdır. Ülkemizde hemen her yerleşim biriminde bu problemle karşılaşılmaktadır.

Filtrelenmiş su pompalama işleminde amaç fonksiyonu en büyük yapacak çözümün, optimal proje büyüklüğü, proje bileşenlerinin optimal boyutları, optimal işletme kuralları gibi teknolojik, ekonomik, vb. uyarak belirlenmesi istenir.

Bu çalışmada GA, su dağıtım şebekesinin optimizasyonu için bölgesel bir probleme uygulanmıştır. Osmaniye İli Fakıuşağı Mevkiindeki su kaynaklarının belirli bir depoda toplanıp, şebekeye dağıtılması işleminde, GA uygulanmış ve klasik metoda göre çok iyi sonuçlar verdiği gözlemlenmiştir. Yapılan uygulama sonucunda günlük elektrik sarfiyatı 80kW'tan 65.6kW'a indirgenerek kazanç elde edilmiştir.

4.2. Su Dağıtım Şebekelerinin Genetik Algoritma İle Optimizasyonu

Son yıllarda su dağıtım şebekelerinin tasarımında optimizasyon tekniklerinin kullanımı giderek artmaktadır. Gelişmiş ülkelerde, su dağıtım sistemlerinin tasarlanması ve işletilmesinde, en iyi optimizasyon seçeneklerinden biri olan GA ön plana çıkmıştır. Yöntem basit ve uygulanması kolaydır. GA tekniğinde mevcut çözüm uzayı içerisinden çok küçük bir kısım kullanılarak optimum bir sonuç elde edilebilmektedir. 1990'lı yıllarda geliştirilen ve mühendislik problemlerinde optimizasyon amaçlı kullanılmaya

başlanan genetik algoritmalar, gerek yöntemin basitliği gerekse de gerçek sistemlerde uygulanabilirliğinin kolay olması nedenleriyle son zamanlarda su dağıtım şebekelerinin tasarımında ve işletilmesinde de yaygın olarak kullanılmaya başlanmıştır.

Su dağıtım şebekelerinde genetik algoritmalar çok çeşitli alanlarda optimizasyon amacı ile kullanılmaktadır. Bunlardan başlıcaları aşağıda verilmiştir:

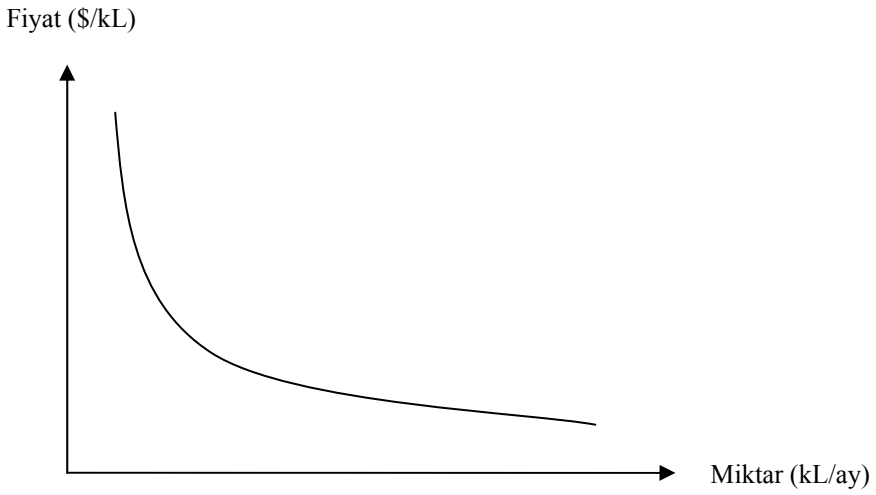
- Yeni su dağıtım şebekesi boyutlandırılmasında,
- Karmaşık ya da çok büyük su dağıtım sistemleri için pompa çalışma zamanlarının ayarlanmasında,
- Su haznelerinin, basınç vanalarının ve pompaların işletme noktalarının ayarlanmasında,
- Farklı kaynaklardan alınan suların minimum maliyetle istenilen su kalitesi standartlarını sağlamasında,
- Haznelerin sistemin dengelenmesi, yangın suyu ihtiyaç ve acil durumlar göz önünde bulundurularak boyutlandırılması ve optimum hazne yerinin belirlenmesinde,
- Sistemdeki sızıntı sularının minimize edilmesi için gereken kontrol vanalarının yerlerinin belirlenmesinde,
- Mevcut su dağıtım şebekelerinin genişletilmesi, rehabilitasyonu ve kalibrasyonu çalışmalarında,
- Ölçüm ve klorlama istasyonlarının optimum yerleşiminde.

GA tabanlı optimizasyon sayesinde, büyük şebekelerin inşaat ve işletme maliyetlerinde %15 ila %50 arası kazançlar sağlanmakta ve ayrıca hidrolik açıdan da daha iyi şebekeler elde edilebilmektedir [9].

4.3. Problemin Tanımlanması

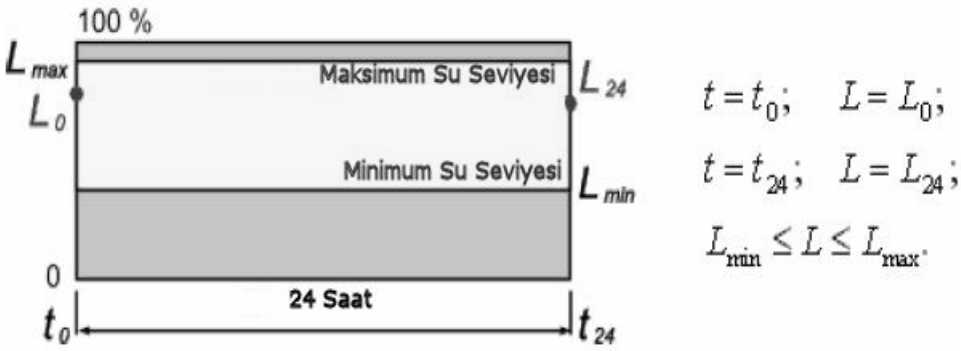
Bu çalışmada genetik algoritmalar, su dağıtım şebekesinin optimizasyonunda kullanılmaktadır. Birçok su filtreleme işleminde temiz su depolarından pompalar yardımıyla su şebekeye dağıtılır. Su genellikle kuyudan yükseklik oranı fazla olan başka bir depoya alınır [2]. Başlama ve bitiş durumlarındaki pompanın kontrolü genellikle yüksek ve düşük su seviyelerine bağlı olan tetikleme seviyeleri yardımıyla yapılır. Çalışmada öncelikle GA ile tetikleme seviyelerinin tespiti belirlenmiştir. GA, karar

değişkenleri ve bunlara ait muhtemel seçimler içerecek yönde bir sayı dizisi kullanır. Pompa optimizasyonu için burada geliştirilmiş olan genetik algoritma uygulamasında, pompalama suyun her kL'si başına pompalama maliyeti ile orantılı olarak her dizinin uygunluğu belirlenir. Toplam maliyet, tanımlanan sınırlamalarla ilgili olarak istenmeyen maliyetleri de içine alacak şekilde, 24 saat müddetince pompalama işlemi için harcanan elektrik maliyeti ile hesaplanır. GA sürecinde, en uygun sonuçların varlığının sürdürülmesi ve bu sonuçların çaprazlama ile karıştırılması, GA yönteminin sürekli daha uygun çözümlerin elde edilmesine imkan sağlar. GA'lar, lineer olmayan optimizasyon problemleri için oldukça verimli bir araştırma metodudur. GA araştırması dizilerin bir popülasyonunun rastgele oluşturulmasıyla başlar. Bu dizilerden her biri, pompalama maliyetini belirleyen faktörlerin seçimine ait muhtemel bir çözümü içerir. GA yöntemi, 24 saatlik peryot üzerinden, bir dakikalık bir zaman aralığındaki istekler ve tank seviyelerindeki değişimi simüle etmek için EPANET'i çalıştırmaktadır. EPANET bilgisayarda şebeke modeli kurmakta kullanılan ABD (Amerika Birleşik Devletleri)'de hükümete ait Amerikan Çevre Koruma Ajansı (EPA) tarafından geliştirilen bir programdır [41]. Her EPANET modelinde, tank için gerçek fiziksel maksimum ve minimum seviyelerinin altında ve üstünde tampon oluşturulmuştur. Su talebi ve maliyet arasındaki ilişki Şekil 4.1.'de gösterilmiştir [42].



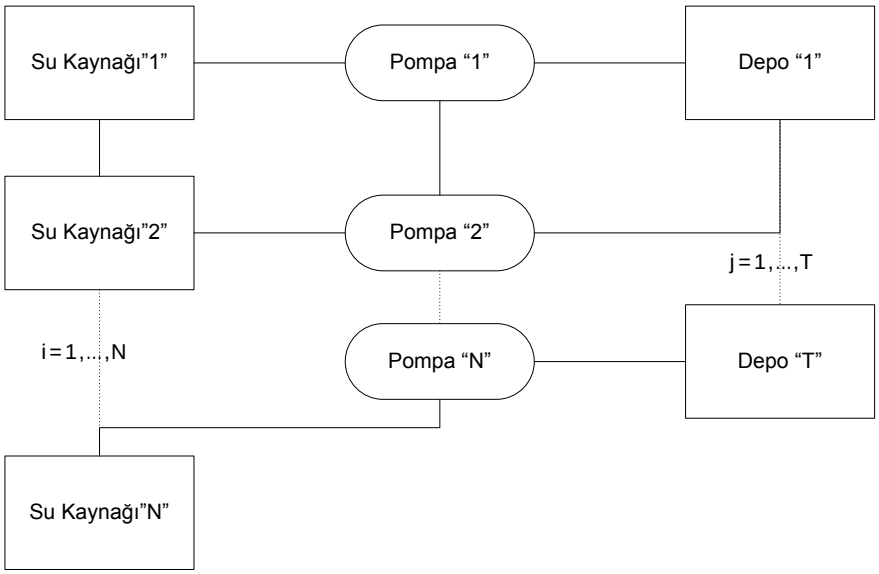
Şekil 4.1. Su Talebi ve Maliyet Arasındaki İlişki

Şekil 4.2.'de sistemdeki tanka ait başlangıç değerleri ve sınırlamalar verilmiştir.



Şekil 4.2. Sistemdeki Tanka ait Başlangıç Değerleri ve Sınırlamalar

Şekil 4.3.'de su şebekesi dağıtım sistemi modeli görülmektedir.



Şekil 4.3. Su Şebekesi Dağıtım Sistemi Modeli

İçme suyu şebekelerinin optimizasyonu problemlerinde amaç fonksiyonu genellikle toplam maliyetin minimize edilmesiyle oluşturulur. Yani amaç fonksiyonu Eşitlik 4.1'de verildiği gibidir.

$$\text{Amaç Fonksiyonu} = F_{maliyet} + F_{ceza} \tag{4.1}$$

Amaç fonksiyonu genel anlamda fonksiyon minimizasyonunu içermektedir. Eşitlik 4.1'e göre problem için aşağıdaki tanımlamalar yapılabilir [43].

$$C=f(x) \tag{4.2}$$

$$\text{Minimize } \{f(x)\}=\sum_{i=1}^{24} x_i(t)\,;\ i=1,2,\ldots,N \tag{4.3}$$

Eşitlik 4.4'de bir gün boyunca sistemde kullanılan su kaynakları görülmektedir.

$$x_i(t)\ (i=1,2,\ldots,N,\ t=1,2,\ldots,24) \tag{4.4}$$

Eşitlik 4.4 $x_{i\min} \le x_i(t) \le x_{i\max}$ ilişkisi yardımıyla vektör formda yazılırsa Eşitlik 4.5 elde edilir.

$$x=\left[x_1(1), x_2(2),..., x_1(24), x_i(1), x_i(2),..., x_i(24),..., x_N(1),..., x_N(24)\right]^T \tag{4.5}$$

Depodaki su seviyesi Eşitlik 4.6'daki gibi verilmektedir.

$$L_{j\min} \le L_j(t) \le L_{j\max} \quad (j=1,\ldots,T,\ t=1,\ldots,24) \tag{4.6}$$

$L_j(t)$: t zamanında j tankındaki su seviyesini göstermektedir. Su seviyelerine ait ceza fonksiyonu Eşitlik 4.7 ve 4.8'de verilmiştir.

$$C_j(t)^T=\begin{cases} 0 & L_{j\min} \le L_{j\max} \\ a_T\left[L_{j\min}-L_j(t)\right]^2 & L_j(t) < L_{j\min} \\ a_T\left[L_j(t)-L_{j\max}\right]^2 & L_j(t) > L_{j\max} \end{cases} \tag{4.7}$$

$$C_j^{\ T}=\sum_{t=1}^{24} C_j(t)^T \quad (j=1,\ldots,T) \tag{4.8}$$

Eşitlik 4.7'de verilen a_T su değişiminden tanktaki lineer olmayan su değişiminden kaynaklanan karasızlığı temsil etmektedir. Pompalama maliyeti, Eşitlik 4.9'da gösterildiği gibi güç maliyetlerinin hesaplaması ile bulunmaktadır.

$$P=\frac{\gamma Q H_P}{e_p e_m} \tag{4.9}$$

P : Güç

γ : Yoğunluk

Q : Debi (Gpm)

H_p : Pompa Yüksekliği

e_p : Pompa Verimi

e_m : Motor Verimi

Pompaya ait motor verimliliği %97 alınmıştır. Yapılan gözlemlere göre sistemde iki güç tarifesi mevcuttur.

- 07:00-23:00 arası 14.5 kW
- 23:00-07:00 arası 6.2 kW

GA ile tetikleme seviyelerinin optimizasyonu için iki karar değişkenini tanımlamada ikili kodlanmış sayı dizileri kullanılmaktadır. Bu probleme ait GA yöntemine dahil edilen sınırlamalar şunlardır:

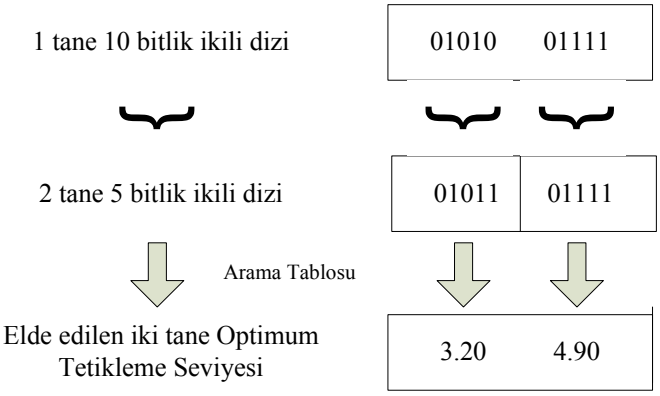
1. Üst tetikleme seviyesi (pompaların çalışması için), alt tetikleme seviyesinden (pompaların kapatılması için) daha küçük olmamalıdır. Ga ile yapılan simülasyonda böyle bir durumla karşılaşılırsa, güç maliyetine ceza maliyeti de eklenecektir.
2. Tanktaki su seviyesinin minimum su seviyesinin altına (%49) ve maksimum seviyenin (%96) üzerine çıkmasına izin verilmemiştir. Böyle bir durumla karşılaşılırsa ceza maliyeti uygulanır.

Şekil 4.4.'de gösterildiği gibi alt ve üst tetikleme noktaları 10 bitlik bir dizi ile temsil edilmektedir. Her bir tetikleme seviyesi dizideki 5 bitle temsil edilir.

Bu çalışmada temiz su deposunun yüksekliği, olası tetikleme seviyelerine göre 0.5 metreden 5.2 metreye kadar 0.1 ve 0.2 metrelik artırımlarla 10 bitten oluşan 32 kombinasyonla gösterilmiştir. Genetik algoritma uygulamasında rulet tekerleği seçme tekniği kullanılmıştır. Algoritmada, çaprazlama işleminde iki noktalı çaprazlama kullanılmış ve çaprazlama oranı 1.0 olarak alınmıştır. Popülasyon büyüklüğü 100 olarak alındığında 1024 olası tetikleme kombinasyonu bulunmuştur.

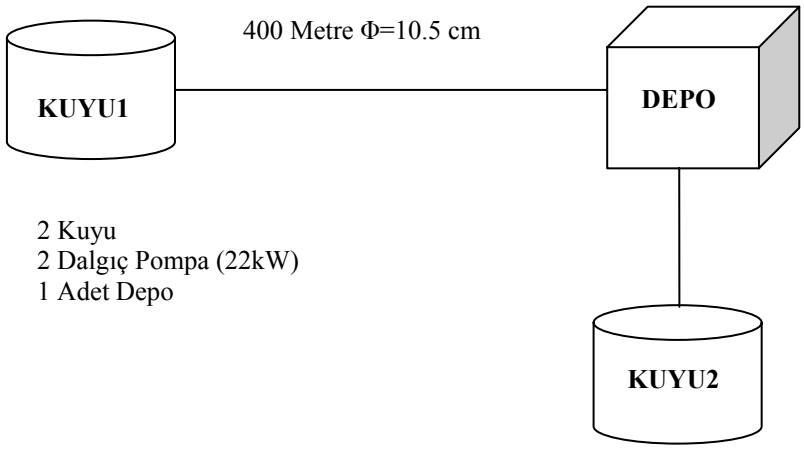
Sistemde iki adet kuyu ve bir adet depo bulunmaktadır. Kuyulardaki su, dalgıç pompalar yardımıyla depoya aktarılmaktadır. Klasik metoda göre su ünitesinde görevli bir kişi, sistemdeki elemanları (motor, dalgıç pompa, vb.) kendisi (manuel-el ile) kontrol etmektedir. Depoda su bittiği zaman dalgıç pompaları çalıştırmakta, aynı şekilde

depo dolduğu zaman, motorları durdurmakta ve şebekeye su dağıtan motorları çalıştırmaktadır.



Şekil 4.4. 10 Bitlik Dizinin İki Tane Tetikleme Seviyesine Dönüştürülmesi

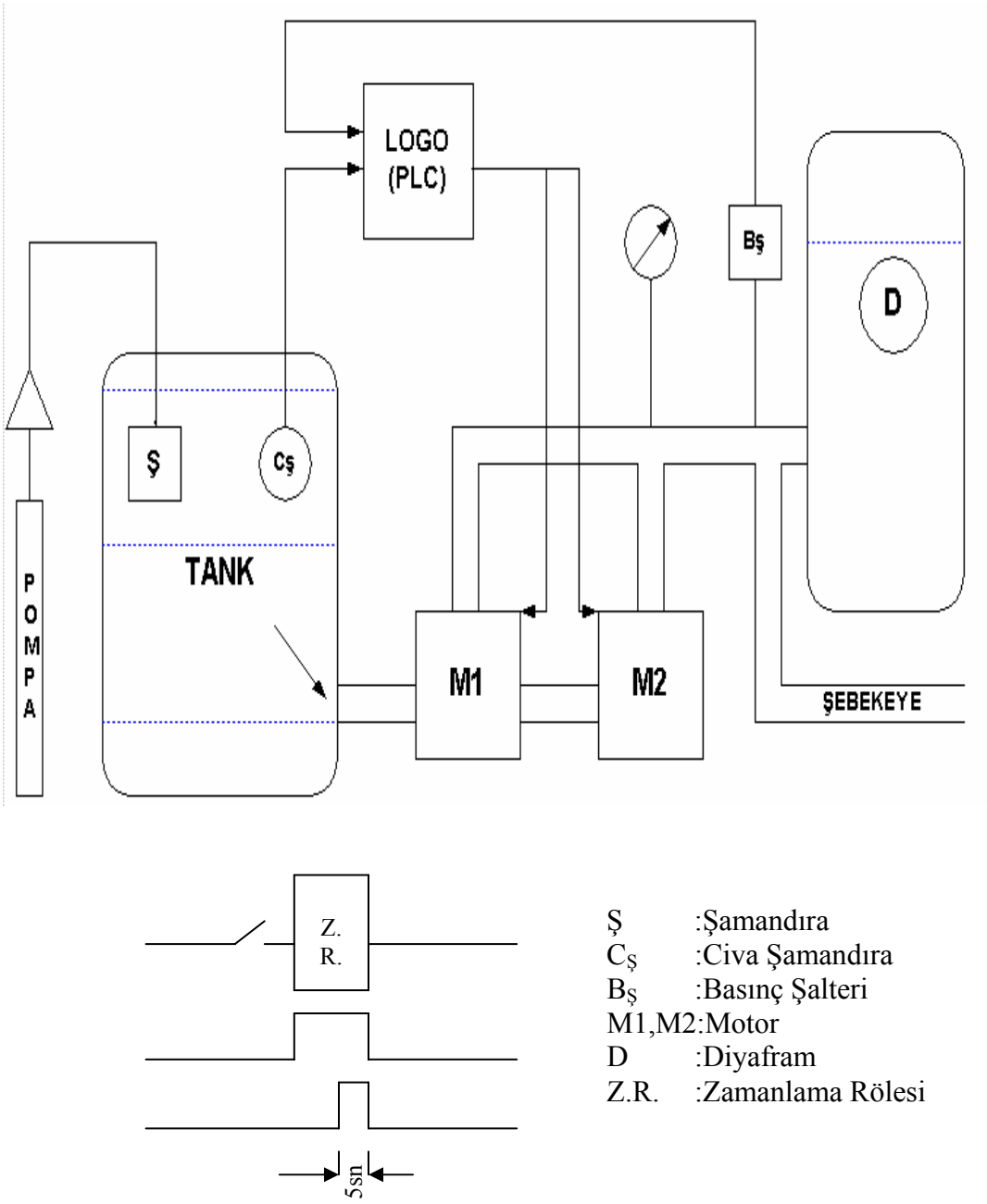
Seçilen bölgedeki su kaynakları Şekil 4.5.'de gösterildiği gibidir. Problemi klasik metottan kurtarmak ve daha iyi sonuçlar elde etmek için sezgisel algoritmalardan genetik algoritmaların kullanılması tercih edilmiştir. EPANET paket programı vasıtasıyla sistemdeki elemanlar programa aktarılmış ve optimizasyon işlemi gerçekleştirilmiştir [41].



Şekil 4.5. Osmaniye Fakıuşağı Mevkii Su Kaynakları

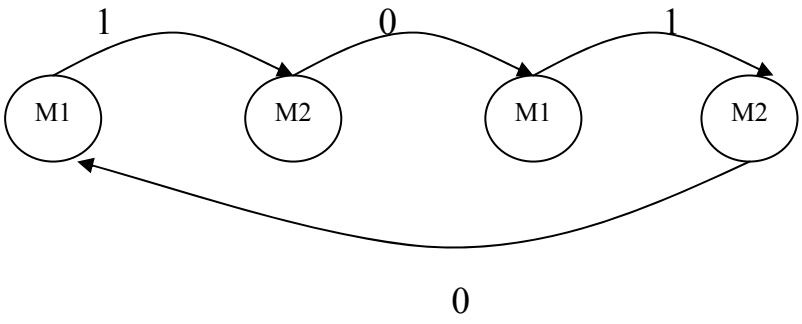
Optimizasyon işlemi sonrasında elde edilen sonuçlar Şekil 4.6.'daki Osmaniye İli Fakıuşağı Mevkii kuyu suları dağıtım ünitesine uygulanmıştır. Sistemin kontrolü klasik metottaki kullanıcı faktörü ortadan kaldırılarak PLC ile gerçekleştirilmiştir. Basınç şalteri ve civalı şamandıradan gelen bilgiler, 5 sn. gecikmeyle PLC' ye aktarılmıştır. Yapılan gözlemlere göre 5 sn. yeterli olmaktadır. 5 sn.' lik gecikme için zaman rölesi

kullanılmış olup bu gecikme, ani dalgalanmalarda sistemin motorlarının deforme olmaması için düşünülmüştür. Yani 5 sn. gecikme ile sistemdeki su seviyesi kararlı hale gelmektedir.



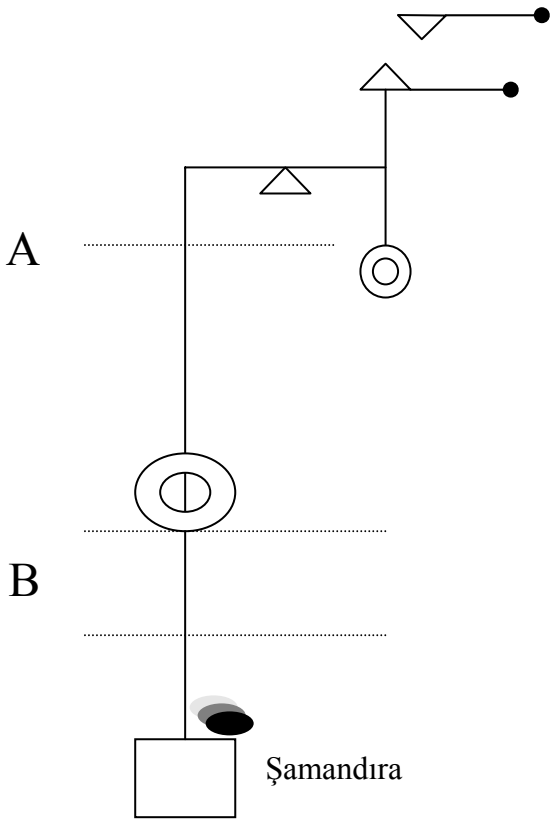
Şekil 4.6. Osmaniye Fakıuşağı Mevkii Kuyu Suları Dağıtım Ünitesi

PLC' ye girilen basınç şalteri ve şamandıra bilgisi ile sistem çıkışındaki M1 ve M2 motorlarının Şekil 4.7.'deki gibi sırasıyla çalıştırılması sağlanmıştır. Bu şekilde motorlar dinlendirilerek çalıştırılmış olması olası arızaların da azalmasına imkan sağlamıştır.



Şekil 4.7. Su Ünitesinin Kontrol Sistemi

Şekil 4.8.'deki A ve B noktaları alt ve üst tetikleme noktalarını göstermektedir. Şamandıradan gelen bilgiye göre motorlar çalışmakta veya durmaktadır.

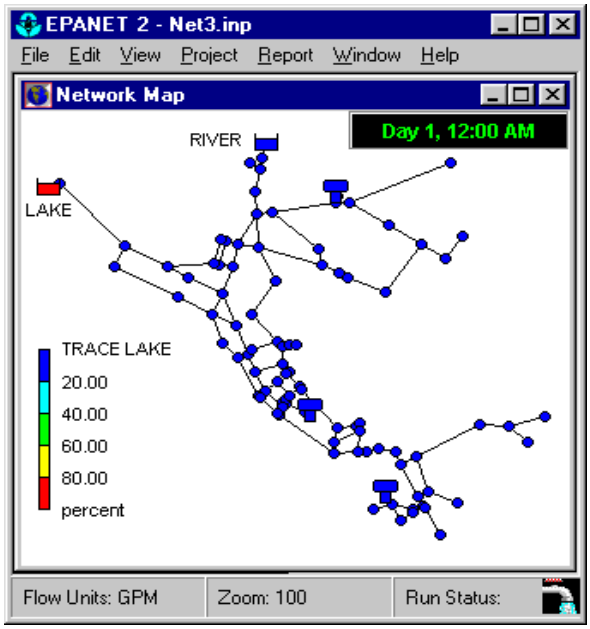


A: Optimize edilecek üst tetikleme noktası
B: Optimize edilecek alt tetikleme noktası

Şekil 4.8. Şamandıranın İç Yapısı, Alt ve Üst Tetikleme Noktaları

Epanet içme suyu dağıtım şebekesi hesap programıdır. Dünya genelinde, su dağıtım şebekelerinin tasarımı ve işletilmelerinde, ençok Epanet adlı yazılım kullanılmaktadır. Epanet2 programı "gradient yöntemi" ile çözüm yapan bir su dağıtım şebekesi hidrolik hesap programıdır. Epanet'te belirli bir zaman ve belirli bir noktada şebekenin hidrolik durumunu karakterize eden debi sürekliliği ve yük kaybı eşitliklerini hesaplamak için kullanılan yöntem hibrit bir nokta-ağ yaklaşımı olarak söylenebilir.

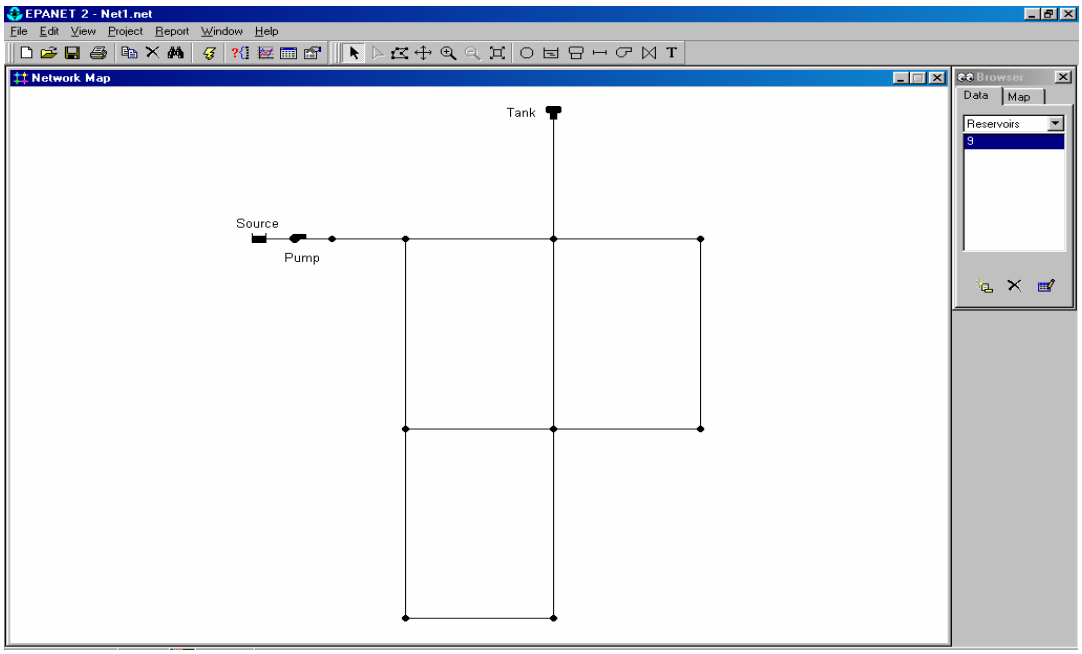
Epanet programına boru uzunlukları, çapları, pürüzlülük katsayıları, düğüm noktalarında su talepleri, arazi kotları, su alma yapıları, pompa, vana gibi ekipmanlarla ilgili veriler girilerek şebekede tek kararlı hidrolik durum (steady state simulation) ya da birden fazla hidrolik durumlarına (extended period simulation) göre şebeke hidrolik hesapları yapılabilmektedir. Şekil 4.9.'da optimizasyonda kullanılan EPANET 2.0 paket programında yer alan iki kaynaklı bir sistemin çözümünün ekran çıktısı görülmektedir.



Şekil 4.9. İki Kaynaklı Sistemin Epanet Programıyla Modellenmesi

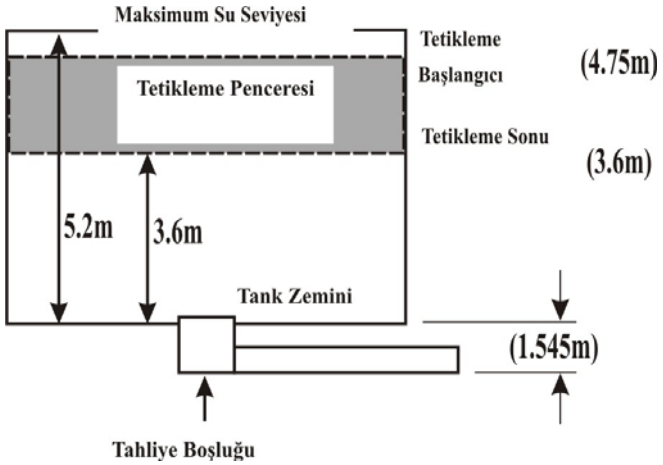
İçme suyu şebekelerinin projelendirilmesinde, önce şebekenin geometrisi ve dağıtacağı debiler belirlenir, daha sonra çaplar seçilerek hidrolik benzeşim modeli ile çözüm yapan bir program yardımıyla hız ve basınçlar hesaplanır. Çaplar değiştirilerek şebekenin kritik noktalardaki basınç ve hız değerlerini sağlayacak minimum maliyetli çözüme ulaşıncaya kadar çap değiştirme işlemi yapılır ve şebekedeki pompa çalışma zamanı değiştirilir. Bu tür klasik çözüm yaklaşımlarında optimum çözüm için uzun ve yorucu

çalışmalar yapılması gerekmektedir. Bu aşamaların başarı düzeyi büyük ölçüde proje mühendisinin deneyimine bağlıdır. Ayrıca bu konuda en deneyimli mühendis bile global optimum çözümü elde edebileceğini garanti edemez. Genetik algoritmalar ise deneme yanılma aşamasını evrim teorisine dayandırarak, yine bir hidrolik benzeşim modeli yardımı ile aylar sürecek bir çalışmayı saatler mertebesine indirebilmektedir. Üstelik genetik algoritma ile bulunan çözümler proje mühendislerinin bulduğu klasik çözümlere kıyasla çok daha ekonomik ve hidrolik açıdan da daha uygun şebekeler olmaktadır [1,9]. Şekil 4.10.'da su şebekesinin programa tanıtılması görülmektedir.



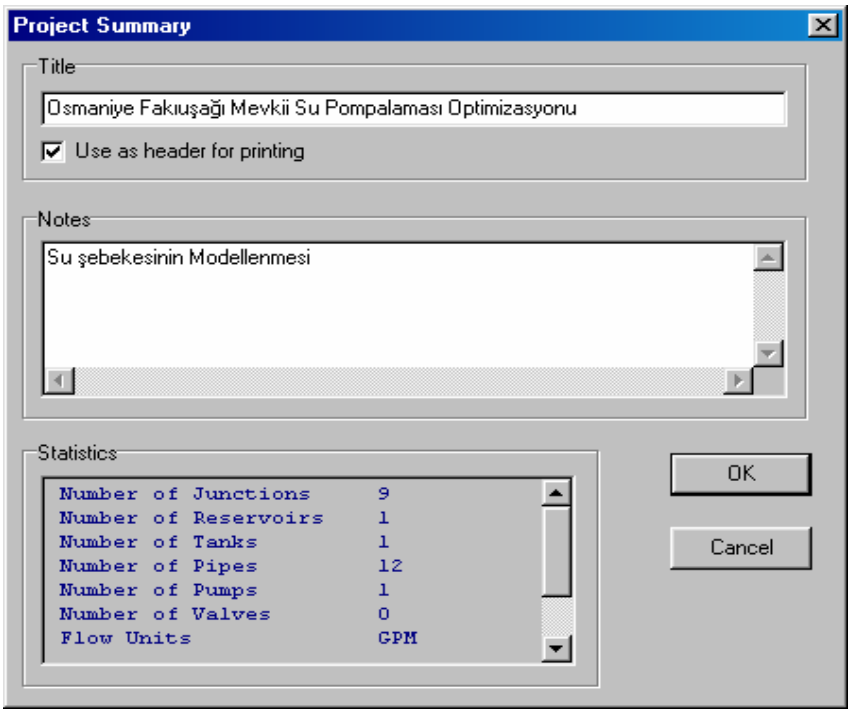
Şekil 4.10. Su Şebekesinin Programa Tanıtılması

Şekil 4.11.'de, var olan pompalama tetikleme seviyeleri görülmektedir. Maksimum su seviyesinin altında, alt ve üst tetikleme seviyesine göre tetikleme penceresi oluşmuştur.



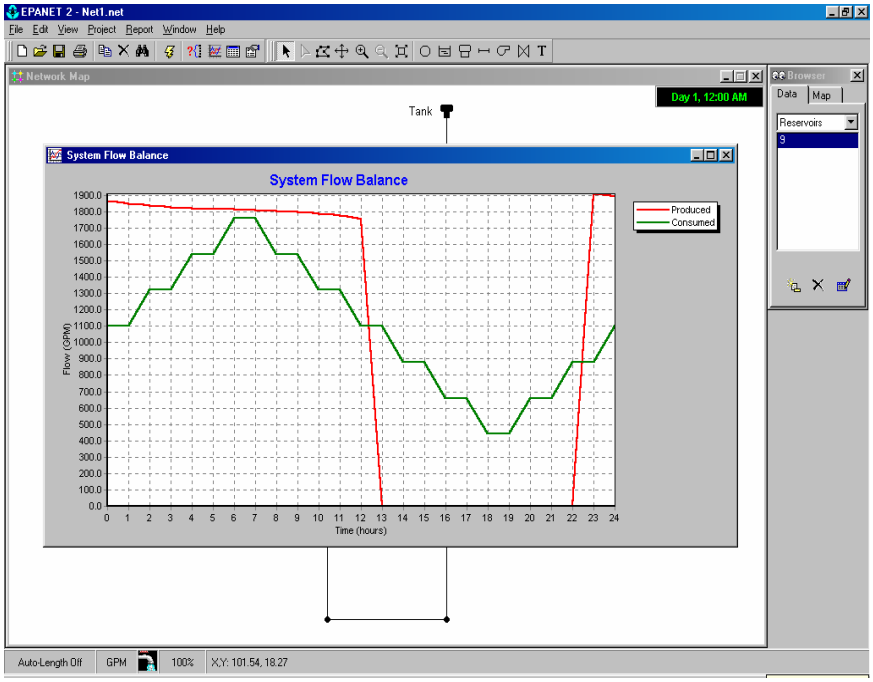
Şekil 4.11. Temiz Su Depolama Tankında Var Olan Pompalama Tetikleme Seviyeleri

Şekil 4.12.'de şebekenin programa tanıtılmasında kullanılan elemanların özeti görülmektedir.



Şekil 4.12. Şebekede Kullanılan Elemanların Özeti

Yapılan simülasyona göre üretilen ve tüketilen su miktarı Şekil 4.13.'de verilmiştir.



Şekil 4.13. Üretilen ve tüketilen su miktarı

Yapılan optimizasyon işlemi sonucunda alt ve üst tetikleme noktaları sırasıyla 3.2 ve 4.9 metre olarak bulunmuştur. Alt seviye tetikleme noktasının 3.2 metreye, üst seviye tetikleme noktasının 4.9 metreye değişimi sonucunda %18'lik kazanç elde edilmiş ve böylelikle günlük elektrik sarfiyatının 80 kW'tan 65.6 kW'a indirgeneceği belirlenmiştir. Klasik metottaki kullanıcı faktörü ortadan kaldırılmış ve böylelikle sisteme uyarlanan PLC programı ile sistem sorunsuz olarak çalışmıştır.

5. BÖLÜM

İRDELEME VE SONUÇ

5.1. Giriş

Mühendislik problemlerinin çözümünde göz önüne alınan alternatiflerden, pratikte uygulanması mümkün olanlar arasında, en iyi çözümün bulunması optimizasyon tekniklerinin kullanılmasıyla mümkün olmaktadır. Kaynaklar sınırlı olduğu zaman, bunların verimli bir şekilde kullanılmasında optimizasyon tekniklerinin geliştirilmesi önem arz etmektedir. Çoğu problemlerde, birden fazla amacın aynı anda gerçekleştirilmesi arzu edilmektedir. Bu gibi problemler çok değişkenli optimizasyon problemleri olarak adlandırılmakta ve çeşitli optimizasyon metotlarıyla çözülmektedir. Sezgisel optimizasyon tekniklerinden biri olan GA' lar çözümü çabuk bulması ve problemin doğasıyla ilgili bilgiye ihtiyaç duymaması gibi özelliklerinden dolayı birçok problemin çözümüne yardımcı olmuştur. Bu özellikler göz önüne alınarak bu tez çalışmasında GA kullanılmıştır.

Bu çalışmada ilk olarak Su Filtreleme İşleminde Pompalamanın Optimizasyonu için Genetik Algoritma kullanılmıştır. Daha sonra sistemin kontrolü klasik metottaki kullanıcı faktörü ortadan kaldırılarak PLC ile gerçekleştirilmiştir. Basınç şalteri ve civalı şamandıradan gelen bilgiler, 5 sn. gecikmeyle PLC' ye aktarılmıştır. 5 sn.' lik gecikme için zaman rölesi kullanılmış olup bu gecikme, ani dalgalanmalarda sistemin motorlarının deforme olmaması için düşünülmüştür. Yani PLC sayesinde 5 sn. gecikme ile sistemdeki su seviyesi kararlı hale gelmektedir. Genetik algoritmanın başarısı, kısaca problemi parçalayarak çalışmasıyla açıklanabilmektedir. Bu aşamada bir dinamik programlama yöntemi gibi çalışmaktadır. Genetik algoritmanın bu yöntemlere göre daha başarılı olmasının en önemli nedeni problemin boyutundan bağımsız olmasıdır. Değişken sayısı ve bu değişkenlerin değerlerinin sonuçlara etkisi araştırma yöntemini değiştirmez. Araştırma arama uzayını dinamik olarak böler. Algoritmanın çalışması sırasında biçim dosyasında belirtilen özellikler, rasgele sayı üreticinin ilk değeri gibi

çevre etkenlerine bağlı olarak arama uzayı farklı şekilde taranır. En alt düzeyde işlemler oldukça karmaşık ve değişkendir; ancak üst düzey bir bakışla değişik yollarla da olsa genetik algoritma sonuca ulaşır.

Tez çalışmasında Osmaniye İli Fakıuşağı Mevkiindeki su kaynaklarının belli bir depoda toplanarak, şebekeye dağıtılması işleminde GA uygulamasının simülasyonu (modellenmesi) gerçekleştirilmiştir. Yapılan optimizasyon işlemi sonucunda alt ve üst tetikleme noktaları sırasıyla 3.2 ve 4.9 metre olarak bulunmuştur. Sistemde iki adet kuyu ve bir adet depo bulunmaktadır. Kuyulardaki su, dalgıç pompalar yardımıyla depoya aktarılmaktadır. Klasik metoda göre su ünitesinde görevli bir kişi, sistemdeki elemanları (motor, dalgıç pompa, vb.) kendisi kontrol etmektedir. Depoda su bittiği zaman dalgıç pompaları çalıştırmakta, aynı şekilde depo dolduğu zaman, motorları durdurmakta ve şebekeye su dağıtan motorları çalıştırmaktadır. Problemi klasik metottan kurtarmak ve daha iyi sonuçlar elde etmek için sezgisel algoritmalardan genetik algoritmaların kullanılması tercih edilmiştir. Klasik metottaki kullanıcı faktörü ortadan kaldırılmış ve böylelikle sisteme uyarlanan PLC programı ile sistem sorunsuz olarak çalışmıştır. Yapılan uygulama sonucunda gözlemlere göre %18'lik kazanç elde edilmiş, günlük elektrik sarfiyatı 80 kw' tan 65.6 kw' a indirgenmiştir. Klasik metottaki kullanıcı faktörü ortadan kaldırılmış ve sisteme uyarlanan PLC programı ile sistem sorunsuz olarak çalışmıştır.

Bu tez çalışmasında gerçekleştirilen uygulamalar geliştirilerek farklı bölgelerdeki su depoları için de uyarlanabilir. Algoritmaya su depolarının durumları ile ilgili eklemeler kolaylıkla yapılabilir. Böylelikle sistemin daha verimli olarak çalışması sağlanabilir.

Ek-1:

İki Değişkenli Fonksiyonun A ve B Katsayılarının GA Kullanılarak Optimize Edilmesinde Kullanılan Kaynak Kodlar,

```
clc

% uygulamadaki kromozom sayısı=ks
% herbir kromozomdaki gen sayısı=gs
% K'daki A/B bitlerinin sayısı=abit/bbit
% jenerasyon sayısı=js
% çaprazlama oranı=pc
% mutasyon oranı=pm


js=20;
ks=100;
gs=38;
pc=0.8;
pm=0.5;
abit=18;
bbit=20;
say=1;


U = sign(randn(ks,1));
prbs = randn(ks,1);

for i=2:ks
    Y(1)=prbs(1);
    Y(i)=0.2*Y(i-1)+0.7*U(i-1)+prbs(i);
    Y=(Y)';
end
%end

%pause
% Başlangıç popülasyonu olarak K matrisi oluşturulması.
%-------------------------------------------------------
% K matrisinden A ve B matrisleri çekilir.
% Bunların kodu çözülür ve E ve F matrisleri olarak saklanırlar.

K=[round(rand(ks,gs))];
jen=1;
for p=1:js,
%C2=0.0;
for n=1:ks
    A(n,:)=K(n,1:abit);
    C=A(n,:);
```

```
    c=decode(C,0,1.5,abit);
    D(n)=c;
% çaprazlama noktasının (cn) üretilmesi.
%-----------------------------------

cn=round(gs*rand(1));
if cn==0 | cn==1,
    cn=cn+2;
end;
if cn==gs,
    cn=cn-2;
end;

% çaprazlamanın yapılması
%------------------------

ka=0.0;
for f=1:2:c1,
    xa=R(f);
    ya=R(f+1);
    XA=K(xa,:);
    YA=K(ya,:);
    X1=XA(:,1:(cn-1));
    X2=YA(:,cn:gs);
    Y1=YA(:,1:(cn-1));
    Y2=XA(:,cn:gs);
    XA=[X1,X2];
    YA=[Y1,Y2];
    K(xa,:)=XA;
    K(ya,:)=YA;
    ka=ka+1;
end

'çaprazlamaya uğrayan kromozom çifti sayısı=',disp(ka)

% mutasyonun uygulanması
% mutasyon oranı=pm=0.01
%-----------------------


% mutasyona uğrayacak olan genlerin seçimi
%-----------------------------------------

ro=rand((ks*gs),1);
k=1;
for t=1:(ks*gs),
    while ro(t)>(pm) & t<(ks*gs),
        t=t+1;
    end;
```

```
    M(k)=t;
    k=k+1;
end;

    P1=(D)';
%D(1)=Dop;

    B(n,:)=K(n,(abit+1):gs);
    E=B(n,:);
    e=decode(E,0,1.5,bbit);
    F(n)=e;
    P2=(F)';
%   F(1)=Fop;
end

%D(1)=Dop;
%F(1)=Fop;


% uygunluk değerlerinin hesaplatılması.
%---------------------------------------
% istenen parametreler P1 ve P2'dir.
for k=2:ks
    YG(k)=P1(k)*Y(k-1)+P2(k)*U(k-1)+prbs(k);
    hata(1)=prbs(2);
    hata(k)=abs(Y(k)-YG(k));
%   uyg(1)=mu;
    uyg(k)=100-hata(k);
        if hata(k)<=0.5
            yd1=P1(k)*Y(4)+P2(k)*U(4)+prbs(5);
            yd2=P1(k)*Y(5)+P2(k)*U(5)+prbs(6);
            yd3=P1(k)*Y(6)+P2(k)*U(6)+prbs(7);
            yd4=P1(k)*Y(7)+P2(k)*U(7)+prbs(8);
            yd5=P1(k)*Y(8)+P2(k)*U(8)+prbs(9);

            hatad1=abs(Y(5)-yd1);
            hatad2=abs(Y(6)-yd2);
            hatad3=abs(Y(7)-yd3);
            hatad4=abs(Y(8)-yd4);
            hatad5=abs(Y(9)-yd5);
```

KAYNAKLAR

1. Karaboğa, N., Loğoğlu, A., Genetik Algoritmaların Su Filtreleme İşleminde Pompalamanın Optimizasyonu İçin Kullanılması, Pamukkale Üniversitesi, 3.Otomasyon Sempozyumu, 221-224, 11-12 Kasım 2005.
2. Simpson A.R., Sutton D., Keane D. and Sherriff S., Optimal Control of Pumping at a Water Filtration Plant Using Genetic Algorithms, Conference on Computing and Control for the Water Industry (CCWI ’99), Exeter, UK, 1999.
3. Ackley, D. J., An Empirical Study of Bit Vector Function Optimization, in Genetic Algorithms and Simulated Annealing, edited by L. Davis, London, Pitman, 194-200, 1987.
4. Wood, D. J., Funk, J. E., Hydraulic Analysis of Water Distribution Systems, in Water Supply Systems, State of the Art and Future Trends, E. Cabrera and F. Martinez, eds., Computational Mechanics Publications, Southampton, 41-85,1993.
5. Alperovits, E., Shamir, U., Design of Optimal Water Distribution Systems, Water Resour. Res., 13(6), 885-900, 1977.
6. Simpson, A. R., Optimization of Design of Water Distribution Systems Using Genetic Algorithms, Slovenian Society of Hydraulic Research, Seminar Series, Vol.1, Ljubljana, Slovenia, 10-14, 2000.
7. The Optimatics Letter, Issue, No:12, Advances in Optimization for Water Distribution System Design & Operations, 2001.
8. Kahraman, A. M., Özdağlar, D., Su Dağıtım Sistemlerinin Genetik Algoritma İle Optimizasyonu, DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi, Cilt: 6, Sayı: 3, s. 1-18, 2004.
9. Alperovits, E., Shamir, U., Design of Optimal Water Distribution Systems, Water Resour. Res., 13(6), 885-900, 1977.
10. Fujiwara, O., Khang, D. B., A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks, Water Resour. Res., 26(4), 539-549,1990.
11. Savic, D. A., Walters, G. A., Genetic Algorithms for Least-Cost Design of Water Distribution Networks, J. Water Resour. Plng. and Mgmt., ASCE, 123(2), 67-77,1997.

12. Abebe, A.J., Solomatine, D.P., Application of Global Optimization to the Design of Pipe Networks, 3rd International Conferences on Hydroinformatics, Copenhagen, Denmark, 989-996,1998.
13. Cunha, M.D.C., Sousa, J., Water Distribution Network Design Optimization: Simulated Annealing Approach, Journal of Water Resources Planning and Management, Vol. 125, No. 4, 215-221, 1999.
14. Eusuff, M. M., Lansey, K. E., Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, J. Water Resources Planning and Management, ASCE, Vol. 129, No. 3, 210-255,2003.
15. Liong, S., Md. Atiquzzaman, Optimal Design of Water Distribution Network Using Shuffled Complex Evolution,Journal of The Institution of Engineers, Singapore Vol. 44, Issue 1, 2004.
16. Karaboğa D., Yapay Zeka Optimizasyon Algoritmaları, s.1-20, Nobel Yayınevi, Ankara, 2004.
17. Barker, T.G., The Application of PROLOG++ to the ELSYM Departmental Timetable, M. Sc. Thesis, University of Wales, College of Cardiff, 1991.
18. Belew, R.K.& Booker, L.B.(eds), Procedings of the 4th International Conference on Genetic Algorithms, 13-16th July 1991 at the University of California, San Diego, Morgan Kaufmann Publishers, California,1991.
19. De Jong, K.A., Genetic Algorithms: A Ten Year Perspective in Grefenstette, J.J. (ed), Procedings of the 1st International Conference on Genetic Algorithms and their Applications. June 24-26 1985 at Carneige-Mellon University, Pittsburgh, Lawrence Erlbaum Assoc, Publishers, 1985.
20. Choudbury, S.S, Computer Timetabling, M. Sc. Thesis, University of Wales, College of Cardiff, 1989.
21. Karaboğa, N., Sayısal Filtre Katsayılarının Genetik Algoritma Kullanılarak Yuvarlatılması, Doktora Tezi, Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Bölümü, Kayseri, 1994.
22. Holland, J.H., Adaptation in Natural and Artificial Systems (2nd Edition) MIT Press, Massechusetts,1992.
23. İşlier, A.A, İmalat Problemlerinde Genetik Algoritmalar, Osmangazi Üniversitesi, Eskişehir, Otomasyon Dergisi, 1995.

24. Ergül, A., Genetik Algoritmalar, PC Günlüğü Dergisi, 1994.
25. Davis, L. (editor) Genetic Algorithms and Simulated Annealing, Pitman Publishing, London, 1987.
26. Davis, L., Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York., 1991.
27. Mansfield, R.A., Genetic Algorithms, M. Sc. Thesis, University of Wales, College of Cardiff, 1990.
28. Davis, L. Job Shop Scheduling with Genetic Algorithms in Grefenstette, J.J., (ed) Procedings of the 1st International Conference on Genetic Algorithms and their Applications, June 24-26 1985 at Carneige-Mellon University, Pittsburgh, Lawrence Erlbaum Assoc, Publishers, 1985.
29. Cleveland, G.A. &Smith, S.F. Using Genetic Algorithms to Schedule Job Shop Releases. in Schaffer, D.J.(ed), Procedings of the 3th International Conference on Genetic Algorithms, June 4–7 1989 at George Mason University, Morgan Kaufmann Publishers Inc., California, 1989.
30. Syswerda, G. & Palmucci, J., The Application of Genetic Algorithms to Resource Scheduling. in Belew, R.K. & Booker, L.B. (eds), Procedings of the 4th International Conference on Genetic Algorithms, 13–16 July 1991 at the University of California, San Diego, Morgan Kaufman Publishers, California, 1991.
31. Syswerda, G., Schedule Optimisation using Genetic Algorithms in Davis, L. Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
32. Whitley, D., Starkweather, T. & Shaner, D., The Travelling Salesman and Sequence Scheduling, Quality Solutions using Genetic Edge Recombination in Davis, L., Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
33. Cohoon, J.P., Genetic Placement, IEEE Transactions on Computer-Aided Design, Vol. CAD–6., No.6., 1987.
34. Cohoon, J.P., Hedge, S.U., Martin, W.N. & Richards, D. Floorplan Design using Distributed Genetic Algorithms, IEEE Transactions, 1988.
35. Syswerda, G. Uniform Crossover in Genetic Algorithms in Schaffer, D.J. (ed), Procedings of the 3th International Conference on Genetic Algorithms,June 4–7

1989 at George Mason University, Morgan Kaufman Publishers, Inc. California, 1989.

36. Booker, L., Improving Search in Genetic Algorithms in Davis, L.(editor), Genetic Algorithms and Simulated Annealing, Pitman Publishing, London,1987.
37. Pham, D.T., Karaboğa, D., Intelligent Optimisation Techniques Springer –Verlag, London, Surrey, UK, 2000.
38. Etaner Uyar Ş., ITU (İstanbul Teknik Üniversitesi) Computer Engineering Department Artificial Intelligence Undergraduate Course Notes, Dec.12 - 2000.
39. Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer- Verlag New York, 1996.
40. Bağış, A., Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Bölümü, Kayseri,1996.
41. U.S. Environmental Protection Agency, www.epa.gov
42. Howe, C., W., and Linaweaver, F.P., The Impact of Price on Residential Water Demand and Its Relation to System Design and Structure, Water Resour. Res. 3(1) 13-32, 1967
43. Aral, M. M., Guan, J., Maslia M.L., Sautner, J.B., Reconstruction of Hydraulic Management of a Water-Distribution System Using Genetic Algorithms, Multimedia Environmental Simulations Laboratory, MESL-01-01, Georgia Institue of Technology School of Civil and Environmental Engineering, 2001

ÖZGEÇMİŞ

Ahmet Loğoğlu, 1975 yılında Kadirli' de doğdu. İlk ve orta öğrenimini Kadirli' deki çeşitli okullarda tamamladı. 1994 yılında girdiği Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü'nden 1998 yılında mezun oldu. Aynı yıl Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans eğitimine başladı. Halen Yüksek Lisans eğitimine devam etmektedir.

Tel : 0 328 813 7374
E-posta :alogoglu@gmail.com
Adres :Rahime Hatun M. Hilmi Karaboran Cd. Kırcılıoğlu Sitesi
B Bl. K:4 N:8 80000, OSMANİYE