

SU KAYNAKLARI SİSTEMLERİNDE DİNAMİK PROGRAMLAMA PROBLEMLERİNİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ

Ahmet BAYLAR¹, Mualla ÖZTÜRK¹ ve Abdussamet ARSLAN²

ÖZET : Genetik Algoritma tekniği, evrim sürecinden etkilenerek, canlılarda yaşanan genetik sürecin bilgisayar ortamında gerçekleştirilmesi işlemidir. İşlemler bilgisayar hafızasına depo edilmiş kromozomlar üzerinde icra edilmektedir. Çaprazlama operatörü vasıtasıyla, kromozomlar arasındaki genetik bilgi sürekli olarak değişmekte ve topluluğun başarısı artmaktadır.

Bu çalışmanın asıl amacı, su kaynakları sistemlerindeki dinamik programlama problemlerinin Genetik Algoritma ile hesaplanmasıdır. Bu maksatla, Genetik Algoritma esaslı bir optimizasyon programı geliştirilmiştir. Elde edilen sonuçlara göre Genetik arama metodunun hızlı ve hassas sonuçlar verdiği gözlenmiştir.

ANAHTAR KELİMELER: Genetik Algoritma, Optimizasyon, Dinamik Programlama

SOLUTION OF DYNAMIC PROGRAMMING PROBLEMS IN WATER RESOURCES SYSTEMS BY GENETIC ALGORITHM

ABSTRACT : Genetic Algorithm methodology is a genetic process treated on computer which is considering evolution process in the nature. The genetic operations takes place within the chromosomes stored in computer memory. By means of various operators, the genetic knowledge in chromosomes change continuously and success of the community progressively increases as a result of these operations.

The primary purpose of this study is calculation of dynamic programming problems in water resources systems by Genetic Algorithm. For this purpose a Genetic Algorithm based optimization program were developed. It can be concluded that the results obtained from the Genetic search based method give the precise and rapid results.

KEYWORDS: Genetic Algorithm, Optimization, Dynamic Programming

¹ Fırat Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 23279 ELAZIĞ
² Gazi Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 06690 ANKARA

I. GİRİŞ

İnsanođlu tarih boyunca hep dođayı taklit ederek belli hedeflere ulařmıřtır. Son yıllarda insanođlu tarafından yapılan bazı aralar neredeyse dođadaki benzerlerinin bire bir mekanik veya elektronik taklitleri řeklinde oldukları aıka grlmektedir. İnsanın dođasında olan bu davranıř řekli ile insan zekasının yođrulması sonucu elde edilenler, insanođlunu gnmzdeki teknoloji dzeyi ve modern toplum yapısına ulařtırmıřtır.

Mhendislik problemleri genellikle amaca ynelik olarak en iyi zm arayan lineer, lineer olmayan ve dinamik programlama problemlerinden ibarettir. Bu nedenle optimizasyon yntemleri mhendislik aısından zel bir nem arz etmektedir. Bilgisayarların yaygınlařmasından nce ok kk optimizasyon problemlerinin zmnde dahi byk glklerle karřılařılmıř, elle zme uygun birok optimizasyon yntemi geliřtirilmiřtir. Bilgisayarların yaygınlařması ile birlikte ođu problemler daha kolay ve daha kesin zmlere kavuřmuř, ancak byk ve karmařık problemlerin zmnde bilgisayarlar da yetersiz kalmıřtır. Yetersizliklerin temel nedenleri, problemlerin analitik zmlerinin olmayıřı ve bilgisayarlarla iterasyona dayalı zm yntemlerinde ortaya ıkan hız sorunlarıdır. Ortaya ıkan sorunlar insanları yeni zm yntemleri arayıřlarına teřvik etmiřtir.

Son yıllarda bilgisayar donanım ve yazılımlarının geliřmesine paralel olarak, dřnebiyen, ğrenebiyen ve karar verme yeteneđine sahip bilgisayarların yapılması en ok zlenen hedeflerden biri haline gelmiřtir. Kısmen gerekleřen bu hedeflerin kilometre tařlarını yapay zeka konuları oluřturmaktadır. Gndemi yođun olarak iřgal eden yapay zeka konularının bařlıcaları Genetik Algoritmalar, Yapay Sinir Ađları, Uzman Sistemler, Fuzzy Setleri, Isı Yayılım Simlasyonu ve Nesneye Ynelik Programlama gibi konulardır. Bu konuların hemen hemen hepsindeki en belirgin zellik, insan, dođa ve sosyal evre yapılarının taklit edilmesine dayanmalarıdır.

Genetik Algoritmalar, insan ve ekosistemdeki dođal geliřme, sosyal sistemleri taklit etme ve psikolojideki sonuları deđerlendirmeyi iine alan dinamik geliřimlerin modellenmesinden oluřmaktadır. Modellenen rneklerin bir zaman dilimi ierisinde ktden iyiye dođru gitmesi, bir optimizasyon probleminin bařlangı zmnden optimum zme dođru yaklařımını andırır. Genetik Algoritmalar zellikle

optimizasyon algoritmalarının yetersiz veya yavaş kaldığı birçok problemde sonuca ulaşabilmek için kullanılabilir en iyi araçlardan birisi, Genetik Algoritmalarıdır.

Su kaynakları mühendisliğinde genel olarak Yapay Zeka bilim dalının uygulamaları literatürde oldukça az yer almaktadır. Yapılan uygulamaların büyük bir çoğunluğunu Uzman Sistemler oluşturmaktadır. Yapay Zekanın bir alt dalı olan Genetik Algoritmalar konusunda yeterli sayılabilecek derecede uygulama mevcut değildir. Oysa su kaynakları mühendisliğinde diğer mühendislik dallarının çoğunda olduğu gibi göz önüne alınan sistemin matematik modeli kurularak sistemin performansı belirlenmekte ve sistem analizi yöntemlerini kullanarak optimizasyon yapılmaktadır. Birçok hallerde birden fazla amacı birlikte göz önüne almak gerektiğinden optimum çözüm belirlenememektedir, ancak her bir seçeneğin çeşitli amaçlar açısından değeri belirlenebilmektedir.

Bu çalışmada, su kaynakları sistemlerinde dinamik programlama problemlerinin çözümünde Genetik Algoritma yöntemi kullanılmıştır.

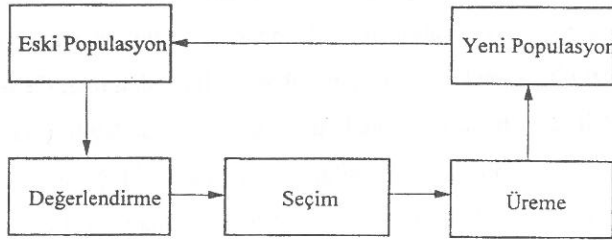
II. GENETİK ALGORİTMA

Ünlü matematikçi J. Hadamard “Gerek matematikte, gerek başka alanlarda buluş ve icatlar farklı alanlardan düşüncelerin bir araya getirilmesiyle gerçekleşir” demiştir [1]. Bu söz Genetik Algoritmalar’ın nasıl doğduğunu özetlediği gibi temel çalışma ilkesini de açıklamıştır. Genetik Algoritmalar’ın temelinde mümkün olan birçok varyans olmasına rağmen topluluk üzerinde icra edilen belli başlı işlemler relatif olarak standartlaştırılmıştır. Bu işlemler üç aşamadan oluşmaktadır.

- Bireylerin uygunluklarının değerlendirilmesi
- Gen havuzunun oluşturulması
- Rekombinasyon ve mutasyon

Bir sonraki nesillerin oluşturacağı topluluk, bu üç operasyondan sonra oluşmaktadır. İşlem başarılı bireyler bulununcaya kadar tekrarlanmaktadır. Şekil 1’de Genetik Algoritma’nın temel yapısı gösterilmektedir. Şekil 1’de görüldüğü gibi eski topluluk değerlendirme işlemine tabi tutulduktan sonra, başarılı bireyler seçilmektedir. Başarısız bireylerin yerini alan başarılı bireyler kendi aralarında rasgele çiftleştirilerek yeni topluluk oluşturulmaktadır. Bir sonraki adımda yeni topluluğun ismi, eski

topluluk olarak değiştirildikten sonra, işlemler daha önceden verilmiş bir generasyon sayısına kadar devam etmektedir.



Şekil 1. Genetik Algoritma'nın genel yapısı .

Aşağıda Genetik Algoritma'nın yapısı genel hatları ile verilmektedir. Bu yapı altında Genetik Algoritma'nın toplumsal bakış açısı ile verilmiş işlem düzenini göstermektedir.

1. Başlangıç anını belirle
 $t:=0;$
2. Toplukta yer alacak bireyleri oluştur.
 $P(t);$
3. Bireylerin yaşam koşullarına uygunluğunu değerlendir.
Değerlendir $P(t);$
4. İstenilen düzeye ulaşılanaya kadar
While not başarılı-birey-bulundu **do**
5. Zaman sayacını artır.
 $t:= t+1;$
6. Bir sonraki topluluğa döl verecek bireyleri seç
 $P'(t) :=Ebeveynleri seç P(t);$
7. Seçilen bireyleri çiftleştir.
Çiftleştir $P(t);$
8. Geçici toplulukta rasgele değişimler olmasını sağla
Değişime- uğrat $P(t);$
9. Geçici topluluğun başarı durumunu değerlendir.
Değerlendir $P(t);$
10. Bir sonraki topluluğu oluştur.
 $P(t+1):=P(t);$
End do ;

Buradaki terimlerde topluluk havuz ile, birey kromozom ile, çiftleşme çaprazlama ile, uygunluk fonksiyonu tasarımın kalitesi ile, sosyal başarı da tasarımın optimalitesi ile yer değiştirdiğinde ortaya Genetik Algoritmalar çıkmaktadır. Görüldüğü gibi başlangıçta topluluğu oluşturma işlemi yapılmaktadır. Bu adımda genellikle uygulanan

yöntem başlangıç topluluğunun rasgele oluşturulması şeklindedir. Fakat topluluk oluşturulurken bireylerin mümkün olduğu kadar birbirine benzememesini sağlamak toplumsal çeşitliliği artırmak açısından faydalı olmakta ve dolayısıyla çözüme daha hızlı yaklaşılmaktadır.

Toplulukları evrim sürecine sokmadan önce yapılması gereken bir başka işlem de, başlangıç bireylerinin değerlendirilmesidir. Bu aşama evrim süreci içerisinde bir sonraki nesle döl verecek olan bireylerin belirlenmesi için gerekmektedir. Daha sonra istenilen düzeyde başarılı birey bulununcaya kadar veya topluluk başarıda artış sağlayamaz duruma gelince veya önceden belirlenen generasyon sayısı tamamlanuncaya kadar sürecek olan evrim başlayacaktır. Her evrim sürecinde tekrar edilecek olan işlemler ise aşağıdaki şekilde sıralanacaktır.

Bir sonraki nesle döl verecek olan bireyler daha önce hesaplanmış olan başarı değerlerine bağlı olarak seçilmektedir. Daha sonra, seçilen bireyler istenilen bir yöntemle çiftleştirilmektedir. Çiftleştirme sonucu oluşturulan bireyler genellikle, % 0.1 olasılıkla değişime uğratılmaktadır. Son adım olarak ta oluşturulan yeni bireylerin başarı değeri hesaplanmaktadır.

III. GENETİK ALGORİTMA İŞLEMLERİ

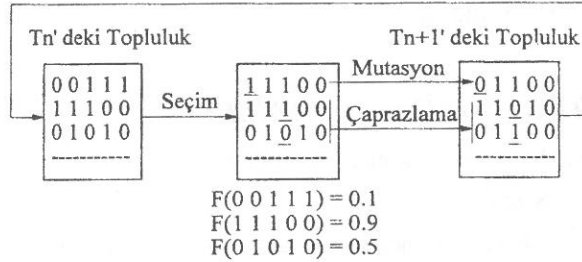
Genetik Algoritma ile yapılan işlemler genellikle seçim, çaprazlama ve mutasyon operatörleri yardımıyla yapılmaktadır.

Seçim işlemi, bir topluluk içerisinde uygunluğu yüksek bireylerin bulunmasını sağlamaktadır. Seçim işlemi topluluk içerisinde uygunlukları düşük olan bireyleri eleyip, bu bireylerin yerine uygunluğu yüksek olan bireylerin birkaç adet kopyasını yaparak tamamlanmaktadır.

Çaprazlama, biyolojik terim olarak bir bireyin içerisindeki değişimi kastetmesine rağmen, Genetik Algoritma'da bireyler arasındaki benzer alt kromozomların değişimini ifade etmektedir. Genetik Algoritma'da çaprazlama iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. En çok kullanılan çaprazlama türleri bir noktalı, iki noktalı ve çok noktalı çaprazlama türleridir.

Sınırlı bir topluluk üzerinde çalışıldığında, toplulukta birkaç genetik bilginin erkenden kaybolma ihtimali bulunmaktadır. Örnek olarak, bir kromozomu oluşturan genlerin tamamı bir ya da sıfır olabilmektedir. Böyle bir kromozomu çaprazlama operatörü ile değiştirmek mümkün olamamaktadır. Çaprazlama vasıtasıyla üretilmeyen uygunluk değeri yüksek kromozomları mutasyon operatörüyle üretmek mümkün olmaktadır.

Şekil 2.'de seçim, çaprazlama ve mutasyon işlemlerinin nasıl yapıldığı gösterilmektedir. Başlangıçta topluluk rasgele oluşturulmaktadır. Burada 00111'e kromozom, 0 ve 1 rakamlarına ise gen ismi verilmektedir. Kromozom beş adet genden oluştuğundan uzunluğu beştir. Buradaki 00111, 11100 ve 01010 kromozomları herhangi bir işlemde değerlendirildikten sonra sırasıyla 0.1, 0.9 ve 0.5 değerlerini almaktadır. Görüldüğü gibi uygunluğu en düşük olan birey 00111, en yükseği ise 11100'dır. En düşük uygunluklu birey ile en yüksek uygunluklu birey seçim operatörü yardımıyla bulunmaktadır. Şekil 2.'de ikinci kutuda uygunluğu en yüksek olan 11100 bireyi, uygunluğu en düşük olan 00111 bireyi gen havuzu dışına atılarak bunun yerine kopyalanmaktadır. Gen havuzu olarak adlandırılan ikinci kutuda 11100 bireyinde altı çizili olan 1 geni yerine 0 geni yazılarak kromozom mutasyona uğratılmakta ve 01100 bireyi elde edilmektedir. İkinci 1100 bireyi ve üçüncü 01010 bireyi arasında üç nolu sitede tek noktalı çaprazlama yapılarak 11010 ve 01100 bireyleri elde edilmektedir. Yapılan bu işlemler daha önceden verilmiş bir generasyon sayısına kadar devam etmektedir [2].



Şekil 2. Genetik Algoritma işlemleri.

Diğer bir çaprazlama türü inversiyondur. İversiyon operatörü tek bir birey üzerinde işlem yapmaktadır. Bu nedenle tek bir birey üretilmektedir. İversiyon siteleri random olarak seçilmektedir. Örneğin aşağıdaki bireyde random olarak seçilen 1 ve 9 nolu genler arasında inversiyon yapılarak 2. birey elde edilmektedir.

1. Birey	2. Birey
<u>1</u> 23456789	<u>1</u> 23456789
10110111 <u>0</u>	00110111 <u>1</u>

IV. DİNAMİK PROGRAMLAMA PROBLEMLERİNİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ

Su kaynaklarını geliştirme çalışmalarında amaç fonksiyonunu en büyük yapacak çözümün (optimal proje büyüklüğü, proje bileşenlerinin optimal boyutları, optimal işletme kuralları) çeşitli kısıtlara (teknolojik, ekonomik,...) uyarak belirlenmesi istenir.

Nispeten bazı basit problemlerin optimum çözümleri sistem analizinin lineer programlama, lineer olmayan programlama ve dinamik programlama gibi algoritmaları ile kesin olarak belirlenebilir. Ancak bu yöntemlerin su kaynakları sistemlerine uygularken yapılması gereken kabuller çoğu zaman sistemin gerçek niteliğinden ayrılmasına yol açtığından varılan çözümler de gerçekçi olmamaktadır. Hesap süresini gerçekçi ölçüler içinde tutabilmek için basitleştirici kabuller yapılması gerekir.

Matematiksel optimizasyon yöntemlerinden biri olan dinamik programlamada amaç fonksiyonunun ve kısıtların doğrusal yapıda olması koşulu yoktur, bunların matematik ifadeleri herhangi bir yapıda olabilir, hatta amaç fonksiyonu karar değişkenlerinin sadece belli değerleri için tablolanmış süreksiz bir fonksiyon şeklinde de verilebilir. Bir optimizasyon problemine dinamik programlamanın uygulanabilmesi için tek koşul şudur: Amaç fonksiyonu ve kısıtlar, her biri tek bir karar değişkenine bağlı olan terimlerin toplamı şeklinde olmalıdır.

$$\text{Max(Min)} V = \sum_{i=1}^n v_i(x_i)$$

(1)

$$\text{Kısıtlar : } \sum_{i=1}^n g_i(x_i) \leq 0$$

Bu ifadelerde v_i ve g_i herhangi bir yapıda fonksiyonlar olabilir.

Bir Genetik Algoritma uygulaması, rasgele oluşturulmuş bireylerden teşkil edilen bir toplum ile başlar. Başlangıçtaki toplumun her bir bireyine ait kromozom, temsil ettiği veri yapısına uygun şekilde çözülerek problemin bağımsız değişkenleri elde edilir. Elde edilen bu değişkenler uygunluk fonksiyonunun parametreleri olarak alınır ve uygunluk değerleri hesaplanır. Bunun genetik bilimindeki karşılığı, canlının ortamın şartlarına

uyum sağlamanın ölçüsüdür. Genetik Algoritmalarındaki anlamı ise, kromozom ile temsil edilen değişkenlerin arzu edilen çözüme yakınlığıdır [3].

Genetik Algoritmaların klasik optimizasyon yöntemlerine göre bazı önemli avantajları vardır. Bunlar aşağıdaki gibi sıralanabilir.

1. Genetik Algoritma açısından problemin lineerlik, non-lineerlik ve süreklilik gibi özelliklerinin hiç bir anlamı yoktur.
2. Optimize edilecek amaç fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonunun türevi de olamayacağından, türev almaya dayalı optimizasyon yöntemleri uygulanamamaktadır. Oysa Genetik Algoritma optimizasyonunda, işlemlere fonksiyonların türevi girmediğinden, böyle bir sakınca söz konusu değildir.
3. Genetik Algoritma ile optimizasyon işlemlerinde, karmaşık matematiksel ifadeler yer almamaktadır.
4. Değişkenlerinin alabilecekleri değerler, sonlu bir tamsayı listesi veya yeterli hassasiyette sonlu ve ayrık bir değer uzayından alınabilir.
5. Genetik Algoritmalar ile optimizasyon algoritmasının en önemli avantajlarından biri; problem ile optimizasyon algoritması arasında iyi bir geçişin sağlanmasıdır. Bu özellik, algoritmanın esnekliğini önemli ölçüde artırmaktadır.
6. Genetik Algoritma ile optimizasyon için, her probleme özel bir optimizasyon tekniği kullanılması söz konusu değildir. Optimizasyonu gerçekleştirmek için boyutlandırma değişkenlerini, değişkenlerin değer uzayını ve problemin uygunluk fonksiyonunu belirlemek yeterlidir.
7. Alışılmış optimizasyon tekniklerine göre yapılan çözümlerde boyutlandırma değişkenlerinin ilk değerlerine bağlı olarak, algoritma her zaman sonuç vermeyebilir. Genetik Algoritma ile optimizasyonda ise, böyle bir sorunla karşılaşılsa dahi, bir sonraki adımda çözüm için uygun olmayan değerler grubu çözümden çıkartılmaktadır.

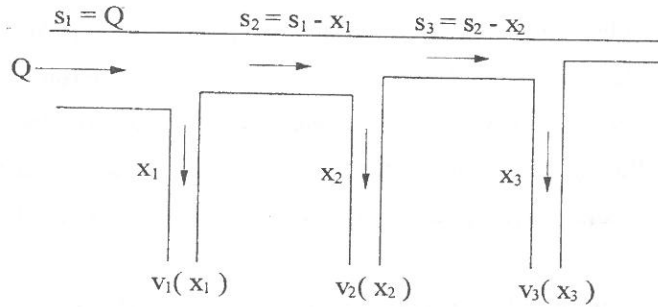
IV. 1. Sayısal Örnekler

Bu çalışmada, su kaynakları sistemlerindeki dinamik programlama problemleri Genetik Algoritma yöntemi kullanılarak çözülmüştür. Çalışma daha sonra yapılacak bir dizi araştırma için bir ilk düzey çalışmasıdır. Bu nedenle kolay anlaşılabilmesi için basit örnekler seçilmiştir. Algoritma IBM uyumlu bilgisayarlara yönelik olarak BASIC programlama dilinde kodlanmıştır.

Bu tip problemler, karar deęişkenlerine alabilecekleri bütün deęerleri verip V 'yi en büyük yapan kombinezonu belirlemek yolu ile çözülmek istenirse, karar deęişkeni sayısının (n) ve bu deęişkenlerin alabilecekleri deęerlerin sayısının (m) artmasıyla seçenek sayısı (m^n) hızla büyüyeceğinden problemin bu şekilde çözümlenmesi zorlaşır ve giderek imkansız hale gelir [3]. Problemin Genetik Algoritma ile çözümünde ise n ve m sayıları ne kadar büyük olursa olsun problem kolay bir şekilde çözülebilmektedir.

IV. 1. 1. Beş Birim Suyun Üç Kullanıcı Arasında En İyi Şekilde Paylaşımı

$Q = 5$ birim suyun üç kullanıcı arasında en iyi şekilde nasıl paylaşılacağı bulunacaktır. Fayda fonksiyonları Çizelge 1'de verilmiştir [3].



Şekil 3. Suyun üç kullanıcı arasında en iyi şekilde paylaşımının belirlenmesi .

Çizelge 1. Fayda fonksiyonları

x_i	$v_1(x_1)$	$v_2(x_2)$	$v_3(x_3)$
1	1.0	6.5	0.5
2	3.0	10.1	2.0
3	6.6	10.9	6.3
4	10.0	9.6	11.5
5	13.1	7.0	15.6

Genetik Algoritmalar ile problemin çözümünde izlenecek en önemli aşamalardan birisi uygun bir gen haritası oluşturmaktır. Yanlış oluşturulmuş bir gen haritasıyla problem doğru bir şekilde çözülemeyecektir. Bu problemde gen haritası iki tabanındaki sayıların yan yana dizilmesiyle oluşturulmaktadır. Bu gen haritasına aynı zamanda kromozom ismi de verilmektedir.

Suyun 3 kullanıcı arasında en iyi şekilde paylaşılmasında amaç fonksiyonu;

$$\text{MAKSİMİZE } | \text{FAYDA } (Q_1, Q_2, Q_3) |$$

ile ifade edilmektedir. Burada fayda Q_1, Q_2, Q_3 debilerinin bir fonksiyonudur. Genetik Algoritmada Q_1, Q_2, Q_3 değerleri ayrı ayrı alt kromozomlar ile ifade edilmekte ve bunların bütünü esas kromozomu vermektedir.

MAKSİMİZE | FAYDA (000, 100, 001) |

Burada ikilik tabandaki sayı on tabanına çevrilirse,

$$Q_1 = 0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 \rightarrow Q_1 = 0$$

$$Q_2 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 \rightarrow Q_2 = 1$$

$$Q_3 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 \rightarrow Q_3 = 4$$

değerleri elde edilmektedir. Burada $Q_1 = 0$ 'da 1. boruya su verilmemektedir. $Q_1 = 0$ 'da fayda 0, $Q_2 = 1$ 'de 6.5 ve $Q_3 = 4$ 'de 11.5 değerini almaktadır. Toplam fayda $Q_1 + Q_2 + Q_3 = 18$ değerini vermektedir. Genetik Algoritmanın yaptığı hesaplar Çizelge 2'de verilmektedir. Bu problemde generasyon sayısı 50, çaprazlama site sayısı 5, kromozom uzunluğu (000, 100, 001) 9, kopyalama sayısı 2 ve populasyon büyüklüğü 14 seçilmiştir.

Çizelge 2'de 1. kutuda B.N. ile birey numarası verilmiştir. Başlangıçta 2. kutucukta topluluk rasgele üretilmektedir. İkinci aşamada, random olarak üretilen kromozoma karşılık gelen Q_1, Q_2 ve Q_3 değerleri tespit edilmektedir. Örneğin 6. kromozomda (100, 100, 110)

$$Q_1 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 \rightarrow Q_1 = 1$$

$$Q_2 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 \rightarrow Q_2 = 1$$

$$Q_3 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 \rightarrow Q_3 = 3$$

olarak bulunmaktadır. $Q_1 + Q_2 + Q_3$ toplamının toplam debiye eşit olmaması durumunda fayda değeri 0 alınmaktadır. Amaç fonksiyonu faydayı maksimum yapmak olduğu için, toplam debiye eşit olmayan birey, kromozom havuzunun dışına atılmaktadır. 7. kutucukta kopyalama işlemi yapılmaktadır. Örneğin 2. kutucukta başarılı olan 1,2 kromozomları, başarısız olan 13 ve 14 kromozomları üzerine kopyalanmaktadır. 8. kutucukta bireyler eşleştirilmektedir. Örneğin 1.(000100001) ve 9.(110100100)

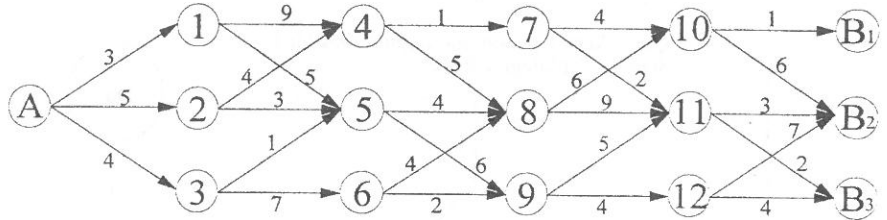
bireyleri eşleştirilmektedir. 9. kutucukta bu bireyler 3, 5, 4, 6 ve 7 noktalarında çaprazlanmaktadır.

0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 1 0 0

Çaprazlama sonucunda 1. bireyde 000100101 ve 9. bireyde 110100000 kromozomları oluşmaktadır. Bu işlem 50. generasyona kadar devam etmektedir. Çizelge 2 'de görüleceği gibi her generasyonda bireyin başarısı (fayda değeri) artmaktadır. Genetik Algoritmanın yüksek performansı sayesinde 1. generasyonda dahi maksimum fayda değeri elde edilmektedir. 50. generasyonda bireylerin başarı değeri maksimum olduğu için ve artık bireylerde birbirine benzediği için çaprazlama işlemi benzer bireyler arasında yapılmaktadır. Bu nedenle 50. generasyonda topluluktaki bireylerin tamamı başarılı bireyi temsil etmektedir. Genetik Algoritmanın yaptığı işlemleri gösteren akış diyagramı şekil 4'de verilmektedir.

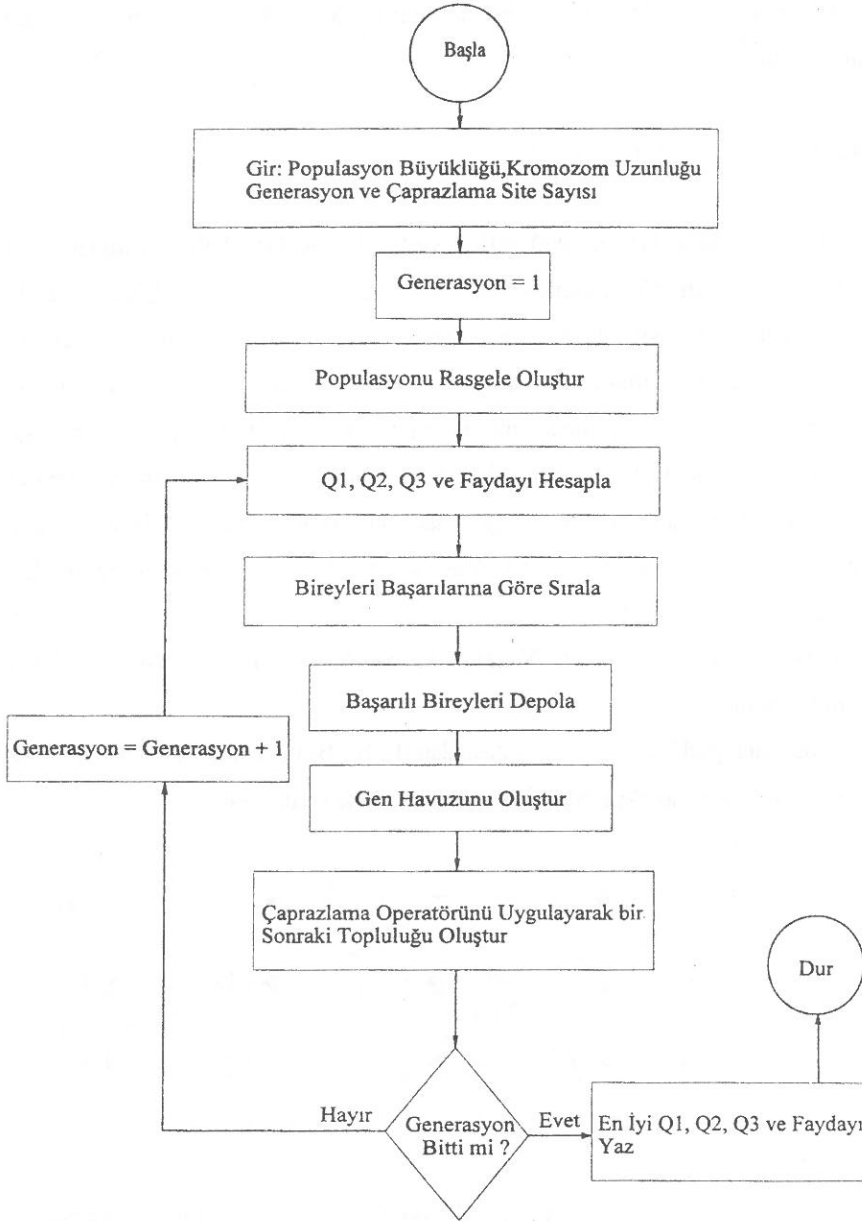
IV.1.2. A Noktasından B₁, B₂, B₃ Noktalarına Su İletecek En Az Maliyetli Boru Hattının Belirlenmesi

Şekil 5 'de gösterilen geçki üzerinde A noktasından B₁, B₂, B₃ noktalarına su iletecek en az maliyetli boru hattı belirlenecektir. Maliyetler şekil üzerinde verilmiştir [3].



Şekil 5. A noktasından B₁, B₂, B₃ noktalarına su iletecek geçki .

Bu problemde en önemli aşama daha önceki problemde olduğu gibi gen haritasını (kromozomu) uygun bir şekilde oluşturmaktır. Amaç fonksiyonu ise maliyeti minimize edecek geçki güzergahını tayin etmektir. Kromozom uzunluğu 15 olup 5 adet 3 genli alt kromozomlardan oluşmaktadır. Şekil 6'da kesik çizgilerle gösterilen yollar illegal turları göstermektedir.

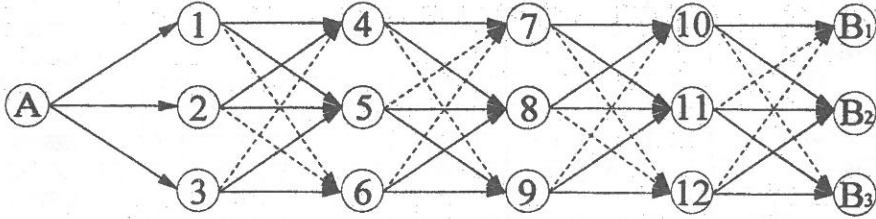


Şekil 4. Dinamik programlama problemlerinin genetik algoritma ile çözümü için akış diyagramı .

Çizelge 2. 1.,15. ve 50. Generasyonlara ait hesaplamalar

1. GENERASYON													
1	2	3	4	5	6	7	8	9				10	
B.N.	Eski Populasyon	Q ₁	Q ₂	Q ₃	Fayda	Kopyalama	Eğilime	Çaprazlama				Yeni Populasyon	
1.	000100001	0.00	1.00	4.00	18.00	000100001	9	3	5	4	6	7	000100101
2.	110010000	3.00	2.00	0.00	16.70	110010000	11	7	9	8	4	5	110110010
3.	001100000	4.00	1.00	0.00	16.50	001100000	12	7	3	4	2	8	000000000
4.	001100000	4.00	1.00	0.00	16.50	001100000	14	4	6	3	9	2	000100001
5.	000010110	0.00	2.00	3.00	16.40	000010110	8	5	1	7	8	9	000010100
6.	100100110	1.00	1.00	3.00	13.80	100100110	10	7	9	2	8	4	100000010
7.	110100100	3.00	1.00	1.00	13.60	110100100	13	5	3	8	6	2	110110100
8.	010010100	2.00	2.00	1.00	13.60	010010100	5	5	1	7	8	9	010010110
9.	110100100	3.00	1.00	1.00	13.60	110100100	1	3	5	4	6	7	110100000
10.	100010010	1.00	2.00	2.00	13.10	100010010	6	7	9	2	8	4	100110110
11.	000110010	0.00	3.00	2.00	12.90	000110010	2	7	9	8	4	5	000010000
12.	100000001	1.00	0.00	4.00	12.50	100000001	3	7	3	4	2	8	101100001
13.	100001000	1.00	4.00	0.00	10.60	110010000	7	5	3	8	6	2	110000000
14.	000101000	0.00	5.00	0.00	7.00	000100001	4	4	6	3	9	2	001100000
15. GENERASYON													
1.	000100001	0.00	1.00	4.00	18.00	000100001	13	8	2	4	1	6	000100001
2.	000100001	0.00	1.00	4.00	18.00	000100001	8	8	1	4	7	2	100100101
3.	000100001	0.00	1.00	4.00	18.00	000100001	14	7	5	1	8	4	000100001
4.	000100001	0.00	1.00	4.00	18.00	000100001	10	5	8	7	6	1	100100001
5.	001100001	4.00	1.00	4.00	0.00	001100001	9	7	4	8	3	6	001100001
6.	000100100	0.00	1.00	1.00	0.00	000100100	11	6	7	5	2	8	010100000
7.	000110001	0.00	3.00	4.00	0.00	000110001	12	7	6	1	2	3	000110001
8.	100100100	1.00	1.00	1.00	0.00	100100100	2	8	1	4	7	2	000100000
9.	011100001	6.00	1.00	4.00	0.00	011100001	5	7	4	8	3	6	011100001
10.	100100001	1.00	1.00	4.00	0.00	100100001	4	5	8	7	6	1	000100001
11.	011000000	6.00	0.00	0.00	0.00	011000000	6	6	7	5	2	8	001000100
12.	000100000	0.00	1.00	0.00	0.00	000100000	7	7	6	1	2	3	000100000
13.	000000000	0.00	0.00	0.00	0.00	000100001	1	8	2	4	1	6	000100001
14.	000000001	0.00	0.00	4.00	0.00	000100001	3	7	5	1	8	4	000100001
50. GENERASYON													
1.	000100001	0.00	1.00	4.00	18.00	000100001	8	6	7	3	8	1	000100001
2.	000100001	0.00	1.00	4.00	18.00	000100001	9	4	6	8	2	3	000100001
3.	000100001	0.00	1.00	4.00	18.00	000100001	10	2	1	8	5	4	000100001
4.	000100001	0.00	1.00	4.00	18.00	000100001	14	7	9	5	8	2	000100001
5.	000100001	0.00	1.00	4.00	18.00	000100001	11	5	4	3	9	8	000100001
6.	000100001	0.00	1.00	4.00	18.00	000100001	12	6	9	7	1	5	000100001
7.	000100001	0.00	1.00	4.00	18.00	000100001	13	5	1	9	6	8	000100001
8.	000100001	0.00	1.00	4.00	18.00	000100001	1	6	7	3	8	1	000100001
9.	000100001	0.00	1.00	4.00	18.00	000100001	2	4	6	8	2	3	000100001
10.	000100001	0.00	1.00	4.00	18.00	000100001	3	2	1	8	5	4	000100001
11.	000100001	0.00	1.00	4.00	18.00	000100001	5	5	4	3	9	8	000100001
12.	000100001	0.00	1.00	4.00	18.00	000100001	6	6	9	7	1	5	000100001
13.	000100001	0.00	1.00	4.00	18.00	000100001	7	5	1	9	6	8	000100001
14.	000100001	0.00	1.00	4.00	18.00	000100001	4	7	9	5	8	2	000100001

Buralarda maliyeti çok yüksek tutularak güzergahın bu kısımlardan geçmesi engellenmiştir. Sistem için seçilen kromozom aşağıda verilmektedir.



Şekil 6. İlegal turlar.

																B ₁	B ₂	B ₃
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
0	1	0	1	0	0	0	1	0	0	0	1	1	0	0				

Burada 1 rakamı güzergahın geçtiği noktayı göstermektedir. Örneğin bu kromozomda güzergah $A \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow B_1$ yolunu takip etmektedir. Bu problemde bireyleri değişime uğrutup yeni bireyler oluşturmak için inversiyon operatörü kullanılmıştır. Örneğin;

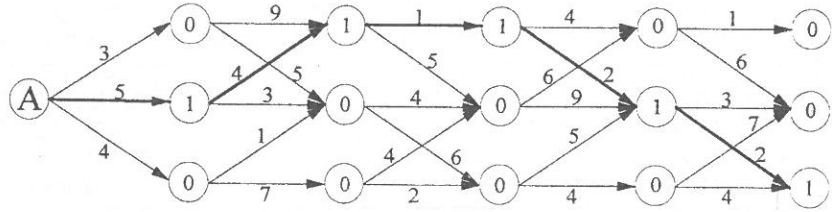
																			B ₁	B ₂	B ₃
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15							
0	1	0	1	0	0	0	1	0	0	0	1	1	0	0							

bireyinde 1-2, 4-6, 8-9, 10-12 ve 13-14 noktalarında inversiyon yapılarak aşağıdaki birey oluşturulmuştur.

1 0 0 0 0 1 0 0 1 1 0 0 0 1 0 (Yeni birey)

Şekil 7'de en az maliyetin olduğu yol ve gen haritası görülmektedir. Burada 1 rakamı güzergahın o noktada geçeceğini, 0 rakamı ise geçmeyeceğini göstermektedir. Genetik

Algoritma, daha karmaşık tipteki boru hatları problemlerinin çözümünde de etkili olmaktadır.



Şekil 7. En az maliyetin oluştuğu yol ve gen haritası.

Bu problemde generasyon sayısı 20, kromozom uzunluğu 15, popülasyon büyüklüğü 14 ve kopyalama sayısı 6 seçilmiştir. Çizelge 3 'de Genetik Algoritmanın yaptığı hesaplar verilmektedir.

V. SONUÇ VE TARTIŞMA

Genetik Algoritma ile optimizasyon, canlılardaki biyolojik ve sosyal işleyişin örneklenmesi ile elde edilmektedir. Hayali bir biyolojik topluluk ve bu topluluğun geçirdiği evrim sürecini incelemek Genetik Algoritmanın işleyişini somutlaştırmak açısından iyi bir yaklaşımdır.

Bu çalışmada, dinamik programlama problemlerinin çözümünde Genetik Algoritma tekniği kullanılmıştır. Bu maksatla, Genetik Algoritma esaslı bir optimizasyon programı geliştirilmiştir. Dinamik programlama problemlerinde karar değişkeni sayısının (n) ve bu değişkenlerin alabilecekleri değerlerin sayısının (m) artmasıyla seçenek sayısı (m^n) hızla büyüyeceğinden problemin bu şekilde çözülmesi zorlaşır ve giderek imkansız hale gelir. Bununla beraber, karar değişkeni sayısı ve bu değişkenlerin alabilecekleri değerlerin sayısı ne kadar büyük olursa olsun bu tip problemler Genetik Algoritma ile hızlı ve hassas olarak çözülebilir.

Çizelge 3. 1., 5. ve 20. Generasyonlara ait hesaplamalar

1. GENERASYON				
1	2	3	4	5
B.N.	Eski Populasyon	Maliyet	Kopyalama	Yeni Populasyon
1.	010100100010010	15.00	010100100010010	001010010010010
2.	001010001010010	19.00	001010001010010	010010001010100
3.	100100100100010	23.00	100100100100010	100010100010001
4.	010010010010010	24.00	010010010010010	001001001010001
5.	001001010010001	26.00	001001010010001	100100100100010
6.	100100010010001	28.00	100100010010001	010001001010100
7.	100010010001001	1000016.00	100010010001001	100010010010001
8.	100010010001001	1000016.00	100010010001001	010010100100100
9.	100010010001001	1000016.00	100100010010001	001100001100010
10.	010010010100001	1000018.00	001001010010001	001100100010001
11.	010001010100010	1000021.00	010010010010010	100001100001001
12.	010001010100001	2000015.00	100100100100010	010100100100001
13.	001100010001010	2000016.00	001010001010010	001010010100001
14.	001100010010100	2000018.00	010100100010010	001010100001001
5. GENERASYON				
1.	010100100010001	14.00	010100100010001	100010001010001
2.	100100100010001	17.00	100100100010001	001100100100001
3.	010010001001001	22.00	010010001001001	010100100100100
4.	100010010010010	24.00	100010010010010	100010100100010
5.	100100010010001	28.00	100100010010001	001001100010010
6.	100001001001001	1000013.00	100001001001001	100001001001001
7.	010001001001001	1000015.00	010001001001001	001010010100010
8.	001010010100001	1000015.00	001010010100001	100010100010010
9.	100010010001001	1000016.00	100001001001001	010100010001001
10.	010100001010001	1000016.00	100100010010001	100100010010100
11.	010010100100010	1000018.00	100010010010010	100100100100001
12.	100010100010100	2000010.00	010010001001001	010100001100001
13.	100100010001100	2000017.00	100100100010001	100100001100001
14.	100100001100010	2000018.00	010100100010001	001100010010100
20. GENERASYON				
1.	010100100010001	14.00	010100100010001	010100100010001
2.	010100100010001	14.00	010100100010001	010100100010001
3.	010100100010001	14.00	010100100010001	010100100010001
4.	010100100010001	14.00	010100100010001	010100100010001
5.	010100100010001	14.00	010100100010001	010100100010001
6.	010100100010001	14.00	010100100010001	010100100010001
7.	010100100010001	14.00	010100100010001	010100100010001
8.	010100100010001	14.00	010100100010001	010100100010001
9.	010100100010001	14.00	010100100010001	010100100010001
10.	010100100010001	14.00	010100100010001	010100100010001
11.	010100100010001	14.00	010100100010001	010100100010001
12.	010100100010001	14.00	010100100010001	010100100010001
13.	010100100010001	14.00	010100100010001	010100100010001
14.	010100100010001	14.00	010100100010001	010100100010001

Çalışma daha sonra yapılacak bir dizi araştırma için bir ilk düzey çalışmasıdır ve bu amaçla başarısı önem kazanmaktadır. Klasik çözüm yöntemlerinin yetersiz kaldığı daha karmaşık dinamik programlama problemlerinin çözümünde kullanılmak üzere çalışmalar devam etmektedir. Sunulan yöntemin başarısı bu çözümlerle daha açıkça görülebilecektir.

KAYNAKLAR

- [1] D.E. Goldberg, "Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning", Ph.D. Dissertation University of Michigan, Ann Arbor, 1983.
- [2] S. Forrest, "Genetic Algorithms", *Principles of Natural Selection Applied to Computation Science*, V.261, pp: 872-878, 1993.
- [3] M. Bayazit, "*Su Kaynakları Sistemleri*", İTÜ İnşaat Fakültesi, Sayı: 1549, İstanbul, 1994.