ÇOKLU-DİZİ HİZALAMA PROBLEMİ İÇİN GENETİK ALGORİTMA

Seda Arslan **Taner Tuncer** **Ali Karcı**

Bilgisayar Mühendisliği Bölümü
Mühendislik Fakültesi
Fırat Üniversitesi 23119 Elazığ
e-posta: akarci@firat.edu.tr

Anahtar sözcükler:Çoklu Dizi Hizalama, Genetik Algoritma,

ÖZET

Çoklu-Dizi Hizalama moleküler biyolojide, jeolojide ve bilgisayar biliminde önemli bir problemdir. Protein yapılarını analiz etmek, yeni proteinler tasarlamak ve DNA sıralılarından evrim ağaçları inşa etmede kullanılır. Genelde Çoklu-Dizi Hizalama zor optimizasyon problemler sınıfına aittir ve kombinasyonal problemler olarak adlandırılır. Bu tip problemlerin çözümü için son yıllarda geliştirilen metotlardan biri Genetik Algoritmadır. Genetik Algoritmalar zor optimizasyon problemlerinin çözümü için uygun bir metottur. Bu makale Çoklu-Dizi Hizalama probleminin çözümünü elde etmede Genetik Algoritmaların nasıl kullanılacağını gösterir.

1.GİRİŞ

Çoklu-Dizi Hizalama (ÇDH) moleküler biyolojide önemli bir problemdir [1,2]. ÇDH Protein yapılarını analiz etmek ve yeni proteinler tasarlamak ve DNA zincirlerinden (Adenin-Timin, Guanin-Stozin) evrim ağaçları inşa etmede kullanılır [3,4,5]. ÇDH son yıllarda birçok farklı bilim alanında da kullanılmaktadır. Çoklu Dizi Hizalama dinamik programlama yaklaşım tabanlıdır. Bu yaklaşımlar üstel zaman karmaşıklığına sahiptir. Bu nedenle analitik metotlarla etkili çözümler elde edilmesi zor ve zaman alıcıdır.

Genetik Algoritma (GA) doğal seçim ve genetik kurallara dayandırılmış global bir arama ve analitik olmayan optimizasyon işlemidir [6, 7]. GA' da rasgele çözümden bir populasyon oluşturulur ve bu populasyondan seçim, çaprazlama, mutasyon gibi doğada geçerli kanunlar kullanılarak çözüm aranır [8, 9,10, 11, 12, 13, 14, 15, 16]. ÇDH gibi eksponansiyel zaman karmaşıklığına sahip problemlerin çözümünde analitik yöntemlere göre daha hızlı ve doğru çözüm veren GA kullanılır. ÇDH problemini çözmek için dinamik programlama, çiftlerin toplamı (SP) gibi farklı yöntemler kullanılmıştır. Fakat bu çalışmada ÇDH probleminin GA ile çözümü üzerinde durulacaktır; çünkü GA ile çözüm yapmak hem basit ve hem de karmaşıklığı eksponansiyel olmayacaktır ve polinomsal olacaktır.

Bu çalışma aşağıdaki gibi organize edilmiştir. 2. bölümde ÇDH probleminin tanımı verilerek 3. bölümde ÇDH problemini çözmek için kullanılan GA' dan bahsedilerek onun parametreleri ve işleyişi açıklanmıştır. 4. bölümde ÇDH probleminin GA ile nasıl çözüleceğinden bahsedilmiş ve problemin kodlanması gerçekleştirilmiştir. 5. bölümde ise örnek bir uygulama gerçekleştirilmiş ve GA ile elde edilen sonuçlar açıklanmıştır. Son olarak ÇDH problemine GA' nın uygulanmasından elde edilen sonuçlar hakkında yorum yapıldıktan sonra elde edilen sonuçların genel bir değerlendirmesi yapılmıştır.

2.ÇOKLU-DİZİ HİZALAMA

Çoklu-Dizi Hizalama problemine moleküler biyolojide, jeolojide ve bilgisayar biliminde sıkça rastlanır. Moleküler biyolojiye en önemli katkısı evrimsel analizin keşfidir. Bu keşif bize farklı organizmalar için yeni protein yapıları ve DNA sıralılarını elde etmeye yardımcı olmuştur. ÇDH eksponansiyel zaman karmaşıklığına sahip bir optimizasyon problemidir ve kombinasyonal problemler olarak adlandırılır. Zaman karmaşıklığı denklem 2.1 deki gibi değişkendir. Burada L, dizilerin uzunluğu N ise dizilerin sayısıdır.

$$F(L,N)=L^N \qquad (2.1)$$

ÇDH iki veya daha fazla sıralı dizilerin optimal bir şekilde hizalanmasını ve yeni sıralı dizilerin elde edilmesini amaç edinmiştir. Hizalama işlemi, dizilerin belirli bir değerde kaydırılması ve kaydırma işlemi yapıldığında boşlukların dikkate alınmadan üst üste çakışan benzer sembollerin veya çiftlerin toplamının maksimum olması işlemidir.

3.GENETİK ALGORİTMA

GA doğal seçim ve genetik kurallara dayandırılmış global bir arama ve optimizasyon işlemidir. Doğada geçerli olan, en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için en iyinin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için çaprazlama ve mutasyon gibi operatörler kullanılır.

Genetik algoritmaların en uygun olduğu problemler, geleneksel yöntemler ile çözümü mümkün olamayan ya da çözüm süresi problemin büyüklüğü ile üstel orantılı olarak artan problemlerdir. GA' nın sıkça kullanılmasının bazı önemli nedenleri vardır

- Matematikte türev ve entegrale ihtiyaç duymaz.
- Problemin ne olduğunu bilmesine gerek duymaz.
- Problem hakkında fazla bir bilgiye sahip olmayıp bir kör arama metodur.
- Problemin kendisi yerine parametrelerin kodları ile uğraşır.
- Nümerik yöntemlerden farklı olarak birden fazla aday çözüm ile çözüme başlar.
- İşlem süreci stokastiktir ve deterministik değildir.

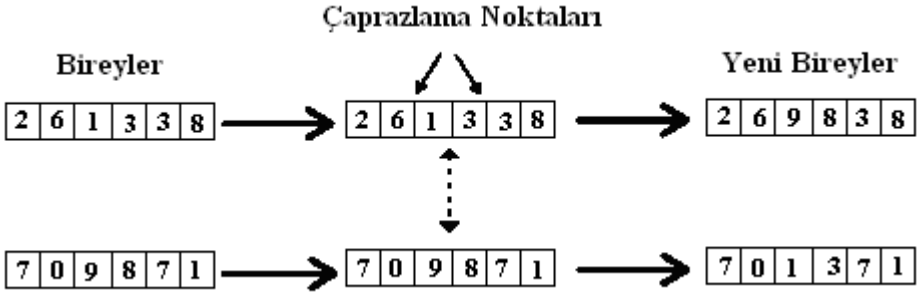
Genellikle GA aşağıdaki gibi ifade edilir.

```
Prosedür(Genetik Algoritma)
M=populasyon boyutu
Ng=Evrim sayısı
No=Döllerin sayısı
Pm=Mutasyon olasılığı
P‹E(M)
For j=1 to M
        Uygunluk(P[j])
End
For i=1 to Ng
        For j=1 to No
                (x,y)‹Φ(P)
                dφl[j]<X(x,y)
                uygunluk(dφl[j])
        End
        For j=1 to No
                Mutasyon[j]‹m(y)

        Uygunluk(Mutasyon[j])
        End
P‹Seç(P,Döl)
End
```

GA' nın çalışması için ilk adım başlangıçta kullanılacak olan populasyonun üretilmesidir. Burada uygulanan yöntem genellikle populasyonun rasgele oluşturulmasıdır. Eğer çözümün ne olabileceği tahmin edilebiliyorsa, mümkün olduğunca başlangıç populasyonu bilinen parametre değerleri ile oluşturularak çözüm hızlandırılmalıdır. Her evrim sürecinde tekrar edilecek işlemler ise şöyle sıralanacaktır. Bir sonraki nesile döl verecek olan bireyler hesaplanmış başarı değerlerine göre Rulet tekerleği yada turnuva seçimi yardımıyla seçilir. Bu seçim türleri bir çok araştırmacı tarafından kullanılması tercih edilen seçim yöntemleridir; aksi halde literatürde tanımlanmış çok sayıda seçim yöntemi vardır. Seçilen bireylerle populasyondaki çeşitliliği sağlayabilmek için, bireyler istenilen bir yöntemle tek ya da çok noktalı çaprazlama işlemine tabi tutulur. Şekil 1. Kısmi Harita Çaprazlama işlemini göstermektedir. Bu çaprazlama şekli her kromozomda genlerin farklı olması zorunlu olduğu durumlarda kullanılması tercih edilir.

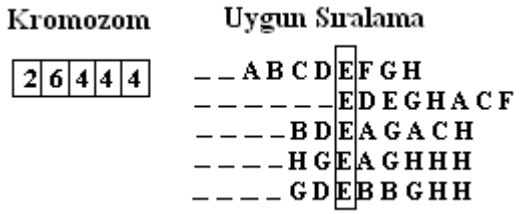
Çaprazlama sonucu elde edilen bireyler mutasyon işlemine tabi tutulur. Mutasyon bireylerdeki gen ya da genlerin olasılığının %5' ten büyük olmaması şartıyla değiştirilmesidir. Mutasyon olasılığının düşük seçilmesinin önemli bir nedeni vardır. Eğer mutasyon olasılığı büyük seçilirse, kromozomlarda çok sık değişmeler meydana gelecektir ve bunun sonucunda genetik süreç uzayacaktır. Bundan dolayı o anda populasyonda var olmayan bilginin populasyona eklenmesi düşük bir olasılıkla yapılmış olur. Uygunluk değerleri hesaplanarak en iyi bireylerle yeni populasyon oluşturulur. Evrim süreci ya belirlenen evrim sayısına göre ya da bireylerde istenilen başarı elde edilene kadar devam ettirilir.



Şekil 1. Kısmi Harita Çaprazlama

4. PROBLEMİN KODLANMASI

Her bir olası sıraya dizme N genden oluşan bir kromozom gibi gösterilebilir. Burada N sıralanmış dizinin sayısıdır. Kromozomdaki her bir genin içeriği, o gene ait dizinin ne kadar kaydırılarak dizilerle hizalanacağını gösterir. Şekil 2' de örnek bir kromozom için dizilerin kaydırma işlemini göstermektedir. Kromozomdaki 1. gen 1. dizinin 2 karakter sağa, 2. gen 2. dizinin 6 karakter sağa v.s kaydırılacağını gösterir.



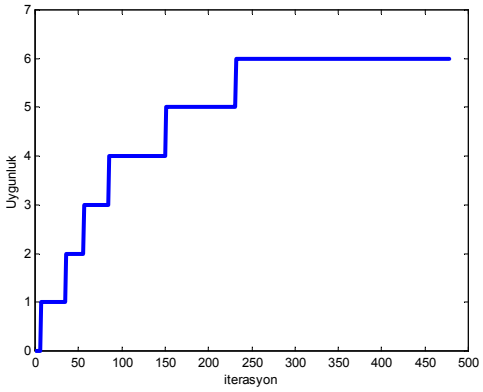
Şekil 2. Problemin kodlanması ve kodun çözümlenmesi.

Uygunluk fonksiyonu, uyumlu karakterlerin toplam sayısı, aşağıdaki gibi bir fonksiyonla ifade edilebilir. Tüm sıralılarda her bir sütun değerinin aynı olası durumunda uygunluk fonksiyonuna 1 sayısı ekleme yapılır. Örneğin yukarıdaki kromozom için uygunluk değeri birdir.

```
Uygunluk( ){
i maksimum değere ulaşmadığı sürece aşağıdaki
işlemleri tekrarla
  Eğer dizi1[i]=dizi2[i]{
    Eğer dizi2[i]=dizi3[i]{
       .
       .
       .
       Uygunluk=Uygunluk+1
       i=i+1
    }
    değilse i=i+1
  .
  .
  .
  }
  değilse i=i+1
  }
  değilse i=i+1
}
```

5. DENEYSEL SONUÇLAR

{A,B,C,D} kümesi göz önüne alınarak 6 farklı dizi oluşturulmuş ve bu dizilerin GA kullanılarak en iyi şekilde hizalanması sağlanmaya çalışılmıştır. Şekil 3' teki diziler için olası hizalama sayısı 191102976 dır. Populasyon boyutu, diğer bir deyişle başlangıçta rastgele üretilen aday çözüm sayısı 100 olarak seçilmiştir. Performansının iyi olmasından dolayı PMX çaprazlama kullanılmıştır ve %5 mutasyon oranı kullanılmıştır. Yapılan deneylerin sonucunda en uygun birey 478 iterasyon sonucunda bulunmuştur. En uygun çözüm bulunurken problemin sonlandırılması belirli bir iterasyon sonucu uygunluk fonksiyonunun değişmemesi durumunda gerçekleştirilmiştir. Şekil 3' te uygunluk fonksiyonunun iterasyon sayısı ile değişimi verilmiştir. Şekil 4 için uygunluk değeri 6 olur. Verilen diziler için bulunabilecek maksimum uygunluk sayısı 6 olur.

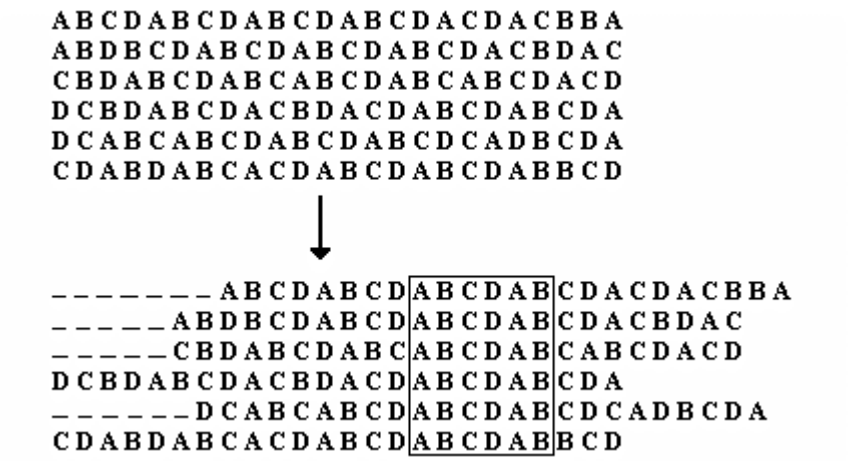


Şekil 3. İterasyon sayısına göre uygunluk değişimi

6. SONUÇ

Yapılan uygulama sonucunda GA' nın ÇDH problemi için uygun bir yöntem olduğu görüldü. ÇDH problemi NP sınıfına ait bir problem olup, çözümü için deterministik bir algoritma tasarımı günümüze kadar yapılabilmiş değildir. Bundan dolayı bu problemin çözümü yaklaşım yöntemleri ile yapılmaktadır. GA yöntem olarak olasılık kuralları ile yaklaşım yapan bir yöntemdir. Bir yaklaşım yöntemi olmasına rağmen bulmuş olduğu sonuç çok iyi ve bu sonucu bulurken harcamış olduğu zamanda önemli derecede kısalmıştır.

Geleneksel algoritmalar, eğer dizi boyutları küçük ve dizi içinde değerlerde birbirine benzerlik çok fazla ise, bulacakları sonuç veya sonuçlar iyi olabilmektedir. Fakat GA' nın böyle bir sınırlaması yoktur.



Şekil 4. Uygulamada kullanılan diziler.

KAYNAKLAR

[1]T. Akutsu, "Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots", Discrete Applied Mathematics, vol: 104, 2000, pp:45-62.

[2]C.E. Ferreira, C.C. de Souza; Y. Wakabayashi, "Rearrangement of DNA fragments: a branch-and-cut algorithm", Discrete Applied Mathematics, vol: 116, 2002, pp:161-177.

[3] C. Notredame, D.G. Higgins, "SAGA: sequence alignment by genetic algorithm", Nucleic Acids Research, vol:24, 1996, pp:1515-1524.

[4] J.-T. Horng, L.-C. Wu, C.-M. Lin, "A genetic algorithm for multiple sequence alignment", Soft Computing, 2004.

[5] R. König, T. Dandekar, "Refined genetic algorithm simulations to model proteins", Journal of Molecular Modeling, vol:5, 1999, pp:317-324.

[6] D. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning", Massachusetts, Addison-Wesley Publishing Company Inc., 1989.

[7] S. Mahfoud, D. Goldberg, "A Genetic Algorithm for Parallel Simulated Annealing", Parallel Problem Solving from Nature 2, Elsevier Science Publishers B.V., 1992.

[8] A. Karcı, A. Arslan, "Bidirectional evolutinary heuristic for the minimum vertex-cover problem" Journal of Computers and Electrical Engineerings, vol. 29, pp.111-120, 2003.

[9] A. Karcı, A. Arslan, " Uniform Population in Genetic algorithms", İ.Ü. Journal of Electrical & Electronics, vol.2 (2), pp.495-504, 2002.

[10] Ali Karcı, " Novelty in the Generation of Initial Population for Genetic Algorithms" Knowledge-Based Intelligent Information & Engineering Systems Wellington, New Zealand. Lecture Notes in Artificial Intelligence, 2004.

[11] A. Karcı, A. Çınar, B. Ergen, "Genetik Algoritma Kullanılarak Minimum Düğüm Kapsama Probleminin Çözümü", Elektrik-Elektronik-Bilgisayar Mühendisliği 8. Ulusal Kongresi, 6-12 Eylül, Gaziantep, Türkiye, 20-23, 1999.

[12] A. Karcı, A. Çınar, "Comparison of Uniform Distributed Initial Population Method and Random Initial Population Method in Genetic Search", 15th International Symposium on Computer and Information Sciences", October, 11-13, Istanbul, Turkey, 2000, pp.159-166.

[13] A. Karcı, A. Arslan, "Genetik Algoritmalarda Düzenli Populasyon", GAP IV: Mühendislik konferansı, 06-08 Haziran 2002, Harran Üniversitesi, Şanlıurfa.

[14] A. Karcı, "Genetik Algoritmalarda Iraksama ve Yerel Çözümde Kalma Problemlerinin Giderilmesi", F. Ü. Fen Bilimleri Enst., 2002.

[15] K. K. Gündoğan, B. Alataş, A. Karcı, "Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator", Turkish Journal of Electrical Engineering and Computer Sciences (ELEKTRİK), Volume 12, Number 1, 43-52, 2004.

[16] A. Karcı, "Imitation of Bee Reproduction as a Crossover Operator in Genetic Algorithms", 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zeland, 2004.