



Erciyes University Journal of the Institute of Science and Technology

Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi

ISSN 1012-2354

Cilt (Volume): 29, Sayı (Issue): 1, Ocak/January-2013

<http://fbe.erciyes.edu.tr/>



Polinomal Olmayan Problemler için DNA Hesaplama Algoritması

Uğur ÇİĞDEM¹, Mehmet KARAKÖSE²

¹Bilgisayar Programcılığı Bölümü, Gaziosmanpaşa Üniversitesi, Tokat

²Bilgisayar Mühendisliği Bölümü, Fırat Üniversitesi, Elazığ

ÖZET

DNA moleküllerinin özelliklerinden yararlanılarak geliştirilen sayısal DNA hesaplama algoritmasının son yıllarda günümüz problemlerinin çözümündeki kullanımı artmaktadır. Paralel işlem yapma ve büyük miktarda veri saklama özelliği bulunan DNA hesaplamanın laboratuvar ortamında uygulanması zor ve pahalıdır. Bu çalışmada sayısal DNA hesaplama algoritması polinomal olmayan (NP) problemlere uygulanmıştır. NP problemler kesin çözümü olmayıp yaklaşık çözümü bulunabilen zor problem sınıfında yer alırlar. Bu çalışmadaki yapılan simülasyonlarda NP problemlerin verilen sayısal DNA hesaplama algoritması ile daha kolay ve hızlı çözülebileceği gösterilmiştir. Bunun için gezgin satıcı ve sırt çantası problemi seçilerek geliştirilen DNA hesaplama algoritmasının performansı matlab'da alınan simülasyon sonuçları ile doğrulanmıştır.

Anahtar

Kelimeler:

DNA Hesaplama,
Gezgin Satıcı
Problemi,
Sırt Çantası

DNA Computing Algorithm for NP problems

ABSTRACT

The numerical algorithm for DNA computing has been developed on the basis of the characteristics of the DNA molecules. Its use for the resolution of the current problems is getting common. It is capable of parallel processing and storing huge amount of data but it is difficult and costly to be implemented within the laboratory environment. In this study, Numerical DNA computing algorithm is implemented on non-polynomial problems (NP). NPs are known to be difficult problems with only a close solution rather than a definite one. The simulations conducted as part of this study have shown that NP problems can be solved more easily and quickly by the given numerical algorithm for DNA computing. The algorithm is developed upon the selection of travelling salesman and knapsack problems and its performance has been verified by the results of matlab simulations.

Key Words:

DNA computing,
Travelling
Salesman
Problem,
Knapsack.

1. Giriş

DNA hesaplama canlıların genetik bilgisini saklayan DNA moleküllerinin özelliklerini kullanarak hesaplama yapan yeni bir araştırma alanıdır. Bu alanda yapılan çalışmalar çözelti ortamı ve elektronik bilgisayarlarda olmak üzere iki farklı şekilde uygulanmaktadır. Çözelti ortamında yapılan uygulamalarda, problemler DNA molekülleri ile kodlanmakta ve biyolojik tepkimeler kullanılarak çözüm elde edilmektedir. Sayısal ortamda uygulanan DNA hesaplama algoritması ise genetik algoritmanın benzer özellikleri kullanılarak geliştirilmektedir. DNA çözeltisi kullanılarak yapılan ilk çalışma 1994 yılında Leonard Adleman tarafından gerçekleştirilen gezgin satıcı probleminin DNA dizileri ile kodlanması ve çözümünün bulunmasıdır [1]. Bu çalışma DNA hesaplama ile yapılan ilk uygulamadır. Uygulamada 7 şehirden oluşan bir gezgin satıcı problemi tasarlanmıştır. Şehirleri ve şehirlerarasındaki mesafeleri DNA dizileri ile kodlayıp deney tüplerinde gerçekleştirdiği biyokimyasal tepkimeler sonucu elde ettiği dizileri çıktı olarak kullanmıştır. Lipton, Adleman tarafından gerçekleştirilen deneye benzer bir yöntemle 3-SAT problemlerinin çözümü için DNA hesaplamayı kullanmıştır [2].

3-SAT problemleri, içerisinde birçok mantıksal denklem içerir. Bu nedenle çözümleri normal yollarla çok zor olmaktadır. Bu çalışmada sentez, arama, polimerleşme ve ayırma biyolojik adımlarını izleyerek laboratuvar ortamında milyonlarda biyokimyasal tepkime gerçekleştirmiştir. DNA dizileri ile muhtemel bütün çözüm kümesini kodlayıp çoğaltarak aynı özellikli test tüplerine yerleştirip paralel işlem gerçekleştirmiştir.

Kimyasal tepkimeler sırasında oluşan hatalar ve DNA bazları arasındaki hidrojen bağlarının farklı erime sıcaklığında ayrılması çalışmanın istenilen şekilde verimli sonuçlanmasını engellemiştir. Ouyang ve arkadaşları, DNA moleküllerini kullanarak maksimum klikleme problemlerinin çözümü için yeni bir yöntem geliştirmiştir [3]. Yapılan çalışmada, 7 düğümlü ve her düğümün kenarlarla birbirine bağlandığı grafik problemini örnek alınmıştır. Uygulama ile maksimum düğümden oluşan küme elde edilmeye çalışılmıştır. Her bir düğüm 6 bitlik ikili sayı sistemi ile kodlanmıştır. Seçilen örnekte düğüm sayısı az olduğu için çok fazla sorun çıkmamakla beraber düğüm sayısı artırıldığında problemin zorlaştığı görülmüştür. Head ve arkadaşları maksimum bağımsız küme problemlerinin optimizasyonu için DNA moleküllerini kullanan bir çalışma gerçekleştirmişlerdir [4].

Yapılan çalışmada laboratuvar ortamında DNA hesaplama ile grafiği oluşturan düğüm kümesi içerisinden maksimum bağımsız alt küme sayısını bulmayı amaçlamışlardır. Düğümleri DNA dizi segmentleri ile kodlamış enzimler kullanarak çözüm kümesini bulmaya çalışmışlardır. Bu tür çalışmalarda düğüm sayısı arttığında enzim sayısındaki sınırlamadan dolayı çözüm kümesinin bulunması zor olmaktadır. Yang ve arkadaşları mantıksal denklemlere sahip problemlerin çözümü için DNA moleküllerinin hesaplama yeteneğinden yararlanmıştır [5]. Bu çalışmada Lipton'un çalışmasından daha fazla veri DNA ile kodlanabilmiştir. Wood ve arkadaşları Max-1 problemlerin çözümü için DNA hesaplamasının kullanılabilirliğini gösteren bir çalışma yapmıştır [6].

Ding ve Ren bulanık mantık ile DNA hesaplamayı birlikte kullanan bir çalışma gerçekleştirmiştir [7]. Shin ve arkadaşları DNA molekülleri arasındaki hidrojen bağları sayısını kullanarak yeni bir ağırlık kodlama metodu geliştirmişlerdir [8]. Yapılan çalışmada DNA dizileri gerçek değerler ile gösterilerek moleküler programlama gerçekleştirilmiştir. Kodlama işlemi sabit kod uzunluğu ve kenarların ağırlıkları dikkate alınarak yapılmıştır. Lee ve arkadaşları DNA moleküllerinin erime sıcaklık değerlerini kullanarak Shin'den farklı bir ağırlık kodlama metodu ileri sürmüşlerdir [9].

Liu ve arkadaşları NP problemler, minimum aralıklı ağaç problemi ve grafik renk problemlerinin çözümü için DNA moleküllerini kullanmıştır [10]. Zhixiang ve arkadaşları DNA molekülleri ile maksimum bağımsız küme problemlerinin çözümü üzerine bir çalışma yapmıştır [11]. Yukarıda verilen uygulamalar sonucu DNA hesaplama hızla gelişerek birçok bilimsel çalışmada kullanılmıştır [12-16]. Özellikle NP problemlerde, denetleyici parametrelerinin ayarlanmasında grafiklerin renklendirilmesi problemlerinde, aritmetik işlemlerde, sinyal işleme problemlerinde ve verilerin şifrelenmesinde sıklıkla kullanılmıştır [17-21].

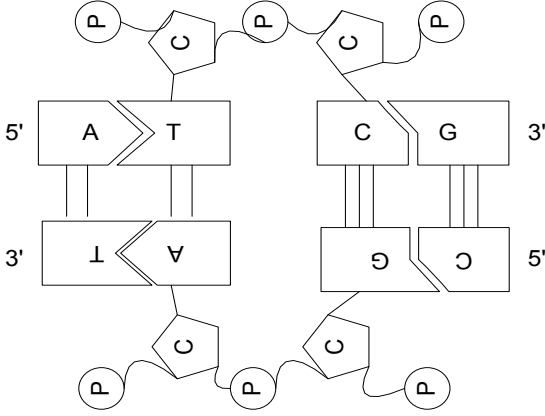
Bu çalışmada DNA hesaplamasının sayısal algoritması kullanılarak NP problemlerden olan gezgin satıcı ve sırt çantası problemlerinin optimizasyonu gerçekleştirilmektedir. Literatürde bu problemlerin DNA hesaplama ile çözümü mevcut olmasına karşın, yapılan bu çalışmalar çözelti ortamında gerçekleştirilmiştir [22]. Ancak çözelti ortamı uygulaması zor ve maliyetli olduğu için burada sayısal DNA hesaplama algoritması önerilmektedir. Önerilen bu algoritmada DNA moleküllerinin genetik algoritma ile olan benzerliklerinden yararlanılmakla birlikte sadece bu moleküle ait olan enzim ve virüs mutasyonunun sayısal ortama uyarlanmış şekli problemlere uygulanmaktadır.

Bu çalışma aşağıda verilen bölümlerden oluşmaktadır: Bölüm 2'de DNA hesaplama hakkında geniş bilgi verilmektedir; bölüm 2'nin devamında gezgin satıcı ve sırt çantası problemleri ve DNA ile kodlaması anlatılmaktadır; bölüm 3'te simülasyon sonuçları ayrıntılı bir şekilde verilmektedir; bölüm 4'te ise bu çalışma ile elde edilen sonuçlar verilmektedir.

2. DNA Hesaplama

DNA hesaplama moleküler biyolojiyi kullanarak optimizasyon problemlerinin çözümünü gerçekleştiren yeni bir araştırma alanıdır. DNA hesaplama kullanılırken DNA molekülünü oluşturan bazlardan yararlanır. Bu bazlar Adenin, Guanin, Sitozin ve Timin olup sırayla A, G, C ve T harfleri ile gösterilir [10-16]. DNA molekülü yapısındaki hidrojen bağlarından dolayı kararlı bir yapıya sahiptir. Adenin ile Timin arasında iki, Guanin ile Sitozin arasında üç hidrojen bağı vardır. Guanin ve Sitozin üçlü hidrojen bağı nedeniyle Adenin ve Timin'e göre daha kararlıdır.

DNA moleküllerinin yapısında bunların yanında 5 karbonlu şeker ve fosfat grubu bulunmaktadır. DNA molekülündeki karbonlar molekülün yönünü belirlemektedir. DNA moleküllerinin pozitif yönü 5'-3' ile, negatif yönü ise 3'-5' ile gösterilmektedir.

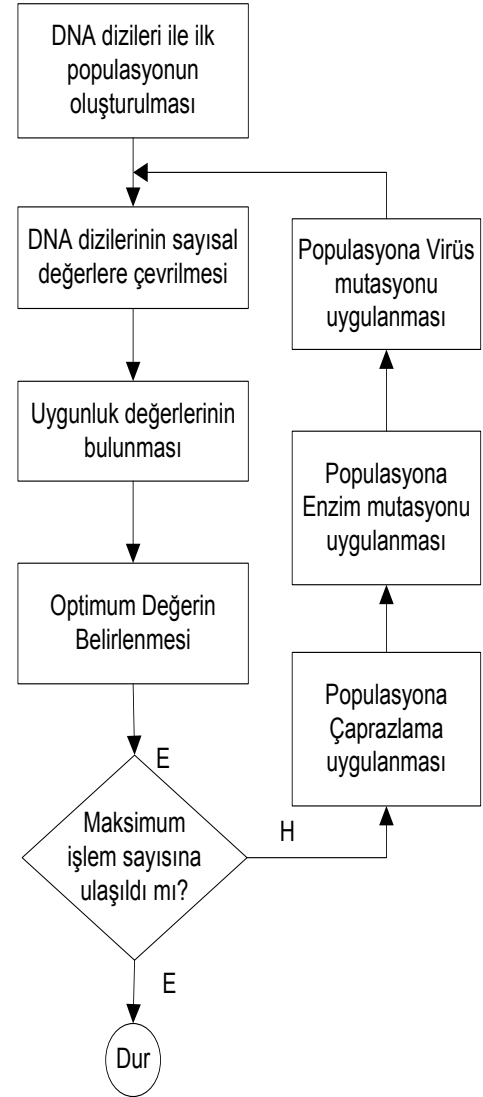


Şekil 1. Çift sarmal DNA molekülünün genel yapısı

Şekil 1’de DNA molekülünün genel yapısı verilmiştir. DNA molekülleri tekli dizi ve çift sarmal dizi şeklinde bulunmaktadır. Tekli DNA dizileri kullanarak DNA moleküllerinin sentezi ve çoğaltılması gerçekleştirilir. Çift sarmal DNA dizileri Watson–Crick tütleme kuralına göre oluşur [15-18]. Bu kurala göre Adenin ile Timin ve Guanin ile Sitozin birleşir. Bunun dışında bir oluşum mümkün değildir. Buraya kadar DNA molekülünün genel yapısı ayrıntılı olarak anlatılmaktadır. Şimdi ise DNA molekülünün hesaplamada kullanıldığında ne tür avantaj ve dezavantajları olduğu anlatılacaktır. Paralel işlem yapma yeteneği sayesinde çözümü yapılacak optimizasyon problemi alt gruplara ayrılarak işlemin daha hızlı ve daha kısa sürede gerçekleşmesi sağlanır. Veri saklama kapasitesinin çok yüksek olması DNA moleküllerinin elektronik veri saklama aygıtlarından 10^{12} kat daha fazla veri saklanabilmesine olanak sağlar [24,25]. Ayrıca DNA molekülünün yıllarca bozulmadan kalması sayesinde insanların genetik bilgisi uzun süre korunmaktadır. Uzun süreli korunması gereken verilerin saklanmasında DNA moleküllerinden yararlanılmaktadır. Bütün bunların yanında moleküllerin diziliminde oluşacak bir hatanın hesaplama sonucunu doğrudan etkileme riski vardır. Bu şekilde bir hata yapıldığında sonuçlar çok farklı oluşmaktadır. DNA moleküllerinin hesaplama özelliklerinden azami fayda sağlanabilmesi için uygulamaların çözelti ortamında gerçekleştirilmesi ise daha fazla maliyet gerektirmekte ve uygulanabilirliği azaltmaktadır. Ayrıca bütün optimizasyon algoritmalarında olduğu gibi DNA hesaplamada da bulunan sonucun doğruluk oranı % 100 değildir.

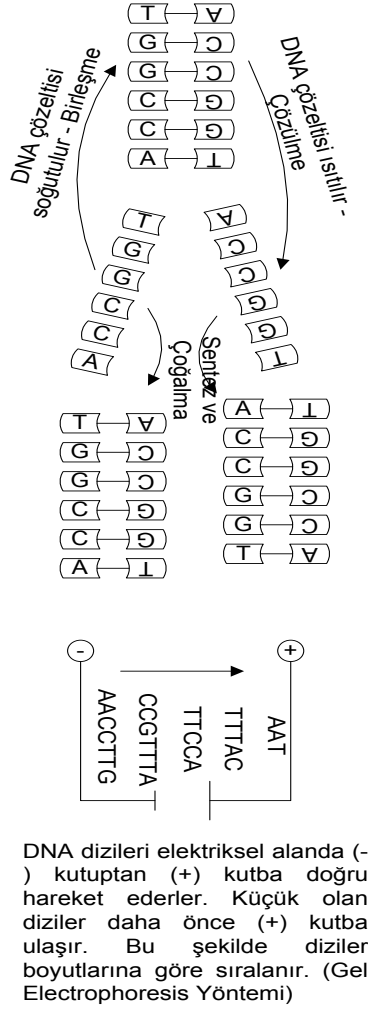
DNA hesaplamasının bugüne kadarki uygulamaları genellikle çözelti ortamında gerçekleştirilmiştir [24,25]. Yapılan çalışmalarda biyokimyasal tepkimeler kullanılmıştır. Bu tepkimelerde kullanılan algoritmik adımlar hibritleşme (hybridization), onarma (ligation), polimer zincir reaksiyonu (polymerase chain reaction–PCR), DNA dizilerinin sıralanması (gel electrophoresis), DNA dizilerinin ayrılması (melting) ve DNA dizilerinin birleştirilmesi (annealing) olarak sıralanır. Şekil 2’de DNA hesaplamasının çözelti ortamındaki adımları verilmektedir [24,25]. Çözelti ortamında problem çözümü zor olduğu için son zamanlarda yapılan literatür çalışmalarında sayısal ortamda DNA hesaplama kullanımına yönelik çalışmalar gerçekleştirilmektedir. Bu çalışmalardan bir

kısmı grafik problemleri şeklinde diğer bir kısmı ise genetik algoritmanın DNA hesaplama ile olan benzerliklerinden yararlanılarak uygulanmaktadır [21]. DNA hesaplama genetik algoritmadan farklı olarak iki tür mutasyon uygulaması kullanılmaktadır. Bunlar enzim ve virüs mutasyonudur [21]. Enzim mutasyonu kullanılarak populasyondan belirli orandaki elemanların silinmesi sağlanır. Virüs mutasyonu kullanılarak populasyona yeni elemanların eklenmesi sağlanır. Bu iki mutasyon kullanılarak algoritmanın her bir iterasyonu işletildiğinde popülasyonun yenilenmesi sağlanır. Böylece DNA hesaplamasının yerel aramalardan kurtulup genel arama yeteneği gelişmiş olur. Bu çalışmada önerilen model ile DNA molekülleri sayısal verilere dönüştürülmekte ve problemlerin kodlaması bu şekilde yapılmaktadır. Ayrıca DNA moleküllerinin çözelti ortamında gerçekleştirdikleri enzim ve virüs mutasyonunun sayısal ortamda da yapılabileceği gösterilmektedir. Önerilen model ile DNA hesaplamasının elektronik ortamda kullanımı gerçekleştirilmektedir. Şekil 3’te çözelti ortamı algoritması verilmektedir.



Şekil 2. Sayısal DNA hesaplama algoritması adımları

Şekil 2’de NP problemlerin optimizasyonu için kullanılan sayısal DNA hesaplama algoritması verilmektedir.



Şekil 3. DNA Hesaplama çözelti ortamı adımları

2.1. Gezgin Satıcı Problemi

Gezgin satıcı problemi en kısa yolu bulma işlemidir. Gezgin Satıcı Probleminde gezgin satıcı olarak isimlendirdiğimiz bir kişi problemde verilen şehirlerden her birine *en az ve sadece bir kez* uğramak zorundadır [5,22]. Ayrıca bizim istediğimiz bir başlangıç noktasından başlayıp yine bizim istediğimiz bir bitim noktasına ulaşmak zorundadır. Bu problemde amacımız bütün bu şartların gerçekleştiği minimum mesafeli yolu bulmaktır [19,22].

Gezgin satıcı probleminde t tane şehir olsun. Bu durumda şehirlerin kümesi $S = \{s_1, s_2, s_3, \dots, s_t\}$ şeklinde olur. Şehirlerarasındaki mesafeleri $m(s_i, s_j)$ ile gösterelim. Bu durumda problemin çözümü için Denklem 1'de verilen uygunluk fonksiyonu kullanılmaktadır [26].

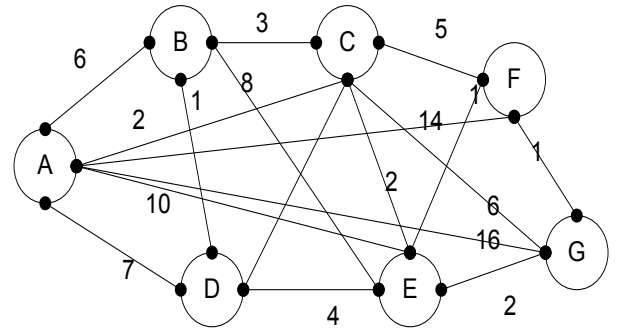
$$\min \left(\sum_{i=1}^{t-1} m(s_i, s_{i+1}) + m(s_t, s_1) \right) \quad (1)$$

Geliştirilen sayısal ortam algoritmasında şehir ve mesafelerin kodlaması için DNA dizileri kullanılmaktadır. Literatürde yapılan bazı çalışmalarda [22] sabit uzunluklu DNA dizileri kullanılmasına karşın bu çalışmada değişken sayısına bağlı olarak dizi uzunluğu seçilmektedir. Çünkü sabit uzunluklu diziler kullanıldığında değişken sayısı çok olan problemlerin kodlanması zor olmaktadır. Polinomal olmayan problem sınıfından olan gezgin satıcı probleminin kodlanması değişken sayısı ile orantılı olarak sürekli zorlaşmaktadır. Bazı literatür çalışmalarında bu tip problemin kodlanması için DNA

moleküllerine ait erime sıcaklığı değerleri (T_m) kullanılmıştır [9].

Bu alandaki çalışmalar incelendiğinde açıkça görüleceği üzere şehirlerin kodlanması ve şehirlerarasındaki mesafe değerlerine göre erime sıcaklıklarının ayarlanması ayrıca bir zorluk oluşturmaktadır. Bizim önerdiğimiz kodlama yöntemi erime sıcaklığından bağımsız olarak sadece şehir sayısı dikkate alınarak DNA uzunluğu belirlenmekte, enzim ve virüs mutasyonu gibi iki önemli özellik sayısal ortama uyarlanarak hesaplama yapılmaktadır. Yaptığımız çalışmada şehir sayısı 7 olarak seçilmiş olup her bir şehir ve mesafe 3'lü DNA dizisi ile gösterilmekte A-0, G-1, C-2 ve T-3 değerleri kullanılarak sayısal veriler oluşturulmaktadır. Örneğin A şehri ile B şehri arasındaki mesafe 6 birim olduğundan AGC ile gösterilmektedir. Çünkü AGC dizisi dörtlük sistemde 012 ile ifade edilir. Bu değer onluk sisteme çevrildiğinde 6 değeri elde edilmektedir. A şehri 1. şehir olduğu için AAG ile kodlanmaktadır. Örnek problemde şehir sayısı 7 ve şehirlerarasındaki en uzun mesafe 16 olduğu için 3 DNA bazı ile kolayca kodlanmaktadır. Şekil 1'de verilen örnek problem için şehirlerin ve yolların DNA ile kodlanması Tablo 2 ve Tablo 3'te verilmektedir. Tablo 4'te şehirlerarasındaki mesafelerin sayısal değerlere dönüştürülmüş şekli gösterilmektedir. Yapılan çalışma için geliştirilen algoritma ve program bölümü değişken sayılarının durumuna göre kolayca uyarlanabilir haldedir. Bu çalışmada Şekil 4'te verilen gezgin satıcı problemi için DNA hesaplama algoritması kullanılarak optimum çözüm bulunmaktadır. Gezgin satıcı probleminin DNA hesaplama ile optimizasyonunda kullanılan algoritmik adımlar sırayla aşağıda açıklanmaktadır.

Adım 1. Şehir sayısı 7 olarak belirlendi. Şehirler arasındaki yollar ve uzunlukları DNA molekülleri kullanılarak kodlanıp ilk popülasyon oluşturuldu. Kodlama işlemi Tablo 2 ve Tablo 3'te ayrıntılı olarak verilmektedir. Şekil 4'te örnek bir gezgin satıcı problemi verilmektedir.



Şekil 4. Örnek gezgin satıcı problemi

Adım 2. Yollar için oluşturulan popülasyonun her bir elemanı için uygunluk değerleri hesaplanır. Uygunluk değerinin hesaplanması için Denklem 1 kullanılmaktadır.

Adım 3. Problem için bir başlama ve bitim noktası belirlenip oluşturulan yollardan bu koşula uymayanlar çözüm kümesinden çıkarılır. Örnek problemde A şehri başlangıç G şehri ise bitim noktası olarak alınmaktadır.

Adım 4. Problem için belirlenen şehirlerin tamamına uğramayan yollar çözümden çıkarılır.

Adım 5. Minimum mesafe ve en kısa yol belirlenir.

Adım 6. Yollar için oluşturulan popülasyona çaprazlama uygulayarak yeni popülasyon oluşturulup 2, 3, 4 ve 5 adımları uygulanarak minimum mesafe ve en kısa yol belirlenir.

Adım 7. Yollar için oluşturulan popülasyona enzim ve virüs mutasyonu uygulanarak yeni popülasyon oluşturulup 2, 3, 4

ve 5 adımları uygulanarak minimum mesafe ve en kısa yol belirlenir.

Adım 8. Maksimum işlem sayısına ulaşıldığında bulunan minimum mesafe ve en kısa yol problemin çözümü olarak alınmaktadır.

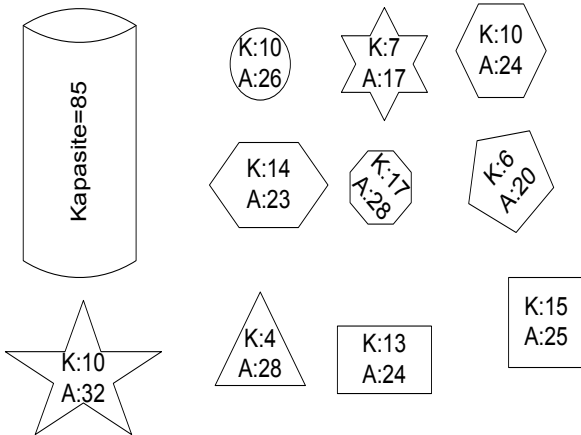
2.2. Sırt Çantası Problemi

Sırt çantası problemi şu şekilde tanımlanır. Elimizde ağırlıkları ve kazançları belirlenmiş olan elemanlar bulunmaktadır. Bu elemanlar kapasitesi sınırlı olan ve bilinen bir kutu veya benzeri bir nesne içerisine yerleştirilecektir [19]. Bu problemin çözümünde amaç *maksimum kazanç sağlanacak ve ağırlıklar toplamı kapasiteyi aşmayacak şekilde en fazla elemanı* yerleştirmektir [5,23]. Bu tür problemler 0/1 planlama problemi olarak tanımlanır. Burada kazanç kümesi içerisine girebilen elemanlar 1 ile diğer elemanlar 0 ile işaretlenmekte ve hesaplama ona göre gerçekleştirilmektedir. n tane eleman verilsin ve çantanızın kapasitesi c olsun. Burada her bir i elemanı için kazanç k_i ve ağırlık a_i olsun. Toplam ağırlık c değerini aşmamak üzere toplam kazanç maksimum olmak zorundadır. Elemanların, çözüm kümesinin bir parçası olup olmadığını d kümesi ile gösterirsek $d = \{0, 1\}$ olur. Burada 0 çözüme dahil olmayan, 1 ise çözümün bir parçası olan elemanları göstermektedir [19]. Sırt çantası probleminin çözümü için uygunluk fonksiyonu olarak Denklem 2 ve Denklem 3 kullanılmaktadır [23].

$$\sum_{i=1}^n a(i) * d(i) \leq c \quad (2)$$

$$\text{Maksimum_Kazanc} = \sum_{i=1}^n k(i) * d(i) \quad (3)$$

Şekil 5'te verilen sırt çantası problemi uygulamaya örnek olarak kullanılmaktadır. Bu problem için DNA hesaplama algoritması kullanılarak optimum çözüm elde edilmektedir. Şekil 5'te verilen sırt çantası problemi için Ağırlık, kazanç ve eleman seçiminde kullanılacak popülasyonun DNA dizileri ile kodlaması yapılmaktadır. Örnek problem için ağırlık ve kazançların DNA ile kodlaması Tablo 6'da ayrıntılı olarak verilmektedir. Her ağırlık ve kazanç 3'lü DNA dizisi ile gösterilmekte A-0, G-1, C-2 ve T-3 değerleri kullanılarak sayısal veriler elde edilmektedir. Örneğin 1. elemanın ağırlığı 25 olduğu için bu değere karşılık gelen GCG dizisi kullanılır. Yine 1. elemanın kazanç değeri 9 olduğu için ACG dizisi ile gösterilir.



Şekil 5. Örnek sırt çantası problemi

Elemanlar = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

Kazanç değerleri = { 10, 7, 10, 14, 17, 6, 13, 4, 10, 15 }

Ağırlık değerleri = {26,17,24,23,28,20,24,28,32,25}

Çantanın kapasitesi c=85

Sırt çantası probleminin DNA hesaplama ile optimizasyonunda kullanılan algoritmik adımlar sırayla aşağıda açıklanmaktadır.

Adım 1. Eleman sayısı 10 olarak belirlenen problemin elemanlarının ağırlıkları, kazanç değerleri ve kapasitesi belirlenir. Ağırlıklar ve kazançlar DNA molekülleri kullanılarak kodlanır. DNA molekülleri kullanılarak rastlantısal olarak ilk popülasyon oluşturulup popülasyondaki diziler 0 ve 1 olarak sayısal değerlere çevrilir. 0 seçilmeyecek elemanları 1 ise seçilecek elemanları göstermektedir. DNA dizileri ile kodlama Tablo 6'da ayrıntılı olarak verilmektedir.

Adım 2. Popülasyonda seçimi belirlenen her bir elemanın kazanç ve ağırlık değerleri kullanılarak uygunluk değerleri hesaplanır. Uygunluk değerlerinin hesaplanmasında Denklem 2 ve Denklem 3 kullanılmaktadır.

Adım 3. Problem için belirlenen kapasite değerini aşan popülasyon elemanları çözümden çıkarılır.

Adım 4. Kalan popülasyon içerisinde maksimum kazançlı elemanlar belirlenir.

Adım 5. İlk popülasyona çaprazlama uygulanarak yeni popülasyon oluşturulup 2, 3 ve 4 adımları uygulanarak maksimum kazanç değerleri belirlenir.

Adım 6. Popülasyona enzim ve virüs mutasyonu uygulanarak yeni popülasyon oluşturulur ve 2, 3, 4 adımları uygulanarak maksimum kazanç değerleri belirlenir.

Adım 7. Maksimum işlem sayısına ulaşıldığında yukarıda belirtilen adımlar sonucunda bulunan maksimum kazanç değeri problemin çözümü olarak kabul edilir.

3. Simülasyon Sonuçları

Yukarıda örnek olarak seçilen NP problemlerin optimizasyonu için matlab programı ile m-file yazılıp program çalıştırılarak sonuçlar elde edilmektedir. Tablo 1 ve Tablo 5'te simülasyon için uygulanan parametre değerleri verilmiş olup verilen bu değerler uygulayıcı tarafından belirlenir. Problemin türüne ve zorluk derecesine göre popülasyon büyüklüğü ve maksimum işlem sayısı artırılıp azaltılabilir. Enzim ve virüs oranı her bir iterasyon için kontrol edilerek ideal değerler belirlenebilir. Bu oranlar gereğinden fazla artırıldığında veya çok sınırlı düzeyde uygulandığında çözüm kümesi için uygun olan değerlerin kaybına veya yerel optimum noktalara odaklanmaya neden olmaktadır. Problemlerin optimizasyonu için matlab ile m-file yazılarak aşağıda verilen sonuçlara ulaşılmıştır.

3.1. Gezgin Satıcı Problemi

Gezgin satıcı problemi için Şekil 4 örnek problem olarak kullanılmaktadır. Bilgisayar simülasyonunda DNA hesaplama ile gezgin satıcı problemi için Tablo 1'de verilen parametre değerleri uygulanmaktadır. Tabloda verildiği üzere şehir sayısı 7 olarak alınmıştır. Şehir sayısı artırıldığında sayısal ortam algoritması ile çözüm zor olmaktadır. Bu durumda çözelti ortamı algoritmasının kullanılması daha doğru olmaktadır. Gezgin satıcı probleminin optimizasyonu için Denklem 1'de verilen uygunluk fonksiyonu kullanılmaktadır. Tablo 2'de şehirlerin, Tablo 3'te ise şehirlerarasındaki mesafelerin DNA dizileri ile kodlanması gösterilmektedir. Tablo 4'te, Tablo 3'te verilen DNA dizilerine karşılık gelen sayısal değerler verilmektedir.

Tablo 1. Gezgin Satıcı Problemi için DNA hesaplama parametreleri

Parametre	Değeri
Şehir Sayısı	7
Populasyon Büyüklüğü	60
Maksimum İşlem Sayısı	100
Mutasyon Oranı	0.3
Enzim ve Virüs Oranı	0.3

Tablo 1’de verilen değerler, program için kullanılan ve deneme yanılma yöntemiyle belirlenen parametre değerleridir.

Tablo 2. Şehirlerin DNA dizileri ile gösterimi

Şehirler	DNA ile gösterimi	karşılığı	
		4’lü tabanda	10’lu tabanda
A	AAG	001	1
B	AAC	002	2
C	AAT	003	3
D	AGA	010	4
E	AGG	011	5
F	AGC	012	6
G	AGT	013	7

Tablo 3. Şehirlerarasındaki mesafelerin DNA dizileri ile gösterimi

Şehir	1	2	3	4	5	6	7
1	AAA	AGC	AAC	AGT	ACC	ATC	GAA
2	AGC	AAA	AAT	AAG	ACA	ACG	ACC
3	AAC	AAT	AAA	AGG	AAC	AGG	AGC
4	AGT	AAG	AGG	AAA	AGA	AAT	AGG
5	ACC	ACA	AAC	AGA	AAA	AAG	AAC
6	ATC	ACG	AGG	AAT	AAG	AAA	AAG
7	GAA	ACC	AGC	AGG	AAC	AAG	AAA

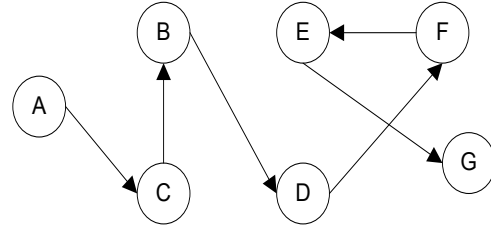
Tablo 3’te verilen DNA dizileri Tablo 4’te verilen sayısal değerlere karşılık gelen kodlamalardır. Örneğin 1. Şehir ile 2. Şehir arasındaki mesafe 6 birim olup AGC dizisi ile kodlanmaktadır. A-0, G-1, C-2 olup AGC -012 olarak 4’lü sayı sistemine ve 6 olarak onluk sisteme dönüştürülmektedir. DNA dizilerinin 4’lü sistemle kullanılmasının nedeni 4 DNA bazının olmasıdır. Bu çalışmada A-0, G-1, C-2 ve T-3 olarak ifade edilmektedir. Gezgin satıcı problemi için yazılan matlab programının çalıştırılması sonucunda en kısa yol uzunluğu 12 olarak bulunmaktadır. Uzunluğu 12 olan 2 tane yol güzergâhı elde edilmektedir. Şekil 6 ve Şekil 7’de gezgin satıcı problemi için bulunan optimum yolların gösterimi verilmektedir.

1. yol güzergâhı: 1 → 3 → 2 → 4 → 6 → 5 → 7

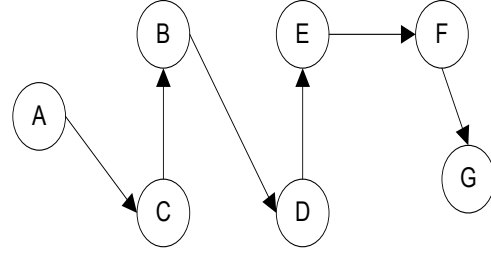
2. yol güzergâhı: 1 → 3 → 2 → 4 → 5 → 6 → 7

Tablo 4. Mesafelerin sayısal değerleri

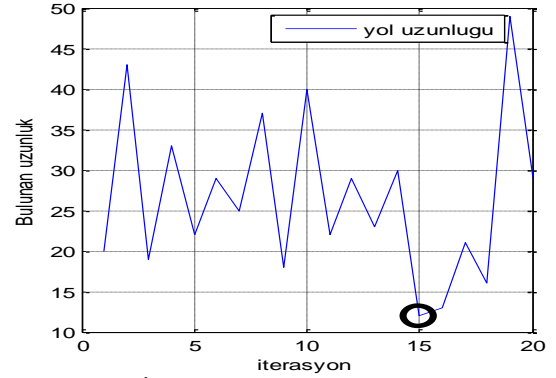
Şehirler	1	2	3	4	5	6	7
1	0	6	2	7	10	14	16
2	6	0	3	1	8	9	10
3	2	3	0	5	2	5	6
4	7	1	5	0	4	3	5
5	10	8	2	4	0	1	2
6	14	9	5	3	1	0	1
7	16	10	6	5	2	1	0



Şekil 6. Gezgin Satıcı Problemi için bulunan 1. Yol güzergâhı



Şekil 7. Gezgin Satıcı Problemi için bulunan 2. Yol güzergâhı



Şekil 8. İterasyon ve bulunan yol uzunlukları

Matlab programı ile yapılan uygulama sonucunda Şekil 8 elde edilmiştir. Şekil üzerinde her bir iterasyon sonucu bulunan yol uzunlukları gösterilmektedir. 15. İterasyonda minimum yol uzunluğu 12 olarak bulunmuştur. Diğer iterasyonlarda bulunan değerler daha büyük olup optimum çözüm olarak 15. İterasyon seçilmiştir. İşlem sayısı 100 olarak kullanılmasına rağmen şekil üzerinde gösterimi zor olduğu için çözüm değerinin bulunduğu konum gösterilmiştir. 12 birim uzunluklu değer elde edildiği yol güzergâhları ise Şekil 6 ve Şekil 7’de verilmiştir.

3.2. Sırt Çantası Problemi

Sırt çantası problemi için Şekil 5’te verilen örnek problem kullanılmaktadır. Sırt çantası problemi için Tablo 5’te verilen parametre değerleri uygulanmaktadır. Sırt çantası problemi için Denklem 2 ve Denklem 3 uygunluk fonksiyonu olarak alınmaktadır. Sırt çantası problemindeki elemanların ağırlık ve kazanç değerlerinin DNA dizileri ile kodlanması Tablo 6’da ayrıntılı olarak gösterilmektedir.

Tablo 5. Sırt Çantası Problemi için DNA hesaplama parametreleri

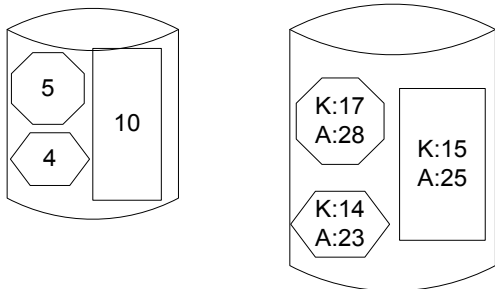
Parametre	Değeri
Eleman Sayısı	10
Popülasyon Büyüklüğü	150
Maksimum İşlem Sayısı	100
Kapasite	85
Enzim ve Virüs Mutasyonu Oranı	0.3

Tablo 6. Ağırlıkların ve Kazançların gösterimi

Ağırlık Değeri	DNA ile kodlama	4'lü kodlama	Kazanç Değeri	DNA ile kodlama	4'lü kodlama
26	GCC	122	10	ACC	022
17	GAG	101	7	AGT	013
24	GCA	120	10	ACC	022
23	GGT	113	14	ATC	032
28	GTA	130	17	GAG	101
20	GGA	110	6	AGC	012
24	GCA	120	13	ATG	031
28	GTA	130	4	AGA	010
32	CAA	200	10	ACC	022
25	GCG	121	15	ATT	033

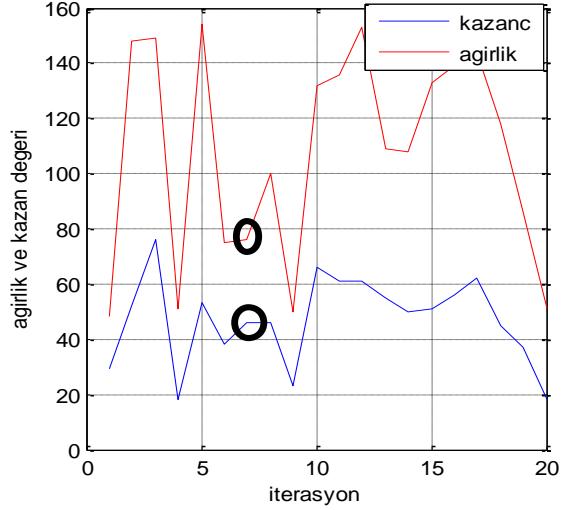
Tablo 6'da görülen ağırlık değeri sütunu, ağırlıkların gerçek değerlerini, 2. Sütun bu değerlerin DNA dizileri ile kodlamasını göstermektedir. Örneğin gerçek değeri 26 olan ağırlığın DNA dizisi olarak karşılığı GCC olarak alınmaktadır. 3. Sütunda ise DNA dizilerinin dördü sayı sisteminde hangi rakam ile gösterildiğini belirtmektedir. Örneğin; G-1, C-2 olduğu için GCC-122 olarak kodlanmaktadır. DNA molekülüne ait 4 baz olduğu için kodlamada 4'lü sayı sistemi kullanılmıştır. Benzer kodlama kazanç değerleri içinde aynı şekilde kullanılmaktadır. Sırt çantası problemi için yazılan matlab programının çalıştırılması sonucunda aşağıda belirtilen değerler elde edilmektedir. Bilgisayar simülasyonu sonucunda 4, 5 ve 10 numaralı elemanlar seçilerek çantaya konulmaktadır. Yapılan işlem sonucunda maksimum kazanç değeri 46 olarak bulunmakta ve toplam ağırlık 76 olmaktadır.

Seçilen elemanlar = 4 5 10
 Maksimum kazanç değeri = 46
 Toplam ağırlık değeri = 76



Şekil 9. Sırt çantası problemi için bulunan çözüm elemanları

Sırt çantası problemi için matlab ile elde edilen sonuçlar Şekil 9 ve Şekil 10'da gösterilmektedir.



Şekil 10. İterasyon ile ağırlık ve kazanç değerleri

Matlab programı ile yapılan uygulama sonucunda Şekil 10 elde edilmiştir. Şekil üzerinde her bir iterasyon sonucu bulunan ağırlık ve kazanç değerleri gösterilmektedir. 7. İterasyonda çantanın kapasitesi olan 85 değerini geçmeyecek şekilde maksimum kazanç olan 46 değeri bulunmuştur. Diğer iterasyonlarda daha büyük kazanç değerleri olmasına karşın çantanın kapasitesini aştığı için optimum çözüm olarak 7. İterasyon seçilmiştir. İşlem sayısı 100 olarak kullanılmasına rağmen şekil üzerinde gösterimi zor olduğu için çözüm değerinin bulunduğu konum gösterilmiştir.

4. Sonuçlar

Sayısal DNA hesaplama algoritması optimizasyon problemlerinde etkin şekilde kullanılabilir. Bu çalışmada, DNA hesaplama algoritması NP problemlerin çözümü için kullanılarak performansı incelenmiştir. Gezgin satıcı ve sırt çantası problemlerine uygulanan DNA hesaplama algoritmasının etkinliği, simülasyon sonuçları ile doğrulanmıştır. Literatürde var olan çalışmalar çözelti ortamında gerçekleştirilmesine karşın bu çalışmada DNA hesaplama sayısal ortama uyarlanmıştır. DNA moleküllerinin çözelti ortamında gerçekleştirdikleri ve problem çözümünde çok etkili olduğu literatürde var olan çalışmalarda açıkça görülen enzim ve virüs mutasyonu, sayısal ortama uyarlanarak kullanılmıştır. Önerilen yöntemle DNA hesaplama sayısal ortama uyarlanarak çözelti ortamına göre uygulanabilirliği artırılmış ve maliyeti düşürülmüştür. Ayrıca çözelti ortamına göre çözüm zamanı belirgin bir şekilde kısalmıştır. Yapılan çalışma ile sayısal ortam DNA hesaplama algoritması polinomal olmayan problemlere uygulanmıştır. Uygulamadaki zorluklar nedeniyle seçilen problemlerin boyutu küçük tutulmuştur. Problem boyutunun artmasıyla beraber hata payının artacağı öngörülse de bu hata payının algoritmadaki geliştirmelerle tolere edilebileceği düşünülmektedir.

5. Kaynaklar

1. Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems, Science, 266(4), p. 1021-1025.
2. Lipton, R.J., 1995. Using DNA to solve NP-complete problems, Science, vol. 268(4), p. 542-545.

3. Ouyang, Q., Kaplan, P., Liu, S., Libchaber, A., 1997. DNA solution of the maximal Clique Problem, Science, vol. 278, no.5337, pp.446-449.
4. Head, T., Rozenberg, G., Bladergroen, S., Breek C., Computing with DNA by operating on Plasmids, BioSystems, vol. 57, p.87-93, 2000.
5. Yang, C.N., Chao, C.H., Cheng, H.P. and Lin, C.H., 2006. A MEMS Based DNA Computer for Solving SAT Problems, Nano/Micro Engineered and Molecular Systems NEMS '06, 1st IEEE International Conference, 18-21 January, p. 172 – 177.
6. Wood, D., Chen, J., Antipov, E., Lemieux, B., and Cedeño, W., 1999. A DNA Implementation of the Max 1s Problem, Proceedings of Genetic and Evolutionary Computation, Orlando, FL, USA, p. 1-7.
7. Ding, Y., and Ren, L., 2000. DNA Genetic Algorithm for Design of the Generalized Membership-Type Takagi-Sugeno Fuzzy Control System, Proceedings of IEEE International Conference on Systems, man, and Cybernetics, 8-11 Oct., vol. 5, p. 3862-3867.
8. Shin, S.Y., Lee, I.H., D., Kim and B.T., Zhang, 2005. Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing, Evolutionary Computation, IEEE Transactions on Vol. 9, p. 143 – 158.
9. Lee, J.Y., Shin, S.Y., Augh, S.J., Park T.H., and Zhang, B.T., 2003. Temperature gradient-based DNA computing for graph problems with weighted edges, M. Hagiya & A. Ohuchi (eds.) DNA computing, 8th international workshop on DNA-based computers. Springer-Verlag, Berlin Heidelberg, p. 73–84.
10. Liu, W., Sun, S., Guo, Y., DNA Computing Model of Perceptron, [Circuits, Communications and Systems, PACCS '09. Pacific-Asia Conference on](#) Page(s):726–729, 2009.
11. Zhixiang, Y., Bosheng, S. and Cheng, Z., 2010. Molecular beacon based DNA computing model for maximum independent set problem, International Conference on Intelligent Computation Technology and Automation 107.
12. Xuncaı, Z., Ying, N. and Yanfeng, W., 2011. DNA Computing in Microreactors A Solution to the Minimum Vertex Cover Problem, Bio-Inspired Computing: Theories and Applications (BIC-TA), Page(s): 236 – 240.
13. Sridhar, R., Balasubramaniam, S., 2011. GIS integrated DNA computing for solving Travelling Salesman Problem, Computers & Informatics (ISCI), pages 402 – 406.
14. Çiğdem, U., Karaköse, M., 2011. PI ve bulanık denetleyici parametrelerinin ayarlanmasında DNA hesaplamının kullanılması, Otomatik kontrol ulusal toplantısı, DEU, İzmir, 14-16 Eylül, S. 665-670.
15. Fei, L., Zheng, L. and Jin, X., 2011. DNA Computing Model Based on Photoelectric Detection System with Magnetic Beads, Bio-Inspired Computing: Theories and Applications (BIC-TA), pages 170 – 175.
16. Yanyan, H. and Ling, H., 2011. DNA computing research progress and application, Computer Science & Education (ICCSE), pages 232 – 235.
17. Xu, J., Qiang, X., Yang, Y., Wang, B., Yang, D., Luo, L., Pan, L. And Wang, S., 2011. An Unenumerative DNA Computing Model for Vertex Coloring Problem, NanoBioscience, Vol. 10 , page(s): 94 – 98.
18. Jiao, H., Zhong, Y., Zhang, L., and Li, P., 2011. Unsupervised remote sensing image classification using an artificial DNA computing, Natural Computation (ICNC), Vol. 3, Page(s): 1341 – 1345.
19. Zhixiang, Y., Chen, H. and Bosheng, S., 2010. Plasmid DNA computing model of 0–1 programming problem, Bio-Inspired Computing: Theories and Applications (BIC-TA), Page(s): 148 – 15.
20. Xu, J., Qiang, X. and Zhang, K., 2010. A paralel type of DNA computing model for graph vertex coloring problem, [Bio-Inspired Computing: Theories and Applications \(BIC-TA\), IEEE Fifth International Conference on](#), pages 231-235.
21. Lin, C.L., Jan, H.Y. and Huang, T.H., 2004. Self-Organizing PID Control Design Based on DNA Computing Method, [Control Applications, Proceedings of the IEEE International Conference on](#) Volume 1, 568–573.
22. Lee, J.Y., Shin, S.Y., Park, T.H., Zhang, B.T., 2004. Solving traveling salesman problems with DNA molecules encoding numerical values, BioSystems 78 (2004) 39–47.
23. Henkel, C.V., Bäck, T., Kok, J.N., Rozenberg, G., Spaink, H.P., 2007. DNA computing of solutions to knapsack problems, BioSystems 88 (2007) 156–162.
24. Henkel, C.V., 2005. Experimental DNA Computing, Doktora Tezi, Leiden Üniversitesi, Gravenhage.
25. Qiu, Z.F., 2003. Advance the DNA computing, Doktora Tezi, Texas A&M University, Texas.
26. Li, 2007. Traveling Salesman Problem Based on DNA Computing, ICNC 2007 Third International Conference on Volume 4, Page(s):28 – 34.