Journal of the Faculty of Engineering and Architecture of Gazi University 41:1 (2026) 479-494

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University
Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Comparison of multiple sequence alignment approaches based on heuristic methods for DNA sequences

Hatice Erdirik[1]*, Abdullah Ammar Karcıoğlu[2], Bahattin Tanyolaç[3], Hasan Bulut[4]

[1]Department of Computer Engineering, Electrical-Electronic Faculty, Yıldız Technical University 34220, İstanbul, Türkiye
[2]Department of Software Engineering, Engineering Faculty, Ataturk University, 25240 Erzurum, Türkiye
[3]Department of Bioengineering, Engineering Faculty, Ege University, 35000, Izmir, Türkiye
[4]Department of Computer Engineering, Engineering Faculty, Ege University, 35000 Izmir, Türkiye

Highlights:

- GA, DE, and hybrid GASA-DESA algorithms were proposed with improvements to address the multiple sequence alignment problem in DNA sequences.
- The proposed algorithms were compared to the ClustalW software in terms of score and runtime.
- The results generally demonstrated superior performance compared to the ClustalW method.

Keywords:

- Multiple Sequence Alignment
- Heuristic Approach
- Genetic Algorithm
- Differential Evaluation
- Simulated Annealing
- Bioinformatic

Article Info:

Research Article
Received: 31.12.2024
Accepted: 06.01.2026

DOI:

10.17341/gazimmfd.1610635

Correspondence:

Author: Hatice Erdirik
e-mail: herdirik@yildiz.edu.tr
phone: +90 530 247 1513

Graphical/Tabular Abstract

In this study, heuristic algorithms based on Genetic Algorithms (GA), Differential Evolution (DE), and Simulated Annealing (SA) are proposed to improve the alignment of divergent sequences. The Needleman–Wunsch (NW) algorithm is used to generate the initial population. The proposed methods are evaluated against ClustalW in terms of alignment accuracy, runtime efficiency, alignment productivity (alignment score/runtime), and computational complexity. Figure A illustrates the proposed methodology.

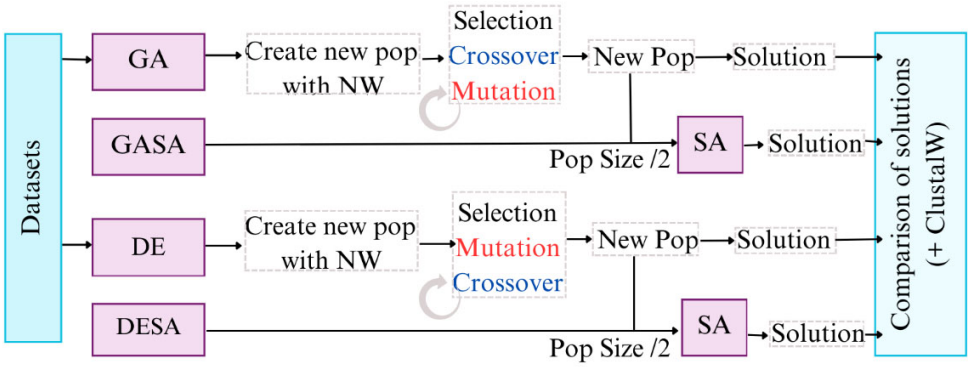


Figure A. Operational framework of the proposed methodologies

Purpose:
This study addresses the multiple sequence alignment (MSA) problem by proposing four heuristic algorithms—GA, DE, GASA, and DESA—that reorder DNA sequences and apply the Needleman–Wunsch algorithm to improve alignment quality. The proposed methods are compared with ClustalW in terms of alignment score, runtime, and computational complexity.

Theory and Methods:
Bioinformatics applies mathematical and computational methods to analyze biological data. To address the complexity of sequence alignment, this study proposes four heuristic algorithms—GA, DE, GASA, and DESA—that integrate the Needleman–Wunsch algorithm to optimize DNA sequence alignment. The proposed methods are evaluated against ClustalW, particularly for datasets with equal-length sequences.

Results:
The proposed algorithms outperformed ClustalW in alignment scores on four of six datasets (trnN-GUU_rps12, rrn4.5_rps12, rrn5_rps12, psbT_pbf1). GA achieved the fastest runtime, while DE and DESA produced the highest alignment scores with longer execution times. Although ClustalW showed the best productivity (alignment score/runtime), the proposed methods provide advantages when alignment accuracy is prioritized, particularly DE and DESA on equal-length sequence datasets.

Conclusion:
The proposed heuristic algorithms (GA, DE, GASA, DESA) achieved higher alignment accuracy than ClustalW in most datasets. Although ClustalW is faster and computationally efficient, it delivered lower alignment quality in several cases. These methods are particularly suitable for bioinformatics tasks requiring high alignment accuracy.

Journal of the Faculty of Engineering and Architecture of Gazi University 41:1 (2026) 479-494

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University
Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

DNA dizileri için sezgisel yöntemlere dayalı çoklu dizi hizalama yaklaşımlarının karşılaştırılması

Hatice Erdirik[1]*, Abdullah Ammar Karcıoğlu[2], Bahattin Tanyolaç[3], Hasan Bulut[4]
[1]Yıldız Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34220, İstanbul, Türkiye
[2]Atatürk Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, 25240, Erzurum, Türkiye
[3]Ege Üniversitesi, Mühendislik Fakültesi, Biyomühendislik Bölümü, 35000, İzmir, Türkiye
[4]Ege Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35000, İzmir, Türkiye

ÖNEÇIKANLAR

- DNA çoklu dizilim problemi için GA, DE ve hibrit GABT-DEBT algoritmaları önerilmiştir
- Önerilen algoritmalar, skor ve çalışma süresi açısından ClustalW yazılımıyla karşılaştırılmıştır
- Önerilen algoritmalar, ClustalW'ye kıyasla çoğu veri kümesinde daha yüksek hizalama puanları elde etmiştir

Makale Bilgileri
Araştırma Makalesi
Geliş: 31.12.2024
Kabul: 06.01.2026

DOI:
10.17341/gazimmfd.1610635

Anahtar Kelimeler:
Çoklu dizi hizalaması, sezgisel yaklaşım, genetik algoritma, diferansiyel evrim, benzetimli tavlama, biyoenformatik

ÖZ

Biyoenformatik, biyolojik verileri kavramsallaştırmak ve aralarındaki ilişkileri anlamak için matematik, bilgi işlem ve istatistikten faydalanan bir bilim dalıdır. Artan biyolojik veri hacmi, dizi hizalama problemini daha karmaşık hale getirerek manuel çözümleri imkânsız kılmıştır. Bu nedenle, otomatik hesaplama sistemleri geliştirilmiştir. Dizi hizalaması, çiftli ve çoklu olmak üzere iki kategoriye ayrılır. Bu çalışma, Genetik Algoritma (GA), Diferansiyel Evrim (DE) ve Benzetimli Tavlama (BT) algoritmaları gibi sezgisel yöntemleri kullanarak çoklu dizi hizalama problemine odaklanmaktadır. GA, DE, GABT ve DEBT algoritmaları önerilmiş ve DNA dizileri üzerinde Needleman-Wunsch algoritmasıyla hizalamalar yapılmıştır. Deneysel sonuçlar, GA algoritmasının en kısa çalışma süresine sahip olduğunu, DE ve DEBT algoritmalarının ise daha uzun sürede çalıştığını ortaya koymaktadır. GA, DE ve DEBT algoritmaları, GABT'ye kıyasla daha yüksek hizalama skorları üretmiştir. Önerilen algoritmalar, ClustalW ile karşılaştırıldığında dört veri kümesinde (trnN-GUU_rps12, rrn4.5_rps12, rrn5_rps12, psbT_pbf1) daha yüksek hizalama doğruluğu sağlamıştır. ClustalW, çalışma süresi açısından en hızlı ve teorik olarak en düşük hesaplama karmaşıklığına sahip yöntem olarak öne çıkmıştır. Verimlilik (hizalama skoru/süre) metriği açısından ise ClustalW genel olarak en yüksek verimliliği sunmuştur. Ancak, önerilen algoritmalar doğruluk odaklı uygulamalarda avantaj sağlamaktadır ve özellikle hassas hizalama gereken senaryolarda tercih edilebilir.

Comparison of multiple sequence alignment approaches based on heuristic methods for DNA sequences

HIGHLIGHTS

- GA, DE, and hybrid GABT-DEBT algorithms are proposed for the DNA multiple sequence alignment problem
- The proposed algorithms were compared to the ClustalW software in terms of score and runtime
- The proposed algorithms achieved higher alignment scores than ClustalW on most datasets

Article Info
Research Article
Received: 31.12.2024
Accepted: 06.01.2026

DOI:
10.17341/gazimmfd.1610635

Keywords:
Multiple sequence alignment, heuristic approach, genetic algorithm, differential evolution, simulated annealing, bioinformatic

ABSTRACT

Bioinformatics is a scientific discipline that utilizes mathematics, computing, and statistics to conceptualize biological data and understand the relationships among them. The increasing volume of biological data has made the sequence alignment problem more complex, rendering manual solutions impractical. As a result, automated computational systems have been developed. Sequence alignment is categorized into pairwise and multiple sequence alignment. This study focuses on solving the multiple sequence alignment problem using heuristic methods such as Genetic Algorithm (GA), Differential Evolution (DE), and Simulated Annealing (SA). The GA, DE, GASA, and DESA algorithms are proposed, and alignments were performed on DNA sequences using the Needleman-Wunsch algorithm.Experimental results show that the GA algorithm has the shortest runtime, while the DE and DESA algorithms take longer to complete. GA, DE, and DESA produced higher alignment scores compared to GASA. When compared to ClustalW, the proposed algorithms achieved higher alignment accuracy in four datasets (trnN-GUU_rps12, rrn4.5_rps12, rrn5_rps12, psbT_pbf1). ClustalW stands out as the fastest method in terms of runtime and has the lowest theoretical computational complexity. In terms of efficiency (alignment score/runtime), ClustalW generally provides the highest efficiency. However, the proposed algorithms offer advantages in accuracy-focused applications and are particularly preferable in scenarios requiring highly precise alignments

*Sorumlu Yazar / Yazarlar / *Corresponding Author / Authors: *herdirik@yildiz.edu.tr, ammar.karcioglu@atauni.edu.tr, bahattin.tanyolac@ege.edu.tr, hasan.bulut@ege.edu.tr / Tel: +90 530 247 1513

1. Giriş (Introduction)

Biyoloji, bilimler arasında anahtar bir rol oynamaktadır. Biyologlar tarafından toplanan büyük ölçekli verilerin yorumlanması ihtiyacı, biyoenformatik alanının ortaya çıkmasına yol açmıştır [1]. Biyoenformatik, biyolojideki molekülleri kavramsallaştırmak ve aralarındaki ilişkileri anlamak amacıyla matematik, bilgisayar ve istatistik gibi disiplinlerden yararlanan bir bilim dalıdır [2] ve birçok uygulama alanına sahiptir [3,4]. Biyolojik dizilerin analizinde en yaygın görevlerden biri dizi hizalamasıdır. DNA dizilerinin analizinde hizalama ve dizi eşleşme problemlerine yönelik geliştirilen algoritmik yaklaşımlar literatürde geniş yer bulmuştur [5,6]. Genetik dizi hizalaması, diziler arasındaki benzer bölgelerin belirlenmesi, evrimsel süreçte korunan bölgelerin tespiti ve genetik hastalıklarla ilişkili değişimlerin ortaya çıkarılması gibi birçok çalışmada kullanılmaktadır. Bu nedenle dizi hizalaması, biyoenformatikte önemli bir araştırma alanıdır [7].

Dizi hizalaması, ikili hizalama ve çoklu dizi hizalama (ÇDH, İng. Multiple Sequence Alignment – MSA) olmak üzere ikiye ayrılır. İkili hizalama yalnızca iki dizinin karşılaştırılmasını ifade ederken, ÇDH üç veya daha fazla dizinin aynı anda hizalanmasını sağlar [8]. ÇDH; protein yapı ve fonksiyon tahmini, filogenetik analiz ve dizi karşılaştırmaları gibi birçok biyoinformatik uygulamada kullanılmaktadır. Bununla birlikte ÇDH problemi NP-tam olarak sınıflandırılmakta olup çözümü üstel hesaplama karmaşıklığına sahiptir [9]. Bu nedenle literatürde farklı yaklaşımlar geliştirilmiştir. Örneğin Zhu vd. çok çekirdekli bilgisayarlarda biyolojik dizilerin hizalanması için veri paralel bir strateji önermiştir [10]. Liu vd. farklı hizalama tiplerine ilişkin üç teorem geliştirmiştir [11]. Naorem vd. ise IL-5 indükleyen peptitlerin tahmini için hizalamaya dayalı ve hizalamasız yöntemleri kullanan bir web sunucusu geliştirmiştir [12].

ÇDH probleminin çözümünde dinamik programlama, sezgisel yöntemler ve istatistiksel yaklaşımlar kullanılmaktadır [13]. Dinamik programlama matematiksel olarak en iyi hizalamayı garanti etmesine rağmen yüksek bellek ve hesaplama maliyeti nedeniyle pratik uygulamalarda sınırlıdır. Bu nedenle literatürde çoğunlukla sezgisel tabanlı algoritmalar tercih edilmektedir. Birçok sezgisel yaklaşım küresel ve yerel hizalama yöntemlerini birleştirerek ilerici hizalama stratejileri geliştirmektedir. Ancak ilerici yöntemlerde erken aşamalarda oluşan hatalar sonraki adımlarda düzeltilememektedir. Bu yaklaşımın ilk örneklerinden biri ClustalW algoritmasıdır ve daha sonra bu yaklaşımı temel alan birçok yöntem geliştirilmiştir [13].

Sezgisel algoritmalar genel olarak tek çözüm tabanlı ve popülasyon tabanlı yöntemler olarak sınıflandırılmaktadır. Tepe Tırmanma, Benzetimli Tavlama ve Tabu Arama tek çözüm tabanlı yöntemlere örnek olarak verilebilirken; Genetik Algoritma ve Diferansiyel Evrim gibi yöntemler popülasyon tabanlı yaklaşımlardır. Son yıllarda yapılan çalışmalar genellikle uygunluk fonksiyonlarının geliştirilmesi, operatörlerin iyileştirilmesi ve hibrit sezgisel yöntemlerin kullanılması üzerine yoğunlaşmıştır [14]. Ayrıca sezgisel ve meta-sezgisel optimizasyon yöntemlerinin yalnızca biyoenformatikte değil, biyomedikal veri analizi ve çeşitli mühendislik problemlerinde de karmaşık optimizasyon problemlerinin çözümünde etkili olduğu gösterilmiştir [15, 16].

Bu çalışmada, sezgisel algoritmalara dayalı yeni ÇDH yaklaşımları önerilmiştir. Genetik Algoritma (GA), Benzetimli Tavlama (BT) ve Diferansiyel Evrim (DE) algoritmaları kullanılarak GA, GABT, DE ve DEBT olmak üzere dört farklı algoritma geliştirilmiştir. Dinamik programlama yöntemi GA ve DE algoritmalarında başlangıç popülasyonunun oluşturulmasında kullanılarak çeşitlilik artırılmış, elde edilen çözümler BT algoritması ile iyileştirilmiştir. Önerilen algoritmalar hizalama doğruluğu ve çalışma süresi açısından ClustalW ile karşılaştırılmıştır. Deneysel sonuçlar GA, DE ve DEBT algoritmalarının GABT'ye göre daha yüksek hizalama skorları elde ettiğini ve GA'nın daha kısa çalışma süresi sağladığını göstermektedir. Ancak çalışma süresi açısından ClustalW en hızlı yöntem olarak öne çıkmaktadır. Bu çalışma, ÇDH probleminin çözümüne yönelik yeni algoritmalar önererek literatüre metodolojik ve performans açısından katkı sunmaktadır.

Makalenin geri kalan kısmı şu şekilde düzenlenmiştir: Bölüm 2'de DNA, RNA ve protein yapıları ile dizi hizalaması ve ÇDH problemi üzerine yapılan çalışmalar ele alınmıştır. Bölüm 3'te GA, BT ve DE algoritmaları ile bu algoritmaların ÇDH problemine uygulanışı açıklanmıştır. Bölüm 4'te kullanılan veri kümesi ve performans ölçütleri tanıtılmıştır. Bölüm 5'te deneysel sonuçlar karşılaştırmalı olarak sunulmuştur. Son olarak Bölüm 6'da elde edilen sonuçlar değerlendirilmiş ve çalışmanın literatüre katkıları tartışılmıştır.

2. Teorik Arka Plan ve İlgili Çalışmalar (Theoretical Background and Related Works)

Bu bölümde ÇDH probleminin özellikleri, gerçekleştirilen hizalamaların nasıl değerlendirildiği ve ÇDH problemini çözmek için geliştirilen algoritmalar ele alınmaktadır.

2.1 Dizi Hizalama Problemi (The Sequence Alignment Problem)

Dizi hizalaması, diziler arasında benzer bölgeleri bulmak için DNA, RNA ve protein dizilerini düzenlemeye yardımcı olur. Dizi hizalama problemindeki her dizi $S=s_1,\ldots,s_n$ olarak gösterilir. İndel ('-' sembolüyle gösterilir), bir dizinin parçalarının eklendiğini veya silindiğini gösterir. Nükleotid bazları Adenin (A), Sitozin (C), Guanin (G), Timin (T) ve Urasil'dir (U). Alfabe sırasıyla DNA ve RNA için A, C, G, T ve A, C, G, U'dur.
Dizi hizalama problemi, yerel hizalama ve küresel hizalama olarak ikiye ayrılır. Küresel hizalamada, diziler bir bütün olarak hizalanırken, yerel dizi hizalamasında benzerlikler tespit edilir [20]. Smith-Waterman algoritması, yerel hizalama algoritmasının bir örneğidir. Küresel dizi hizalaması, her iki dizi için de en iyi hizalamayı bulmak için kullanılır. Needleman-Wunsch algoritması, küresel hizalama algoritmasının bir örneğidir [10]. Ek olarak, ikili hizalama algoritmalarında genellikle dinamik programlama tercih edilir [21].

2.2 Çoklu Dizi Hizalama (ÇDH) Problemi (The MSA Problem)

ÇDH problemindeki temel amaç, diziler arasındaki eşleşen sembol sayısını en üst düzeye çıkarmak ve ayrıca boşluklara izin veriliyorsa yalnızca minimum boşluk kullanmaktır [20]. ÇDH, kombinatoryal problemler adı verilen, üstel zaman karmaşıklığına sahip bir optimizasyon problemleri sınıfına aittir. Zaman karmaşıklığı O (L^N)'dir. L, hizalanacak dizilerin ortalama uzunluğudur ve N, dizilerin hizalanma sayısıdır. Biyolojide, dizilerin uzunlukları yüzlerce (protein), binlerce (RNA) veya milyonlarca (DNA) olduğundan çalışma süreleri çok uzundur. Bu nedenle, ÇDH problemi için genellikle dinamik programlama ve sezgisel yöntemler kullanılır [22].

2.3 Hizalanmış Dizilerin Puanlarının Hesaplanması (Calculating Scores of The Aligned Sequences)

Hizalamaların maliyetinin tanımlanması, optimal hizalamayı diğer olası hizalamalardan ayırt edebilmek için gereklidir. Bu amaçla, birden fazla hizalamanın değerlendirilmesi için çeşitli amaç (hedef) fonksiyonları kullanılır. Bu çalışmada, önerilen yöntemlerin başarısını ölçmek için yaygın olarak kullanılan Çift Toplamı (ÇT) fonksiyonu tercih edilmiştir. ÇT, literatürde Sum-of-Pairs (SP) olarak bilinen ve en sık kullanılan hedef fonksiyonlardan biridir.

ÇDH problemlerinde, N adet hizalanmış dizi bulunduğunda, algoritma bu dizileri $\binom{N}{2}$ adet olası çift halinde hizalar. Bir j. sütun için ÇT puanı Eş. 1 ile, tüm hizalama boyunca elde edilen toplam ÇT puanı ise Eş. 2 ile hesaplanmaktadır.

$$P(j) = \sum_{i=0}^{N-1} \sum_{K=i+1}^{N} PS(C_{ij}, C_{ik}) \quad (1)$$

$$ÇT(A) = \sum_{i=1}^{L} \sum_{J=0}^{N-1} \sum_{K=J+1}^{N} PS(C_{ij}, C_{ik}) \quad (2)$$

Eş. 1 ve Eş. 2'de verilen PS fonksiyonu dizileri çiftler halinde puanlayarak hizalamanın toplam skorunu hesaplamaktadır. PS fonksiyonunda yer alan C_{ij} ve C_{ik} parametreleri, i. sütundaki j. ve k. dizilerin değerlerini temsil etmektedir. PS fonksiyonunun C_{ij} ve C_{ik} parametreleri için kullanılacak puanlama değerleri için belirli bir kural bulunmamaktadır. Bu değerler, her algoritmanın amacına bağlı olarak farklı şekillerde belirlenmektedir.

Boşluk, uyumsuzluk ve uyum puanlarına göre bir hizalama skoru belirlenmektedir. Bu puanlama değerlerinin hesaplanmasında BLOSUM ve PAM matrisleri yaygın olarak kullanılmaktadır. Bu çalışmada hizalama skorunun hesaplanması için BLOSUM matrisi tercih edilmiştir. Blok İkame Matrisi (BLOSUM), protein dizilerinin hizalanmasını puanlamak amacıyla geliştirilmiş bir matristir. Temel yaklaşım olarak PAM matrisine benzemekle birlikte, BLOSUM matrisleri daha geniş bir dizi veri kümesi kullanmakta ve bağımsız kalıntı hizalamaları yerine yerel hizalama bloklarını veya yüksek derecede korunan bölgeleri dikkate almaktadır [23].

2.4 ÇDH Probleminin Çözümüne İlişkin Çalışmalar
(Studies Related to Solving the ÇDH Problem)

ÇDH, ikiden fazla dizinin aynı anda hizalanmasını ifade eder. Filogenetik ağaç inşasında, proteinlerin ikincil ve üçüncül yapılarının tahmininde, fonksiyon tahmininde, primer tasarımında ve tam genom dizisi analizinde yaygın olarak kullanılmaktadır. Çoklu dizi hizalama problemi, hesaplama açısından oldukça karmaşık olup literatürde NP-complete bir problem olarak sınıflandırılmaktadır. Bu nedenle problemin tam çözümü yüksek hesaplama maliyetleri gerektirir ve üstel zaman karmaşıklığına yol açar. Bu zorluk nedeniyle literatürde ÇDH problemini çözmek için farklı algoritmik yaklaşımlar geliştirilmiştir. Genel olarak bu yaklaşımlar iki ana grupta incelenmektedir: Dinamik Programlama tabanlı yöntemler ve sezgisel yöntemler. Sezgisel yöntemler içerisinde ise ilerleyen küresel hizalama (progressive alignment) ve tekrarlayan küresel hizalama (iterative alignment) gibi yaklaşımlar yaygın olarak kullanılmaktadır.

2.4.1. Dinamik programlama kullanılarak geliştirilen çözümler
(Developed solutions using dynamic programming)

Dinamik programlama, bir problemi daha küçük alt problemlere bölerek en uygun çözümü bulmayı amaçlayan bir yaklaşımdır. [24]'te belirtildiği gibi böl ve yönet stratejisine dayanır. Çoklu dizi hizalaması (ÇDH) için teorik olarak en doğru çözümü sağlayabilmesine rağmen, N dizi için N boyutlu bir matris oluşturulmasını gerektirir ve bu durum arama uzayının üstel biçimde büyümesine neden olur. Bu nedenle dinamik programlama ÇDH problemlerinde yaygın olarak kullanılmaz; ancak çiftli dizi hizalamalarında tercih edilir. Bu bağlamda Needleman–Wunsch ve Smith–Waterman algoritmaları en bilinen dinamik programlama tabanlı yöntemlerdir [25].

Küresel hizalama, dizilerin tamamını hizalamayı amaçlar ve özellikle yüksek benzerlik gösteren dizilerde etkilidir. Needleman–Wunsch algoritması [26], dinamik programlama tabanlı bir yöntem olup boşluk ekleyerek optimal küresel hizalama elde eder [27]. Buna karşılık Smith–Waterman algoritması [28], yalnızca en benzer alt bölgeleri hizalayarak yerel hizalama gerçekleştirir ve küresel hizalamada ortaya çıkabilecek yerel uyumsuzluk problemlerini azaltır.

2.4.2. Sezgisel yöntemler kullanılarak geliştirilen çözümler
(Developed solutions using heuristic methods)

Sezgisel yöntemler, ÇDH problemini çözmek için genel olarak artımlı, yinelemeli ve sezgisel yaklaşımlar olarak sınıflandırılır.

Aşamalı küresel hizalama**,** en benzer dizi çiftinden başlayarak daha az benzer çiftlere doğru ilerleyen çiftli hizalamaların birleştirilmesiyle çoklu hizalama oluşturur. Bu yaklaşımın temel sorunları yerel minimuma takılma ve hizalama parametrelerinin seçimidir. Algoritma dizileri kılavuz ağacına göre açgözlü (greedy) biçimde birleştirdiği için erken aşamalarda yapılan hatalar sonraki adımlarda düzeltilemez. Ayrıca ağırlık matrisi ve boşluk cezalarının uygun seçilmemesi, optimuma yakın çözümlerin elde edilmesini zorlaştırabilir. Bu yaklaşımı kullanan yaygın programlar arasında ClustalW, T-Coffee ve MUSCLE bulunmaktadır [21, 29].

İteratif küresel hizalama ise başlangıçta elde edilen hizalamayı, daha iyi bir sonuç elde edilene kadar tekrarlı biçimde iyileştiren algoritmalara dayanır. Bu yöntemler deterministik veya stokastik olabilir. Deterministik yaklaşımlar genellikle dizilerin hizalamadan çıkarılıp kalan dizilerle yeniden hizalanmasına dayanırken, stokastik yaklaşımlar arasında Gizli Markov Modelleri (HMM), Benzetimli Tavlama ve genetik algoritmalar yer alır. İyileştirme sağlanamadığında algoritma sonlandırılır [11, 21, 30].

3. Önerilen Yaklaşımlar (The Proposed Approaches)

3.1 Veri Kümesi için Ön İşlem (The Preprocessing for Dataset)

Başlangıç popülasyonunu oluştururken eşit uzunluktaki dizilerde çiftler halinde hizalama kullanılmıştır. Tüm dizilerin her veri kümesi için aynı uzunluğa sahip olması mümkün değildir. Veri kümesindeki dizilerin hizalanabilir olması için, daha kısa olanlara boşluklar eklenerek uzunluklar eşitlenmiştir. Veri kümesindeki en uzun dizi uzunluğuna dayanarak, dizinin başına, sonuna veya her dizinin bu uzunluğa olan uzaklığına eşit dizinin herhangi bir endeks değerine rastgele boşluklar eklenmiştir. Önceden işlenmiş ve önceden işlenmemiş veri kümelerinin örnekleri Şekil 1'de gösterilmiştir.

3.2. Genetik Algoritma (GA) (Genetic Algorithm (GA))

Genetik Algoritma (GA), doğal seçilim ve genetik prensiplerine dayanan popülasyon tabanlı bir arama algoritmasıdır. GA'da aday çözümler kromozom olarak ifade edilir ve kromozomları oluşturan her bir eleman gen olarak adlandırılır. Algoritma, aday çözümlerden oluşan bir popülasyon üzerinde çalışır. Bu nedenle popülasyon büyüklüğü, GA'nın performansı ve ölçeklenebilirliği üzerinde önemli bir etkiye sahiptir. Ancak popülasyon boyutunun çok büyük olması hesaplama süresini artırabilir. Uygun popülasyon boyutu genellikle farklı değerler denenerek belirlenir [31]. GA'da yeni çözümler üretmek için seçim, çaprazlama ve mutasyon operatörleri kullanılır.

Başlangıç popülasyonu*:* GA algoritmasının performansı açısından önemli bir rol oynar çünkü algoritmanın arama alanını keşfetmesi için temel oluşturur. Bu çalışmada, veri seti her yinelemede karıştırılarak çeşitli bir başlangıç popülasyonu oluşturulur ve her diziye konumunu izlemek için bir kimlik indeksi atanır. Karıştırılan veri setindeki ilk dizi, Needleman–Wunsch algoritması kullanılarak rastgele seçilen farklı bir diziyle hizalanır. Elde edilen hizalanmış dizi yeni kromozomun ilk elemanını oluşturur ve kimlik indeksleri korunur. Bu yaklaşım, yalnızca rastgele başlatmaya dayanan yöntemlere kıyasla daha dengeli ve çeşitli bir başlangıç popülasyonu sağlayarak daha tutarlı hizalama sonuçlarına katkıda bulunur.

(a) Ön işlemden önce :

```
CTATCGAGTCTTCCCTCCCTCCTTCTCTGCCCCCTCCGCTCCCGCT
CCCTCCACCCTACAAGTGGCCTACAGGGCACAGGTGAGGCGGGACTGGAC
AGCTCCTGCTTTGATCGCCGGAGATCTGCAAATTCTGCCC
TGCAGAGCACTCCGACGTGTCCCATAGTGTTTCCAAACTTGGAAAGGGCG
GGGGAGGGCGGGAGGATGCGGAGGGCGGAGGTATGCAGACAACGAGT
AGTTTCCCCTTGAAAGCCTCAAAAGTGTCCACGTCCTCAAAAAGAATGGA
ACCAATTTAAGAAGCCAGCCCCGTGG
CCTCCTCTGCGCCCCCGCAGGCTCCTCCCAGC
CAGCCCCAGCCCTCCCATTGGTGGAGGCCCTTTTGGAGGCACCCTAGGGC
CAGGGAAACTTTTGCCGTATAAATAGGGCAGATCCGGGCTTTATTA
```

(b) Ön işlemden sonra:

```
CTATCGAGTCTTCCCTCCCTCCTTCTCTGCCCCCTCCGCTCCCGCT----
CCCTCCACCCTACAAGTGGCCTACAGGGCACAGGTGAGGCGGGACTGGAC
--AGCTCCTGCTTTGATCGCCGGAGATCTGCA-AATTCTGCCC-------
TGCAGAGCACTCCGACGTGTCCCATAGTGTTTCCAAACTTGGAAAGGGCG
GGGGAGGGCGGGAGGATGCGGAGGGCGGAGGTATGCAGACAACGAGT---
AGTTTCCCCTTGAAAGCCTCAAAAGTGTCCACGTCCTCAAAAAGAATGGA
----------ACCAATTTA-AGAAGC-CAG--CC-CCGTGG---------
---------CCTCCTCT-G-CGCCCC-GCA-GGCTC-CTCCCAGC-----
CAGCCCCAGCCCTCCCATTGGTGGAGGCCCTTTTGGAGGCACCCTAGGGC
-CAGGGAAACTTTTGCCGTATAAA-TAG-GGCAGATCCGGGCT-TTATTA
```

Şekil 1. Veri setinin ön işleme tabi tutulmasından önce (a) ve sonra (b) (Before(a) and after(b) pre-processing of dataset.)

Seçim: Doğal seçilimden esinlenerek, yüksek uygunluk değerlerine sahip bireylerin özelliklerinin sonraki nesillere aktarılmasını sağlar [32]. Bu çalışmada rulet tekerleği seçim yöntemi kullanılmıştır. Bu yöntemde bireylerin uygunluk değerleri normalleştirilerek bir çiftleşme havuzu oluşturulur ve bireyler uygunluklarına bağlı olarak farklı olasılıklarla temsil edilir. Ardından bu havuzdan iki ebeveyn rastgele seçilerek daha uygun bireylerin seçilme olasılığı artırılır.

Çaprazlama: Seçim sürecinden sonra ebeveyn bireyler yavru üretmek amacıyla çaprazlanır. Çaprazlama, iki ebeveynden gelen genetik materyali birleştirerek popülasyondaki genetik çeşitliliği artıran önemli bir işlemdir. Kullanılan çaprazlama yöntemi, ele alınan optimizasyon problemine bağlı olarak değişebilir.

Bu çalışmada, genetik algoritma tabanlı yöntemlerde bireyler arasındaki genetik çeşitliliği sağlamak amacıyla tek noktalı çaprazlama yöntemi uygulanmıştır. Her iki ebeveynin kromozomu satır sayısına bağlı olarak rastgele belirlenen bir noktadan yatay olarak kesilmiş, bu noktadan itibaren genetik materyal karşılıklı olarak değiştirilmiştir. Bu işlem sonucunda, ebeveynlerin genetik özelliklerinin kombinasyonundan oluşan iki yeni yavru birey elde edilmiştir. Kullanılan tek noktalı çaprazlama yöntemi, basitliği ve geniş uygulanabilirliği nedeniyle tercih edilmiştir. Çaprazlama işlemi Şekil 2'te örneklendirilmiştir.

Mutasyon: Çözümlerin birbirine benzemesini engeller ve GA algoritmasının yerel çözümlerden kaçma olasılığını artırır. GA'da mutasyon oranı düşük tutulur. Bunun nedeni, yüksek mutasyon oranlarının GA'yı ilkel bir rastgele algoritmaya dönüştürmesidir [32].

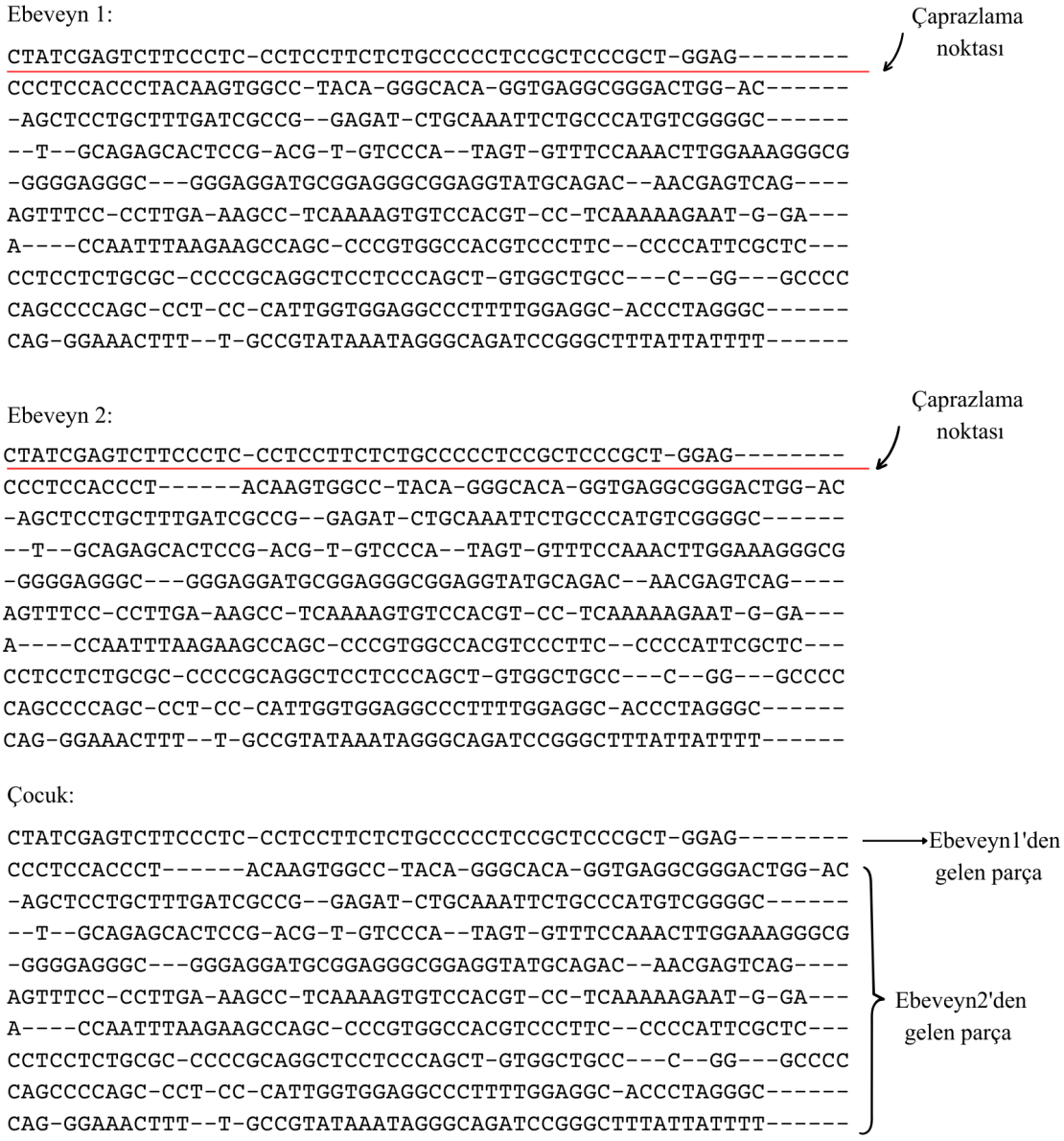
Bu çalışmada, genetik algoritmanın yerel çözümlere sıkışmasını önlemek ve çözüm çeşitliliğini artırmak amacıyla çaprazlama işleminden sonra yavru kromozoma mutasyon uygulanmıştır. Kromozomlar, nükleotid ve boşluk karakterlerinden oluşmaktadır. Mutasyon işlemi, kromozom üzerinde rastgele bir konuma boşluk ekleme veya boşluk kaldırma şeklinde gerçekleştirilmiş olup, her iki işlem eşit olasılıkla (0,5) uygulanmıştır. Bu yöntemle, çözümlerde küçük yapısal değişiklikler sağlanarak yeni potansiyel hizalama alternatiflerinin keşfedilmesi hedeflenmiştir. Mutasyon işlemi Şekil 3'te örneklendirilmiştir.

3.3 Diferansiyel Evrim Algoritması (DE)
(Differential Evolution Algorithm (DE))

İkili kodlama (0,1) kullanan standart genetik algoritmaların gerçek parametreli problemlerde bazı zorluklar oluşturması nedeniyle bu sınırlamaları aşmak amacıyla Diferansiyel Evrim (DE) algoritması geliştirilmiştir. DE, özellikle sürekli veriler içeren optimizasyon problemlerinde başarılı sonuçlar veren popülasyon tabanlı bir yöntemdir. Algoritmanın temel avantajları arasında başlangıç parametre değerlerinden bağımsız olarak küresel minimuma yakın çözümler bulabilmesi, hızlı yakınsama sağlaması ve az sayıda kontrol parametresi gerektirmesi yer almaktadır [33].

DE algoritmasında, D değişkenli bir optimizasyon problemi D boyutlu bir vektör ile temsil edilir ve bu vektörler bir popülasyon oluşturur. Optimizasyon süreci, popülasyon üzerinde uygulanan mutasyon, çaprazlama ve seçim operatörleri aracılığıyla gerçekleştirilir. Bu operatörler sayesinde popülasyondaki bireyler her nesilde güncellenir ve daha iyi çözümlere doğru bir evrim süreci gerçekleşir.

Parametreler:
NP: popülasyon boyutu (kromozom sayısı) $N \geq 4(1,2,3,\ldots i)$,
D: varyant sayısı (gen sayısı) $(1, 2, 3, \ldots j)$,
CR: çaprazlama oranı $(0.1,1.0)$,
G: nesil $(G1, G2, G3, \ldots, Gmax)$,
F: ölçekleme faktörü.
Algoritmada kullanılan bazı temel değişkenler aşağıdaki şekilde tanımlanmaktadır:



Şekil 2. Çalışmada gerçekleştirilen bir çaprazlama örneği (Example of a crossover performed in the study)

$X_{j,i,G}$: G neslinde kromozom j'nin parametresi i(gen)
$n_{j,i,G+1}$: mutasyon ve çaprazlama geçirmiş ara kromozom
$u_{j,i,G+1}$: $X_{j,i,G}$ ' den sonraki nesil için üretilen kromozom
$r_{1,2,3}$: yeni kromozomlar oluşturmak için rastgele seçilen kromozomlar,$r_{1,2,3}$ ε 1,2,3,… NP , $r_1 \neq r_{2\neq}r_{3\neq}i$.

Bu kromozomlar mutasyon işlemi sırasında kullanılır ve popülasyondan birbirinden farklı bireyler olacak şekilde rastgele seçilir. Probleme ait değişkenlerin sayısı, her kromozoma ait gen sayısını belirler. Bir popülasyondaki kromozom sayısı kullanıcı tarafından belirlenir. Bir popülasyondaki kromozom sayısı her zaman üçten fazla olmalıdır. Çünkü algoritmadaki mutasyon süreci üç kromozom seçilerek başlar. Bu nedenle NP parametresinin yeterince büyük seçilmesi algoritmanın arama kabiliyetini artırır.

Başlangıç popülasyonu ve popülasyondaki her genin ifade edilme şekli probleme bağlı olarak farklılık gösterir. Bu nedenle, probleme uygun bir temsil yöntemi ve başlangıç popülasyonu seçimi algoritmanın performansı açısından önemli bir rol oynamaktadır.

Başlangıç popülasyonu: Başlangıç popülasyonu, DE algoritmasının performansı ve yakınsaması için kritik öneme sahiptir. Bu çalışmada, başlangıç popülasyonu veri setinin orijinal sırası korunarak oluşturulmuştur. Her bir dizi, yeni kromozomun ilk elemanını oluşturan, rastgele seçilen bir diziyle Needleman-Wunsch algoritması kullanılarak çiftler halinde hizalamaya tabi tutulur. İkinci dizi, ikinci eleman olmak için aynı işlemden geçer ve bu yinelemeli yaklaşım, tüm popülasyon dolana kadar devam eder. Bu yapılandırılmış yöntem, daha tutarlı ve yüksek kaliteli bir başlangıç popülasyonu sağlayarak, iyileştirilmiş hizalama performansı ve genel algoritma verimliliği için güçlü bir temel oluşturur.

Mutasyon: Diferansiyel gelişim algoritmasında, mutasyona uğrayacak kromozom haricinde üç farklı kromozom seçilir. İlk iki kromozomun farkları alınır. Yeni oluşan fark kromozomu sabit F ile çarpılır. Elde edilen yeni kromozom, seçilen üçüncü kromozom ile toplanır.

Sonuç olarak çaprazlama işleminde kullanılacak kromozom elde edilir. DE'deki mutasyon, Eş. 3'de gösterildiği gibi hesaplanır.

Şekil 3. Çalışmada gerçekleştirilen mutasyon örneği (Example of mutation performed in this study)

$$\forall_j \leq D: \mathrm{n}_{j,i,G+1} = x_{j,r_3,G} + F \times (x_{j,r_1,G} - x_{j,r_2,G}) \quad (3)$$

Çaprazlama: Çaprazlama işlemi sırasında, mutasyondan kaynaklanan fark kromozomu ve x_(i,G) kromozomu kullanılarak yeni bir u_(i,G+1) kromozomu oluşturulur. Yeni u_(i,G+1) kromozomunun genleri CR'ye (çaprazlama oranı) göre belirlenir. Çaprazlama, Eş. 4'de gösterildiği gibi hesaplanır.

$$\forall_j: \mathrm{x}_{j,u,G+1} = \begin{cases} \mathrm{x}_{j,n,G+1} \; if \; rand[0,1] \leq RC \vee j = j_{rand} \\ \mathrm{x}_{j,n,G+1} \; aksi \; taktirde \end{cases} \quad (4)$$

Uygunluk Fonksiyonu ve Seçim: Mutasyon ve çaprazlama işlemleri sonucu oluşan kromozom ve hedef kromozom kullanılarak yeni bir deneme kromozomu elde edilir. Hangi kromozomun G+1 yeni jenerasyona aktarılacağına hedef kromozomun uygunluk değeri ile deneme kromozomunun uygunluk değeri karşılaştırılarak karar verilir. Uygunluk değeri yüksek olan kromozom yeni G+1 jenerasyona aktarılır. Uygunluk değeri ÇT yöntemi ile hesaplanır. Seçim işlemi Eş. 5'de gösterildiği gibi hesaplanır.

$$\forall_j \leq NP: \mathrm{x}_{i,G+1} = \begin{cases} \mathrm{x}_{u,G+1} \; if \; f(\mathrm{x}_{u,G+1}) \geq f(\mathrm{x}_{u,G}) \\ \mathrm{x}_{j,i,G} \; aksi \; takdirde \end{cases} \quad (5)$$

Çoklu Dizi Hizalama (ÇDH) problemine Diferansiyel Evrim (DE) algoritmasının uygulanması sürecinde, başlangıç popülasyonundaki kromozomlar ikili biçimde temsil edilir. Bu temsilde boşluk karakterleri "0" ile, nükleotid karakterleri ise "1" ile kodlanır; bu aşama ileri dönüşüm (encoding) olarak adlandırılır. Mutasyon ve çaprazlama gibi evrimsel işlemler bu ikili temsiller üzerinde gerçekleştirilir. Ardından, bireylerin uygunluk değerlerinin hesaplanabilmesi için ikili format, tekrar nükleotid dizilerine dönüştürülür; bu işlem geri dönüşüm (decoding) olarak tanımlanır. Söz konusu dönüşümler ve evrimsel adımlar sırasında uygulanan işlemler Şekil 4'te özetlenmiştir.

3.4 Benzetimli Tavlama (BT) (Simulated Annealing (SA))

Benzetimli Tavlama (BT) algoritması özellikle, yerel minimumlara takılmadan küresel optimumu bulma olasılığının yüksek olduğu, büyük ve karmaşık çözüm uzaylarına sahip optimizasyon problemleri için tercih edilir. BT algoritması, katıların tavlama işlemi ile optimizasyon problemleri arasındaki ilişkiye dayanır [34]. Bu işlemde, katıyı yüksek sıcaklıkta eritip daha sonra yavaşça soğutarak enerjiyi en aza indirme ilkesi optimizasyon problemlerine uyarlanır.

BT'nın temel adımlarından biri olan döngü, kromozom üzerinde rastgele hareket seçer ve bu hareket mutasyon olarak tanımlanır. Mevcut değişken, hizalanmış dizilere sahip bir kromozomdur. Bu kromozoma %50 olasılıkla bir mutasyon işlemi uygulanır. Mutasyon işleminde, boşluk ekleme ve boşluk silme gibi işlemler yapılır. Bu işlemde amaç uygunluk değerini artırmaktır. Mutasyon nedeniyle elde edilen değer uygunluk değerini artırıyorsa, bu hemen kabul edilir ve mutasyon nedeniyle elde edilen kromozom mevcut değer olarak alınır. Ancak mutasyon uygunluk değerini artırmıyorsa, $e^{(-(DE/T))}$ ifadesi 0-1 aralığından rastgele seçilen bir sayıdan büyükse, mutasyon nedeniyle elde edilen kromozom kabul edilir.

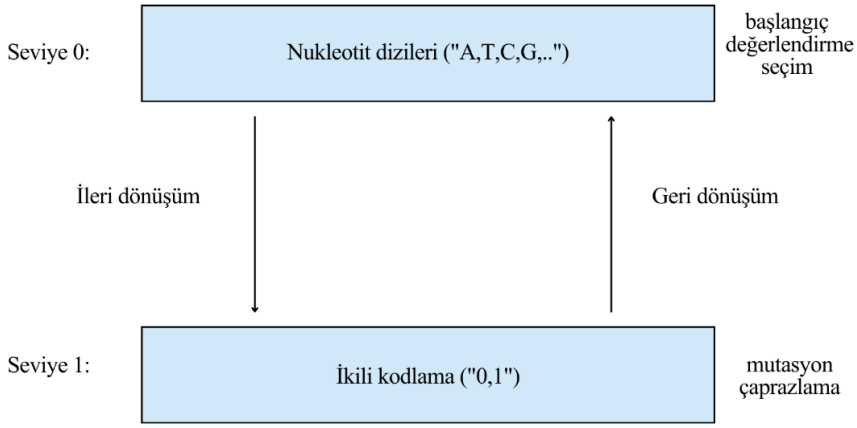
Bu algoritmanın temel prensibi, sıcaklık yavaşça düşürüldükçe, olasılık 1'e yaklaştıkça algoritmanın küresel optimumu bulma şansının artmasıdır. BT algoritmasında, başlangıç sıcaklığı ve soğuma hızı algoritmanın doğruluğu için çok önemlidir. BT algoritması, deterministik algoritmaların yerel minimumlara takılma riskinin olduğu ve çözüm alanının çok büyük olduğu problemler için ideal bir seçim olabilir. Örneğin, protein dizilerinin hizalanması, karmaşık rota optimizasyon problemleri veya üretim planlaması gibi problemler BT ile etkili bir şekilde çözülebilir.

3.5. Hibrit GABT Algoritması (Hybrid GASA Algorithm)

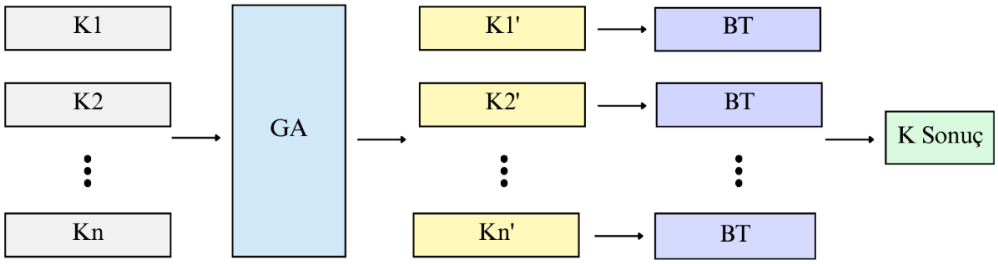
Önerilen GABT algoritması, daha verimli bir optimizasyon süreci oluşturmak için Genetik Algoritma (GA) ve Benzetimli Tavlamanın (BT) güçlü yönlerini birleştirir. Başlangıçta, her bireyin çözüm alanında potansiyel bir çözümü temsil ettiği bireylerden oluşan bir başlangıç popülasyonu, GA'ya sağlanır. GA, çözümleri geliştirmek, daha uygun kromozomları teşvik etmek ve çeşitliliği korumak için seçim, çaprazlama ve mutasyon kullanarak çalışır. GA algoritması sonlanmadan önce oluşan son popülasyondaki ilk 50 kromozom daha fazla iyileştirme için BT algoritmasına verilir. Bu aşamada, BT, yerel optimumlardan kaçınmak ve neredeyse küresel bir optimuma doğru yakınsamak için kontrollü rastgeleleştirme (tavlama) kullanarak çözüm kalitesini artırmak için çözüm alanını hızla araştırır. Sonuç olarak, BT, GA tarafından sağlanan iyi çözümleri iyileştirerek optimuma en yakın çözümü üretir. Bu hibrit GABT algoritması, GA'nın küresel arama ve çeşitlilik bakımından yararlanırken, BT, optimuma yakın sonuçlar için çözümleri ince ayarlayarak süreci optimum çözümleri bulmada daha verimli ve etkili hale getirir. Önerilen hibrit GABT algoritmasının çalışması Şekil 5'te gösterilmektedir.

3.6. Hibrit DEBT Algoritması (Hybrid DESA Algorithm)

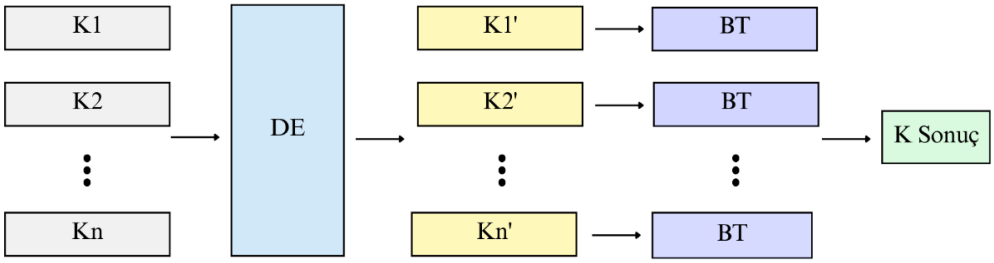
Önerilen DEBT algoritması, optimizasyon sürecini geliştirmek için Diferansiyel Evrim (DE) ve Benzetimli Tavlamanın (BT) güçlü yönlerini birleştirir. DE algoritmasına girdi olarak sağlanan olası çözümlerin ilk popülasyonunu oluşturarak başlar. Her birey, çözüm alanında bir aday çözümü temsil eder. DE algoritması daha sonra üç temel işlemi kullanarak birkaç yineleme üzerinde çalışır: mutasyon, çaprazlama ve seçim. Birden fazla yinelemeden sonra, DE popülasyon kalitesini iyileştirerek optimum çözümlere yaklaşır. Benzetimli Tavlama aşamasında, DE'den gelen ilk 50 kromozom daha da rafine edilir. Bu aşamada, BT, yerel optimumlardan kaçınmak ve neredeyse küresel bir optimuma doğru yakınsamak için kontrollü rastgeleleştirme (tavlama) kullanarak çözüm kalitesini artırmak için çözüm alanını hızla araştırır. Sonuç olarak, BT, küresel optimuma en yakın hizalama değerine sahip bir kromozom üretir ve keşif ile yerel iyileştirme arasında dengeli bir optimizasyon sağlamak için en iyi DE çözümlerini iyileştirir. DEBT algoritması, DE'nin keşfini BT'nın yerel optimizasyonuyla etkili bir şekilde birleştirerek yüksek kaliteli çözümleri verimli bir şekilde bulma şansını artırır. Önerilen hibrit DEBT algoritmasının çalışması Şekil 6'da gösterilmiştir.



Şekil 4. DE algoritmasının dönüşüm adımları (The transformation steps of DE algorithm)



Şekil 5. Önerilen GABT algoritmasının çerçevesi (The framework of the proposed GASA algorithm)



Şekil 6. Önerilen DEBT algoritmasının çerçevesi (The framework of the proposed DESA algorithm)

4. Veri Seti ve Performans Metrikleri
(The Datasets and Performance Metrics)

Bu bölümde, önerilen algoritmaların performansını test etmek ve performanslarını diğer iyi bilinen algoritmalarla karşılaştırmak amacıyla veri kümeleri ve performans ölçümleri açıklanmaktadır.

4.1. Veri Seti (The Datasets)

Bu çalışmada, Cicer reticulatum kloroplast geni (GenBank Assembly: GCA_003689015.2), NCBI'den indirilerek algoritmaların doğruluğu ve çalışma süresi performanslarını değerlendirmek için kullanılmıştır. Referans veri setindeki CDS (Coding Sequence – Kodlayan Dizi) başlıkları, genlerin başlangıç ve bitiş indekslerini belirtmektedir. Belirli başlangıç ve bitiş indekslerine sahip genler seçilmiş ve tamamlayıcı genler oluşturulmuştur (örn. A ↔ T, C ↔ G dönüşümü). Sezgisel bir yaklaşım ile tamamlayıcı genlerin son veya ilk 10 nükleotidinin eşleştiği indeksler bulunmuş, uyumsuzluk durumunda eşleşme kriterleri azaltılarak yaklaşık hizalamalar yapılmıştır. Tam eşleşmeler bulunduğunda, tamamlayıcı gen parçaları başlangıç veya bitiş indekslerine göre hizalanmış ve puan hesaplanmıştır. "rps12", "pbf1" ve "rps18" genleri, Cicer reticulatum kloroplast geninde başlangıç ve bitiş indeksleri olarak seçilmiştir. Bu süreç, olası eşleşme indekslerini belirlemek ve dizi hizalamayı optimize etmek amacıyla gerçekleştirilmiştir. Bu genlere ait özellikler Tablo 1'de gösterilmiştir.

Cicer reticulatum kloroplast veri setinden elde edilen genler hariç, Opuntia veri seti (farklı uzunluk) önerilen algoritmaların performansını test etmek için kullanılmıştır. Dizi setinde sekiz DNA dizisi bulunmaktadır. Veri setindeki en uzun dizi boyutu 902, en kısa dizi boyutu ise 893'tür. Gen dizi benzerliği yüzdesi %63'tür.

4.2. Performans Metrikleri (The Performance Metrics)

Önerilen ÇDH algoritmalarının performansını test etmek ve bunları diğer iyi bilinen algoritmalarla karşılaştırmak için bazı performans ölçütlerine ihtiyaç vardır. ÇDH probleminde bir yöntem seçerken üç husus dikkate alınır; biyolojik doğruluk, çalışma süresi, bellek kullanımı. ÇDH için en önemli husus doğruluktur [35]. Bu çalışmada, algoritmaların performans ölçütleri biyolojik doğruluk ve çalışma süresi olarak seçilmiştir.

Hizalamanın ÇT puanı, ÇDH sonucundan elde edilen hizalama BLOSUM62 matrisi kullanılarak hesaplanır. Aynı veri kümesi her algoritmaya girdi olarak verilir ve elde edilen hizalamaların puanları birbirleriyle karşılaştırılır. Daha yüksek ÇT puanına sahip algoritmanın hizalama doğruluğu daha iyi kabul edilir. Çalışma süresi, algoritmanın verimliliğini gösteren bir diğer parametredir. Algoritmaların çalışma süreleri saniye cinsinden karşılaştırılır.

Tablo 1. Genlerin Özellikleri (The properties of the genes)

Gen Kodları	Gen Veri Seti Adı	Dizi Uzunlukları	Dizi Sayısı	Benzerlik (%)
	trnN-GUU	82	3	28
rps12	rrn5	143	4	27
	rrn4.5	124	3	30
pbf1	psbT	118	3	26
rps18	rpl33	242	11	33

5. Deneysel Çalışmalar (Experımental Studies)

Bu bölümde önerilen algoritmanın parametreleri değiştirilerek elde edilen puanlar karşılaştırılmıştır. Önerilen algoritmalar, en yüksek puanın elde edildiği parametrelere göre karşılaştırılmıştır. Algoritmalar, python programlama dilinde uygulanmıştır.

5.1. Önerilen Algoritmaların Parametre Seçimleri
(Parameters of The Proposed Algorithms)

Parametreler algoritmaların başarısını etkileyen önemli faktörlerden biridir. Önerilen GA, GABT, DE ve DEBT algoritmalarının doğruluğu algoritmadaki parametre değerlerine göre değişmektedir. Önerilen algoritmalarda ÇDH probleminin çözümünde kullanılacak en iyi parametre değerlerini belirlemek için algoritma parametrelerine farklı değerler atanmış ve algoritmanın doğruluğu değerlendirilmiştir.

GA'nın ÇDH problemine uygun parametre değerlerini belirlemek amacıyla, iterasyon sayısı, popülasyon büyüklüğü ve mutasyon oranı farklı değerlerle test edilmiştir. Her bir parametre kombinasyonu, her veri kümesinde 30 kez çalıştırılmış; hizalama skorları ve çalışma sürelerinin ortalama ve standart sapmaları hesaplanarak analiz edilmiştir.

İlk olarak, popülasyon büyüklüğü 100 ve mutasyon oranı 0,05 sabitlenerek iterasyon sayısı 100, 200 ve 400 olarak test edilmiştir. İterasyon sayısı 400'e çıkarıldığında, genellikle daha yüksek hizalama skorları elde edilmiş; artan çalışma süresine rağmen doğruluktaki iyileşme bu artışı haklı çıkarmıştır. Bu nedenle iterasyon sayısı 400 olarak seçilmiştir.

Ardından, iterasyon sayısı 400 ve mutasyon oranı 0,05 sabit tutularak popülasyon büyüklüğü 50, 100 ve 150 olarak denenmiştir. 100 ve 150 bireylik popülasyonlar benzer doğruluk sunarken, daha kısa çalışma süresi nedeniyle 100 bireylik popülasyon değeri tercih edilmiştir.

Son olarak, iterasyon sayısı 400 ve popülasyon büyüklüğü 100 sabit tutularak mutasyon oranı 0,01, 0,015 ve 0,05 olarak test edilmiştir. 0,05 mutasyon oranı en iyi hizalama skorlarını sağlamış, bu oranlardaki değişimin çalışma süresi üzerinde önemli bir etkisi olmamıştır.

Tüm bu değerlendirmeler sonucunda, ÇDH probleminin çözümü için GA algoritması için önerilen parametreler: iterasyon sayısı = 400, popülasyon büyüklüğü = 100 ve mutasyon oranı = 0,05 olarak belirlenmiştir. İlgili tüm deneysel sonuçlar Tablo 2'de sunulmuştur.

DE'nin ÇDH problemine uygulanabilirliğini artırmak amacıyla, iterasyon sayısı, popülasyon büyüklüğü, F-sabiti ve çaprazlama oranı gibi temel parametreler farklı değerlerle test edilmiştir. Her bir parametre kombinasyonu, her veri kümesi için algoritma 30 kez çalıştırılarak; elde edilen hizalama skorları ve çalışma sürelerinin ortalama ve standart sapmaları hesaplanmış, kapsamlı bir performans analizi gerçekleştirilmiştir.

İlk olarak iterasyon sayısının etkisini incelemek amacıyla popülasyon büyüklüğü 100, F-sabiti ve çaprazlama oranı 0,8 olarak sabit tutulmuş; iterasyon sayısı 100, 200 ve 400 olarak denenmiştir. 100 iterasyonda düşük hizalama skorları elde edilirken, 200 iterasyonda daha başarılı sonuçlar gözlemlenmiş, ancak 400 iterasyon hem daha fazla zaman gerektirmiş hem de anlamlı bir doğruluk artışı sağlamamıştır. Bu nedenle, optimal yineleme sayısı 200 olarak belirlenmiştir.

Ardından, popülasyon büyüklüğü 50, 100 ve 150 olarak test edilmiş; diğer parametreler sabit tutulmuştur. 50 birey içeren popülasyon ile daha düşük puanlar elde edilirken, 100 ve 150 bireyli popülasyonlarla benzer doğrulukta ve daha yüksek hizalama skorları gözlemlenmiştir. Ancak artan popülasyon büyüklüğüyle birlikte çalışma süresi de uzamıştır. Dolayısıyla, doğruluk ve zaman açısından dengeli bir yapı sunan 100 bireyli popülasyon değeri tercih edilmiştir.

Tablo 2. GA'nın popülasyon büyüklüğü, mutasyon oranı ve yineleme sayısı değiştirildikten sonra hizalama skoru ve çalışma süresi sonuçları (The results of the alignment score and runtime after changing the population size, mutation rate and iteration number of GA)

Parametreler	Veri Seti		PS =50		PS= 100		PS=150	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Popülasyon Boyutu (PS)	trnN-GUU_rps12	AVG	1048,33	1,09	1061,93	2,74	1051,66	4,92
		SD	26,59	0,02	33,69	0,02	30,06	0,05
	rrn4,5_rps12	AVG	1151,93	1,27	1178,93	3,08	1166,33	5,39
		SD	50,09	0,04	48,22	0,08	34,55	0,10
	rrn5_rps12	AVG	2391,26	1,81	2456,46	4,19	2446,20	7,09
		SD	90,45	0,03	56,64	0,04	57,57	0,07
	psbT_pbf1	AVG	1124,66	1,32	1144,66	3,21	1152,06	5,55
		SD	34,21	0,03	26,92	0,03	15,63	0,06
	rpl33_rps18	AVG	23973,93	8,55	24682,26	17,49	24917,00	27,10
		SD	614,23	0,12	476,51	0,27	507,04	0,32
	Opuntia	AVG	128837,53	18,87	133458,80	39,04	135426,00	58,87
		SD	12168,61	0,27	11697,72	0,96	10036,00	1,06
	Veri Seti		MR = 0,01		MR = 0,015		MR = 0,05	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Mutasyon Oranı (MR)	trnN-GUU_rps12	AVG	1017,00	2,65	1042,86	2,72	1049,46	2,75
		SD	21,63	0,03	27,09	0,02	23,41	0,02
	rrn4,5_rps12	AVG	1142,26	3,06	1149,13	3,12	1154,60	3,17
		SD	44,42	0,05	38,43	0,09	49,47	0,07
	rrn5_rps12	AVG	2282,80	4,14	2374,86	4,16	2443,80	4,26
		SD	48,93	0,06	77,57	0,05	75,59	0,06
	psbT_pbf1	AVG	1135,46	3,13	1135,66	3,19	1133,90	3,24
		SD	7,45	0,02	20,77	0,03	27,32	0,03
	rpl33_rps18	AVG	23745,80	17,38	24110,00	17,07	24452,20	17,30
		SD	398,92	0,23	589,15	0,24	523,19	0,32
	Opuntia	AVG	127130,13	39,12	133767,80	38,27	135261,93	38,83
		SD	11450,98	0,68	11505,87	0,66	11315,98	0,66
	Veri Seti		İterasyon = 100		İterasyon = 200		İterasyon = 400	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
İterasyon Sayısı	trnN-GUU_rps12	AVG	1048,13	0,71	1057,00	1,37	1056,70	2,77
		SD	26,37	0,02	19,79	0,03	26,45	0,03
	rrn4,5_rps12	AVG	1140,13	0,83	1143,33	1,59	1168,06	3,13
		SD	33,61	0,02	29,40	0,04	35,29	0,08
	rrn5_rps12	AVG	2341,60	1,14	2390,80	2,17	2436,46	4,20
		SD	57,22	0,02	69,64	0,09	62,67	0,06
	psbT_pbf1	AVG	1140,40	0,85	1141,26	1,61	1144,73	3,22
		SD	21,22	0,02	18,35	0,03	23,43	0,06
	rpl33_rps18	AVG	23854,13	6,18	24466,46	9,97	24525,73	17,69
		SD	417,68	0,07	513,31	0,19	531,58	0,33
	Opuntia	AVG	128103,66	18,89	128101,73	25,42	135427,33	37,80
		SD	14210,90	0,38	8986,48	0,31	10600,38	0,78

Tablo 3. DE algoritmasının popülasyon büyüklüğü, çaprazlama oranı, F-sabiti ve yineleme sayısı değiştirildikten sonra hizalama skoru ve çalışma süresine ilişkin sonuçlar
(The results of the alignment score and runtime after changing the population size, crossover rate, F-constant and iteration number of DE algorithm)

Parametreler	Veri Seti		İterasyon = 100		İterasyon = 200		İterasyon = 400	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
İterasyon Sayısı	trnN-GUU_rps12	AVG	1016,73	3,66	1019,06	7,35	1016,13	14,95
		SD	6,89	0,03	7,22	0,12	8,83	0,18
	rrn4,5_rps12	AVG	1288,60	5,16	1290,20	10,30	1286,20	20,59
		SD	10,13	0,03	10,43	0,03	19,59	0,13
	rrn5_rps12	AVG	2216,73	8,55	2278,80	16,70	2379,06	32,17
		SD	48,39	0,15	85,77	0,26	83,93	0,81
	psbT_pbf1	AVG	1148,06	5,57	1151,00	11,21	1152,26	22,05
		SD	24,69	0,04	28,71	0,11	28,21	0,29
	rpl33_rps18	AVG	29072,73	39,37	29214,86	75,36	29166,86	149,02
		SD	145,65	0,61	166,59	0,95	175,16	2,66
	Opuntia	AVG	175309,73	110,89	177452,60	210,71	177782,33	413,91
		SD	4162,95	2,67	767,65	3,62	168,10	5,47
	Veri Seti		PS = 50		PS=100		PS=150	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Popülasyon Boyutu	trnN-GUU_rps12	AVG	1013,80	3,71	1019,06	7,34	1021,06	11,31
		SD	10,09	0,02	7,22	0,12	8,14	0,15
	rrn4,5_rps12	AVG	1021,06	11,31	1290,20	10,30	1289,60	15,78
		SD	8,14	0,15	10,43	0,02	8,28	0,12
	rrn5_rps12	AVG	2270,20	8,23	2278,80	16,70	2295,00	24,52
		SD	72,26	0,16	85,77	0,26	66,26	0,49
	psbT_pbf1	AVG	1123,80	5,49	1151,46	11,21	1170,40	16,71
		SD	33,67	0,07	28,72	0,11	19,68	0,14
	rpl33_rps18	AVG	28776,93	37,53	29214,86	75,35	29330,13	114,09
		SD	249,29	0,52	166,59	0,95	114,44	1,32
	Opuntia	AVG	177427,53	105,42	177452,60	210,71	177471,20	322,61
		SD	672,19	1,62	767,65	3,62	414,32	15,78
	Veri Seti		F =0,3		F = 0,5		F =0,8	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
F Sabiti	trnN-GUU_rps12	AVG	1018,86	7,47	1020,33	7,49	1020,13	7,45
		SD	8,32	0,04	6,26	0,08	8,32	0,09
	rrn4,5_rps12	AVG	1286,53	10,29	1288,60	10,34	1289,66	10,24
		SD	10,67	0,05	13,65	0,11	10,24	0,05
	rrn5_rps12	AVG	2280,20	16,57	2291,60	16,75	2322,66	16,87
		SD	89,88	0,29	72,85	0,33	123,44	0,24
	psbT_pbf1	AVG	1158,86	11,31	1149,66	11,38	1153,66	11,38
		SD	18,70	0,12	20,41	0,07	23,93	0,15
	rpl33_rps18	AVG	29185,80	75,74	29193,53	76,93	29211,20	77,80
		SD	105,60	1,01	165,92	2,34	129,66	1,01
	Opuntia	AVG	177464,06	208,61	177320,93	208,75	177576,93	205,22
		SD	705,51	4,73	779,66	3,50	345,99	2,21
	Veri Seti		CR = 0,3		CR = 0,5		CR = 0,8	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Çaprazlama Oranı	trnN-GUU_rps12	AVG	1035,20	7,31	1031,06	7,33	1018,00	7,29
		SD	9,46	0,04	10,90	0,10	7,24	0,14
	rrn4,5_rps12	AVG	1243,06	10,15	1268,60	10,21	1280,00	10,07
		SD	37,79	0,08	32,64	0,12	12,76	0,08
	rrn5_rps12	AVG	2198,33	17,25	2187,40	16,82	2279,40	16,14
		SD	25,42	0,16	46,73	0,15	78,83	0,36
	psbT_pbf1	AVG	1162,33	10,83	1155,46	10,71	1158,20	10,84
		SD	25,95	0,07	25,19	0,08	33,44	0,09
	rpl33_rps18	AVG	26880,66	78,69	29051,40	78,30	29211,20	77,80
		SD	247,85	0,50	106,74	1,22	129,66	1,01
	Opuntia	AVG	176990,53	207,50	176776,46	209,21	177576,93	205,22
		SD	286,89	2,41	1249,60	2,19	345,99	2,21

F-sabitinin değerlendirilmesinde, 0,3, 0,5 ve 0,8 değerleri test edilmiş; en iyi hizalama skorları 0,8 ile elde edilmiştir. Farklı F değerleri arasında çalışma süresi açısından anlamlı bir fark gözlenmemiştir. Bu nedenle F-sabiti, 0,8 olarak belirlenmiştir.

Son olarak, çaprazlama oranı 0,3, 0,5 ve 0,9 olarak test edilmiştir. 0,9 değeri ile en yüksek hizalama skorları elde edilmiş, çalışma süresi ise önemli ölçüde etkilenmemiştir. Tüm bu değerlendirmeler sonucunda, DE algoritması için önerilen parametre değerleri: iterasyon sayısı = 200, popülasyon büyüklüğü=100, F-sabiti=0,8 ve çaprazlama oranı=0,9 olarak belirlenmiştir. İlgili tüm deneysel sonuçlar Tablo 3'te sunulmuştur.

GABT algoritmasının bir bileşeni olan GA için, daha önce belirlenen optimal parametreler kullanılmıştır. BT için ise başlangıç sıcaklığı ve soğuma hızı parametrelerinin etkilerini değerlendirmek amacıyla her veri seti, her parametre değeriyle 30 kez çalıştırılmış; elde edilen hizalama skorları ve çalışma sürelerinin ortalama ve standart sapmaları hesaplanarak analiz gerçekleştirilmiştir.

BT algoritmasının başlangıç sıcaklık değerini belirlemek üzere, soğuma oranı 0,005 ve son sıcaklık 1 olarak sabitlenmiş; başlangıç sıcaklığı 50.000, 100.000 ve 200.000 olarak test edilmiştir. 100.000 değeri, tüm veri setleri için olmasa da genellikle daha iyi hizalama skorları üretmiş; başlangıç sıcaklığı arttıkça çalışma süresinin de anlamlı biçimde uzadığı gözlemlenmiştir. Bu değerlendirmeler doğrultusunda, başlangıç sıcaklığı 100.000 olarak belirlenmiştir.

Soğuma oranının belirlenmesinde ise başlangıç sıcaklığı 100.000 ve son sıcaklık 1 sabit tutularak oranlar 0,001, 0,003 ve 0,005 olarak denenmiştir. En iyi hizalama skorları 0,001 oranında elde edilmiş olsa da çalışma süresi belirgin şekilde artmıştır. Bu nedenle, doğruluk ve süre arasında denge sağlamak amacıyla soğuma oranı 0,003 olarak tercih edilmiştir.

Sonuç olarak, ÇDH probleminin çözümü için önerilen GABT algoritmasının parametreleri: başlangıç sıcaklığı 100.000, soğuma oranı 0,003 ve son sıcaklık 1 olarak belirlenmiştir. GA bileşeni için daha önce optimize edilen parametreler kullanılmıştır. İlgili tüm deneysel sonuçlar Tablo 4'te sunulmuştur.

Tablo 4. GABT algoritmasının başlangıç sıcaklığı/soğutma oranı değiştirilerek elde edilen hizalama skoru ve çalışma süresi sonuçları (ortalama ve standart sapma ile)
(The results of the alignment score and runtime (with average and standard deviation) by changing the initial temperature/cooling rate of the GASA algorithm)

Parametreler	Veri Seti		Başlangıç Sıcaklığı=50.000		Başlangıç Sıcaklığı=100.000		Başlangıç Sıcaklığı=200.000	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Başlangıç Sıcaklığı	trnN-GUU_rps12	AVG	1057,93	3,35	1060,40	3,92	1059,40	4,52
		SD	30,00	0,07	25,02	0,06	25,41	0,09
	rrn4,5_rps12	AVG	1,17	4,09	1178,73	4,76	1189,40	5,53
		SD	40,60	0,15	40,07	0,16	35,65	0,15
	rrn5_rps12	AVG	2414,93	6,73	2409,06	7,81	2390,80	9,03
		SD	73,42	0,11	67,19	0,11	79,68	0,13
	psbT_pbf1	AVG	1134,00	4,26	1139,73	4,83	1145,26	5,64
		SD	36,28	0,05	28,61	0,09	17,40	0,03
	psl33_rps18	AVG	23820,33	41,46	23928,06	48,83	23672,20	56,07
		SD	364,05	0,73	554,91	0,94	474,33	2,24
	Opuntia	AVG	129107,53	69,07	128942,86	79,72	128590,73	96,44
		SD	10021,02	0,73	10674,32	0,49	11959,41	6,86
	Veri Seti		Soğuma Oranı=0,001		Soğuma Oranı=0,003		Soğuma Oranı=0,005	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Soğuma Oranı	trnN-GUU_rps12	AVG	1072,00	18,45	1058,93	5,98	1056,40	3,89
		SD	25,94	0,30	30,80	0,10	26,37	0,23
	rrn4.5_rps12	AVG	1214,33	22,91	1206,33	7,55	1178,73	4,76
		SD	46,50	0,57	55,10	0,28	40,07	0,16
	rrn5_rps12	AVG	2460,06	37,23	2421,06	12,80	2409,06	7,81
		SD	87,69	0,66	81,53	0,19	67,19	0,11
	psbT_pbf1	AVG	1152,60	23,85	1139,33	7,98	1139,73	4,83
		SD	36,02	0,28	27,28	0,15	28,61	0,09
	psl33_rps18	AVG	24316,60	227,92	23875,73	77,49	23928,06	48,83
		SD	551,78	5,51	557,80	1,19	554,91	0,94
	Opuntia	AVG	136675,66	361,82	131969,26	126,50	128942,86	79,72
		SD	12972,69	4,95	10934,93	1,47	10674,32	0,49

DEBT algoritmasının bir bileşeni olan DE için, daha önce belirlenen optimal parametreler kullanılmıştır. BT için ise başlangıç sıcaklığı ve soğuma hızı parametrelerinin etkilerini analiz etmek amacıyla, her veri seti üzerinde her parametre değeriyle algoritma 30 kez çalıştırılmış; hizalama skorları ve çalışma sürelerinin ortalama ve standart sapmaları hesaplanarak performans değerlendirmesi yapılmıştır.

Başlangıç sıcaklığını belirlemek üzere soğuma oranı 0,005 ve son sıcaklık 1 olarak sabitlenmiş; başlangıç sıcaklığı 50.000, 100.000 ve 200.000 olarak test edilmiştir. Bu testlerde 100.000 değeri, tüm veri setlerinde olmasa da genellikle en yüksek hizalama skorlarını sağlamış; sıcaklık arttıkça çalışma süresinde anlamlı bir artış gözlemlenmiştir. Bu nedenle başlangıç sıcaklığı 100.000 olarak belirlenmiştir.

Soğuma oranının belirlenmesinde ise başlangıç sıcaklığı 100.000 ve son sıcaklık 1 sabit tutularak oranlar 0,001, 0,003 ve 0,005 olarak denenmiştir. En yüksek doğruluk 0,001 ile elde edilmiş olsa da bu durumda çalışma süresi önemli ölçüde artmıştır. Doğruluk ve verimlilik arasında denge sağlamak amacıyla soğuma oranı 0,003 olarak tercih edilmiştir.

Sonuç olarak, ÇDH probleminin çözümünde kullanılan DEBT algoritması için BT bileşenine ait parametreler: başlangıç sıcaklığı 100.000, soğuma oranı 0,003 ve son sıcaklık 1 olarak belirlenmiştir. DE bileşeni için ise daha önce optimize edilen parametreler esas alınmıştır. İlgili tüm deneysel sonuçlar Tablo 5'te sunulmuştur.

5.2. Önerilen Algoritmaların Sonuçları
(The Results of The Proposed Algorithms)

Bu çalışmada önerilen GA, GABT, DE ve DEBT algoritmalarının performansları karşılaştırılmıştır. Önerilen algoritmaların performansları hizalama skoru, çalışma süresi ve bu metriklerin standart sapmaları açısından karşılaştırılmıştır. Önerilen algoritmalar hizalama skoru açısından karşılaştırıldığında, GA "rrn5" veri setinde, DEBT algoritması "trnN-GUU" ve "rrn4.5" veri setlerinde DE algoritması "psbT", rpl33" ve "Opuntia" veri setlerinde diğer algoritmalardan daha iyidir. GA, DE ve DEBT algoritmalarında hizalama skoru açısından GABT algoritmasına göre daha iyi sonuçlar elde edilmiştir. GA ve GABT algoritmalarının standart sapmaları, DE ve DEBT algoritmalarının standart sapmalarından daha yüksektir. DE algoritması hizalama skorunun en düşük standart sapmasına sahiptir. Önerilen algoritmaların hizalama skorları ve hizalama skorlarının standart sapmaları Tablo 6'da gösterilmiştir.

Tablo 5.DEBT algoritmasının başlangıç sıcaklığı/soğutma oranı değiştirilerek hizalama skoru ve çalışma süresi sonuçları (ortalama ve standart sapma ile) (The results of the alignment score and runtime (with average and standard deviation) by changing the initial temperature/cooling rate of the DESA algorithm)

Parametreler	Veri Seti		Başlangıç Sıcaklığı=50.000		Başlangıç Sıcaklığı=100.000		Başlangıç Sıcaklığı=200.000	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Başlangıç Sıcaklığı	trnN-GUU_rps12	AVG	1053,86	4,80	1064,46	5,32	1062,20	5,89
		SD	25,00	0,07	20,99	0,03	22,51	0,03
	rrn4,5_rps12	AVG	1269,13	6,09	1254,53	6,76	1256,00	7,60
		SD	36,63	0,05	40,91	0,06	41,04	0,05
	rrn5_rps12	AVG	2235,86	9,82	2278,00	10,87	2232,06	11,99
		SD	114,62	0,14	89,82	0,14	96,48	0,15
	psbT_pbf1	AVG	1128,06	6,44	1127,86	7,18	1124,66	7,98
		SD	39,40	0,04	31,68	0,06	28,94	0,10
	psl33_rps18	AVG	29048,86	54,23	29034,93	62,97	29013,66	66,43
		SD	159,39	0,50	214,23	0,23	205,95	0,63
	Opuntia	AVG	176115,86	112,03	176474,86	122,22	175639,40	131,58
		SD	3195,53	1,26	2264,20	0,86	3521,28	0,83
	Veri Seti		Soğuma Oranı=0,001		Soğuma Oranı=0,003		Soğuma Oranı=0,005	
			Skor	Çalışma Süresi	Skor	Çalışma Süresi	Skor	Çalışma Süresi
Soğuma Oranı	trnN-GUU_rps12	AVG	1063,33	19,37	1066,40	7,51	1063,40	5,17
		SD	22,21	0,15	22,15	0,07	21,31	0,05
	rrn4,5_rps12	AVG	1274,13	24,29	1265,13	9,68	1253,26	6,78
		SD	21,68	0,23	41,23	0,07	39,71	0,05
	rrn5_rps12	AVG	2273,20	10,99	2290,40	15,41	2256,86	10,86
		SD	91,61	0,14	98,41	0,17	91,47	0,12
	psbT_pbf1	AVG	1126,00	25,98	1129,13	10,25	1129,93	7,16
		SD	32,81	0,31	24,18	0,14	31,98	0,08
	psl33_rps18	AVG	29320,13	226,43	29130,26	87,62	29034,93	62,97
		SD	148,59	2,26	190,61	0,61	214,23	0,23
	Opuntia	AVG	175931,60	391,02	176884,53	166,87	176474,86	122,22
		SD	3723,52	3,48	1659,69	2,21	2264,20	0,86

Tablo 6. Önerilen algoritmaların hizalama skorları ve çalışma sürelerinin sonuçları
(Results of the alignment scores and runtimes of the proposed algorithms)

Parametreler	Veri Seti		GA	GABT	DE	DEBT
Hizalama Skorları	trnN-GUU_rps12	AVG	1049,46	1058,93	1018,00	1066,40
		SD	23,41	30,80	7,25	22,15
	rrn4,5_rps12	AVG	1154,60	1206,33	1282,00	1265,13
		SD	49,47	55,10	12,76	41,23
	rrn5_rps12	AVG	2443,80	2421,06	2279,40	2290,40
		SD	75,59	81,53	78,83	98,41
	psbT_pbf1	AVG	1133,90	1139,33	1158,20	1129,13
		SD	27,32	27,28	33,44	24,18
	rpl33_rps18	AVG	24452,20	23875,73	29211,20	29130,26
		SD	523,19	557,80	129,66	190,61
	Opuntia	AVG	135261,93	131969,26	177576,93	176884,53
		SD	11315,98	10934,93	346,00	1659,69
	Veri Seti		GA	GABT	DE	DEBT
Çalışma Süresi	trnN-GUU_rps12	AVG	2,75	5,98	7,30	7,51
		SD	0,02	0,10	0,15	0,07
	rrn4,5_rps12	AVG	3,17	7,55	10,07	9,68
		SD	0,07	0,28	0,08	0,07
	rrn5_rps12	AVG	4,26	12,80	16,14	15,41
		SD	0,06	0,19	0,36	0,17
	psbT_pbf1	AVG	3,24	7,98	10,84	10,25
		SD	0,03	0,15	0,09	0,14
	rpl33_rps18	AVG	17,30	77,49	77,80	87,62
		SD	0,32	1,19	1,01	0,61
	Opuntia	AVG	38,83	126,50	205,22	166,87
		SD	0,66	1,47	2,21	2,21

Tablo 7.Önerilen algoritmalar ve ClustalW'nin hizamla skoru açısından karşılaştırılması
(Comparison of the proposed algorithms and ClustalW in terms of alignment score)

Veri Seti		Hizalama Skoru				
		ClustalW	GA	GABT	DE	DEBT
trnN-GUU_rps12	AVG	844	1049,46	1058,93	1018,00	1066,40
	SD	0,0	23,41	30,80	7,25	22,15
rrn4,5_rps12	AVG	1194	1154,60	1206,33	1282,00	1265,13
	SD	0,0	49,47	55,10	12,76	41,23
rrn5_rps12	AVG	2044	2443,80	2421,06	2279,40	2290,40
	SD	0,0	75,59	81,53	78,83	98,41
psbT_pbf1	AVG	1046	1133,90	1139,33	1158,20	1129,13
	SD	0,0	27,32	27,28	33,44	24,18
rpl33_rps18	AVG	29780	24452,20	23875,73	29211,20	29130,26
	SD	0,0	523,19	557,80	129,66	190,61
Opuntia	AVG	194472	135261,93	131969,26	177576,93	176884,53
	SD	0,0	11315,98	10934,93	346,00	1659,69

Önerilen algoritmalar hizalama skoru açısından karşılaştırıldığında; GA en iyi çalışma süresine ulaşmıştır. DE ve DEBT algoritmaları en uzun çalışma zamanlarına, GA ve GABT algoritmaları ise daha iyi çalışma zamanlarına sahiptir. Çalışma süresinin standart sapması her veri seti için farklı olmuştur. Tablo 6'da önerilen algoritmaların çalışma süresi ve çalışma süresinin standart sapması verilmiştir.

5.3. Önerilen Algoritmaların Sonuçlarının ClustalW ile Karşılaştırılması
(Comparison of the Results of the Proposed Algorithms with the Clustal)

Bu çalışmada önerilen algoritmalar, ÇDH problemini çözmek için sıklıkla kullanılan Clustal algoritması ile karşılaştırılmıştır.

Tablo 8. Önerilen algoritmalar ve ClustalW'nin çalışma süresi açısından karşılaştırılması
(Comparison of the proposed algorithms and ClustalW in terms of runtime)

Veri Seti		Çalışma Süresi (sn)				
		ClustalW	GA	GABT	DE	DEBT
trnN-GUU_rps12	AVG	0,01	2,75	5,98	7,30	7,51
	SD	0,00	0,02	0,10	0,15	0,07
rrn4,5_rps12	AVG	0,01	3,17	7,55	10,07	9,68
	SD	0,00	0,07	0,28	0,08	0,07
rrn5_rps12	AVG	0,01	4,26	12,80	16,14	15,41
	SD	0,00	0,06	0,19	0,36	0,17
psbT_pbf1	AVG	0,01	3,24	7,98	10,84	10,25
	SD	0,00	0,03	0,15	0,09	0,14
rpl33_rps18	AVG	0,03	17,30	77,49	77,80	87,62
	SD	0,00	0,32	1,19	1,01	0,61
Opuntia	AVG	0,29	38,83	126,50	205,22	166,87
	SD	0,00	0,66	1,47	2,21	2,21

Tablo 9. Önerilen algoritmalar ve ClustalW'nin verimlilik (birim zamanda elde edilen skor başarısı) açısından karşılaştırılması
(Comparison of the proposed algorithms and ClsutalW in terms of efficiency (score success achieved per unit time))

Veri Seti	Verimlilik (Hizalama Skoru/Süre(sn))				
	ClustalW	GA	GABT	DE	DEBT
trnN-GUU_rps12	140666,67	381,62	177,11	139,49	142,09
rrn4,5_rps12	199000,00	364,34	159,72	127,33	130,72
rrn5_rps12	292000,00	574,34	189,17	141,19	148,68
psbT_pbf1	174333,33	349,97	142,77	106,84	110,14
rpl33_rps18	1075090,25	1413,50	308,11	375,45	332,45
Opuntia	663726,96	3483,35	1043,24	865,28	1060,04

Performans karşılaştırması, hizalama skoru ve çalışma süresi dikkate alınarak yapılmıştır. Önerilen GA, GABT, DE ve DEBT algoritmalarıyla karşılaştırıldığında, ClustalW, hizalama skoru açısından "rpl33" ve "Opuntia" veri kümelerinde daha iyi hizalama skoru elde etmiştir. ClustalW'nin yüksek hizalama skoru elde ettiği veri kümeleri, diğer veri kümelerine göre yüksek dizi benzerliğine, daha yüksek dizi numarasına ve dizi uzunluğuna sahip veri kümeleridir. ClustalW, hizalama skorunun standart sapması açısından açgözlü yaklaşıma dayandığından, hizalama skorunun standart sapması sıfırdır. ClustalW ile önerilen yöntemlerin elde ettiği hizalama skorları ve bu skorlara ait standart sapma değerleri Tablo 7'de sunulmuştur. Çalışma süresi açısından değerlendirildiğinde ise ClustalW, önerilen algoritmalara kıyasla daha kısa sürelerde sonuç üretmiştir. Algoritmaların ortalama çalışma süreleri ile bu sürelerin standart sapmaları Tablo 8'de verilmiştir.

Tablo 9'da sunulan verimlilik sonuçlarına göre, ClustalW genel olarak önerilen GA, GABT, DE ve DEBT algoritmalarına kıyasla daha yüksek verimlilik değerleri elde etmiştir. Özellikle kısa sürelerde sonuç üretebilmesi ve sabit bir yapı ile çalışması, ClustalW'nin bu alandaki avantajını ortaya koymaktadır. Bununla birlikte, önerilen sezgisel algoritmaların verimlilik değerleri görece düşük olmasına rağmen, bu durum algoritmaların doğası gereği daha geniş çözüm uzayını taramaları ve potansiyel olarak daha kaliteli hizalamalar üretme hedefinden kaynaklanmaktadır. Özellikle büyük ve karmaşık veri setlerinde, sezgisel yöntemlerin doğruluk odaklı çalışmaları ön plana çıkabilmektedir. Bu nedenle önerilen yöntemler, hız öncelikli uygulamalara göre değil, doğruluk ve esneklik gerektiren senaryolara yönelik güçlü alternatifler olarak değerlendirilebilir. Dolayısıyla bu çalışmada, geleneksel algoritmaların hız avantajına karşılık, sezgisel yaklaşımların farklı veri türlerine uyarlanabilirliği ve geliştirilebilir yapıları ortaya konmuştur.

6. Sonuçlar (Conclusions)

Bu çalışmada, ÇDH problemini çözmek için GA, BT ve DE algoritmaları seçilmiştir. GABT algoritması, GA ve BT algoritmalarının birleştirilmesiyle, DEBT algoritması ise DE ve BT algoritmalarının birleştirilmesiyle önerilmiştir. Önerilen algoritmalar (GA, BT, GABT, DEBT) altı farklı veri kümesi için değerlendirilmiştir. Çalışmanın amacı, yüksek hizalama skorlarına ve zaman açısından verimli bir modele sahip algoritmalar geliştirmektir. GA, DE ve DEBT algoritmaları farklı veri kümeleri için yüksek hizalama skorlarına ulaşmıştır. GABT algoritmasının hizalama skoru, GA, DE ve DEBT algoritmalarına kıyasla daha düşüktür. Çalışma süresi açısından en iyi sonuç GA algoritması ile elde edilmiştir. DE ve DEBT'nin çalışma süreleri ise GA ve GABT'ye göre daha uzundur. ÇDH problemini çözmek için önerilen algoritmalar, hizalama skoru ve çalışma süresine göre tercih edilebilir.

Önerilen algoritmalar ClustalW ile karşılaştırılmıştır. ClustalW, çalışma süresi açısından en iyi sonucu elde etmiştir. Ancak hizalama skoru açısından, önerilen algoritmalar ClustalW'ye kıyasla dört veri kümesi (trnN-GUU_rps12, rrn4.5_rps12, rrn5_rps12, psbT_pbf1) için daha iyi sonuçlar elde etmiştir. Verimlilik (hizalama skoru/süre) açısından değerlendirildiğinde ise ClustalW, genel olarak daha yüksek verimlilik sunmuştur. Bununla birlikte, önerilen algoritmaların bazı veri kümelerinde daha yüksek doğruluk sağlaması, verimlilikten ziyade doğruluk odaklı senaryolarda tercih edilebilirliğini artırmaktadır.

Teorik zaman karmaşıklığı analizine göre, GA ve DE algoritmaları için karmaşıklık $O(G \times P \times N^2 \times L^2)$, GABT ve DEBT algoritmaları için ise $O((G \times P + T) \times N^2 \times L^2)$ olarak belirlenmiştir. ClustalW'nin

teorik karmaşıklığı ise O ($N^2 \times L^2$) düzeyindedir. Bu analizler, algoritmaların büyük ölçekli veri kümelerinde nasıl performans göstereceğini öngörmek açısından önemlidir.

Gelecek çalışmalarda, önerilen algoritmaların büyük ölçekli veri kümelerinde daha etkin çalışabilmesi için paralel işlemeye dayalı yapılar (örneğin GPU hızlandırma) ve veri ön işleme stratejileri (örneğin, dizi kümeleme, boşluk optimizasyonu) entegre edilebilir. Ayrıca başlangıç popülasyonunun iyileştirilmesi ve parametre seçim stratejilerinin optimize edilmesiyle hem hizalama doğruluğu hem de çalışma süresi açısından daha verimli sonuçlar elde edilebilir.

Teşekkür (Acknowledgement)

Bu çalışma, Ege Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından "Cicer ve Lens Türlerinin Kloroplast DNA Dizilerinin Yeni Nesil Dizileme ile Dizilenmesi ve Genom Organizasyonlarının Belirlenerek Karşılaştırmalı Genom Analizlerinin Yapılması" başlıklı proje kapsamında "FO A-2020-20981" proje numarası ile maddi olarak desteklenmiştir.

Kaynaklar (References)

1. Cohen, J., Bioinformatics-an introduction for computer scientists, ACM Comput. Surv. ,36 (2), 122-158, 2004.
2. Luscombe N.M., Greenbaum D., Gerstein M., What is bioinformatics? An introduction and overview, Yearbook of medical informatics**,** 10 (1), 83-100, 2001.
3. Karcioglu A.A., Bulut H., Improving hash-q exact string-matching algorithm with perfect hashing for DNA sequences, Computers in Biology and Medicine,131, 104292, 2021.
4. Karcıoğlu A.A, Bulut H., Q-gram hash comparison based multiple exact string matching algorithm for DNA sequences, Journal of the Faculty of Engineering and Architecture of Gazi University**,** 38 (2), 2023.
5. Karcioglu A.A., Bulut H., The WM-q multiple exact string matching algorithm for DNA sequences, Computers in Biology and Medicine, 136, 104656, 2021.
6. Karcioglu A.A., Bulut H., q-frame hash comparison based exact string matching algorithms for DNA sequences, Concurrency and Computation: Practice and Experience, 34 (9), 2022.
7. Botta M., Negro G., Multiple sequence alignment with genetic algorithms, Computational Intelligence Methods for Bioinformatics and Biostatistics, Springer, Berlin, Germany, 206-214, 2009.
8. Haque W., Aravind A., Reddy B., Pairwise sequence alignment algorithms: a survey, Information Science, Technology and Applications Conference, 96-103, 2009.
9. Lee Z.J., Su S.F., Chuang C.C., Liu K.H., Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment, Applied Soft Computing, 8 (1), 55-78, 2008.
10. Zhu X., Li K., Salah A., A data parallel strategy for aligning multiple biological sequences on multi-core computers, Computers in biology and medicine, 43 (4), 350-361, 2013.
11. Liu X., Yang X., Wang C., Yao Y., Dai Q., Number of distinct sequence alignments with k-match and match sections, Computers in biology and medicine, 63, 287-292, 2015.
12. Naorem L. D., Sharma N., Raghava G. P., A web server for predicting and scanning of IL-5 inducing peptides using alignment-free and alignment-based method,Computers in Biology and Medicine, 158, 106864, 2023.
13. Pais F.S.M., Ruy P.D.C., Oliveira G., Coimbra R.S., Assessing the efficiency of multiple sequence alignment programs, Algorithms for Molecular Biology, 9 (1), 4, 2014.
14. Aktan M.N., Bulut H., Metaheuristic task scheduling algorithms for cloud computing environments, Concurrency and Computation: Practice and Experience, 34 (9), 2022.
15. Çavga S. H., Performance of neural networks and heuristic models for disease prediction from liver enzymes: Application to biochemistry device output, Journal of the Faculty of Engineering and Architecture of Gazi University, 39 (4), 2263-2270, 2024
16. Kaya F., Conker Ç., Hybrid input shaper design and genetic algorithm-based multi-objective optimization for elimination of residual vibrations at specific frequencies in flexible systems, Journal of the Faculty of Engineering and Architecture of Gazi University (Advanced Online Publication), 2541-2552, 2025.
17. Arıkan M., Hybrid simulated annealing–tabu search algorithms for solving U-shaped type-2 assembly line balancing problems with workload smoothing objective, Journal of the Faculty of Engineering and Architecture of Gazi University, 39 (3), 1457–1472, 2024.
18. Yalcin A., Deliktas D., Genetic algorithm based on weighted goal programming for doctor rostering problem, Journal of the Faculty of Engıneerıng and Archıtecture of Gazı Unıversıty, 39 (4), 2567-2585, 2024.
19. Erdirik H., Karcıoğlu A. A., Tanyolaç B., Bulut H., Meta-Sezgisel Tabanlı Clustal-SA Algoritmasını Kullanarak DNA Sekanslarında Çoklu Dizi Hizalama, Journal of the Institute of Science and Technology, 14 (2), 544-562, 2024.
20. Bucak, İ. Ö., Uslan, V., Sequence alignment from the perspective of stochastic optimization: a Survey, Turkish Journal of Electrical Engineering and Computer Sciences, 19 (1), 157-173, 2011.
21. Nalbantoğlu O.U., Dynamic Programming, Multiple Sequence Alignment Methods, Methods in Molecular Biology, 1079, Russell D.J., Springer, 3–27, 2014.
22. Karadimitriou K., Kraft D.H., Genetic algorithms and the multiple sequence alignment problem in biology, Second Annual Molecular Biology and Biotechnology Conference, Baton Rouge, 1–7, 1996.
23. Onwubolu G., Davendra D., Scheduling flow shops using differential evolution algorithm, European Journal of Operational Research, 171 (2), 674-692, 2006.
24. Major Differences. Difference between Global and Local Sequence Alignment. https://www.majordifferences.com/2016/05/difference-between-global-and-local.html. Erişim tarihi: 28.12.2024.
25. Likic V., The Needleman–Wunsch Algorithm for Sequence Alignment, Lecture Notes, 7th Melbourne Bioinformatics Course, Molecular Science and Biotechnology Institute, University of Melbourne, 1–46, 2008.
26. Needleman S. B., Wunsch C. D., A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology, 48 (3), 443–453, 1970.
27. Diamantis S., Charissi A., Comparison of Multiple Sequence Alignment Programs, MSc Bioinformatics Report, National and Kapodistrian University of Athens, 2005.
28. Smith T.F., Waterman M.S., Identification of common molecular subsequences, Journal of Molecular Biology, 147 (1), 195–197, 1981.
29. Thompson J.D., Higgins D.G., Gibson T.J., CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Research, 22 (22), 4673–4680, 1994.
30. Doğan H., Otu H.H., Objective Functions, Multiple Sequence Alignment Methods, Methods in Molecular Biology, Cilt 1079, Editör: Russell D.J., Springer, 45–58, 2014.
31. Sastry K., Goldberg D., Kendall G., Genetic Algorithms, Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Editör: Burke E.K., Kendall G., Springer, 97–125, 2005.
32. Mirjalili S., Evolutionary Algorithms and Neural Networks, Studies in Computational Intelligence, Springer, Berlin, 780, 2019.
33. Karaboğa D., Ökdem S., A simple and global optimization algorithm for engineering problems: differential evolution algorithm, Turkish Journal of Electrical Engineering and Computer Sciences, 12 (1), 53–60, 2004.
34. Aarts E.H., Van Laarhoven P.J., Simulated annealing: a pedestrian review of the theory and some applications, Pattern Recognition Theory and Applications, Springer, Berlin, 179–192, 1987.
35. Edgar R.C., Batzoglou S., Multiple sequence alignment, Current Opinion in Structural Biology, 16 (3), 368–373, 2006.