T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ÇOKLU DİZİ HİZALAMA PROBLEMİNİN YAPAY ARI KOLONİ ALGORİTMASI İLE ÇÖZÜLMESİ

(Yüksek Lisans Tezi)

Hazırlayan
Selçuk ASLAN

Danışman
Yrd. Doç. Dr. Celal ÖZTÜRK

Haziran 2013
KAYSERİ

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Selçuk ASLAN

"Çoklu Dizi Hizalama Probleminin Yapay Arı Koloni Algoritması İle Çözülmesi" adlı yüksek lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Selçuk ASLAN

Danışman

Yrd. Doç. Dr. Celal ÖZTÜRK

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Derviş KARABOĞA

Yrd. Doç. Dr. Celal ÖZTÜRK danışmanlığında **Selçluk ASLAN** tarafından hazırlanan **"Çoklu Dizi Hizalama Probleminin Yapay Arı Koloni Algoritması İle Çözülmesi"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

26 / 06 / 2013

JÜRİ:

Danışman : Yrd. Doç. Dr. Celal ÖZTÜRK .....

Üye : Prof. Dr. Derviş KARABOĞA .....

Üye : Yrd. Doç. Dr. Rıfat KURBAN .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 09/07/2013 tarih ve 2013/30-08 sayılı kararı ile onaylanmıştır.

09 / 07 / 2013

Doç. Dr. Hasan YALÇIN
Enstitü Müdürü V.

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde bilgi ve tecrübelerini esirgemeyen Bilgisayar Mühendisliği Bölüm Başkanı Prof. Dr. Derviş KARABOĞA'ya ve danışman hocam Yrd. Doç. Dr. Celal ÖZTÜRK'e sonsuz teşekkürlerimi sunarım.

NVIDIA Profesyonel Ürünler Grubu'ndan sorumlu Türkiye Müdürü Oğuzhan OĞUZ'a CUDA destekli ekran kartlarının kullanımı için desteklerinden dolayı teşekkür ederim.

Yaptıkları kıymetli katkılardan dolayı çalışma arkadaşlarım Arş. Gör. Turgay BATBAT Arş. Gör. Muhammed YÜCE'ye ve kıymetli aileme teşekkürlerimi sunarım.

ÇOKLU DİZİ HİZALAMA PROBLEMİNİN YAPAY ARI KOLONİ ALGORİTMASI İLE ÇÖZÜLMESİ

Selçuk ASLAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Haziran 2013

Danışman: Yrd. Doç. Dr. Celal ÖZTÜRK

KISA ÖZET

Çoklu dizi hizalama, biyolojik sekansların yorumlanabilmesi için en sık başvurulan analiz yöntemidir. Makro moleküllerin yapı ve fonksiyon tahmininden filogenetik ağaçların oluşturulmasına, polimeraz zincir reaksiyon birincil tasarımından hücre simülasyonuna kadar Biyoinformatik'in hemen hemen tüm problemlerinde çoklu dizi hizalamalarına ihtiyaç duyulmaktadır.

Protein, DNA ya da RNA makro moleküllerine ait sekans bileşenlerinin hizalanması için kullanılan yaklaşımları genel olarak dinamik programlama algoritmaları ve sezgisel algoritmalar olarak iki gruba ayırmak mümkündür. İkiden fazla olan sekans sayısı ve sekans uzunlukları dikkate alındığında dinamik programlama bazı hesaplama sınırlarını beraberinde getirmektedir. Hem dinamik programlama sınırlamaları dikkate alındığında hem de çoklu dizi hizalama işlemi ayrık optimizasyon problemi olarak değerlendirildiğinde sezgisel algoritmaların bu problemlerin çözümünde kullanılabilirliği daha önemli hale gelmiştir. Bu tez bağlamında, arıların yiyecek arama davranışlarındaki zekiliği modelleyen Yapay Arı Koloni Algoritması kullanılarak çoklu dizi hizalama problemine ait optimum çözümler bulunmaya çalışılmıştır. Yapay Arı Koloni Algoritması'nın sağlam, esnek yapısı ve algoritmanın işçi, gözcü, izci arı fazlarının çoklu dizi hizalama problemi için önerilen yeni modelleri sayesinde literatürle karşılaştırılabilir sonuçlar alınmıştır. Aynı zamanda yapılan çalışmalar CUDA platformu ile birleştirilerek paralel çalışma performansları da araştırılmıştır.

Anahtar Kelimeler: Biyoinformatik, Dizi Hizalama, Ayrık Optimizasyon, Yapay Arı Koloni Algoritması, CUDA Programlama

SOLVING MULTIPLE SEQUENCE ALIGNMENT PROBLEM WITH ARTIFICIAL BEE COLONY ALGORITHM

Selçuk ASLAN

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, June 2013

Supervisor: Assist. Prof. Dr. Celal ÖZTÜRK

ABSTRACT

Multiple Sequence Alignment is one of the most commonly used analysis tool for interpreting and identifying biological sequences. Almost all problems in Bioinformatics which can be extended from structure and function prediction to phylogenetic tree construction and polymerase chain reaction primer design to cell simulation needs methods of Multiple Sequence Alignment.

The approaches chosen for aligning residues of the protein, DNA or RNA macro molecules roughly fall into two categories named dynamic programming algorithms and heuristic algorithms. When considering number of the aligned sequences are more than two and length of the aligned sequences dynamic programming algorithms have disadvantages and computational limitations. Considering main drawbacks of dynamic programming algorithms and accepting the multiple sequence alignment process as a combinatorial optimization problem, heuristic algorithms are more eligible than other greedy approaches in solving these problems. In the context of this thesis, Artificial Bee Colony Algorithm which is inspired by the intelligent foraging behaviors of honey bees is used to find the optimal solutions of multiple sequence alignments. The robust, flexible structure of Artificial Bee Colony Algorithm and newly proposed model for employed, onlooker, scout bee phases have given comparable results with similar works in the literature. In addition, performance of the parallelized methods has been analyzed by incorporating the proposed models and CUDA platform.

Keywords: Bioinformatics, Multiple Sequence Alignment Problem, Combinatorial Optimization, Artificial Bee Colony Algorithm, CUDA Programming

İÇİNDEKİLER

ÇOKLU DİZİ HİZALAMA PROBLEMİNİN YAPAY ARI KOLONİ ALGORİTMASI İLE ÇÖZÜLMESİ

3. BÖLÜM

ÇOKLU DİZİ HİZALAMA PROBLEMİNİN TEMEL ÖZELLİKLERİ

4. BÖLÜM

YAPAY ARI KOLONİ ALGORİTMASI VE ÇOKLU DİZİ HİZALAMA

5. BÖLÜM

UYGULAMA

6. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

EKLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

1. BÖLÜM

GİRİŞ

Temel bilim dalları, disiplinleri arasındaki etkileşim geçmişten günümüze ara disiplinlerin, bilim dallarının doğmasına sebep olmuştur. Canlı organizmalarda hayati önem taşıyan nükleik asitlerin, protein ve enzim işlevlerinin tam olarak açıklanabilmesi amacıyla kimyasal reaksiyonların tanımlarından ve fiziksel olaylardan referans alan biyokimya ve biyofizik yeni disiplinler olarak bilim tarihde yerini almıştır.

Teknolojik gelişmelerin laboratuvar ve deney ortamlarında kullanımı; tekrarlanabilirlik, yüksek hassasiyet ve en önemlisi deney ya da gözlemlerden elde edilecek olası veri yoğunluğu artışını beraberinde getirmiştir. Biyologlar tarafından ulaşılan oldukça büyük boyutlu bu verilerin toplanması, depolanması, tekrar erişimi ve daha önemlisi bu bilgilerden faydalanılarak çıkarım ve genellemeler yapılabilmesi amacıyla bilgisayar bilimlerinden faydalanılması Biyoinformatik (Bioinformatics) ve Hesaplamalı Biyoloji (Computational Biology) olarak isimlendirilmiş yeni iki ara disiplin tanımının yapılmasını zorunlu kılmıştır [1, 4-6].

Biyoinformatik disiplini dahilinde tanımlanmış problemlerin tamamına yakını, canlı organizmaların gerçekleştirdiği temel yaşamsal faliyetlerdeki, bu faliyetler besin arama, eş bulma, göç yollarının tespit edilmesi gibi daha genel konularda gösterilen zeki davranışlar içerisinde değerlendirilebileceği gibi, kalıtsal materyalin aktarılması, hücrenin haberleşme ve sinyalizasyon mekanizmaları gibi mikro boyutlarda da zekiliğin eylemlere yansıdığı durumlarda da gözlenmektedir, zeki davranışların tahmini veya algoritmik olarak ifadesi, DNA ve RNA gibi kalıtsal materyallerin hücrenin canlılık faliyetlerinin sürdürülebilmesi için nasıl işlem gördüğü, kromozomlar üzerinde bir ya da daha fazla protein sentezinden sorumlu gen ile genlerin bulunması ve canlı gruplar arasındaki evrimsel akrabalık tayini gibi konuların içerdiği ayrık türde problemlerdir.

Bu problemlerin doğrudan Simülasyon-Modelleme, Makine Öğrenmesi (Machine Learning), Yapay Sinir Ağları (Artificial Neural Network), Veri Madenciliği (Data Minding) ve en önemlisi Optimizasyon Algoritmaları (Optimization Algorithms) gibi temelde bilgisayar bilminin alt araştırma alanı olan Yapay Zeka (Artificial Intelligence) ile ilişkilendirilebileceğini söyleyebiliriz [4].

1.1 Biyoinformatikte Dizi Hizalama Problemi ve Optimizasyon

Karar parametreleri, karar değişkenleri ya da tasarım parametreleri olarak da isimlendirilen ve problem ile ilgili sınırlama fonksiyonlarını (constraints) sağlayan bilinmeyen parametre seti, değerlerinin en küçük yapılacak (minimizasyon) maliyet fonksiyonu (cost function) veya en büyük yapılacak (maksimizasyon) kâr fonksiyonunda kullanıldığı herhangi bir problem optimizasyon problemi olarak adlandırılır [2, 3, 11].

$x = (x_1, x_2, \dots x_i, \dots x_n)$ n elemanlı tasarım parametrelerini tanımlayan vektörü, x_i ise i. parametreyi göstermek üzere, maliyet fonksiyonu $f(x) = f(x_1, x_2, \dots x_i, \dots x_n)$;

$$h_j(x) = h_j(x_1, x_2, \dots x_i, \dots x_n) = 0, 1 \leq j \leq p \quad 1.1$$

$$g_k(x) = g_k(x_1, x_2, \dots x_i, \dots x_n) = 0, 1 \leq k \leq m \quad 1.2$$

p tane eşitlik sınırlamasına, k tane de eşitsizlik sınırlamasına sahip olabilir [2, 14]. Bazı problemler birden fazla en küçük ya da en büyük yapılacak maliyet fonksiyonları ya da kâr fonksiyonları barındırabilir. Bu tür optimizasyon problemleri, çok amaçlı optimizasyon problemi olarak tanımlanmaktadır [2, 15].

Optimizasyon problemleri, amaç fonksiyonu $f(x)$ ile ilgili herhangi bir sınırlama olup olmamasına göre, amaç fonksiyonu $f(x)$ ve sınırlama fonksiyonlarının lineer olup olmamasına göre gruplara ayrılabilir [2].

Optimizasyon problemlerine diğer bir yaklaşım, optimize edilecek problemin sürekli (continuous) veya ayrık (combinatorial, discrete) olmasına göre iki alt grubun oluşturulması şeklindedir [2, 16]. Tasarım değişkenleri ya da parametrelerinin alacağı değerler sürekli değerler ise optimizasyon problemleri sürekli optimizasyon problemleri olarak tanımlanır [2]. Ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması

sıraya konulması veya seçilmesi durumlarını içeren optimizasyon problemleri ise ayrık optimizasyon problemleri olarak adlandırılır [2, 16].

Ayrık optimizasyon problemlerini, optimal çözümü elde etmek için gerekli hesaplama işlemleri ve ilgilenilen optimizasyon algoritmasına göre modellenebilirlik açısından değerlendirdiğimizde oldukça zor problemler oldukları hemen anlaşılabilir. Problem ile ilgili tasarım parametre sayısı arttıkça uygun sürelerde en kaliteli çözümlerin elde edilebilirliği azaldığından, en kaliteli çözüm veya bu çözüme yakın optimalitede çözümler bulabilmek amacıyla sezgisel yaklaşımların kullanılabilirliği önem kazanmıştır. Sezgisel algoritmalar, sezgisel yaklaşımlara dayanan ve çözüm uzayında optimum çözüme yakınsaması garanti edilemeyen ancak yakınsama özelliğine sahip algoritmalar olarak tanımlanabilir [2, 4, 5].

Yukarıda genel olarak sınırları çizilmiş Biyoinformatik araştırma konuları özellikleri nedeniyle, yine Biyoinformatikte Dizi Hizalama Problemi şeklinde belirtilen ayrık türde bir optimizasyon problemin çözümüne dayandırılır [4-6]. Hizalama kavramı; iki ya da daha fazla DNA, RNA makro moleküllerine ait nükleotid dizilerinin ve iki ya da daha fazla protein makro moleküllerine ait amino asit dizilerinin, hizalanacak dizi sayısı iki ise İkili Dizi Hizalama (Pairwise Sequence Alignment), hizalanacak dizi sayısı ikiden fazla ise Çoklu Dizi Hizalama (Multiple Sequence Alignment) kavramları kullanılacaktır, kendi aralarında evrimsel akrabalık dereceleri, benzerlik ölçütleri maksimum olacak şekilde, bu makro moleküllerin sentezi sırasında gerçekleşmesi olası biyolojik olaylar da referans alınarak düzenlenmesi durumunu özetlemektedir.

Hizalama probleminin bir ayrık optimizasyon problemi olarak nitelendirilmesinin temel sebebi, diziler arasında evrimsel akrabalık ya da benzerlik ölçütü en büyük değere sahip sonlu sayıda farklı elemandan oluşan, DNA ve RNA makro molekülleri için dört temel nukleotid, protein makro molekülü için ise yirmi temel amino asit kullanıldığı varsayımı üzerine, sonuç hizalamalarının elde edilmesi amacının bulunması durumuna bağlanabilir [4].

Çoklu dizi hizalama probleminin; dizi sayısı, her bir diziye ait eleman sayısı ve ayrık optimizasyon problemi olması özellikleri dikkate alındığında, Dinamik Programlama yöntemleri kullanılarak, yüksek maliyet ve karmaşıklıklara sahip işlemler yaptırmak

yerine, yaklaşık çözümlerin iteratif tabanlı olarak optimizasyon algoritmaları ile bulunmasının daha iyi olabileceği düşünülmektedir.

Üç veya daha fazla dizi için hizlama problemi temelde, iki dizi için kullanılan hizalama yöntemleri üzerinden genelleştirilebilir. Ancak iki ait diziye hizalama işleminin bile üstel karmaşıklıklara sahip olduğu düşünüldüğünde ve dizi uzunlukları da dikkate alındığında, yukarıda belirtildiği üzere ayrık problemlerin çözümü için oldukça sık tercih edilen bazı sezgisel optimizasyon algoritmalarının kullanılması başvurulan bir yol haline gelmiştir. Genetik Algoritma, Isıl İşlem Algoritması, Parçacık Sürü Optimizasyon Algoritması ve Yapay Arı Koloni Algoritması Çoklu Dizi Hizalama Problemi için ayrık optimizasyon problemlerinin çözümüne uygun olarak tekrar modellenip kullanılmıştır [15-17].

1.2 Tezin Organizasyonu

Çoklu Dizi Hizalama Problemi'nin temel işlem verisi olan DNA, RNA ve protein makro moleküllerinin, kalıtsal bilginin depolanması, jenerasyonlar boyunca çok iyi korunarak aktarılması ve canlılık için gerekli tüm faaliyetlerin sürdürülebilirliğinde oldukça önemlidir. Hizalama Problemi'nin genelinde bu makro moleküller ile ilgili ihtiyaç duyulacak giriş seviyesindeki biyolojik alt yapı Bölüm 2'de verilmektedir.

Bölüm 3, Dizi Hizalama Problemi ve Çoklu Dizi Hizalama kavramları üzerine yapılan tanım ve çalışmaları kapsamaktadır. Problemin çerçevesi tam olarak çizildikten sonra, problemin çözümü için kullanılan klasik metodlardan bahsedilmiş, elde edilen hizalamaların kalitesini göstermek adına kullanılan metrikler verildilkten sonra, ayrık optimizasyon problemi olarak Çoklu Dizi Hizalama probleminin çözümü için sezgisel algoritmalar kullanılarak yapılan çalışmalar özetlenmiştir.

Bölüm 4'de, tezin iskeletini oluşturan Yapay Arı Koloni Algoritması'nın temel işlem adımlarının verilmesi, başlangıçta nümerik optimizasyon problemlerinin çözümü için sunulmuş modelin Çoklu Dizi Hizalama Problemine göre düzenlenmesi konularını kapsamakta, Çoklu Dizi Hizalama Problemi'nin çözümü için Yapay Arı Koloni Algoritması'nın yiyecek kaynaklarının üretilmesinden, işçi, gözcü ve işçi arı fazlarında önerilen iki yeni yaklaşımından bahsedilmektedir.

Bölüm 5, Çoklu Dizi Hizalama Problemi'nin daha önce Yapay Arı Koloni Algoritması, Genetik Algoritma ve Parçacık Sürü Algoritmaları ile çözülmüş çalışma sonuçları ile önerilen Yapay Arı Koloni Algoritması'ndan elde edilen sonuçların karşılaştırmalı olarak verildiği bölümdür. Bu bölümde ayrıca Genetik Algoritma ve dinamik programlama teknikleri ile desteklenmiş iki uygulamadan elde edilen skorlar, önerilen Yapay Arı Koloni Algoritması'ndan elde edilen sonuçlar ile karşılaştırılmıştır. Son olarak önerilen Yapay Arı Koloni Algoritması'nın güncel paralel hesaplama platformlarından olan CUDA'ya aktarılabilirliği üzerinde durulmuştur, farklı ekran kartları üzerinde paralelleştirilebilirliği en yüksek metodlardan birinin test işlemleri gerçekleştirilmiştir. Son bölümde ise tez kapsamında elde edilen sonuçlar yorumlanmış ve önerilere yer verilmiştir.

2. BÖLÜM

DNA, RNA MOLEKÜLLERİ VE PROTEİN SENTEZİ

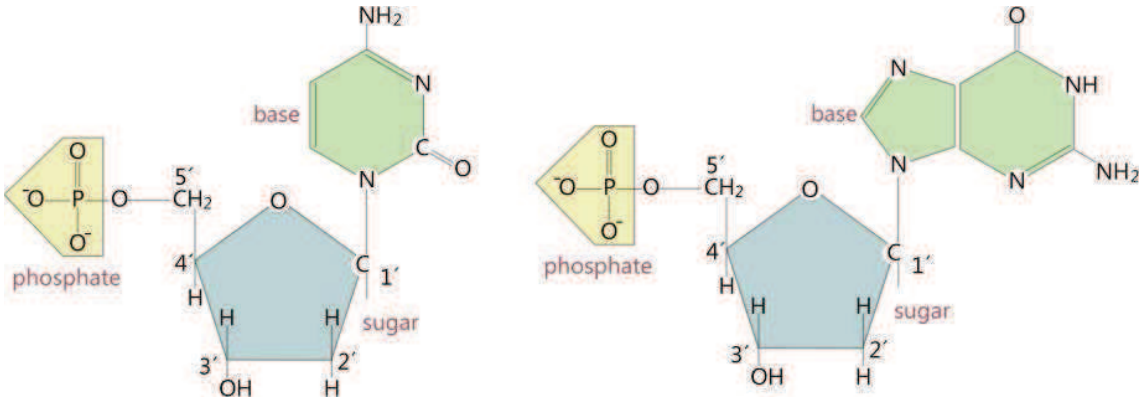
Dizi hizalama probleminin temel işlem verisi, DNA (deoksiribonükleik asit) ve RNA (ribonükleik asit) moleküllerine ait nükleotid dizileri ile birçok biyokimyasal reaksiyonda görev alan protein moleküllerinin amino asit dizileridir. Bu bölümde, dizi hizalama probleminin bir çok aşamasında karşılaşılacak olan kalıtsal bilginin bir jenerasyondan diğerine aktarımından sorumlu DNA molekülünün yapısı, canlılığın devamı için gerekli biyomoleküllerin genomik DNA üzerinde nasıl kodlandığı, DNA üzerinde kodlanmış bilginin RNA aracılığı ile temel işlevsel moleküller olan proteinlere dönüştürüldüğü Central Dogma gibi, bazı durum ve kavramlar üzerinde durulacaktır.

2.1 DNA ve RNA Moleküllerinin Yapı Taşı: Nükleotidler

DNA ve RNA moleküllerinin, kalıtsal bilginin aktarılması, kodlanması ve protein sentezindeki etkileri dikkate alındığında, üstlendikleri görevlerdeki zorluğa karşın, basit sayılabilecek kimyasal yapıya sahip oldukları görülür. Hem DNA hem de RNA biyomolekülleri, farklılıkları özellikle baz olarak isimlendirilen alt birimlerinde gözlenen, dört farklı nükleotid yapı taşından oluşan zincirler halinde bulunur [5]. Kimyasal yapısındaki görece basitliğe karşın genomik DNA molekülleri yüzbinlerce hatta milyonlarca nükleotid barındıracak sekans, zincirlere sahip olabilirken, RNA molekülleri DNA moleküllerine kıyasla daha az sayıda nükleotid barındırır ve üç-boyutlu yapıları DNA molekülleri kadar düzenli olmayabilir [5].

DNA ve RNA moleküllerinin yapı taşı olan nükleotidler temelde, beş-karbonlu şeker halkası, bu beş-karbonlu şeker halkasına 1` (bir üssü) pozisyonunda bağlı azotlu baz ile 5` (beş üssü) pozisyonunda bağlı fosfat grubundan oluşmaktadır [5]. Beş-karbonlu şeker halkası, 2` (iki üssü) pozisyonunda hidroksil grubunun bağlı olup olmamasına göre riboz ve deoksiriboz şekeri olarak iki gruba ayrılır. Beş-karbonlu şeker halkası riboz

şekeri olan nükleotidler ribonükleotid, beş-karbonlu şeker halkası deoksiriboz şekeri olan nükleotidler ise deoksiribonükleotid olarak tanımlanır.



Şekil 2.1 Beş-karbonlu şeker omurgası, azotlu baz ve fosfat grupları [5].

İsimlerinden de anlaşılacağı üzere, DNA (deoksiribonükleik asit) molekülünü oluşturan nükleotid'ler, beş-karbonlu şeker halkasında 2` pozisyonunda hidroksil grubu bulundurmayan deoksiriboz şekerine sahip iken (deoksiribonükleotidlerden oluşmaktadır), RNA (ribonükleik asit) molekülünü oluşturan nükleotidler, beş-karbonlu şeker halkasında 2` pozisyonunda hidroksil grubu (ribonükleotidlerden oluşmaktadır) bulunduran riboz şekerine sahiptir.

Nükleotidler, nükleik asitleri oluşturmak üzere bir araya geldiklerinde; DNA moleküllerinin oluşturulması için sadece deoksiribonükleotidlerin ve RNA moleküllerinin oluşturulması için sadece ribonükleotidlerin seçilmesi canlı sistemlerde özenle kontrol altında tutulmaktadır; nükleotidlerden birinin fosfat grubu ile diğerinin beş-karbonlu şekerinin hidroksil grubu arasına fosfodiester bağları kurulur. Sonuç itibariyle, nükleotid'lerin bir araya gelerek oluşturdukları zincirin bir ucu, fosfodisester bağı yapmamış serbest fosfat grubu barındırken, diğer uç ise yine fosfodiester bağı yapmamış hidroksil grubu barındırır [5, 8].

Nükleik asit zincirinin fosfodiester bağı yapmamış, serbest fosfat grubu barındıran ucu 5` ucu-sonu olarak isimlendirilirken, yine nükleik asit zincirinin fosfodiester bağı yapmamış serbest hidroksil grubu barındıran ucu 2` ucu-sonu olarak isimlendirilmektedir [5, 8]. Nükleotid ya da baz dizileri tek harften oluşan kodları ile belirtilirken, 5` sonundan 3` sonuna doğru yerleştikleri sıra referans alınmaktadır.

2.2 DNA Molekülünün Yapısı

DNA biyomolekülünün üzerine yapılan ve araştırmacılarına Nobel Ödülü kazandıran en önemli çalışmalardan biri, James Watson ve Francis Crick'in Rosalind Franklin'in de çalışmalarından faydalanarak yapıtıkları, DNA biyomolekülünün çift-sarmal (double-helical) yapısının keşfi konusunda olmuştur [4, 5, 8]. James Watson ve Francis Crick'in önerdikleri DNA biyomolekülünün çift-sarmal yapısı, iki DNA zincirinin, bu zincirleri ya da sekansları oluşturan nükleotidlerin baz alt birimleri arasında oluşturulan hidrojen bağları sebebi ile birbirleri üzerine sarılarak helezonik bir yapı aldıkları durumu açıklamaktadır [4, 5].

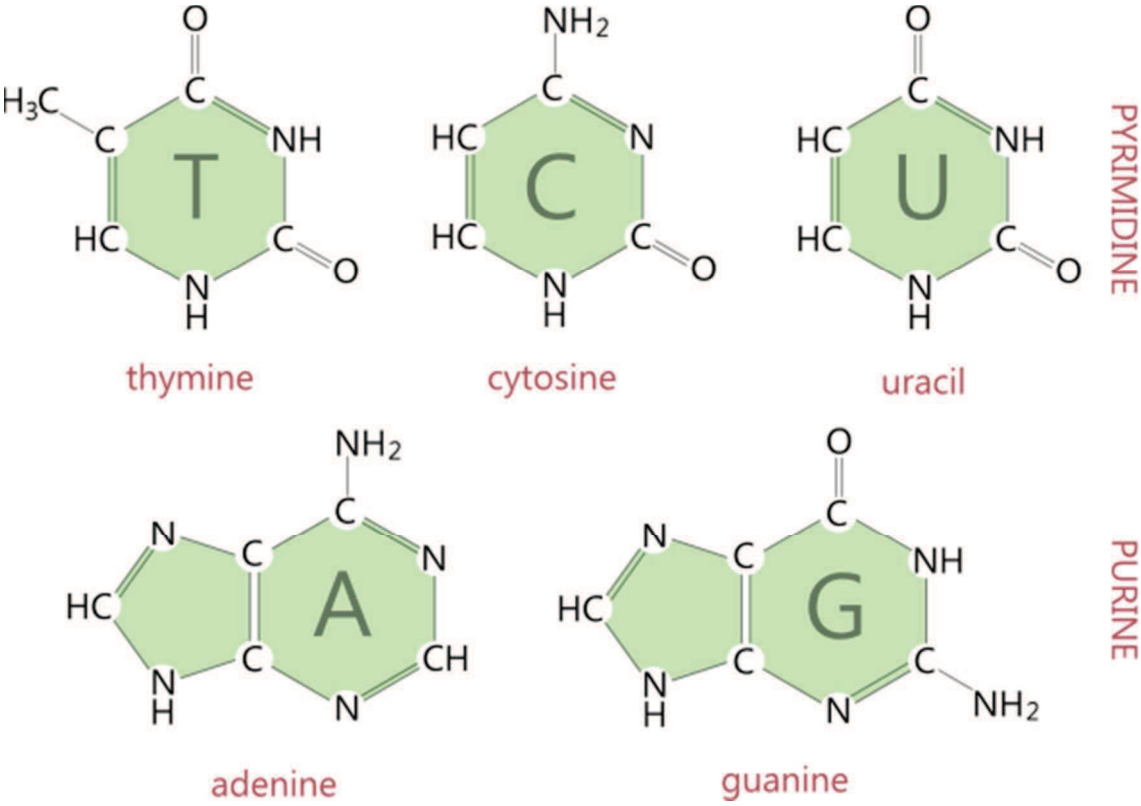
James Watson ve Francis Crick'in çalışmalarındaki en önemli keşif, DNA molekülünün tek değil çift nükleotid zincirini kapsadığı ve çift-sarmal yapının oluşturulması için zincirlerdeki bazlar arasında hidrojen bağlarının sadece belirli baz çiftleri arasında yapıldığı konularında olmuştur. Watson-Crick baz çift modeline göre (Watson-Crick Base Pairing) adenin bazı ile timin bazları arasında, guanin bazı ile stozin bazları arasında hidrojen bağları kurularak, baz çiftleri oluştururlar [5].

Watson-Crick baz çift modelinin önerdiği, belirli primidin bazlarının yine belirli pürin bazları ile baz çifleri oluşturacağı yaklaşımı, adenin ve timin bazları arasında iki, guanin ve sitozin bazları arasında üç hidrojen bağı çift-sarmal yapı kurulurken yapılmaktadır, çift-sarmal yapıyı oluşturan iki nükleotid zincir ya da sekansının birbirlerinin eşleniği (complementary) olduğunu, yani nükleotid zincirlerinden bir belirlenmiş ise, ikincisinin önerilen baz çift modeline göre çıkarılabileceği durumunu açıklamaktadır.

DNA molekülünün çift-sarmal yapısını oluşturan iki nükleotid zincirinin karşılıklı bazları arasındaki hidrojen bağları, 5 ile 20 $kJ\,mol^{-1}$ arasında değişen bağlanma (binding) enerjilerine sahip kovalent olmayan bağlardır [5, 8]. DNA çift-sarmalının birbirinden ayrılmasını gerektiren pek çok durumda, bu durumlardan en önemlisi DNA replikasyonudur (DNA Replication), hidrojen bağları görece sınırlı, az miktarda enerji harcanarak kırılır.

Gerçekte oldukça karmaşık olan, pek çok protein ve enzimin de görev aldığı DNA replikasyon işlemi ile hücrenin bölünmesi sırasında DNA molekülünün çift-sarmal yapısı açılarak çift-sarmalın her bir zinciri, kopyalama işleminin DNA Polimeraz (DNA

Polymerase) enziminin kullanılarak gerçekleştirildiği, şablon olarak davranıp bir önceki DNA molekülünün hemen hemen aynısı olan iki örnek sentezlendiği bilinmektedir. DNA Polimeraz enzimi, şablon DNA zinciri üzerinde hareket ederek, gelen nükleotidler arasından şablonda bulunan nükleotid ile eşleşenin seçilmesi ve seçilen nükleotidlerin birleştirilip şablona ait tümleyenin sentezini, son aşamada da şablon ve oluşturulan tümleyen nükleotid zincirinin birleştirilmesi gibi kompleks bir görevi yerine getirir.

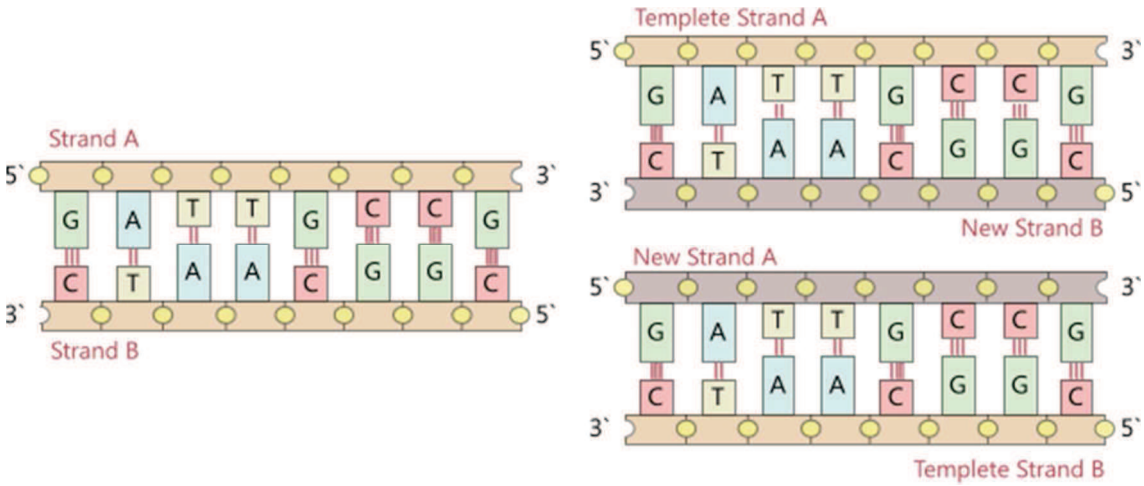


Şekil 2.2 Adenin, Timin, Guanin, Stozin ve Urasil bazları [5].

DNA molekülünün en önemli özelliklerinde biri de, DNA replikasyonu sırasında şablon ya da kalıp olarak davranan nükleotid zincirlerinin eşlenikleri oluşturulurken kalıtsal bilginin korunması amacıyla oldukça az sayıda hatalı eşleşmiş baz çiftleri barındırılacak şekilde kopyalanıyor olmasıdır. DNA replikasyonu ile oluşturulan iki yeni DNA molekülünde hatalı eşleştirildiği düşünülen nükleotid sayısı 10^9 nükleotid arasında bir nükleotid olarak belirlenmiştir [5, 8].

DNA replikasyonu sırasında yanlış eşleşmiş baz çiftlerinin minimizasyonu üzerine geliştirilmiş kontrol mekanizmaları kalıtsal bilginin değiştirilmeden ya da çok iyi korunarak bir sonraki jenerasyona aktarılması durumunun önemini göstermektedir.

DNA molekülüne ait tek nükleotid zinciri, bu nükleotid zincirinin eşleniği ya da tümleyeni formda oluşmuş, DNA ya da RNA nükleotid zinciri ile çift zincirli form oluşturmak üzere birleşebilir [5, 8]. Nükleik asit hibridizasyonu (Nucleic Acid Hybridization) olarak isimlendirilen bu özellik, bir dizi deneysel moleküler biyoloji uygulamasının temelini oluşturmaktadır [5, 8].



Şekil 2.3 DNA çift-sarmalının sentez boyunca şablon olarak kullanılması [5].

RNA molekülleri, DNA moleküllerinin çift-sarmal yapısının aksine, çoğunlukla tek zincir, iplikli yapıda bulunurlar. RNA moleküllerinin bu tek zincir, iplikli yapısı, RNA moleküllerinin DNA moleküllerine kıyasla daha esnek üç boyutlu formlarda bulunmasına olanak sağlar. Bazı RNA moleküllerinin kimyasal reaksiyonları katalize eden enzimler olarak görev aldığı da bilinmektedir [5, 8].

RNA molekülünü oluşturan nükleotid zincirinin kısa alt birimleri birbirinin tam ya da kısmen tümleyeni, eşleniği formunda olabilir. Birbirinin tam ya da kısmen eşleniği durumundaki bu alt nükleotid zincirleri, tıpkı DNA molekülününde olduğu gibi eşleşen nükleotid çiftlerinde sarmal yapı, sarmal yapının sonunda eşleşmeyen nükleotid çiftleri sebebiyle de döngüler oluşturarak özgün bir üç boyutlu yapı alırlar [5, 8].

RNA moleküllerinin özgün üç boyutlu yapılarını düzenleyen nükleotid, bazlar arasındaki bu etkileşimler genelde Watson-Crick baz çift modeline uymadığından, proteinler ile benzerlik kurularak RNA moleküllerinin bu yapıları RNA ikincil yapı (RNA Secondary Structure) olarak isimlendirilmiştir [4, 5, 8].

2.3 Protein Sentezi ve Central Dogma

Moleküler Biyoloji'de DNA, RNA ve protein sentezi arasındaki bağıntıyı tanımlamak adına Central Dogma adı verilen bir kavram kullanılmaktadır. Central Dogma adı verilen bu kavrama göre, genetik bilginin, bu bilginin kodlandığı DNA molekülünden, protein sentezinde sorumlu RNA molekülüne doğru tek yönlü bir akışı olduğu kabul edilmektedir [4, 5, 8].

DNA molekül veya moleküllerinde taşınan genetik bilginin tamamının, sadece organizmanın canlılığı için gerekli tüm reaksiyonlarda görev alan temel işlevsel bileşen olan proteinleri ve protein sentezinde ya da biyokimyasal reaksiyonlarda enzim olarak görev alan RNA moleküllerini kodlamadığı, bazı bölgelerin henüz tespit edilememiş fonksiyon ya da sentezden sorumlu olduğu bilinmektedir [5, 8].

Canlılığın sürdürülebilmesi için gerekli genetik bilginin kodlandığı DNA, bilinen en uzun ve büyük moleküller arasında yer almaktadır. Örneğin, insan genomuna ait 46 kromozomu oluşturan 23 çift DNA molekülü arasında, en küçük olanının 47 milyon, en büyük olanının 247 milyon baz içermesi, bütün insan genomun toplamda yaklaşık 3 milyar baz içerdiği bilinmektedir ki genetik bilginin kodlanması ve diğer bileşenlerin sentezi için kullanılması konusundaki karmaşıklığı göstermektedir [5, 8].

Her bir kromozomun DNA molekülü, yüzlerce hatta binlerce farklı proteini, kromozoma bağlı olarak, DNA molekülü üzerindeki belirli segment, parçalarda kodlamaktadır. Bir ya da birden fazla proteinin kodlandığı bu segment, parçalar ile bu segment, parçalara oldukça yakın posizyonlarda olup karmaşık kontrol görevlerini üstlenmiş bölge veya bölgeler, birlikte protein veya proteinlere ait genleri oluşturular [5]. Kısaca gen, DNA molekülü üzerinde belirli bir proteinin ya da proteinlerin kodlandığı alt birim, parçalar gibi düşünülebilir. Genin protein kodlayan sekansın protein ile ilgili taşıdığı bilgiyi genetik kod olarak isimlendirilen kodlanmış formda taşıdığı bilinmektedir,

DNA üzerindeki gen ya da genlerin nükleotid, baz dizilerinin genetik kod olarak adlandırışmış bir formda, tek ya da birden fazla proteine ait amino asit zincirini tanımladığı belirtilmişti. Ancak, gen ya da genler üzerindeki genetik kodun doğrudan, genelde 20 değişik amino asit türü kullanılarak oluşturulan ve bu 20 amino asitin farklı

özelikleri sayesinde biyokimyasal reaksiyonlarda oldukça kullanışlı proteinlere doğrudan geçilemez. Protein sentezinin gerçekleştirileceğine dair uyarı sinyalleri, proteinin genetik kodunu barındıran gen üzerinde, genin tek-zincirli RNA kopyasının oluşturması ile sonuçlanan ve transkripsiyon (Transcription) olarak isimlendirilen işlemi başlatır. Transkripsiyon, temelde DNA Replikasyon işlemine benzemesine karşın, DNA zincirlerinden sadece biri şablon olarak kullanılır ve sentez ürünü DNA değil RNA molekülüdür [4, 5, 8].

Transkripsiyon işlemini, RNA Polimeraz (RNA Polymerase) enzimi ile katalize edilir. RNA Polimeraz enzimi, tıpkı DNA Polimeraz enziminin yaptığı gibi, şablon görevindeki DNA sekansı üzerinde ilerleyerek, DNA sekansındaki bazlara karşılık gelen ribonükleotidleri seçip, seçilen ribonükleotidleri birleştirerek RNA zincirini oluşturur [5]. Protein kodlayan gen aracılığı ile protein sentezini yöndendirmesi için sentezlenen bu RNA, mesajcı RNA (messenger RNA) olarak isimlendirilir [5, 8]. RNA zinciri, DNA zincirinin belirli bir proteinin sentezinden sorumlu parçası yani geni üzerinden sentezlendiği için sentezlendiği DNA zincirinden oldukça küçüktür.

Verilen gene ait DNA zincirlerinden sadece biri transkripsiyon işleminde şablon, kalıp olarak kullanılır. Transkripsiyon sonucu sentezlenen RNA zinciri, son ürün olan protein molekülüne ait tüm bilgiyi içerirken, RNA sentezi için şablon, kalıp olarak kullanılan DNA zinciri protein molekülünü kodlayan bilgiyi doğrudan tutmaz, bunun yerine protein molekülünü kodlayan bilginin eşleniğini barındırır.

Mesajcı RNA zincirinin sentezi için şablon, kalıp görevindeki DNA zincirine kodlamayan zincir (noncoding, anticonding, antisense strand), Mesajcı RNA zincirinin sentezi için şablon, kalıp görevinde olmayan ancak sentezlenen Mesajcı RNA molekülüne ait DNA eşleniğini barındıran DNA zincirine ise kodlayan zincir (coding sense) adı verilmektedir. Kodlayan zincir, gene ait nükleotid dizisi verilirken kullanılır ve bu zincir ile genetik kodlama kurallarına uyularak protein zinciri çıkartılabilir [5].

Genomik DNA sekansında tutulan bilgi sadece protein zincirine ait değildir. Genomik DNA protein zincirine ait bilginin yanı sıra, transkripsiyon işleminin ne zaman ve hangi pozisyonda başlayacağına ilişkin kontrol dizileri de içermektedir.

2.4 Genetik Koda Göre Amino Asitlerin Belirlenmesi

Genetik kod, DNA veya RNA moleküllerindeki nükleotid, baz dizilerinin karşılık geldiği amino asit zincirinin belirlenmesi için tanımlanmış kuralları ifade etmektedir. Genetik koda göre her bir amino asit, üç bazdan oluşan ardışık gruplar ile tarif edilmektedir [5]. RNA molekülünde üç bazdan oluşan ardışık grupları kodon (codon) olarak adlandırılıp, genetik kod tabloları, genetik kodu RNA kodonlarını referans alınarak oluşturulmaktadır [5]. Aşağıda standart genetik kodları içeren bir diyagram verilmiştir.

5` Sonu (Kodonun ilk harfi)	Kodonun ikinci harfi U	Kodonun ikinci harfi C	Kodonun ikinci harfi A	Kodonun ikinci harfi G	3` Sonu (Kodonun üçüncü harfi)
U	UUU (Phe)	UCU (Ser)	UAU (Tyr)	UGU (Cys)	U
	UUC (Phe)	UCC (Ser)	UAC (Tyr)	UGC (Cys)	
	UUA (Leu)	UCA (Ser)	UAA (Stop)	UGA (Stop)	
	UUG (Leu)	UCG (Ser)	UAG (Stop)	UGG (Trp)	
C	CUU (Leu)	CCU (Pro)	CAU (His)	CGU (Arg)	C
	CUC (Leu)	CCC (Pro)	CAC (His)	CGC (Arg)	
	CUA (Leu)	CCA (Pro)	CAA (Gln)	CGA (Arg)	
	CUG (Leu)	CCG (Pro)	CAG (Gln)	CGG (Arg)	
A	AUU (Ile)	ACU (Thr)	AAU (Asn)	AGC (Ser)	A
	AUC (Ile)	ACC (Thr)	AAC (Asn)	AGC (Ser)	
	AUA (Ile)	ACA (Thr)	AAA (Lys)	AGA (Arg)	
	AUG (Met)	ACG (Thr)	AAG (Lys)	AGG (Arg)	
G	GUU (Val)	GCU (Ala)	GAU (Asp)	GGU (Gly)	G
	GUC (Val)	GCC (Ala)	GAC (Asp)	GGC (Gly)	
	GUA (Val)	GCA (Ala)	GAA (Glu)	GGA (Gly)	
	GUG (Val)	GCG (Ala)	GAG (Glu)	GGG (Gly)	

Şekil 2.4 Standart genetik kodlar ve karşılığı amino asitler örneği [5].

Standart genetik kodların gösterildiği diyagramdan da anlaşılacağı üzere, amino asitler birden fazla genetik kod ile tanımlanmaktadır. Bu durum standart genetik kodun dejenere (degenerate) olduğunu, yani verilen bir DNA ya da RNA zincirinden protein molekülünü oluşturacak amino asit dizisinin bulunacağı ancak verilen protein molekülüne ait amino asit dizisi için tek bir DNA ya da RNA zincirinin oluşturulamayacağını belirtir [5, 8].

Standart genetik kodlardan 3 tanesi, herhangi bir amino asidi kodlamayıp, amino asit zincirinin, sentezinin tamamlandığını belirten bitiş (stop) kodonları olarak görev yapar. Protein sentezinin başlatıldığı kontrol mekanizması ise, protein sentezinin

tamamlanmasını kontrol eden mekanizmaya kıyasla daha karmaşıktır. Pek çok durumda, metiyonin (methionine) amino asitinin kodlandığı AUG kodonu, protein sentezinin başlatılmasını yani RNA zincirinde uygun nükleotid seçilerek, elemanları birbiriyle çakışmayan ardışık üç bazlı gruplar ile belirtilen kodonlara karşılık gelen amino asitlerin birleştirildiği translasyon (translation) işleminin başlamasını sağlar [5,8].

Bazların amino asitlere translasyonunun genetik koda göre, çakışık olmayan, ardışık üç bazlı gruplar ile yapıldığı göz önüne alınırsa, translasyonun başlayacağı baza göre üç farklı protein sentezinin yapılabileceği, amino asit zincirinin oluşturulabileceği görülür. Bazların amino asitlere translasyonunun başlatıldığı ilk ardışık üç bazlı grup okuma çerçevesi (reading frame) olarak adlandırılır [5, 8].

DNA molekülünde protein kodlayan sekansın tahmini için ise, her bir DNA zinciri için üç olmak üzere, toplamda altı farklı durum test edilmelidir. Hem RNA hem de DNA zincirleri için yapılacak tahminlerden sadece biri fonksiyonel bir protein sentezi, bu proteinlerin uzunlukları genelde 100 amino asit üzerinde olmaktadır, ile sonuçlanır. Hatalı okuma çerçeveleri ise, genellikle fonksiyonel protein molekülerine kıyasla daha erken bitiş kodonuna, sonuç olarak daha kısa amino asit zincirlerine sahip olmaktadır.

3. BÖLÜM

ÇOKLU DİZİ HİZALAMA PROBLEMİ ve ÜZERİNE YAPILAN ÇALIŞMALAR

Rekombinant DNA (Recombinant DNA) teknolojisi ve DNA makro moleküllerine ait sekanslarnın belierlenmesi (DNA Sequencing) yöntemlerindeki gelişme, milyonlarca farklı nükleotid ya da amino asit dizilerine sahip yeni veri tabanlarının kurulmasını sağlamıştır [4]. Türler ve biyokimyasal reaksiyonladaki çeşitlilik dikkate alındığında yeni çalışmalardan elde edilecek verilerin değerlendirilmesi, mevcut veriler ile karşılaştırılabilirliği neredeyse doğrudan sekansların çoklu analiz edilebilmesine bağlı hale gelmiştir. Sekansların analiz yöntemleri arasında en önemli olanı, evrimsel akrabalık, benzerlik tahmini, filogenetik ağaçaların oluşturulması, yapı ve fonksiyon tahminleri için de kullanılabilecek dizi hizalama yaklaşımlarıdır.

Hizalanacak dizi sayısına göre, hizalanacak dizi sayısının iki olduğu durumlarda İkili Dizi Hizalama (Pairwise Sequence Alignment) Problemi ve hizalanacak dizi sayısı ikiden fazla oldugu durumlarda Çoklu Dizi Hizalama (Multiple Sequence Alignment) Problemlemi ayrımı yapılmaktadır, iki temel gruba ayrılan dizi hizalama probleminin tanımı, Çoklu Dizi Hizalama Probleminin temel özellikleri, evrimsel mekanizmaların etkisinin dizi hizalama işlemi üzerinde nasıl modellendiği, elde edilen hizalamaların biyolojik geçerliliklerini gösterecek temel skor sistemleri bu bölümde verilmektedir.

Farklı türlere ait ancak benzer fonksiyonlara sahip DNA, RNA ya da protein makro moleküllerinin, DNA makro molekülleri kendi aralarında, RNA makro molekülleri kendi aralarında, protein makro molekülleri de kendi aralarında hizalanmaktadır, çoklu hizalanması, hizalanacak sekanslar üzerinde evrimsel mekanizma etkilerinin dahil edilmesiyle sütun bazında eşleşen nükleotid ya da amino asit türünün maksimizasyonunu düzenlediğinden, Çoklu Dizi Hizalama Problemi optimizasyon

problemleri arasında değerlendirilebilir. Çoklu Dizi Hizalama Problemi üzerine geliştirilmiş ve dinamik programlama olarak isimlendirilen algoritmik yaklaşım ile sezgisel algoritmaların kullanıldığı temel çalışmalara bu bölümde değinilecektir.

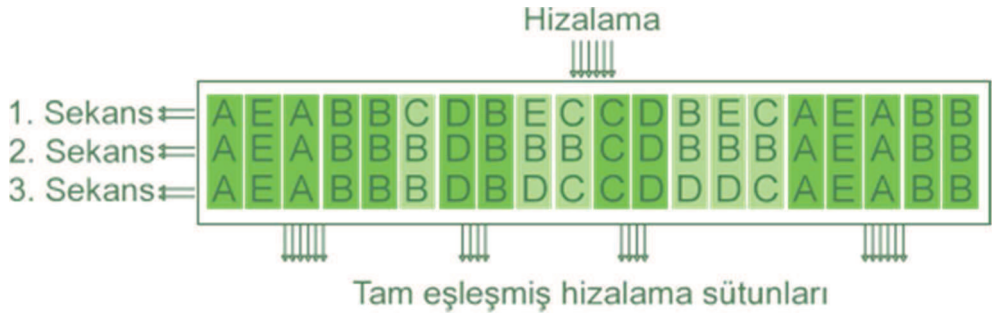
Çoklu Dizi Hizalama Problemi'nin ayrık optimizasyon problemi olarak değerlendirilebilmesi sayesinde, hizalama problemi yönlü graflar ile ifade edilip dinamik programlama yöntemlerinin getirdiği işlem yoğunluğundan sıyrılmış, sezgisel algoritmaların kullanılabilirliğinin önü açılmıştır. Hem dinamik programlama hem de sezgisel algoritmalar kullanılıp en kaliteli hizalamalar üretilmiş olsa dahi, hizalamanın kalitesi, geçerliliği kabul edilmiş aynı sekansları barındıran hizalama ile karşılaştırılmadan anlaşılamayabilir.

3.1 Dizi Hizalama Probleminin Temel Özellikleri

Mutasyon (mutation), insersyon (insertion), delesyon (deletion) ve seleksiyon (selection) gibi evrimsel mekanizmalar, aynı ortak genden türediği düşünülen şu anki gen, nükleotid dizileri üzerinde dikkate değer oranlarda ayrılma, ıraksamaya sebep olmuştur [4-7]. Evrimsel mekanizmaların milyonlarca yıllık etkisi üzerine türetildikleri ortak genden ıraksamış şu anki nükleotid dizileri, değişen yapıları ve uzunlukları sebebi ile başlangıçta türedikleri gen ya da kendi aralarında olduğu varsayılan benzerliklerini kaybetmiştir [4-7].

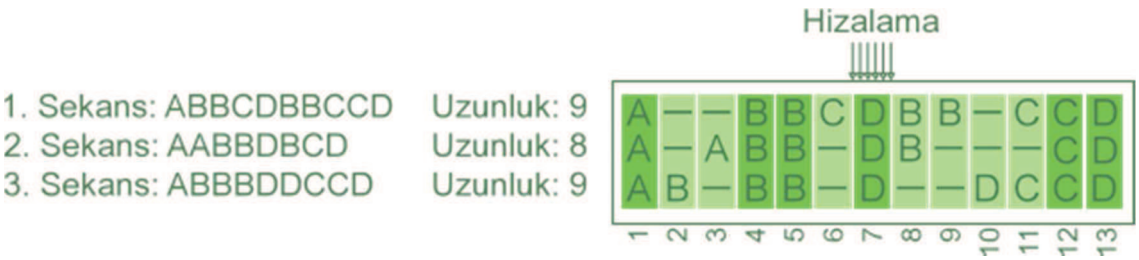
Evrimsel süreç boyunca ilgilenilen nükleotid dizileri ya da bu dizilerin kodladığı proteinlere ait amino asit dizilerinin atası durumundaki dizi ile başlangıç benzerliğinin kabul edilmesi durumunda, tüm sekansların uzunlukları kontrollü şekilde eşit yapılıp aynı pozisyonlardaki sekans bileşenleri (residue) kontrol edildiğinde, eşleşen sekans bileşen sayısının maksimum olması beklenir [4-7]. Bu durum tam olarak dizi hizalama analiz yönteminin ya da dizi hizalama probleminin çözüm dayanak noktasıdır. Dizi hizalama, nükleotid veya amino asit sekanslarının kendi aralarında yukarıda bahsedilen mutasyon, insersyon, delesyon ve seleksiyon gibi evrimsel mekanizmaların etkileri dahil edilerek eşit uzunluklara getirildiğinde aynı pozisyon, pozisyonlarda eşleşen sekans bileşen sayısının maksimum olması, ata sekansın tahmin edilebilmesi durumunu özetler [4-7].

Sekanslar hizalandığında, sekansların türediği ata sekansın mutasyon, insersyon veya delesyon evrim mekanizmalarından korunmuş şablonlarının, hizalamalarda da denk gelmiş olması olasıdır. Aşağıda, evrimsel sürecin sekans uzunluklarını değiştirmediği kabulu ile bileşenleri rastgele belirlenmiş üç sekansa ait hizalama durumu gösterilmektedir.



Şekil 3.1 Bileşenleri rastgele belirlenmiş üç sekansın örnek hizalanması.

Şekil 3.1'de verilen üç sekans için hizalama işlemi, hizalama durumunun yukarıda belirlenen özellikleri sağlanacak, eşleşen sütun sayısı maksimum olacak şekilde kolaylıkla yapılmıştır. Ancak, hizalancak sekansların çoğu, örnekteki gibi evrimsel mekanizmaların ıraksatıcı etkisine az sayılabilecek derecede maruz kalmaz. Sekanslar birbirinden uzaklaştıkça karşılaştırılması veya hizalanması işlemi de ıraksmayla orantılı olarak zorlaşmaktadır. Aşağıda aralarındaki ıraksama, ilk duruma kıyasla daha çok olan üç sekansın, hizalama durumunun istenilen özellikleri sağlayacak şekilde karşılaştırılması işlemi gösterilmektedir.



Şekil 3.2 Uzunlukları aynı olmayan üç sekansın hizalanması.

Şekil 3.2'de verilen örnekte de anlaşılacağı üzere mutasyon, insersyon ve delesyon gibi evrimsel mekanizmaların etkisi, boşluk (gap, "-") karakteri ya da karakterlerinin tüm sekansların uzunluğu eşit olana kadar eşleşen sütun sayısının maksimum olması dikkate alınarak belirli pozisyonlara, orijinal sekanstaki bileşenlerin kendi içlerindeki

pozisyonlara değiştirilmeden yerleştirilmesi şeklinde modellenmiştir. Dizi hizalama probleminin temel işlem sahası, boşluk karakterlerinin tek olmayan en iyi hizalamayı bulabilmek adına, uygun sayıda, gerekli pozisyonlara yerleştirilmesi durumunu kapsamaktadır.

Dizi Hizalama Problemi'nin çözümü ile bulunmuş hizalamalar, eşleşen sütun sayısı bakımından değerlendirildiğinde en kaliteli hizalamalar arasında yer alsalar da, hizalanacak sekansların homolog olup olmadığı hususunda yeterli bilgi taşıdıkları söylenemeyebilir [5, 7]. Homoloji kavramı, Moleküler Biyoloji ve Evrim alanlarında farklı kavramlara karşılık gelmesine rağmen, homoloji kavramının genel itibariyle ortak ataya benzerlikten bahsettiğini söyleyebiliriz [5, 7].

DNA, RNA gibi kalıtsal makro molekülleri ile protein makro moleküllerine ait sekansların, karşılaştırıldıkları diğer sekanslar ile homolog olabilmesi, biyokimyasal yapı ve fonksiyonel görev benzerliklerinin de kontrol edilmesini gerektirir [4, 5, 7]. Hizalama sonucu sekansların akrabalık dereceleri ile ilgili karar verilirken bu akrabalığın gerçekten sekansların ortak bir ataya sahip olmalarından mı kaynaklandığı yoksa evrimin yakınsatıcı etkisi ile mi oluştuğu husunda ayrım yapılması da oldukça önemlidir [5].

3.2 Hizalanmış Dizilerin Skor Değerlerinin Belirlenmesi

Homolog sekans sayısı ve sekans uzunlukları arttıkça, sekans uzunlukları birbirlerine yakın olsalar dahi hizalama probleminin kolaylıkla çözülebilmesi, eşleşen sütun sayısını maksimize edebilmek adına belirli sayıda boşluk karakteri eklenecek algoritmik bir yaklaşıma ihtiyaç duyar. Kullanılan algoritmik yaklaşım sonucu hizalanmış dizilerin, diğer hizalamalar ile karşılaştırılabilmesi için biyolojik olarakta geçerli kalitelerinin bir metrik, sayısal değer ile, daha çok eşleşen sütun barındıran hizalamaların daha yüksek skorlara sahip olması beklenmektedir, ifade edilmesi gerekmektedir.

Hizalamalara ait kalite değerlerini gösterecek skorları üretmek için evrimsel mekanizmaların karmaşık etkisini tam olarak yansıtacak skor şeması, evrimsel mekanizmaların kompleks etkisi basit skor şemaları ile tanımlanamayabilir, henüz mevcut değildir [4, 26-29]. Skor şema ya da sistemlerinin biyolojik geçerlilik ve uygunluk adına yapılan işlemleri de barındırıyor olması, skor şemasının hizalamalarda

kullanılabilirliğini arttırmaktadır. Aşağıda hizalamalara ait kalitelerin belirlenmesi için kullanılan skor şemalarından en sık kullanılanları sırasıyla tanıtılmaktadır.

3.2.1 Percentage Identity Skoru ve Hesaplanması

Identity, her biri kendi türleri içinde hizalanmış iki ya da ikiden fazla DNA, RNA ve protein makro moleküllerine ait sekansların aynı hizalama pozisyonunda DNA ile RNA için ortak nükleotid, protein için ortak amino asit bulundurması olayını özetler, aynı hizalama pozisyonunda DNA ile RNA makro molekülleri tam eşleşen nükleotid, protein makro molekülleri için ise tam eşleşen amino asit sütun sayısı toplanarak hesaplanan en temel benzerlik, kalite ölçütüdür [4-7].

Percentage Identity ya da Percent Identity skoru ise, hizalama için bulunan identity değerinin hizalama uzunluğuna bölünmesi ile elde edilen skor olarak tanımlanmaktadır [4-7]. Aşağıda, evrimsel sürecin sekans uzunluklarını değiştirdiği kabulu ile, bileşenleri rastgele belirlenmiş üç homolog sekansa ait hizalama durumu ve hizalama için Percentage Identity skorunun hesaplanışı gösterilmektedir.

Şekil 3.3 Percent Identity skorunun hesaplanması.

Percent Identity skoru, hizalanmış sekanslar arasında hiç tam eşleşmemiş sekans bileşeni yok ise sıfır değerine sahip olacaktır. Ancak hizalacak sekanslar, DNA veya RNA makro moleküllerine ait ise nükleotid olarak adlandırılan en çok dört farklı sekans bileşeni, protein makro moleküllerine ait ise amino asit olarak adlandırılan en çok yirmi farklı sekans bileşeni bulunduracağından, tam eşleşmiş sütunlara sonlu olasılıkla rastlanabilir [4, 5].

Percent Identity skorunun hesaplanmasında hizalancak sekansların uzunlukları oldukça önemlidir. Yapılan çalışmalara göre, uzun sekansların hizalanması sonucu Percent Identity skorunun tesadüfi olarak %30 olması, kısa sekansların hizalanması sonucu Percent Identity skorunun tesadüfi olarak %30 olmasından daha düşük olasılığa sahiptir [5, 6].

3.2.2 Sum Of Pairs (SP) Skoru ve Hesaplanması

Hizalamaların kalite hesaplama işlemlerinde en yaygın kullanılan skor yöntemi Sum Of Pairs (SP) skor yöntemidir. Sum Of Pairs Skor hesaplama yönteminin temel yaklaşımı, elde edilecek skorun hizalamanın geneline ya da aynı sütun üzerindeki olası tüm ikili eşleşmelere yayılması üzerine kurulmuştur [4, 5, 26, 27, 29, 32]. N hizalanmış sekans sayısı olmak üzere Sum Of Pairs (SP) skorunun hizalamanın j. sütununda hesaplanması için aşağıdaki formül kullanılmaktadır [32].

$$SP(j) = \sum_{i=0}^{N-1} \sum_{k=i+1}^{N} p(c_{ij}, c_{kj}) \tag{3.1}$$

L hizalamanın uzunluğu, N hizalanmış sekans sayısı olmak üzere, Sum Of Pairs (SP) skorunun tüm hizalama için hesaplanması amacıyla aşağıdaki formül kullanılabilir [32].

$$SP(A) = \sum_{i=1}^{L} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} p(c_{ij}, c_{ik}) \tag{3.2}$$

Eşitlik 3.2'de p fonksiyonunun parametreleri olan c_{ij} ve c_{ik} parametreleri sırasıyla hizalamanın j. ve k. sekansların i. sütununda bulunan sekans bileşenlerini temsil eder. p fonksiyonunun c_{ij} ve c_{ik} parametreleri için üreteceği değerler ile ilgili genel bir kural olmamasına karşın, yaygın olarak kullanılacak yaklaşım c_{ij} ve c_{ik} parametrelerinin eşleşip eşleşmediği, boşluk karakterleri olup olmadıkları kontrolüne dayanır. Aşağıda p fonksiyonunun c_{ij} ve c_{ik} parametreleri için üreteceği değerler ile ilgili olası durumlar gösterilmektedir [32].

$$p(c_{ij}, c_{ik}) = \begin{cases} 1, c_{ij} = c_{ik}\ ve\ c_{ij} \neq '-' ve\ c_{ik} \neq '-' \\ 0, c_{ij} = c_{ik}\ ve\ c_{ij} = '-' ve\ c_{ik} = '-' \\ 0, c_{ij} = c_{ik}\ ve\ c_{ij} \neq '-' ve\ c_{ik} \neq '-' \end{cases} \tag{3.3}$$

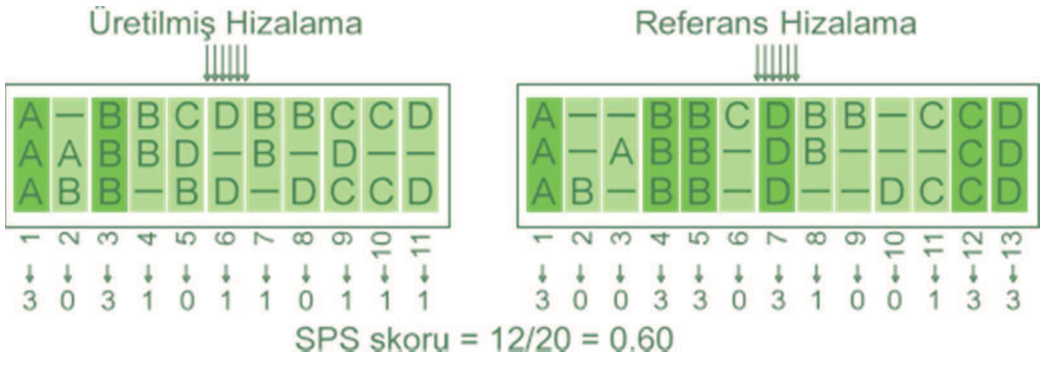
p fonksiyonunun c_{ij} ve c_{ik} parametreleri için üreteceği değerler, akrabalıkları ve fonksiyonel özellikleri bilinen sekanslar kullanılarak yapılan istatistiksel çalışmalar sonucu üretilmiş Değişim (Substitution) Matrisleri kullanılarak da belirlenebilir. Bu matrislerden en çok kullanılanları, ilerleyen konu başlıkları altında bu matrisler ile ilgili bilgi verilecektir, PAM ve BLOSUM matrisleridir.

3.2.3 Relative Sum Of Pairs Skoru (SPS) ve Hesaplanması

Sum Of Pairs Skor hesaplama yönteminin temel yaklaşımı, elde edilecek skorun hizalamanın geneline ya da aynı sütun üzerindeki olası tüm ikili eşleşmelere yayılması üzerine kurulmuştu. Sum Of Pairs skor hesaplama metodu bu yaklaşımıyla, hizalanan tüm sekansların evrimsel olarak birbirinden bağımsız olduğu kabulünü yapmış, sütun üzerindeki eşleşmelere bağımlılığı artmıştır [4-7, 32]. Relative Sum Of Pairs skoru, Sum Of Pairs skor hesaplama yönteminin yukarıda bahsedilen bağımlılığı azaltabilmek adına, tekrar hizalancak sekanslara ait kabul edilmiş standart hizalama var ise, bu hizalamanın Sum Of Pairs skorunun da kullanılması temeline dayanmaktadır. A_{new} üretilen yeni hizalama, L_{new} üretilen hizalamanın uzunluğu, A_{std} kabul edilmiş standart hizalama, L_{std} kabul edilmiş standart hizalama uzunluğu ve sekans sayısı N olmak üzere Relative Sum Of Pairs (SPS) skoru aşağıdaki gibi hesaplanabilir [26, 29, 32].

$$SPS(A_{new}) = \frac{\sum_{i=1}^{L_{new}} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} p(c_{ij}, c_{ik})}{\sum_{i=1}^{L_{std}} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} p(c_{ij}, c_{ik})} \qquad 3.4$$

Relative Sum Of Pairs skorunun, üretilen ve stantart hizalamalara ait Sum Of Pairs skorları kullanılarak hesaplanmasına ait örnek uygulama Şekil 3.4'te gösterilmektedir.



Şekil 3.4 Relative Sum Of Pairs skorunun hesaplanması.

Relative Sum Of Pairs skoru, elde edilen hizalama ile karşılaştırılabilecek standart hizalama yok ise, üretilen hizalama uzunluğu kullanılarak aşağıdaki yöntem ile de hesaplanabilir [32].

$$SPS(A_{new}) = \frac{\sum_{i=1}^{L_{new}} \sum_{j=0}^{N-1} \sum_{k=j+1}^{N} p(c_{ij}, c_{ik})}{L \; x \; N \; x \; (N-1) \; / \; 2} \qquad 3.5$$

3.2.4 PAM Değişim Matrisi ve PAM Değişim Matrisinin Üretilmesi

Margaret Dayhoff ve çalışma arkadaşları tarafından, 1978 yılında aralarında Cytochrome C, Insulin A ve B zincirleri, Ferredoxin ile α ve β globin isimli protein ailelerinin de olduğu çoklu hizalamalar kullanılarak üretilmiş PAM (Point Accepted Mutations) ya da diğer adıyla MDM (Mutation Data Matrix), proteinlere ait hizalamaların kalitelerini belirlemek amacıyla en sık başvurulan skor sistemleri arasında gelmektedir [5].

PAM değişim skor matrisleri oluşturulurken kullanılan çoklu hizalamalar, hizalanan ailelerin mutasyon sonucu değişebilirliği, sekansların amino asit kompozisyonu ya da dağılımı ve sekans uzunluklarının etkilerini minimize edebilmek adına, kullanılan her bir hizalama için amino asitlere ait mutasyona maruz kalma miktarları ayrı ayrı hesaplanmaya çalışılmıştır [4, 5].

Amino asitlere ait mutasyona maruz kalma oranlarının hesaplanması işleminde, ilgilenilen amino asidin hizalamadaki bulunma oranı (compositional fraction) ile yüz hizalama pozisyonunda kabul edilen mutasyon sayısı çarpılmaktadır. PAM değişim matrislerinin hesaplanmasında en önemli husus yüz hizalama pozisyonunda gerçekleşebilirliği kabul edilen mutasyon sayısıdır.

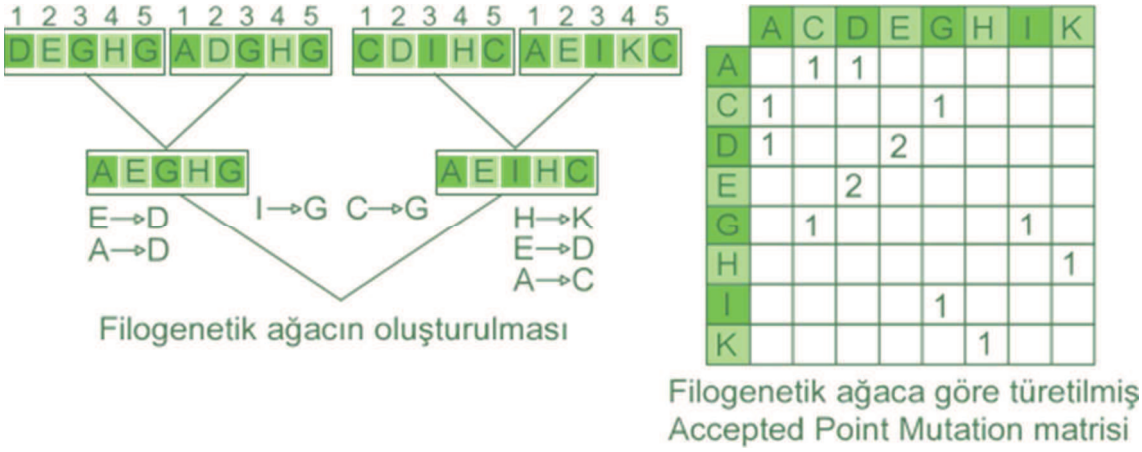
PAM değişim matrisleri hesaplanırken yüz hizalama pozisyonunda gerçekleşebilirliği kabul edilen mutasyon sayısı, bir dizi işlem ve kabul sonrasında karar verilen parametredir. Değişim matrisinde her bir terimin belirlenebilmesi için yapılan ilk işlem ilgilenilen amino asidin relative mutability değerinin hesaplanmasıdır.

Özel bir amino asidin mutability değeri, bütün hizalamalarda ilgilenilen amino asidin mutasyona uğrama sayısı ile bu amino asidin tüm hizalamalarda elde edilen mutasyona maruz kalma oranlarına ait toplamın oranı şeklinde elde edilir. Relative mutability

değeri ise, ilgilenilen amina asidin alanin isimli amino aside ait mutability değerine bağlı ifade edilmesiyle bulunur [5]. Alanin amino asidine göre a amino asidinin relative mutability değeri m_a şeklinde gösterilmektedir [5].

Değişim matrisinin her bir teriminin elde edilebilmesi için yapılan ikinci işlem, hizalanacak sekanslar üzerinden türetilecek filogenik ağaçların oluşturulması durumu ile başlar [5]. Her bir hizalama için, ara katmanlarda olası ortak atayı temsil edecek bireylerin bulunmaya çalışıldığı filogenetik ağaç oluşturulurken, tüm mutasyonların gerçekleştiği kabul edilir. Gerçekleşen mutasyonların tamamı eşit olasılıklı olup, a türü bir amino asidin b türüne mutasyonu, hem a türünden b türüne, hem de b türünden a türüne mutasyonda, toplam mutasyon sayısına dahil edilir [5].

Her biri eşit olasılıklı ve çift yönlü sayılan mutasyonlar kullanılarak A ile gösterilen Accepted Point Mutation matrisi elde edilir. Filogenetik ağacın oluşturulması ve Accepted Point Mutation matrisi ile ilgili aşağıda gösterilen örnek durum incelenebilir.



Şekil 3.5 Filogenetik ağacın oluşturulması ve Accepted Point Mutation matrisi [5].

Birinci adımda hesaplanan relative mutability ve ikinci adımda hesaplanan Accepted Point Mutation matrisi, mutasyonların gerçekleşme olasılıklarını gösterecek ve M ile gösterilecek Mutation Probability matrisinin hesaplanması için kullanılacaktır.

$M_{a,b}$ değeri b tipi sekans bileşeninin a tipi sekans bileşenine mutasyon olasılığını, m_b, b sekans bileşeninin relative mutability değerini, Λ değeri ise belirlenmemiş evrimsel periyodu, $A_{a,b}$, Accepted Point Mutation matrisinin a sekans bileşenininden b sekans

bileşenine mutasyon sayısını temsil etmek üzere, M matrisinin köşegen üzerinde olmayan elemanları hesaplanırken aşağıdaki formülasyon kullanılabilir [5].

$$M_{a,b} = \frac{\Lambda m_b A_{a,b}}{\sum_a A_{a,b}} \tag{3.6}$$

M ile temsil edilen Mutation Probability matris, bir olasılık matrisi olduğuna göre M matrisinin köşegen değerleri aşağıdaki gibi bulunabilir [5].

$$M_{b,b} = 1 - \Lambda m_b \tag{3.7}$$

Mutation Probability matrisinin temsil ettiği evrimsel sürecin sonrasında, ortalama dağılıma sahip bir sekansta değişmeden kalan sekans bileşeni sayısı bulunurken, f_b ortalama dağılıma sahip sekanstaki b tipi sekans bileşeninin frekansı, $M_{b,b}$ Mutation Probability matrisinin köşegen üzerindeki elemanı olmak üzere değişmeden kalan sekans bileşen sayısı, Mutation Probability matrisi M bir olasılık matrisi, f_b frekansı da b sekans bileşeninin beklenen değeri olduğundan matristeki tüm elemanlar ile beklenen değerlerinin çarpımlarının toplamı bire eşit olacaktır, aşağıdaki gibi bulunabilir [5].

$$100 \sum_b f_b M_{b,b} = 100 \sum_b f_b (1 - \Lambda m_b) \tag{3.8}$$

PAM matrisinin temel düzenlenme stratejisi Eşitlik 3.8'deki formülasyona bağlıdır. Toplam operatörüne ait işlem sonuçlandığında, seçilen Λ parametresine göre sonuç istenildiği gibi düzenlenebilir. Örneğin sonucun doksan dokuz olması istenerek Λ parametresi seçilirse, yüz hizalama pozisyonunda gerçekleşmiş mutasyonlardan sadece birinin kabul edileceği sonucu çıkarılır [5]. Yüz hizalama pozisyonunda tek bir nokta mutasyonunun kabul edildiği senaryoya göre üretilen PAM matrisi 1-PAM, üretilen matris henüz skor matrisi değildir, şeklinde gösterilir.

Margaret Dayhoff evrimsel işleyişi Markov süreci ile modellediğinden, 100 hizalama pozisyonunda daha fazla mutasyonun kabul edilebildiği PAM matrislerinin üretilebilmesi için 1-PAM matrisinin kuvvetleri alınabilir [5]. 2-PAM, 1-PAM matrisinin kare alınarak, 3-PAM, 1-PAM matrisinin kübü alınarak elde edilir. 2-PAM matrisi 1-PAM matrisinin üretildiği veri kümesindeki sekansların birbirlerinden ıraksadığı sürenin 1-PAM matrisinin iki katı, 3-PAM matrisi 1-PAM matrisinin

üretildiği veri kümesindeki sekansların birbirinden ıraksadığı sürenin 1-PAM matrisinin üç katı kadar olduğu şeklinde de yorumlanabilir. Yüz hizalama pozisyonunda kabul edilir mutasyon sayısının belirlenmesi, sekans bileşenlerinin tamamının mutasyona uğrayacağı ile ilgili doğrudan bilgi sağlamamaktadır. Yani sekans bileşenlerinden sadece biri tüm mutasyonları barındırıyor olabilir [5].

Şu ana kadar hesaplanan matris bir skor matrisi değil olasılık matrisi idi. PAM matrisinin hizalama skorların belirlenebilmesi amacıyla kullanılması, evrimsel modeli temsil eden Mutation Probability matrisi M ile, rastgele modeli temsil eden ilgili sekans bileşenine ait olasılığın odd-ratio değerinin bulunabilmesi gerekmektedir [5].

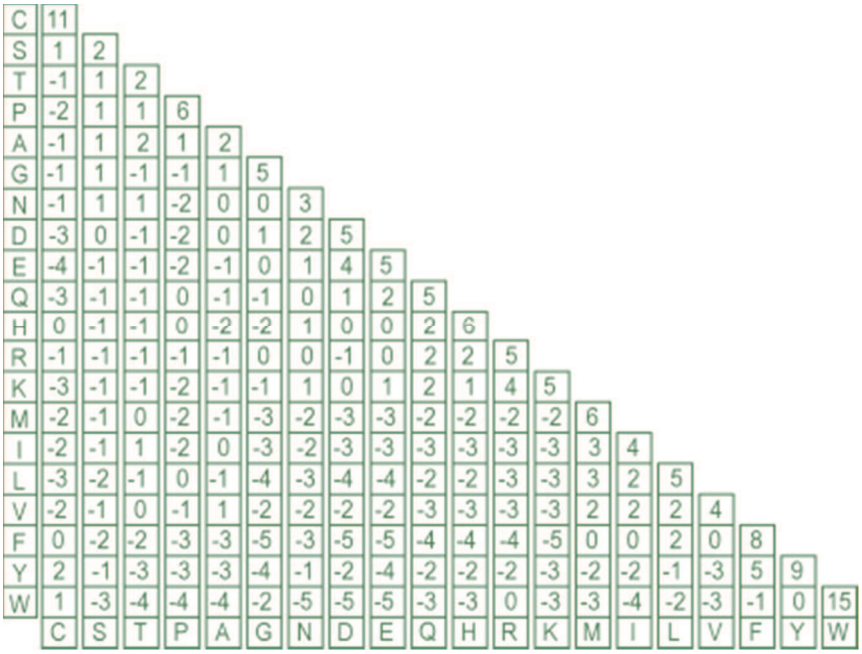
$M_{a,b}$ değeri Mutation Probability matrisinde b tipi sekans bileşeninin, a tipi sekans bileşenine mutasyon sonucu dönüşme olasılığı, f_a değeri ise a tipi sekans bileşeninin sekans popülasyonunda rastlanma olasılığı olmak üzere, b tipi sekans bileşeninin, a tipi sekans bileşenine mutasyon sonucu dönüşmesine verilecek skor değeri $s_{a,b}$ aşağıdaki gibi hesaplanabilir [5].

$$s_{a,b} = 10 \log_{10} \left(\frac{M_{a,b}}{f_a} \right) \tag{3.9}$$

Eşitlik 3.9'da verilen $s_{a,b}$ değerlerinin tüm sekans bileşenleri için düzenlenmesiyle simetrik PAM skor matrisleri düzenlenmiş olur. Elde edilen PAM skor matrisinde yüz hizalama pozisyonunda kabul edilecek mutasyon sayısını belirtmek amacıyla, PAMn notasyonu kullanılmaktadır. Örneğin PAM120, yüz hizalama pozisyonunda yüz yirmi mutasyonun kabulunü gösterir skor matrisini temsil etmektedir.

Margaret Dayhoff 1978 yılında PAM skor matrislerini oluştururken işlem verisini temsil eden protein aileleri ve ailelerdeki sekans sayısındaki sınırlama, Margaret Dayhoff tarafından önerilen skor üretme sistemi kullanılarak 1991 yılında yeni protein aileleri dahil edilerek PAM250 için tekrar düzenlenmiş ve PET91 adıyla duyrulmuştur [5].

Şekil 3.6'da PET91 skor matrisi $2 \log_2$ ile ölçeklenmiş versiyon gösterilmektedir. PAM skor matrisine kıyasla, PET91 skor matrisinde bazı sekans değişimleri negatif değil pozitif skorlar almaktadır.



Şekil 3.6 PET91 skor matrisi [5].

3.2.5 BLOSUM Değişim Matrisi ve BLOSUM Değişim Matrisinin Üretilmesi

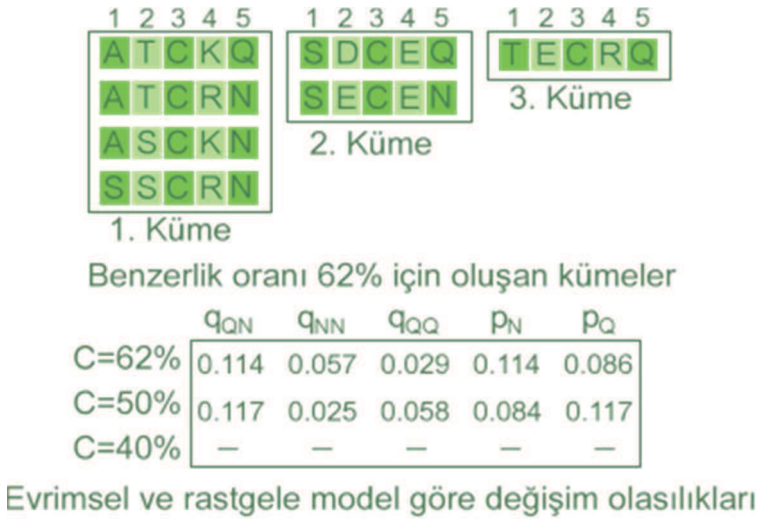
BLOSUM (Block Substitution Matrix) matrisi, Steven ve Jorja Henikoff tarafından, aynı hizalama pozisyonunda çoklu sekans bileşen değişimi barındıran, birbirleri ile akrabalık dereceleri zayıf protein dizilerine ait korunmuş bölgelerin boşluk karakterleri kullanılmadan hizalandığı 1991 yılından itibaren erişilebilir BLOCKS veri tabanı kullanılarak üretilmiş değişim matrisidir [5].

BLOSUM değişim matrislerini, PAM değişim matrislerinden ayıran en önemli özellik, işlem verisi olarak kullanılan protein sekanslarının benzerlik oranları arasındaki farklılıklardır [5]. PAM değişim matrisleri üretilirken Margaret Dayhoff tarafından seçilen protein ailelerindeki sekanslar arasında benzerlik oranı %85 ve altı olarak seçildiğinde tüm sekanslar aynı kümeye dahil edilmektedir [5].

BLOSUM değişim matrisleri üretilirken seçilen sekansların evrimsel homolojileri doğrulanmıştır ancak, akrabalık dereceleri PAM değişim matrisleri için seçilmiş sekanslara kıyasla oldukça zayıftır [5]. BLOSUM değişim matrislerini, PAM değişim matrislerinden ayıran diğer özelliklerden biri de hizalamaların analizi sırasında filogenetik ağaçların üretimine ihtiyaç duyulmamasıdır [5].

PAM değişim matrislerinden de hatırlanacağı üzere, kullanılacak hizalamalardaki sekans bileşen kompozisyonu, protein ailelerinin mutasyon sonucu değişebilirlik miktarları ve sekans uzunluklarının etkisini analiz aşamasında bir dizi işlem yapılmıştır. Benzer bir durum BLOSUM değişim matrisleri üretilirken ağırlıklandırma (weighting) ve kümeleme (clustering) işlemleri ile gerçekleştirilmektedir [5].

Kümeleme aşamasında benzerlik oranları $\%C$ ve üzerinde olan sekanslar aynı gruba dahil edilirken, kümeleme işlemi yapıldıktan sonra oluşturulan hizalanmış sekansların ilgilenilen hizalama pozisyonundaki sekans bileşenine göre, N_{seq} aynı gruptaki sekans sayısı olmak üzere, N_{seq} değerine bağlı ağırlıklandırılması işlemi yapılmaktadır. Böylece hizalanacak sekanslar grup içinde değil, grup ya da kümeler arasında sekanslara verilen ağırlık değerleri de kullanılarak değerlendirilecektir [5]. BLOSUM değişm matrislerinin isimlendirilmesinde, analiz edilecek sekansların kümelenmesi için kullanılacak ve C ile gösterilen benzerlik oranı kullanılmaktadır. C ile gösterilen benzerlik oranı 62 olarak belirlenip oluşturulan BLOSUM değişim matrisi, BLOSUM-62, C ile gösterilen benzerlik oranı 50 olarak belirlenip oluşturulan BLOSUM değişim matrisi, BLOSUM-50 adı ile tanımlanmaktır [5]. Aşağıda C benzerlik oranının üç farklı değeri için 7 sekanstan oluşan ailenin kümeleme örnekleri gösterilmektedir.



Şekil 3.7 Kümelerin oluşturulması ve değişm olasılık değerleri [5].

BLOSUM değişim matrisinin sunduğu evrimsel (non-random) modelde bir sekans bileşenin başka bir sekans bileşenine dönüşüm olasılığı hesabı ve sonrasında odd-ratio işlemi ile BLOSUM değişim matrisinin oluşturulması işleminde farklı C değerlerinin

kullanıldığı örnek uygulama için ayrıntıları Şekil 3.7'de verilen sekansları inceleyebiliriz. Şekil 3.7'de kümeleme için kullanılan C parametresinin 50 olarak alındığı kabulü ile dört, iki ve bir elemanlı üç grubun oluştuğu, her bir grupta ise ağırlıkların sırasıyla $1/4$, $1/2$ son olarakta 1 olarak belirlendiği grupların eleman sayılarına bakıldığında görülebilir [5]. Bu ağırlık değerleri kullanılarak, a tipi sekans bileşeninin b tipi sekans bileşenine ya da b tipi sekans bileşeninin a tipi sekans bileşenine mutasyon sonucu dönüşme olasılığı aşağıdaki gibi hesaplanır [5].

$$q_{a,b} = \frac{f_{a,b}}{\sum_{1 \le a \le b}^{20} f_{a,b}} \tag{3.10}$$

BLOSUM değişim matrisleri oluşturulurken temel yaklaşım, grup içi sekanslar değil, gruplar arası sekanslar dahilinde yapılacak hizalamaların kullanılması durumunu barındırmakta idi. C parametresi 50 olarak alındığında oluşan üç grup için, üç farklı grup çifti oluşturulur. Oluşturulan bu grup çiftlerini kullanarak Q tipi sekans bileşeninin N tipi sekans bileşenine ya da N tipi sekans bileşeninin Q tipi sekans bileşenine mutasyon sonucu dönüşme olasılığını hesaplamak için önce $f_{Q,N}$ değerinin hesaplanması gerekir [5]. Bu değer $\left(\frac{1}{4} x \frac{1}{2}\right) + \left(\frac{3}{4} x \frac{1}{2}\right) + \left(\frac{1}{2}\right) + \left(\frac{3}{4}\right) = \frac{14}{8}$ olarak Q ve N tipi sekans bileşenlerinin üç gruptaki tüm durumlar dikkate alınarak hesaplanmıştır. Üç grup çifti, her grup çiftinde de beş sekans bileşeni hizalanacağından, $q_{Q,N}$ olasılık değeri $\left(\frac{14/8}{15}\right) \cong 0.117$ hesaplanabilir [5]. Hesaplanan $q_{Q,N}$ değeri evrimsel modele göre ilgili sekans bileşenlerinin mutasyon olasılığını verirken, rastgele modele ait ilgili sekans bileşenlerinin mutasyon olasılık değerinin hesaplanması ile BLOSUM değişim matrisinin oluşturulması aşamasına geçilebilir [5]. Rastgele modele ait mutasyon olasılık değerinin hesaplanması için tıpkı PAM değişim matrisinde de kullanıldığı üzere, mutasyona uğrayacak sekans bileşeninin işlem verisindeki frekansı ya da dağılımının bulunması yeterli olacaktır. p_a değeri, rastgele modelde a tipi sekans bileşeninin dağılımını temsil etmektedir, aşağıdaki gibi hesaplanabilir [5].

$$p_a = q_{a,a} + \sum_{a \neq b} \frac{q_{a,b}}{2} \tag{3.11}$$

Evrimsel model ve rastgele modelden elde edilen olasılık değerleri odd-ratio gereğince birbirine oranlanırsa, genelikle odd-ratio değeri $\log_2$ ile ölçeklenmektedir, BLOSUM değişim matrisinin ilgili değişimlere ait atadığı skor değerleri bulunmuş olur.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	0	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-1	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	2	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-2	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

Şekil 3.8 BLOSUM-62 değişim matrisi [5].

3.2.6 COFFEE Skoru ve Hesaplanması

Mevcut skor hesaplama yöntemlerinin evrimsel mekanizmaların tüm özelliklerini modelleyemediği var sayılacak olursa, üretilecek çoklu hizalamaların genel skorlarının iyileştirilmesi çalışmalarının devamı yerine, bu hizalamaların biyolojik olarak geçerli alt bölge ya da alt hizalamalarının tesbit edilebilirliği yönünde uygulamaların düzenlenmesi daha da önemli olabilir [20, 26-29, 33]. Çoklu Hizalama veya ikili hizalamaların biyolojik geçerlilik, uygunluklarının kontrolü amacıyla bir dizi çalışma mevcuttur. COFFEE (Consistency Based Objective Function For Alignment Evalution) skoru ve skor hesaplama yöntemi de bu çalışmaların bir ürünü olarak değerlendirilebilir [29].

COFFEE skorunun temelinde, protein sekanslarının çoklu hizalamalarına ait kalitenin hesaplanacağı matematiksel model farklılığı ile üretilen hizalama ve hizalamayı oluşturan tüm sekansların ikili hizalamaları kullanılarak oluşturulmuş bilinen referans kütüphane arasındaki doğruluk, tutarlılık kontrolünün bulunduğu söylenebilir. COFFEE skoru, elde edilen çoklu hizalamalardaki tüm ikili sekansları, doğruluğu bilinen ve

hizalamada kullanılacak sekansların tamamının oluşturduğu referans ikili hizalamalar ile sekans bileşen pozisyonları üzerinden karşılaştırıldığında yakalanan uygunluk sayesinde, hizalama skoruna ikili hizalamalar temelinde geçerlilik kazandırmış olacaktır [20, 29].

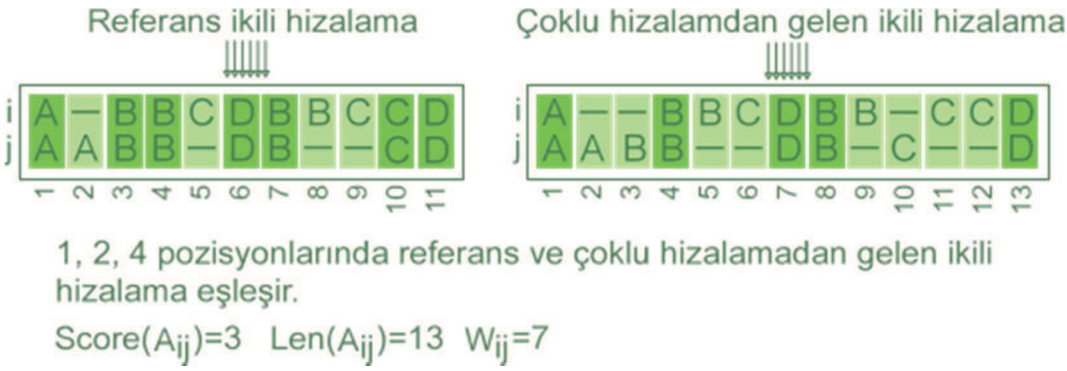
Referans kütüphanenin oluşturulması için farklı yaklaşımlar COFFEE skoru hesaplanırken denenmiş olmasına karşın, en yaygın kullanılan kütüphane oluşturma yöntemi, N adet sekansa sahip çoklu hizalamanın, $(N^2 - N)/2$ adet ikili hizalamasının kullanılması şeklindedir [29]. COFFEE skoru doğrudan referans kütüphaneye bağımlı olmadığından, kütüphaneyi oluşturacak referans ikili hizalamaların üretilmesi için istenilen veya kabul edilen bir uygulama ya da protein veritabanı, ClustalW uygulaması [35], FSSP veritabanı SAGA uygulamasında tercih edilmiştir [26], kullanılabilir [29].

COFFEE skoru hesaplanacak çoklu hizalama ve sekansların tüm ikili hizalamalarını barındıran referans kütüphanenin oluşturulduğunu kabul ederek, COFFEE skorunun hesaplandığı formülasyon; N hizalanacak sekans sayısı, $S_1, S_2 \dots, S_N$ hizalanacak sekanslar, A_{ij}, i ve j sekanslarının ikili hizalaması, $Score(A_{ij})$, A_{ij} ikili hizalamasının referans kütüphanedeki i ve j sekanslarının ikili hizalaması arasında aynı lokasyonda bulunan amino asit çifti sayısını, $Len(A_{ij})$, çoklu hizalamadan alınmış A_{ij} ikili hizalamasının uzunluğunu, W_{ij} ise referans kütüphanedeki i ve j sekanslarının ikili hizalamasına ait benzerlik (identity) değeri olmak üzere, aşağıdaki gibi gösterilebilir [29].

$$COFFEE\ Skoru = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} W_{ij} Score(A_{ij})}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} W_{ij} Len(A_{ij})} \qquad 3.12$$

Eşitlik 3.12'de verilen formülasyon dikkatli incelendiğinde, COFFEE skorunun alabileceği en büyük değerin 1, bu durumda referans kütüphanedeki tüm ilgili ikili hizalamalar ile üretilen çoklu hizalamadaki ilgili ikili hizalamanın tüm sekans bileşen çiftleri aynı pozisyonda bulunmaktadır, en küçük değerinin ise 0, bu durumda referans kütüphanedeki tüm ilgili ikili hizalamalar ile üretilen çoklu hizalamadaki ilgili ikili hizalamanın tüm sekans bileşen çiftleri farklı pozisyonlardadır, değerlerini alabilmektedir.

Farklı çalışmalarda da kullanılmış olmakla beraber, COFFEE skoru hesaplanırken şimdiye kadar açıklanan skor hesaplama sistemlerinden ayrı olarak, işlem uygulanacak çoklu hizalama içindeki ikili hizalamalara ağırlık değerleri, skor hesabında W_{ij} ile gösterilmekte idi, atanmıştır. Bu sistem sayesinde çoklu hizalama içerisinde ya da genel skor üzerinde etkili olacak ikili hizalamalara daha çok ağırlık verilmiş, çoklu hizalamada görülme, korunma olasılığı arttırılmış olur [29]. Aşağıda üretilen çoklu hizalama ile referans kütüphanedeki ilgili ikili hizalamanın karşılaştırılması örneklenmiştir.



Şekil 3.9 COFFEE skorunun hesaplanması işlemi.

Şekil 3.9'da da görülebileceği üzere, referans kütüphanenin ikili hizalamaları ile çoklu hizalamadan üretilecek iki hizalamalar, aynı pozisyonda sekans bileşenleri ya da sekans bileşeni ile boşluk karakterinin eşleşmesi durumunu bir, diğer durumları ise sıfır değeri ile temsil ederek $Score(A_{ij})$ değeri hesaplamaktadır.

3.3. Boşluk Karakterlerinin Kullanımı ve Boşluk Karakterlerinin Cezalandırılması

Evrimsel sürecin iki temel mekanizması olan insersyon (insertion) ve delesyon (deletion) Çoklu Dizi Hizalama Problemi için boşluk ("-", gap) karakterinin belirli pozisyonlara, çoklu hizalamanın skoru maksimum olacak biçimde yeteri sayıda boşluk karakteri seçilmelidir, eklenmesi şeklinde modellenmişti. Eklenecek boşluk karakterine ait ceza, hizalamaya ait skor belirlenirken ya doğrudan boşluk karakterinin değiştirdiği hizalama uzunluğu dikkate alınarak ya da elde edilen genel skordan çoklu hizalamanın her bir sekansındaki boşluk karakterlerine verilen negatif skor çıkartılarak hesaplanabilir [4-7].

Boşluk karakterlerine ait cezanın belirlenmesi için kullanılan yöntemlerden ilki Lineer Boşluk Cezası (Linear Gap Penalty) yöntemi, tek boşluk karakterine atanan cezanın hizalamadaki toplam boşluk karakter sayısıyla çarpılıp genel skordan çıkartılmasına dayanır, matematiksel model, E bir boşluk karakterine verilecek ceza ve n_{gap} hizalamadaki toplam boşluk sayısı olmak üzere, aşağıdaki gibi verilebilir [4-7].

$$g(n_{gap}) = -n_{gap}E \tag{3.12}$$

İnsersyon mekanizması Çoklu Dizi Hizalama Problemi için modellenirken tek bir boşluk karakteri değil ardışık boşluk karakterlerinin eklenmesi şeklinde faaliyet gösterdiği dikkate alındığından tek bir boşluk karakteri ya da bir dizi boşluk karakterinden oluşan sekans bölgelerine ait ceza bu modele bağlı kalınarak tekrar düzenlenebilir. Boşluk karakterlerine ait cezanın belirlenmesi için bu düzenlemeyi barından yöntemlerden en sık kullanılanı Düzgün Boşluk Cezası (Affine Gap Penalty) yöntemidir.

I boşluk açılışına ait ceza (Gap Opening Penalty), E ilk boşluk karakterinden sonra gelen boşluk karakterlerine ait ceza (Gap Extension Penalty), n_{gap} tek bir sekanstaki toplam boşluk karakter sayısı olmak üzere, Düzgün Boşluk Cezası yöntemine için belirlenmiş matematiksel model aşağıdaki gibi verilebilir [4-7].

$$g(n_{gap}) = -I - (n_{gap} - 1)E \tag{3.13}$$

Düzgün Boşluk Cezası yönteminde, boşluk açılış cezası için genellikle 7 ile 15 arasında, boşluk genişletme cezası için ise genellikle 0.5 ile 2 arasında değerler protein sekansları hizalanırken verilmektedir.

Boşluk karakterleri için belirlen cezalar, Düzgün Boşluk Cezası ya da Lineer Boşluk Cezasının önerdiği matematiksel modeli, boşluk karakteri ile hizalanan sekans bileşenine göre E parametresinin değiştirilmesi şeklinde düzenlenebilir [4-7]. Kullanılan skor sistemi boşluk karakterlerini genel skor üzerine yansıtmıyor ise, kullanılan skor sistemine göre uygun boşluk ceza yönteminin seçilememesi, seçilen boşluk ceza yöntemine göre gerekli parametrelerin belirlenememesi, hizalama kalitesinin beklenenden aşağıda çıkmasına sebep olabilir [4-7].

3.4. Çoklu Dizi Hizalama Probleminin Çözümünde Kullanılan Yaklaşımlar

İkincil ve üçüncül yapı tahmini, filogenetik analiz, fonksiyon tahmini ve polimeraz zincir reaksionlarının belirlenmesi gibi Biyoinformatik'in diğer pek çok konusunda da çözümüne ihtiyaç duyulan Çoklu Dizi Hizalama Problemi, hangi skor hesaplama ve boşluk ceza yöntemleri seçilirse seçilsin, ayrık türde bir optimizasyon probleminin bütün çözüm zorluklarını taşımaktadır [4, 5, 10, 17, 18].

Çoklu Dizi Hizalama Problemi'nin oldukça zor yapısı dikkate alındığında, kendi içlerinde bazı kabulleri ve birbirlerine göre bazı avantajları barındıran iki yaklaşım problemin çözümü için kullanılmaktadır [4, 5, 10, 17, 18]. Bu yaklaşımlardan birincisi İkili Dizi Hizalama Problemi'nin çözümüne dayalı dinamik programlama isimli deterministik (deterministic) algoritmik yaklaşım ve tüm sekansların aynı anda hizalanmasını sağlayan Genetik Algoritma (Genetic Algorithm), Isıl İşlem Algoritması (Simulated Annealing Algorithm) gibi algoritmaların kullanıldığı sezgisel (heuristic) yaklaşımlardır.

3.4.1. Çoklu Dizi Hizalama Probleminin Dinamik Programlama Algoritmaları Kullanılarak Çözülmesi

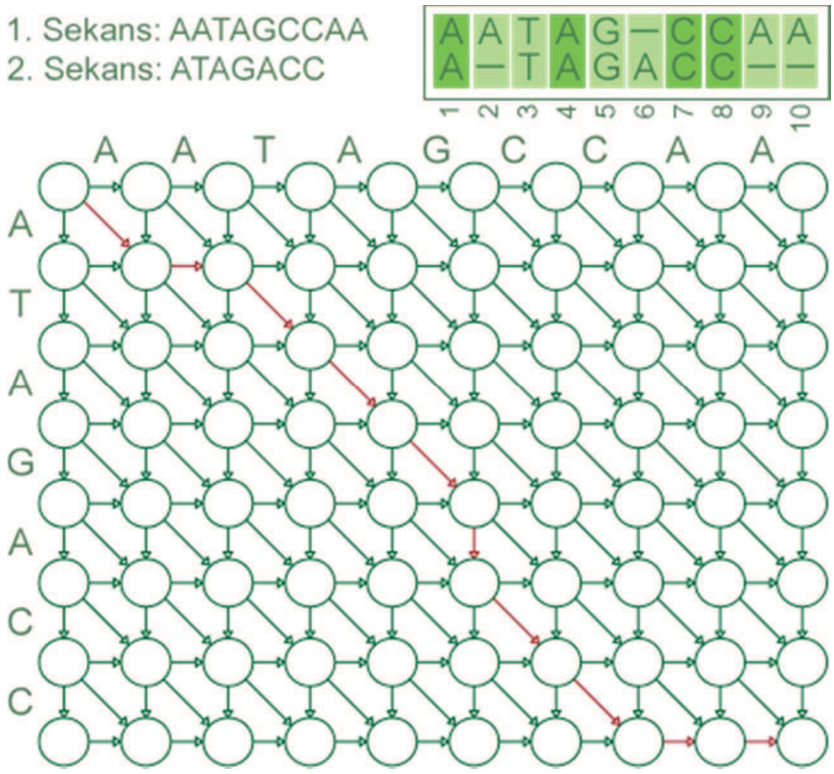
Dinamik programlama algoritmaları ya da yaklaşımları adıyla Biyoinformatik alanında yapılan ilk çalışma Needleman ve Wunsch tarafından 1970 yılında sunulmuş ve Sum Of Pairs skoru kullanılarak elde edilen hizalamaların kaliteleri belirlenmiştir [4, 5, 26, 29].

Eğer problem, çözümleri birbirlerinin bulunmasında kullanılacak alt problemlere ayrıştırılıyor, bu alt problemler de genel çözümü temsil etmek üzere birleştirilebiliyor ise dinamik programlama, bellek kullanımından işlem maliyetine kadar pek çok konuda diğer yaklaşımlara kıyasla daha başarılı olmaktadır [9]. Şayet hizalanacak diziler, yönlü graf düğümlerinden oluşacak şekilde modellenebilir ve her bir düğüme ait kenarlara uygun ağırlık değerleri atanabilir ise, graf veri modeli üzerinde kenar değerleri toplamının minimizasyonu veya maksimizasyonu temelli dolaşımlar dinamik programlama yaklaşımı ile kesinlikle bulunabilir [4, 9].

Needleman ve Wunsch tarafından önerilen yaklaşıma göre, hizalanacak iki sekans, birden fazla giriş ve tek çıkış düğümü olan yönlü graf veri yapısı üzerinden hizalanabilir [4]. Düğümlerin yatay veya düşey doğrultudaki komşuları ile olan

bağlantıları, sekans bileşeni ile boşluk karakterinin eşleşmesi durumuna karşılık gelecek şekilde, köşegen ya da çapraz doğrultudaki komşusu ile olan bağlantısı ise eşleşen sekans bileşenlerinin bulunması durumuna karşılık gelecek şekilde ağırlıklandırılır [4].

Başlangıç düğümünden bir sonraki düğüm gidilmek üzere seçilecek iken, başlangıç düğümü ile seçilecek düğüm arasındaki maliyetin en az olması gerektiği şeklinde yapılacak aç gözlü (greedy) seçim, yol boyunca her işlem adımına dağıtıldığından ve bu işlem adımları birleştirilerek hedefe ulaşıldığından, yönlü graf üzerinde dolaşma işlemi dinamik programlama yaklaşımlarının kullanılabilirliğine olanak tanır [4, 5]. Aşağıda rastgele üretilmiş iki sekans, Needleman ve Wunsch tarafından önerilen yaklaşıma bağlı kalınarak hizalanması örnek olarak hizalanmıştır.



Şekil 3.10 Dinamik Programlama ve yönlü graf düğümleri.

Needleman ve Wunsch tarafından önerilen yaklaşımın ikiden çok sekansın hizalanmasında kullanılabilirliğini kaybetmesi araştırmacıları, dinamik programlama yaklaşımının getirdiği avantajdan farklı modeller türeterek faydalanmaya yönlendirmiştir. 1988 yılında Higgins ve Sharp Pileup isimli uygulamada, yine 1988 yılında Taylor Multal isimli uygulamada, 1994 yılında da Thompson ve arkadaşları şu an yaygın olarak kullanılan CLUSTALW isimli uygulamada dinamik programlama ile

filogenetik ağaçları birleştirerek çoklu hizalamalar üretmişlerdir [19, 26, 27, 29]. Filogenetik ağaç ve Dinamik Programlamanın birlikte kullanılması ile aralarındaki akrabalık en yüksek olan ikili hizalamalar, daha sonra da hizalanmış ikilileri birleştirerek çoklu hizalamalar iteratif olarak elde edilmektedir [26, 27, 29].

Dinamik programlamanın ikili hizalamaları kademli (progressive) birleştirerek çoklu hizalamaları elde etmesi, ikili hizalamalarda yapılan bir hatanın elde edilecek çoklu hizalamaya kadar artarak korunacağı durumunu beraberinde getirmektedir [26, 27, 29]. Dinamik programlama ve filogenetik ağaçların birlikte kullanımından doğacak bu dezavantajın giderilebilmesi amacıyla, çoklu hizalamayı oluşturacak sekansların tamamına ait ikili hizalamaların kullanılılabilirliği 1988 yılında Lipman ve arkadaşları tarafından geliştirilen Multiple Sequence Alignment (MSA) isimli programda gösterilmiştir [26, 27, 29]. Lipman ve arkadaşları tarafından önerilen yaklaşım 10'a kadar sekansın Dinamik Programlama yaklaşımı ile çoklu hizalamalar üretmede kullanılabileceğini göstermesine karşın, 1994 yılında Wang ve Jiang Çoklu Dizi Hizalama Problemini NP-Complete problemler sınıfına dahil ederek alternatif yaklaşımların kullanılabileceğine vurgu yapmıştır [26, 27, 29].

Hizalama probleminin yönlü graf modeli üzerinden dinamik programlama ile çözüme ulaştırılması, hizalanacak dizilerin uzunlukları birbirlerine yakın ise, dizi boyutlarının çarpımları taban olmak üzere ikinci dereceden üstel zaman ve uzay karmaşıklıkları barındırmaktadır [4, 5]. Ancak Şekil 3.10'daki basit örnekte de görüldüğü üzere, uzunlukları 10'u geçmeyen iki dizinin bile yönlü graf modeli üzerinden dinamik programlama ile hizalanması, en iyi sonucun eldesinin garanti edilmesi ve birden fazla en az maliyetli yol bulunma olasılığı dikkate alındığında ikinci dereceden üstel zaman ve uzay karmaşıklıklarını üç veya dördüncü dereceden üstel karmaşıklıklara dönüştürebilir [4].

DNA, RNA ve protein makro moleküllerinin oldukça kompleks yapıları gereği on binler ile ifade edilen dizi uzunluklarına sahip oldukları, çoklu dizi hizalama probleminin üç veya daha fazla diziyi barındırdığı ve evrimsel mekanizmaların çalışma mekanizmalarındaki karmaşıklık göz önüne alındığında dinamik programlama yöntemlerinin bazı sezgisel ya da olasılık tabanlı tekniklerle birleştirilerek yaklaşık hizalamalar üretilmesi gerektiği düşünülmektedir [4, 5, 17-19].

3.4.2. Çoklu Dizi Hizalama Probleminin Sezgisel Algoritmalar Kullanılarak Çözülmesi

Çoklu Dizi Hizalama Problemi'nin çözümü için sunulan algoritmik yaklaşımların, Kesin (exact), İlerlemeli (progressive) ve İteratif (iterative) Algoritmalar olarak belirlenmiş üç gruba ayrılarak değerlendirilmesi mümkündür [28]. Kesin Algoritmalar başlığı altında değerlendirilebilecek algoritmalar, iyi tanımlanmış belirli sınırlamalar dahilinde ilgilenilen sekanslara ait en iyi ya da kabul edilebilir kalitede (sub-optimal) çoklu hizalamanın üretilebileceğini garanti eden yaklaşımlardır. Ancak sekans sayısı, sekans uzunlukları ve evrimsel gelişimi daha iyi modelleyen kalite metriklerinin adaptasyonu hususundaki kısıtlamalar Kesin Algoritmalar grubuna dahil edilmiş yaklaşımların kullanılabilirliğini azaltmaktadır [26, 28, 29].

İlerlemeli Algoritmalar başlığı altında değerlendirilebilecek yaklaşımlar ise genellikle Dinamik Programlama temelli olup, sadece iki sekansın aynı anda hizalanıp, hizalanmış ikilileri birleştirilerek çoklu hizalamaların üretilmesi şeklinde kurgulanmıştır [26, 28, 29]. İkili hizalamaların üretilmesi hususunda Dinamik Programlama Algoritmaları'nın getirdiği avantajlardan faydalanmasına karşın, başlangıçta rastlanan bir hatayı sonuç hizalama elde edilinceye kadar devam ettireceğinden en kaliteli hizalamanın üretileceğini garanti edemeyebilir [26, 28, 29].

İteratif Algoritmalar grubunda ise, çoklu hizalamayı oluşturan sekansların hepsi aynı anda hizalanıp, hizalamaya ait kalite değeri elde edilecek hizalamanın özellikleri dikkate alınarak belirlenmiş durdurma kriteri sağlanıncaya kadar iterasyonlar, boyunca hizalama üzerinde değişiklikler yapılarak geliştirilmeye çalışılır [26, 28, 29]. İteratif Algoritmalar deterministik ya da stokastik yapıda olabilirler. Ancak deterministik yapıda olan İteratif Algoritmalar Kesin ya da İlerlemeli Algoritma üyeleri gibi bazı sınırlamaları beraberinde getirmektedir. Üç gruba dahil algoritmik yaklaşımlar dikkate alındığında, Dizi Hizalama Problemi'nin karmaşıklığı ve hizalamayı oluşturan sekansların ıraksama durumları dikkate alındığında Sezgisel (Stochastic) İteratif Algoritmalar'ın kullanımı, ayrık türde olan Dizi Hizalama Problemine göre modellenmesi oldukça önemli bir çalışma sahası haline gelmiştir.

Sezgisel Algoritmaların Dizi Hizalama Problemi'nde tercih edilebilirliğine ilişkin ilk çalışma Isıl İşlem Algoritması (Simulated Annealing) kullanılarak yapılmıştır [17, 18,

21, 31]. 1988 yılında Myers ve Miller doğrudan Isıl İşlem Algoritması'nı kullanarak, 1993 yılında Isikhawa ve arkadaşları sundukları paralel modeli Isıl İşlem Algoritması ile birleştirerek çalışmalar yapmışlardır [21]. Ancak Isıl İşlem Algoritması'nın ürettiği çoklu hizalamalarda yerel minimumlara sık sık takılması, Isıl İşlem Algoritma'sının doğrudan çoklu hizalamalar üretmede kullanılması yerine, Dinamik Programlama kullanılarak hizalanmış sekansların düzenlenmesi için kullanılması doğrultusunda araştırmaları yönlendirmiştir [26, 29].

Çoklu Dizi Hizalama Problemi'nin çözümünde Sezisel Algoritmalar kullanılarak yapılan çalışmaların genelini, Genetik Algoritma (Genetic Algorithm) ile yapılanlarının oluşturduğu söylenebilir [17-19]. Genetik Algoritma'nın esnek yapısı ve problem özgü uygun şekilde modellenebilirliği oldukça kaliteli hizalamaların üretilmesini sağlamaktadır. 1993 yılında Isakhawa ve arkadaşları, 1997 yılında Zhang ve Wong, 1998 yılında Anbarasu sunduğu paralel yapıda ve yine 1998 yılında Gonzalez ve arkadaşları Genetik Algoritma'yı mutasyon, çaprazlama ve seleksiyon işlem adımları için önerdikleri farklı modeller ile kullanmışlardır [17-19, 23, 24, 25].

1996 yılında Notredame ve Higgins tarafından Genetik Algoritma kullanılarak geliştirilen SAGA (Sequence Alignment By Genetic Algorithm), hem önerdiği skor hesaplama yaklaşımı hem de mutasyon, çaprazlama ve seleksiyon operatörlerinin tamamı için kullandığı yirmi iki farklı senaryoyu adaptif olarak ayarlayarak (dynamic scheduling) oldukça kaliteli hizalamalar elde etmiştir [26, 28, 29]. 1997 yılında Notredame ve Higgins, SAGA ile yakaladıkları başarıyı RNA makro moleküllerinin hizalandığı RAGA (RNA Sequence Alignment by Genetic Algorithm) isimli uygulamayı sunarak devam ettirmişlerdir [27].

Notredame ve arkadaşları, Genetik Algoritma tabanlı SAGA uygulamasına, 1997 yılında çoklu hizalamaların kalitesini ölçtükleri yeni bir metrik ekleyerek, COFFEE (Consistency Based Objective Function For Alignment Evalution) isimli skor ikili hizalamalardan oluşan referans kütüphane kullanarak çalışmalarını sürdürmüşlerdir [26, 33].

Sürü zekâsına dayalı sezgisel algoritmalar Çoklu Dizi Hizalama Problemi'nin çözümü için evrimsel algoritmalar kadar yaygın kullanılmamakla beraber, özellikle son dönemde Parçacık Sürü Algoritması (Particle Swarm Optimization Algorithm), Karınca

Koloni Algoritması (Ant Colony Optimization Algorithm) ve Yapay Arı Koloni Algoritması'nın (Artificial Bee Colony Algorithm) tek başına kullanıldığı çalışmalar mevcuttur [17-19, 30, 31]. Sürü zekasına dayalı sezgisel algoritmaların Çoklu Dizi Hizalama Problemi'nin çözümü için kullanımı daha çok, mevcut Evrimsel Algoritmalar'ın yerel ya da global arama özelliğini arttırma amacıyla oluşturulan hibrit modellerde gözlenmektedir [30, 31].

4. BÖLÜM

YAPAY ARI KOLONİ ALGORİTMASI VE ÇOKLU DİZİ HİZALAMA PROBLEMİ

Çoklu Dizi Hizalama Problemi'nin; sekans sayısı, her bir sekansa ait eleman sayısı ve ayrık optimizasyon problemi olması özellikleri dikkate alındığında, dinamik programlama yöntemleri kullanılarak, yüksek maliyet ve karmaşıklıklara sahip işlemler yaptırmak yerine, yaklaşık çözümlerin iteratif tabanlı olarak optimizasyon algoritmaları ile bulunmasının daha iyi olabileceği düşünülmektedir.

Çoklu Dizi Hizalama Problemleri'ne ait hızlı üretilmiş kaliteli çözümler Biyoinformatik ile ilişkilendirilebilecek pek çok probleme ışık tutacaktır. Evrimsel olarak akrabalık ilişkisi olduğu düşünülen canlılara ait DNA, RNA gibi kalıtsal makro moleküller ile, protein gibi işlevsel makro moleküllerin sahip oldukları uzun dizilerin hizalanmasıyla, filogenetik ağaçların tanımlanmasıdan, gen yapısı ve işlevlerinin belirlenmesine, protein moleküllerinin üç boyutlu yapıları ve görevleri üzerine tahminlerin yapılabilmesinden ilaç üretimine kadar pek çok konuda alternatif çözümlere ulaşılacaktır [4, 5, 17-19].

Bu bölümde Yapay Arı Koloni Algoritması'nın referans aldığı bal arılarının yiyecek ararken gösterdiği zeki davranış, temel işlem adımları ve bu işlem adımlarında ihtiyaç duyulan, duyulacak kontrol parametreleri verilip, algoritmanın nümerik bir optimizasyon probleminin çözümü için başlangıç yiyecek kaynaklarının oluşturulması aşamasında, işçi arı fazı (employed bee phase), gözcü arı fazı (onlooker bee phase) ve kâşif arı fazlarında (scout bee phase) sunduğu matematiksel modellerin, ayrık türdeki Çoklu Dizi Hizalama Problemi'nin çözümü için tekrar düzenlenmesi, düzenlemelerin sunulmuş matematiksel modeller ile uygunluğu konuları üzerinde durulacaktır.

4.1 Arıların Yiyecek Arama Davranışlarındaki Zekilik

Arı kolonisinde, belirli görevlerin yerine getirilmesi adına arılar arasında merkezi bir otoriteye ihtiyaç duyulmaksızın iş bölümünün yapıldığı, arıların kendi kendilerine organize olabildikleri görülmektedir [2, 3]. İş bölümü yapabilme (division of labour) ve kendi kendine organizasyon (self-organization) aynı zamanda sürü zekâsının iki temel özelliğidir [2, 3].

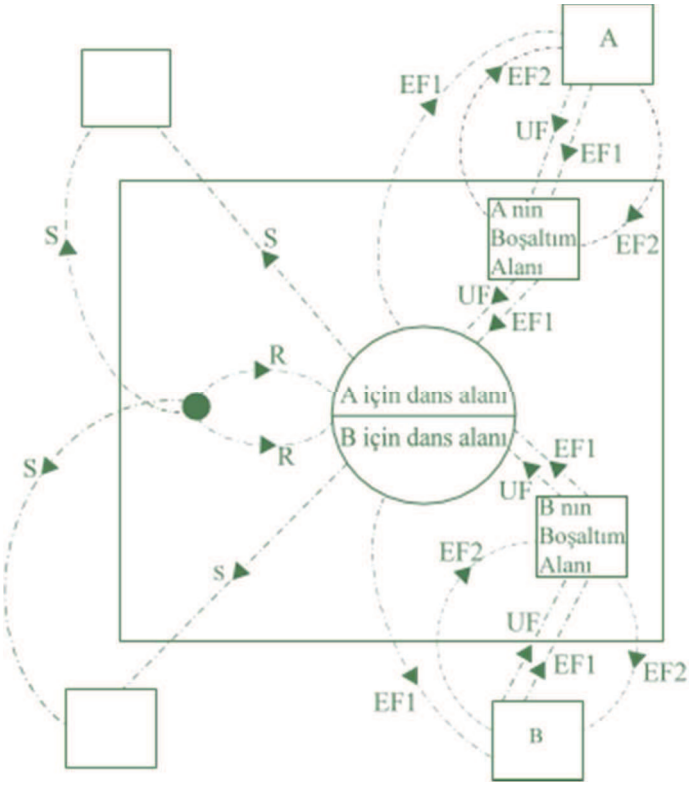
Kaynağın nektar miktarı ile ilgili kalitesi arttıkça kaynağı seçen arı sayısının artmasını sağlayan pozitif geribesleme (positive feedback), nektarı tükenen kaynağın terk edilmesinde etkili negatif geribesleme (negative feedback), arıların yiyecek kaynağı bulabilmek amacıyla rastgele arama yaptığı salınım (fluctuation) ve kaynak ile ilgili bilginin kovanda arılar tarafından paylaşıldığı çoklu etkileşim (multiple interaction) özellikleri, arıların kendi kendine organize olabilmeleri için oldukça önemlidir [2, 3].

İş bölümü yapabilme ve kendi kendine organize olabilme özelliklerini barındıran arı kolonisine ait kolektif zekâyı, arıların nektar veya polen elde etmek için gittikleri ve kalitesi nektar konsantrasyonu, nektarın çıkarılabilirliği ile yuvaya yakınlığı metriklerine bağlanmış yiyecek kaynakları, önceden belirlenmiş kaynaklara giderek nektarın kovana getirilmesine ek olarak kaynak kalitesi ve yeriyle ilgili bilgiyi kovandaki diğer arılara aktarmakla sorumlu görevli işçi arılar ile iç güdüsel veya ortam şartlarına bağlı olarak rastgele yeni kaynaklar arayan kâşif arılar ve görevli işçi arıları takip edip bu arılar tarafından aktarılan bilgiyi kullanarak yeni kaynaklara yönelen gözcü arılardan oluşan görevsiz işçi arıların meydana getirdiği üç bileşenli minimal yiyecek arama modelinde gözlemek mümkündür [2, 3].

Arılar arasında kolektif bilginin oluşması için görevli işçi arılar, buldukları kaynaklar ile ilgili verinin paylaşılması amacıyla, kesin olmamakla birlikte kovanda uçuş yapılan yere yakın bir bölgede, bu bölge dans alanı olarak adlandırılır, dans ederler [2, 3, 11]. Dans alanında ziyaret ettiği kaynağa daha fazla arı yönlendirebilmek için görevli işçi arı dans ederken, görevsiz işçi arılar bu arılara antenlerini kullanıp dokunarak kaynak ile ilgili bilgi toplarlar [2, 3, 11].

Kaynağın kovana olan uzaklığı, kaynağın tadı ve kokusu bilgi aktaran arılar tarafından yapılan farklı danslar ile modellenmiştir [2, 3]. Kaynaklar ile ilgili bütün bilgiler, dans

alanında tüm gözcü arılara görevli işçi arılar tarafından aktarıldıktan sonra; gözcü arılar hangi kaynağı tercih edeceklerine karar verirler.



Şekil 4.1 Yiyecek kaynakları, görevli ve görevsiz arılar [2].

Yukarıdaki şekilde, yiyecek kaynakları, görevli işçi arılar ve görevsiz işçi arıların durumları gösterilmektedir. Kovandaki görevsiz işçi arılardan biri, kaşif arı (S) olup yeni nektar kaynakları aramak üzere rastgele bölgelere gidebilirken, gözcü arı (R) olup A veya B kaynaklarına ait bilgilerin aktarıldığı dans alanlarından A veya B kaynağına yönelebilir. Kovandaki görevli bir arı ise, gittiği kaynağı terk ederek gözcü arı (UF) olabilir, gittiği kaynağa ait bilgiyi dans bölgesinde kovandaki gözcü arılarla paylaşarak (EF1) aynı kaynağa tekrar dönebilir veya gittiği kaynak ile ilgili bilgiyi paylaşmadan aynı kaynağa nektar almak (EF2) için tekrar gidebilir [2, 3].

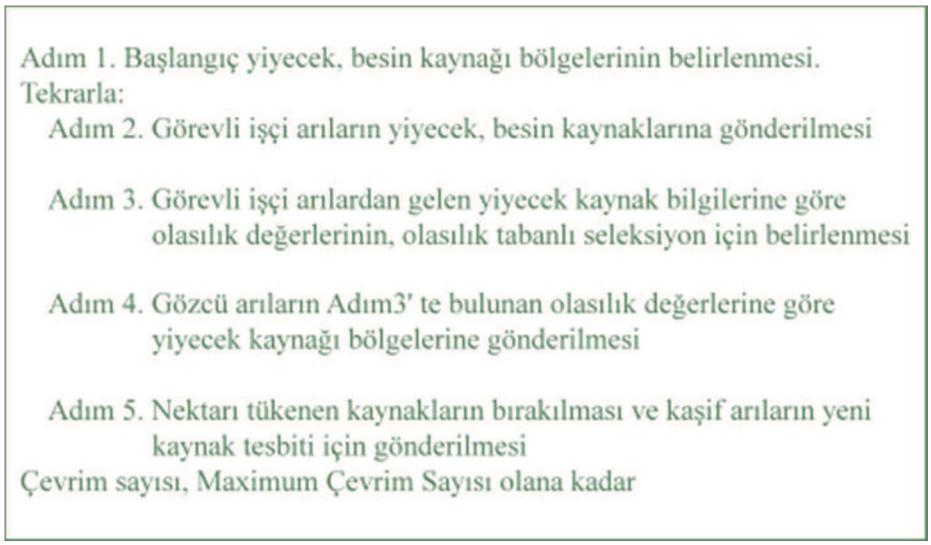
4.2 Yapay Arı Koloni Algoritmasının Temel İşlem Adımları ve Özellikleri

Yapay Arı Koloni (Artificial Bee Colony Algorithm, ABC) Algoritması, 2005 yılında KARABOĞA tarafından sunulmuş, sürü zekası temelli (swarm intelligence), populasyon tabanlı (population based), sezgisel (heuristic) bir optimizasyon algoritmasıdır [2, 3, 11-14, 16]. Nümerik optimizasyon problemlerinin çözümü için geliştirilmiş olmasına karşın sağlam, esnek yapısı ve içerdiği az sayıda kontrol

parametresi ile tez kapsamında ilgilenilen Çoklu Dizi Hizalama Problemi gibi, ayrık türdeki optimizasyon problemlerinin çözümünde de Yapay Arı Koloni Algoritması çok kez kullanılmıştır [11, 15, 16].

Arıların nektar kaynağı ararken gösterdikleri zeki davranışları referans alan Yapay Arı Koloni Algoritması, sürü davranışında gösterilen zekiliği modellerken basitlik amacıyla bir takım kabulleri beraberinde getirmektedir. Yapay Arı Koloni Algoritması, probleme göre farklı olarak seçilebilmekle beraber, her kaynağın nektarının bir görevli işçi arı tarafından alınıyor olması, görevli işçi arı sayınının araştırma uzayındaki toplam kaynak sayısına eşit alınması, kovandaki gözcü arı sayısının görevli işçi arı sayısına eşit olması ve nektarı tükenen kaynakla ilgilenen görevli işçi arının kâşif arı olarak devam edeceği gibi kabulleri barındırmaktadır [2, 3].

Yapay Arı Koloni Algoritması bir optimizasyon probleminin çözümünde kullanılırken besin kaynaklarının yerlerini ilgilenilen optimizasyon probleminin olası çözümleri ile, kaynakların nektar miktarlarını da kaynakla modellenmiş çözümlerin kalite, maliyet değerlerine eşleştirmektedir. Sonuç olarak Yapay Arı Koloni Algoritması, en fazla nektara sahip besin kaynağını diğer bir ifadeyle ilgilenilen optimizasyon probleminin araştırma uzayındaki minimum ya da maksimum değerini temsil eden çözüm veya çözümleri bulmaya çalışmaktadır [2, 3, 11]. Yapay Arı Koloni Algoritması'nın modellediği zeki davranış ve algoritmanın basitliği açısından yapılan kabul durumları da göz önüne alınarak, temel işlem adımları aşağıdaki gibi verilebilir.



Şekil 4.2 Yapay Arı Koloni Algoritmasının Temel İşlem Adımları.

4.3 Başlangıç Yiyecek Kaynaklarının Çoklu Dizi Hizalama Problemine Göre Üretilmesi

Yapay Arı Koloni Algoritmasının temel adımlarının ilkinde de belirtildiği üzere başlangıç yiyecek, besin kaynak bölgelerinin belirlenmesi adına kâşif arılar, kovan çevresinde rastgele arama yaparlar. Yani, Yapay Arı Koloni Algoritması ilgilenilen optimizasyon probleminin çözümlerine karşılık gelen, rastgele yiyecek kaynaklarının üretilmesi ile çalışmaya başlamaktadır [2, 3, 11].

Nümerik bir optimizasyon problemi için, $i = 1, \dots, SN$ ve SN yiyecek kaynak sayısı, $j = 1, \dots, D$ ve D optimize edilecek parametre sayısı, x_j^{max} j. parametrenin üst sınırı, x_j^{min} ise j. parametrenin alt sınırı olmak üzere, bir parametrenin alt ve üst sınırları kullanılarak, bu alt ve üst sınırlar arasında rastgele üretilecek yiyecek kaynakları Eşitlik 4.1 ile formülize edilmiştir [2, 3, 11].

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \tag{4.1}$$

Çoklu Dizi Hizalama Problemi, ayrık türde bir optimizasyon problemi olduğundan Yapay Arı Koloni Algoritması'na özgü matematiksel ifadelerin Çoklu Dizi Hizalama Problemine göre yeniden düzenlenmesi gerekmektedir. Yapay Arı Koloni Algoritması Çoklu Dizi Hizalama Problemi'nin çözümü için kullanılırken yiyecek kaynaklarının yerleri olası hizalamalar, yiyecek kaynaklarının nektar miktarları da hizalamaların kalite ya da uygunluk değerleri ile eşleştirilecektir.

Çoklu hizalamalar genellikle uzunlukları eklenecek boşluk karakter ya da karakterleri ile eşitlenen, amino asit, DNA veya RNA sekansları içeren iki boyutlu matris formunda gösterilmektedir [17-19]. Yeni türetilecek iki boyutlu matris formundaki hizalamaların uzunlukları, hizalamadaki en büyük sekans ya da dizi uzunluğunun 1.2 ile 2 katı arasında değişecek ve yeteri miktarda boşluk karakterinin orijinal sekanslara eklenmesi ile oluşturulacaktır [17-19, 33].

Yeteri kadar boşluk karakterinin hizalamayı oluşturan sekanslara eklenme aşaması, bu boşluk karakteri ya da karakterlerinin orijinal sekansta hangi pozisyonlara yerleştirileceği, teker teker ya da bloklar halinde eklenip eklenmeyeceğine göre farklılıklar göstermektedir.

Yapay Arı Koloni Algoritması ile çözülürken, yiyecek kaynaklarını temsil ediyor olan çoklu hizalamaların üretilmesindeki en yaygın yöntem, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki bloklar halinde, sekansın baş ve sonuna (sequence terminal) eklenmesi şeklindedir [17-19, 26-29]. Aşağıda, boşluk karakterlerinin bloklar halinde, sekansın baş ve sonuna eklenmesi ile oluşturulmuş hizalama örnekleri gösterilmektedir.

Şekil 4.3 Terminal boşluk karakterlerinin eklenmesi.

Üyeleri sırasıyla çoklu hizalamalar ile eşleştirilen başlangıç popülasyonunun, boşluk karakterlerinin sekanslar üzerine nasıl dağıtılarak oluşturulduğu, iterasyonlar (çevrim sayısı) boyunca bireylerin gelişimi için önemli olabilir. Orijinal sekansların benzerlik oranları, gerçekleşmiş mutasyon etkisinin bir dizi boşluk karakteri barındırma durumu ve ardışık boşluk karakterinin genel kalite üzerinde daha az ceza etkisi, bloklar halinde boşluk karakterlerinin dağıtıldığı stratejilerin önemini arttırmaktadır. Aşağıda pozisyon ile sayıları rastgele belirlenmiş üç blok halinde boşluk karakterlerinin dağıtıldığı çoklu hizalama popülasyonu gösterilmektedir.

Şekil 4.4 Terminal ve rastgele belirlenmiş pozisyona boşluk karakterlerinin eklenmesi.

Boşluk karakterlerinin sayıları rastgele belirlenmiş iki blok halinde terminallere eklendiği durum, pozisyon ile sayıları rastgele belirlenmiş üç blok halinde boşluk karakterlerinin dağıtıldığı durum ve her birinin pozisyonu rastgele belirlenmiş tek boşluk karakterinin dağıtıldığı durumların tamamının kullanıldığı başlangıç popülasyonları araştırma uzayını daha iyi tarıyor olabilir. Bu etki dikkate alarak Yapay

Arı Koloni Algoritması'nın Çoklu Dizi Hizalama Problemi'nin çözümü için kullanılan modelinde başlangıç yiyecek kaynaklarının yerlerini temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 1.2 veya 1.5 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği durum ile seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin pozisyon ile sayıları rastgele belirlenmiş üç blok halinde dağıtıldığı durum birlikte kullanılarak oluşturulmuştur.

4.4 Görevli Arıların Yiyecek Kaynak Bölgelerinin Çoklu Dizi Hizalama Problemine Göre Belirlenmesi

Yapay Arı Koloni Algoritması'nda kolaylık sağlamak amacıyla her nektar kaynağı tek bir görevli işçi arı ilişkilendirmektedir, sonuç olarak kovandaki görevli işçi arı sayısı ile nektar kaynak sayılarını eşit alınmaktadır ki bu durum her zaman geçerli olmayabilir. Görevli işçi arılar, nektar topladıkları kaynaktan daha kaliteli kaynak bulmak adına, çalıştığı yiyecek kaynağı komşuluğunda yeni bir kaynak belirleyip, bu kaynağın kalitesini mevcut kaynağın kalitesi ile karşılaştırır [2, 3, 11-13]. Komşuluk tabanlı üretilen bu yeni kaynak, mevcut kaynaktan daha kaliteli ise, görevli işçi arı artık yeni kaynağı kovana nektar taşımak için kullanır ve yeni kaynağı hafızasına alır. Görevli işçi arının yeni kaynağı, daha önce çalıştığı kaynak komşuluğunda üretebilmesi için nümerik optimizasyon problemlerinde kullanılacak matematiksel benzetimi Eşitlik 4.2'de tanımlanmıştır [2, 4, 11-13].

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \tag{4.2}$$

Eşitlik 4.2'ye göre, D adet parametreye sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} parametresi, diğer yiyecek kaynakları arasından rastgele seçilmiş ve yine D parametreli x_k çözümünün x_{kj} parametresi ile $[-1, 1]$ aralığında rastgele değer alan $\emptyset_{ij}$ sayısı kullanılarak değiştirilmekte, x_i komşuluğunda v_i çözümü üretilmektedir. x_{ij} ve x_{kj} değerleri arasındaki fark azaldıkça, x_{ij} parametresindeki değişim miktarı da azalacak, bölgesel optimal çözüme yaklaşıldıkça değişim miktarı adaptif olarak düzenlenmiş olacaktır [2, 3, 11-14]. Üretilmiş v_{ij} parametre değerinin, j.

parametreye ait alt ve üst sınırları aşıp aşmadığının kontrolü yapılarak komşuluk tabanlı v_i parametre vektörü üretilmiş olur [2, 3].

$$v_{ij} = \begin{cases} x_j^{min} & , \quad v_{ij} < x_j^{min} \\ v_{ij} & , \quad x_j^{min} \leq v_{ij} \leq x_j^{max} \\ x_j^{max} & , \quad v_{ij} > x_j^{max} \end{cases} \tag{4.3}$$

x_i çözümü vektörü ve x_i çözüm vektörü komşuluğunda üretilen v_i çözüm vektörü arasında, çözümlerin maliyet ya da kaliteleri hesaplandıktan sonra, maliyet veya kalitelerine göre belirlenmiş uygunluk (fitness) değerlerine göre aç gözlü (greedy) seleksiyon işlemi gerçekleştirilir [2, 3]. f_i, v_i çözüm vektörünün maliyeti olmak üzere, v_i çözüm vektörünün kalite uygunluk değeri aşağıdaki gibi hesaplanır.

$$fitness_i = \begin{cases} 1/(1+f_i) & , \quad f_i \geq 0 \\ 1 + abs(f_i) & , \quad f_i < 0 \end{cases} \tag{4.4}$$

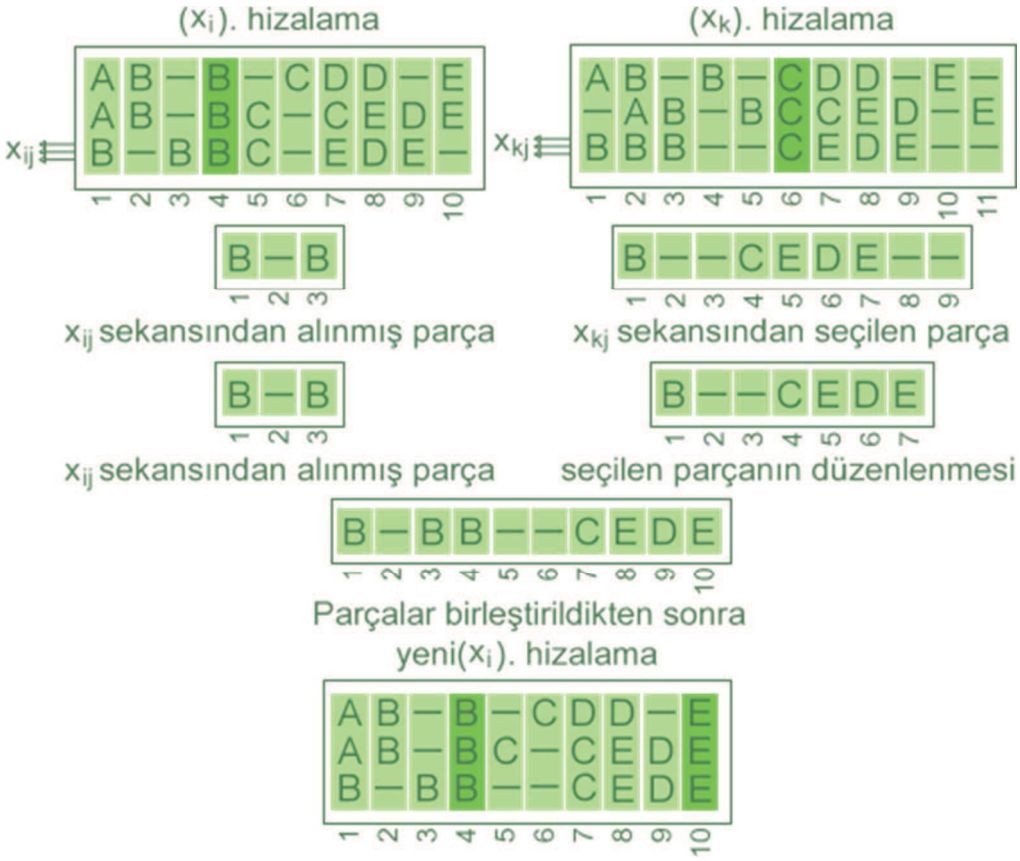
Özetle yeni bulunan v_i çözümü daha yüksek kaliteli bir çözüm ise, işçi arı eski kaynağa ait bilgilerini v_i kaynağına ait bilgiler ile günceller. v_i çözümü x_i çözümüne kıyasla daha kaliteli değil ise, işçi arı x_i kaynağını kullanmaya devam eder ve x_i kaynağının daha kaliteli bir kaynakla geliştirilemediğini gösteren sayaç, sayaç daha kaliteli bir kaynak ile karşılaşıldığında ya da kaynak bırakılmak istendiğinde sıfırlanmaktadır, bir arttırılmaktadır [2, 3, 11-14].

Çoklu Dizi Hizalama Problemi'nin Yapay Arı Koloni Algoritması'yla çözüldüğü uygulamada, görevli işçi arıların yiyecek kaynak bölgelerine gönderilmesi, problemin ayrık türde optimizasyon problemi olması sebebiyle, Eşitlik 4.2'deki temel matematiksel modele bağlı kalınarak değiştirilmeye çalışılmıştır.

Çoklu Dizi Hizalama Problemine göre işçi arıların yiyecek kaynak bölgelerine gönderildiği modelde, i. çoklu hizalamayı temsil eden ve her birinin uzunluğu L_i olan D adet sekansa sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} sekansı rastgele bir pozisyondan bölünerek alınan ilk parçasının, diğer çoklu hizalamalar arasından rastgele seçilmiş her birinin uzunluğu L_k olan D adet sekansa sahip x_k çözümünün x_{kj} sekansının uygun parçasıyla birleştirilmesiyle, x_{kj} sekansının uygun parçası x_{ij} sekansının rastgele bir pozisyondan bölünerek alınan ilk parçası ile birleştirilirken yeni sekansın uzunluğu L_i yapılmak adına yeteri kadar boşluk karakteri

x_{kj} sekansının uygun parçasının rastgele pozisyonlarına eklenmekte veya rastgele pozisyonlarından silinmektedir, yine herbirinin uzunluğu L_i olan yeni bir çoklu hizalama oluşturulmaktadır.

x_{ij} sekansının rastgele bir pozisyondan bölünerek alınan ilk parçasının x_{kj} sekansının uygun parçasıyla birleştirilmesi aşamasında, bu iki parçanın çoklu hizalama üretilirken referans alınan j. sekansın bütün elemanlarını barındırıyor olması kontrol edilmekte, ihtiyaç duyulursa x_{kj} sekansının uygun parçasının rastgele pozisyonlarına boşluk karakter ya da karakterlerinin eklenmesi, rastgele pozisyonlarından boşluk karakter ya da karakterlerinin silinmesi işlemleri gerçekleştirilmektedir. Aşağıda Çoklu Dizi Hizalama Problemine göre işçi arıların yiyecek kaynak bölgelerine gönderildiği durum komşuluk tabanlı yeni çoklu hizalama üretilirken örneklenmiştir.



Şekil 4.5 Önerilen modele göre yeni çözümün oluşturulması.

Komşuluk tabanlı çözümün Çoklu Dizi Hizalama Problemi'ne göre üretildiği durumda x_{ij} sekansının rastgele bir pozisyondan bölünerek ilk parçasının alınmasıyla, x_i çoklu hizalamasının x_{ij} sekansına ait bilgi doğrudan kullanılmış, diğer çoklu hizalamalar

arasından rastgele seçilmiş x_k çözümünün x_{kj} sekansının uygun parçasının alınmasıyla x_{ij} ve x_{kj} sekansları arasında fark alma işlemi gerçekleştirilmiş, ihtiyaç duyulursa x_{kj} sekansının uygun parçasının rastgele pozisyonlarına boşluk karakter ya da karakterlerinin eklenmesi, rastgele pozisyonlarından boşluk karakter ya da karakterlerinin silinmesi işlemleri ile de $[-1, 1]$ aralığında rastgele değer alan $\emptyset_{ij}$ parametresinin etkisi yeni çözüme dahil edilmiş olur.

4.5 Gözcü Arıların Yiyecek Kaynak Bölgelerini Çoklu Dizi Hizalama Problemine Göre Seçmeleri

Görevli işçi arılar, nektar kaynakları ile ilgili sahip oldukları bilgiyi, görevli işçi arıları takip edip bu arılar tarafından aktarılan bilgiyi kullanarak yeni kaynaklara yönelen gözcü arılara, kaynağın kovana olan uzaklığı, kaynağın tadı ve kokusuna göre değişen danslar ile aktarırlar. Gözcü arılar dans alanında paylaşılan bilgilerden, kaynakların nektar miktarı ile orantılı olasılıksal bir yaklaşımla seçim yaparak yeni yiyecek kaynak bölgesi belirlerler [2, 3].

Yapay Arı Koloni Algoritması, gözcü arıların olasılık tabanlı seçme işlemini, yiyecek, nektar kaynaklarının uygunluk (fitness) değerlerini ve rulet tekerleği (roulette wheel) seleksiyon şemasını kullanarak, rulet tekerleği, sıralamaya dayalı, stokastik örnekleme veya turnuva yöntemleri örnek seleksiyon şemalarındandır, sürü zekasının en önemli bileşenlerinden biri çoklu etkileşim özelliğini algoritmaya dahil etmiş olur [2]. $fitness_i$, $i.$ kaynağın kalitesini, SN görevli arı sayısını göstermek üzere, gözcü arıların $i.$ kaynağı seçerek bu kaynağa ait bilgiyi kullanma olasılığı p_i aşağıdaki gibi hesaplanmaktadır [2].

$$p_i = \frac{fitness_i}{\sum_j^{SN} fitness_j} \tag{4.5}$$

İlgilenilen kaynağın uygunluk değerinin, tüm kaynakların uygunluk değerleri toplamına bölünmesiyle hesaplanan kaynak seçim olasılık değeri, kaynak uygunluk değeri arttıkça bu kaynaktan faydalanan gözcü arı sayısını da arttıracaktır. Yapay Arı Koloni Algoritması, daha çok gözcü arının kaliteli nektar, yiyecek kaynağına ait bilgiden faydalanmasını sağlayarak, doğal arı kolonisinde gözlenen ve sürü zekasının bileşenlerinden olan pozitif geribesleme özelliğini de içermektedir [2, 3].

Rulet tekerleği seleksiyon şemasında kullanılacak bütün kaynaklara ait seçim olasılık değeri hesaplandıktan sonra, her bir kaynak için $[0, 1]$ aralığında değişen rastgele sayı üretilir. Kaynağın seçilim olasılık değeri, kaynak için $[0, 1]$ aralığında rastgele belirlenen sayıdan büyükse, gözcü arılardan biri bu kaynak bilgisini kullanıp komşuluk tabanlı yeni bir besin, nektar kaynağı belirleyerek görevli arıya dönüşür [2, 3, 11-14].

Gözcü arının, görevli arıya dönüşerek bulduğu yeni kaynak, bilgisinden faydalanılan, eski kaynaktan daha kaliteli ise, yeni kaynak bilgisinden faydalanılan kaynak ile değiştirilir ve çözümün geliştirilip geliştirilemediğinin kontrol edildiği sayaç sıfırlanır [2, 3, 11-14]. Aksi durumda eski çözüm korunur, bu çözümün geliştirilip geliştirilemediğinin kontrol edildiği sayaç bir arttırılır. Bu işlemler, kovandaki tüm gözcü arılar bir kaynağa gönderilene kadar, gözcü arıların sayısı görevli arıların sayısına eşit alınmakta idi, devam eder [2, 3, 11-14].

Kovandaki görevli ve gözcü arıların, kaynakların geliştirilmeye çalışıldığı arama süreçleri tamamlandıktan sonra, kaynakların geliştirilip geliştirilemediğinin kontrol edildiği sayaç, çözüm geliştirememe sayaç ($failure_i$) değerleri belirli bir eşik üzerinde olan kaynaklara artık görevli arı gönderilmez, kaynak terk edilmiş olur. Terkedilen kaynağın görevli arısı artık kâşif arı haline gelmiş ve rastgele çözüm, kaynak arama süreci başlatılmış olur. Kaynağın bırakılıp bırakılmayacağının kontrol edildiği geliştirememe sayacının eşik değeri, Yapay Arı Koloni Algoritması'nın önemli bir kontrol parametresidir ve $limit$ olarak adlandırılır.

Temel Yapay Arı Koloni Algoritmasında, her çevrimde eşik değeri en çok aşılmış kaynağın görevli arısı yani tek bir arı kâşıf arı olarak rastgele kaynak arama sürecine dahil edilmektedir. Yapay Arı Koloni Algoritması, nektarı tüketilen kaynaktan sorumlu görevli arının kâşif arıya dönüşmesi ile küresel araştırma, görevli ve gözcü arılar aracılığı ile de bölgesel araştırma özelliklerini dengeli şekilde harmanlamış olur [2, 3, 11-14].

Önerilen Yapay Arı Koloni Algoritması'nda ise, $limit$ değeri için belirlenen eşiği aşan her görevli işçi arı kâşif arıya dönüşüp başlangıç yiyecek kaynaklarının oluşturulduğu çoklu hizalama üretme yaklaşımını kullanarak, yani seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği

durum ile seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin pozisyon ile sayıları rastgele belirlenmiş üç blok halinde dağıtıldığı durum arasında olasılık tabanılı seçim yapılarak yeni bir hizalama üretilmekte idi, bulunan yeni hizalama eski hizalamadan daha kaliteli ise bu hizalama başlangıç değeri için sıfır atanmış $limit$ değeri ile, değil ise önceki hizalama bu hizalama için belirlenmiş $limit$ değeri sıfır yapılarak geliştirilmeye devam edilmektedir. $limit$ değeri için eşiğini en çok aşan görevli işçi arının kâşif arıya dönüşmesi yaklaşımı, önerilen Yapay Arı Koloni Algoritması'nın kâşif arı fazında $limit$ değeri için eşiği aşan her görevli işçi arının kâşif arıya dönüşmesi modeli kullanılarak [32] biraz daha esnek yorumlanmıştır.

5. BÖLÜM

UYGULAMA

Yapay Arı Koloni Algoritması'nın işlem adımlarının, Çoklu Dizi Hizalama Problemi için önerilen modeli kullanılarak BAliBASE (Benchmark Alignment Database) ve 3D_ali protein veri tabanlarından seçilmiş, bu veri tabanları seçilen protein aileleri için referasns, yapısal çoklu hizalamaları da içermektedir, protein aileleri kullanılarak üretilen çoklu hizalamaların COFFEE (Consistency Based Objective Function For Alignment Evaluation) skorları üzerinden SAGA (Sequence Alignment With Genetic Algorithm) ve CLUSTALW programları, SPS (Relative Sum Of Pairs) skorları üzerinden daha önce önerilmiş Yapay Arı Koloni Algoritması, Genetik Algoritma (Genetic Algorithm) ve Parçacık Sürü Algoritmaları (Particle Swarm Optimization Algorithm) ile karşılaştırmalı sonuçları bu bölümde verilmektedir.

Çoklu dizi hizalama probleminin Yapay Arı Koloni Algoritması kullanılarak çözüleceği uygulamanın profil sonuçlarına bakılarak, işçi (employed bee phase), gözcü (onlooker bee phase) ya da izci (scout bee phase) arı fazlarının modellendiği fonksiyonlar ile kalite değerinin hesaplandığı fonksiyonun, Nvidia ekran kartları ile ilgili 2006 yılının sonlarına doğru duyrulan görece yeni paralel hesaplama mimarisi CUDA (Compute Unified Device Architecture) platformu hakkında giriş seviyesinde bilgi verilip yine bu platforma aktarılabilirliği üzerinde yine bu bölümde durulmaktadır.

Elde edilen profil sonuçlarına göre paralelleştirilebilirliği en yüksek fonksiyonlardan birinin seçilmesi işlemi ile, bu fonksiyon hizalamaların kalite değerlerinin hesaplandığı fonksiyondur, Tesla M2050, Tesla M2090, Tesla K10, Tesla K20 ve Tesla K20X profesyonel kartlar ile giriş seviyesinde CUDA destekli GeForce GT 430 kartı üzerinde farklı paralelleştirme senaryolarının sonuçları karşılaştırmalı olarak incelenecektir.

5.1 Yapay Arı Koloni Algoritması İle BAliBASE Veritabanından Alınan Sekansların Hizalanması

Yapay Arı Koloni Algoritması'nın Çoklu Dizi Hizalama Problemi'nin çözümü için kullanılan modelinde başlangıç besin kaynaklarının yerlerini temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 1.2 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği durum ile seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin pozisyon ile sayıları rastgele belirlenmiş üç blok halinde dağıtıldığı durum birlikte kullanılarak oluşturulmuştur.

Gözcü arıların yiyecek kaynaklarına gönderilmesi için i. çoklu hizalamayı temsil eden ve her birinin uzunluğu L_i olan D adet sekansa sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} sekansı rastgele bir pozisyondan bölünerek alınan ilk parçasının, diğer çoklu hizalamalar arasından rastgele seçilmiş her birinin uzunluğu L_k olan D adet sekansa sahip x_k çözümünün x_{kj} sekansının uygun parçasıyla birleştirilmesiyle, x_{kj} sekansının uygun parçası x_{ij} sekansının rastgele bir pozisyondan bölünerek alınan ilk parçası ile birleştirilirken yeni sekansın uzunluğu L_i yapılmak adına yeteri kadar boşluk karakteri x_{kj} sekansının uygun parçasının rastgele pozisyonlarına eklenmekte veya rastgele pozisyonlarından silinmektedir, yine herbirinin uzunluğu L_i olan yeni bir çoklu hizalama oluşturulurken kullanılan durum seçilmiş, çoklu hizalamaların kalite değerlerinin belirlenmesi için de Relative Sum Of Pairs (SPS) skor hesaplama yöntemi kullanılmıştır.

Yapay Arı Koloni Algoritması'nın temel işlem adımlarına ilişkin düzenlemelerin yukarıdaki gibi yapıldığı uygulamanın test edilmesi amacı ile BAliBASE protein veritabanından 9 farklı protein ailesi, BAliBASE mevcut protein aileleri ile ilgili referans çoklu hizalamalarıda da barındırmaktadır, seçilmiştir [32-34]. BAliBASE protein veritabanından seçilen protein aileleri, protein ailelerini oluşturan farklı türlere ait amino asit dizi, sekansların uzunlukları ve seçilen protein ailesinin kendi içindeki benzerlik oranlarına göre Kısa, Orta ve Uzun olmak üzere üç gruba ayrılmıştır. Seçilmiş protein ailelerinin sekans uzunluğu, sekans sayısı ve benzerlik metriklerine göre

gruplandırılarak seçilmesi, algoritmaların performans değerlendirme aşamasında faydalı olabilir [26, 27, 32]. Tablo 5.1'de Yapay Arı Koloni Algoritması'nın test sürecinde kullanılacak BAliBASE protein veritabanından alınmış protein aileleri üç grup halinde gösterilmektedir.

Tablo 5.1. BAliBASE protein veritabanından seçilmiş protein ailelerinin özellikleri [32, 34]

Sekans Tipi	Protein Ailesi	Sekans Sayısı	En Kısa-Uzun Sekans	Benzerlik
Kısa	SH3	5	49 - 80	<25%
	Cytochrome C	5	70 - 87	20% - 40%
	Serine Protease	5	66 - 82	>35%
Orta	Protein Kinase	5	263 - 276	<25%
	Anthranilate Isomerase	4	247 - 259	20% - 40%
	Serine Protease	5	222 - 245	>35%
Uzun	Aminotransferase	4	358 - 387	<25%
	Glutamyl-TRNA Synthetase	5	438 - 486	20% - 40%
	Taq DNA Polymerase	5	806 - 928	>35%

SH3 için 1aboA, Cytochrome C için 451c, Serine Protease için 1krn, Protein Kinase için kinase, Anthranilate Isomerase için 1pii, Serine Protease için 1thm, Aminotransferase için 1ajsA, Glutamly-TRNA Synthetase için glg ve Taq DNA Polymerase için 1taq kısaltmaları BAliBASE protein veritabanında kullanılmaktadır [32, 34]. Test edilecek protein ailelerindeki sekans sayısı, her bir sekansın uzunluğu ve aile içi ortalama benzerlik, Yapay Arı Koloni Algoritmasının problemin çözümü için seçilecek uygun çevrim sayısı, kolonideki işçi, gözcü, kaşif arı oranları, kaynağın bırakılıp bırakılmayacağı kararının verildiği *Limit* parametresinin seçiminde önem teşkil etmektedir. Yapay Arı Koloni Algoritması'nın yukarıda önerilen işlem adımları için seçilmiş parametreleri aşağıdaki tabloda [32] referans alınarak gösterilmektedir.

Tablo 5.2. Yapay Arı Koloni Algoritması'nın BAliBASE veritabanından alınan protein aileleri için parametre değerleri [32]

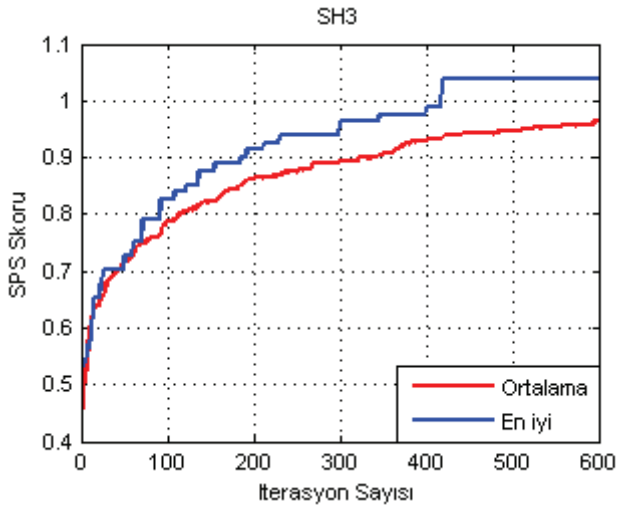
Sekans Tipi	Limit Değeri	Popülasyon Büyüklüğü	İterasyon Sayısı
Kısa	5	20	600
Orta	10	20	1000
Uzun	10	20	1200

BAliBASE protein veritabanından seçilen her bir protein ailesi, Kısa, Orta veya Uzun gruplarına üyeliklerine göre, sırasıyla 600, 1000 ve 1200 iterasyon 10 işçi ve 10 gözcü arıdan oluşan başlangıç popülasyonu kullanılıp, *Limit* değerleri Kısa grubuna dahil aileler için 5, Orta ve Uzun grublarına dahil aileler için 10'u aşan her hizalama için bir kaşif arı üretilerek 10 farklı kez Yapay Arı Koloni Algoritmasıyla hizalanmıştır.

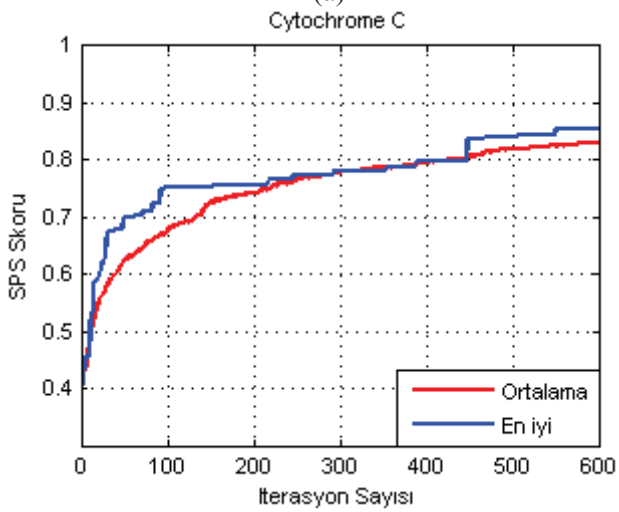
Tüm koşmaların her iterasyonda en kaliteli çoklu hizalama skorları kaydedilmiş, en kaliteli çoklu hizalamayı üreten koşmadan elde edilen gelişim, yakınsama eğrisi ile tüm koşmalarda elde edilen değerlerin ortalamaları alınarak elde edilen gelişim, yakınsama eğrisi çizdirilmiş, 10 farklı koşmaya ait en iyi değerlerin ortalama, en yüksek, en düşük skor değerleri ve 10 farklı koşmaya ait en iyi değerlerin standart sapması Tablo 5.3'te protein ailelerinin tam isimleri kullanılarak aşağıda gösterilmiştir.

Tablo 5.3. BAliBASE protein veritabanından alınmış aileleri için ABC algoriması ile elde edimiş SPS skorları

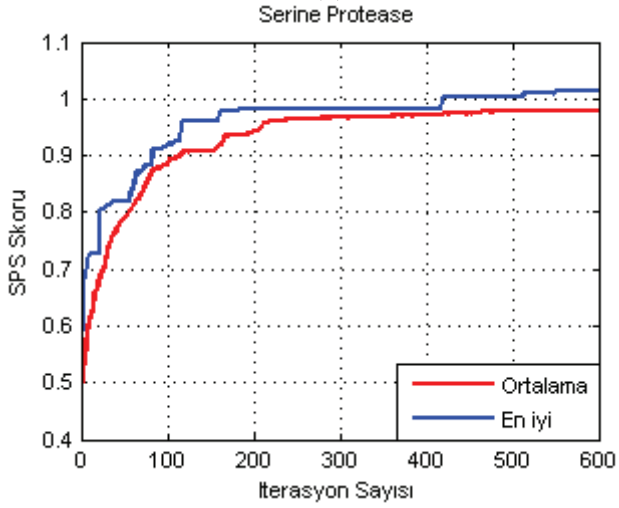
Protein Aileleri	**En Yüksek SPS**	**Ortalama SPS**	**En Düşük SPS**	**Std. Sapma**
SH3	1.037037	0.962963	0.913580	0.044322
Cytochrome C	0.854369	0.828155	0.791262	0.019578
Serine Protease	1.011730	0.979765	0.947214	0.024279
Protein Kinase	0.619137	0.605629	0.585366	0.012184
Anthranilate Isomerase	0.677560	0.626144	0.594771	0.025223
Serine Protease	0.823821	0.795533	0.769231	0.016766
Aminotransferase	0.738710	0.711613	0.674194	0.020525
Glutamyl-TRNA Synthetase	0.667614	0.620455	0.583807	0.028371
Taq DNA Polymerase	0.578620	0.538969	0.506663	0.023312



(a)

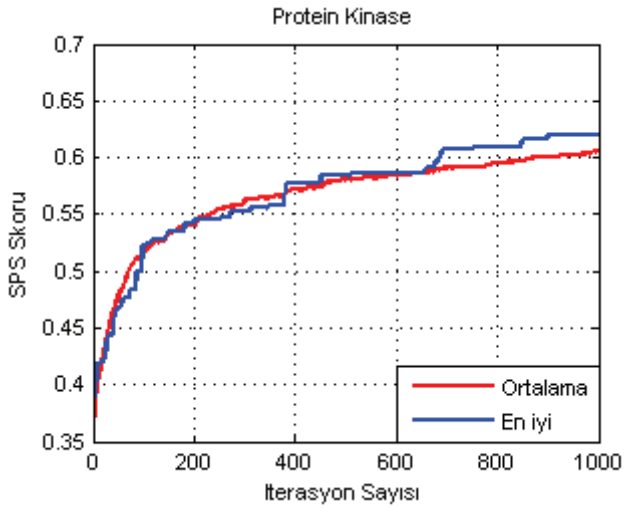


(b)

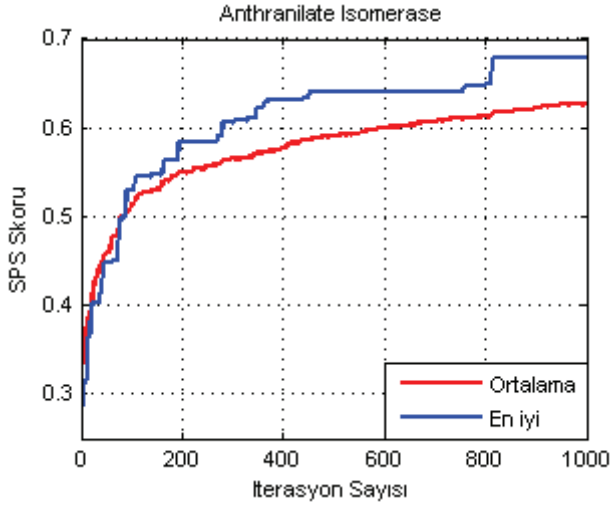


(c)

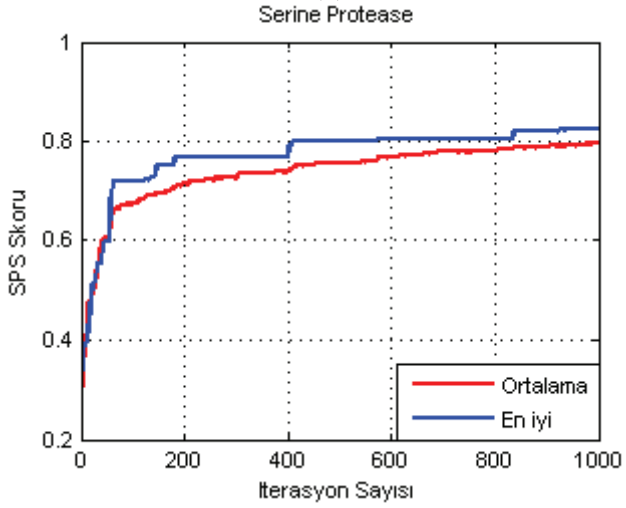
Şekil 5.1 (a) SH3, (b) Cytochrome C ve (c) Serine Protease protein aileleri için ABC algoritmasının yakınsama grafikleri.



(a)

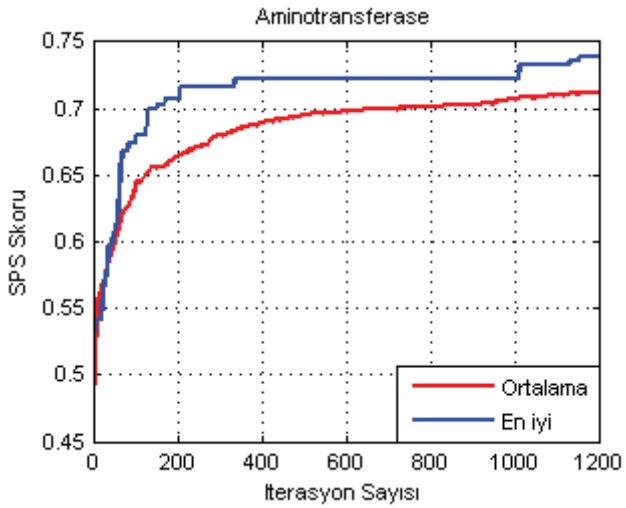


(b)

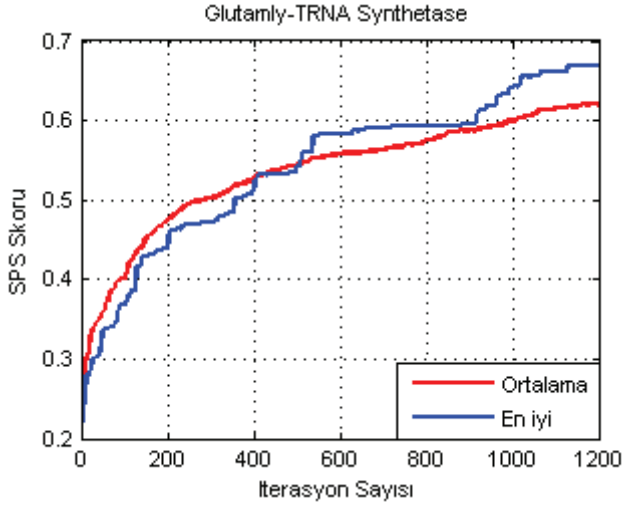


(c)

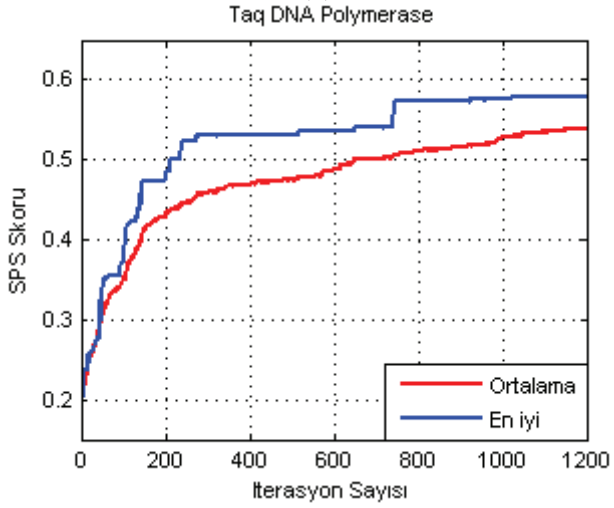
Şekil 5.2 (a) Protein Kinase, (b) Anthranilate Isomerase , (c) Serine Protease protein aileleri için ABC algoritmasının yakınsama grafikleri.



(a)



(b)



(c)

Şekil 5.3 (a) Aminotransferase, (b) Glutamyl-TRNA Synthetase, (c) Taq DNA Polymerase protein aileleri için ABC algoritmasının yakınsama grafikleri.

Test sonuçları dokuz farklı protein ailesi için yukarıda ayrıntılı olarak verilmiş önerilen Yapay Arı Koloni Algoritması (YABC), aynı protein ailelerin kullanılarak test edilmiş diğer bir Yapay Arı Koloni Algoritması, Genetik Algoritma ve Parçacık Sürü Algoritması ile Relative Sum Of Pairs skorları üzerinden karşılaştırılmıştır.

Önerilen Yapay Arı Koloni Algoritması'nın karşılaştırılacağı Yapay Arı Koloni Algoritması'nda, başlangıç yiyecek, besin kaynaklarının yerlerini temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 1.2 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin yine rastgele belirlenmiş pozisyonlara, çoklu hizalamanın hiçbir sütunu sadece boşluk karakterlerinden oluşmayacak şekilde dağıtılmasıyla oluşturulmuş, gözcü arıların yiyecek kaynaklarına gönderilmesi için $i.$ çoklu hizalamayı temsil eden ve her birinin uzunluğu L_i olan D adet sekansa sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} sekansı başlangıç besin kaynakları oluşturulurken referans alınan her hangi bir sütunun sadece boşluk karakterleri barındıramayacağı yaklaşımı ile mevcut boşluk karakterlerinin rastgele tekrar yerleştirilmesi yaklaşımı kullanılmıştır [32].

Aşağıda karşılaştırılan Yapay Arı Koloni Algoritması, Genetik Algoritma ile Parçacık Sürü Algoritmasının kontrol parametreleri, Kısa, Orta ve Uzun sınıflara ait protein ailelerine göre belirlenmiştir ve ilgili parametreler önerilen Yapay Arı Koloni Algoritması'nda da kullanılmıştır, farklı tablolar halinde gösterilmektedir [32].

Tablo 5.4 Karşılaştırmada kullanılacak Yapay Arı Koloni Algoritmasının Kontrol Parametreleri [32]

Yapay Arı Koloni Algoritması		
Limit	**Popülasyon Büyüklüğü**	**Çevrim Sayısı**
5/10/10	20	600/1000/1200

Tablo 5.5 Karşılaştırmada kullanılacak Genetik Algoritmanın Kontrol Parametreleri [32]

Genetik Algoritma		
Çaprazlama Oranı	**Mutasyon Oranı**	**Jenerasyon Sayısı**
0.6	0.05	600/1000/1200

Tablo 5.6. Karşılaştırmada Kullanılacak Parçacık Sürü Algoritmasının Kontrol Parametreleri [32]

Parçacık Sürü Algoritması		
α Değeri	**β Değeri**	**Jenerasyon Sayısı**
0.8	0.8	600/1000/1200

Tablo 5.7. SH3, Cytochrome C ve Serine Protease protein aileleri için algoritmaların karşılaştırma sonuçları [32]

Protein Ailesi	**Algoritma**	**En iyi SPS Skoru**	**Ortalama SPS Skoru**	**En Düşük SPS Skoru**
SH3	GA	0.6829	0.6090	0.5463
	PSO	0.6543	0.6444	0.6296
	ABC	0.7654	0.6895	0.6543
	YABC	**1,0370**	**0,9629**	**0,9135**
Cytochrome C	GA	0.5398	0.5141	0.4891
	PSO	0.6407	0.6252	0.6116
	ABC	0.6854	0.6451	0.5922
	YABC	**0,8543**	**0,8281**	**0,7912**
Serine Protease	GA	0.8065	0.7455	0.6921
	PSO	0.7331	0.7120	0.7008
	ABC	0.8168	0.7956	0.7732
	YABC	**1,0117**	**0,9797**	**0,9472**

Tablo 5.8. Protein Kinase, Anthranilate Isomerase ve Serine Protease protein aileleri için algoritmaların karşılaştırma sonuçları [32]

Protein Ailesi	**Algoritma**	**En iyi SPS Skoru**	**Ortalama SPS Skoru**	**En Düşük SPS Skoru**
Protein Kinase	GA	0.3620	0.3508	0.3453
	PSO	0.5065	0.4848	0.4765
	ABC	0.5257	0.4951	0.4765
	YABC	**0,6191**	**0,6056**	**0,5853**
Anthranilate Isomerase	GA	0.3440	0.3317	0.3244
	PSO	0.4117	0.3813	0.3529
	ABC	**0.4292**	**0.3825**	**0.3682**
	YABC	0,6775	0,6261	0,5947
Serine Protease	GA	0.4334	0.3792	0.3579
	PSO	0.2725	0.2626	0.2507
	ABC	0.4480	0.4178	0.4036
	YABC	**0,8238**	**0,7955**	**0,7692**

Tablo 5.9. Aminotransferase, Glutamly-TRNA Synthetase ve Taq DNA Polymerase protein aileleri için algoritmaların karşılaştırma sonuçları [32]

Protein Ailesi	Algoritma	En iyi SPS Skoru	Ortalama SPS Skoru	En Düşük SPS Skoru
Aminotransferase	GA	0.3719	0.3646	0.3548
	PSO	0.5903	0.5819	0.5645
	ABC	0.5936	0.5819	0.5741
	YABC	**0,7387**	**0,7116**	**0,6741**
Glutamyl-TRNA Synthetase	GA	0.3824	0.3738	0.3648
	PSO	0.4154	0.3987	0.3828
	ABC	0.4254	0.4040	0.3857
	YABC	**0,6676**	**0,6204**	**0,5838**
Taq DNA Polymerase	GA	0.2644	0.2500	0.2441
	PSO	0.2588	0.2533	0.2484
	ABC	0.2615	0.2569	0.2520
	YABC	**0,5786**	**0,5389**	**0,5066**

Yapay Arı Koloni Algoritması'nın başlangıç yiyecek kaynaklarının oluşturulması ve komşuluk tabanlı yeni aday çözüm üretmek için sunulan modelleri, karşılaştırılan Yapay Arı Koloni Algoritması, Genetik Algoritma ve Parçacık Sürü Algoritması'na kıyasla oldukça başarılı sonuçlar üretmiştir.

Hizalanacak tüm sekansların ikişerli grupları arasında hesaplanan benzerlik oranı artıp, sekans uzunlukları azaldıkça elde edilecek çoklu hizalamaların kaliteli olma olasılığı, önerilen Yapay Arı Koloni Algoritması ile daha da yükselmiştir.

Orta ve Uzun gruplara dahil protein ailelerinde ise, hizalanacak sekans sayısı ve sekansların uzunlukları sebebi ile araştırma uzayı oldukça genişlemiş olmasına karşın önerilen Yapay Arı Koloni Algoritması, karşılaştırılan algoritmalara kıyasla yüksek benzerlik oranlı sekansları iyi yöneterek daha kaliteli çözümleri, bir miktar artmış standart sapma oranları ile üretebilmiştir.

BAliBASE protein veritabanın alınan test verileri için, önerilen Yapay Arı Koloni Algoritması ile elde edilen sonuçlar, uzun sekansların parçalanarak ilk önce kendi içlerinde çoklu hizalanabileceği, alt sekanslardan bulunan çoklu hizalamaların birleştirilerek, tüm sekans bileşenlerini barındıran genel çoklu hizalamanın bulunabileceği hususunda ipucu vermektedir.

5.2 Yapay Arı Koloni Algoritması İle 3D_ali Veritabanından Alınan Sekansların Hizalanması

Yapay Arı Koloni Algoritması'nın Çoklu Dizi Hizalama Problemi'nin çözümü için kullanılan modelinde başlangıç yiyecek, besin kaynaklarının yerlerini temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 1.5 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği durum ile seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin pozisyon ile sayıları rastgele belirlenmiş üç blok halinde dağıtıldığı durum birlikte kullanılarak oluşturulmuştur.

Gözcü arıların yiyecek kaynaklarına gönderilmesi için , i. çoklu hizalamayı temsil eden ve her birinin uzunluğu L_i olan D adet sekansa sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} sekansı rastgele bir pozisyondan bölünerek alınan ilk parçasının, diğer çoklu hizalamalar arasından rastgele seçilmiş her birinin uzunluğu L_k olan D adet sekansa sahip x_k çözümünün x_{kj} sekansının uygun parçasılya birleştirilmesiyle, x_{kj} sekansının uygun parçası x_{ij} sekansının rastgele bir pozisyondan bölünerek alınan ilk parçası ile birleştirilirken yeni sekansın uzunluğu L_i yapılmak adına yeteri kadar boşluk karakteri x_{kj} sekansının uygun parçasının rastgele pozisyonlarına eklenmekte veya rastgele pozisyonlarından silinmektedir, yine herbirinin uzunluğu L_i olan yeni bir çoklu hizalama oluşturulurken kullanılan durum seçilmiş, çoklu hizalamaların kalite değerlerinin belirlenmesi için de COFFEE (Consistency Based Objective Function For Alignment Evaluation) skor hesaplama yöntemi kullanılmıştır.

Yapay Arı Koloni Algoritması'nın temel işlem adımlarına ilişkin düzenlemelerin yukarıdaki gibi yapıldığı uygulamanın test edilmesi amacı ile 3D_ali protein veritabanından 9 farklı protein ailesi, 3D_ali mevcut protein aileleri ile ilgili referans çoklu hizalamalarıda barındıracak şekilde oluşturulmuştur, seçilmiştir [26, 29].

3D_ali veritabanından seçilen protein aileleri, sekans sayısı ve sekanslar arası benzerlik açısından BAliBASE veritabanından seçilmiş protein ailelerine kıyasla farklı belirlenmiştir [26, 31]. 3D_ali veritabanından seçilmiş protein aileleri 17 ile 6 arasında

sekans grupları barındırmaktadır ve %21 ile %61 arasında değişen ortalama benzerlik oranlarına sahiptir. Tablo 5.27'de Yapay Arı Koloni Algoritması'nın test sürecinde kullanılacak 3D_ali protein veritabanından alınmış protein aileleri gösterilmektedir.

Tablo 5.10. 3D_ali protein veritabanından seçilmiş protein aileleri [26]

Protein Ailesi	**Sekan Sayısı**	**Referans Hizalama Uzunluğu**	**Ortalama Benzerlik**
Acid Protease	14	248	21
Sugar/Amino Acid Binding	7	500	31
Cytochrome C	6	146	42
Fibronectin Type III	9	136	17
Crystallins	8	52	36
Globins/Phycocyanins/Collicins	17	183	24
Lyzozymes/Lactalbumin	6	213	39
Dihydroxybiphenyl Dioxygenase	8	90	22
Subtilisin	7	331	61

Acid Protease için Ac_Prot, Sugar/Amino Acid Binding için Binding, Cytochrome C için Cytc, Fibronectin Type III için fniii, Crystallins için Gcr, Globins/Phycocyanins/Collicins için Globin, Lyzozymes/Lactalbumin için Lzm, Dihydroxybiphenyl Dioxygenase için Phenyldiox, Subtilisin için S_Prot kısaltmaları kullanılmıştır [26]. Test edilecek protein ailelerindeki sekans sayısı, her bir sekansın uzunluğu ve aile içi ortalama benzerlik, Yapay Arı Koloni Algoritması'nın problemin çözümü için seçilecek uygun çevrim sayısı, kolonideki işçi, gözcü, kâşif arı oranları, kaynağın bırakılıp bırakılmayacağı kararının verildiği $Limit$ parametresinin seçiminde önem teşkil etmektedir. Yapay Arı Koloni Algoritması'nın yukarıda önerilen işlem adımları için seçilmiş parametreleri aşağıdaki tabloda verilmiştir.

Tablo 5.11. Yapay Arı Koloni Algoritması'nın 3D_ali aileleri için parametre değerleri

Sekans Sayısı	**Limit Değeri**	**Popülasyon Büyüküğü**	**Çevrim Sayısı**
<10	10	40	3000
≥10	10	40	4000

3D_ali protein veritabanından seçilen her bir protein ailesi; sekans sayısı 10'dan küçük olanlar için 3000 iterasyon, sekans sayısı 10'dan büyük olanlar için 4000 iterasyon, 20 işçi ve 20 gözcü arıdan oluşan popülasyon ile *Limit* değerleri için 10'u aşan her hizalama için bir kaşif arı kullanılarak 10 farklı kez Yapay Arı Koloni Algoritmasıyla hizalanmış, tüm koşmaların her iterasyonda en kaliteli çoklu hizalamala skorları kaydedilmiş, en kaliteli çoklu hizalamayı üreten koşmadan elde edilen gelişim, yakınsama eğrisi ile tüm koşmalarda elde edilen değerlerin ortalamaları alınarak elde edilen gelişim, yakınsama eğrisi çizdirilmiş, 10 farklı koşmaya ait en iyi değerlerin ortalama, en yüksek, en düşük skor değerleri tablolar halinde aşağıda gösterilmiştir.

Tablo 5.12. Acid Protease Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Acid Protease		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.362842	0.333477	0.307112
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.019700		

Tablo 5.13. Acid Binding Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Sugar/Amino Acid Binding		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.790084	0.747664	0.662991
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.040400		

Tablo 5.14. Cytochrome C Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Cytochrome C		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.676544	0.617444	0.542580
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.045900		

Tablo 5.15. Fibronectin Type III Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Fibronectin Type III		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.584063	0.537877	0.505568
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.023800		

Tablo 5.16. Crystallins Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Crystallins		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0,949420	0,941054	0,925125
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.008920		

Tablo 5.17. Globins/Phycocyanins/Collicins Protein Ailesinin hizalanmasında YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

Protein Ailesi: Globins/Phycocyanins/Collicins		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.562535	0.510465	0.471329
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.029000		

Tablo 5.18. Lyzozymes/Lactalbumin Protein Ailesinin hizalanmasından YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

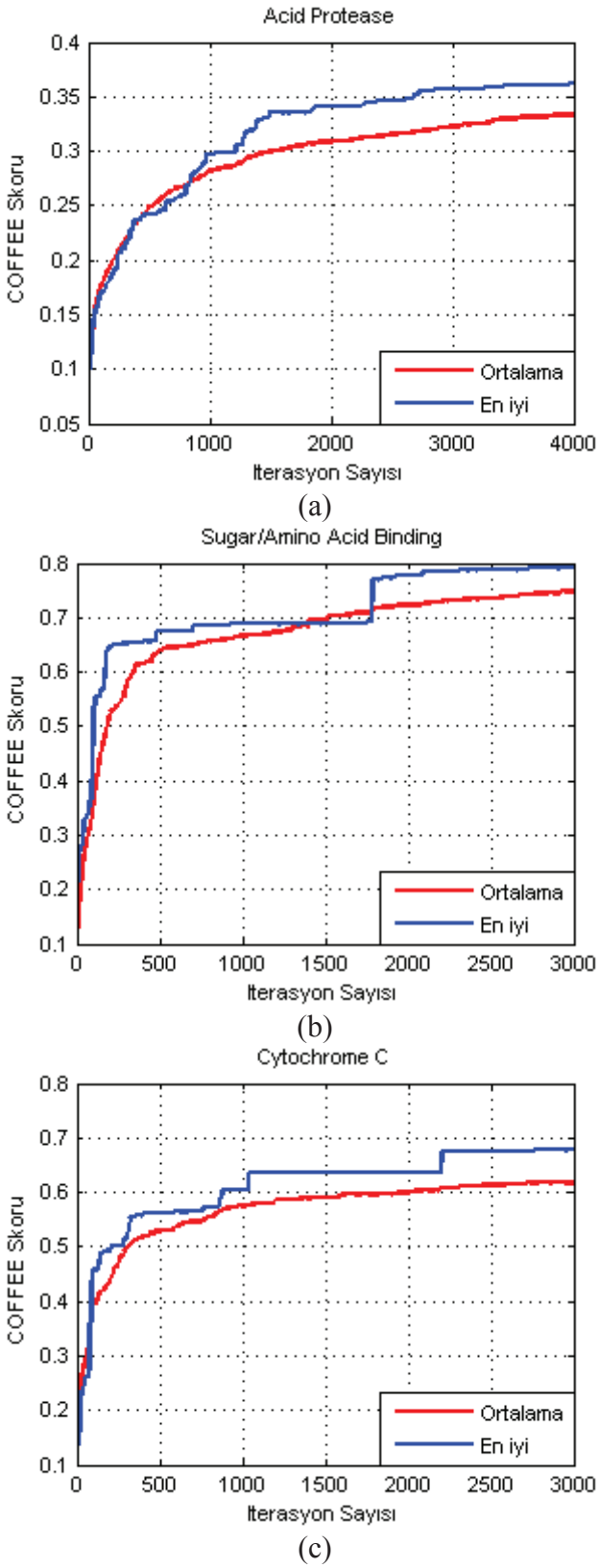
Protein Ailesi: Lyzozymes/Lactalbumin		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0.817407	0.685750	0.548796
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.087600		

Tablo 5.19. Dihydroxybiphenyl Dioxygenase Protein Ailesinin hizalanmasından YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

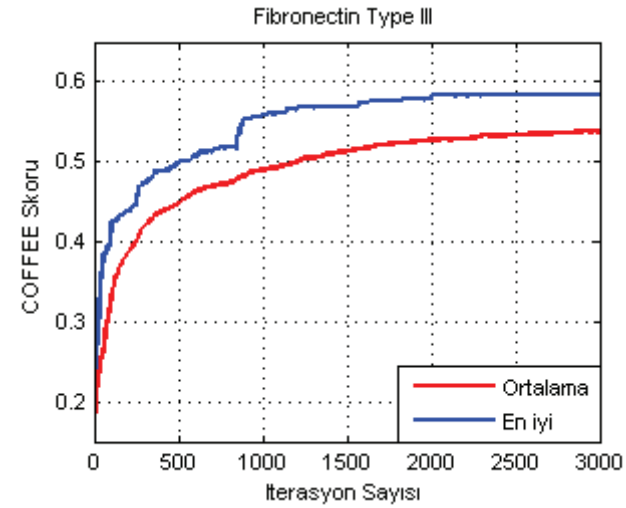
Protein Ailesi: Dihydroxybiphenyl Dioxygenase		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0,799573	0,746544	0,708863
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.033333		

Tablo 5.20. Subtilisin Protein Ailesinin hizalanmasından YABC ile elde edilen en yüksek, ortalama, en düşük COFFEE skoru ve standart sapma

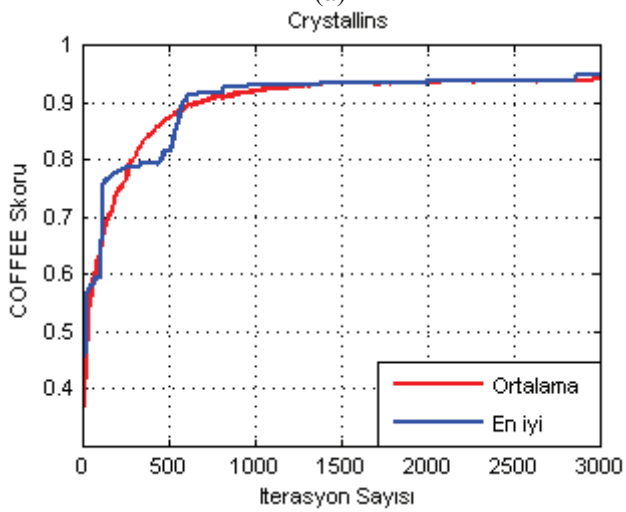
Protein Ailesi: Subtilisin		
En Yüksek COFFEE Skoru	**Ortalama COFFEE Skoru**	**En Düşük COFFEE Skoru**
0,799573	0,746544	0,708863
Bulununan en yüksek on COFFEE skoru için standard sapma: 0.029800		



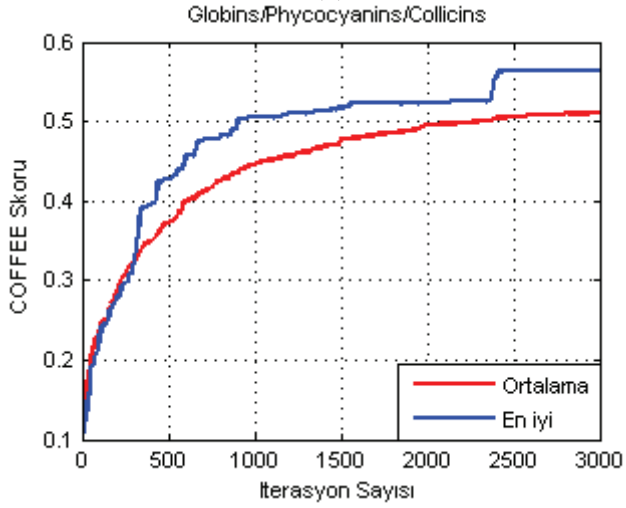
Şekil 5.4 (a) Acid Protease, (b) Sugar/Amino Acid Binding, (c) Cytochrome C protein aileleri için ABC algoritmasının yakınsama grafikleri.



(a)

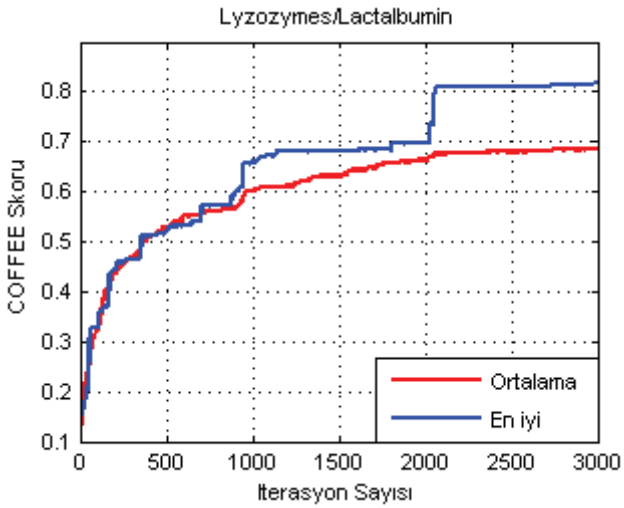


(b)

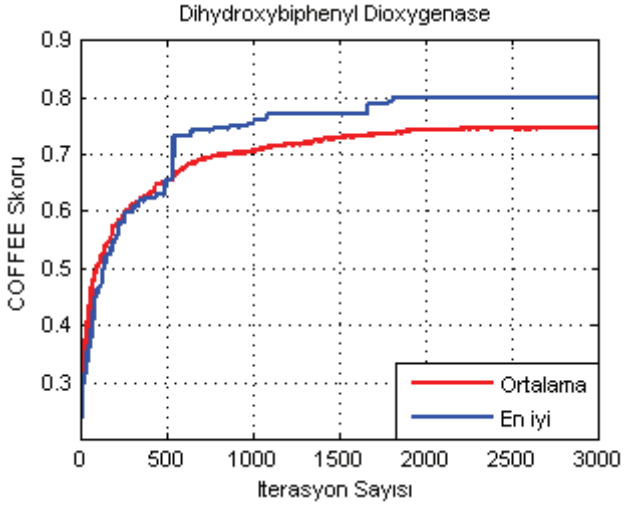


(c)

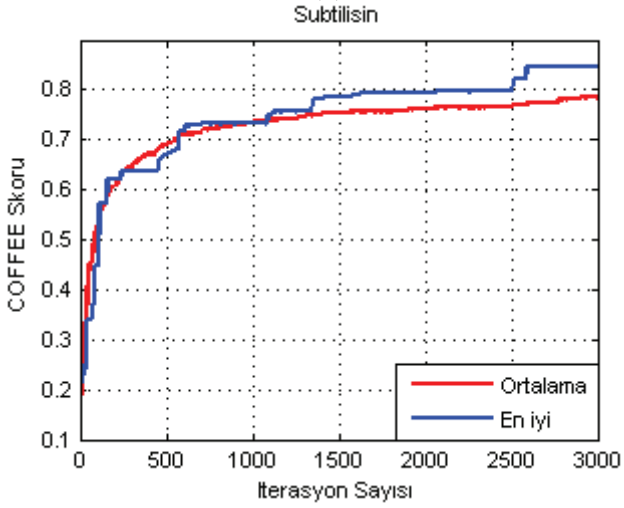
Şekil 5.5 (a) Fibronectin Type III, (b) Crystallins, (c) Globins/Phycocyanins/Collicins protein aileleri için ABC algoritmasının yakınsama grafikleri.



(a)



(b)



(c)

Şekil 5.6 (a) Lyzozymes/Lactalbumin, (b) Dihydroxybiphenyl Dioxygen (c) Substilin protein aileleri çin ABC algoritmasının yakınsama grafikleri.

Test sonuçları dokuz farklı protein ailesi için yukarıda ayrıntılı olarak verilmiş önerilen Yapay Arı Koloni Algoritması, aynı protein ailelerin kullanılarak test edilmiş Genetik Algoritma tabanlı SAGA isimli uygulama ve dinamik programlama tabanlı CLUSTALW isimli uygulama ile COFFEE skorları üzerinden [29] çalışması referans alınarak karşılaştırılmıştır.

Önerilen Yapay Arı Koloni Algoritmasının, karşılaştırılacağı Genetik Algoritma tabanlı SAGA uygulamasında, SAGA uygulamasında başlangıç popülasyonunun büyüklüğü tüm hizalamalar için varsayılan olarak 100 seçilmektedir, başlangıç popülasyonundaki bireyleri temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 2 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği durum kullanılmıştır [26, 29].

SAGA uygulaması, Genetik Algoritma'nın bir sonraki jenerasyona dahil edilmek üzere yeni birey ürettiği Çaprazlama (Crossover) ve Mutasyon (Mutation) aşamalarını, Çoklu Dizi Hizalama Problem'inin zorluğu dikkate alınarak, toplamda 22 farklı operatörü olasılık tabanlı bir seçim mekanizması ile seçip, Çaprazlama için iki ve Mutasyon için tek birey kullanılmaktadır, yönetmektedir [26, 29, 33]. Önerilen Yapay Arı Koloni Algoritması, yeni çoklu hizalamayı temsil eden komşuluk tabanlı üretilmiş besin kaynağını seçilmiş hizalamanın tek bir sekansı üzerinde değişiklik yaparak üretirken, SAGA Çaprazlama işlemi sırasında iki çoklu hizalamanın tüm sekanslarından faydalanmakta, Mutasyon işlemi sırasında da çoklu hizalamanın geneli üzerinde değişim etkisi oluşturmaktadır [26, 29]. Genetik Algoritmanın esnek yapısı, Çaprazlama ve Mutasyon operatörlerindeki çeşitlilik ile de birleştirilince, SAGA uygulamasının aynı protein aileleri üzerinde, en iyi çözüm ya da çoklu hizalamanın on jenerasyon boyunca iyileştirilememesi durumunda çalışmayı sonlandırdığı zaman elde ettiği en kaliteli çoklu hizalama referens alınmıştır, ürettiği çoklu hizalamaların önerilen Yapay Arı Koloni Algoritması ve CLUSTALW uygulaması ile üretilen çoklu hizalamalardan daha kaliteli olacağı tahmin edilebilir.

CLUSTALW ise iki dizinin hizalanması için oldukça sık kullanılan ve dinamik programlama olarak isimlendirilmiş algoritmik bir yaklaşımı, dinamik programlama

algoritmik yaklaşımı sekans sayısı ve sekans uzunluğu arttıkça uygunluğunu kaybetmektedir, referans almaktadır [26, 33]. Çoklu hizalamalar, sekansların benzerlikleri kullanılarak oluşturulmuş filogenetik ağaca göre belirlenen sırada, sekansların tamamı dahil edilinceye kadar ikili hizalamaların (Pairwise Alignments) birleştirilmesi şeklinde oluşturulur. CLUSTALW kullanılan yaklaşım sebebi ile çoklu hizalamaları hızlı şekilde üretmesine karşın, ilk hizalamalar oluşturulurken yapılan hatalar, yaklaşımın getirdiği kademeli ilerleme ve en iyi odaklı, aç gözlü (greedy) seçim mekanizması sebebiyle çoklu hizalama üretilinceye kadar korunmuş olur [26, 28]. Aşağıda, önerilen Yapay Arı Koloni Algoritması'nın herbir aile için on farklı koşmasında elde edilen en iyi değer ile, SAGA ve CLUSTALW uygulamalarından aynı protein aileleri kullanılarak elde edilmiş sonuçlar verilmiştir.

Tablo 5.21. 3D_ali protein veritabanından alınan sekansların karşılaştırmalı sonuçları[29]

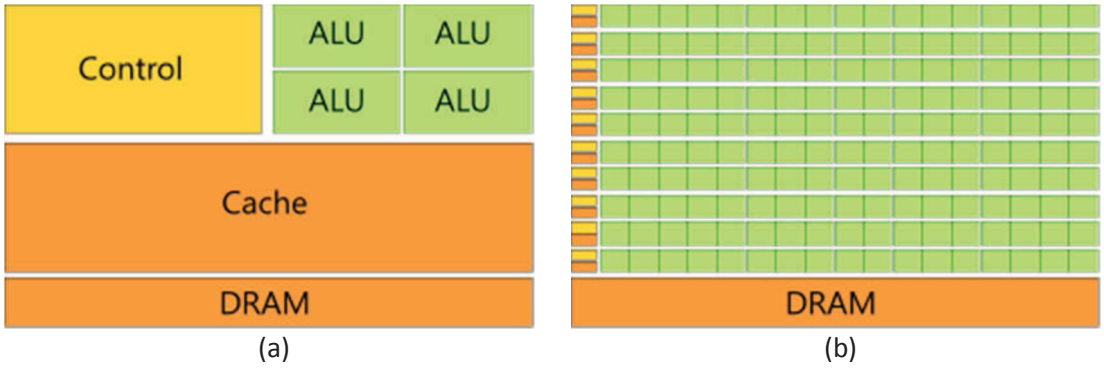
Protein Ailesi	SAGA [29]	CLUSTALW [29]	YABC
Acid Protease	**0.56**	0.48	0.36
Sugar/Amino Acid Binding	**0.84**	0.72	0.79
Cytochrome C	**0.87**	0.84	0.67
Fibronectin Type III	**0.62**	0.49	0.58
Crystallins	0.89	0.86	**0,94**
Globins/Phycocyanins/Collicins	**0.80**	0.78	0.56
Lyzozymes/Lactalbumin	**0.87**	0.87	0.81
Dihydroxybiphenyl Dioxygenase	0.64	0.59	**0,79**
Subtilisin	**0.97**	0.96	0.84

SAGA, CLUSTALW ve önerilen Yapay Arı Koloni Algoritması'nın karşılaştırmalı COFFEE skorlarından da anlaşılabileceği üzere, Genetik Algoritma tabanlı SAGA uygulamasının çoklu hizalamaların genelinde en kaliteli skorlara sahip olduğu görülmektedir. Önerilen Yapay Arı Koloni Algoritması ise, BAliBASE protein veritabanından alınıp hizalanan aileler içinde elde edilen sonuçlar da dikkate alındığında, benzerlik oranı %20 ile %40 arasında olan 5 ile 10 arasında sekans barındıran ve bu sekans uzunlukları 50 ile 300 arasında değişen aileler için başarılı sayılabilecek sonuçları yine üretmiştir. 3D_ali protein veri tabanından seçilmiş ailelerin, önerilen Yapay Arı Koloni Algoritması'nın ürettiği COFFEE skorları üzerinde sekans sayılarının, benzerlik ve uzunluklarına kıyasla daha etkili olduğu da söylenebilir.

5.3 Relative Sum Of Pairs (SPS) Skor Hesaplama Yönteminin CUDA Platformuna Aktarılması

Kullanıcı taleplerinin gerçek zamanlı (realtime), yüksek çözünürlüklü üç boyutlu (high definition 3D) grafikler üzerinde yoğunlaşması, paralelleştirilebilirliği oldukça yüksek, çok sayıda çekirdeğe (manycore) sahip, çok kanallı (multithreaded), yüksek işlem gücü ve bant genişlikli (high memory bandwidth) grafik işleme birimlerinin (graphical processing unit, gpu) geliştirilip, yaygınlaştırılmasının önünü açmıştır [37, 41, 42].

Grafik işleme birimleri ya da ekran kartlarının bu özellikleri, merkezi işlem birimlerinde (cpu), grafik işlem birimlerine kıyasla daha çok transistör iş akış kontrolü (flow control) ile veri depolama (data caching) için ayrılmıştır, aritmetik işlem miktarının veri kopyalama işlem miktarına oranının daha yüksek olduğu, işlem yoğunluklu (high arithmetic intensity, compute-intense, data-parallel computations) üç boyutlu modelleme, görüntü, sinyal, örüntü işleme ve biyoinformatik problemlerin çözüldüğü uygulamaların pek çoğunda hesaplama maliyetlerini düşürmüştür [37, 41, 42]. Aşağıda grafik işlem birimleri ile merkezi işlem birimleri arasında kontrol ve depolama işlemlerine adanmış birimler sembolik olarak gösterilmektedir.

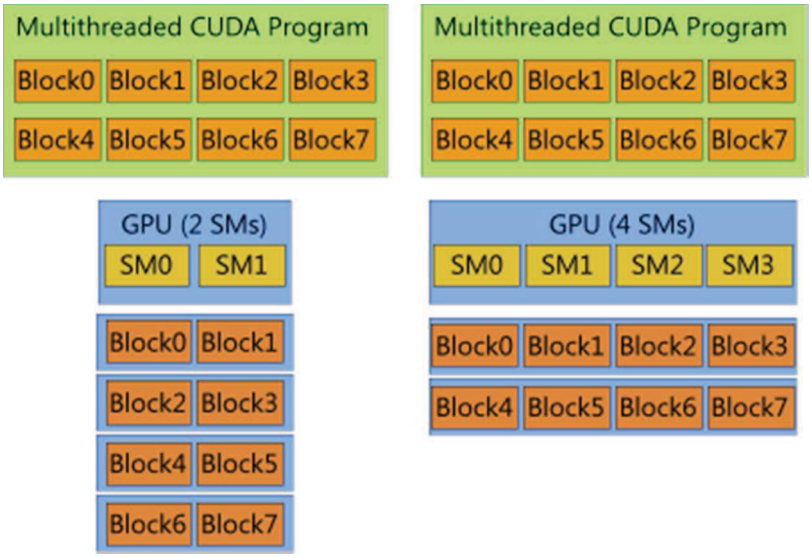


Şekil 5.7 (a) CPU ve (b) GPU hafıza ve işlem ünite dağılımları [37].

2006 yılının Kasım ayında NVIDIA firması, ürettikleri ekran kartlarındaki grafik işlem birimlerinin paralelleştirilebilirliği yüksek ve veri bağımlılığı az kompleks problemlerin çözümünde kullanılabilmesi için CUDA (Compute Unified Device Architecture) adı verdikleri genel paralel hesaplama, programlama platformunu, C gibi üst seviye bir programa dilinin de aralarında bulunduğu pek çok dil desteği ile, uygulama geliştiricilere duyurmuştur [37].

Hem işlemci, hem de ekran kartlarının gelişen paralel mimarisi, bu mimarilerin getireceği avantajlardan daha fazla yararlanmak adına üretilecek uygulamaların da düzenlenmesini gerekmektedir. Geliştirilen uygulamaların paralelleştirme yaklaşımını, uygulamanın çalıştırıldığı çoklu paralel işlem birimine sahip platformlara göre dinamik olarak düzenleyebilmesi oldukça önemlidir.

CUDA paralel programlama platformu, uygulamaların farklı mimarilere göre dinamik düzenlenmesi gereken paralel yapısını, uygulamanın taşındığı NVIDIA ekran kartının özellikleri ve geliştiricinin tanımladığı thread-block-grid yapısına göre dinamik ayarlamaktadır [37, 41, 42]. Aşağıda grafik işlem birimlerinin en temel elemanları olan Streaming Multiprocessor (SM) sayısına göre CUDA uygulamasının çalıştırılma yaklaşımı gösterilmiştir.



Şekil 5.8 Thread-Block-Grid ve Streaming Multiprocessor yapısı [37].

Yapay Arı Koloni Algoritması'nın Çoklu Dizi Hizalama probleminin çözümü için kullanılan ve CUDA platformuna aktarılacak metod ya da metodların belirleneceği modelinde; başlangıç yiyecek, besin kaynaklarının yerlerini temsil eden çoklu hizalamalar, çoklu hizalamaların uzunlukları en büyük sekansın 1.2 katını geçmeyecek şekilde rastgele belirlenmiştir, seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk karakterinin rastgele belirlenmiş boyutlardaki iki blok halinde, sekansın baş ve sonuna (sequence terminal) eklendiği durum ile seçilen hizalama uzunluğu ve orijinal sekans uzunluğu arasındaki fark kadar boşluk

karakterinin pozisyon ile sayıları rastgele belirlenmiş üç blok halinde dağıtıldığı durum birlikte kullanılarak oluşturulmuştur.

Gözcü arıların yiyecek kaynaklarına gönderilmesi için i. çoklu hizalamayı temsil eden ve her birinin uzunluğu L_i olan D adet sekansa sahip x_i çözümünün, j değeri $[1, D]$ aralığında rastgele belirlenmektedir, x_{ij} sekansına ait boşluk karakter lokasyonları ile sahip olduğu j. sekansın uzunluğu L_i yapılmak adına yeteri kadar boşluk karakterinin rastgele pozisyonlara eklendiği veya rastgele pozisyonlardan silindiği diğer çoklu hizalamalar arasından rastgele seçilmiş, her birinin uzunluğu L_k olan D adet sekansa sahip x_k çözümünün düzenlenmiş x_{kj} sekansının boşluk karakter lokasyonlarının karşılaştırıldığı, x_{ij} sekansı ile düzenlenmiş x_{kj} sekansı arasındaki ortak boşluk karakterleri pozisyonlarını korurken, ortak olmayan boşluk karakterleri arasında olasılık tabanlı bir seçim yapılarak, yine herbirinin uzunluğu L_i olan yeni bir çoklu hizalama oluşturulurken kullanılan durum seçilmiş, çoklu hizalamaların kalite değerlerinin belirlenmesi için de SPS skor hesaplama yöntemi kullanılmıştır.

Yapay Arı Koloni Algoritması'nın Çoklu Dizi Hizalama probleminin çözümü amacıyla özellikleri yukarıda verilen modelinin test işlemlerinde, BAliBASE protein veri tabanından alınmış 9 protein ailesi arasından en uzun sekanslara sahip Taq DNA Polymerase seçilmiştir. Bu aile için üretilecek çoklu hizalamaların uzunluğu 1024, CUDA platformu için her bir blokta seçilen 32 thread eş zamanlı yönetildiğinden 32'nin katları bloktaki toplam thread sayısı olarak seçilebilir, olarak alınmıştır [37, 38]. Çoklu hizalama uzunluklarının aynı alınması, daha kaliteli hizalamaların aranmasını sağlayan seçilmiş sekans düzenleme işlemini sınırlamış olsa da, bu durum uygulamanın paralelleştirilebilir metod ya da metodlarının CUDA platformuna önerilen şekilde aktarılabilirliği için önemlidir.

CUDA platformuna aktarılacak metod ya da metodların belirlenmesi amacıyla Yapay Arı Koloni Algoritması, tarif edilen özelliklere göre C programlama dilinde kodlanıp işlemci numarası i7-2600, çalışma hızı 3.4 GHz olarak belirtilen Intel işlemcili ve 1333MHz çalışma frekanslı DDR3 (Double Data Rate) RAM (Random Access Memory) modüllerine sahip Ubuntu 11.10 işletim sistemi üzerinde çalıştırılıp, GCC (Gnu Compiler Collection) üyesi *gprof* [36] uygulaması ile profil sonuçları elde edilmiş ve Tablo 5.22'de gösterilmiştir.

Tablo 5.22. CUDA platformuna aktarılacak metodun belirlenmesi için profil sonuçları

Metod Adı	İstek Sayısı	Sadece Metod (ms)	Toplam Geçen (ms)
generateNewAlignmentUsingNeighbourhood	24000	0.37	0.72
getValueOfDifferentElements	47744	0.05	0.05
getValueOfEqualElements	24000	0.06	0.06
getNumberOfDifferentElements	24000	0.06	0.06
getNumberOfEqualElements	24000	0.06	0.06
***calculateRelativeSumOfPairsScore**	**34640**	**0.04**	**0.04**
copyAlignmentContentToEmptyAlignment	72000	0.01	0.01
sortInAscendingOrder	23872	0.03	0.03
getNumberOfMatchedGapCharactersBetweenResidues	34640	0.01	0.01
removeMatchedGapCharactersInTerminals	34640	0.01	0.01
removeMatchedGapCharactersBetweenResidues	34640	0.01	0.04
generateAdjustedAlignment	34640	0.01	0.06
getIndexOfMatchedGapCharactersBetweenResidues	24169	0.01	0.01
generateAlignmentUsingTerminalOrBlockGapCharacters	10630	0.02	0.02
getIndexOfGapCharactersInSequence	48000	0.00	0.00
getNumberOfGapCharactersInSequence	24000	0.00	0.00
calculateFitnessValueOfAlignment	34640	0.00	0.00
getMaximumValue	34640	0.00	0.00
getMinimumValue	34640	0.00	0.00
getRandomAlignmentIndex	24000	0.00	0.00
getRandomSequenceIndex	24000	0.00	0.00
getRelativeSumOfPairsScoreValueOfBestAlignment	1201	0.00	0.00
getRelativeSumOfPairsScoreValueOfWorstAlignment	1201	0.00	0.00
calculateProbabilityValuesOfAlignments	1200	0.00	0.00
employedBeePhase	1200	0.00	8.16
onlookerBeePhase	1200	0.00	8.16
scoutBeePhase	1200	0.00	1.01
copyAdjustedAlignmentContentToExistingAdjustedAlignment	359	0.00	0.00
copyAlignmentContentToExistingAlignment	359	0.00	0.00
getIndexOfBestAlignment	47	0.00	0.00
generateAdjustedReferenceAlignment	1	0.00	0.00
generateInitialAdjustedAlignments	1	0.00	0.49
generateInitialAlignmentsUsingTerminalOrBlockGapCharacters	1	0.00	0.00
printAdjustedAlignments	1	0.00	0.00

Hem profil sonuçları hem de Yapay Arı Koloni Algoritması için önerilen modelin veri bağımlılığı dikkate alındığında uygulamanın tamamını CUDA destekli ekran kartları

üzerine taşımak yerine, veri bağımlılığı görece az, paralelleştirilebilirliği ve CPU'da işlendiği zaman yüksek metodlardan en uygun olanının seçilmesi daha doğru olacaktır.

Deklerasyonu *int calculateRelativeSumOfPairsScore(char*)* şeklinde yapılmış ve SPS (Relative Sum Of Pairs) skorunun hesaplandığı, 34600 kez çağrılmış, her çağrılışında ortalama 0.04 ms (milisaniye), 40 µs (mikrosaniye), CPU'da işlenmiş metod CUDA platformuna aktarılmak üzere seçilmiştir. Üç farklı paralelleştirme senaryosunda mimarisi, çalışma frekansları ve hafıza özellikleri aşağıda verilmiş Tesla K20X, Tesla K20, Tesla K10, Tesla M2090, Tesla M2050 ve GeForce GT 430 ekran kartları üzerinde aynı uygulama, *void calculateRelativeSumOfPairsScoreKernel(char*, int*)* deklerasyonlu kernelin, CUDA platformuna aktarılmış metod kernel olarak isimlendirilmektedir, en az 34600 sefer çağrılacağı şekilde çalıştırılıp, kernel ve ekran kartı ile bilgisayar hafızası arasında veri kopyalama işlemleri için *nvprof* uygulaması [40] aracılığıyla profil sonuçları elde edilmiştir.

Tablo 5.23. Ekran kartlarının grafik işlemci, mimari ve major/minor özellikleri

Ekran Kartı	**Grafik İşlemci**	**Mimari**	**Major/Minor**
Tesla K20Xm	1 x GK110	Kepler	3.5
Tesla K20m	1 x GK110	Kepler	3.5
Tesla K10	2 x GK104	Kepler	3.0
Tesla M2090	1 x GF110	Fermi	2.0
Tesla M2050	1 x GF104	Fermi	2.0
GeForce GT 430	1 x GF104	Fermi	2.1

Tablo 5.24. Ekran kartlarının CUDA çekirdek, Multiprocessor sayıları ve çalışma hızları

Ekran Kartı	**CUDA Çekirdek Sayısı**	**Multiprocessor Sayısı**	**Clock Rate(KHz)**
Tesla K20Xm	2688	14	732000
Tesla K20m	2496	13	705500
Tesla K10	2 x 1536	8	745000
Tesla M2090	512	16	1301000
Tesla M2050	448	14	1147000
GeForce GT 430	96	1	1400000

Tablo 5.25. Ekran Kartlarının hafıza özellikleri ve blok başına paylaşılan hafıza miktarları

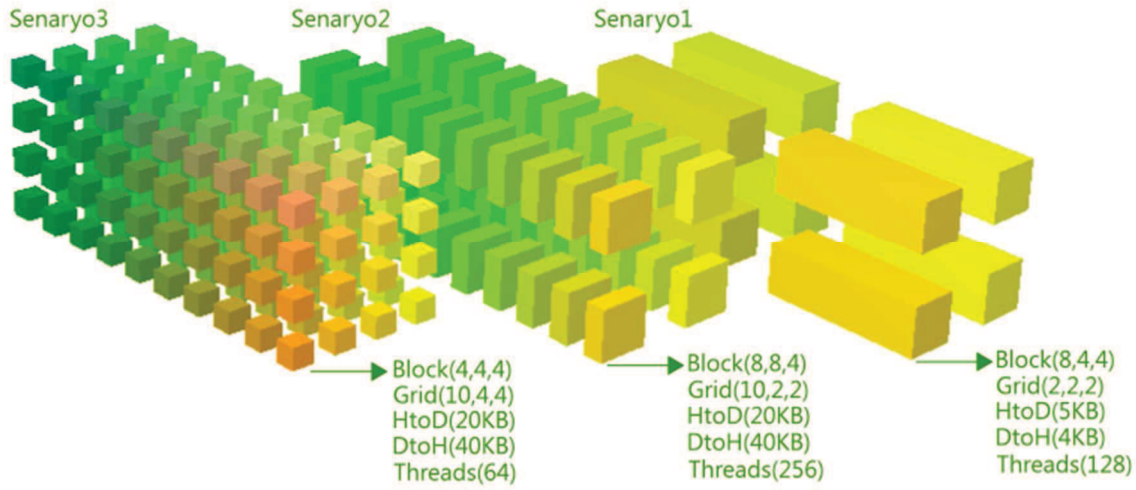
Ekran Kartı	Memory Bus Width(Bit)	Memory Clock Rate(KHz)	Shared Memory(Byte)
Tesla K20Xm	384	2600000	49152
Tesla K20m	320	2600000	49152
Tesla K10	256	2500000	49152
Tesla M2090	384	1848000	49152
Tesla M2050	384	1546000	49152
GeForce GT 430	128	533000	49152

İlk paralelleştirme senaryosunda, Taq DNA Polymerase (1taq) ailesine ait 5 KB veri, aile her birinin uzunluğu boşluk karakterleri ile birlikte 1024 yapılmış 5 sekanstan oluşmaktadır, ekran kartına, ekran kartı için CUDA platformunda device sözcüğü kullanılacaktır, kopyalanmış, kopyalanan veri, çoklu hizalama boyutları 8x4x4 biçiminde seçilmiş blok, buna göre her blok toplamda 128 thread barındırmaktadır, ve boyutları 2x2x2 şeklinde seçilmiş grid, buna göre kernel toplamda 8 bloğa sahiptir, kullanılarak SPS skoru hesaplanmış, 4 KB (KiloByte) skor verisi, her bir hizalama 1024 karakter uzunluktaydı, ekran kartından (device) kodun seri kısmının çalıştırıldığı sisteme, bu sistem için CUDA platformunda host sözcüğü kullanılacaktır, aktarılmıştır.

İkinci paralelleştirme senaryosunda, Taq DNA Polymerase ailesine ait 20 KB veri, aile her birinin uzunluğu boşluk karakterleri ile birlikte 1024 yapılmış 5 sekansın tüm ikili gruplarından oluşan veri kümesi toplamda 20 KB yapmaktadır, ekran kartına kopyalanmış, kopyalanan veri, çoklu hizalama boyutları 8x8x4 biçiminde seçilmiş blok, buna göre her blok toplamda 256 thread barındırmaktadır, ve boyutları 10x2x2 şeklinde seçilmiş grid, buna göre kernel toplamda 40 bloğa sahiptir, kullanılarak SPS skoru hesaplanmış, 40 KB skor verisi, aile uzunlukları boşluk karakterleri ile birlikte 1024 yapılmış 5 sekansın tüm ikili gruplar için türetilen 1024 elemanlı tam sayı vektörleri toplamda 40 KB yapmaktadır, ekran kartından (device) kodun seri kısmının çalıştırıldığı sisteme (host) aktarılmıştır.

Son paralelleştirme senaryosunda, Taq DNA Polymerase ailesine ait 20 KB veri, aile her birinin uzunluğu boşluk karakterleri ile birlikte 1024 yapılmış 5 sekansın tüm ikili gruplarından oluşan veri kümesi toplamda 20 KB yapmaktadır, ekran kartına

kopyalanmış, kopyalanan veri, çoklu hizalama boyutları 4x4x4 biçiminde seçilmiş blok, buna göre her blok toplamda 64 thread barındırmaktadır, ve boyutları 10x4x4 şeklinde seçilmiş grid, buna göre kernel toplamda 160 bloğa sahiptir, kullanılarak SPS skoru hesaplanmış, 40 KB skor verisi, aile uzunlukları boşluk karakterleri ile birlikte 1024 yapılmış 5 sekansın tüm ikili gruplar için türetilen 1024 elemanlı tam sayı vektörleri toplamda 40 KB yapmaktadır, ekran kartından (device) kodun seri kısmının çalıştırıldığı sisteme (host) aktarılmıştır. Aşağıda üç paralelleştirme senaryosuna ait thread-block-grid hiyerarşisi ve device ile host arasında her hesaplamada aktarılan veri boyutları gösterilmektedir.



Şekil 5.9 Paralelleştirme senaryolarına ait thread-block-grid yapıları.

İlk paralelleştirme senaryosunda, 128 thread'lik 8 bloktan oluşan kernel belirtilen boyutlarda veri transferleri kullanılarak Tesla M2050, Tesla M2090, Tesla K10, Tesla K20 ve Tesla K20X profesyonel kartlar ile giriş seviyesinde CUDA destekli GeForce GT 430 ekran kartları üzerinde çalıştırılmıştır. Device ile host arasında yapılan kopyalamaların boyutları düşünüldüğünde, bu işlemler için geçen sürenin diğer iki senaryoya göre daha az olması beklenebilir. Ancak blok ve thread sayıları dikkate alındığında, Streaming Multiprocessor'lerden bazılarının boşta (idle) tutulabileceği, bu cihazın istenilen Streaming Multiprocessor doluluklarına (occupancy) erişilemeyeceği anlamına gelebilir [37, 38, 40].

Tablo 5.26'da ilk paralelleştirme senaryosuna ait Yapay Arı Koloni Algoritması'nın önerilen modelinde en az 34600 defa SPS skorunun hesaplanması işlemi için kullanılan

CUDA uygulamasının çağrılma sayısı üzerinden ortalama değerler alınarak elde edilmiş profil sonuçları gösterilmektedir.

Tablo 5.26. İlk Paralelleştirme senaryosununa ait profil sonuçları

Ekran Kartı	DeviceToHost(DToH)(µS)	HostToDevice(HToD)(µS)	Kernel(µS)
Tesla K20Xm	3.10	2.51	7.39
Tesla K20m	3.01	2.47	7.66
Tesla K10	2.91	1.82	6.74
Tesla M2090	3.20	2.32	3.25
Tesla M2050	2.76	1.64	3.75
GeForce GT 430	7.18	6.10	9.24

İkinci paralelleştirme senaryosunda, 256 thread'lik 40 bloktan oluşan kernel belirtilen boyutlarda veri transferleri kullanılarak Tesla M2050, Tesla M2090, Tesla K10, Tesla K20 ve Tesla K20X profesyonel kartlar ile giriş seviyesinde CUDA destekli GeForce GT 430 ekran kartları üzerinde çalıştırılmıştır. Device ile host arasında yapılan kopyalamaların boyutları düşünüldüğünde, bu işlemler için geçen sürenin ilk senaryoya göre fazla olması beklenebilir. Ancak blok ve thread sayıları dikkate alındığında, Streaming Multiprocessor'lerin ilk senoryaya göre daha çok meşgul edilmesi olasıdır [37, 38, 40]. Tablo 5.27'de ikinci paralelleştirme senaryosuna ait Yapay Arı Koloni Algoritması'nın önerilen modelinde en az 34600 defa SPS skorunun hesaplanması işlemi için kullanılan CUDA uygulamasının çağrılma sayısı üzerinden ortalama değerler alınarak elde edilmiş profil sonuçları gösterilmektedir.

Tablo 5.27. İkinci Paralelleştirme senaryosuna ait profil sonuçları

Ekran Kartı	DeviceToHost(DToH)(µS)	HostToDevice(HToD)(µS)	Kernel(µS)
Tesla K20Xm	8.48	4.42	3.62
Tesla K20m	8.55	4.58	3.87
Tesla K10	6.33	8.95	3.69
Tesla M2090	14.21	7.28	2.05
Tesla M2050	9.37	5.01	2.81
GeForce GT 430	28.28	24.20	18.28

Son paralelleştirme senaryosunda, 64 thread'lik 160 bloktan oluşan kernel belirtilen boyutlarda veri transferleri kullanılarak Tesla M2050, Tesla M2090, Tesla K10, Tesla

K20 Tesla K20X profesyonel kartlar ile giriş seviyesinde CUDA destekli GeForce GT 430 ekran kartları üzerinde çalıştırılmıştır. Device ile host arasında yapılan kopyalamaların boyutları düşünüldüğünde, bu işlemler için geçen sürenin ilk senaryoya göre daha fazla, ikinci senaryoya göre hemen hemen eşit olması beklenebilir. Blok ve thread sayıları dikkate alındığında, Streaming Multiprocessor'lerin ilk senoryaya göre daha çok meşgul edilmesi olasıdır, ancak blok sayılarının fazlalığı tamamlanan blokların yenileri ile değiştirilmesi işlemlerinin maliyetini, kernelin çalışma süresine ekleyecektir [37, 38, 40]. Tablo 5.28'de son paralelleştirme senaryosuna ait Yapay Arı Koloni Algoritması'nın önerilen modelinde en az 34600 defa SPS skorunun hesaplanması işlemi için kullanılan CUDA uygulamasının çağrılma sayısı üzerinden ortalama değerler alınarak elde edilmiş profil sonuçları gösterilmektedir.

Tablo 5.28. Son paralelleştirme senaryosuna ait profil sonuçları

Ekran Kartı	**DeviceToHost(DToH)(µS)**	**HostToDevice(HToD)(µS)**	**Kernel(µS)**
Tesla K20Xm	8.51	4.42	3.56
Tesla K20m	8.58	4.57	3.75
Tesla K10	6.35	8.94	3.53
Tesla M2090	14.22	7.28	2.74
Tesla M2050	9.36	5.00	3.25
GeForce GT 430	28.29	24.22	25.32

Her üç senaryoda da, device ile host arasında kopyalanacak veri miktarları ve kopyalama maliyetleri ihmal edildiğinde, Tesla M2090 NVIDIA ekran kartı ile en başarılı profil sonuçlarının elde edildiği görülebilir. Bu durum, veri kopyalama işlemlerine olan ihtiyacın en aza indirgendiği, yani uygulamanın tamamen ekran kartı üzerine aktarılacağı durumlarda CUDA platformunun sağlayacağı performans artışı için önemli bir göstergedir.

Aynı metodun host üzerinde 0.04 ms yaklaşık 40 µs sürede çalıştığı, Tablo 5.22'de uygulamanın kernel barındırmadan çalıştırıldığında elde edilmiş profil sonuçları gösterilmekteydi, dikkate alındığında 3.25 µs, 2.05 µs ve 2.74 µs çalışma performanslarıyla Tesla M2090 NVIDIA ekran kartları metodun genel performansını %95'a yakın geliştirmektedir. Tablo 5.29'da kopyalama maliyetleri ve kernel çalışma sürelerinin tamamı dikkate alınarak Tesla M2050, Tesla M2090, Tesla K10, Tesla K20

ve Tesla K20X profesyonel kartlar ile giriş seviyesinde CUDA destekli GeForce GT 430 ekran kartlarının üç senaryo için performansları gösterilmiştir.

Tablo 5.29. Kopyalama maliyetleri ve kernel çalışma sürelerinin toplamı

Ekran Kartı	**Senaryo1(µS)**	**Senaryo2(µS)**	**Senaryo3(µS)**
Tesla K20Xm	13.00	**16.52**	**16.49**
Tesla K20m	13.14	17.00	16,90
Tesla K10	11.47	18.97	18,82
Tesla M2090	8.77	23.54	24.24
Tesla M2050	**8.15**	17.19	17.61
GeForce GT 430	22.52	70.76	77.83

Veri kopyalama işlemlerinin etkisi de kernel çalışma süresi üzerine eklendiğinde, ekran kartlarının performansları için daha belirleyici olan iki ve üçüncü senaryolarda Tesla K20Xm beklendiği üzere en kaliteli performansı göstermiştir. Aynı metodun host üzerinde 0.04 ms yaklaşık 40 µs sürede çalıştığı dikkate alındığında üçüncü paralelleştirme senaryosunda %58.7, ikinci paralelleştirme senaryosunda ise %58.9 performans kazancı sağlanmaktadır. Giriş seviyede CUDA destekli olup, profesyonel ekran kartları ile arasındaki performans farkını görebilmek amacıyla test sürecine dahil edilen GeForce GT 430 NVIDIA ekran kartı, uygun paralelleştirme yaklaşımı ve veri kopyalama maliyetleri ile, ilk paralelleştirme senaryosunda GeForce GT 430 40 µs'nin altında çalışmaktadır, uygulamanın çalışma performansını arttırmıştır.

Son paralelleştirme senaryosunda blok başına thread sayısı azaltılmış ancak ikinci paralelleştirme senaryosu ile aynı işi yapabilmek adına blok sayısı arttırılmıştır. Artan blok ve azalan thread sayısı blok başına işlem zamanının her zaman azaltılacağı anlamına gelmemektedir [37, 38]. Tesla K20Xm, Tesla K20X ve Tesla K10 artan blok sayısını Streaming Multiprocessor'ler üzerine dağıtma işlemini, Tesla M2090 ve Tesla M2050 kartlarına kıyasla önerilen iki ve üçüncü senaryolarda daha iyi yönetebilmiştir.

6. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

Biyoinformatik'te Çoklu Dizi Hizalama Problemi olarak isimlendirilmiş ayrık türdeki optimizasyon probleminin çözümünde, Çoklu Dizi Hizalama Problemi'nin çözülmesi Biyoinformatik'te tanımı yapılmış diğer pek çok problem için de önemlidir, filogenetik, soy ağaçlarının üretilmesinden, ikincil ya da üçüncül protein yapı tahminlerine, üç boyutlu molekül yapılarının oluşturulmasından ilaç üretimine problem sahasını genişletmek mümkündür, bal arılarının yiyecek, nektar arama davranışlarında gösterdiği zekiliği modelleyen Yapay Arı Koloni Algoritması'nın yeni hali kullanılmıştır ve elde edilen sonuçların Yapay Arı Koloni Algoritması, Genetik Algoritma ile Parçacık Sürü Optimizasyon Algoritması kullanılarak daha önce yapılmış çalışmalarla ve Genetik Algoritma tabanlı SAGA (Sequence Alignment By Genetic Algorithm) ile dinamik programlama tabanlı CLUSTALW uygulamamalarından elde edilen sonuçlarla farklı skor metrikleri üzerinden karşılaştırılmıştır. Yapay Arı Koloni Algoritması için önerilen modelin CUDA platformuna aktarılabilirliği ve paralelleştirilebilirliği hususunda yapılan profil çalışmalarıyla tez çalışması sonlandırılmıştır.

6.1 Sonuç ve Öneriler

Önerilen Yapay Arı Koloni Algoritması'nın, başlangıç çoklu hizalamalarının üretilmesi ve komşuluk tabanlı aday çözüm oluşturma aşamaları modellenirken, Yapay Arı Koloni Algoritması'nın nümerik türdeki optimizasyon problemleri için sunulan formülasyonlarına mümkün olduğunca bağlı kalınmaya çalışılmıştır. Yapay Arı Koloni Algoritması'nın sağlam ve esnek yapısı, işçi, gözcü, kâşif arı fazları için önerilen yaklaşımlar ile birleşince, BAliBASE protein veri tabanından alınan 9 aile için karşılaştırıldığı diğer algoritmalar arasında en iyi, 3D_ali protein veri tabanından

alınan 9 aile için ise diğer uygulamalarla karşılaştırılabilir kalitede çözümler elde edildiği gözlemlenmiştir. Bulunan sonuçların geneli üzerinden yapılacak değerlendirmede ise önerilen Yapay Arı Koloni Algoritması'nın, benzerlik oranları %20 ile %40, sekans uzunlukları 50 ile 300 ve son olarak da sekans sayısı 5 ile 10 arasında değişen protein ailelerinde daha başarılı sonuçlar ürettiği görülmüştür. Bu durum, sekans uzunlukları ve sekans sayıları açısından belirtilen sınırları aşan çoklu hizalamaların önerilen Yapay Arı Koloni Algoritması ile çözüldüğü uygulamaların, önce alt çoklu hizalamalar üretmesi sonra alt çoklu hizalamaların birleştirilerek hizalamanın genel skorunun elde edilmesi şeklinde bir yöntemin kullanılabilirliğini akla getirmektedir.

Elde edilecek çoklu hizalamaların kalitesini arttırabilmek adına, birden fazla başlangıç besin kaynağı oluşturma stratejisinin birleştirilmesi yaklaşımı, komşuluk tabanlı çoklu hizalama üretme stratejisinde de kullanılabilir. Komşuluk tabanlı çözümlerin üretildiği yaklaşımlar tek bir uygulamada beraber kullanılırsa Yapay Arı Koloni Algoritması'nın yerel ve global arama özelliklerine Çoklu Dizi Hizalama Problemi için katkıda bulunulabileceği düşünülmektedir.

Çoklu Dizi Hizalama Problemi'nin çözüldüğü diğer uygulamalarda da olduğu üzere, hizalanacak sekans sayısı, sekanslara ait uzunluklar arttıkça önerilen Yapay Arı Koloni Algoritması ile de kaliteli hizalamaların kabul edilebilir çalışma zamanlarında elde edilme olasılığı azalmaktadır. Kaliteli ve hızlı bir şekilde sekansların hizalanabilmesi adına paralelleştirilebilir algoritmik yaklaşımların kullanılması veya geliştirilmesi oldukça önemlidir.

Önerilen Yapay Arı Koloni Algoritması'nın SPS skorunun hesaplandığı metodu, NVIDIA firması tarafından 2006 yılının sonlarına doğru duyurulmuş CUDA isimli yeni sayılabilecek paralel hesaplama platformuna aktarılmış, Tesla M2050, Tesla M2090, Tesla K10, Tesla K20 ve Tesla K20X profesyonel kartları ile giriş seviyesinde CUDA destekli GeForce GT 430 kartı üzerinde farklı paralelleştirme senaryolarının sonuçları karşılaştırmalı olarak incelenmiştir.

Tek bir metodu CUDA platformuna aktarılan Yapay Arı Koloni Algoritması'nın, ekran kartlarında elde edilen çalışma süreleri, uygun paralelleştirme senaryosu

kullanıldığında, seri kodlara kıyasla %60'a yakın performans kazancı sağlamıştır. Bu durum, önerilen algoritmanın veri bağımlılığının azaltılıp, tamamen CUDA platformuna aktarıldığında en az iki ya da üç kat performans artışı sağlayacağı, ekran kartları kullanılarak oluşturulacak hibrit hesaplama mimarilerinin kullanılabilirliğinin araştırılması için planlar yapılmasını sağlamıştır.

Laboratuvar çalışmalarında türlere ait tam gen haritalarının elde edilmesiyle hizalama işlemi sadece belirli proteinlerin sentezinden sorumlu DNA ya da RNA sekansları üzerinden değil, tüm genomlar kullanılarak yapılmaya başlanmıştır. Basit bakteri formlarına ait genomların bile bir kaç milyon baz içeriyor olması, daha kompleks canlı türleri için bu sayı trilyonlara kadar çıkmaktadır, sınırlı sayıda uzunluk ve sekans için bile NP-Complete olduğu kabul edilen Çoklu Dizi Hizalama Problemi'nin standart donanımlar ile çözülebilmesini imkansız kılacağından, paralelleştirme senaryoları oldukça önemli hale gelecektir. Tek bir metodu CUDA platformuna aktarılan Yapay Arı Koloni Algoritması'nın, ekran kartlarında elde edilen çalışma süreleri uygulanabilir paralelleştirme senaryolarının problemin çözümü için oldukça önemli olacağını göstermektedir.

Yapay Arı Koloni Algoritması'nın önerilen modeli Yapay Arı Koloni Algoritması, Genetik Algoritma, Parçacık Sürü Algoritması gibi sezgisel algoritmalar ile daha önceden yapılmış çalışmalarla ve Genetik Algoritma tabanlı SAGA ile dinamik programlama tabanlı CLUSTALW uygulamaları kullanılarak yapılmış çalışmalar üzerinden karşılaştırılmıştı. Önerilen modelin izci, gözcü, kâşif arı fazları için kullanılan yaklaşımları, Yapay Arı Koloni Algoritması'nın yerel ve global araştırma özelliklerine katkıda bulunacak şekilde yeni düzenlemeler yapılarak farklı sürü zekâsı temelli ya da evrimsel algoritmalar ile karşılaştırılarak, birden fazla algoritmanın kullanıldığı hibrit yapıların oluşturulabilmesi adına gerekli çalışmaların yapılması hedeflenmektedir.

KAYNAKÇA

1. Huerta M., Haseltine F., Liu Y., Downing G., Seto B., 2000. **NIH working definition of bioinformatics and computational biology**.
2. Karaboğa D., 2005. **An idea based on bee swarm for numerical optimization**. Technical Report.
3. Karaboğa D., 2011. Yapay Zekâ Optimizasyon Algoritmaları. Nobel Yayın Dağıtım, Ankara, 232pp.
4. Cohen, J., 2004. Bioinformatics-an introduction to computer scientists. ACM Computing Surveys, 36 (2): 122-158.
5. Zvelebil, M., Baum, J., O., 2008. Understanding Bioinformatics. Garland Science, New York, 772pp.
6. Krawetz, S., A., Womble, D., D., 2003. Introduction to Bioinformatics A Theorical and Practical Approach. Human Press, New Jersey, 762pp.
7. Lesk, A. M., 2002. Introduction to Bioinformatics. Oxford University Press **15** (3): Incorporation, Oxford, 255pp.
8. Alberts, B., Johnson A., Lewis J., Raff, M., Roberts K., Walter P., 2007. Molecular Biology of the Cell. Garland Science, New York, 1392pp.
9. Cormen T., H., Leiserson C., E., Rivest R., L., Stein C., 2009. Introduction to Algorithms. The MIT Press, Cambridge, 1299pp.
10. Ezzihane Z., 2006. Applications of artificial intelligence in bioinformatics: a review. **Expert Systems with Applications, 30** (1): 2-8.
11. Karaboga, D., 2010. Artificial Bee Colony Algorithm. Scholarpedia, 5 (3): 6915.
12. Karaboga, D., Basturk B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. **Journal of Global Optimization, 39** (3): 459-471.
13. Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (abc) algorithm. **Applied Soft Computing, 8** (1): 687-697.
14. Karaboga, D., Basturk, A., 2011. A modified artificial bee colony algorithm (abc) for constrained optimization problems. **Applied Soft Computing, 11** (3): 3021-3031.

15. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2012. A comprehensive survey: artificial bee colony (abc) algorithm and applications. **Artificial Intelligence Review**, In Press.
16. Karaboga, D., Gorkemli B., 2011. A combinatorial artificial bee colony algorithm for traveling salesman problem. INISTA 2011: Internationa Symposium on Innovations in Intelligence Systems and Applications, 15-18 June, 2011, Istanbul, Turkey.
17. Bucak, İ., Ö., Uslan, V., 2011. Sequence alignment from the perspective of stochastic optimization: a survey. **Turkish Journal of Electrical Engineering & Computer Sciences, 19** (1): 157-173.
18. Omar, M., F., Salam, R., A., Abdullah, R., Rashid, N., A., 2005. Multiple sequence alignment using optimization algorithms. **Internation Journal of Computational Intelligence, 1**: 81-89.
19. Notredame, C., 2002. Recent progress in multiple sequence alignment: a survey. **Pharmagenomics, 3** (1): 131-144.
20. Altschul, S., F., Carroll R., J., Lipman D., J., 1989. Weights for data related by a tree. **Journal of Molecular Biology, 207**: 647-653.
21. Myers, E., W., Miller, W., 1988. Multiple sequence alignment using simulated annealing. **Computer Applications for Bioscience, 4** (1): 11-17.
22. Isikhawa, M., Toya, T., Hoshida M., Nitta, K., Ogiwara, A., Kanehisa, M., 1993. Multiple sequence alignment by parallel simulated annealing. **Computer Applications for Bioscience, 9**: 419-426.
23. Isikhawa, M., Wayama, M., Shimizu, T., 1996. Multiple sequence alignment using a genetic algorithm. **Genome Information, 7**: 176-186.
24. Zhang, C., Wong A., K., C., 1997. A genentic algorithm for multimolecular sequence alignment. **Computer Applications for Bioscience, 13** (6): 565-581.
25. Anbarasu, L., A., Narayanasamy, P., Sundararajan, V., 1998. Multiple sequence alignment using parallel genetic algorithms. **Simulated Evolution and Learing Lecture Notes in Computer Science**, 1585: 130-137.
26. Notredame, C., Higgins, D., G., 1996. SAGA: Sequence alignment by genetic algorithm. **Nucleic Acid Research, 24** (8): 1515-1524.
27. Notredame, C., O'Brien, E., A., Higgins, D., G., 1997. RAGA: RNA sequence alignment using genetic algorithm. **Nucleic Acid Research, 25** (22): 4570-4580.

28. Notredame, C., 2003. Chapter-5 Using genetic algorithms for pairwise and multiple sequence alignments. Evolutionary Computation in Bioinformatics, 87-111.
29. Notredame C., Holm L., Higgins, D., G., 1998. COFFEE: an objective function for multiple sequence alignments. **Bioinformatics, 14** (5): 407-422.
30. Lee, Z.-J., Su, S.-F., Chuang, C.-C., Liu, K.-H., 2008. Genetic algorithm with ant colony optimization (ga-aco) for multiple sequence alignment. **Applied Soft Computing, 8** (1): 55-78.
31. Xu, F., Chen, Y., 2009. A method for multiple sequence alignment based on particle swarm optimization. ICIC'09 Proceedings of the Intelligent computing 5th international conference on Emerging intelligent computing technology and applications, Heidelberg, 965-973.
32. Lei, X., Sun, J., Xu, X., Guo, L., 2010. Artificial bee colony for solving multiple sequence alignment. Bio-inspired Computing: Theories and Applications 2010 IEEE Fifth International Conference, 337-342.
33. Thompson, J., D., Plewniak, F., Poch, O., 1999. A comprehensive comparison of multiple sequence alignment programs. **Nucleic Acid Research, 27** (13): 2682-2690.
34. Thompson, J., D., Plewniak, F., Poch, O., 1999. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. **Bioinformatics Applications Note, 15** (1): 87-88.
35. Larkin, M., A., Blackshields, G., Brown, N., P., Chenna, R., 2007. Clustal W and Clustal X version 2.0. **Bioinformatics Applications Note, 23** (21): 2947-2948.
36. Gough, B., 2004. An Introduction to GCC for the GNU Compilers gcc and g++. Network Theory Limited, Bristol, 116pp.
37. NVIDIA, 2012. C Programming Guide-Design Guide. 185pp.
38. NVIDIA, 2012. C Best Practises Guide-Design Guide. 75pp.
39. NVIDIA 2012. CUDA API Reference Manual-Version 5.0. 680pp.
40. NVIDIA, 2012. Profiler-User's Guide, 53pp.
41. Farber, R., 2011. CUDA Application Design and Development. Morgan Kauffman, Waltham, 315pp.
42. Cook, S., 2013. CUDA Programming-A Developer's Guide to Parallel Computing with GPUs. Morgan Kauffmann, Waltham, 576pp.

EKLER

EK-A. SPS Skorları Üzerinden Karşılaştırılan Protein Aileleri İçin Bulunan Skorlar

Tablo EK-A.1 SH3 Protein Ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: SH3		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.925926	0.419753
2	0.913580	0.419753
3	0.987654	0.432099
4	0.950617	0.419753
5	1.037037	0.530864
6	1.012346	0.419753
7	0.913580	0.543210
8	0.925926	0.444444
9	0.962963	0.469136
10	1.000000	0.407407

Tablo EK-A.2 Cytochrome C Protein Ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Cytochrome C		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.854369	0.388350
2	0.834951	0.427184
3	0.815534	0.441748
4	0.825243	0.334951
5	0.805825	0.446602
6	0.839806	0.388350
7	0.844660	0.393204
8	0.791262	0.402913
9	0.844660	0.485437
10	0.825243	0.402913

Tablo EK-A.3 Serine Protease protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Serine Protease		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	1.011730	0.495601
2	0.947214	0.431085
3	0.973607	0.624633
4	0.964809	0.466276
5	0.950147	0.577713
6	0.967742	0.662757
7	0.970674	0.331378
8	1.002933	0.568915
9	1.011730	0.319648
10	0.997067	0.410557

Tablo EK-A.4 Protein Kinase protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Protein Kinase		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.594747	0.378987
2	0.598499	0.360225
3	0.617261	0.384615
4	0.598499	0.380863
5	0.619137	0.388368
6	0.613508	0.350844
7	0.585366	0.371482
8	0.596623	0.324578
9	0.619137	0.360225
10	0.613508	0.362101

Tablo EK-A.5 Anthrailate Isomerase protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Anthranilate Isomerase		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.605664	0.313725
2	0.607843	0.366013
3	0.618736	0.318083
4	0.677560	0.265795
5	0.612200	0.339869
6	0.655773	0.331155
7	0.623094	0.300654
8	0.640523	0.435730
9	0.625272	0.267974
10	0.594771	0.259259

Tablo EK-A.6 Serine Protease protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Serine Protease		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.807692	0.263027
2	0.792804	0.286600
3	0.806452	0.276675
4	0.807692	0.404467
5	0.769231	0.236973
6	0.772953	0.338710
7	0.823821	0.300248
8	0.794045	0.269231
9	0.795285	0.248139
10	0.785360	0.228288

Tablo EK-A.7 Aminotransferase protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Aminotransferase		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.709677	0.425806
2	0.703226	0.470968
3	0.729032	0.516129
4	0.738710	0.500000
5	0.706452	0.458065
6	0.674194	0.477419
7	0.690323	0.561290
8	0.729032	0.483871
9	0.732258	0.461290
10	0.703226	0.464516

Tablo EK-A.8 Glutamyl Synthetase protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Glutamyl-TRNA Synthetase		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.667614	0.217330
2	0.585227	0.269176
3	0.626420	0.271307
4	0.620028	0.229403
5	0.659801	0.271307
6	0.617898	0.272727
7	0.625710	0.279830
8	0.625710	0.280540
9	0.583807	0.241477
10	0.592330	0.227273

Tablo EK-A.9 Taq DNA Polymerase protein ailesinin on farklı koşmasına ait SPS skorları

Protein Ailesi: Taq DNA Polymerase		
Koşma Numarası	**En Yüksek SPS Skoru**	**En Düşük SPS Skoru**
1	0.550192	0.172342
2	0.578620	0.173823
3	0.525022	0.181818
4	0.561149	0.186556
5	0.521469	0.206988
6	0.564406	0.180041
7	0.506663	0.212615
8	0.535683	0.187148
9	0.521765	0.225940
10	0.524726	0.225940

EK-B. COFFEE Skorları Üzerinden Karşılaştırılan Protein Aileleri İçin Bulunan Skorlar

Tablo EK-B.1 Acid Protease Protein Ailesinin COFFEE skorları

Protein Ailesi: Acid Protease		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.321224	0.099370
2	0.359381	0.097743
3	0.307112	0.080113
4	0.335934	0.117457
5	0.332220	0.083368
6	0.341627	0.087143
7	0.313809	0.090130
8	0.312287	0.085588
9	0.362842	0.093429
10	0.348333	0.094137

Tablo EK-B.2 Sugar/Amino Acid Binding Protein Ailesinin COFFEE skorları

Protein Ailesi: Sugar/Amino Acid Binding		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.750460	0.114810
2	0.752936	0.103156
3	0.662991	0.041883
4	0.777675	0.094199
5	0.751507	0.080144
6	0.696247	0.188852
7	0.790084	0.164053
8	0.737082	0.104571
9	0.774601	0.115569
10	0.783053	0.050987

Tablo EK-B.3 Cytochrome C protein ailesinin COFFEE skorları

Protein Ailesi: Cytochrome C		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.664939	0.207461
2	0.542580	0.209376
3	0.615417	0.103920
4	0.573330	0.166249
5	0.660057	0.153845
6	0.676544	0.125951
7	0.589681	0.138100
8	0.574446	0.088208
9	0.625603	0.088697
10	0.651839	0.219409

Tablo EK-B.4 Fibronectin Type III Protein Ailesinin COFFEE skorları

Protein Ailesi: Fibronectin Type III		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.556077	0.166053
2	0.509706	0.133820
3	0.530942	0.141592
4	0.584063	0.228871
5	0.527068	0.191545
6	0.541109	0.147499
7	0.505568	0.103387
8	0.545874	0.163565
9	0.522301	0.152401
10	0.556061	0.223730

Tablo EK-B.5 Crystallins protein ailesinin COFFEE skorları

Protein Ailesi: Crystallins		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0,943563	0,374293
2	0,949420	0,449389
3	0,943737	0,241655
4	0,949420	0,216938
5	0,944970	0,437282
6	0,943563	0,188227
7	0,925125	0,434071
8	0,931356	0,171570
9	0,929970	0,386041
10	0,949420	0,463477

Tablo EK-B.6 Globins/Phycocyanins/Collicins protein ailesinin COFFEE skorları

Protein Ailesi: Globins/Phycocyanins/Collicins		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.534636	0.154630
2	0.504813	0.183247
3	0.471329	0.100357
4	0.489314	0.140483
5	0.503518	0.120663
6	0.518194	0.169902
7	0.513442	0.147525
8	0.471478	0.055318
9	0.535388	0.061226
10	0.562535	0.103181

Tablo EK-B.7 Lyzozymes/Lactalbumin protein ailesinin COFFEE skorları

Protein Ailesi: Lyzozymes/Lactalbumin		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.643082	0.102409
2	0.548796	0.089233
3	0.636347	0.125402
4	0.678660	0.094343
5	0.717953	0.088790
6	0.518194	0.159182
7	0.646748	0.102118
8	0.817407	0.143960
9	0.720932	0.111089
10	0.720240	0.036169

Tablo EK-B.8 Dihydroxybiphenyl Dioxygenase protein ailesinin COFFEE skorları

Protein Ailesi: Dihydroxybiphenyl Dioxygenase		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0,708863	0,183342
2	0,716398	0,339560
3	0,799573	0,158932
4	0,726340	0,212218
5	0,739943	0,244235
6	0,787858	0,337146
7	0,760874	0,323991
8	0,726956	0,334837
9	0,781667	0,210932
10	0,716966	0,145735

Tablo EK-B.9 Subtilisin Protein Ailesinin COFFEE skorları

Protein Ailesi: Subtilisin		
Koşma Numarası	**En Yüksek COFFEE Skoru**	**En Düşük COFFEE Skoru**
1	0.801121	0.108021
2	0.757405	0.137278
3	0.794497	0.252171
4	0.741059	0.089029
5	0.767853	0.136212
6	0.844966	0.208061
7	0.805618	0.199345
8	0.770111	0.143137
9	0.798698	0.215751
10	0.770112	0.166866

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Selçuk Aslan

Doğum Tarihi ve Yeri: 2 Şubat 1988, Kayseri

email: selcukaslan@erciyes.edu.tr, selcuk.aslan@bil.omu.edu.tr

Yazışma Adresi: Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü 38039 Melikgazi/KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi Fen Bilimleri Enstitüsü	2013
Lisans	Erciyes Üniversitesi Bilgisayar Mühendisliği	2011

İŞ DENEYİMİ

Yıl	Kurum	Görev
2012-	Erciyes Üniversitesi Fen Bilimleri Enstitüsü	Araştırma Görevlisi
2011–2012	Ondokuz Mayıs Üniversitesi, Mühendislik Fakültesi, Bilgisyar Mühendisliği	Araştırma Görevlisi