

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

YAPAY ZEKÂ VE NANO-TOPAKLAR;
GENETİK ALGORİTMA UYGULAMASI

Enes AYAN

AĞUSTOS 2015

Bilgisayar Mühendisliği Anabilim Dalında Enes AYAN tarafından hazırlanan YAPAY ZEKÂ VE NANO-TOPAKLAR; GENETİK ALGORİTMA UYGULAMASI adlı Yüksek Lisans Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.



Prof. Dr. Hasan ERBAY
Anabilim Dalı Başkanı

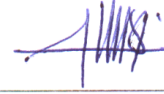
Bu tezi okuduğumu ve tezin **Yüksek Lisans Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.



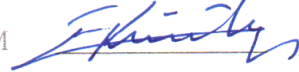
Danışman

Jüri Üyeleri

Başkan : Doç. Dr. Murat ATIŞ



Üye (Danışman) : Prof. Dr. Erdem Kamil YILDIRIM



Üye : Yrd. Doç. Dr. Taner TOPAL



.../.../...

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onaylamıştır.



Prof. Dr. Mustafa YİĞİTOĞLU

Fen Bilimleri Enstitüsü Müdürü

ÖZET

YAPAY ZEKÂ VE NANO TOPAKLAR; GENETİK ALGORİTMA UYGULAMASI

AYAN, Enes

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi

Danışman: Prof. Dr. Erdem Kamil YILDIRIM

Ağustos 2015, 50 sayfa

Teorik ve deneysel olarak nano topaklar son yıllarda oldukça fazla ilgi çekmeye başladı. Benzer bulk materyallerle karşılaştırıldığında çok farklı kimyasal ve fiziksel özelliklere sahip olabilmektedirler. Nano topak yapıları nano malzemelerin temel taşları olduklarından geometrilerinin ve kararlı yapılarının bulunması oldukça önemlidir. Nano materyallerin istikrarlı geometrilerini tahmin etmek için, araştırmacılar farklı yöntemler kullanmışlardır bunlardan bazıları; monte carlo, moleküler dinamik, rassal arama metotları, genetik algoritmalar ve benzetim tavlama algoritmasıdır.

Bu çalışmada, tek tip atomdan oluşan topak yapılarının istikrarlı geometrilerini tahmin etmek için daha önce geliştirilmiş genetik algoritma kodlarından ayrı olarak çaprazlama ve mutasyon operatörleri farklı genetik algoritma kodu geliştirildi. Geliştirilen kod DFT (Yoğunluk Fonksiyon Teorisi) ve genetik algoritmayı birlikte kullanan örneklerden bir tanesidir. Bu kodun doğru çalıştığını doğrulamak amacıyla B₄, B₅, B₆, B₈, Li₅, Li₆, Li₇ topakları üzerinde test edilerek literatürdeki sonuçlar ile karşılaştırıldı. Elde edilen sonuçlar geliştirilen kodun çalıştığımız yapıların kararlı geometrilerini başarılı bir şekilde tahmin ettiğini ortaya koymuştur. Buna ek olarak geliştirilen kod hesaplama sürecindeki farklılıklardan dolayı daha önceki genetik algoritma kodlarından daha hızlı çalışmaktadır.

Anahtar Kelimeler: Nano Topaklar, Optimizasyon Algoritmaları, Geometri Optimizasyonu, Genetik Algoritma

ABSTRACT

ARTIFICIAL INTELLIGENCE AND NANO CLUSTERS; APPLICATION OF GENETIC ALGORITHM

AYAN, Enes

Kırıkkale University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering, M. Sc. Thesis

Supervisor: Prof. Dr. Erdem Kamil YILDIRIM

August 2015, 50 pages

Theoretically and experimentally nano-clusters have attracted much attention in recent years. They may have very different physical and chemical properties from their corresponding bulk materials. As they are building blocks for nano-materials finding their stable geometries are very important. To predict the stable geometries of nano-materials, researchers have used different methods such as monte carlo, molecular dynamics, stochastic search methods, genetic algorithms, simulated annealing etc.

In this study, to predict the stable geometries of mono atom nano-clusters we developed a genetic algorithm code which has different crossover and mutation process from the previously developed genetic algorithm codes. This code is one of the few examples of performing genetic algorithm+DFT (Density Functional Theory) method. We tested our code on B₄, B₅, B₆, B₈, Li₅, Li₆, Li₇ clusters and compared the results with the literature. Our results show that the code successfully predicts the stable geometries for these clusters. In addition, due to differences in the calculation process our code may be much faster than the previously developed genetic algorithm codes.

Key Words: Nano Clusters, Optimization Algorithms, Geometry Optimization, Genetic Algorithm

TEŐEKKÜR

Tezimin hazırlanması esnasında yardımlarını esirgemeyen ve büyük destek olan tez yöneticisi hocam, Sayın Prof. Dr. Erdem Kamil YILDIRIM'a, tez çalışmalarım esnasında, bilimsel konularda yardımını gördüğüm hocam Sayın Yrd. Doç. Dr. Taner TOPAL'a, üzerimde büyük emeđi olan hocam, Sayın Doç. Dr. Habil KALKAN'a, tezimin birçok aşamasında yardım gördüğüm arkadaşlarım, Enes ALTINKAYA ve Ahmet Seçkin BİLGİ'ye, tezimin düzenlemelerini yaparken yardımlarını esirgemeyen Öğr. Gör. Volkan ATEŐ'e teşekkürlerimi sunarım.

Maddi ve manevi her zaman yanımda olan desteklerini hiçbir zaman esirgemeyen aileme, özellikle gösterdiği sabır ve verdiği destekten ötürü eşime teşekkür ederim.

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	iv
İÇİNDEKİLER DİZİNİ	v
ŞEKİLLER DİZİNİ	vii
KISALTMALAR DİZİNİ	viii
1. GİRİŞ	1
1.1. Nanoteknoloji Alanındaki İlerlemeler	3
1.2. Nanoişleyiciler	4
1.3. Nanoteknoloji ve Bilgisayar Bilimleri	5
1.4. Nanoteknolojinin Kullanım Alanları	6
1.5. Nanoteknoloji Ürünleri ve Uygulamaları	7
1.5.1. Spor Ürünleri	7
1.5.2. Araç Boyaları	8
1.5.3. Anti Bakteriyel Temizleyiciler	8
1.5.4. Medikal Bandajlar	8
1.5.5. Giyim Sektörü	9
1.5.6. Güneş Kremleri ve Kozmetik Ürünleri	9
1.6. Nanoteknoloji için Gelecekte Önem Verilen Uygulama Alanları	9
2. MATERYAL VE YÖNTEM	10
2.1. Temel Kavramlar ve Genel Tanımlar	10
2.2. Genetik Algoritmalar (GA)	12
2.3. Genetik Algoritmanın Çalışma Mekanizması	14
2.3.1. Uyguluk Fonksiyonu	15
2.3.2. Genetik Algoritma Parametreleri	15
2.3.3. Kromozom Tasarımı	16
2.3.4. Başlangıç Popülasyonu	16
2.3.5. Genetik Operatörler	16
2.4. Genetik Algoritma Avantaj ve Dezavantajları	19

2.5. Nano Topak Yapılarda Optimizasyon	19
2.6. Genetik Algoritmanın Nano Topak Yapılarda Kullanımı	20
2.7. Birmingham Topak Genetik Algoritması	21
3. ARAŞTIRMA BULGULARI VE GELİŞTİRLEN YAZILIM	24
3.1. Genetik Algoritma Tasarımı	25
3.1.1. Programın Ara Yüzünün Tasarımının Tanıtımı	26
3.1.2. Kromozom Tasarımı	27
3.1.3. Uygunluk Fonksiyonu	28
3.1.4. İlk Popülasyonun Oluşturulması	29
3.1.5. Seçim Operatörü	31
3.1.6. Çaprazlama Operatörü	32
3.1.7. Mutasyon Operatörü	33
3.1.8. Elitizm	34
3.1.9. Durdurma Ölçütü	34
3.2. Programın Sınıf Yapısı ve Sınıfların Üstlendikleri Görevler	34
3.3. Programın Çalışma Aşamaları	35
4. TARTIŞMA VE SONUÇ	38
4.1. B ₄ Nano Topağı için Algoritmanın Çalıştırılması	38
4.2. B ₅ Nanotopağı için Algoritmanın Çalıştırılması	39
4.3. B ₆ Nanotopağı için Algoritmanın Çalıştırılması	39
4.4. B ₈ Nano Topağı için Algoritmanın Çalıştırılması	41
4.5. Li ₅ Nano Topağı için Algoritmanın Çalıştırılması	41
4.6. Li ₆ Nano Topağı için Algoritmanın Çalıştırılması	42
4.7. Li ₇ Nano Topağı için Algoritmanın Çalıştırılması	43
4.8. Sonuçları Değerlendirme ve Gelecekte Yapılacaklar	43
KAYNAKLAR	45
EKLER	49
EK 1. PROGRAMIN UML DİYAGRAMI	49
EK 2. PROGRAMIN DOSYA YAPISI	50

ŞEKİLLER DİZİNİ

<u>ŞEKİL</u>	<u>Sayfa</u>
1.1. Metreden nanoya doğru boyutlar	1
1.2. Nanoteknolojinin etkileşimli olduğu alanlar	7
2.1. Genetik algoritma akış şeması	13
2.2. Rulet tekeri	17
2.3. Çaprazlama yöntemi.....	18
2.4. Mutasyon yöntemi.....	19
2.5. Potansiyel enerji yüzeyi	21
2.6. Birmingham genetik algoritma akış şeması	22
2.7. Deaven ve Ho çaprazlama yöntemi	23
3.1. Programın algoritma akış şeması	25
3.2. Program arayüzü	26
3.3. Kromozom tasarımı.....	27
3.4. Linux kabuk kodu	28
3.5. DFT enerji hesabı formülü.....	29
3.6. Topak yapısı	30
3.7. İki boyutlu düzlemsel yapı.....	31
3.8. Çaprazlama.....	32
3.9. Mutasyon.....	33
3.10. Gaussian 09 com dosyası	36
4.1. B ₄ topağı sonuç ve literatür karşılaştırması.....	38
4.2. B ₅ topağı sonuç ve literatür karşılaştırması.....	39
4.3. B ₆ topağı sonuç ve literatür karşılaştırması.....	40
4.4. B ₆ topağı izomeri sonuç ve literatür karşılaştırması	40
4.5. B ₈ topağı sonuç ve literatür karşılaştırması.....	41
4.6. Li ₅ topağı sonuç ve literatür karşılaştırması.....	42
4.7. Li ₆ topağı sonuç ve literatür karşılaştırması.....	42
4.8. Li ₇ topağı sonuç ve literatür karşılaştırması.....	43

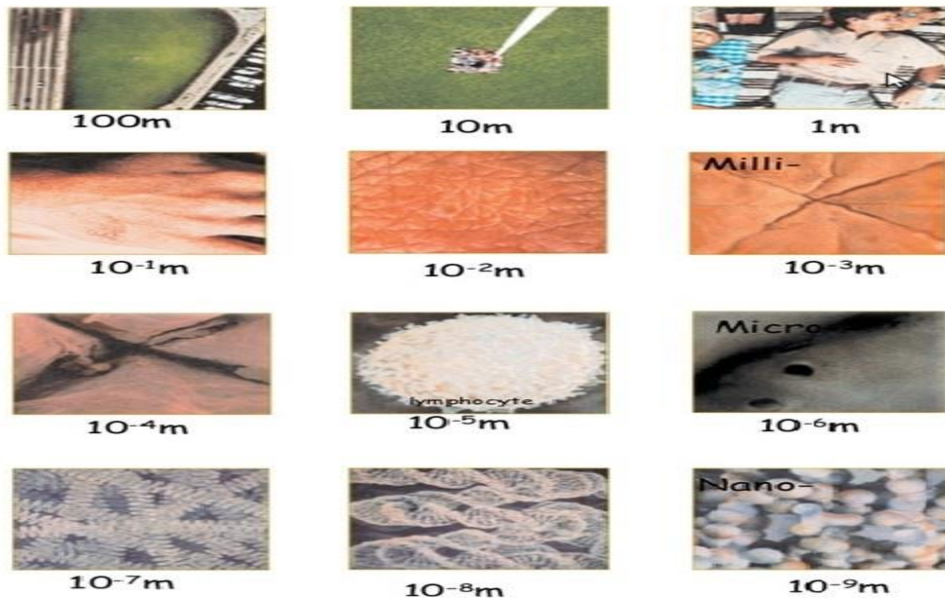
KISALTMALAR DİZİNİ

GA	Genetik Algoritma
STM	Scanning tunnelling microscope Tarama tünel mikroskopyu
SPM	Scanning probe microscope / Tarama mikroskopyu
AFM	Atomic force microscope / Atomik kuvvet mikroskopyu
Nm	nono metre
NNI	National Nanotechnology Initiative / Ulusal nanoteknoloji girişimi
UML	Unified Modeling Language / Birleşik modelleme dili
DFT	Density functional theory / Yoğunluk fonksiyon teorisi
a.u.	Atomic Units / Atomik birimler

1. GİRİŞ

Nanoteknoloji maddeleri nano ölçekte gözlemlene, ölçme, işleme ve üretme yeteneğidir [1]. Nanoteknoloji kelimesi içinde geçen nano terimi bir uzunluk ölçü birimidir. 1 nano metre (nm) 10^{-9} metreye denk gelmektedir. Bu ölçek atomlar ve moleküller düzeyini temsil etmektedir. 1-100 nanometre arasında deęişen boyuta sahip olan parçacıklara nano partikül denilmektedir. Nano kavramını daha iyi anlayabilmek için Şekil 1.1'de metreden nanoya doğru temsili bir örnek verilmiştir. Çeşitli yapıların nano boyutta ölçüleri aşağıda verilmiştir:

Atom	1 nano metre
DNA molekülü	2 nano metre
Bilgisayar içindeki bir kablo	100 nano metre
Ultraviyole ışığın dalga boyu	300 nano metre
Toz partikülleri	800 nano metre
Bakteri	1,000 nano metre
Kırmızı kan hücresi	10,000 nano metre
Saç teli	75,000 ile 100,000 nano metre



Şekil 1.1. Metreden nanoya doğru boyutlar [2]

Nano bilimler hakkında ilk tartışmayı ateşleyen 1959 yılında “There’s Plenty of Room at the Bottom.” adlı konuşmasıyla Amerikalı fizikçi Richard Feynman’dır [3]. Feynman’a göre makinelerin daha küçük boyutlarda üretilebilmesi atomik düzeyden itibaren oluşturulmaya başlanmasıyla mümkündür. Atomik düzeyde bilginin yazılabilmesi ile küp şeker boyutundaki elektronik ortamda kütüphaneler dolusu bilginin saklanacağını savunmuştur [3]. 1974 yılında Japon Bilim adamı Norio Taniguchi ilk kez nanoteknoloji terimini bilimsel bir makalede kullanmıştır [4]. 1986 yılında Kim Eric Drexler tarafından yazılan “Engines of Creation” adlı kitap nanoteknolojide dönüm noktası olmuştur [5]. Drexler nanoteknolojinin günümüzde popüler olmasının arkasındaki kişi olarak anılmaktadır. Feynman ve Drexler’in görüşleri, fizik, materyal bilimi, kimya, biyoloji ve mühendislik alanında çalışan birçok araştırmacıya ilham kaynağı olmuştur.

Teknolojiyi hayatımıza kolaylıklar getirmesi için geliştirmeye devam etmekteyiz. Bu gün bir bilgisayarı tek elimizde taşıyabiliyoruz bundan 40 yıl önce bilgisayar bir oda büyüklüğünde olmasına rağmen günümüze oranla yüzlerce kez daha yavaştı. İşlemciler mevcut teknoloji ile artık nano ölçekte üretilmektedir [6]. Günümüzde modern teknoloji ile üretim yapılmasına rağmen eski bir üretim tekniği olan saf malzemeden işlenmiş ürün elde etme yöntemini kullanmaya devam etmekteyiz. Bu teknik yukarıdan aşağıya (top-down) üretim tekniği olarak adlandırılmaktadır [5]. Alışlagelmiş yöntemlerle küçültülerek işlemciler içine yerleştirilen transistörler için belli bir süre sonra limitlere ulaşılacak ve yukarıdan aşağıya üretim tekniği ile üretim pahalı ve zor hale gelecektir [6]. Feynman ve Drexler bu ihtiyaçları göz önünde bulundurarak yeni bir teknoloji terimi ortaya attılar. Bu teknolojiye aşağıdan yukarı üretim tekniği (bottom-up) adını verdiler. Ürünü meydana getirirken atomların temelden itibaren yerleştirilerek yapılması gerektiğini belirttiler. Bu öneriden sonra farklı disiplinlerden birçok araştırmacı bu üretim tekniği üzerinde çalışmalar yapmıştır. Günümüzde cihazlar yapılırken üretim teknolojisi olarak yukarıdan aşağıya üretim tekniği kullanılıyor olsa da aşağıdan yukarı üretim tekniği hızla gelişmeye devam etmektedir [6].

Aşağıdan yukarı üretim tekniği ortaya atıldıktan sonra bu teknikle alakalı bazı sorunlar ortaya çıkmıştır. Bunlardan bir tanesi fizik kurallarının nano ölçeklerde uygulanabilirliğinin zor olmasıdır [7]. Nano ölçekte maddenin özellikleri

karmaşık klasik fizik kuralları ve kuantum mekaniği tarafından yönetilmektedir bu da üretime bazı noktalarda engel teşkil etmektedir [7]. Her şeye rağmen aşağıdan yukarı üretim tekniği ile nano tüpler ve kuantum noktaları başarı ile üretilmiştir [6]. Söz konusu yöntem şimdilik karmaşık elektronik devreleri veya işlemcileri üretmek için uygun olmasa da çalışmalar devam etmektedir.

Son zamanlarda bilgisayar bilimleri ile nanoteknoloji yakın ilişki içerisindedir. Bilgisayar bilimleri denilince akla gelen alanlar; yazılım mühendisliği, network, internet güvenliği, görüntü işleme, sanal gerçeklik, yapay zekâ ve yapay zekâ sistemlerdir. Bilgisayar bilimlerinin nanoteknoloji ile ilk etkileşimi araştırma araçları etrafında odaklanmıştır. Araştırmacılar tarama mikroskopları ile çalışırken bilgisayar grafikleri ve görüntü işleme tekniklerini kullanarak interaktif sanal bir ortam oluşturmuşlar ve bu ortamdan atomları keşfetme, işleme, düzenleme gibi aşamalarda faydalanmışlardır [8, 9].

Bilgisayar bilimleri nanoteknoloji için birçok fırsat vadetmektedir. Yumuşak hesaplama teknikleri, sürü zekâsı, genetik algoritma, hücresel otomata gibi teknikler birçok gerçek problemin çözümü için kullanılmış ve başarılı sonuçlar elde edilmiştir. Bu tarz tekniklerin nanoteknoloji için modellenmesiyle trilyonlarca nanoişleyici ve nano robotun senkronizasyonu yapılabilir, bunun sonucunda daha verimli üretim gerçekleştirilebilir [10].

1.1. Nanoteknoloji Alanındaki İlerlemeler

Feynman'ın ön görüşleri farklı disiplinlerde çalışan araştırmacılara ilham kaynağı olmuştur. Drexler "Engine of Creation" adlı kitabında Nanoteknoloji ve Moleküler Mühendislik terimlerinden bahsetmiştir [5]. Drexler, Feynman'ın ön görüşlerini araştırmış ve bunları birçok açıdan incelemiştir, potansiyel faydaları ve insanlık için ortaya çıkabilecek tehlikeleri üzerinde durmuştur. Ürünleri atomik düzeyde aşağıdan yukarı üretim tekniği ile inşa etmenin yaşam kalitemizi artırırken çevreye vermiş olduğumuz zararı düşüreceğini iddia etmiştir. Çeşitli örnekler verecek olursak; ürünler küçüldüğünde paketlemek için kullanılacak kâğıt da küçülecek hem kirlilik azalacak hem de kâğıt israfı engellenmiş olacaktır,

bir grup programlanabilir nano robot damardan enjekte edilerek vücut içinde zarar görmüş organları tamir etmek, ameliyat yapmak için kullanılabilir.

Nanoteknoloji insanlığın geleceği için büyük önem taşımaktadır. Faydalı olacağı alanların yanında eğer dikkat edilmese insanlığa zararı dokunabilecek bir teknolojidir. Drexler kontrolsüz bir şekilde çoğalan nano yapıcılarının dünya yaşamı için büyük tehlike oluşturacağını öne sürmüştür [5]. Toplumunu tehlikeden korumak için nanoteknoloji alanında yapılan çalışmalara etik ve sosyal değerler göz önünde bulundurularak kabul edilebilir standartlar getirilmesi çok önemlidir [10].

Günümüzde nanoteknoloji toplumun ilgisini fazlasıyla çekmiştir. Birçok şirket ticari amaçlar için nanoteknoloji üzerine araştırmalar yapmaktadır. Ülkeler bu alanda araştırma yapan firmaları desteklemenin yanında bu alanda devlet bünyesinde çalışan kurumlar kurarak bu teknolojiyi devlet politikası haline getirmektedirler.

1.2. Nanoişleyiciler

Aşağıdan yukarı üretim tekniği nanoteknoloji için büyük önem arz etmektedir. Aşağıdan yukarı üretim teknolojisinde ürün içine atomlar birer birer yerleştirilerek üretim yapılmaktadır. Bu üretim tekniğinde ilk adım ürün için gerekli atomların nano ölçekte bir araya getirilmesi işlemidir. Bu yüzden nanoişleyiciler nano materyalleri işleme sürecinde çok kritik görev yapan araçlardır [10].

Nano ölçekte görüntüleme yapabilen ilk elektron mikroskobu M.Knoll ve E.Ruska tarafından 1931 yılında geliştirildi [11]. 1981 de G. Binnig ve H. Rohrer tarama tünel mikroskopunu(Scanning tunnelling microscope STM) icat ettiler. STM ile atomları görüntülemeyi başardılar. Bu icatları onlara Nobel ödülü kazandırdı [11].

STM'nin başarısı farklı türde mikroskopların icadına öncülük etmiştir. Bunlar; tarama mikroskobu (Scanning probe microscopes, SPM), atomik kuvvet mikroskobu (Atomic force microscope, AFM)'dir. Her birinin çalışma prensibi ve

görüntüledikleri maddeler farklılık göstermektedir. Örneğin STM ile iletken maddeler görüntülenebilirken AFM ile iletken olmayan maddeler görüntülenebilmektedir [10].

1993 yılında W. Robinett ve R. S. Williams STM’i kullanarak atomları görmeyi ve dokunmayı sağlayan sanal gerçeklik sistemi geliştirdiler [11, 12]. Bu buluş nanoişleyicilerin başlangıcı olarak kabul edilmektedir. Kuzey Carolina Üniversitede bilgisayar bilimleri, fizik ve kimyacıardan oluşan bir araştırma grubu bir nanoişleyici geliştirmiş ve atomları sanal olarak bilgisayar grafikleri yardımı ile görüntülemeyi başarmışlardır. Bu sistem bilim insanlarına atomları nano düzeyde işleme ve izleme imkânı sunmuştur. Örneğin; bilim insanları bu sistem sayesinde karbon nano tüplerinin elektriksel ve mekaniksel özelliklerini incelemişlerdir [9].

Nanoişleyiciler ticari olarak satılmaktadır ancak nano ölçekte fizik kuralları tam olarak kavranmadıkça güvenli nano makineler yapılması pek mümkün gözükmemektedir.

1.3. Nanoteknoloji ve Bilgisayar Bilimleri

National Nanotechnology Initiative (NNI, Ulusal Nanoteknoloji Girişimi) 2001 yılında Amerika’da resmi olarak kurulan bir kuruluştur. Bu kuruluşun amacı nano ölçekli bilimler ve teknolojiler arasında koordinasyonu sağlamaktır. NNI nanoteknolojinin ticarileşmesi için bir zaman çizelgesi oluşturmuştur [13]. Bu çizelgede nanoteknolojinin gelişim sürecini 4 aşamaya bölmüştür. İlk aşama olan pasif nano yapıların oluşturulması 2004 yılında sona ermiştir. Bu aşamanın örnekleri; kaplama, nano partiküller, nano yapıda metaller, polimerler, seramiklerdir. 2004 ten 2010’a kadar devam eden ikinci aşamada aktif nano yapılar oluşturulmuştur. Bunlar içerisinde transistörler, yükselticiler, bazı ilaçlar, aktivatörler, ve adaptif yapılar yer almaktadır. 2010 yılından sonra üçüncü aşamaya geçilmiştir. Bu aşamada tahmini olarak yapılacak olan nano sistemlerdir. Kılavuzla molekül birleştiren sistemler, 3D network sistemleri, nano sistemler için yeni bir mimari, robotlar, süper moleküler cihazlar, gibi yenilikler

beklenmektedir. 2020 yılından itibaren başlayacak olan dördüncü aşamanın moleküler nano sistemleri beraberinde getirmesi öngörülmektedir. Bu sayede parçalar ya da cihazlar atomik düzeyde tasarlanırken evrimsel sistemler ile birleştirilebilecektir.

Günümüzde nanoteknoloji çoğunluklu olarak fizik, kimya, materyal bilimleri ve biyoloji bilimleri tarafından geliştirilmektedir. Şimdiye kadar bilgisayar bilimleri bu gelişim sürecinde araştırma araçları olarak kullanılmıştır. Örneğin nano işleyici projesinde sanal gerçeklik sistemi ile mikroskopun birleştirilmesi gibi yan rollerde bulunmuştur. NNI'ya göre dördüncü ve üçüncü aşama araştırmalarında ağırlıklı olarak bilgisayar bilimleri rol alacaktır.

Bilgisayar bilimleri çok geniş bir alanı kapsamaktadır. Bu yüzden birçok açıdan nanoteknoloji ile ilişkilendirilmesi mümkündür. Bilgisayar bilimleri nanoteknoloji için geliştirilen yazılımlar, simülasyon programları ve yapay zeka algoritmaları nano yapıları oluşturmak için kullanılmaktadır.

1.4. Nanoteknolojinin Kullanım Alanları

Nanoteknoloji Şekil 1.2'de görüldüğü gibi fizik, kimya biyoloji, materyal bilimi, mühendislik, elektronik tasarım, tarım, enerji, gibi bir çok bilim alanında yapılan çalışmalar ve bu alanda geliştirilen ürünler ile karşımıza çıkmaktadır.



Şekil 1.2. Nanoteknolojinin etkileşimli olduğu alanlar

1.5. Nanoteknoloji Ürünleri ve Uygulamaları

2005 yılında global olarak 32 milyon dolardan fazla nano materyal içeren ürün satılmıştır. 2014 yılında 2,6 trilyon dolar hacminde nano ürün satılmıştır. Ekonomi uzmanları 2015'in sonlarına doğru bu hacmin 1 trilyon dolar daha artacağını öngörmektedirler [1]. Hali hazırda 200 den fazla şirket pazarda nanoteknoloji ürünlerini satmaktadır. Bu şirketler 700 den fazla nano ürünü Amerika pazarında sunmaktadır [1]. Günümüzde bulunan bazı nano ürünlere kısaca bakacak olursak.

1.5.1. Spor Ürünleri

Karbon nano partiküllerinden yapılmış raket başlığı ve sapı çelikten yapılmış olan muadillerine göre 100 kat daha sert ve 6 kat daha hafiftir. Karbon nanotüp epoksi matris sayesinde üretilen hokey çubukları diğer çubuklardan daha dayanıklıdır.

Nano tenis topları içten yapılan nano düzey zar sayesinde basıncı daha kolay tahliye etmektedir.

1.5.2. Araç Boyaları

Günümüzde nanoteknoloji kullanılarak geliştirilen yeni nesil araç boyaları geleneksel araç boya koruyucularına göre çizilmeye karşı daha dayanıklıdır. Nanoteknoloji ile üretilen araç boya koruyucuları çok ince lekeleri ve çizikleri doldurduğu için muadillere göre boyayı daha parlak göstermektedir.

1.5.3. Anti Bakteriyel Temizleyiciler

Piyasada nano emülsiyon teknolojisini kullanarak üretilen birçok anti bakteriyel temizleyici bulunmaktadır. Bu temizleyiciler patojenlerin öldürülmesinde çok etkilidir. Temizleyiciler tüberküloz ve bakterileri de başarılı bir şekilde yok etmektedir. Kullanım esnasında ya da kullanım sonrasında herhangi bir zararlı etkisi yoktur.

1.5.4. Medikal Bandajlar

Gümüş metali antibiyotik özellikleri sayesinde yaraların ve yanıkların tedavisinde kullanılmaktadır. Yanıklar için üretilen özel elbiseler nano gümüş parçacıklarından oluşan bir anti bakteriyel koruma bariyeri oluşturmaktadır. Bu medikal bandajlar derinin tedavisi sırasında enfeksiyonlardan korunmasını sağlamaktadır. Ayrıca diğer bandajlar gibi yaraya yapışmadığı için değişim sırasında daha az acı vermektedir.

1.5.5. Giyim Sektörü

Birçok giyim markası nanoteknoloji ile üretilmiş kumaşlarını kullanıcılarına sunmaktadır. Kumaş kendini çeşitli sıvı türlerine ve kirlenmelere karşı korumakta (içecekler, salata sosları), serin ve rahat hissettirmektedir. Üzerinde toz tüy gibi çevresel etkenlerin bulaşmasını engellemektedir.

1.5.6. Güneş Kremleri ve Kozmetik Ürünleri

Kozmetik firmaları nanoteknoloji ile ürettikleri güneş kremleri, deodorantlar, kırışıklık önleyici kremler gibi ürünlerini kullanıcıya sunmaktadır. İçerisinde çinko oksit nano parçacıklar bulunduran bir güneş kremi deriyi korurken deri üzerinde beyaz lekeler bırakmamaktadır.

1.6. Nanoteknoloji için Gelecekte Önem Verilen Uygulama Alanları

Yukarıda bahsedilen ve hali hazırda inşaların kullanımına sürülen ürünlerin yanı sıra halen geliştirilmekte olan ve bunun için yatırım yapılan alanlar aşağıda listelenmektedir.

- Çevre
- Güneş enerjisi
- Yakıt hücreleri
- Yiyecek ve Tarım
- Otomobil ve Havacılık
- Sağlık uygulamaları

Başta Amerika olmak üzere Rusya, Japonya, Brezilya, İsrail, Çin, Tayvan, Meksika, Güney Afrika ve birçok Avrupa ülkesi nanoteknolojiyi desteklemekte ve bu alanda gelecek için çalışmalarını sürdürmektedir.

2. MATERYAL VE YÖNTEM

Tezin bu bölümünde çeşitli optimizasyon algoritmalarına kısaca değinilecek, genetik algoritma hakkında detaylı bilgi verilecektir. Genetik algoritmanın işleyişi problemlere uygulanma yöntemi olumlu ve olumsuz yönleri üzerinde durulacaktır.

2.1. Temel Kavramlar ve Genel Tanımlar

Optimizasyon, çözümlerin bulunduğu bir küme içerisinde en iyi çözümün seçilmesi işlemidir [14]. Fiziksel bir problemi matematiksel olarak modelleyerek en iyi çözümleri bulan bilim olarak da tanımlanmaktadır [15].

Optimizasyonun matematiksel tanımı ise, bir amaç fonksiyonunun, verilen kısıtları göz önünde bulundurarak, söz konusu problemin en küçük ya da en büyük çözüm değerini tespit etmesi işlemidir [14].

Günlük yaşamda gelirler veya kar değerlerin en fazla olması istenirken, giderler, maliyet gibi değerlerinde en az olması beklenmektedir. Optimizasyon teknikleri bu dengeyi sağlamak için bulunması gereken en uygun değeri hesaplarırken çeşitli metotlar, matematiksel işlemler ve algoritmalar kullanılmaktadır [15].

Optimizasyon problemleriyle mühendislik, fizik, matematik, sosyal bilimler, ekonomi gibi birçok alanda karşılaşmaktadır. Mühendislikte bilgisayar, makine, elektronik, inşaat gibi alanlarda yapıların dizayn edilmesi, üretim süreci, üretim planlama, kalite kontrol gibi birçok alanda optimizasyon uygulamaları yapılmaktadır [15].

Karşılaşılan bir problemin birden çok çözümü olabilmektedir. Şayet problemin tek bir çözümü mevcut ise optimizasyon uygulanması mümkün olmamaktadır [15]. [16]

Optimizasyon problemlerinin genel kabul gören sınıflandırması aşağıdaki gibidir;

- Amaç fonksiyonunda herhangi bir sınırlama olmaksızın yapılan optimizasyon türüne; sınırlamasız optimizasyon,
- Amaç fonksiyonunda sınırlamaların olduğu optimizasyon problemi; sınırlamalı optimizasyon
- Amaç fonksiyonu ve parametrelerinin doğrusal olup olmamasına göre; doğrusal optimizasyon
- Amaç fonksiyon parametreleri doğrusal değil ise; doğrusal olmayan optimizasyon
- Ayrık değerlerin en uygun olarak düzenlenmesi, gruplanması; ayrık optimizasyon
- Parametrelerin alacağı değerler sürekli ise bu tür optimizasyon problemlerine; sürekli optimizasyon problemi denilmektedir [16].

Araştırmacılar optimizasyon problemlerinin çözümünde yapay zeka teknikleri kullanmışlardır. Bu teknikler problemin çözümüne ulaşmak için gerçekleştirilmesi gereken ölçütler veya bilgisayar metotları bütünüdür [16].

Sonuçlar net olarak gösteriyor ki yapay zekâ optimizasyon algoritmaları optimizasyonun uygulanacağı probleme bağımlı algoritmalarıdır. Bir problem için iyi sonuç veren bir yapay zekâ optimizasyon algoritması başka bir problem için aynı başarıyı göstermeyebilir. Yapay zekâ optimizasyon algoritmalarından bazıları; tabu, tavlama benzetimi, karınca kolonisi ve genetik algoritmadır [16].

Tabu Algoritması: Algoritma Glover tarafından 1986 yılında önerilmiştir [17]. Temel çalışma mekanizması, son çözüme götüren adımın dairesel hareketler yapmasının önüne geçmek için bir sonraki döngüde tekrarın yasaklanması veya cezalandırılması prensibine dayanarak işlemektedir. Tabu Algoritması var olan çözüm uzayını tarayarak en iyi çözümü bulur ve bu çözümü yeni çözüm olarak sunar. Şayet sunduğu çözüm mevcut çözümden daha kötü olsa bile komşulardan yeni çözümler bulmaya devam eder. Bu sayede Tabu algoritması mevcut çözüm

uzayını büyük ölçüde tarayarak yerel minimum noktalarından kaçınılmasını sağlar [18].

Tavlama Benzetimi: 1983 yılında Kirkpatrick tarafından önerilmiş bir algoritmadır [19]. Katı hal fiziğinin temel disiplini ve katı veya sıvı haldeki maddelerin atom özelliklerinin analizinde kullanılan metot olan istatistiksel ve mekanik temellere dayanan rastlantısal bir tekniktir. Katıların tavlama işlemine benzediği için bu isimle anılmaktadır [18]. Algoritmanın en önemli dezavantajı, süreç içinde giderek azalan bir olasılıkla daha kötü bir çözümü bulmaya sebep olabilesidir [16].

Karınca Kolonisi Algoritması: 1992 yılında Dorigo tarafından önerilmiştir [20]. Algoritmanın yapısı karıncaların yiyeceğe giden en kısa yolu bulmak için kullandıkları yöntemin matematiksel olarak modellenmesi ile oluşturulmuştur [18].

Genetik Algoritma: Doğal seçilim ilkelerinden destek alan bir arama ve optimizasyon algoritmasıdır. Temelleri John Holland tarafından ortaya atılmıştır [21]. Genetik algoritmanın optimizasyon, makine öğrenmesi, tasarım, planlama gibi daha birçok alanda başarılı uygulamaları bulunmaktadır [22].

2.2. Genetik Algoritmalar (GA)

İnsanlık tarihi incelendiğinde birçok önemli buluş doğanın taklidi yoluyla ortaya çıkmıştır. Doğa insanlara sonsuz bir esin kaynağı olmuş ve olmaya devam etmektedir. Michigan Üniversitesinde makine öğrenmesi üzerine çalışmalar yapan John Holland evrim kuramından etkilenerek canlılarda meydana gelen genetik süreci bilgisayar ortamında gerçekleştirmeyi başarmıştır. Holland 1975 yılında çalışmalarını bir kitap haline getirmiş ve bulduğu yöntemin adını genetik algoritmalar koymuştur [21].

O günlerde genetik algoritma araştırmasının pratik karşılığı olmadığı düşünülmekteydi. 1985 yılında Holland'ın doktora öğrencisi inşaat mühendisi David E. Goldberg gaz borusu hatlarının denetimi üzerine yaptığı çalışmada genetik algoritma kullanarak GA'nın pratik kullanımının da mümkün olduğunu kanıtlamış oldu [23].

GA rastlantısal arama yöntemlerini kullanarak çözüm arayan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir. GA'nın çalışma prensibi Darwin'in en iyi olan yaşar prensibine dayalı olarak popülasyonu oluşturan bireylerin birbirleri ile rekabet etmelerini ve sonuç olarak elenmelerini sağlayan evrimsel süreci matematiksel olarak modelleyen algoritmalarıdır [16]. Şekil 2.1'de çalışma adımları gösterilmiştir. Genetik algoritma bazı varsayımlar üzerine işlemektedir.

- Bireyler yaşamını devam ettirebilmek için sürekli mücadele halindedir. Bu mücadele sonucu bir kısmı kaybolurken bir kısmı sonraki nesilde varlıklarını devam ettirebilmektedir.
- Varlığını devam ettiren bireylerin daha sonraki nesillere aktarım ihtimali daha yüksektir.
- İlk olarak bir başlangıç popülasyonu oluşturularak bu popülasyonun sonraki nesiller için bir temel oluşturması sağlanır [24].



Şekil 2.1. Genetik algoritma akış şeması

Genetik biliminde olan bazı kavramlar genetik algoritmaya göre düzenlenerek kullanılmaktadır.

Gen: Genetik bilgi taşıyan anlamlı en küçük yapı birimidir. Genetik algoritmada bit olarak tanımlanmaktadır.

Kromozom(Birey): Problemin olası çözümlerinden biri ifade eder ve genlerden oluşmaktadır.

Popülasyon(Nesil): Problemin çözümlerini ifade eden kromozomlardan oluşan bir havuzdur. Algoritma başlamadan önce popülasyon içindeki birey sayısı belirlenir ve süreç boyunca sabit kalır.

Seçim: Genetik algoritma iteratif olarak çalışır. Belli sayıda işlemde geçen popülasyonun içindeki hangi bireylerin bir sonraki iterasyona aktarılacağını belirler.

Çaprazlama: Popülasyon içinden seçilen iki kromozomunun genlerinin yer değiştirmesidir. Bu olay sonucunda iki yeni farklı birey oluşmaktadır.

Mutasyon: Bir veya birden fazla bireyin rastgele seçilen bir geninde yapılan değişikliktir. Mutasyon tür içi çeşitliliği sağlar.

Uygunluk Fonksiyonu: Problemin amaç fonksiyonudur. Bireylerin problemin çözümüne uygunluğu hesaplanarak bir sonraki nesle aktarılıp aktarılmayacağı uygunluk fonksiyonuna göre belirlenmektedir [24]

2.3. Genetik Algoritmanın Çalışma Mekanizması

Bir genetik algoritmanın içermesi gereken aşamalar Şekil 2.1’de verilmiştir. Genetik algoritmanın bir probleme uygulanma süreci aşağıdaki gibidir [16, 24].

- Problemin uygunluk fonksiyonu ve başlangıç parametreleri belirlenir. Başlangıç parametrelerinden kasıt popülasyon sayısı, mutasyon oranı, durma ölçütü gibi değerlerdir.
- Problemin olası çözümlerini temsil eden kromozomların ifade biçimleri oluşturulur.
- Rastgele ilk popülasyon oluşturulur.
- İlk popülasyondaki her bir birey için uygunluk değeri hesaplanır.

- En yüksek uygunluk değerine sahip olan bireyler üreme havuzu için seçilir.
- Üreme havuzundan seçilen bireyler önce çaprazlama sonra mutasyon işlemine tabi tutulur.
- Çaprazlama ve mutasyon sonucu oluşturulan bireylerin uygunluk değerleri hesaplanır.
- Yeni bireyler popülasyona eklenir.
- Durma ölçütü sağlandı ise durulur, şayet sağlanmadı ise bir sonraki iterasyon için 5. maddeden devam edilir.
- Son popülasyonda en iyi uygunluk değerine sahip birey problemin en iyi çözümüdür [16, 24].

2.3.1. Uygunluk Fonksiyonu

Bireylerin problemin çözümüne uygunluğu hesaplayan fonksiyona denir. Bir örnek üzerinden açıklayacak olursak $f(x)=x^2$ 'nin $[0,9]$ aralığında alacağı maksimum değeri bulmak istiyoruz. Burada $f(x)=x^2$ bizim uygunluk fonksiyonumuz 0-9 arası gelecek rastgele sayılar ise kromozom değerlerimizdir. İlk popülasyon için rastgele gelen bireylerimiz 2,6,7 olsun bu değerlerin uygunlukları hesaplanacak olursa 4,36,49 olacaktır. Bu durumda en iyi birey 7 olmaktadır. Bireylerin uygunluk değerleri problemde istenen maksimum ya da minimum değerine göre hesaba alınmaktadır.

2.3.2. Genetik Algoritma Parametreleri

Algoritmanın işlemeye başlamadan önce tanımlanması gereken parametrelerdir.

Mutasyon Olasılığı: Popülasyondaki herhangi bir bireyin mutasyon için seçilme olasılığına denilmektedir. Ayrıca bir popülasyonda kaç bireyin popülasyona uğratılacağı da başlangıçta verilmesi gereken parametrelerdendir.

Popülasyon Büyüklüğü: Popülasyonun kaç bireyden oluşacağını belirleyen parametredir. Problemin yapısına göre popülasyon büyüklüğü değişkenlik gösterebilir.

Durma Ölçütü: Genetik algoritmada her zaman en mükemmel sonuca ulaşma mümkün olmayabilmektedir. Bunun için algoritma adımlarının belli bir iterasyon sayısında ulaşması veya çözüm için belirli bir değere yaklaşması durma ölçütü olarak kullanılabilir.

2.3.3. Kromozom Tasarımı

Genetik algoritma ile problem çözülürken her bir bireyi ifade eden bir dizi kodlaması yapılmaktadır. Kromozom kodlaması problemin hızlı çözülmesi ve algoritmanın en doğru çözüme ulaşması için en önemli adımlardan biridir. İkili kodlama, tam sayılı kodlama, ondalık sayılarla kodlama, harflerle kodlama gibi çeşitli kodlama yöntemleri vardır. Kromozomun kodlanması problemin yapısına uygun olmalıdır ve olası her çözüme karşılık verebilecek nitelikte tasarlanmalıdır.

2.3.4. Başlangıç Popülasyonu

Genetik algoritma bir tek çözüm yerine çözümler kümesi ile çalışmaktadır. Başlangıç popülasyonu genelde rassal olarak oluşturulur.

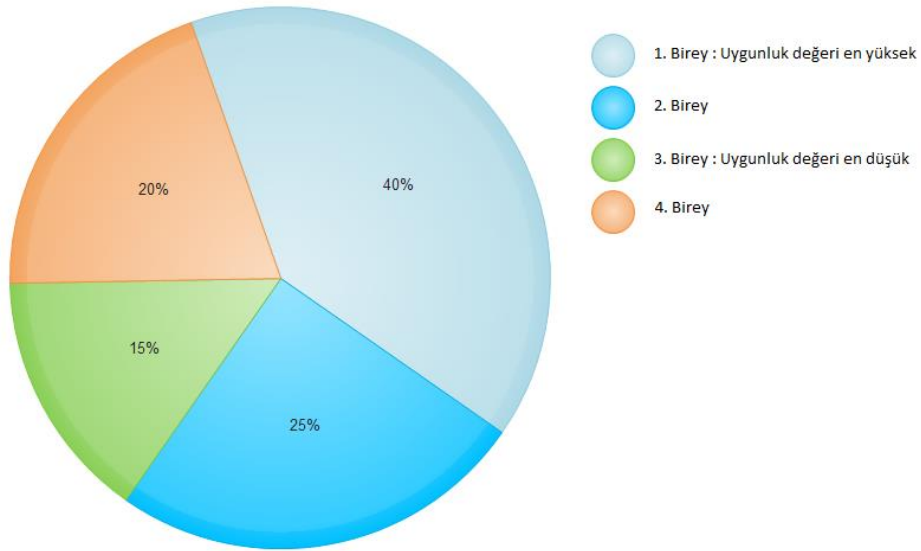
2.3.5. Genetik Operatörler

İteratif olarak çalışan genetik algoritmada mevcut nesilden bir sonraki nesil oluşturulurken gerçekleştirilmesi gereken aşamalar seçim, çaprazlama ve mutasyondur.

Seçim: Mevcut popülasyondan hangi bireylerin bir sonraki popülasyona aktarılacağını veya hangi bireylerin çaprazlamaya, mutasyona uğratılacağını belirleyen operatördür. Seçim işlemini bireyin uygunluk değerine göre

yapmaktadır. En çok kullanılan seçim yöntemi rulet tekeri yöntemidir. Bunun yanında rassal üreme, turnuva yöntemi, elitizm gibi çeşitli seçim yöntemleri vardır.

Rulet Tekerı Yöntemi, temel olarak uygunluk değeri yüksek olan bireyin seçilme olasılığının yüksek olmasını temel alır. Popülasyonda yer alan bireyler bir rulet tekeri üzerine yerleştirilir. Bu yerleştirmede uygunluk değeri yüksek olan bireyin tekerde kapladığı yüzdelik dilim daha fazla iken uygunluk değeri az olan bireyin yüzdelik dilimi daha azdır. Bu yöntemle uygunluk değeri yüksek olan bireylerin sonraki nesle aktarılma şansı daha yüksek olmaktadır. Şekil 2.2’de bireylerin uygunluk değerlerine göre tekerleğe yerleşme yüzdeleri gösterilmiştir

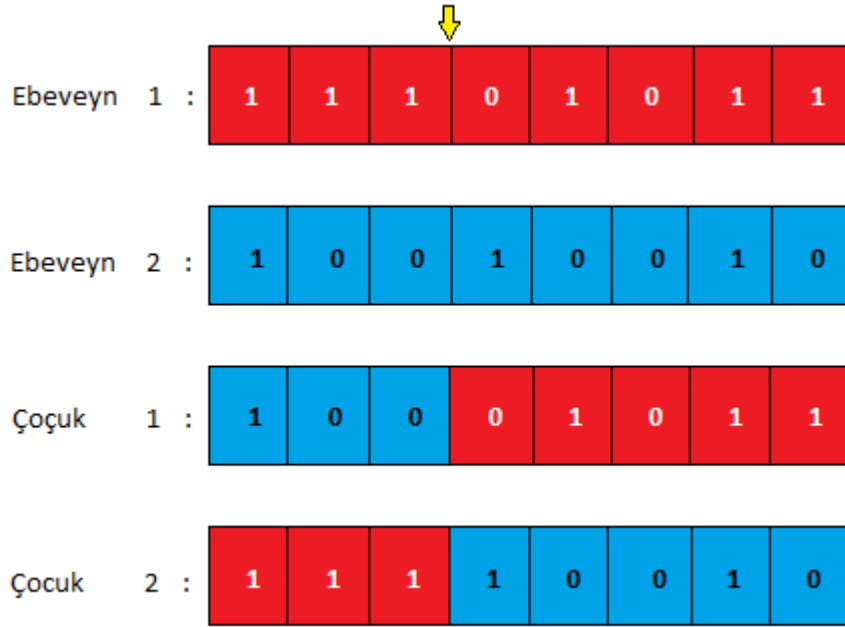


Şekil 2.2. Rulet tekeri

Turnuva Yöntemi, popülasyon içinden rastgele seçilen bireyler arasında turnuva yapılır. Bireyler ikili olarak karşılaştırılır. Uygunluk değeri daha iyi olan birey turnuvayı kazanarak seçilmiş olur.

Üreme Havuzu: Seçim operatörleri vasıtasıyla seçilen bireyler bir havuza alınır. Bu havuza üreme havuzu denilmektedir. Çaprazlama için alınacak bireyler bu havuzdan alınarak çaprazlama işlemine tabi tutulurlar.

Çaprazlama: Üreme havuzundan seçilen bireyler rastgele olarak eşleştirilir ve çaprazlama işlemine tabi tutulur. Seçilen bireylerin çaprazlanması sayesinde mevcut popülasyonda olmayan yeni yavru bireyler oluşturulur. Oluşturulan yavru bireyler atalarının genlerini taşımaktadır. Bu sayede çeşitlilik artmaktadır. Çaprazlamanın en basit hali kromozomlar üzerinde rastgele bir nokta seçilerek genler arasında yapılan karşılıklı değişimdir. Şekil 2.3’de örnek yöntem gösterilmiştir. Literatür incelendiğinde problem tiplerine göre birçok çaprazlama tekniği mevcuttur. Gerçek sayılı çaprazlama, permütasyon çaprazlama, sıralı çaprazlama bunlardan bazılarıdır.



Şekil 2.3. Çaprazlama yöntemi

Mutasyon: Popülasyonda çeşitliliği sağlayan diğer bir yöntemdir. Bu yöntemde ikinci bir bireye ihtiyaç duyulmamaktadır. Rastgele seçilen bir bireyin rastgele seçilen bir geninin değiştirilmesi işlemidir. Şekil 2.4’de bit düzeyinde mutasyon yöntemi gösterilmiştir.

Ebeveyn :	1	0	1	1	0	0	1	0
Çocuk :	1	0	1	0	0	0	1	0

Şekil 2.4. Mutasyon yöntemi

2.4. Genetik Algoritma Avantaj ve Dezavantajları

Genetik algoritma probleme çözüm ararken diğer algoritmalarından ayrıldığı bazı önemli noktalar vardır. Genetik Algoritma birden fazla çözüm ile çalışır ve çözüm kümesinde iyi olanların yanında kötü çözümlerin de olma ihtimali vardır. Diğer algoritmalar belli bir başlangıç noktası seçip çözümü adım adım daha iyiye taşıırken genetik algoritmada bunun garantisi yoktur.

Diğer algoritmaların yerel minimum ya da maksimum noktalarında takılma olasılığı varken genetik algoritmalarda bu durum çaprazlama ve mutasyon ile ortadan kalkmaktadır. Bu yolla doğruya en yakın sonuca ulaşılır. Hesaplama adımları fazla olduğu için zaman konusunda pek verimli değildir.

2.5. Nano Topak Yapılarda Optimizasyon

Atomik topaklar günümüzde deneysel ve teorik olarak çalışılan aktif konuların başında gelmektedir. Teorik olarak topakların yapılarını tahmin etmek çok zordur çünkü atomlar arası yerel minimumlar üstel olarak artmaktadır.

Nano topak yapıları nano malzemelerin temel taşları olduklarından geometrilerinin ve kararlı yapılarının bulunması oldukça önemlidir. Özellikle aşağıdan yukarı üretim tekniği için nano yapılar büyük önem arz etmektedir. Kararlı yapıları elde edilen nano yapılar yeni malzemelerin geliştirilmesinde temel

yapı taşları olarak kullanılabilirler. Atomik topakların minimum enerjisini bulmak, atomik topaklar için maksimum kararlılık demektir. Buda nano yapıda bir malzeme üretilirken göz önünde bulundurulması gereken en önemli etmenlerden biridir. Bu bir global minimum bulma problemidir.

N tane atomdan oluşan bir topakta uygun yerel minimum enerji seviyesini bulmak için bu güne kadar birçok metot kullanılmıştır. Bunlardan bazıları şunlardır. Newton-Raphson, eşlenik gradyan, daha gelişmiş metotlar monte carlo, benzetim tavlama ve genetik algoritma olarak sayılabilir.

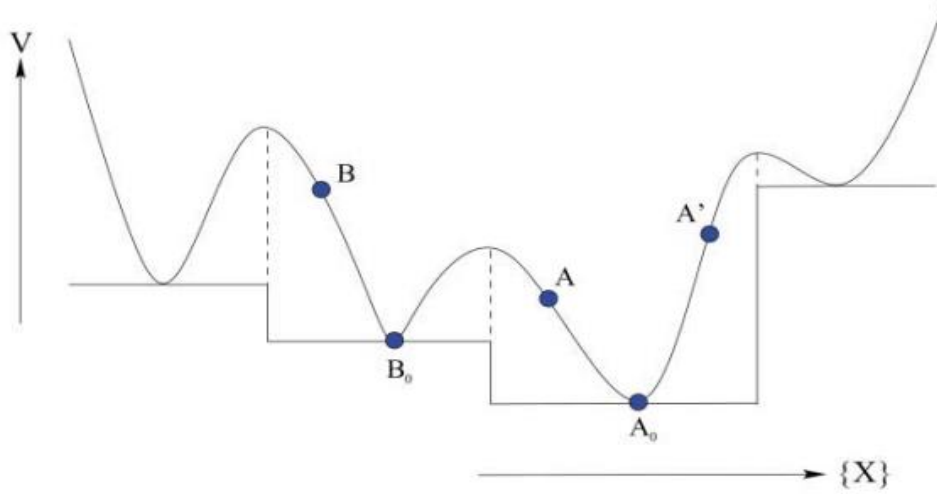
Genetik algoritma evrimsel süreçleri temel alarak çalışan bir genel optimizasyon algoritmasıdır. Bünyesinde çaprazlama, mutasyon, elitizm gibi biyolojik süreçleri bulundurur. Topak geometri optimizasyonunda özellikle genetik algoritmanın verimliliği kanıtlanmış ve bu alanda gelişmeler devam etmektedir.

2.6. Genetik Algoritmanın Nano Topak Yapılarda Kullanımı

GA'nın nano topakların geometri optimizasyonunda kullanılması 1990'ların başlarında ortaya çıkmıştır [25]. Hartke küçük silikon nano topaklar için Xiao ve Williams ise moleküler nano topaklar için GA'yı kullanmışlardır [25, 26]. Her iki durumda da bireyler ikili olarak kodlanmış ve genetik operatörler gerekli işlemleri bit düzeyinde yapmışlardır. Hartke yaptığı çalışmadan sonra değişik nano topaklar (silikon, su, cıva) üzerinde GA ile yaptığı geometri optimizasyonu sonuçlarını yayınlamıştır [27-29].

Nano topakların optimizasyonunda kullanılan GA'da önemli bir gelişme Zeiri tarafından ortaya sürülmüştür. Zeiri kromozom oluştururken gerçek değerli kartezyen koordinatları kullanarak GA'yı nano topakların geometri optimizasyonunda kullanmıştır [30]. Bu yöntem bit düzeyinde kodlama ve geri çözümleme işlemlerini ortadan kaldırmıştır [31]. GA'nın nano topaklara uygulanmasının gelişim sürecinde ortaya çıkan diğer bir önemli gelişmede Deaven ve Ho tarafından gerçekleştirilmiştir. Deaven ve Ho yeni bir nano topak oluşturulduktan sonra oluşturulan nano topağın gradient odaklı yerel minimum enerjisini hesaplamışlardır [32]. Doye ve Wales nano topak potansiyel enerji hiper yüzeyinin yerel minimizasyonu yapılmış basamaklı bir enerji yüzeyine

dönüştürülmesi üzerine çalışmalar yapmışlardır [33]. Her bir adım, Şekil 2.5'te görüldüğü gibi nano topağın yerel minimum enerji noktasını göstermektedir.

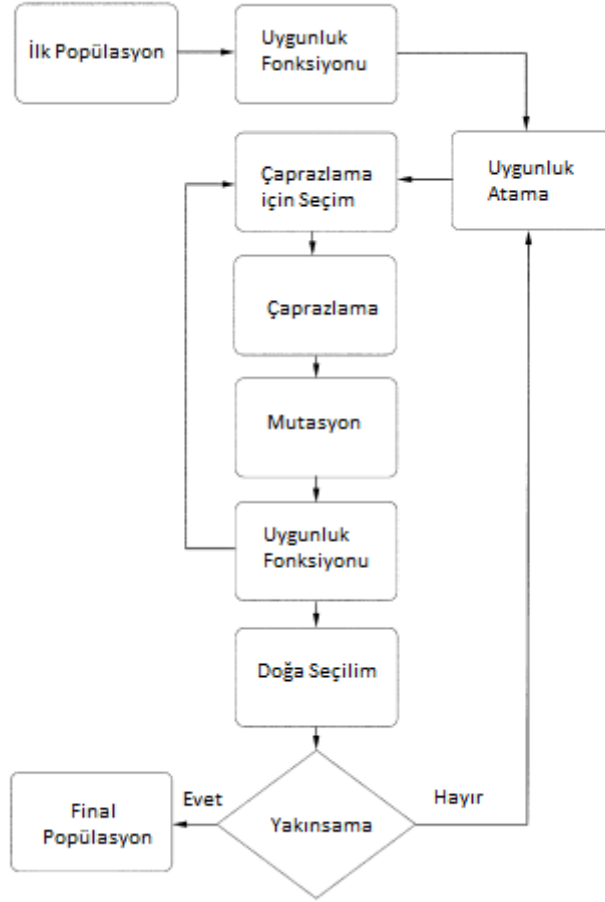


Şekil 2.5. Potansiyel enerji yüzeyi [31]

Bu basitleştirme GA'nın potansiyel enerji yüzeyinde arayacağı alanı daraltarak global minimumu daha rahat bulmasına olanak sağlamıştır. Bu alanda diğer bir önemli gelişmede yine Deaven ve Ho tarafından gerçekleştirilen kesme ve bölme yöntemiyle yapılan çaprazlama işlemidir [32]. Bu yöntemde nano topaklar belli düzlemlerden kesilerek çaprazlama yapılması ile yeni bireylerin oluşması sağlanmıştır. 1990'lardan sonra geometri optimizasyonu konusunda birçok GA yazılımı geliştirilmiştir.

2.7. Birmingham Topak Genetik Algoritması

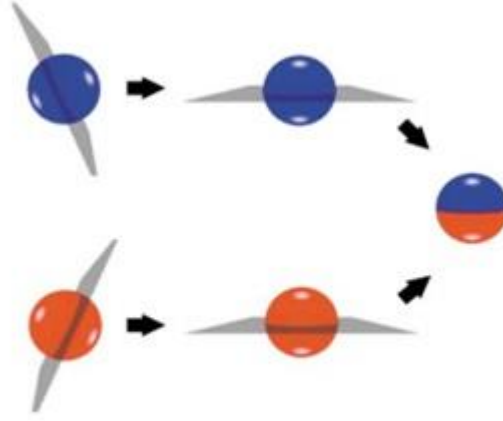
Birmingham üniversitesinde geliştirilen bu genetik algoritma kodu adını geliştirildiği üniversitenin isminden almaktadır. Algoritmanın akış şeması Şekil 2.6 'da gösterilmiştir.



Şekil 2.6. Birmingham genetik algoritması akış şeması [31]

Algoritmada ilk popülasyon rastgele oluşturulmaktadır ve popülasyon sayısı 10-30 arası değişmektedir. Kromozom oluşumunda gerçek değerli kartezyen koordinatlar kullanılmıştır. Her bir atomun x , y , z olmak üzere 3 adet koordinat değeri bulunmaktadır. N (nuclearity) olmak üzere x , y , z değerler $[0, N^{1/3}]$ aralığından rassal olarak oluşturulmuştur. Kullanılan uygunluk fonksiyonları: Exponential, Linear, Hyperbolic tanget 'dır [31].

Çaprazlama için bireylerin seçiminde rulet tekeri ya da turnuva yöntemi kullanılmıştır. Çaprazlama yöntemi ise Deaven ve Ho'nun önerdiği kes ve böl yöntemidir. Şekil 2.7'de yöntem gösterilmektedir.



Şekil 2.7. Deaven ve Ho çaprazlama yöntemi [31]

Mutasyon için birden çok yöntem önerilmektedir. Bu yöntemlerden bazıları:

Atom yeri değiştirme: Topak içindeki belli sayıda atomun koordinatlarının rassal olarak yeniden belirlenmesi işlemidir.

Döndürme: Topağın yarısının ya da belli bir kısmının rassal olarak üretilen bir açı kadar z eksenini etrafında döndürülmesi işlemidir.

Yeni Topak: Sıfırdan bir topak oluşturulup popülasyona eklenmesidir.

Atom Permütasyonu: Topak içinde bir çift veya birden fazla çift atomun topağın mevcut yapısını bozmadan değiştirilmesi işlemidir. Genellikle iyonik topaklarda kullanılmaktadır.

Bu aşamadan sonra çeşitliliği sağlamak amacı ile topakların potansiyel enerjileri arası kıyaslama yapılarak belli bir eşik değerinden az olan topaklar popülasyondan silinmektedir. Bu eşik değeri 10^{-6} a.u.'dur. Daha sonra popülasyona elitizm işlemi uygulanarak en yüksek potansiyel enerjiye sahip olan bireyler popülasyon sayısı sabit kalması da göz önünde bulundurularak popülasyondan silinmektedir. Böylelikle bir sonraki iterasyon için yeni popülasyon hazırlanmış olmaktadır. Ayrıca kötü çözümler atılarak mevcut iyi çözümlerin elde tutulması sağlanmış olmaktadır [31].

3. ARAŞTIRMA BULGULARI VE GELİŞTİRLEN YAZILIM

Bu bölümde uygulama geliştirme ortamı ve uygulama araçlarının tanıtılacak, uygulamanın çalışma sistemi ve detaylarına inilecektir.

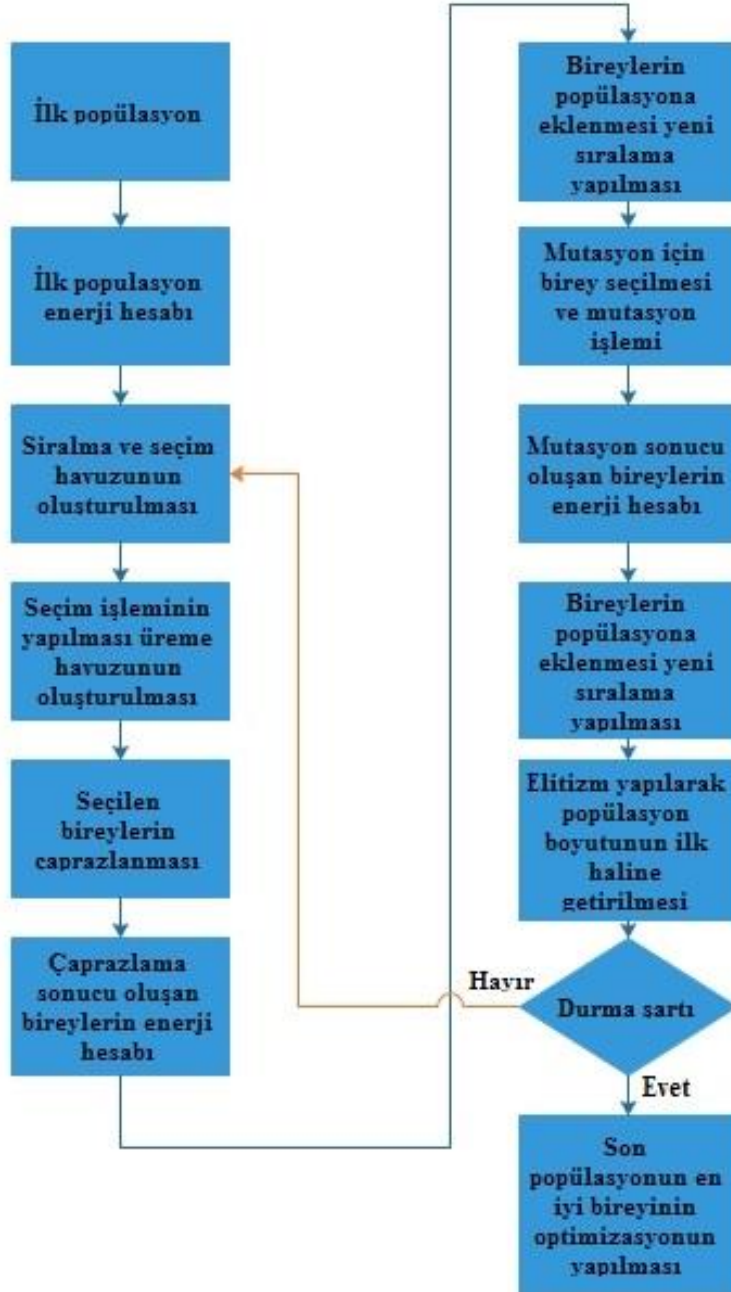
İşletim Sistemi: Uygulama geliştirmek için işletim sistemi olarak Linux'un Mint 17 dağıtımı tercih edilmiştir. Linux; unix çekirdeği temel alınarak geliştirilmiş açık kaynak kodlu özgür ve ücretsiz bir işletim sistemi çekirdeğidir. Linux'un birçok dağıtımı bulunmaktadır. Linux çekirdeği, X pencere sistemi GNU araçları ve bir masaüstü ortamının bir araya getirilerek sürdürülebilir ve tam teşekküllü bir işletim sistemi haline getirilmiş uygulamalar bütününe Linux dağıtımı denilmektedir. Piyasada bulunan en popüler Linux dağıtımları, Debian, Ubuntu, Mint, Fedora, Suse ,RetHat ve benzerleridir. Linux Mint seçilmesinin başlıca sebepleri açık kaynak olması, özgür yazılım olması, ücretsiz olması, kullanıcı dostu olması, hızlı ve kararlı çalışmasıdır.

Uygulama Yazılım Dili: Uygulama yazılım dili olarak Java seçilmiştir. Nesne yönelimli bir dil olması, yazılan kodun tüm işletim sistemleri tarafından çalıştırılabilmesi, uygulama geliştirmek için uygun araçlarının olması Java'nın seçilmesindeki temel sebeplerdir.

Kod Geliştirme Aracı: Uygulamanın kodu NetBeans aracı üzerinde yazılmıştır. Kod geliştirme sürecinde kullanıcı dostu bir ara yüzü olması, ücretsiz olması, java haricinde başka dillere de verdiği destek Netbeans'in seçilme sebepleri arasındadır.

3.1. Genetik Algoritma Tasarımı

Uygulanacak Genetik Algoritmanın iş akış diyagramı Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Programın algoritma akış şeması

3.1.1. Programın Ara Yüzünün Tasarımının Tanıtımı

Kullanıcılar programı çalıştırdıklarında Şekil 3.2’de gösterilen ana menü ekranı üzerinden algoritmayı çalıştırmadan önce bazı parametreleri girmeleri gerekmektedir. Bu parametreler algoritma için başlangıç değerlerini temsil etmektedir.



Şekil 3.2 Programın ara yüzü

İterasyon Sayısı: Algoritmanın kaç kez işletileceğini belirleyen değerdir.

Popülasyon Sayısı: Algoritma başlangıcında belirlenen ve tüm iterasyonlar boyunca sabit kalan bir değerdir. Bir popülasyon içinde kaç adet birey olacağını belirler.

Atom Sayısı: Bir nano topağın kaç adet atomdan oluşacağını belirleyen değerdir.

Atom Türü: Topağa eklenecek atom türünü belirler.

Mutasyon Oranı: Popülasyonda çaprazlamadan sonra kaç bireyin mutasyona uğratılacağını belirler.

Atomlar Arası Mesafe: Nano topak içindeki atomların birbirlerine olan uzaklık ve yakınlık mesafelerini belirler.

Çaprazlama Türü: Hangi tür çaprazlama işleminin yapılacağını belirler.

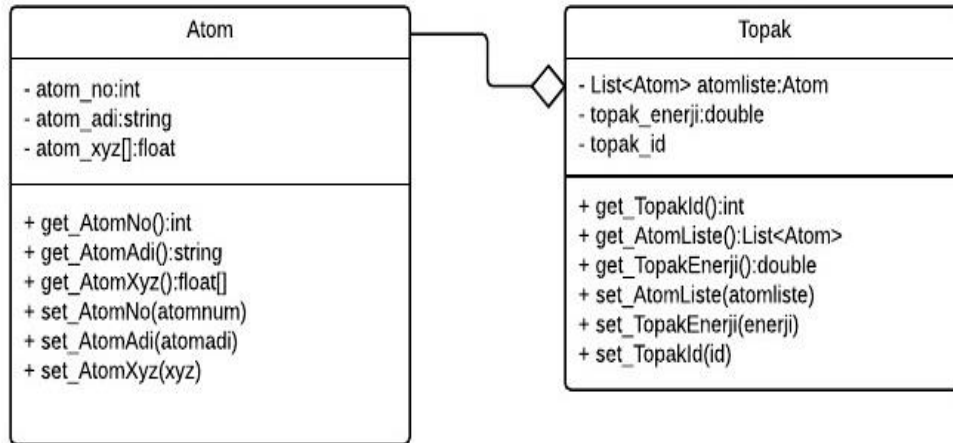
Atom Koordinat Aralığı: Üretilecek atomların koordinatlarının hangi değerler arası üretileceğini belirler.

Mutasyon Türü: Sahip olunan iki farklı mutasyon türünden hangisinin seçileceğini belirler.

Topak Yapı: Üretilen topağın hangi kısıtlara göre oluşturulması gerektiğini belirler.

3.1.2. Kromozom Tasarımı

Nano topakların GA algoritmada birer birey olarak temsil edilmeleri için kromozom tasarımının yapılması gereklidir. Literatürde topak içinde bulunan atomların 3 boyutlu gerçek kartezyen koordinatları kullanılarak yapılan çalışmalar görülmektedir [31, 34, 35]. Bu çalışmada da topakta bulunan her bir atomun 3 boyutlu uzayda sahip olduğu x, y, z koordinat değerlerine [-4, 4] aralığında reel sayılardan atama işlemi yapılmıştır. Her bir topak bir bireyi temsil ederken her bir atom ise bir geni temsil etmektedir. Java dilinde nesne yapısıyla ifade edilmesi Şekil 3.3'teki UML diyagramındaki gibidir.



Şekil 3.3. Kromozom tasarımı

Atom sınıfının içinde atom numarası, atom adı isimli değişkenler ve atom koordinatlarını tutmak için xyz adlı bir dizi bulunmaktadır.

Topak sınıfı; nano topağı oluşturan atomların tutulduğu bir liste, topak id değeri ve topağın enerjisini tutan topak enerji adlı değişkenlerden oluşmaktadır.


3.1.3. Uygunluk Fonksiyonu

Algoritma bireylerin uygunluk değerlerini hesaplamak için Gaussian 09 programını kullanmaktadır [36]. Gaussian 09 topakların tek nokta enerjisini (Single Point Energy) yoğunluk fonksiyonel teorisi (Density functional theory DFT) yöntemiyle hesaplamaktadır. Şekil 3.5'te DFT formülü ve açıklamaları verilmiştir. B3LYP düzeyi ve temel veri seti 3-21G kullanılmıştır. GA programı ve Gaussian 09'un haberleşmesi için Şekil 3.4'te gösterilen Linux kabuk kodları kullanılmıştır. Şekil 3.4'te verilen kod parçasığında com klasörü içindeki, .com uzantılı dosya isimleri bir listeye alındıktan sonra her dosya bir döngü ile teker teker Gaussian 09 programına gönderilmiş ve hesaplanan enerjiler, .log uzantılı kayıt (log) dosyalarında saklanmıştır.

```
#!/bin/bash
g09root=/home/sene/gauss/
export g09root
. $g09root/g09/bsd/g09.profile

files=$( ls -rth /home/sene/Dropbox/Master/TezDeneme/com/* )
c=0
for i in $files ; do
  g09 <"$i" > /home/sene/Dropbox/Master/TezDeneme/log/log$c.log
  let c=c+1
done
exit
```

Şekil 3.4. Linux kabuk kodu

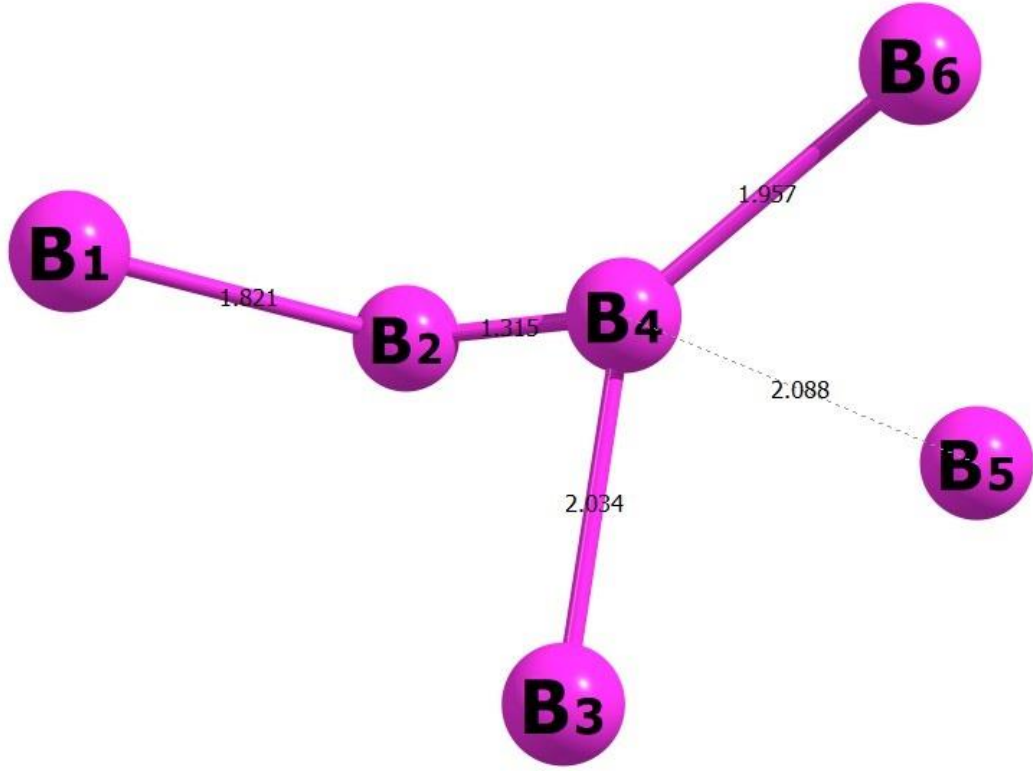
$$E[\rho(\mathbf{r}_i), \mathbf{R}_I] = T_e[\rho(\mathbf{r}_i)] + T_n + E_{en}[\rho(\mathbf{r}_i), \mathbf{R}_I] + E_{ee}[\rho(\mathbf{r}_i)] + E_{nn}(\mathbf{R}_I)$$


↓ ↓ ↓ ↓ ↓
Elektronik kinetik enerji Nükleer kinetik enerji Elektron-çekirdek etkileşimi Elektron-elektron etkileşimi Çekirdek-çekirdek etkileşimi

Şekil 3.5. DFT enerji hesabı formülü

3.1.4. İlk Popülasyonun Oluşturulması

Algoritmanın kromozom tasarımı yapıldıktan ve uygunluk fonksiyonunun belirlenmesinden sonraki adım ilk popülasyonun oluşturulmasıdır. Popülasyon sayısı önceden belirlenmelidir. Bu değer elle girilebileceği gibi program tarafından otomatik olarak da atanabilir. Elle girildiğinde popülasyon sayısı 10' dan daha düşük girilememektedir. Literatüre bakıldığında popülasyon sayısının kaç olması gerektiği üzerine çalışmalar mevcuttur. Genel kani oluşturulan topak N tane atomdan oluşuyorsa popülasyon sayısının 5xN olması gerektiğidir [34]. Popülasyon sayısı belirlendikten sonra sıradaki adım üretilecek olan topakların yapısını belirleme işlemidir. Yapı için iki seçenek sunulmuştur; birinci seçenekte iki koşula göre topaklar oluşturulur. İlk koşul topak içindeki atomlar arası mesafenin en az 1 angström olmasıdır. Topak içindeki her bir atomun birbirlerine olan yakınlığı 1 angströmden daha yakın olamamaktadır. İkinci koşul ise topak içindeki atomların bireysel olarak en az bir tane 3 angströmden küçük mesafede bir komşusunun olmasıdır. Üretilen atomlar topağa eklenirken bu koşullar göz önünde bulundurulur. Bu koşullara uyum sağlamayan atomların koordinatları tekrar oluşturulur. Şekil 3.6'da bu koşullara göre üretilen yapıyı temsil eden bir topak gösterilmektedir.

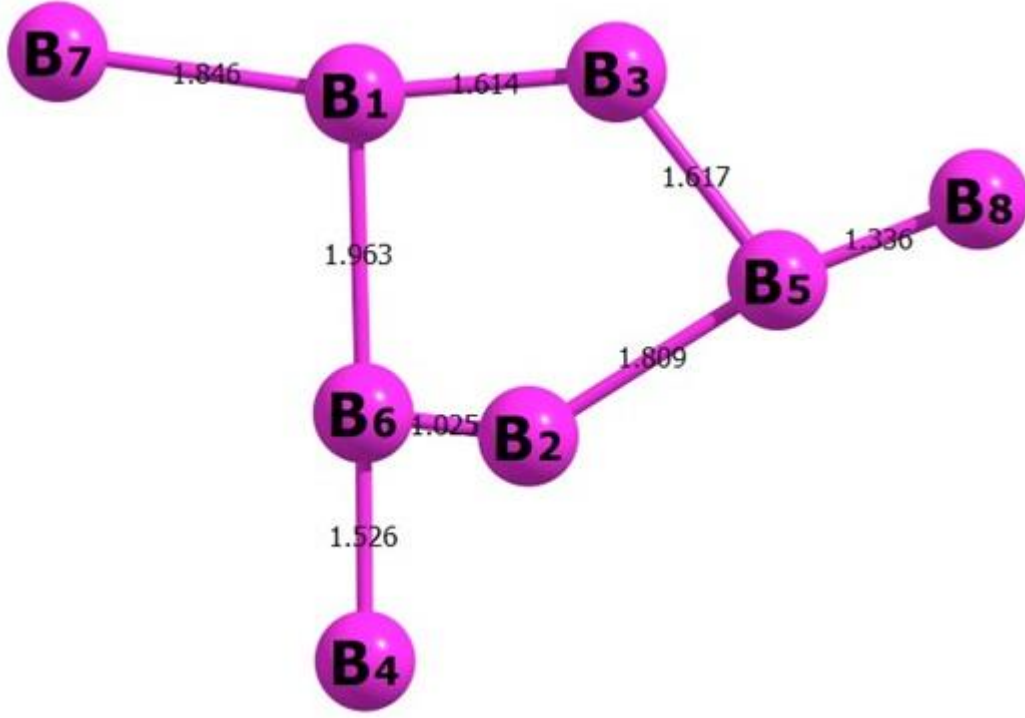


Şekil 3.6. Topak yapısı

Atomlar arası mesafe öklid uzaklık formülüne göre hesaplanmaktadır. Üç boyutlu uzayda yer alan $P = (a, b, c)$ ve $Q = (x, y, z)$ noktalarının öklid uzaklık formülüne göre mesafe hesabı aşağıdaki gibidir.

$$uzaklık = \sqrt{(a - x)^2 + (b - y)^2 + (c - z)^2}$$

Popülasyonunun oluşturulma aşamasında son 10 bireyin atom koordinatlarının z eksenine sıfır atanarak x, y düzleminde bireyler üretilmiştir. Böylece düzlemsel olarak en iyi enerjiye sahip olması ihtimali olan toplarlarda göz önünde bulundurulmuştur. Kullanıcı ikinci seçeneği seçerse ilk popülasyonun tüm toplarını 2 boyutlu düzlemde ürettirebilmektedir. Şekil 3.7’de düzlemsel olarak üretilen bir topak yapısı verilmiştir.



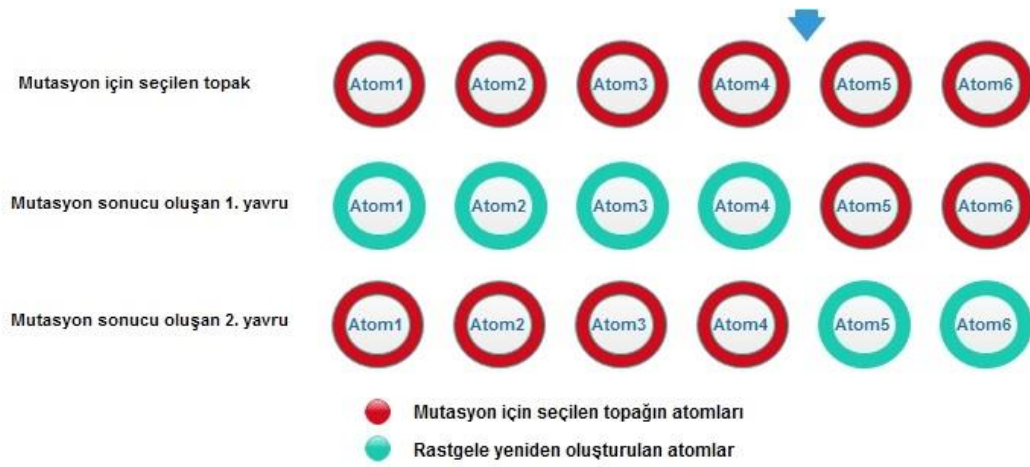
Şekil 3.7. İki boyutlu düzlemsel yapı

3.1.5. Seçim Operatörü

İlk popülasyon oluşturulduktan sonra her bireyin uygunluk değeri hesaplanmaktadır. Uygunluk değeri en yüksek topakların çaprazlama işlemi için seçilmesi gerekmektedir. Seçim için turnuva yöntemi kullanılmıştır. Seçim operatörüne bireyler gönderilmeden önce uygunluk değerleri hesaplanmış bireylerin popülasyon içinde sıralanması yapılır. Sıralamadan sonra popülasyonun yarısı kadar en iyi birey bir seçim havuzuna eklenir ve seçim işlemi turnuva yöntemi ile bu havuzdan yapılır. Böylelikle her durumda en iyi bireyler arasında seçim işlemi gerçekleştirilir. Turnuva yöntemiyle seçilen bireyler üreme havuzuna eklenir. Çaprazlama için seçilecek bireyler bu havuzdan ikişer ikişer alınarak çaprazlama yapılır.

3.1.6. Çaprazlama Operatörü

Üreme havuzundan seçilen bireyler çaprazlamaya tabi tutulur. Bu amaçla iki yöntem kullanılmıştır. İlk yöntem çaprazlama için seçilen iki kromozomun (topağın) genlerinin (atomların) rastgele bir noktadan bölünerek yer değiştirilmesidir. Çaprazlama işlemi yapıldıktan sonra atomlar arası mesafe kontrol edilir eğer uygun değilse bölünmesi için farklı bir nokta seçilir. Şekil 3.8'de çaprazlama yöntemi gösterilmektedir.



Şekil 3.8. Çaprazlama

İkinci çaprazlama yöntemi ise gerçek değerli çaprazlama yöntemidir. Seçilen iki topağın atom koordinatları aşağıdaki formüllere göre işleme tabi tutularak yeni iki adet birey üretilmektedir. Formüllerde y_1, y_2 üretilen yeni bireyleri x_1, x_2 çaprazlama için seçilen ebeveyn bireyleri temsil etmektedir. $a = [0,1]$ aralığında rassal olarak üretilen bir reel sayıdır. Oluşturulan yavru topakların atomlar arası mesafelerinin uygunluğuna bakılmakta, eğer uyumsuz birey oluşmuşsa a değeri tekrar oluşturularak yeni bireyler üretilmektedir.

$$y_1 = ax_1 + (1 - a)x_2$$

$$y_2 = ax_2 + (1 - a)x_1$$

3.1.7. Mutasyon Operatörü

Mutasyon için bireylerin seçimi popülasyon içinden rastgele yapılmıştır. Popülasyon boyutundan büyük olmayacak şekilde rastgele bir sayı üretilir ve bu sayıya popülasyonda karşılık gelen birey alınarak mutasyon operatörüne gönderilir. Mutasyon işlemi için iki yöntem kullanılmıştır. İlk yöntem; mutasyon için herhangi bir birey seçilimi yapılmadan sıfırdan yeni bir topak üretilip popülasyona eklenmesi işlemidir. İkinci yöntemdeyse mutasyon için popülasyondan rastgele seçilen topağın atomları için rassal bir kesme noktası oluşturulur, bu noktadan kesilen topak iki parçaya ayrılır. Bu iki parçanın eksik atomları rassal olarak oluşturularak tamamlanır. Böylelikle bir topaktan iki adet topak elde edilmiş olmaktadır. Şekil 3.9’da uygulanan mutasyon yöntemi gösterilmektedir. Üretilen topakların atomları arası mesafeler göz önünde bulundurulmuştur. Kaç adet mutasyon yapılacağı bilgisi başlangıçta kullanıcıdan alınmaktadır.



Şekil 3.9. Mutasyon

3.1.8. Elitizm

Tüm işlemler bittikten sonraki adım ise popülasyonun bir sonraki iterasyon için hazırlanmasıdır. Çaprazlamadan ve mutasyondan gelen bireylerin popülasyona eklenmesiyle popülasyon boyutu artmıştır. Popülasyon boyutunun ilk haline geri döndürülmesi gerekmektedir. Tüm eklemeleri yapılmış popülasyon uygunluk değerlerine göre en küçük potansiyel enerjiye sahip topaktan en büyük enerjiye sahip topağa doğru sıralanır. Sıralama işleminden sonra ilk girilen popülasyon sayısına ulaşılan kadar sondan itibaren topaklar silinir. Sabit popülasyon sayısına gelindiğinde silme işlemi tamamlanarak popülasyon bir sonraki iterasyon için hazır hale gelir. Silinen topaklar kötü bireyler oldukları için bir sonraki nesle kalan bireyler mevcut durumun en iyi bireyleri olmaktadır.

3.1.9. Durdurma Ölçütü

Algoritmanın durma ölçütü girilen iterasyon sayısına bağlıdır. Algoritma iterasyon sayısına ulaşılan kadar işlemlerine devam etmektedir.

3.2. Programın Sınıf Yapısı ve Sınıfların Üstlendikleri Görevler

Program java dilinde nesne yönelimli programlama yöntemi kullanılarak geliştirilmiştir. Program içinde Atom, Topak, Hesaplamalar, Çaprazlama, Genetik Algoritma, Terminal, Sıralama, Ana Menü, Kontrolcü, isminde sınıflar vardır.(Bkz. EK 1) Sınıfların içeriği ve yaptıkları işlerden biraz bahsedecek olursak:

Atom: Bu sınıf topak içine eklenecek atomların çeşitli özelliklerini modelleyen ve bu özelliklerin gerektiğinde değiştirilmesini sağlayan fonksiyonlara sahiptir. Sınıfın detaylı yapısı Şekil 3.3' te detaylı olarak verilmiştir.

Topak: Bu sınıf program içinde bireyi (kromozomu) temsil etmektedir. Sınıf içinde topağa ait atomlar ve topağın enerjisini tutan alanlar bulunmaktadır. Şekil 3.3'te bu sınıfın yapısı detaylı olarak verilmiştir.

Hesaplamalar: İlk popülasyonun üretilmesi rastgele koordinatların oluşturulması, koordinatlar arası mesafelerin kontrol edilmesi, topakların .com uzantılı dosyalara yazılması, .log uzantılı dosyaların açılıp içinden topak enerjilerinin okunması, topakların yapılarının kontrol edilmesi, işlemlerinden sorumlu olan sınıftır.

Genetik Algoritma: Seçilim yöntemleri ve mutasyon işlemlerinden sorumlu sınıftır.

Terminal: Linux komutlarının çalıştırılmasından sorumlu sınıftır.

Sıralama: Topakların sahip oldukları enerjilerine göre sıralanmasından sorumlu sınıftır.

Çaprazlama: Topakların çaprazlama işlemlerinden sorumlu sınıftır.

Kontrolcü: Tüm süreçleri kontrol eden ve sınıflar arasında koordinasyonu sağlayan sınıftır.

Ana Menü: Programın ara yüzünden sorumlu olan sınıftır.

3.3. Programın Çalışma Aşamaları

Programın çalıştırıldığında ara yüz ekranı görüntülenir. Ara yüzden popülasyon sayısı, atom sayısı, mutasyon oranı, iterasyon sayısı, atomlar, çaprazlama türü, mutasyon türü, topak yapısı, atomlar arası mesafe, atom koordinat aralığı gibi algoritma parametreleri girildikten sonra başlat butonu ile birlikte algoritma koşturmaya başlamaktadır. İlk olarak rastgele koordinatlar oluşturulmakta, bu koordinatlar atom sınıfı içindeki x, y, z değerlerini tutan diziye atanmakta, atomlar ise topaklara dahil edilmektedir. İlk popülasyon oluşturulduktan sonra bireylerin (kromozomların) uygunluk değerleri hesaplanır. Topakların uygunluk değerlerini hesaplamak için Gaussian 09 programı kullanılmıştır. Topakların uygunluk değerlerinin hesaplanabilmesi için, atomik koordinatların Gaussian 09 programına uygun formatta gönderilmesi gerekmektedir.

```
# B3LYP/3-21G scf=(XQC,MaxConventionalCycle=400, MaxCycle=400,Direct,Tight)

Bor atom

0 1
5 -3.6941562 -2.3226223 -1.7457862
5 -3.875657 -2.3862572 -2.7981606
5 -2.6286454 -3.1866627 -0.44695568
5 -3.5807457 -0.05835533 -0.6005297
5 -3.1403518 2.013444 0.416811
5 -1.5939884 -0.20625353 -1.5696507
```

Şekil 3.10. Gaussian 09 com dosyası

Gaussian 09 programı .com uzantılı dosyalar ile işlem yapmaktadır. Şekil 3.10'da böyle bir dosyanın içeriği görülmektedir. Dosyanın ilk satırında hesaplamalar için kullanılması gereken yöntem ve diğer ayrıntılar yer almaktadır. Bunu izleyen satırlarda topağın toplam yükü ve topağı oluşturan atomların kartezyen koordinat sisteminde ki koordinatları bulunmaktadır. GA programı bir klasör içine tüm bireylerin (topakların) .com dosyalarını uygun formatta yazdırmakta ve her bir topak için bir com dosyası oluşturmaktadır. Bu aşamadan sonra yazılan bir Linux kabuk kodu çalıştırılarak klasörün içindeki .com uzantılı dosyalar teker teker Gaussian 09 programına yollanmaktadır. Gaussian 09 programı enerji hesabına dair ayrıntıları, ayrı bir klasöre her topak için ayrı bir kayıt (log) dosyasına yazmaktadır. Bu dosyalarda uygunluk hesabı yapılan topakların potansiyel enerjisi bulunmaktadır. Bu enerjiler kayıt dosyalarından okunarak topaklara yazılır. Tüm popülasyonun enerjileri hesaplandıktan sonra popülasyon küçükten büyüğe doğru sıralanır. Sıralamadan sonra popülasyon sayısının yarısı kadar birey seçim havuzuna alınır. Seçim havuzuna alınan bireyler seçim operatörüne gönderilerek çaprazlama için üreme havuzu oluşturulur. Üreme havuzundan alınan bireyler çaprazlama operatörüne gönderilerek çaprazlama işlemine tabi tutulur. Çaprazlama sonucu oluşan ve enerjileri hesaplanan bireyler popülasyona eklenir. Popülasyon tekrar sıralama işlemine tabi tutularak küçükten büyüğe sıralanır. Popülasyon içinden rastgele seçilen topaklar mutasyon için mutasyon operatörüne gönderilir. Mutasyon işlemi yapıldıktan sonra yeni yavrular com dosyasına yazdırılıp enerji hesabı için Gaussian 09 programına gönderilir ve

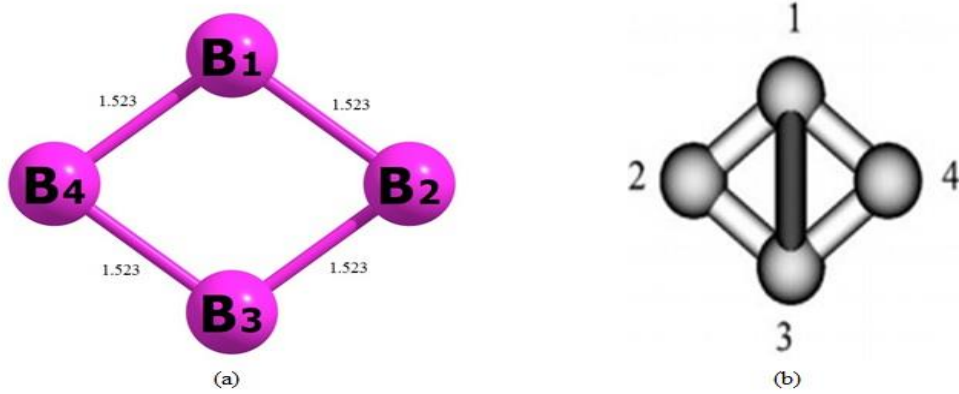
hesaplanan enerjileri kayıt (log) dosyalarına yazılır. Bu kayıt dosyaları okunarak mutasyon sonucu oluşan yavru topakların enerji atamaları yapılır. Enerji atamaları yapılan topaklar popülasyona eklendikten sonra popülasyon tekrar küçükten büyüğe sıralanır. Sıralanan popülasyondan kötü bireyleri elemek için sondan itibaren ilk popülasyon boyutuna varıncaya kadar silme işlemi yapılır. Son haline ulaşan popülasyondaki tüm bireylerin enerjileri bir klasörde enerjilerepok1.txt isimli dosyaya yazdırılır. Her iterasyon sonunda tüm topakların enerjileri ayrı enerjilerepok.txt dosyasında saklanır ve dosya isminin sonuna bulunan enerjilerin ait olduğu iterasyonun numarası da eklenir. Sıralı popülasyonun en başındaki topak alınır ve o iterasyonun en iyi bireyi olarak bir klasörde saklanır. Bu aşamadan sonra program elindeki mevcut popülasyonla seçim havuzu oluşturma adımına giderek işlemleri iterasyon sayısı bitene kadar tekrar ettirir. İterasyon sayısı bittiğinde elde kalan son popülasyonun tüm bireyleri saklanmak üzere ayrı bir klasör içinde .com uzantılı dosyalara yazdırılır. Programın son aşamasında bulunan son popülasyonun en iyi bireyi alınarak optimizasyon için Gaussian 09 programına yollanır.

4. TARTIŞMA VE SONUÇ

Algoritma Bor ve Lityum atomlarından oluşan nano topaklar üzerinde denenmiştir. Çalışmada seçilen bor ve lityum topakları üzerinde daha önceden yapılmış deneysel ve teorik çalışmalar mevcuttur [34], [35]. Bir çok araştırmacı topakların global minimumunu bulmak için küçük yapıların seçilmesi üzerinde hem fikir olmuşlardır. Bu sebepten ötürü seçilen B₄, B₅, B₆, B₈, Li₅, Li₆, Li₇ topakları geliştirilen program için iyi bir test verisi olmuştur. Program farklı algoritma parametreleri girilerek her yapı için 10'ar kez çalıştırılmış en kısa sürede en iyi sonuç veren parametreler sonuçlarda paylaşılmıştır.

4.1. B₄ Nano Topağı için Algoritmanın Çalıştırılması

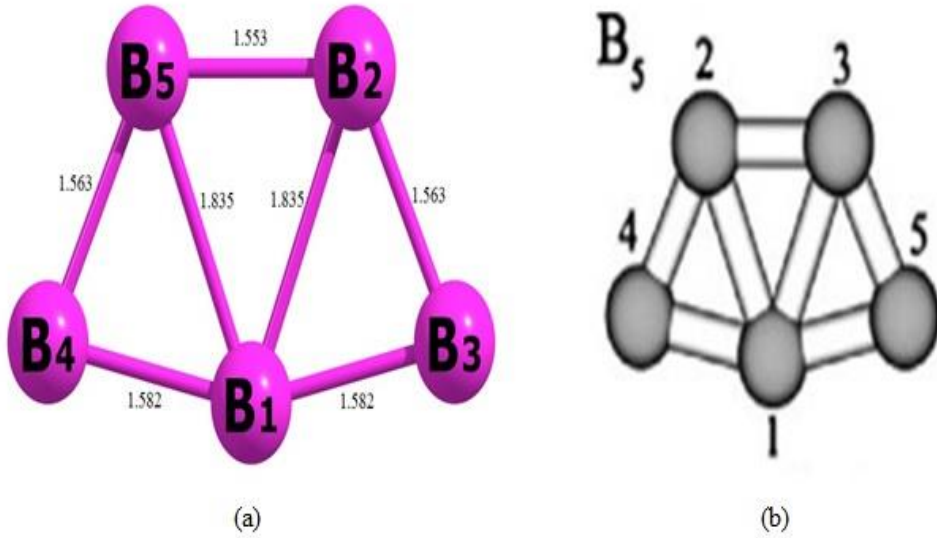
B₄ topağı için popülasyon sayısı 20, mutasyon oranı %40 ve iterasyon sayısı 50 girilerek program çalıştırılmış ve aşağıdaki sonuçlar alınmıştır. Çaprazlama işleminde gerçek çaprazlama yöntemi kullanılmıştır. Algoritmanın 50 iterasyon sonunda bulunduğu yapının enerjisi:-99,157 a.u.'dur. Literatürde bulunan B₄ topağının en düşük enerjisi:-99.158 a.u.'dur [34]. Şekil 4.1'de a ve b sırasıyla, programın bulunduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.1. B₄ topağı sonuç ve literatür karşılaştırması (b) [34]

4.2. B₅ Nano Topağı için Algoritmanın Çalıştırılması

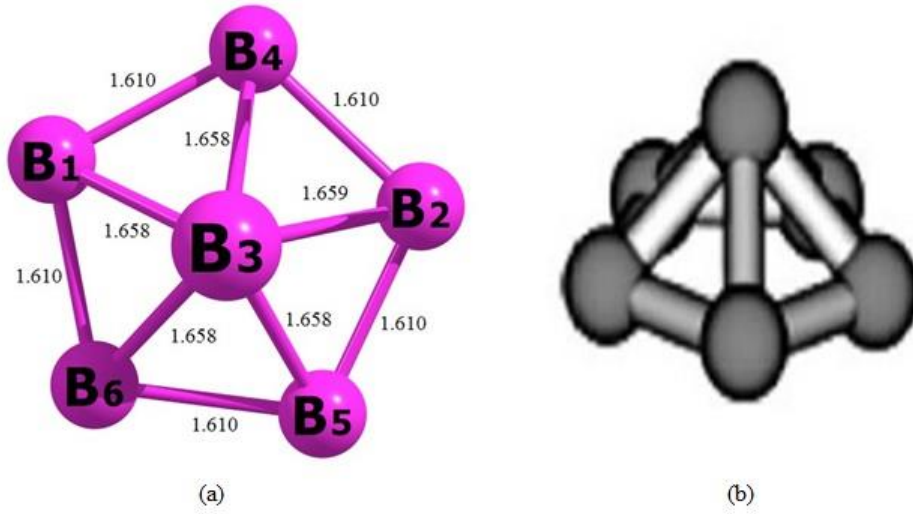
B₅ topağı için popülasyon sayısı 20, mutasyon oranı %40, iterasyon sayısı 50 girilerek program çalıştırılmıştır. Çaprazlama işleminde gerçek çaprazlama yöntemi kullanılmıştır. Algoritmanın 50 iterasyon sonunda bulduğu yapının enerjisi :-124,003 a.u.'dur. Literatürde bulunan B₅ topağının en düşük enerjisi: -124,003 a.u.'dur [34]. Şekil 4.2'de a ve b'de sırasıyla, programın bulduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.2. B₅ topağı sonuç ve literatür karşılaştırması (b) [34]

4.3. B₆ Nano Topağı için Algoritmanın Çalıştırılması

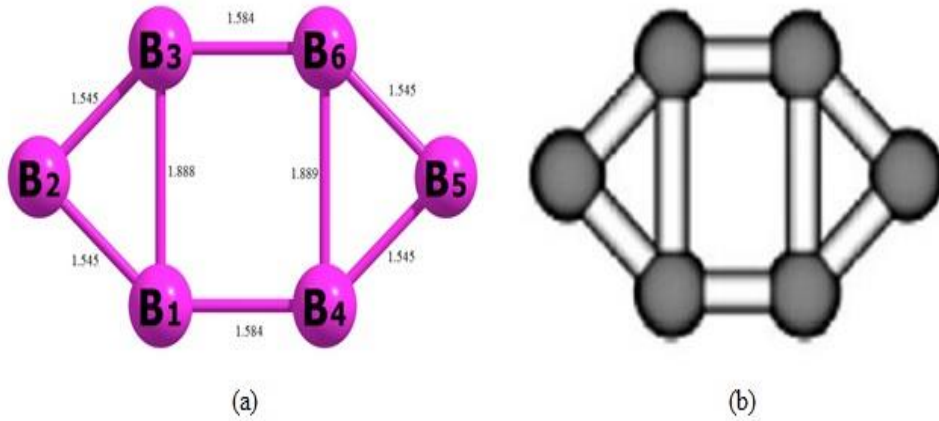
B₆ topağı için popülasyon sayısı 40, mutasyon oranı %40 ve iterasyon sayısı 100 girilerek program çalıştırılmıştır. Çaprazlama işleminde gerçek çaprazlama yöntemi kullanılmıştır. Algoritmanın 100 iterasyon sonunda bulduğu yapının enerjisi :-148,827 a.u.'dur. Literatürde bulunan B₆ topağının mevcut izomerinin enerjisi: -148,827 a.u.'dur [34]. Şekil 4.3'te a ve b'de sırasıyla, programın bulduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.3. B₆ topağı sonuç ve literatür karşılaştırması (b) [34]

Program B₆ için popülasyon sayısı 30, mutasyon oranı %20, iterasyon sayısı 100, girilerek çalıştırıldığında bulunan izomer topağın enerjisi: -148,828 a.u. olarak hesaplanmıştır. Literatürde bulunan söz konusu B₆ izomer topağının en düşük enerjisi: -148,829 a.u.'dur [34].

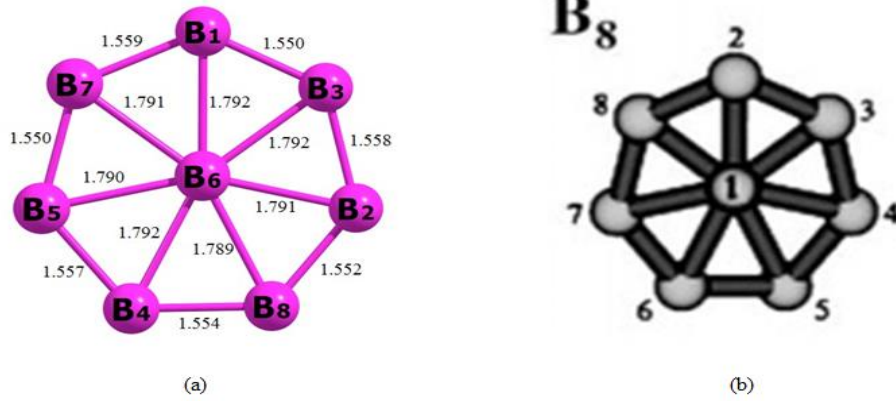
Şekil 4.4'te a ve b'de sırasıyla, programın bulduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.4. B₆ topağı izomeri sonuç ve literatür karşılaştırması (b) [34]

4.4. B₈ Nano Topađı için Algoritmanın alıřtırılması

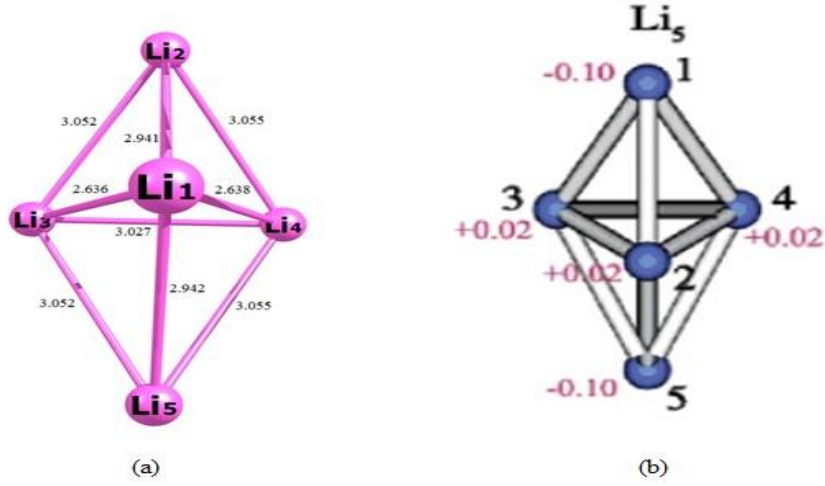
B₈ topađı için poplasyon sayısı 40, mutasyon oranı %40, iterasyon sayısı 100, girilerek program alıřtırılmıřtır. aprazlama iřleminde gerek aprazlama yntemi kullanılmıřtır. Algoritmanın 100 iterasyon sonunda bulduđu yapının enerjisi: -198,590 a.u.'dur. Literatrde bulunan en iyi yapının enerjisi: -198.596 a.u.'dur [34]. Őekil 4.5'te a ve b'de sırasıyla, programın bulduđu yapı ve literatrde bulunan en iyi yapıyı temsil etmektedir.



Őekil 4.5. B₈ topađı izomeri sonu ve literatr karřılařtırması (b) [34]

4.5. Li₅ Nano Topađı için Algoritmanın alıřtırılması

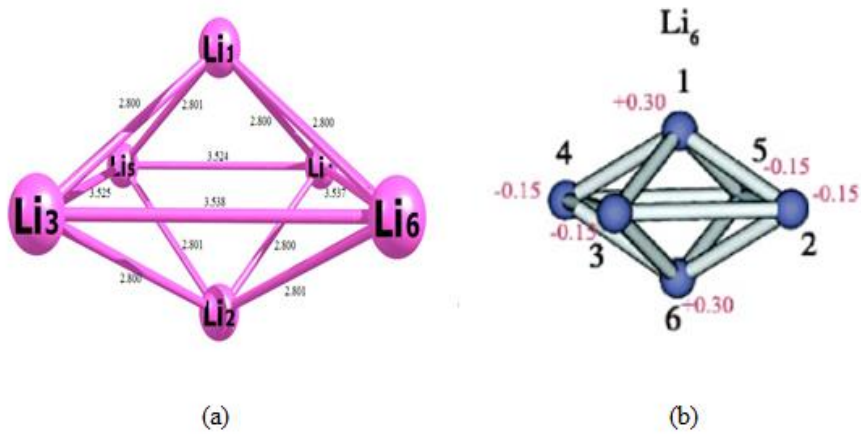
Li₅ topađı için algoritma deđerleri; poplasyon sayısı 30, iterasyon sayısı 50, %40 mutasyon oranı girilerek program alıřtırılmıřtır. aprazlama iřleminde gerek aprazlama yntemi kullanılmıřtır. Programın 50 iterasyon sonunda bulduđu yapının enerjisi : -37,580 a.u.'dur. Literatre bakıldıđında L₅ topađı için bulunan en dřk enerji : -37,580 a.u.'dur [35]. Őekil 4.6'da a ve b sırasıyla programın bulduđu yapı ve literatrde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.6. Li_5 topağı sonuç ve literatür karşılaştırması (b) [35]

4.6. Li_6 Nano Topağı için Algoritmanın Çalıştırılması

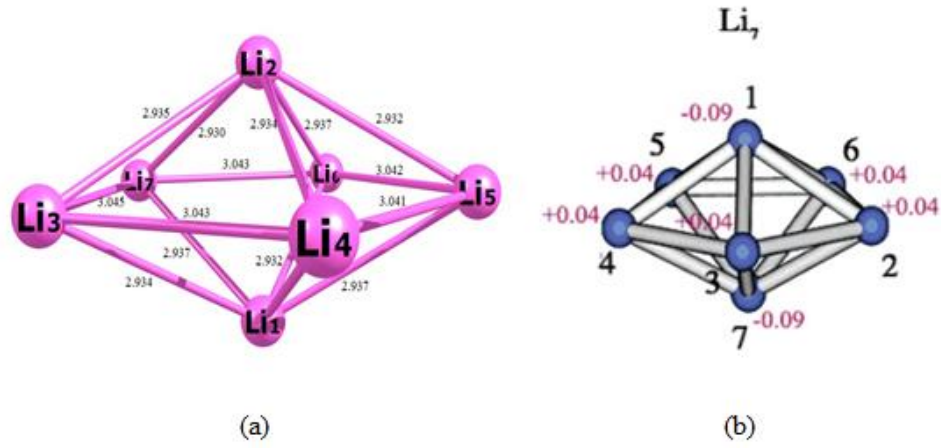
Li_6 topağı için algoritma değerleri; popülasyon sayısı 30, iterasyon sayısı 50, %40 mutasyon girilerek program çalıştırılmıştır. Çaprazlama işleminde gerçek çaprazlama yöntemi kullanılmıştır. Programın 50 iterasyon sonunda bulunduğu yapının enerjisi : -45,116 a.u.'dur Literatüre bakıldığında L_6 topağı için bulunan en düşük enerji : -45,116 a.u.'dur [35]. Şekil 4.7'de a ve b sırasıyla programın bulunduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.7. Li_6 topağı sonuç ve literatür karşılaştırması (b) [35]

4.7. Li₇ Nano Topağı için Algoritmanın Çalıştırılması

Li₇ topağı için algoritma değerleri; popülasyon sayısı 30, iterasyon sayısı 50, %40 mutasyon oranı girilerek program çalıştırılmıştır. Çaprazlama işleminde gerçek çaprazlama yöntemi kullanılmıştır. Programın 50 iterasyon sonunda bulunduğu yapının enerjisi : -52,652 a.u.'dur. Literatüre bakıldığında L₇ topağı için bulunan en düşük enerji : -52,652 a.u.'dur [35]. Şekil 4.8'de a ve b sırasıyla programın bulunduğu yapı ve literatürde bulunan en iyi yapıyı temsil etmektedir.



Şekil 4.8. Li₇ topağı sonuç ve literatür karşılaştırması (b) [35]

4.8. Sonuçları Değerlendirme ve Gelecekte Yapılacaklar

B₄, B₅, B₆, B₈, Li₅, Li₆, Li₇ topak yapıları ile yapılan denemeler sonucunda başarılı sonuçlar elde edilmiştir. Sonuçlar incelendiğinde bulunan yapılar ve enerjileri literatürdeki yapılar ile örtüştüğü görülmektedir. Bulunan yapılar için algoritma parametreleri değiştirilerek program çalıştırılmış ve en uygun parametreler paylaşılmıştır. Topaklar için üretilen atomların koordinatları [-4,4] aralığında ürettirilmiştir. Bu aralığın fazla yada az olması üretilen topağın yapısını bozduğunu gözlemlenmiştir. Literatür ile karşılaştırıldığında ilk kez gerçek sayılı çaprazlama ve farklı bir mutasyon tekniği kullanılmıştır. Kullanılan gerçek

aprazlama yntemi kes yapıştır aprazlama yntemine gre daha hızlı alıřtıđı grlmřtr. Mutasyon iin Őekil 3.9’da gsterilen yntemin diđer ynteme gre daha hızlı sonular verdiđi grlmřtr. Ayrıca algoritma her seferinde optimizasyon iin zaman kaybı yařamadıđı iin literatrdeki rneklere gre daha hızlı alıřmaktadır.

Bu ařamadan sonra ki hedef farklı tr atomlar ieren ve atom sayısı daha fazla olan topaklar zerinde programın denemesidir.

KAYNAKLAR

- [1] Mongillo, J., Nanotechnology 101, GREENWOOD PRESS, London, 2007.
- [2] Goldstein, S. C., Introduction to Nanotechnology Lecture Notes, Carnegie Mellon University, 2004.
- [3] Feynman, R. P., There's Plenty of Room at the Bottom, Engineering and Science, 1959.
- [4] Taniguchi, N., On the Basic Concept of 'Nano-Technology, Proc. Intl. Conf. Prod. Eng. Tokyo, Part II, Japan Society of Precision Engineering, 1974.
- [5] Drexler, K. E., Engines of Creation: the coming era of nanotechnology, Anchor Press, 1986.
- [6] Love, C., Whitesides, G. M., The Art of Building Small, Scientific American, 2001.
- [7] Roukes, M. L., Plenty of Room, Indeed, Scientific American, 2001.
- [8] Levit, C., Bryson, S. T., Henze, C. E., Virtual Mechanosynthesis, Fifth Foresight Conference on Molecular Nanotechnology, 1997.
- [9] Falvo, M. R., Clary, G. J., Taylor, R. M., Chi, V., Brooks, F. P., Washburn, S., Superfine, R., Bending and buckling of carbon nanotubes under large strain, Nature, s. 389, 582-584, 1997.
- [10] Kaewkamnerdpong, B., Bently, P. J., Computer Science for Nanotechnology: Needs and Opportunities, Department of Computer Science, University College London, UK .

- [11] Stix, G., Little Big Science, Scientific American, 2001.
- [12] Taylor II, R. M., The Nanomanipulator: A Virtual-Reality Interface for a Scanning Tunneling , Proceedings of SIGGRAPH, 1993.
- [13] Bentley, S. Kumar and P. J., On Growth, Form and Computers, Academic Press, 2003.
- [14] M. M., Kevran, Çoklu Sensör Konumlandırma Problemlerinin Genetik Algoritmalar İle Çözülmesi, Yüksek Lisans Tezi, Hava Harp Okulu Havacılık ve Uzay Teknolojileri Enstitüsü, s. 8-14, 2009.
- [15] K.Çakar, Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu, Doktora Tezi, Erzurum, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü, s 39-76 , 2009
- [16] A.,Coşkun, Genetik Algoritma Kullanılarak Kimyasal Maddelerin Deriden Geçiş Katsayılarının ve Molekül Yapılarının Bulunması, Doktora Tezi, Ankara, Gazi Üniversitesi Fen Bilimleri Enstitüsü, s. 27-52, 101-106, 2006.
- [17] Glover, F., Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research, 1983.
- [18] B.,Kemer, Araç Rotalama Problemlerine Genetik Algoritma Yaklaşımı: Bir Gıda Dağıtım Firması Uygulaması, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü, s. 3-11 , 2010.
- [19] Kirkpatrick, S., Gelatt, C. D. ve Vecchi, M. P., Optimization by Simulated Annealing, Science, 1993.
- [20] Colormi,A., Dorigo, M., Maniezzo,V., Distributed Optimization by Ant Colonies, Elsevier, 1991.
- [21] Holland, J. H, Genetic algorithms, Scientific American, 1992.

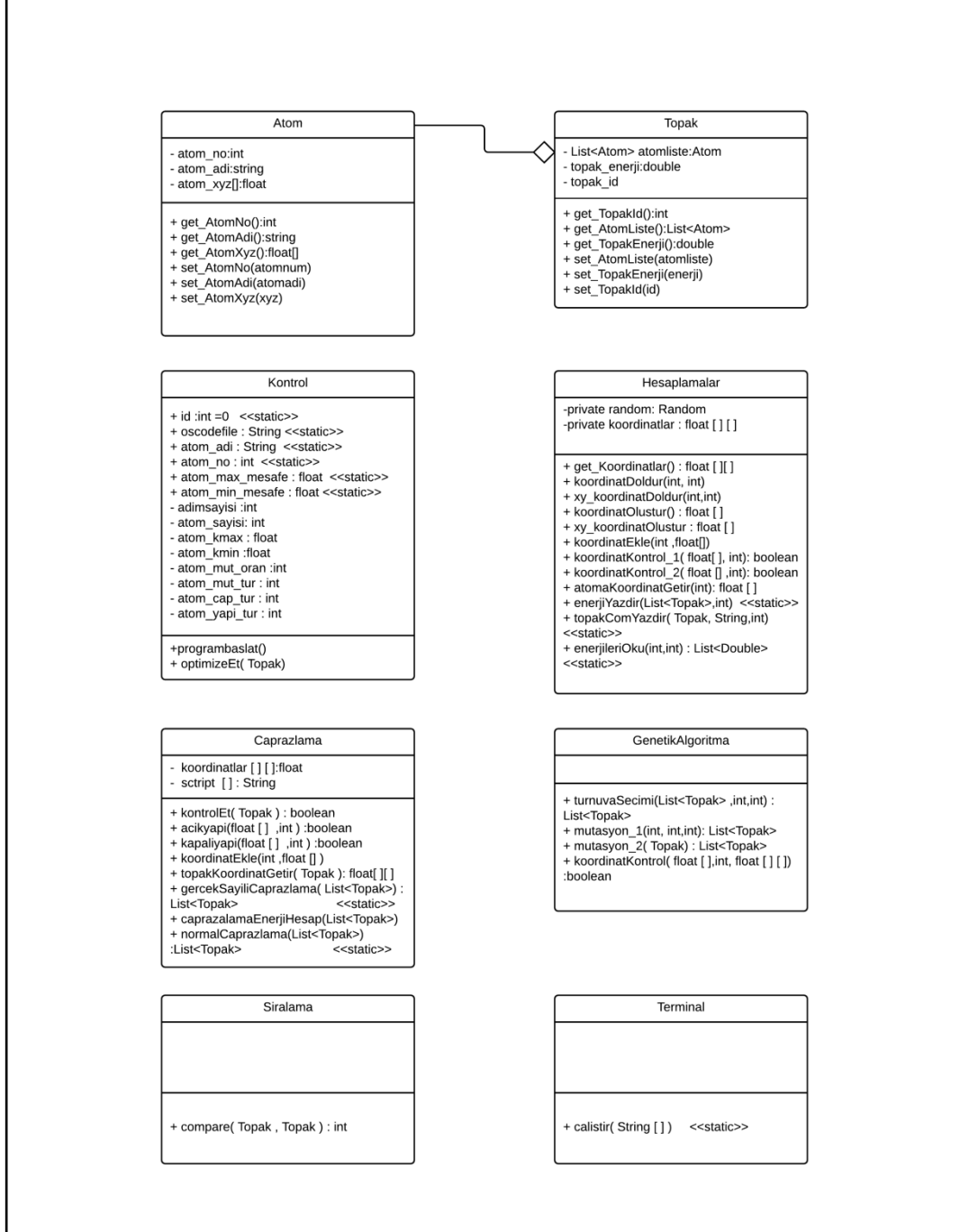
- [22] Emel, G. G., Taşkın, Ç., Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, no. 1, s. 129-152, 2002.
- [23] Turgut, P., Gümüşçü, M., Arslan, A., Genetik Algoritmalar ve Çalışma Prensipleri, Mühendislik Kongresi Bildiriler Kitabı, 2002.
- [24] F., Akpınar, Yerleştirme Rotalama Problemi İçin Bir Genetik Algoritma, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, s 15-23, 2009.
- [25] Hartke, B. J. Phys. Chem, s. 97, 1993.
- [26] Williams, Y. Xiao and D. E., Chem. Phys. Lett., s. 215, 1993.
- [27] Hartke, B., Chem. Phys. Lett., s. 258, 1996.
- [28] Hartke, B., Phys. Chem. Chem. Phys, s. 5, 2003.
- [29] Hartke, B., Flad, H.-J., Dolg, M., Phys. Chem. Chem. Phys, s. 3, 2001.
- [30] Zeiri, Y., Phys. Rev. E, s. 51, 1995.
- [31] Johnston, R. L., Evolving better nanoparticles: Genetic algorithms for optimising, Royal Society of Chemistry, 2003.
- [32] Deaven, D. M., Ho, K. M., Phys. Rev. Lett, p. 75, 1995.
- [33] Wales, D. J., Doye ,J. P. K., J. Phys. Chem. A, p. 101, 1997.
- [34] Alexandrova, A. N., Boldyrev,A. I., Zhai,H. J., Wang,L.S., All-boron aromatic clusters as potential new inorganic ligands and building blocks in chemistry, Coordination Chemistry Reviews, 2006.

[35] Alexandrova, A. N., Boldyrev, A. I., Search for the Li_n ($n=5-7$) Lowest-Energy Structures Using the ab Initio Gradient Embedded Genetic Algorithm (GEGA). Elucidation of the Chemical Bonding in the Lithium Clusters, *Journal of Chemical Theory and Computation*, 2005.

[36] Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, Gaussian 09, 2009.

EKLER

EK 1. PROGRAMIN UML DİYAGRAMI



EK 2. PROGRAMIN DOSYA YAPISI

```
KUGA\kabukcode
KUGA\runfiles
KUGA\kabukcode\runc.sh
KUGA\kabukcode\rungausse.sh
KUGA\kabukcode\runm.sh
KUGA\runfiles\bestcom
KUGA\runfiles\ccom
KUGA\runfiles\clog
KUGA\runfiles\com
KUGA\runfiles\eniyienerjiler
KUGA\runfiles\eniyipopulasyon
KUGA\runfiles\log
KUGA\runfiles\mcom
KUGA\runfiles\mlog
KUGA\runfiles\optimum
KUGA\src\com\kku\tr
KUGA\src\com\kku\tr\Atom.java
KUGA\src\com\kku\tr\Topak.java
KUGA\src\com\kku\tr\Kontrol.java
KUGA\src\com\kku\tr\Caprazlama.java
KUGA\src\com\kku\tr\GenetikAlgoritma.java
KUGA\src\com\kku\tr\AnaMenu.java
KUGA\src\com\kku\tr\Hesaplamalar.java
KUGA\src\com\kku\tr\Siralama.java
KUGA\src\com\kku\tr\Terminal.java
```