

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY ATOM ALGORİTMASI
VE
AYRIK PROBLEMLERE UYGULANMASI

Ayşe ERDOĞAN YILDIRIM

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ŞUBAT-2018

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY ATOM ALGORİTMASI
VE
AYRIK PROBLEMLERE UYGULANMASI

Ayşe ERDOĞAN YILDIRIM

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ŞUBAT-2018

Tezin Başlığı: Yapay Atom Algoritması ve Ayrık Problemlere Uygulanması

Tezi Hazırlayan: Ayşe ERDOĞAN YILDIRIM

Sınav Tarihi: 02/02/2018

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliği Ana Bilim Dalında Doktora Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı:

Prof. Dr. Ali KARCI

İnönü Üniversitesi

Prof. Dr. İbrahim TÜRKOĞLU

Fırat Üniversitesi

Doç. Dr. Mehmet Emin TAĞLUK

İnönü Üniversitesi

Doç. Dr. Galip AYDIN

Fırat Üniversitesi

Doç. Dr. Davut HANBAY

İnönü Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL

Enstitü Müdürü

ONUR SÖZÜ

Doktora Tezi olarak sunduğum “Yapay Atom Algoritması ve Ayrık Problemlere Uygulanması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Ayşe ERDOĞAN YILDIRIM



ÖZET

Doktora Tezi

YAPAY ATOM ALGORİTMASI VE AYRIK PROBLEMLERE UYGULANMASI

Ayşe ERDOĞAN YILDIRIM

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

97+xv sayfa

2018

Danışman: Prof. Dr. Ali KARCI

Gerçek dünyada, bir problemin en ideal çözümünün arandığı birçok alan mevcuttur. Bunlar arasında mühendislik, tıp, diyetetik, lojistik, elektronik gibi alanlar sayılabilir. Bu farklı alanlardaki optimizasyon problemlerinin çözümünde, hem klasik matematiksel yöntemler hem de sezgisel yöntemler tercih edilebilir. Buna karşın doğadan esinlenerek geliştirilen sezgisel algoritmalar, işlem hızı ve doğrulukları ile günümüzde optimizasyon problemlerinin çözümü için popüler yöntemlerdir. Son zamanlarda, bu yöntemlere bir yenisi daha eklenmiştir. O da, birden çok sezgisel tekniği içerisinde bulunduran, meta-sezgisel yöntemler olarak isimlendirilmektedir. Yakın geçmişte literatüre giren ve atomların bileşik oluşturma süreçlerinden esinlenerek geliştirilen meta-sezgisel yöntem, bu tez çalışmasının konusunu oluşturmaktadır. Kimyasal iyonik bağ ve kovalent bağ süreçlerinin modellenmesiyle geliştirilen bu algoritma, ayrık optimizasyon problemlerine uygulanarak, algoritmanın bu alandaki performansı test edilmeye çalışılmıştır.

Bir optimizasyon algoritmasının performansında, ayrık optimizasyon problemlerinde gösterdiği başarı önemli bir kriterdir. Bu nedenle, ilk olarak literatürde yaygın kullanıma sahip olan gezgin satıcı problemi üzerinde çalışılmıştır. Küçük ölçekli ve simetrik problemlere uygulanan yaklaşım; tabu arama, parçacık sürü optimizasyonu, karınca koloni optimizasyonu, genetik algoritma gibi sıklıkla tercih edilen meta-sezgisel yöntemlerle karşılaştırılmıştır. Ardından, bireyin günlük enerji ihtiyacına uygun gıdalardan oluşan beslenme çizelgesinin elde edilmesini amaçlayan bir optimizasyon problemi üzerinde algoritmanın performansı izlenmiştir. Çalışmada, bireyin fiziksel, fizyolojik ve sosyolojik özellikleri dikkate alınarak günlük enerji ihtiyacı hesaplanmıştır. Son olarak, bir binada çalışan birden çok asansörün tek bir merkezden kontrol edilerek, kullanıcılarına etkin

asansör hizmeti sunulması amacıyla bir optimizasyon gerçekleştirilmiştir. Elde edilen sonuçlar karşılaştırmalı olarak verilmiştir. Bu tez çalışmasında yapılan uygulamalarla, makul sürelerde başarılı sonuçlar elde edildiği görülmüştür.

ANAHTAR KELİMELEER: Hesapsal zekâ, optimizasyon, meta-sezgisel algoritma, yapay atom algoritması, gezgin satıcı problemi, beslenme optimizasyonu, grup asansör kontrol optimizasyonu



ABSTRACT

Ph.D. Thesis

ARTIFICIAL ATOM ALGORITHM AND ITS APPLICATION TO DISCRETE PROBLEMS

Ayşe ERDOĞAN YILDIRIM

Inonu University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

97+xv pages

2018

Supervisor: Prof. Dr. Ali KARCI

In the real world, there are many areas where the ideal solution of a problem is sought. These areas include engineering, medicine, dietetic, logistics and electronics. In order to solve optimization problems in these different areas, both classical mathematical methods and heuristic methods can be preferred. On the other hand, heuristic algorithms were developed by inspired from nature are popular methods for the solution of optimization problems with their run-time speed and accuracy. Recently, a new one has been added to these methods. It is also called meta-heuristic method, which is contained in multiple intuitive techniques. The meta-heuristic method, which was recently introduced into the literature and developed by inspired the compounding processes of atoms, is the subject of this thesis. This algorithm which was developed by modeling chemical ionic bond and covalent bond processes, was applied to discrete optimization problems and tried to test the performance of the algorithm in this area.

In the performance of an optimization algorithm, the success in discrete optimization problems is an important criterion. For this reason, firstly, the traveling salesman problem which has widespread use in the literature has been studied. The approach which was applied to small scale and symmetric problems, was compared with frequently-used meta-heuristic methods such as tabu search, particle swarm optimization, ant colony optimization, genetic algorithms. Then, the performance of the algorithm was monitored on an optimization problem which aims to obtain a nutrition schedule consisting of food according to the daily energy requirement of person. In the study, the daily energy requirement was calculated considering the physical, physiological and sociological characteristics of the individual.

Finally, an optimization was performed to provide efficient elevator service to its users by controlling multiple elevators from a center in the building. The obtained results were given comparatively. In this thesis, successful results for the applications were achieved at reasonable times.

KEYWORDS: Computational intelligence, optimization, meta-heuristic algorithm, artificial atom algorithm, traveling salesman problem, nutrition optimization, group elevator control optimization



TEŞEKKÜR

Doktoram süresince ve bu tez çalışmasının her aşamasında önerilerini ve desteklerini benden esirgemeyerek, bilgi ve tecrübesiyle beni aydınlatan çok kıymetli hocam Sayın Prof. Dr. Ali KARCI' ya;

Çalışmalarım sırasında her zaman verdikleri motivasyonla beni destekleyen, her konuda fikirlerine başvurduğum sevgili arkadaşlarım Mekatronik Mühendisliği Anabilim Dalı Arş. Gör. Beyda TAŞAR' a, Bilgisayar Mühendisliği Anabilim Dalı Arş. Gör. Gülşah KARADUMAN' a ve Arş. Gör. Betül KARAKUŞ' a;

Tez ve makale çalışmalarım sırasında fikirlerine ve yardımlarına başvurduğum doktora arkadaşım Yrd. Doç. Dr. Murat CANAYAZ' a;

Hayatımın her aşamasında yanımda olup, benden maddi ve manevi hiçbir yardımı esirgemeyen canım anneme, babama, özellikle tez yazımım esnasında oğlumun bakımını üstlenerek beni destekleyen sevgili eşim Fatih YILDIRIM' a, neşe kaynağım biricik oğlum H. Emre YILDIRIM' a, kardeşim M. Yusuf ERDOĞAN' a ve tüm aileme;

Tez ve makale çalışmalarım esnasında farklı konularda desteğini aldığım, gücünün yetmediği durumlarda bile tatlı diliyle ve güzel yüreğiyle yanımda olduğunu hissettiren sevgili yeğenim merhum Ahsen Elif TURĞUT' a teşekkür ederim.

İÇİNDEKİLER

ONUR SÖZÜ	iii
ÖZET.....	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER LİSTESİ.....	xi
ÇİZELGELER LİSTESİ.....	xiii
SİMGELER VE KISALTMALAR	xv
1. GİRİŞ	1
1.1. Tez Çalışmasının Amacı ve Kapsamı	1
1.2. Literatür Taraması ve Değerlendirmesi.....	2
1.3. Tez Çalışmasının Organizasyonu.....	3
2. META-SEZGİSEL YÖNTEMLER	4
2.1. Optimizasyon Problemleri	4
2.1.1. Optimizasyon problemlerinin sınıflandırılması	4
2.2. Optimizasyon Algoritmaları.....	5
2.2.1 Matematiksel optimizasyon algoritmaları.....	6
2.2.2. Sezgisel optimizasyon algoritmaları	6
2.2.3. Meta-sezgisel optimizasyon algoritmaları	7
3. YAPAY ATOM ALGORİTMASI (A³)	15
3.1. Temel Kavramlar	16
3.2. Kovalent Bağ Operatörü	18
3.3. İyonik Bağ Operatörü.....	22
3.4. Elektron Etkisi.....	24
3.4.1. Amaç fonksiyonu bağımsız değişkenlere ayrılabilen elektronların etki değeri	24
3.4.2. Amaç fonksiyonu bağımsız değişkenlere ayrılamayan elektronların etki değeri.....	25
3.5. Algoritma Adımları.....	25
4. TEZİN LİTERATÜRE KATKISI.....	28
5. DENEYSEL UYGULAMALAR.....	32
5.1. Gezgin Satıcı Problemi	32
5.1.1. A ³ ile GSP uygulaması.....	33
5.2. Beslenme Optimizasyonu Problemi	42
5.2.1. A ³ ile beslenme optimizasyonu	47
5.3. Grup Asansör Kontrol Optimizasyonu Problemi	52
5.3.1. A ³ ile grup asansör kontrol optimizasyonu uygulaması	54

6.	BULGULAR.....	59
6.1.	GSP için Deneysel Sonuçlar	59
6.2.	Beslenme Optimizasyonu Problemi için Deneysel Sonuçlar	72
6.3.	Grup Asansör Kontrol Optimizasyonu için Deneysel Sonuçlar.....	75
7.	SONUÇ.....	79
8.	KAYNAKLAR	82
	EKLER.....	91
	ÖZGEÇMİŞ.....	97



ŞEKİLLER LİSTESİ

Şekil 3.1. Kovalent bağ sürecinin temsili gösterimi.....	15
Şekil 3.2. İyonik bağ sürecinin temsili gösterimi.....	16
Şekil 3.3. Atomun gösterimi	16
Şekil 3.4. Atom kümesi, kovalent alan ve iyonik alanın gösterimi.....	17
Şekil 3.5. Kovalent bağ operatörünün gösterimi.....	20
Şekil 3.6. Kovalent bağ operatörünün akış diyagramı	21
Şekil 3.7. İyonik bağ operatörünün akış diyagramı	23
Şekil 3.8. A^3 'ün akış diyagramı	26
Şekil 4.1. Kıyaslama problemleri için A^3 'ün yapısı.....	29
Şekil 4.2. 8 şehirli bir GSP için A^3 'ün sürekli problemlerdeki kullanımına göre iyonik bağ operatörünün uygulanışı.....	29
Şekil 4.3. GSP için A^3 'ün yapısı	30
Şekil 4.4. 8 şehirli bir GSP için A^3 'ün ayrık problemler için geliştirilen formuna göre iyonik bağ operatörünün uygulanışı.....	30
Şekil 5.1. GSP uygulaması için A^3 'ün akış diyagramı.....	35
Şekil 5.2. A^3 'ün iyonik bağ operatörünün GSP için uygulanışına bir örnek.....	37
Şekil 5.3. 2-opt. yöntemi (A) Uygulanmadan önceki hali (B) Uygulandıktan sonraki hali.	38
Şekil 5.4. Eğer $k = 2$, $l = 6$ ise, 2-opt. yöntemiyle bir tur (A) Uygulanmadan önce (B)Uygulandıktan sonra.....	41
Şekil 5.5. Eğer $k = 2$, $l = 6$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim	41
Şekil 5.6. Eğer $k = 7$, $l = 1$ ise, uygulanan yönteme göre bir atomda oluşacak değişim ..	41
Şekil 5.7. Eğer $k = 7$, $l = 3$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim	42
Şekil 5.8. Eğer $k = 6$, $l = 1$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim	42
Şekil 5.9. Eğer $k = 4$, $l = 2$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim	42
Şekil 5.10. Besin Piramidi	44
Şekil 5.11. A^3 ile beslenme optimizasyonu problemi için atom kümesinin temsili.....	49
Şekil 5.12. A^3 ile beslenme optimizasyonu problemi için atom kümesinin sıralanması.....	49
Şekil 5.13. A^3 ile beslenme optimizasyonu problemi için kovalent bağ operatörünün işleyişi	50
Şekil 5.14. A^3 ile beslenme optimizasyonu problemi için iyonik bağ operatörünün işleyişi.	51
Şekil 5.15. Bir binada çoklu asansör kullanımı	52
Şekil 5.16. Asansör kabinlerinin başlangıçtaki konumları ve kat çağrıları.....	55
Şekil 6.1. A^3 ile burma14 GSP uygulaması için ulaşılan optimum rota.....	60
Şekil 6.2. A^3 ile bayg29 GSP uygulaması için ulaşılan optimum rota	61
Şekil 6.3. A^3 ile wi29 GSP uygulaması için ulaşılan optimum rota.....	62

Şekil 6.4. A^3 ile att48 GSP uygulaması için ulaşılan optimum rota.....	63
Şekil 6.5. A^3 ile eil51 GSP uygulaması için ulaşılan optimum rota.....	64
Şekil 6.6. A^3 ile berlin52 GSP uygulaması için ulaşılan optimum rota	65
Şekil 6.7. A^3 ile st70 GSP uygulaması için ulaşılan optimum rota	66
Şekil 6.8. A^3 ile pr76 GSP uygulaması için ulaşılan optimum rota	67
Şekil 6.9. A^3 ve karşılaştırılan diğer algoritmaların GSP örnekleri için optimum çözümünden yüzdesel sapmaları.....	71
Şekil 6.10. A^3 ve karşılaştırılan diğer algoritmaların GSP örnekleri için ortalama yüzdesel sapmaları.....	71
Şekil 6.11. A^3 ile grup asansör kontrol optimizasyonunda her atomun amaç fonksiyonu değeri	76
Şekil 6.12. A^3 ile grup asansör kontrol optimizasyonunda atom kümesinin amaç fonksiyonu ortalaması.....	77

ÇİZELGELER LİSTESİ

Çizelge 4.1.	GSP için A ³ uygulamasında kullanılan parametre değerleri.....	34
Çizelge 4.2.	Kovalent bağ operatörü uygulanırken oluşan 1. durum.....	39
Çizelge 4.3.	Kovalent bağ operatörü uygulanırken oluşan 2. durum.....	39
Çizelge 4.4.	Kovalent bağ operatörü uygulanırken oluşan 3. durum.....	40
Çizelge 4.5.	Kovalent bağ operatörü uygulanırken oluşan 4. Durum.....	40
Çizelge 4.6.	Kovalent bağ operatörü uygulanırken oluşan 5. Durum.....	40
Çizelge 4.7.	BKİ'ye göre şişmanlık durumunun sınıflandırılması.....	44
Çizelge 4.8.	Fiziksel aktivite yoğunluğuna göre EER1'in hesaplanması	46
Çizelge 4.9.	Hamile bayanların hamilelik dönemine göre EER2'nin hesaplanması	46
Çizelge 4.10.	Çocuk emziren bayanların doğum sonrası dönemine göre EER2'nin hesaplanması.....	46
Çizelge 4.11.	A ³ ile beslenme optimizasyonu için belirlenen parametre değerleri.....	48
Çizelge 4.12.	Binanın ve asansörün fiziksel özellikleri	56
Çizelge 4.13.	A ³ ile grup asansör kontrol optimizasyonu için parametre değerleri.....	57
Çizelge 5.1.	A ³ 'ün uygulandığı GSP'ler için performans analizi	59
Çizelge 5.2.	A ³ ile burma14 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası.....	60
Çizelge 5.3.	A ³ ile bayg29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	61
Çizelge 5.4.	A ³ ile wi29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	62
Çizelge 5.5.	A ³ ile att48 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	63
Çizelge 5.6.	A ³ ile eil51 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	64
Çizelge 5.7.	A ³ ile berlin52 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	65
Çizelge 5.8.	A ³ ile st70 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	66
Çizelge 5.9.	A ³ ile pr76 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası	67
Çizelge 5.10.	att48 problemi için A ³ ve diğer sezgisel tekniklerin performans değerlendirmesi.....	68
Çizelge 5.11.	eil51 problemi için A ³ ve diğer sezgisel tekniklerin performans değerlendirmesi.....	68

Çizelge 5.12. berlin52 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi.....	69
Çizelge 5.13. st70 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi.....	69
Çizelge 5.14. pr76 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi.....	70
Çizelge 5.15. GSP örnekleri için A^3 'ün çalışma zamanı performansı	72
Çizelge 5.16. Örnek bireylerin fiziksel, fizyolojik ve sosyolojik özellikleri	73
Çizelge 5.17. Örnek bireylerin BMR, EER1 ve EER2 değerleri	73
Çizelge 5.18. A^3 ile beslenme optimizasyonu uygulamasının 1. birey için elde ettiği besin çizelgesi	74
Çizelge 5.19. A^3 ile beslenme optimizasyonu uygulamasının 2. birey için elde ettiği besin çizelgesi	74
Çizelge 5.20. Karşılaştırılan tekniklere göre asansörlerin cevaplayacakları kat çağrıları ve cevaplama sıraları	78
Çizelge 5.21. Tüm kat çağrıları için yolcuların asansör hizmetini bekleme süreleri toplamının karşılaştırılması.....	78

SİMGELER VE KISALTMALAR

α	İyonik oran
β	Kovalent oran
η	(0-1) arasında rastgele üretilen katsayı
c_p	Asansör kabininin başlangıçta bulunduğu kat
c_t	Asansör kabininin hedef katı
t_T	Kabinin katlar arasındaki ortalama seyahat süresi
t_D	Kabin kapısının ortalama açılış/kapanış süresi
α_1, α_2	Ağırlık katsayıları
A^3	Yapay Atom Algoritması
AYBS	Adaptif Yapay Bağışık Sistemler
BKİ	Beden-kitle indeksi
BMR	Bazal metabolizma oranı
DSÖ	Dünya Sağlık Örgütü
EER1	Fiziksel aktivite yoğunluğuna göre tahmin edilen enerji gereksinimi
EER2	Bayanların hamilelik/emzirme durumuna göre tahmin edilen enerji gereksinimi
GA	Genetik Algoritma
GSP	Gezgin Satıcı Problemi
İA	İyonik Alan
KA	Kovalent Alan
KKO	Karınca Koloni Optimizasyonu
PSO	Parçacık Sürü Optimizasyonu
TAA	Tabu Arama Algoritması

1. GİRİŞ

1.1. Tez Çalışmasının Amacı ve Kapsamı

Ayrık problemler, parametreleri sınırlı sayıda ve sürekli olmayan değerler alan, optimizasyonun bu ayrık parametre değerlerinin seçilmesi, sıralanması ya da gruplanması şeklinde yapıldığı, tanımlanması kolay ancak çözümü zor problemlerdir. Optimizasyon algoritmasının başarısını gösteren önemli kriterlerden biri de, ayrık optimizasyon problemlerinde gösterdiği başarıdır. Bu tez çalışmasında, yakın zamanda geliştirilen meta-sezgisel bir yaklaşım olan Yapay Atom Algoritması (Artificial Atom Algorithm) (A^3)'nın kıyaslama (benchmark) problemlerindeki başarısı göz önünde bulundurularak, algoritmanın farklı yapıdaki ayrık problemlere uyarlanması ve aynı performansı ayrık optimizasyon problemlerinde gösterip göstermeyeceğinin test edilmesi ve değerlendirilmesi amaçlanmıştır. Bu amaçla, algoritmanın uygulanacağı üç ayrık problem belirlenmiştir.

Birincisi, Gezgin satıcı problemi (GSP)'dir. Literatürde sıklıkla kullanılan bu problem, NP-zor yapıda olup, şehir sayısı arttıkça problemin matematiksel yollarla çözümü, hesaplama zamanındaki artış nedeniyle imkânsız hale gelmektedir. Bu nedenle çözümünde sezgisel ya da meta-sezgisel yöntemler tercih edilmektedir.

İkinci problem, beslenme optimizasyonu problemi olup, bireylerin fiziksel, fizyolojik ve sosyolojik özelliklerine uygun beslenme çizelgesinin elde edilmesini, yani bireyin ihtiyaç duyduğu günlük kalori miktarına uygun besinlerden oluşan optimum beslenme çizelgesinin kullanıcıya sunulmasını hedeflemektedir. Bu da yine ayrık bir optimizasyon problemi olarak karşımıza çıkmaktadır.

Son olarak, bina içerisinde aynı anda çalışan birden çok asansörün mevcut olduğu durumlarda, bu asansörlerin bir kontrol mekanizmasını kullanarak optimum çağrı cevaplama süresini yakalaması amaçlanır. Bu da grup asansör kontrol optimizasyonu problemi olarak tanımlanmaktadır.

Bu tez çalışmasında, A^3 'ün performans analizi için GSP, Beslenme optimizasyonu ve Grup asansör kontrol optimizasyonu problemleri kullanılmıştır. Çalışmada, A^3 'ün sürekli optimizasyon problemleri için tasarlanan yapısının, ayrık optimizasyon problemlerine uyarlanabilmesi için geliştirilen teknikler ayrıntılı olarak sunulmuştur. Ayrıca, gerçekleştirilen uygulamalar neticesinde elde edilen sonuçlar, literatürde aynı alanda yapılan çalışmalarla karşılaştırmalı olarak verilmiştir.

1.2. Literatür Taraması ve Değerlendirmesi

Literatürde GSP'nin çözümünde farklı meta-sezgisel yöntemler kullanılmıştır. Bunlardan bazıları; Tabu arama algoritması (TAA) (Malek vd., 1989), Karınca koloni optimizasyonu (KKO) (Dorigo ve Gambardella, 1997; Ellabib vd., 2007), Simule edilmiş tavlama benzetimi (Lo ve Hus, 1998), Parçacık sürü optimizasyonu (PSO) (Onwubolu ve Clerc, 2004), Genetik algoritmalar (GA) (Hua and Huang 2006), Arı koloni algoritması (Wong vd., 2008), Nöral ağlar (Masutti and de Castro 2009), Evrimsel algoritmalar (Shen ve Zhang, 2011), Memetik algoritması (Acampora vd., 2011), Ateş böceği algoritması (Suyanto, 2011), Guguk kuşu algoritması (Ouaarab vd., 2014; Demirtaş ve Zengin, 2016) ve Harmoni arama algoritması (Tseng, S.P., 2016) gibi yöntemlerdir. Bu algoritmalarından bazıları ikinci bölümde daha ayrıntılı olarak anlatılacaktır. Bahsi geçen algoritmaların çeşitli GSP'lerde başarılı sonuçlar verdikleri görülmektedir.

Literatürde beslenme optimizasyonu alanında da birçok çalışma bulunmaktadır. Bu çalışmalarda genellikle Doğrusal programlama ve GA tekniklerini kullanılmıştır. Yapılan çalışmalarda, beslenme problemi farklı parametreler için optimize edilmiştir. Örneğin; Stigler tarafından yapılan çalışmada, Doğrusal programlama ile ortalama bir bireyin minimum maliyet ile beslenme ihtiyacını karşılaması hedeflenmiştir (Stigler, 1939). Başka bir çalışmada ise, yine Doğrusal programlama kullanılarak hem maliyet optimizasyonu yapılırken hem de besin çizelgesinin lezzet düzeyi arttırılmaya çalışılmıştır (Smith, 1959). Doğrusal programlama kullanılarak gerçekleştirilen bir diğer beslenme optimizasyonu uygulamasında ise, bireyin enerji gereksinimi değerlendirilmeden sabah, öğle, akşam öğünleri için maliyet optimizasyonu yapılması amaçlanmıştır (Balintfy, 1964). Doğrusal programlama ve hedef programlama teknikleri kullanılarak gerçekleştirilen uygulamada ise; besin öğelerini, içerdiği kalori, protein, vitamin ve mineraller bakımından ortalama bir bireyin ihtiyaç duyduğu oranları sağlayacak şekilde optimize edecek bir beslenme çizelgesi oluşturulmaya çalışılmıştır. Ancak çalışmada, IBM LPS/1130 paket programını kullanarak optimizasyon yapılmıştır (Anderson ve Earle, 1983). Yapılan başka bir çalışma ise, QSBWIN paket programını kullanarak, Doğrusal programlama tekniği ile düşük gelirli ve fiziksel aktivite yoğunluğu olan bir işte çalışan genç bir birey için haftalık optimum beslenme çizelgesinin oluşturulmasıdır (Kılınç, 2007). Yapılan başka bir çalışma ise, çok amaçlı beslenme optimizasyonu probleminin Kuantum genetik algoritma ile çözülmesidir. Çalışma aynı problemin GA çözümüne göre, işlem zamanı performansı arttırmıştır (Lv, 2009). Yapılan bir diğer çalışma ise, GA kullanılarak, sadece kahvaltı öğünü için ortalama bir bireyin ihtiyaçlarına uygun besin bileşenlerine sahip ve maliyet bakımından optimum bir beslenme çizelgesi hazırlanmasıdır (Gümüştakin ve Şenel, 2013). Yapılan çalışmaların farklı beslenme optimizasyonu problemleri için başarılı sonuçlar verdikleri görülmektedir.

Literatürde grup asansör kontrol optimizasyonu probleminin çözümünde, farklı yapay zeka optimizasyon tekniklerinin kullanıldığı görülmektedir. Bunlar arasında GA (Gudwin ve Gomide, 1994; Cortes vd., 2004; Bolat, 2006), Bulanık mantık teknikleri (Kim vd., 1998), Takviyeli öğrenme teknikleri (Crites ve Barto, 1998), DNA hesaplama yöntemi (Muhammad vd., 2005), Parçacık sürü benzetimli tavlama optimizasyonu (Fei vd., 2010), KKO (Zhang vd., 2010) ve Adaptif yapay bağışık sistemler (AYBS) (Baygın ve Karaköse, 2011) bulunmaktadır. Bu yöntemlerin farklı grup asansör kontrol optimizasyonu problemleri için başarılı sonuçlar verdikleri görülmektedir.

1.3. Tez Çalışmasının Organizasyonu

Tez çalışması aşağıda verilen şekilde organize edilmiştir:

İkinci bölümde optimizasyon kavramı, optimizasyon problemleri ve optimizasyon algoritmaları üzerinde durulmaktadır. Özellikle meta-sezgisel optimizasyon algoritmaları ayrıntılı olarak anlatılmaktadır. Literatürde bu sınıfta bulunan çeşitli meta-sezgisel yöntemler açıklamalı olarak verilmektedir.

Üçüncü bölümde, tez çalışmasında kullanılan meta-sezgisel algoritma olan Yapay Atom Algoritması (A^3)'nın çalışma mantığı, kavramları, operatörleri, operatörlerine ait sözde kodları ve akış diyagramları ayrıntılı şekilde verilmektedir. Ayrıca diğer meta-sezgisel yöntemlerden farkları üzerinde durulmaktadır.

Dördüncü bölümde, A^3 yönteminin ayrık problemlerin çözümünde kullanılabilmesi için, sürekli optimizasyon problemleri için tasarlanan şekli üzerinde yapılan değişiklikler anlatılarak, tezin literatüre sağladığı katkılar açıklanmaktadır.

Beşinci bölümde, A^3 'ün uygulandığı ayrık optimizasyon problemleri olan gezgin satıcı, beslenme optimizasyonu ve grup asansör kontrol optimizasyonu problemleri tanımlanarak, bu problemler için belirlenen amaç fonksiyonları verilmektedir. Ek olarak, beslenme optimizasyonu problemi için gerekli bazı öncül hesaplamalar gösterilmektedir. Ayrıca, tüm problemler için gerçekleştirilecek uygulamaların parametre ayarlamaları belirtilmekte ve A^3 'ün bu problemlere uygulanması aşamasında geliştirilen çeşitli teknikler ve algoritma içerisinde yapılan değişiklikler anlatılmaktadır.

Altıncı bölümde, A^3 ile gezgin satıcı, beslenme optimizasyonu ve grup asansör kontrol optimizasyonu uygulamalarının elde ettiği deneysel sonuçlar şekiller ve tablolarla karşılaştırmalı olarak gösterilmektedir. Uygulamaların çalışma zamanı performansları da bu bölümde değerlendirilmektedir.

Son bölümde ise, A^3 yönteminin ayrık problemlere uygulanmasındaki performansının genel olarak değerlendirildiği ve gelecekte yapılabilecek çalışmaların anlatıldığı sonuç kısmı yer almaktadır.

2. META-SEZGİSEL YÖNTEMLER

2.1. Optimizasyon Problemleri

Optimum kelime anlamı olarak, bir sorunun en ideal çözümüdür. Optimizasyon ise, bir sistem veya tasarım için en iyi sonucu bulmak amacı ile yapılan işlemler bütünüdür. Belirli kısıt şartlarını sağlayarak karar değişkenlerinin değerlerini hesaplamaya çalışan problemlere optimizasyon problemleri denir. Bir optimizasyon probleminde temel amaç; faydayı veya kârı maksimize etmek, hatayı, maliyeti ya da harcanan enerjiyi minimize etmektir (Gass, 2000).

Optimizasyon problemlerinde, problemin çözümü için karar değişkenleri arasındaki bağıntıları kullanan fonksiyonlara amaç fonksiyonu denir. Amaç fonksiyonu, problemin performans ölçütüdür. Her optimizasyon probleminin bir amaç fonksiyonu vardır. Denklem (2.1)'de bir optimizasyon probleminin matematiksel gösterimi verilmiştir.

$$\begin{aligned} \min/\max f(x), \quad x = [x_1, x_2, x_3, \dots, x_N]^T \in R^N \\ x_i^l \leq x_i \leq x_i^u, i = 1, 2, 3, \dots, N \end{aligned} \quad (2.1)$$

Burada, problemin karar değişkenleri x ile gösterilmekte, problemin amaç fonksiyonu ise $f(x)$ ile belirtilmektedir. x_i^l ve x_i^u sırasıyla, x_i karar değişkeninin alt ve üst sınırlarını belirtmektedir. Problemin çözümü, amaç fonksiyonunu minimize ya da maksimize eden karar değişkenlerinin tespit edilmesiyle gerçekleştirilir (Wu, 2010).

2.1.1. Optimizasyon problemlerinin sınıflandırılması

Optimizasyon problemleri, içerdikleri karar değişkenlerinin, amaç fonksiyonunun ve problem sınırlayıcılarının özelliklerine göre çeşitli şekillerde sınıflandırılmaktadırlar.

a) Optimizasyon problemlerinin minimize yada maksimize edilecek bir amaç fonksiyonu olmak zorundadır. Buna karşın sınırlayıcıları olmayabilir. Bu sebeple optimizasyon problemleri, sınırlayıcılı ve sınırlayıcısız optimizasyon problemleri olarak sınıflandırılmaktadırlar. Sınırlayıcısız optimizasyon problemleri, karar değişkenleri için herhangi bir sınırlayıcı şart bulunmayan problemlerdir. Genel olarak Denklem (2.2)'deki gibi ifade edilirler.

$$\min/\max f(x), \quad x = [x_1, x_2, x_3, \dots, x_N]^T \in R^N \quad (2.2)$$

Burada, $f(x)$ amaç fonksiyonunu, x ise karar değişkenlerini belirten vektörü gösterir. Ek olarak, $x \in S$ ve $S \subseteq \mathbb{R}^N$ 'dir. S , arama uzayını ifade eder. S , N boyutludur ve parametrik kısıtlarla sınırlandırılmıştır.

Sınırlayıcı optimizasyon problemleri ise, Denklem (2.3)'teki gibi ifade edilirler. Burada x ve y karar değişkenleri ve $f(x, y)$ amaç fonksiyonu olmak üzere;

$$\begin{aligned} \min/\max z &= f(x, y), \quad x \in \mathbb{R}^N, \quad y \in \{0, 1, 2, \dots, m\} \\ g(x, y) &= 0 \\ h(x, y) &\leq 0 \end{aligned} \quad (2.3)$$

dır. Burada $g(x, y)$, eşitlik sınırlamasına sahip bir kısıtlayıcıyı belirtirken; $h(x, y)$, eşitsizlik sınırlamasına sahip bir kısıtlayıcıyı belirtir. Ayrıca, karar değişkenleri de iki şekilde ifade edilmiştir. Herhangi bir reel değer alabilen sürekli değişkenler x ve herhangi bir tamsayı değer alabilen ayrık değişkenler ise y ile belirtilmiştir (Balas, 1965).

b) Optimizasyon problemleri, tek bir amaca yönelik olabilecekleri gibi, birden çok amaca yönelik de olabilirler. Tek amaca yönelik problemlerde, sınırlayıcı sayısından bağımsız olarak, problemin tek bir amaç fonksiyonu olur. Çok amaçlı optimizasyon problemlerinde ise, gerçekleşmesi gereken birden çok amaç fonksiyonu bulunmaktadır.

c) Optimizasyon problemlerindeki bir diğer sınıflandırma, lineer ve non-lineer şeklinde yapılmaktadır. Eğer probleme ait amaç fonksiyonu ve sınırlayıcılar doğrusalsa, lineer optimizasyon problemi olarak sınıflandırılır. Eğer probleme ait amaç fonksiyonu ve sınırlayıcılar doğrusal değilse, lineer olmayan (non-lineer) optimizasyon problemi olarak sınıflandırılır.

d) Başka bir sınıflandırma ise, problemin karar değişkenlerinin sürekli ya da ayrık olmasına göre yapılmaktadır. Eğer karar değişkenleri (parametreler), sürekli değerler alıyorsa, bu tür problemlere sürekli optimizasyon problemleri denir. Eğer karar değişkenleri ayrık değerler alıyorsa ve bu ayrık değerlerin düzenlenmesi, gruplanması, seçilmesi veya sıralanması ile problemin optimizasyonu yapılıyorsa, bu tür problemlere ayrık optimizasyon problemleri denilmektedir.

2.2. Optimizasyon Algoritmaları

Optimizasyon algoritmaları, karar alma süreçlerini hızlandırmak ve karar kalitesini yükseltmek amacıyla, matematiksel ya da sezgisel teknikler kullanarak, minimum hata,

minimum maliyet, maksimum kâr ve maksimum fayda ile en iyi sonucu bulmaya çalışan algoritmalar. Optimizasyon algoritmaları karar sürecinde kullandıkları teknikler bakımından üç şekilde sınıflandırılırlar. Bunlar; matematiksel optimizasyon algoritmaları, sezgisel optimizasyon algoritmaları ve meta-sezgisel optimizasyon algoritmalarıdır. Bu tez çalışmasında, meta-sezgisel optimizasyon algoritmaları üzerinde durulacaktır.

2.2.1 Matematiksel optimizasyon algoritmaları

Matematiksel optimizasyon algoritmaları, probleminin çözümünü bir matematiksel modele dayandırır ve optimizasyon probleminin yapısına uygun olarak çözüm üretmeye çalışırlar. Yani matematiksel optimizasyon algoritmalarında, karar verme sürecinde, problemin karar değişkenleri arasındaki bağıntılar belirlenir. Karar değişkenleri arasındaki bağıntılara göre oluşturulan denklemlerin çözümü, problemin çözümünü verir.

2.2.2. Sezgisel optimizasyon algoritmaları

Sezgisel optimizasyon algoritmaları, doğadan esinlenerek üretilen ve sezgilere dayalı olan operatörlerle işlem görürler. Matematiksel yöntemlerden farklı olarak, probleme özgü çözüm geliştirmezler, önceden belirlenen çözüm tekniklerini problem üzerine uygulamaya çalışırlar. Sağ el kuralına dayalı algoritmalar. Sezgisel algoritmalar her çalıştırıldıklarında aynı performansı göstermeyi garanti etmezler. Aynı zamanda, optimum çözüme ulaşmayı da garanti etmezler. Ancak birçok optimizasyon problemi için, matematiksel yöntemlere göre daha kolay ve hızlı şekilde optimuma yakın çözümler elde ederler.

Sezgisel algoritmalara gerek duyulmasının başlıca sebepleri şunlardır:

- a) Optimizasyon probleminin yapısı, karar değişkenleri arasındaki bağıntılarla kesin çözümün elde edilmesine olanak tanımayabilir.
- b) Matematiksel denklemlerde genellikle problemin en zor bölümü ihmal edilir. Yani probleme uygun çözüm tanımlanırken, gerçek hayattaki haliyle tam olarak örtüşmediği görülür.
- c) Matematiksel model oluşturulurken, problemin karar değişkenlerinin seçilmesinde yapılacak bir hata, sezgisel teknikle elde edilen optimum olmayan çözüme göre daha büyük bir hataya sebep olabilir.
- d) Sezgisel algoritmaların anlaşılabilirliği ve uygulanabilirliği daha yüksektir.
- e) Bazı problemlerin matematiksel yöntemlerle çözümü tanımlanabilse bile, çözümü için harcanacak zaman ve enerji nedeniyle sezgisel tekniklerin kullanılması avantaj sağlar. Çünkü sezgisel teknikler, genellikle daha hızlı çalışma performansına sahiptir.

- f) Sezgisel yöntemler, problemin bir bölümünün çözümünde kullanılabilirler.
- g) Problemin çözümünü öğrenme amaçlı olarak da kullanılabilirler.

Sezgisel optimizasyon algoritmalarının değerlendirilmesinde; çözüm kalitesi, hesaplama zamanı performansı, farklı problemlere uygulanabilirliği, operatörlerinin basit ve esnek yapısı, analiz edilebilirliği ve diğer hesaplama teknikleriyle etkileşimli kullanılabilirliği önem taşımaktadır (Karaboğa, 2017).

2.2.3. Meta-sezgisel optimizasyon algoritmaları

Meta-sezgisel optimizasyon algoritmaları, arama ve optimizasyon problemlerinin çözümünde kullanılan yöntemlerdir. Sezgisel algoritmalar gibi, doğadan esinlenmiş, sezgilere dayalı, basit, uygulaması kolay operatörleri bulunmaktadır. Temel olarak komşu çözümleri araştırarak iteratif olarak arama yapan ve bu yolla optimum çözümü elde etmeye çalışan, ancak elde edeceğini garanti etmeyen algoritmalarlardır. Sıklıkla kullanılan bu algoritmalar, güçlü problem çözüm tekniklerine sahiptir. Buna karşın hesaplama karmaşıklığı ve hesaplama zamanı düşüktür. Bu algoritmaların, farklı amaç fonksiyonu ve kısıtları olan problemlere uygulanması da oldukça kolaydır. Sezgisel yöntemlerden farkı, birden çok sezgisel teknik içerisinde en kullanışlı olanlarını seçerek, çözüm uzayında arama yapmaları ve elde edilen sonuçların doğruluğunun kanıtlanabilir olup olmadığını önemsememeleridir.

Meta-sezgisel algoritmaların birçoğu popülasyon yapısına sahiptir. Yani algoritmanın yürütülmesine, rastgele oluşturulan karar değişkeni değerleriyle başlanır. Karar değişkenlerine rastgele değerler atarken, problem yapısı ve değişken sınırları dikkate alınır. Popülasyondaki her birey, bir aday çözümdür. Bu rastgele belirlenen çözümler amaç fonksiyonu ya da uygunluk fonksiyonuna göre değerlendirilir ve gerekirse optimizasyon algoritmasının operatörleri kullanılarak yenilenir. Algoritmalar, belirlenen iterasyon sayısına kadar ya da belli bir hata tolerans değerine ulaşıncaya kadar sürdürülür.

Meta-sezgisel optimizasyon algoritmaları, farklı şekillerde sınıflandırılabilirler. Bunlar; mevcut bilginin arama sırasında kullanılıp kullanılmaması bakımından, doğadan esinlendikleri faktörler bakımından, aramada kullanılan çözüm sayısı bakımından, kullandıkları amaç fonksiyonu ya da uygunluk fonksiyonu bakımından, kullandıkları komşuluk bakımından sınıflandırmadır. Ancak en temel sınıflandırma, doğadan esinlendikleri faktör bakımından yapılmaktadır. Buna göre; fizik tabanlı, kimya tabanlı, biyoloji tabanlı, matematik tabanlı, sürü tabanlı, sosyal tabanlı, spor tabanlı, müzik tabanlı ve melez şeklinde kategorilere ayrılabilirler (Canayaz, 2015).

a) Fizik tabanlı algoritmalar, fiziğin temel prensiplerinden esinlenerek geliştirilmişlerdir. Bu alanda literatürde rastlanılan algoritmalarından bazıları; Benzetilmiş tavlama algoritması (Kirkpatrick vd., 1983), Elektromanyetizma algoritması (Birbil ve Fang, 2003), Büyük patlama-büyük büzülme algoritması (Erol ve Eksin, 2006), Merkezi kuvvet optimizasyonu (Formato, 2007; Green vd., 2012), Yerçekimsel arama algoritması (Rashedi vd., 2009), Yapay fizik optimizasyon algoritması (Xie vd., 2009), Yüklü sistem arama algoritması (Kaveh ve Talatahari, 2010)'dır.

Elektromanyetizma algoritması, fizik biliminde elektrik yüklü parçacıklar arasında meydana gelen itme ve çekme prensibinden esinlenerek tasarlanmıştır. Her parçacık çözüm uzayında bir konum belirtir. Ayrıca her parçacığın bir yükü bulunmaktadır. Bu yük amaç fonksiyonuyla hesaplanır. Parçacıklar, yüklerine bağlı olarak birbirine itme ya da çekme şeklinde bir vektörel kuvvet uygularlar. Tüm parçacıkların vektörel kuvvetleri hesaplandıktan sonra, hesaplanan bu değer kullanılarak bileşke kuvvet değeri belirlenir. Elektromanyetizma algoritması, çözüm uzayında bulunan tüm parçacıkların bu bileşke kuvvet yönünde hareket ettirilmesiyle optimum çözüme ulaşmaya çalışır (Birbil ve Fang, 2003; Gürsu, 2014; Özdağ ve Karcı, 2015).

Büyük patlama-büyük büzülme algoritması, Big Bang teorisine dayanmakta olup, evren oluşurken meydana geldiği düşünülen olayların birleşiminden esinlenerek tasarlanmıştır. Algoritma, çözümlerin rastgele dağıtımı ve rastgele toplanarak bir araya getirilmesiyle, patlama ve büzülme süreçlerini taklit eder. Bir araya gelen parçacıkların kütle merkezi hesaplanır. Yeni patlama ve büzölmeler ile algoritma sürdürölür. Her defasında yeni kütle merkezi hesaplanır. Kütle merkezi artık değışmediğinde algoritma sonlandırılır. Optimum durum, bu son kütle merkezi olarak kabul edilir (Erol ve Eksin, 2006).

Yerçekimsel arama algoritması, Newton'un yerçekimi prensiplerinden yola çıkarak geliştirilmiştir. Algoritmaya göre, her cisim kütlesi oranında diğler cisimlere bir çekim kuvveti uygulamaktadır. Kütlesi hafif olan cisimler, ağır olanlara doğru hareket ederler. Yani, çözüm uzayında amaç fonksiyonu değeri düşük olan elemanlar, yüksek olanlara yaklaşırlar ve sonuçta hareket bitince algoritma sonlanır. Algoritma sonunda, kütlesi en ağır olan cisim, optimum çözümü gösterir (Rashedi vd., 2009; Can, 2014).

Yüklü sistem arama algoritması, mekanik ve elektrostatik yasalarına dayanmaktadır. Cisimlerin içediğı yüklü parçacıklar birbirlerine etki etmektedirler. Bu etki aralarındaki mesafenin karesiyle ters orantılı olarak uygunluk fonksiyonu ile hesaplanır. Yüklü parçacıklar, problemin aday çözümleridir. Birbirlerine yaptıkları etkilere bağlı olarak, algoritma sonunda global çözüme ulaşılmaya çalışılmaktadır (Kaveh ve Talatahari, 2010).

b) Kimya tabanlı algoritmalar, kimya bilimindeki temel prensiplerden esinlenerek geliştirilen algoritmalarlardır. Kimya tabanlı algoritmalarından bazıları; Yapay atom algoritması (A^3) (Karcı, 2012) ve Yapay kimyasal tepkime algoritması (Alataş, 2012)'dır.

A³, bu tez çalışmasında kullanıldığı için 3. bölümde detaylı olarak anlatılacaktır. Kısaca, maddenin yapıtaşı olan atomun bileşik oluşturma süreçlerinden esinlenerek tasarlanmıştır. Kovalent bağ ve iyonik bağ adında iki operatörü bulunmaktadır. Diğer meta-sezgisel yöntemlerden farklı olarak, parametre değerlerinin çözüm üzerindeki etkilerini de değerlendirir. Bu şekilde optimum çözüme erişmeye çalışır (Karcı, 2012).

Yapay kimyasal tepkime algoritması ise, kimyasal tepkimeleri model olarak geliştirilmiştir. Buna göre ilk olarak tepkimenin kaç moleküllü olacağına karar verilir. Bu karara göre, tepkimeye girecek moleküller belirlenir. Moleküller problem türüne bağlı olarak ikili kodlama ya da metin kodlama şeklinde kodlanırlar. Tepkimeye giren moleküller bazı reaksiyonlara maruz kalır. Buna göre, kimyasal kararlılık testi yapılır. Bu test sonucunda iyi değerler alan moleküller korunurken, kötü değerler alanlar atılır. Algoritma sonunda en iyi moleküller, optimum sonuç olarak belirlenirler. Bu algoritmanın diğer yöntemlere göre avantajı az sayıda parametre içermesidir (Alataş, 2012).

c) Biyoloji tabanlı algoritmalar, genellikle doğada yaşayan canlı türlerinin yapılarının ya da canlı türlerine ait davranışların taklit edilmesiyle ortaya çıkmıştır. Bu algoritmalar içinde en çok bilineni GA'dır (Goldberg, 1989). Biyoloji tabanlı algoritmalarından diğerleri; Yapay bağışık sistemler (Dasgupta, 1993), Diferansiyel gelişim algoritması (Storn ve Price, 1997), Bakteriyel besin arama algoritması (Passino, 2002), Fidan gelişim algoritması (Karcı, 2007), Biyo-coğrafya temelli optimizasyon algoritması (Simon, 2008), (Baygın ve Karaköse, 2011)'dir.

GA, evrim teorisinden esinlenerek tasarlanan bir yöntemdir. Problem içinde, her parametre bir gen ile belirtilir. Genler, kromozomları oluşturarak aday çözümleri gösterirler. Kromozomlardan oluşan çözüm uzayı ise, popülasyon adını alır. Buna göre kromozomlar, uygunluk değerlerine göre işlem görürler. Popülasyona dört farklı operatör uygulanır. Bunlar; kodlama, seçim, çaprazlama ve mutasyondur. Gen olarak bilinen karar değişkenlerinin, değerler dizisi oluşturacak şekilde birleştirilmesine kodlama denir. Kodlama; ikili kodlama, permütasyon kodlama ve değer kodlama olarak farklı yöntemlerle yapılabilir. Seçim, iyi nesillerin çoğalmasını sağlamak amacıyla yapılır. Çeşitli seçim yöntemleri mevcuttur. Bunlar; rulet tekeri, turnuva yöntemi, sıralamalı seçim gibi tekniklerdir. Popülasyon içinden seçilen iki kromozoma çaprazlama ve mutasyon operatörleri uygulanır. Çaprazlama, seçilen iki kromozom arasında gen alış-verişi yapılarak, iki yeni kromozomun elde edilmesidir. Mutasyon ise, seçilen kromozomlarda mutasyon oranında gende değişiklik yapılmasıdır. Yapılan değişiklikler neticesinde, algoritma sonunda elde edilen son nesil, optimum nesil olarak sunulur (Goldberg, 1989).

Yapay Bağışık Sistemler, canlıların doğal bağışıklık sistemlerinden esinlenerek geliştirilmiş meta-sezgisel bir algoritmadır. İki temel yöntem içermektedir. Bunlar; negatif seçim ve klonal seçim algoritmalarıdır. Negatif seçimde, öncelikle bir aday dedektör kümesi

belirlenir. Bu aday küme, öz küme ile eşleştirilmeye çalışılır ve eşleşenler asıl dedektör kümesine alınır. Ardından test verileri ile sistemin performansı test edilir. Bu yöntem çoğunlukla virüs tanıma sistemlerinde tercih edilir. Klonal seçim algoritmasında ise, antikor denilen aday çözümlerle başlangıç kümesi oluşturulur. Antikorların benzerlikleri hesaplanır. Belirli sayıda en yüksek benzerliğe sahip antikor alınarak, benzerlik oranıyla doğru orantılı olarak kopyalanır. Ardından benzerlik oranıyla ters orantılı olarak, mutasyon operatörü uygulanır. Mutasyona uğrayan antikorların benzerlik oranları tekrar hesaplanır. Belirli sayıda yüksek benzerliğe sahip antikor çözüm kümesine alınarak algoritma tekrarlanır. Klonal seçim algoritması ise, genellikle optimizasyon problemlerinde tercih edilir (Dasgupta, 1993; Baygın ve Karaköse, 2011).

Diferansiyel gelişim algoritması, genetik algoritmaya benzeyen, sürekli veriler için etkili, popülasyon temelli bir çözüm tekniğidir. Lig Şampiyonası, Politik rekabet, Kedi sürüsü optimizasyon algoritması gibi birçok teknik Diferansiyel gelişim algoritmasına dayandırılmaktadır. Algoritma operatörleri, popülasyon içerisindeki her kromozoma tek tek uygulanır. İlk olarak, sırası gelen kromozom ile rastgele seçilen iki kromozoma çaprazlama operatörü uygulanır. Ardından bir kromozom daha seçilir ve mutasyon operatörü uygulanır. Elde edilen yeni birey, mevcut bireyle karşılaştırılır. Uygunluk değeri iyi olan, popülasyona alınır. Algoritma sonunda uygunluk değeri en iyi kromozom, optimum çözüm olarak belirlenir (Storn ve Price, 1997; Kesintürk, 2006, Karcı, 2017).

Bakteriyel besin arama algoritması; kemotaksis, üreme ve eliminasyon süreçlerinden oluşan bakterilerin besin arama yöntemini örnek alarak tasarlanan bir algoritmadır. Kemotaksis, bakterilerin bir kimyasal maddeye doğru ya da bir kimyasal maddeden uzağa doğru hareketini belirtir. Buna göre, kemotaksis operatörüyle bakterinin pozisyonunda bir değişiklik olup olmayacağı, olacaksa hangi yöne kaç birim değişiklik olacağı belirlenir. Üreme, ilk olarak belirli sayıda bakterinin popülasyondan atılmasıyla başlar. Ardından geriye kalan bakteriler, amaç fonksiyon değerlerine göre sıralanır. Popülasyonun ilk yarısı korunur ve kopyalanır. İkinci yarısı ise, popülasyondan atılır. Eliminasyon ise, önceden belirlenen bir sayıyla, her bakteri için rastgele üretilen bir sayının karşılaştırılması sonucunda, bakterilerin popülasyondan elenmesi işlemidir. Algoritma, tüm bu operatörlerin belirli sayıda uygulanmasıyla optimum sonuca ulaşmaya çalışır (Passino, 2002; Hezer ve Kara, 2013).

Fidan gelişim algoritması, biyoloji tabanlı bir algoritma olup, fidanların gelişim sürecini modelleyerek oluşturulmuştur. Algoritmada aday çözümler fidanlarla belirtilir. Fidanların gelişim süreçleri, algoritmanın operatörleri olarak belirlenmiştir. Bu operatörler; fidan ekimi, eşleştirme, dallanma ve aşılama operatörleridir. Fidan ekimi ile düzgün dağılımlı çözüm kümesinin oluşturulması sağlanır. Eşleştirme ile fidanlar arasında genetik bilgi alış-verişi sağlanır (Karcı, 2007). Dallanma ve aşılama operatörleri ise, çözümden fazla

uzaklaşmaya sebep oldukları için sonradan birleştirilerek karıştır-geliştir adını almıştır. Son şekliyle iki fidan girdi olarak alınmış ve fidanlar arasındaki uzaklığa bağlı olarak iki yeni fidan elde edilmiştir. Buna göre, algoritma sonunda en iyi değerleri veren fidan, optimum çözüm olarak sunulur (Demir vd., 2011).

Biyo-coğrafya temelli optimizasyon algoritması ise, canlıların hayatta kalmaları için en uygun yaşam alanını belirleme esasına dayanmaktadır. En uygun yaşamsal şartlar, yaşam alanı uygunluk indeksi ile hesaplanır. Bir bölgenin yaşam için uygun bir bölge olup olmadığı önemlidir. Buna göre, eğer yaşam için uygunsa, canlı sayısı artar; aksi halde canlıların soyu tükenir. Algoritma, bu uygunluk indeksine göre göç ve mutasyon işlemlerini gerçekleştirir. Algoritma sonunda en iyi yaşam alanı, problemin en ideal çözümü olur (Simon, 2008).

d) Matematik tabanlı algoritmalar, matematiksel ve sezgisel algoritma tekniklerinin birleştirilmesiyle oluşturulmuştur (Boschetti vd., 2009).

e) Sürü tabanlı algoritmalar, doğadaki canlıların sürüler halinde yaşamaları ve sürüye özgü davranışlar sergilemeleri ve bu davranışların hayatlarını kolaylaştıracak etkili seçimler yapmalarını sağlaması fikrinden yola çıkarak, sürü davranışlarının modellenmesiyle ortaya çıkmıştır. Bu algoritmalar en çok bilineni PSO'dur (Kennedy ve Eberhart, 1995). Diğer bazıları; KKO (Dorigo ve Stützle, 2004), Kedi sürüsü algoritması (Chu vd. 2006), Yapay arı koloni algoritması (Karaboğa ve Baştürk, 2007), Cırcır böceği algoritması (Canayaz ve Karcı, 2015) gibi algoritmalar.

PSO, kuş ve balık sürülerinin yiyecek aramaları esnasında liderin (en öndekinin) davranışını taklit etme özelliklerini modelleyerek geliştirilmiştir. Algoritma, rastgele parçacık değerleriyle oluşturulan popülasyon ile başlar. Her parçacık için uygunluk değeri hesaplanır. Uygunluk değeri en küçük olan parçacığın pozisyon değeri ayrı bir değişkene atılır ve bu değere göre parçacıkların pozisyon ve hız değerleri güncellenir. Algoritma, iterasyon sayısınca yürütülür. Uygunluk değeri en küçük olan parçacığın pozisyonu, optimum çözümü verir (Kennedy ve Eberhart, 1995).

KKO, bir karıncanın bulduğu bir besine ulaşmak için takip ettiği yolda, feromon adında bir salgı bırakarak, arkasından gelen karıncaların bu besine kolay şekilde ulaşmasını sağlaması olayını modellemektedir. Algoritmaya, karıncaların takip edecekleri yollar rastgele oluşturularak başlanır. Yoldaki feromon miktarlarına bakarak, en kısa yoldan yiyeceğe ulaşmaya çalışılır. Feromonun belirli bir süre sonra kaybolduğu da dikkate alınır. Algoritma bitiminde karıncaların takip ettiği en kısa yol, optimum çözüm olarak sunulur. Genellikle en kısa yol problemleri ve GSP'lerde kullanılmaktadır (Dorigo ve Stützle, 2004).

Kedi sürüsü algoritması, kedilerin avını yakalama sürecinin temsiliyle geliştirilmiştir. İki aşamadan oluşur. Birinci aşama, kedinin avını takip aşamasıdır. İkinci aşama ise, kedinin avını gördükten sonra pozisyonunu ayarlaması aşamasıdır. Algoritma yürütülürken, öncelikle hafızadaki tüm kedilerin pozisyonları kopyalanır. Değişen boyut sayısına bağlı

olarak, seçilen boyutun arama aralığını belirten değişkenler, yüzdesel ve rastgele olarak ya arttırılır ya da azaltılır. Daha sonra uygunluk fonksiyonu hesaplanır. Eğer tüm kediler için uygunluk fonksiyonu değeri aynı değilse, her kedi için seçilme olasılığı hesaplanır ve kediler rastgele bir noktaya yönlendirilir. İkinci aşamada, kedilerin hızları algoritmanın limitleri içerisine çekilerek güncellenir. Bu hız değerlerine göre, kedilerin bulundukları konumlar güncellenir. Sonlandırma şartı sağlandığında, en iyi pozisyonadaki kedi, optimum çözümü verir (Chu vd., 2006).

Yapay arı koloni algoritması, arıların kovanlarında bal yapmak için bulmaları gereken nektar adındaki maddeyi arama ve bulduğunda kovandaki arılara bilgi verme işlemlerini modelleyen bir algoritmadır. Algoritmada nektar arama süreci, arama bölgelerinin alt ve üst limitlere göre rastgele üretilerek hafızaya eklenmesiyle başlar. Her bölge için aynı sayıda işçi arı görevlendirilir. Bu işçi arılar uygunluk fonksiyonuna göre bölgenin komşularının nektara yakın olup olmadığını ve kalitesini değerlendirir. Buna göre eğer komşudaki hafızadakinden daha iyiye, eski arama bölgesinin yerine yenisi yazılır. Algoritmanın ikinci aşamasında, hafızasındaki nektar bilgisine göre uygunluk fonksiyonu hesaplanır. Bu uygunluk fonksiyonu kullanılarak olasılıksal değerler hesaplanır. $[0,1]$ arasında rastgele belirlenen bir sayı ile bu olasılıksal değerler karşılaştırılır. Rastgele üretilen sayı, olasılıksal değerden büyükse, yeni çözüm üretilir. Aksi halde, eskisi korunur. Eğer yeni çözüm kullanılırsa, bu çözümün hafızadakinden daha iyi olup olmadığı da değerlendirilir. Nektarın bitme durumunda kâşif arılar, yeni nektar kaynakları ararlar. Algoritmada bu işlem, rastgele çözümler üreterek gerçekleştirilir. Algoritmanın iterasyon değeri hafızadaki bilgiden daha iyisine ulaşılmadığı durumda arttırılır (Akay, 2009).

Cırcır böceği algoritması, literatüre yeni girmiş bir algoritmadır. Cırcır böceklerinin kanat çırpışlarıyla havanın sıcaklığını tahmin etme özelliklerinin, optimizasyon problemlerinde kullanılmak amacıyla modellendiği algoritmadır. Algoritma, popülasyon değerlerinin rastgele oluşturulmasıyla başlar. Problemin uygunluk fonksiyonu değeri, cırcır böceklerinin çıkardıkları sesin şiddetinin belirlenmesiyle elde edilir. Ardından böceklerin kanat çırpış sayısı rastgele üretilir. Üretilen kanat çırpış sayısı kullanılarak hava sıcaklığı, sesin hızı ve sesin frekansı hesaplanır. Buna göre böceklerin koordinatları güncellenir. Güncellenen değerler amaç fonksiyonunun hesaplanmasında kullanılır. Eğer hesaplanan değer eski değerlerden iyi sonuç verirse, bu değer popülasyona alınır. Sonlandırma şartı sağlanana kadar algoritma bu şekilde yürütülür (Canayaz, 2015; Canayaz ve Karcı, 2015; Canayaz ve Karcı, 2016).

f) Sosyal tabanlı algoritmalar, insanlar ve insanların sosyal hayatlarında karşılaştıkları olaylardan esinlenerek ortaya çıkmıştır. Bu alanda literatürde bulunan algoritmalarından bazıları; TAA (Glover, 1989), Emperyalist yarışmacı algoritması (Gargari ve Lucas, 2007), Beyin fırtınası optimizasyon algoritması (Zhan vd., 2012), Öğretme-öğrenme tabanlı

optimizasyon algoritması (Rao ve Patel, 2013), Sosyal tabanlı algoritma (Ramezani ve Lotfi, 2013), Grup liderleri optimizasyon algoritması (Pooranian vd., 2013) ve Parlamenter optimizasyon algoritması (Altunbey ve Alataş, 2015)'dir. Bu algoritmalarından bazılarının genel çalışma prensiplerinden bahsedilecektir.

TAA, yeni çözümler üretirken en iyi hamleyi yapmayı hedefler. Bazı yeni çözümlere izin verirken, bazılarının da üretimini engeller. Buna tabu denilmektedir. Böylece algoritmanın yerel optimumda sınırlı kalmasını önler. Bazen de tabuları yıkarak, yeni çözüm üretmez hale gelme durumunun önüne geçer. Bu nedenle belirli aralıklarla tabu yıkımını gerçekleştirir. Algoritma, bir başlangıç çözümü oluşturarak çalışmaya başlar. Bu çözüm optimum çözüm olarak hafızaya alınır. Belirli dönüşüm tekniklerini kullanarak, geçilebilecek komşu çözümler hesaplanır. Bu teknikler, değişken değerini değiştirme, eleman ekleme veya çıkartma, iki çözüm arasında belirlenen değerlerin değiştirilmesi gibidir. Sonra, tabu olmayan ya da tabu olan fakat, tabu yıkma kriterlerini taşıyan komşu çözüme geçilir. Bu geçiş tabu olarak belirlenir ve eğer yeni çözüm, şu ana kadar ulaşılan en iyi çözümse, yeni çözüm, en iyi çözüm olarak atanır. Algoritma, tabu oluşturma ve yıkma süreçlerinin tekrarıyla yürütülür. Algoritma sonunda en iyi olarak kalan çözüm, optimum olarak sunulur (Glover, 1989; Gürbüz, 2015).

Emperyalist yarışmacı algoritmasına göre, başlangıçta arama uzayında güçlerine göre emperyalist ülkeler ve onların kolonileri bulunmaktadır. Algoritma boyunca emperyalist ülkeler güçlerini arttırmak için bir yarışa girerler ve birbirlerini elerler. Tek emperyalist ülke kalınca, yarış biter ve bu ülke optimum çözümü belirtir (Gargari ve Lucas, 2007).

Beyin fırtınası optimizasyon algoritması, bir problem için tüm önerilerin sunulması ve içlerinden en iyi önerinin seçilmesi yoluyla çözüme giden bir algoritmadır (Zhan vd., 2012).

Öğretme-öğrenme tabanlı optimizasyon algoritması, öğrencilerin dersleri anlama ve öğretmenlerin de dersleri anlatabilme becerilerini değerlendiren bir yöntemdir. En iyi öğretmen, optimum çözümü belirtmektedir. En iyi öğretmenin öğrencileri de, iyi çözümleri gösterir (Rao ve Patel, 2013).

Sosyal tabanlı algoritma, emperyalist yarışmacı algoritması ile evrimsel algoritmanın birleştirilmesiyle oluşturulmuştur. Bu algoritmaya göre, başlangıçta güçlü liderleri olan imparatorluklar kurulmaktadır. Daha sonra emperyalist maliyet fonksiyonuna bağlı olarak, güçlü imparatorluklar, diğer ülkeleri de yöneticisiyle birlikte kendi imparatorluğuna taşımaktadır. Algoritma sonunda, en güçlü imparatorluğun lideri, optimum çözümü belirtir (Ramezani ve Lotfi, 2013).

Parlamenter optimizasyon algoritması, bir meclis ve içinde seçim yoluyla birbiriyle yarışan partilerin bulunduğu ve partilerin içinde de kendi grupları içerisinde birbiriyle yarışan adayların olduğu sistemleri temsil ederek tasarlanmıştır. Buna göre grup içi ve grup

dışı yarışın sonunda yapılan birleştirme işlemi, seçimin sonucunu belirtmektedir. En güçlü aday ise, optimum sonucu göstermektedir (Altunbey ve Alataş, 2015).

g) Spor tabanlı algoritmalar, literatüre yakın zamanda girmiştir. En bilineni Lig şampiyonası algoritmasıdır (Kashan, 2009).

Lig şampiyonası algoritmasında, lig içindeki takımların birbirleriyle yaptıkları maçlar sonucu en iyi takımın belirlenmesi süreci, optimizasyon problemlerinin çözümüne uyarlanmıştır. Buna göre, her takım bir aday çözümü belirtir. En iyi takım ise, optimum çözümü gösterir. Takımlar kendi aralarında rastgele eşleştirilmektedir. İki takım maçtaki performansına göre ya galip gelir ya da mağlup olur. Takımlar güçlerini arttırmak için oyuncu transferi yaparlar. Sezon sonunda en yüksek skorlu takım, optimum çözüm olarak sunulur. (Bingöl ve Alataş, 2015).

h) Müzik tabanlı algoritmalar; notalar, çalgılar ve sesler arasındaki uyumu yakalama tekniklerini, problem optimizasyonu için uyarlayan yöntemlerdir. En çok bilineni Harmoni arama algoritmasıdır (Geem vd., 2001).

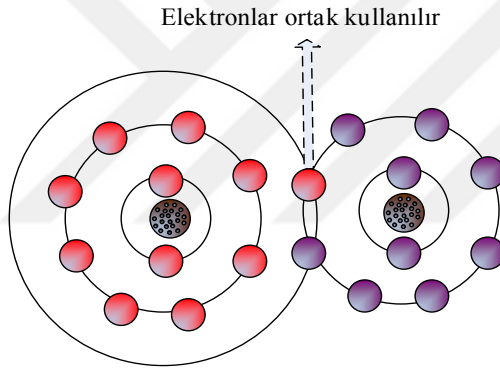
Harmoni arama algoritması, müzik notaları arasındaki uyumu yakalama amacıyla yapılan işlemlere dayandırılmaktadır. Algoritma, rastgele notalardan oluşan harmoni hafızasını üreterek çalışmaya başlar. Her nota, problemin bir parametresini belirtir. Buna göre bir harmoni birden çok notadan oluşur ve aday çözümü gösterir. Hafızanın güncellenmesi için yeni harmoniler geliştirilmelidir. Yeni harmoni üretme üç kurala göre yapılır. Bunlar; rastgele seçim, harmoniyi bozan durumları göz önünde bulundurma ve ton ayarlamadır. Bu durumlar aynı zamanda harmoni arama algoritmasının operatörleridir. Algoritmanın parametre değerlerine göre hangi operatörün uygulanacağı belirlenir ve bu operatör kullanılarak yeni harmoni oluşturulur. Yeni harmoni, harmoni hafızasındaki en kötü durumla karşılaştırılır. Eğer daha iyiye, kötü harmoni hafızadan atılarak, yerine yeni üretilen harmoni hafızaya alınır. Algoritma, sonlandırma şartı sağlanana kadar yürütülür ve sonuçta elde edilen en iyi harmoni, optimum çözümü gösterir (Geem vd., 2001; Lee ve Geem, 2004; Lee ve Geem, 2005).

i) Melez algoritmalar, birden çok sezgisel tekniğin bir arada kullanıldığı, bu şekilde optimum çözüme ulaşmaya çalışan yöntemlerdir.

3. YAPAY ATOM ALGORİTMASI (A^3)

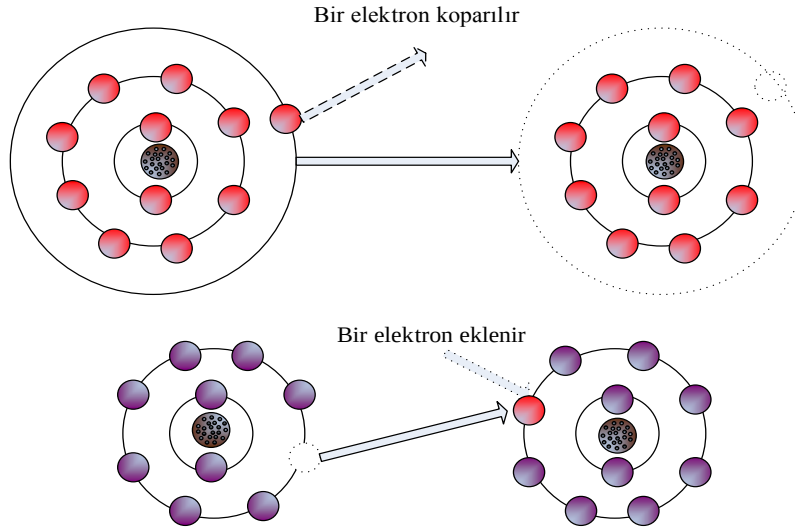
Yapay Atom Algoritması (A^3), optimizasyon problemleri için etkin çözümler sunan yeni nesil bir meta-sezgisel yaklaşımdır. Maddenin yapıtaşı olan atomun, elektron ortak kullanımı ya da elektron alış-verişine dayanan kimyasal bileşik oluşturma sürecinden esinlenerek tasarlanmıştır. A^3 yaklaşımının, atomun bu kimyasal süreçlerini esas alan iki ana operatörü bulunmaktadır. Bunlar; iyonik bağ ve kovalent bağ operatörleridir (Karcı, 2012; Yıldırım ve Karcı, 2013; Karadoğan ve Karcı, 2013).

Kimya bilimindeki kovalent bağ, bir atom içerisindeki bazı elektronların başka bir atom ile ortaklaşa kullanılmasıyla oluşan bağdır. Şekil 3.1.'de kimyadaki bu sürecin temsili gösterimi bulunmaktadır. A^3 'de kovalent bağ operatörü ise, sonuç üzerinde daha yüksek etkiye sahip parametre değerlerinin iki olası çözümde birden ortak kullanılmasıdır. Yani daha iyi değere sahip elektronların daha kötü olanların üzerine kopyalanarak, aynı değerlere sahip iki atom oluşturulmasıdır.



Şekil 3.1. Kovalent bağ sürecinin temsili gösterimi

Kimyada iyonik bağ süreci ise, atomlar arasında elektron alma ve verme ile oluşan bağdır. Elektron alışverişine dayanan bu sürecin temsili gösterimi ise, Şekil 3.2.'de verilmiştir. A^3 'de iyonik bağ operatörü ise, sonuç üzerinde düşük etkiye sahip parametre değerlerinin çözüm kümesinden çıkartılarak, yerine yeni değerlerin çözüm kümesine alınmasıyla gerçekleştirilir (Karcı, 2012; Demir ve Karcı, 2015; Canayaz ve Demir, 2016; Yıldırım ve Karcı, 2017).



Şekil 3.2. İyonik bağ sürecinin temsili gösterimi (Yıldırım ve Karcı, 2017)

A^3 , klasik sezgisel yöntemlerden farklı olarak, parametre değerlerinin sonuç üzerindeki etkisini hesaplayarak, etkisi az olan parametre değerlerini çözüm kümesinden çıkarmaktadır. Bu nedenle kıyaslama problemlerine uygulandığında, kısa sürede etkili çözümler elde ettiği görülmüştür. A^3 yürütülmeye başladığındaki atom sayısını algoritma süresince korur. Yine, tek noktadan değil birden çok noktadan arama yaparak çalıştığı için daha hızlı olarak çözüme ulaşır. Diğer bazı sezgisel yaklaşımlardan farklı olarak, çözüm kalitesini belirlemek için amaç fonksiyon haricinde başka fonksiyonlara ihtiyaç duymaz. A^3 'ün tek bir parametresi bulunmaktadır. Bu da iyonik oran (α) ve kovalent oran (β) toplamının 1'e eşit olmasıdır ($\alpha + \beta = 1$). Bu nedenle karmaşık olmayan, kolay uygulanabilir bir yapısı vardır (Karcı, 2012).

3.1. Temel Kavramlar

Elektron: Sonuç üzerinde etkisi bulunan her bir parametre değerine elektron denir. Elektron bir nokta belirtir. Genetik algoritmalarındaki gen yapısına benzemektedir.

Atom: Amaç fonksiyonu üzerinde etkisi olan tüm parametre değerlerinin oluşturduğu kümeye atom denir. Atom, elektronlardan oluşan bir aday çözümdür. Genetik algoritmalarındaki kromozom yapısına benzer ve bir vektör belirtir (Goldberg, 1989). Şekil 3.3.'de n adet elektron içeren bir atomun gösterimi verilmiştir (Karadoğan ve Karcı, 2014; Yıldırım ve Karcı, 2015; Çifci vd., 2017).

A[1]	A[2]	A[3]		A[n-2]	A[n-1]	A[n]
------	------	------	--	--------	--------	------

Şekil 3.3. Atomun gösterimi

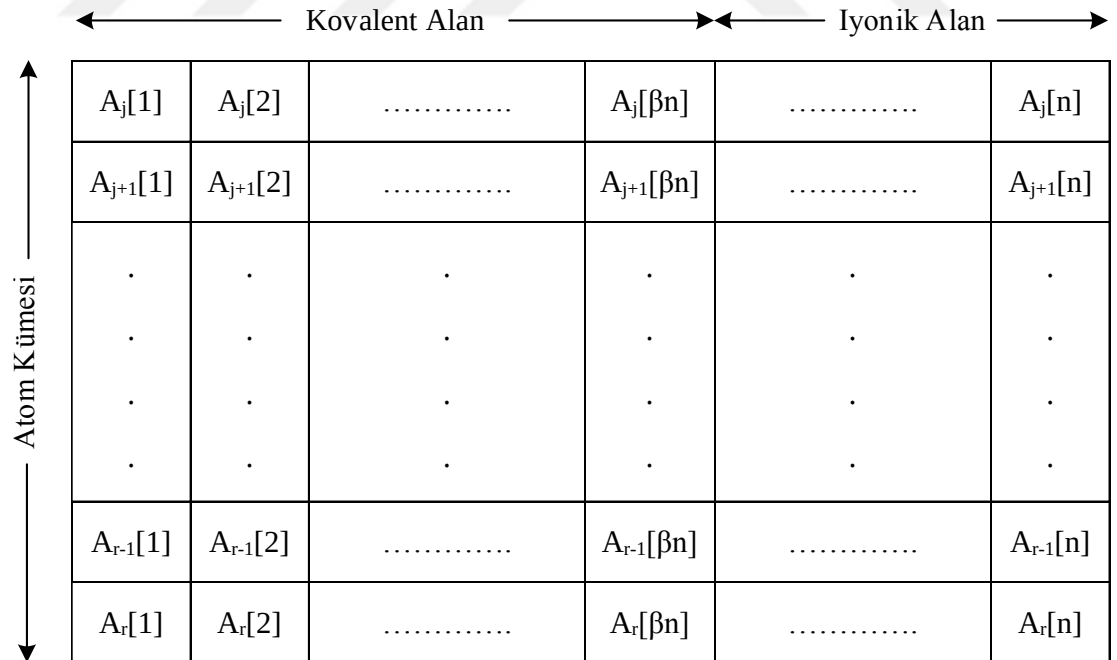
Atom Kümesi: Başlangıçta rastgele üretilen atomların oluşturduğu kümeye atom kümesi denir. Atom kümesi bir matris belirtir. Genetik algoritmalarındaki popülasyon kavramına karşılık gelir (Goldberg, 1989). Atom kümesindeki eleman sayısı, problemin boyutuna uygun olarak belirlenir ve algoritma süresince sabit kalır.

İyonik Oran (α): Probleme uygun olarak, iyonik bağ operatörünün uygulanacağı alanı belirlemek için kullanılan katsayıdır.

Kovalent Oran (β): İyonik orana benzer şekilde, kovalent bağ operatörünün uygulanacağı alanı belirlemek için kullanılan katsayıya kovalent oran denir. İyonik oran ve kovalent oranın toplamı 1'e eşittir ($\alpha + \beta = 1$).

İyonik Alan (İA): Atom kümesinde iyonik bağ operatörünün uygulandığı bölgeye iyonik alan denir. Problemin içerdiği parametre sayısına bağlı olarak, bir atomda bulunan elektron sayısına n denilecek olursa, iyonik alan αn ile belirlenir.

Kovalent Alan (KA): Atom kümesinde kovalent bağ operatörünün uygulandığı bölgeye ise kovalent alan denir. İyonik alandakine benzer şekilde, problemin içerdiği parametre sayısına bağlı olarak, bir atomda bulunan elektron sayısına n denilecek olursa, kovalent alan βn ile belirlenir. $(\alpha n + \beta n) = n$ 'dir. Şekil 3.4.'de elektron sayısı n , atom sayısı r olan bir atom kümesinin gösterimi bulunmaktadır. Atom kümesi üzerinde kovalent alan ve iyonik alanın gösterimi de yapılmıştır (Yıldırım ve Karcı, 2017a, b).



Şekil 3.4. Atom kümesi, kovalent alan ve iyonik alanın gösterimi (Yıldırım ve Karcı, 2017a)

3.2. Kovalent Bağ Operatörü

Kimya biliminde kovalent bağ, maddenin en küçük yapıtaşı olan atomun elektronlarını ortaklaşa kullanarak bileşik oluşturmaktır. A^3 bu ortak kullanım sürecini taklit ederek, atom adı verilen aday çözümlerde, sonuç üzerinde olumlu etkiye sahip elektron adı verilen parametre değerlerini başka bir atoma daha kopyalayarak, ortak kullanımını sağlamaktadır. Böylece iki atom da aynı parametre için aynı değere sahip olmaktadır. Bu işlem kovalent bağ olarak adlandırılmaktadır (Çifci vd., 2017). Kovalent bağ operatörü, atom kümesinin yalnızca kovalent alanına uygulanır. Karmaşık bir operatör değildir. Oldukça kolay uygulanır.

A_j ve A_r iki farklı atom olsun. n her bir atomun bulundurduğu elektron sayısı ve βn kovalent alandaki elektron sayısını belirtir. $A_j[i]$, A_j atomunun i . elektronunu belirtmektedir. $A_r[i]$ ise, A_r atomunun i . elektronunu göstermektedir. $E[A_j[i]]$, $A_j[i]$ elektronunun amaç fonksiyonu üzerindeki etki değerini belirtirken, $E[A_r[i]]$, $A_r[i]$ elektronunun amaç fonksiyonu üzerindeki etki değerini belirtir. Buna göre, kovalent bağ operatörünün sözde kodu aşağıdaki gibidir (Yıldırım ve Karcı, 2017a, b):

```
 $i \leftarrow 1, 2, \dots, \beta n \quad // i \leq \beta n$   
Eğer  $E[A_j[i]]$  ,  $E[A_r[i]]$  ‘den daha iyiye,  
     $A_j[i]$  değerini  $A_r[i]$  ‘ye kopyala  
Aksi halde,  
     $A_r[i]$  değerini  $A_j[i]$  ‘ye kopyala
```

Atom kümesinin ilk atomu A_j ve son atomu A_r olsun. Kovalent bağ operatörü, atom kümesindeki ilk ve son atomlar eşleştirilerek işletilmeye başlar. Yani başlangıçta A_j ve A_r eşleştirilir. Sonraki adımlarda j arttırılarak, r ise azaltılarak yürütülür ve atom kümesindeki tüm atomlara kovalent bağ operatörü uygulanana kadar devam eder. Kovalent bağ operatörünün algoritma adımları Algoritma 3.1’de verilmiştir (Yıldırım ve Karcı, 2017b).

Algoritma 3.1: Kovalent Bağ Operatörü

KovalentBag (AtomKumesi, EtkiDegerKumesi (E), β , n, m)

$K \leftarrow \text{AtomKumesi} \ // \ \text{AtomKumesi} = \{A_1, A_2, \dots, A_n\}$

$j \leftarrow 1$

$r \leftarrow m \ // \ m : \text{Atom sayısı}$

while $K \neq \emptyset$

A_j ve A_r atomlarını eşleştir

// $A_j[i], A_r[i] \in \text{AtomKumesi}$

$i \leftarrow 1, 2, \dots, \beta n \ // \ \beta : \text{Kovalent oran} \ n : \text{Elektron sayısı}$

if ($E[A_j[i]] < E[A_r[i]]$)

$A_j[i] \leftarrow A_r[i]$

else

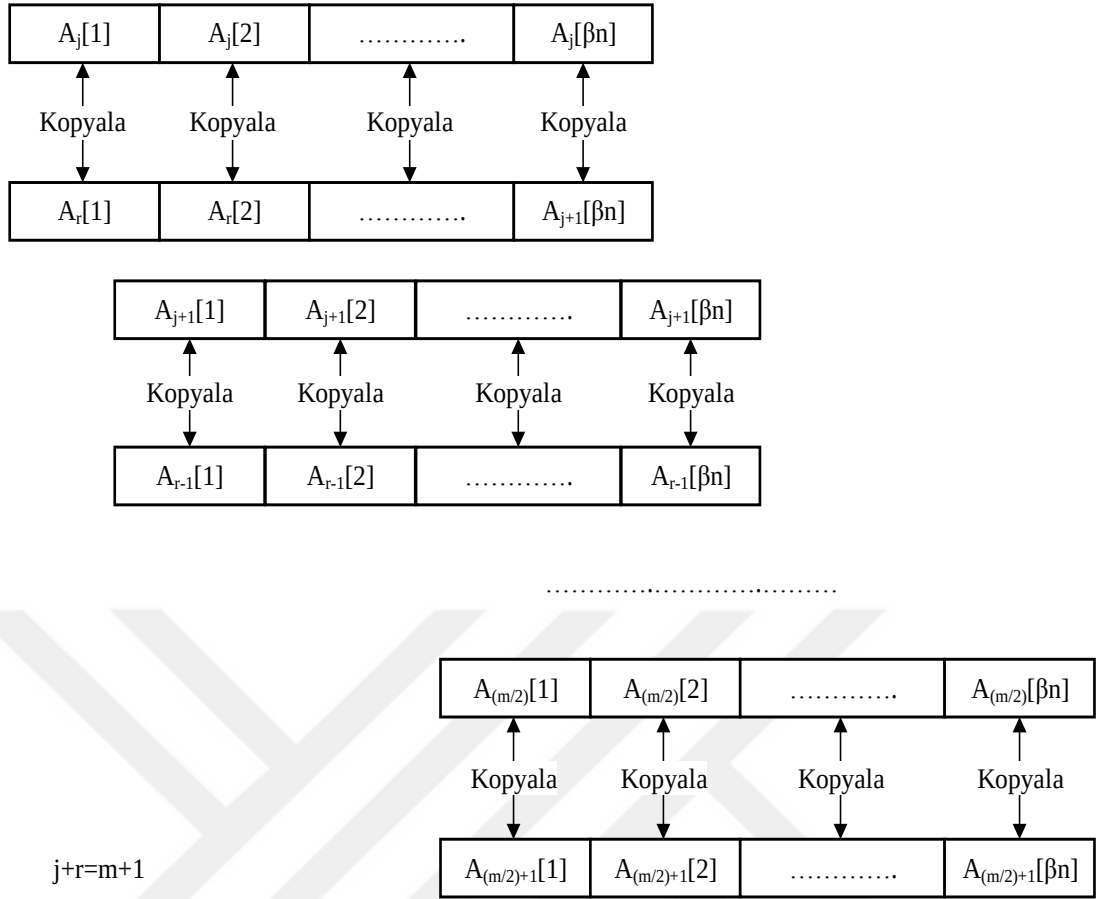
$A_r[i] \leftarrow A_j[i]$

$K \leftarrow K - \{A_j, A_r\}$

$j \leftarrow j + 1$

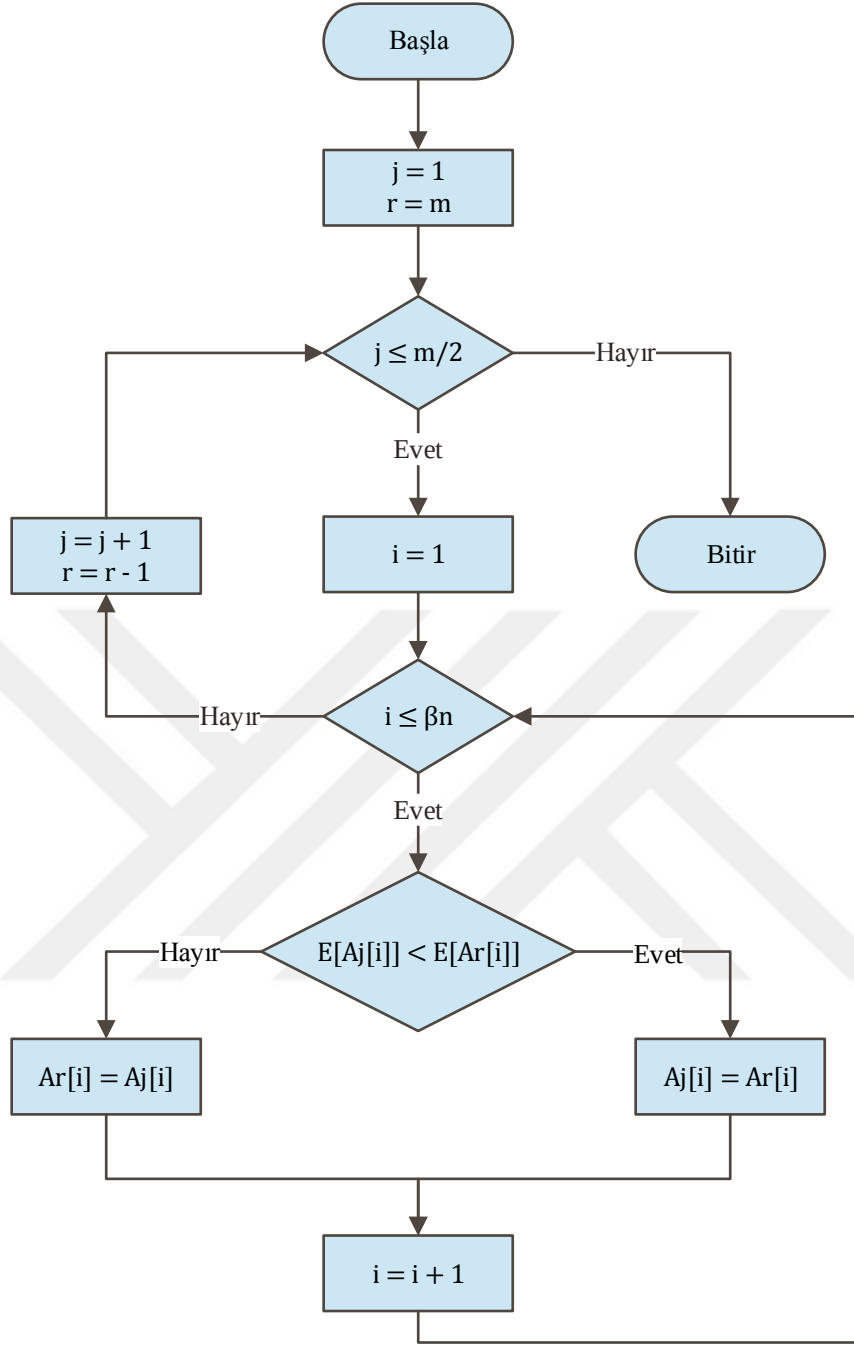
$r \leftarrow r - 1$

Kovalent bağ operatörünün gösterimi Şekil 3.5.'deki gibidir. Burada, m atom kümesinde bulunan atom sayısını göstermektedir. Buna göre A_j ve A_r gibi iki atom eşleştirilir. $j = 1$ iken, $r = m$ 'dir. Sonraki adımda $j = j + 1$ olurken, $r = r - 1$ olur. Bu şekilde yürütülerek son adımda $j = (m/2)$ olurken, $r = (m/2) + 1$ olur. Yani $j + r = m + 1$ 'dir (Yıldırım ve Karcı, 2017a, b).



Şekil 3.5. Kovalent bağ operatörünün gösterimi

Kovalent bağ operatörünün akış diyagramı Şekil 3.6.'da gösterildiği gibidir. Buna göre, bir atomun bulundurduğu elektron sayısı n , kovalent alandaki elektron sayısı βn ve atom kümesinde bulunan atom sayısı m ile temsil edilmektedir.



Şekil 3.6. Kovalent bağ operatörünün akış diyagramı

Kovalent bağ operatörünün uygulanmasıyla, atom kümesinde iyi sonuç veren elemanların korunması, çoğaltılması ve böylece optimum sonuca ulaşma hedeflenir. Kovalent bağ operatörü sezgisel bir operatör değildir, deterministlik bir operatördür.

3.3. İyonik Bağ Operatörü

Kimyasal bir süreç olan iyonik bağ, elektron alarak veya vererek gerçekleşmektedir. A^3 'de iyonik bağ operatörü, çözüm üzerinde olumsuz etkiye sahip olan elektronların atom kümesinden atılarak, yerlerine rastgele yeni elektronların atom kümesine dâhil edilmesiyle gerçekleştirilir. Bu değişim süreci iyonik bağ olarak isimlendirilmiştir (Çifci vd., 2017). İyonik bağ operatörü, atom kümesinin yalnızca iyonik alanına uygulanır. Bir atomun iyonik alanı αn sayıda elektron içerir ve bu elektronlar atomun sonunda konumlanmaktadır.

$A_j[i]$ atom kümesinin iyonik alanında bulunmak üzere; j . atomun i . indeksindeki elektronunu belirtmektedir. L_i , problemin i . niteliği için alt sınırı; U_i ise, problemin i . niteliği için üst sınırı göstermektedir. η , Matlab'ın random() fonksiyonuyla 0 ile 1 arasında üretilen bir sayıdır. Buna göre iyonik bağ operatörünün sözde kodu aşağıdaki gibidir (Yıldırım ve Karcı, 2017a, b):

$i \leftarrow \beta n + 1, \dots, n$ // β : Kovalent oran n : Elektron sayısı
 $A_j[i] \leftarrow L_i + \eta * (U_i - L_i)$

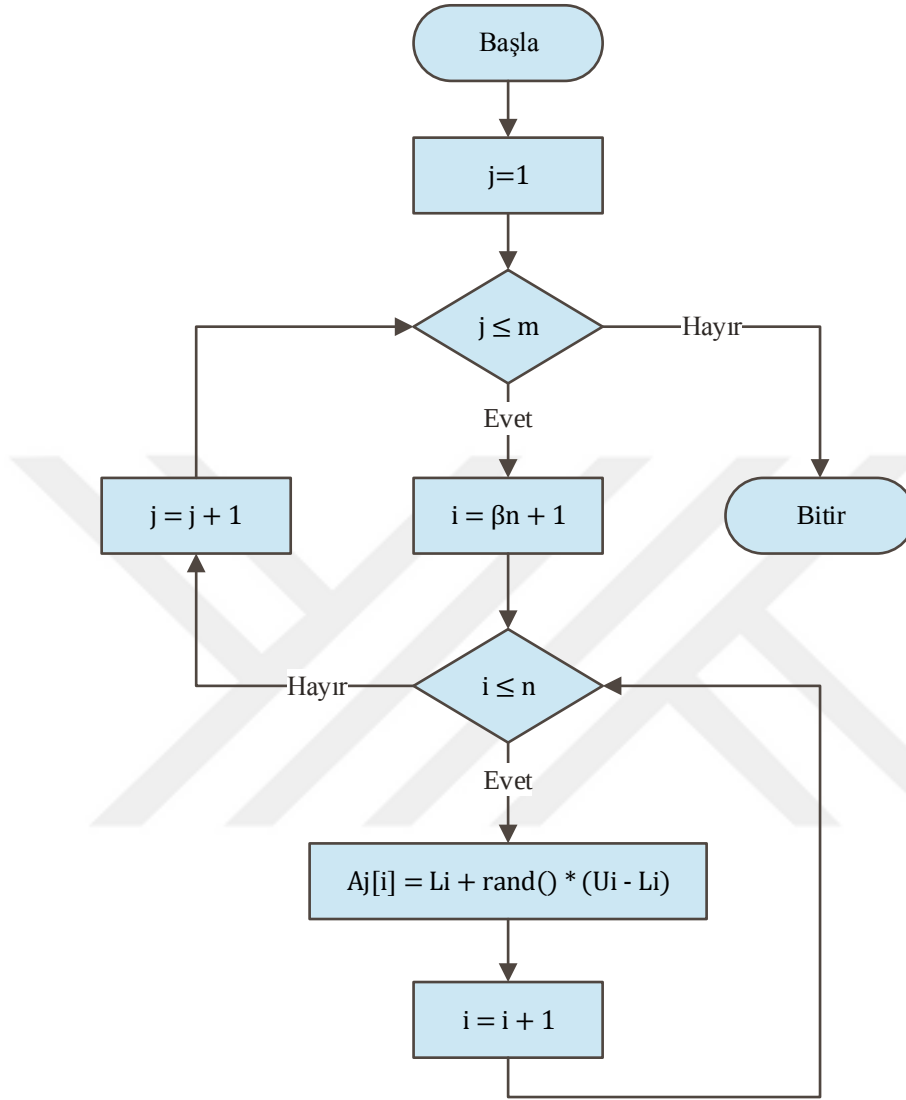
Atom kümesinde bulunan atom sayısı m , her bir atomun sahip olduğu elektron sayısı ise n ile gösterilmektedir. Buna göre, iyonik bağ operatörünün algoritma adımları Algoritma 3.2'de verilmiştir (Yıldırım ve Karcı, 2017b).

Algoritma 3.2: İyonik Bağ Operatörü

IyonikBag (*AtomKumesi*, m , n , β)
 $j \leftarrow 1, \dots, m$ // m : Atom sayısı
 $i \leftarrow \beta n + 1, \dots, n$ // β : Kovalent oran n : Elektron sayısı
 $A_j[i] \leftarrow L_i + \eta * (U_i - L_i)$
// $A_j[i] \in \text{AtomKumesi}$
// η : (0-1) arasında rastgele üretilen bir sayı
// U_i : i . nitelik için üst sınır
// L_i : i . nitelik için alt sınır

Bir atomun bulundurduğu elektron sayısı n , kovalent alandaki elektron sayısı βn , iyonik alandaki elektron sayısı αn ve $(\beta n + \alpha n) = (\beta + \alpha) n = n$ 'dir. Dolayısıyla iyonik alanın başlangıç indeksi $(\beta n + 1)$, son indeksi ise n olmaktadır. Atom kümesinde bulunan atom

sayısı ise m ile temsil edilmektedir. Buna göre, iyonik bağ operatörünün akış diyagramı Şekil 3.7.'de gösterildiği gibidir.



Şekil 3.7. İyonik bağ operatörünün akış diyagramı

İyonik bağ olasılıksal bir operatördür. İyonik alandaki eski değerleri, rastgele belirlenen yeni değerlerle değiştirdiği için daha iyiye gidişi garanti etmez. Ancak değer uzayında bulunan farklı elemanların da en iyi çözüm olma olasılığını değerlendirdiği için, yerel optimumda sınırlı kalmayarak, global optimumu yakalamayı sağlar.

İyonik bağ operatörünün uygulanması oldukça kolaydır. Operatörün uygulanması, elektron etki değerlerine göre sıralı atomların iyonik alandaki elektronlarını, Matlab'ın random() fonksiyonuyla üretilen yeni değerlerle değiştirmekten ibarettir. Ancak yeni değerler üretilirken, parametre sınırlarına dikkat edilmelidir.

3.4. Elektron Etkisi

Kovalent bağ operatörü uygulanırken, elektronların etki değerlerine göre karşılaştırma yapılır ve amaç fonksiyon üzerinde etkisi çok olan elektronun değeri, amaç fonksiyon üzerinde etkisi az olan elektrona kopyalanır. Bu nedenle elektronların amaç fonksiyon üzerindeki etkileri tek tek hesaplanmalıdır. Algoritma başlangıcında elektron etki değer matrisi rastgele olarak doldurulur. Amaç fonksiyon değerleri hesaplandıktan sonra, her bir elektronun amaç fonksiyon üzerindeki etki değeri de hesaplanarak matris güncellenir. Bu güncelleme iyonik bağ ve kovalent bağ operatörleri uygulandıktan sonra da yapılır. Ayrıca kovalent bağ operatörü ile bir elektronun değeri diğerine kopyalanınca, etki değeri de kopyalanır. Böylece algoritma süresince güncel etki değerleri ile karşılaştırma yapılır.

Amaç fonksiyon üzerindeki elektron etki değerlerini hesaplamak için Denklem (3.1)'deki formül kullanılabilir.

$$f(\vec{x}) = \sum_{i=1}^n f(x_i) \quad (3.1)$$

Eğer amaç fonksiyonu bağımsız değişkenlere ayrılabilirse, her elektronun amaç fonksiyon üzerindeki etkisi kesinlikle hesaplanabilir. Öte yandan, eğer amaç fonksiyon bağımsız değişkenlere ayrılmaz ise, her bir elektronun amaç fonksiyon üzerindeki etkisi yaklaşık olarak hesaplanabilir (Yıldırım ve Karcı, 2017b).

3.4.1. Amaç fonksiyonu bağımsız değişkenlere ayrılabilen elektronların etki değeri

Eğer amaç fonksiyonu bağımsız değişkenlerine ayrılabilirse, her elektronun etki değeri kesin olarak hesaplanabilir. Bu hesaplama için Denklem (3.2) kullanılmalıdır.

$$E[A_j[i]] = f(x_i), \quad i = 1, 2, \dots, n \\ j = 1, 2, \dots, m \quad (3.2)$$

Denklem (3.2)'de, bir atomda bulunan elektron sayısı n , atom kümesinde bulunan atom sayısı ise m ile gösterilmektedir. $A_j[i]$, j . indeksli atomun i . indeksli elektronunu belirtmektedir. $E[A_j[i]]$ ise, $A_j[i]$ atomunun elektron etkisini göstermektedir (Yıldırım ve Karcı, 2017b).

3.4.2. Amaç fonksiyonu bağımsız değişkenlere ayrılamayan elektronların etki değeri

Amaç fonksiyonu bağımsız değişkenlerine ayrılamazsa, elektron etki değerleri yaklaşık olarak hesaplanabilir.

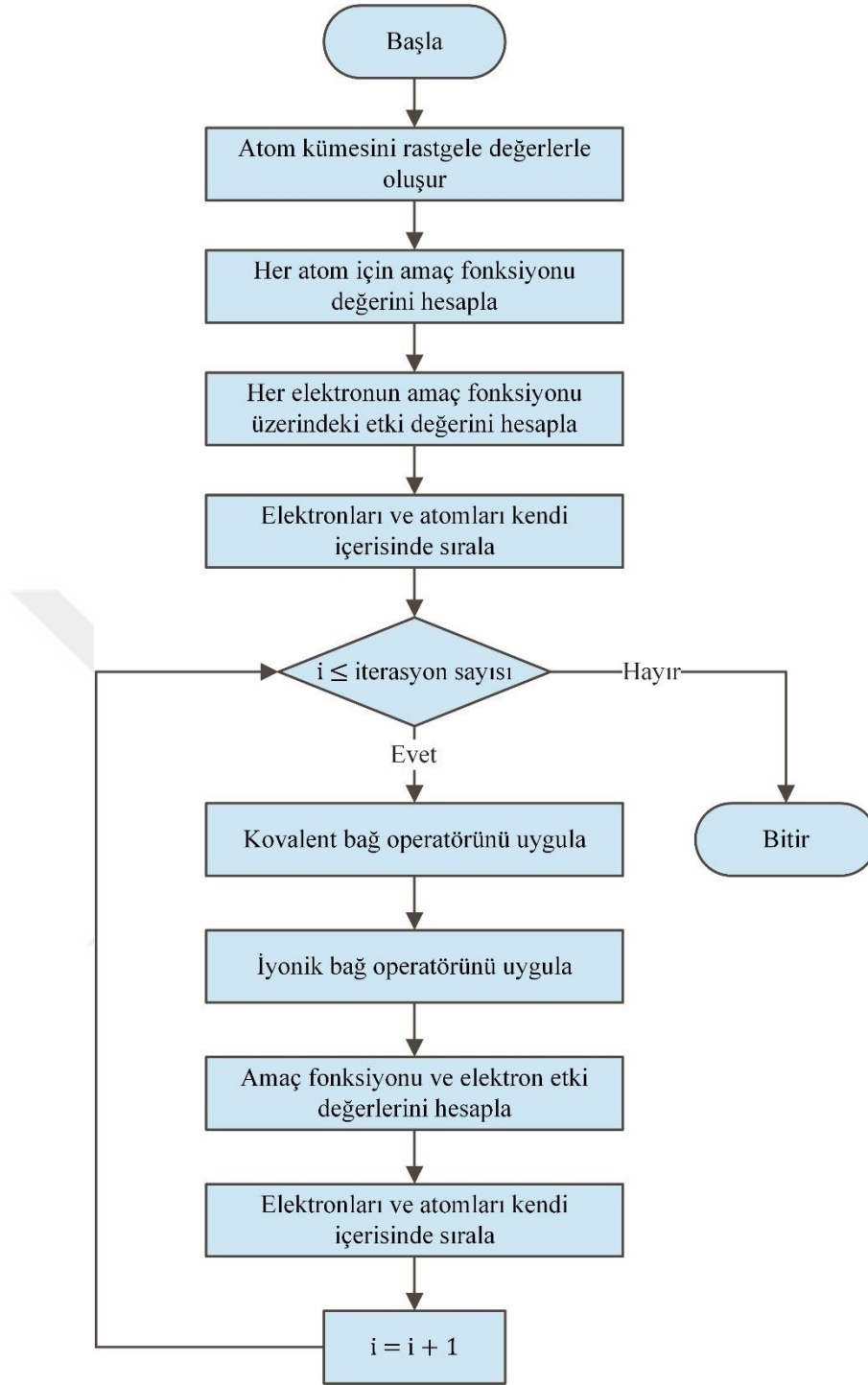
$$E[A_j[i]] = \sum_{k=1}^n f(x_k) - \sum_{k=1, k \neq i}^n f(x_k) \quad (3.3)$$

Denklem (3.3)'de, bir atomda bulunan elektron sayısı n ile gösterilmektedir. $A_j[i]$, j . indeksli atomun i . indeksli elektronunu belirtmektedir. $E[A_j[i]]$ ise, $A_j[i]$ atomunun elektron etkisini göstermektedir. Eğer amaç fonksiyonu bağımsız değişkenlere ayrılamazsa, aynı atom üzerindeki elektronların etki değerleri Denklem (3.3) kullanılarak hesaplanabilir (Yıldırım ve Karcı, 2017b).

3.5. Algoritma Adımları

A^3 , başlangıçta rastgele üretilen atom kümesi üzerinde operatörlerini uygular. Atom kümesi algoritma sonuna kadar sabit sayıda atom içerir. Algoritmanın ilk adımı, problem kısıtlarına uygun aralıklarda ve her bir parametre değeri bir elektron ile temsil edilecek şekilde, aday çözümleri belirten atomlardan oluşan, problemin boyutuna ve karmaşıklığına uygun atom kümesinin rastgele oluşturulmasıdır. Atom kümesi oluşturulduktan sonra her bir atom için amaç fonksiyonu değeri hesaplanır. Ardından, her bir elektronun amaç fonksiyonu üzerindeki etki değeri hesaplanır. Hem atom kümesi, amaç fonksiyonu değerine göre sıralanır hem de her bir atom, elektron etki değerine göre sıralanır. Sıralanan atom kümesine öncelikle iyonik bağ operatörü uygulanır. İyonik bağ operatörü ile problemin çözümü için olumsuz etkiye sahip elektronlar atom kümesinden çıkartılırken, yerine rastgele üretilen yeni elektronlar atom kümesine dâhil edilir. İyonik bağ operatöründen sonra, atom kümesine kovalent bağ operatörü uygulanır. Kovalent bağ operatörü ise, iki atomu elektron etkileri bakımından karşılaştırır ve etki değeri büyük olan elektronları diğerinin yerine kopyalar. Böylece iki atom içerdiği elektronlar bakımından birebir aynı olur. Atomların yeni hali için amaç fonksiyonu değeri ve elektron etkileri hesaplanır. Ardından atomlar, amaç fonksiyonu değerlerine göre, elektronlar ise etki değerlerine göre kendi içerisinde tekrar sıralanır. Sonlandırma şartı sağlanıncaya kadar ya da belirlenen iterasyon sayısına erişilinceye kadar iyonik bağ ve kovalent bağ operatörlerinin uygulanmasıyla algoritma devam eder (Karadoğan, 2014; Yıldırım ve Karcı, 2017a, b).

i iterasyon sayısını belirten değişken olmak üzere, A^3 'ün algoritma adımlarını içeren akış diyagramı Şekil 3.8.'de verilmiştir.



Şekil 3.8. A³'ün akış diyagramı

A³ yaklaşımında, atom kümesindeki kovalent alan içerisindeki elektronlar, tüm iterasyonlarda çözümün bir parçası iken, iyonik alan içerisindeki elektronlar sürekli olarak yenilenir. İyonik alan içerisindeki elektronlar, kovalent bağ operatörü uygulanırken kullanılmaz ve amaç fonksiyonun değeri hesaplanırken çözüme dâhil edilmez. Yani iyonik alan, yedek elektronların bulunduğu bir bölgedir denilebilir. Bu bölgedeki elektronların etki

değerleri kovalent alandaki elektronlardan daha iyiye, çözüm alanına girme şansı bulurlar. Bu yapı, doğadan esinlenen diğer algoritmalarda mevcut değildir.



4. TEZİN LİTERATÜRE KATKISI

Bir meta-sezgisel algoritmanın alanında başarılı olduğunun ispatı, o yöntemin sürekli optimizasyon problemlerinde gösterdiği başarıyı ayrık problemlerde de gösterebilmesinden geçmektedir. Yani bir meta-sezgisel yöntemin ayrık problemlerin çözümünde gösterdiği performans, o algoritmanın literatürde kabul görmesi için önemli bir ölçüttür. Bu sebeple, yapılan tez çalışmasında, yakın zamanda kimyasal bileşik oluşturma süreçlerinin modellenmesiyle geliştirilen A^3 yönteminin, daha önce kıyaslama (benchmark) problemlerinde elde ettiği başarılı sonuçlar göz önünde bulundurularak, algoritmanın literatürde kabul görebilecek güçlü bir yapıya sahip, alanında başarılı bir algoritma olduğunu ispatlamak adına, ayrık problemlere uygulanabilirliği test edilmiştir. Elde edilen sonuçlar, aynı problemlerin çözümünde kullanılan farklı meta-sezgisel yöntemlerle performans bakımından karşılaştırılmış ve değerlendirilmiştir.

Yapılan tez çalışmasında A^3 , farklı parametre değerlerine sahip farklı yapıdaki ayrık problemlerin çözümünde kullanılmıştır. Ayrık problemler, sınırlı sayıda ayrık parametre değerlerinin seçilmesi, sıralanması ya da gruplanması şeklinde optimizasyonun yapıldığı, tanımlanması kolay, ancak çözümü zor problemlerdir. Bu nedenle, her ne kadar algoritma güçlü operatörlere sahip olsa da, bu operatörlerin ayrık problemlere uygulanması için temel algoritma mantığını bozmadan çeşitli teknikler geliştirilmesi şarttır.

Bu amaçla ilk olarak, GSP'nin çözümü için A^3 yöntemi üzerinde bazı düzenlemeler yapılmıştır. Şekil 4.1.'de görüldüğü gibi A^3 sürekli parametre değerleri alan problemlere uygulanırken, atom kümesinin sol bölgesi kovalent alan, sağ bölgesi iyonik alan olarak belirlenmiştir. Kovalent alan, aday çözümleri oluşturan elektronları içerirken; iyonik alan da, yedek parametre değerlerini (yedek elektronları) bulundurmaktadır. İyonik bağ operatörü uygulanırken, çözüm için kötü değerlere sahip elektronlar atom kümesinden çıkarılır, yerine rastgele oluşturulan yeni elektronlar atom kümesine dahil edilir.

← Kovalent Alan →			← İyonik Alan →	
$A_1[1]$		$A_1[\beta n]$		$A_1[n]$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$A_r[1]$		$A_r[\beta n]$		$A_r[n]$

Şekil 4.1. Kıyaslama problemleri için A^3 'ün yapısı

GSP'nin çözümü için atom kümesi tekrarsız ve sıralı (permütasyon) yapıda oluşturulmuştur. Bu sebeple, A^3 'ün temel mantığındaki gibi iyonik alanı rastgele değerlerle güncellemek, atom kümesinin yapısını bozacaktır. Bu durum, GSP için şehir tekrarlarına ya da bazı şehirlerin hiç ziyaret edilmemesine sebep olacağı için problem kısıtlarına uygun değildir. Örneğin; 8 şehirli bir GSP için, A^3 yönteminde herhangi bir değişiklik yapılmadan iyonik bağ operatörü uygulandığında, bir aday çözüm içerisinde tekrar tekrar ziyaret edilen ya da hiç ziyaret edilmeyen şehirler olduğu Şekil 4.2.'deki gibi görülmektedir.

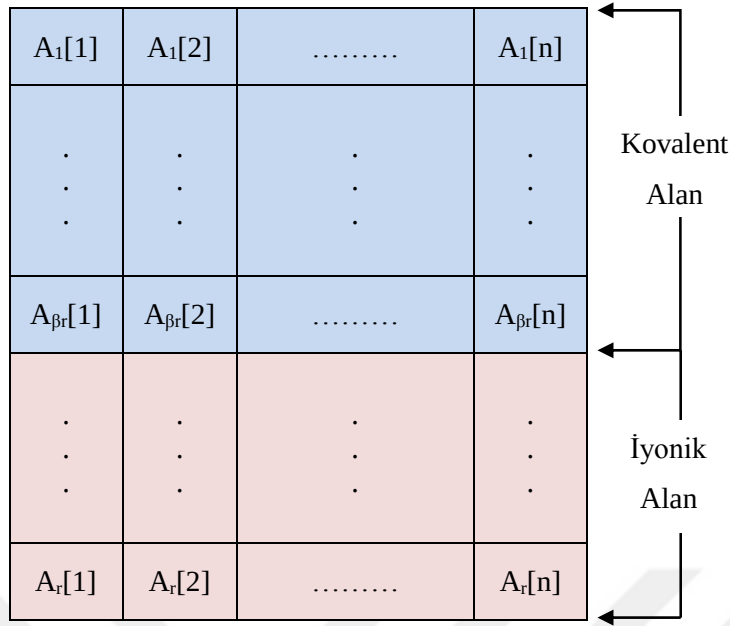
8	2	6	1	4	7	3	5
---	---	---	---	---	---	---	---

↓

8	2	6	1	4	8	1	3
---	---	---	---	---	---	---	---

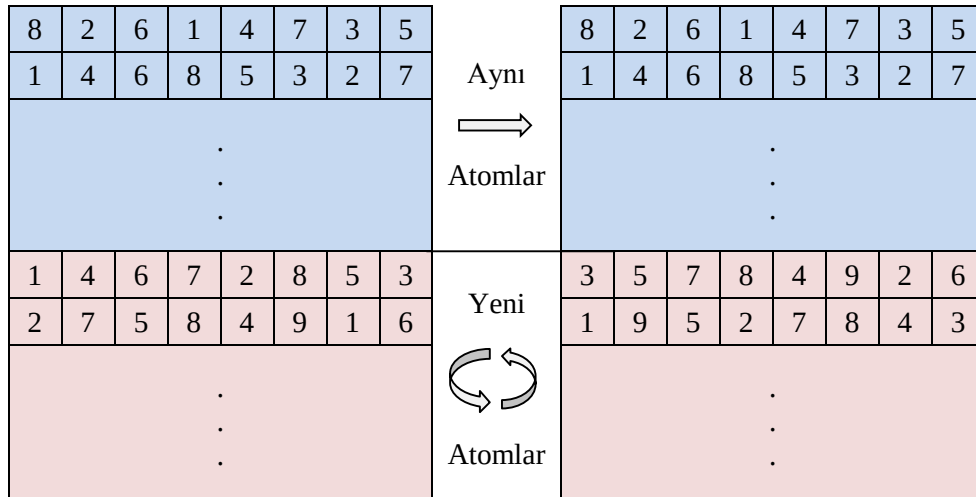
Şekil 4.2. 8 şehirli bir GSP için A^3 'ün sürekli problemlerdeki kullanımına göre iyonik bağ operatörünün uygulanışı

Bu nedenle, GSP'nin A^3 ile çözümü için yeni bir model tasarlanmıştır. Şekil 4.3.'de görüldüğü gibi, atom kümesinin üst bölgesi kovalent alan, alt bölgesi ise iyonik alan olarak belirlenmiştir. Böylece iyonik bağ operatörünün uygulanmasıyla, atom kümesine amaç fonksiyon değerlerine göre yeni parametre değerleri değil de, onun yerine doğrudan aday çözümler dâhil edilmektedir.



Şekil 4.3. GSP için A^3 'ün yapısı

Bu tez çalışmasında, A^3 üzerinde yapılan değişikliklerle, bu yönteminin GSP gibi ayrık parametre değerlerinin sıralanışının önemli olduğu optimizasyon problemlerine uygulanabilmesi sağlanmıştır. Şekil 4.4.'de görüldüğü gibi, geliştirilen teknik ile 8 şehirli bir GSP için iyonik bağ operatörünün uygulanmasıyla bir aday çözüm içerisinde, tekrar eden ya da hiç ziyaret edilmeyen şehirlerin oluşması engellenmiştir.



Şekil 4.4. 8 şehirli bir GSP için A^3 'ün ayrık problemler için geliştirilen formuna göre iyonik bağ operatörünün uygulanışı

Ayrıca yapılan tez çalışmasıyla, GSP'nin A^3 ile çözümünde iyonik bağ operatörü uygulanırken olduğu gibi, kovalent bağ operatörü uygulanırken de, her şehrin bir kez ziyaret edilmesi ve şehir tekrarından kaçınılması kısıtları göz önünde bulundurularak, aday çözümlerin bu kısıt şartları sağlayan yapılarının bozulmaması amacıyla bazı teknikler geliştirilmiştir. Aksi halde, A^3 'ün sürekli problemlerin çözümü için geliştirilen formuna göre kovalent bağ operatörü uygulandığında, eşleştirilen iki atom arasında elektronlar kopyalanırken, bazı şehirlerin tekrar ziyareti ve bazı şehirlerin ise hiç ziyaret edilmemesi söz konusu olmaktadır. Bu amaçla literatürden de yararlanılarak geliştirilen tekniklere, Deneysel Uygulamalar bölümünde ayrıntılı olarak yer verilmiştir.

A^3 'ün sürekli problemlerin çözümü için geliştirilen formuna göre, atomlar amaç fonksiyon değerlerine göre kendi aralarında sıralanırken, elektronlar da etki değerlerine göre sıralanmaktadır. GSP için bu algoritma adımı, yalnızca atomların amaç fonksiyonu değerlerine göre sıralanması şeklinde değiştirilmiştir. Böylece, sıralanışıyla aday çözümleri belirten atomların, bu sıralı yapılarının bozulmasının önüne geçilmiştir.

Tez çalışmasında kullanılan bir diğer ayrık problem olan Grup asansör kontrol optimizasyonu probleminde aday çözümler, asansörlerin hangi sırayla hangi kat çağrısını cevaplayacağını belirtmektedir. Bu nedenle atom kümesinin yapısında ve iyonik bağ operatörünün uygulanmasında GSP için yapılan değişiklikler, parametre değerlerinin sıralanışının önemli olduğu grup asansör kontrol optimizasyonu problemi için de uygun bulunmuştur. Buradan da anlaşılabileceği üzere; yapılan değişikliklerle A^3 yöntemi, sıralı yapıdaki farklı ayrık optimizasyon problemlerine uygulanabilecek formata dönüştürülmüştür. Bu tez çalışmasıyla geliştirilen teknikler, ileride ayrık optimizasyon alanında A^3 yöntemi kullanılarak yapılacak çalışmaların önünü açan niteliktedir.

Ayrıca, parametre değerlerinin sıralanışlarının önem taşımadığı beslenme optimizasyonu problemi için ise, A^3 'ün sürekli problemler için geliştirilen yapısı kullanılmıştır. Böylece, farklı yapıdaki ayrık problemlerin çözümünde A^3 yaklaşımı kullanılarak, elde edilen başarılı sonuçlarla yeni bir meta-sezgisel algoritma olan A^3 'ün alanında güçlü ve uygulanabilir bir algoritma olduğu ispatlanmıştır.

5. DENEYSEL UYGULAMALAR

5.1. Gezgin Satıcı Problemi

Gezgin Satıcı Problemi (GSP), kombinasyonel optimizasyon alanında sıklıkla kullanılan önemli bir NP-zor problemidir (Held ve Karp, 1970). GSP’de bir satıcı mallarını bütün şehirleri gezerek satmak ister. Bu amaçla satıcı, bir şehirden yola çıkarak, bütün şehirleri yalnızca bir kez ziyaret etmeli ve sonuçta başlangıç şehrine geri dönmelidir. GSP’de amaç, uzaklıkları bilinen şehirler arasında, mümkün olan en kısa rotayı kullanarak seyahati tamamlamaktır (Held ve Karp, 1970; Lin, 1965; Lawler vd., 1985; Applegate vd., 2007). Problemin zorluğu, şehir sayısı arttıkça problemin matematiksel yöntemlerle çözümünün zorlaşmasından kaynaklanmaktadır. Rotayı belirlerken, başlangıçta satıcının seçebileceği n şehir bulunmaktadır. İlk seçimden sonra, $(n - 1)$ şehir kalır. İkinci şehirdeyken, satıcının seçebileceği $(n - 2)$ şehir vardır. Böylece, satıcının tüm seçimleri $(n - 1)!$ karmaşıklığına sahiptir. Bu sebeple problem boyutu büyüdükçe, yani şehir sayısı arttıkça, problemin matematiksel yöntemlerle çözülmesi de zorlaşmaktadır (Çolak, 2010; Dantzig vd., 1954).

GSP matematiksel olarak aşağıdaki gibi ifade edilir:

$$\text{Min } z = \sum_{i=1}^n \sum_{j=1, i \neq j}^n x(i, j) d(i, j) \quad (5.1)$$

Kısıtlayıcılar:

$$\sum_{i=1, i \neq j}^n x(i, j) = 1, j = 1, 2, \dots, n, i = 1, 2, \dots, n \quad (5.2)$$

$$\sum_{j=1, i \neq j}^n x(i, j) = 1, i = 1, 2, \dots, n, j = 1, 2, \dots, n \quad (5.3)$$

$$\sum_{i, j \in S, i \neq j}^n x(i, j) \leq |S| - 1, \forall S \subset \{1, 2, \dots, n\}, 2 \leq |S| \leq n - 2 \quad (5.4)$$

Denklem (5.1)’de GSP için amaç fonksiyonu verilmiştir. $d(i, j)$, i ve j noktaları arasındaki uzaklığı göstermektedir. $x(i, j)$, i noktasından j noktasına giden bir yol olup olmadığını belirtmektedir. Eğer i noktasından j noktasına bir yol mevcutsa $x(i, j) = 1$, aksi halde $x(i, j) = 0$ ’dır. Denklem (5.2) ve (5.3) bir şehrin yalnızca bir kez ziyaret edildiğini garanti eder. Denklem (5.4) alt turda eliminasyonun kısıtlayıcısıdır (Miller vd., 1960; Gavish ve Graves, 1978; Gutin ve Punnen, 2002).

GSP’ler iki şekilde sınıflandırılabilir. Birincisi, problemin boyutuna göre sınıflandırmadır. Boyutuna göre yapılan sınıflandırmada GSP’ler büyük ölçekli ve küçük

ölçekli olmak üzere iki sınıfa ayrılırlar. GSP için problemin boyutunda belirleyici olan şehir sayısıdır. Şehir sayısı arttıkça, problemin çözüm süresi artmakta ve çözüm başarısı azalmaktadır. İkincisi ise, şehirlerarasındaki mesafelerde simetriklik olup olmaması durumuna göre yapılan sınıflandırmadır. Buna göre, simetrik ve asimetrik GSP olarak sınıflandırma yapılır. GSP için i ve j iki şehri belirtmek üzere; $d(i, j) = d(j, i)$ ise simetrik GSP, $d(i, j) \neq d(j, i)$ ise asimetrik GSP'dir. Örneğin, A ve B seyahat edilmesi gereken iki şehir olsun. Simetrik GSP'de, A şehrinden B şehrine gitmek için alınması gereken yol, B şehrinden A şehrine gitmek için alınması gereken yola eşittir. Asimetrik GSP'de ise, A'dan B'ye gitmek için alınması gereken yol, B'den A'ya gitmek için alınması gereken yoldan farklıdır.

Şehir sayısı arttıkça GSP'lerin matematiksel yöntemlerle çözümleri neredeyse imkânsız hale gelmektedir. Bu nedenle GSP'lerin çözümünde sezgisel tekniklerin kullanılması daha uygun olmaktadır (Mudaliar ve Modi, 2013). Yapılan çalışmalarda, doğadan esinlenerek geliştirilen çeşitli sezgisel ve meta-sezgisel yöntemler, büyük ölçekli ve küçük ölçekli, simetrik ve asimetrik GSP'lerin çözümünde kullanılmış ve olumlu sonuçlar vermiştir. Bu nedenle meta-sezgisel bir yöntem olan ve kıyaslama (benchmark) problemlerinin çözümünde oldukça iyi sonuçlar elde eden A³'ün, GSP'lerinin çözümünde de olumlu sonuçlar vermesi beklenmektedir.

5.1.1. A³ ile GSP uygulaması

Bu bölümde, ayrı bir problem olan GSP'nin, yakın zamanda geliştirilmiş bir algoritma olan A³ kullanılarak çözümü gerçekleştirilmiştir. A³, literatüre girmiş 8 adet simetrik GSP'nin çözümüne uygulanmıştır. Elde edilen sonuçlar diğer bazı sezgisel yöntemlerle karşılaştırılmıştır. Bu yöntemler; TAA, GA, PSO, KKO ve bu algoritmaların farklı kombinasyonları ile üretilmiş olan PSO-KKO ve GA-PSO-KKO'dur (Deng vd., 2012).

A³ ile GSP uygulaması, Matlab 2010b yazılımı kullanılarak Intel Core-i3 3.3 GHz işlemcili PC üzerinde gerçekleştirilmiştir. Uygulamada kullanılan veriler, TSPLib kütüphanesinden elde edilmiştir. Şehir sayısı sırasıyla 14, 29, 29, 48, 51, 52, 70, 76 olan burma14, bayg29, wi29, att48, eil51, berlin52, st70, pr76 olmak üzere, literatürde sıklıkla kullanılan küçük boyutlu simetrik GSP'ler için uygulama gerçekleştirilmiştir.

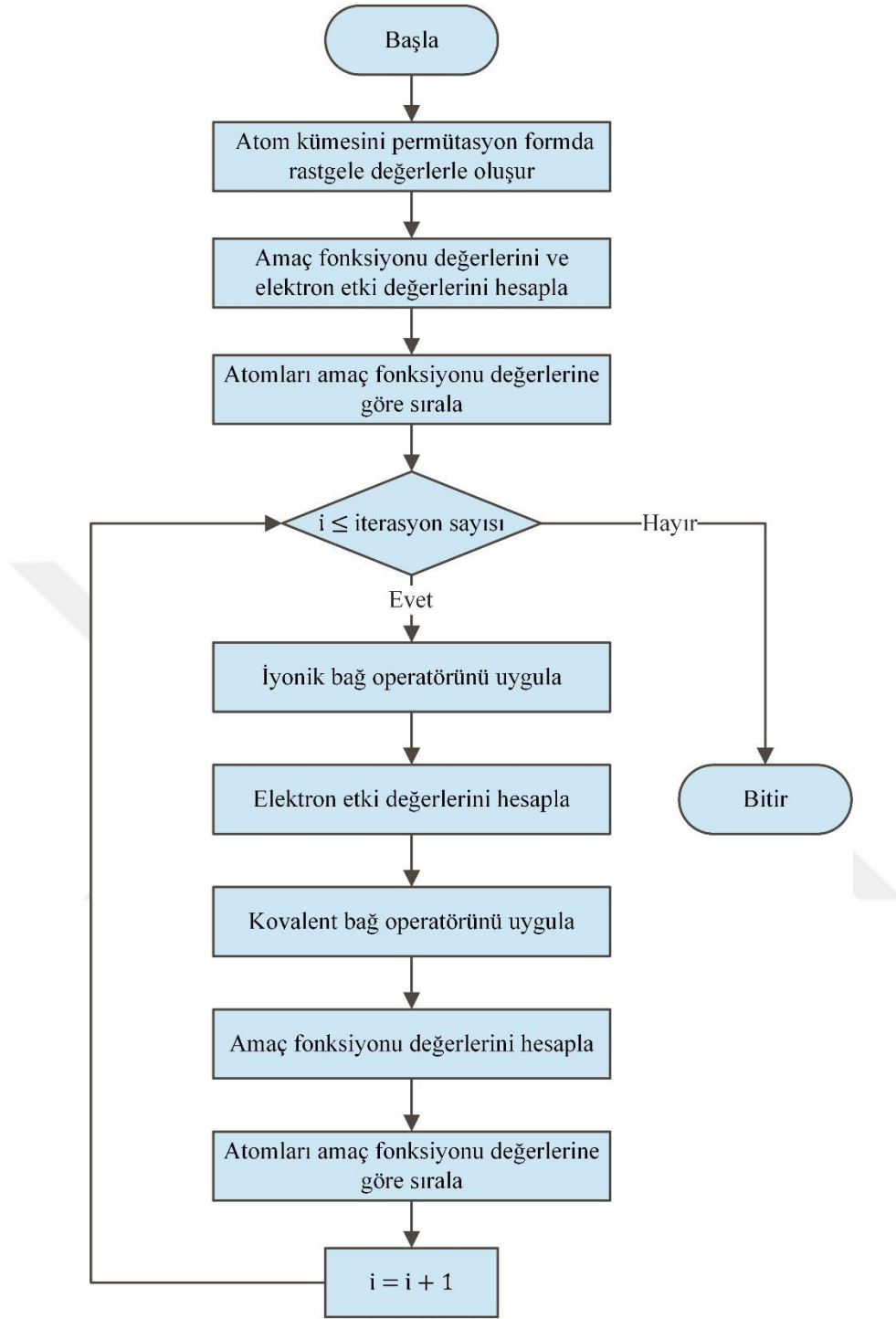
GSP uygulaması gerçekleştirilirken, A³ için kullanılan parametre değerleri Çizelge 5.1.'de verilmiştir.

Çizelge 5.1. GSP için A^3 uygulamasında kullanılan parametre değerleri
(*GSP'deki şehir sayısına göre değişmektedir.)

Parametre	Değeri
Elektron sayısı	14-29-48-51-52-70-76 *
Atom sayısı	100
Kovalent Oran (β)	0.5
İyonik Oran (α)	0.5
İterasyon sayısı	1000

GSP uygulaması için A^3 'ün algoritmik adımlarında bazı değişiklikler yapılmıştır. Öncelikle, daha önce ziyaret edilen bir şehrin tekrar ziyaretinden kaçınmak için, atom kümesinin oluşturulmasında kıyaslama (benchmark) problemlerinden farklı olarak permütasyon yöntemi kullanılmıştır. Aynı şekilde A^3 operatörleri uygulanırken, atom kümesinin tekrarsız ve sıralı formunu korumak için bazı teknikler geliştirilmiştir. Atomların amaç fonksiyon değerleri ve elektronların etki değerleri hesaplandıktan sonra, GSP için iyonik bağ operatörü, kovalent bağ operatöründen önce uygulanmıştır. İyonik bağ operatörü uygulandığında, atom kümesine rastgele belirlenmiş yeni atomlar girdiği için, onların etki değerlerini de hesaplamak gerekmektedir. Bu nedenle kovalent bağ operatöründen önce, elektronların amaç fonksiyon üzerindeki etki değerleri tekrar hesaplanmıştır. Ardından kovalent bağ operatörü uygulanarak, atom kümesi için amaç fonksiyon değerleri hesaplanmıştır. Atom kümesi sıralandıktan sonra, iterasyon sayısı kontrol edilerek, belirlenen iterasyon sayısına ulaşıncaya kadar A^3 operatörleri uygulanmaya devam edilmiştir. Kıyaslama problemleri için A^3 uygulanırken, hem amaç fonksiyon değerlerine göre atom kümesi sıralanır hem de elektron etki değerlerine göre her atom kendi içerisinde sıralanır. Oysaki GSP için elektronların sıralanışı, şehirlerin ziyaret sırasını belirtmektedir. Bu sebeple elektronların etki değerlerine göre sıralanması, aday çözümü tamamen bozacağı için GSP'nin yapısına uygun değildir. Bu nedenle A^3 'ün GSP için uyarlanan formunda, atomlar, elektron etki değerlerine göre sıralanmazlar.

GSP uygulaması için A^3 'ün akış diyagramı Şekil 5.1.'de verilmiştir.



Şekil 5.1. GSP uygulaması için A^3 'ün akış diyagramı

A^3 ile GSP uygulamasında, atomların amaç fonksiyon değerleri Denklem (5.1)'deki formülasyon ile hesaplanmıştır. GSP için amaç fonksiyonu, rastgele bir şehirden başlayarak tüm şehirlerin gezilmesiyle alınan toplam mesafe ve başlangıç noktasına geri dönüş mesafesinin toplamı olarak belirlenmiştir. Ayrıca, A^3 ile GSP uygulamasında, her bir elektronun bir şehir belirttiği kabul edilmiştir. Buna göre elektronların amaç fonksiyonu üzerindeki etki değerleri hesaplanırken, bir şehrin etki değeri, kendisinden sonra ziyaret

edilen komşu şehre olan uzaklığı ile hesaplanmıştır. GSP için bir atomda bulunan elektron sayısı n ile belirtilirse; $A_r[i]$, A_r atomunun i . indeksli elektronu ve $E[A_r[i]]$, $A_r[i]$ elektronunun etki değeri olmak üzere; eğer $i \neq n$ ise, elektronların etki değerlerini hesaplamak için Denklem (5.5) kullanılır. Eğer $i = n$ ise, elektronların etki değerlerini hesaplamak için Denklem (5.6) kullanılır.

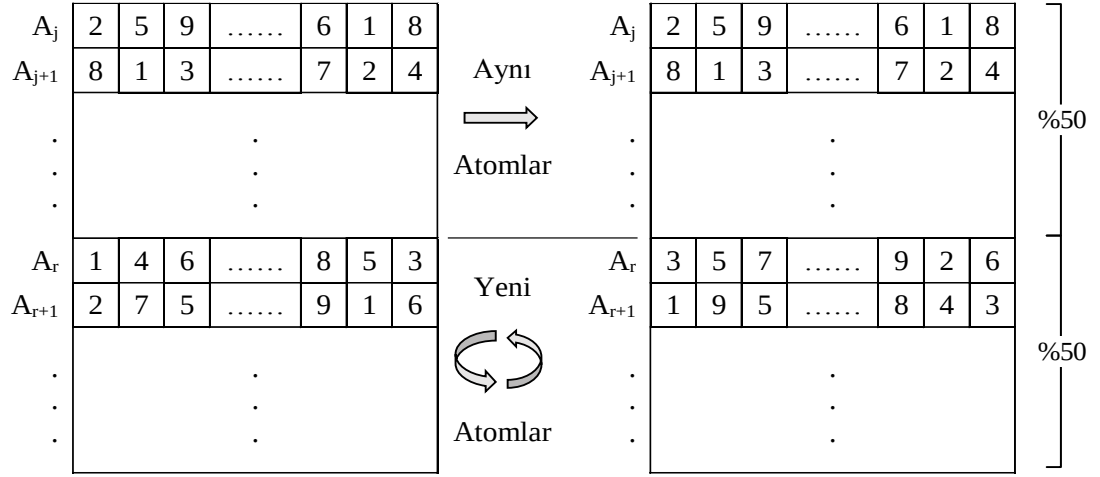
$$E[A_r[i]] = x(i, i + 1) * d(i, i + 1) \quad (5.5)$$

$$E[A_r[i]] = x(n, 1) * d(n, 1) \quad (5.6)$$

Denklem (5.5)'de $x(i, i + 1)$, i . şehir ile $i + 1$. şehir arasında bir yol olup olmadığını belirtirken; $d(i, i + 1)$, i . şehir ile $i + 1$. şehir arasındaki mesafeyi göstermektedir. Denklem (5.6)'da ise, $x(n, 1)$, n . şehir ile 1. şehir arasında bir yol olup olmadığını belirtirken; $d(n, 1)$, n . şehir ile 1. şehir arasındaki mesafeyi göstermektedir.

A^3 'ün GSP için uyarlanan şeklinde, atom kümesinin tekrarsız permütasyon yapısını bozmamak için iyonik alan, atom kümesinin alt bölgesi olarak belirlenmiştir. Yani her atomun belli bir bölümü iyonik alan olarak değil de, atom kümesinin belli bir bölümü iyonik alan olarak ayrılmıştır. İyonik oran ve kovalent oran birbirine eşit olarak belirlendiği için, atom kümesinin ilk yarısı kovalent alan, ikinci yarısı ise iyonik alan olarak tanımlanmıştır. Yani atom kümesinde bulunan atom sayısı m olmak üzere; 1 ile $(m/2)$ arası kovalent alan, $(m/2) + 1$ ile m arası iyonik alan olarak belirlenmiştir.

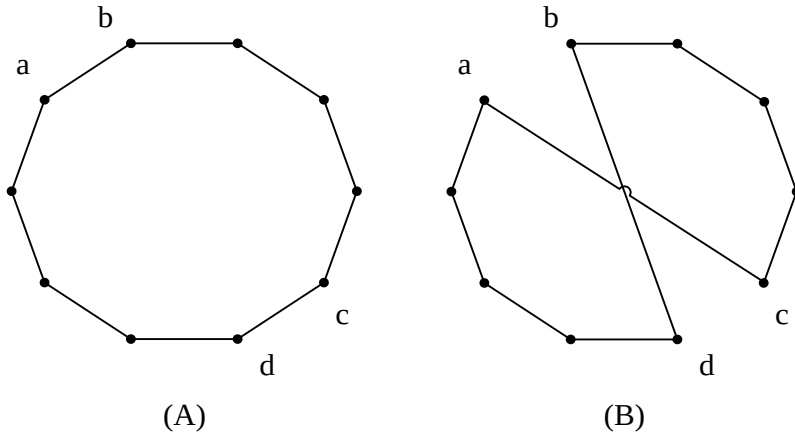
İyonik bağ operatörü ile atom kümesinin iyonik alanındaki elemanlar rastgele şekilde yenilenirken, kovalent alandaki elemanlar aynı şekliyle korunmuştur. İyonik alandaki atomların yerine, yine tekrarsız permütasyon formda rastgele atomlar oluşturulmuştur. Şekil 5.2.'de iyonik bağ operatörü uygulanırken atom kümesinde meydana gelen değişiklikler bir örnekle gösterilmektedir.



Şekil 5.2. A^3 'ün iyonik bağ operatörünün GSP için uygulanaşına bir örnek

GSP için kovalent bağ operatörü uygulanırken, atomlar kıyaslama problemlerinde uygulanandan farklı şekilde eşleştirilmiştir. m atom kümesinin içerdğı atom sayısı, A_j ve A_r ise kovalent bağ operatörünün uygulanacağı iki atomu belirtmektedir. A^3 ile kıyaslama problemlerinin çözümünde sırasıyla $j = 1, \dots, (m/2)$ ve $r = m, \dots, (m/2) + 1$ değerlerini alırken, GSP'nin çözümünde sırasıyla $j = 1, \dots, (m/2)$ ve $r = (m/2) + 1, \dots, m$ değerlerini alır.

Kovalent bağ operatörü uygulanırken, atomların permütasyon yapısını korumak için bazı teknikler kullanılmıştır. Atomlar kendi aralarında ikişerli olarak eşleştirildikten sonra, elektronların amaç fonksiyonu üzerindeki etki değerleri karşılaştırılır. Daha küçük etki değerine sahip olan elektron, diğer atom üzerinde bulunur ve bu elektron a olarak isimlendirilir. a 'dan bir sonraki indeksteki elektron b olarak adlandırılır. Daha büyük etki değerine sahip olan elektron c olarak belirlenir ve onun sonraki indeksindeki elektron d olarak adlandırılır. Bu tanımlamalar yapıldıktan sonra, daha büyük etki değerine sahip elektrona Şekil 5.3.'deki 2-opt. yöntemi (Croes, 1958; Ouaraab vd., 2014) ve tez çalışmasıyla bu yöntem esas alınarak geliştirilen benzer teknikler uygulanır.



Şekil 5.3. 2-opt. yöntemi (A) Uygulanmadan önceki hali (B) Uygulandıktan sonraki hali (Ouaarab vd., 2014)

Kovalent bağ operatörünü uygulamak için atom kümesi içerisinde A_j ve A_r gibi iki atomun eşleştirildiği farz edilsin ve atom kümesinde bulunan atomların içerdiği elektron sayısı n ile gösterilsin. Denklem (5.7)'de gösterildiği gibi k ve l değişkenleri, a elektronunun eşleştirilen iki atom üzerindeki indekslerini göstermektedir.

$$A_j[k] = A_r[l] = a \quad // \quad k = \{1, \dots, n-1\}, l = \{1, \dots, n\} \quad (5.7)$$

Buna göre, GSP için A^3 'ün kovalent bağ operatörünün uygulanması aşamasında 5 farklı durum oluşmaktadır. Bu farklı durumlar için, kovalent bağ operatörünün ana mantığını koruyarak, tekrarsız permütasyon yapıdaki bu ayrık problem için tez çalışması ile geliştirilen teknikler aşağıda gösterildiği gibidir.

Eğer $k < l$ ise, $A_r[k+1:l] = A_r[l:-1:k+1]$ olur.

Eğer $l \leq k$ ise; a) eğer $l = 1$ ve $k = n-1$ ise,

$$A_r[k+1:l] = A_r[l:k+1] \text{ olur.}$$

b) eğer $l \neq 1$ ve $k = n-1$ ise,

$$A_r[k+1:-1:l] = A_r[l:k+1] \text{ olur.}$$

c) eğer $l = 1$ ve $k \neq n-1$ ise,

$$A_r[k:-1:l+1] = A_r[l+1:k] \text{ olur.}$$

d) eğer $l \neq 1$ ve $k \neq n-1$ ise,

$$A_r[k+1:-1:l] = A_r[l:k+1] \text{ olur.}$$

$A_j(k)$ ve $A_r(k)$ elektronlarının etki değerleri $E[A_j(k)]$ ve $E[A_r(k)]$ ile ifade edilmektedir. Buna göre, $E[A_j(k)] < E[A_r(k)]$ ise kovalent bağ operatörü A_r atomuna

uygulanır, aksi halde A_j atomuna uygulanır (Yıldırım ve Karcı, 2017b). GSP’de elektron etkilerinin şehirlerarası uzaklığa göre belirlenmesi nedeniyle, kıyaslama problemlerinden farklı olarak, etki değeri küçük olan değil de, büyük olan kovalent bağ operatörü ile güncellenmektedir.

A^3 ile GSP’de kovalent bağ operatör uygulanırken oluşan farklı durumlar Çizelge 5.2, 5.3, 5.4, 5.5 ve 5.6’da gösterilmektedir (Yıldırım ve Karcı, 2017b).

Çizelge 5.2. Kovalent bağ operatörü uygulanırken oluşan 1. durum

A_j		a	
	k		
A_r		c	d
	k		l

Çizelge 5.3. Kovalent bağ operatörü uygulanırken oluşan 2. durum

A_j		a	
	k		
A_r	a		b
	l	k	

Çizelge 5.4. Kovalent bağ operatörü uygulanırken oluşan 3. durum

A_j					a	
	k					
A_r	c		b	a		d
	l			k		

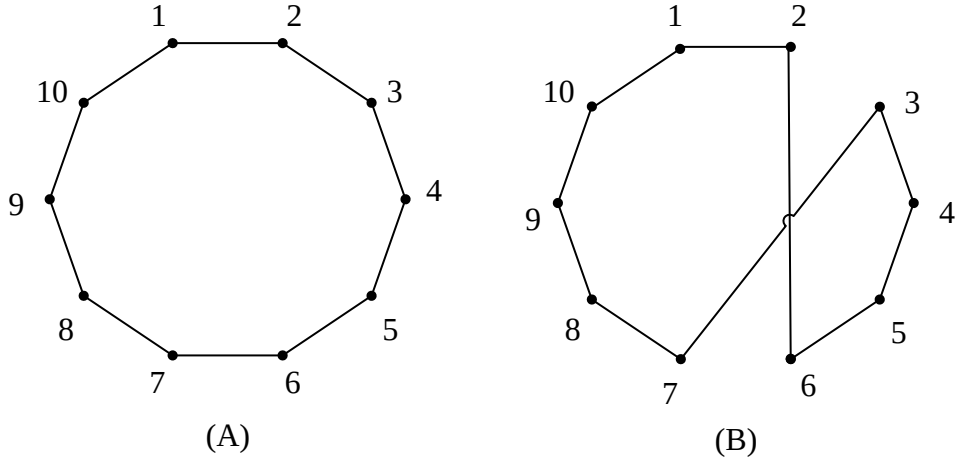
Çizelge 5.5. Kovalent bağ operatörü uygulanırken oluşan 4. Durum

A_j					a	
	k					
A_r	a	b		c	d	
	l			k		

Çizelge 5.6. Kovalent bağ operatörü uygulanırken oluşan 5. Durum

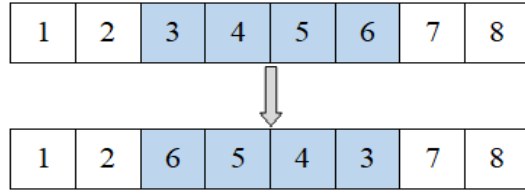
A_j					a	
	k					
A_r	b	a			d	c
	l			k		

$n = 10$ için 2-opt. yönteminin bir tur üzerinde uygulandığı Şekil 5.4.'de gösterilmektedir.

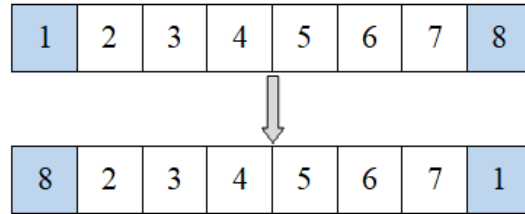


Şekil 5.4. Eğer $k = 2$, $l = 6$ ise, 2-opt. yöntemiyle bir tur (A) Uygulanmadan önce (B)Uygulandıktan sonra

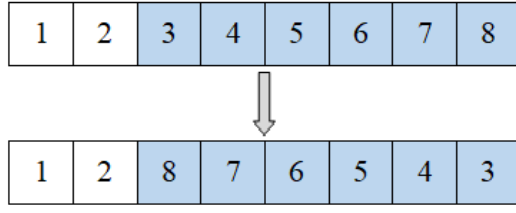
$n = 8$ için kovalent bağ operatörü uygulanırken, oluşan farklı durumlara karşı geliştirilen teknikler, örnek atomlar üzerinden Şekil 5.5, 5.6, 5.7, 5.8 ve 5.9’da verilmiştir (Yıldırım ve Karcı, 2017b).



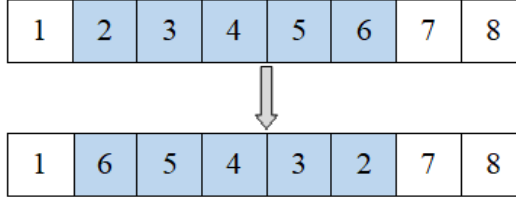
Şekil 5.5. Eğer $k = 2$, $l = 6$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim



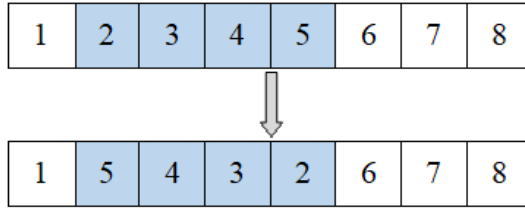
Şekil 5.6. Eğer $k = 7$, $l = 1$ ise, uygulanan yönteme göre bir atomda oluşacak değişim



Şekil 5.7. Eğer $k = 7$, $l = 3$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim



Şekil 5.8. Eğer $k = 6$, $l = 1$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim



Şekil 5.9. Eğer $k = 4$, $l = 2$ ise, 2-opt. yöntemine göre bir atomda oluşacak değişim

A^3 ile GSP'lerin çözümünde, şehirler arasındaki mesafeler Öklidyen uzaklık metodu kullanılarak hesaplanmış ve elde edilen sonuçlar tamsayıya yuvarlanmıştır. Aksi halde, TSPLib kütüphanesinde bulunan problemlere ait sunulan en iyi sonuçların, verilen optimum rotaya göre hesaplanan uzaklık toplamından farklı çıktığı görülmüştür (Shang, 2005; Xu vd., 2012). Bu nedenle uzaklıklar hesaplandıktan sonra, en yakın tamsayı değere yuvarlanmıştır.

5.2. Beslenme Optimizasyonu Problemi

Sağlıklı bir yaşam sürdürmek için, sağlıklı beslenmek şarttır. Beslenme (nütrisyon), bireylerin bedensel, zihinsel ve ruhsal faaliyetlerini etkileyen önemli bir unsurdur. Özellikle yatan hastalar ve sporcular için beslenme planlaması oldukça önemlidir (Demirel ve Bahçecioğlu, 2010; Ghisalberti, 1979; Nazni ve Vimala, 2010). Beslenme alışkanlığındaki bozukluklar, aşırı kiloya veya tersi şekilde aşırı kilo kayıplarına yol açabilir. Ayrıca düşünme ve kavrama yeteneğinin azalmasına, Alzheimer gibi beyni doğrudan etkileyen hastalıklara neden olabilir.

Beslenme bozukluklarının en sık sebebiyet verdiği hastalık obezitedir. Dünya Sağlık Örgütü (DSÖ) obeziteyi, sağlığı bozacak ölçüde vücutta aşırı yağ birikmesi olarak tanımlamaktadır. Obezite; kalp-damar hastalıkları, yüksek kolesterol, hipertansiyon, inme, diyabet, rahim, meme, prostat ve kalın bağırsak kanseri, reflü gibi gastrointestinal sistem bozuklukları, romatizma, varis, uyku apnesi, doğum zorlukları, yumurtalık kisti ve depresyon gibi hastalıklar için önemli risk faktörüdür (Bahçeci, 2013). Yine DSÖ'ye göre obezite sıklığı 1995'ten 2000 yılına kadar %50 oranında artış göstermiştir. Buna göre, özellikle gelişmiş ülkeler için aşırı kilolu ya da obez birey sayısındaki artış ciddi risk faktörüdür (Bahçeci, 2011). Türkiye'deki obezite sıklığı; kadınlarda %30, erkeklerde %13, genelde ise %22,3 düzeyinde tespit edilmiştir (Satman vd., 2002). Obezitenin sebep olduğu hastalıklara ve bu hastalıkların tedavisi için yapılan tıbbi harcamalara ve buna rağmen bu hastalıklara bağlı gerçekleşen ölüm oranlarının yüksek olmasına bakıldığında; yeterli ve dengeli beslenmenin önemi görülmektedir (Klein ve Romijn, 2008).

Sağlıklı beslenme için, tüm gıdalardan yeterli ve dengeli şekilde alınması ve alınan enerji ile harcanan enerjinin dengede tutulması gerekmektedir. Bireylerin yaşına, cinsiyetine, kilo-boy oranına, yaptığı işe, hamilelik ya da hastalık durumuna göre günlük enerji ihtiyaçları değişkenlik gösterir. Bu ihtiyaca göre bir beslenme planlaması yapmak, insan vücudunu oluşturan hücrelerin düzenli ve dengeli çalışmasını sağlayarak, bireyleri daha sağlıklı kılacaktır. Ayrıca günümüzde sıklıkla rastlanan şişmanlık ve obezite gibi hastalıklardan korunmayı sağlayacaktır (Nişancı Kılınç ve Çağdaş, 2013). Ek olarak, beslenme planlamasının; hamilelik, emzirme gibi özel durumlarda hem annenin sağlığı, hem de çocuğun fiziksel ve zihinsel gelişimi açısından olumlu etkisi olacaktır.

Obeziteden korunmak ve sağlıklı bir yaşam için, bireyin kilo-boy oranının belirli bir düzeyde tutulması şarttır. Bunun kontrolünde beden-kitle indeksi (BKİ) kavramı ön plana çıkmaktadır. BKİ, hesaplaması kolay bir kavram olup, obezite veya şişmanlık seviyesini ölçmek için yaygın olarak kullanılır. BKİ'nin hesaplanması için kullanılan formül Denklem (5.8)'de verilmiştir.

$$BKİ = \text{Ağırlık (kg)} / \text{Boy}^2 (\text{m}^2) \quad (5.8)$$

Örneğin; 1.70 m boyunda, 68 kilo ağırlığında bir birey için $BKİ = 68 / (1.70)^2 = 23.53 \text{ kg/m}^2$ dir (Fuller vd., 1991). Cinsiyete göre ideal BKİ değeri değişkenlik gösterir. Kadınlar için ideal ortalama BKİ değeri 19 - 21 kg/m^2 iken, erkekler için ideal ortalama BKİ değeri 20–22 kg/m^2 dir.

BKİ değerine göre kişilerin zayıf, normal, şişman, obez gibi sınıflandırılması yapılmakta ve buna göre gerekli tedbirler alınmaktadır. Bu sınıflandırma Çizelge 5.7.'de gösterilmektedir.

Çizelge 5.7. BKİ'ye göre şişmanlık durumunun sınıflandırılması (Booth vd., 2000; Eker ve Melih, 2002)

BKİ Değeri (kg/m ²)	Obezite Durumu
< 18.5	Zayıf
18.5 - 24.9	Normal
25 - 29.9	Hafif Şişman
30 - 39.9	Şişman
30 - 34.9	1. derece obez
35 - 39.9	2. derece obez
≥ 40	3. derece obez (ileri derecede obez)

Yeterli ve dengeli beslenmenin gerçekleşebilmesi, tüm besin öğelerinden gerektiği kadar almaya bağlıdır. Besin öğeleri; yağlar, karbonhidratlar, proteinler, vitaminler ve minerallerden oluşur. İnsan vücudunun tüm bu besin maddelerine ihtiyacı vardır. Bu nedenle tek taraflı beslenmek doğru olmaz. Çünkü tek bir grup besin öğesi, vücudun ihtiyaç duyduğu bileşenlerin elde etmesi için yeterli olmaz.

Temel besin öğelerinin, günlük hayatta tükettiğimiz yiyecek ve içecekler içerisindeki dağılımları Şekil 5.10.'daki besin piramidinde görülmektedir. Besin piramidi 5 ana besin grubunu içerir. Piramidin en altında, sıklıkla tüketilmesi gereken besinler bulunurken, yukarıya doğru çıkıldıkça daha az tüketilmesi gereken gıdalar yer almaktadır. Besin piramidi hangi gıdalardan hangi miktarlarda alınması gerektiği konusunda bilgi verir (Eker, 2006).



Şekil 5.10. Besin Piramidi

Her besin maddesi farklı kalori değerine sahiptir. Bu nedenle beslenme çizelgesi uygulamasında, besinlerin kalori değerleri göz önünde bulundurularak, bireylerin günlük alması gereken kalori miktarlarını yakalamak için en uygun besin dağılımı yapılabilir.

Sağlıklı bir bireyin günlük kalori ihtiyacı ortalama 2100 kkal olup yaş, cinsiyet, boy, kilo, aktivite yoğunluğu, hamilelik ve emzirme durumu gibi özelliklerine göre kişiden kişiye değişiklik gösterir. Genel olarak günlük kalori ihtiyacı; bireyin bazal metabolizma hızı, besin sindirimi için harcanan kalori miktarı ve günlük fiziksel aktivite durumuna göre hesaplanır. Bazal metabolizma enerjisi, vücut dinlenme halindeyken, insan yaşamının devamı için gereken minimum enerjiyi belirtmektedir. Bu enerji kalbin atması, soluk alış-verişi, karaciğer işlevlerinin yerine getirilmesi, vücut ısısının sabit tutulması gibi hayati fonksiyonlar için harcanır. Bazal metabolizma enerjisi, bir bireyin günlük aldığı kalori miktarının yaklaşık %60-%70'ine denktir. Kadınların, erkeklere göre daha düşük bazal metabolizma enerjisine ihtiyaçları vardır (Kreymann, 2009).

Bazal metabolizma oranı (BMR) yaygın olarak Harris-Benedict formülasyonu kullanılarak hesaplanır. Eğer birey erkekse, BMR; yaşı, boyu ve kilosuna bağlı olarak Denklem (5.9) ile hesaplanabilir (Harris ve Benedict, 1918; Harris ve Benedict, 1919; Roza ve Shizgal, 1984).

$$BMR = 88.362 + (13.397 \times \text{kilo (kg)}) + (4.799 \times \text{boy (cm)}) - (5.677 \times \text{yaş}) \quad (5.9)$$

Eğer birey kadınsa, BMR; yaşı, boyu ve kilosuna bağlı olarak Denklem (5.10) ile hesaplanabilir.

$$BMR = 447.593 + (9.247 \times \text{kilo (kg)}) + (3.098 \times \text{boy (cm)}) - (4.330 \times \text{yaş}) \quad (5.10)$$

Besinlerin sindirimi için de bir miktar enerji harcanmaktadır. Bu miktar, alınan kalori miktarının %10'una denk düşmektedir. Bu oran bireyin belirgin bir özelliği ile ilişkili görülmediği için, yapılan çalışmada ihmal edilmiştir.

Fiziksel aktivite durumu da günlük kalori ihtiyacında belirleyici bir faktördür. Hareketsiz bir yaşam tarzına sahip bir birey ile fiziksel olarak ağır bir işte çalışan bireyin kalori ihtiyaçları birbirinden farklıdır (Pekcan, 2008). Bireyin cinsiyet, yaş, boy ve kilo durumuna bağlı olarak BMR değeri hesaplandıktan sonra, fiziksel aktivite yoğunluğuna bağlı olarak tahmin edilen enerji gereksinimi (EER1) hesaplanabilir. Bu hesaplama bireylerin egzersiz yoğunluğuna bağlı olarak 5 farklı sınıflandırma ile yapılabilir. Bunlar; çok az egzersiz yapanlar, hafif egzersiz yapanlar, orta seviye egzersiz yapanlar, ağır egzersiz yapanlar ve çok ağır egzersiz yapanlar olarak ayrılabilir. Bu sınıflandırmaya göre; EER1'in

hesaplanması Çizelge 5.8.'de gösterilmiştir. Bu gruplardan ağır egzersiz yapanlar ve çok ağır egzersiz yapanlar, genellikle beden işçileri ve sporcular olarak dikkat çekmektedir.

Çizelge 5.8. Fiziksel aktivite yoğunluğuna göre EER1'in hesaplanması

Fiziksel Aktivite Yoğunluğu	EER1'in Hesaplanması
Çok az egzersiz	$EER1 = BMR \times 1.2$
Hafif egzersiz	$EER1 = BMR \times 1.375$
Orta seviye egzersiz	$EER1 = BMR \times 1.55$
Ağır egzersiz	$EER1 = BMR \times 1.725$
Çok ağır egzersiz	$EER1 = BMR \times 1.9$

Bireyin egzersiz durumuna göre EER1 hesaplandıktan sonra, eğer bireyin cinsiyeti bayansa ve hamilelik durumu söz konusu ise Çizelge 5.9.'daki gibi EER2 hesaplanabilir.

Çizelge 5.9. Hamile bayanların hamilelik dönemine göre EER2'nin hesaplanması

Hamilelik Dönemi	EER2'nin Hesaplanması
1. Trimester (0-3 aylık) dönem	$EER2 = EER1 + 0$
2. Trimester (3-6 aylık) dönem	$EER2 = EER1 + 340$
3. Trimester (6-9 aylık) dönem	$EER2 = EER1 + 452$

Eğer bireyin cinsiyeti bayansa ve emzirme durumu söz konusu ise; bireyin enerji gereksinimi de farklılık gösterir. Bu durumda, bireyin egzersiz durumuna bağlı olarak hesaplanan EER1 kullanılarak, Çizelge 5.10.'daki gibi EER2 hesaplanabilir (Ulusal Tıp Akademileri Enstitüsü, 2005).

Çizelge 5.10. Çocuk emziren bayanların doğum sonrası dönemine göre EER2'nin hesaplanması

Emzirme Dönemi	EER2'nin Hesaplanması
Doğum sonrası 0-6 ay	$EER2 = EER1 + 330$
Doğum sonrası 7-12 ay	$EER2 = EER1 + 400$

Yukarıda verilen hesaplamalardan da görüldüğü gibi, kişilerin kalori ihtiyacı fiziksel, fizyolojik ve sosyolojik özelliklerine göre farklılık gösterir. Her bireyin, vücudunun ihtiyaç duyduğu kalori gereksinimini göz önünde bulundurarak bir beslenme planlaması yapması, sağlıklı bir yaşam sürdürmesi için şarttır. Bireyin, vücudunun ihtiyaç duyduğu oranda enerjiyi, farklı kalori değerlerine sahip çeşitli yiyeceklerden alabilmesi için optimum beslenme çizelgesinin hazırlanması gerekmektedir. Bu karmaşık optimizasyon probleminin çözümünde, sezgisel tekniklere dayanan ve sürekli optimizasyon problemleri için iyi sonuçlar veren A³ yönteminin kullanılması mümkündür.

5.2.1. A³ ile beslenme optimizasyonu

Bireylerin fiziksel, fizyolojik ve sosyolojik özelliklerine göre belirlenen günlük kalori gereksinimlerine uygun olarak, optimum beslenme çizelgesinin oluşturulması için A³ yöntemi kullanılmıştır. Uygulama MATLAB 2013a yazılımı kullanılarak Intel Core-i3 3.3 GHz işlemcili PC üzerinde gerçekleştirilmiştir. Geliştirilen yazılım ile kullanıcıdan cinsiyet, yaş, boy, kilo, egzersiz yoğunluğu, eğer cinsiyet bayan ise hamilelik durumu ve dönemi ile emzirme durumu ve dönemi bilgileri istenmektedir. Alınan bilgilere göre bireyin günlük enerji gereksinimi hesaplanmaktadır. Uygulama rastgele seçilen 20 birey için denenmiştir. Uygulama için kahvaltılıklar, çorbalar, ana yemekler, tatlılar, kuruyemişler, meyveler ve içeceklerden oluşan 158 adet besin ve bu besinlere ait kalori değerleri kullanılmıştır (Baysal vd., 1991; Anonim, 2016; Anonim, 2017). Bu besinlerin 100 gramları için kalori değerleri Ek 1-7’de verilmiştir. Besin gruplarının günlük tüketilme oranları dikkate alınarak her besin grubu için tercih katsayısı belirlenmiştir (Ek 8). Bu katsayılar kullanılarak optimum beslenme çizelgesi uygulaması gerçekleştirilmiştir. Böylece kuruyemiş gibi gün içerisinde az tüketilen besinlerin, beslenme optimizasyonu uygulamasında daha düşük gramajlı seçilmesi sağlanmıştır. Benzer şekilde, ana yemeklerin tercih katsayısı ise daha yüksek belirlenmiştir.

A³ ile beslenme optimizasyonu probleminin çözümünde ilk olarak, Harris-Benedict formülasyonu kullanılarak, kullanıcının girdiği cinsiyet, yaş, boy ve kilo bilgileri ile bireyin bazal metabolizma oranı (BMR) hesaplanmıştır. Ardından, BMR değeri kullanılarak kişinin fiziksel aktivite yoğunluğuna göre tahmin edilen günlük enerji gereksinimi (EER1) hesaplanmıştır. Bu aşamadan sonra, eğer birey bayansa kullanıcıya hamilelik durumu olup, olmadığı ve varsa dönemi soruları yöneltilmektedir. Kullanıcıdan alınan bilgilere göre, EER1 değeri kullanılarak EER2 hesaplanmıştır. Bu durumda EER2 hamile bayanın enerji gereksinimi göstermektedir. Eğer birey bayansa ve hamile değilse, o zaman kullanıcıya emzirme durumu olup olmadığı ve varsa dönemi soruları yöneltilmektedir. Yine kullanıcıdan alınan bilgilere göre, EER1 değeri kullanılarak, bu sefer emziren bayanın enerji gereksinimi (EER2) hesaplanmıştır. Sonuç olarak yapılan hesaplamalar neticesinde, bireylerin fiziksel,

fizyolojik ve sosyolojik özelliklerine göre günlük tahmin edilen enerji gereksinimleri hesaplanmaktadır.

A^3 ile beslenme optimizasyonu problemi için amaç fonksiyonu, Denklem (5.11)'deki gibi belirlenmiştir.

$$\min f(x) = \left| \sum_{i=1}^{\beta n} (x_i) - EER \right| \quad (5.11)$$

Burada n , A^3 ile beslenme optimizasyonu uygulaması için belirlenen elektron sayısını göstermektedir. x_i ise, karar değişkenlerini belirtmektedir. Bu problem için karar değişkenleri, bireyin günlük enerji gereksinimini karşılamak için alması mümkün olan besin maddeleridir (Arora, 1989). Bu problem için gün boyu 12 farklı besin tüketileceği düşünülmüştür. Bu nedenle $n = 20$ olarak belirlenmiştir. (8 besin maddesi, iyonik alanda kullanılır. İyonik alandakiler sonuca dahil edilmezler.)

Beslenme optimizasyonu problemi için A^3 yöntemi uygulanırken, Çizelge 5.11.'deki parametre değerleri belirlenmiştir.

Çizelge 5.11. A^3 ile beslenme optimizasyonu için belirlenen parametre değerleri

Parametre	Değeri
Elektron sayısı	20
Atom sayısı	100
Kovalent oran (β)	0.6
İyonik oran (α)	0.4
İterasyon sayısı	100

A^3 ile beslenme optimizasyonu probleminde atom kümesi, Matlab'ın randperm() fonksiyonu kullanılarak rastgele seçilen besinlerin kalori değerlerinden oluşmaktadır. Ayrıca besinlerin isimlerinin tutulduğu bir matris daha mevcuttur. A^3 operatörleriyle, besinlerin kalori değerleri arasındaki karşılaştırma sonucunda atom kümesinde yapılan değişiklikler, aynı şekilde isimlerin tutulduğu matriste de yapılmaktadır. Şekil 5.11.'de beslenme optimizasyonu problemi için atom kümesinin temsili bulunmaktadır.

	Kovalent Alan					İyonik Alan	
A_1	Kestane 245 kkal	Enginar 48 kkal		Dana eti 187 kkal	Kivi 30 kkal		Süt 50 kkal
A_2	Kavurma 670 kkal	Ayran 37 kkal		Kiraz 70 kkal	B.Peynir 289 kkal		Dondurma 193 kkal
	.	.	.	.	.	.	.
A_{r+1}	Yumurta 158 kkal	Levrek 93 kkal		Şeftali 38 kkal	Profiterol 236 kkal		Portakal 49 kkal
A_{r+2}	Brokoli 35 kkal	İhlamur 1 kkal		Reçel 272 kkal	Tavuk 215 kkal		Ceviz 651 kkal
	.	.	.	.	.	.	.

Şekil 5.11. A^3 ile beslenme optimizasyonu problemi için atom kümesinin temsili

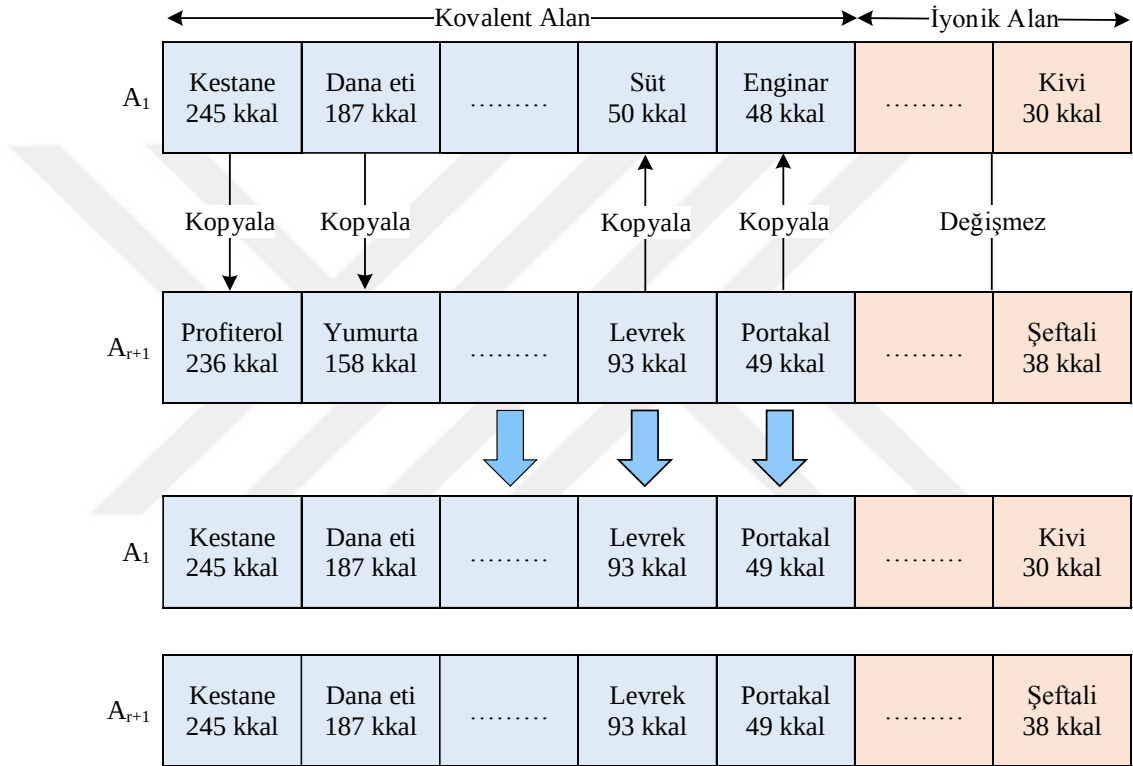
Atom kümesi şekildeki gibi rastgele değerlerle oluşturulduktan sonra, bireyin enerji gereksinimiyle besinlerin toplam kalori değerleri arasındaki farka göre (yani amaç fonksiyonuna göre) atom kümesi ve her atom kendi içerisinde sıralanır. Eğer bireyin enerji gereksinimi, atomun kovalent alanındaki besinlerin toplam kalori değerinden büyükse, kovalent alandaki elektronlar büyükten küçüğe doğru sıralanır; aksi halde küçükten büyüğe doğru sıralanır. Yani $\sum_{i=1}^{\beta n}(x_i) - EER < 0$ ise, elektronlar büyükten küçüğe doğru sıralanır. $\sum_{i=1}^{\beta n}(x_i) - EER > 0$ ise, elektronlar küçükten büyüğe doğru sıralanır. Şekil 5.12.'de $\sum_{i=1}^{\beta n}(x_i) - EER < 0$ olduğu varsayılmakta ve beslenme optimizasyonu problemi için atom kümesinin sıralanmasına bir örnek verilmektedir.

A_1	Kestane 245 kkal	Enginar 48 kkal		Dana eti 187 kkal	Kivi 30 kkal		Süt 50 kkal
				↑			
				Büyükten Küçüğe Sırala			
				↓			
A_1	Kestane 245 kkal	Dana eti 187 kkal		Süt 50 kkal	Enginar 48 kkal		Kivi 30 kkal

Şekil 5.12. A^3 ile beslenme optimizasyonu problemi için atom kümesinin sıralanması

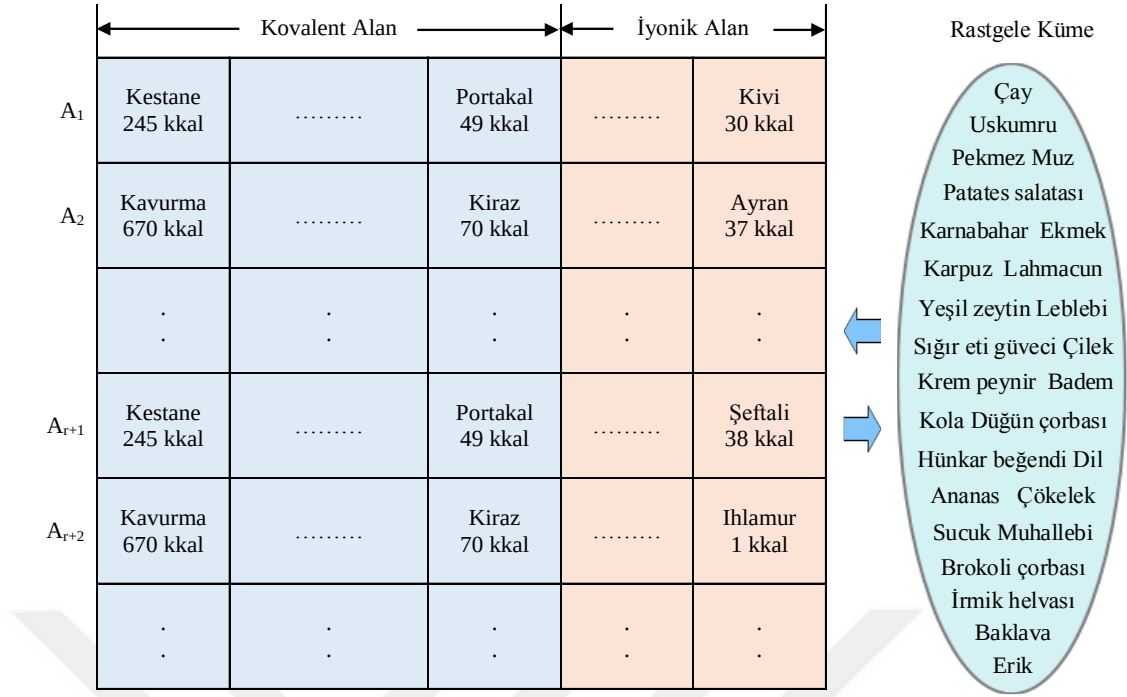
Atom kümesindeki her atom kendi içerisinde sıralandıktan sonra, kovalent bağ ve iyonik bağ operatörleri uygulanır. Böylece, besin listelerindeki toplam kalori değerlerinin, bireyin enerji gereksinimine yakın değerlere ulaşması sağlanır.

Kovalent bağ operatörünün temel mantığı, ikişerli olarak eşleştirilen atomların kovalent alan içerisindeki elektronlarının etki değerlerine göre karşılaştırılarak, elektron etkisi büyük olanın küçük olan üzerine kopyalanmasıdır. A^3 ile beslenme optimizasyonu problemi için kovalent bağ operatörünün işleyişi $\sum_{i=1}^{\beta n}(x_i) - EER < 0$ olduğu durumda, Şekil 5.13.'de gösterildiği gibidir.



Şekil 5.13. A^3 ile beslenme optimizasyonu problemi için kovalent bağ operatörünün işleyişi

İyonik bağ operatörü, atom kümesinin iyonik alanına uygulanır. Bu alanda amaç fonksiyonu içerisinde değerlendirilmeyen karar değişkenleri bulunmaktadır. Şekil 5.14.'de görüldüğü gibi, beslenme optimizasyonu problemi için iyi olarak değerlendirilen besinler seçilerek kovalent alana alındıktan sonra, iyonik alanda kalan besinler rastgele yeni değerlerle değiştirilir. Böylece, iyonik bağ operatörüyle global optimuma ulaşmak mümkün olur (Yıldırım ve Karcı, 2015).



Şekil 5.14. A^3 ile beslenme optimizasyonu problemi için iyonik bağ operatörünün işleyişi

Beslenme optimizasyonu problemi için A^3 'ün algoritma adımları aşağıdaki gibi güncellenmiştir.

1. Atom kümesini, seçilen besinlerin kalori değerlerini içeren şekilde randperm() fonksiyonuyla oluştur
2. Atom kümesinde bulunan kalori değerlerine ait besinlerin isimlerini de aynı sırayla başka bir matriste tut
3. Elektron etki değerlerini hesapla
4. Amaç fonksiyonu değerlerini hesapla
5. Atom kümesini amaç fonksiyon değerlerine göre sırala
6. Atomları kendi içerisinde elektronların etki değerlerine göre sırala
7. Algoritma belirlenen iterasyon sayısına ulaşıncaya kadar aşağıdaki adımları yürüt
 - a) Kovalent bağ operatörünü uygula (değişiklikleri besinlerin isimlerinin tutulduğu matrise de uygula)
 - b) İyonik bağ operatörlerini uygula (değişiklikleri besinlerin isimlerinin tutulduğu matrise de uygula)
 - c) Elektron etki değerlerini hesapla
 - d) Amaç fonksiyon değerlerini hesapla
 - e) Atom kümesini amaç fonksiyon değerlerine göre sırala
 - f) Atomları kendi içerisinde elektronların etki değerlerine göre sırala

5.3. Grup Asansör Kontrol Optimizasyonu Problemi

Asansörler günümüzde yaygın olarak kullanılan sistemlerdir. Bina içerisinde dikey yönde taşıma yaparlar. Yük ve yolcu taşımak gibi çeşitli amaçlarla kullanılmaktadırlar. Özellikle çok katlı ve kalabalık binalarda vazgeçilmez bir teknoloji halini almışlardır.



Şekil 5.15. Bir binada çoklu asansör kullanımı

Asansörlerin kullanımı yaygınlaştıkça, daha etkin kullanılabilmesi de önem kazanmıştır. Mesela taşıma kapasitesinin artırılması, daha hızlı ve güvenli çalışması, kat içerisinden yapılan çağrılara daha çabuk cevap verebilmesi gibi özellikleri önemli hale gelmiştir. Kısacası, gelişen teknolojiyle birlikte asansörlerin servis kalitesinin artırılması beklenmektedir. Asansörün servis kalitesini, asansörün fiziksel özellikleri etkilemektedir. Binanın büyüklüğüne göre yükleme ve taşıma kapasitesi, katlar arası seyahat hızı, durma ve kalkma zamanı, kapısının açılması ve kapanması için geçen süreler asansörün fiziksel özellikleridir ve servis kalitesinde belirleyici etkisi bulunmaktadır.

Fiziksel özelliklerinin yanı sıra, asansörlerin çağrı toplamada kullandıkları stratejiler de önemlidir. Özellikle tek asansörlü sistemlerde çağrı toplama stratejileri asansörün etkin çalışması açısından önemlidir. Bu amaçla 4 metot kullanılmaktadır. Bunlar; basit çağrı toplama, yukarı doğru çağrı toplama, aşağı doğru çağrı toplama, çift yönlü çağrı toplama. Basit çağrı toplama, asansörün herhangi bir süzgeçten geçirmeden, gelen her kat çağrısına sırayla cevap vermesidir. Diğer yöntemlerde ise, trafik yoğunluğunun olduğu yöne doğru çağrı toplama stratejisi belirlenir. Yukarı yönlü çağrı toplamada, asansör gelen çağrıları yukarıya doğru sırayla listesine eklerken; aşağı yönlü çağrı toplamada da aynı mantıkla,

gelen çağrılarını cevaplamak üzere aşağı doğru sırayla listesine alır. Çift yönlü toplama, trafik akış yoğunluğu iki yönde de hemen hemen aynıysa kullanılan sistemlerdir (Ayaz, 2012).

Asansörler, kullanım amacına göre yük ve yolcu taşıyan asansörler olarak sınıflandırılabilirler. 8 veya daha az katlı binalarda yolcu taşıyan asansörlerde, hidrolik ya da elektriksel sistemler kullanılırlar. Hidrolik asansörler 1 m/sn 'ye kadar hızlanabilirler. Oysa elektrikli asansörler 2.5 m/sn 'ye kadar hızlanabilirler. 8 ile 10 kat arasındaki binalarda, elektriksel ve aynı zamanda vitessiz sistemler tercih edilirler. Elektrikli ve vitessiz asansörler de ancak 2.5 m/sn 'ye kadar hızlanırlar. 10 kattan daha yüksek binalarda asansör hızının çok daha yüksek olması beklenir. Genellikle binanın kat sayısına göre, 10 m/sn 'ye varan bir hızlanma mümkün olur (Barney ve dos Santos, 1976; Siikonen, 1997a; Yong, 2008). Asansör hızı belli bir seviyeye kadar yükseltilebilir. Bu nedenle servis kalitesi açısından, çağrı cevaplamada izlenecek stratejiler önem kazanmaktadır.

Binaların büyüklüğüne ya da yoğunluğuna göre içinde bulundurduğu asansör sayıları değişkenlik göstermektedir. Örneğin, az sayıda dairenin bulunduğu 5-6 katlı binalarda tek asansör yeterli olurken, hastaneler gibi aciliyeti olan binalarda, oteller, alışveriş merkezleri gibi yoğun binalarda ve gökdelenler gibi yüksek katlı binalarda birden çok asansör hizmet vermektedir. Yani asansörler, içinde bulunduğu binanın içerdiği asansör sayısına bağlı olarak da sınıflandırılabilirler. Buna göre; tek asansörlü ve birden çok asansörlü sistemler olarak ikiye ayrılırlar (Şekil 5.15). Çok asansörlü sistemler iki şekilde çalışabilirler. Birincisi, asansörlerin kendi aralarında etkileşimli çalışmalarını sağlayan bir kontrol mekanizması olmadan, birbirinden bağımsız olarak çağrılara cevap veren sistemlerdir. İkincisi ise, asansörlerin aralarında koordinasyonu sağlayan bir kontrol mekanizmasının bulunduğu sistemlerdir. Bu sistemlere grup asansör kontrol sistemleri denilmektedir.

Birden çok asansörün bulunduğu kalabalık binalarda, asansörlerin kendi içerisinde koordineli şekilde çalışması oldukça önemlidir. Örneğin, bir kat çağrısı olduğu zaman, ona cevap verecek asansörün doğru şekilde belirlenmesi ve diğer asansörlerin bu çağrıyla meşgul edilmemesi gerekmektedir. Bu nedenle birden çok asansörün hizmet verdiği binalar için, grup asansör kontrol optimizasyonu önemli bir problem olarak görülmektedir. Bu problemin etkili bir şekilde çözümü, ancak uygun bir kontrol algoritmasıyla mümkün olur (Barney ve Santos, 1976). Uygun bir kontrol algoritmasının kullanılması, asansörün fiziksel özellikleri iyileştirilmeden, servis kalitesinin geliştirilmesini sağlar (Siikonen, 1997a).

Yapay zeka optimizasyon algoritmaları, günlük hayatta karşılaşılan birçok optimizasyon problemi için etkin çözümler sunarlar. Grup asansör kontrol optimizasyon probleminin çözümü için de kullanılabilirler. Bu problemin çözümünde daha önce farklı yapay zekâ algoritmaları kullanılmıştır. Bunlardan bazıları; bulanık mantık, genetik algoritmalar, DNA hesaplama, PSO, AYBS, KKO ve takviyeli öğrenme yöntemidir (Siikonen, 1997b; Kim vd., 1998; Crites ve Barto, 1998; Bolat, 2006; Gudwin ve Gomide, 1994; Cortes vd., 2004;

Muhammad vd., 2005; Fei vd., 2010; Zhang vd., 2010; Baygın ve Karaköse, 2011). Asansör trafik kontrolünde yapay zekâ yöntemlerinin kullanılması, çağrı cevaplama sürelerinde belirgin iyileşme sağlamıştır (Ayaz, 2012). Bu yöntemlerin yanı sıra, optimizasyon problemlerinin çözümü için etkili sonuçlar veren bir meta-sezgisel yaklaşım olan A^3 'ün, grup asansör kontrol optimizasyonu probleminin çözümü için de iyi sonuçlar vermesi mümkündür.

5.3.1. A^3 ile grup asansör kontrol optimizasyonu uygulaması

A^3 ile grup asansör kontrol optimizasyonu, 7 asansörün bulunduğu 16 katlı bir binanın asansör sistemi için gerçekleştirilerek, optimum sonuca ulaşma hedeflenmiştir. Uygulama MATLAB 2013a yazılımı kullanılarak Intel Core-i3 3.3 GHz işlemcili PC üzerinde gerçekleştirilmiş ve elde edilen sonuçlar, daha önce aynı problemin çözümünde kullanılan başka bir yapay zekâ algoritması olan AYBS yöntemiyle karşılaştırılmıştır.

Asansör sistemlerinde iki tip çağrı bulunmaktadır. Bunlar kat çağrısı ve kabin çağrısıdır. Kat çağrısı, asansör kabininin dışındaki bir kullanıcının asansöre binmek için, asansör çağırma butona basmasıyla oluşan çağrıdır. Diğeri ise kabin çağrısıdır ki, bu da çağırıldığı asansörün gelmesiyle birlikte asansör kabinine giren kullanıcının, gitmek isteği katı belirten butona basmasıyla oluşan çağrıdır. Bu uygulama sırasında kat çağrıları dikkate alınmıştır. 16 kat içerisinde farklı konumlarda bulunan kullanıcıların 10 adet kat çağrısı yaptığı farz edilmiştir. Asansörlerin başlangıç konumları ve kullanıcılar tarafından yapılan kat çağrıları Şekil 5.16.'da verilmiştir.

ASANSÖR KAT	1	2	3	4	5	6	7	KAT ÇAĞRISI
16								
15								
14								
13								
12								
11								
10								
9								
8								
7								
6								
5								
4								
3								
2								
1								

 .. Kabin Konumu
  .. Kat Çağrısı

Şekil 5.16. Asansör kabinlerinin başlangıçtaki konumları ve kat çağrıları (Baygın ve Karaköse, 2011)

A³ kullanılarak gerçekleştirilen grup asansör kontrol optimizasyonu uygulamasında, optimizasyon algoritmasının başarısını etkilemediği için, asansörün ve binanın fiziksel özellikleri sabit kabul edilmiştir. Binanın ve asansörün fiziksel özellikleri Çizelge 5.12.'de gösterilmektedir. Binanın fiziksel özellikleri; binada bulunan kat ve asansör sayısıdır. Asansörün fiziksel özellikleri ise; bu problem için asansörün katlar arası geçiş süresi ve asansör kapısının açılış/kapanış süresi olarak belirlenmiştir (Baygın ve Karaköse, 2011).

Çizelge 5.12. Binanın ve asansörün fiziksel özellikleri

Özellik	Değeri
Binadaki kat sayısı	16
Binada bulunan asansör sayısı	7
Asansör kapısının ortalama açılma/kapanma süresi (sn.)	4
Asansörün katlar arasındaki ortalama geçiş süresi (sn.)	3

A^3 ile grup asansör kontrol optimizasyonu için, elektronların amaç fonksiyon üzerindeki etki değerleri Denklem (5.12)'ye göre hesaplanmıştır.

$$E[A(k)] = [| c_p(i) - c_t(i) |] * t_T * \alpha_1 + t_D * \alpha_2 \quad // i = \{1, 2 \dots m\}$$

$$// k = \{1, 2 \dots n\} \quad (5.12)$$

Burada, A bir atom olmak üzere; $A(k)$, A atomunun k . elektronunu belirtir. $E[A(k)]$ ise, $A(k)$ elektronunun etki değerini göstermektedir. Grup asansör kontrol optimizasyonu problemi için, her kat çağrısı birer elektron ile ifade edilmiştir. Yani; n , toplam kat çağrısını göstermektedir. Ayrıca bu denklemde, c_p asansör kabininin başlangıçta bulunduğu katı, c_t ise asansör kabininin hedef katını göstermektedir. Buna göre; i , asansör indeksini tutmakta, m ise, binadaki asansör sayısını belirtmektedir. t_T , kabinin katlar arasındaki ortalama seyahat süresini; t_D ise, kabin kapısının ortalama açılış/kapanış süresini göstermektedir. Ayrıca amaç fonksiyonu hesaplanırken, α_1 ve α_2 gibi iki ağırlık katsayısı kullanılmaktadır. Bu problem için $\alpha_1 = 0.8$ ve $\alpha_2 = 0.2$ olarak belirlenmiştir (Baygın ve Karaköse, 2011).

A^3 ile grup asansör kontrol optimizasyonu için amaç fonksiyonu Denklem (5.13)'de verilmektedir.

$$\min \sum_{i=1}^m ([| c_p(i) - c_t(i) |] * t_T * \alpha_1 + t_D * \alpha_2) \quad // i = \{1, 2 \dots m\} \quad (5.13)$$

Grup asansör kontrol optimizasyonu problemi için A^3 'ün parametre değerleri Çizelge 5.13.'deki gibi belirlenmiştir.

Çizelge 5.13. A^3 ile grup asansör kontrol optimizasyonu için parametre değerleri

Parametre	Değeri
Elektron sayısı	10
Atom sayısı	100
Kovalent oran (β)	0.5
İyonik oran (α)	0.5
İterasyon sayısı	1000

A^3 kullanılarak grup asansör kontrol optimizasyonu probleminin çözümüyle, kat çağrılarına en uygun sırayla cevap verecek asansör listesi elde edilmeye çalışılmıştır. Problem için elektron sayısı, kat çağrılarının sayısı kadar belirlenmiştir. Her kat çağrısına karşılık, ona cevap verecek asansör numarası tanımlanmıştır. Bu işlem atom kümesinin başlangıçta 1 ile 7 arasında rastgele değerlerle doldurulmasıyla gerçekleştirilmiştir. Bir asansör birden çok kat çağrısına cevap verebileceği için, her atom tekrarlı bir permütasyon belirtmektedir.

Grup asansör kontrol optimizasyonu uygulamasında amaç fonksiyonun değeri, tüm asansörlerin kendilerine düşen kat çağrılarını cevaplamak için geçen sürelerin toplamı şeklinde hesaplanır. Bu nedenle, eğer bir asansör birden çok kat çağrısını cevaplayacaksa, uğradığı ilk kattan sonra, her adımda asansörün başlangıç konumunun güncellenmesi gerekmektedir. Bu çalışmada, asansör her kat çağrısını cevapladığında bu güncelleme gerçekleştirilmiştir. Böylece amaç fonksiyonun değerinin doğru şekilde hesaplanması sağlanmıştır.

A^3 ile grup asansör kontrol optimizasyonu uygulaması gerçekleştirilirken, algoritma adımlarında problemin özelliklerine bağlı olarak bazı değişiklikler yapılmıştır. Buna göre, yapılan değişikliklerle aşağıdaki algoritma adımları yürütülmüştür:

1. Elektronlar, asansör indeksini belirtecek şekilde; tekrarlı ve rastgele değerlerle atom kümesini oluştur
2. Her atomun amaç fonksiyon değeri hesapla
3. Her elektronun amaç fonksiyon üzerindeki etki değeri hesapla
4. Belirlenen iterasyon sayısına ulaşıncaya kadar aşağıdaki adımları yürüt
 - a) İyonik alana, iyonik bağ operatörü uygula
 - b) Elektron etkilerini ve amaç fonksiyonu değerlerini hesapla
 - c) Kovalent alana, kovalent bağ operatörü uygula
 - d) Atom kümesini amaç fonksiyonun değerine göre sırala

Kovalent oran (β) ve iyonik oran (α) eşit olarak ($\alpha = \beta = 0.5$) belirlenmiş ve problem yapısına daha uygun olduğu için, atom kümesinin ilk yarısı kovalent alan, ikinci yarısı ise iyonik alan olarak tanımlanmıştır. Kovalent bağ operatörü uygulanırken, yapılan değişikliklerin amaç fonksiyon üzerinde olumlu etkisi olup olmadığına dikkat edilmiştir. Eğer amaç fonksiyon değeri kötüye gidiyorsa, değişiklik gerçekleştirilmemiştir. Bu şekilde, A³ yöntemi kullanılarak bir grup asansör kontrolü probleminin optimizasyonu yapılmıştır (Yıldırım ve Karcı, 2017a).



6. BULGULAR

6.1. GSP için Deneysel Sonuçlar

Ayrık bir problem olan GSP, literatürde sıklıkla kullanılan ve optimizasyon algoritması için başarı ölçütü sayılabilecek bir NP-zor problemidir. Bu nedenle GSP, A³ algoritmasının başarısını ispatlamak adına önemli bir kombinasyonel optimizasyon problemidir.

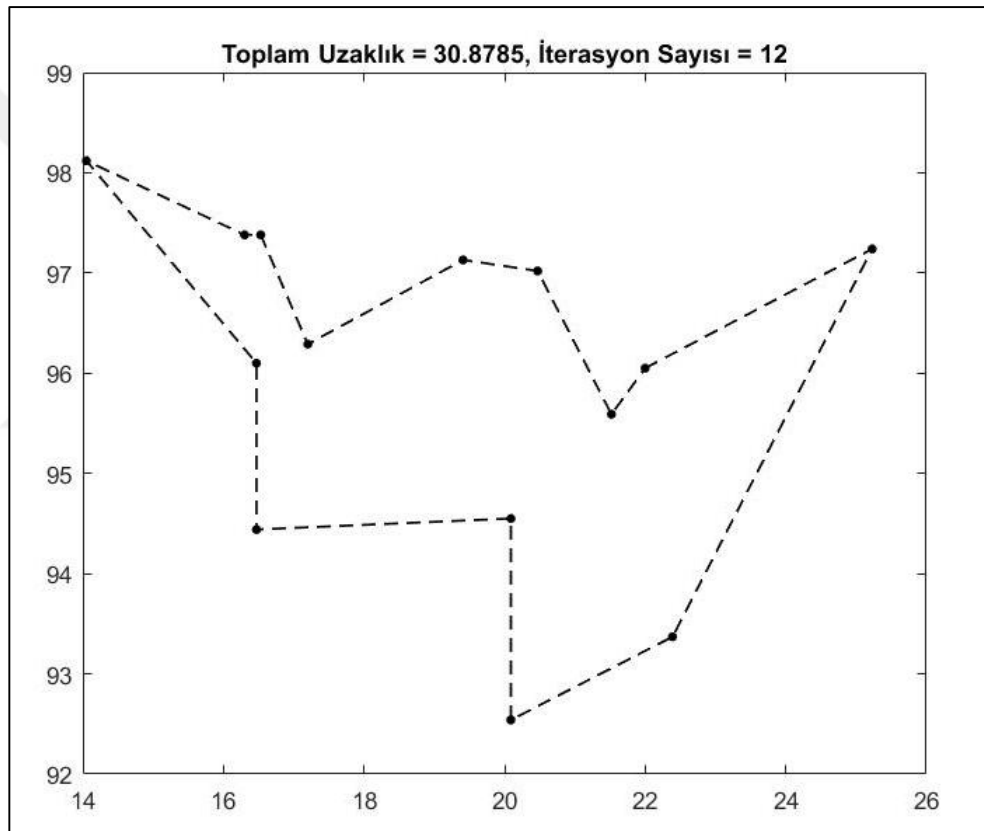
Bu tez çalışmasında, A³ yönteminin uygulanacağı ilk ayrık problem olarak GSP seçilmiştir. Bu yöntem, literatürde sıklıkla kullanılan 8 adet küçük ölçekli ve simetrik GSP için uygulanmıştır. Bunlar; şehir sayıları sırasıyla 14, 29, 29, 48, 51, 52, 70, 76 olan burma14, bayg29, wi29, att48, eil51, berlin52, st70, pr76 problemleridir. Yazılım, MATLAB 2010b kullanılarak Intel Core-i3 3.3 GHz işlemcili PC üzerinde gerçekleştirildi. Uygulama, her veri seti için 20 kez çalıştırılarak test edildi ve en iyi, en kötü ve ortalama sonuçlar elde edildi. A³'ün uygulandığı GSP'lerdeki başarısı Çizelge 6.1.'de verilmiştir. Burada optimum çözüm, TSPLib kütüphanesine göre belirlenen optimum rota için hesaplanan şehirler arası mesafeler toplamıdır. Yani problemin çözümü için en kısa yoldur. En iyi durum, algoritmanın 20 kez çalıştırılması sonucu ulaşılan en iyi sonucu gösterir. En kötü durum, algoritmanın 20 kez çalıştırılması sonucunda ulaşılan en kötü sonucu gösterir. Ortalama ise, algoritmanın 20 kez çalıştırılmasıyla elde edilen sonuçların ortalamasıdır. Fark, en iyi durumla optimum çözüm arasındaki farkı belirtir. Hata (%) ise, en iyi durumla optimum çözüm arasındaki farkın %'lik değerini gösterir.

Çizelge 6.1. A³'ün uygulandığı GSP'ler için performans analizi

GSP Örnekleri	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
burma14	14	30.8785	30.8785	30.8785	30.8785	0.0	0.0
bayg29	29	9,073	9,073	9,073	9,073	0.0	0.0
wi29	29	27,603	27,603	27,603	27,603	0.0	0.0
att48	48	33,522	33,522	33,783	33,655	0.0	0.0
eil51	51	426	426	431	428.2	0.0	0.0
berlin52	52	7,542	7,542	7,618	7,563.1	0.0	0.0
st70	70	675	675	682	679.5	0.0	0.0
pr76	76	108,159	108,339	110,343	109,47	180	0.166

Bu çalışmada, şehirler arasındaki uzaklıklar Öklidyen uzaklık metodu kullanılarak hesaplandı. Ardından elde edilen sonuçlar en yakın tam sayıya yuvarlanarak, TSPLib kütüphanesindeki optimum sonuçlarla karşılaştırıldı. Çünkü TSPLib kütüphanesindeki optimum rotalara göre, şehirler arası uzaklıklar tamsayıya çevrilerek toplamı hesaplandığında, optimum çözüm elde edilmiştir. Yani TSPLib kütüphanesindeki optimum çözümler de, şehir uzaklıklarının tamsayıya yuvarlandığı varsayılarak hesaplanmıştır. (Shang, 2005; Xu vd., 2012).

A³ kullanılarak gerçekleştirilen GSP uygulamalarından burma14 için en iyi rotayı gösteren grafik Şekil 6.1.'de verilmiştir. A³ ile burma14 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.2.'de verilmiştir.

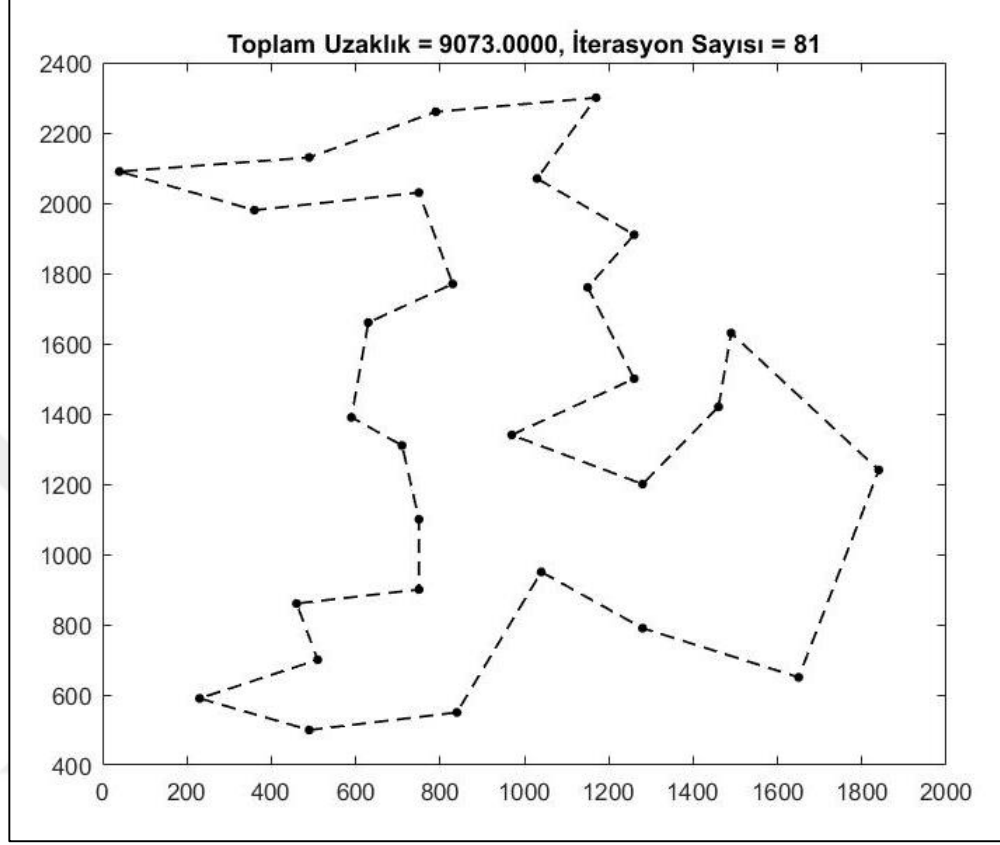


Şekil 6.1. A³ ile burma14 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.2. A³ ile burma14 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	30.8785
En iyi rota	13 → 7 → 12 → 6 → 5 → 4 → 3 → 14 → 2 → 1 → 10 → 9 → 11 → 8 →

A^3 kullanılarak gerçekleştirilen GSP uygulamalarından bayg29 için en iyi rotayı gösteren grafik Şekil 6.2.'de verilmiştir. A^3 ile bayg29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.3.'de verilmiştir.

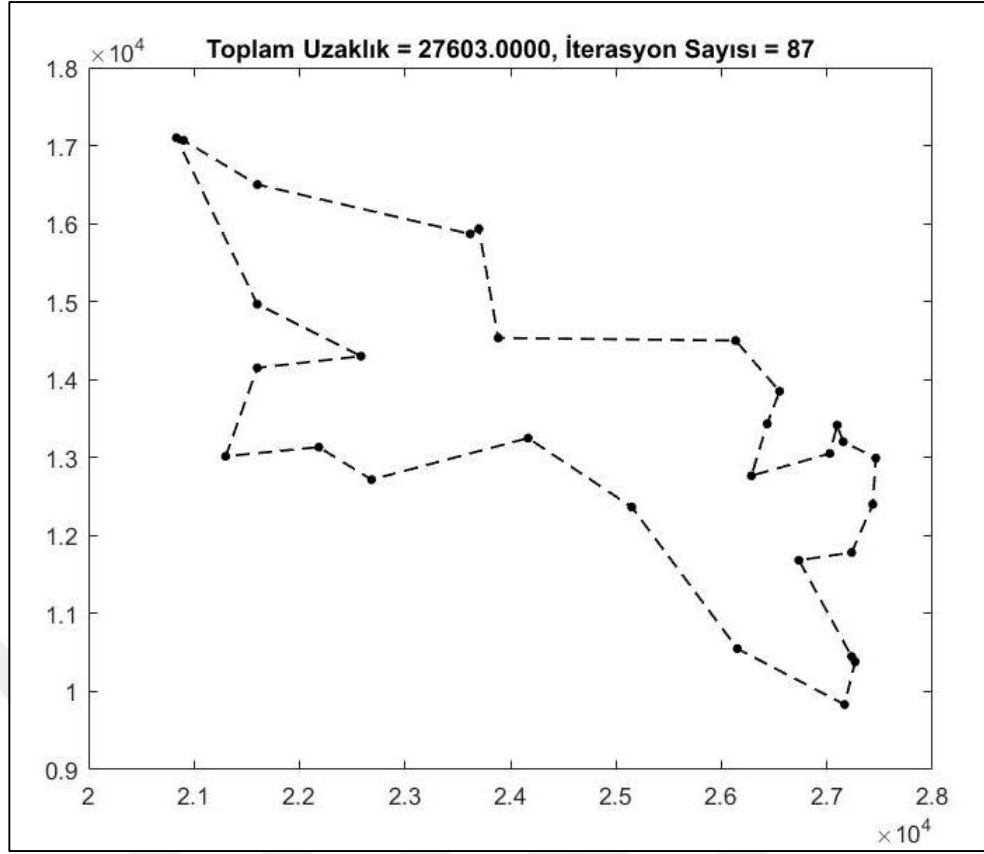


Şekil 6.2. A^3 ile bayg29 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.3. A^3 ile bayg29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	9,073
En iyi rota	27 → 8 → 23 → 7 → 25 → 19 → 11 → 22 → 17 → 14 → 18 → 15 → 4 → 10 → 20 → 2 → 21 → 5 → 29 → 3 → 26 → 9 → 12 → 6 → 28 → 1 → 24 → 13 → 16 →

A^3 kullanılarak gerçekleştirilen GSP uygulamalarından wi29 için en iyi rotayı gösteren grafik Şekil 6.3.'de verilmiştir. A^3 ile wi29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.4.'de verilmiştir.

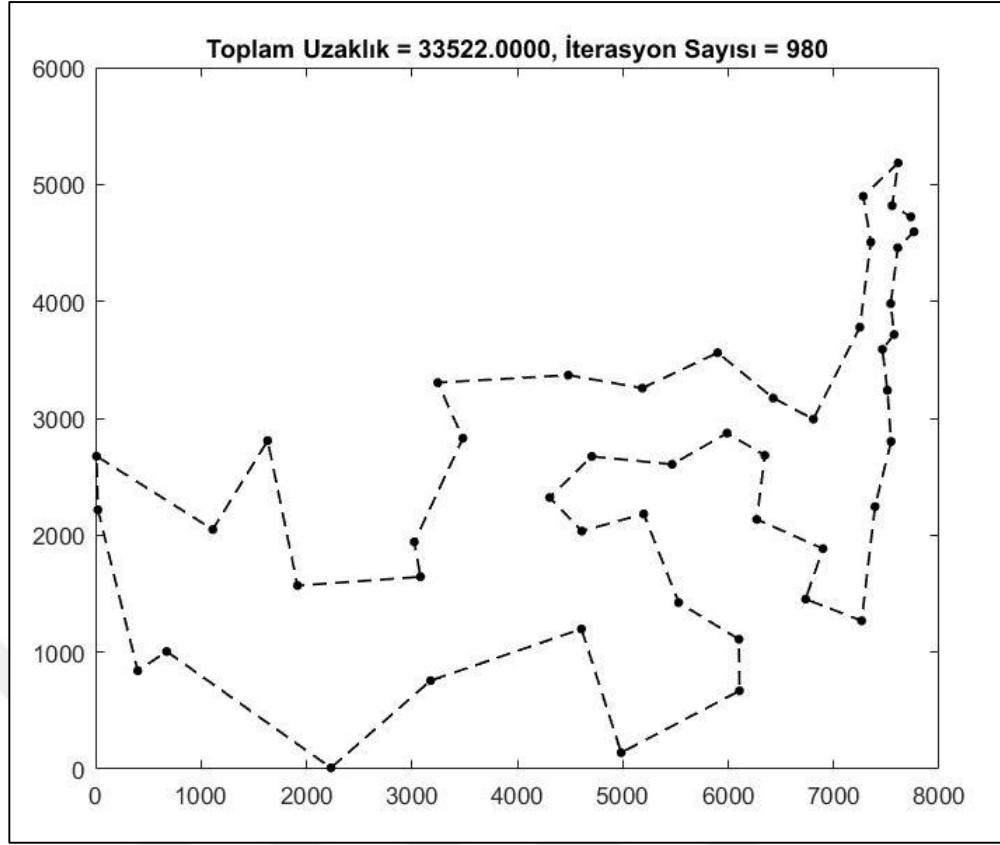


Şekil 6.3. A³ ile wi29 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.4. A³ ile wi29 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	27,603
En iyi rota	1 → 2 → 6 → 10 → 11 → 12 → 15 → 19 → 18 → 17 → 21 → 22 → 23 → 29 → 28 → 26 → 20 → 25 → 27 → 24 → 16 → 14 → 13 → 9 → 7 → 3 → 4 → 8 → 5 →

A³ kullanılarak gerçekleştirilen GSP uygulamalarından att48 için en iyi rotayı gösteren grafik Şekil 6.4.'de verilmiştir. A³ ile att8 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.5.'de verilmiştir.

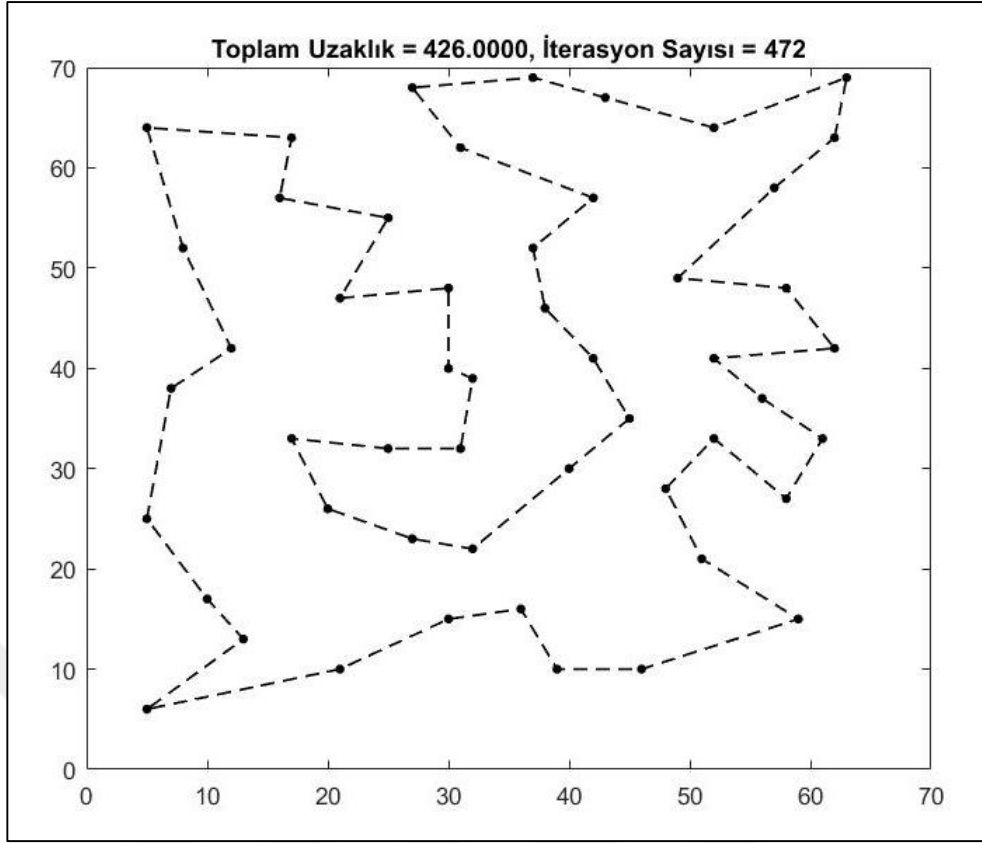


Şekil 6.4. A^3 ile att48 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.5. A^3 ile att48 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	33,522
En iyi rota	47 → 20 → 33 → 46 → 36 → 30 → 43 → 17 → 27 → 19 → 37 → 6 → 28 → 7 → 18 → 44 → 31 → 38 → 8 → 1 → 9 → 40 → 15 → 12 → 11 → 13 → 25 → 14 → 23 → 3 → 22 → 16 → 41 → 34 → 29 → 2 → 26 → 4 → 35 → 45 → 10 → 24 → 42 → 5 → 48 → 39 → 32 → 21 →

A^3 kullanılarak gerçekleştirilen GSP uygulamalarından eil51 için en iyi rotayı gösteren grafik Şekil 6.5.'de verilmiştir. A^3 ile eil51 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.6.'da verilmiştir.

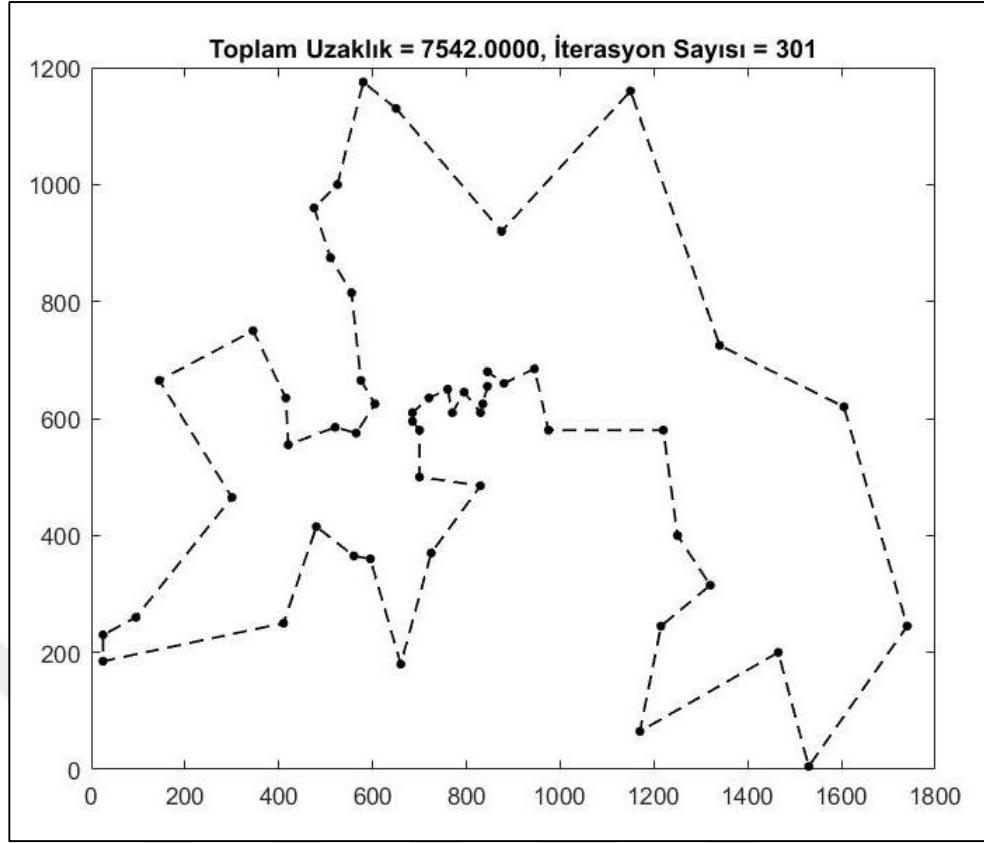


Şekil 6.5. A³ ile eil51 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.6. A³ ile eil51 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	426
En iyi rota	35 → 36 → 3 → 28 → 31 → 26 → 8 → 22 → 1 → 32 → 11 → 38 → 5 → 37 → 17 → 4 → 18 → 47 → 12 → 46 → 51 → 27 → 6 → 48 → 23 → 7 → 43 → 24 → 14 → 25 → 13 → 41 → 19 → 40 → 42 → 44 → 15 → 45 → 33 → 39 → 10 → 49 → 9 → 30 → 34 → 50 → 16 → 21 → 29 → 2 → 20 →

A³ kullanılarak gerçekleştirilen GSP uygulamalarından berlin52 için en iyi rotayı gösteren grafik Şekil 6.6.'da verilmiştir. A³ ile berlin52 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.7.'de verilmiştir.

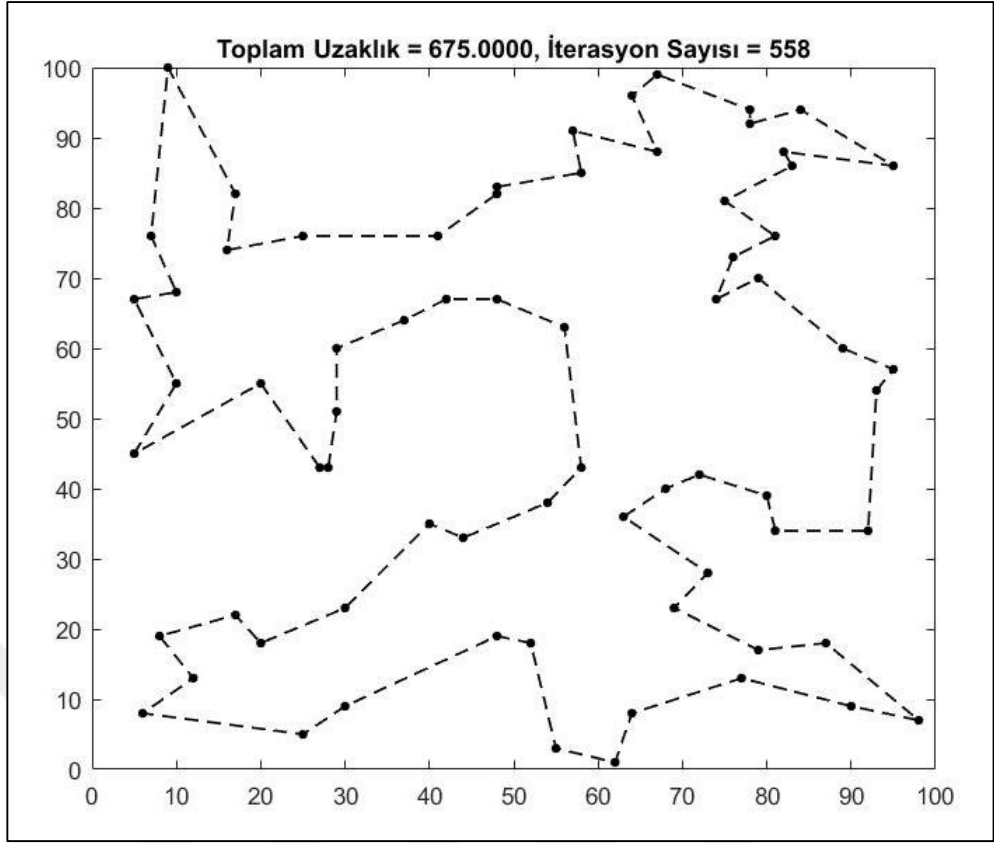


Şekil 6.6. A^3 ile berlin52 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.7. A^3 ile berlin52 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	7,542
En iyi rota	10 → 9 → 8 → 41 → 19 → 45 → 32 → 49 → 1 → 22 → 31 → 18 → 3 → 17 → 21 → 42 → 7 → 2 → 30 → 23 → 20 → 50 → 29 → 16 → 46 → 44 → 34 → 35 → 36 → 39 → 40 → 37 → 38 → 48 → 24 → 5 → 15 → 6 → 4 → 25 → 12 → 28 → 27 → 26 → 47 → 13 → 14 → 52 → 11 → 51 → 33 → 43 →

A^3 kullanılarak gerçekleştirilen GSP uygulamalarından st70 için en iyi rotayı gösteren grafik Şekil 6.7.'de verilmiştir. A^3 ile st70 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.8.'de verilmiştir.

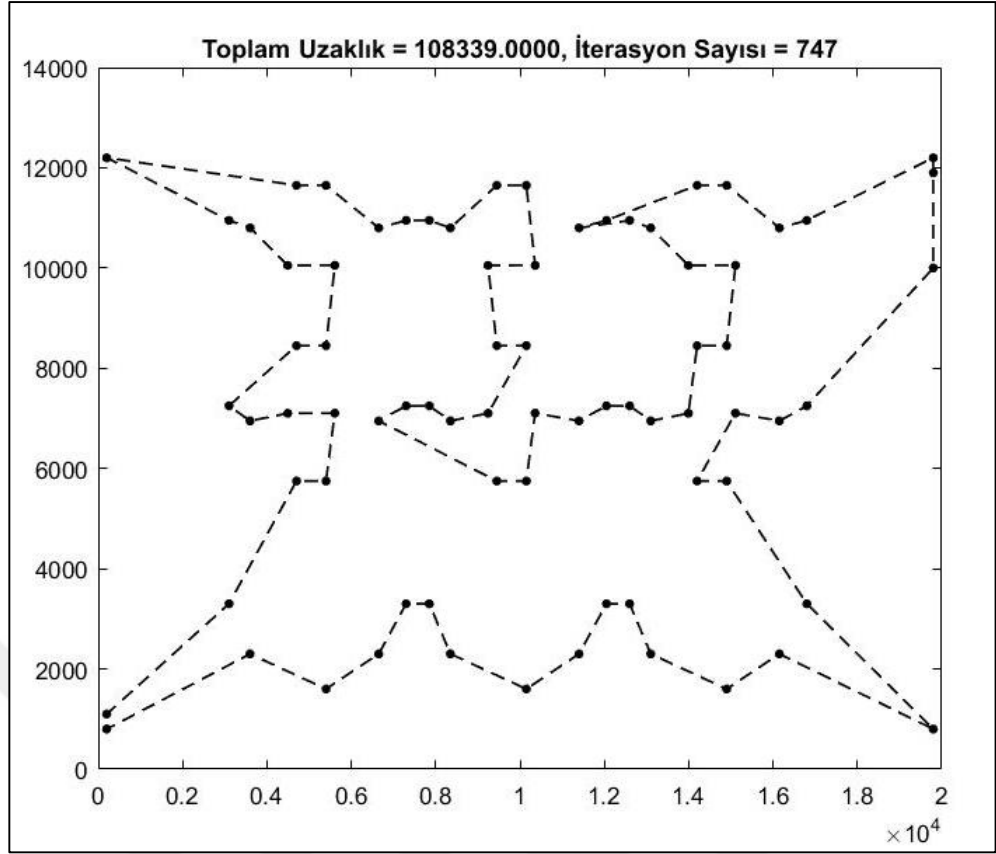


Şekil 6.7. A³ ile st70 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.8. A³ ile st70 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	675
En iyi rota	37 → 47 → 16 → 23 → 1 → 36 → 29 → 13 → 70 → 35 → 31 → 69 → 38 → 59 → 22 → 66 → 63 → 57 → 15 → 24 → 19 → 7 → 2 → 4 → 18 → 42 → 32 → 3 → 8 → 26 → 55 → 49 → 28 → 14 → 20 → 30 → 44 → 68 → 27 → 46 → 25 → 45 → 39 → 61 → 40 → 9 → 17 → 43 → 41 → 6 → 53 → 5 → 10 → 52 → 60 → 12 → 21 → 34 → 33 → 62 → 54 → 48 → 67 → 11 → 64 → 65 → 56 → 51 → 50 → 58 →

A³ kullanılarak gerçekleştirilen GSP uygulamalarından pr76 için en iyi rotayı gösteren grafik Şekil 6.8.'de verilmiştir. A³ ile pr76 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotasını gösteren tablo da Çizelge 6.9.'da verilmiştir.



Şekil 6.8. A³ ile pr76 GSP uygulaması için ulaşılan optimum rota

Çizelge 6.9. A³ ile pr76 uygulamasının ulaştığı en iyi çözüme göre, seyahat maliyeti ve rotası

En iyi seyahat maliyeti	108,339
En iyi rota	73 → 72 → 71 → 65 → 66 → 51 → 49 → 50 → 67 → 70 → 68 → 69 → 47 → 48 → 44 → 45 → 46 → 24 → 25 → 21 → 22 → 23 → 1 → 76 → 75 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 74 → 15 → 16 → 17 → 18 → 37 → 36 → 38 → 39 → 34 → 35 → 32 → 33 → 29 → 30 → 31 → 19 → 20 → 26 → 27 → 28 → 43 → 42 → 54 → 53 → 52 → 55 → 56 → 57 → 58 → 59 → 60 → 40 → 41 → 61 → 62 → 63 → 64 →

A³ ile GSP uygulaması 6 farklı meta-sezgisel yöntem ile karşılaştırılmıştır. Bunlar; TAA, GA, PSO, KKO ve bu algoritmaların farklı kombinasyonları ile üretilmiş olan PSO–KKO ve GA-PSO-KKO’dur (Deng vd., 2012). A³ ile GSP uygulaması ve TAA, GA, PSO, KKO, PSO–KKO ve GA-PSO-KKO için deneysel sonuçlar Çizelge 6.10., 6.11., 6.12., 6.13. ve 6.14.’de karşılaştırmalı olarak verilmiştir.

Çizelge 6.10. att48 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi

Algoritma	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
TAA	48	33,522	34,198	35,886	34,978	676	2.017
GA	48	33,522	34,572	36,299	35,002	1,050	3.132
PSO	48	33,522	34,759	37,672	36,179	1,237	3.690
KKO	48	33,522	34,357	35,197	34,460	835	2.491
PSO-KKO	48	33,522	33,641	34,730	33,956	119	0.354
GA-PSO-KKO	48	33,522	33,524	34,164	33,662	2	0.006
A^3	48	33,522	33,522	33,783	33,655	0.00	0.000

Çizelge 6.11. eil51 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi

Algoritma	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
TAA	51	426	445.44	472.76	459.32	19.44	4.563
GA	51	426	446.96	537.67	481.07	20.96	4.920
PSO	51	426	447.51	454.99	446.53	21.51	5.049
KKO	51	426	436.85	454.99	446.53	10.85	2.547
PSO-KKO	51	426	427.40	442.51	433.09	1.74	0.408
GA-PSO-KKO	51	426	426	436.20	431.83	0.00	0.000
A^3	51	426	426	431	428.2	0.00	0.000

Çizelge 6.12. berlin52 problemi için A³ ve diğer sezgisel tekniklerin performans değerlendirmesi

Algoritma	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
TAA	52	7,542	7,976.84	8,286.68	8,014.60	434.84	5.766
GA	52	7,542	8,201.17	8,443.02	8,376.55	659.17	8.740
PSO	52	7,542	8,197.79	8,589.31	8,319.51	655.79	8.695
KKO	52	7,542	7,647.55	7,780.57	7,732.31	105.55	1.399
PSO-KKO	52	7,542	7,568.54	7,618.31	7,586.42	26.54	0.352
GA-PSO-KKO	52	7,542	7,544.37	7,544.37	7,544.37	2.37	0.031
A ³	52	7,542	7,542	7,618	7,563.1	0.00	0.000

Çizelge 6.13. st70 problemi için A³ ve diğer sezgisel tekniklerin performans değerlendirmesi

Algoritma	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
TAA	70	675	702.27	738.45	718.56	27.56	4.083
GA	70	675	715.43	744.31	731.72	40.43	5.990
PSO	70	675	720.41	753.29	741.09	45.41	6.727
KKO	70	675	697.76	716.83	705.58	22.76	3.372
PSO-KKO	70	675	684.16	710.47	698.75	9.16	1.357
GA-PSO-KKO	70	675	679.60	704.25	694.60	4.60	0.681
A ³	70	675	675	682	679.5	0.00	0.000

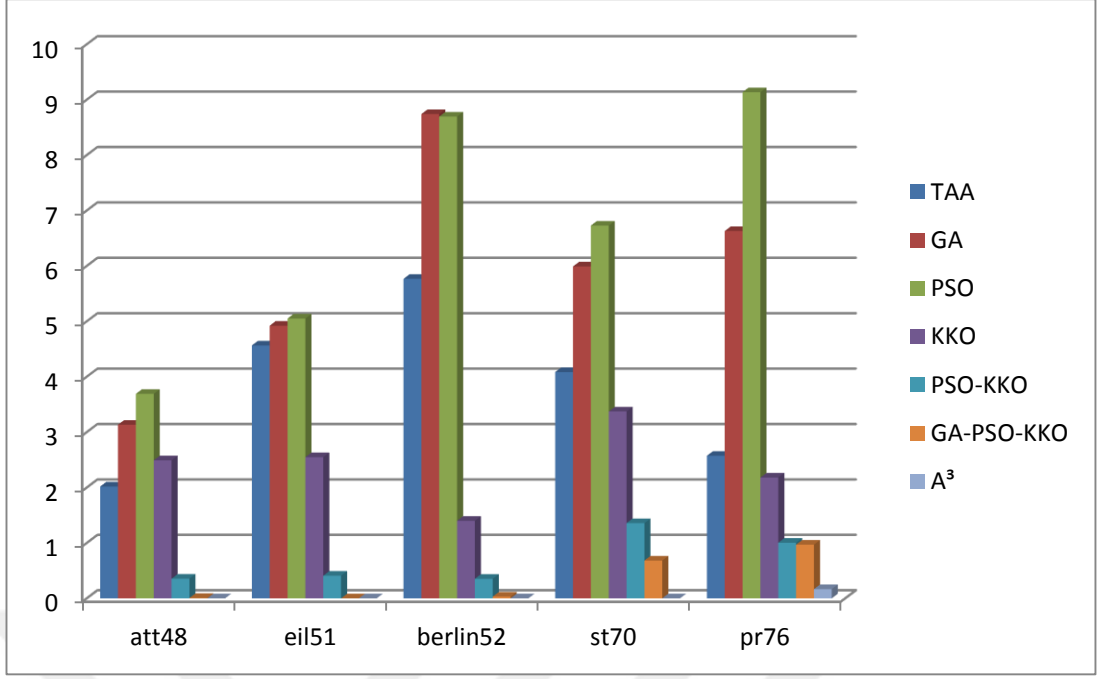
Çizelge 6.14. pr76 problemi için A^3 ve diğer sezgisel tekniklerin performans değerlendirmesi

Algoritma	Şehir Sayısı	Optimum	En iyi	En kötü	Ortalama	Fark	Hata (%)
TAA	76	108,159	110,941	130,637	122,104	2,782	2.572
GA	76	108,159	115,329	124,851	120,245	7,170	6.629
PSO	76	108,159	118,038	126,583	122,735	9,879	9.134
KKO	76	108,159	110,517	120,922	114,964	2,358	2.180
PSO-KKO	76	108,159	109,244	113,120	110,162	1,085	1.003
GA-PSO-KKO	76	108,159	109,206	112,443	110,023	1,074	0.968
A^3	76	108,159	108,339	110,343	109,47	180	0.166

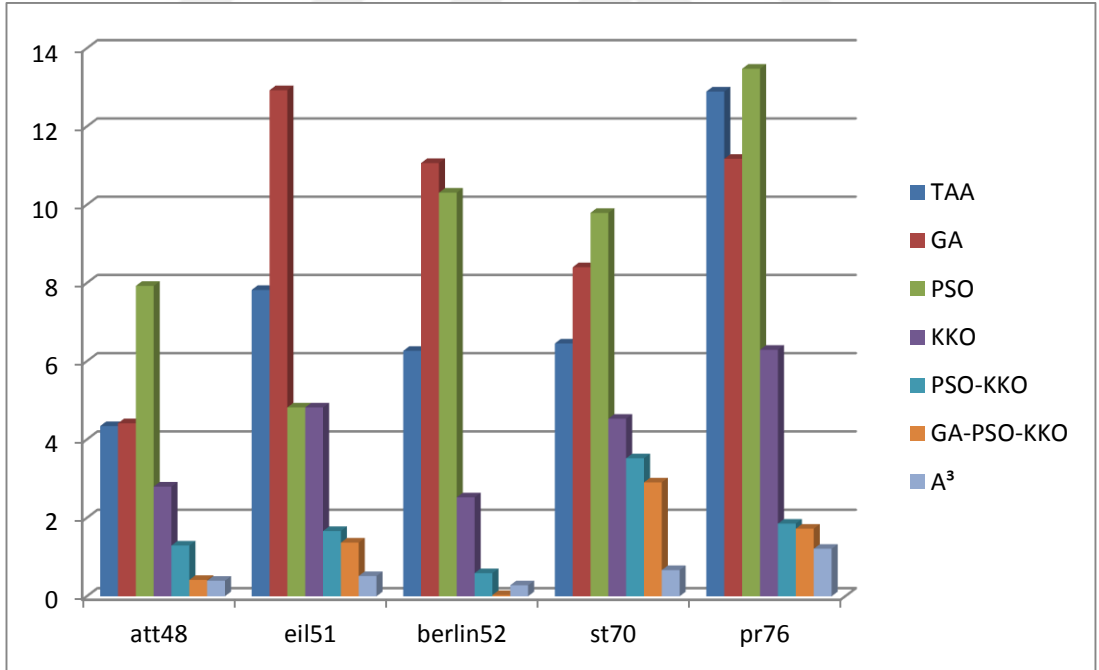
A^3 ile diğer sezgisel tekniklerin karşılaştırılması sonucunda, küçük ölçekli simetrik GSP için A^3 'ün diğer tekniklere göre daha başarılı sonuçlar verdiği görülmüştür. En iyi durumlar bakımından; A^3 , karşılaştırılan diğer yöntemlere göre daha başarılıdır. Ortalama sonuçlar bakımından ise, berlin52 problemi haricinde yine en iyi sonuçlara A^3 ile ulaşılmıştır. Berlin52 probleminde GA-PSO-KKO yönteminin elde ettiği toplam uzaklık ortalaması A^3 'e göre daha iyi görünmektedir.

A^3 küçük ölçekli simetrik GSP'lerin birçoğu için bilinen en iyi çözümlere ulaşmıştır. Bu problemler ve bilinen en iyi çözümleri sırasıyla şu şekildedir: burma14 için 30.8785, bayg29 için 9073, wi29 için 27603, att48 için 33522, eil51 için 426, berlin52 için 7542, st70 için 675'dir.

A^3 ve karşılaştırıldığı diğer sezgisel algoritmalar TAA, GA, PSO, KKO, PSO-KKO ve GA-PSO-KKO için optimum çözümden yüzdesel sapmaları gösteren grafik Şekil 6.9.'da verilmektedir. Ayrıca A^3 ve karşılaştırıldığı diğer sezgisel yöntemler için ortalama yüzdesel sapmaları gösteren grafik ise, Şekil 6.10.'da verilmektedir. Bu grafikler de, A^3 'ün diğer sezgisel tekniklere göre daha düşük yüzdesel sapmalar ile çözüme ulaştığını göstermektedir (Deng vd., 2012; Yıldırım ve Karcı, 2017b).



Şekil 6.9. A³ ve karşılaştırılan diğer algoritmaların GSP örnekleri için optimum çözümden yüzdesel sapmaları



Şekil 6.10. A³ ve karşılaştırılan diğer algoritmaların GSP örnekleri için ortalamadan yüzdesel sapmaları

A³ ile gerçekleştirilen burma14, bayg29, wi29, att48, eil51, berlin52, st70 ve pr76 GSP örnekleri için çalışma zamanı performansı Çizelge 6.15.'deki gibidir. Şehir sayısındaki artışa bağlı olarak, çalışma zamanında meydana gelen artış göz önüne alındığında, bu veri setleri

için A³'ün çalışma zamanı performansının, diğer birçok sezgisel yöntemle göre daha kabul edilebilir bir seviyede olduğu görülmektedir (Peker vd., 2013; Yıldırım ve Karcı, 2017b).

Çizelge 6.15. GSP örnekleri için A³'ün çalışma zamanı performansı

GSP Örnekleri	Şehir Sayısı	Optimum Çözüm	A ³ için En İyi Çözüm	Çalışma zamanı (sn.)
burma14	14	30.8785	30.8785	0.1
bayg29	29	9,073	9,073	0.3
wi29	29	27,603	27,603	0.3
att48	48	33,522	33,522	4.43
eil51	51	426	426	3.14
berlin52	52	7,542	7,542	7.89
st70	70	675	675	13.84
pr76	76	108,159	108,339	18.36

6.2. Beslenme Optimizasyonu Problemi için DeneySEL Sonuçlar

Yapılan tez çalışmasında, diğer bir ayrık problem olan beslenme optimizasyonu problemi, A³ kullanılarak çözülmüş ve algoritmanın bu problem için performansı izlenmiştir. Buna göre, gerçekleştirilen uygulamayla, klavyeden bireyin fiziksel, fizyolojik ve sosyolojik özellikleri girildiğinde, gerekli hesaplamalar yapılarak bireye özgü günlük beslenme çizelgesi kullanıcıya sunulmaktadır.

Çizelge 6.16.'da örnek iki bireyin cinsiyet, yaş, boy, kilo, aktivite yoğunluğu, bayansa hamilelik durumu olup olmadığı ve varsa dönemi, yine bayansa emzirme durumu olup olmadığı ve varsa dönemi bilgileri verilmektedir.

Çizelge 6.16. Örnek bireylerin fiziksel, fizyolojik ve sosyolojik özellikleri

Özellik	1. Birey	2. Birey
Cinsiyet	Bay	Bayan
Yaş	35	25
Kilo (kg)	73	58
Boy (cm)	174	168
Hamilelik durumu	-	Var
Hamilelik dönemi (1.Trimester / 2.Trimester / 3.Trimester)	-	2.Trimester
Emzirme durumu	-	-
Emzirme dönemi (0-6 ay / 7-12 ay)	-	-
Aktivite durumu (Hiç / Hafif / Orta / Ağır / Çok ağır)	Orta	Hafif

Bu bilgiler programa girdi olarak verildiğinde, öncelikle bireyin cinsiyet, yaş, boy, kilo bilgilerine bağlı olarak; bireyin kalp atışı, soluk alıp verme gibi temel vücut fonksiyonlarını yerine getirebilmesi için gereken enerjiyi belirten BMR değerleri hesaplanmıştır. Ardından BMR değeri kullanılarak, aktivite yoğunluğuna bağlı olarak tahmin edilen enerji gereksinimi olan EER1 değerleri hesaplanmıştır. Son olarak, eğer birey bayansa ve hamilelik ya da emzirme durumu varsa, bu durumlarda bireyin enerji gereksiniminin artması nedeniyle hamilelik ya da doğum sonrası dönemine bağlı olarak EER2 değerleri hesaplanmıştır. Örnek bireyler için hesaplanan bu değerler Çizelge 6.17.'de gösterilmektedir.

Çizelge 6.17. Örnek bireylerin BMR, EER1 ve EER2 değerleri

	BMR	EER1	EER2
1. Birey	1702.7	2639.1	2639
2. Birey	1396.1	1919.7	2260

Çizelge 6.18. ve Çizelge 6.19.'da, A³ yöntemi kullanılarak gerçekleştirilen beslenme optimizasyonu uygulaması ile örnek bireyler için hazırlanan beslenme çizelgeleri gösterilmektedir. Uygulamada besinlerin 100 gramı için geçerli kalori değerleri kullanılmıştır.

Çizelge 6.18. A³ ile beslenme optimizasyonu uygulamasının 1. birey için elde ettiği besin çizelgesi

1. Birey	
Besin Öğesi	Kalori Değeri (kkal)
Otlu Peynir (50 gr)	164
Tahin Helvası (50 gr)	258
Pekmez (50 gr)	147
Ekmek (50 gr)	138
Brokoli Çorbası (400 gr)	348
Yaprak Sarması (200 gr)	282
Fajita (200 gr)	234
Sığır Eti Güveci (200 gr)	190
Peynirli Pide (200 gr)	550
Avakado (200 gr)	334
TOPLAM	2645

Çizelge 6.19. A³ ile beslenme optimizasyonu uygulamasının 2. birey için elde ettiği besin çizelgesi

2. Birey	
Besin Öğesi	Kalori Değeri (kkal)
Krem Peynir (50 gr)	175
Salam (50 gr)	225
Sosis (50 gr)	161
Ekmek (100 gr)	276
Hünkar Beğendi (200 gr)	348
Ekmek (100 gr)	276
Zeytinyağlı Dolma (200 gr)	346
Patates Püresi (200 gr)	166
Ekmek (100 gr)	276
TOPLAM	2249

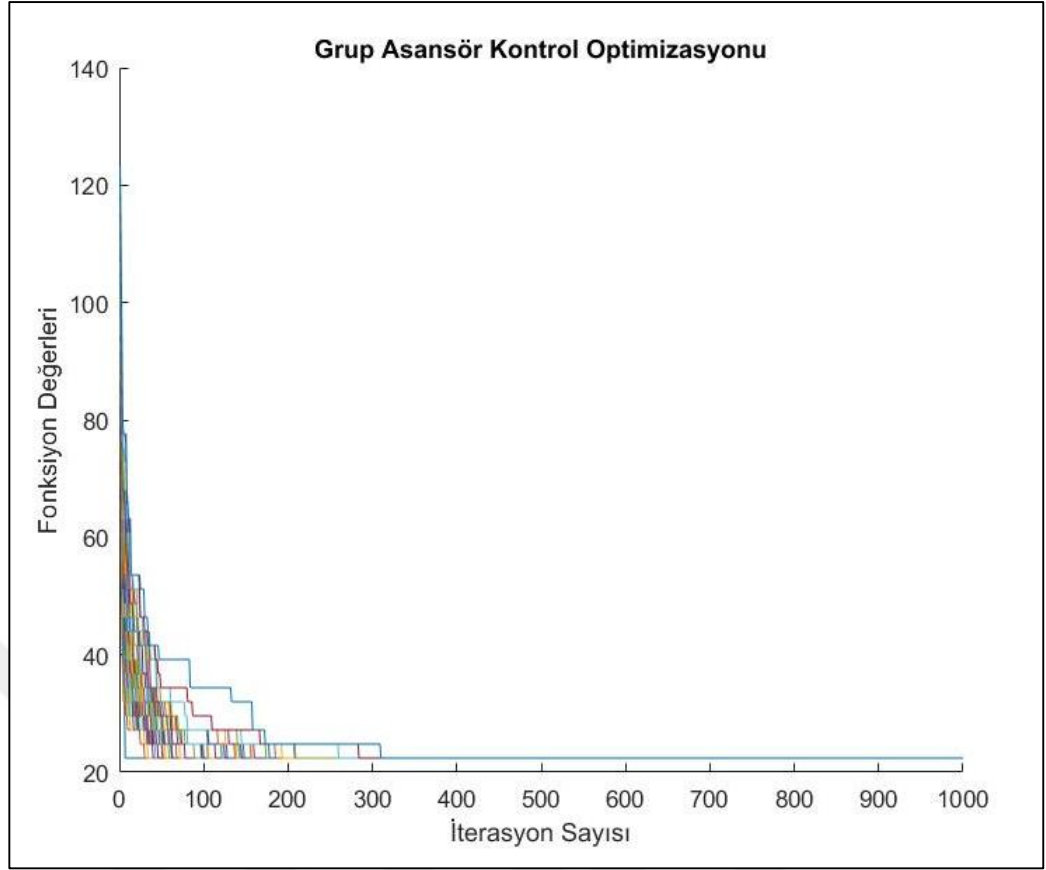
A^3 ile beslenme optimizasyonu uygulaması, 1. örnek birey için % 0.22 hatayla, 2. örnek birey için % 0.48 hatayla sonuca ulaşmıştır. Uygulama rastgele belirlenen 20 birey için test edilmiştir. Buna göre, hazırlanan beslenme çizelgeleri için ortalama hata % 0.9 olarak hesaplanmıştır. Uygulamanın çalışma zamanı performansı ortalama 1.05 sn. olarak tespit edilmiştir.

Literatürde beslenme optimizasyonu alanında birçok çalışma yapılmıştır. Ancak A^3 kullanılarak yapılan uygulamayla içerik bakımından farklılıklar bulunması nedeniyle performans karşılaştırılması yapılamamıştır. Ancak sonuçların düşük hata oranıyla elde edilmesi ve çalışma zamanı performansı uygulamanın başarısını göstermektedir.

6.3. Grup Asansör Kontrol Optimizasyonu için Deneysel Sonuçlar

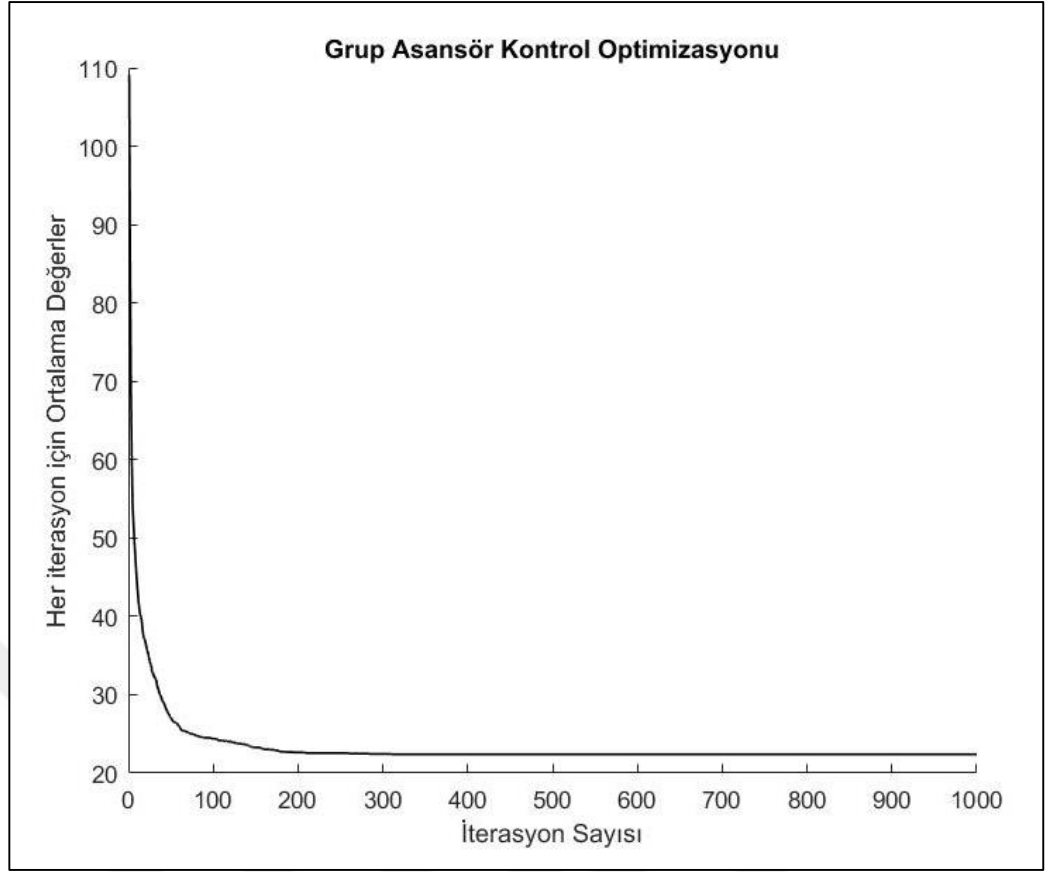
Bu tez çalışmasında, son olarak A^3 kullanılarak bir diğer ayrık problem olan grup asansör kontrol optimizasyonu probleminin çözümü yapılmıştır. 9 asansöre sahip, 16 katlı bir bina için gelen kat çağrılarının minimum bekleme süresiyle cevaplanması hedeflenmektedir. Uygulama yazılımı, MATLAB 2013a ile Intel Core-i3 3.3 GHz işlemcili PC kullanılarak gerçekleştirilmiştir. A^3 kullanılarak elde edilen sonuçlar, daha önce aynı problemin çözümünde kullanılan AYBS yöntemiyle elde edilen sonuçlarla karşılaştırılmıştır.

A^3 ile grup asansör kontrol optimizasyonu uygulamasında, en ideal asansör-çağrı eşleştirmesi için tüm atom kümesinin elde ettiği sonuçlar Şekil 6.11.'deki grafik ile gösterilmektedir.



Şekil 6.11. A^3 ile grup asansör kontrol optimizasyonunda her atomun amaç fonksiyonu değeri

Grup asansör kontrol problemi için A^3 yöntemiyle elde edilen sonuçlara göre, atom kümesinin amaç fonksiyon değerlerinin ortalaması; yani, yolcuların toplam bekleme zamanı ortalaması Şekil 6.12.'deki grafikte verilmektedir.



Şekil 6.12. A^3 ile grup asansör kontrol optimizasyonunda atom kümesinin amaç fonksiyonu ortalaması

Grafiklerden de görüldüğü gibi, algoritma kısa sürede optimum sonuca ulaşmıştır. Yani, A^3 yöntemi kullanılarak, kat çağrıları ile onu cevaplayacak asansör eşleştirmesi en ideal şekilde gerçekleştirilmiştir.

Bu tez çalışmasında, A^3 ile çözümü gerçekleştirilen grup asansör kontrol optimizasyonu probleminin performansı, herhangi bir optimizasyon yöntemi kullanılmadan ve AYBS yöntemi kullanılarak elde edilen sonuçlarla karşılaştırılmıştır. A^3 kullanılarak, AYBS yöntemi kullanılarak ve herhangi bir optimizasyon yöntemi kullanılmaksızın belirlenen asansör-çağrı eşleştirmeleri ve asansör rotaları, Çizelge 6.20.'de verilmiştir (Baygın ve Karaköse, 2011).

Çizelge 6.20. Karşılaştırılan tekniklere göre asansörlerin cevaplayacakları kat çağrıları ve cevaplama sıraları

Asansör	Optimizasyon olmaksızın izlenecek yol	AYBS ile izlenecek yol	A ³ ile izlenecek yol
1	3. kat	3-5. katlar	1. kat
2	6. kat	6. kat	5-6. katlar
3	9. kat	9. kat	8-9. katlar
4	8-13-14-16. katlar	13-14-16. katlar	11. kat
5	5. kat	1. kat	13-14. katlar
6	-	-	3. kat
7	1-11. katlar	8-11. katlar	16. kat

Grup asansör kontrolünün, optimizasyon yöntemi kullanılmaksızın yapıldığı sistemlerde, yolcuların asansör bekleme süreleri toplamı oldukça uzundur. Buna karşın AYBS yöntemi kullanılarak yapılan grup asansör kontrol optimizasyonu uygulaması da optimum sonuca ulaşamamıştır. Ancak A³ yöntemiyle gerçekleştirilen uygulamanın optimum sonuca ulaştığı görülmüştür. Buna göre, optimizasyon olmaksızın, AYBS ile optimizasyon yapıldığında veya A³ ile optimizasyon gerçekleştirildiğinde, asansör kullanıcılarının asansör hizmetini bekleme süreleri toplamı Çizelge 6.21.'de gösterilmektedir (Baygın ve Karaköse, 2011).

Çizelge 6.21. Tüm kat çağrıları için yolcuların asansör hizmetini bekleme süreleri toplamının karşılaştırılması

Kullanılan yöntem	Toplam bekleme zamanı (sn.)
Optimizasyon olmaksızın	326.4
AYBS ile optimizasyon	246.4
A ³ ile optimizasyon	22.4

Sonuçlar değerlendirildiğinde, A³'ün 16 katlı ve 9 asansörlü bir binanın grup asansör kontrol optimizasyonu için, diğer tekniklere nazaran çok başarılı olduğu görülmektedir. Bu optimizasyon probleminde A³'ün kullanılması, yolcuların asansör hizmetinden yararlanmak için geçirdikleri süreyi oldukça kısaltmıştır.

A³ ile grup asansör kontrol optimizasyonu uygulamasının hesaplama zamanı performansı, programın 20 kez çalıştırılmasıyla elde edilmiştir. Buna göre, A³ uygulamasının ortalama hesaplama zamanı 0.368 sn. olarak belirlenmiştir.

7. SONUÇ

Literatürde yüzü aşkın sezgisel ve meta-sezgisel algoritma bulunmakta ve bu algoritmalar her gün yenisi eklenmektedir. Yeni algoritmaların ortaya çıkması, aslında her problem için en ideal çözümü elde eden bir algoritmanın henüz keşfedilememiş olmasından kaynaklanmaktadır (Canayaz, 2015).

Bu tez çalışmasında, yakın zamanda geliştirilen bir meta-sezgisel yöntem olan Yapay atom algoritması (A^3) tanıtılarak, bu yöntemin ayrık optimizasyon problemlerindeki performansı test edilmiştir. A^3 , kimyasal bileşik oluşturma süreçleri olan elektron alıp verme ve elektron ortak kullanma süreçlerinden esinlenerek geliştirilmiştir. Bu süreçler, iyonik bağ ve kovalent bağ operatörleri olarak modellenmiştir. Diğer meta-sezgisel yöntemlerden farklı olarak, parametre değerlerinin sonuç üzerindeki etkisini değerlendirmekte, bu da algoritmanın performansını arttırmaktadır.

Günümüzde tıp, mühendislik, diyetetik, lojistik, elektronik gibi farklı alanlardaki optimizasyon problemlerinde meta-sezgisel algoritmalar kullanılmaktadır. Çözümü matematiksel yöntemlerle zor ve karmaşık, hesaplama zamanı da uzun olan problemlerin meta-sezgisel yöntemlerle çözümü; hem hesaplama zamanını düşürmekte, hem de işlem karmaşıklığını azaltmaktadır. Bir optimizasyon algoritmasının performansında, ayrık optimizasyon problemlerinde gösterdiği başarı belirleyici bir ölçüttür. Bu amaçla yapılan tez çalışmasında, A^3 yaklaşımı üç ayrık optimizasyon problemlerine uygulanmıştır. Bunlar; gezgin satıcı, beslenme optimizasyonu ve grup asansör kontrol optimizasyonu problemleridir.

Bu tez çalışmasında ilk olarak, TSPLib kütüphanesinde bulunan ve literatürde sıklıkla kullanılan 8 adet küçük ölçekli simetrik GSP'nin A^3 yaklaşımıyla çözümü gerçekleştirilmiştir. GSP'nin çözümünde, A^3 yönteminin sürekli optimizasyon problemleri için tasarlanan formu üzerinde değişiklikler yapılmıştır. Problem parametrelerinin tekrarsız ve sıralı yapısını korumak için A^3 operatörleri için yeni teknikler geliştirilmiştir. Ayrık problemler için güncellenen A^3 yaklaşımı, uygulandığı problemlerin 7'si için optimum çözüme ulaşmıştır. Bu alanda TSPLib kütüphanesinde yüzlerce problem ve çözümünde kullanılan onlarca sezgisel ve meta-sezgisel yöntem bulunmaktadır. Buna karşın henüz bu problemlerin hepsi için en iyi sonuçlara ulaşan bir algoritma keşfedilmemiştir. Bu nedenle yapılan çalışmalar, performans bakımından literatürde aynı GSP'lere uygulanan yöntemlerle karşılaştırılmıştır. Tez çalışması kapsamında yapılan GSP uygulamaları TAA, GA, PSO, KKO, PSO-KKO ve GA-PSO-KKO ile gerçekleştirilen uygulamalarla karşılaştırılmıştır. Karşılaştırılan GSP'ler arasında berlin52 haricinde en iyi, en kötü ve ortalama sonuçlar bakımından en başarılısı, ayrık problemlere uyarlanan A^3 yöntemi ile elde edilmiştir. berlin52 problemi için de, yine A^3 yaklaşımı optimum sonuca ulaşmasıyla avantaj

sağlamaktadır. Yalnızca en kötü ve ortalama durum bakımından GA-PSO-KKO kombinasyonu daha iyi sonuç elde etmiştir. Ayrıca, ayrık problemlere uyarlanan A^3 yaklaşımı ile gerçekleştirilen GSP uygulamalarında, şehir sayısındaki artışa oranla, hesaplama zamanında meydana gelen artış makul seviyededir.

İkinci uygulama, beslenme optimizasyonu probleminin A^3 yaklaşımı ile çözülmesidir. Yapılan tez çalışmasında, bireyin fiziksel, fizyolojik ve sosyolojik özellikleri dikkate alınarak, bireyin günlük enerji ihtiyacına uygun kalori değerini gün içinde yediği besinlerden alabilmesi için optimum bir beslenme çizelgesi oluşturulması amaçlanmıştır. Yapılan çalışmayla bireyin günlük kalori ihtiyacına oldukça yakın değerlerde besin çizelgesi elde edilmiştir. Bu tez çalışmasında kullanılan beslenme optimizasyonu probleminin, literatürde benzerleri bulunmasına karşın çoğunlukla paket programlarla çözümü gerçekleştirilmiştir. Ayrıca literatürde beslenme optimizasyonu alanında yapılan çalışmaların birçoğu, maliyet yönüyle optimizasyon sağlamaktadır. Buna karşın, GA kullanılarak kalori bakımından optimizasyon sağlayan bir çalışmadaysa, günün sadece kahvaltı öğünü değerlendirilmekte ve bireye özgü bir çözüm sunulmamaktadır. Problemler arasındaki farklılıklardan dolayı, A^3 ile beslenme optimizasyonunun başka yöntemlerle karşılaştırması yapılamamıştır. Buna karşın, yapılan uygulama, bireye özgü bir hesaplama ile optimuma yakın beslenme çizelgeleri elde ederek, daha önce yapılmayan bir çalışmayla literatüre katkı sağlamıştır. Ayrıca çalışma zamanı performansı da makul düzeydedir.

Tez çalışmasında yapılan son uygulama ise, literatürde AYBS yöntemi kullanılarak gerçekleştirilen bir grup asansör kontrol optimizasyonu probleminin, A^3 yöntemi kullanılarak çözülmesidir. Problem parametreleri, asansörlerin hangi sırayla kat çağrılarını cevaplayacaklarını belirttiği için, GSP'de olduğu gibi sıralamanın önemli olduğu bir problemdir. Dolayısıyla grup asansör kontrol optimizasyonu probleminin çözümü için de A^3 yaklaşımının sürekli optimizasyon problemleri için geliştirilen şekli üzerinde değişiklikler yapılmıştır. A^3 yaklaşımı, bu kez tekrarlı permütasyon yapıda bir problem için uyarlanmıştır. Yapılan çalışma, optimum çözüme ulaşması bakımından literatürdeki örneğine nazaran oldukça başarılı bir sonuç elde etmiştir. Ayrıca hesaplama zamanı performansı da milisaniyeler düzeyinde olup, problem için oldukça iyi bir seviyededir.

Yapılan tez çalışmasında, A^3 yönteminin ayrık problemlere uygulanması amacıyla geliştirilen teknikler, ileride ayrık optimizasyon alanında A^3 yöntemi kullanılarak yapılacak çalışmaların önünü açan niteliktedir. A^3 yönteminin ayrık problemlere uygulanmasıyla elde edilen başarılı sonuçlar, bu algoritmanın literatürde bulunan birçok arama ve optimizasyon problemi için kullanışlı bir yöntem olduğunu göstermektedir. Ayrıca yapılan tez çalışmasıyla, A^3 yaklaşımının, literatürde bulunan birçok meta-sezgisel yöneme göre, farklı alanlardaki problemlere uygulanabilirliğinin ve doğruluğunun yüksek, hesaplama zamanı ve karmaşıklığının ise makul düzeyde olduğu kanıtlanmıştır.

Gelecekte A^3 yönteminin farklı alanlardaki gerçek dünya problemlerine uygulanabilmesi mümkündür. Ayrıca bu tez çalışmasıyla yapılan uygulamaların, geliştirilmesi ve çeşitlendirilmesinin de önü açıktır. Örneğin; bu tez çalışmasında küçük ölçekli ve simetrik GSP'ler için A^3 yöntemi uygulanmıştır. Gelecekte bunu asimetrik veya büyük ölçekli GSP'ler için de çeşitlendirmek mümkündür. Beslenme optimizasyonu problemini, besin öğeleri (yağ, protein, karbonhidrat, vitamin vb.) ve maliyet bakımından optimizasyon için çok amaçlı tasarlayarak A^3 yöntemiyle çözmek de, ileride yapılabilecek çalışmalar arasında sayılabilir.



8. KAYNAKLAR

- Acampora, G., Gaeta, M., Loia, V. (2011). Combining multi agent paradigm and memetic computing for personalized and adaptive learning experiences. *Journal of Computational Intelligence* **27**, 141-165.
- Akay, B. (2009). *Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi*. Doktora tezi, Erciyes Üniversitesi, Kayseri.
- Alataş, B. (2012). A novel chemistry based metaheuristic optimization method for mining of classification rules. *Journal of Expert Systems with Applications* **39**, 11080–11088.
- Altunbey, F., Alataş, B. (2015). Overlapping community detection in social networks using parliamentary optimization algorithm. *International Journal of Computer Networks and Applications* **2**, 12-19.
- Anderson, A.M., Earle, M.D. (1983). Diet planning in the third world by linear and goal programming. *Journal of the Operational Research Society* **34**, 9-16.
- Anonim. (2016). <https://www.diyetkolik.com/> (internet erişimi 15 Kasım, 2017).
- Anonim. (2017). <http://www.kaloricetveli.org/yiyecek/yemekler-ve-oeguenler> (internet erişimi 11 Kasım, 2017).
- Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J. (2007). *Traveling salesman problem: A computational study*. Princeton University Press, New Jersey, USA, 608 p.
- Arora, J.S. (1989). *Introduction to Optimum Design*. McGraw-Hill, New York, USA, 880 p.
- Ayaz, S. (2012). Grup asansör çalışmalarında etkin çağrı paylaşırma ve trafik denetimi (s: 27-30). *Asansör Sempozyumu*, 4-6 Ekim, İzmir, Türkiye.
- Bahçeci, M. (2011) Obezite (pp: 50-80). In: TEMD Obezite, Dislipidemi, Hipertansiyon Çalışma Grubu (Ed.), *Türkiye Endokrinoloji ve Metabolizma Derneği Hipertansiyon, Obezite ve Lipid Metabolizması Tanı ve Tedavi Rehberi*, 5. Baskı, Miki Matbaacılık, Ankara.
- Bahçeci, M. (2013). Obezite tedavi kılavuzu ve yaşam tarzı önerileri, Türkiye Endokrinoloji ve Metabolizma Derneği, http://www.turkendokrin.org/icerik_hastalar.php?id=6 (internet erişimi 15 Mayıs, 2015).
- Balas, E., (1965). An additive algorithm for solving linear programs with zero-one variables. *Journal of Operations Research* **13**, 517-546.
- Balintfy, J.L. (1964). Menu planning by computer. *Journal of Communications of the ACM* **7**, 255-259.
- Barney, G.C., dos Santos, S.M. (1976). *Lift traffic analysis, design and control*. Peter Peregrinus, London, UK, 386 p.
- Baygın, M., Karaköse, M. (2011). Adaptif yapay bağışık sistem tabanlı grup asansör kontrol algoritması (s: 205-210). *Elektrik - Elektronik ve Bilgisayar Sempozyumu*, Fırat Üniversitesi, 5-7 Ekim, Elazığ, Türkiye.

- Baysal, A., Keçecioglu, S., Arslan, P., Yücecan, S., Pekcan, G., Güneyli, U., Birer, S., Sağlam, F., Yurttagül, M., Çehreli, R. (1991). *Besinlerin Bileşimleri*. Türkiye Diyetisyenler Derneği Yayını, Ankara, Türkiye, 53 s.
- Bingöl, H., Alataş, B. (2015). Metasezgisel optimizasyon tekniklerine spor tabanlı yeni bir yaklaşım: lig şampiyonası algoritması. *Fırat Üniversitesi Fen Bilimleri Dergisi* **27**, 1-11.
- Birbil, S.I., Fang, S.C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization* **25**, 263-282.
- Bolat, B. (2006). Genetik algoritma ile asansör kontrol sistemlerinin simülasyonu ve optimizasyonu. *Journal of Engineering and Natural Sciences* **2**, 14-20.
- Booth, M.L., Hunter, C., Gore, C.J., Bauman, A., Owen, N. (2000). The relationship between body mass index and waist circumference: implications for estimates of the population prevalence of overweight. *International Journal of Obesity and Related Metabolic Disorders* **24**, 1058-1061.
- Boschetti, M.A., Maniezzo, V., Roffilli, M., Röhrer, A.B. (2009). Matheuristics: optimization, simulation and control (pp: 171-177). In: Blesa, M.J., Blum, C., Di Gaspero, L., Roli, A., Sampels, M., Schaerf, A. (Ed.), *Hybrid Metaheuristics*. Vol. 5818, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg.
- Can, Ü. (2014). *Yerçekimsel arama algoritmasıyla nicel birliktelik kurallarının keşfi*. Yüksek lisans tezi, Tunceli Üniversitesi, Tunceli.
- Canayaz, M. (2015). *Cırcır böceği algoritması: Yeni bir meta-sezgisel yaklaşım ve uygulamaları*. Doktora tezi, İnönü Üniversitesi, Malatya.
- Canayaz, M., Karcı, A. (2015). Investigation of cricket behaviours as evolutionary computation for system design optimization problems. *Journal of Measurement* **68**, 225-235.
- Canayaz, M., Demir, M. (2016). Veri Kümelemede Yapay Atom Algoritması ve Cırcır Böceği Algoritmasının Karşılaştırılmalı Analizi (pp: 1230-1239). *Proceedings of 4th International Symposium on Innovative Technologies in Engineering and Science (ISITES-2016)*, November 3-5, Antalya, Turkey.
- Canayaz, M., Karcı, A. (2016). Cricket behaviour-based evolutionary computation technique in solving engineering optimization problems. *Journal of Applied Intelligence* **44**, 362-376.
- Chu, S.C., Tsai, P.W., Pan, J.S. (2006). Cat swarm optimization (pp: 854-858). In: Yang, Q., Webb, G. (Ed.), *PRICAI 2006: Trends in Artificial Intelligence*. Vol. 4099, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg.
- Cortes, P., Larraneta, J., Onieva, L. (2004). Genetic algorithm for controllers in elevator groups: analysis and simulation during lunchpeak traffic. *Journal of Applied Soft Computing* **4**, 159-174.
- Crites, R.H., Barto, A.G. (1998). Elevator group control using multiple reinforcement learning agents. *Journal of Machine Learning* **33**, 235-262.
- Croes, G.A. (1958). A method for solving traveling salesman problems. *Journal of Operations Research* **6**, 791-812.

- Çifci, E., Açma, B., Selçuk, B. (2017). Yapay atom algoritması kullanarak bireye özgü öğün programlı beslenme çizelgesi hazırlanması. *Proceedings of 2nd International Conference on Artificial Intelligence and Data Processing (IDAP-2017)*, Inonu University, September 16-17, Malatya, Turkey.
- Çolak, S. (2010). Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* **19**, 423-438.
- Dantzig, G., Fulkerson, R., Johnson, S. (1954). Solution of a large-scale travelling salesman problem. *Journal of Operations Research Society of America* **2**, 393-410.
- Dasgupta, D. (1993). An Overview of Artificial Immune Systems and Their Applications (pp: 3-21). In: Dasgupta, D. (eds), *Artificial Immune Systems and Their Applications*. Springer, Berlin, Heidelberg.
- Demir, M., Karcı, A., Özdemir, M. (2011). Fidan gelişim algoritması yardımı ile DNA motiflerinin keşfi. *Çankaya University Journal of Science and Engineering* **8**, 51-62.
- Demir, M., Karcı, A. (2015). Yapay atom algoritması yöntemi ile veri kümeleme. *Muş Alparslan Üniversitesi Fen Bilimleri Dergisi* **3**, 340-353.
- Demirel, U., Bahçecioğlu, İ.H. (2010). Enteral ve Parenteral Beslenmeye Klinik Yaklaşım. *Güncel Gastroenteroloji Dergisi* **14**, 149-154.
- Deng, W., Chen, R., He, B., Liu, Y., Yin, L., Guo, J. (2012). A novel two-stage hybrid swarm intelligence optimization algorithm and application. *Journal of Soft Computing* **16**, 1707-1722.
- Dorigo, M., Gambardella, L.M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *Journal of IEEE Transactions on Evolutionary Computation* **1**, 53-66.
- Dorigo, M., Stützle, T. (2004). *Ant colony optimization*. The MIT Press, Cambridge, Massachusetts, London, England.
- Eker, E., Melih, Ş. (2002). Birinci basamakta obeziteye yaklaşım. *TBB Sürekli Tıp Eğitimi Dergisi* **11**, 246-249.
- Eker, E. (2006). *Edirne ili kentsel alanında yaşayan erişkinlerde beslenme durum değerlendirilmesi*. Tıpta Uzmanlık Tezi, Trakya Üniversitesi, Edirne.
- Ellabib, I., Calamai, P., Basir, O. (2007). Exchange strategies for multiple ant colony system. *Journal of Information Sciences* **177**, 1248-1264.
- Erol, O.K., Eksin, I. (2006). A new optimization method: Big bang-big crunch. *Journal of Advances in Engineering Software* **37**, 106-111.
- Fei, L., Xiaocui, Z., Yuge, X. (2010). A new hybrid elevator group control system scheduling strategy based on particle swarm simulated annealing optimization algorithm. *Proceedings of the 8th World Congress on Intelligent Control and Automation*, July 7-9, Jinan, China.

- Formato, R.A. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Journal of Progress in Electromagnetics Research* **77**, 425–491.
- Fuller, N., Jebb, S.A., Goldberg, G., Pullicino, E., Adams, C., Cole, T.J., Elia, M. (1991). Inter-observer variability in the measurement of body composition. *European Journal of Clinical Nutrition* **45**, 43-49.
- Gargari, E.A., Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition (pp: 4661–4667). *Proceedings of IEEE Congress on Evolutionary Computation*, September 25-28, Singapore.
- Gass, S.I., Harris, C.M. (2000). *Encyclopedia of Operations Research and Management Science*. Springer, New York, USA, 960 p.
- Gavish, B., Graves, S.C. (1978). *The travelling salesman problem and related problems*. Operations Research Center, MIT, Cambridge, MA, 32 p.
- Geem, Z.W., Kim, J.H., Loganathan, G.V. (2001). A new heuristic optimization algorithm: harmony search. *Journal of Simulation* **76**, 60–68.
- Ghisalberti, E.L. (1979). Propolis: a review. *Journal of Bee World* **60**, 59-84.
- Glover, F. (1989). Tabu search-part 1. *ORSA Journal on Computing* **1**, 190-206.
- Goldberg, D.E. (1989). *Genetic algorithm in search, optimization and machine learning*. Addison- Wesley Publishing Company, Boston, USA, 432 p.
- Green, R.C., Wang, L., Alam, M. (2012). Training neural networks using central force optimization and particle swarm optimization: Insights and comparisons. *Journal of Expert Systems with Applications* **39**, 555–563.
- Gudwin, R., Gomide, F. (1994). Genetic algorithms and discrete event systems: an application (pp: 742-745). *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, June 27-29, Orlando, FL, USA.
- Gutin, G., Punnen, A.P. (2007). *The traveling salesman problem and its variations*. Springer, New York, USA, 829 p.
- Gümüştekin, S., Şenel, T. (2013). Beslenme Probleminde Genetik Algoritma Kullanılması. *International Anatolia Academic Online Journal* **1**, 19-26.
- Gürbüz, Ö. (2015). *Tabu arama algoritmasının kuyruk problemine uygulanması*. Yüksek lisans tezi, Hacettepe Üniversitesi, Ankara.
- Gürsu, B. (2014). Elektromanyetizm algoritması ile radyal şebekelerde ters zamanlı yönsüz selektif aşırı akım röle koordinasyonu (s: 24-28). *Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, 27 – 29 Kasım, Bursa.
- Harris, J.A., Benedict, F.G. (1918). A Biometric Study of Human Basal Metabolism (pp: 370–373). *Proceedings of the National Academy of Sciences of USA*, December 1, USA.
- Harris, J.A., Benedict, F.G. (1919). *A biometric study of basal metabolism in man*. Carnegie Institution of Washington Publication, Washington, USA, 266 p.

- Held, M., Karp, R.M. (1970). The traveling salesman problem and minimum spanning trees. *Journal of Operations Research* **18**, 1138–1162.
- Hezer, S., Kara, Y. (2013). Eşzamanlı dağıtım ve toplamalı araç rotalama problemlerinin çözümü için bakteriyel besin arama optimizasyonu tabanlı bir algoritma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi* **28**, 373-382.
- Hua, Z., Huang, F. (2006). A variable-grouping based genetic algorithm for large-scale integer programming. *Journal of Information Sciences* **176**, 2869-2885.
- Institute of Medicine of the National Academies (2005). *Dietary reference intakes for energy, carbohydrate, fiber, fat, fatty acids, cholesterol, protein, and amino acids*. The National Academies Press, Washington, USA, 1332 p.
- Karaboğa, D., Baştürk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization* **39**, 459-471.
- Karaboğa, D. (2017). *Yapay Zekâ Optimizasyon Algoritmaları*. Nobel Yayınları, İstanbul, Türkiye, 246 s.
- Karadoğan, A., Karıcı, A. (2013). Artificial atom algorithm for reinforcement learning (p:379). *Proceedings of 2nd International Eurasian Conference on Mathematical Sciences and Applications (IECMSA-2013)*, International Sarajevo University, August 26-29, Sarajevo, Bosnia and Herzegovina.
- Karadoğan, A. (2014). *Takviyeli öğrenme için yapay atom algoritması (A^3) kullanımı*. Yüksek lisans tezi, İnönü Üniversitesi, Malatya.
- Karacı, A. (2007). Theory of saplings growing up algorithm (pp: 450-460). In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (Ed.), *Adaptive and Natural Computing Algorithms*. Vol. 4431, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg.
- Karacı, A. (2012). A new meta-heuristic algorithm based on chemical process: Atom algorithm (pp: 85-86). *Proceedings of 1st International Eurasian Conference on Mathematical Sciences and Applications (IECMSA-2012)*, Prishtine University, September 3-7, Prishtine, Kosovo.
- Karacı, A. (2017). Differential evolution algorithm and its variants. *Anatolian Journal of Computer Sciences* **2**, 10-14.
- Kashan, A.H. (2009). League championship algorithm: a new algorithm for numerical function optimization (pp: 43-48). *Proceedings of IEEE Soft Computing and Pattern Recognition*, December 04-07, Malacca.
- Kaveh, A., Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Journal of Acta Mechanica* **213**, 267–289.
- Kennedy, J., Eberhart, R.C. (1995). Particle swarm optimization (pp: 1942-1948). *Proceedings of IEEE International Conference on Neural Network*, Nov. 27–Dec. 1, Perth, WA, Australia.
- Keskintürk, T. (2006). Diferansiyel gelişim algoritması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi* **5**, 85-99.

- Kreymann, G., Adolph, M., Mueller, M.J. and Working group for developing the guidelines for parenteral nutrition of The German Association for Nutritional Medicine (2009). Energy expenditure and energy intake - Guidelines on Parenteral Nutrition - Chapter 3. *Journal of German Medicine Science* **7**, 1-7.
- Kılınç, E. (2007). *Diyet problemlerinin optimizasyonu ve bir uygulama*. Yüksek lisans tezi, Süleyman Demirel Üniversitesi, Isparta.
- Kılınç, F.N., Çağdaş, D.N. (2013). Diet and physical activity interventions do have effects on body composition and metabolic syndrome parameters in overweight and obese adolescents and their mothers. *The Turkish Journal of Pediatrics* **55**, 292-299.
- Kim, C., Seong, K.A., Lee-kwang, H. (1998). Design and implementation of a fuzzy elevator group control system. *Journal of IEEE Transactions on Systems, Man, and Cybernetics* **28**, 277-287.
- Klein, S., Romijn, J.A. (2008). Obesity (pp: 1563-1587), In: Kronenberg, H.M., Melmed, S., Polonsky, K.S., Larsen, P.R. (Ed.), *Williams Textbook of Endocrinology*. 11th Edition, Saunders, Elsevier, Philadelphia, PA.
- Lawler, E.L., Lenstra, J.K., Shmoys, D.B. (1985). *The traveling salesman problem: A guided tour of combinatorial optimization*. Series in discrete mathematics & optimization, John Wiley, Chichester, UK, 476 p.
- Lee, K.S., Geem, Z.W. (2004). A new structural optimization method based on the harmony search algorithm. *Journal of Computers and Structures* **82**, 781-798.
- Lee, K.S., Geem, Z.W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Journal of Computer Methods in Applied Mechanics and Engineering* **194**, 3902–3933.
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell Systems Technical Journal* **44**, 2245-2269.
- Lo, C.C., Hus, C.C. (1998). Annealing framework with learning memory. *Journal of IEEE Transactions on Systems, Man, and Cybernetics - Part A* **28**, 1-13.
- Lv, Y. (2009). Multi-objective nutritional diet optimization based on quantum genetic algorithm (pp: 336-340). *Proceedings of International Conference on Computing, Networking and Communications*, August 14-16, Tianjin, China.
- Malek, M., Guruswamy, M., Pandya, M., Owens, H. (1989). Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Journal of Annals of Operations Research* **21**, 59-84.
- Masutti, T.A., de Castro, L.N. (2009). A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Journal of Information Sciences* **179**, 1454-1468.
- Miller, C.E., Tucker, A.W., Zemlin, R.A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery* **7**, 326-329.

- Mudaliar, D.N., Modi, N.K. (2013). Unraveling travelling salesman problem by genetic algorithm using m-crossover operator (pp: 127–130). *Proceedings of International Conference on Signal Processing, Image Processing and Pattern Recognition*, February 7-8, Coimbatore, India.
- Muhammad, M.S., Ibrahim, Z., Ueda, S., Ono, O., Khalid, M. (2005). DNA computing for complex scheduling problem. *Journal of Advances in Natural Computation* **3611**, 1182-1191.
- Nazni, P., Vimala, S. (2010). Nutrition knowledge, attitude and practice of college sportsmen. *Asian Journal of Sports Medicine* **1**, 93-100.
- Onwubolu, G.C., Clerc, M. (2004). Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research* **42**, 473-491.
- Ouaarab, A., Ahiod, B., Yang, X.S. (2014). Discrete cuckoo search algorithm for the travelling salesman problem. *Journal of Neural Computing and Applications* **24**, 1659-1669.
- Özdağ, R., Karçı, A. (2015). Sensor node deployment based on electromagnetism-like algorithm in mobile wireless sensor networks. *International Journal of Distributed Sensor Networks* **11**, 1-15.
- Passino, K.M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *Journal of IEEE Control Systems* **22**, 52–67.
- Pekcan, G. (2008). *Diyet El Kitabı*, Hatipoğlu Yayınevi, Ankara, Türkiye, 592 s.
- Pooranian, Z., Shojafar, M., Abawajy, J.H., Singhal, M. (2013). GLOA: a new job scheduling algorithm for grid computing. *International Journal of Artificial Intelligence and Interactive Multimedia* **2**, 59-64.
- Ramezani, F., Lotfi, S. (2013). Social-based algorithm (SBA). *Journal of Applied Soft Computing* **13**, 2837–2856.
- Rao, R.V., Patel, V. (2013). An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Journal of Scientia Iranica* **20**, 710–720.
- Rashedi, E., Nezamabadi-pour, H., Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Journal of Information Sciences* **179**, 2232–2248.
- Roza, A.M., Shizgal, H.M. (1984). The Harris Benedict equation reevaluated: Resting energy requirements and the body cell mass. *American Journal of Clinical Nutrition* **40**, 168-182.
- Sarria, A., Moreno, L.A., Garcia-Clop, L.A., Fleta, S., Morellan, M.P., Bueno, M. (2001). Body mass index, triceps skinfold and waist circumference in screening for adiposity in male children and adolescent. *Journal of Acta Paediatrica* **90**, 387-392.
- Satman, İ., Şengül, A.M., Uygur, S., Salman, F., Baştar, İ., Sargın, M., Tütüncü, Y., Karşıdağ, K., Dinççağ, N., Özcan, C., Yılmaz, M. T. (2002). Population based study of diabetes and risk characteristics in Turkey: results of the Turkish diabetes epidemiology study (TURDEP). *Journal of Diabetes Care* **25**, 1551-1556.

- Shang, G. (2005). The chaotic ant colony algorithm to solve TSP problem. *Journal of Systems Engineering-theory & Practice* **9**, 100.
- Shen, G., Zhang, Y.Q. (2011). A new evolutionary algorithm using shadow price guided operators. *Journal of Applied Soft Computing* **11**, 1983-1992.
- Siikonen, M.L. (1997). *Planning and control models for elevators in high-rise buildings*. Helsinki University of Technology Systems Analysis Laboratory Research Reports, Kone Corporation, Helsinki, Finland.
- Siikonen, M.L. (1997). *Elevator group control with artificial intelligence*. Helsinki University of Technology Systems Analysis Laboratory Research Reports, Kone Corporation, Helsinki, Finland.
- Simon, D. (2008). Biogeography-based optimization. *Journal of IEEE Transactions on Evolutionary Computation* **12**, 702–713.
- Smith, V.E. (1959). Linear programming models for the determination of palatable human diets. *Journal of Farm Economics* **41**, 272-283.
- Stigler, G.J. (1945). The cost of subsistence. *Journal of Farm Economics* **27**, 303-314.
- Storn, R., Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341–359.
- Suyanto, G.K.J. (2011). Evolutionary discrete firefly algorithm for travelling salesman problem (pp: 393-403). In: Bouchachia A. (Ed.), *Adaptive and Intelligent Systems*. Vol. 6943, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg.
- Tseng, S.P. (2016). An improved harmony search for travelling salesman problem. *Proceedings of 2nd IEEE International Conference on Computer and Communications*, October 14-17, Chengdu, China.
- Wong, L.P., Low, M.Y.H., Chong, C.S. (2008). A bee colony optimization algorithm for traveling salesman problem (pp: 818-823). *Proceedings of Second Asia International Conference on Modelling & Simulation (AMS)*, May 13-15, Kuala Lumpur, Malaysia.
- Wu, J.Y. (2010). Real-coded genetic algorithm-based particle swarm optimization method for solving unconstrained optimization problems (pp: 194-198). *Proceedings of International Conference on Electronics and Information Engineering (ICEIE 2010)*, August 1-3, Kyoto, Japan.
- Xie, L., Zeng, J., Cui, Z. (2009). General framework of artificial physics optimization algorithm (pp: 1321–1326). *Proceedings of World Congress on Nature & Biologically Inspired Computing*, December 9-11, Coimbatore, India.
- Xu, H., Qian, X., Zhang, L. (2012). Study of ACO algorithm optimization based on improved tent chaotic mapping. *Journal of Information & Computational Science* **9**, 1653-1660.
- Yıldırım, A.E., Karcı, A. (2013). Solutions of travelling salesman problem using genetic algorithm and atom algorithm (p: 134). *Proceedings of 2nd International Eurasian Conference on Mathematical Sciences and Applications (IECMSA-2013)*, International Sarajevo University, August 26-29, Sarajevo, Bosnia and Herzegovina.

- Yıldırım, A.E., Karcı, A. (2015). Bireye özgü optimum beslenme çizelgesinin yapay atom algoritması kullanılarak hazırlanması. *Mustafa Kemal Üniversitesi Tıp Dergisi* **6**, 1-11.
- Yıldırım, A.E., Karcı, A. (2017). Group elevator control optimization using artificial atom algorithm. *Proceedings of 2nd International Conference on Artificial Intelligence and Data Processing (IDAP-2017)*, Inonu University, September 16-17, Malatya, Turkey.
- Yıldırım, A.E., Karcı, A. (2017). Applications of artificial atom algorithm to small-scale traveling salesman problems. *Journal of Soft Computing* (in press).
- Yong, L.C. (2008). *Elevator traffic flow prediction using artificial intelligence*. Thesis, Malaysia University, Malaysia.
- Zhan, Z., Zhang, J., Shi, Y., Liu, H. (2012). A modified brain storm optimization (pp: 1–8). *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, June 10-15, Brisbane, QLD.
- Zhang, J.I., Tang, J., Zong, Q., Li, J.F. (2010). Energy-saving scheduling strategy for elevator group control system based on ant colony optimization (pp: 37-40). *Proceedings of IEEE Youth Conference on Information Computing and Telecommunications*, November 28-30, Beijing, China.

EKLER

Ek 1. Kahvaltılıklar İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Bal	315
Beyaz Peynir (Yağlı)	289
Beyaz Peynir (Yağsız)	99
Çikolata	528
Çökelek	215
Domates	22
Ekmek	276
Kaşar	404
Krem Peynir	349
Meyveli Yoğurt	125
Otlu peynir	327
Pastırma	250
Pekmez	293
Reçel	272
Salam	450
Salatalık	15
Sosis	322
Sucuk	452
Tahin helvası	516
Tereyağı	37
Tulum peyniri	215
Yoğurt	62
Yumurta	158
Zeytin (Siyah)	207
Zeytin (Yeşil)	144

Ek 2. Ana Yemekler İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Alabalık	168
Beyin	125
Böbrek	105
Brokoli	35
Bulgur Pilavı	215
Dalak	104
Dil	130
Etli Kuru Fasulye	133
Fajita	117
Fırında Tavuk	164
Humus	177
Hünkâr Beğendi	174
Ispanak	26
Kalkan	193
Karaciğer	140
Karnabahar	27
Karnıyarık	134
Kavurma	670
Kebap	215
Kıymalı Dolma	114
Kıymalı Pide	297
Kuşbaşı Pide	297
Kuzu Tandır	150
Lahmacun	147
Levrek	93
Palamut	168
Patates Püresi	83
Patates Salatası	143
Peynirli Makarna	370
Peynirli Pide	275
Pirinç Pilavı	352
Pizza	267
Rulo Köfte	254
Sığır Eti Güveci	95
Somon füme	171
Tavuklu Salata	127

Ek 2. (Devam) Ana Yemekler İçin Kalori Cetveli

Ton balığı	121
Uskumru	159
Yaprak Sarması	141
Yürek	124
Zeytinyağlı Dolma	173
Zeytinyağlı Enginar	166
Zeytinyağlı Kereviz	66
Zeytinyağlı Taze Fasulye	56

Ek 3. Çorbalar İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Bezelye Çorbası	75
Brokoli Çorbası	87
Domates Çorbası	30
Domatesli Pirinç Çorbası	47
Düğün Çorbası	53
Et Çorbası	33
Havuç Çorbası	25
Kremalı Brokoli Çorbası	45
Kremalı Kereviz Çorbası	37
Kremalı Mantar Çorbası	39
Kremalı Tavuk Çorbası	48
Köfteli Çorba	49
Mantar Çorbası	35
Mercimek Çorbası	56
Sebze Çorbası	28
Tarhana Çorbası	60
Tavuklu Sebze Çorbası	31
Yayla Çorbası	47

Ek 4. Tatlılar İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Burma tatlısı	318
Cevizli Baklava	285
Dondurma	193
Fıstıklı Baklava	329
Göllaç	168
Hanımöbeği	211
İrmik helvası	302
Lokma Tatlısı	518
Kadayıf	244
Milkshake (çikolatalı)	120
Milkshake (vanilyalı)	110
Muhallebi	173
Profiterol	236
Söllaç	139
Şekerpare	282

Ek 5. Kuruyemişler İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Ay çekirdeği	560
Antep fıstığı	562
Badem	598
Ceviz	651
Fındık	634
Kabak çekirdeği	610
Kaju	553
Kestane	194
Kuru incir	274
Kuru kayısı	260
Kuru üzüm	290
Leblebi	350
Patlamış mısır	386
Şam fıstığı	594
Yer fıstığı	582

Ek 6. Meyveler İin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Ahududu	57
Ananas	52
Armut	61
Avokado	167
Ayva	57
Böğürtlen	58
Çilek	93
Elma	58
Erik	66
Greyfurt	41
İncir	80
Karpuz	26
Kavun	33
Kayısı	51
Kiraz	70
Kivi	30
Mandalina	46
Muz	85
Nar	63
Portakal	49
Şeftali	38
Üzüm	67
Vişne	58

Ek 7. İçecekler İçin Kalori Cetveli

Besin	Kalori (kkal) (100gr)
Ayran	37
Adaçayı	1
Buzlu çay	30
Çay	0
Elma suyu	47
Ihlamur	1
Kayısı suyu	44
Kola	39
Meyveli gazoz	46
Portakal suyu	45
Salep	95
Sıcak Çikolata	77
Süt (Tam Yağlı)	61
Süt (Yarım Yağlı)	50
Süt (Yağsız)	35
Şeftali suyu	55
Türk Kahvesi (orta)	17
Vişne suyu	58

Ek 8. Besin Gruplarına Göre Belirlenen Tercih Katsayıları

Besin Grubu	Tercih Katsayısı
Kahvaltılıklar	0,5
Ana Yemekler	2
Çorbalar	2
Tatlılar	1
Kuruyemişler	0,2
Meyveler	1
İçecekler	1

ÖZGEÇMİŞ

Ad Soyadı: Ayşe Erdoğan Yıldırım

Doğum Yeri ve Tarihi: Elazığ – 15.08.1985

Adres: Fırat Üniversitesi Bilgisayar Mühendisliği Bölümü / Elazığ

E-Posta: ayseerdogan@firat.edu.tr , aerdoganyildirim@gmail.com

Lisans:

- Fırat Üniversitesi-Mühendislik Fakültesi-Bilgisayar Mühendisliği Bölümü-2008
- Anadolu Üniversitesi-İşletme Fakültesi-İşletme Bölümü-2009

Yüksek Lisans: Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Müh. ABD, Yazılım BD

Mesleki Deneyim ve Ödüller:

- Zemin Harita Müh. Müt. San. Tic. Ltd. Şti. (2009) – Yazılım Geliştirici
- Yonca CBS Bilişim Sistemleri Bilgisayar - Harita Müh. Şti. (2009-2010) – Yazılım Geliştirici
- İçişleri Bakanlığı Bilgi-İşlem Dairesi Başkanlığı (2010-2011) – Programcı
- Fırat Üniversitesi Müh. Fak. Bilgisayar Mühendisliği Bölümü (2011- Halen) - Araştırma Görevlisi

DOKTORA TEZ KAPSAMINDA TÜRETİLEN AKADEMİK YAYINLAR

Makale:

- **Yıldırım, A.E.,** Karcı, A. (2015). Bireye özgü optimum beslenme çizelgesinin yapay atom algoritması kullanılarak hazırlanması. *Mustafa Kemal Üniversitesi Tıp Dergisi*. 6(24), 1-11.
- **Yıldırım, A.E.,** Karcı, A. (2017) Applications of artificial atom algorithm to small-scale traveling salesman problems. *Springer Soft Computing* (in press). doi: <https://doi.org/10.1007/s00500-017-2735-z>.

Bildiri:

- **Yıldırım, A.E.,** Karcı, A. (2013). Solutions of travelling salesman problem using genetic algorithm and atom algorithm (p:134). *Proceedings of 2nd International Eurasian Conference on Mathematical Sciences and Applications (IECMSA-2013)*, International Sarajevo University, August 26-29, Sarajevo, Bosnia and Herzegovina.
- **Yıldırım, A.E.,** Karcı, A. (2017). Group elevator control optimization using artificial atom algorithm. *Proceedings of 2nd International Conference on Artificial Intelligence and Data Processing (IDAP-2017)*, Inonu University, September 16-17, Malatya, Turkey.