



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



MOTSA: Çok Amaçlı Ağaç Tohum Algoritması

Gül ÖZCAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Eylül-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gül ÖZCAN tarafından hazırlanan “Çok Amaçlı Ağaç Tohum Algoritması” adlı tez çalışması 05/09/2019 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Mesut GÜNDÜZ

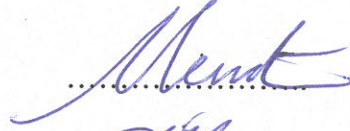
Danışman

Doç. Dr. Mustafa Servet KIRAN

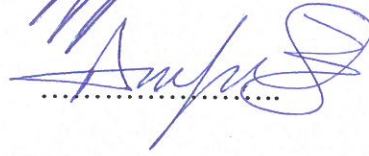
Üye

Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN

İmza

.....


.....


.....


Yukarıdaki sonucu onaylarım.

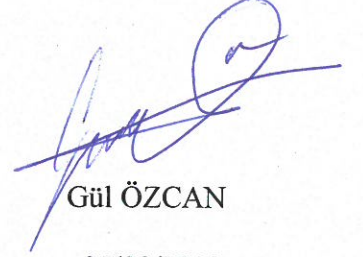
Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.



Gül ÖZCAN

05/09/2019

ÖZET

YÜKSEK LİSANS TEZİ

ÇOK AMAÇLI AĞAÇ TOHUM ALGORİTMASI

Gül ÖZCAN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mustafa Servet KIRAN

2019, 64 Sayfa

Jüri

Doç. Dr. Mustafa Servet KIRAN
Doç. Dr. Mesut GÜNDÜZ
Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN

Gerçek dünya problemlerinde belirlenen amaçlar genellikle birbiriyle çelişen birden fazla amacın eş zamanlı optimizasyonundan oluşmaktadır. Bu noktada tek amaçlı bir optimizasyon yönteminin çok amaçlı versiyonunu oluşturulması birden fazla soruna aynı anda çözüm üretme imkânı sağlayacaktır. Ağaç Tohum Algoritması, kısaca TSA, ağaçlarla tohumların doğadaki ilişkisinde ilham alınarak tek amaçlı optimizasyon algoritması çözümü için önerilmiştir. Önerilen çok amaçlı varyant ağaç tohumu algoritması MOTSA, TSA'nın tek amaçlı problemler üzerindeki performansını temel alarak çok amaçlı optimizasyon problemlerini çözmek için önerilmiştir. Önerilen algoritma birçok çok amaçlı optimizasyon algoritması yayılım, dağılım ve yakınsama başarı ölçütleri kullanılarak test edilmiştir. Java ortamında gerçekleştirilen testlerde 9 farklı uygunluk fonksiyonu kullanılmıştır. Bu çalışmada belirlenen kriterlere göre yapılan testlerde elde edilen sonuçlar sunulmuştur. MOTSA'nın elde ettiği başarı değerlendirilmiştir. Algoritmanın optimum sonuca ulaşma yöntemindeki özellikleri değerlendirilmiştir.

Çok amaçlı problemlerde tek amaçlıdan farklı olarak seçim sorununun üstesinden gelmek için, NSGA-II'nin iyi bilinen stratejileri, baskın olmayan sıralama ve kalabalık mesafesi özellikleri önerilen MOTSA ile entegre edilmiştir. Bunu yaparak, MOTSA algoritmasında birleşik ağaç ve tohum popülasyonlarından en yüksek kalitede çözümler seçilir ve bir sonraki nesle aktarılır. TSA ve NSGA-II algoritmalarının en iyi yönlerinin sentezlenmesi ile çok amaçlı optimizasyon problemlerine MOTSA algoritması ile çözüm bulunmaktadır.

Anahtar Kelimeler: Çok Amaçlı Optimizasyon, MOTSA, NSGA-II, Optimizasyon, TSA

ABSTRACT

MS THESIS

MOTSA: A MULTI-OBJECTIVE VARIANT OF TREE SEED ALGORITHM

Gül ÖZCAN

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineer**

Advisor: Assoc. Prof. Dr. Mustafa Servet KIRAN

2019, 64 Pages

Jury

**Assoc. Prof. Mustafa Servet KIRAN
Assoc. Prof. Mesut GÜNDÜZ
Asst. Prof. Dr. Mehmet Akif ŞAHMAN**

The goals identified in real-world problems often consist of concurrently optimization of multiple conflicting goals. At this point, the creation of a multi-objective version of a single-objective optimization method will provide the opportunity to produce solutions to multiple problems concomitantly. Tree Seed Algorithm, TSA for short, has been proposed for the solution of a single-targeted optimization algorithm by inspiring the relationship between trees and seeds in nature. A multiobjective variant tree seed algorithm MOTSA has been proposed to solve multiobjective optimization problems based on the performance of the TSA on single objective problems. The algorithm that proposed has been tested using several criterions optimization algorithms, propagation and convergence success criteria. 9 different conformity functions were used in the tests that performed in Java. In the study, the results obtained in the tests performed according to the specified criteria that are determined. The success that achieve of MOTSA was evaluated. The properties of the algorithm in reaching the optimum result method were evaluated.

In order to overcome the problem of selection, as different to single-purpose for multi-objective problems, NSGA-II's well-known strategies, non-dominant sorting and crowd distance features are integrated with the proposed MOTSA. By doing this, the highest quality solutions are selected from the combined tree and seed populations in the MOTSA algorithm and passed on to the next generation. By synthesizing the best aspects of TSA and NSGA-II algorithms, MOTSA algorithm is used to solve multi-objective optimization problems.

Keywords: MOTSA, Multi-Objective Optimization, NSGA-II, Optimization, TSA

ÖNSÖZ

Tez çalışmamda planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. Mustafa Servet KIRAN'a sonsuz teşekkürlerimi sunarım. Yine çalışmamda konu, kaynak ve yöntem açısından bana sürekli yardımda bulunarak yol gösteren sayın hocam Dr. Öğr. Üyesi Ahmet ÖZKIŞ'a teşekkürlerimi sunarım.

Teşekkürlerin az kalacağı diğer üniversite hocalarımdan da bana üniversite ve yüksek lisans hayatım boyunca kazandırdıkları her şey için teşekkürlerimi sunuyorum.

Gül ÖZCAN
KONYA-2019

İÇİNDEKİLER

T.C.....	iv
TEZ KABUL VE ONAYI	v
TEZ BİLDİRİMİ	vi
ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	x
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	8
2.1. Ağaç-Tohum Algoritması (TSA)	8
2.2. Genetik Algoritma (GA).....	15
2.2.1. Bireylerin seçimi.....	17
2.2.3. Çaprazlama işlemi	18
2.2.3. Mutasyon işlemi	18
2.3. Bastırılmamış Sınıflandırma Genetik Algoritma II -NSGAI.....	19
2.3.1. Kalabalık mesafe (Crowding Distance)	21
2.3.2. Bastırılmamış sıralama (Non-Dominated Sorted)	22
2.3.3. Elitizm işlemi	23
3. MATERYAL VE YÖNTEM.....	25
3.1. Materyal Listesi.....	25
3.2. Yöntem.....	28
3.3. MOTSA Akış Diyagramı.....	29
3.4. MOTSA Algoritma Modelinin Oluşturulması	30
3.4. MOTSA Modelleme Süreci Benchmark Testleri	32

3.4.1. Modelleme test fonksiyonları.....	32
3.4.1.1. Kursawe fonksiyonu	33
3.4.1.2. Fonseca fonksiyonu.....	34
3.4.1.3. Poloni fonksiyonu.....	35
3.5. MOTSA: A Multi-Objective Tree Seed Algorithm	36
5.6. Sonuçların Elde Edilme Aşaması	38
3.6.1. Başarı ölçütleri	38
3.6.1.1. Hypervolume	39
3.6.1.2. Spread	40
3.6.1.3. Invented Generational Distance (IGD)	40
3.6.1.4. Epsilon.....	40
3.6.2. Karşılaştırılan algoritmalar	41
3.6.2.1. AbYSS (Multiobjective Scatter Search)	41
3.6.2.2. CelIDE (Cellular Genetic Algorithm with Differential Evolution)	42
3.6.2.3. DENSEA (Duplicate Elimination Nondominated Sorting Evolutionary Algorithm)	42
3.6.2.4. dMOPSO (Decomposition-Based Multi-Objective Particle Swarm Optimizer).....	42
3.6.2.5. GDE3 (Generalized Differential Evolution)	42
3.6.2.6. WASFGA (Weighting Achievement Scalarizing Function Genetic Algorithm)	43
3.6.2.7. GWASFGA (Global Weighting Achievement Scalarizing Function Genetic Algorithm)	43
3.6.2.8. MOCeLL (Multiobjective Cellular Genetic Algorithm)	43
3.6.2.9. IBEA (Indicator-Based Evolutionary Algorithm)	43
3.6.2.10. MOCHC (Multiobjective CHC Algorithm)	44
3.6.2.11. MOEAD (Multiobjective Evolutionary Algorithm with Decomposition)	44
3.6.2.12. OMOPSO (Multiobjective Particle Swarm Optimization)	44
3.6.2.13. PAES (Pareto Archived Evolution Strategy)	44

3.6.2.14. PESA2 (Pareto Envelope-based Selection Algorithm)	44
3.6.2.15. SMPSO (Speed-Constrained Multiobjective Particle Swarm Optimization)	45
3.6.2.16. SPEA2 (Strength-based Evolutionary Algorithm)	45
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	46
4.1. Araştırma Sonuçları	46
4.2. Tartışma.....	58
KAYNAKLAR	59
ÖZGEÇMİŞ	64



SİMGELER VE KISALTMALAR

Simgeler

P_0	Başlangıç ebeveyn popülasyonu
P_{t+1}	Yeni nesil popülasyon
Q_0	Çocuk popülasyonu
$I_{distance}$	Küboidin çevre uzunluğu
R	Bileşik popülasyon
p	Çözüm uzayında bulunan bir çözüm
q	Çözümün üstünlük değeri
N	Popülasyon boyutu
F_1	1. Amaç fonksiyonu
F_2	2. Amaç fonksiyonu
Δ	Spread başarı ölçütü

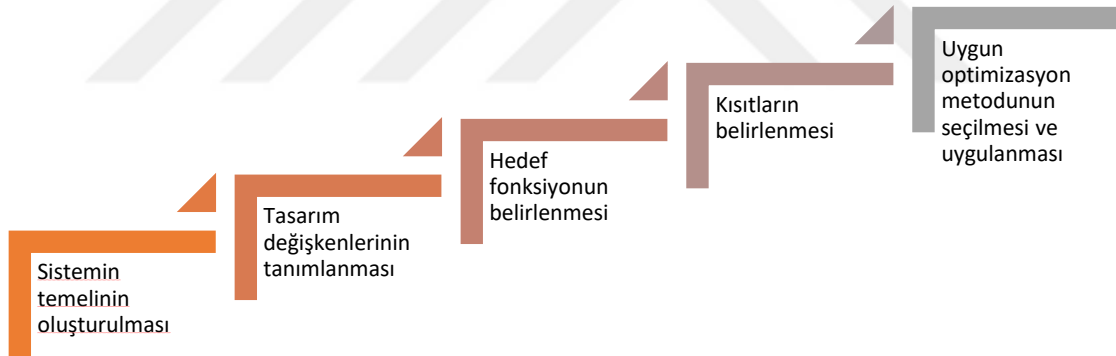
Kısaltmalar

NSGA-II	Sıralı Seçkin Bastırılmayan Genetik Algoritma
GA	Genetik Algoritma (Genetic Algorithm)
SI	Sürü Zekâsı (Swarm Intelligence)
AI	Yapay Zekâ (Artificial Intelligence)
GP	Genetik Programlama (Genetic Programming)
ANN	Yapay Sinir Ağları (Artificial Neural Network)
TSA	Ağaç-Tohum Algoritması (Tree-Seed Algorithm)
ACO	Karınca Koloni Optimizasyonu (Ant Colony Optimization)
PSO	Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
ABC	Yapay Arı Kolonisi (Artificial Bee Colony)
EC	Evrimsel Hesaplama (Evolutionary Computation)
ST	Araştırma eğilimi (Search Tendency)
UMA	Tek düze bellek erişimi (Uniform Memory Access)
CPU	Merkezi İşlem Birimi (Central Processing Unit)
KNN	K En Yakın Komşu
PCA	Temel Bileşen Analizi
VK	Veri Kümesi
RS	Rulet Çarkı Seçimi
TS	Turnuva Seçimi
CD	Yoğunluk Mesafesi
SF	Sınıflandırma Fonksiyonu
EA	Evrimsel Algoritma
HV	Hypervolume
IGD	Inverse Generational Distance

1. GİRİŞ

Teknolojik gelişimlerin belirlenen bir hedefe yönelik olduğu düşünülmektedir. Ancak bu durumun aksine birden fazla probleme çözüm bulan gelişmeler gelecek nesillere aktarılabilir. Bu durumun temelinde gerçek dünyada karşılaşılan problemlerin aynı zaman diliminde çoğu zaman birbirine zıt çözümleri hedefleyen problemler olmasıdır. Tüm hedefler için aynı anda istenen koşulun elde edilmesi, birbiriyle çelişen hedefler için bir gerekliliktir (Seifollahi-Aghmiuni ve Bozorg Haddad, 2018).

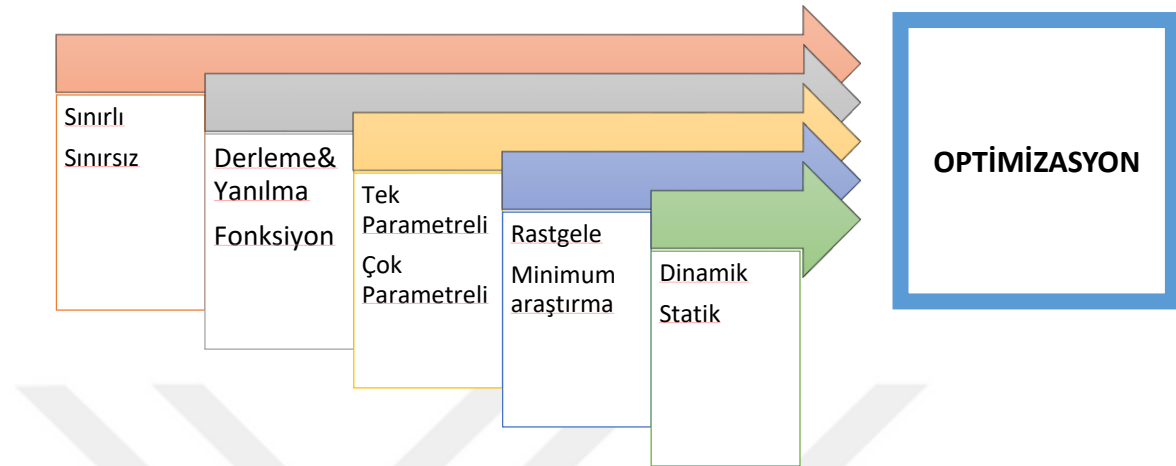
Optimizasyon terimi belirlenen hedefe ulaşan en uygun çözümü bulma yöntemi olarak karşımıza çıkmaktadır. Temelde problem çözümünde amaçların belirlenme sürecinin tamamlanmasından problem çözümüne kadar olan akış, Şekil 1.1’de gösterilen optimizasyon aşamalarını oluşturmaktadır. Gerçek dünyada karşılaşılan problemlerin matematiksel olarak modellenmesi, hedeflerin ve bu hedefler doğrultusunda ortaya çıkan kısıtların belirlenmesi işlemi optimizasyon sürecini başlatır. Optimizasyon modeli kısaca, verilen kısıtlamaları sağlayan karar değişkenleridir (Winston, 2003).



Resim 1.1. Optimizasyon aşamaları

Optimizasyon ve mühendislik terimleri birbiri ile belirli amaca en uygun çözümü optimum kaynak tüketimi ile sağlama bakımından benzerlik göstermektedir. Bu durum ışığında optimizasyon algoritmalarının gelişimi öngörülebilir. Mevcut kaynak tüketimi ve sürdürülebilir teknoloji gelişiminde optimizasyon algoritmaları önemli rol almaktadır. Bu süreçte var olan kaynakların (zaman, maliyet, işgücü vb.) en uygun kullanımı ile optimum sonuca ulaşma yolunda birçok yöntem geliştirilmiştir. Problem hedef fonksiyonuna, çözüm kümesine, aldığı değişken türü gibi özelliklere göre

sınıflandırılan optimizasyon işleminde uygun programlama yöntemi seçilmelidir, bazı durumlarda birden fazla programlama yöntemi aynı anda kullanılabilir. Resim 1.2’de bu optimizasyon sınıfları gösterilmektedir.

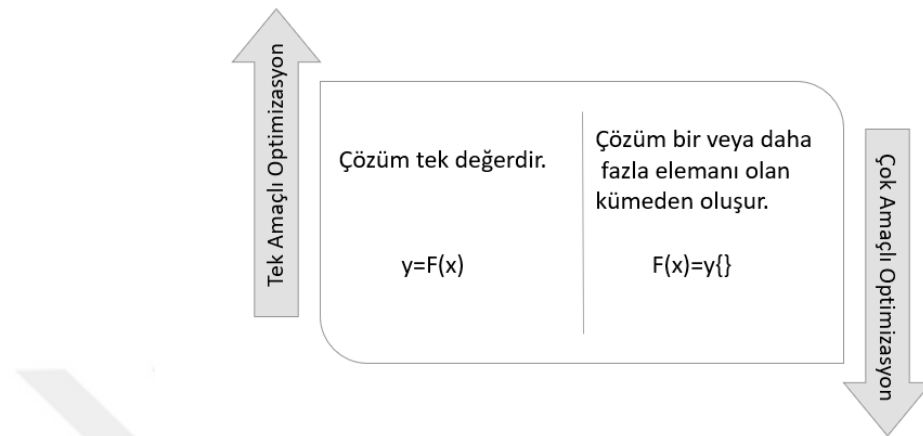


Resim 1.2. Optimizasyon sınıfları

Optimizasyon teknolojisi, karar verme süreçlerini hızlandırmakta ve karar kalitesini arttırmakta kullanılarak gerçek hayatta karşılaşılan problemlerin etkin, doğru ve gerçek zamanlı çözümünde yararlanılmaktadır (Winston, 2003). Optimizasyon işlemi temelde iki aşamaya ayırırsak ilk aşama problemin belirlenmesi ve matematiksel modelinin oluşturulması olabilir. İkinci aşama da problemin çözüm metodunun işlenmesi kabul edersek, probleme uygun olarak seçilen programlama yöntemlerinin kullanılmasıdır. Optimizasyon programlama yöntemlerinde temelde probleme atanan amaç yani uygunluk fonksiyonunu çözüm değerine en hızlı ve uygun kaynak kullanımı ile ulaşımının sağlanmasıdır. Bu programlama türleri performans, kaynak tüketimi gibi durumları göz önünde bulundurularak oluşturulan ölçütlere göre başarı değeri belirlenmektedir. Optimizasyon teknolojisinin daha başarılı sonuçlar için geliştirilme sürecinde matematiksel modelin oluşturulmasına öncelik gösterilmiştir (Türkay, 2019).

Uzun yıllar boyunca tek hedefli optimizasyon problemleri üzerinde çalışılmış ve bu problem çözümünde sıklıkla evrimsel teknikler kullandıktan sonra, amaç fonksiyonuna birden fazla hedefin eklenmesi ile dikkat çekici bir araştırma alanı doğmuştur (Coello Coello, 2002). Birden fazla amacın, aynı zaman diliminde ortak kaynak tüketimi ile çözüm kümesinin belirlenmesine çok amaçlı optimizasyon denir. Çok amaçlı optimizasyon problemlerine temel yaklaşım, tek bir çözüm değil, bir dizi egemen olmayan çözüm bulmaya çalışmaktır (Ishibuchi ve ark., 1997). Tek amaçlı fonksiyondan farklı olarak amaç fonksiyonunun modellenmesinin daha zor olduğu çok amaçlı

fonksiyonlarda tek bir matematiksel modellenmiş fonksiyonla birden fazla amaca çözüm bulmak her zaman mümkün olmamaktadır. Bu durumlarda birden fazla çözüm söz konusu olmakta, elde edilen sonuç çözüm kümesini oluşturmaktadır. Resim 1.3'te optimizasyon türlerine göre elde edilen çözüm gösterilmektedir.



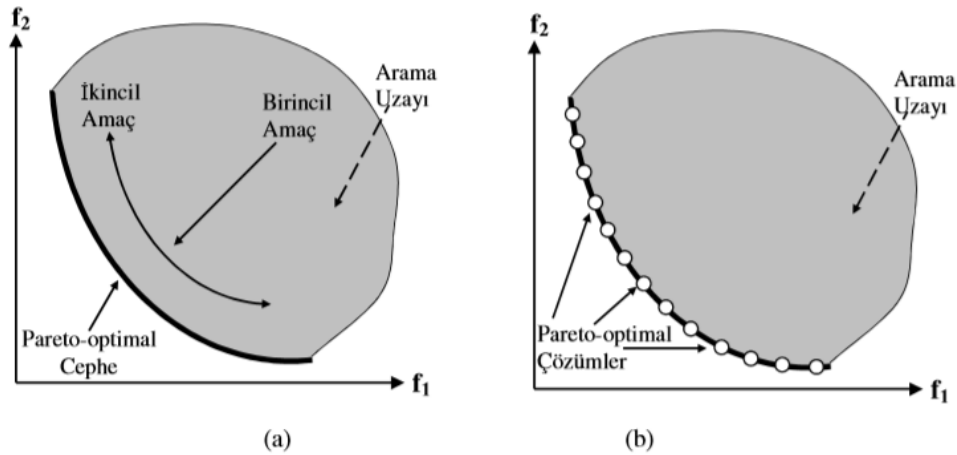
Resim 1.3. Optimizasyon türleri

Optimizasyon problem çözüm metotları problemde bulunan amaç sayısına göre farklılık göstermektedir. Bir amaca yönelik problemlerde belirlenen hedefe ulaşmada optimum çözüme ulaşırken sadece o problem ve kısıtları göz önünde bulundurulurken birden fazla amaç hedeflendiğinde çözüm yöntemi farklılık göstermektedir. Bu kapsamda uygulanan yöntemden elde edilen sonuç çözümü değil çözüm kümesini ifade etmektedir. Algoritma bu çözüm kümesinden en uygun çözümü seçerken hangi amacın diğer amaca göre daha üstün olduğunu belirleyemediği için sonuç kümesini oluşturur ve kullanıcıya sunar. Problemlere ve problemlerin çözücüsüne bağlı olarak kullanıcı, sunulan çözümlerden kendisi için en uygun olanı seçer. Bu olay tek amaçlı ve çok amaçlı optimizasyon arasındaki yaklaşım farkını ortaya koymaktadır.

Çeşitli amaçlara sahip olan bu problemlere “çoklu hedef” veya “vektör” optimizasyon problemleri de denir ve başlangıçta ekonomi bağlamında incelenmiştir (Coello Coello, 2002). Son yıllarda çok amaçlı optimizasyon alanında yapılan çalışmalar sonucunda optimal çözüm kümesini bulmada tüm çözüm uzayını hedef alan yöntemler geliştirilmiştir. Bu yöntemlere sabit ağırlıklı amaç fonksiyonu, değişken ağırlıklı amaç fonksiyonu, vektör hesaplamalı genetik algoritma ve pareto optimal yöntemleri örnek olarak verilebilir.

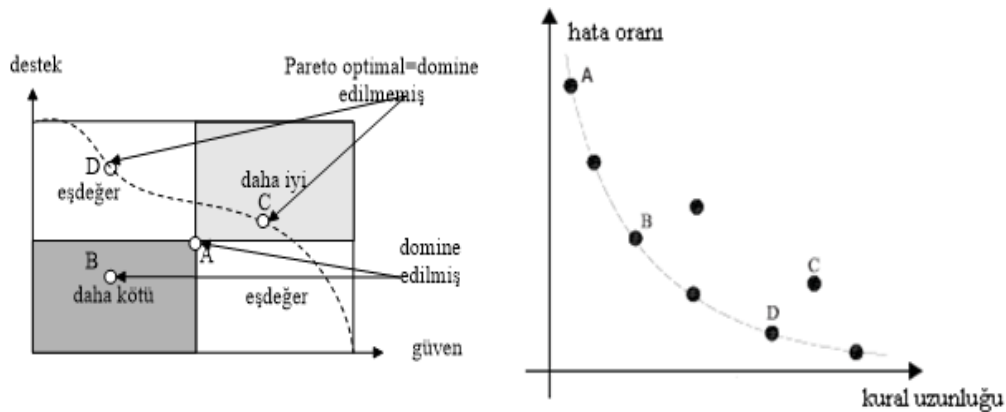
Pareto-optimal cephenin matematiksel olarak ifade edilmesi işlemlerin başlangıcı ve problem için önem arz ettiğinden, bazı karmaşık Pareto-optimal cephelerin

matematiksel olarak yazılması zor olabilir (Deb ve ark., 2005). Pareto optimal ve birbirini bastıran çözümler kavramları birlikte açıklanabilir. Resim 1.4’ de gösterilen destek ve güven amaçlarının maksimum problem çözümlerinde bulunan dört noktanın birbirine karşı baskınlıkları ifade edilmektedir.



Resim 1.4. (a) Çok amaçlı optimizasyonda ulaşılmaması istenen iki amaç
(b) İdeal bir Pareto-optimal çözüm kümesi (Ergül, 2015)

Çok amaçlı bir probleme optimal bir çözüm, her biri diğer çözümün hakimiyetine girmeden, amaçları kabul edilebilir bir düzeyde yerine getiren bir çözüm kümesinin araştırılmasıdır (Konak ve ark., 2006). Resim 1.5’te minimum edilen problem için örnek Pareto-optimal cephe gösterilmektedir. Burada Pareto yüzeyi üzerinde bulunan C ve D noktaları için biri diğerinden daha iyi veya kötü ifadesinin kullanılmadığı, A ve B noktalarının ise maksimum optimizasyon probleminde C ve D noktalarına göre daha kötü oldukları söylenebilmektedir. Dolayısıyla A ve B noktaları C ve D nokta çözümleri tarafından domine edilmektedir. Diğer grafik ise minimum optimizasyon için A, B ve D nokta çözümlerinin, C çözümünü domine ettiğini göstermektedir. Pareto-optimal çözümlerin hiçbiri diğer çözümlerden daha iyi olarak sınıflandırılmadığından, çok amaçlı optimizasyon mantığında öncelikli hedef mümkün olduğu kadar çok Pareto-optimal çözüm elde etmektir (Deb, 1999).



Resim 1.5. Pareto-Optimal Kavramı (M Kaya, 2007)

Problem türü bakımından oldukça fazla konuyu ele alabilen çok amaçlı algoritmalar geniş bir uygulanma alanına sahiptir (A. Coello Coello ve ark., 2007). Çok amaçlı optimizasyonun iki temel amacı vardır:

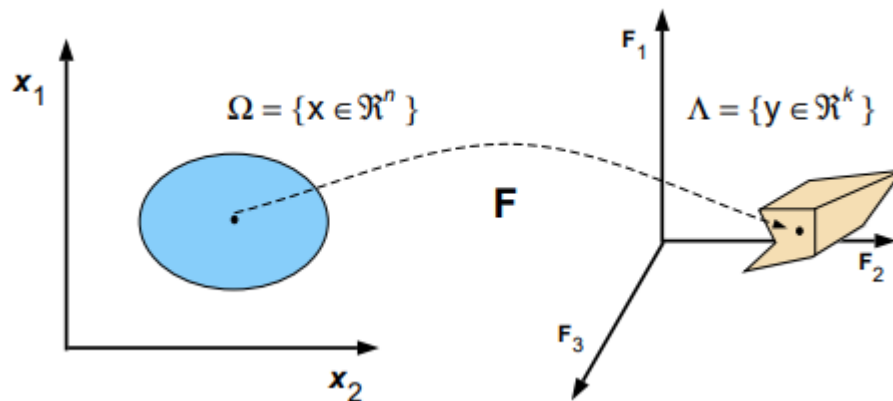
- (i) Pareto-optimal cepheye (önceden biliniyorsa) mümkün olduğunca yakın çözümler bulmak (yakınsama),
- (ii) Bulunan bastırılmayan(Pareto) bireylerin Pareto cephesinde mümkün olduğunca düzgün dağılması (dağılım veya çeşitlilik) (Ergül, 2015).

Çok amaçlı optimizasyonların temelindeki bu iki amaç, yakınsama ve dağılım, çözüm kümesinde meydana gelen pareto optimal cepheyi oluşturmaktadır. Çok amaçlı bir optimizasyon problemi, minimize edilmesi veya maksimize edilmesi gereken bir dizi objektif fonksiyon içerir (Deb, 2011). Yakınsama amacı olarak elde edilen sonuçlar probleme bağlı olarak maksimize ya da minimize bir cephe ortaya çıkarmalıdır. Yine çok amaçlı optimizasyonda dağılım amacını ele aldığımızda oluşan cephede bulunan çözümlerin birbirine uzaklığı ve tek bir yüzey doğrusu oluşturacak şekilde yerleşimidir. Bu iki ana amaç doğrusunda çok amaçlı optimizasyon algoritmalarının başarı kriterleri hesaplanmaktadır. Çok amaçlı optimizasyon algoritmalarıyla birlikte bu algoritmalar arasında başarı kriterini belirlemek amacıyla birçok ölçüt geliştirilmiş ve hala geliştirilmektedir. Bu noktada yaygın olarak kullanılan başarı ölçütleri:

- C-metriği (C-measure) (Zitzler, 1999)
- Ters Nesilsel Mesafe (Inverted Generational Distance, IGD) (Zitzler ve ark., 2003)

- Hypervolume (HV) (Zitzler ve Thiele, 1999)
- Epsilon (ϵ) Indicator (Zitzler ve ark., 2003)
- Nesilsel Mesafe (Generational Distance, GD) (Deb, 2001)
- Yayılım (Spread) (Deb, 2001)

Çok amaçlı optimizasyon algoritmaları arama (popülasyonda bulunan her bir çözümün ifade ettiği matematiksel değer) uzayı ve çözüm (arama uzayında bulunan değerlerin probleme verdikleri çözüm değeri) uzayı olmak üzere iki keşif alanı kullanmaktadır. Resim 1.6'da gösterilen ilk grafik çok amaçlı optimizasyon problemlerinin arama uzayını ifade ederken ikinci grafik 3 boyutlu bir optimizasyon problemi için çözüm uzayının grafiğini oluşturur. Arama uzayında bulunan başlangıç popülasyonun evrimsel arama yöntemleri ile çözüm uzayında Pareto optimal cepheyi oluşturması olarak akış izlerler. Optimal çözüme ulaşma aşamasında, başlangıç popülasyonu zamanla çözüm uzayında pareto cephelerini oluşturmaktadır. Bu cepheler birbiri ile kıyaslanamayan çözümlerin sınıflandırılmış halini ifade eder.



Resim 1.6. Çok amaçlı optimizasyon haritası (Lamont, 1999)

1970'li yıllardan itibaren optimizasyon problemlerinin çözümünde kullanılan klasik yöntemler, gerçek hayat problemlerini çözmede yetersiz kaldığı, çözüm sürelerinin ve algoritma karmaşıklığının problem boyutuyla orantılı olarak artması, sezgisel araştırma yöntemlerinin gelişmesini sağlamıştır (Erdoğan, 2016). Bu sezgisel araştırma yöntemlerinin birçoğu doğadan esinlenilerek geliştirilmiştir. Doğadaki birçok canlı sahip oldukları kusursuz tasarımlarıyla var olan kaynakları minimum kullanarak, maksimum başarı elde ederek en zorlu şartlarda bile hayatta kalmışlardır (Erdoğan, 2016). Temelde belirlenen probleme bağlı olarak sezgisel tahminler kullanarak çözüm oluşturma yöntemi olarak ortaya çıkan sezgisel algoritmalar birçok optimizasyon algoritmasının ana

mantığını oluşturmaktadır. Doğal yaşamın matematiksel olarak modellenmesi ile oluşturulan algoritmalar sahip oldukları çözüm havuzu, işlevselliği, hızlı ve kolay entegre edilebilmesi ve etkin çalışabilmesi nedeniyle optimizasyon problemi çözümünde sıklıkla tercih edilmektedir. Bu algoritmalar zor problemlerin çözümünde göstermiş oldukları başarılarından dolayı birçok bilim dalında, mühendislik problemlerinde ve askeri uygulamalarda sıklıkla kullanılmaktadır (Haklı, 2013).

Gerçek dünyada bulunan birbiriyle çelişen problemlere en hızlı ve kolay yolla çözüm bulunması noktasında geliştirilen algoritmaların farklı yöntemler kullanılarak daha da güçlendirilmesi sağlanmalıdır. Gelişen teknolojiyle birlikte problemlerde ve yöntemlerde gelişim göstermekte, her geçen gün farklı amaçlara yönelik algoritmalar geliştirilmektedir. Bu tez çalışmasında ise doğal yaşamdan sentezlenerek elde edilen algoritmaların yaygın kullanımı ile geliştirilmesindeki öneminin artması doğrultusunda sürekli optimizasyon problemlerini çözmek maksadıyla önerilen Ağaç-tohum algoritmasının (Tree-seed algorithm- TSA) (Kıran, 2015) çok amaçlı optimizasyon işlemleri için bir varyantı oluşturulmuştur. Çok amaçlı TSA ile birden fazla probleme eş zamanlı olarak çözüm bulunması sağlanmıştır. Çalışmada son yıllarda yaygın olarak kullanılan doğal yaşamın matematiksel modellenmesi ile geliştirilmiş sezgisel tek amaçlı optimizasyon algoritması TSA 'nın çok amaçlı optimizasyon problemlerinde en çok kullanılan NSGA-II ile bilinen en iyi yönlerinin sentezlenmesi ile Çok Amaçlı Ağaç-Tohum Algoritması (MOTSA) tanıtılmıştır.

Çalışma sonuçları elde edilme sürecinde MOTSA kaynak kodlaması algoritmanın gelişimi tamamlanana kadar MATLAB ortamında gerçekleştirilmiştir. Modelleme süreci tamamlanan algoritma java (NetBeans) ortamında JMetal standartlarına uygun olarak kodlanmıştır. Günümüz teknolojisinde açık kaynak olarak sunulan bu platform kullanılarak MOTSA java kaynak kodları üretilmiş, belirlenen 16 algoritma ile 9 kıyaslama fonksiyonu üzerinde test sonuçları elde edilmiştir.

Tez çalışmasının bundan sonraki bölümünde MOTSA'nın temelini oluşturan TSA'nın kaynak araştırması kısmı yer almaktadır. TSA ile gerçekleştirilen mevcut çalışmalardan bahsedilmiştir. Genetik Algoritma (GA) ve Bastırılmamış Sıralama Genetik Algoritma (NSGA-II) 'larının özellikleri incelenmiş; üçüncü bölümde bu çalışma kapsamında kullanılan materyal ve yöntemlerden; dördüncü bölümde ise MOTSA'nın 9 uygunluk fonksiyonu üzerinde elde edilen sonuçlar değerlendirilmiş, algoritmanın özgün ve geliştirilebilir yönleri belirlenerek gelecekte yapılabilecek çalışmalar hakkında bilgi verilmiştir.

2. KAYNAK ARAŞTIRMASI

2.1. Ağaç-Tohum Algoritması (TSA)

Kıran tarafından 2015 yılında önerilen Ağaç Tohum Algoritması (TSA), sürekli optimizasyon problemlerinin çözümü için yeni bir meta-heuristik algoritmadır (Kıran, 2015). Popülasyon temelli algoritma doğada bulunan ağaç tohum ilişkisini modellemektedir. Arama uzayında birçok noktadan arama işlemi gerçekleştiren algoritma yeni nesil oluşturmak için ağaçtan tohumları üretmektedir. TSA'da bir ağaçtan üretilen tohum sayısı rastgele değer olarak elde edilir. Optimizasyon probleminde çözüm uzayında arama işleminde ağaçlar ve tohumları temel alan algoritma yinmeli, sezgisel araştırma mantığında şekillenmiştir.

Resim 2.1'de TSA algoritmasının sözde kodu verilmiştir. Bir problem için olası tüm çözümlerin ağaçlar ve tohumlar olarak ifade edildiği algoritmada başlangıç popülasyonu belirlenen miktarda rastgele olarak oluşturulmaktadır.

N miktarda oluşturulan popülasyonda her bir ağaç için modellenmesi yapılmış problemin amaç fonksiyonu için değeri hesaplanır. Amaç değerlerinin hesaplanmasının ardından çözüm araştırma yöntemi her bir ağaçtan tohum üretme yöntemiyle gerçekleşir. Üretilen tohumların amaç fonksiyon değerleri hesaplandıktan sonra kendi ebeveyn ağacı ve tohum kümesiyle amaç değeri optimizasyon probleminin minimize ya da maksimize olmasına bağlı olarak kıyaslama yapılır. Bu kıyas grubundan çıkan başarılı tohum ya da ağaç bir sonraki nesle aktarılmak üzere yeni neslin ağacı olarak kabul edilir. Her bir ağaç ve tohumların arasında yapılan grup kıyaslamasının ardından yeni nesil olarak oluşturulan popülasyon arasından en optimize ağaç tespit edilir. Algoritma başlangıcında belirlenen maksimum sayıya ulaşıncaya kadar algoritma tohum üretim ve kıyaslama işlemi devam eder. Maksimum sayıya ulaşıldığında elde edilen en optimize sonuç algoritmanın ilgili değerler için probleme bulunduğu çözüm olarak kabul edilir.

Adım 1: Algoritma parametrelerini belirleme

- 0 ve 1 arasında ST parametresinin değerini belirle
- Problemin boyutunu ve popülasyon içindeki bireylerin numaralarını belirle
- Durdurma kriterini belirle

Adım 2: Başlangıç popülasyonunu oluşturma

- Kısıtlar ve problemin boyutuna göre başlangıç popülasyonunu rastgele oluştur
- Uygunluk değerini popülasyonun her bir bireyi için hesapla
- Ağacı en iyi uygunluk değeri ile seç

Adım 3: Tohumlarla araştırma

- Bütün ağaçlar için,
 - Tohumların numarasını I ve K arasında rastgele seç
 - $1 < i < k$ için,
 - $1 < j < D$ için,
 - Eğer $\text{rand} < \text{ST}$ ise,
 - Eq.(1)'e göre ilgili boyutu güncelle
 - Değilse,
 - Eq.(2)'ye göre ilgili boyutu güncelle
 - Tohumu tüm tohumlar arasında en iyi uygunluk değeri ile seç
 - En iyi tohumu ebeveyn ağaçla karşılaştır
 - Eğer tohum ebeveyn ağaçlardan daha iyi konumda ise tohumun pozisyonu tarafından ebeveyn ağacın pozisyonunu güncelleştir

Adım 4: Popülasyonda en iyi çözümü belirleme

- En iyi uygunluk değeriyle ağacı seç

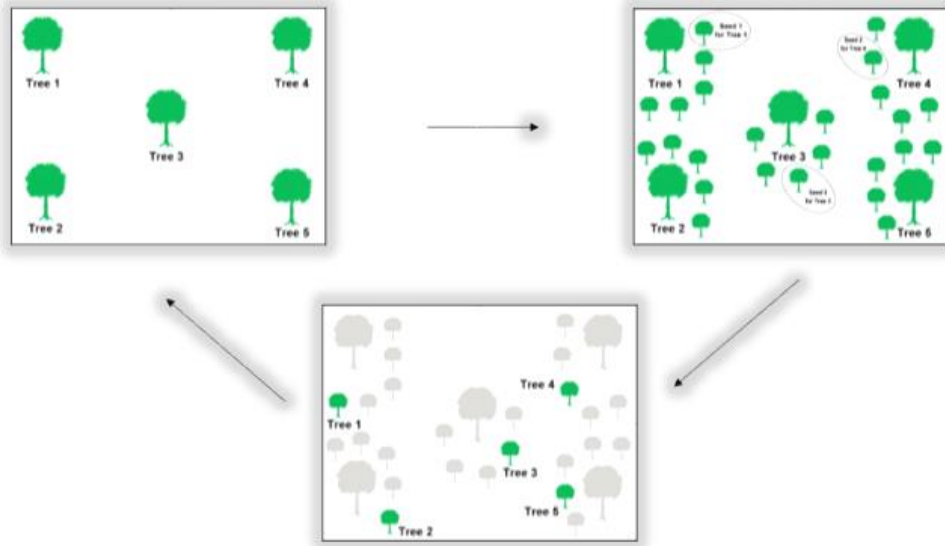
Adım 5: Durdurma şartını kontrol et

- Eğer durdurma şartı sağlanmazsa adım 3
Eğer durdurma şartı sağlanırsa en iyi çözümü raporla

Resim 2.1. TSA Çalışma Prensipleri (Aslan ve ark., 2018)

Algoritma başlangıcında popülasyon sayısı, n_s ve ST parametre değerlerinin belirlenmesi gerekmektedir. Burada ST parametresi arama eğilimini, n_s tohum sayısını ifade etmektedir. Bu iki parametre algoritma içerisinde analiz edildiğinde genellikle ST parametresi için 0.1 uygun kabul edilirken, n_s parametresi %10 ila %25 arasında olabilmektedir. (Kıran, 2015). Bu iki parametre algoritmanın problemdeki başarısına doğrudan etki etmesinden kaynaklı probleme bağlı olarak farklılık göstermektedir. ST 'nin daha yüksek değeri, güçlü bir yerel arama ve hızlı yakınsama sağlar; ST 'nin daha düşük değeri, yakınsamada yavaşlığa neden olurken güçlü küresel arama sağlar (Kıran,

2015). Popülasyon miktarının 5 olarak kabul edildiği Resim 2.2’de akış şeması gösterilen algorithmada öncelikle arama uzayında ağaçların konumları rastgele olarak üretilir. Üretilen her ağaç için probleme bağlı amaç fonksiyon değerleri hesaplanır ve başlangıç popülasyonunda bulunan en iyi ağaç seçilir.



Resim 2.2. TSA akış şeması (Kıran, 2015)

Algorithmada bulunan parametrelerden tohum sayısı (n_s) parametresini 5 olarak kabul edelim. Arama uzayında bulunan 5 ağaçtan her biri için 5 tohum üretmek için öncelikle “if (rand < ST) “ şartı gereği rastgele gelen değer ST parametresinden küçükse formül 2.1’de bulunan denklem, büyükse formül 2.2’de bulunan denkleme göre tohum üretimi gerçekleşir. Denklemlerde i, j değerleri problem boyutunu ifade ederken $T_{i,j}$ tohumu üretilen ağacı, $T_{r,j}$ popülasyonda bulunan rastgele seçilmiş ağacı, B_j ise popülasyonda bulunan en iyi ağacı ifade etmektedir. Yani ST parametresine göre popülasyona katılacak olan tohumların ebeveyn seçimi yapılmaktadır.

$$S_{i,j} = T_{i,j} + a_{i,j} \times (B_j - T_{r,j}) \quad (2.1)$$

$$S_{i,j} = T_{i,j} + a_{i,j} \times (T_{i,j} - T_{r,j}) \quad (2.2)$$

Her bir ağaç için ST parametresine bağlı olarak 5 tohum üretimi sonrasında popülasyon görünümü Resim 2.2’ de bulunan ikinci görselde gösterilmiştir. Bu aşamada amaç fonksiyonuna göre tohumların uygunluk değerleri hesaplanır. Her bir ağaç ve tohumları kendi gruplarını oluşturacak şekilde beş grup halinde ağaçları ve tohumları

uygunluk değerlerine göre kıyaslama yapılır. Bu kıyaslama yeni nesle aktarılacak olan popülasyon elemanını belirlemektedir. Resim 2.2’de verilen üçüncü görsele göre 1 numaralı ağacın dahil olduğu grup için 1 numaralı tohumun daha kaliteli olduğu gözlemlenmektedir. Ağaçları kendi tohumları ile gruplandırılmış şekilde incelediğimizde her grup kendi içinde en kaliteli çözümü belirler ve sonraki nesle bu çözümü aktarır. Tüm gruplar için yapılan kıyaslama sonrası gruplarda belirlenen kaliteli çözümler bir sonraki iterasyon için ızgara üzerindeki konumlarını korurken, diğer ağaç ve tohumlar ızgara üzerinden kaldırılır.

Kıyaslama sonrasında seçilen tohumlar artık yeni popülasyonun ağaçlarını oluştururlar. Bu işlem sonrasında yeni popülasyonda bulunan ağaçlar arasında en iyi ağaç seçimi yapılır. Algoritmanın durma kriteri kontrol edilerek yeni iterasyona geçilir veya sonuç çözüm algoritma sonucu olarak sunulur.

Kıran (2016) tarafından yapılan çalışmada TSA kullanılarak bir kısıtlama optimizasyon problemi, iyi bilinen basınçlı kap tasarımı PVD problemine çözüm aranmıştır. Ceza işlevi kullanılarak sorunun sınırlarının üstesinden gelinmenin amaçlandığı çalışmada, sorun tek bir amaç optimizasyon sorunu olarak kabul edilmiştir. TSA tarafından elde edilen deneysel sonuçlar, yapay arı kolonisi (ABC) ve parçacık sürüsü optimizasyonu (PSO) gibi yöntemlerin sonuçları ile kıyaslanmıştır. Çözüm kalitesi ve sağlamlığı baz alınarak incelenen sonuçlarda başarılı sonuçlar elde edilmiştir (Kıran, 2016).

Kıran (2016) tarafından yapılan çalışmada TSA’nın geliştirilerek ikili optimizasyon problemlerine çözüm bulması amaçlanmıştır. Çalışmada kullanılan dönüştürme mekanizması ile sürekli değerler ikili değerlere dönüştürülür. Çalışmada uygulanan yaklaşımın performansı, ulaşılamayan tesis yerleşimi problemleri (UFLP) kullanılarak test edilmiş ve ümit verici sonuçlar elde edilmiştir.

2016 yılında Kıran ve Çınar tarafından yapılan çalışmada TSA, GPU kullanılarak CUDA içerisinde uygulanmıştır. TSA’nın işlem süresini ölçmek için gerçekleştirilen çalışmada arama işlemlerinde umut verici sonuçlar elde edilmiştir. Paralel çalışma prensibi üzerine yapılan çalışmada paralel TSA ’nın, TSA ’nın seri versiyonuna göre yaklaşık 150 kat hızlandırıldığı gözlemlenmiştir (Çınar ve Kıran, 2016).

2016 yılında Kıran yaptığı çalışmada algoritmanın yerel ve evrensel optimuma takılmasını engellemek için TSA’yı geliştirmiştir. Arama uzayında sezgisel olarak dağılım gösteren algoritmalar optimizasyon problemlerinde her zaman doğru sonuca ulaşmayı garanti etmezler. Tamamen sezgisel dağılım gerçekleştirirken oluşturulan

nesiller bir önceki nesle bağlı olarak arama uzayında konumlanırlar. Fakat bu çözümler bazen local minimum veya maksimum alanlarda çoğalıp kilitlenmektedirler. Bu problemin önüne geçebilmek adına TSA algoritmasının güncel sürümüne solma işlemi (WP) adı verilen yeni bir kontrol parametresi eklenerek TSAWP olarak adlandırılan yeni sürümü M. Kıran tarafından önerilmiştir (Kıran, 2017).

Babalık ve arkadaşları (2017) tarafından yapılan çalışmada kısıtlı optimizasyon problemini çözmek amaçlı TSA geliştirilmiştir. TSA 'nın temel versiyonu ilk önce kısıtlanmamış optimizasyon problemlerini çözmek için kullanılmıştır (Babalık ve ark., 2018). Çalışmada Deb'in kuralları bir sonraki nesli seçme aşamasında kullanılmıştır. CTSA olarak geliştirilen algoritma 13 kısıtlı optimizasyon fonksiyonları altında analiz edilmiştir. Analiz sonuçları partikül sürüsü optimizasyonu (PSO), yapay arı kolonisi algoritması (ABC), genetik algoritma (GA) ve diferansiyel gelişim (DE) algoritmasının sonuçlarıyla kıyaslanmış, başarılı sonuçlar elde edilmiştir (Babalık ve ark., 2017).

Çınar ve Kıran (2017) tarafından yapılan çalışmada TSA'da Absorbing, Reflecting, Damping, Randomly olmak üzere dört sınır koşulu uygulanmıştır. Çalışma TSA'daki sınırlama yöntemleri arasında anlamlı bir bağlantının var olup olmadığını araştırmak amacıyla yapılmıştır. Çalışma sonucunda TSA için sınır şartları yöntemleri arasında anlamlı bir fark bulunmadığı görülmüştür (Çınar ve Kıran, 2017).

Korkmaz ve arkadaşları (2017) TSA'yı ayrık arama alanında çalışacak şekilde geliştirmişlerdir. Gezgin Satıcı Problemi (TSP) üzerinde tasarlanan TSA Berlin52 TSP örnek performans fonksiyonu üzerinde iki grup halinde test edilmiştir. Deneysel testlerde noktadan noktaya operatörler rasgele takas (RS), rasgele yerleştirme (RI) ve alt sıra operatörleri, alt dizilerin rastgele değişmesi (RSS), alt dizinin rastgele yerleştirilmesi (RIS), alt dizinin rastgele ters çevrilmesi (RRS), alt dizilerin rastgele ters takas (RRS) RRSS), alt dizinin (RRIS) tesadüfi olarak tersine çevrilmesi yöntemleri kullanılarak gerçekleştirilmiş, örnek TSP için başarılı sonuçların elde edildiği gözlenmiştir (Korkmaz ve ark., 2017).

Çınar ve Kıran (2018) tarafından yapılan çalışmada TSA ayrık optimizasyon alanından ikili optimizasyon problemlerini çözmek için tasarlanmıştır. Çalışmada TSA mantık kapıları (LogicTSA) ve benzerlik ölçüm teknikleri (SimTSA) kullanılmıştır. Geliştirilen hibrid TSA'nın yapılan çalışmada ikili optimizasyon problemlerinde çözüm kalitesi açısından diğer algoritmalarından daha iyi sonuç verdiği gözlenmiştir (Kıran ve Çınar, 2018).

Çınar, İşcan ve Kıran'ın (2018) gerçekleştirdiği çalışmada büyük boyutlu ikili optimizasyon problemlerini çözmek için TSA'daki xor mantık operatörünü kullanmışlardır. Algoritmanın probleme bağlı olarak geliştirilen versiyonlarının diğer optimizasyon algoritmalarına karşı daha iyi sonuç verdiği gözlenmektedir. Örneğin TSA'nın XORTSA varyantının, sürekli fonksiyonların çözümünde daha iyi performans gösterdiği 2018 de yapılan çalışmada sonuçlarıyla birlikte sunulmuştur (Çınar ve ark., 2018).

Çınar ve Kıran (2018) tarafından yapılan çalışmada TSA, CUDA platformunda paralel olarak uygulanmış, algoritmanın performansı problem boyutunun 10 olarak alındığı analizler ile test edilmiştir. Çalışma sonuçlarına göre belirli problemlere göre TSA'nın paralel versiyonu seri versiyonuna göre 184.65 kata kadar arttığı gözlenmiştir (Çınar ve Kıran 2018).

Kaya ve arkadaşlarının (2018) yaptığı çalışmada yıldız kümelerinin hareketlerini modelleyen Galaktik Sürü Optimizasyonu (GSO) kullanılarak TSA'dan GSO_TSA modeli oluşturulmuştur. Bu modelin performansı kıyaslama fonksiyonları ile test edilmiştir (Kaya ve ark., 2018).

Haklı ve arkadaşları (2018) tarafından gerçekleştirdikleri çalışmada daha önce 2,3,4 ve 5 boyutlu fonksiyonlarda performansı test edilen TSA'nın problem boyutu arttığında performansının düştüğü tespit edilmiş, algoritma boyutu yüksek problemlerde de kullanılabilir şekilde geliştirilmiştir. Temelde iki modifikasyon içeren geliştirmede ardışık tohum üretme ve en kötü tohumun soldurulma işlemleri yer almaktadır. Yapılan geliştirme ile TSASSGW adı ile TSA'nın yeni bir varyantı sunulmuştur (Haklı ve ark., 2018).

2018 yılında Aslan ve arkadaşları, TSA'da tohum üretme denklemlerini formül 2.3 ve 2.4'de gösterildiği şekilde güncellemiştir (Aslan ve ark., 2018). Formül 2.5'de bulunan denklemde D parametresi ile problemin boyutu ifade edilmektedir. Farklı parametrelerle yapılan deneysel çalışmaların sonunda, C parametresinin değeri 2'nin veya 1'in üzerine çıktığında algoritmanın başarısının azaldığı fark edilmiştir (Aslan ve ark., 2018).

$$\Delta_{i,j} = (BestTree_j - Trees_{r,j}) * (rand - 0.5) * C \quad (2.3)$$

$$\Delta_{i,j} = (ParentTee_j - Trees_{r,j}) * (rand - 0.5) * C \quad (2.4)$$

$$C = 2 - (D^2 * 0.0001) \quad (2.5)$$

Çınar ve arkadaşları (2018) gerçekleştirdiği çalışmada 0-1 sırt çantası problemini çözmek için TSA'nın hibrit ikili versiyonunu geliştirmişlerdir. 12 adet 0-1 sırt çantası problemi hibrit algoritma ile çözülmüş ve parametrelerin etkinliği analiz edilmiştir. Bu parametreler arama eğilimidir - ST, stand büyüklüğü ve sonlandırma koşulu. ST (0.1-0.9), stand büyüklüğü (10-100) ve sonlandırma koşulu maksimum fonksiyon değerlendirme sayısıdır. Yapılan 50 adet test sonucunda algoritmanın kontrol parametreleri soruna bağlı olarak hassasiyet göstermektedir.

Güngör ve arkadaşlarının (2019) yaptığı çalışmada yüksek boyutlu problemlerin TSA ile çözümünde güncelleme kuralları için farklı entegrasyonlar önerilmiştir. Çalışmada önerilen yöntem performansı ölçülürken Congress on Evolutionary Computation (CEC) ve temel 30 boyutlu iki amaçlı kıyaslama fonksiyonu kullanılmıştır. Deneysel sonuçlara göre geliştirilen TSA, temel TSA'dan çözüm kalitesi, yakınsama ve sağlamlık kriterleri bakımından daha başarılı olduğu gözlenmiştir (Güngör ve ark., 2019). Ayrıca TSA'nın yerel arama kabiliyeti hakkında çalışmalar yapılmıştır (Jiang ve ark., 2019).

TSA literatürü incelendiğinde, 2015'te önerilen algoritma günümüze kadar birçok problem üzerinde test edilmiş, farklı optimizasyon problemlerine göre tasarlanmıştır. Çalışmalarda yapılan deneysel test sonuçlarında ortaya çıkan başarısı ile dikkat çeken algoritma, seri ve paralel çalışma durumlarına göre performans testleri yapılmış, hem kısıtlı hem de kısıtsız optimizasyona göre uyarlanmış, ayrık ve sürekli fonksiyonlarda uygulanabilirliği araştırılmış, problem boyutuna karşı verdiği tepkiler incelenmiştir. Tüm bu varyantların yanında tek amaçlı optimizasyon için başarı gösteren algoritmanın çok amaçlı optimizasyona göre uyarlanabilirliğinin literatürde incelenmediği dikkat çekmektedir. Bu durum ışığında bu tez çalışması TSA'nın çok amaçlı optimizasyon için bir varyantı olarak MOTSA diye adlandırılan çok amaçlı ağaç tohum algoritmasını sunmaktadır. TSA'da bulunan ağaç tohum ilişkisini çok amaçlı problemler üzerinde modelleyen çalışma, algoritmanın tohum oluşturma yönteminde kullanılan denklem ve parametreler ile arama uzayında sağladığı başarıyı çok amaçlı optimizasyona taşımıştır. Bu çalışma sayesinde algoritmanın gerçek hayatta bulunan birçok problemin çözümünde kullanılabilirliği sağlanmıştır. Çalışmada çok amaçlı optimizasyon versiyonunda gerçekleştirilen algoritma performans testlerinden elde edilen veriler sunulmuş, alınan deneysel sonuçlar değerlendirilmiştir.

2.2. Genetik Algoritma (GA)

Doğa sentezlemesini örnek alan genetik algoritmalar biyolojik yaşamın bir modellemesi olan sezgisel evrim algoritmasıdır. Sezgisel algoritmalar optimum sonuca garanti veremezler ancak optimal bölgelere en azından yaklaşık çözümleri bulmak için etkili çözümlerdir (Kremmel ve ark., 2010). Evrimsel algoritmalarda, rastgele karar değişkenleri bir simülasyon modeli için girdi verisi olarak üretilir (Fallah-Mehdipour ve ark., 2012). Sezgisel yaklaşım kullanılırken hedeflerde açık bir iyileşme görülmektedir (Chan ve ark., 2016). Evrimsel algoritmaların genel yapısı incelendiğinde genetik algoritmaların mühendislik açısından tasarımını oluşturduğu gözlenmektedir (Steven, 2002).

Genetik algoritma temelinde kromozomlardan bireylerin meydana gelme süreci yatmaktadır. Bu kapsamda algoritma içerisinde kromozomlar bireyleri, bireyler de popülasyonu oluşturmaktadır. Popülasyon içerisinde güçlü bireyler seçilip yeni nesle aktarılır. Bir süreç akışı olarak incelediğimizde seçim, çaprazlama ve mutasyon evreleri ile ebeveynlerden meydana gelen bireylerden oluşan popülasyon yoğunluğu belli parametreler ile kontrol altına alınır. Sağladığı avantajlardan dolayı optimizasyon algoritmalarında sıklıkla kullanılan genetik algoritmalar son yıllarda artan talep üzerine oldukça fazla kullanım alanına sahiptir.

Darwin'nin evrim teorisinden yola çıkılarak geliştirilen algoritma arama uzayında bir noktadan değil, bir topluluk kullanılarak birden fazla alanda aynı anda arama işlemi gerçekleştirilmesiyle diğer algoritmalara oranla hızlı ve etkili bir algoritmadır. Bu özelliği ile yerel bir arama işlemi terkedilerek evrensel arama işlemi başlatılır. Optimizasyon problemlerinde karşılaşılan yerel minimum yada maksimum noktalarına takılmaktan kurtaran bu özellik genetik algoritmaların çok amaçlı optimizasyon problemlerine uygulanabilirliğini göstermiştir (Ergül, 2010).

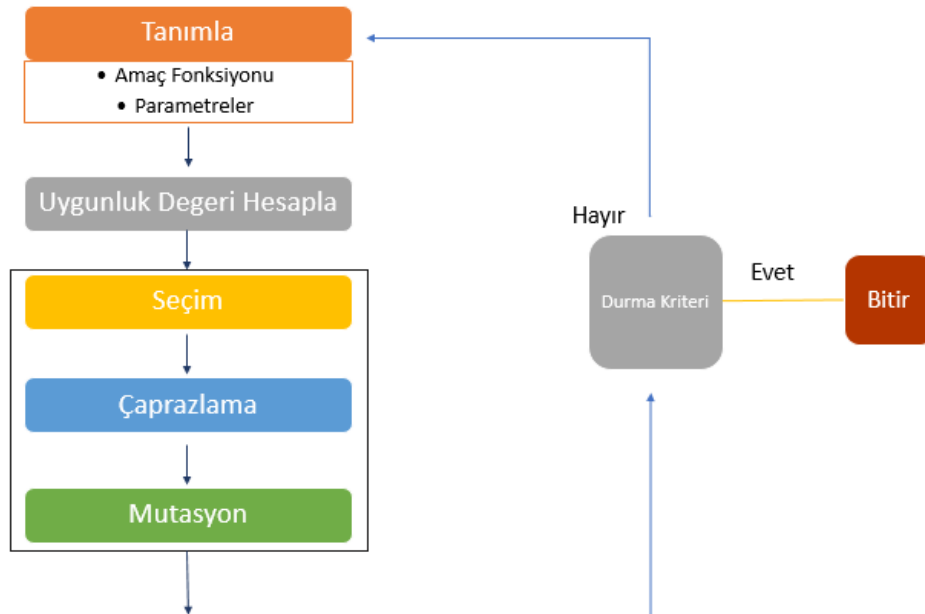
Genetik algoritmalar günümüzde birçok problem çözümünde kullanılmıştır. Bu problemlerden bazıları:

- Optimizasyon,
- Otomatik programlama ve bilgi sistemleri,
- Mekanik öğrenme,
- Finans,
- Pazarlama,
- Çizelgeleme problemleri,
- Montaj hattı dengeleme problemleri,

- Tesis yerleşim problemi,
- Sistem güvenliği problemi,
- Gezgin satıcı problemi (Emel, 2002).

Doğal seçim ilkesini kullanarak ebeveyn üreten genetik algoritmalar tüm arama uzayını değil seçimli oluşan alanlarda arama gerçekleştirdiği için diğer arama algoritmalarına göre çok daha hızlı sonuca erişebilmektedir. Seçimli arama işlemini gerçekleştiren genetik algoritma mantığı olasılık kurallarına dayanmaktadır. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar (Goldberg ve Holland, 1988). Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır (Emel, 2002).

Resim 2.3’da verilen genetik algoritmanın akış süreci incelendiğinde, ilk önce gerekli parametre ve amaç fonksiyonları tanımlanma aşaması gelmektedir. Rastsal yöntemle başlangıç popülasyonu üretilir. Başlangıç popülasyonunda bulunan her bir ebeveyn için uygunluk değerleri hesaplanır. Başlangıçta tanımlanan olasılık değerine göre yeni nesli oluşturacak ebeveynler seçilir. Seçilen iki ebeveyn çaprazlama ve mutasyon yöntemlerine tabii tutularak çocuk bireyleri oluşturur. Bu işlemler durma kriteri sağlanana kadar devam eder. Her iterasyonda yeni bir kuşak meydana gelen genetik algoritmalarda birkaç iterasyon sonrasında kaliteli bireylerden yeni kuşaklar oluşturulmaya başlanır. Bu özellik genetik algoritmanın arama uzayında iyileştirme yaparak sonuca erişmesini sağlamaktadır.



Resim 2.3. Genetik Algoritma'nın Akış Diyagramı

2.2.1. Bireylerin seçimi

Biyolojik yaşam teorisine dayanan algoritmada yeni birey oluşturmada çaprazlama ve mutasyon işlemleri için bireylerin seçilmesi gereklidir. Birey seçiminde bireyin uygunluk fonksiyon değeri o bireyin çözüm kalitesini ifade eder. Yapılan seçim işleminde kaliteli bireylerin seçimi oluşturulacak kuşakta bulunan bireylere ait çözüm kalitesini doğrusal olarak etkilemektedir.

Genetik algoritmalarda seçim yöntemi için rulet, turnuva ve sıralı seçim yöntemleri en bilinen seçim yöntemleridir.

- **Rulet Tekerleği Yöntemi:**

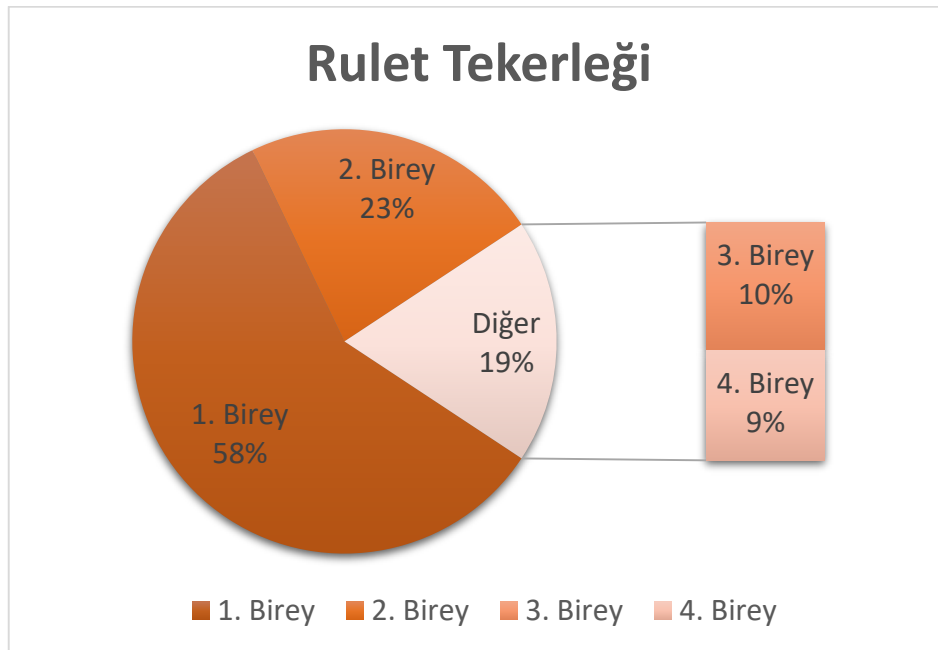
Popülasyonda bulunan bütün bireylerin uygunluk değeri toplamının her bireye ait uygunluk değerine oranı o bireyin seçilme olasılığını verir. Resim 2.4’da rulet tekerleğinde 4 bireyli seçim oranları gösterilmektedir.

- **Turnuva Seçimi Yöntemi:**

Popülasyon içinde rastgele sayıda seçilen bireyler arasında uygunluk değeri en yüksek bireyin seçilmesidir.

- **Elitist Seçim Yöntemi:**

Uygunluk değeri kalitesine göre numaralandırılan bireylerin en iyi bireyden başlayarak seçilmesidir.



Resim 2.4. Rulet Tekerleği

2.2.3. Çaprazlama işlemi

Seçilen ebeveyn bireylerden yeni bireyler elde edilmesi işlemine denir. Uygunluk değeri yüksek, dolayısıyla kaliteli iki bireyin kromozomlarını belirli yöntemle değiştirilerek yeni bireyler elde edilir. Çaprazlama işlemi tek noktadan ve çift noktadan olmak üzere iki farklı şekilde yapılabilir.

- Tek Noktadan Çaprazlama Yöntemi:

Kromozomlarda bir noktadan itibaren çaprazlama işlemi yapılır.

Ebeveynler: 11110|**00100110111** - **11011**|11000011110

Çocuk-1: **1101100100110111**

Çocuk-2: 1111011000011110

- Çok Noktadan Çaprazlama Yöntemi:

Kromozomlarda rastgele seçilen iki nokta arasındaki kısımda çaprazlama işlemi gerçekleştirilir.

Ebeveynler: 11000|11100|110110 - **11011**|11110|**011110**

Çocuk-1: 1100011110110110

Çocuk-2: **1101111100011110**

2.2.3. Mutasyon işlemi

Mutasyon, yeni kuşakta oluşturulacak bireylerde kromozomda meydana gelen farklılaştırma işlemidir. Amaç çaprazlama sonrasında oluşan bireyde yapılan değişiklik ile arama uzayında kısır döngüye girilmesini ve yerel minimum ya da maksimum noktaya takılmayı engellemektedir. Mutasyon işlemi pozisyona, sıraya ve kromozomdaki bite göre üç farklı şekilde gerçekleştirilebilir. Mutasyon işlemi popülasyonda çeşitlilik sağlayarak problem sonucunun yerel çözümlere takılmasını önler (Özsağlam ve Cunkaş, 2008).

- Pozisyona göre mutasyon:

Kromozomdaki rastgele seçilen genlerde yapılan değişimdir.

ABCDFE: EBCDFA

- Sıraya göre mutasyon:

Kromozomda rastgele seçilen iki nokta arasında ilk kısmın ikinci kısmın önüne getirilmesiyle yapılan değişimdir.

ABCDFE: EABCDF

- Kromozoma göre mutasyon:

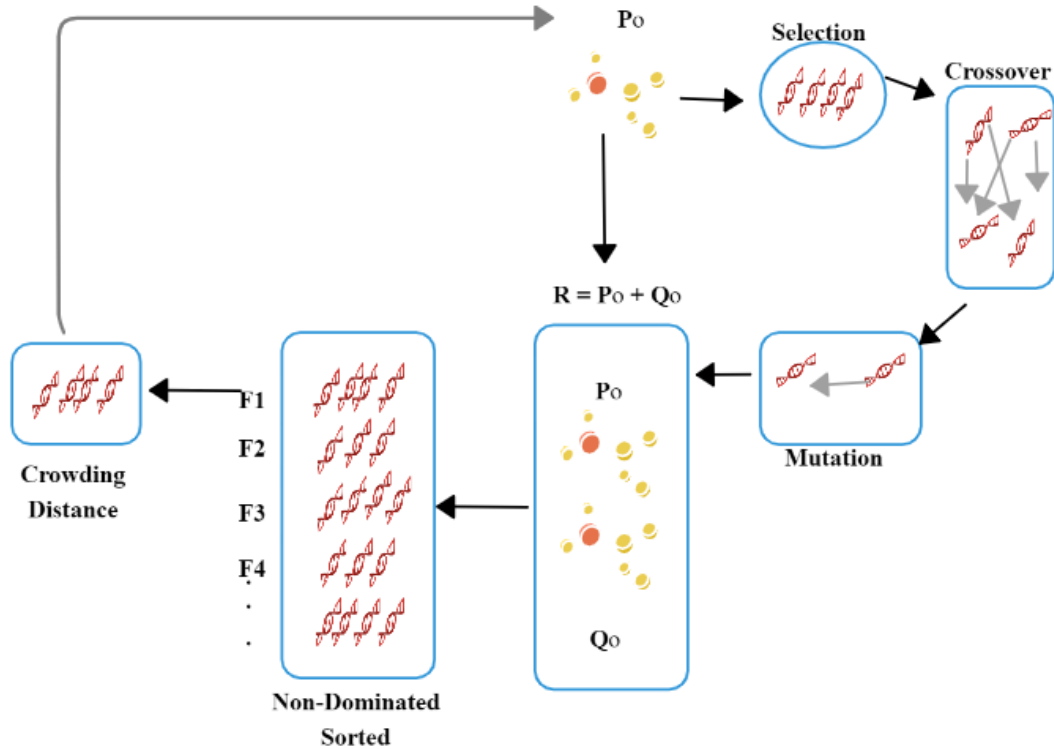
Kromozomda seçilen rastgele genin değiştirilmesidir.

1100011110110110 - 1100011111110110

2.3. Bastırılmamış Sınıflandırma Genetik Algoritma II -NSGAI

Pareto-optimal bölgeye yönelik geliştirilen NSGA algoritması Srinivas ve Deb tarafından önerilmiştir. (Srinivas ve Deb, 1994). Çok amaçlı evrimsel algoritmalarından biri olan NSGA-II (Non-dominated Sorting Genetic Algorithm), bir önceki nesli olan NSGA da karşılaşılan eksikliklerin giderilmesi sonucu ortaya çıkmıştır. (Deb ve ark., 2000). Pareto yüzeyinin araştırılmasında genetik algoritma özelliklerini kullanmaktadır. Genetik algoritmalarda bulunan çaprazlama ve mutasyona ek olarak yeni nesilde eleman farklılığını sağlamak için kalabalık mesafe (crowding distance) hesaplamasını ve oluşturulan geniş popülasyondan yeni neslin daha iyi çözümlerden olması için seçimli sıralama kısaca elitizm yöntemini kullanır.

Baskın olmayan sıralama (Non-dominated Sorting), bir popülasyondaki çözümlerin kalitesini belirlemek için sıklıkla evrimsel algoritmalarda kullanılan bir tekniktir (Gustavsson ve Syberfeldt, 2017). NSGA-II algoritması birçok etkin özelliklerinden dolayı birçok çok amaçlı optimizasyon çalışmasında kullanılmıştır (Ergül ve Eminoglu, 2014; Chan ve ark., 2015; xu ve ark., 2015).



Resim 2.5 NSGAI Algoritması

Algoritma Resim 2.5'te gösterilen akış şeklinde başlangıcında belirlenen popülasyon miktarına (N) göre rastgele olarak başlangıç popülasyonu oluşturulur. NSGA-II algoritması N miktardaki popülasyondan yine N miktarında popülasyon oluşturulması ve bu iki popülasyondan bastırılmamış çözümlerin seçilmesi işlemlerini içermektedir. Arama uzayında bulunan başlangıç popülasyonundaki her bir eleman için çözüm değerleri yani çözümlerin kaliteleri, uygunluk fonksiyonu hesaplanarak bulunmaktadır. Genetik algoritma tabanında bulunan sistemde Turnuva yöntemi ile gerçekleşen seçim işleminde rastgele belirlenen k kadar çözüm kalitelerine göre kıyaslanarak iki çözüm seçilir.

Seçilen çözümler çaprazlama ve başlangıçta belirlenen mutasyon oranına göre yeni iki çözüm oluştururlar. N miktarında yeni popülasyon elde edilene kadar çözüm üretme işlemi devam eder. Başlangıç popülasyonu ve yeni popülasyon birleştirilir. Oluşan kombine popülasyon, kısaca R popülasyonu $2N$ miktarındadır.

Genetik algoritma işlemleri sonrasında oluşan R kombine popülasyonunda bulunan çözümlerin Pareto yüzeyleri belirlenir. Elde edilen Pareto yüzeyler kombine popülasyonunda bulunan grupları oluşturur. Pareto yüzeylerinde bulunan çözüm sayıları eşit olmamakla birlikte her iterasyon sonucunda aynı sayıda Pareto yüzeyi çıkması da beklenemez. Pareto yüzeylerin sayısı sezgisel algoritmanın arama uzayındaki dağılımına göre farklılık göstermektedir.

NSGAIİ algoritması diğer algoritmalarından farklı olarak seçkinlik işleminde kalabalık mesafe metodunu kullanmaktadır. Popülasyonda herhangi bir noktanın yoğunluğunu tahmin etmek için; ele alınan noktanın her iki kenarında bulunan iki noktanın ortalama uzaklığı tüm amaçlar için hesaplanır. Bu miktar ($I_{distance}$), en yakın komşular olan köşeler kullanılarak şekillendirilen küboidin çevre uzunluğunun tahmini için kullanılır ve buna kalabalık mesafesi (crowding-distance) denir (Sağ, 2008).

NSGAIİ ana döngüsünde her bir t. nesli oluşturmak için gerçekleşen iterasyonda kullanılan denklemlerden formül 2.3'te bulunan denkleme göre R_t birleşik popülasyonu, P_t mevcut popülasyon ile Q_t çocuk popülasyonunun birleşmesinden oluşmaktadır. Hızlı baskın olmayan sıralama yapılan birleşik popülasyon her bir seviye için F ile ifade edilen rank seviyelerine ayrılmaktadır. Yeni popülasyona birey aktarılırken kullanılan elitizm yöntemi formül 2.6'daki denklemde gösterildiği gibi popülasyonun birey sayısına ulaşana kadar önce rank seviyelerinin aktarılması ile gerçekleşir. Daha sonra F_i rank seviyesinde bulunan birey sayısı yeni popülasyona tamamen yerleşemediği durumda F_i popülasyonunda kalabalık mesafeye göre sıralama

yapılır. Formül 2.10'da bulunan denkleme göre oluşturulan yeni popülasyon bir sonraki nesil için yeni iterasyona aktarılır.

t. nesil için;

$$R_t = P_t U Q_t \quad (2.3)$$

$$F = \text{hızlı baskın olmayan sıralama} (R_t) \quad (2.4)$$

$$P_{t+1} = \{\emptyset\} \text{ ve } i = 1 \quad (2.5)$$

$$|P_{t+1}| + |F_i| \leq N \text{ olana kadar yığılma uzaklığı ataması} (F_i) \quad (2.6)$$

$$P_{t+1} = P_{t+1} U F_i \quad (2.7)$$

$$i = i + 1 \quad (2.8)$$

$$\text{Sırala} (F_i, a_n) \quad (2.9)$$

$$P_{t+1} = P_{t+1} U F_i [1 \div (N - |P_{t+1}|)] \quad (2.10)$$

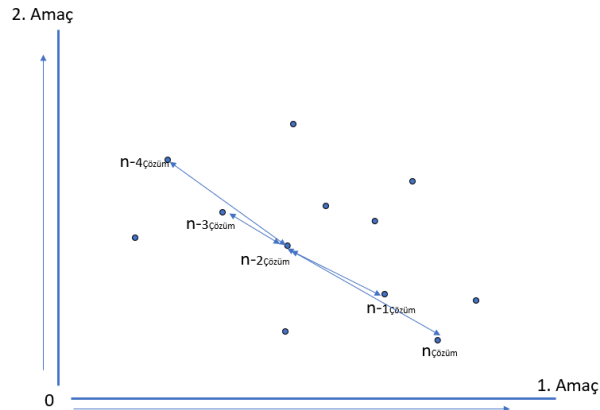
$$Q_{t+1} = \text{yeni yığın oluştur} (P_{t+1}) \quad (2.11)$$

$$t = t + 1 \quad (2.12)$$

2.3.1. Kalabalık mesafe (Crowding Distance)

NSGA-II'nin en büyük avantajı, bastırılmayan çözümler arasındaki çeşitliliğin, turnuva seçiminde ve elitizm aşamasında kullanılan kalabalık mesafeye göre sınıflandırma prosedürü kullanılarak ortaya konmasıdır (Roshanian ve ark., 2017).

Bir yüzeyde bulunan çözümlerin kalabalık mesafesi hesaplanırken öncelikle yüzeyin uç kısımlarında bulunan çözümler belirlenir. Bu çözümlerin I_{distance} değerine sonsuz atanır. Sonsuz atanması bu çözümlerin elitizm işleminde yeni nesil popülasyonuna atanma şansını artırmaktadır. Resim 2.6'de $n_{\text{çözüm}}$ olarak ifade edilen çözümlerin kalabalık mesafe hesaplaması gösterilmiştir.



Resim 2.6. Kalabalık mesafe hesaplaması

Kalabalık mesafe mantığında eğer bir yüzeyde bulunan çözümler yeni nesle aktarılacaksa öncelik olarak uç çözümlerin atanması sağlanmalıdır ki, gelecek nesillerde bu yüzeyde oluşacak çözümler yerel minimum ya da maksimum noktasına takılmamalı ve bir alan çerçevesinde üretilmemelidir. Böylelikle yüzeyde daha sağlıklı ve düzgün bir yayılım elde edilebilmektedir.

2.3.2. Bastırılmamış sıralama (Non-Dominated Sorted)

Bastırılmama durumu, en az bir diğer amaç için kötü sonuç vermeden hiçbir amaç için hiçbir çözümün geliştirilemediği bir eşdeğerlik olarak tanımlanabilir (Srinivas ve Deb, 2000). İki çözümün kıyaslanmasında eğer bir çözüm optimizasyon probleminde bulunan amaçlardan en az biri için diğer çözümden daha kötü değilse bu çözüm ikinci çözüm tarafından bastırılmaz.

Resim 2.7’te gösterilen hızlı bastırılmamış çözümleri sınıflandırma yönteminde, her bir çözüm için o çözümü bastıran çözümlerin kümesi ve bu çözümün bastırdığı çözüm sayısı hesaplanarak işlem gerçekleştirilir.

```

for each  $p \in P$ 
   $S_p = \emptyset$ 
   $n_p = 0$ 
  for each  $q \in P$ 
    if ( $p < q$ )
       $S_p = S_p \cup q$ 
    else if ( $q < p$ )
       $n_p = n_p + 1$ 
    if  $n_p = 0$ 
       $P_{rank} = 1$ 
       $F_1 = F_1 \cup p$ 
   $i = 1$ 
  while  $F_i \neq \emptyset$ 
     $Q = \emptyset$ 
    for each  $q \in F_i$ 
      for each  $q \in S_p$ 
         $n_p = n_p - 1$ 
        if  $n_p = 0$ 
           $q_{rank} = i + 1$ 
           $Q = Q \cup q$ 
       $i = i + 1$ 
     $F_i = Q$ 

```

Eğer p, q yi bastırırsa
q yu p tarafından bastırılan çözümler kümesine ekle

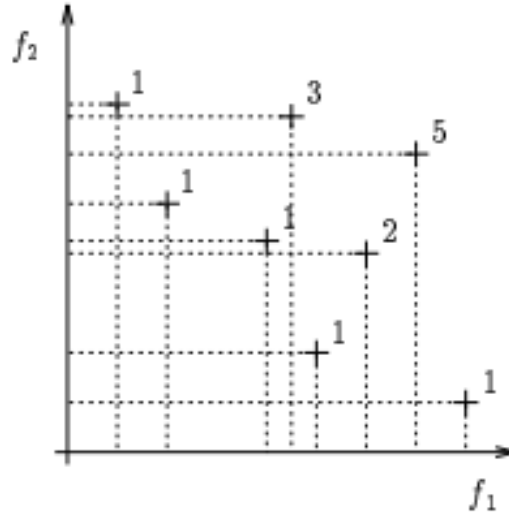
yüzey sayacı başlangıç durumuna getir

Sonraki yüzeyin elemanlarını depolamak için kullanılır

q sonraki yüzeye mi ait ?

Resim 2.7. Hızlı bastırılmayan sıralama prensibi (Deb ve ark., 2002; Sağ, 2008)

Sıralama işleminde oluşturulan rank seviyeleri ilk olarak 0 (sıfır) dan başlamaktadır ve işlem döngüsü her bir rank seviyesi için tekrar edilmektedir. Şekil 2.8’te iki amaçlı optimizasyon için çözümler buldukları rank seviyesine göre numaralandırılmıştır.

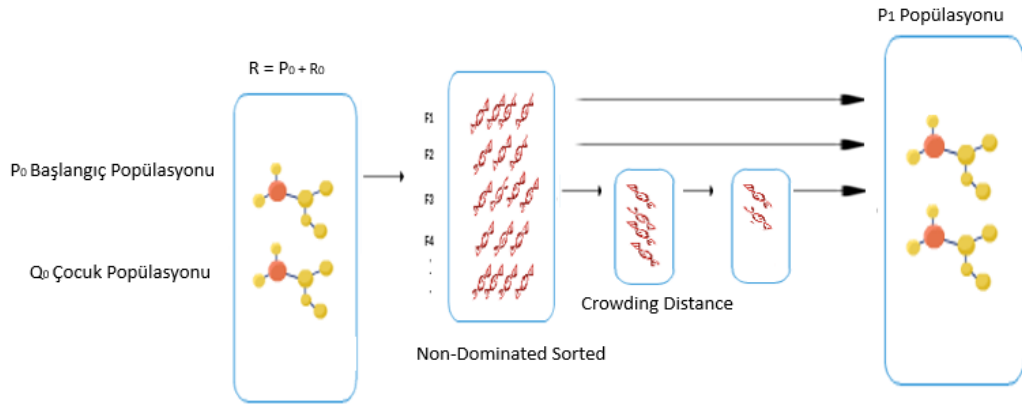


Resim 2.8. Çok amaçlı optimizasyon rank seviyeleri (Fonseca ve Fleming, 1999)

Pareto yüzeylerine ayrılmış ve kalabalık mesafeleri hesaplanmış kombine popülasyonu içerisinde önce rank seviyelerine (Pareto yüzeylerine) göre sıralanır. Daha sonra her bir rank seviyesindeki çözümler kendi içlerinde $I_{distance}$ değerine göre sıralama yapılır. Böylelikle en kaliteli çözümler ve yine bu çözümler arasında yeni nesil için en avantajlı çözümler ön sıralara getirilmiş olur.

2.3.3. Elitizm işlemi

Başlangıçta popülasyon miktarı N olduğu gibi her bir iterasyon sonucunda elde edilen popülasyon miktarı da N 'dir. $2N$ boyutlu popülasyonda elitizm işlemi uygulanırken kalite değerlerine göre sıralanmış, ranklarına ayrılmış çözümler bastırılmamışlık derecesine göre yeni popülasyona aktarılır. Şekil 2.9'te elitizm işlem aşamaları gösterilmektedir.



Resim 2.9. Elitizm İşlemi

İterasyon sonucunda oluşan rank ve ranklardaki çözüm sayısı belirsiz olduğu için uygulanan bu işlemde k çözüm sayısına sahip bir rank seviyesi taşınırken $k < NP_1$ (ilgili yüzeyde bulunan çözüm sayısı eklendiğinde yeni nesil popülasyonu N boyutunu aşar mı?) kontrolü yapılmaktadır. Ranktaki tüm çözümler aktarılamadığı durumda çözümler kalabalık mesafe değeri temel alınarak yeni popülasyona aktarılır. NSGAIİ algoritması elitizm işlemi ile aynı rank seviyesinde bulunan çözümler arasında seçim yapmış, birbirini bastırmayan çözümleri kalabalık mesafesine göre, yeni nesilde daha iyi bir yayılım sağlamak amacıyla sıralamıştır. NSGA, yalnızca özdeş Pareto derecesine sahip vektörleri değerlendiren çözümler arasındaki netliği paylaşır (Van Veldhuizen ve Lamont, 2000).

3. MATERYAL VE YÖNTEM

3.1. Materyal Listesi

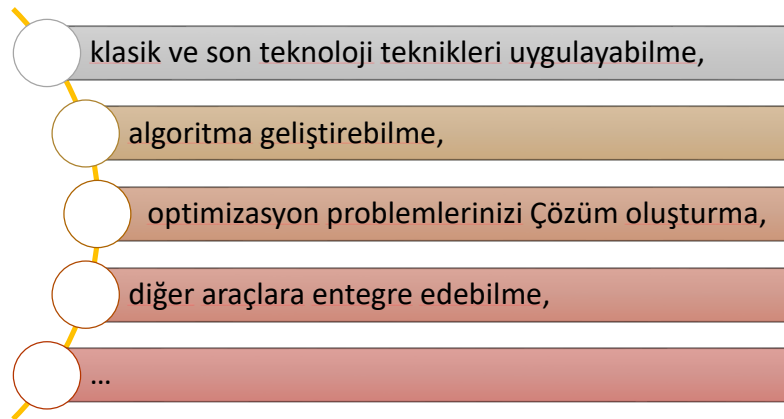
Bu tez çalışmasında java JMETAL platformu üzerinde çalışan bir bilgisayar yazılımı kullanılmıştır. Tablo 3.1’de bu tez çalışması için kullanılan bilgisayara ait teknik özellikler belirtilmiştir.

Tablo 3.1. Tez çalışması için kullanılan bilgisayarın teknik özellikleri

Teknik Özellikler	TOSHIBA SATELLITE L750
İşlemci	Intel Core i7 2670QM - 2.20 GHz
Ram	4 GB DDR4
Sistem	64 Bit Operating System x-64
Harddisk Kapasitesi	594 GB
İşletim Sistemi	Windows 10 Education
Ekran Kartı	NVIDIA GeForce GTX 525M
Çözünürlük	1366x768

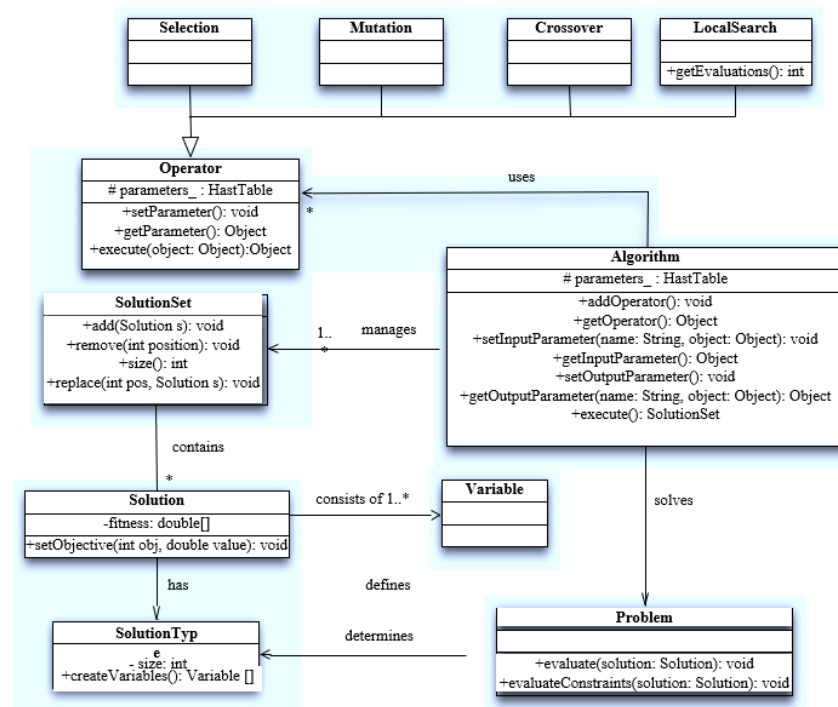
Bu çalışmada çok amaçlı optimizasyon algoritmaları üzerine artan çalışmalardan doğan algoritmaların birbiri ile kıyaslanmasını ve belli bir sistemde kodlanmasını sağlayan java frameworkü kullanılmıştır. jMetal, algoritmaların performansını değerlendirmek için bir dizi klasik ve modern teknolojiye sahip en iyi duruma getirici, geniş bir kıyaslama sorun kümesi ve bir dizi iyi bilinen kalite göstergesini içerir (Durillo ve Nebro, 2011). Kullanımı son yıllarda artan bu framework üzerinde önerilen algoritma kodlanmıştır. Framework geniş test sistemi ve grafik özellikleri sağlamaktadır. Çok çeşitli uygulamalarda kullanılan esnek, genişletilebilir ve kullanımı kolay bir yazılım paketidir (Durillo ve ark., 2010).

Nesne yönelimli mimariye dayanan framework kullanıcıya birçok avantaj sağlamaktadır. Resim 3.1’ de bu avantajlar gösterilmektedir.



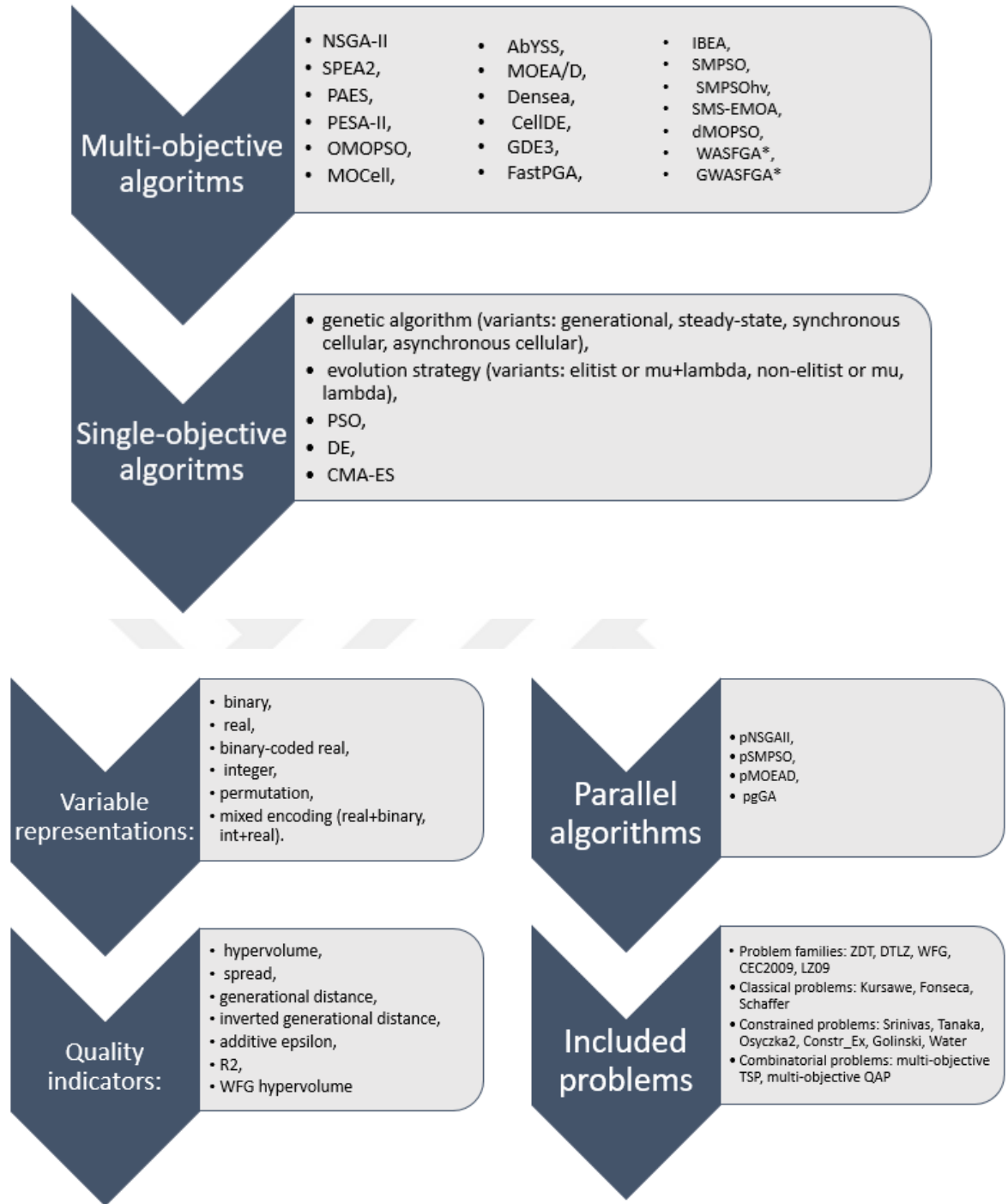
Resim 3.1. JMetal avantajları

Java Meta-heuristic Algorithm olarak geliştirilen platform sezgisel algoritma yaklaşımında bulunan özellikleri fonksiyonel şekilde sunmaktadır. Resim 3.2’de gösterilen temel mimariye bakıldığında problem, operatör, çözüm ve fonksiyonlar bulunur.



Resim 3.2. Jmetal mimarisi (Durillo ve ark., 2010)

Geliştiricileri tarafından eklenen çok amaçlı algoritmalar sayesinde diğer algoritmalara hızla ulaşabilir ve yeni uygulamalarla kıyaslama gerçekleştirilebilmektedir. Resim 3.3’de Jmetal özellikleri gösterilmektedir.

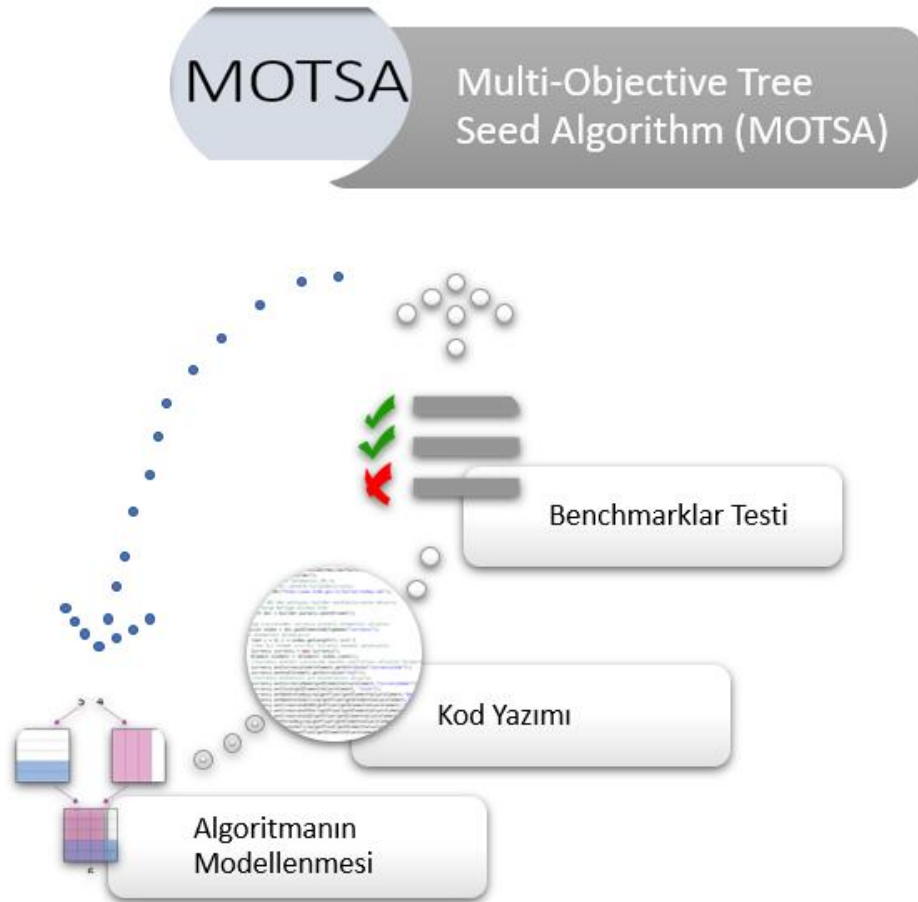


Resim 3.3. JMetal platformunun sağladığı yöntemler

3.2. Yöntem

Bu tez çalışmasında çok amaçlı optimizasyon problemlerinin çözümünde kullanılması için MOTSA sezgisel algoritma yaklaşımı önerilmiştir. Algoritma literatürde optimizasyon problemleri için kullanılan iki algoritmanın iyi bilinen yanlarının sentezlenmesi ile oluşturulmuştur. TSA algoritmasının çok amaçlı optimizasyon problemlerine çözüm aramada bir varyantı olarak geliştirilen Çok Amaçlı Ağaç Tohum Algoritması (Multi- Objective Tree Seed Algorithm), kısaca MOTSA olarak adlandırılan algoritma, tohum üretimi ve bastırılmamış sınıflandırma olarak iki temel adımdan oluşmaktadır.

MOTSA algoritmasında TSA algoritmasının birden fazla amaç için NSGAIİ algoritmasına benzer bir çözüm yolu izlemesi planlanmıştır. Algoritmada TSA algoritmasının doğası gereği ağaçlardan tohum üretme işlevinden ziyade tohumlar arasında ve tohumla ağaç arasında gerçekleştirilen kıyaslama işleminde NSGAIİ'nin elitizm yöntemi kullanılmıştır. Resim 3.4'te bu çalışmanın akış şeması gösterilmiştir.

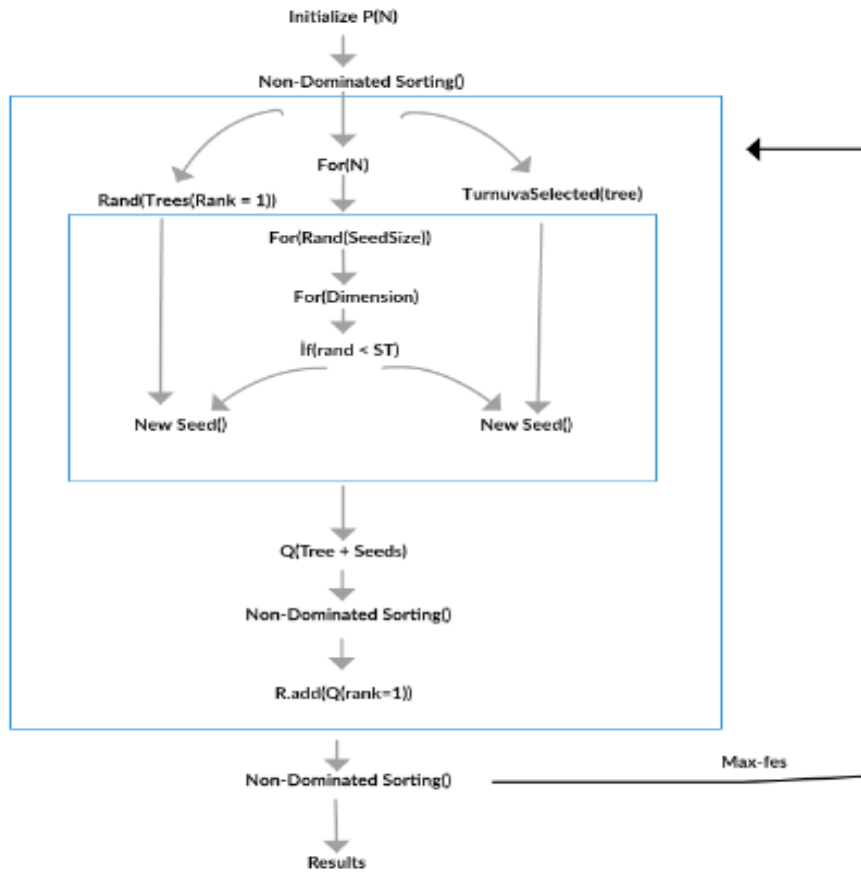


Resim 3.4. Çalışmanın Akış Şeması

Gerçekleştirilen tez çalışması öncelikli olarak TSA ve NSGAIİ algoritmalarının en iyi özelliklerini kullanarak algoritmanın temel adımları belirlenmiştir. Algoritma modelinin oluşturulma süresince MATLAB ortamında gerçekleştirilen çalışmada farklı parametre değerlerine göre test edilerek, parametrelerin kararlılık değerleri belirlenmiştir. Algoritma modeli belirli uygunluk fonksiyonlarında doğru pareto yüzey elde edebilecek şekilde geliştirilmiştir. Modellemesi gerçekleştirilen algoritma java platformunda kodlanarak kalite analiz testleri gerçekleştirilmiştir.

3.3. MOTSA Akış Diyagramı

Resim 3.5'te gösterilen akış diyagramında başlangıç popülasyonundan başlayan algoritma bastırılmamış sıralama, turnuva seçimi, tohum üretimi, kalabalık mesafe gibi işlemleri birbirini takip eden for döngüleri ile if koşulları çerçevesinde gerçekleştirilmektedir.



Resim 3.5. MOTSA Akış Diyagramı

Problemin boyutuna bağılı olarak gerçekleşen döngülerde TSA' da bulunan ST kontrol parametresi ile tohum üretme mekanizması gerçekleşmektedir. Ağaçların ve tohumların bulunduğu Q popülasyonu NSGAIİ elitizm yöntemiyle yeni nesli oluşturur. Oluşturulan yeni nesil bir sonraki iterasyon için başlangıç popülasyonunu ifade eder. Max-fes olarak ifade edilen durdurma parametresi sağlandığı takdirde algoritma sonlandırılır ve çözüm sunulur.

3.4. MOTSA Algoritma Modelinin Oluşturulması

Algoritma modeli oluşturma süreci MATLAB ortamında gerçekleşmiştir. Bu süreçte iki algoritmada bulunan özellikler belirli bir sıra ile benchmark testine tutularak algoritmaya eklenmiştir. Gelişim evresinin başlarında TSA algoritmasının tohum üretme yönteminin yanında NSGAIİ de bulunan non-dominated sıralama eklenmiştir. İlerleyen süreçlerinde ise uygulamaya genetik algoritmada bulunan mutasyon özelliği eklenerek son şeklini almıştır. Modelleme süreci maddeler halinde gösterilecek olursa:

- i. Başlangıç popülasyonunda bulunan ağaçların arama uzayına yerleştirilmesi ve çözüm uzayındaki konumlarının birden fazla amaca göre belirlenmesi.
- ii. Çözüm ızgarasında bulunan ağaçlar arasında en kaliteli ağacın seçiminde bastırılmamış sınıflandırma yönteminin kullanılması. Bu durumda en kaliteli tek ağaç yerine kaliteli ağaçların bulunduğu küme elde edilmektedir.
- iii. Tohum üretimi mekanizmasında TSA' da bulunan ST parametresi olasılığına bağılı kullanılan en iyi ağacın seçimi turnuva seçiminin rank seviyesi bir olan ağaçlar arasında kalabalık mesafe sıralamasına uygun olarak yapılmasıdır.
- iv. Oluşan tohum ve ağaçlardan NSGAIİ' de bulunan kombine popülasyonunun oluşturulması. Burada kombine popülasyonun oluşmasında çeşitli versiyonlar test edilmiştir.
 - a. Ağaç ve tohumları arasında sıralama işlemi uygulamak ve bu işlem sonrasında kombine popülasyonu oluşturmak,
 - b. Tüm ağaç ve tohumları tek bir kombine popülasyonda birleştirmek,

- c. Yeni nesil popülasyon miktarını sabitlememek (Oluşacak olan nesillerin sayıları sezgisel oranda artıp azalması),
- d. Her bir ağaç ve tohum grubundan kombine popülasyona belli sayıda çözüm çekmek,
- e. Sadece ait olduğu grupta bulunan ağacı domine edebilen tohumları kombine popülasyona eklemek,
- v. Kombine popülasyonunda NSGAIİ' de bulunan sıralama ve elitizm işlemlerini gerçekleştirip yeni nesli oluşturmak,
- vi. Algoritmayı benchmarklar üzerinde uygulamak ve benchmarkların gerçek front yüzeylerini elde etmek,
- vii. Max-fes (durma kriteri), ST parametresi ve tohum sayısının farklı değerlerine göre algoritmanın test edilmesi ve en kararlı değerlerin belirlenmesi,
- viii. Tohum üretme işlemine çaprazlama ve mutasyon özelliklerinin kombinasyonel şekilde uygulanması,
- ix. Mutasyon oranının belirlenmesi,
- x. Tohum üretim sınırlarına kontrol işleminin uygulanması,

Modelleme sürecinde gelişen MOTSA uygulaması sonucunda parametreleri değerleri Tablo 3.2'deki gibi belirlenmiştir.

Tablo 3.2. Modelleme sonucu belirlenen parametre değerleri

TEKNİK ÖZELLİKLER	PARAMETRELER
Max-fes	25000
ST	0.1
MT	0.3
minSeedNumber	0.01*N
maxSeedNumber	0.03*N
N	100
Turnuva yöntemi	Binary

Gerçekleştirilen modelleme sürecinde özellikle mutasyon oranı ve ST parametre değeri üzerinde durulmuştur. Bunun yanı sıra en iyi ağacı belirlerken kullanılan turnuva yöntemi için model süreçlerinde rulet tekerleği de denenmiştir. Tohum üretiminde bir ağaçtan kaç tohum üretilmesi gerektiği noktasında MOTSA' nın gelişim sürecindeki ilk versiyonlarında tohum sayısı sabit tutulmuş daha sonra olasılık değerine göre olması gerektiği kararlaştırılmıştır.

3.4. MOTSA Modelleme Süreci Benchmark Testleri

Algoritma modelleme aşamasında amaç fonksiyonu olarak kullanılan benchmarklardan alınan yüzey sonuçları gerçek yüzeylerle kıyaslanmıştır. Belirlenen geliştirme MATLAB ortamında kodlaması yapılarak çok amaçlı optimizasyon algoritması olarak geliştirilen MOTSA için farklı revizyonları hazırlanmıştır. En uygun geliştirme araştırılmış, test edilerek bir sonraki aşamaya alınmıştır. Bu süreçte 16 farklı revizyonda MOTSA algoritması tasarlanmıştır.

3.4.1. Modelleme test fonksiyonları

Çok amaçlı genetik algoritmaların geliştirilmesi aşamasında doğru pareto yüzeylerinin bulunması performans kriterini belirlemektedir (Sağ ve Cunkaş, 2009). Evrimsel algoritmalar teknolojisinde kullanılan çok amaçlı test problemleri güçlü ve doğru bir sonuç çıkarmak için dikkatli bir şekilde analiz edilmelidir (Huband ve ark., 2006). Nitekim çok amaçlı optimizasyon algoritmalarının gelişim sürecinde bu algoritmaların ölçeklenebilmeleri için üretilen test fonksiyonları da önem kazanmış ve hızla geliştirilmiştir (Deb, 2001).

Çok amaçlı optimizasyon algoritma geliştirmeleri genellikle öncelikle iki amaca göre modelleme ile başlamakta, algoritmanın gelişimine göre 2 ve daha fazla amaç fonksiyonuna uyarlanmaktadır. Bu noktada geliştirilen algoritmaların fonksiyonel yapıda kurulması ile dinamik amaç fonksiyonu kullanımı sağlanmaktadır.

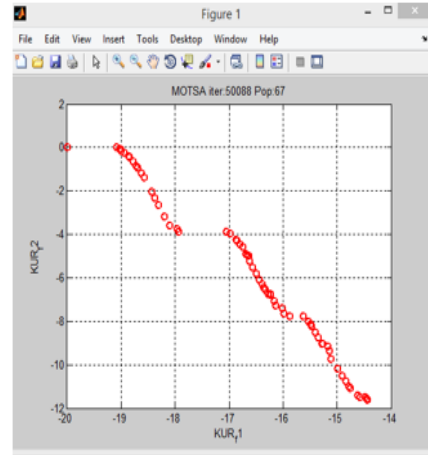
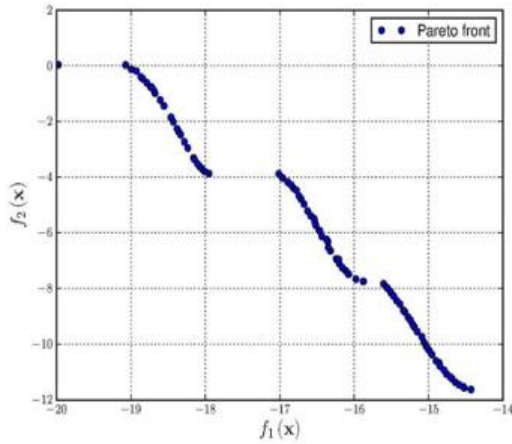
Algoritmanın gerçek Pareto yüzeye yakın sonuç üretebilmesi, o algoritmanın problem için başarı oranının yüksek olduğunu göstermektedir. Gerçekleştirilen testlerde kullanılan benchmarklardan bazıları Kursawe, Poloni ve Fonseca fonksiyonlarıdır.

3.4.1.1. Kursawe fonksiyonu

Kursawe benchmarkı çoklu hedef minimize problemini tanımlar. Bu fonksiyon n boyutlu bir girdiyi kabul eder ve iki boyutlu bir çıktı üretir. Formül 3.1’de Kursawe uygunluk fonksiyonunun hesaplama fonksiyonu, formül 3.2’de fonksiyonun değer aralıkları verilmiştir. Resim 3.6’da fonksiyonun gerçek pareto yüzeyi ve modelleme işleminde MOTSA’dan aldığımız pareto yüzeyi gösterilmiştir. Resimde gösterilen plot çizimler incelendiğinde MOTSA’nın Kursawe uygunluk fonksiyonunda gerçek pareto yüzeyi yakalayabildiği gözlenmektedir.

$$\text{Minimize} = \begin{cases} f_1(x) = \sum_{i=1}^2 \left[-10 \exp(-0.2 \sqrt{x_i^2 + x_i^2}) \right] \\ f_2(x) = \sum_{i=1}^3 [|x_i|^{0.8} + 5 \sin(x_i^3)] \sqrt{n}^2 \end{cases} \quad (3.1)$$

$$s. t = \begin{cases} -5 \leq x_i \leq 5 \\ 1 \leq i \leq 3 \end{cases} \quad (3.2)$$



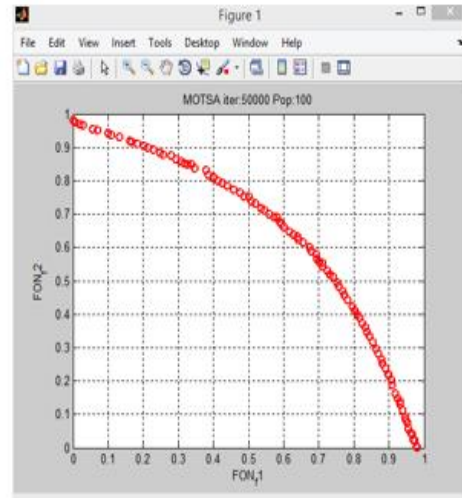
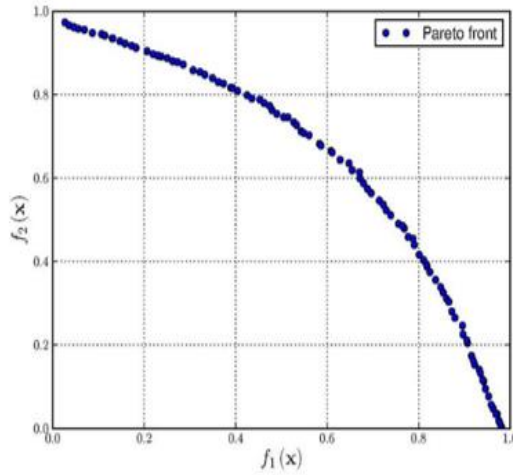
Resim 3.6. Kursawe fonksiyonu gerçek yüzey ve MOTSA sonucu

3.4.1.2. Fonseca fonksiyonu

Fonseca benchmarkı çoklu hedef maksimize problemini tanımlar. Fonseca ve Fleming tarafından önerilen iki amaçlı fonksiyondur (Fonseca ve Fleming, 1998). Formül 3.3’de Fonseca uygunluk fonksiyonunun hesaplama fonksiyonu, formül 3.4’te fonksiyonun değer aralıkları verilmiştir. Resim 3.7’de fonksiyonun gerçek pareto yüzeyi ve modelleme işleminde MOTSA’dan aldığımız pareto yüzeyi gösterilmiştir. Resimde gösterilen plot çizimler incelendiğinde MOTSA’nın Fonseca uygunluk fonksiyonunda gerçek pareto yüzeyi yakalayabildiği gözlenmektedir.

$$\text{Minimize} = \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \end{cases} \quad (3.3)$$

$$\text{s. t.} \begin{cases} -4 \leq x_i \leq 4 \\ 1 \leq i \leq n \end{cases} \quad (3.4)$$



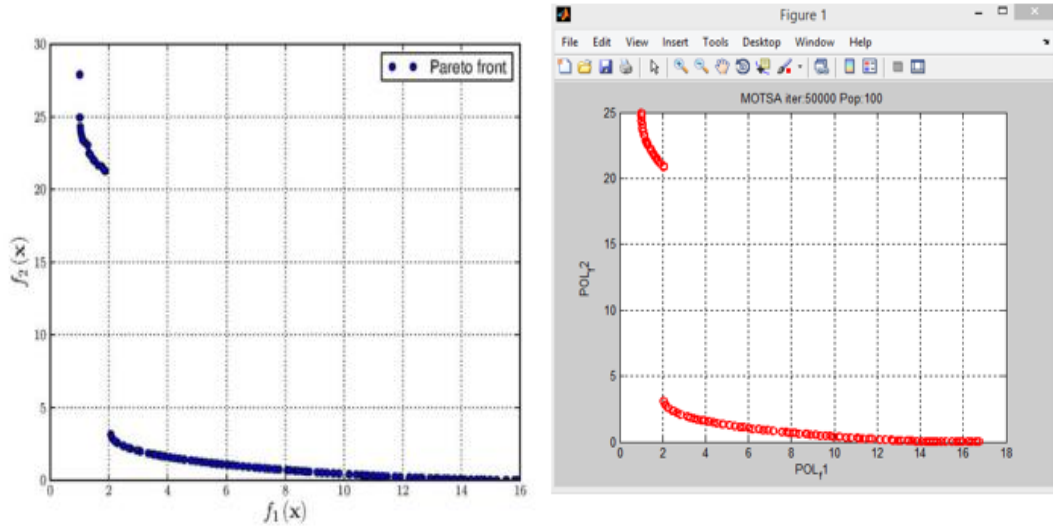
Resim 3.7. Fonseca fonksiyonu gerçek yüzey ve MOTSA sonucu

3.4.1.3. Poloni fonksiyonu

Poloni benchmarkı birçok hedef için minimize problemini tanımlar. 2 amaçlı süreksiz fonksiyondur. Formül 3.5’de Poloni uygunluk fonksiyonunun hesaplama fonksiyonu, formül 3.6’da fonksiyonun değer aralıkları verilmiştir. Resim 3.8’de fonksiyonun gerçek pareto yüzeyi ve modelleme işleminde MOTSA’dan aldığımız pareto yüzeyi gösterilmiştir. Resimde gösterilen plot çizimler incelendiğinde MOTSA’nın Poloni uygunluk fonksiyonunda gerçek pareto yüzeyi yaklaştığı ancak tam yakalayamadığı gözlenmektedir.

$$\text{Minimize} = \begin{cases} f_1(x, y) = \left[1 + (A_1 - B_1(x, y))^2 + (A_2 - B_2(x, y))^2 \right] \\ f_2(x, y) = (x + 3)^2 + (y + 1)^2 \end{cases} \quad (3.5)$$

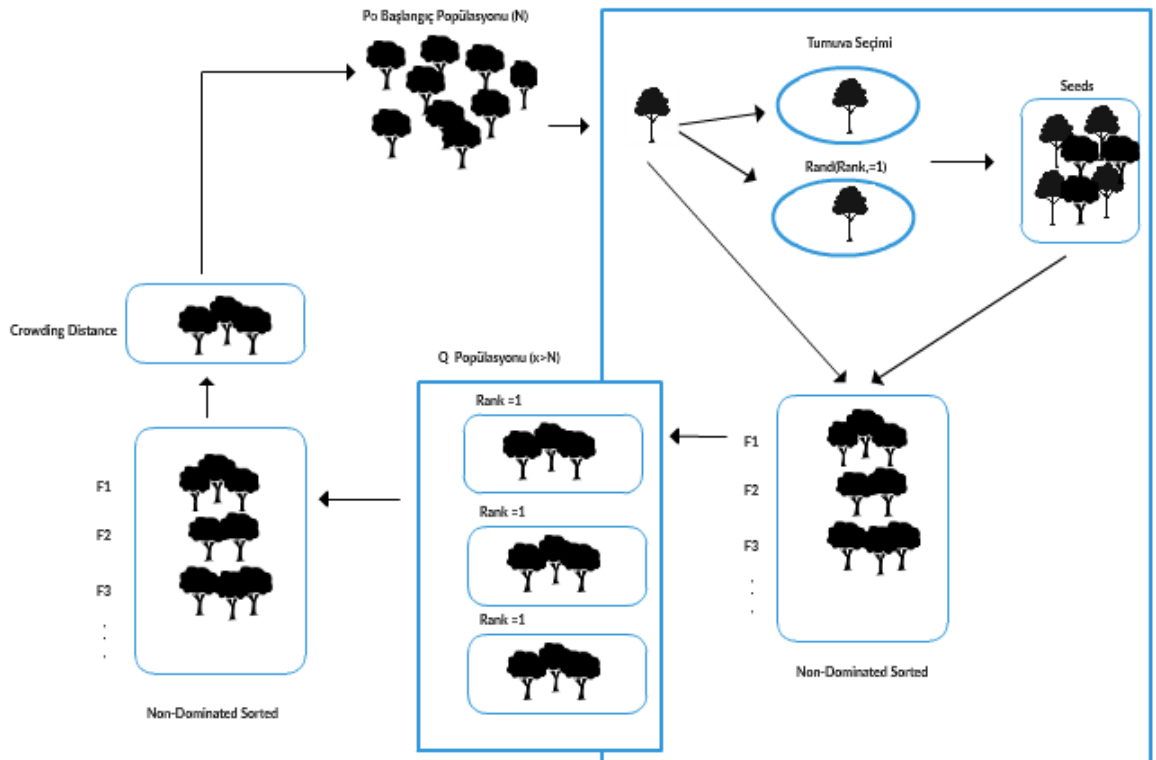
$$s. t = \begin{cases} A_1 = 0.5 \sin(1) - 2 \cos(1) + \sin(2) - 1.5 \cos(2) \\ A_2 = 1.5 \sin(1) - \cos(1) + 2 \sin(2) - 0.5 \cos(2) \\ B_1(x, y) = 0.5 \sin(x) - 2 \cos(x) + \sin(y) - 1.5 \cos(y) \\ B_2(x, y) = 1.5 \sin(x) - \cos(x) + 2 \sin(y) - 0.5 \cos(y) \\ -\pi \leq x, y \leq \pi \end{cases} \quad (3.6)$$



Resim 3.8. Poloni fonksiyonu gerçek yüzey ve MOTSA sonucu

3.5. MOTSA: A Multi-Objective Tree Seed Algorithm

Bu çalışmada, tek amaçlı bir optimizasyon algoritması olan bir ağaç tohumu algoritması (TSA) geliştirilmiş ve yeni birçok amaçlı optimizasyon algoritması olan çok amaçlı bir ağaç tohumu algoritması (MOTSA) önerilmiştir. MOTSA, 2 hedefli kıyaslama problemine uygulanmıştır ve bulunduğu çözümler incelenmiştir. MOTSA, TSA algoritmasını yeni nesil oluşturmada bir örnek olarak alırken, en iyi bireylerin belirlenmesinde, yeni ebeveynlerin ve gelecek nesle aktarılacak bireylerin seçiminde Pareto tabanlı seçkinlik uygulanarak seçilir. Uygulamalı seçkinlik sürecinde, NSGA-II algoritmasında baskın olmayan sıralama ve kalabalık mesafeli sıralama yöntemleri uygulanmaktadır. Baskın olmayan sıralama, pareto yüzeyini belirlemek için kullanılır. Kalabalık mesafesi ile algoritmada difüzyon sağlanarak, popülasyonda bulunan en göze çarpan çözümler belirlenerek güçlendirilir. Resim 3.9’da MOTSA’nın optimal çözümlü arama yöntemi gösterilmiştir.



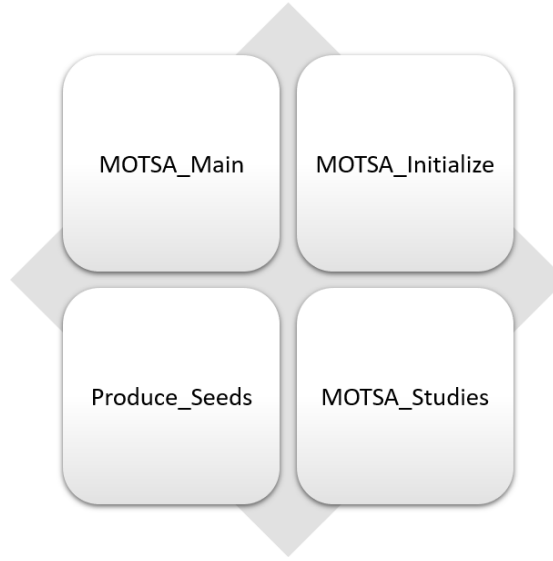
Resim 3.9. MOTSA metodu

MOTSA algoritmasında, her ağaç ve tohum soruna bir çözüm sunar. TSA algoritmasına göre kullanılan ST parametresi ile mevcut popülasyondaki bireylerden yeni bireyler oluşturulur. Çözüm alanındaki tüm bireyler, Pareto'ya dayanan bastırma

seviyelerine göre sıralanır ve yeni kuşağa aktarılacak nüfus belirlenir. MOTSA, tek amaçlı problemler üzerindeki performansını motive ederek çok amaçlı optimizasyon problemlerini çözmek için önerilmiştir. Çok amaçlı problemlerde seçim sorununun üstesinden gelmek için, NSGA-II 'nin iyi bilinen stratejileri, baskın olmayan sıralama ve kalabalık mesafesi, önerilen MOTSA ile entegre edilmiştir. Bu yöntemle, birleşik ağaç ve tohum popülasyonlarından en yüksek kalitede çözümler seçilir ve bir sonraki nesle aktarılır.

Öncelikle başlangıç popülasyon boyutu belirlenir ve çözüm uzayında ağaçlar rastgele oluşturulur. Oluşturulan ağaçların uygunluk değerleri hesaplanır. Başlangıç popülasyonunda bulunan elemanların bastırılmışlık seviyeleri yani rank değerleri belirlenir. Popülasyonda bulunan her bir ağacın tohum sayısı rastgele belirlenir. Her bir tohum ST parametresine göre Paretoya dayalı turnuva seçimiyle gelen bir tree veya popülasyon içerisinde rank değeri 1'e eşit olanlar arasından rastgele seçilen bir ebeveyn kullanılarak oluşturulur. Oluşturulan tohumlar mutasyon işlemine tabi tutulur ve birleşik popülasyonda elitizm işlemi uygulanır. Bu işleme göre tohumlar ve ağaç bastırılmışlık sıralamasına gönderilerek rank değerleri belirlenir. Rank değerleri belirlenen ağaç ve tohumlara crowding distance sıralaması yapılarak en seçkin çözümler belirlenir. Tek amaçlı optimizasyonda olduğu gibi tek bir sonuç elde edilmeyeceği için yapılan sıralama işleminde en iyi rank seviyesine sahip olan bireyler çocuk popülasyonuna eklenir. Bu tohum üretme ve elitizm işlemi her ağaç için tamamlandıktan sonra oluşan çocuk popülasyonunda tekrar bir elitizm işlemi yapılarak yeni popülasyon oluşturulur. Bu işlemler maksimum iterasyon süresince devam ederek en iyi çözümler kümesi aranmaktadır.

JMetal platformuna algoritma eklemesi 4 temel aşamadan oluşmaktadır. Platformda algoritma frameworkün sağladığı fonksiyonel altyapıya uygun olarak kodlanmıştır. NSGA-II algoritmasının da içerisinde entegre bulunan platforma öncelikle algoritmanın başlangıç değerlerinin atanacağı Initialize kısmı eklenmiş, bu kısımda operatör tanımlamaları gerçekleştirilmiştir. Tohum üretme yapısının operatör sisteme entegre şekilde kodlanması ile jMetal yapısıyla örtüşme sağlanmıştır. MOTSA'nın tüm işlevlerini gerçekleştireceği ve probleme sonuç üreteceği sınıf yapısı kurulduktan sonra jMetal'de bulunan experiments yapısına eklenen MOTSA_Studies kodlaması ile MOTSA'nın sistemde bulunan diğer algoritmalarla kıyaslanması sağlanmıştır. Bu sayede istenilen fonksiyon ve başarı ölçütüne göre algoritmalar karşılaştırılabilmektedir. Resim 3.10'da Jmetal platformuna eklenen MOTSA sınıfları gösterilmiştir.



Resim 3.10. MOTSA JMetal parçaları

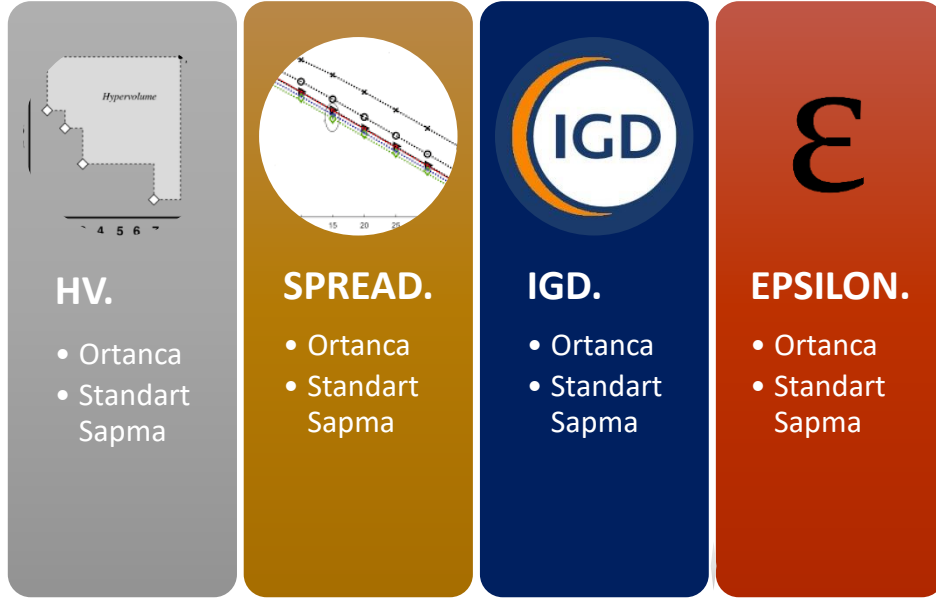
5.6. Sonuçların Elde Edilme Aşaması

Bu tez çalışmasında sonuçların elde edilmesi 3 aşamadan oluşmaktadır. İlk aşamada algoritmaların başarılarının karşılaştırılması için hangi ölçütlerin kullanılacağı seçilmiştir. İkinci aşamada MOTSA ile kıyaslanacak 16 algoritma belirlenmiş, üçüncü aşamada ise kıyaslamada kullanılacak amaç fonksiyonları için literatürde bulunan Benchmarklar seçilmiştir.

Java ortamından alınan sonuçlar 4 başarı ölçütü ile 16 farklı algoritma ile elde edilmiştir. Tüm bu sonuçlar jMetal ortamında MOTSA için özel geliştirilen experiment uygulamasından elde edilmiştir.

3.6.1. Başarı ölçütleri

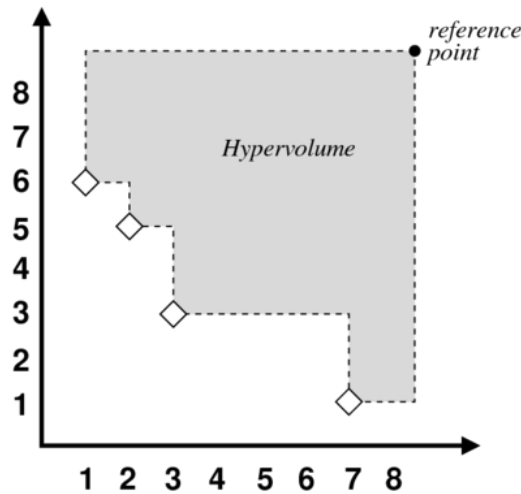
Çok amaçlı optimizasyon algoritmalarının karşılaştırmasında kullanılan birçok kriter bulunmaktadır. Bu ölçütler yayılım, dağılım, performans gibi kaynaklardan bir ya da birkaçına verdiği ağırlığa göre farklılık göstermektedir. Bu çalışmada başarı ölçütü olarak Hypervolume, Spread, IGD ve Epsilon ölçütlerinin her biri için ortanca ve standart sapma olarak hesaplanmıştır. Resim 5.11’de test sonuçlarında kullanılan başarı ölçütleri gösterilmiştir.



Resim 3.11. Çalışmada kullanılan başarı ölçütleri

3.6.1.1. Hypervolume

Hypervolume, çözüm kümesine yakınsama ve dağılım bakımından kalite seviyesini belirleyen başarı ölçütlerinden biridir. Kriterin sonuç değerinin 1'e yakınlığı çözümün kaliteli olduğunu göstermektedir. Gerçek yüzeyin HV değeri ile algoritmanın HV değerinin oranını ifade eder. Hypervolume göstergesi (veya S-metrik), uygun özelliklerinden dolayı özellikle önemlidir (Beume ve ark., 2009). Resim 3.12'de hypervolume hesaplama işlemi gösterilmiştir. Çözüm kümesi üzerinde gerçekleştirilen hesaplama, referans noktası ile çözümlerin oluşturduğu alan kullanılmaktadır.



Resim 3.12. Hypervolume

Bu kriter matematiksel olarak çözüm uzayında elde edilen çözüm kümesi yani rank seviyesinde bulunan her bir çözüm ile çözüm kümesinde bulunan en kötü nokta arasında kalan alanı ifade etmektedir. Bu metrik çözüm uzayının baskın olan kısmını ölçer (Fonseca ve ark., 2006). Hypervolume göstergesinin, çok amaçlı evrimsel algoritmalarındaki çözümleri kabul etmek için bir rehberlik kriteri olarak önerilmiş olan bazı makul özelliklere sahip olduğu düşünülmektedir (Fonseca ve ark., 2006).

3.6.1.2. Spread

Spread kriteri dağılım ve çeşitlilik bakımından algoritmaları karşılaştırmaktadır. Hypervolume kriterinin tersine bu kriterin 0'a yakın olması kalite düzeyinin daha iyi olduğu anlamına gelmektedir. Temel olarak, bu metrik, her bir hedefin genişliğinin toplamını kullanarak yayılma değerini verir (Riquelme-Granada ve ark., 2015). Spread kriteri hesaplama denklemi formül 3.7'te gösterilmektedir.

$$\frac{\sum_{m=1}^m dm^e + \sum_{i=1}^m |di - \bar{d}|}{\sum_{m=1}^m d_m^2 + |a|d^-} \quad (3.7)$$

3.6.1.3. Invented Generational Distance (IGD)

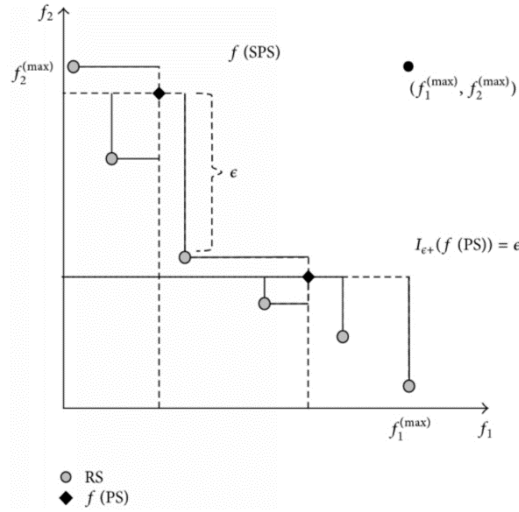
IGD algoritmanın dağılım değerini ölçmek için kullanılan başarı kriteridir. Çözüm uzayında bulunan noktalardan referans noktasına bağlı olarak öklid mesafesi ile hesaplanmaktadır. Referans noktalarının seçimi kriterin doğruluğuna etki etmektedir. Bu kriter 0'a yaklaştıkça kaliteli çözüm elde edildiğini ifade eder. Formül 3.7'te IGD ölçütünün hesaplanmasında kullanılan denklem verilmiştir (Ishibuchi ve ark., 1997).

$$IGD_Z(A) = \frac{1}{|Z|} \sum_{j=1}^{|Z|} \min\{d(z_j, a_k) \mid a_k \in A\} \quad (3.8)$$

3.6.1.4. Epsilon

Algoritmanın yakınsama değerini ölçen bu kriter 0'a yaklaştıkça kaliteli çözüm elde edildiğini ifade eder. Epsilon kriterinde, her bir çözüm çifti arasındaki fark anlamlı

olduğu için her bir amaç fonksiyonun istenen hassasiyet seviyesi de farklıdır (Asadzadeh, 2016). Resim 5.13'te epsilon başarı kriterleri gösterilmiştir.



Resim 3.13. Epsilon

3.6.2. Karşılaştırılan algoritmalar

MOTSA algoritması java platformunda NetBeans derleyicisi kullanılarak jMetal platformuna göre kodlanmıştır. jMetal platformunda bulunan diğer algoritmalarla kıyaslama yapılabilmesi için yapılan experiments kodu ile AbYSS, CellDE, DENSEA, dMOPSO, GDE3, GWASFGA, IBEA, MOCeLL, MOCHC, MOEAD, NSGAI, OMOPSO, PAES, PESA2, SMPSO, SPEA2, WASFGA algoritmaları ile karşılaştırılmıştır. Tüm karşılaştırmalar java ortamında eşit parametreler ile yapılmıştır.

3.6.2.1. AbYSS (Multiobjective Scatter Search)

Tek amaçlı optimizasyonun çok amaçlı optimizasyona uyarlanması olan algoritma Pareto optimal mantığı çerçevesinde şekillendirilmiştir. Yoğunluk tahmini ve bastırılmış sınıflandırma özelliklerini içinde barındırmaktadır. Dağınık arama yapısını izleyen, ancak evrimsel algoritmalarından mutasyon ve çaprazlama işleçlerini kullanan Arşiv Tabanlı Hibrit Dağılım Arama (AbYSS) adlı hibrit metaheuristik bir algoritmadır (Nebro ve ark., 2008).

3.6.2.2. CelIDE (Cellular Genetic Algorithm with Differential Evolution)

2008 yılından yeni bir hibrit hücresel genetik algoritma olarak sunulmuştur (Durillo ve ark., 2008). MOCeII algoritmasını temel olarak alan algoritma genetik algoritma yöntemlerinin yerine diferansiyel evrim çoğalma mantığını benimsemiştir.

3.6.2.3. DENSEA (Duplicate Elimination Nondominated Sorting Evolutionary Algorithm)

Bastırılmamış sınıflandırma algoritması mantığını içeren algoritma çakışan çözümlerin engellenmesi üzerine geliştirilmiştir. İterasyonda oluşturulan çözümlerden duplicate çözümlerin silinmesi ve yeniden yerleştirilmesi işlemlerini gerçekleştirir (Obayashi ve ark., 2007).

3.6.2.4. dMOPSO (Decomposition-Based Multi-Objective Particle Swarm Optimizer)

Parçacık sürü optimizasyon algoritma temelinde oluşturulmuştur. PSO algoritmasının bir varyantıdır. Algoritmada simüle edilmiş tavlama stratejisi, PSO'nun arama performansını iyileştirmek için yerel bir arama operatörü olarak algoritmaya dahil edilir (Zhang ve ark., 2018).

3.6.2.5. GDE3 (Generalized Differential Evolution)

GDE algoritmasının bir varyantı olan diferansiyel evrim algoritmasıdır. GDE3, isteğe bağlı bir dizi amaç ve kısıtlamayla küresel optimizasyon için farklı evrimin (DE) bir uzantısıdır (Kukkonen ve Lampinen, 2005). DE'nin çözüm arama sürecinde öncelikli olarak, gerçek sayılardan oluşan kromozom yapılarıyla temsil edilen bireyler, yani rastgele üretilmiş amaç vektör (değişken) değerleri, yeni donör (donor) bireyler elde etmek için mutasyon işlemi içerisine dâhil edilmekte ve bu yolla arama uzayının da genişlemesi sağlanmaktadır (Köse, 2017).

3.6.2.6. WASFGA (Weighting Achievement Scalarizing Function Genetic Algorithm)

Tercih bilgisinin evrimsel algoritmaya aktarıldığı çok amaçlı evrimsel optimizasyon algoritmasıdır. Karar mekanizmasından ifade edilen tercihleri mümkün olan optimal şekilde yerine getiren Pareto cepheyi oluşturmayı hedefler. 2014 yılında önerilen yaklaşım, bir başarı ölçeklendirme fonksiyonunun (ASF) kullanılmasına ve bireylerin çeşitli cephelere sınıflandırılmasına dayanmaktadır (Ruiz ve ark., 2014).

3.6.2.7. GWASFGA (Global Weighting Achievement Scalarizing Function Genetic Algorithm)

GWASFGA evrimsel algoritma tabanlı, topluluğa dayalı, çok amaçlı optimizasyon algoritmasıdır. Uygunluk fonksiyonu, iki referans noktasının dikkate alındığı ve kullanılan ağırlık vektörünün bir ağırlık vektörü kümesinden alındığı Tchebychev mesafesine dayanan bir başarı ölçeklendirme fonksiyonu (ASF) ile tanımlanır (Saborido Infantes ve ark., 2016). Pareto cepheye yaklaşma stratejisinde ağırlıklandırma yöntemi etrafında geliştirilmiştir.

3.6.2.8. MOCCell (Multiobjective Cellular Genetic Algorithm)

Hücresel çok amaçlı evrimsel algoritmadır. Genetik algoritmanın hücresel mantığıyla kurulan algortmada hücrelerin etkileşime girme işlemini sadece o hücrenin komşuları arasında gerçekleştirmektedir. Bulunulan arama uzayında yavaş dağılımın hedeflendiği algoritma gruplama ile arama gerçekleştirmektedir (Alba, 2006).

3.6.2.9. IBEA (Indicator-Based Evolutionary Algorithm)

Indicator tabanlı çok amaçlı evrimsel algoritmadır. Mevcut algoritmalarından farklı olarak, IBEA, kullanıcının isteğine göre uyarlanabilir ve kullanılacak amaç fonksiyonu paylaşımı gibi ek çeşitlilik koruma mekanizması gerektirmez (Zitzler ve Künzli, 2004).

3.6.2.10. MOCHC (Multiobjective CHC Algorithm)

Tek amaçlı CHC algoritmasının çok amaçlı varyantıdır. Yeni nesil seçiminde elitizm ve kalabalık mesafe yöntemlerini kullanır. Algoritmada önerilen yeni iterasyonu başlatma işleminde, rank ve kalabalıklık mesafesine göre sıralama yaptıktan sonra seçilen çözümlerin bir yüzdesini tutmaktır (Durillo, 2007).

3.6.2.11. MOEAD (Multiobjective Evolutionary Algorithm with Decomposition)

Ayrıştırma temelli çok amaçlı optimizasyon algoritmasıdır. DE ve polinaminal mutasyon kullanılarak çeşitlilik sağlanmıştır. MOEA / D aynı zaman diliminde olarak bir dizi tek amaçlı optimizasyon alt problemini optimize eder (Li ve Zhang, 2009).

3.6.2.12. OMOPSO (Multiobjective Particle Swarm Optimization)

Pareto hakimiyetine ve mevcut liderlerin listesini filtrelemek için kalabalık bir faktör kullanılmasına dayanan çok amaçlı optimizasyon algoritmasıdır (Reyes Sierra ve Coello Coello, 2005). PSO algoritmasının geliştirilmesinde bastırılmışlık sıralama ve mutasyon özelliklerinin eklenmiş versiyonudur.

3.6.2.13. PAES (Pareto Archived Evolution Strategy)

Evrimsel tabanlı çok amaçlı optimizasyon algoritmasıdır. Algoritma, bir popülasyondan yerel arama kullanılarak fakat mevcut ve aday çözüm vektörlerinin yaklaşık baskınlık sıralamasını belirlemek için önceden bulunan çözümlerin referans arşivini kullanarak (1 + 1) bir evrim stratejisi olarak tanımlanır (Knowles ve Corne, 1999). Yeni çözümler üretirken arşivleme yöntemi kullanarak çeşitliliği sağlamaktadır.

3.6.2.14. PESA2 (Pareto Envelope-based Selection Algorithm)

Bölge temelli yaklaşımı benimseyen çok amaçlı algoritmadır. Amaç fonksiyonunda seçim işleminde hyperbox yöntemi kullanılır. Çözümlere değil oluşturulan bu kutulara uygunluk değeri atanır. Böylece sonuçta hiperbox seçilir ve ortaya çıkan seçilmiş birey bu hiper-kutudan rastgele seçilir (Corne ve ark., 2001).

3.6.2.15. SMPSO (Speed-Constrained Multiobjective Particle Swarm Optimization)

PSO algoritmasının geliştirilmesi ile oluşan varyantıdır. Algoritmada hızın aşırı arttığı zaman dilimleri tespit edilmekte ve bu durumlarda yeni pozisyonların belirlenip atanması sağlanmaktadır. Doğruluk ve hız bakımından diğer algoritmalara oranla daha iyi sonuçlar gözlemlenmiştir (Nebro ve ark., 2009).

3.6.2.16. SPEA2 (Strength-based Evolutionary Algorithm)

Popülasyon temelli, arşiv mantığında kurulan SPEA' nın geliştirilerek elde edilen çok amaçlı genetik algoritmadır. Uygunluk atama, yoğunluk tahmini ve arşiv kısaltma yöntemleri ile geliştirilmiştir (Zitzler ve ark., 2001).

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Araştırma Sonuçları

Bu tez çalışmasında çok amaçlı optimizasyon problemlerine çözüm üretmek amacıyla TSA prensibinden geliştirilen MOTSA tanıtılmıştır. Önerilen algoritmanın gelişim süreci tanımlanmış, uygulanan testler ile çözüm kalitesi belirlenmiştir.

Gerçekleştirilen çalışmada, öncelik olarak algoritma tasarımı gerçekleştirilmiştir. Önerilen algoritma Hypervolume, Epsilon, Spread ve IGD kalite ölçüm birimleri hesaplanmıştır. Algoritma eş giriş parametreleri kullanılarak çok amaçlı optimizasyon algoritmaları ile kıyaslanmıştır. Önerilen algoritma diğer algoritmalar ile kıyaslanırken doğru Pareto çözümlere olan yakınlığı, Pareto yüzeyde yayılımı ve çözümler arasındaki yakınsama değerleri ölçümleri gerçekleştirilmiştir. Bu tez çalışmasında tanıtılan 16 algoritmayla birlikte gelişiminde örnek alınan NSGAIII algoritması, Fonseca, Kursawe, Poloni, ConstrEx, Srinivas, Viennet2, Viennet3, Viennet4 ve OKA1 test fonksiyonları ile HV, Spread, IGD ve Epsilon kriterlerinin ortanca ve standart sapma kullanılarak sonuçlar elde edilmiştir. Seçilen özellikler üzerinde gerçekleştirilen toplam 648 testin sadece %10,64'ünde diğer algoritmalar MOTSA 'dan daha başarılı olmuştur. Bu sonuç MOTSA'nın %89,35 oranında testlerde başarı elde ettiğini göstermektedir.

Her bir amaç fonksiyonu için 68 testin gerçekleştirildiği bu çalışmada, MOTSA başarı oranı olarak Fonseca, Kursawe, Poloni ve Srinivas amaç fonksiyonlarında bütün ölçütlerde daha kaliteli değere sahip olduğu gözlenmiştir. ConstrEx ve OKA1 için %75, Viennet4 ve Viennet2 için %82,35 ve son olarak Viennet3 için %83,82 diğer algoritmalarından daha yüksek kalitede değere sahip olan MOTSA çalışma genelinde yaklaşık olarak %89 başarı oranına sahiptir.

Kullanılan amaç fonksiyonuna bağlı olarak farklı sonuçların elde edildiğinin gözlemlendiği çalışmada fonksiyonlara bağlı olarak performans kriterlerinin sonuçları farklılık göstermektedir. Çalışmada seçilen performans kriterlerinden bir referans noktasından çözümler arasındaki uzaklığın kullanılarak hesaplandığı IGD ölçütünün yapılan testlerin tamamında MOTSA 'yı daha kaliteli olarak seçtiği gözlemlenmektedir. Algoritma yakınsama değerini belirleyen bu kriter MOTSA 'nın yakınsama açısından diğer algoritmalarından daha kaliteli bir değere sahip olduğunu göstermektedir.

Çizelge 4.1’de Fonseca uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Fonseca fonksiyonu için belirlenen ölçütlerin tümünde diğer 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir.

Çizelge 4.1. Fonseca fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	3.10e - 01 _{2.9e-04}	2.85e - 01 _{2.7e-02}	2.18e - 04 _{1.2e-05}	1.15e - 02 _{1.5e-03}
AbYSS	3.08e - 01 _{3.8e-04}	4.00e - 01 _{3.0e-02}	3.21e - 04 _{1.4e-05}	1.33e - 02 _{1.7e-03}
CellIDE	3.08e - 01 _{3.7e-04}	3.98e - 01 _{3.6e-02}	3.19e - 04 _{1.2e-05}	1.33e - 02 _{1.9e-03}
DENSEA	3.08e - 01 _{3.7e-04}	4.01e - 01 _{3.7e-02}	3.21e - 04 _{1.6e-05}	1.33e - 02 _{2.0e-03}
dMOPSO	3.08e - 01 _{4.1e-04}	4.02e - 01 _{3.0e-02}	3.22e - 04 _{1.4e-05}	1.33e - 02 _{1.7e-03}
GDE3	3.08e - 01 _{4.4e-04}	4.01e - 01 _{3.5e-02}	3.21e - 04 _{1.4e-05}	1.32e - 02 _{1.8e-03}
GWASFGA	3.08e - 01 _{4.2e-04}	4.03e - 01 _{3.3e-02}	3.21e - 04 _{1.3e-05}	1.35e - 02 _{1.8e-03}
IBEA	3.08e - 01 _{4.1e-04}	4.00e - 01 _{3.8e-02}	3.21e - 04 _{1.3e-05}	1.35e - 02 _{1.8e-03}
MOCcell	3.08e - 01 _{3.2e-04}	4.05e - 01 _{3.1e-02}	3.21e - 04 _{1.1e-05}	1.32e - 02 _{1.8e-03}
MOCHC	3.08e - 01 _{4.0e-04}	4.08e - 01 _{3.4e-02}	3.21e - 04 _{1.4e-05}	1.33e - 02 _{1.8e-03}
MOEAD	3.08e - 01 _{4.2e-04}	4.06e - 01 _{3.6e-02}	3.21e - 04 _{1.5e-05}	1.32e - 02 _{1.8e-03}
NSGAI	3.08e - 01 _{3.7e-04}	4.09e - 01 _{3.5e-02}	3.21e - 04 _{1.5e-05}	1.33e - 02 _{2.0e-03}
OMOPSO	3.08e - 01 _{3.9e-04}	3.98e - 01 _{3.6e-02}	3.21e - 04 _{1.5e-05}	1.32e - 02 _{1.8e-03}
PAES	3.08e - 01 _{4.4e-04}	4.02e - 01 _{4.0e-02}	3.22e - 04 _{1.5e-05}	1.32e - 02 _{1.7e-03}
PESA2	3.08e - 01 _{3.9e-04}	4.03e - 01 _{3.4e-02}	3.17e - 04 _{1.2e-05}	1.32e - 02 _{2.1e-03}
SMPSO	3.08e - 01 _{3.3e-04}	3.97e - 01 _{2.8e-02}	3.20e - 04 _{1.3e-05}	1.30e - 02 _{1.5e-03}
SPEA2	3.08e - 01 _{4.0e-04}	4.01e - 01 _{3.4e-02}	3.20e - 04 _{1.2e-05}	1.34e - 02 _{2.0e-03}
WASFGA	3.08e - 01 _{3.7e-04}	4.01e - 01 _{3.2e-02}	3.19e - 04 _{1.3e-05}	1.35e - 02 _{1.9e-03}

Çizelge 4.2’de Kursawe uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Kursawe fonksiyonu için belirlenen ölçütlerin tümünde diğer 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir.

Çizelge 4.2. Kursawe fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	4.00e - 01 _{5.1e-04}	5.05e - 01 _{2.2e-02}	1.58e - 04 _{7.1e-06}	7.18e - 02 _{1.0e-02}
AbYSS	4.00e - 01 _{2.5e-04}	5.67e - 01 _{2.4e-02}	1.80e - 04 _{8.4e-06}	7.93e - 02 _{1.2e-02}
CellIDE	4.00e - 01 _{2.6e-04}	5.70e - 01 _{2.6e-02}	1.83e - 04 _{1.0e-05}	8.05e - 02 _{1.2e-02}
DENSEA	4.00e - 01 _{2.7e-04}	5.69e - 01 _{2.5e-02}	1.80e - 04 _{9.3e-06}	8.16e - 02 _{1.2e-02}
dMOPSO	4.00e - 01 _{2.5e-04}	5.67e - 01 _{2.5e-02}	1.80e - 04 _{8.2e-06}	7.97e - 02 _{1.0e-02}
GDE3	4.00e - 01 _{2.2e-04}	5.69e - 01 _{2.4e-02}	1.80e - 04 _{1.0e-05}	7.94e - 02 _{1.0e-02}
GWASFGA	4.00e - 01 _{2.4e-04}	5.70e - 01 _{2.4e-02}	1.80e - 04 _{9.5e-06}	7.89e - 02 _{9.7e-03}
IBEA	4.00e - 01 _{2.6e-04}	5.65e - 01 _{2.4e-02}	1.80e - 04 _{9.4e-06}	8.05e - 02 _{1.4e-02}
MOCcell	4.00e - 01 _{2.5e-04}	5.70e - 01 _{2.6e-02}	1.81e - 04 _{9.2e-06}	8.12e - 02 _{1.3e-02}
MOCHC	4.00e - 01 _{2.4e-04}	5.64e - 01 _{2.5e-02}	1.82e - 04 _{8.9e-06}	8.05e - 02 _{1.3e-02}
MOEAD	4.00e - 01 _{2.5e-04}	5.68e - 01 _{2.8e-02}	1.80e - 04 _{9.3e-06}	7.93e - 02 _{1.2e-02}
NSGAI	4.00e - 01 _{2.5e-04}	5.72e - 01 _{2.5e-02}	1.80e - 04 _{9.4e-06}	8.16e - 02 _{1.3e-02}
OMOPSO	4.00e - 01 _{2.6e-04}	5.67e - 01 _{2.4e-02}	1.81e - 04 _{8.5e-06}	8.06e - 02 _{1.3e-02}
PAES	4.00e - 01 _{2.4e-04}	5.65e - 01 _{2.6e-02}	1.82e - 04 _{9.8e-06}	7.90e - 02 _{1.1e-02}
PESA2	4.00e - 01 _{2.2e-04}	5.71e - 01 _{2.6e-02}	1.82e - 04 _{9.8e-06}	8.16e - 02 _{1.2e-02}
SMPSO	4.00e - 01 _{2.2e-04}	5.67e - 01 _{2.7e-02}	1.79e - 04 _{1.0e-05}	7.91e - 02 _{1.1e-02}
SPEA2	4.00e - 01 _{2.7e-04}	5.62e - 01 _{2.2e-02}	1.81e - 04 _{1.1e-05}	8.01e - 02 _{1.3e-02}
WASFGA	4.00e - 01 _{2.5e-04}	5.66e - 01 _{2.6e-02}	1.80e - 04 _{9.2e-06}	7.88e - 02 _{1.2e-02}

Çizelge 4.3’de Poloni uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Poloni fonksiyonu için belirlenen ölçütlerin tümünde diğer 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir.

Çizelge 4.3. Poloni fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	9.13e - 01 _{5.2e-05}	7.79e - 01 _{2.2e-02}	3.89e - 05 _{1.4e-06}	1.16e - 01 _{1.8e-02}
AbYSS	9.13e - 01 _{8.6e-05}	8.05e - 01 _{2.1e-02}	4.51e - 05 _{3.0e-06}	1.40e - 01 _{3.0e-02}
CellIDE	9.13e - 01 _{8.2e-05}	7.87e - 01 _{1.7e-02}	4.50e - 05 _{3.1e-06}	1.46e - 01 _{3.2e-02}
DENSEA	9.13e - 01 _{7.6e-05}	8.03e - 01 _{1.8e-02}	4.53e - 05 _{2.4e-06}	1.40e - 01 _{2.7e-02}
dMOPSO	9.13e - 01 _{8.4e-05}	8.07e - 01 _{2.1e-02}	4.51e - 05 _{2.9e-06}	1.45e - 01 _{3.1e-02}
GDE3	9.13e - 01 _{8.2e-05}	8.05e - 01 _{2.0e-02}	4.51e - 05 _{3.7e-06}	1.44e - 01 _{3.0e-02}
GWASFGA	9.13e - 01 _{7.9e-05}	8.04e - 01 _{1.8e-02}	4.52e - 05 _{2.8e-06}	1.46e - 01 _{3.3e-02}
IBEA	9.13e - 01 _{8.6e-05}	8.03e - 01 _{1.8e-02}	4.50e - 05 _{2.2e-06}	1.45e - 01 _{3.0e-02}
MOCcell	9.13e - 01 _{8.5e-05}	8.01e - 01 _{2.0e-02}	4.52e - 05 _{2.5e-06}	1.41e - 01 _{2.9e-02}
MOCHC	9.13e - 01 _{9.8e-05}	8.07e - 01 _{2.0e-02}	4.48e - 05 _{2.9e-06}	1.44e - 01 _{3.2e-02}
MOEAD	9.13e - 01 _{7.2e-05}	8.07e - 01 _{2.3e-02}	4.52e - 05 _{3.0e-06}	1.44e - 01 _{2.9e-02}
NSGAI	9.13e - 01 _{8.3e-05}	8.06e - 01 _{2.1e-02}	4.56e - 05 _{3.1e-06}	1.46e - 01 _{3.2e-02}
OMOPSO	9.13e - 01 _{8.8e-05}	8.03e - 01 _{1.7e-02}	4.55e - 05 _{2.9e-06}	1.48e - 01 _{3.1e-02}
PAES	9.13e - 01 _{7.2e-05}	8.06e - 01 _{2.1e-02}	4.51e - 05 _{2.8e-06}	1.40e - 01 _{2.9e-02}
PESA2	9.13e - 01 _{9.9e-05}	8.04e - 01 _{2.3e-02}	4.44e - 05 _{2.1e-06}	1.47e - 01 _{3.3e-02}
SMPSO	9.13e - 01 _{8.3e-05}	8.05e - 01 _{2.0e-02}	4.53e - 05 _{3.4e-06}	1.40e - 01 _{2.8e-02}
SPEA2	9.13e - 01 _{8.8e-05}	8.09e - 01 _{1.9e-02}	4.49e - 05 _{2.7e-06}	1.42e - 01 _{3.0e-02}
WASFGA	9.13e - 01 _{8.0e-05}	8.04e - 01 _{1.9e-02}	4.55e - 05 _{2.9e-06}	1.40e - 01 _{2.5e-02}

Çizelge 4.4'de ConstrEx uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde ConstrEx fonksiyonu için belirlenen ölçütlerden Hipervolume dışında üç ölçütte diğer 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir.

Çizelge 4.4. ConstrEx fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	7.74e - 01 _{4.3e-04}	3.42e - 01 _{2.6e-02}	2.65e - 04 _{2.0e-05}	1.34e - 02 _{1.6e-03}
AbYSS	7.74e - 01 _{3.2e-04}	4.65e - 01 _{3.8e-02}	2.79e - 04 _{1.5e-05}	1.57e - 02 _{2.4e-03}
CellIDE	7.74e - 01 _{3.0e-04}	4.70e - 01 _{4.0e-02}	2.81e - 04 _{1.7e-05}	1.66e - 02 _{3.4e-03}
DENSEA	7.74e - 01 _{2.9e-04}	4.64e - 01 _{3.7e-02}	2.78e - 04 _{1.3e-05}	1.58e - 02 _{2.7e-03}
dMOPSO	7.74e - 01 _{3.0e-04}	4.62e - 01 _{3.6e-02}	2.78e - 04 _{1.4e-05}	1.57e - 02 _{2.7e-03}
GDE3	7.74e - 01 _{3.2e-04}	4.66e - 01 _{4.0e-02}	2.82e - 04 _{1.7e-05}	1.65e - 02 _{3.3e-03}
GWASFGA	7.74e - 01 _{2.9e-04}	4.64e - 01 _{4.0e-02}	2.79e - 04 _{1.6e-05}	1.56e - 02 _{3.0e-03}
IBEA	7.74e - 01 _{2.9e-04}	4.68e - 01 _{3.5e-02}	2.81e - 04 _{1.2e-05}	1.61e - 02 _{2.6e-03}
MOCcell	7.74e - 01 _{3.0e-04}	4.65e - 01 _{3.4e-02}	2.77e - 04 _{1.6e-05}	1.56e - 02 _{2.5e-03}
MOCHC	7.74e - 01 _{3.1e-04}	4.62e - 01 _{4.2e-02}	2.79e - 04 _{1.6e-05}	1.60e - 02 _{2.9e-03}
MOEAD	7.74e - 01 _{2.8e-04}	4.61e - 01 _{3.9e-02}	2.78e - 04 _{1.5e-05}	1.64e - 02 _{4.1e-03}
NSGAI	7.74e - 01 _{3.1e-04}	4.69e - 01 _{4.1e-02}	2.82e - 04 _{1.6e-05}	1.64e - 02 _{3.1e-03}
OMOPSO	7.74e - 01 _{2.9e-04}	4.65e - 01 _{3.6e-02}	2.79e - 04 _{1.6e-05}	1.62e - 02 _{3.4e-03}
PAES	7.74e - 01 _{3.0e-04}	4.69e - 01 _{4.2e-02}	2.82e - 04 _{1.7e-05}	2.78e - 04 _{1.8e-05}
PESA2	7.74e - 01 _{2.8e-04}	4.62e - 01 _{3.7e-02}	2.78e - 04 _{1.4e-05}	1.61e - 02 _{3.0e-03}
SMPSO	7.74e - 01 _{2.8e-04}	4.60e - 01 _{3.9e-02}	2.77e - 04 _{1.3e-05}	1.63e - 02 _{3.4e-03}
SPEA2	7.74e - 01 _{3.0e-04}	4.68e - 01 _{3.9e-02}	2.80e - 04 _{1.4e-05}	1.61e - 02 _{3.0e-03}
WASFGA	7.74e - 01 _{3.1e-04}	4.60e - 01 _{3.8e-02}	2.78e - 04 _{1.6e-05}	1.59e - 02 _{2.8e-03}

Çizelge 4.5’de Srinivas uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Srinivas fonksiyonu için belirlenen ölçütlerin tümünde diğer 17 fonksiyondan daha kaliteli çözüm üretebildiği görülmektedir.

Çizelge 4.5. Srinivas fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	5.39e - 01 _{2.2e-04}	2.90e - 01 _{2.5e-02}	8.48e - 05 _{2.8e-06}	2.56e + 00 _{3.0e-01}
AbYSS	5.38e - 01 _{4.0e-04}	4.06e - 01 _{3.2e-02}	1.01e - 04 _{7.4e-06}	3.55e + 00 _{6.9e-01}
CellIDE	5.38e - 01 _{3.6e-04}	4.05e - 01 _{3.1e-02}	1.01e - 04 _{7.0e-06}	3.42e + 00 _{6.1e-01}
DENSEA	5.38e - 01 _{3.5e-04}	4.04e - 01 _{3.5e-02}	1.01e - 04 _{7.3e-06}	3.43e + 00 _{6.0e-01}
dMOPSO	5.38e - 01 _{4.1e-04}	4.06e - 01 _{3.3e-02}	1.02e - 04 _{7.9e-06}	3.46e + 00 _{7.1e-01}
GDE3	5.38e - 01 _{3.3e-04}	3.99e - 01 _{3.1e-02}	1.01e - 04 _{8.1e-06}	3.42e + 00 _{6.4e-01}
GWASFGA	5.38e - 01 _{3.6e-04}	3.99e - 01 _{3.4e-02}	1.00e - 04 _{6.4e-06}	3.40e + 00 _{5.9e-01}
IBEA	5.38e - 01 _{3.8e-04}	4.00e - 01 _{3.3e-02}	1.01e - 04 _{6.9e-06}	3.32e + 00 _{5.2e-01}
MOCcell	5.38e - 01 _{3.7e-04}	4.08e - 01 _{3.5e-02}	1.01e - 04 _{6.6e-06}	3.40e + 00 _{5.9e-01}
MOCHC	5.38e - 01 _{3.6e-04}	3.96e - 01 _{3.0e-02}	9.96e - 05 _{6.0e-06}	3.32e + 00 _{5.8e-01}
MOEAD	5.38e - 01 _{4.2e-04}	4.09e - 01 _{3.3e-02}	1.02e - 04 _{8.5e-06}	3.46e + 00 _{6.4e-01}
NSGAI	5.38e - 01 _{3.3e-04}	4.01e - 01 _{3.4e-02}	1.00e - 04 _{7.0e-06}	3.38e + 00 _{5.6e-01}
OMOPSO	5.38e - 01 _{3.4e-04}	3.98e - 01 _{3.0e-02}	9.99e - 05 _{6.3e-06}	3.41e + 00 _{5.7e-01}
PAES	5.38e - 01 _{3.3e-04}	3.97e - 01 _{3.0e-02}	1.01e - 04 _{8.3e-06}	3.42e + 00 _{6.9e-01}
PESA2	5.38e - 01 _{2.8e-04}	4.00e - 01 _{2.6e-02}	1.01e - 04 _{7.2e-06}	3.40e + 00 _{5.5e-01}
SMPSO	5.38e - 01 _{4.0e-04}	3.99e - 01 _{2.9e-02}	1.00e - 04 _{8.0e-06}	3.33e + 00 _{6.2e-01}
SPEA2	5.38e - 01 _{3.8e-04}	4.10e - 01 _{3.4e-02}	.02e - 04 _{1.0e-05}	3.44e + 00 _{7.0e-01}
WASFGA	5.38e - 01 _{3.7e-04}	4.03e - 01 _{3.2e-02}	1.00e - 04 _{6.7e-06}	3.34e + 00 _{5.7e-01}

Çizelge 4.6’de Viennet2 uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Viennet2 fonksiyonu için belirlenen ölçütlerden Spread dışında diğer 3 ölçütte belirlenen 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir. Bu fonksiyonda Spread ölçütünde NSGAIİ’nde içinde bulunan 5 algoritmaya göre MOTSA daha kaliteli sonuç elde etmiştir.

Çizelge 4.6. Viennet2 fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	9.22e - 01 _{1.4e-03}	8.44e - 01 _{8.6e-02}	3.06e - 04 _{3.1e-05}	3.08e - 02 _{5.8e-03}
AbYSS	9.20e - 01 _{1.4e-03}	8.50e - 01 _{8.7e-02}	3.42e - 04 _{4.1e-05}	3.44e - 02 _{7.3e-03}
CellDE	9.20e - 01 _{1.3e-03}	8.36e - 01 _{9.1e-02}	3.32e - 04 _{3.8e-05}	3.42e - 02 _{8.0e-03}
DENSEA	9.20e - 01 _{1.4e-03}	8.29e - 01 _{9.4e-02}	3.33e - 04 _{3.9e-05}	3.35e - 02 _{7.4e-03}
dMOPSO	9.20e - 01 _{1.6e-03}	8.29e - 01 _{8.4e-02}	3.36e - 04 _{4.7e-05}	3.48e - 02 _{8.9e-03}
GDE3	9.20e - 01 _{2.2e-03}	8.45e - 01 _{8.6e-02}	3.40e - 04 _{5.6e-05}	3.61e - 02 _{9.9e-03}
GWASFGA	9.20e - 01 _{1.6e-03}	8.39e - 01 _{8.7e-02}	3.36e - 04 _{4.3e-05}	3.40e - 02 _{9.9e-03}
IBEA	9.20e - 01 _{1.6e-03}	8.31e - 01 _{8.9e-02}	3.24e - 04 _{3.6e-05}	3.41e - 02 _{8.7e-03}
MOCeII	9.20e - 01 _{1.7e-03}	8.38e - 01 _{9.7e-02}	3.35e - 04 _{4.4e-05}	3.39e - 02 _{8.3e-03}
MOCHC	9.20e - 01 _{1.5e-03}	8.28e - 01 _{9.4e-02}	3.41e - 04 _{4.4e-05}	3.40e - 02 _{8.5e-03}
MOEAD	9.20e - 01 _{1.7e-03}	8.21e - 01 _{9.7e-02}	3.39e - 04 _{4.1e-05}	3.41e - 02 _{9.0e-03}
NSGAIİ	9.20e - 01 _{1.4e-03}	8.40e - 01 _{8.5e-02}	3.30e - 04 _{3.8e-05}	3.40e - 02 _{8.0e-03}
OMOPSO	9.20e - 01 _{1.3e-03}	8.38e - 01 _{9.4e-02}	3.33e - 04 _{3.7e-05}	3.32e - 02 _{8.1e-03}
PAES	9.20e - 01 _{1.5e-03}	8.27e - 01 _{8.5e-02}	3.37e - 04 _{4.2e-05}	3.29e - 02 _{7.5e-03}
PESA2	9.20e - 01 _{1.5e-03}	8.21e - 01 _{1.0e-01}	3.39e - 04 _{4.0e-05}	3.39e - 02 _{8.8e-03}
SMPSO	9.20e - 01 _{1.9e-03}	8.28e - 01 _{8.7e-02}	3.38e - 04 _{4.6e-05}	3.49e - 02 _{9.9e-03}
SPEA2	9.20e - 01 _{1.4e-03}	8.45e - 01 _{9.3e-02}	3.33e - 04 _{3.8e-05}	3.40e - 02 _{8.5e-03}
WASFGA	9.20e - 01 _{1.5e-03}	8.40e - 01 _{9.2e-02}	3.37e - 04 _{4.5e-05}	3.44e - 02 _{8.7e-03}

Çizelge 4.7’de Viennet3 uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Viennet3 fonksiyonu için belirlenen ölçütlerden Epsilon dışında diğer 3 ölçütte belirlenen 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir. Bu fonksiyonda Spread ölçütünde 17 algoritmadan 6’sında MOTSA daha kaliteli sonuç elde etmiştir.

Çizelge 4.7. Viennet3 fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	8.33e - 01 _{5.7e-04}	6.93e - 01 _{5.8e-02}	1.63e - 04 _{1.8e-05}	4.83e - 02 _{1.1e-02}
AbYSS	8.33e - 01 _{6.6e-04}	7.26e - 01 _{6.1e-02}	1.73e - 04 _{1.9e-05}	4.98e - 02 _{1.1e-02}
CellDE	8.33e - 01 _{6.3e-04}	7.33e - 01 _{5.1e-02}	1.70e - 04 _{1.7e-05}	4.94e - 02 _{1.1e-02}
DENSEA	8.32e - 01 _{7.8e-04}	7.29e - 01 _{5.4e-02}	1.74e - 04 _{2.4e-05}	4.98e - 02 _{1.1e-02}
dMOPSO	8.33e - 01 _{6.2e-04}	7.19e - 01 _{5.9e-02}	1.75e - 04 _{2.2e-05}	4.94e - 02 _{1.1e-02}
GDE3	4.94e - 02 _{1.6e-02}	7.30e - 01 _{5.6e-02}	1.76e - 04 _{2.5e-05}	5.00e - 02 _{1.1e-02}
GWASFGA	8.33e - 01 _{5.9e-04}	7.22e - 01 _{5.3e-02}	1.73e - 04 _{1.9e-05}	5.08e - 02 _{1.1e-02}
IBEA	8.33e - 01 _{6.0e-04}	7.27e - 01 _{5.7e-02}	1.73e - 04 _{1.9e-05}	4.70e - 02 _{1.0e-02}
MOCeII	8.33e - 01 _{5.9e-04}	7.25e - 01 _{5.6e-02}	1.72e - 04 _{1.8e-05}	4.70e - 02 _{1.0e-02}
MOCHC	8.33e - 01 _{6.9e-04}	7.23e - 01 _{5.5e-02}	1.71e - 04 _{2.0e-05}	4.81e - 02 _{1.0e-02}
MOEAD	8.33e - 01 _{7.4e-04}	7.30e - 01 _{5.8e-02}	1.70e - 04 _{1.9e-05}	4.92e - 02 _{1.2e-02}
NSGAI	8.33e - 01 _{6.1e-04}	7.28e - 01 _{5.5e-02}	1.72e - 04 _{1.8e-05}	4.89e - 02 _{1.2e-02}
OMOPSO	8.33e - 01 _{5.6e-04}	7.31e - 01 _{5.3e-02}	1.70e - 04 _{1.7e-05}	4.75e - 02 _{9.6e-03}
PAES	8.33e - 01 _{7.7e-04}	7.22e - 01 _{6.0e-02}	1.74e - 04 _{2.0e-05}	4.89e - 02 _{1.2e-02}
PESA2	8.32e - 01 _{7.6e-04}	7.30e - 01 _{5.5e-02}	1.72e - 04 _{1.9e-05}	5.14e - 02 _{1.1e-02}
SMPSO	8.33e - 01 _{6.6e-04}	7.28e - 01 _{5.8e-02}	1.75e - 04 _{2.4e-05}	4.78e - 02 _{1.1e-02}
SPEA2	8.33e - 01 _{6.1e-04}	7.26e - 01 _{5.3e-02}	1.73e - 04 _{1.7e-05}	4.67e - 02 _{1.1e-02}
WASFGA	8.33e - 01 _{7.3e-04}	7.29e - 01 _{5.3e-02}	1.72e - 04 _{2.0e-05}	4.87e - 02 _{1.2e-02}

Çizelge 4.8’de Viennet4 uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde Viennet4 fonksiyonu için belirlenen ölçütlerden Hipervolume ve IGD ölçütlerinde belirlenen 17 fonksiyondan daha kaliteli çözüm üretilebildiği görülmektedir. Yine grafiğe göre bu fonksiyonda Spread ölçütünde 17 algoritmadan 13’ünde, Epsilon ölçütünde 9’unda MOTSA daha kaliteli sonuç elde etmiştir.

Çizelge 4.8. Viennet4 fonksiyonu test sonuçları

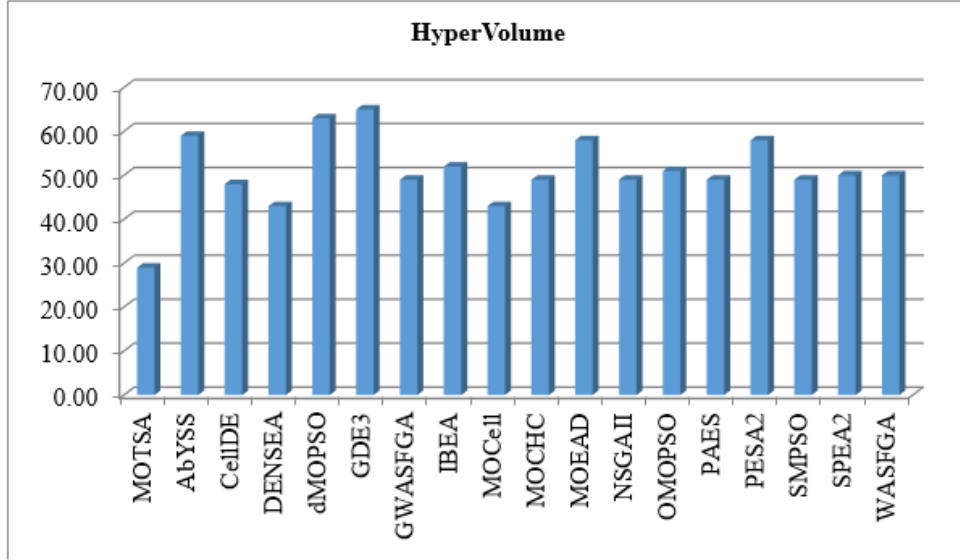
Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	8.55e - 01 _{2.1e-03}	9.26e - 01 _{5.0e-02}	3.69e - 04 _{1.9e-05}	2.89e - 01 _{3.8e-02}
AbYSS	8.54e - 01 _{2.7e-03}	9.12e - 01 _{5.4e-02}	3.91e - 04 _{2.5e-05}	2.94e - 01 _{4.8e-02}
CellIDE	8.54e - 01 _{2.5e-03}	9.24e - 01 _{5.7e-02}	3.91e - 04 _{3.0e-05}	2.91e - 01 _{4.8e-02}
DENSEA	8.55e - 01 _{2.2e-03}	9.19e - 01 _{5.4e-02}	3.81e - 04 _{2.4e-05}	2.80e - 01 _{4.2e-02}
dMOPSO	8.54e - 01 _{2.5e-03}	9.29e - 01 _{5.1e-02}	3.84e - 04 _{2.2e-05}	2.89e - 01 _{4.1e-02}
GDE3	8.54e - 01 _{2.3e-03}	9.15e - 01 _{5.1e-02}	3.87e - 04 _{2.8e-05}	2.86e - 01 _{4.5e-02}
GWASFGA	8.54e - 01 _{2.3e-03}	9.23e - 01 _{5.5e-02}	3.82e - 04 _{2.3e-05}	2.80e - 01 _{3.8e-02}
IBEA	8.54e - 01 _{2.2e-03}	9.21e - 01 _{5.0e-02}	3.87e - 04 _{2.6e-05}	2.80e - 01 _{4.0e-02}
MOCeII	8.54e - 01 _{2.1e-03}	9.21e - 01 _{5.1e-02}	3.90e - 04 _{2.9e-05}	2.81e - 01 _{4.2e-02}
MOCHC	8.55e - 01 _{2.2e-03}	9.18e - 01 _{6.0e-02}	3.87e - 04 _{2.4e-05}	2.81e - 01 _{4.1e-02}
MOEAD	8.55e - 01 _{2.2e-03}	9.12e - 01 _{5.3e-02}	3.83e - 04 _{2.8e-05}	2.83e - 01 _{3.9e-02}
NSGAI	8.54e - 01 _{2.4e-03}	9.16e - 01 _{5.3e-02}	3.85e - 04 _{2.3e-05}	2.85e - 01 _{4.6e-02}
OMOPSO	8.54e - 01 _{2.4e-03}	9.28e - 01 _{4.8e-02}	3.88e - 04 _{2.4e-05}	2.91e - 01 _{4.8e-02}
PAES	8.55e - 01 _{2.1e-03}	9.18e - 01 _{5.2e-02}	3.83e - 04 _{2.1e-05}	2.81e - 01 _{4.7e-02}
PESA2	8.54e - 01 _{2.1e-03}	9.25e - 01 _{5.3e-02}	3.84e - 04 _{2.2e-05}	2.83e - 01 _{4.5e-02}
SMPSO	8.54e - 01 _{2.5e-03}	9.23e - 01 _{5.8e-02}	3.86e - 04 _{2.4e-05}	2.83e - 01 _{4.5e-02}
SPEA2	8.54e - 01 _{2.2e-03}	9.26e - 01 _{5.5e-02}	3.88e - 04 _{2.6e-05}	2.88e - 01 _{4.5e-02}
WASFGA	8.55e - 01 _{2.3e-03}	9.28e - 01 _{5.2e-02}	9.32e - 01 _{6.6e-02}	2.81e - 01 _{4.5e-02}

Çizelge 4.9’de OKA1 uygunluk fonksiyonu için yapılan test sonuçları gösterilmektedir. Tablo incelendiğinde OKA1 fonksiyonu için belirlenen ölçütlerden Hipervolume dışında tüm ölçütlerde belirlenen 17 fonksiyondan daha kaliteli çözüm üretilbildiği görülmektedir.

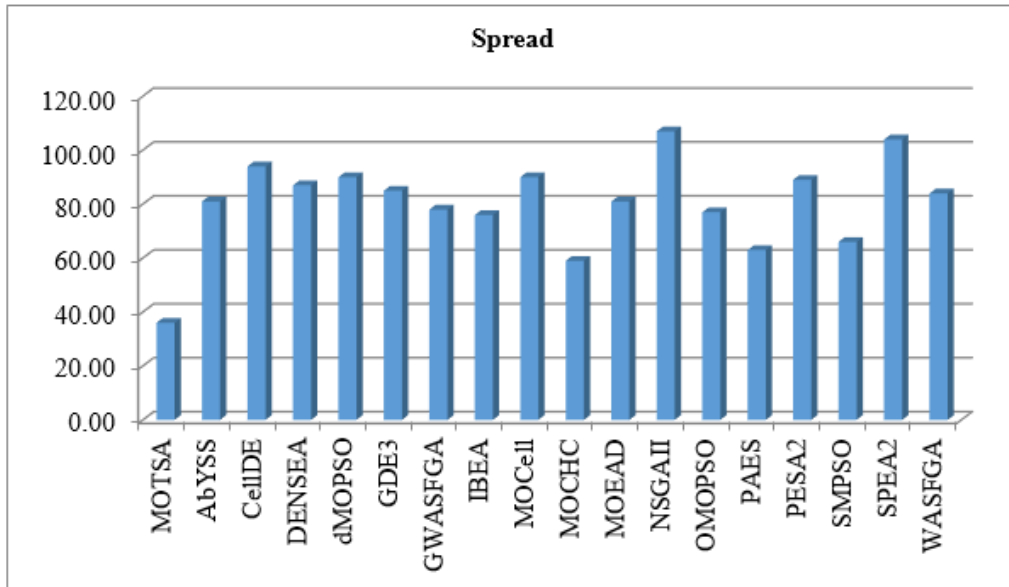
Çizelge 4.9. OKA1 fonksiyonu test sonuçları

Algoritma	HV. Ortalama ve standart sapma	SPREAD. Ortalama ve standart sapma	IGD. Ortalama ve standart sapma	EPSILON. Ortalama ve standart sapma
MOTSA	6.94e - 01 _{5.9e-03}	1.00e + 00 _{5.4e-02}	1.78e - 04 _{2.5e-05}	4.11e - 01 _{5.8e-02}
AbYSS	7.15e - 01 _{2.8e-03}	1.13e + 00 _{4.4e-02}	3.51e - 03 _{2.4e-06}	4.74e - 01 _{8.0e-03}
CellIDE	7.15e - 01 _{3.0e-03}	1.13e + 00 _{4.2e-02}	3.51e - 03 _{1.9e-06}	4.73e - 01 _{8.2e-04}
DENSEA	7.16e - 01 _{3.1e-03}	1.14e + 00 _{4.9e-02}	3.52e - 03 _{2.6e-06}	4.74e - 01 _{1.3e-02}
dMOPSO	7.14e - 01 _{3.7e-03}	1.14e + 00 _{4.7e-02}	3.46e - 03 _{3.5e-06}	4.75e - 01 _{2.0e-02}
GDE3	7.14e - 01 _{3.0e-03}	1.13e + 00 _{4.1e-02}	3.48e - 03 _{7.8e-07}	4.73e - 01 _{0.0e+00}
GWASFGA	7.14e - 01 _{3.7e-03}	1.13e + 00 _{4.4e-02}	3.55e - 03 _{1.3e-06}	4.73e - 01 _{4.8e-03}
IBEA	7.16e - 01 _{3.6e-03}	1.14e + 00 _{4.9e-02}	3.56e - 03 _{1.5e-06}	4.73e - 01 _{9.3e-04}
MOCcell	7.16e - 01 _{3.6e-03}	1.13e + 00 _{4.6e-02}	3.52e - 03 _{2.4e-06}	4.74e - 01 _{8.7e-03}
MOCHC	7.17e - 01 _{2.5e-03}	1.13e + 00 _{4.0e-02}	3.58e - 03 _{8.4e-07}	4.73e - 01 _{2.6e-08}
MOEAD	7.14e - 01 _{3.7e-03}	1.13e + 00 _{3.9e-02}	3.50e - 03 _{5.0e-06}	4.76e - 01 _{2.6e-02}
NSGAI	7.14e - 01 _{3.9e-03}	1.14e + 00 _{4.2e-02}	3.53e - 03 _{4.5e-06}	4.78e - 01 _{3.1e-02}
OMOPSO	7.13e - 01 _{3.2e-03}	1.13e + 00 _{4.2e-02}	3.55e - 03 _{1.3e-06}	4.73e - 01 _{0.0e+00}
PAES	7.15e - 01 _{4.2e-03}	1.13e + 00 _{4.1e-02}	3.53e - 03 _{6.6e-06}	.79e - 01 _{3.7e-02}
PESA2	7.13e - 01 _{3.7e-03}	1.14e + 00 _{4.4e-02}	3.52e - 03 _{5.2e-06}	4.77e - 01 _{3.2e-02}
SMPSO	7.16e - 01 _{3.3e-03}	1.14e + 00 _{5.2e-02}	3.53e - 03 _{3.7e-06}	4.76e - 01 _{1.7e-02}
SPEA2	7.17e - 01 _{3.4e-03}	1.13e + 00 _{4.7e-02}	3.57e - 03 _{2.9e-06}	4.75e - 01 _{1.4e-02}
WASFGA	7.15e - 01 _{3.1e-03}	1.14e + 00 _{3.8e-02}	3.51e - 03 _{1.2e-06}	4.73e - 01 _{9.5e-04}

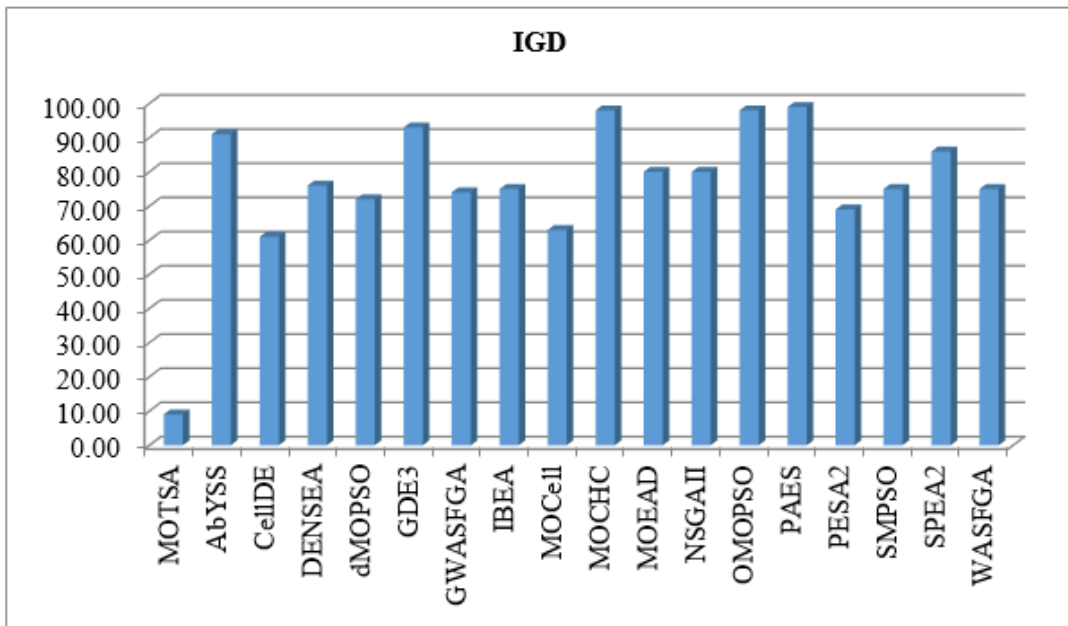
Belirlenen uygunluk fonksiyonları için yapılan test sonuçları toplam rank değeri alınıp Resim 4.1 ile HyperVolume, 4.2 ile Spread, 4.3 ile IGD ve 4.4 ile Epsilon başarı ölçütlerine göre gruplandırılmış grafikleri sunulmuştur. Bu grafiklere göre çalışmada önerilen MOTSA'nın belirlenen ölçütlerde karşılaştırıldığı fonksiyonlardan daha kaliteli IGD değerine sahip olduğu, nesilseller arası mesafe bakımından daha sağlam çözümler ürettiği gözlenmiştir. Bu durum yakınsama özelliğinin diğer algoritmalara oranla çok daha güçlü olduğunu göstermektedir.



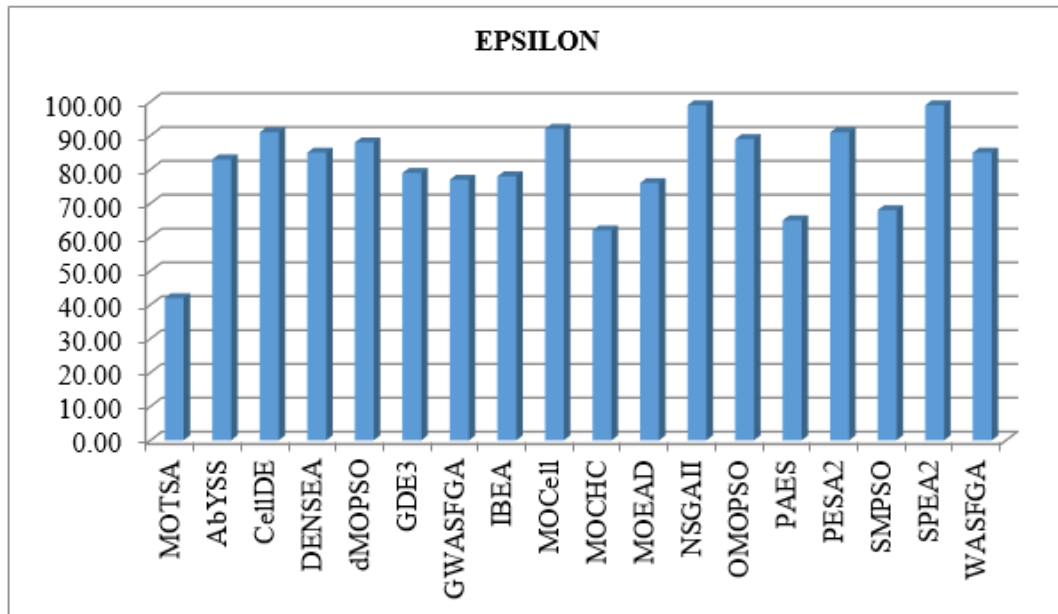
Resim 4.1. HV başarı ölçütü test sonuçları



Resim 4.2. Spread başarı ölçütü test sonuçları



Resim 4.3. IGD başarı ölçütü test sonuçları



Resim 4.4. Epsilon başarı ölçütü test sonuçları

4.2. Tartışma

MOTSA çok amaçlı Pareto'ya dayalı sezgisel bir algoritma olarak önerilmiştir. Bu noktada geniş kullanım alanları sahip algoritma iyi bilinen NSGAIİ'nin genetik birey üretim yapısını ağaç tohum mekanizması sentezleyerek farklılaşmaktadır. Tohum üretiminde kullanılan TSA denklemleri ile arama uzayında hızlı ve iyiye yönelik ilerleyiş sağlanmıştır. Algoritma gelişim sürecinde yapılan testlerde mutasyon işleminin eklenmesi tohum üretiminde çeşitliliği arttırmış, algoritmanın başarı yüzdesine olumlu etki etmiştir.

Optimizasyon algoritmaları probleme yönelik çözümler geliştirirken, problemin boyutu, amaç sayısı kısıtlar gibi etkenlere göre kalite düzeyinde değişim gerçekleşmektedir. Bu çalışmada elde edilen sonuçlar incelendiğinde MOTSA kalitesi problem karakteristiklerinden toplam 9 fonksiyonda %10.64 oranında etkilenmiştir. MOTSA bazı problemlerde tüm algoritmalarından daha kaliteli sonuçlar elde ederken farklı bir amaç fonksiyonunda probleme bağlı olarak kalite ölçütü değeri düşebilmektedir. Literatürde bulunan algoritmaların uygunluk fonksiyonunda aynı ya da birbirine çok yakın kalite değerine sahip olduğu durumlarda MOTSA'nın kalite oranının kısmi bir farkla yüksek olması algoritmanın gerçek Pareto yüzeylere yakınlığı konusundaki başarısını ortaya koymaktadır. Sonuçlarda alınan düşük değerlerin hypervolume ölçütünde yoğunlaşması ve IGD ölçütünde başarı göstermesi Algoritmanın dağılım eğilimi hakkında geliştirilmesi gerektiğini göstermektedir.

Bu çalışmada çok amaçlı optimizasyon için öncelikle iki amaç hedef alınmış, algoritmanın iki amaçlı problemlere karşı tepkisi belirlenmiştir. Yeni önerilen algoritma yapılan testlerde umut veren sonuçlar elde etmiştir. Algoritmanın daha çok ve farklı karakteristikte amaç fonksiyonları üzerinde geliştirilmesi ile bilinen algoritmalarından daha etkili çözümler elde edilebileceği öngörüsü doğmaktadır.

KAYNAKLAR

- Alba, E., Nebro, A. J., Durillo, J. J., Luna, F. ve Dorronsoro, B., 2006, A Cellular Genetic Algorithm for Multiobjective Optimization, Spain.
- Asadzadeh, M., 2016, Desired Precision in Multi-Objective Optimization : Epsilon Archiving or Rounding Objectives?.
- Aslan, M., Beskirli, M., Kodaz, H. ve Kiran, M. S., 2018, An Improved Tree Seed Algorithm for Optimization Problems, *International Journal of Machine Learning and Computing*, 8 (1), 20-25.
- Babalık, A., Çınar, A. C. ve Kiran, M. S., 2018, A modification of tree-seed algorithm using Deb's rules for constrained optimization, *Applied Soft Computing*, 63, 289-305.
- Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L. ve Vahrenhold, J., 2009, On the Complexity of Computing the Hypervolume Indicator, *IEEE Transactions on Evolutionary Computation*, 13 (5), 1075-1082.
- Chan, T. S., Jha, A. ve Tiwari, M. K., 2015, Bi-Objective Optimization of Three Echelon Supply Chain involving Truck Selection and Loading using NSGA-II with Heuristics algorithm, *Applied Soft Computing*, 38.
- Chan, F. T. S., Jha, A. ve Tiwari, M. K., 2016, Bi-objective optimization of three echelon supply chain involving truck selection and loading using NSGA-II with heuristics algorithm, *Applied Soft Computing*, 38, 978-987.
- Coello Coello, C. A., Van Veldhuizen, D. A. ve Lamont, G. B., 2007, Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition.
- Coello Coello, C. A., 2000, An updated survey of GA-based multiobjective optimization, *ACM Computing Surveys*, 32, 109-143.
- Coello Coello, C. A., 2002, Evolutionary Multi-Objective Optimization: Evolutionary Optimization, Springer US, Boston, MA, 117-146.
- Corne, D., Jerram N. R., Knowles, J. D. ve Oates, M. J., 2001, PESA-II: Region-based selection in evolutionary multiobjective optimization, *Proc. 6th Int. Conf. Pparallel Prob. Solving from Nature PPSN-VI*.
- Çınar, A. C., Iscan, H. ve Kiran, M. S., 2018, Tree-Seed Algorithm for Large-Scale Binary Optimization, *KnE Social Sciences*, 3 (1), 48.
- Çınar A. C. ve Kiran, M. S., 2016, A Parallel Version of Tree-Seed Algorithm (TSA) within CUDA Platform, *International Scientific Conference on Applied Sciences, Antalya*.
- Çınar A. C. ve Kiran, M. S., 2017, Boundary conditions in Tree-Seed Algorithm: Analysis of the success of search space limitation techniques in Tree-Seed Algorithm., *Computer Science and Engineering (UBMK), 2017 International Conference on., IEEE*.
- Çınar A. C. ve Kiran, M. S., 2018, A Parallel Implementation of Tree-Seed Algorithm on CUD-supported graphical processing unit, *Journal of the Faculty of Engineering and Architecture of Gazi University, Ankara*.
- Çınar, A. C. ve Kiran, M. S., 2018, Similarity and Logic Gate-Based Tree-Seed Algorithms for Binary Optimization, *Computers & Industrial Engineering*, 115, 631-646.
- Çınar, A. C., Korkmaz, S. ve Kiran, M. S., 2018, A Binary Hybrid Approach Based on Tree Seed Algorithm for Solving 0 1 Knapsack Problems, *7th International Conference on Advanced Technologies (ICAT'18), Antalya, Turkiye*.

- Deb, K., Agrawal, S., Pratap, A. ve Meyarivan, T., 2000, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, *Parallel Problem Solving from Nature PPSN VI*, Berlin, Heidelberg, 849-858.
- Deb, K., Pratap, A., Agarwal, S. ve Meyarivan, T., 2002, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6 (2), 182-197.
- Deb, K., Thiele, L., Laumanns, M. ve Zitzler, E., 2001, Scalable Test Problems for Evolutionary Multi-Objective Optimization, *India: Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol.*
- Deb, K., Thiele, L., Laumanns, M. ve Zitzler, E., 2005, Scalable Test Problems for Evolutionary Multiobjective Optimization, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, Springer London, London, 105-145.
- Deb, K., 1999, Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design, *Evolutionary Algorithms in Engineering and Computer Science*, 8.
- Deb, K., 2001, Multiobjective Optimization Using Evolutionary Algorithms, Wiley, New York.
- Deb, K., 2011, Multi-Objective Optimization Using Evolutionary, *Department of Mechanical Engineering Indian Institute of Technology Kanpur Kanpur*.
- Durillo, J., Nebro, A., Luna, F., Chicano F. ve Moloni, G., 2007, Optimal antenna placement using a new multi-objective chc algorithm, 876--883.
- Durillo, J., Nebro, A., Luna, F. ve Alba, E., 2008, Solving Three-Objective Optimization Problems Using a New Hybrid Cellular Genetic Algorithm, 661-670
- Durillo, J., Nebro, A. ve Alba, E., 2010, The jMetal framework for multi-objective optimization: Design and architecture, 1-8
- Durillo, J. ve Nebro, A., 2011, jMetal: A Java framework for multi-objective optimization, *Advances in Engineering Software*, 42 (10), 760-771.
- Durmaz, E. D. ve Şahin, R., 2017, Çok Amaçlı Tek Sıra Tesis Düzenleme Probleminin Çözümü İçin Nsga-Iı ve Hedef Programlama Yaklaşımı, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 32 (3).
- Emel, G. G. ve Taşkın, Ç., 2002, Genetik Olgaritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*.
- Erdoğan, P., 2016, Doğadan Esinlenen Optimizasyon Algoritmaları ve Optimizasyon Algoritmalarının Optimizasyonu, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 4.
- Ergül, E. ve Eminoglu, I., 2014, DOPGA: A new fitness assignment scheme for multi-objective evolutionary algorithms, *International Journal of Systems Science*, 45, 407-426.
- Ergül, E., 2010, Çok amaçlı Genetik Algoritmalar: Temelleri ve Uygulamaları, *Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Doktora Tezi*.
- Ergül, E., 2015, Çok Amaçlı Genetik Algoritma Yöntemlerinin Başarımının Belirlenmesi için İki Yeni Ölçüt Önerisi, *Journal of Advanced Technology Sciences*.
- Fallah-Mehdipour, E., Bozorg-Haddad, O., mohammad rezapour tabari, M. ve Mariño, M. A., 2012, Extraction of decision alternatives in construction management projects: Application and adaptation of NSGA-II and MOPSO, *Expert Syst. Appl.*, 39, 2794-2803.
- Fonseca, C. M., Knowles, J., Thiele, L. ve Zitzler, E., 2006, A tutorial on the performance assesment of stochastic multiobjective optimizers, *Tik Report*, 216.

- Fonseca, C. M., Paquete, L. ve Lopez-Ibanez, M., 2006, An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator, *2006 IEEE International Conference on Evolutionary Computation*, 1157-1163.
- Fonseca, C. M. ve Fleming, P. J., 1998, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28 (1), 26-37.
- Fonseca, C. M. ve Fleming, P. J., 1999, Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization, *the fifth Intl conference on Genetic Algorithms*, 93.
- Goldberg, D. E. ve Holland, J. H., 1988, Genetic Algorithms and Machine Learning, *Machine Learning*, 3 (2), 95-99.
- Güngör, I., Emiroğlu, B. G., Çınar, A. C. ve Kıran, M. S., 2019, Integration Search Strategies in Tree Seed Algorithm for High Dimension Function Optimization, *International Journal of Machine Learning and Cybernetics*.
- Gustavsson, P. ve Syberfeldt, A., 2017, A New Algorithm Using the Non-Dominated Tree to Improve Non-Dominated Sorting, *Evolutionary Computation*, 26 (1), 89-116.
- Haklı, H., Korkmaz, S. ve Kıran, M. S., 2018, An improved version of tree seed algorithm for continuous optimization, *7th International Conference on Advanced Technologies (ICAT'18)*, 47.
- Haklı, H., 2013 Sürekli fonksiyonların optimizasyonu için doğa esinli algoritmaların geliştirilmesi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Huband, S., Hingston, P., Barone, L. ve While, L., 2006, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Transactions on Evolutionary Computation*, 10 (5), 477-506.
- Ishibuchi, H., Murata, T. ve Türkşen, I. B., 1997, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, *Fuzzy Sets and Systems*, 89 (2), 135-150.
- Jiang, J., Jiang, S., Meng, X. ve Qiu, C., 2019, EST-TSA: An effective search tendency based to tree seed algorithm, *Physica A: Statistical Mechanics and its Applications*, 534, 122323.
- Kaya, E., Uymaz, O., Korkmaz, S., Sırınkaya, E. ve Kıran, M. S., 2018, Performance analysis of Galactic Swarm Optimization with Tree Seed Algorithm, *International Conference on Advanced Technologies, Computer Engineering and Science*.
- Kaya, M. ve Güngör, S., 2007, Çok-Amaçlı Genetik Algoritma Kullanan Bulanık Sınıflandırıcı Etmenlerle Hastalık Teşhisi, *Elektrik Elektronik Bilgisayar Mühendisliği 12. Ulusal Kongresi ve Fuarı*.
- Kaya, S ve Fırlı, N., 2016, Çok Amaçlı Optimizasyon Problemlerinde Pareto Optimal Kullanımı, *Socia Sciences Research Journal*, 5.
- Kıran, M. S. ve Çınar, A. C., 2018, Ağaç-tohum algoritmasının CUDA destekli grafik işlem birimi üzerinde paralel uygulaması, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 2018 (2018).
- Kıran, M. S., 2015, TSA: Tree-seed algorithm for continuous optimization, *Expert Systems with Applications*, 42 (19), 6686-6698.
- Kıran, M. S., 2016, Tree-Seed Algorithm for BinaryOptimization, *International Scientific Conference on Applied Sciences*, Cham, 189-197.
- Kıran, M. S., 2016, An Implementation of Tree-Seed Algorithm (TSA) for Constrained Optimization, *Intelligent and Evolutionary Systems*, Antalya, Turkey.

- Kıran, M. S., 2016, Withering process for tree-seed algorithm, *8th International Conference on Advances in Information Technology, Macau, China*.
- Kıran, M. S., 2017, Withering process for tree-seed algorithm, *Procedia Computer Science*, 111, 46-51.
- Knowles, J. ve Corne, D., 1999, The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1, 98-105.
- Konak, A., Coit, D. W. ve Smith, A. E., 2006, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering & System Safety*, 91 (9), 992-1007.
- Korkmaz, S., Çınar, A. C., Seyfi, G., Kıran, M. S., 2017, A Discrete Variant of Tree-Seed Algorithm., *International Conference on Engineering Technologies - ICENTE*.
- Köse, U., Temmuz-2017, Yapay Zeka Tabanlı Optimizasyon Algoritmaları Geliştirilmesi., *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Kremmel, T., Kubalik, J. ve Biffel, S., 2010, Multiobjective evolutionary algorithm for software project portfolio optimization.
- Kukkonen, S. ve Lampinen, J., 2005, GDE3: The third evolution step of generalized differential evolution.
- Lamont, G. B., van Veldhuizen, D. A., 1999, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, *Air Force Institute of Technology*.
- Li, H. ve Zhang, Q., 2009, Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II, *IEEE Transactions on Evolutionary Computation*, 13 (2), 284-302.
- Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello Coello, A. C., Luna, F. ve Alba, E., 2009, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*, 66-73.
- Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. ve Beham, A., 2008, AbYSS: Adapting Scatter Search to Multiobjective Optimization, *Evolutionary Computation, IEEE Transactions on*, 12, 439-457.
- Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T. ve Murata, T., 2007, Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings.
- Özsağlam, M. Y. ve Cunkaş, M., 2008, Optimizasyon Problemlerinin Çözümü için Parçaçık Sürü Optimizasyonu Algoritması, *Journal of Polytechnic*.
- Reyes Sierra, M. ve Coello Coello, A. C., 2005, Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ -Dominance.
- Riquelme-Granada, N., Von Lüken, C. ve Baran, B., 2015, Performance metrics in multi-objective optimization.
- Roshanian, J., Bataleblu, A., Farghadani, M. ve Ebrahimi, B., 2017, Multi-Objective Multidisciplinary Design Optimization of a General Aviation Aircraft, *Modares Mechanical Engineering*, 17, 199-210.
- Ruiz, A. B., Saborido Infantes, R. ve Luque, M., 2014, A preference-based evolutionary algorithm for multiobjective optimization: The weighting achievement scalarizing function genetic algorithm, *Journal of Global Optimization*.
- Saborido Infantes, R., Ruiz, A. B. ve Luque, M., 2016, Global WASF-GA: An Evolutionary Algorithm in Multiobjective Optimization to Approximate the Whole Pareto Optimal Front, *Evolutionary Computation*, 25.
- Sağ, T., 2008, Çok Kriterli Optimizasyon için Genetik Algoritma Yaklaşımları.
- Sağ, T. ve Cunkaş, M., 2009, Çok Amaçlı Genetik Algoritmalar İçin Bir Çevrimdışı Performans Değerlendirmesi, *Konya*.

- Seifollahi-Aghmiuni, S. ve Bozorg Haddad, O., 2018, Multi Objective Optimization with a New Evolutionary Algorithm, *Water Resources Management*, 32 (12), 4013-4030.
- Srinivas, N. ve Deb, K., 1994, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, 2 (3), 221-248.
- Srinivas, N. ve Deb, K., 2000, Multiobjective Function Optimization Using Nondominated Sorting Genetic Algorithms, 2.
- Steven, G., 2002, Evolutionary algorithms for single and multicriteria design optimization. A. Osyczka. Springer Verlag, Berlin, 2002, ISBN 3-7908-1418-01, *Structural and Multidisciplinary Optimization - Struct Multidiscip Optim*, 24, 88-89.
- Türkay, M., 2019, Optimizasyon Modelleri ve Çözüm Metodları, *Koç Üniversitesi, Endüstri Mühendisliği Bölümü, İstanbul*.
- Van Veldhuizen, D. A. ve Lamont, G. B., 2000, Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art, *Evolutionary Computation*, 8, 125-147.
- Weise, T., 2009, Global Optimization Algorithm: Theory and Application, p.
- Winston, W. L., 2003, Operations Research: Applications and Algorithms, *International Thomson Publishing, Belmont, CA*.
- xu, Z., Ming, X. G., Zheng, M., Li, M., He, L. ve Song, W., 2015, Cross-trained workers scheduling for field service using improved NSGA-II, *International Journal of Production Research*, 53.
- Zhang, H., Wu, J., Sun, C., Zhong, M. ve Yang, R., 2018, A Multi-objective Particle Swarm Optimizer Based on Simulated Annealing and Decomposition, *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 262-273.
- Zitzler, E., Laumanns, M. ve Thiele, L., 2001, SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. ve Fonseca, V. G. d., 2003, Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Transactions on Evolutionary Computation*, 7 (2), 117-132.
- Zitzler, E. ve Künzli, S., 2004, Indicator-Based Selection in Multiobjective Search.
- Zitzler, E. ve Thiele, L., 1999, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, 3 (4), 257-271.
- Zitzler, E., 1999, Evolutionary Algorithms for Multiobjective Optimization: Methods and Application, *Swiss Federal Institute of Technology, Switzerland*.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Gül ÖZCAN
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Ankara – Polatlı 03.05.1994
Telefon : (543) 208 52 55
Faks :
E-Posta : ozcanggull@gmail.com

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: Polatlı Lisesi	Polatlı	Ankara	2011
Üniversite	: Selçuk Üni.	Selçuklu	Konya	2016
Yüksek Lisans	:			
Doktora	:			

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2017-2018	Emse Mühendislik	Bilgisayar Müh.
2018-2019	RKSOFT Bilişim Teknoloji AŞ.	Bilgisayar Müh.

UZMANLIK ALANI

YABANCI DİLLER

İngilizce – B2 Level (British Culture Certificate)

KONFERANSLAR

ICATCES 2018 – MOTSA: A multiobjective variant of Tree Seed Algorithm