



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**KISITLI VE AYRIK OPTİMİZASYON
PROBLEMLERİNİN ÇÖZÜMÜ İÇİN
AĞAÇ-TOHUM ALGORİTMASININ
UYARLANMASI VE ANALİZİ**

Ahmet Cevahir ÇINAR

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Mart-2020
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ahmet Cevahir ÇINAR tarafından hazırlanan “**Kısıtlı ve Ayrık Optimizasyon Problemlerinin Çözümü için Ağaç-Tohum Algoritmasının Uyarlanması ve Analizi**” adlı tez çalışması 20/03/2020 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Erkan ÜLKER

Danışman

Doç. Dr. Mustafa Servet KIRAN

Üye

Doç. Dr. Oğuz FINDIK

Üye

Doç. Dr. Mesut GÜNDÜZ

Üye

Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN

İmza











Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

Bu tez çalışması Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 18201015 nolu proje ile desteklenmiştir.

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 material and results that are not original to this work.

İmza 

Ahmet Cevahir ÇINAR

Tarih: 20/03/2020

ÖZET

DOKTORA TEZİ

KISITLI VE AYRIK OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜ İÇİN AĞAÇ-TOHUM ALGORİTMASININ UYARLANMASI VE ANALİZİ

Ahmet Cevahir ÇINAR

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Doç. Dr. Mustafa Servet KIRAN

2020, 150 Sayfa

Jüri

**Prof. Dr. Erkan ÜLKER
Doç. Dr. Mustafa Servet KIRAN
Doç. Dr. Mesut GÜNDÜZ
Doç. Dr. Oğuz FINDIK
Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN**

Ağaç-tohum algoritması (TSA), 2015 yılında sürekli kısıtsız optimizasyon problemlerini çözmek için önerilmiş olan metasezgisel bir arama algoritmasıdır. Bu tez çalışmasında küçük boyutlu sürekli kısıtsız optimizasyon problemlerine kaliteli sonuçlar üreten TSA'nın sürekli kısıtlı, ayrık ikili ve ayrık tam sayılı optimizasyon problemlerine çözüm üretecek şekilde iyileştirilmesi ve geliştirilmesi ile ilgili çalışmalar yapılmıştır. Sürekli kısıtlı optimizasyon problemlerini çözmek için literatürde yer alan kural tabanlı bir seçim stratejisi TSA'ya entegre edilerek kısıtlı TSA (CTSA) önerilmiştir. CTSA ile hem sayısal kıyas fonksiyonları, hem de mühendislik tasarım problemleri çözülmüş ve CTSA, literatürdeki güncel algoritmalarla karşılaştırılmıştır. Ayrık ikili optimizasyon problemlerini çözmek için TSA ikili çözüm uzayında çalışacak şekilde yapılandırılmış, yeni çözümler üretmek içinse mantıksal kapı kullanan (LogicTSA) ve benzerlik tabanlı (SimTSA) yeni algoritmalar önerilmiştir. Performansı daha üst seviyeye çıkarmak için bu iki yaklaşım hibritlenerek SimLogicTSA isimli yeni bir yaklaşım önerilmiş ve mevcut algoritmalarla kıyaslandığında daha başarılı sonuçlar elde edildiği görülmüştür. Ayrık tam sayılı optimizasyon problemlerinin çözümü için TSA permütasyon kodlu gösterim (DTSA) ile çalışacak hale getirilmiştir. DTSA, takas, öteleme ve simetri gibi komşuluk operatörleri ile yeni çözümler üretmektedir. En yakın komşu sezgiseli ve 2-opt yerel arama algoritması ile DTSA'nın elde ettiği çözümlerin kalitesi artırılmaya çalışılmıştır. Bu tez kapsamında TSA tabanlı önerilen yöntemlerin literatürdeki benzerlerinden daha kaliteli veya rekabetçi çözümler ürettiği gösterilmiştir.

Anahtar Kelimeler: ayrık optimizasyon, ağaç-tohum algoritması, evrimsel hesaplama, ikili optimizasyon, optimizasyon, sürü zekası

ABSTRACT

PhD THESIS

ADAPTATION AND ANALYSIS OF TREE-SEED ALGORITHM FOR SOLVING CONSTRAINED AND DISCRETE OPTIMIZATION PROBLEMS

Ahmet Cevahir ÇINAR

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Assoc. Prof. Dr. Mustafa Servet KIRAN

2020, 150 Pages

Jury

Prof. Dr. Erkan ÜLKER
Assoc. Prof. Dr. Mustafa Servet KIRAN
Assoc. Prof. Dr. Mesut GÜNDÜZ
Assoc. Prof. Dr. Oğuz FINDIK
Asst. Prof. Dr. Mehmet Akif ŞAHMAN

Tree-seed algorithm (TSA) is a metaheuristic search algorithm proposed in 2015 to solve continuous unconstrained optimization problems. In this thesis, the studies have been carried out on the improvement and development of TSA, which produces high-quality results on small-sized continuous unconstrained optimization problems, in a way to provide solutions to continuous constrained, discrete binary and discrete integer optimization problems. A rule-based selection strategy which is in the literature has been integrated with TSA, and a constrained TSA (CTSA) has been proposed in order to solve continuous constrained optimization problems. Both numerical benchmark functions and engineering design problems were solved by CTSA, and it has been compared with the state-of-art algorithms in the literature. To solve discrete binary optimization problems, TSA is re-structured to work on binary search space, and logic-gate based TSA (LogicTSA) and similarity-based TSA (SimTSA) are proposed to generate new binary solutions. In order to obtain better results, these two approaches have been hybridized and a new approach, called SimLogicTSA, has been proposed, and more successful results have been obtained when compared to existing algorithms. In order to solve discrete integer optimization problems, TSA has been modified for working with permutation coded representation (DTSA). DTSA produces new solutions with neighborhood operators such as swap, shift, and symmetry. The quality of the solutions obtained by DTSA has been improved with the nearest neighbor heuristic and 2-opt local search approach. It has been shown that all proposed methods in this thesis content produce better quality or more competitive solutions than their counterparts in the literature.

Keywords: binary optimization, discrete optimization, evolutionary computation, optimization, swarm intelligence, tree-seed algorithm,

ÖNSÖZ

Polinomsal zamanda en iyi çözümü elde edilemeyen optimizasyon problemlerine makul zamanda kabul edilebilir çözümler üretmek için kırk yıldan uzun bir süredir geliştirilen ve geliştirilmeye devam eden metasezgisel algoritmalara olan ilgi, günümüzde verinin çoğalmasına paralel olarak büyüyen boyutlardan dolayı daha da artmıştır. Bu kapsamdaki problemlere çözüm üretmek için önerilmiş olan Ağaç-Tohum Algoritması'nın sürekli kısıtsız optimizasyon problemlerini çözmekte gösterdiği başarıyı, bu tez çalışması kapsamında sürekli kısıtlı, ayrık ikili ve ayrık tam sayılı optimizasyon problemlerini çözecek şekilde geliştirip literatüre farklı yaklaşımlar kazandırılması bu tez kapsamında amaçlanmıştır.

Bu süreçte büyük emeği olan ve tez çalışmasına fikirleriyle yön veren danışman hocam Doç. Dr. Mustafa Servet KIRAN'a, yapıcı eleştirileriyle bana yol gösteren tez izleme kurulu üyeleri Doç. Dr. Oğuz FINDIK ve Dr. Öğretim Üyesi Mehmet Akif ŞAHMAN'a;

Tez çalışmama yaptıkları katkı ve incelemeleri için tez savunma sınavı jüri üyeleri olan Prof. Dr. Erkan ÜLKER ve Doç. Dr. Mesut GÜNDÜZ'e;

Tez çalışmam sırasında yapmış olduğum akademik çalışmalara yaptıkları katkılardan dolayı Dr. Öğretim Üyesi Ahmet BABALIK ve Dr. Öğretim Üyesi Sedat KORKMAZ'a;

Bu araştırmaya 18201015 numaralı proje ile maddi destek sağlayan Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'ne;

2006 yılından beri çalıştığım Konya Meydan Meteoroloji Müdürlüğü'ndeki mesai arkadaşlarıma;

Gösterdikleri sabır, hoşgörü, maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Ahmet Cevahir ÇINAR
KONYA-2020

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
KISALTMALAR	ix
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	6
2.1. Sürekli Kısıtlı Optimizasyon Algoritmaları.....	6
2.2. İkili Optimizasyon Algoritmaları.....	9
2.2.1. Parçacık sürü optimizasyonu tabanlı ikili optimizasyon algoritmaları.....	9
2.2.2. Yapay arı kolonisi tabanlı ikili optimizasyon algoritmaları	11
2.2.3. Farksal gelişim (DE) tabanlı ikili optimizasyon algoritmaları	12
2.2.4. Diğer ikili optimizasyon algoritmaları.....	13
2.3. Ayırık Tam Sayılı Optimizasyon Algoritmaları	15
2.4. Ağaç-Tohum Algoritmasını Kullanan Diğer Çalışmalar.....	22
3. MATERYAL VE YÖNTEM	31
3.1. Ağaç-Tohum Algoritması	31
3.2. Kısıtlı Ağaç-Tohum Algoritması	39
3.2.1. Kural tabanlı kısıtlı TSA.....	39
3.3. İkili Ağaç-Tohum Algoritmaları.....	40
3.3.1. Benzerlik tabanlı ikili TSA	41
3.3.2. Mantıksal kapı tabanlı ikili TSA.....	44
3.3.3. Benzerlik ve mantıksal kapı tabanlı ikili TSA.....	45
3.4. Ayırık Tam sayılı Ağaç-Tohum Algoritması.....	46
3.4.1. Komşuluk operatörleri	47
3.4.2. DTSA'nın algoritmik çerçevesi	48
3.5. Optimizasyon Problemleri	49
3.5.1. Sürekli kısıtlı nümerik kıyas fonksiyonları.....	49
3.5.2. Sürekli kısıtlı mühendislik tasarım problemleri.....	50
3.5.3. Kapasitesiz tesis yerleşim problemi.....	50
3.5.4. Ayırık tam sayılı gezgin satıcı problemi	52
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	54
4.1. CTSA Deneysel Sonuçları ve Karşılaştırmaları	54
4.1.1. Standart kısıtlı kıyas fonksiyonları ile yapılan deneylerin sonuçları	54
4.1.2. Mühendislik tasarım problemleri ile yapılan deneylerin sonuçları	73
4.2. İkili TSA'ların Deneysel Sonuçları ve Karşılaştırmaları.....	76

4.2.1. LogicTSA deneysel sonuçları ve karşılaştırmaları	76
4.2.2. SimTSA deneysel sonuçları ve karşılaştırmaları	86
4.2.3. SimLogicTSA deneysel sonuçları ve karşılaştırmaları.....	94
4.2.4. SimLogicTSA'nın LogicTSA ve SimTSA ile karşılaştırmaları	99
4.2.5. SimLogicTSA'nın diğer algoritmalarla karşılaştırılması	104
4.3. DTSA deneysel sonuçları ve karşılaştırmaları.....	112
4.3.1. TSP için komşuluk operatörlerinin karşılaştırılması	112
4.3.2. DTSA için N ve ST parametrelerinin belirlenmesi	113
4.3.3. DTSA'nın SA, ACO ve STA ile karşılaştırılması	114
4.3.4. DTSA'nın SA ve DSTA varyantları ile karşılaştırılması	115
4.3.5. DTSA'nın ABC varyantları ile karşılaştırılması	116
4.3.6. DTSA'nın ACO, ABC, HA ve DPSO ile karşılaştırılması.....	122
4.3.7. DTSA'nın ACO, PSO, GA ve BH ile karşılaştırılması	124
5. SONUÇLAR VE ÖNERİLER	126
5.1 Sonuçlar	126
5.2 Öneriler ve Gelecekteki Çalışmalar	127
KAYNAKLAR	128
EKLER	141

KISALTMALAR

AAA	: Yapay alg algoritması (artificial algae algorithm)
ABC	: Yapay arı kolonisi (artificial bee colony)
ACS	: Karınca kolonisi sistemi (ant colony system)
ACO	: Karınca kolonisi optimizasyonu (ant colony optimization)
AFSA	: Yapay balık sürüsü algoritması (artificial fish swarm algorithm)
AMABC	: Açık modülasyonlu ABC (angle-modulated ABC)
AMDE	: Açık modülasyonlu DE (angle-modulated DE)
AMPSON	: Açık modülasyonlu PSO (angle-modulated PSO)
APX	: Alternatif pozisyon çaprazlama (alternating position crossover)
BA	: Yarasa algoritması (bat algorithm)
BAAA	: İkili AAA (binary AAA)
bAFSA	: İkili AFSA (binary AFSA)
BAS	: İkili karınca sistemi (binary ant system)
binde	: Binary DE (BDE)
binABC	: İkili ABC (binary ABC)
BitABC	: Bitsel ABC (bitwise ABC)
BGSA	: İkili GSA (binary GSA)
BH	: Kara delik (black hole)
BHTPSO	: İkili hibrit topoloji PSO (binary hybrid topology PSO)
BHTPSO-QI	: BHTPSO kuadratik enterpolasyon (BHTPSO quadratic interpolation)
BLDE	: İkili öğrenme DE (binary learning DE)
BPSO	: İkili PSO (binary PSO)
BPSOTVAC	: TVAC entegreli BPSO
CABC	: Kombinatoriyal ABC (combinatorial ABC)
CBPSO	: Kısıt temelli PSO (constrained-based PSO)
CBPSOTVAC	: TVAC entegreli CBPSO
CEC	: Evrimsel hesaplama kongresi (congress on evolutionary computation)
CMA	: Kovaryans Matris Adaptasyonu (Covariance Matrix Adaptation)
CMA-ES	: Kovaryans Matris Adaptasyonu Evrim Stratejisi
CS	: Guguk kuşu arama (cuckoo search)
CTSA	: kural tabanlı kısıtlı TSA (constrained TSA, CTSA)
CX	: Çevrim çaprazlama (cycle crossover)
DABC	: Ayrık ABC (discrete ABC)
DBHS	: Ayrık ikili HS (discrete binary HS)
DCS	: Ayrık CS (discrete CS)
DE	: Farksal Gelişim (Differential Evolution)
DEPSO	: DE entegreli PSO
DisABC	: Benzemezlik tabanlı ABC (Dissimilarity-based ABC)
DM	: Yer değiştirme mutasyonu (displacement mutation)
DPSO	: Ayrık PSO (discrete PSO)
DSTA	: Ayrık STA (discrete STA)
DTSA	: Ayrık TSA (discrete TSA)
EM	: Değişim mutasyonu (exchange mutation)
ERX	: Geliştirilmiş kenar rekombinasyon çaprazlama (improved edge recombination crossover)
ES	: Evrim Stratejisi (evolution strategy)
ESA	: Evrimsel SA (evolutionary SA)
EST-TSA	: Efektif ST TSA (effective ST TSA)
FA	: Ateşböceği algoritması (firefly algorithm)

GA	: Genetik algoritma (genetic algorithm)
GADS	: Çift dizili GA (genetic algorithm with double strings)
Gbest	: Genel en iyi çözüm (global best)
GB-ABC	: Genetik operatörlere dayalı ABC (genetic operators based ABC)
GCBPSO	: Yakınsama garantili BPSO (guaranteed convergence BPSO)
GNX	: Genelleştirilmiş N-nokta çaprazlama (generalized N-point crossover)
GOA	: Çekirge optimizasyon algoritması (grasshopper optimization algorithm)
GRNN	: Genelleştirilmiş regresyon sinir ağı (generalized regression neural network)
GSA	: Yerçekimi arama algoritması (gravitational search algorithm)
G-TSA	: Geliştirilmiş TSA
HC	: Tepe tırmanma (hill climbing)
HMMA	: Hibrit max-min karınca sistemi (hybrid max-min ant system)
HS	: Harmoni arama (harmony search)
HX	: Sezgisel çaprazlama (heuristic crossover)
IbAFSA	: Geliştirilmiş bAFSA (Improved bAFSA)
ICA	: Emperyalist rekabet algoritması (imperialist competitive algorithm)
INBPSO	: Geliştirilmiş NBPSO (Improved NBPSO)
ISM	: Ekleme mutasyonu (insertion mutation)
IVM	: Devrik mutasyon (inversion mutation)
iTSA	: Geliştirilmiş TSA (improved TSA)
KP	: Sırt çantası problemi (knapsack problem)
KNN	: K-en yakın komşu (K-nearest neighbor)
LogicTSA	: Mantıksal kapı tabanlı ikili TSA (logic-gate based TSA)
maxFEs	: maksimum fonksiyon değerlendirme sayısı (maximum function evaluation number)
MBPSO	: Değiştirilmiş BPSO (Modified BPSO)
MKP	: Çoklu KP (multiple KP)
MPX	: Maksimum koruyucu çaprazlama (maximal preservative crossover)
MTSA	: Değiştirilmiş TSA (modified TSA)
MVO	: Çoklu evren optimizasyonu (multi-verse optimizer)
N	: Popülasyon büyüklüğü (population size)
NBPSO	: Yeni BPSO (New BPSO)
NBSG	: Yeni ikili çözüm üretim algoritması (new binary solution generator)
NFL	: Bedava yemek yok (No free lunch)
NN	: Sinir ağı (neural network)
normDE	: Normalleştirilmiş DE (normalization DE)
OBX	: Sıra tabanlı çaprazlama (order based crossover)
OOTSA	: Sıralı optimizasyon TSA (ordinal optimization TSA)
OX	: Düzenli çaprazlama (order crossover)
PACO-3Opt	: Paralel kooperatif hibrit algoritması (parallel cooperative hybrid algorithm)
PBX	: Pozisyon tabanlı çaprazlama (position-based crossover)
PEO	: Fiziksel enerji optimizasyonu (physarum-energy optimization)
PESO	: Parçacık evrimsel SO (particle evolutionary SO)
PMX	: Kısmen eşlenmiş çaprazlama (partially-mapped crossover)
PRSA	: Saf rastgele arama algoritması (pure random search algorithm)
PSO	: Parçacık sürü optimizasyonu (particle swarm optimization)
PTSA	: Ceza temelli TSA (penalty based TSA)
PVDP	: Basınçlı tank tasarımı problemi (pressure vessel design problem)
RBFN	: Radyal temelli fonksiyon NN (radial basis function NN)
RI	: Rastgele ekleme (random insertion)

RIS	: Rastgele alt dizilerin eklenmesi (random insertion of subsequence)
RKCS	: Rastgele anahtar guguk kuşu arama (random key CS)
RRIS	: Rastgele alt dizilerin ters yüz edilerek eklenmesi (random reversing insertion of subsequence)
RRS	: Rastgele alt dizilerin ters yüz edilmesi (random reversing of subsequence)
RRSS	: Rastgele alt dizilerin ters yüz edilerek takas edilmesi (random reversing swap of subsequences)
RS	: Rastgele takas (random swap)
RSS	: Rastgele alt dizilerin takası (random swap of subsequences)
SA	: Benzetimli tavlama (simulated annealing)
SabDE	: Kendinden uyarlamalı ikili DE (self-adaptive binary DE)
SBHS	: Basitleştirilmiş ikili HS (Simplified binary HS)
SIM	: Basit inversiyon (simple inversion mutation)
SimLogicTSA	: Benzerlik ve mantıksal kapı tabanlı ikili TSA
SimTSA	: Benzerlik tabanlı ikili TSA (similarity based TSA)
SM	: Çırpma mutasyon (scramble mutation)
SMX	: Sıralanmış eşlenmiş çaprazlama (sorted match crossover)
SO	: Sürü optimizasyonu (swarm optimization)
SR	: Stokastik sıralama (stochastic ranking)
SSA	: Salp sürü algoritması (salp swarm algorithm)
SVM	: Destek vektör makineleri (support vector machines)
SVM-TSA	: TSA ile eğitilmiş SVM
ST	: Araştırma eğilimi (search tendency)
STA	: Durum geçiş algoritması (state transition algorithm)
STSA	: Sinüs TSA (Sine TSA)
S-bAFSA	: Basitleştirilmiş bAFSA (simplified bAFSA)
TSA	: Ağaç-tohum algoritması (tree-seed algorithm, TSA)
TSAWP	: Soldurma süreci eklenmiş TSA (TSA with withering process)
TS	: Tabu arama (tabu search)
tsABC	: TSA aramalı ABC
tsABC-GRNN	: tsABC ile eğitilmiş GRNN
TSP	: Gezgin satıcı problemi (traveling salesman problem)
TVAC	: Zamanla değişen hızlanma katsayıları (time varying acceleration coefficients)
UC	: Birim yüklenme (unit commitment)
UFLP	: Kapasitesiz tesis yerleşim problemi (uncapacitated facility location problem)
VRX	: Oylama rekombinasyon çaprazlama (voting recombination crossover)
VTSA-CA	: Rekabetçi yığılma ile VTSA (competitive agglomeration VTSA)
VTSA	: Değişken uzunluklu TSA (variable-length TSA)
qABC	: Hızlı ABC (quick ABC)
qCABC	: Hızlı kombinatoriyal ABC (quick combinatorial ABC)
WBDP	: Kaynaklı giriş tasarım problemi (welded beam design problem)
WP	: Soldurma süreci (withering process)

1. GİRİŞ

Optimizasyon hayatın her alanında, minimum maliyet ile maksimum kazancı hedefleyen bütün süreçlerde yer almaktadır. Yaşam, yegane amacın hayatta kalmak olduğu çok karmaşık bir optimizasyon problemi olarak düşünülebilir. Mühendislikte birçok gerçek dünya problemi optimizasyon problemi olarak modellenmekte ve çeşitli tekniklerle çözülmektedir. Optimizasyon problemlerini çözmek için klasik deterministik matematiksel yöntemler (doğrusal programlama, doğrusal olmayan programlama) kullanılmaktayken, bu yöntemlerin makul zamanda çözüm üretememeleri operasyonel kullanım için uygun olmadığından yeni çözümler aranmaya başlanmıştır. İkinci dünya savaşı sırasında Alan Turing'in Almanların Enigma makinesinin şifrelerini çözmesinden sonra yöneylem araştırması (operation research) araştırma alanı gündeme gelmiş, savunma temelli başlayan bu süreç endüstriyel alana da kayarak gittikçe popüler hale gelmiştir.

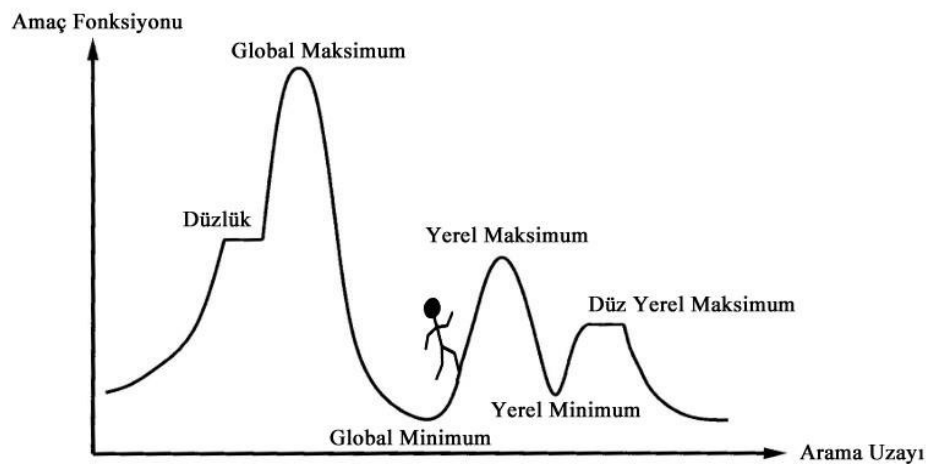
Polinomsal zamanda çözüm üretilmeyen (NP-hard) kesme ve paketleme problemleri, tesis yerleşim düzenlemesi, kaynak tahsisi problemi, parti büyüklüğü belirleme, ürün karması belirleme, montaj hattı dengeleme ve çizelgeleme, gezgin satıcı problemi, araç rotalama, havayolu filosu çizelgeleme, akış tipi çizelgeleme, hat dengeleme, yükleme ve paketleme, ekonometrik talep analizi, personel planlama, ağ saldırı tespit sistemi başta olmak üzere çözülmesi ekonomik anlamda önemli olan minimum maliyet ile maksimum kazancı hedefleyen problemlerin çözümü için deterministik matematiksel yöntemlerin esnek olmaması ve uygulanmasının zor olması bilim insanlarını yeni arayışlara itmiş, Holland (1975)'in genetik algoritma (GA) fikrini ortaya atmasıyla evrimsel hesaplama bilim insanlarının dikkatini çekmeye başlamıştır. Sezgisel yöntemler optimum çözümü garanti etmese dahi makul zamanlarda makul çözümler ürettiğinden ilgi görmüş ve her geçen gün artan bir ivmeyle hem araştırmacıların hem de uygulayıcıların ilgisini çekmeye devam etmektedir. 1975-2020 yılları arasında 300'den fazla metasezgisel algoritma çeşitli optimizasyon problemlerini çözmek için önerilmiştir.

Evrimsel sürecini taklit eden GA'dan önce evrim stratejisi (evolution strategy, ES) ismiyle anılan bir arama stratejisi Rechenberg (1973) tarafından önerilmiş, bu yöntem kovaryans matris adaptasyonu (covariance matrix adaptation, CMA) ile entegre edilerek CMA-ES (Hansen ve Ostermeier, 2001) isimli bir evrimsel hesaplama algoritmasının

ortaya çıkmasına vesile olmuştur. 1997 yılında ise Storn ve Price (Storn ve Price, 1997) tarafından farksal gelişim (differential evolution, DE) algoritması önerilmiştir.

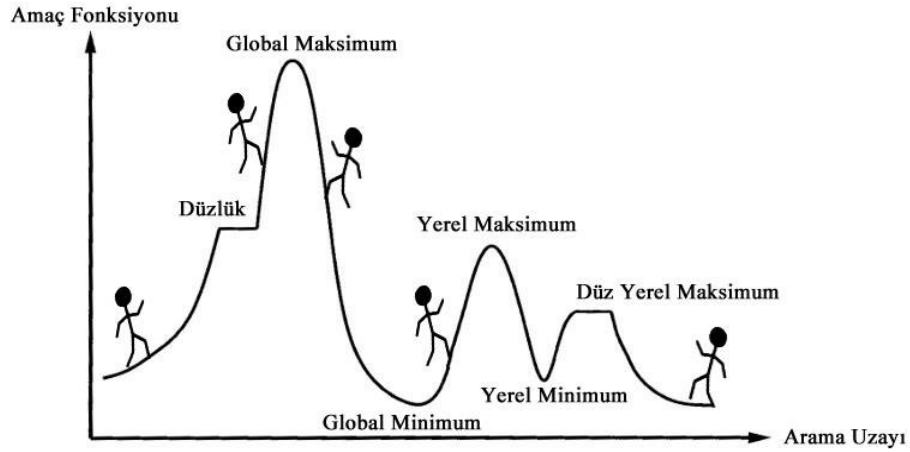
1995 yılında Eberhart ve Kennedy (1995) tarafından parçacık sürü optimizasyonu (particle swarm optimization, PSO) algoritmasının önerilmesiyle kuş ve balık gibi sürülerin yiyecek arama ve yaşamsal davranışlarını taklit eden popülasyon tabanlı sürü zekası algoritmalar dönemi başlamıştır. 1995 yılından bugüne doğa esinli optimizasyon algoritması geliştirme süreci hızlanmış çeşitli hayvan, bitki ve diğer varlıkların davranışları incelenerek bunlar optimizasyon metodu olarak modellenmeye başlanmıştır. Günümüzde insan ve topluluk davranışlarından esinlenen algoritmalar da bulunmaktadır.

Sezgisel araştırmayı Şekil 1.1'deki görsel ile birlikte tepe tırmanma (hill climbing, HC) yerel arama algoritması ekseninde düşünecek olursak verilen kararlar ile tepenin zirvesinin bulunması mümkün olabileceği gibi belki de yerel minimum noktaya takılı kalınarak ilerleme sağlanamayacaktır.



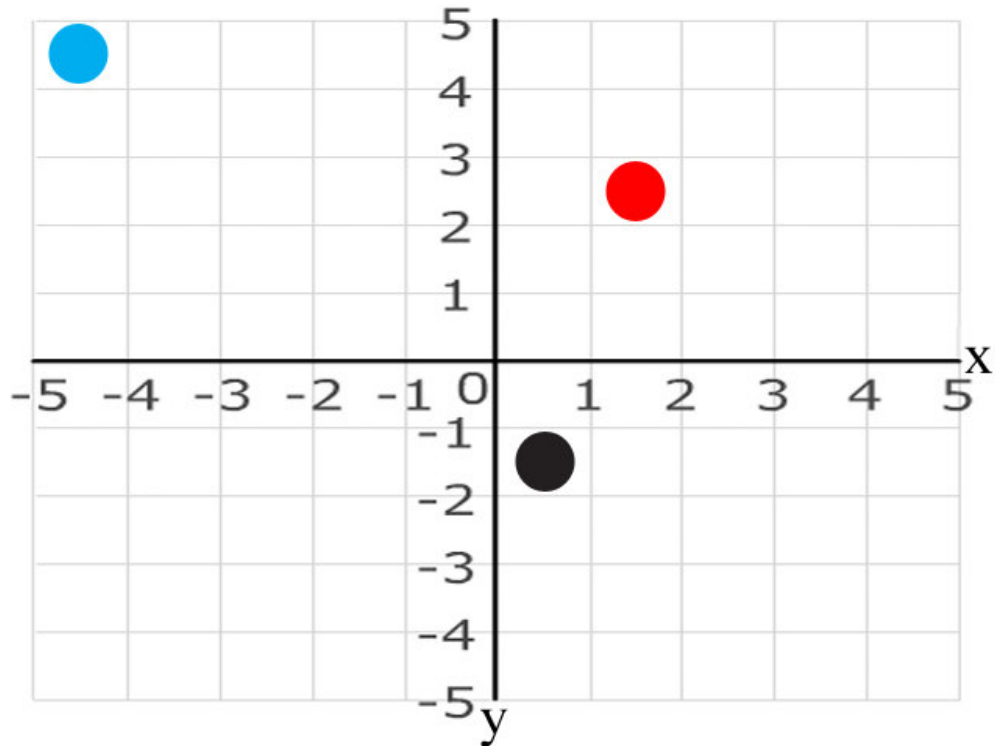
Şekil 1.1. Optimizasyon problemlerindeki temel kavramlar

Örneğin mevcut konumda bulunan bir dağcı, eğim giderek arttığından dolayı yukarıya doğru ilerlemeyi seçecek fakat bu onu yerel maksimum noktaya ulaştıracaktır. O yüzden bu tarz arama algoritmalarında başlangıç noktası sonuca direkt etki etmektedir. Bu dezavantajı ortadan kaldırmak için tek noktadan başlayan arama stratejileri yerine çok noktadan başlayan arama stratejileri (Şekil 1.2) önerilmiştir. Bu iyileştirme ile elde edilen sonuçların kalitesi artmış, bu noktalar bir popülasyon olarak adlandırılmış, popülasyon içerisindeki bireylerin birbirleriyle haberleştirilmeleri sonucunda ise daha kaliteli çözümler üretilmiştir.



Şekil 1.2. Arama uzayına yayılmış çok noktalı arama

Şekil 1.3'teki 2 boyutlu arama uzayında bulunan 3 bireyin de amacı (0,0) noktasına ulaşmaktır. Bu aşamada amaca en yakın siyah nokta iken, en uzak mavi noktadır. Popülasyon tabanlı algoritmalar, hem popülasyondaki en iyi birey ile hem de diğer bireyler ile bilgi paylaşımı (konum) yaparak optimum noktaya daha hızlı yakınsama yapmak için yardımlaşmaktadırlar.



Şekil 1.3. 2 boyutlu arama uzayına yayılmış 3 bireyin durumu

Burada ortaya çıkan problem optimizasyon probleminin sahip olduğu yerel optimum noktaların bu adımların başarısını olumsuz yönde etkilemesidir. Bu dezavantajı

kaldırmak için popülasyona yeni rastgele bireyler eklemek, popülasyonun bir kısmını karışık olarak yok etmek gibi çözümler üretilmiştir.

Arama uzayına rastgele dağılarak başlatılan araştırma sürecinin belirli/makul bir zamanda optimum veya optimuma yakın sonuç üretmesi beklenmektedir. Bu yüzden araştırma süresince atılacak adımların büyüklüğü ile boşa adım atmamak en önemli amaçtır. Bunu kontrol etmek için algoritmalar keşif (exploration) ve faydalanma (exploitation) şeklinde iki ana özelliği barındırır. Optimum çözüm bilinmediğinden, araştırma sürecinin sonuna kadar bu ikisi arasında kurulacak denge, iyi sonuçlar üreteceğinden dolayı popülasyon çeşitliliğini (population diversity) kontrol etmek algoritmaların en önemli yönlerinden birini oluşturmaktadır. Metasezgisel algoritmalar, çözüm için problem bilgisini kullanan sezgisel algoritmaları ek mekanizmalarla bir adım öteye taşıyarak geliştirilmiş arama stratejileri sunan yaklaşımlardır. Wolpert ve Macready (1997)'nin öne sürdüğü bedava yemek yok (no free lunch, NFL) teoreminde de ispatlandığı üzere bir algoritmanın tüm optimizasyon problemlerini başarıyla çözememesinden dolayı yeni metasezgisel algoritmalar önerilmeye devam edilmiştir.

2015 yılında bu eksikleri tespit eden Kiran (2015a), anlaşılır, kolay uygulanabilir ve keşif-faydalanma dengesinin her adımda kontrol edilerek popülasyonun ona göre yönlendirildiği ağaç-tohum algoritmasını (tree-seed algorithm, TSA) önermiştir. Doğa esinli sürü zekası tabanlı algoritmalar kategorisinde popülasyon tabanlı bir stokastik arama algoritması olan ve ağaçlar ile tohumları arasındaki ilişkiden esinlenilerek üretilen TSA, küçük boyutlu sürekli kısıtsız optimizasyon problemleri için kaliteli sonuçlar üreterek literatürde kabul görmüştür.

Bu tez çalışması kapsamında, literatürde TSA ile çözülmüş örnekleri bulunmayan çeşitli optimizasyon problemleri üzerinde TSA'nın başarısı incelenmiş, iyileştirilmiş ve geliştirilmiştir. Sürekli kısıtlı, ayrık ikili ve ayrık tam sayılı optimizasyon problemleri TSA ile çözülmüş, kural tabanlı kısıtlı TSA (constrained TSA, CTSA), benzerlik tabanlı ikili TSA (similarity based TSA, SimTSA), mantıksal kapı tabanlı ikili TSA (logic-gate based TSA, LogicTSA), benzerlik ve mantıksal kapı tabanlı ikili TSA (similarity and logic-gate based TSA, SimLogicTSA), ayrık TSA (discrete TSA, DTSA) algoritmaları önerilmiştir. Tezin organizasyonu aşağıdaki şekilde yapılmıştır.

İlk bölümde optimizasyon, evrimsel hesaplama, sürü zekası gibi temel kavramlardan bahsedilerek giriş yapılmıştır.

İkinci bölümde kaynak araştırması sunulmuş, Bölüm 2.1'de sürekli kısıtlı, bölüm 2.2'de ayrık ikili, bölüm 2.3'te ayrık tam sayılı optimizasyon problemlerinin çözümü için

kullanılan yöntemlerin kaynak araştırması verilmiş, bölüm 2.4'te ise bugüne kadar TSA ile yapılmış bütün çalışmalar kısaca anlatılmıştır.

Üçüncü bölümde TSA, CTSA, SimTSA, LogicTSA, SimLogicTSA ve DTSA yöntemleri ile bu yöntemleri test etmek için kullanılan optimizasyon problemleri olan sürekli kısıtlı nümerik kıyas fonksiyonları, sürekli kısıtlı mühendislik tasarım problemleri, UFLP ve TSP anlatılmıştır.

Dördüncü bölümde yapılan deneysel çalışmalar detaylı bir şekilde anlatılarak gerekli karşılaştırma ve tartışmalar yapılmıştır.

Beşinci bölümde ise tez kapsamında elde edilen sonuçlar açıklanmış ve gelecekteki çalışmalar için bazı öneriler sunulmuştur.



2. KAYNAK ARAŞTIRMASI

Tezin amacı TSA ile sürekli kısıtlı, ayrık ikili ve ayrık tam sayılı optimizasyon problemlerine çözüm aramak olduğundan kaynak araştırması bölümü daha faydalı ve toplu olması için dört alt bölüm olarak organize edilmiştir. Bölüm 2.1’de sürekli kısıtlı, bölüm 2.2’de ayrık ikili, bölüm 2.3’te ayrık tam sayılı optimizasyon problemlerinin çözümü için kullanılan yöntemlerin literatür özeti verilmiş, bölüm 2.4’te ise bugüne kadar TSA ile yapılmış bütün çalışmalar kısaca anlatılmıştır.

2.1. Sürekli Kısıtlı Optimizasyon Algoritmaları

Gerçek dünya problemleri genellikle kısıtlı problemlerdir. Kısıtların içerisinde uygun (feasible) çözümler üretmek sezgisel yaklaşımlar için bazen çok zor olmaktadır, bu da yüksek hesaplama maliyetine sebep olmaktadır. Makul bir zamanda sorunun çözülememesi çözücüye olan güveni azaltmaktadır. Kısıtlı optimizasyon problemlerinin çözümünde ceza fonksiyonu kullanmak en temel yaklaşımdır. Burada amaç kısıtlı problemlerin ceza fonksiyonu ile kısıtsız hale getirilmesidir. Bu şekilde kısıtsız optimizasyonda başarılı olan sezgisel algoritmaların bu tip problemleri çözmesi hedeflenmiştir. Yaygın kullanımına rağmen klasik ceza (penalty) fonksiyonları bazı güçlüklerle karşılaşmaktadır, bu yüzden evrimsel hesaplama ve sürü zekası teknikleri ile çalışabilen farklı kısıtlarla baş etme yöntemleri önerilmiştir. Bu bölümde kronolojik olarak kısıtlı optimizasyon problemlerinin çözümü için literatürde bulunan belli başlı çalışmalardan kısaca bahsedilmiştir. Kısıtlı optimizasyon problemlerinin çözümü için literatürde bir çok yaklaşım bulunmaktadır. Bu yaklaşımlar Michalewicz’in (1995) çalışmasında aşağıdaki şekilde gruplandırılmıştır:

- Ceza fonksiyonuna dayanan yöntemler
- Uygun olmayan aday çözümlerin reddedilmesine dayanan yöntemler
- Çok amaçlı optimizasyona dayalı yöntemler
- Uygun çözümlere, uygun olmayan çözümlere göre öncelik veren yöntemler
- Tamir yöntemleri
- Özel operatörlere dayalı yöntemler
- Seçime dayalı yöntemler
- Kültürel algoritmaların kullanıldığı yöntemler
- Hibrit yöntemler

Koziel ve Michalewicz (1999) çalışmalarında kısıtlarla başa çıkma yaklaşımlarını dört gruba ayırmıştır. Bu gruplandırma Michalewicz'in (1995) çalışmasındakine benzer olduğundan yeniden bahsedilmesine gerek görülmemiştir.

Runarsson ve Yao (2000)'nun çalışmalarında amaç ve ceza fonksiyonları arasındaki dengeyi stokastik olarak ayarlanması ile kısıtlı optimizasyon problemleri çözülmeye çalışılmıştır. Farklı oranlarla amaç ve ceza fonksiyonları üzerinde analizler yapılmıştır. 13 kısıtlı kıyas fonksiyonu ES ile çözümlenerek testler yapılmıştır.

Coello (2002), çalışmasında evrimsel hesaplama tekniklerinin kısıtlı optimizasyon problemlerinin çözümü için nasıl modifiye edildiği ile ilgili bir araştırma yapmıştır. Bağışıklık, karınca ve kültür temelli evrimsel hesaplama algoritmaları üzerinde ceza fonksiyonlarının etkilerinin analiz edildiği çalışmada, basit bir karşılaştırma da yapılmıştır.

Parsopoulos ve Vrahatis (2002)'in çalışmasında PSO'nun yaygın olarak kullanılan kısıtlı kıyas fonksiyonlarını çözümedeki başarısı araştırılarak ES ve GA ile karşılaştırması yapılmıştır. Çalışmada durağan olmayan çok aşamalı atama yapan ceza fonksiyonu (non stationary multi stage assignment penalty function) kullanılmıştır. Uygun olanlar tüm kısıtlara uyanlar, uygun olmayan olanlar ise en az bir tanesine veya daha fazlasına uymayanlardır. Ceza fonksiyonu büyüdükçe minimizasyon algoritmaları yerel minimuma daha çok takılmaktadır. Ceza fonksiyonu küçük olduğu durumlarda ise uygun optimum çözümleri bulmakta zorlanmaktadır. Ceza fonksiyonları iki kısımda incelenebilir: durağan (stationary) ve durağan olmayan (nonstationary). Literatür incelendiğinde durağan olmayan ceza fonksiyonlarının çözüme daha iyi ulaştığı görülecektir. Durağan olmayan ceza fonksiyonlarının değerleri dinamik olarak değişmektedir.

Zhang ve Xie (2003), çalışmalarında DE ile PSO'yu birleştirerek DEPSO algoritmasını önermişlerdir. DEPSO, Deb'in kuralları ile kısıtlarla mücadele eden bir yaklaşımdır. 11 kısıtlı kıyas fonksiyonu üzerinde yapılan testler ile önerilen algoritmanın performansı ortaya konmuştur. Elde edilen sonuçlar ES, DE ve PSO ile kıyaslanmıştır. Ayrıca mühendislik tasarım problemlerinden biri olan basınçlı tank tasarımı problemi (pressure vessel design problem, PVDP) de çözülmüştür.

Pulido ve Coello (2004)'nun çalışmasında önerilen mekanizma PSO'daki uygun alana en yakın parçanın lider olmasına dayanmaktadır. Ayrıca PSO'nun araştırma kapasitesini geliştirmek için bir de türbülans operatörü önerilmiştir. Türbülans operatörü ile yerel minimuma takılmış parçacıkların kurtarılması hedeflenmiştir. Bu mekanizma

literatürde bilinen ve kullanılan 3 kısıtın üstesinden gelme çalışmasıyla (Koziel ve Michalewicz, 1999; Runarsson ve Yao, 2000; Hamida ve Schoenauer, 2002) kıyaslandığında rekabetçi sonuçlar ürettiği görülmüştür.

Mezura-Montes ve Coello (2005)'nin çalışmasında ES, Deb'in kuralları kullanılarak kısıtlı fonksiyonları çözecek hale getirilmiştir. Önerilen yaklaşım 13 kısıtlı kıyas fonksiyonu üzerinde test edilmiştir.

Muñoz Zavala ve ark. (2005)'nin çalışmasında parçacık evrimsel sürü optimizasyonu (particle evolutionary swarm optimization, PESO) algoritması önerilmiştir. PESO, 2 farklı pertürbasyon operatörüyle (C ve M) donatılmış PSO olarak düşünülebilir. Pertürbasyon operatörleri genel manada çeşitliliği ve araştırmayı sürdürmek için kullanılmıştır. PSO, PSO-C, PSO-M ve PESO 13 kısıtlı kıyas fonksiyonu üzerinde test edilmiştir.

Cagnina ve ark. (2006)'nin çalışmasında eşitlik şeklinde olan kısıtların işlenmesinin daha zor olduğu belirtilerek eşitsizliğe çevrilmiştir. Komşu olarak anılan parçanın seçimi için bir sistem önerilmiş, yerel minimumlara takılmayı ve uzun süre gelişememe durumunu ortadan kaldırmak için dinamik bir mutasyon operatörü önerilmiştir.

Akay (2009), doktora tezinin bir bölümünde kısıtlı optimizasyon problemlerinin çözümü için ABC algoritmasını Deb'in kuralları ile birleştirerek literatürde bulunan sonuçlar ile karşılaştırmıştır. 13 kısıtlı kıyas fonksiyonu üzerinde önerilen yaklaşım test edilmiştir.

Mallipeddi ve Suganthan (2010)'in çalışmasında son 30 yıldır literatürde çalışılan kısıtlarla mücadele etme yöntemleri üzerine analizler yapılmıştır. Wolpert ve Macready (1997) tarafından önerilen NFL teoremi bir yöntemin bütün problemlerde başarılı olamayacağını ortaya koymuştur. Çözüm uzayı ile uygun değerlerin çözüm uzayının oranı, çözülecek olan problemin modu (unimodal, multimodal vb.), araştırma sürecindeki keşif ve faydalanma dengesi, uygulanan kısıtla mücadele yöntemi sonuca doğrudan etki etmektedir. Uygun çözümlerin ön plana çıkarılması, kendinden uyarlamalı ceza yöntemi, epsilon kısıt yöntemi, stokastik sıralama, (stochastic ranking, SR) ve çok amaçlı kısıt yöntemi bu çalışma kapsamında değerlendirilmiştir. 24 fonksiyondan oluşan CEC2006 kısıtlı test kümesi ile analizler yapılmıştır. Yeni önerilen kısıtla başa çıkma tekniklerinin birleştirilmesi (ensemble of constraint handling techniques, ECHT) yaklaşımı, mevcut yöntemlerden daha iyi sonuçlar üretmiştir.

Karatas (2011), yüksek lisans tezinde DE algoritması ile birlikte kısıtlarla başa çıkma yöntemlerinden yakın uygunluk eşiği (near feasibility threshold) ve epsilon kısıtı (epsilon constraint) metotlarını kullanmış ve yaklaşımı 18 kısıtlı test problemi üzerinde denemiştir. Sonuçlar incelendiğinde Epsilon kısıtı metodunun yakın uygunluk eşiği metodundan daha iyi olduğu görülmüştür.

Akay ve Karaboga (2017), çalışmalarında ABC algoritmasının 9 farklı çeşidini 13 kısıtlı kıyas fonksiyonu üzerinde test etmiştir. Kısıtlarla başa çıkabilmek için Deb'in kuralları mekanizması kullanılmıştır.

Bu bölümdeki kaynak araştırmasının ışığında, sürekli kısıtlı optimizasyon problemlerinin çözümü için TSA'nın ağaç ve tohum seçim mekanizması, Deb'in kuralları ile değiştirilerek TSA'nın sürekli kısıtlı optimizasyon problemlerini çözecek hale getirilmesine karar verilmiştir.

2.2. İkili Optimizasyon Algoritmaları

Bu bölümde ikili optimizasyon problemlerinin çözümü için kullanılan yöntemlerden kısaca bahsedilmiş olup, kıyas için tercih edildiğinden dolayı PSO, ABC, DE algoritmaları alt bölüm başlığı şeklinde ayrılmıştır. Bu bölümdeki kaynak araştırmasının ışığında, ayrık ikili optimizasyon problemlerinin çözümü için TSA ikili uzayda çalışacak şekilde yeniden dizayn edilerek, tohum üretme için XOR mantıksal kapısı ve benzerlik tabanlı yaklaşım kullanılmasına karar verilmiştir.

2.2.1. Parçacık sürü optimizasyonu tabanlı ikili optimizasyon algoritmaları

Kennedy ve Eberhart (1997), PSO algoritmasını yeniden yapılandırarak ayrık ikili uzayda çalışacak şekilde dizayn ederek BPSO olarak isimlendirmiştir. Bu çalışmada, sürekli değerli değişkenlerden ikili değerlere dönüşüm için sigmoid fonksiyonu bir lojistik fonksiyon olarak kullanılmıştır.

Sevki ve Guner (2006)'in çalışmasında UFLP'yi çözmek için, sürekli PSO algoritması transfer fonksiyonları yardımıyla ikilileştirilerek kullanılmıştır. Çözüm kalitesini artırmak için, PSO algoritmasına yerel arama eklenmiştir. Önerilen algoritmada, sürekli uzaydaki çözümlerin, ikili uzaya dönüştürülmesinde mod_2 ve kırpma (floor) lojistik fonksiyonları uygulanmıştır. Sonuçlar, GA ve Evrimsel SA (ESA) gibi iki metasezgisel yaklaşımın sonuçlarıyla karşılaştırılarak sunulmuştur.

Nezamabadi-pour ve ark. (2008), yeni bir ikili PSO algoritması ortaya koyarak ismini NBPSO olarak belirlemişlerdir. NBPSO, sürekli değerli değişkenleri ikili değerlere dönüştürmek için hız tabanlı bir sigmoid fonksiyonu kullanılmıştır. Ayrıca NBPSO geliştirilerek Guaranteed Convergence BPSO (GCBPSO) ve Improved NBPSO (INBPSO) isimli yöntemler de önerilmiştir. GCBPSO için farklı bir hız güncelleme denklemi önerilmiştir. INBPSO yönteminde algoritmanın durağanlaşması (stagnation) denetlenmiş ve bir durağanlaşma kontrol parametresi kullanılarak sigmoid fonksiyonu değiştirilmiştir.

Guner ve Sevkli (2008), ayrık optimizasyonda çokça çalışılmış UFLP'leri çözmek için ayrık parçacık sürüsü optimizasyonu (discrete particle swarm optimization, DPSO) yöntemini önermiştir. Bu çalışmanın yazarları, sonuçları geliştirmek için yerel bir arama mekanizmasıyla önerdikleri yöntemi melezleştirmiştir.

Yuan ve ark. (2009), BPSO'ya lamda yineleme yöntemini (lambda-iteration method) entegre ederek, birim yüklenme (unit commitment, UC) probleminin çözümü için yeni geliştirilmiş bir ikili PSO (IBPSO) yöntemi önermiştir. IBPSO yönteminin başarısını doğrulamak amacı ile 10-100 aralığında birim sayılı UC sistemleri üzerinde, literatürdeki diğer yöntemler karşılaştırılmıştır. Sayısal sonuçlar, IBPSO'nun daha düşük üretim maliyeti ve daha kısa hesaplama süresi açısından literatürdeki bilinen diğer yöntemlerden üstün olduğunu göstermiştir.

Saha ve ark. (2011), çalışmalarında PSO'yu ikili optimizasyon problemleri için uyarlamışlardır. İkileştirme işlemi sürekli değerlerin ikiye göre modunun alınması ile gerçekleştirilmiştir. PSO'nun özel bir parametresi olan atalet ağırlığı (inertia weight) 0.9 – 0.4 aralığında iterasyon sayısına bağlı olarak azalan şekilde ayarlanmıştır. Sürekli değerlerle işlem yaptığı için hedeflenen optimum değerlere yakın değerler üretilebilmiştir.

Bansal ve Deep (2012)'in çalışmasında özellikle 0-1 sırt çantası problemi (knapsack problem, KP) ve çok boyutlu KP'nin (MKP) çözümü için yeni bir modifiye edilmiş BPSO (Modified BPSO-MBPSO) algoritması önerilmiştir. Temel BPSO ile karşılaştırıldığında, bu geliştirilmiş algoritma, KP'lerin çözümünde, sürüdeki çeşitliliği muhafaza etmek ve daha keşfedici hale getirmek için yeni bir olasılık fonksiyonunu önermiştir. Parçacığın hızının normalleştirilmesi için sigmoid fonksiyonu kullanılmıştır.

Beheshti ve ark. (2015), memetik ikili PSO yaklaşımını önermişlerdir. Binary hybrid topology PSO (BHTPSO) ismi verilen yaklaşıma, global araştırmayı artıran yeni bir yöntem entegre edilerek BHTPSO quadratic interpolation (BHTPSO-QI) ismiyle

literatüre kazandırılmıştır. Çalışmada 0-1 MKP'ler kıyas için kullanılmıştır. Elde edilen sonuçlar BPSO ve ikili yerçekimi arama algoritması (binary gravitational search algorithm, BGSA) ile karşılaştırılmıştır. Yakınsama hızı ve çözüm doğruluğu çerçevesinde önerilen yöntemin başarısı ortaya konmuştur.

2.2.2. Yapay arı kolonisi tabanlı ikili optimizasyon algoritmaları

Pampará ve Engelbrecht (2011)'in çalışmasında, ikili değerli alanlarla optimizasyon problemlerine uygulanması amacı ile ABC'nin üç versiyonu önerilmiştir. Bu versiyonlardan en iyisi olan açı modülasyonlu ABC (angle-modulated ABC, AMABC), açı modülasyonlu PSO (angle-modulated PSO, AMPSO) ve açı modülasyonlu DE (angle-modulated differential evolution, AMDE) algoritmaları ile karşılaştırılmıştır. Açı modülasyonu telekomünikasyon endüstrisinde sıklıkla kullanılan bir sinyal işleme tekniğidir.

Kashan ve ark. (2012), ABC algoritmasını ikili optimizasyon için yeniden dizayn ederek DisABC isimli yeni bir yaklaşım önermişlerdir. DisABC, yeni bireyler oluşturmak için ikili vektörler arasındaki benzerlik ölçüsünü kullanmaktadır. Çalışmada benzerlik ölçüm birimi olarak Jaccard (1901)'in benzerlik katsayısı esas alınmıştır. DisABC ile yerel arama modülü hibritlenmiştir. Bu yerel arama modülü 0 olan bitleri 1, 1 olan bitleri 0 yapmaktadır. Çözümdeki toplam 1 sayısı değişmediğinden alternatif çözümler hızlıca değerlendirilmiştir.

Kiran ve Gunduz (2013), binABC ismini verdikleri bir ABC varyantı önermişlerdir. Bu çalışmada aday çözüm üretme denklemi, XOR mantıksal kapısı ile yeniden dizayn edilerek ikili bireyler üretilmesi sağlanmıştır. binABC'nin başarısını test etmek için UFLP'ler kullanılmış ve sonuçlar BPSO ve DisABC algoritmaları ile kıyaslanmıştır.

Jia ve ark. (2014), temel ABC'ye benzer bir çerçeveye sahip, BitABC olarak isimlendirilen, ancak işçi arıların ve kâşif arıların hareketleri için bitsel (bitwise) bir operatör kullanan etkili bir ikili ABC algoritması önermiştir. Hareket mekanizmasında kullanılan bitsel operatörler, XOR, AND ve OR operatörleridir.

Kiran (2015b)'in çalışmasında ABC mod tabanlı lojistik fonksiyonlar ile ikilileştirilmiş ve ABCbin ismi verilmiştir. Bu çalışmada sürekli uzayda çalışan arılar, amaç fonksiyonunda değerlendirmeye girmeden mod işlemi ile ikili hale getirilmiştir.

Ozturk ve ark. (2015), ABC algoritmasını genetik ve komşuluk operatörleri kullanarak ikili aday çözüm üretecek hale getirmişlerdir. Bu yaklaşıma genetik operatörlere dayalı yapay arı kolonisi algoritması (artificial bee colony algorithm based on genetic operators, GB-ABC) ismi verilmiştir. GB-ABC yaklaşımı ile ikili uzayda ABC algoritmasının global ve yerel arama kapasitesi artırılmıştır. Temel ABC algoritmasının komşu arama mekanizmasına entegre edilmiş olan modifikasyon dört aşamadan oluşmaktadır: (1) Güncel bir gıda kaynağının komşuluklarından, popülasyondan rastgele iki gıda kaynağı seçilir. (2) Çocuk besin kaynakları üretmek için mevcut, iki komşu, küresel en iyi ve ikili sıfır değerleri ile oluşturulan çözümler arasında iki noktadan çaprazlama (two-point crossover) operatörü uygulanır. (3) Torun gıda kaynakları üretmek için, çocuk gıda kaynaklarına takas (swap) operatörü uygulanır. (4) Çocuk ve torun besin kaynakları arasında, en iyi besin kaynağını seçilir.

2.2.3. Farksal gelişim (DE) tabanlı ikili optimizasyon algoritmaları

Pampara ve ark. (2006), açı modülasyonu ile DE algoritmasını ikili optimizasyon için uyarlamışlardır. Açı modülasyonu yaklaşımı, sürekli değerli bir arama uzayını, ikili değerli bir arama uzayına eşlemek için bir mekanizma sağlamaktadır.

Engelbrecht ve Pampara (2007), angle modulated DE (AMDE), binary DE (binDE) ve normalization DE (normDE) isimlerini verdikleri 3 farklı ikili DE varyantı önermişlerdir. normDE algoritması sürekli değerleri 0-1 arasına normalize etmekte ve normalize edilmiş bu değer 0.5'ten küçükse 0, büyükse 1 değeri atayarak ikilileştirmektedir.

Yang (2008) ve Wang ve ark. (2010a), ikili olarak kodlanmış problemlerin çözümü için DE algoritmasını kullanmışlardır.

Deng ve ark. (2011), mantıksal kapılar (XOR, OR, AND, NOT) kullanarak ikili aday çözümler üretmişlerdir. Binary DE (BDE) ismi verilen bu yaklaşım ile ikili optimizasyon problemleri çözülmüştür.

Kashan ve ark. (2013), DE algoritmasını Jaccard (1901)'in benzerlik katsayısı temelli yeni bir birey üretme stratejisi ile donatmıştır.

Chen ve ark. (2015), binary learning DE (BLDE) ismini verdikleri bir algoritma ile PSO içerisindeki öğrenme mekanizmasını kullanan bir ikili algoritma geliştirmişlerdir.

Banitalebi ve ark. (2016), bir benzerlik ölçütü ile kendinden uyarlamalı ikili DE (self-adaptive binary DE, SabDE) ismini verdikleri bir algoritma önermişlerdir. SabDE

yeni bireyleri seçmek için uyarlamalı mekanizma ve parametre değerlerini adapte etmek için kaotik bir süreç kullanmaktadır. Donör vektör oluşumu için mantıksal kapı tabanlı bir yaklaşım önerilmiştir. Önerilen algoritma, yüksek boyutlu sırt çantası problemlerinde ve Evrimsel Hesaplama Kongresinde (Congress on Evolutionary Computation, CEC 2015) sunulmuş 15 adet öğrenme temelli problem üzerinde çalıştırılmış ve güncel algoritmalar ile kıyaslanmıştır.

2.2.4. Diğer ikili optimizasyon algoritmaları

Alves ve Almeida (1992), benzetimli tavlama (simulated annealing, SA) algoritmasını ikili optimizasyona uygulayarak iyi sonuçlar elde etmişlerdir, fakat bu sonuçların üretilmesi uzun zaman aldığından efektif olarak kullanılması mümkün olmamıştır. Aydın ve Fogarty (2004), kombinatoriyal optimizasyon problemleri için dağıtık evrimsel SA (distributed ESA) algoritmasını önermiştir. Bu yöntem SA'dan daha iyi sonuçları daha kısa sürede elde etmeyi başarmıştır.

Al-Sultan ve Al-Fawzan (1999) ve Sun (2006), tabu arama (tabu search, TS) algoritmasını UFLP'yi çözmek için modifiye etmişlerdir. Michel ve Van Hentenryck (2004) TS algoritmasını depo yerleşim problemini (warehouse location problem) çözmek için kullanmışlardır.

Rashedi ve ark. (2010) yerçekimi arama algoritması'nın (gravitational search algorithm, GSA) ikili versiyonunu önererek ismini BGSA olarak belirlemiştir. BGSA, tanjant hiperbolik fonksiyonları, lojistik fonksiyonu olarak kullanarak sürekli değerleri ikili değerleri dönüştürmüştür. Performans analizi için nümerik fonksiyonlar üzerinde test yapılmıştır. Mirjalili ve ark. (2014b), GSA'yı BPSO ile hibritleştirerek BPSOGSA isimli yeni bir ikili optimizasyon algoritması önermişlerdir. Çalışmada ikilileştirme için transfer fonksiyonu yaklaşımı kullanılmıştır. Nezamabadi-pour (2015) kuantum hesaplama temelli bir GSA yaklaşımı önererek BQIGSA ismini vermiştir. Kuantum biti, süper pozisyon ve değiştirilmiş rotasyon Q-geçit stratejileri, GSA'nın konum ve hız güncelleme kurallarıyla değiştirilmiştir.

Harmoni arama (harmony search, HS) algoritması ile ikili optimizasyon problemlerini çözmek için Wang ve ark. (2010b) ayrık ikili HS (discrete binary HS, DBHS), Salman ve ark. (2015) adaptif olasılıklı (adaptive probabilistic) HS, Kong ve ark. (2015a) basitleştirilmiş ikili HS (simplified binary harmony search, SBHS) algoritmalarını önermişlerdir. Kong ve ark. (2015b), MKP'leri çözmek için, yeni SBHS

algoritmasını önermişlerdir. SBHS ve diğer HS yöntemleri arasındaki temel fark doğaçlama (improvisation) sürecindedir. Akort ayarlama oranı (pitch adjustment rate) ve adım bant genişliği (step bandwidth) yerine harmoni belleğinde saklanan harmoniler arasındaki farklar, yeni çözümler üretmek için kullanılmıştır. Ayrıca, algoritmanın yakınsamasını artırmak için harmonik bellek dikkate alma oranı (harmony memory considering rate), dinamik olarak, boyut sayısı dikkate alınarak ayarlanmıştır. Dolayısıyla, önerilen yöntem ile uygun parametre seçim işlemi gerekmemiştir. Popülasyon çeşitliliğini daha da artırmak için, uygun olmayan çözümler etrafında, sezgisel tabanlı yerel arama yapılmakta ve daha iyi kalitede çözümler elde edilmesi amaçlanmaktadır. Ayrıca Kong ve ark. (2015b), yeni ikili HS (new binary HS, NBHS) isimli yeni bir yaklaşım ile büyük ölçekli MKP'leri (large-scale MKP) çözmüşlerdir.

Kratica ve ark. (2001) ve Jaramillo ve ark. (2002), GA ile yerleşim problemlerini çözmüştür. Tohyama ve ark. (2011), UFLP'nin çözümü için GA tabanlı bir yaklaşım önermiştir. UFLP'de tesis yerleşimi maliyeti ve tesis kullanıcısı maliyeti oranına göre, tesis yeri sayısı kabaca tahmin edilebilir. Bu nedenle, iyi bir çözüm ihtimali bulunan kısmi çözüm uzaylarının bir dereceye kadar sınıflandırma endeksine dayanarak öngörülebileceği düşünülmüştür. Mutasyonla, iyi bir çözüm ihtimali bulunan çözüm alanını arayan işlem birlikte kullanıldığında, önerilen yöntem bütün çözüm uzayını verimli bir şekilde arayabilmiştir.

Ghosh (2003), UFLP'yi çözmek için komşuluk arama yöntemlerini kullanmıştır. Kong ve ark. (2008), yeni bir karınca kolonisi optimizasyonu (ant colony optimization, ACO) yaklaşımı geliştirerek ismini ikili karınca sistemi (binary ant system, BAS) koymuştur ve MKP'nin çözümünde kullanmıştır. BAS, ikili çözüm yapısı için özel olarak tasarlanmış bir feromon döşeme yöntemi kullanmış ve çözüm yapım prosedüründe uygun olmayan çözümlerin üretilmesine izin vermiştir.

Dağılım arama metodolojisi (scatter search methodology), Gortázar ve ark. (2010) tarafından ikili optimizasyon problemlerinin çözümü için uyarlanmıştır.

Azad ve ark. (2012) yapay balık sürüsü algoritmasının (artificial fish swarm algorithm, AFSA) ikili versiyonunu ortaya koyarak 0-1 MKP'yi çözmüştür. Uygun olmayan çözümler bir kod çözme algoritması (decoding algorithm) kullanılarak uygun çözümlere dönüştürülmüştür. Yine Azad ve ark. (2013), basitleştirilmiş bAFSA (simplified binary version of AFSA, S-bAFSA) isimli bir yöntem ile UFLP'nin çözümü için modifikasyonlar yapmıştır. Bu çalışmada, çaprazlama ve mutasyon kullanarak yeni bireyler oluşturulmuştur. Çözümlerin kalitesini artırmak için, popülasyonun çevrimsel

olarak yeniden başlatılması gerçekleştirilmiştir. Çözümün doğruluğunu artırmak için önceden tanımlanmış sayıda noktaya yerel arama uygulanmıştır.

Yarasa algoritması (bat algorithm, BA) Babaoğlu (2016) tarafından ikilileştirilmiştir.

Laalaoui ve M'Hallah (2016), bağlı liste veri yapısı ve dinamik eşik kabul kriteri ile çalışan sezgisel değişken komşuluk arama algoritmasını önermiş ve MKP'leri çözmek için kullanmıştır.

Zhang ve ark. (2016), çalışmalarında MKP'yi etkin bir şekilde çözmek için ikili yapay bir alg algoritması (binary artificial algae algorithm, BAAA) önermişlerdir. Önerilen BAAA, ayrık süreç (discrete process), onarım operatörleri (repair operators) ve seçkin yerel arama (elite local search) adımlarından oluşmaktadır. Ayrık süreçte, farklı eğri katsayılarına sahip iki lojistik fonksiyon kullanılmıştır. Onarım operatörleri, çözümün verimliliğini artırmak için gerçekleştirilmiş ve son olarak seçkin yerel arama adımında ise çözümün kalitesini artırmak amaçlanmıştır. Önerilen algoritmanın verimliliğinin kanıtlanması amacı ile 94 kıyas problemi üzerinde çalıştırılmış ve MBPSO, BPSOTVAC, CBPSOTVAC, çift dizili GA (genetic algorithm with double strings, GADS), bAFSA ve IbAFSA gibi biyolojik esinli güncel algoritmalar ile karşılaştırılmıştır.

Korkmaz ve ark. (2018)'nin çalışması ile Korkmaz ve Kiran (2018)'in çalışmasında AAA ile UFLP problemlerinin çözümü için yeni sezgisel yaklaşımlar önerilmiştir.

2.3. Ayrık Tam Sayılı Optimizasyon Algoritmaları

Gezgin satıcı problemi (traveling salesman problem, TSP) anlaşılması ve matematiksel modeli oldukça kolay olan fakat NP-Hard (Arora, 1998) problem kümesine dahil olduğundan çözülmesi polinomsal zamanda mümkün olmayan kombinatoriyal optimizasyon problemlerindedir. TSP'de ana problem ağırlıklandırılmış graf üzerinde minimum maliyet ile bir Hamilton yolu bulunmasıdır. Hamilton yolu geçilen bir noktadan bir daha geçilmemek kaydıyla tüm noktalardan geçerek tekrar başlangıç noktasına geldiği durumdaki yoldur. Çizelgeleme (scheduling), rotalama (routing), test edilebilir çok büyük ölçekli-entegre taramalı hücrelerin işlenmesi (threading of scan cells in a testable very-large-scale-integrated), bilgisayar kablolama (computer wiring), baskı devre kartlarının ve devrelerin otomatik delinmesi (automatic drilling of printed circuit

boards and circuits), X-ışını kristalografisi (X-ray crystallography) gibi önemli problemler TSP ile modellenip çözülebildiğinden TSP literatürde önemli bir kıyas fonksiyonu olmuştur (Ravikumar, 1992). TSP'yi çözmek için birçok kesin ve sezgisel yöntem önerilmiştir. Kesin yöntemlere dal-sınır (branch and bound) (Lawler ve Wood, 1966), dal-kes (branch and cut) (Padberg ve Rinaldi, 1987), dal-fiyat (branch and price) (Barnhart ve ark., 1998), kesme düzlemi (cutting planes) (Laporte ve Nobert, 1980) gibi yöntemler örnek verilebilir. Sezgisel yöntemlere ise ABC (Karaboga ve Gorkemli, 2011; Li ve ark., 2011; Görkemli ve Karaboga, 2013; Kiran ve Gunduz, 2013; Gunduz ve ark., 2015), PSO (Wang ve ark., 2003; Pang ve ark., 2004; Shi ve ark., 2007; Chen ve Chien, 2011; Deng ve ark., 2012; Mahi ve ark., 2015), SA (Chen ve Chien, 2011), ACO (Dorigo ve Gambardella, 1997; Chen ve Chien, 2011; Deng ve ark., 2012; Gunduz ve ark., 2015; Mahi ve ark., 2015; Yong, 2015; Gulcu ve ark., 2016), sinir ağı (neural network) (Potvin, 1993), TS (Knox, 1994), yapay bağışıklık sistemleri (artificial immune systems) (Baykasoglu ve ark., 2013), CS (Ouaarab ve ark., 2014; 2015), BH (Hatamlou, 2017), durum geçiş algoritması (state transition algorithm, STA) (Chunhua ve ark., 2012; Zhou ve ark., 2016), fruit fly (Iscan ve Gunduz, 2017), emperyalist rekabet algoritması (imperialist competitive algorithm, ICA)(Chen ve ark., 2017a), fiziksel enerji optimizasyonu (physarum-energy optimization, PEO) (Feng ve ark., 2017) ve GA (Grefenstette ve ark., 1985; Oliver ve ark., 1987; Mühlenbein, 1989; Whitley ve ark., 1989; Braun, 1990; Ulder ve ark., 1990; Starkweather ve ark., 1991; Whitley ve ark., 1991; Potvin, 1996; Larranaga ve ark., 1999; Bryant ve Benjamin, 2000; Üçoluk, 2002; Tsai ve ark., 2004; Ahmed, 2010; Albayrak ve Allahverdi, 2011; Chen ve Chien, 2011; Deng ve ark., 2012; Hussain ve ark., 2017; Jubeir ve ark., 2017) örnek verilebilir.

Genellikle, permütasyon kodlu optimizasyon problemlerini çözmek için iki ana yaklaşım göze çarpmaktadır. Bu yaklaşımlar yol yapımına (path construction) ve yol geliştirmeye (path improvement) dayanır (Gunduz ve ark., 2015). Yol yapım yöntemleri genellikle probleme ait bilgileri kullanmışlardır, bu nedenle yeni çözümler üretmek için daha fazla zaman harcarlar. Yol geliştirme yöntemleri ise çaprazlama, mutasyon, takas vb. operatörlerle yeni çözümler ürettiğinden daha az zaman harcarlar fakat kaliteli bir çözüm elde etmek için uzun bir yineleme süresine ihtiyaç duyarlar.

TSP'leri çözmek için kullanılan evrimsel hesaplama ve sürü zekası yöntemleri aşağıda kronolojik olarak verilerek kısaca bahsedilmiştir.

Goldberg ve Lingle (1985), TSP'yi çözmek için kısmen eşlenmiş çaprazlama (partially-mapped crossover, PMX) operatörünü önermiştir. Grefenstette ve ark. (1985)

TSP için GA üzerinde sıralı (ordinal) ve bitişik (adjacency) gösterimi tartışmıştır. Sezgisel çaprazlama (Heuristic Crossover, HX) bu çalışmada önerilmiştir. HX ile 50, 100 ve 200 şehirli yapay olarak oluşturulan problemler çözülmüş ve elde edilen sonuçlar, HX'in TSP için iyi bir çözücü olmadığını göstermiştir. Ancak bu çalışma, TSP'lerin çözümü için gelecekteki bazı çalışmalara öncü olmuştur. Fogel (1988), alternatif evrimsel bir yaklaşım ile TSP'yi çözmüştür. Önerdiği yaklaşım PMX ile entegre edilmiş GA ile karşılaştırıldığında GA'ya göre çözüm kalitesi açısından daha iyi performans gösterdiği ortaya konulmuştur. Başka bir çalışmada, Braun (1990), büyük boyutlu TSP'leri çözmek için GA kullanmıştır. Bu çalışmada, GA en yakın komşu sezgiseliyle başlatılmış ve 2-opt veya or-opt yöntemleri ile sonlandırılmıştır. Aday çözümleri oluşturmak için düzenli çaprazlama (order crossover, OX) operatörünü kullanan paralel bir yaklaşım olan adalı GA (Insular GA) önerilmiştir. Popülasyon, adalar şeklinde gruplandırılmış ve işlemler paralel olarak sürdürülmüştür. Deneysel sonuçlar, bu yaklaşımın TSP'lerin çözümü için kullanılabilir olduğunu göstermiştir.

Ulder ve ark. (1990), genetik yerel aramalı (genetic local search) GA ve çok başlatmalı yerel aramalı (multi-start local search) GA'da 2-opt ve Lin-Kernighan yerel arama yöntemlerini analiz etmiş, elde edilen sonuçlar 2-opt SA ve 2-opt eşik kabulü (threshold accepting) yöntemleri ile karşılaştırılmıştır. Lin-Kernighan yerel arama yöntemleriyle güçlendirilmiş genetik yerel arama, diğer yaklaşımlardan daha iyi performans göstermiştir. Starkweather ve ark. (1991), çalışmalarında kör TSP ve klasik TSP arasındaki farkları tartışmışlardır. Kör TSP'de şehirlerarası uzunluklar sadece tur uzunluğunu belirlemek için kullanılmışken, klasik TSP'de ise şehirlerarası uzunluklar her turun oluşturulması aşamasında kullanılmıştır. Bu çalışmada, geliştirilmiş kenar rekombinasyon çaprazlama (improved edge recombination crossover, ERX), sıra tabanlı çaprazlama (order based crossover, OBX), pozisyon tabanlı çaprazlama (position-based crossover, PBX), PMX ve çevrim çaprazlama (cycle crossover, CX) karşılaştırılmıştır. Ayrıca ebeveyn bireylerdeki önemli kenarların korunması için geliştirilmiş ERX yöntemi bu çalışmada geliştirilmiştir. Oliver30 TSP bu altı yöntemle çözülmüş ve sonuçlar karşılaştırılmıştır. Ayrıca, bu yöntemlerle bir depo / nakliye (warehouse/shipping) çizelgeleme problemi çözülmüştür. Oliver30 TSP'yi çözmek için algoritmanın performansı kabul edilebilir düzeyde iken çizelgeleme problemini çözmek için değildir. Bu durum operatörlerin başarısının probleme bağlı olduğunu ortaya koymaktadır. Potvin (1996), TSP'yi çözmek için kullanılan genetik algoritmalar hakkında bir literatür araştırması hazırlamıştır. Bu çalışmada, yazar TSP için çaprazlama ve mutasyon

tekniklerini tartışmış ve paralel uygulamalara değinmiştir. PMX, CX, değiştirilmiş çaprazlama (modified crossover), OX, OBX, PBX, ERX, alternatif kenarlar çaprazlama (alternate edges crossover), HX çaprazlama teknikleri detaylı olarak ele alınmıştır. Mutasyon yaklaşımları olarak ise takas, yerel tepe tırmanma (local hill-climbing) ve çırpma (scramble) teknikleri incelenmiştir. Larranaga ve ark. (1999), çalışmalarında ikili, yol, bitişiklik, sıra ve matris gibi çeşitli gösterimleri tartışmıştır. Sekiz çaprazlama operatörünün (alternatif pozisyon çaprazlama (alternating position crossover, APX), CX, ERX, OX, OBX, PMX, PBX ve oylama rekombinasyon çaprazlama (voting recombination crossover, VRX)) ve 6 mutasyon operatörünün (değişim mutasyon (exchange mutation, EM), devrik mutasyon (inversion mutation, IVM), yer değiştirme mutasyonu (displacement mutation, DM), ekleme mutasyonu (insertion mutation, ISM), basit inversiyon (simple inversion mutation, SIM), çırpma mutasyonu (scramble mutation, SM)) kombinasyonu olan 48 farklı yaklaşım ile entegre edilmiş GA, üç TSP üzerinde incelenmiştir. Bu çalışmada elde edilen üç önemli sonuç vardır. Birincisi, ERX diğer tüm çaprazlama operatörlerini geride bırakmıştır. İkincisi, mutasyon operatörleri arasında anlamlı bir fark yoktur. Üçüncüsü, TSP için en iyi temsil tipi yol gösterimidir.

Bryant ve Benjamin (2000), GA'da çaprazlama ve mutasyon yöntemlerinin TSP'lere etkilerini araştırmıştır. Sonuçlar, matris temsiline ve HX'in çözüm kalitesi açısından iyi sonuçlar verdiğini göstermiştir. Üçoluk (2002), PMX tekniğini temel alan yeni bir çaprazlama operatörü önermiştir. Önerilen yöntem Bays29, Berlin52 ve Eil101 problemlerinde PMX'ten daha kötü sonuçlar vermiştir. Ayrıca, hem PMX hem de önerilen yöntem optimum sonuçlara ulaşamamıştır. Wang ve ark. (2003), Burma14 problemi için takas ve takas dizisi mutasyon yöntemleriyle donatılmış PSO algoritmasını kullanmıştır. Tsai ve ark. (2004), genetik operatörlerin TSP'lerdeki performansını araştırmış ve popülasyon çeşitliliğini koruyan bir yaklaşım önermiştir. DPSO algoritması, simetrik ve genelleştirilmiş TSP'lerin çözümü için Shi ve ark. (2007) tarafından önerilmiştir. 5 simetrik ve 19 genelleştirilmiş TSP, DPSO ile çözülmüştür. Sonuçlar DPSO'nun bu tür problemler için alternatif bir çözücü olduğunu göstermiştir. DPSO'da, aday çözümler kaydırma (slide) ve ters yüz (reverse) operatörleri tarafından oluşturulmuştur. Ayrıca, silme-çaprazlama işlemi (delete-crossover process) (2-opt'a benzer), çözüm kalitesini artırmak için bu algoritmayla bütünleştirilmiştir.

Ahmed (2010), TSP'ler için sıralı yapıcı çaprazlama (sequential constructive crossover, SCX) tekniğini önermiştir. SCX, simetrik ve asimetrik TSPLIB örneklerinde genelleştirilmiş N-nokta çaprazlama (generalized N-point crossover, GNX) ve ERX ile

karşılaştırılmıştır. Deneysel sonuçlar SCX'in ERX ve GNX'ten daha iyi çözümler ürettiğini göstermiştir. ABC algoritması, Li ve ark. (2011) tarafından takas operatörleri ile ayrıklaştırılmıştır. Önerilen ayrık ABC (discrete ABC, DABC), PSO (Wang ve ark., 2003) ve GA (Braun, 1990) ile kıyaslanmış, sonuçlar DABC'nin Burma14 ve Dantzig42 problemlerinde PSO'dan daha iyi olduğunu ve GA ile yarıştığını göstermiştir. Albayrak ve Allahverdi (2011), açgözlü alt tur mutasyon (greedy sub tour mutation, GSTM) adını verdikleri yeni bir mutasyon operatörü önermiş ve EM, DISP, INV, ISM, SIM, SCM ve GSM mutasyon teknikleri ve DPX, OX, CX ve PMX çaprazlama teknikleri ile bu yaklaşımı kıyaslamışlardır. Deneysel sonuçlar, GSTM'nin diğer tekniklerle karşılaştırıldığında makul bir sürede kabul edilebilir çözümler ürettiğini göstermiştir. Berlin52, Kroa100, Pr144, Ch150, Krob150, Pr152, Rat195, D198, Kroa200, Ts225, Pr226, Pr299, Lin318 ve Pcb442 problemleri kıyaslama seti olarak kullanılmıştır. Karaboga ve Gorkemli (2011), kombinatoriyal optimizasyon problemlerini çözmek için kombinatoriyal ABC (combinatorial ABC, CABC) algoritmasını geliştirmiştir. CABC en yakın komşu tur oluşturma sezgiseli ile oluşturduğu popülasyonu GSTM (Albayrak ve Allahverdi, 2011) kullanarak oluşturduğu aday çözümler için geliştirerek arama sürecini yürütmüştür. Bu yaklaşımın verimliliğini analiz etmek için Krob150 ve Kroa200 problemleri kullanılmış ve iyi sonuçlar alınmıştır. Chen ve Chien (2011), TSP'lerin çözümü için parçacık sürüsü optimizasyonlu genetik benzetimli tavlama karınca kolonisi sistemi (genetic simulated annealing ant colony system with particle swarm optimization, GSA-ACS-PSOT) olarak adlandırdıkları yeni bir yöntem önermiştir. GSA-ACS-PSOT'da, ACS ile ilk çözümler oluşturulmakta, daha sonra bu çözümler genetik benzetimli tavlama tekniklerine verilmektedir. Belirli bir yinelemenin geçmesinden sonra, elde edilen çözümler, PSO ile geliştirilmiştir. Bu yaklaşım, dört farklı metasezgisel algoritmayı hibritleyerek oluşturulmuştur. GSA-ACS-PSOT'un performansı 25 farklı TSP'de test edilmiş ve rekabetçi çözümler ürettiği gösterilmiştir. Chunhua ve ark. (2012), ayrık STA'yı (DSTA) önermiştir. DSTA, aday çözümler oluşturmak için takas, öteleme ve simetri isimli komşuluk operatörlerini kullanmıştır. Bu çalışmada DSTA, SA ve ACO ile karşılaştırılmış ve sonuçlar, DSTA'nın çok daha az zaman harcadığını ve SA ve ACO'dan daha iyi arama yeteneğine sahip olduğunu göstermiştir. Deng ve ark. (2012), GA-PSO-ACO olarak adlandırdıkları hibrit bir yaklaşım önermiştir. GA-PSO-ACO'da, GA ve PSO keşif aşamasında, ACO ise faydalanma aşamasında çalışmaktadır. Gorkemli ve Karaboga (Gorkemli ve Karaboga, 2013), TSP'lerin çözümü için hızlı ABC'yi (quick ABC, qABC) değiştirmiştir. qABC'de gözcü arı fazı temel ABC'den farklıdır. Bu

çalışmada yeni bir benzerlik ölçütü geliştirilmiş ve aday çözümler üretilirken gözcü arılar kullanılmıştır. Bu yaklaşım hızlı kombinatorial ABC (quick combinatorial ABC, qCABC) olarak adlandırılmış ve Kroa150 ve Kroa200 problemleri qCABC ile çözülmüştür. qCABC, CABC ve GSTM ile karşılaştırılmış ve bu metotlardan daha iyi bir performans gösterdiği ortaya çıkarılmıştır. Kiran ve ark. (2013), ayırık ABC algoritmasında aday çözümler oluşturmak için komşuluk operatörlerini kullanmıştır. Bu çalışmada rastgele takas (random swap, RS), rastgele ekleme (random insertion, RI), rastgele alt dizilerin takası (random swap of subsequences, RSS), rastgele alt dizilerin eklenmesi (random insertion of subsequence, RIS), rastgele alt dizilerin ters yüz edilmesi (random reversing of subsequence, RRS), rastgele alt dizilerin ters yüz edilerek takas edilmesi (random reversing swap of subsequences, RRSS) ve rastgele alt dizilerin ters yüz edilerek eklenmesi (random reversing insertion of subsequence, RRIS) isimli komşuluk operatörleri kullanılmıştır. Ayrıca bu operatörler gruplandırılmış ve takas grubu (RS, RSS ve RRSS) ve ekleme grubu (RI, RIS ve RRIS) olarak ayrı ayrı analiz edilmiştir. Oliver30, Eil51, Berlin52, St70, Pr76, Kroa100, Eil101, Tsp225 ve A280 problemleri kıyaslama seti olarak kullanılmıştır. Deneysel sonuçlar umut verici sonuçların üretildiğini ortaya koymuştur. Ouaarab ve ark. (2014), ayırık guguk kuşu arama (discrete CS, DCS) algoritmasını önermiştir. 41 simetrik TSP, DCS tarafından çözülmüştür. DCS ile elde edilen sonuçlar bazı yaklaşımlarla karşılaştırılmıştır. Sonuçlar DCS'nin simetrik TSP'ler için alternatif bir çözücü olduğunu göstermiştir. DCS, 2-opt ve çift köprü (double bridge) (Martin ve ark., 1991) teknikleriyle donatılmıştır. 2-opt yerel arama olarak kullanılmış ve her yinelemede çift köprü genel arama yöntemi olarak kullanılmış, ancak bu teknikler çok zaman alıcı yaklaşımlar olduğundan makul bir zamanda çözüm üretimi zorlaşmaktadır. Bu çalışmada en küçüğü 51, en büyüğü 1379 şehre sahip TSP'ler DCS ile çözülmüştür. Gunduz ve ark. (2015), ACO-ABC olarak adlandırdıkları melez bir algoritma önermiştir. Bu yöntemde ACO ve ABC, TSP'lerin çözümü için birleştirilmiştir. Bu hibrit yaklaşım, ABC ve ACO algoritmalarının başarılı yönleri kullanılarak önerilmiştir. Bu çalışmada, ABC ve ACO'nun zayıf yönleri belirtilmiş ve hibrit yaklaşım savunulmuştur. Mahi ve ark. (2015), PSO, ACO ve 3-opt algoritmasını içeren bir hibrit çözücü önermiştir. Bu yöntemde başlangıç aşamasında α ve β olarak gösterilen ACO'nun kendine özgü parametreleri, PSO tarafından optimize edilmiş ve ACO, TSP'leri bu parametrelere göre çözmüştür. Sonlandırma kriteri karşılandığında, ACO tarafından elde edilen en iyi tur 3-opt algoritmasına verilmiş ve bu yerel arama yöntemi çözümü geliştirmeye çalışmıştır. Bu yaklaşım Eil51, Berlin52, St70,

Eil76, Rat99, Kroa100, Eil101, Lin105, Ch150 ve Kroa200 problemlerinde test edilmiş ve rekabetçi çözümler üretilmiştir. Sürekli arama uzayındaki değişkenleri tamsayı arama uzayına aktarmak için rastgele anahtar kodlama şemasını kullanan rastgele anahtar guguklu arama (random key cuckoo search, RKCS) yaklaşımı Ouaarab ve ark. (2015) tarafından önerilmiştir. RKCS'de, 2-opt yerel arama algoritması daha iyi bir sonuç elde edilene kadar çalıştığından dolayı makul sonuçlar üretmek uzun zaman almaktadır. Yerel arama algoritmasına sahip hibrit max-min karınca sistemi (hybrid max-min ant system, HMMA) Yong (2015) tarafından önerilmiştir. Zhou ve ark. (2016), TSP'lerin çözümü için DSTA'yı geliştirmiştir. DSTA, aday çözümler oluşturmak için takas, öteleme, simetri ve yerine koyma (substitute) olarak isimlendirilmiş komşuluk operatörlerini kullanmıştır. Bu çalışmada Kroa100, Krob100, Kroc100, Krod100 ve Kroe100 problemleri performans testi için kullanılmış ve elde edilen sonuçlar SA ile karşılaştırılmıştır. Gulcu ve ark. (2016), TSP'lerin çözümü için ACO temelli yeni bir paralel kooperatif hibrit algoritması (parallel cooperative hybrid algorithm, PACO-3Opt) önermiştir. PACO-3Opt, 3-opt yerel arama algoritmasını kullanmıştır ve 3-opt zaman alan bir işlem olduğundan paralelleştirme algoritmayı hızlandırmak için kullanılmıştır. Yazarlar, yaklaşımlarının ACO'nun erken yakınsamasının üstesinden geldiğini iddia etmişlerdir. Hussain ve ark. (2017), CX operatörünü temel alan yeni bir çaprazlama operatörü (CX2) önermiştir. Bu operatör PMX, OX ve CX operatörleri ile karşılaştırılmıştır. Deneyler, CX2'nin bazı TSP'lerde PMX ve OX operatörlerinden daha iyi olduğunu göstermiştir. Jubeir ve ark. (2017), GA'nın seçim yönteminin TSP'ler üzerindeki etkisini araştırmıştır. Stokastik evrensel seçim (stochastic universal selection), sıra seçimi (rank selection), turnuva seçimi (tournament selection) ve rulet tekerleği seçimi (roulette wheel selection) genellikle GA'da ebeveyn seçimi için kullanılmaktadır. Bu çalışmada yazarlar GA için yeni ve geliştirilmiş bir ebeveyn seçim yöntemi önermiştir. Bu seçim yöntemi minimum mesafe bilgisini kullanmakta ve RS'ye benzemektedir, ancak sıralama yapmamaktadır. Önerilen seçim yönteminin, TSP'lerin Gr24, Brazil58, Sil75 ve Pa561 problemlerinde stokastik evrensel seçimden daha iyi performans gösterdiği deneysel sonuçlarla ortaya konmuştur. Meyve sineği optimizasyon algoritması (fruit fly optimization algorithm) TSP'lerin çözülmesi için Iscan ve Gunduz (2017) tarafından modifiye edilmiştir. Bu çalışmada önerilen yaklaşımla bir gerçek dünya problemi çözülmüştür. Hatamlou (2017), TSP'leri kara delik (black hole, BH) algoritmasıyla çözmüştür. Yıldırım ve Karci (2017) küçük ölçekli TSP'ler için yapay atom algoritmasını yeniden tasarlamıştır. Bu çalışmada, aday çözümler 2-opt yerel arama yöntemiyle oluşturulmuştur. Bu yöntemin performansı

Wi29, Att48, Eil51, Berlin52, St70, Pr76 problemlerinde incelenmiştir. Chen ve ark. (2017a), TSP'lerin çözümü için politika öğrenme işlevi (policy-learning function) ile ICA algoritmasını birleştirmiştir. Feng ve ark. (2017) TSP'lerin çözümü için PEO algoritmasını önermiştir.

Bu bölümdeki kaynak araştırmasının ışığında ayrık tam sayılı optimizasyon problemlerinden TSP'yi çözmek için metasezgisel algoritmaların üç ana aşamada kurulduğu tespit edilmiştir. Bu aşamalar başlatma, çaprazlama ve mutasyondur. Permütasyon kodlamalı yol gösterimli TSP'lerin çözümü için başlatma sırasında rastgele oluşturma ve en yakın komşu sezgiseli kullanılmaktadır. Çaprazlama aşamasında yeni bireyler üretmek için mevcut popülasyondan seçilen bireyler PMX (Goldberg ve Lingle, 1985), OX (Davis, 1985), CX (Oliver ve ark., 1987), OBX (Syswerda, 1991), PBX (Syswerda, 1991), HX (Grefenstette, 1987), sıralanmış eşlenmiş çaprazlama (sorted match crossover, SMX) (Brady, 1985), ERX (Whitley ve ark., 1989; Whitley ve ark., 1991), maksimum koruyucu çaprazlama (maximal preservative crossover, MPX) (Mühlenbein ve ark., 1988), VRX (Mühlenbein, 1989), APX (Larrañaga ve ark., 1997), ve SCX (Ahmed, 2010) gibi yöntemler kullanılarak oluşturulabilir. Mutasyon aşamasında ise popülasyondan seçilen tek birey üzerinde EM (Banzhaf, 1990), DM (Michalewicz, 1992), IVM (Fogel, 1993), ISM (Fogel, 1988), SIM (Grefenstette ve ark., 1985), SM (Syswerda, 1991), GSTM (Albayrak ve Allahverdi, 2011), komşuluk operatörleri (RS, RI, RSS, RIS, RRS, RRSS, RRIS) (Kiran ve ark., 2013) ve dönüştürme operatörleri (Chunhua ve ark., 2012; Zhou ve ark., 2016) gibi teknikler kullanılarak yeni çözümler üretilir. Sonlandırma kriterine erişildikten sonra ise elde edilen en iyi sonuç 2-opt (Croes, 1958), Or-opt (Or, 1976), double bridge (Martin ve ark., 1991) veya Lin ve Kernighan komşuluğu (Lin ve Kernighan, 1973) gibi yerel arama teknikleriyle geliştirilebilir.

2.4. Ağaç-Tohum Algoritmasını Kullanan Diğer Çalışmalar

Kiran (2015a) önerdiği TSA'yı literatürdeki PSO (Kennedy ve Eberhart, 1995), ABC (Karaboga ve Basturk, 2008) (Karaboga ve Akay, 2009), ateşböceği algoritması (firefly algorithm, FA) (Yang, 2010a), BA (Yang, 2010b), HS (Geem ve ark., 2001) yöntemleri ile kıyaslamış ve özellikle küçük boyutlu problemlerdeki başarısını ortaya koymuştur. Önerildiği 2015 yılından bugüne (03/02/2020) 79 atıf (Google Scholar) alan TSA, 1 yüksek lisans tezi, 2 kitap bölümü, 12 sempozyum bildirisi ve 24 dergi

makalesinde derinlemesine incelenmiş ve kullanılmıştır. Bu bölümde kronolojik olarak TSA'nın literatür özeti sunulmuştur.

Cinar ve Kiran (2016), CUDA platformu üzerinde TSA'yı paralel hale getirmiştir. Bu sempozyum bildirisi (Cinar, 2016; Cinar ve Kiran, 2018a) çalışmalarının ön çalışması niteliğindedir.

Kiran (2016), PVDP'yi çözmek için TSA'ya statik ceza ekleyen yeni bir yaklaşım önermiştir. Bu çalışma, kısıtlı optimizasyon problemlerini çözmek için TSA'yı modifiye eden ilk çalışmadır.

Cinar (2016), TSA'nın paralelleştirilmesi üzerine bir yüksek lisans tezi hazırlamıştır. Bu tez çalışmasında, on kıyas fonksiyonu farklı ST ve popülasyon büyüklüğü (N) değerleri ile çözülmüştür. Detaylı analizler ile paralelleştirmenin TSA için uygun olduğu gösterilmiştir.

Zheng ve ark. (2016), dengeli voltaj regülasyonu (balanced voltage regulation) ve sönümlenme performansı (damping performance) için TSA tabanlı bir yuvarlanma (rolling) optimizasyonu önermişlerdir. Bu yaklaşım TSA-MPC olarak adlandırılmıştır. TSA-MPC, türbin yönetiminin ve jeneratör uyarımının kontrolünde iyi performans göstermiştir.

Muneeswaran ve Rajasekaran (2016), TSA (RBF-TSA) temelli bir radyal temel fonksiyon sinir ağı (radial basis function neural network, RBFN) geliştirmiştir. RBFN'nin parametreleri (kümeleme merkezleri, genişlik ve ağırlıklar) TSA tarafından optimize edilmiştir. Deneyler için, iki sayısal fonksiyon yaklaşımı kullanılır. TSA bu çalışmada PSO'dan daha iyi performans göstermiştir.

Muneeswaran ve Rajasekaran (2017) görüntü işleme alanındaki benek azaltma (speckle reduction) problemleri için TSA kullanmıştır. Bu çalışma, benek azaltma için kullanılan en iyi gürültü giderici filtrenin katsayılarını belirlemek için TSA kullanmıştır.

Kiran (2017b), soldurma süreci (withering process, WP) ismini verdiği bir parametreyi TSA'ya ekleyerek durağanlaşmayı (stagnation) önlemeye çalışmıştır. Bu yaklaşım TSAWP olarak adlandırılmıştır. TSA ve TSAWP 13 çok modlu kıyas fonksiyonu üzerinde karşılaştırmış ve TSAWP'nin TSA'dan daha iyi olduğu ortaya konmuştur.

Chen ve ark. (2017b), TSA ile Li-ion piller için eşdeğer devre modellerinin parametrelerini belirlemiştir. Bu çalışmada, TSA deneysel sonuçlarda GA'dan daha iyi performans göstermiştir.

Cinar ve Kiran (2017) yedi farklı arama uzayı sınırlama tekniği ile TSA'nın içerisindeki arama uzayı bariyer formüllerini değiştirerek bu yaklaşımı beş standart kıyas fonksiyonu üzerinde test etmiştir. Deneyler, TSA'nın kıyas fonksiyonlarını optimize etmek için sınır koşulu seçimine karşı dayanıklı (robust) olduğunu göstermiştir.

Cinar ve Kiran (2018a), CUDA platformunda paralel TSA'yı geliştirmiştir. Temel ve paralel TSA ile 10 standart kıyas fonksiyonu çözülmüştür. Paralel TSA, ağaç sayısı 100 ve boyut 10 olduğunda Schwefel fonksiyonunda sıralı TSA'ya göre 184.65 kat daha hızlı çalışmıştır.

Jiang Xue-ying ve ark. (2018) RBFN eğitimi için TSA'yı kullanmıştır.

Chen ve ark. (2018), çalışmasında özellik seçimi için TSA ve K-en yakın komşu (K-nearest neighbor, KNN) algoritmalarını kullanmıştır. Bu yaklaşım KNN-TSA olarak adlandırılmıştır ve ağa izinsiz giriş tespit problemini çözmüştür. KDDCUP1999 veri kümesi PSO, GA ve KNN-TSA ile çözülmüştür. KNN-TSA %87.34, PSO % 83.65 ve GA% 78.78 doğruluk ile sınıflandırma yapmıştır. KNN-TSA'da sürekli değişkenler 0 ile 1 arasında normalize edilmiş ve bir eşik değer yardımı ile ikili değerlere dönüştürülmüştür.

Babalik ve ark. (2018a) kısıtlı optimizasyon problemlerini çözmek için TSA'yı geliştirmiştir. Temel TSA, kısıtlı optimizasyon problemlerini çözemez; bu nedenle bir kısıt mücadele tekniği olan Deb'in kuralları (Deb, 2000), TSA ile bütünleştirilmiş ve bu yaklaşım, CTSA olarak adlandırılmıştır. Deneyler için 13 kısıtlı maksimizasyon veya minimizasyon standart kıyas fonksiyonu ve mühendislik tasarım optimizasyon problemleri (PVDP ve kaynaklı kiriş tasarım probleminin (welded beam design problem, WBDP) versiyonları) kullanılmıştır. CTSA ortalama sonuçlara ve standart sapmalara göre çözüm kalitesi ve sağlamlığı bakımından son dönemde önerilmiş literatürdeki yöntemlerden daha iyi sonuçlar üretmiştir.

Horng ve Lin (2018), kısa süre içinde hesaplama yapabilen yakın optimal çözüm bulmak için sıralı optimizasyon TSA (ordinal optimization TSA, OOTSA) ismini verdikleri bir yaklaşım önermişlerdir. OOTSA, PSO, GA ve ES ile kıyaslanmıştır. Olasılıklı kısıtlı simülasyon optimizasyon problemi (probabilistic constrained simulation optimization problem) sürekli kısıtlı bir optimizasyon problemidir ve deneysel çalışmalar için kullanılmıştır. OOTSA'da, tohum üretim oranı (seed produced rate) ile kontrol edilen tohum üretim aşamasının katkısıyla hesaplama süresi önemli ölçüde azaltılmıştır.

Zhou ve ark. (2018) değişken uzunluklu (variable-length) TSA temelli bir rekabetçi yığılma (competitive agglomeration) (VTSA-CA) algoritması sunmuştur. İlk

önce deęişken uzunluklu TSA (VTSA) önerilmekte, daha sonra bu yaklaşım geliştirilmekte ve VTSA-CA olarak adlandırılmaktadır. VTSA-CA bulanık bir kümeleme algoritmasıdır ve arama işlemi sırasında küme sayısını otomatik olarak belirler. Faydalanma aşaması CA tarafından, keşif aşaması ise TSA tarafından yönetilmiştir. Deneyler için, pompalanmış depolama jeneratör motorunun (pumped storage generator motor) karmaşık dinamik davranışlarını tam olarak tanımlayan bir Takagi-Sugeno bulanık modeli kullanılır.

Ding ve ark. (2018), yapısal hasar tespit probleminin çözümü için önerdikleri yaklaşımlarının karşılaştırmalarını yaparken TSA'yı kullanmıştır. Bu çalışmada ST parametresi 0.3 olarak alınmış ve umut verici sonuçlar üretilmiştir.

Muneeswaran ve Rajasekaran (2018) TSA ile safra kesesine ait ultrason görüntülerinde segmentasyon için radyal temelli fonksiyon ağını eğitmiştir.

Cinar ve ark. (2018), çalışmalarında XOR mantıksal kapısı tabanlı ikili TSA'yı (XORTSA) önermişlerdir. XORTSA yeni ikili bireyler oluşturmak için XOR mantık kapısını kullanmaktadır. XORTSA, büyük ölçekli (250, 500 ve 1000 boyut) ikili optimizasyon problemleri üzerinde BPSO (Kennedy ve Eberhart, 1997) ile karşılaştırılmıştır. Test kümesi, beş kıyas fonksiyonu içermekte olup sonuçlar XORTSA'nın BPSO'dan daha iyi çözümler ürettiğini ortaya koymuştur.

Büyük ölçekli elektrik sistemlerinde optimal güç akışı (OPF) problemi El-Fergany ve Hasanien (2018) tarafından TSA ile çözülmüştür. Bu çalışmada TSA tasarımının ayrık ve sürekli kontrol deęişkenlerinin optimum deęerlerini saptamak için kullanılmıştır. Deneyler için IEEE 57 veri yolu ve 300 veri yolu sistemlerine ait farklı durumlar çalıştırılmıştır. Elde edilen sonuçlar TSA tabanlı OPF'nin çözüm kalitesini arttırdığını göstermiştir.

Cinar ve Kiran (2018b), çalışmalarında TSA'nın üç farklı (LogicTSA, SimTSA ve SimLogicTSA) ikili versiyonunu geliştirmiştir. Performans ölçümü için UFLP'nin kullanıldığı çalışmada küçük (16 boyut), orta (25 boyut), büyük (50 boyut) ve çok büyük (100 boyut) boyutlu UFLP'ler etkin bir şekilde çözülmüştür. SimLogicTSA, SimTSA ve LogicTSA'dan daha iyidir. SimLogicTSA, ABC (Kashan ve ark., 2012; Kiran ve Gunduz, 2013), PSO (Kennedy ve Eberhart, 1997; Yuan ve ark., 2009; Kiran ve Gunduz, 2013) ve DE'nin ikili varyantları (Kashan ve ark., 2013) ile karşılaştırılmıştır ve önerilen SimLogicTSA rekabetçi çözümler üreterek başarısını ispatlamıştır.

Ding ve ark. (2019b), TSA ve K-ortalamlar kümeleme algoritması ile yeni bir yapısal hasar tanımlama (structural damage identification) yaklaşımı önermiştir. Bu

yaklaşım C-TSA olarak adlandırılmıştır. K-ortalamlar kümeleme algoritması yeni tohumlar oluşturulurken kullanılır. Deneyler için standart sayısal kıyaslama fonksiyonları ve 61 barlık kafes yapı (61-bar truss structure) problemleri kullanılmıştır. TSA, global en iyi ABC (GABC) (Zhu ve Kwong, 2010), standart PSO'nun (SPSO) 2011 versiyonu (Zambrano-Bigiarini ve ark., 2013), geliştirilmiş global en iyi HS (IGHS) (El-Abd, 2013) ve global beyin fırtınası optimizasyonundan (global brain storm optimization, GBSO) (Hussain ve Haris, 2019) daha iyi performans göstermiştir. Bu yaklaşım endüstriyel bir uygulamada kullanılmıştır.

Oliva ve ark. (2019) görüntü segmentasyonu için TSA'yı kullanmıştır. Bu çalışmada, sınıf varyansı kriterleri arasındaki maksimum (Otsu) amaç fonksiyonu olarak kullanılmıştır. Önerilen yaklaşım, görüntü segmentasyonu için çok seviyeli eşikleme problemindeki diğer yöntemlerden daha iyi performans göstermiştir.

Li ve ark. (2019), TSA kullanılarak optimize edilmiş FIR filtresi ile resimlerin kenarını algılamaya çalışmıştır. Bu çalışmada resimler için yeni bir veri sıkıştırma yöntemi önerilmiş ve JPEG gibi iyi bilinen sıkıştırma teknikleriyle karşılaştırılmıştır.

Ding ve ark. (2019a), TSA'yı geliştirerek doğrusal olmayan histeretik parametre tanımlama problemini (nonlinear hysteretic parameter identification problem) üç tipik histeretik modelle (Bouc-Wen, kinematik sertleşmeli bilinear ve eşit verim kuvvetine sahip bilinear) çözmüştür. Bu yaklaşım geliştirilmiş TSA (Improved TSA, I-TSA) olarak adlandırılmıştır ve Levy uçuşu arama mekanizması ve yeni bir güncelleme denklemi ile temel TSA ile bütünleştirilmiştir. Bu çalışmadaki deneyler standart ve CEC 2014 (Liang ve ark., 2013) kıyas fonksiyonları üzerinde yapılmıştır. I-TSA, GABC (Zhu ve Kwong, 2010), SPSO (Zambrano-Bigiarini ve ark., 2013), GBSO (El-Abd, 2017) ve doğrusal popülasyon büyüklüğü küçültme mekanizması (linear population size reduction mechanism) ile uyumlu bir DE (Tanabe ve Fukunaga, 2014) ile kıyaslanmıştır. Ayrıca, I-TSA ile doğrusal olmayan bir tek serbestlik dereceli (single-degree-of-freedom) sistem ve doğrusal olmayan yedi katlı bir kayma inşaatı problemi (seven-storey shear building) çözülmüştür.

Beşkirli ve ark. (2019) büyük (20, 50, 100 ve 1000) boyutlu sürekli optimizasyon problemlerinin çözümü için TSA'yı geliştirmiştir. Bu yaklaşım değiştirilmiş TSA (modified TSA, MTSA) olarak adlandırılmıştır. Bu yöntemde tohum oluşturmada kullanılan rastgele ağaçlar turnuva seçim yöntemiyle seçilmiştir. Deneysel araştırmalar için 15 kıyas fonksiyonu ve görüntü segmentasyonu kullanılmıştır. MTSA, ABC (Akay ve Karaboga, 2012), PSO (Eberhart ve Kennedy, 1995), GSA (Rashedi ve ark., 2009)

salp sürü algoritması (salp swarm algorithm, SSA) (Mirjalili ve ark., 2017) çekirge optimizasyon algoritması (grasshopper optimization algorithm, GOA) (Saremi ve ark., 2017), çoklu evren optimizasyonu (multi-verse optimizer, MVO) (Mirjalili ve ark., 2016) algoritmaları ile karşılaştırılmıştır. Sonuçlar, MTSA'nın karşılaştırılan algoritmalarından daha iyi olduğunu göstermiştir.

Muneeswaran ve Rajasekaran (2019), yerel kontrast (local contrast) düzenli kontrast sınırlı adaptif histogram eşitleme (contrast limited adaptive histogram equalization)'nin geliştirme parametresini optimize etmek için TSA'yı kullanmıştır. Deneysel sonuçlar, TSA kullanan önerilen yöntemin, görüntüde bulunan yerel bilgileri koruduğunu ve ayrıca optimize kontrast geliştirmeyi sağladığını ortaya koymuştur.

Martin ve ark. (2019), TSA'yı karıştırılmış GWO (Martin ve ark., 2019), PSO (Eberhart ve Kennedy, 1995), GWO (Mittal ve ark., 2016), ABC (Karaboga ve Basturk, 2007), GA (Holland, 1992) ve SA (Fang ve ark., 2018) ile CEC2014 fonksiyonları üzerine karşılaştırmıştır. Sonuçlar, TSA'nın diğer algoritmaları geride bıraktığını göstermiştir. Ayrıca, bu çalışmada gerçek dünyadaki çok bantlı resimler için gürültü giderme ve ayrıştırma uygulaması (denoise and unmix real-world multispectral images application) yapılmıştır.

Eligüzel ve ark. (2019), P-medyan kuruluş yeri seçim probleminin çözümünde TSA'yı kullanmıştır. Küçük ve orta ölçekli P-medyan kuruluş yeri seçim problemi örneklerinden GAMS-CPLEX tarafından üretilmiş optimum çözümler ile kıyaslanan TSA tarafından elde edilmiş sonuçlar, TSA'nın optimum veya optimuma yakın sonuçlar ortaya koyduğunu göstermiştir.

Deperlioglu ve Kose (2019), destek vektör makinelerini (support vector machines, SVM) TSA ile eğiterek zeki bir medikal teşhis sistemi önermiştir. Bu yöntem SVM-TSA olarak isimlendirilmiştir. Dört farklı veri seti üzerinde yapılan testler, TSA ile eğitilen SVM'nin, varsayılan değerlerle çalışan SVM'den daha iyi sonuçlar ürettiğini ortaya koymuştur. ABC, CS, GA ve PSO ile eğitilmiş SVM'ler ile elde edilmiş sonuçlarla, SVM-TSA kıyaslanmıştır. Deneysel sonuçlar SVM-TSA'nın ABC'den sonra ikinci olduğunu ortaya koymuştur.

Cinar ve ark. (2019), ayrık tamsayılı optimizasyon problemlerini çözmek için TSA'nın yapısını değiştirerek takas, öteleme ve simetri operatörleri yardımıyla TSP'lerin çözümü için DTSA'yı önermiştir. DTSA, ABC, ACO, GA, SA, STA, BH, PSO ve varyantlarından daha iyi veya yarışmacı çözümler üreterek literatürdeki yerini almıştır.

Sahman ve ark. (2019), beş farklı sürekli kısıtsız gerçek dünya optimizasyon problemini çözmek için TSA'yı kullanmıştır. Farklı popülasyon büyüklükleri ve ST parametrelerinin sonuçlara etkisi tartışılmıştır.

Zhao ve ark. (2019), genelleştirilmiş regresyon sinir ağı (generalized regression neural network, GRNN) ile çok değişkenli bir sistemin modellemesini TSA aramalı ABC (tsABC-GRNN) ile gerçekleştirmiştir. Ham petrol fraksiyonlama işlemi (crude oil fractionation process) için tsABC-GRNN etkili çözümler üretmiştir.

Beşkirli (2019), yüksek boyutlu test fonksiyonlarının çözümü için TSA'ya turnuva seçim operatörünü ekleyerek G-TSA adını verdiği yeni bir yöntem önermiştir. 10 farklı kısıtsız sürekli kıyas fonksiyonunda yapılan testlerde G-TSA, TSA'dan daha iyi sonuçlar üretmiştir.

Durmuş (2019), beş farklı kaotik harita temelli TSA geliştirmiş ve önerdiği yöntemleri 24 kıyas fonksiyonu üzerinde test etmiştir.

Jiang ve ark. (2019), ST parametresi yerine keşif ve faydalanma sürecini yönetecek yeni bir yaklaşım önererek, bu yeni yöntemle EST-TSA ismini vermişlerdir. Deneysel sonuçlar özellikle yerel arama kapasitesinin arttığını ortaya koymuştur.

Chen ve ark. (2019), Li-Ion piller için kalan şarj süresinin hesaplanması için geliştirilmiş TSA yöntemini önermiştir. ST parametresini iterasyona bağımlı olarak değişken bir şekilde kullanan yöntemde, popülasyon çeşitliliğini artırmak ve yeni aday çözümler üretmek için DE/rand/2/bin yaklaşımı da kullanılmıştır. Önerilen TSA yaklaşımı GA'dan daha iyi sonuçlar üretmiştir.

Ding ve ark. (2019c), TSA'yı geliştirmek için Gauss arama denklemi ile yeni tohumlar üretmiştir. Önerdikleri bu yöntemle BGTSA (Bare-bones Gaussian TSA) ismini vermişlerdir. Kiran (Kiran, 2017b) tarafından önerilen WP yaklaşımı bu çalışmada da kullanılmıştır. Performans ölçümü için CEC05 kıyas seti ile yapısal hasar tanımlama problemi kullanılmıştır. BGTSA, TSA ve TSAWP'den daha iyi sonuçlar üretmiştir.

Sahman ve Cinar (2019), S ve V şekilli transfer fonksiyonlarıyla sürekli TSA'yı ikili TSA'ya çevirmiştir. Önerilen 8 farklı TSA yaklaşımı 15 UFLP üzerinde test edilmiştir. S şekilli transfer fonksiyonlarının V şekilli transfer fonksiyonlarından daha iyi sonuç ürettiği ortaya konmuştur.

Gungor ve ark. (2019), büyük boyutlu optimizasyon problemlerinin çözümü için ST parametresini ortadan kaldırıp DE'den esinlendikleri yeni tohum üretme stratejileri ile TSA'yı geliştirerek iTSA (improved TSA) ismini vermişlerdir. 30 boyutlu CEC2015

kıyas fonksiyonları ile yapılmış deneyler, iTSA'nın ABC, PSO, GA, PRSA, DE ve varyantlarından daha iyi sonuçlar üretmiştir.

Jiang ve ark. (2020), sinüs fonksiyonu yardımıyla TSA'nın başarısını artırarak STSA ismini verdikleri yeni bir yöntem önermişlerdir. 24 kıyas ve 2 mühendislik tasarım problemi üzerinde yapılan testler önerilen yöntemin başarısını ortaya koymuştur.

Yukarıda bahsedilen yöntemler isim, referans, yayımlandığı yıl ve tür bilgilerine göre kategorilendirilerek Çizelge 2.1'de toplu halde sunulmuştur.

Çizelge 2.1. TSA ile ilgili yapılmış çalışmalar

Referans	Tür	Yıl	İsim
(Cinar ve Kiran, 2016)	Sempozyum bildirisi	2016	-
(Cinar, 2016)	Yüksek lisans tezi	2016	-
(Zheng ve ark., 2016)	Dergi makalesi	2016	TSA-MPC
(Muneeswaran ve Rajasekaran, 2016)	Sempozyum bildirisi	2016	RBF-TSA
(Kiran, 2016)	Sempozyum bildirisi	2016	-
(Chen ve ark., 2017b)	Dergi makalesi	2017	-
(Cinar ve Kiran, 2017)	Sempozyum bildirisi	2017	-
(Muneeswaran ve Rajasekaran, 2017)	Sempozyum bildirisi	2017	-
(Kiran, 2017a)	Sempozyum bildirisi	2017	TSAWP
(Cinar ve ark., 2018)	Sempozyum bildirisi	2018	XORTSA
(Zhou ve ark., 2018)	Dergi makalesi	2018	VTSA VTSA-CA
(Ding ve ark., 2018)	Dergi makalesi	2018	-
(Muneeswaran ve Rajasekaran, 2018)	Kitap bölümü	2018	-
(Cinar ve Kiran, 2018b)	Dergi makalesi	2018	SimTSA LogicTSA SimLogicTSA
(El-Fergany ve Hasanien, 2018)	Dergi makalesi	2018	TSA-based OPF
(Babalik ve ark., 2018b)	Dergi makalesi	2018	CTSA
(Horng ve Lin, 2018)	Dergi makalesi	2018	OOTSA
(Cinar ve Kiran, 2018a)	Dergi makalesi	2018	-
(Jiang Xue-ying ve ark., 2018)	Dergi makalesi	2018	MPC-TSA
(Chen ve ark., 2018)	Sempozyum bildirisi	2018	KNN-TSA
(Ding ve ark., 2019a)	Dergi makalesi	2019	I-TSA
(Ding ve ark., 2019b)	Dergi makalesi	2019	C-TSA
(Oliva ve ark., 2019)	Kitap bölümü	2019	-
(Li ve ark., 2019)	Dergi makalesi	2019	-
(Beşkirli ve ark., 2019)	Dergi makalesi	2019	MTSA
(Muneeswaran ve Rajasekaran, 2019)	Sempozyum bildirisi	2019	-
(Martin ve ark., 2019)	Dergi makalesi	2019	-
(Eligüzel ve ark., 2019)	Dergi makalesi	2019	-
(Deperlioglu ve Kose, 2019)	Sempozyum bildirisi	2019	SVM-TSA
(Cinar ve ark., 2019)	Dergi makalesi	2019	DTSA
(Sahman ve ark., 2019)	Sempozyum bildirisi	2019	-
(Zhao ve ark., 2019)	Sempozyum bildirisi	2019	tsABC-GRNN
(Beşkirli, 2019)	Dergi makalesi	2019	G-TSA
(Durmuş, 2019)	Dergi makalesi	2019	-
(Jiang ve ark., 2019)	Dergi makalesi	2019	EST-TSA
(Chen ve ark., 2019)	Dergi makalesi	2019	-
(Sahman ve Cinar, 2019)	Dergi makalesi	2019	-
(Gungor ve ark., 2019)	Dergi makalesi	2019	iTSA
(Jiang ve ark., 2020)	Dergi makalesi	2020	STSA

Kaynak araştırması ve Çizelge 2.1'den de görüleceği üzere TSA az sayıda kontrol parametresi, uygulama kolaylığı, kolay anlaşılabilirlik ve açıklanabilirlik gibi özelliklerinden dolayı hem araştırmacılar, hem de uygulayıcılar tarafından ilgi görmüş ve kısa zamanda birçok alana uygulanıp çeşitli akademik çalışmalarda kullanılmıştır.



3. MATERYAL VE YÖNTEM

Bu bölümde TSA, CTSA, SimTSA, LogicTSA, SimLogicTSA ve DTSA ile bu yöntemleri test etmek için kullanılan optimizasyon problemleri olan sürekli kısıtlı nümerik kıyas fonksiyonları, sürekli kısıtlı mühendislik tasarım problemleri, UFLP ve TSP anlatılmaktadır.

3.1. Ağaç-Tohum Algoritması

TSA doğa esinli, popülasyon tabanlı, sürü zekası kategorisinde bulunan, stokastik, metasezgisel optimizasyon algoritmasıdır. TSA sürekli optimizasyon problemlerinin çözümü için ağaçlar ile tohumları arasındaki ilişkiden yola çıkılarak Kiran (2015a) tarafından önerilmiştir. Ağaçların üreme ve yayılma organı tohumdur. Tohumlar yapılarına uygun olarak taşınır. Bazı tohumlar kuşların tüylerine, hayvanların kürklerine ve giysilere tutunarak farklı yerlere taşınır. Böylece taşınılan noktada uygun yetiştirme ortamı bulan tohum büyüyerek ağaç olur.

Yeryüzü, optimizasyon problemleri için araştırma uzayı olarak düşünüldüğünde, çözüm için üretilen sonuçları ağaçlar ve tohumlar olarak düşünebiliriz. Araştırma sürecinde her ağacın kendi tohumları arasında açgözlü yaklaşım uygulanır ve en iyi tohum belirlenir. En iyi tohum, kendi ağacından daha iyiye artık ağaç kurur ve en iyi tohum o ağacın yerini alır. Algoritmanın iki ana unsuru vardır, keşif ve faydalanma. Keşif aşamasında arama uzayına dağılmış rastgele noktalarda yerleşmiş ağaçlar bulunmaktadır, faydalanma aşamasında ise ağaçlar ile aynı özellikteki tohumlar kullanılmaktadır. Her ağacın tohum sayısı birden az olmamak kaydıyla rastgele olarak belirlenmiştir. Rastgele tohum sayısının toplam ağaç sayısının %10'u ile %25'i arasında olması önerilmiştir. Araştırma eğilimi (search tendency, ST) parametresi ise tohum oluşumu sırasında en iyi ağaç ile rastgele seçilecek ağaç arasında seçim yapmak için kullanılmaktadır. ST parametresi ile algoritmanın yerel bir yoğunlaşma içine girmesi amaçlanmış, optimum veya optimuma yakın değerlere yakınsama yapması sağlanmıştır. Sphere minimizasyon problemi (Denklem 3.1) üzerinde TSA'nın çalışması aşağıda anlatılmıştır.

$$\min f(\vec{X}) = \sum_{j=1}^D X_j^2 \quad (3.1)$$

Burada X fonksiyona gönderilen vektörü, D vektörün boyutunu, j boyutu temsil etmektedir. Amaç, fonksiyonu minimum yapan değer veya değerleri bulmaktır.

TSA başlatılmadan önce N , ST , çözülecek problemin boyutu (D), sonlandırma koşulu (maksimum fonksiyon değerlendirme sayısı, maxFEs) belirlenir. İlk adımda

arama uzayına D boyutlu N adet ağaç sınırlara uygun olarak Denklem 3.2 yardımıyla dağıtılır.

$$T_{i,j} = L_{j,min} + r_{i,j}(H_{j,max} - L_{j,min}) \quad (3.2)$$

Burada, $T_{i,j}$ i.ağacın j.boyutunu, $L_{j,min}$ j.boyut için arama uzayının alt sınırını, $H_{j,max}$ j.boyut için arama uzayının üst sınırını, $r_{i,j}$ i.ağacın j.boyutu için 0-1 arasında üretilen tekdüze (uniform) rastgele sayıyı temsil etmektedir. Arama uzayında rastgele ağaçlar oluşturulduktan sonra en iyi ağaç Denklem 3.3 yardımıyla belirlenir.

$$B = ArgMin\{f(T_i)\} \quad i = 1,2, \dots, N \quad (3.3)$$

Burada B en iyi ağacı, N toplam ağaç sayısını, i mevcut ağacı, T ise popülasyondaki tüm ağaçları belirtmektedir. En iyi ağaç seçilirken amaç fonksiyon değeri en küçük olan (minimizasyon problemi çözüldüğü varsayıldığında) ağaç, en iyi ağaç olarak belirlenir. En iyi ağaç tohum üretim sürecinde kullanılacağından bu adımda belirlenmesi gerekir. Araştırmanın, popülasyondan seçilen rastgele seçilecek ağaca doğru mu, yoksa en iyi ağaca doğru mu yapılacağına ST parametresi ile karar verilir. Tohum oluşumu sırasında ST parametresine göre Denklem 3.4 veya Denklem 3.5 kullanılır. ST parametresi 0-1 arasında bir sayı olarak belirlenir. 0-1 arasında üretilen rastgele bir sayı eğer ST parametresinden küçükse tohum üretimi için Denklem 3.4, büyükse ve eşitse Denklem 3.5 kullanılır. ST parametresi büyüdükçe faydalanma (exploitation) ve hızlı yakınsama sağlanırken, küçüldükçe yavaş yakınsama ve keşif (exploration) sağlanmış olur.

$$S_{i,k,j} = T_{i,j} + \alpha_{i,j}x(B_j - T_{r,j}) \quad (3.4)$$

$$S_{i,k,j} = T_{i,j} + \alpha_{i,j}x(T_{i,j} - T_{r,j}) \quad (3.5)$$

Burada $S_{i,k,j}$ i.ağacın k. tohumunun j.boyutunu, $T_{i,j}$ i.ağacın j.boyutunu, $\alpha_{i,j}$ -1,1 arasında üretilen rastgele bir değeri, B_j mevcut iterasyona kadar elde edilen en iyi ağacın j.boyutunu, $T_{r,j}$ ise r.ağacın j.boyutunu temsil etmektedir. Rastgele seçilen ağaç (r) mevcut ağaçtan (i) farklı bir ağaç olmalıdır.

Tohum üretim sürecinde arama uzayının sınırlarını aşan tohumlar Denklem 3.6 ile kontrol edilerek kendilerine en yakın sınır değerine çekilir.

$$S_{i,k,j} = \begin{cases} H_{j,max}, & \text{eğer } S_{i,k,j} > H_{j,max} \\ L_{j,min}, & \text{eğer } S_{i,k,j} < L_{j,min} \end{cases} \quad (3.6)$$

Burada $S_{i,k,j}$ i.ağacın k. tohumunun j.boyutunu, $L_{j,min}$ j.boyut için arama uzayının alt sınırını, $H_{j,max}$ j.boyut için arama uzayının üst sınırını belirtmektedir.

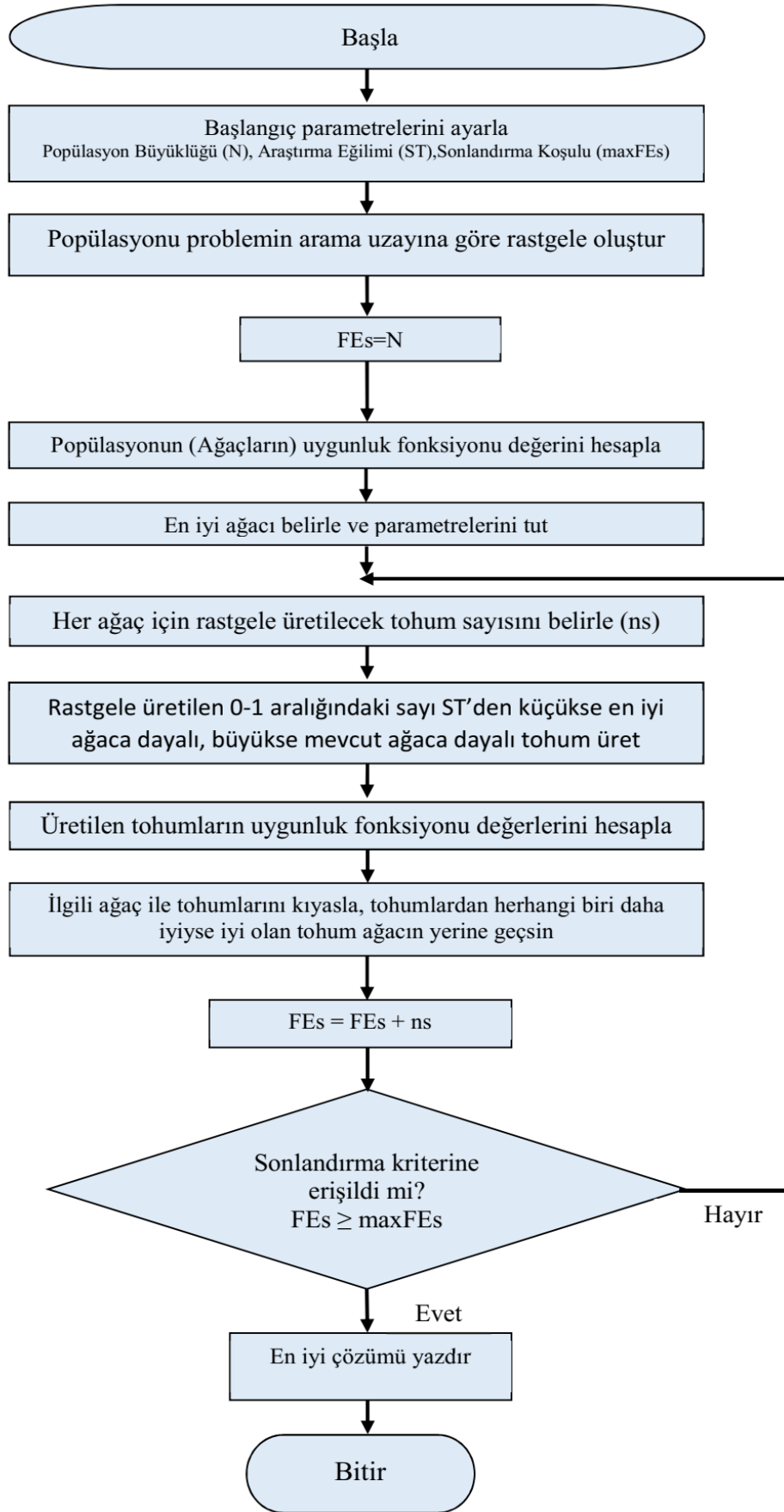
Başlangıçta belirlenmiş olan maxFEs değerine erişilinceye kadar tohum üretim süreci devam eder ve kendi ağacından daha iyi olan tohumların ağaç olması suretiyle ilerleme sağlanır. TSA’da her ağaca ait tohum sayısı farklı olduğundan sabit bir iterasyon sayısı yerine maxFEs sonlandırma kriteri olarak kullanılmalıdır. Bu şekilde yapılmadığı takdirde adil bir kıyaslama için gerekli ortam sağlanmamış olur. Kısıtsız sürekli kıyas fonksiyonları ile yapılan testlerde ST parametresi 0.1 olarak önerilmiştir. Problemin boyutu büyüdükçe algoritmanın iyi sonuçlar üretmediği de açıklanmıştır. Bunun sebebinin boyutun artmasına bağlı olarak arama uzayının üstel artması olduğu belirtilmiştir. TSA, unimodal (tek optimum değeri olan) fonksiyonları çözmekte, multimodal (çok yerel optimum değeri olan) fonksiyonları çözmeye göre daha başarılıdır. TSA literatürde yaygın olarak kullanılan 24 kıyas fonksiyonunu ve eşikleme problemini çözmeye kabul edilebilir ve karşılaştırılabilir düzeyde sonuçlar üretmek başarısını ortaya koymuştur (Kiran, 2015a). Şekil 3.1’de TSA’nın sözde kodu, Şekil 3.2’de ise TSA’nın akış şeması verilmiştir.

```

Algoritmanın başlatılması
Popülasyon sayısı belirlenir (N)
Araştırma eğilimi parametresi belirlenir (ST)
Problemin boyutu belirlenir (D)
Sonlandırma koşulu belirlenir (maxFEs)
N adet D boyutlu ağaç Denklem 3.2 ile oluşturulur
Üretilen ağaçların amaç fonksiyonu değeri hesaplanır
En iyi ağaç seçilir, parametreleri alınır (B)
FEs=N olarak atanır ve tohumlarla arama işlemi başlatılır
WHILE FEs <=maxFEs
    FOR tüm ağaçlar için,
        İlgili ağaç için kaç tohum üretileceği rastgele olarak belirlenir (ns)
        FOR ilgili ağacın ns adet tohumu için
            FOR ilgili tohumun j=1...D boyutu için
                IF (rand<ST)
                    İlgili boyut Denklem 3.4 ile hesaplanır
                    Sınır aşımı varsa Denklem 3.6 uygulanır
                ELSE
                    İlgili boyut Denklem 3.5 ile hesaplanır
                    Sınır aşımı varsa Denklem 3.6 uygulanır
                END IF
            END FOR
        İlgili tohumun amaç fonksiyonu değeri hesaplanır
        FEs=FEs+1 şeklinde FEs sayacı bir artırılır
        END FOR
    En iyi tohum seçilir ve kendi ağacıyla karşılaştırılır
    Eğer en iyi tohum, ağacından daha iyiyse tohum ağaç olur
    END FOR
En iyi ağaç belirlenir
Eğer mevcut en iyi ağaç, şimdiye kadarki en iyi ağaçtan daha iyiyse yer değiştirir ve parametreleri aktarılır
Sonlandırma kıstasına erişilinceye kadar devam eder
END WHILE
En iyi çözüm yazdırılır

```

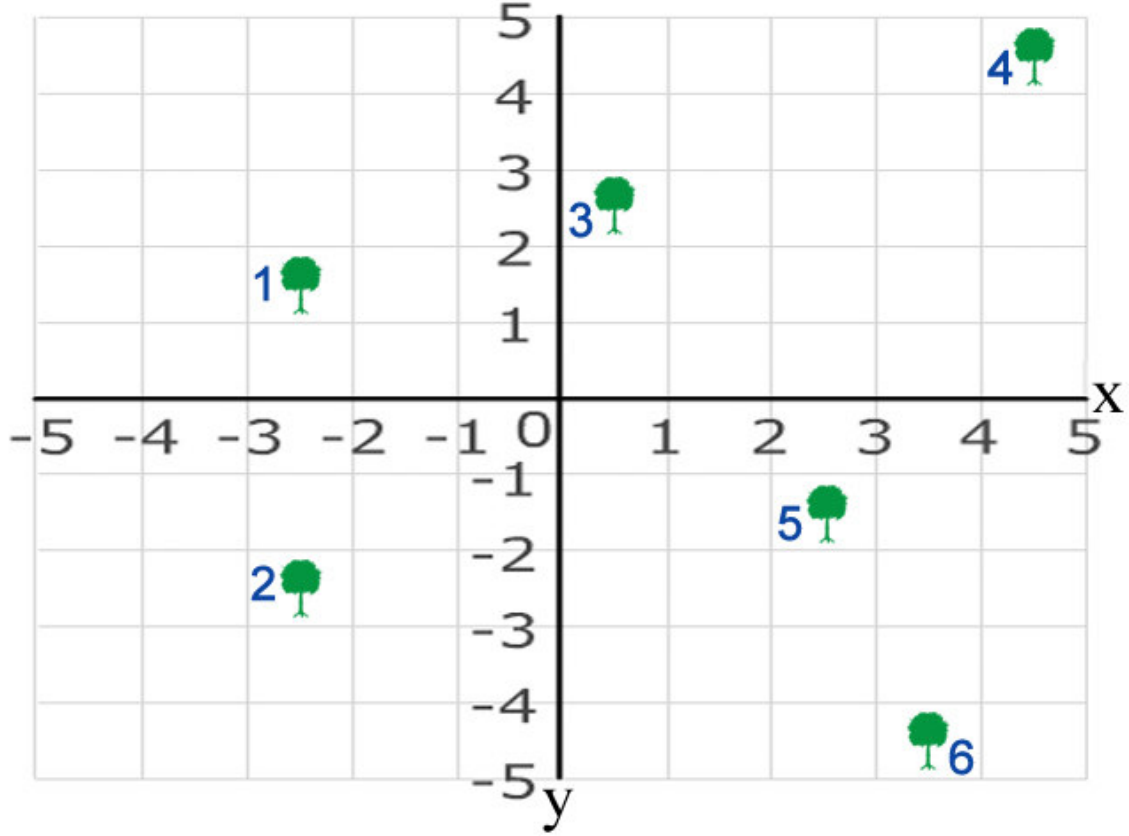
Şekil 3.1. Ağaç-Tohum algoritması (TSA)’nın sözde kodu



Şekil 3.2. Ağaç-Tohum algoritması (TSA)'nın akış şeması

Arama uzayının $-5,+5$ aralığında olduğu ve değişimlerin tam sayı seviyesinde ilerlediği 2 boyutlu ayırık Sphere fonksiyonunu $\sum_{j=1}^2 x_j^2$ minimize etmek için TSA'nın çalışması adım adım görsel olarak izah edilmiştir.

$N=6$ olarak seçilmiştir ve ağaçlar rastgele oluşturulmuştur.



Şekil 3.3. Başlangıç aşamasında arama uzayındaki ağaçların konumları

Şekil 3.3'te görülen ağaçların konumlarının sayısal değerleri Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Başlangıç aşamasında oluşturulan 6 ağacın konumlarının sayısal değerleri

Ağaç İsmi	1.Boyut (x eksen)	2.Boyut (y eksen)
1.Ağaç	-3	2
2.Ağaç	-3	-2
3.Ağaç	1	3
4.Ağaç	5	5
5.Ağaç	3	-2
6.Ağaç	4	-5

Ağaçların amaç fonksiyonu değerleri hesaplanmış ve Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2. Başlangıç aşamasında oluşturulan 6 ağacın amaç fonksiyonu değerleri

Ağaç İsmi	1.Boyut (x eksen)	2.Boyut (y eksen)	Amaç Fonksiyonu Değeri
1.Ağaç	-3	2	13
2.Ağaç	-3	-2	13
3.Ağaç	1	3	10
4.Ağaç	5	5	50
5.Ağaç	3	-2	13
6.Ağaç	4	-5	41

En iyi ağaç değeri minimizasyon problemi olduğu için 3.ağaç olarak belirlenmiştir. Görüleceği üzere 1. 2. ve 5. ağaçlar farklı konumlarda olsalar dahi amaç fonksiyonu değeri olarak algoritma açısından aynı değerdedir. Bu durum popülasyon çeşitliliğinin amaç fonksiyonu bazlı takibi sırasında dikkat edilmesi gereken unsurlardandır. En kötü ağaç ise 4.ağaçtır.

Başlangıç aşamasında belirlenmiş olan maxFEs değerine erişilinceye kadar devam edecek olan iterasyon süreci başlatılır.

Her ağaca ait kaç adet tohum üretileceği popülasyon büyüklüğünün %10'u ile %25'i arasında bir sayı olarak önerildiğinden bu örnekte 1 alınarak işlemler yapılmıştır. Tohum üretme aşamasında kullanılan iki farklı denklemin seçimi için ST parametresi kullanıldığından bu anlatımda 1.ağaç için Denklem 3.4, 3.ağaç için Denklem 3.5. kullanılmıştır.

1.ağaç için rastgele 1 tohum üretilmiş ve bu üretim için Denklem 3.4 kullanılmıştır. Rastgele ağaç 3 seçilmiş olup ilk boyut aşağıdaki şekilde üretilmiştir. -1 ve +1 aralığındaki rastgele üretilen α değeri ise bu adımda 0.5 alınmıştır.

$$S_{(1,1,1)} = T_{(1,1)} + \alpha \times (B_{(1)} - T_{(3,1)})$$

$$S_{(1,1,1)} = -3 + 0.5 \times (1 - 1)$$

$$S_{(1,1,1)} = -3$$

1. ağacın 1. boyutunun değeri (-3) ile üretilmiş tohumun ilk boyutunun değeri aynı çıktı. Bunun sebebi denklemden de görüleceği üzere en iyi ağaç ile seçilen rastgele ağacın değerinin aynı olmasıdır. 1.ağacın 1.tohumunun 2. boyutunun üretilmesi aşamasında rastgele ağaç 6 olarak seçilmiş, α değeri ise 1 alınmıştır.

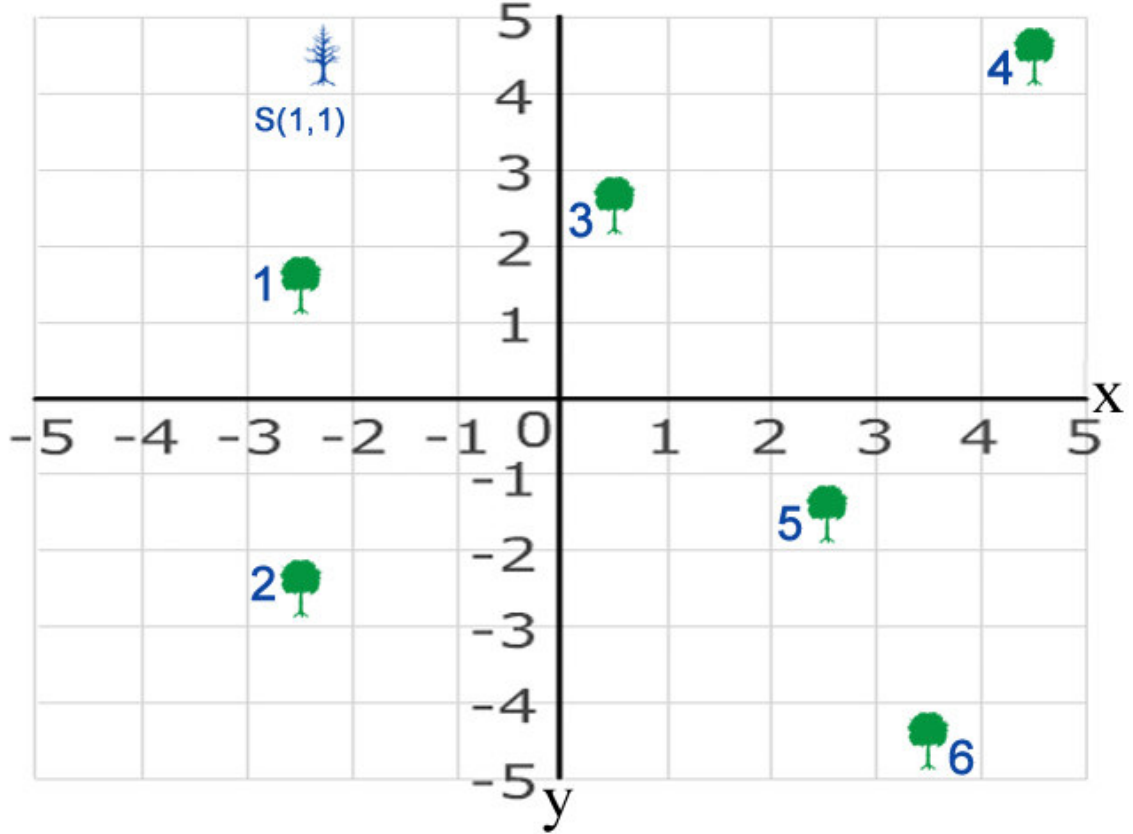
$$S_{(1,1,2)} = T_{(1,2)} + \alpha \times (B_{(2)} - T_{(6,2)})$$

$$S_{(1,1,2)} = 2 + 1 \times (3 - (-5))$$

$$S_{(1,1,2)} = 10$$

$$S_{(1,1,2)} = 5$$

1.ağacın 2. boyutunun değeri 10 olarak hesaplandı. Bu değer arama uzayının dışına taşığından 5 olarak (arama uzayının en yakın sınırı) güncellenmiştir. Böylece $S_{(1,1)} = (-3,5)$ konumuna yerleşmiş, amaç fonksiyonu değeri ise $-3^2 + 5^2 = 34$ olarak hesaplanmıştır. 1.ağacın mevcut amaç fonksiyonu değeri 13 olduğundan üretilmiş tohumun amaç fonksiyon değeri 34 olduğundan büyüyerek ağaç olma imkânını kaybetmiştir. Şekil 3.4'te arama uzayının anlık durumu gösterilmiştir.



Şekil 3.4. 1.ağacın 1.tohumu oluştuktan sonra arama uzayının durumu

Araştırma süreci çok sayıda rastgelelik içerdiğinden konunun anlaşılması için verilen bu örneği daha iyi bir tohum üretmesi beklendiğinden en iyi ağaç olan 3.ağaca ait bir tohum üreterek sonlandıracağız.

3.ağaç için rastgele 1 tohum üretilmiş ve bu üretim için Denklem 3.5 kullanılmıştır. Rastgele ağaç 2 olarak seçilmiş olup ilk boyut aşağıdaki şekilde üretilmiştir. -1 ve +1 aralığında rastgele üretilen α değeri ise bu adımda 1 alınmıştır.

$$S_{(3,1,1)} = T_{(3,1)} + \alpha \times (T_{(3,1)} - T_{(2,1)})$$

$$S_{(3,1,1)} = -3 + 1 \times (1 - (-3))$$

$$S_{(3,1,1)} = 1$$

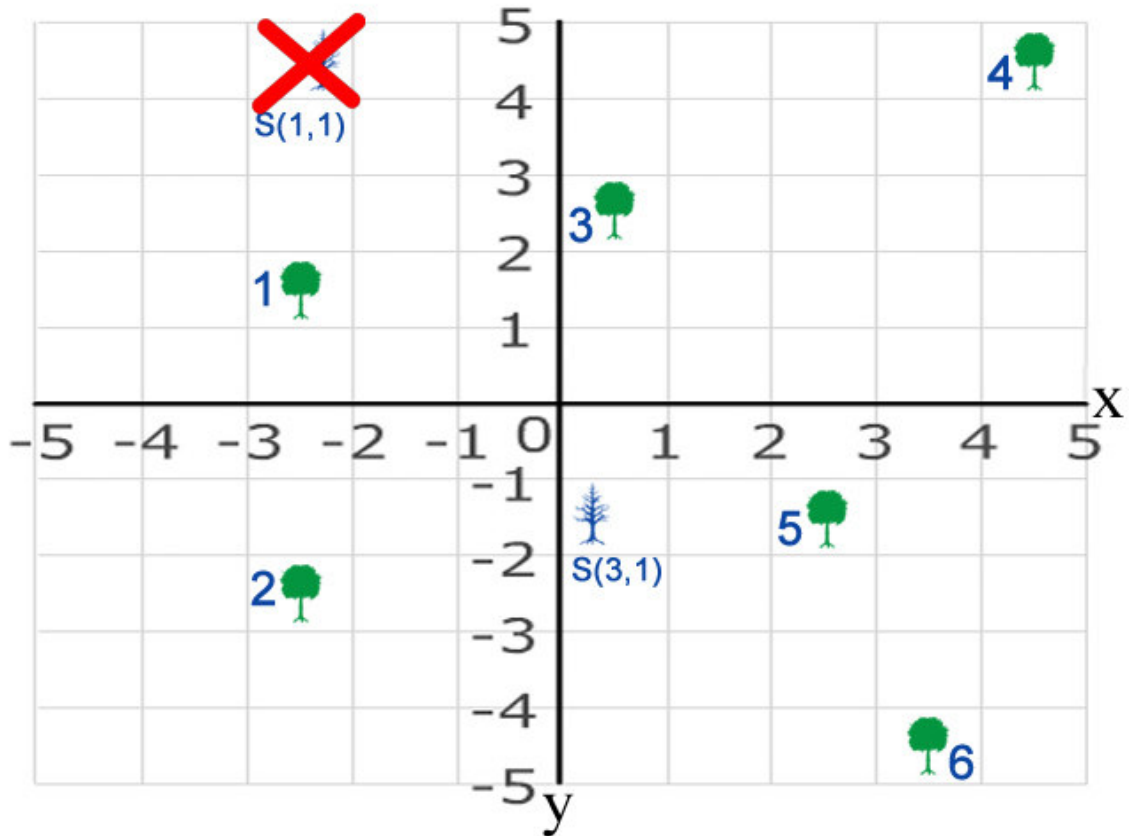
3. ağacın 1.tohumunun 1.boyutunun değeri 1 olarak üretilmiştir. 3.ağacın 1.tohumununun 2. boyutunun üretilmesi aşamasında rastgele ağaç 5 olarak seçilmiş, α değeri ise -1 alınmıştır.

$$S_{(3,1,2)} = T_{(3,2)} + \alpha \times (T_{(3,2)} - T_{(5,2)})$$

$$S_{(3,1,2)} = 3 + (-1) \times (3 - (-2))$$

$$S_{(3,1,2)} = -2$$

3. ağacın 1.tohumununun 2.boyutunun değeri -2 olarak hesaplanmıştır. Bu değer arama uzayının dışına taşmadığından arama uzayı sınırlandırma sürecine maruz kalmamıştır. Böylece $S_{(3,1)} = (1, -2)$ konumuna yerleşmiş, amaç fonksiyonu değeri ise $1^2 + (-2)^2 = 5$ olarak hesaplanmıştır. 3.ağacın mevcut amaç fonksiyonu değeri 10 olduğundan, üretilmiş tohumun amaç fonksiyon değeri 5 olduğundan, büyüyerek ağaç olma imkânına kavuşmuş ve bundan sonraki iterasyonlarda kendisini de en iyi ağaç olarak kaydettirmiştir.



Şekil 3.5. 3.ağacın 1.tohumu oluştuktan sonra arama uzayının durumu

Şekil 3.5'te 3.ağacın 1.tohumu oluştuktan sonra arama uzayının durumu gösterilmiş olup, Çizelge 3.3'te en iyi ağaç güncellendikten sonra 6 ağacın konum ve

amaç fonksiyon değerleri gösterilmiştir. Süreç bu şekilde sonlandırma kriterine erişilinceye kadar devam ettirilerek optimizasyon işlemi gerçekleştirilmektedir.

Çizelge 3.3. En iyi ağaç güncellendikten sonra 6 ağacın konum ve amaç fonksiyon değerleri

Ağaç İsmi	1.Boyut (x eksen)	2.Boyut (y eksen)	Amaç Fonksiyonu Değeri
1.Ağaç	-3	2	13
2.Ağaç	-3	-2	13
3.Ağaç	1	-2	5
4.Ağaç	5	5	50
5.Ağaç	3	-2	13
6.Ağaç	4	-5	41

3.2. Kısıtlı Ağaç-Tohum Algoritması

Kısıtlı optimizasyon Denklem 3.7 ile tanımlı eşitsizlik ve Denklem 3.8 ile tanımlı eşitlik sınırlamalarını dikkate alarak Denklem 3.9'daki tanımlı amaç fonksiyonunu minimize eden \vec{x} değerini bulma işlemidir (Akay, 2009).

$$g_j(\vec{x}) \leq 0, j = 1, \dots, q \quad (3.7)$$

$$h_j(\vec{x}) = 0, j = q + 1, \dots, m \quad (3.8)$$

$$\min f(\vec{x}), \vec{x} = (x_1, \dots, x_n) \in R^n \quad (3.9)$$

$$l_i \leq x_i \leq u_i, i = 1, \dots, n \quad (3.10)$$

Burada f amaç fonksiyonu, n boyutlu R^n 'de bulunan S araştırma uzayında $S \subseteq R^n$ tanımlıdır. Karar değişkenlerinin arama uzayı içerisindeki alt ve üst sınırları Denklem 3.10'da tanımlanmıştır. Uygun çözümlerin üretilebileceği kabul edilebilir bölge $F \subseteq S$, m tane ($m \geq 0$) kısıt fonksiyonuna sahiptir. \vec{x} değeri kabul edilebilir bölgede $\vec{x} \in F \subseteq S$ tanımlıdır. $\vec{x} \in F$ ise $g_k(\vec{x}) = 0$ şartını sağlayan g_k kısıtları \vec{x} 'de aktif olarak isimlendirilir. h_j eşitlik kısıtları ise arama uzayının her bölgesinde aktiftir.

3.2.1. Kural tabanlı kısıtlı TSA

Temel TSA mevcut durumuyla kısıtlı optimizasyon problemlerine çözüm üretemediğinden Deb'in kuralları seçim yönetimi kullanılarak modifiye edilmiştir. Deb'in kuralları ile araştırma kabul edilebilir bölgeye doğru kaydırılmaktadır. Deb'in kuralları iki çözümün aşağıdaki kriterlere göre yarıştırdığı bir turnuva seçim operatörü gibi çalışmaktadır.

- 1- Kabul edilebilir bölgedeki her bir çözüm ($ihlal_i \leq 0$) kabul edilebilir olmayan bölgedeki çözüme ($ihlal_j > 0$) tercih edilir. (çözüm i seçilir)
- 2- Her iki çözümde kabul edilebilir bölgedeyseniz ($ihlal_i \leq 0, ihlal_j \leq 0$) o zaman daha küçük maliyet değerine sahip çözüm tercih edilir. ($f_i < f_j$ ise çözüm i seçilir)
- 3- Her iki çözümde kabul edilebilir olmayan bölgede ise ($ihlal_i > 0, ihlal_j > 0$) daha küçük ihlal miktarına sahip olan çözüm tercih edilir. ($ihlal_i \leq ihlal_j$ ise çözüm i seçilir)

En iyi ağaç ve en iyi tohum seçilirken açgözlü seçim yöntemi yerine yukarıdaki seçim yaklaşımı kullanılır.

3.3. İkili Ağaç-Tohum Algoritmaları

Temel TSA sürekli arama uzayındaki karar değişkenleri ile çalışırken, ikili TSA'lar ikili arama uzayında çalışacak şekilde yapılandırılmıştır. Başlangıç aşamasında problemin her boyutu için 0 ya da 1 rastgele sayılarından birisi üretilerek popülasyon oluşturulur. Bunun için Denklem 3.13 yardımıyla her adımda üretilen 0-1 aralığındaki rastgele sayı 0.5'ten küçükse 0 büyükse ve eşitse 1 değeri üretilmiş olur.

$$P(\text{Açık}) = P(X_{i,j} = 1) = p \quad (3.11)$$

$$P(\text{Kapalı}) = P(X_{i,j} = 0) = 1 - p \quad (3.12)$$

$$X_{i,j} = \begin{cases} 0 & \text{eğer } (r_{i,j} < p) \\ 1 & \text{eğer } (r_{i,j} \geq p) \end{cases} \quad (3.13)$$

Burada p ikili TSA'lar için 0.5 olarak alınan olasılık değeridir, $P(\text{Açık})$ 1 değerini alan diğer bir anlamıyla açık olan tesisleri (Denklem 3.11), $P(\text{Kapalı})$ 0 değerini alan yani kapalı olan tesisleri (Denklem 3.12) belirtir. Başlangıç aşamasında açık ve kapalı olan tesis sayısının olabildiğince eşit olması bu şekilde sağlanmış olur.

Başlangıç popülasyonu oluşturulduktan sonra her bir ağacın tohumları, temel TSA'daki denklemlerin bağlamından koparılmadan oluşturulmuş yeni ikili tohum oluşturma denklemleri kullanılarak oluşturulur. Bu denklemler her algoritmayı ayrı ayrı anlattığımız alt bölümlerde verilmiştir.

3.3.1. Benzerlik tabanlı ikili TSA

İkili bireylerin arasındaki benzerliğin ölçümü, birbirlerinden ne kadar yakında veya uzakta olduklarını göstermektedir. İkili gösterim için uygulanabilecek 76 benzerlik ölçüm tekniği Choi ve ark. (2010)'nın çalışmasında detaylı bir şekilde incelenmiştir. İkili gösterime sahip bireyler arasındaki benzerlik derecesini belirlemek için en yaygın kullanılan genel amaçlı benzerlik ölçümlerinden biri Jaccard (1901)'in benzerlik katsayısıdır. Ayrıca bu benzerlik ölçümü (Kashan ve ark., 2012), (Kashan ve ark., 2013), (Hancer ve ark., 2015) çalışmalarında kullanıldığından SimTSA algoritmasında Denklem 3.14'te verilmiş olan bu benzerlik ölçüm tekniği kullanılmıştır.

$X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ ve $X_j = [x_{j1}, x_{j2}, \dots, x_{jd}]$ benzerliği ölçülecek olan D boyutlu ikili bireylerdir.

$$\text{Benzerlik}(X_i, X_j) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \quad (3.14)$$

$$M_{11} = \sum_{d=1}^D I_d, I_d = \begin{cases} 1, & \text{eğer } x_{id} = 1 \text{ ve } x_{jd} = 1 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.15)$$

$$M_{01} = \sum_{d=1}^D I_d, I_d = \begin{cases} 1, & \text{eğer } x_{id} = 0 \text{ ve } x_{jd} = 1 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.16)$$

$$M_{10} = \sum_{d=1}^D I_d, I_d = \begin{cases} 1, & \text{eğer } x_{id} = 1 \text{ ve } x_{jd} = 0 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.17)$$

$$M_{00} = \sum_{d=1}^D I_d, I_d = \begin{cases} 1, & \text{eğer } x_{id} = 0 \text{ ve } x_{jd} = 0 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.18)$$

$$M_{11} + M_{01} + M_{10} + M_{00} = D \quad (3.19)$$

Burada kıyaslanan iki boyutun değeri de 1 olduğu zaman (Denklem 3.15) M_{11} değeri bir artılır, iki boyutun değeri de 0 olduğu zaman (Denklem 3.18) M_{00} bir artılır. Birinci 1 ikinci 0 olduğu durumda (Denklem 3.17) M_{10} , birinci 0 ikinci 1 olduğu durumda (Denklem 3.16) ise M_{01} değeri bir artılır. Sonuçta elde edilen $M_{11} + M_{01} + M_{10} + M_{00}$ toplamı Denklem 3.19'da belirtildiği gibi D 'ye eşit olmak zorundadır. Benzerlik hesabı yapıldıktan sonra Denklem 3.20 ile benzemezlik hesabı yapılır.

$$\text{Benzemezlik}(X_i, X_j) = 1 - \text{Benzerlik}(X_i, X_j) = 1 - \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \quad (3.20)$$

Örneğin benzerlik değeri 0.7 çıkmış iki bireyin benzemezlik değeri 0.3 olarak bulunacaktır. Bu değerler ile bireylerin birbirleriyle ne kadar benzeştikleri veya

benzeşmedikleri yüzde olarak belirlenmiş olur. Burada iki bireyin tüm değerleri 0 olduğu durumda bu benzerlik ölçütü çalışmamaktadır, tezde kullanılan UFLP'ler için en az bir tesisin açık olması kısıtından dolayı en azından bir boyutun değeri 1 olmak zorunda olduğundan bu ölçütün çalışmasında herhangi bir problem olmamaktadır.

Aşağıdaki anlatılan yeni aday çözüm üretme yöntemi (Kashan ve ark., 2012; Kashan ve ark., 2013) çalışmalarından alınmıştır.

SimTSA'da yeni bir tohum üretmek için temel TSA'ya benzer şekilde mevcut ağaç, en iyi ağaç veya rastgele seçilen komşu bir ağaca ait değerler kullanılmaktadır. Her bir tohum oluşturulurken 0-1 aralığındaki rastgele bir sayı ST parametresi ile kıyaslanır. Bu sayı ST'den küçükse Denklem 3.21, büyükse Denklem 3.22 ile tohum üretme süreci devam ettirilir.

$$\text{Benzemezlik}(S_k, T_i) \sim \varphi \text{Benzemezlik}(T_i, B) \quad (3.21)$$

$$\text{Benzemezlik}(S_k, T_i) \sim \varphi \text{Benzemezlik}(T_i, T_r) \quad (3.22)$$

Burada φ pozitif rastgele ölçekleme faktörünü, T_i mevcut ağacı, S_k mevcut tohumu, T_r mevcut ağaç harici rastgele seçilen bir ağacı ve B en iyi ağacı temsil etmektedir. \sim işleminin kullanılmasının sebebini, Denklem 3.21 üzerinden yorumlayacak olursak, mevcut ağaç ile en iyi ağacın benzemezlik değerine en yakın olacak şekilde mevcut ağaç ile mevcut tohumun benzemezlik değerini üretmek istememizdir. φ parametresi aradaki benzemezlik farkını ortaya koymaktadır.

Yeni bir tohum üretmek için aşağıdaki üç parametrenin değerinin bulunması gerekmektedir:

M_{11} : Hem üretilecek tohumda hem mevcut ağaçta 1 olan boyut sayısı

M_{10} : Üretilecek olan tohumda 1, mevcut ağaçta 0 olan boyut sayısı

M_{01} : Üretilecek olan tohumda 0, mevcut ağaçta 1 olan boyut sayısı

Bu değerleri bulmak için önerilen tam sayılı programlama modelinin matematiksel modeli Denklem 3.23-3.26 arasında, uygun çözüm uzayının gösterimi ise Şekil 3.6'da verilmiştir.

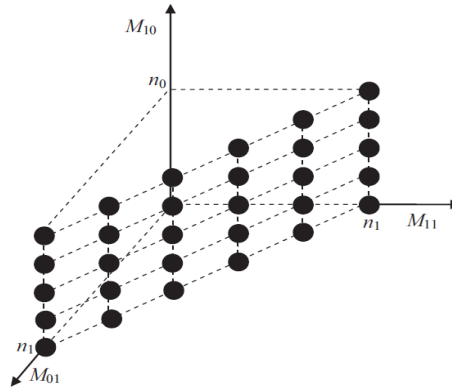
$$\min \left| 1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} - A \right| \quad (3.23)$$

$$M_{11} + M_{01} = n1 \quad (3.24)$$

$$M_{10} \leq n0 \quad (3.25)$$

$$M_{11}, M_{10}, M_{01} \geq 0 \text{ ve tam sayı olmak zorundadır} \quad (3.26)$$

Burada n_0 değeri 0 olan boyutların sayısını, n_1 değeri 1 olan boyutların sayısını, A benzemezlik oranı ile pozitif rastgele ölçkleme faktörünün çarpımını göstermektedir. Tam sayılı programlama modelinin çalıştırılması ile M_{11} , M_{10} , M_{01} değerleri belirlenir ve Şekil 3.7’de anlatıldığı şekilde yeni ikili çözüm üretim algoritması (new binary solution generator, NBSG) (Kashan ve ark., 2012) ile tohum üretimi gerçekleştirilir.



Şekil 3.6. Tam sayılı programlama modelinin uygun çözüm uzayının gösterimi (Kashan ve ark., 2012)

Adım 1. A değeri φ . Benzemezlik (X_i, X_k) işlemi ile hesaplanır ve Denklem 3.23-3.26 arasındaki tam sayılı programlama modelinin çözülmesiyle optimum M_{01} , M_{10} ve M_{11} değerleri tespit edilerek D boyutlu içeriği 0 değeri ile doldurulmuş çözüm vektörü (V_i) oluşturulur.

Adım 2-1 (Miras Fazı) Aday çözüm (V_i) içerisinde, X_i içerisinde 1 olan boyutlardan M_{11} adedi 1 olarak değiştirilir.

Adım 2-2 (Miras Reddi Fazı). Aday çözüm (V_i) içerisinde, X_i içerisinde 0 olan boyutlardan M_{10} adedi 1 olarak değiştirilir.

Aday çözüm (V_i) çıktı olarak verilir.

Şekil 3.7. Yeni ikili çözüm üretim algoritmasının sözde kodu (Kashan ve ark., 2012).

SimTSA'nın en önemli parçası olan NBSG algoritmasının çalışması 16 boyutlu bireyler için sayısal bir örnekle aşağıda açıklanmıştır.

Başlangıçta:

$$X_i = [0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]$$

$$X_k = [0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0]$$

$$\varphi = 0.6$$

olarak ayarlanmıştır. Bu bilgiler ışığında $m_{01}=1$, $m_{10}=2$, $m_{11}=9$, $m_{00}=4$ olarak hesaplanmıştır.

$$\text{Benzemezlik}(X_i, X_k) = 1 - \text{Benzerlik}(X_i, X_k) = 1 - \frac{9}{1 + 2 + 9} = 1 - \frac{9}{12} = 0.25$$

$$A = \varphi. Benzemezlik(X_i, X_k) \rightarrow A = 0.6 \times 0.25 = 0.15$$

$$\min z \left| 1 - \frac{M_{11}}{M_{11}+M_{10}+M_{01}} - A \right| = \min z \left| 1 - \frac{M_{11}}{M_{11}+M_{10}+M_{01}} - 0.15 \right|$$

$$M_{11} + M_{01} = 11 \text{ (n}_1\text{)}$$

$$M_{10} \leq 5 \text{ (n}_0\text{)}$$

$M_{11}, M_{10}, M_{01} \geq 0$ ve tam sayı olmak zorundadır.

Tam sayılı programlama modeli işletildiğinde $M_{01} = 0$, $M_{10} = 2$, $M_{11} = 11$ ve $z = 0.0038$ olarak bulunur. $S_k = [0000000000000000]$ şeklinde içeriği 0 doldurulan tohum oluşturulur. X_i içerisinde 1 olan boyutlar BirlerXi = [2,3,4,5,7,8,9,10,11,13,15] ve $M_{11} = 11$ olduğundan rastgele seçilen 11 boyut [2,3,4,5,7,8,9,10,11,13,15] 1 yapılır ve $S_k = [0111101111101010]$ haline gelir. X_i içerisinde 0 olan boyutlar SıfırlarXi = [1,6,12,14,16] ve $M_{10} = 2$ olduğundan rastgele seçilen [6, 12] iki boyutun değeri 1 olarak değiştirilir ve $S_k = [01111111111111010]$ halini alır. NBSG algoritmasını kullanan farklı örnekler için (Kashan ve ark., 2012; Kashan ve ark., 2013; Kiran ve Gunduz, 2013) çalışmaları incelenebilir. Bu çalışmalarda da kullanıldığı gibi φ parametresi sabit alınmamış, Denklem 3.27'deki gibi maksimum bir değerden başlayarak iterasyonlar sürecinde belirlenmiş olan minimum değere düşürülmüştür.

$$\varphi^t = \varphi_{max} - \left(\frac{\varphi_{max} - \varphi_{min}}{maxFES} \right) FES \quad (3.27)$$

Burada FES mevcut fonksiyon değerlendirme sayısını, $maxFES$ maksimum fonksiyon değerlendirme sayısını, φ_{max} maksimum φ değerini, φ_{min} minimum φ değerini göstermektedir. Deneysel çalışmalarda $\varphi_{max} = 0.9$ ve $\varphi_{min} = 0.5$ alınmıştır. Bunun sebebi başlangıçta daha çeşitli çözümler elde etmek amaçlanırken, araştırmanın bitmesine yakın daha benzer çözümler üretmenin faydalı olduğunun düşünülmesidir.

3.3.2. Mantıksal kapı tabanlı ikili TSA

İkili gösterim için yeni aday çözüm üretimi için mantıksal kapıların kullanımı (Kiran ve Gunduz, 2013) çalışmasında ABC algoritmasına uygulanmış ve umut verici çözümler üretilmiştir. Mantıksal kapıların girişleri ve çıkışları ikili değerler olduğundan herhangi bir modifikasyona gerek duymadan direkt olarak uygulanabilirler. Temel olarak, “VE (AND)”, “VEYA (OR)” ve “DEĞİL (NOT)” olarak adlandırılan üç mantıksal kapı vardır. “ÖZEL VEYA (XOR)” işlemi “VE” ve “VEYA” kapılarının bir birleşimidir ve

mantıksal devrelerde yaygın olarak kullanılır. LogicTSA'da, ağaçlar ikili değerlerle başlatıldıktan sonra tohum üretim aşamasında XOR kapısı kullanılır. Bunun sebebi girdiler bilinmediği ve en çeşitli ve adil sonucun ÖZEL VEYA kapısı ile üretilmesinden dolayıdır. Şekil 3.8'de VE (%25=1 %75=0), VEYA (%75=1 %25=0) ve ÖZELVEYA (%50=1 %50=0) mantık kapılarıyla pozisyon güncelleme işlemi yapıldığı durumlarda hangi oranda 1 veya 0 üretileceği belirtilmiştir. ÖZELVEYA kapısı ile eşit bir güncelleme işlemi yapılabilmektedir.

VEYA		
X_i	X_k	$X_i + X_k$
0	0	0
0	1	1
1	0	1
1	1	1
%75=1 %25=0		

VE		
X_i	X_k	$X_i \& X_k$
0	0	0
0	1	0
1	0	0
1	1	1
%25=1 %75=0		

ÖZEL VEYA		
X_i	X_k	$X_i \oplus X_k$
0	0	0
0	1	1
1	0	1
1	1	0
%50=1 %50=0		

Şekil 3.8. VE, VEYA ve ÖZELVEYA mantık kapılarıyla pozisyon güncelleme

Temel TSA'daki güncelleme kuralının seçiminde ST kullanılırken, LogicTSA'da boyut seçiminde aynı parametre Denklem 3.28'deki şekilde kullanılır.

$$S_{kj} = \begin{cases} T_{ij} \oplus (B_j \oplus T_{rj}), & \text{eğer } rand_{ij} < ST \\ T_{ij} & , \text{diğer durumlarda} \end{cases} \quad (3.28)$$

Burada S_{kj} k.tohumun j.boyutunu, T_{ij} i.ağacın j.boyutunu, B_j en iyi ağacın j.boyutunu, T_{rj} ise rastgele bir ağacın j.boyutunu temsil etmektedir. Kısaca, bir ağaç için bir tohum oluşturulurken, bazı karar değişkenleri doğrudan mevcut ağaçtan tohuma direkt olarak kopyalanırken ve diğerleri mevcut ağaç, komşu ağaç ve en iyi ağaç arasındaki etkileşim ile elde edilir.

3.3.3. Benzerlik ve mantıksal kapı tabanlı ikili TSA

SimLogicTSA, LogicTSA ve SimTSA'nın hibritlenmiş bir kombinasyonudur. Bu yaklaşımda her iterasyonda 0-1 aralığında rastgele bir sayı üretilir ve bu sayı 0.5'ten küçükse, tohum üretimi için SimTSA kullanılır; aksi takdirde LogicTSA kullanılır. Bu durum Denklem 3.29'da gösterilmiştir.

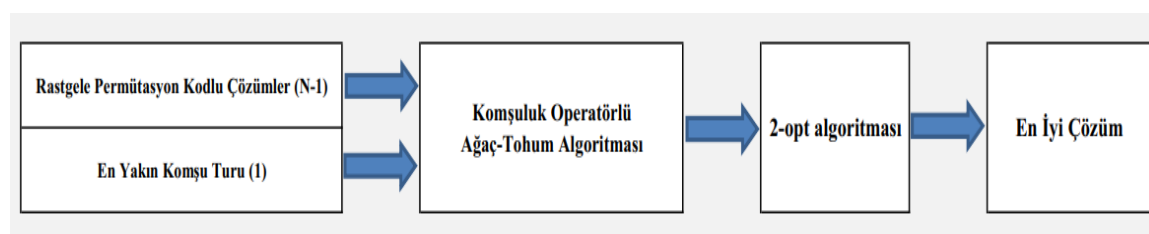
$$\text{Tohum Üretim Metodu} = \begin{cases} \text{SimTSA}, & \text{eğer } rand < 0.5 \\ \text{LogicTSA}, & \text{diğer durumlarda} \end{cases} \quad (3.29)$$

SimLogicTSA'da popülasyon rastgele ikili değerlerle başlatılır ve arama sırasında SimTSA veya LogicTSA'nın tohum üretim mekanizması kullanılır. Her iki yaklaşımın da arama yetenekleri açısından bazı avantajları vardır. Örneğin, mevcut ağacın etrafında

LogicTSA güçlü bir yerel arama özelliğine sahiptir, çünkü tohum üretimi sırasında karar değişkenlerinin bir kısmı değiştirilmeden aktarılır. SimTSA ise çözüm alanındaki yeni noktaları keşfetmekte daha iyidir fakat yoğunlaşmada LogicTSA'dan daha kötüdür. Bu nedenle, SimTSA ve LogicTSA'nın hibritlenmesi, çözüm alanında dengeli arama sağlamak için de faydalıdır. Seçim olasılığı problemin çözümü için yön bilinmediğinden 0.5 olarak alınmıştır. Çözülecek ikili optimizasyon problemlerinin özelliklerine bağlı olarak, bu sabit değer SimTSA veya LogicTSA'yı teşvik etmek için değiştirilebilir.

3.4. Ayrık Tam sayılı Ağaç-Tohum Algoritması

Permütasyon kodlamalı optimizasyon problemlerinin çözmek için yapılan kaynak araştırması doğrultusunda literatürde sıklıkla kullanılan komşuluk operatörleri TSA ile birleştirilerek ayrık TSA (discrete TSA, DTSA) yöntemi önerilmiştir. Bu yöntemde başlangıç popülasyonundaki bir birey hariç tüm bireyler 1'den D'ye kadar olan tam sayıların rastgele bir permütasyonunu içeren vektör şeklinde oluşturur. Buradaki D ilgili problemin boyut sayısıdır. Bir birey ise problem bilgisini kullanarak en kısa turu oluşturan dizi olarak oluşturulur ve popülasyona eklenir. Bu işlemin amacı en yakın komşu sezgiselinin gücünden yararlanmaktır. Çalışmada TSP'ler çözüldüğünden tam turun sağlanması için ilgili bireylerin ilk boyutları D+1.boyuta kopyalanarak tur oluşumu sağlanmıştır. Tohum üretme aşamasında ise komşuluk operatörleri olarak adlandırılan takas, öteleme ve simetri (Chunhua ve ark., 2012; Zhou ve ark., 2016) yöntemleri kullanılır. Sonlandırma kriterine erişildiği zaman ise elde edilen en iyi çözüme 2-opt (Croes, 1958) yerel arama tekniği uygulanarak çözüm kalitesinin artma imkânı olup olmadığı araştırılmakta ve nihai sonuca ulaşılmaktadır. Önerilen yöntemin genel işleyişi Şekil 3.9'da verilmiştir.



Şekil 3.9. DTSA'nın genel işleyişi

3.4.1. Komşuluk operatörleri

DTSA'da tohum üretimi için kullanılan takas, öteleme ve simetri komşuluk operatörleri alt bölümlerde anlatılmıştır.

3.4.1.1. Takas operatörü

Takas operatörü ile tohum üretmek için $[1, D]$ (boyut sayısı) arasında iki farklı rastgele sayı oluşturulur. Ağaçtaki bu iki pozisyon üretilen tohum için yer değiştirilir. Takas operatörü ağaçta küçük değişiklikler yapar. Takas operatörünün çalışmasını gösteren örnek bir işlem Şekil 3.10'da görülmektedir.

Ağaç =	1	2	3	4	5	6
	takas(2,5)					
Tohum =	1	5	3	4	2	6

Şekil 3.10. Takas operatörünün çalışması

3.4.1.2. Öteleme operatörü

Öteleme operatörü ile tohum üretmek için $[1, D]$ (boyut sayısı) arasında iki farklı rastgele sayı (x ve y) oluşturulur. Daha sonra x pozisyonundaki karar değişkeni hafızaya alınır ve diğer karar değişkenleri $[x, y]$ aralığında sola kaydırılır. Daha sonra, hafızaya alınmış karar değişkeni y konumuna atanır. Öteleme operatörü ağaç üzerinde orta düzeyde değişiklik yapar. Öteleme operatörünün çalışmasını gösteren örnek bir işlem Şekil 3.11'de görülmektedir.

Ağaç =	1	2	3	4	5	6
	öteleme(3,5)					
Tohum =	1	2	4	5	3	6

Şekil 3.11. Öteleme operatörünün çalışması

3.4.1.3. Simetri operatörü

Simetri operatörü ile tohum üretmek için $[1, D]$ (boyut sayısı) arasında iki farklı rastgele sayı oluşturulur. Bu rastgele sayılar aynı blok büyüklüğü ve birbirleriyle çakışmayacak şekilde ayarlanır. Her iki blok hem yer değiştirilir hem de simetrisi alınır.

Simetri operatörü ağaç üzerinde büyük düzeyde değişiklik yapar. Simetri operatörünün çalışmasını gösteren örnek bir işlem Şekil 3.12'de görülmektedir.

Ağaç =	1	2	3	4	5	6
	simetri((2,3),(4,5))					
Tohum =	1	5	4	3	2	6

Şekil 3.12. Simetri operatörünün çalışması

3.4.2. DTSA'nın algoritmik çerçevesi

Detaylı algoritmik çerçeve (DTSA'nın sözde kodu), Şekil 3.13'de verilmiştir.

<p>Maksimum fonksiyon değerlendirme sayısını belirleyin (maxFES) Problemdaki toplam boyut sayısını belirleyin (D) Ağaç sayısını belirleyin (N) Araştırma eğilimi parametresini belirleyin (ST) İlk ağaç olarak <u>en yakın komşu turunu</u> ekleyin Tüm ağaçları (ilki hariç) 1 ile D arasında rasgele permütasyonla oluşturun Tüm ağaçların amaç fonksiyon değerini hesaplayın Fonksiyon değerlendirme sayısının (FES) değerini N yapın Amaç fonksiyon değerlerini kullanarak en iyi ağacı belirleyin (En İyi) WHILE FES sayısı maxFES'ten küçük olduğu sürece: Tüm ağaçlar için (Tree(1)'den Tree (N)) Tohum sayısını (ns) 6 olarak belirleyin Rastgele bir ağaç (mevcut ağaç harici) belirleyin (Tree(k)) 0-1 arasında rastgele bir sayı üretin (rand) EĞER rand < ST ise <u>En iyi ağaca takas</u> operatörü uygulayarak bir tohum oluşturun (s1) <u>En iyi ağaca öteleme</u> operatörü uygulayarak bir tohum oluşturun (s2) <u>En iyi ağaca simetri</u> operatörü uygulayarak bir tohum oluşturun (s3) <u>Rastgele bir ağaca takas</u> operatörü uygulayarak bir tohum oluşturun (s4) <u>Rastgele bir ağaca öteleme</u> operatörü uygulayarak bir tohum oluşturun (s5) <u>Rastgele bir ağaca simetri</u> operatörü uygulayarak bir tohum oluşturun (s6) END EĞER EĞER rand ≥ ST ise <u>Mevcut ağaca takas</u> operatörü uygulayarak bir tohum oluşturun (s1) <u>Mevcut ağaca öteleme</u> operatörü uygulayarak bir tohum oluşturun (s2) <u>Mevcut ağaca simetri</u> operatörü uygulayarak bir tohum oluşturun (s3) <u>Rastgele bir ağaca takas</u> operatörü uygulayarak bir tohum oluşturun (s4) <u>Rastgele bir ağaca öteleme</u> operatörü uygulayarak bir tohum oluşturun (s5) <u>Rastgele bir ağaca simetri</u> operatörü uygulayarak bir tohum oluşturun (s6) END EĞER Mevcut ağacın tohumlarının amaç fonksiyonlarını hesaplayın FES=FES+6 En iyi tohumu belirleyin (s1, s2, s3, s4, s5, s6) (En İyi Tohum) En iyi tohum mevcut ağaçtan daha iyiyse, geçerli ağacı en iyi tohumla değiştirin END FOR Geçici en iyi ağacı belirleyin (Geçici En İyi) Eğer geçici en iyi ağaç (Geçici En İyi) en iyi ağaçtan (En İyi) iyiyse değiştirin END WHILE En iyi çözüme 2 opt algoritmasını uygulayın. DTSA tarafında elde edilen en iyi çözümü raporlayın.</p>

Şekil 3.13. DTSA'nın sözde kodu

3.5. Optimizasyon Problemleri

3.5.1. Sürekli kısıtlı nümerik kıyas fonksiyonları

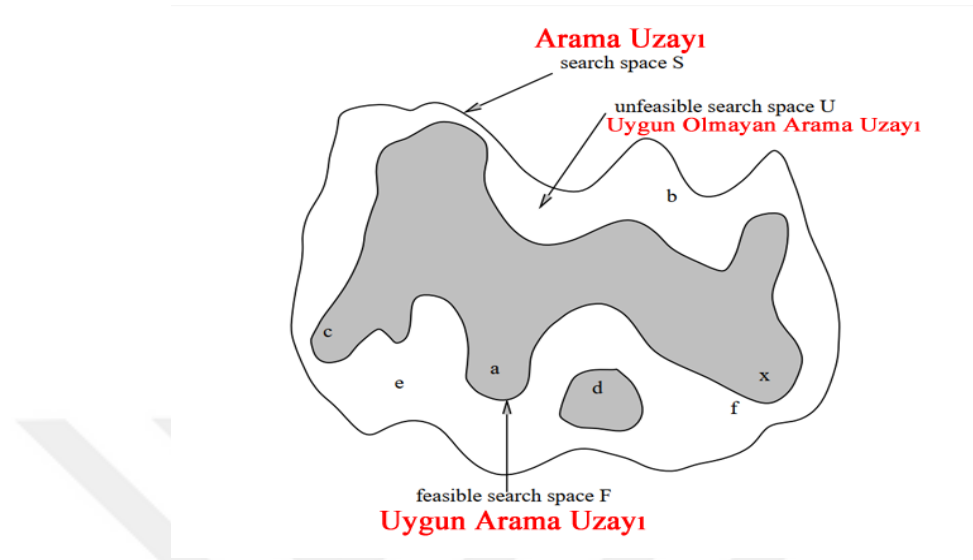
Kısıtlı optimizasyon problemlerinde doğası gereği, sınırlamaların çizdiği kabul edilebilir bölge içerisinde çözümlerin aranması veya bu sınırlar dışında arama yapılsa dahi en son bulunacak en iyi çözümün kabul edilebilir bölgede olması gerekir. Kısıtlı optimizasyon problemlerinin performansını sadece tek bir parametre etkilememektedir. Karar değişkeni sayısı, doğrusal olan ve olmayan eşitlik ve eşitsizliklerin sayısı, fonksiyonun türü, aktif kısıt sayısı ve p değeri problemin çözümünü etkilemektedir (Akay, 2009).

Çizelge 3.4. Sürekli kısıtlı optimizasyon kıyas seti ve özellikleri (Akay ve Karaboga, 2017)

	Boyut	Problem Tipi	p (%)	LI	NI	LE	NE
G01	13	kuadratik	0.0003	9	0	0	0
G02	20	doğrusal olmayan	99.9973	1	1	0	0
G03	10	doğrusal olmayan	0.0026	0	0	0	1
G04	5	kuadratik	27.0079	0	6	0	0
G05	4	doğrusal olmayan	0.0000	2	0	0	3
G06	2	doğrusal olmayan	0.0057	0	2	0	0
G07	10	kuadratik	0.0000	3	5	0	0
G08	2	doğrusal olmayan	0.8581	0	2	0	0
G09	7	doğrusal olmayan	0.5199	0	4	0	0
G10	8	doğrusal	0.0020	3	3	0	0
G11	2	kuadratik	0.0973	0	0	0	1
G12	3	kuadratik	4.7697	0	729	0	0
G13	5	doğrusal olmayan	0.0000	0	0	1	2

Akay ve Karaboga (2017) tarafından yapılan çalışmadan alınan ve Çizelge 3.4'te verilen tabloda LI doğrusal eşitsizlik sayısını, NI doğrusal olmayan eşitsizlik sayısını, LE doğrusal eşitlik sayısını, NE doğrusal olmayan eşitlik sayısını göstermektedir. p değeri, kabul edilebilir bölgedeki çözüm sayısının (F), tüm çözümlerin sayısına (S) bölünmesiyle elde edilen değeri göstermektedir. Bu değer kabul edilebilir alanın tüm araştırma uzayına oranını kestirmek için kullanılmaktadır. Tablodaki değerler hesaplanırken S değeri 1,000,000 olarak alınmıştır. CTSA'nın performansını ölçerken kullanılan 13 sürekli kısıtlı fonksiyonun matematiksel formülleri ve özellikleri EK-1'de sunulmuştur.

Arama uzayı, uygun arama uzayı, uygun olmayan arama uzayı kavramlarının daha iyi anlaşılabilmesi için Michalewicz (1995) tarafından yapılan çalışmadan alınarak Türkçeleştirilmiş olan Şekil 3.14 incelenebilir.



Şekil 3.14. Kısıtlı Optimizasyonda Uygun ve Uygun Olmayan Çözümlerin Gösterimi (Michalewicz, 1995)

3.5.2. Sürekli kısıtlı mühendislik tasarım problemleri

CTSA için kullanılan sürekli kısıtlı mühendislik tasarım problemleri EK-2'de sunulmuştur.

3.5.3. Kapasitesiz tesis yerleşim problemi

Geliştirilen ikili algoritmaların (SimTSA, LogicTSA, SimLogicTSA) performansının ölçülmesi için saf bir ikili test fonksiyonu olan kapasitesiz tesis yerleştirme problemi (uncapacitated facility location problem, UFLP) kullanılmıştır. UFLP sadece ikili karar değişkenlerine sahiptir ve ikili optimizasyon algoritmalarının performans analizlerinde ve karşılaştırmalarında sıklıkla kullanılmaktadır. Bu özelliği nedeniyle önerdiğimiz algoritmaların performans analizleri ve karşılaştırmaları da bu problem üzerinden yapılmıştır. Problemin temelinde müşteriler, tedarikçiler ve bu ikisi arasında gerçekleşen siparişlerden dolayı oluşan maliyetler bulunmaktadır. Her bir müşterinin talebi tesisler tarafından karşılanmak zorundadır. Müşteri talebinin hangi açık tesis tarafından karşılanacağı ise kesin bir yöntem ile belirlenmektedir. Bu bağlamda problem, tesis kurulumu ve transfer maliyeti olmak üzere iki farklı maliyet barındırmaktadır ve problemin amacı, karar değişkenlerine (tesislerin açık olması durumu

1 ve kapalı olması durumu 0 kabulü ile) 0 veya 1 atamak yolu ile toplam maliyeti en aza indirmektir. Karar değişkenlerine atanan 0 veya 1 değerleri, sırasıyla tesisin kapalı veya açık olduğunu göstermektedir. Olası çözüm sayısına, toplam tesis sayısı kullanılarak ulaşılabilir. Toplam tesis sayısı n olarak ifade edildiğinde, problem için olası çözüm sayısı en az bir tesis açık olmak koşuluyla $2^n - 1$ olarak bulunur. Tesislerin açılışlarının maliyeti olduğu durumlarda problem NP-zor (NP-hard) (Cornuéjols ve ark., 1983) (olmadığı durumlarda ise P) problemlerdendir ve literatürde bu tarz problemlerin kesin yöntemlerle çözümünün aşırı zaman almasından dolayı birçok evrimsel veya sürü zekâsı tabanlı yaklaşımlar önerilmiştir (Jakob ve Pruzan, 1983; Monabbati ve Kakhki, 2015).

Problemde, potansiyel tesis lokasyon sayısı bilinmekte fakat kaç tesisin kurulacağı bilinmemektedir. Her tesisin sabit bir kurulum maliyeti mevcuttur. Eğer birden fazla tesis kurulu ise, müşteri siparişlerini, transfer maliyeti en uygun olan tesisten gerçekleştirir (Ghosh, 2003). Literatürde UFLP problemi için örnek veri setlerine OR-Library'den erişilebilmektedir (Beasley, 1990). Problemin matematik modeli (Ghosh, 2003) Denklem 3.30-3.34 arasında sunulmuştur.

$$AA = \{aa_1, aa_2, \dots, aa_{mm}\} \quad (3.30)$$

$$JJ = \{jj_1, jj_2, \dots, jj_{nn}\} \quad (3.31)$$

$$FF = \{ff_1, ff_2, \dots, ff_{mm}\} \quad (3.32)$$

$$CC = \begin{bmatrix} cc_{1,1} & cc_{1,2} & \dots & cc_{1,nn} \\ \dots & & & \\ cc_{mm,1} & cc_{mm,2} & \dots & cc_{mm,nn} \end{bmatrix} \quad (3.33)$$

$$\arg \min = \{ \delta(SS) = \sum_{aa \in SS} ff_{aa} + \sum_{jj \in JJ} \min \{ cc_{aa,jj} | aa \in SS \} \} \quad (3.34)$$

Burada AA potansiyel tesis yerleri kümesini, aa tesisi, mm toplam tesis sayısını, JJ müşterilerin kümesini, jj müşteriye, nn toplam müşteri sayısını, FF her tesis için tanımlanmış olan tesislerin sabit maliyet kümesini, ff ilgili tesisin maliyetini, CC transfer maliyet matrisini göstermektedir. Denklemde SS , AA kümesinin boş olmayan ve minimum maliyet şartını sağlayan alt kümesidir. Amaç müşterilerin taleplerini en düşük maliyetle karşılayacak tesisi açık tutarak tedariki oradan sağlamaktır.

UFLP'yi çözmek için dal-sınır (branch-and-bound) (Holmberg, 1999), doğrusal programlama (linear programming) ve Lagrangian relaxation (Barcelo ve ark., 1990), dual-based procedure (Erlenkotter, 1978) gibi kesin çözüm yöntemleri önerilmiştir. Kesin yöntemler optimum çözümü garanti etse de makul bir zamanda çözüm üretememektedir. Bu yüzden çok sayıda sezgisel yöntem ile UFLP çözülmeye çalışılmıştır. Örneğin, GA (Jaramillo ve ark., 2002), filter-and-fan (F&F) metodu (Greistorfer ve Rego, 2006), tabu

arama (Al-Sultan ve Al-Fawzan, 1999; Sun, 2006), sürekli ve ayrık PSO (Sevklı ve Guner, 2006), binABC (Kiran ve Gunduz, 2013), DisABC (Kashan ve ark., 2012), BPSO (Kennedy ve Eberhart, 1997; Kiran ve Gunduz, 2013) komşuluk arama (neighborhood search) (Ghosh, 2003), S-bAFSA (Simplified Binary Artificial Fish Swarm Algorithm) (Azad ve ark., 2013), DisDE (Kashan ve ark., 2013) UFL problemlerini çözmek için literatürde bulunan metasezgisel yaklaşımlardandır. Önerilmiş olan ikili algoritmaların deneysel çalışmalarında Çizelge 3.5'te özellikleri belirtilen 15 UFL problemi kullanılmıştır.

Çizelge 3.5. Kapasitesiz tesis yerleşim problemleri ve bilgileri (Kiran ve Gunduz, 2013)

Problemin Adı	Problemin Boyutu	Optimum Sonuç
Cap71	16x50	932,615.75
Cap72	16x50	977,799.40
Cap73	16x50	1,010,641.45
Cap74	16x50	1,034,976.98
Cap101	25x50	796,648.44
Cap102	25x50	854,704.20
Cap103	25x50	893,782.11
Cap104	25x50	928,941.75
Cap131	50x50	793,439.56
Cap132	50x50	851,495.33
Cap133	50x50	893,076.71
Cap134	50x50	928,941.75
CapA	100x1000	17,156,454.48
CapB	100x1000	12,979,071.58
CapC	100x1000	11,505,594.33

3.5.4. Ayrık tam sayılı gezgin satıcı problemi

TSP bir satıcının herhangi bir şehirden başladığı turunu yine bu şehirde bitirdiği, satış için verilen tüm şehirleri içeren, asıl amacın bu turu en kısa yoldan tamamlamak olduğu bir problemdir. TSP'nin zorluğunun nedeni D şehir için başlangıç şehri belirli olduğunda $(D-1)!$ farklı tur olmasından dolayıdır. Bu turların hangisinin en uygun olduğunu tespit etmek zor bir problemdir. TSP'nin amaç fonksiyonu Denklem 3.35'te, kısıtları Denklem 3.36-3.39 arasında verilmiştir.

$$\text{Minimize: } \sum_{y \neq z}^D c_{yz} b_{yz} \quad (3.35)$$

$$\sum_{z=1}^D b_{yz} = 1 \quad y = 1, 2, \dots, D \quad (3.36)$$

$$\sum_{y=1}^D b_{yz} = 1 \quad z = 1, 2, \dots, D \quad (3.37)$$

$$\sum_{y,z \in S}^D b_{yz} \leq |S| - 1, \quad S \subset V, 2 \leq S \leq D - 2 \quad (3.38)$$

$$b_{yz} \in \{0,1\}, y, z = 1, \dots, D, y \neq z \quad (3.39)$$

Burada b_{yz} y ve z ile belirtilmiş şehirlerin ziyaret edilip edilmediğini tutan karar değişkenini, c_{yz} y ve z şehirleri arasındaki mesafeyi (maliyeti), D şehir sayısını göstermektedir. Denklem 3.36 ve Denklem 3.37'deki kısıtlar her şehre yalnızca bir kez gidilmesini sağlamaktadır. Denklem 3.38'deki kısıtlar oluşabilecek alt turları engellemeyi ve her şehirden bir kez ayrılmayı sağlamaktadır. Denklem 3.39'da ikili kısıt koşulu verilmiştir.

İki şehir arasındaki uzaklıklar eşitse bu tip TSP'lere simetrik TSP ismi verilir. Simerik TSP'de ise $(D-1)!/2$ farklı tur bulunmaktadır. Deneysel çalışmalarda simetrik TSP'lere çözüm aranmıştır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. CTSA Deneysel Sonuçları ve Karşılaştırmaları

4.1.1. Standart kısıtlı kıyas fonksiyonları ile yapılan deneylerin sonuçları

CTSA'nın performansını analiz etmek için EK 1'de verilen 13 kısıtlı kıyas fonksiyonu kullanılmıştır. Bu fonksiyonlar G01-G13 şeklinde isimlendirilmişlerdir N parametresi {10,20,30,40,50,60,70,80,90,100}, ST parametresi ise {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} alınarak ayrıntılı analiz yapılmıştır. Algoritma her test durumu için rastgele olarak 30 kez çalıştırılmış ve testlerden elde edilen ortalama ve standart sapmalar analiz tablolarında rapor edilmiştir. Durdurma kriteri olarak maxFEs kullanılmış ve literatürdeki kıyas yapılan daha önceki çalışmalarla eşit bir yarış yapılması için $2.4E+5$ 'e ayarlanmıştır.

4.1.1.1. ST parametresinin analizi

Öncelikle ST değeri sabit tutulup N değiştirilerek popülasyon büyüklüğünün sonuçlara etkisi araştırılmıştır. N parametresinin {10,20,30,40,50,60,70,80,90,100} alındığı ve ST parametresinin {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} alındığı durumlarda elde edilen sonuçlar Çizelge 4.1-4.9'da gösterilmiştir. Çizelge 4.1-4.9'daki sonuçlar incelendiğinde G12 problemi için her durumda optimum çözüm elde edildiği görülmektedir. Bu nedenle, önerilen yöntemin performans analizi için G12 probleminin ayırt edici bir problem olmadığı ortadadır. N, 80, 90 veya 100 iken, önerilen yöntem diğer durumlardan daha düşük performans göstermektedir. Bunun sebebi sabit maxFEs ile işlem yapıldığından gelişim için yeterince iterasyon yapılamamasıdır. G13 problemi hiçbir şart altında optimum olarak çözülememiştir. Bu duruma uygun çözümlerin üretileceği alanın arama uzayı içerisinde çok kısıtlı olması neden olmuştur. Genel olarak N, 10, 20, 30, 40 ve 50 iken hemen hemen her durumda daha iyi sonuçlar elde edilmiştir. Bu popülasyon büyüklükleri için sonuçlara bir diğer kontrol parametresi olan ST açısından bakmak için tablolar yeniden düzenlenerek sunulmuştur.

Çizelge 4.1. ST=0.1 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.64E-01	3.38E-02	7.90E-01	8.10E-03	7.99E-01	7.85E-03	8.01E-01	3.50E-03	7.98E-01	3.98E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.26E+03	2.01E+02	5.19E+03	1.26E+02	5.19E+03	8.34E+01	5.20E+03	1.22E+02	5.17E+03	8.63E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.47E+01	4.50E-01	2.45E+01	1.80E-01	2.45E+01	1.51E-01	2.45E+01	8.30E-02	2.45E+01	9.30E-02
G08	9.14E-02	1.69E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	4.00E-03	6.81E+02	3.00E-03	6.81E+02	5.00E-03	6.81E+02	4.00E-03	6.81E+02	6.00E-03
G10	7.35E+03	4.01E+02	7.16E+03	1.28E+02	7.12E+03	8.01E+01	7.13E+03	7.19E+01	7.12E+03	3.16E+01
G11	8.38E-01	8.00E-02	8.25E-01	6.40E-02	8.00E-01	5.60E-02	8.06E-01	5.50E-02	7.95E-01	5.30E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.06E+00	6.56E-01	1.70E+00	2.96E+00	9.19E-01	1.48E-01	9.88E-01	2.82E-01	9.60E-01	1.51E-01
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.97E-01	4.41E-03	7.91E-01	7.15E-03	7.80E-01	1.62E-02	7.43E-01	3.64E-02	7.09E-01	4.54E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03
G05	5.16E+03	5.28E+01	5.18E+03	5.48E+01	5.17E+03	6.32E+01	5.17E+03	6.05E+01	5.18E+03	6.31E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.05E-01	2.46E+01	1.07E-01	2.46E+01	6.60E-02	2.46E+01	9.20E-02	2.47E+01	1.19E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	5.00E-03	6.81E+02	5.00E-03	6.81E+02	6.00E-03	6.81E+02	1.00E-02
G10	7.17E+03	5.40E+01	7.18E+03	4.70E+01	7.18E+03	5.11E+01	7.24E+03	7.18E+01	7.24E+03	6.26E+01
G11	7.91E-01	4.00E-02	8.03E-01	4.80E-02	7.96E-01	4.90E-02	8.30E-01	6.40E-02	8.16E-01	5.40E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.45E-01	1.06E-01	9.14E-01	1.59E-01	9.69E-01	1.98E-01	9.95E-01	1.51E-01	1.04E+00	3.10E-01

Çizelge 4.2. ST=0.2 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.51E-01	6.11E-02	7.96E-01	9.50E-03	8.00E-01	4.54E-03	8.00E-01	5.23E-03	7.97E-01	6.98E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.24E+03	1.60E+02	5.17E+03	7.88E+01	5.17E+03	6.80E+01	5.22E+03	1.31E+02	5.21E+03	1.22E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	2.39E-01	2.45E+01	1.18E-01	2.45E+01	1.21E-01	2.46E+01	1.07E-01	2.46E+01	8.30E-02
G08	9.14E-02	1.69E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	6.00E-03	6.81E+02	5.00E-03
G10	7.32E+03	2.63E+02	7.18E+03	1.06E+02	7.11E+03	4.71E+01	7.14E+03	6.11E+01	7.14E+03	6.65E+01
G11	8.24E-01	7.80E-02	8.07E-01	5.40E-02	8.05E-01	5.80E-02	7.85E-01	3.90E-02	7.99E-01	6.10E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.03E+00	7.94E-01	9.68E-01	2.39E-01	1.03E+00	2.71E-01	9.09E-01	1.69E-01	9.64E-01	1.63E-01
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.89E-01	1.67E-02	7.72E-01	3.17E-02	7.32E-01	3.78E-02	6.74E-01	4.45E-02	6.49E-01	4.53E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	2.00E-03
G05	5.20E+03	1.35E+02	5.21E+03	7.65E+01	5.15E+03	4.64E+01	5.18E+03	4.83E+01	5.17E+03	6.86E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	8.10E-02	2.47E+01	1.12E-01	2.46E+01	8.80E-02	2.47E+01	8.90E-02	2.47E+01	1.31E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	7.00E-03	6.81E+02	6.00E-03	6.81E+02	7.00E-03	6.81E+02	8.00E-03
G10	7.18E+03	8.14E+01	7.21E+03	5.37E+01	7.23E+03	5.62E+01	7.24E+03	4.84E+01	7.27E+03	6.63E+01
G11	8.10E-01	5.80E-02	8.21E-01	5.40E-02	8.01E-01	5.70E-02	8.23E-01	5.00E-02	8.32E-01	6.80E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.80E-01	4.18E-02	9.65E-01	1.09E-01	9.83E-01	5.09E-02	9.70E-01	9.87E-02	9.90E-01	3.10E-02

Çizelge 4.3. ST=0.3 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.71E-01	2.87E-02	7.95E-01	1.43E-02	8.01E-01	4.70E-03	8.01E-01	3.00E-03	7.94E-01	8.56E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.23E+03	1.50E+02	5.20E+03	1.12E+02	5.17E+03	6.79E+01	5.17E+03	4.80E+01	5.18E+03	8.90E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.70E-01	2.45E+01	1.18E-01	2.45E+01	1.03E-01	2.45E+01	8.00E-02	2.46E+01	8.70E-02
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	3.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	6.00E-03	6.81E+02	7.00E-03
G10	7.29E+03	2.40E+02	7.15E+03	9.13E+01	7.14E+03	6.08E+01	7.17E+03	7.11E+01	7.18E+03	5.83E+01
G11	8.29E-01	6.90E-02	8.25E-01	6.90E-02	8.05E-01	5.40E-02	8.02E-01	6.20E-02	8.12E-01	6.20E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.05E+00	7.27E-01	9.31E-01	1.25E-01	9.57E-01	1.79E-01	9.69E-01	1.93E-01	9.64E-01	9.47E-02
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.82E-01	2.20E-02	7.52E-01	3.76E-02	6.84E-01	3.96E-02	6.48E-01	5.23E-02	6.06E-01	3.61E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	3.00E-03
G05	5.17E+03	6.01E+01	5.17E+03	5.82E+01	5.16E+03	4.27E+01	5.17E+03	5.11E+01	5.16E+03	6.99E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	1.00E-03
G07	2.46E+01	1.06E-01	2.47E+01	1.04E-01	2.47E+01	7.80E-02	2.47E+01	1.07E-01	2.48E+01	9.10E-02
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	7.00E-03	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	9.00E-03
G10	7.23E+03	7.12E+01	7.25E+03	7.28E+01	7.27E+03	6.51E+01	7.31E+03	6.53E+01	7.34E+03	6.65E+01
G11	7.96E-01	6.10E-02	8.27E-01	6.80E-02	8.26E-01	6.60E-02	8.11E-01	6.00E-02	8.20E-01	7.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.76E-01	9.55E-02	9.85E-01	3.11E-02	9.87E-01	3.61E-02	9.63E-01	1.16E-01	1.01E+00	1.24E-01

Çizelge 4.4. ST=0.4 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.62E-01	3.76E-02	7.97E-01	8.22E-03	8.02E-01	2.66E-03	7.99E-01	6.22E-03	7.84E-01	1.84E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.20E+03	7.71E+01	5.20E+03	1.09E+02	5.21E+03	1.25E+02	5.18E+03	6.43E+01	5.24E+03	1.41E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.86E-01	2.45E+01	1.66E-01	2.45E+01	1.06E-01	2.46E+01	8.80E-02	2.46E+01	1.18E-01
G08	9.14E-02	1.69E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	2.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	7.00E-03
G10	7.21E+03	1.61E+02	7.14E+03	6.19E+01	7.16E+03	7.28E+01	7.17E+03	6.94E+01	7.24E+03	7.28E+01
G11	8.36E-01	7.30E-02	8.14E-01	6.00E-02	8.31E-01	6.90E-02	8.01E-01	4.20E-02	8.57E-01	7.60E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.78E-01	4.19E-01	9.90E-01	1.86E-01	9.10E-01	1.58E-01	9.68E-01	9.36E-02	9.50E-01	1.09E-01

Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03
G02	7.50E-01	3.24E-02	7.01E-01	4.07E-02	6.53E-01	4.59E-02	6.07E-01	3.79E-02	5.91E-01	3.56E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	2.00E-03	-3.07E+04	8.00E-03
G05	5.20E+03	1.52E+02	5.21E+03	1.60E+02	5.18E+03	8.61E+01	5.16E+03	6.67E+01	5.21E+03	1.18E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	3.00E-03
G07	2.47E+01	8.50E-02	2.47E+01	1.07E-01	2.47E+01	1.15E-01	2.48E+01	1.14E-01	2.49E+01	1.28E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	7.00E-03	6.81E+02	9.00E-03	6.81E+02	1.00E-02	6.81E+02	9.00E-03	6.81E+02	1.20E-02
G10	7.27E+03	7.29E+01	7.31E+03	7.89E+01	7.32E+03	7.39E+01	7.37E+03	7.24E+01	7.40E+03	8.94E+01
G11	8.22E-01	7.10E-02	8.04E-01	5.20E-02	8.18E-01	7.60E-02	8.23E-01	7.20E-02	8.27E-01	6.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.67E-01	9.61E-02	9.91E-01	1.17E-01	9.67E-01	6.72E-02	9.77E-01	4.76E-02	9.18E-01	1.71E-01

Çizelge 4.5. ST=0.5 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.83E-01	2.17E-02	7.99E-01	8.13E-03	8.01E-01	4.41E-03	7.95E-01	9.96E-03	7.67E-01	2.57E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.23E+03	1.47E+02	5.17E+03	5.77E+01	5.19E+03	8.95E+01	5.22E+03	1.78E+02	5.23E+03	1.70E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.41E-01	2.45E+01	1.28E-01	2.45E+01	8.30E-02	2.46E+01	1.12E-01	2.47E+01	1.44E-01
G08	8.92E-02	2.03E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	4.00E-03	6.81E+02	5.00E-03	6.81E+02	5.00E-03	6.81E+02	5.00E-03	6.81E+02	7.00E-03
G10	7.22E+03	9.06E+01	7.19E+03	7.13E+01	7.21E+03	7.48E+01	7.23E+03	6.70E+01	7.26E+03	7.58E+01
G11	8.37E-01	7.70E-02	8.33E-01	6.80E-02	8.14E-01	6.80E-02	8.14E-01	5.70E-02	7.97E-01	5.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.08E-01	1.22E-01	1.13E+00	1.00E+00	9.57E-01	9.28E-02	9.79E-01	2.41E-01	9.76E-01	6.47E-02
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03	-1.50E+01	2.00E-03
G02	7.36E-01	3.55E-02	6.69E-01	3.85E-02	6.20E-01	3.98E-02	5.75E-01	3.54E-02	5.61E-01	3.62E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	6.00E-03	-3.07E+04	1.10E-02
G05	5.25E+03	1.60E+02	5.19E+03	9.64E+01	5.22E+03	1.68E+02	5.19E+03	8.20E+01	5.17E+03	4.53E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	1.00E-03	-6.96E+03	3.00E-03
G07	2.47E+01	1.21E-01	2.48E+01	9.60E-02	2.48E+01	1.29E-01	2.49E+01	1.19E-01	2.49E+01	1.46E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	7.00E-03	6.81E+02	9.00E-03	6.81E+02	9.00E-03	6.81E+02	1.20E-02	6.81E+02	1.50E-02
G10	7.29E+03	7.49E+01	7.36E+03	8.77E+01	7.39E+03	1.07E+02	7.41E+03	9.28E+01	7.48E+03	9.38E+01
G11	8.07E-01	5.30E-02	8.13E-01	7.00E-02	8.34E-01	7.70E-02	8.09E-01	6.20E-02	8.46E-01	6.10E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.17E-01	1.78E-01	9.39E-01	1.49E-01	8.99E-01	1.79E-01	9.66E-01	1.96E-01	9.04E-01	1.66E-01

Çizelge 4.6. ST=0.6 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.88E-01	2.08E-02	8.00E-01	9.41E-03	7.99E-01	5.05E-03	7.83E-01	2.04E-02	7.44E-01	3.65E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.26E+03	1.75E+02	5.24E+03	1.46E+02	5.24E+03	1.69E+02	5.21E+03	9.15E+01	5.19E+03	8.22E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.81E-01	2.45E+01	9.40E-02	2.46E+01	1.15E-01	2.47E+01	1.00E-01	2.47E+01	1.48E-01
G08	9.36E-02	1.22E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	5.00E-03	6.81E+02	6.00E-03	6.81E+02	5.00E-03	6.81E+02	6.00E-03	6.81E+02	7.00E-03
G10	7.22E+03	1.20E+02	7.23E+03	9.94E+01	7.21E+03	6.87E+01	7.30E+03	9.53E+01	7.31E+03	7.63E+01
G11	8.34E-01	7.90E-02	8.24E-01	7.50E-02	8.21E-01	5.40E-02	8.08E-01	6.80E-02	8.13E-01	6.10E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.59E-01	4.88E-01	9.73E-01	3.50E-01	9.87E-01	2.95E-01	9.62E-01	1.52E-01	9.66E-01	8.02E-02
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03	-1.50E+01	2.00E-03	-1.50E+01	4.00E-03
G02	6.87E-01	3.86E-02	6.34E-01	4.47E-02	5.72E-01	3.19E-02	5.59E-01	2.71E-02	5.45E-01	4.04E-02
G03	1.00E+00	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	3.00E-03	-3.07E+04	8.00E-03	-3.07E+04	1.70E-02
G05	5.24E+03	1.54E+02	5.20E+03	1.58E+02	5.20E+03	1.12E+02	5.28E+03	2.37E+02	5.23E+03	1.39E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	1.00E-03	-6.96E+03	3.00E-03
G07	2.48E+01	1.19E-01	2.48E+01	1.27E-01	2.49E+01	1.07E-01	2.50E+01	1.73E-01	2.50E+01	1.61E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	9.00E-03	6.81E+02	1.20E-02	6.81E+02	1.30E-02
G10	7.36E+03	8.24E+01	7.40E+03	9.74E+01	7.43E+03	9.51E+01	7.51E+03	1.06E+02	7.50E+03	1.02E+02
G11	8.11E-01	7.30E-02	8.28E-01	7.20E-02	8.28E-01	7.00E-02	8.28E-01	7.60E-02	8.28E-01	6.10E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.45E-01	1.70E-01	9.31E-01	2.00E-01	9.83E-01	5.38E-01	9.86E-01	3.78E-01	9.21E-01	1.49E-01

Çizelge 4.7. ST=0.7 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.84E-01	1.60E-02	8.01E-01	5.09E-03	7.92E-01	1.15E-02	7.61E-01	2.72E-02	7.12E-01	2.66E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.23E+03	1.30E+02	5.22E+03	1.14E+02	5.27E+03	2.17E+02	5.24E+03	1.42E+02	5.26E+03	1.78E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.47E+01	2.08E-01	2.46E+01	9.90E-02	2.46E+01	1.25E-01	2.47E+01	1.00E-01	2.48E+01	1.04E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	1.00E-02	6.81E+02	0.00E+00	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	1.00E-02
G10	7.22E+03	9.62E+01	7.19E+03	6.04E+01	7.29E+03	9.17E+01	7.28E+03	9.57E+01	7.34E+03	8.89E+01
G11	8.40E-01	8.00E-02	8.10E-01	8.00E-02	8.40E-01	7.00E-02	8.30E-01	7.00E-02	8.10E-01	7.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.47E-01	2.35E-01	9.59E-01	7.20E-02	9.86E-01	3.09E-01	9.22E-01	1.47E-01	1.02E+00	4.55E-01
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03	-1.50E+01	2.00E-03	-1.50E+01	5.00E-03	-1.49E+01	9.00E-03
G02	6.45E-01	3.15E-02	5.99E-01	3.98E-02	5.82E-01	3.23E-02	5.42E-01	2.33E-02	5.22E-01	2.96E-02
G03	1.00E+00	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	3.00E-03	-3.07E+04	7.00E-03	-3.07E+04	1.70E-02	-3.07E+04	4.40E-02
G05	5.31E+03	2.55E+02	5.28E+03	2.67E+02	5.30E+03	2.32E+02	5.33E+03	2.45E+02	5.34E+03	2.34E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	1.00E-03	-6.96E+03	3.00E-03
G07	2.49E+01	1.41E-01	2.49E+01	1.36E-01	2.50E+01	1.60E-01	2.51E+01	1.62E-01	2.52E+01	1.49E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	1.00E-02	6.81E+02	2.00E-02
G10	7.42E+03	8.21E+01	7.44E+03	8.41E+01	7.48E+03	8.89E+01	7.53E+03	7.33E+01	7.54E+03	8.47E+01
G11	8.20E-01	7.00E-02	8.10E-01	7.00E-02	8.30E-01	7.00E-02	8.40E-01	7.00E-02	8.40E-01	7.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.40E-01	1.77E-01	8.88E-01	1.87E-01	9.12E-01	1.25E-01	8.96E-01	2.48E-01	9.73E-01	1.62E-01

Çizelge 4.8. ST=0.8 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.90E-01	1.22E-02	7.95E-01	1.26E-02	7.71E-01	2.31E-02	7.23E-01	2.66E-02	6.60E-01	2.90E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03
G05	5.32E+03	2.41E+02	5.26E+03	2.19E+02	5.34E+03	2.36E+02	5.34E+03	2.78E+02	5.40E+03	2.89E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.47E+01	2.34E-01	2.46E+01	9.90E-02	2.47E+01	1.27E-01	2.47E+01	1.26E-01	2.48E+01	1.43E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	6.00E-03	6.81E+02	7.00E-03	6.81E+02	8.00E-03	6.81E+02	7.00E-03
G10	7.24E+03	9.16E+01	7.24E+03	9.15E+01	7.27E+03	7.41E+01	7.36E+03	8.25E+01	7.39E+03	7.38E+01
G11	8.25E-01	7.50E-02	8.37E-01	8.10E-02	8.22E-01	7.50E-02	8.19E-01	6.70E-02	8.44E-01	6.70E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.43E-01	4.02E-01	9.59E-01	1.90E-01	9.44E-01	2.21E-01	9.68E-01	2.25E-01	9.07E-01	1.58E-01
Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	1.00E-03	-1.50E+01	3.00E-03	-1.50E+01	7.00E-03	-1.49E+01	1.20E-02	-1.49E+01	1.70E-02
G02	6.13E-01	3.23E-02	5.74E-01	4.13E-02	5.37E-01	3.06E-02	5.28E-01	2.73E-02	5.05E-01	2.41E-02
G03	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00
G04	-3.07E+04	2.00E-03	-3.07E+04	1.00E-02	-3.07E+04	2.30E-02	-3.07E+04	3.90E-02	-3.07E+04	7.50E-02
G05	5.35E+03	2.44E+02	5.25E+03	1.72E+02	5.40E+03	2.79E+02	5.28E+03	2.13E+02	5.28E+03	2.16E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	1.00E-03	-6.96E+03	2.00E-03
G07	2.50E+01	1.43E-01	2.50E+01	1.27E-01	2.51E+01	1.83E-01	2.52E+01	1.70E-01	2.53E+01	1.66E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	1.00E-02	6.81E+02	9.00E-03	6.81E+02	1.10E-02	6.81E+02	1.20E-02	6.81E+02	1.20E-02
G10	7.45E+03	9.58E+01	7.51E+03	9.64E+01	7.54E+03	1.09E+02	7.57E+03	1.01E+02	7.62E+03	8.84E+01
G11	7.97E-01	6.00E-02	7.98E-01	5.90E-02	8.01E-01	5.30E-02	8.38E-01	6.60E-02	8.19E-01	7.20E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	8.95E-01	2.07E-01	1.07E+00	7.95E-01	1.00E+00	8.25E-01	8.88E-01	2.37E-01	9.35E-01	3.95E-01

Çizelge 4.9. ST=0.9 alındığında farklı popülasyon büyüklüğünün sonuçlara etkisi

Problem	N=10		N=20		N=30		N=40		N=50	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03	-1.50E+01	2.00E-03
G02	7.82E-01	2.32E-02	7.64E-01	3.10E-02	7.09E-01	2.77E-02	6.54E-01	2.44E-02	5.89E-01	2.38E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	3.00E-03
G05	5.31E+03	2.48E+02	5.27E+03	2.20E+02	5.35E+03	2.95E+02	5.33E+03	2.46E+02	5.31E+03	2.72E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.47E+01	1.67E-01	2.47E+01	9.20E-02	2.47E+01	1.16E-01	2.48E+01	1.01E-01	2.49E+01	1.10E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	4.00E-03	6.81E+02	7.00E-03	6.81E+02	5.00E-03	6.81E+02	9.00E-03
G10	7.27E+03	1.10E+02	7.28E+03	8.77E+01	7.33E+03	7.70E+01	7.40E+03	8.46E+01	7.47E+03	9.19E+01
G11	8.54E-01	8.20E-02	8.29E-01	7.50E-02	8.22E-01	7.40E-02	8.45E-01	7.90E-02	8.18E-01	6.90E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.03E+00	5.50E-01	8.80E-01	1.79E-01	1.08E+00	6.30E-01	8.39E-01	2.60E-01	1.08E+00	9.76E-01

Problem	N=60		N=70		N=80		N=90		N=100	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	6.00E-03	-1.49E+01	1.20E-02	-1.49E+01	2.10E-02	-1.48E+01	3.00E-02	-1.47E+01	5.40E-02
G02	5.52E-01	2.70E-02	5.32E-01	2.90E-02	5.17E-01	2.22E-02	4.86E-01	1.99E-02	4.76E-01	2.73E-02
G03	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.99E-01	0.00E+00	9.98E-01	0.00E+00
G04	-3.07E+04	9.00E-03	-3.07E+04	3.10E-02	-3.07E+04	5.40E-02	-3.07E+04	1.27E-01	-3.07E+04	1.54E-01
G05	5.30E+03	2.28E+02	5.31E+03	2.27E+02	5.41E+03	2.69E+02	5.38E+03	2.40E+02	5.40E+03	2.34E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	3.00E-03
G07	2.50E+01	1.20E-01	2.51E+01	1.91E-01	2.52E+01	1.85E-01	2.53E+01	1.87E-01	2.54E+01	1.69E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	1.00E-02	6.81E+02	1.30E-02	6.81E+02	1.30E-02	6.81E+02	1.60E-02	6.81E+02	1.60E-02
G10	7.50E+03	1.05E+02	7.52E+03	8.85E+01	7.64E+03	1.15E+02	7.61E+03	9.66E+01	7.63E+03	1.08E+02
G11	8.30E-01	7.50E-02	8.47E-01	7.50E-02	8.19E-01	6.90E-02	8.14E-01	6.30E-02	8.02E-01	6.50E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.07E+00	5.86E-01	9.97E-01	5.00E-01	1.55E+00	2.84E+00	1.57E+00	2.63E+00	9.50E-01	4.27E-01

4.1.1.2. Popülasyon büyüklüğünün analizi

N parametresinin {10, 20, 30, 40, 50} alındığı ve ST parametresinin ise {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} alındığı durumlardaki sonuçlar Çizelge 4.10-4.14'te verilmiştir. ST için 0.2 ve N olarak 40 alındığı durumda en iyi sonuçlar üretilmiştir. CTSA'nın literatürdeki yöntemlerle karşılaştırılmasında bu koşullar kullanılmıştır. Ayrıca N, 20, 30 veya 40 iken ve ST değeri 0.1, 0.2 veya 0.3 iken rekabetçi ve standart sapma açısından dayanıklı sonuçlar elde edilmiştir.



Çizelge 4.10. N=10 alındığında farklı ST değerlerinin sonuçlara etkisi

Problem	ST=0.1		ST=0.2		ST=0.3		ST=0.4		ST=0.5	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.64E-01	3.38E-02	7.51E-01	6.11E-02	7.71E-01	2.87E-02	7.62E-01	3.76E-02	7.83E-01	2.17E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.26E+03	2.01E+02	5.24E+03	1.60E+02	5.23E+03	1.50E+02	5.20E+03	7.71E+01	5.23E+03	1.47E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.47E+01	4.50E-01	2.46E+01	2.39E-01	2.46E+01	1.70E-01	2.46E+01	1.86E-01	2.46E+01	1.41E-01
G08	9.14E-02	1.69E-02	9.14E-02	1.69E-02	9.58E-02	0.00E+00	9.14E-02	1.69E-02	8.92E-02	2.03E-02
G09	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	3.00E-03	6.81E+02	2.00E-03	6.81E+02	4.00E-03
G10	7.35E+03	4.01E+02	7.32E+03	2.63E+02	7.29E+03	2.40E+02	7.21E+03	1.61E+02	7.22E+03	9.06E+01
G11	8.38E-01	8.00E-02	8.24E-01	7.80E-02	8.29E-01	6.90E-02	8.36E-01	7.30E-02	8.37E-01	7.70E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.06E+00	6.56E-01	1.03E+00	7.94E-01	1.05E+00	7.27E-01	9.78E-01	4.19E-01	9.08E-01	1.22E-01
Problem	ST=0.6		ST=0.7		ST=0.8		ST=0.9			
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.88E-01	2.08E-02	7.84E-01	1.60E-02	7.90E-01	1.22E-02	7.82E-01	2.32E-02	7.82E-01	2.32E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.26E+03	1.75E+02	5.23E+03	1.30E+02	5.32E+03	2.41E+02	5.31E+03	2.48E+02	5.31E+03	2.48E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.81E-01	2.47E+01	2.08E-01	2.47E+01	2.34E-01	2.47E+01	1.67E-01	2.47E+01	1.67E-01
G08	9.36E-02	1.22E-02	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	5.00E-03	6.81E+02	7.00E-03	6.81E+02	6.00E-03	6.81E+02	6.00E-03	6.81E+02	6.00E-03
G10	7.22E+03	1.20E+02	7.22E+03	9.62E+01	7.24E+03	9.16E+01	7.27E+03	1.10E+02	7.27E+03	1.10E+02
G11	8.34E-01	7.90E-02	8.45E-01	8.30E-02	8.25E-01	7.50E-02	8.54E-01	8.20E-02	8.54E-01	8.20E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.59E-01	4.88E-01	9.47E-01	2.35E-01	9.43E-01	4.02E-01	1.03E+00	5.50E-01	1.03E+00	5.50E-01

Çizelge 4.11. N=20 alındığında farklı ST değerlerinin sonuçlara etkisi

Problem	ST=0.1		ST=0.2		ST=0.3		ST=0.4		ST=0.5	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.90E-01	8.10E-03	7.96E-01	9.50E-03	7.95E-01	1.43E-02	7.97E-01	8.22E-03	7.99E-01	8.13E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.19E+03	1.26E+02	5.17E+03	7.88E+01	5.20E+03	1.12E+02	5.20E+03	1.09E+02	5.17E+03	5.77E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.45E+01	1.80E-01	2.45E+01	1.18E-01	2.45E+01	1.18E-01	2.45E+01	1.66E-01	2.45E+01	1.28E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	3.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	5.00E-03
G10	7.16E+03	1.28E+02	7.18E+03	1.06E+02	7.15E+03	9.13E+01	7.14E+03	6.19E+01	7.19E+03	7.13E+01
G11	8.25E-01	6.40E-02	8.07E-01	5.40E-02	8.25E-01	6.90E-02	8.14E-01	6.00E-02	8.33E-01	6.80E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	1.70E+00	2.96E+00	9.68E-01	2.39E-01	9.31E-01	1.25E-01	9.90E-01	1.86E-01	1.13E+00	1.00E+00
Problem	ST=0.6		ST=0.7		ST=0.8		ST=0.9			
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.		
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00		
G02	8.00E-01	9.41E-03	8.01E-01	5.09E-03	7.95E-01	1.26E-02	7.64E-01	3.10E-02		
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00		
G05	5.24E+03	1.46E+02	5.22E+03	1.14E+02	5.26E+03	2.19E+02	5.27E+03	2.20E+02		
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00		
G07	2.45E+01	9.40E-02	2.46E+01	9.90E-02	2.46E+01	9.90E-02	2.47E+01	9.20E-02		
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00		
G09	6.81E+02	6.00E-03	6.81E+02	4.00E-03	6.81E+02	6.00E-03	6.81E+02	4.00E-03		
G10	7.23E+03	9.94E+01	7.19E+03	6.04E+01	7.24E+03	9.15E+01	7.28E+03	8.77E+01		
G11	8.24E-01	7.50E-02	8.13E-01	7.60E-02	8.37E-01	8.10E-02	8.29E-01	7.50E-02		
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G13	9.73E-01	3.50E-01	9.59E-01	7.20E-02	9.59E-01	1.90E-01	8.80E-01	1.79E-01		

Çizelge 4.12. N=30 alındığında farklı ST değerlerinin sonuçlara etkisi

Problem	ST=0.1		ST=0.2		ST=0.3		ST=0.4		ST=0.5	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.99E-01	7.85E-03	8.00E-01	4.54E-03	8.01E-01	4.70E-03	8.02E-01	2.66E-03	8.01E-01	4.41E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.19E+03	8.34E+01	5.17E+03	6.80E+01	5.17E+03	6.79E+01	5.21E+03	1.25E+02	5.19E+03	8.95E+01
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.45E+01	1.51E-01	2.45E+01	1.21E-01	2.45E+01	1.03E-01	2.45E+01	1.06E-01	2.45E+01	8.30E-02
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	5.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	4.00E-03	6.81E+02	5.00E-03
G10	7.12E+03	8.01E+01	7.11E+03	4.71E+01	7.14E+03	6.08E+01	7.16E+03	7.28E+01	7.21E+03	7.48E+01
G11	8.00E-01	5.60E-02	8.05E-01	5.80E-02	8.05E-01	5.40E-02	8.31E-01	6.90E-02	8.14E-01	6.80E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.19E-01	1.48E-01	1.03E+00	2.71E-01	9.57E-01	1.79E-01	9.10E-01	1.58E-01	9.56E-01	9.28E-02
Problem	ST=0.6		ST=0.7		ST=0.8		ST=0.9			
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.99E-01	5.05E-03	7.92E-01	1.15E-02	7.71E-01	2.31E-02	7.09E-01	2.77E-02	7.09E-01	2.77E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.24E+03	1.69E+02	5.27E+03	2.17E+02	5.34E+03	2.36E+02	5.35E+03	2.95E+02	5.35E+03	2.95E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.46E+01	1.15E-01	2.46E+01	1.25E-01	2.47E+01	1.27E-01	2.47E+01	1.16E-01	2.47E+01	1.16E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	5.00E-03	6.81E+02	5.00E-03	6.81E+02	7.00E-03	6.81E+02	7.00E-03	6.81E+02	7.00E-03
G10	7.21E+03	6.87E+01	7.29E+03	9.17E+01	7.27E+03	7.41E+01	7.33E+03	7.70E+01	7.33E+03	7.70E+01
G11	8.21E-01	5.40E-02	8.44E-01	6.90E-02	8.22E-01	7.50E-02	8.22E-01	7.40E-02	8.22E-01	7.40E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.87E-01	2.95E-01	9.86E-01	3.09E-01	9.44E-01	2.21E-01	1.08E+00	6.29E-01	1.08E+00	6.29E-01

Çizelge 4.13. N=40 alındığında farklı ST değerlerinin sonuçlara etkisi

Problem	ST=0.1		ST=0.2		ST=0.3		ST=0.4		ST=0.5	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	8.01E-01	3.50E-03	8.00E-01	5.23E-03	8.01E-01	3.00E-03	7.99E-01	6.22E-03	7.95E-01	9.96E-03
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.20E+03	1.22E+02	5.22E+03	1.31E+02	5.17E+03	4.80E+01	5.18E+03	6.43E+01	5.22E+03	1.78E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.45E+01	8.30E-02	2.46E+01	1.07E-01	2.45E+01	8.00E-02	2.46E+01	8.80E-02	2.46E+01	1.12E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	4.00E-03	6.81E+02	6.00E-03	6.81E+02	6.00E-03	6.81E+02	4.00E-03	6.81E+02	5.00E-03
G10	7.13E+03	7.19E+01	7.14E+03	6.11E+01	7.17E+03	7.11E+01	7.17E+03	6.94E+01	7.23E+03	6.70E+01
G11	8.06E-01	5.50E-02	7.85E-01	3.90E-02	8.02E-01	6.20E-02	8.01E-01	4.20E-02	8.14E-01	5.70E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.88E-01	2.82E-01	9.09E-01	1.69E-01	9.69E-01	1.93E-01	9.68E-01	9.36E-02	9.79E-01	2.41E-01
Problem	ST=0.6		ST=0.7		ST=0.8		ST=0.9			
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	1.00E-03		
G02	7.83E-01	2.04E-02	7.61E-01	2.72E-02	7.23E-01	2.66E-02	6.54E-01	2.44E-02		
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03		
G05	5.21E+03	9.15E+01	5.24E+03	1.42E+02	5.34E+03	2.78E+02	5.33E+03	2.46E+02		
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00		
G07	2.47E+01	1.00E-01	2.47E+01	1.00E-01	2.47E+01	1.26E-01	2.48E+01	1.01E-01		
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00		
G09	6.81E+02	6.00E-03	6.81E+02	7.00E-03	6.81E+02	8.00E-03	6.81E+02	5.00E-03		
G10	7.30E+03	9.53E+01	7.28E+03	9.57E+01	7.36E+03	8.25E+01	7.40E+03	8.46E+01		
G11	8.08E-01	6.80E-02	8.27E-01	7.30E-02	8.19E-01	6.70E-02	8.45E-01	7.90E-02		
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G13	9.62E-01	1.52E-01	9.22E-01	1.47E-01	9.68E-01	2.25E-01	8.39E-01	2.60E-01		

Çizelge 4.14. N=50 alındığında farklı ST değerlerinin sonuçlara etkisi

Problem	ST=0.1		ST=0.2		ST=0.3		ST=0.4		ST=0.5	
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00
G02	7.98E-01	3.98E-03	7.97E-01	6.98E-03	7.94E-01	8.56E-03	7.84E-01	1.84E-02	7.67E-01	2.57E-02
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00
G05	5.17E+03	8.63E+01	5.21E+03	1.22E+02	5.18E+03	8.90E+01	5.24E+03	1.41E+02	5.23E+03	1.70E+02
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00
G07	2.45E+01	9.30E-02	2.46E+01	8.30E-02	2.46E+01	8.70E-02	2.46E+01	1.18E-01	2.47E+01	1.44E-01
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00
G09	6.81E+02	6.00E-03	6.81E+02	5.00E-03	6.81E+02	7.00E-03	6.81E+02	7.00E-03	6.81E+02	7.00E-03
G10	7.12E+03	3.16E+01	7.14E+03	6.65E+01	7.18E+03	5.83E+01	7.24E+03	7.28E+01	7.26E+03	7.58E+01
G11	7.95E-01	5.30E-02	7.99E-01	6.10E-02	8.12E-01	6.20E-02	8.57E-01	7.60E-02	7.97E-01	5.00E-02
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00
G13	9.60E-01	1.51E-01	9.64E-01	1.63E-01	9.64E-01	9.47E-02	9.50E-01	1.09E-01	9.76E-01	6.47E-02
Problem	ST=0.6		ST=0.7		ST=0.8		ST=0.9			
	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.	Ortalama	Std.Sap.		
G01	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	0.00E+00	-1.50E+01	2.00E-03		
G02	7.44E-01	3.65E-02	7.12E-01	2.66E-02	6.60E-01	2.90E-02	5.89E-01	2.38E-02		
G03	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G04	-3.07E+04	0.00E+00	-3.07E+04	0.00E+00	-3.07E+04	1.00E-03	-3.07E+04	3.00E-03		
G05	5.19E+03	8.22E+01	5.26E+03	1.78E+02	5.40E+03	2.89E+02	5.31E+03	2.72E+02		
G06	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00	-6.96E+03	0.00E+00		
G07	2.47E+01	1.48E-01	2.48E+01	1.04E-01	2.48E+01	1.43E-01	2.49E+01	1.10E-01		
G08	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00	9.58E-02	0.00E+00		
G09	6.81E+02	7.00E-03	6.81E+02	7.00E-03	6.81E+02	7.00E-03	6.81E+02	9.00E-03		
G10	7.31E+03	7.63E+01	7.34E+03	8.89E+01	7.39E+03	7.38E+01	7.47E+03	9.19E+01		
G11	8.13E-01	6.10E-02	8.10E-01	7.30E-02	8.44E-01	6.70E-02	8.18E-01	6.90E-02		
G12	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00	1.00E+00	0.00E+00		
G13	9.66E-01	8.02E-02	1.02E+00	4.55E-01	9.07E-01	1.58E-01	1.08E+00	9.76E-01		

4.1.1.3. CTSA ile literatürdeki diğer yöntemlerin karşılaştırılması

Bu alt bölümde, ortalama sonuçları dikkate alarak önerilen CTSA; PSO, ABC, GA ve DE algoritmaları ile karşılaştırılmıştır. GA için sonuçlar doğrudan Akay'ın (Akay, 2009) çalışmasından alınmış olup diğer algoritmaların kontrol parametreleri bu çalışmadaki gibi alınmıştır. İlgili çalışmada kullanılan GA'nın parametreleri ve çalışma metodu yeterince açık anlatılmadığından yeniden kodlanamamıştır. GA hariç tüm algoritmalar Wilcoxon işaretli sıra testi ve Friedman'ın testine tabi tutularak istatistiksel analizler yapılmıştır. Algoritmalar rastgele başlangıç koşullarıyla 30 kez çalıştırılmış ve tüm yöntemler için sonlandırma koşulu maxFEs olarak $2.4E+5$ 'e ayarlanmıştır. Tüm algoritmalara CTSA'deki gibi Deb'in kuralları entegre edilmiştir. Eşit şartlar sağlanarak algoritmaların sonuçları karşılaştırılıp kısıtlı optimizasyon için CTSA'nın performansı olabildiğince net bir şekilde ortaya konmuştur. Ortalama bazlı karşılaştırma sonuçları Çizelge 4.15'te verilmiştir.

Çizelge 4.15. CTSA, PSO, GA, DE ve ABC algoritmalarının karşılaştırma sonuçları

Problem	Optimal	ABC	PSO	GA	DE	CTSA
G01	-15.000	-15.0205(2)	-10.5551(5)	-14.236(4)	-14.2406(3)	-15.0000(1)
G02	0.803619	0.4795(4)	0.4043(5)	0.788588(2)	0.6660(3)	0.8005(1)
G03	1.000	3.0191(5)	1.1675(3)	0.976(2)	1.1694(4)	1.0158(1)
G04	-30665.539	-30610.974(3)	-30661.740(2)	-30590.455(4)	-30665.540(1)	-30665.540(1)
G05	5126.498	5115.056(1)	5298.284(3)	N/A(5)	5329.197(4)	5172.377(2)
G06	-6961.814	-7579.630(5)	-6961.819(2)	-6872.204(3)	-6765.482(4)	-6961.816(1)
G07	24.306	29.0956(4)	28.7418(3)	34.980(5)	24.3160(1)	24.5008(2)
G08	0.095825	6.5347(3)	0.0847(2)	0.0958(1)	0.0958(1)	0.0958(1)
G09	680.630	683.8941(4)	680.7815(3)	692.064(5)	680.6308(1)	680.6433(2)
G10	7049.250	7259.028(3)	8128.793(4)	10003.225(5)	7162.592(2)	7116.206(1)
G11	0.750	0.7171(3)	0.7626(2)	0.750(1)	0.9545(5)	0.8003(4)
G12	1.000	1.0001(2)	1.000(1)	1.000(1)	1.000(1)	1.000(1)
G13	0.05395	0.0955(1)	1.4228(4)	N/A(5)	0.9492(2)	0.9671(3)
Ortalama Sıra		3.08 (40/13)	3 (39/13)	3.31(43/13)	2.46 (32/13)	1.62 (21/13)

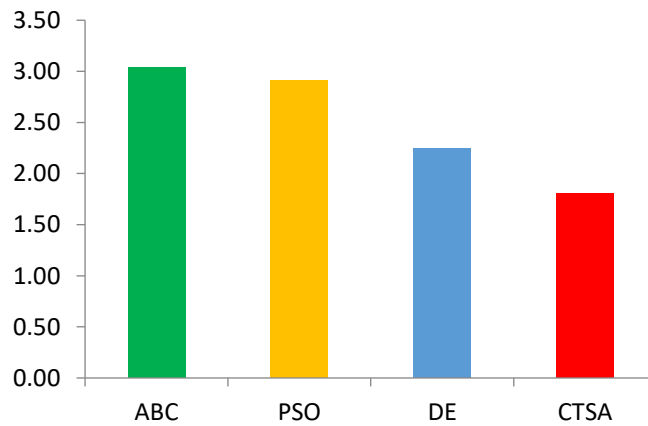
CTSA 1.62 ortalama sıra değeri ile 2.46 sıra değeri ile en yakın rakibi olan DE'yi geride bırakmıştır. PSO 3, ABC 3.08 ve GA 3.31 değerleriyle sıralanmışlardır. 13 problemin sekiz tanesini birincilikle çözen CTSA, etkinliğini ortaya koymuştur. Tüm yöntemler G12 problemi için optimum değeri bulmuştur. CTSA, G02 ve G10 problemleri üzerinde diğer yöntemlerden açık ara daha iyidir. DE, G07 probleminde PSO ise G05 ve G06 problemlerinde diğer algoritmalarından daha iyi sonuçlar üretmiştir.

Wilcoxon işaretli sıra testinin p-değeri 0.05 alınmış ve sonuçlar Çizelge 4.16'da verilmiştir. Sonuçlar arasında istatistiksel olarak anlamlı farklar olduğu gösterilmiştir.

Çizelge 4.16. Wilcoxon işaretli sıra testi sonuçları

CTSA vs.	ABC		PSO		DE	
	p-value	sign	p-value	sign	p-value	sign
G01	1.73E-06	(+)	1.73E-06	(+)	6.73E-01	(-)
G02	1.73E-06	(+)	1.73E-06	(+)	2.60E-06	(+)
G03	4.90E-04	(+)	1.73E-06	(+)	1.73E-06	(+)
G04	3.18E-06	(+)	2.07E-02	(+)	1.73E-06	(+)
G05	1.73E-06	(+)	9.30E-03	(+)	3.11E-05	(+)
G06	4.80E-04	(+)	3.35E-07	(+)	4.50E-06	(+)
G07	1.73E-06	(+)	3.18E-06	(+)	1.73E-06	(+)
G08	2.60E-05	(+)	6.25E-02	(-)	1.00E+00	(-)
G09	1.73E-06	(+)	6.89E-05	(+)	1.92E-06	(+)
G10	4.53E-04	(+)	1.02E-05	(+)	3.88E-04	(+)
G11	1.73E-06	(+)	1.11E-02	(+)	4.73E-06	(+)
G12	1.73E-06	(+)	3.74E-05	(+)	8.89E-06	(+)
G13	1.73E-06	(+)	1.78E-01	(-)	1.20E-01	(-)

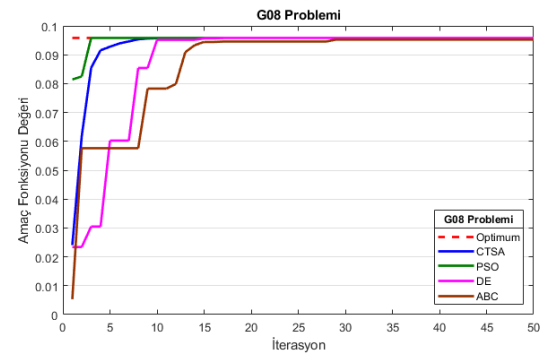
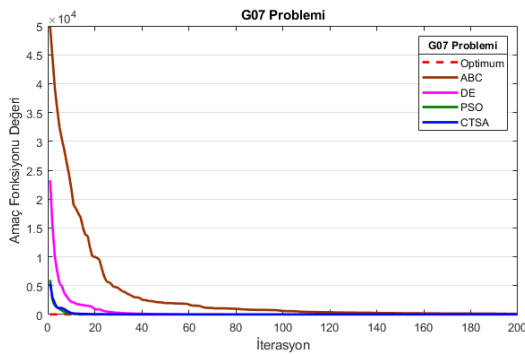
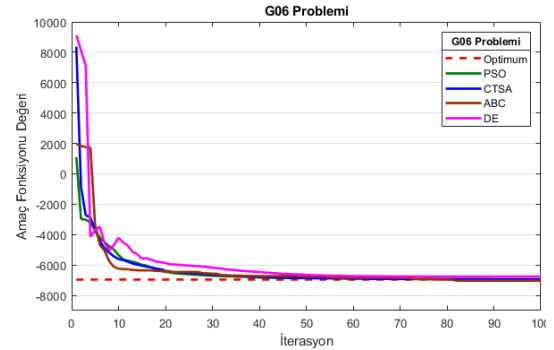
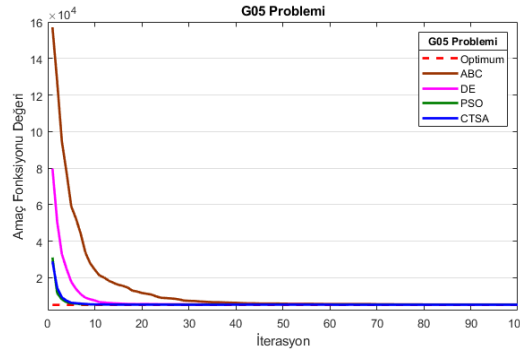
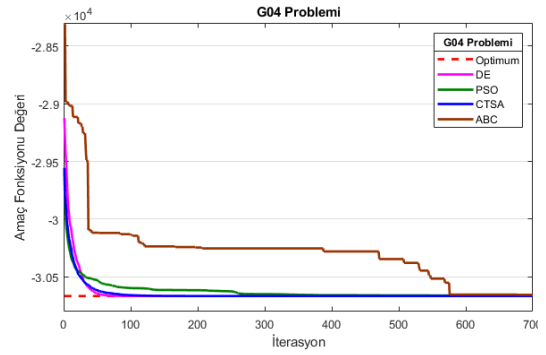
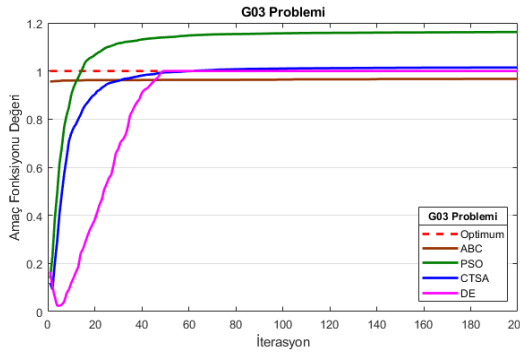
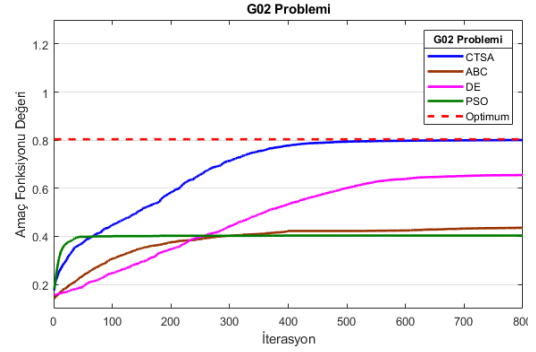
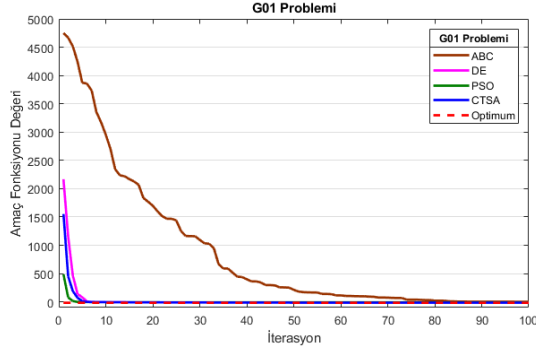
Friedman testinin sonuçları ise Şekil 4.1'de verilmiştir. CTSA 1.81, DE 2.24, PSO 2.91 ve ABC 3.04 değerleri ile sıralanmıştır.

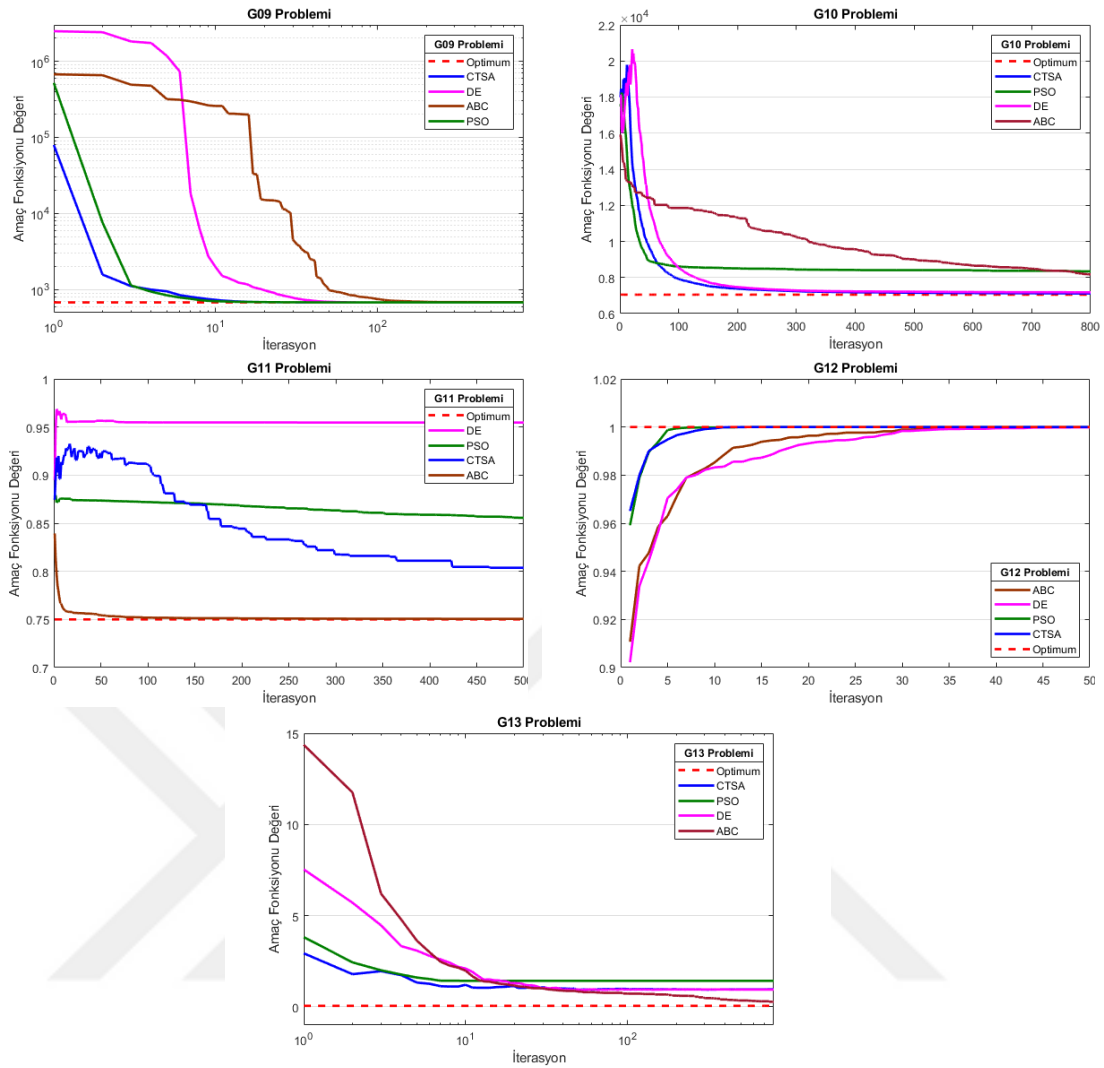


Şekil 4.1. Friedman sıra testinin sonuçları

Ayrıca CTSA için $N=40$ ve $ST=0.2$ alınarak hazırlanmış olan yakınsama grafikleri Şekil 4.2'de görülmektedir. Her bir problem için ayrı ayrı yakınsama grafikleri hazırlanmış, optimum değerler kesikli kırmızı çizgi ile belirtilerek yorumlanması kolaylaştırılmıştır. G03, G06, G10, G11 ve G13 problemleri için hazırlanmış olan yakınsama grafiklerindeki dalgalanmaların sebebi, mevcut iterasyondaki en iyi bireylerin seçiminin Deb'in kuralları doğrultusunda yapılmasıdır. Her iterasyonda en iyi çözüm yeniden belirlenmekte, geçmişteki en iyi çözüm saklanmadığından yakınsama

grafiklerinde gözlenen dalgalanmalar oluşmaktadır. CTSA, DE ve PSO daha erken yakınsama sağlarken, ABC her iterasyonda sadece bir boyut güncellediğinden dolayı daha geç yakınsamaktadır. Sonuç olarak CTSA'nın makul zaman içerisinde optimum ve yakın optimum sonuçlara yakınsadığı ortaya konmuştur.





Şekil 4.2. Algoritmaların standart kıyaslama problemleri üzerinde yakınsama grafikleri

4.1.2. Mühendislik tasarım problemleri ile yapılan deneylerin sonuçları

Bu bölümde CTSA'nın performansını ölçmek için kullanılan mühendislik tasarım problemleri ile yapılan deneysel çalışmalara yer verilmiştir. PVDP ve kaynaklı kiriş dizayn problemine (WBDP) ait birer farklı versiyona ait problem bilgileri (matematiksel modelleri) EK-2'de sunulmuştur. Bu problemlerin bazı farklı versiyonları literatürdeki algoritmaların karşılaştırılmasında kullanıldığından, hem ceza temelli TSA (penalty based TSA, PTSA) hem de CTSA ilk önce bu problemlere uygulanmış ve daha sonra CTSA diğer algoritmalarla karşılaştırılmıştır.

Yukarıdaki deneysel çalışmalar uyarınca PTSA ve CTSA'nın performansını analiz etmek için deneylerde N parametresi 20 ve 40 ve ST parametresi 0.2 ve 0.5 olarak alınmıştır. Sonlandırma koşulu, maksimum fonksiyon değerlendirme sayısıdır ve $3E+4$ olarak ayarlanmıştır. 30 bağımsız çalıştırmanın en iyi, en kötü ve ortanca değerleri

Çizelge 4.17'de raporlanmıştır. Tasarım problemi olduğundan kıyaslama en iyi değerlere göre yapılmıştır. Çizelge 4.17'ye incelendiğinde, CTSA, WBDP'lerin çözülmesinde PTSA'dan daha iyi ve PTSA, PVDP'lerin çözülmesinde CTSA'dan daha iyi sonuç vermiştir. ST, 0.5 ve N, 20 olduğunda iki algoritma benzer sonuçlar üretmiştir.

Çizelge 4.17. PTSA ve CTSA'nın mühendislik tasarım problemleri üzerinde karşılaştırılması

ST=0.2. N=40	PTSA			CTSA		
	En İyi	En Kötü	Ortanca	En İyi	En Kötü	Ortanca
PVDP-V1	6029.8649	6474.1792	6150.8781	6053.1729	6398.9258	6194.5821
PVDP-V2	7198.3806	7225.3487	7204.1104	7198.4885	7214.7302	7200.8153
WBDP-V1	2.3926034	2.6249205	2.4451424	2.3901825	2.5505803	2.4290572
WBDP-V2	1.6987855	1.7825596	1.7061327	1.6962002	1.7871216	1.7066324
ST=0.5. N=20	En İyi	En Kötü	Ortanca	En İyi	En Kötü	Ortanca
PVDP-V1	5961.9491	6292.7948	6056.5666	5930.4781	6459.0682	6104.2594
PVDP-V2	7197.8352	7214.536	7199.1896	7197.9606	7218.7861	7201.0789
WBDP-V1	2.3820242	2.4120713	2.3912587	2.382411	2.4807722	2.3891574
WBDP-V2	1.6954364	1.7161444	1.6967409	1.6954304	1.7057972	1.6968124

4.1.2.1. CTSA ile literatürdeki diğer yöntemlerin karşılaştırılması

Bu bölümde, önceki alt bölümde yapılan analizlere bağlı olarak CTSA'nın N ve ST parametreleri sırasıyla 20 ve 0.5 olarak alınmıştır. Güve-alev optimizasyonu (moth-flame optimization, MFO) (Mirjalili, 2015), membran hesaplamaya dayalı yaşamdan esinlenen algoritma (bio-inspired algorithm based on membrane computing, BIAMC) (Xiao ve ark., 2013), gri kurt algoritması (grey wolf optimizer, GWO) (Mirjalili ve ark., 2014a), modifiye DE (modified DE) (Mezura-Montes ve ark., 2007), CS (Gandomi ve ark., 2013), balina optimizasyon algoritması (whale optimization algorithm, WOA) (Mirjalili ve Lewis, 2016), MVO (Mirjalili ve ark., 2016), toplum ve medeniyet (Ray ve Liew) (Ray ve Liew, 2003a), AAA (Uymaz ve ark., 2015), geliştirilmiş HS (improved HS) (Mahdavi ve ark., 2007), HS (Lee ve Geem, 2005), basit optimizasyon (simple optimization, SOPT) (Hasançebi ve Azad, 2012), büyük patlama büyük çöküş algoritması (big-bang big-crunch algorithm, BBBCA) (Kazemzadeh Azad ve ark., 2011), GA (Wu ve Chow, 1995), dinamik stokastik seçimli DE (differential evolution with dynamic stochastic selection, DSS-MDE) (Zhang ve ark., 2008), genetik algoritmalar (Deb) (Deb, 1991), kültürel algoritma (cultural algorithm, CA) (Yan ve ark., 2012a), FA (Gandomi ve ark., 2011) ve ışın optimizasyonu (ray optimization, RO) (Kaveh ve Khayatizad, 2012) ile PVDP ve WBDP'nin çözümleri referansları verilen çalışmalardan toplanmış ve sonuçlar PVDP-V1 için Çizelge 4.18'de, PVDP-V2 için Çizelge 4.19'da, WBDP-V1 için

Çizelge 4.20'de ve WBDP-V2 için Çizelge 4.21'de verilmiştir. Bu tablolarda, algoritmalar ile elde edilen en iyi çözümler birbiriyle karşılaştırılmıştır.

Çizelge 4.18. PVDP-V1 problemi üzerindeki karşılaştırma sonuçları

Sıra	Yöntem	x_1 (Ts)	x_2 (Th)	x_3 (R)	x_4 (L)	f(x)	Fes	Referans
1	CTSA	0.79626855	0.39513591	41.19931831	188.16101690	5930.47808677	30000	-
2	BIAMC	0.82759900	0.41379400	42.70313700	169.96525400	6029.18105900	80000	(Xiao ve ark., 2013)
3	GWO	0.81250000	0.43450000	42.08918100	176.75873100	6051.56390000	N/A	(Mirjalili ve ark., 2014a)
4	modified DE	0.81250000	0.43750000	42.09844600	176.63604700	6059.70166000	24000	(Mezura-Montes ve ark., 2007)
5	MFO	0.81250000	0.43750000	42.09844500	176.63659600	6059.71430000	N/A	(Mirjalili, 2015)
6	CS	0.81250000	0.43750000	42.09844560	176.63659580	6059.71433480	375000	(Gandomi ve ark., 2013)
7	WOA	0.81250000	0.43750000	42.09826990	176.63899800	6059.74100000	6300	(Mirjalili ve Lewis, 2016)
8	MVO	0.81250000	0.43750000	42.09073820	176.73869000	6060.80660000	N/A	(Mirjalili ve ark., 2016)
9	Ray and Liew	0.81250000	0.43750000	41.97680000	182.28450000	6171.00000000	12630	(Ray ve Liew, 2003b)

Çizelge 4.19. PVDP-V2 problemi üzerindeki karşılaştırma sonuçları

Sıra	Yöntem	x_1 (Ts)	x_2 (Th)	x_3 (R)	x_4 (L)	f(x)	Fes	Referans
1	AAA	1.25000000	0.62500000	58.29015540	43.69265650	7197.72893000	30000	(Uymaz ve ark., 2015)
2	Improved HS	1.12500000	0.62500000	58.29015000	43.69268000	7197.73000000	200000	(Mahdavi ve ark., 2007)
3	CTSA	1.12506871	0.62501470	58.29343297	43.67539950	7197.96056849	30000	-
4	HS	1.12500000	0.62500000	58.27890000	43.75490000	7198.43300000	N/A	(Lee ve Geem, 2005)
5	SOPT	1.12500000	0.62500000	58.29020000	43.69270000	7199.35900000	10000	(Hasançebi ve Azad, 2012)
6	BBBCA	1.12500000	0.62500000	58.28950000	43.69640000	7199.41200000	25000	(Kazemzadeh Azad ve ark., 2011)
7	GA	1.12500000	0.62500000	58.19780000	44.29300000	7207.49400000	N/A	(Wu ve Chow, 1995)

Çizelge 4.20. WBDP-V1 problemi üzerindeki karşılaştırma sonuçları

Sıra	Yöntem	x_1 (h)	x_2 (l)	x_3 (t)	x_4 (b)	f(x)	Fes	Referans
1	HS	0.24420000	6.22310000	8.29150000	0.24430000	2.38070000	110000	(Lee ve Geem, 2005)
2	DSS-MDE	0.24436898	6.21751972	8.29147139	0.24436898	2.38095658	24000	(Zhang ve ark., 2008)
3	CTSA	0.24415742	6.22306595	8.29555011	0.24440474	2.38241101	30000	-
4	Ray and Liew	0.24443828	6.23796723	8.28857614	0.24456618	2.38543470	33095	(Ray ve Liew, 2003b)
5	Deb	0.24890000	6.17300000	8.17890000	0.25330000	2.43311600	5000	(Deb, 1991)

Çizelge 4.21. WBDP-V2 problemi üzerindeki karşılaştırma sonuçları

Sıra	Yöntem	x_1 (h)	x_2 (l)	x_3 (t)	x_4 (b)	f(x)	Fes	Referans
1	CTSA	0.20561590	3.25511618	9.03720651	0.20572699	1.69543039	30000	-
2	MFO	0.20570000	3.47030000	9.03640000	0.20570000	1.72452000	N/A	(Mirjalili, 2015)
3	Improved HS	0.20573000	3.47049000	9.03662000	0.20573000	1.72480000	300000	(Mahdavi ve ark., 2007)
4	BIAMC	0.20567500	3.47099300	9.04058700	0.20572800	1.72550700	80000	(Xiao ve ark., 2013)
5	BBBCA	0.20540000	3.47830000	9.03860000	0.20570000	1.72576000	20000	(Kazemzadeh Azad ve ark., 2011)
6	GWO	0.20567600	3.47837700	9.03681000	0.20577800	1.72624000	N/A	(Mirjalili ve ark., 2014a)
7	MVO	0.20546300	3.47319300	9.04450200	0.20569500	1.72645000	N/A	(Mirjalili ve ark., 2016)
8	CA	0.20236900	3.54421400	9.04821000	0.20572300	1.72802400	100000	(Yan ve ark., 2012b)
10	WOA	0.20539600	3.48429300	9.03742600	0.20627600	1.73049900	9900	(Mirjalili ve Lewis, 2016)
11	FA	0.20150000	3.56200000	9.04140000	0.20570000	1.73121000	50000	(Gandomi ve ark., 2011)
12	RO	0.20368700	3.52846700	9.00423300	0.20724100	1.73534400	N/A	(Kaveh ve Khayatizad, 2012)

Karşılaştırma sonuçlarına dayanarak CTSA'nın mühendislik tasarım problemlerini çözmeye rekabetçi olduğu ve PVDP-V1 ve WBDP-V2'yi çözmeye konusunda çözüm kalitesi açısından diğer algoritmalarından daha iyi olduğu söylenebilir. AAA ve geliştirilmiş HS algoritması, PVDP-V2'yi çözmeye konusundaki CTSA ve diğer algoritmalarından biraz daha iyidir. WBDP-V1'de ise CTSA 3.sıradadır fakat birinci olan HS'nin maxFEs değeri 110E+3 olduğundan tam adil bir karşılaştırma yapılamamıştır. Sonuç olarak, CTSA mühendislik tasarım problemlerinin çözümü için umut verici ve karşılaştırılabilir sonuçlar üretmiştir.

4.2. İkili TSA'ların Deneysel Sonuçları ve Karşılaştırmaları

İkili TSA'ların performans ölçümü için Çizelge 3.5'te verilmiş olan 15 UFLP çözülmüş ve analizler için optimum sonuçtan fark yüzdesi (GAP) değeri Denklem 4.1 ile hesaplanarak alt bölümlerde yorumlanmıştır.

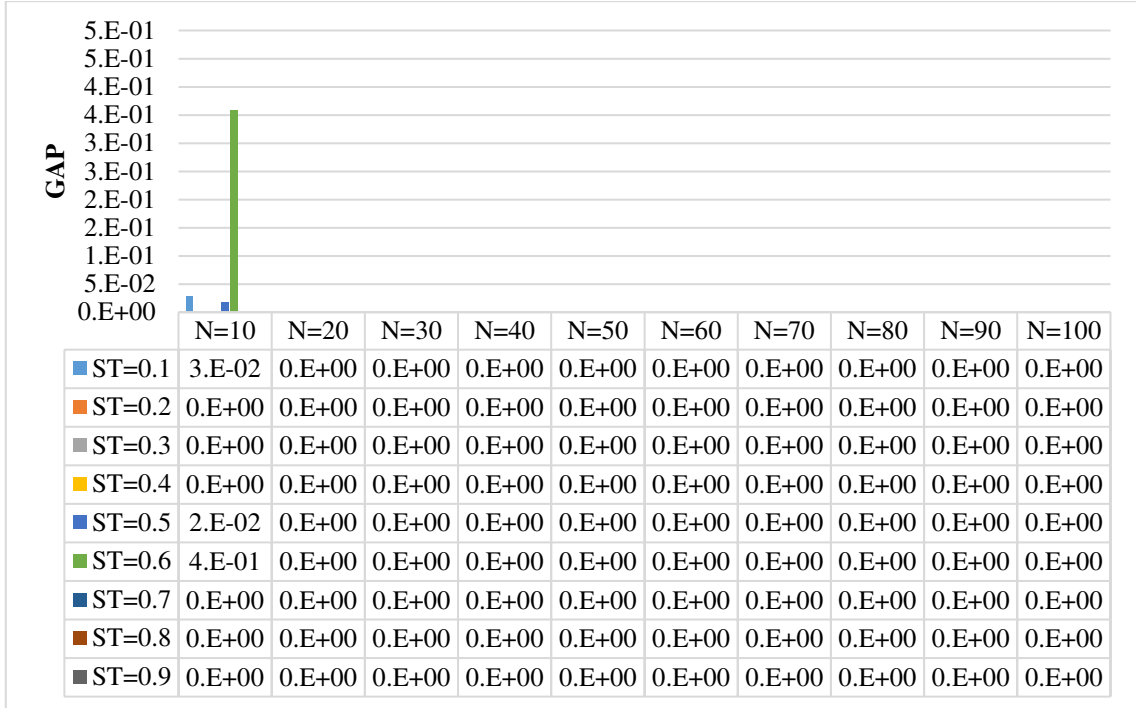
$$GAP = \frac{\text{Elde edilen en iyi çözüm} - \text{Optimum Çözüm}}{\text{Optimum Çözüm}} \times 100 \quad (4.1)$$

Elde edilen en iyi çözüm rastgele başlangıç noktaları ile 30 farklı çalıştırmalardan elde edilen sonuçların ortalamasıdır. Analiz aşamasında 100 boyutlu UFLP'ler (CapA, CapB, CapC) için, GAP değeri 1'den (göreceli hata ≤ 0.01) küçükse bu durumdaki koşullar (N ve ST) kabul edilebilir seviye olarak yorumlanmıştır. UFLP'lerin geri kalanında (Cap71-Cap134) ise GAP değeri 0 ise (göreceli hata = 0.00) algoritmanın olarak kabul edilebilir seviyede olduğu kabul edilmiştir.

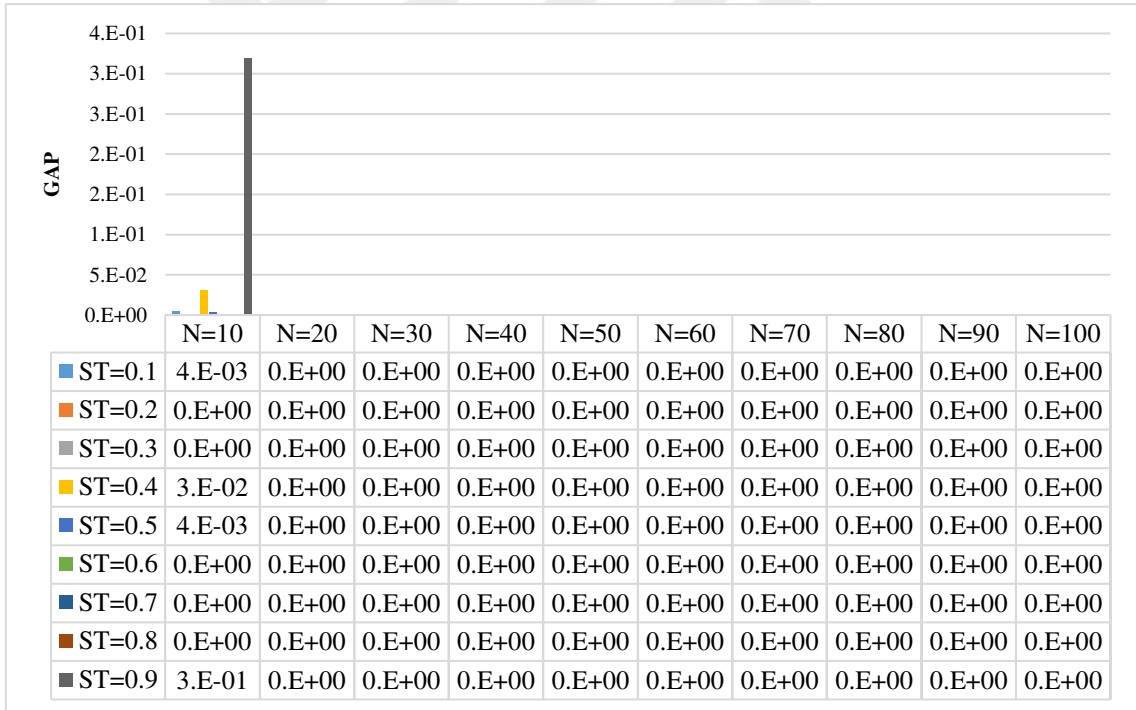
4.2.1. LogicTSA deneysel sonuçları ve karşılaştırmaları

Kontrol parametreleri olan ST ve N'nin LogicTSA'nın performansına etkilerini araştırmak için $ST = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ ve $N = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ alınarak algoritma rastgele başlangıç noktalarından başlatılarak 30 kez çalıştırılmıştır.

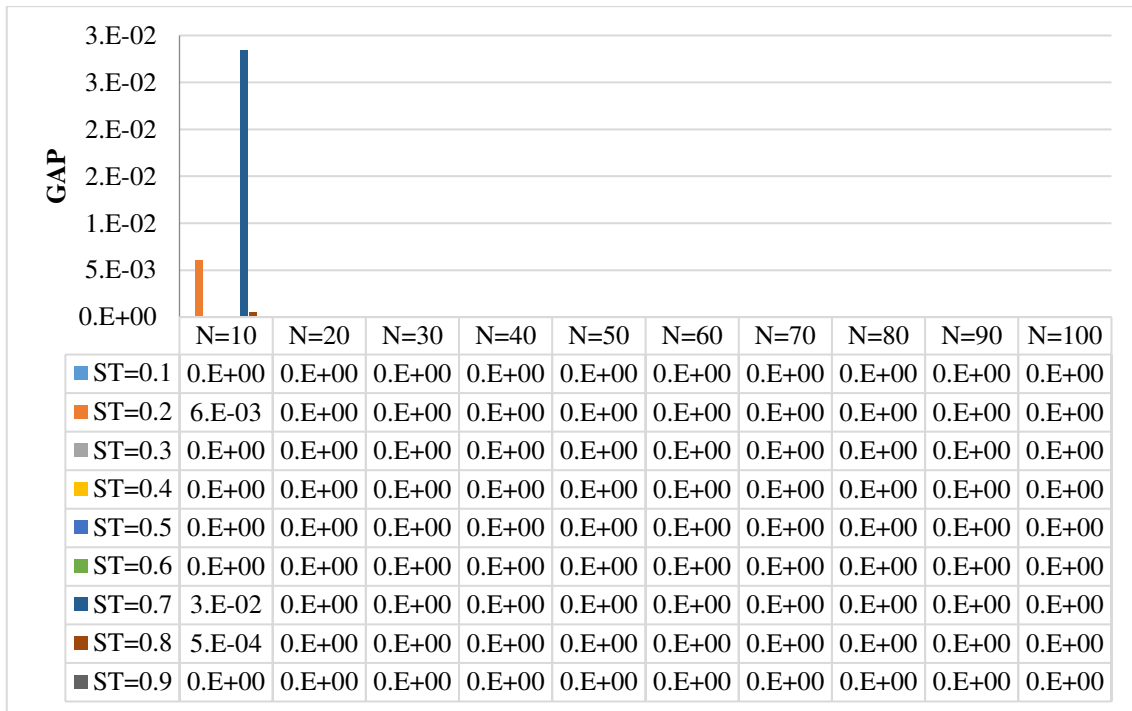
Potansiyel tesis yerlerinin sayısı (16) nispeten düşük olan Cap71 (Şekil 4.3), Cap72 (Şekil 4.4), Cap73 (Şekil 4.5) ve Cap74 (Şekil 4.6) problemleri için üretilen sonuçlar incelendiğinde LogicTSA, N=10 seçildiği durum hariç diğer bütün durumlarda optimum çözümü bulmuştur. Düşük ağaç sayısı durağanlaşmaya neden olmuş ve popülasyondaki ağaçlar birbirine benzediğinden kaliteli çözümler üretilmemiştir.



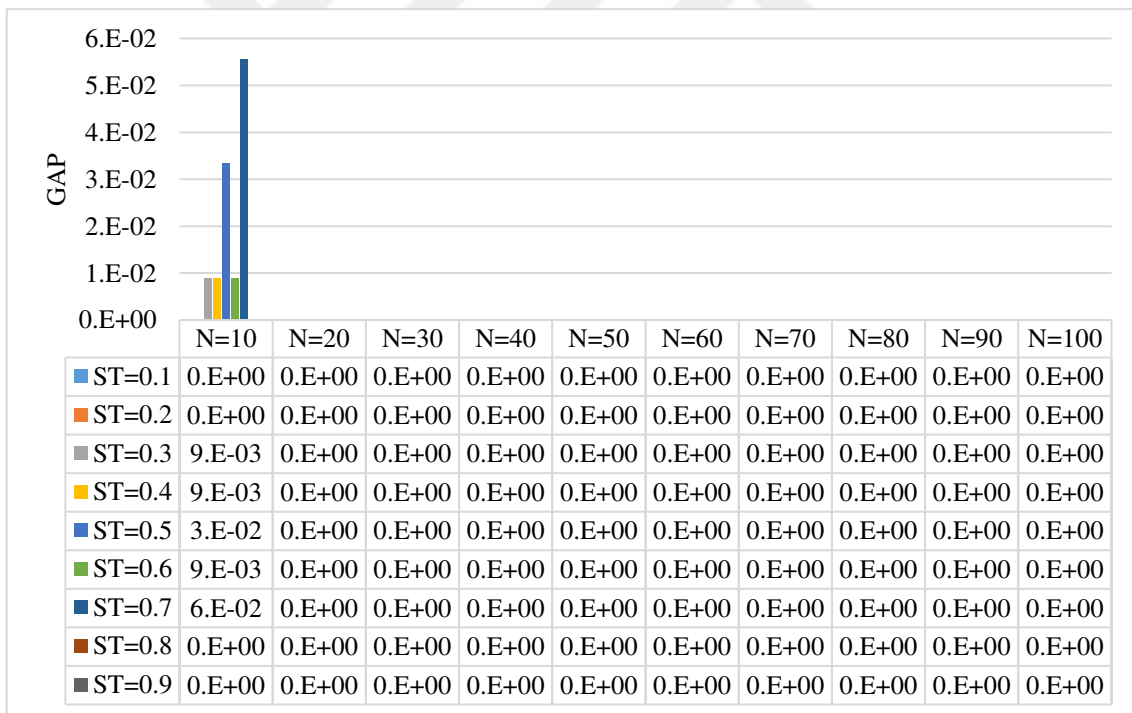
Şekil 4.3. LogicTSA'nın farklı ST ve N değerleri ile Cap71 problemi için ürettiği sonuçlar



Şekil 4.4. LogicTSA'nın farklı ST ve N değerleri ile Cap72 problemi için ürettiği sonuçlar



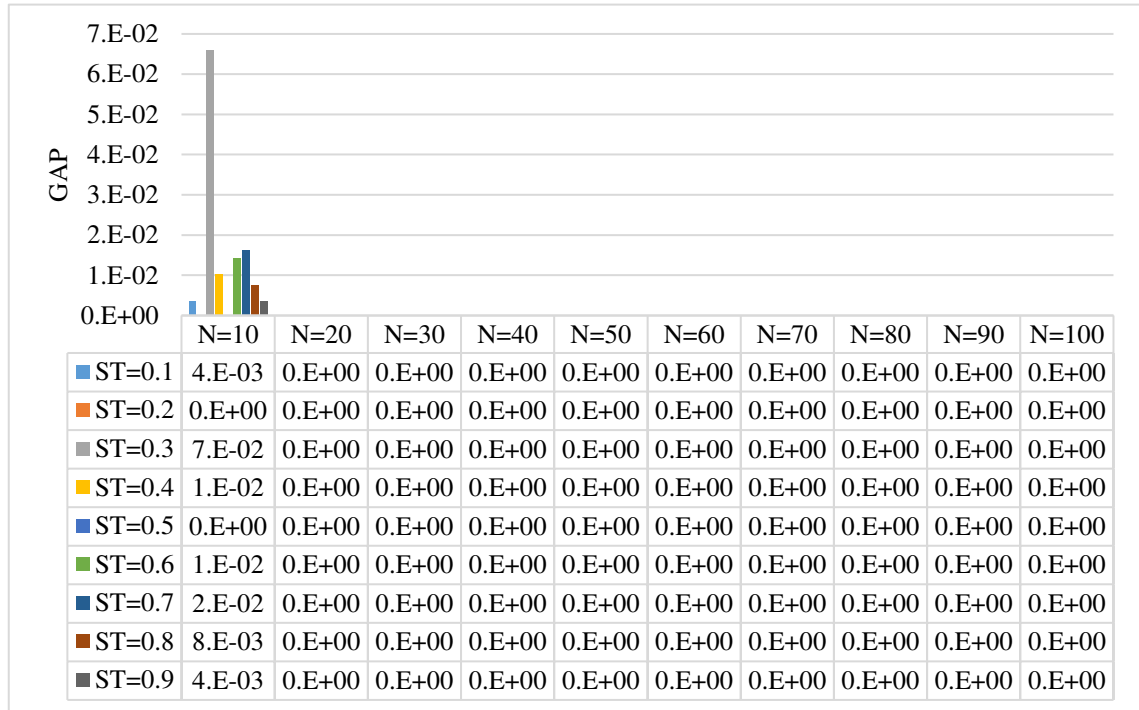
Şekil 4.5. LogicTSA'nın farklı ST ve N değerleri ile Cap73 problemi için ürettiği sonuçlar



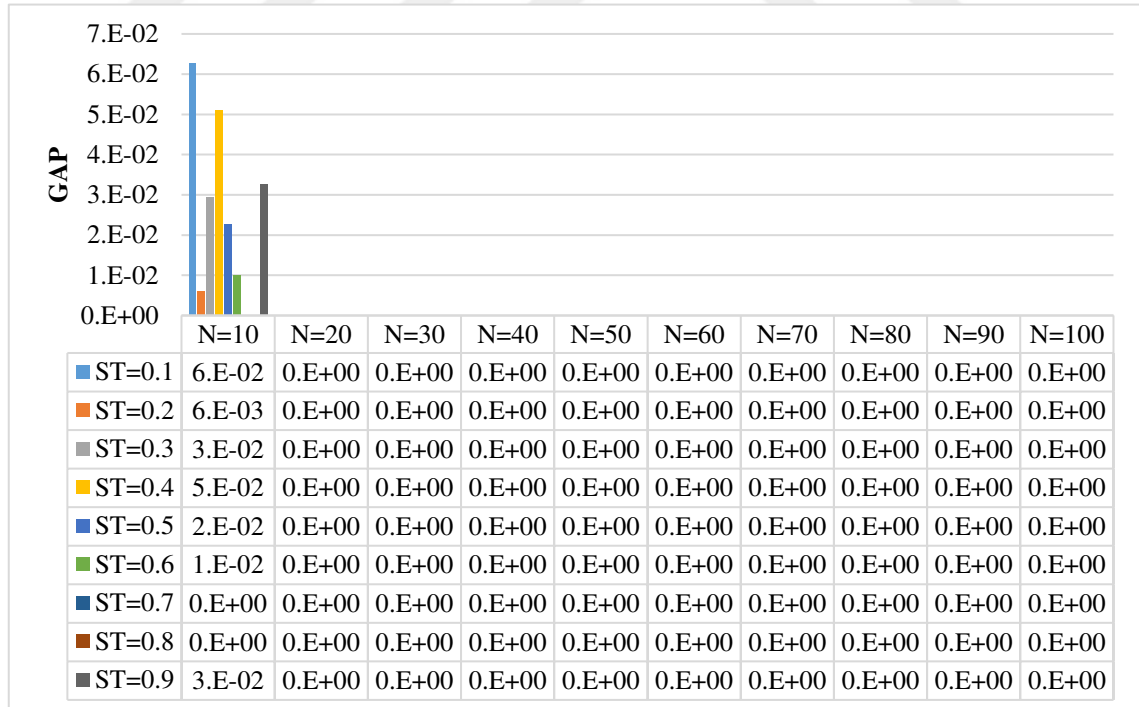
Şekil 4.6. LogicTSA'nın farklı ST ve N değerleri ile Cap74 problemi için ürettiği sonuçlar

25 ikili karar değişkenine sahip orta büyüklükteki problemler olan Cap101 (Şekil 4.7), Cap102 (Şekil 4.8), Cap103 (Şekil 4.9) ve Cap104 (Şekil 4.10) için üretilmiş sonuçlar incelendiğinde LogicTSA'nın bu problemler için de küçük problemlere benzer sonuçlar ürettiği görülmektedir. Bu testlerde de N=10 olduğu bazı ST değerlerinde

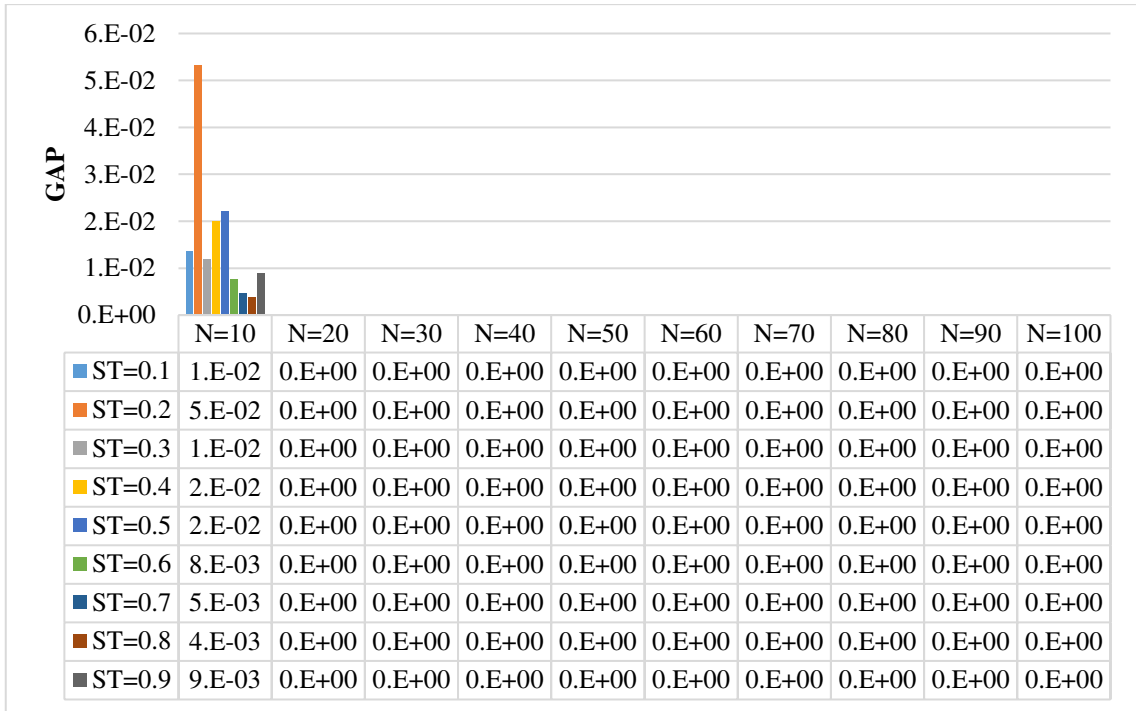
optimum çözüm elde edilememiştir. N parametresi 10'un üzerinde ($N > 10$) olduğunda optimum çözümler LogicTSA tarafından elde edilmiştir.



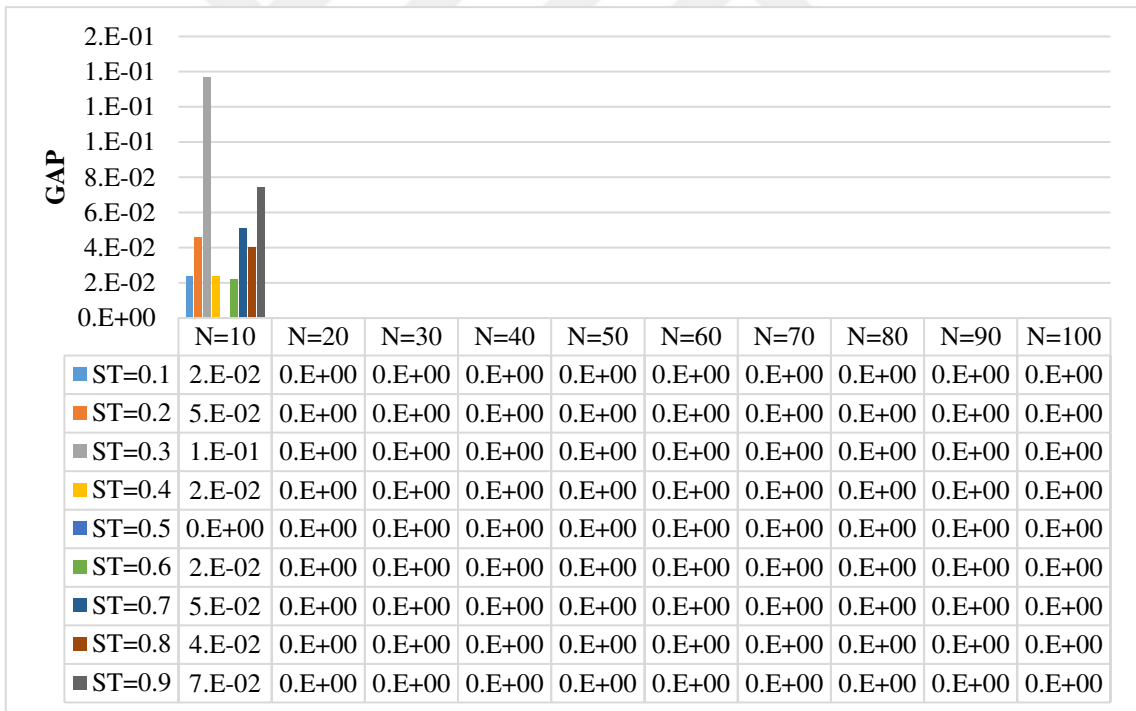
Şekil 4.7. LogicTSA'nın farklı ST ve N değerleri ile Cap101 problemi için ürettiği sonuçlar



Şekil 4.8. LogicTSA'nın farklı ST ve N değerleri ile Cap102 problemi için ürettiği sonuçlar



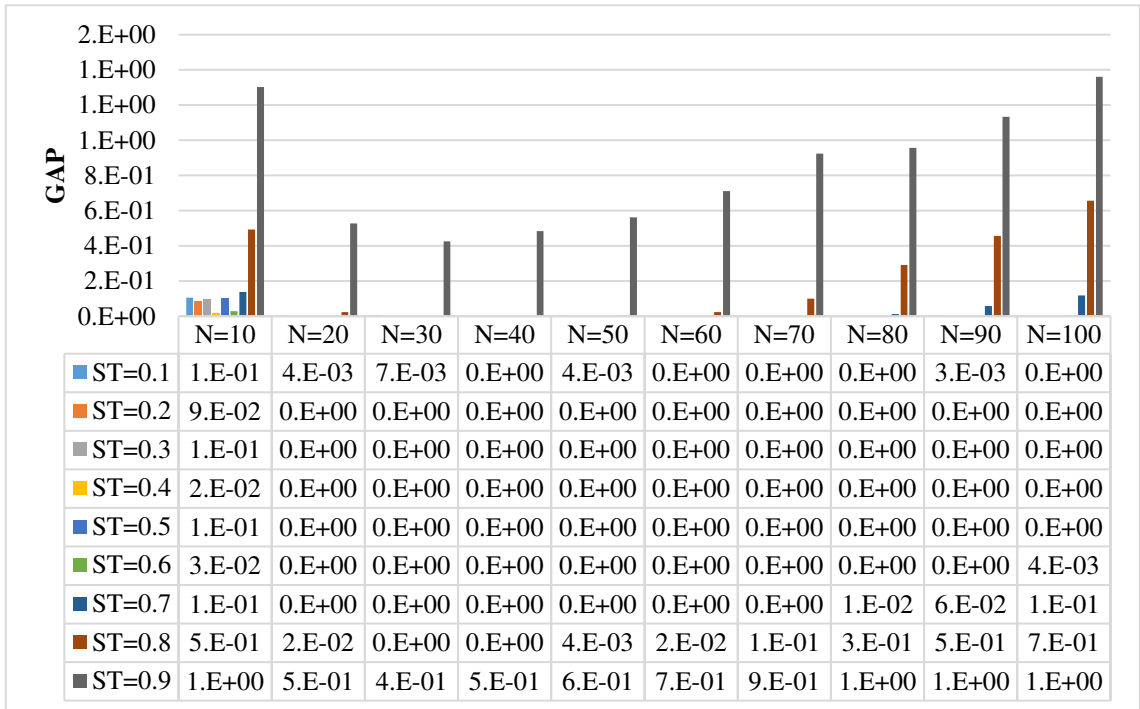
Şekil 4.9. LogicTSA'nın farklı ST ve N değerleri ile Cap103 problemi için ürettiği sonuçlar



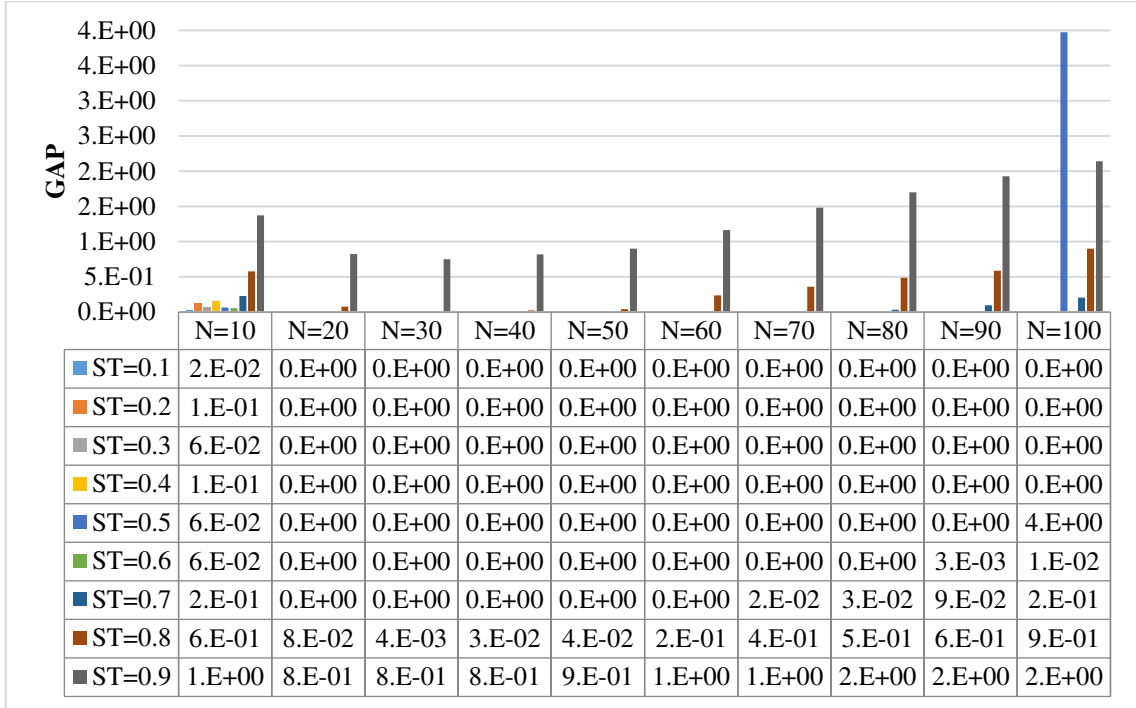
Şekil 4.10. LogicTSA'nın farklı ST ve N değerleri ile Cap104 problemi için ürettiği sonuçlar

Büyük boyutlu problemler olan ve optimize edilmesi gereken 50 karar değişkenine sahip olan Cap131 (Şekil 4.11), Cap132 (Şekil 4.12), Cap133 (Şekil 4.13) ve Cap134 (Şekil 4.14) için elde edilen sonuçlar incelendiğinde yüksek ve düşük popülasyon büyüklükleri ile LogicTSA'nın optimum çözümleri elde edemediği görülmüştür. Popülasyonun büyük olması güçlü global arama özelliği sağlamakta ancak sınırlı sayıda

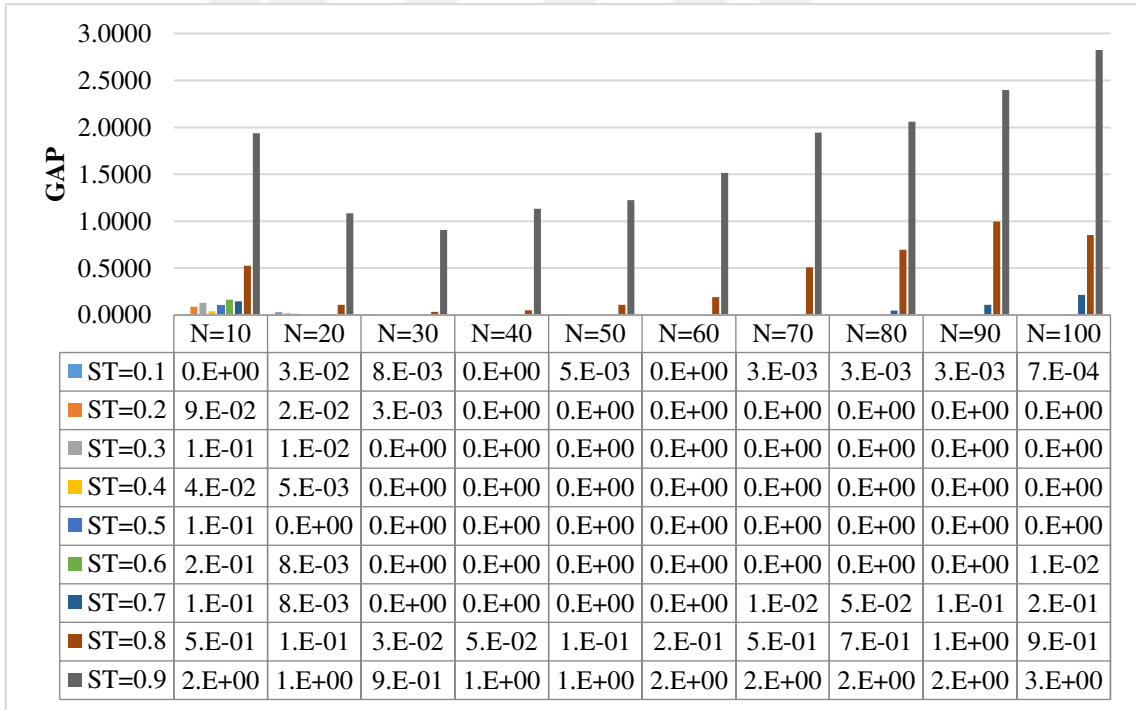
FEs olması, bulunan çözümler etrafındaki faydalanma için yeterli olmamaktadır. Popülasyon büyüklüğünün küçük olması ise faydalanma için yeterli olsa da keşif açısından yetersiz kalınmasına sebep olmaktadır ve süreçte ağaçlar birbirine benzemektedirler. Bu ağaçlardan üretilen yeni tohumlar ağaçlara benzediğinden ilerleme sağlamamaktadır. ST parametresi büyüdükçe çözümler, elde edilen en iyi çözümden daha fazla etkilenir ve bu durum erken yakınsamaya sebep olur, böylece tüm çözümler zaman içinde en iyi çözüme benzer ve çeşitlilik ortadan kalkar. Bu durum LogicTSA'da durağanlaşma davranışına neden olur. Sonuç olarak $ST=\{0.2,0.4\}$ arasında, $N=\{40,60\}$ arasında seçilirse büyük boyutlu problemlerin çözümünde LogicTSA daha kaliteli çözümler üretmektedir.



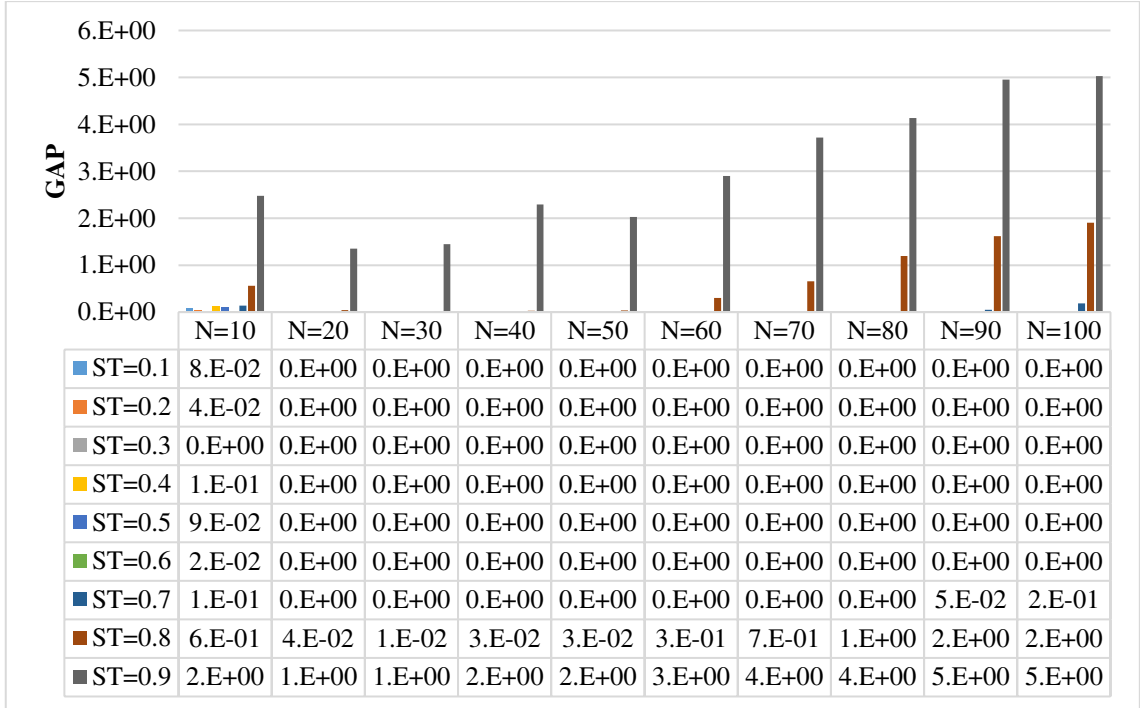
Şekil 4.11. LogicTSA'nın farklı ST ve N değerleri ile Cap131 problemi için ürettiği sonuçlar



Şekil 4.12. LogicTSA'nın farklı ST ve N değerleri ile Cap132 problemi için ürettiği sonuçlar

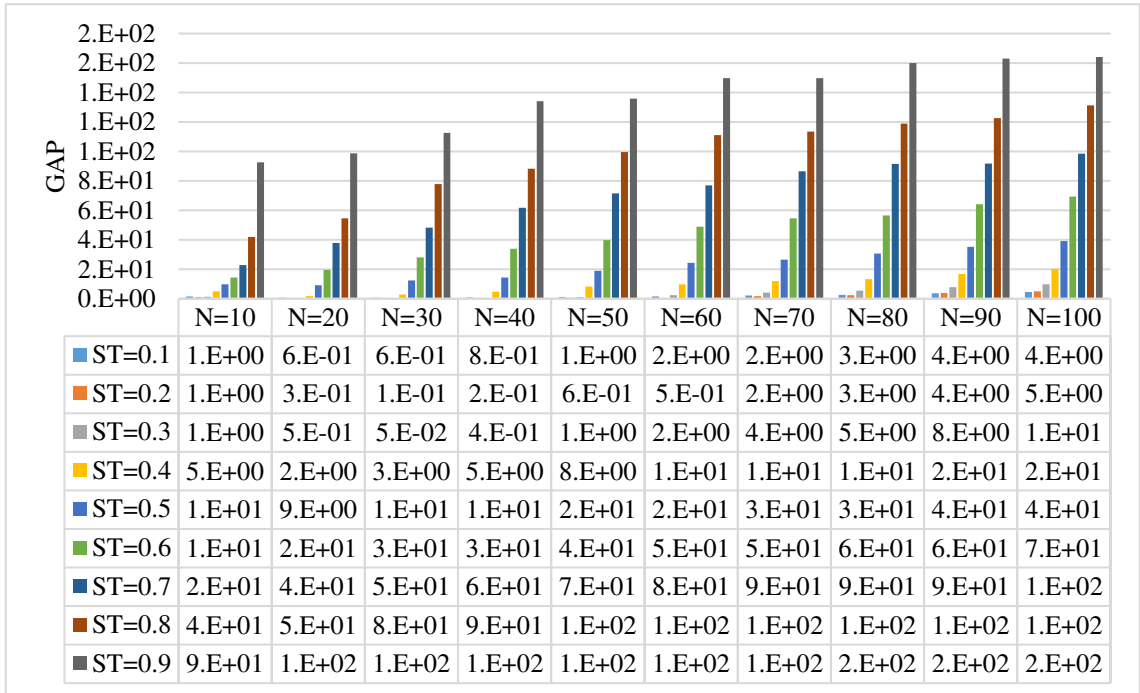


Şekil 4.13. LogicTSA'nın farklı ST ve N değerleri ile Cap133 problemi için ürettiği sonuçlar

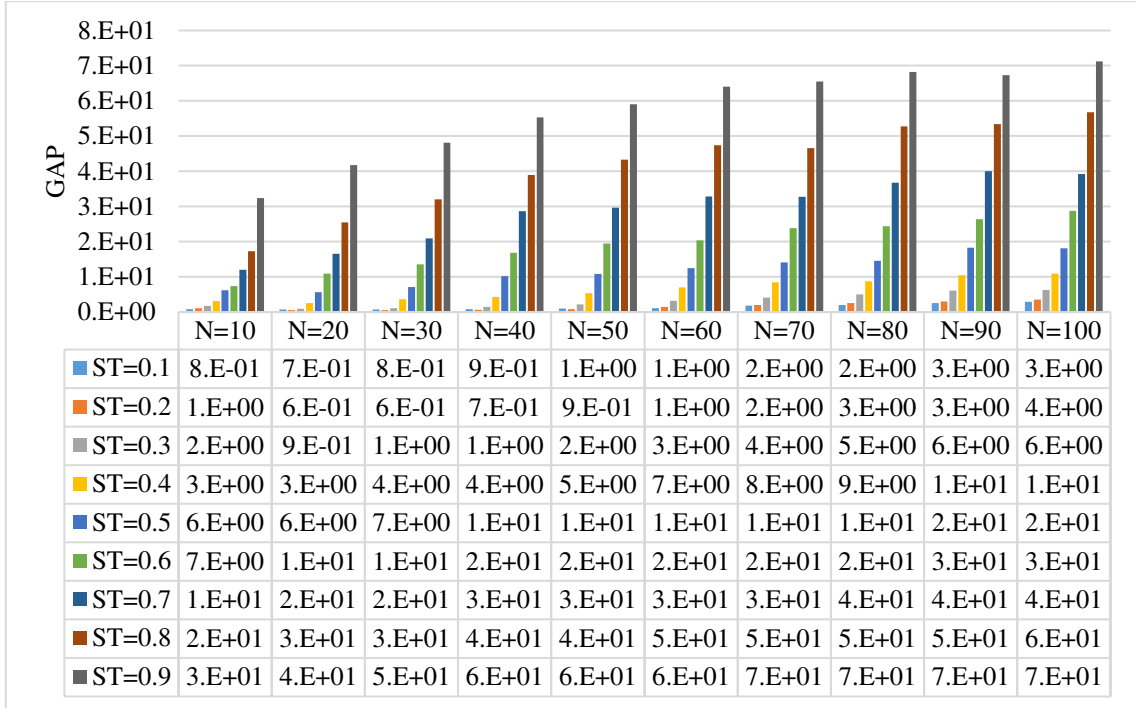


Şekil 4.14. LogicTSA'nın farklı ST ve N değerleri ile Cap134 problemi için ürettiği sonuçlar

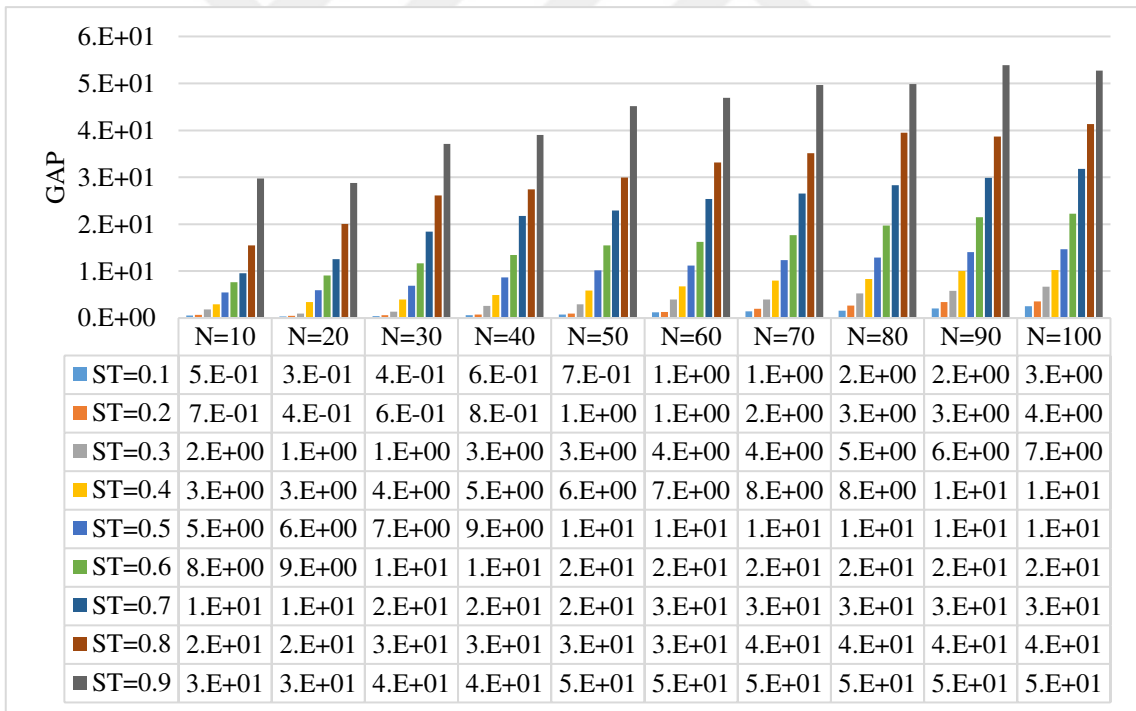
100 karar değişkenine sahip çok büyük boyutlu problemler olan CapA (Şekil 4.15), CapB (Şekil 4.16) ve CapC (Şekil 4.17)'nin $2^{100}-1$ potansiyel çözümü olduğundan optimum çözümleri bulmak için güçlü bir küresel ve yerel arama yeteneğine sahip olunması gerekir. Sonuçlar incelendiği $ST=\{0.1,0.2,0.3\}$ ve $N=\{20,30\}$ olarak seçildiğinde LogicTSA nispeten daha iyi sonuçlar üretmiştir.



Şekil 4.15. LogicTSA'nın farklı ST ve N değerleri ile CapA problemi için ürettiği sonuçlar

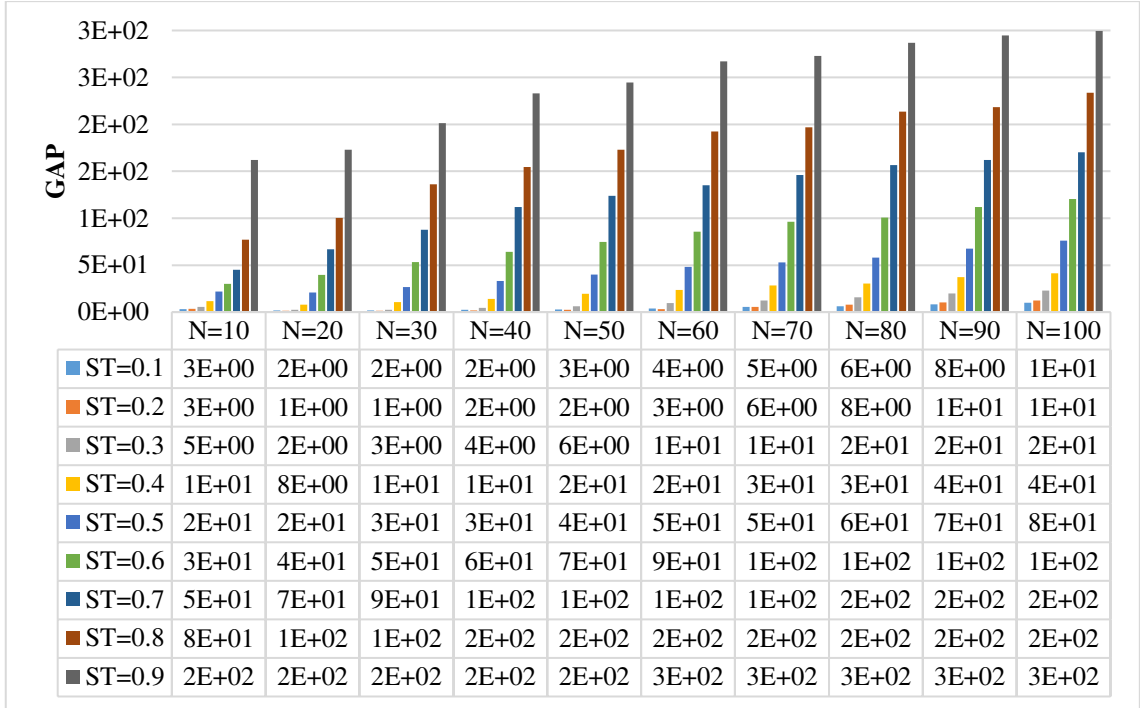


Şekil 4.16. LogicTSA'nın farklı ST ve N değerleri ile CapB problemi için ürettiği sonuçlar



Şekil 4.17. LogicTSA'nın farklı ST ve N değerleri ile CapC problemi için ürettiği sonuçlar

LogicTSA'nın tüm ST ve N değerleri ile ürettiği toplam GAP değerleri Şekil 4.18'de, toplam optimum çözüm sayıları ise Çizelge 4.22'de verilmiştir. Toplam GAP değerlerine göre ST ve N değerleri arttıkça sonuçların daha kötü olduğu, ST=0.2 ve N={20,30} olduğunda en iyi sonuçların üretildiği görülmektedir.



Şekil 4.18. LogicTSA'nın farklı ST ve N değerlerindeki toplam GAP değerleri

Toplam optimum çözüm sayılarına göre en iyi durum $ST=0.3$ ve $N=40$ olduğu durumda elde edilmiştir. Birbirine yakın çözümleri de kabul edilebilir olarak değerlendirdiğimiz vakit LogicTSA için $ST=\{0.2, 0.3, 0.4, 0.5\}$ ve $N=\{40,50,60\}$ alınması iyi sonuçların üretilmesini sağlayacaktır.

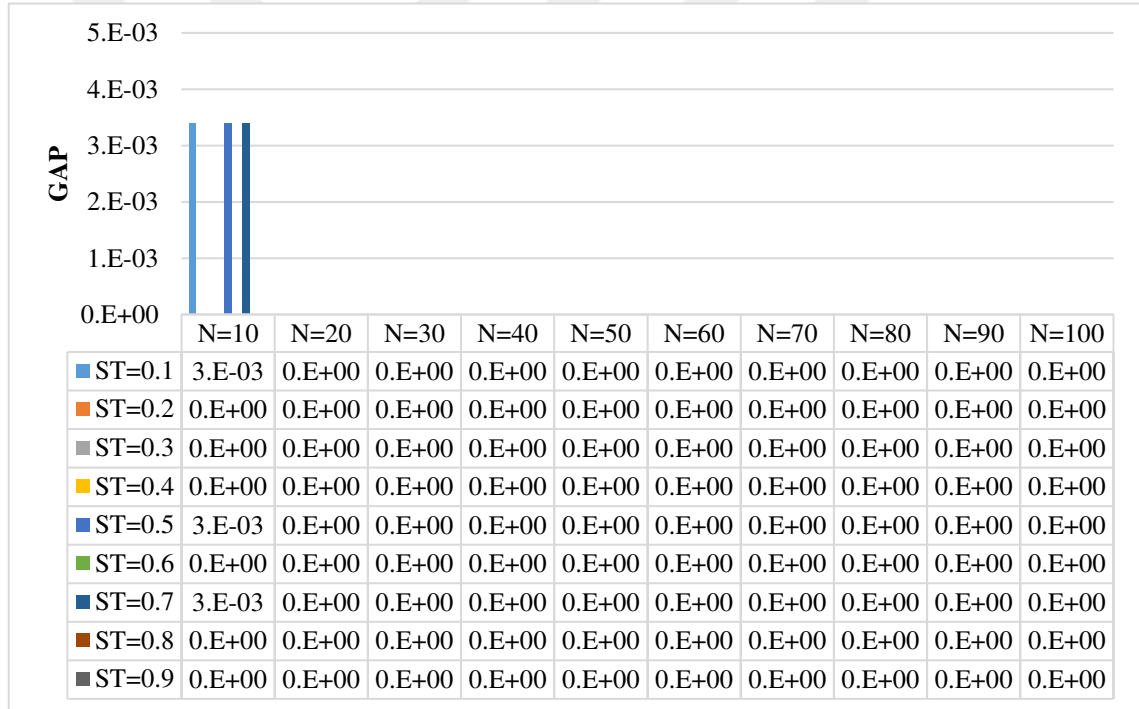
Çizelge 4.22. LogicTSA'da 15 UFLP problemi için toplam optimum çözüm sayısı

	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90	N=100	Toplam	Sıra
ST=0.1	3	10	10	12	10	12	11	11	10	11	100	5
ST=0.2	4	11	11	12	12	12	12	12	12	12	110	2
ST=0.3	4	11	12	12	12	12	12	12	12	12	111	1
ST=0.4	2	11	12	12	12	12	12	12	12	12	109	3
ST=0.5	3	12	12	12	12	12	12	12	12	11	110	2
ST=0.6	2	11	12	12	12	12	12	12	11	9	105	4
ST=0.7	3	11	12	12	12	12	10	9	8	8	97	6
ST=0.8	4	8	9	9	8	8	8	8	8	8	78	7
ST=0.9	3	8	8	8	8	8	8	8	8	8	75	8
Toplam	28	93	98	101	98	100	97	96	93	91	895	
Sıra	8	6	3	1	3	2	4	5	6	7		

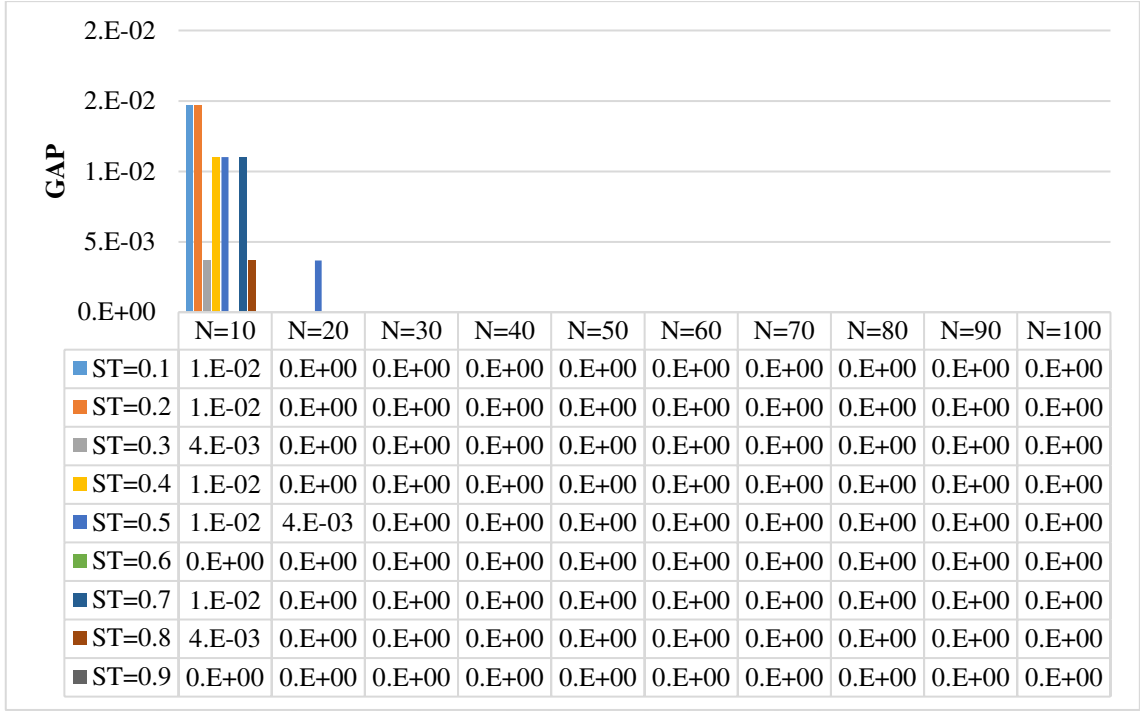
4.2.2. SimTSA deneysel sonuçları ve karşılaştırmaları

TSA'nın kontrol parametreleri olan ST ve N'nin SimTSA'nın performansına etkilerini araştırmak için $ST=\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ ve $N=\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ alınarak algoritma rastgele başlangıç noktalarından başlatılarak 30 kez çalıştırılmıştır.

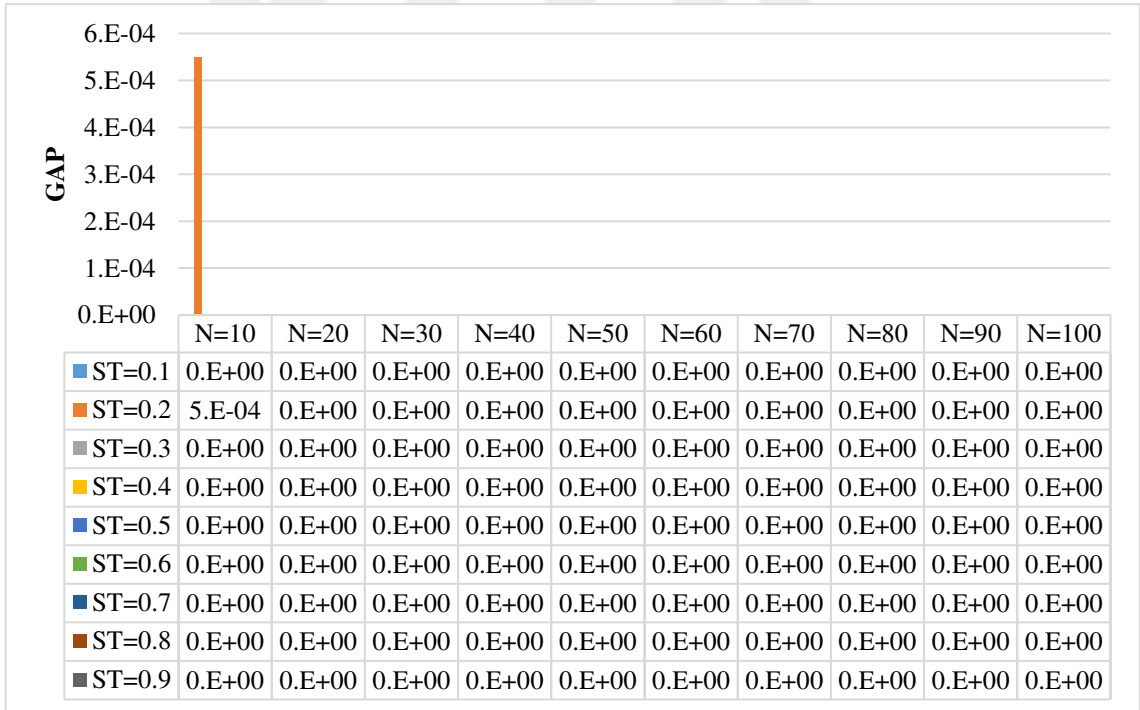
SimTSA'nın Cap71 (Şekil 4.19) ve Cap72 (Şekil 4.20) problemleri için elde ettiği sonuçlara göre, N parametresi 10'dan büyük olduğu durumlarda optimum çözümler elde edilmiştir. Cap73 (Şekil 4.21) probleminde $ST=0.2$ ve $N=10$ iken optimum çözüme ulaşılabilmiş diğer tüm durumlarda optimum çözüm elde edilmiştir. Cap74 (Şekil 4.22) probleminde $ST=0.7$ ve $N=10$ iken optimum çözüm elde edilememiştir. Bu durum haricindeki tüm durumlarda optimum çözüme ulaşılmıştır.



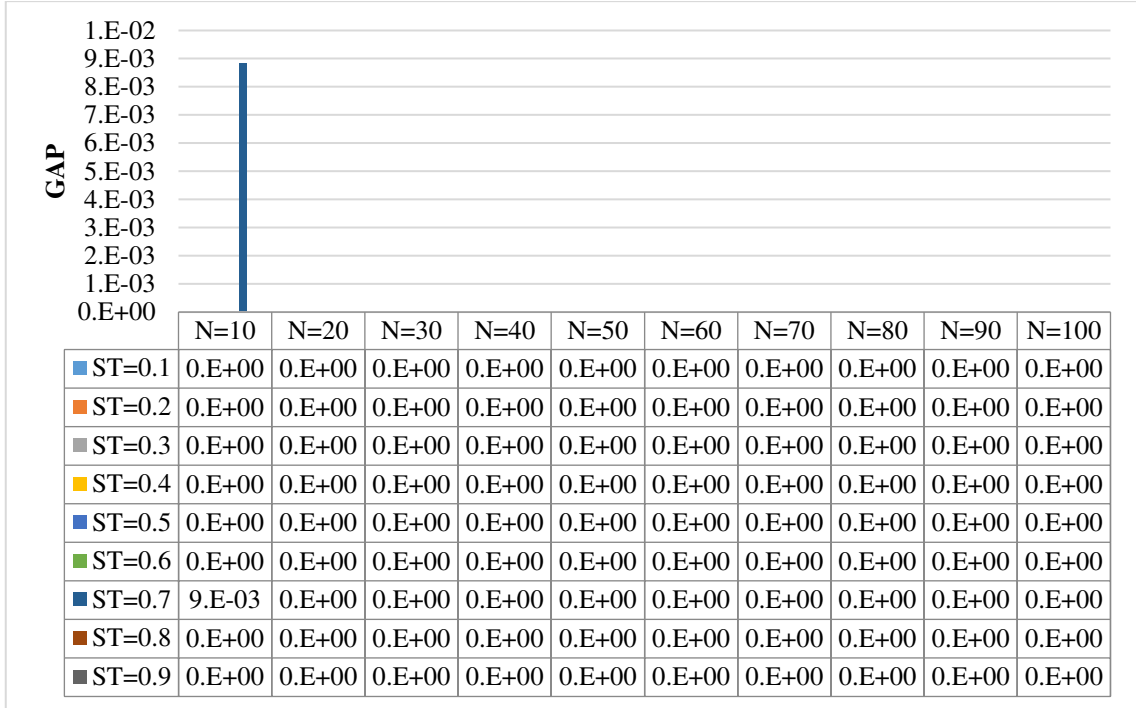
Şekil 4.19. SimTSA'nın farklı ST ve N değerleri ile Cap71 problemi için ürettiği sonuçlar



Şekil 4.20. SimTSA'nın farklı ST ve N değerleri ile Cap72 problemi için ürettiği sonuçlar

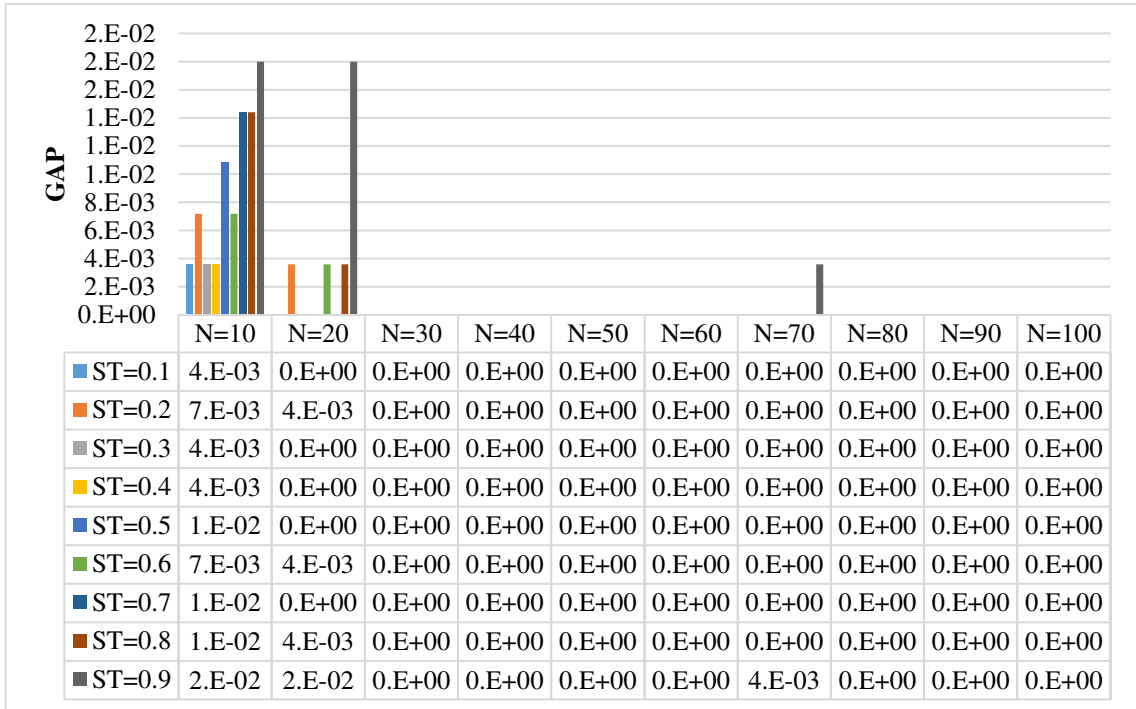


Şekil 4.21. SimTSA'nın farklı ST ve N değerleri ile Cap73 problemi için ürettiği sonuçlar

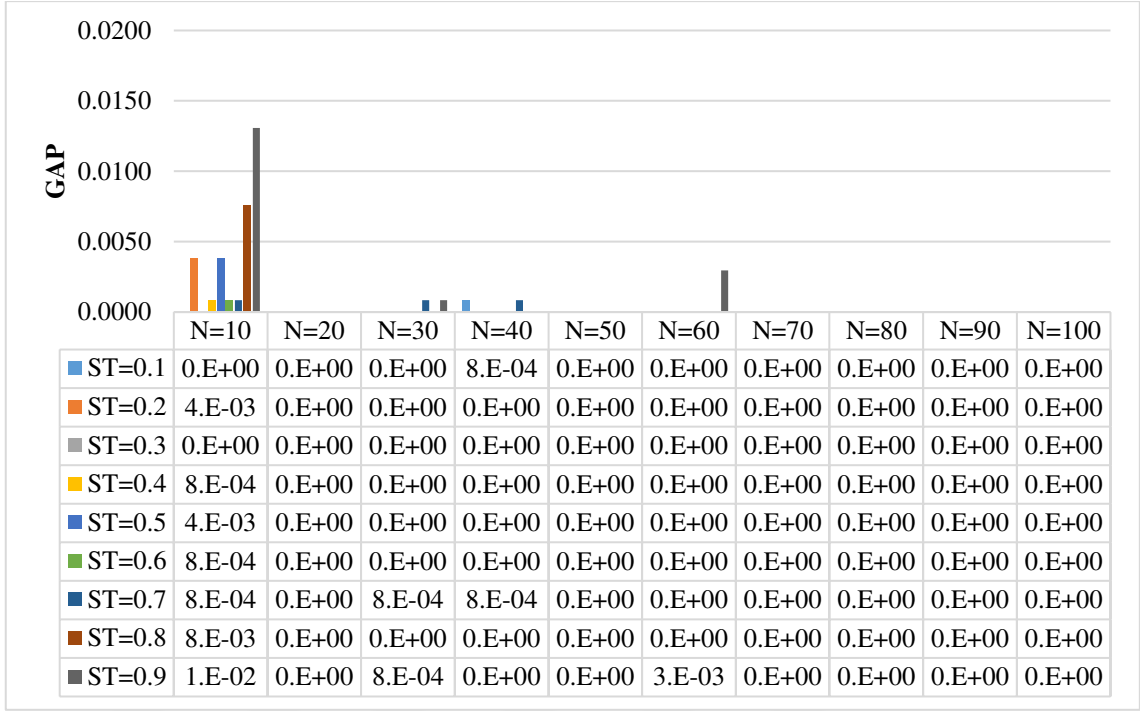


Şekil 4.22. SimTSA'nın farklı ST ve N değerleri ile Cap74 problemi için ürettiği sonuçlar

Cap102 ve Cap104 problemleri için tüm test durumlarında SimTSA tarafından optimum çözümler elde edildiğinden sonuçlar burada raporlanmamıştır. Cap101 (Şekil 4.23) ve Cap103 (Şekil 4.24) problemlerinde ise $ST \in [0.2, 0.8]$ ve $N > 20$ olduğu test durumlarında optimum sonuçlar elde edilmiştir.

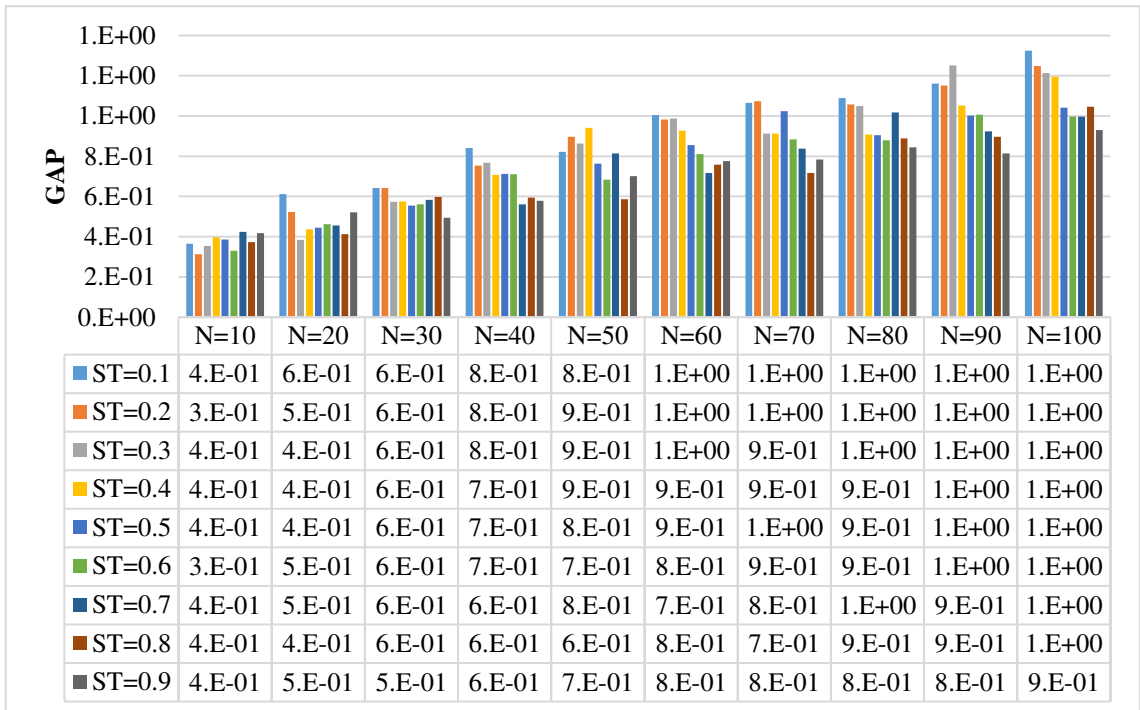


Şekil 4.23. SimTSA'nın farklı ST ve N değerleri ile Cap101 problemi için ürettiği sonuçlar

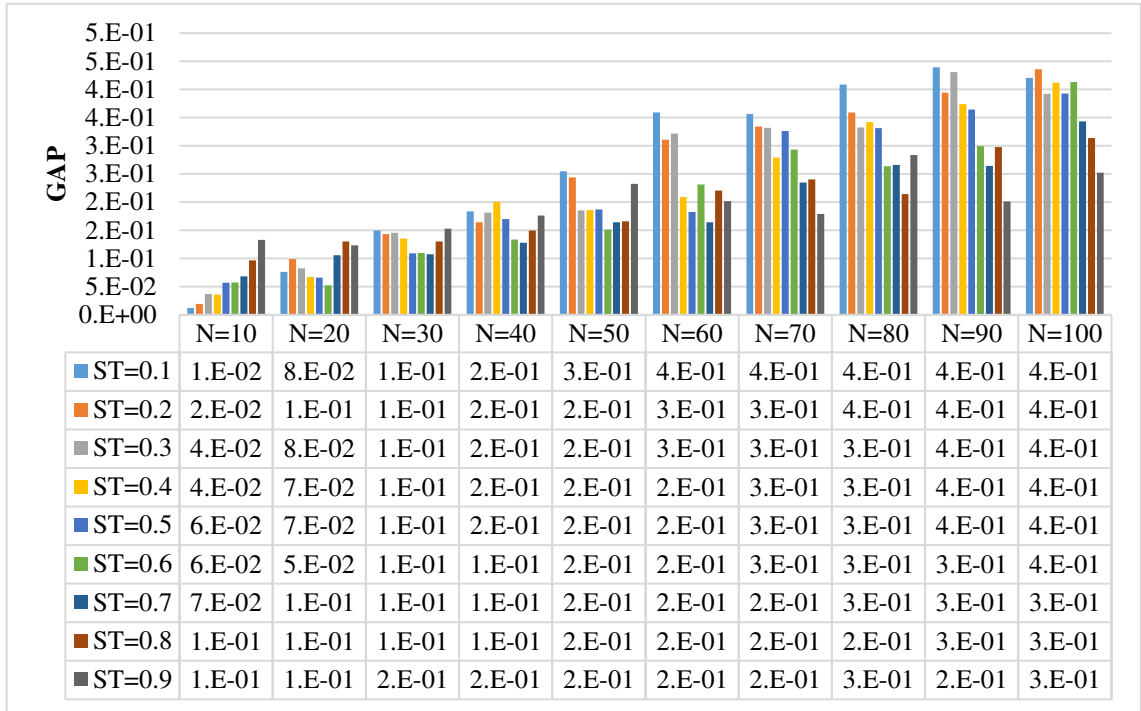


Şekil 4.24. SimTSA'nın farklı ST ve N değerleri ile Cap103 problemi için ürettiği sonuçlar

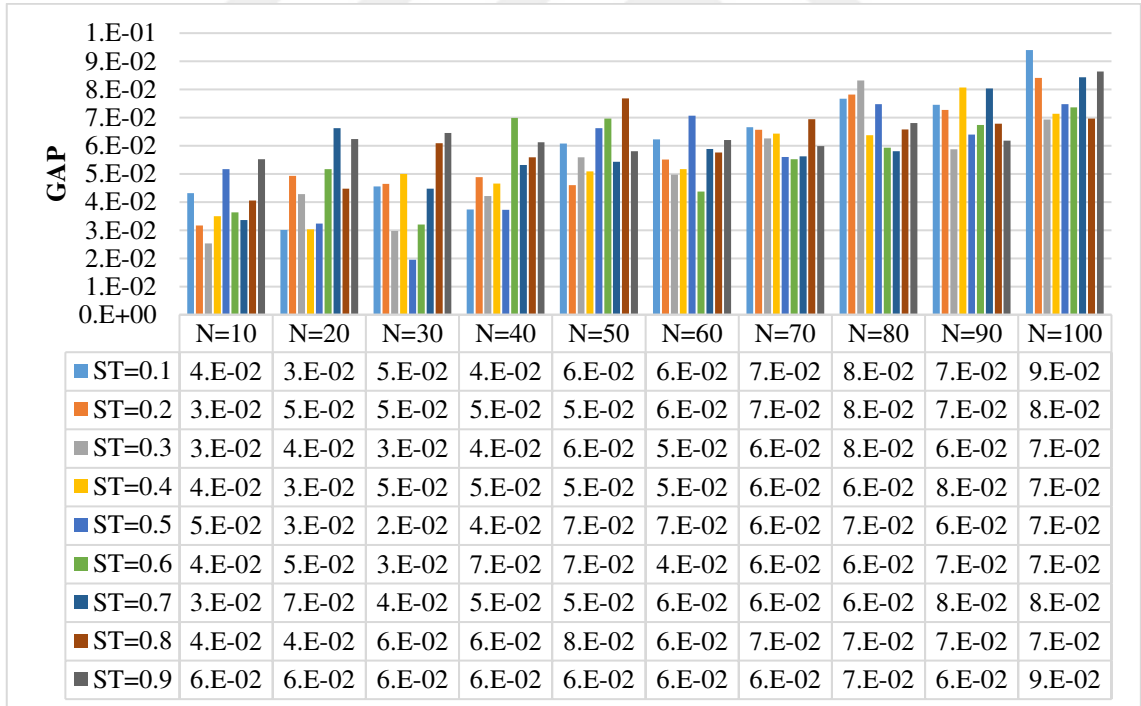
SimTSA, Cap134 problemini için her durumda optimum çözüme ulaştığından sonuçlar burada raporlanmamıştır. Cap131 (Şekil 4.25), Cap132 (Şekil 4.26) ve Cap133 (Şekil 4.27) problemlerinin sonuçları incelendiğinde optimum sonuçlara erişilemediği ancak elde edilen sonuçların kabul edilebilir düzeyde (göreceli hata ≤ 0.01) olduğu görülmektedir.



Şekil 4.25. SimTSA'nın farklı ST ve N değerleri ile Cap131 problemi için ürettiği sonuçlar

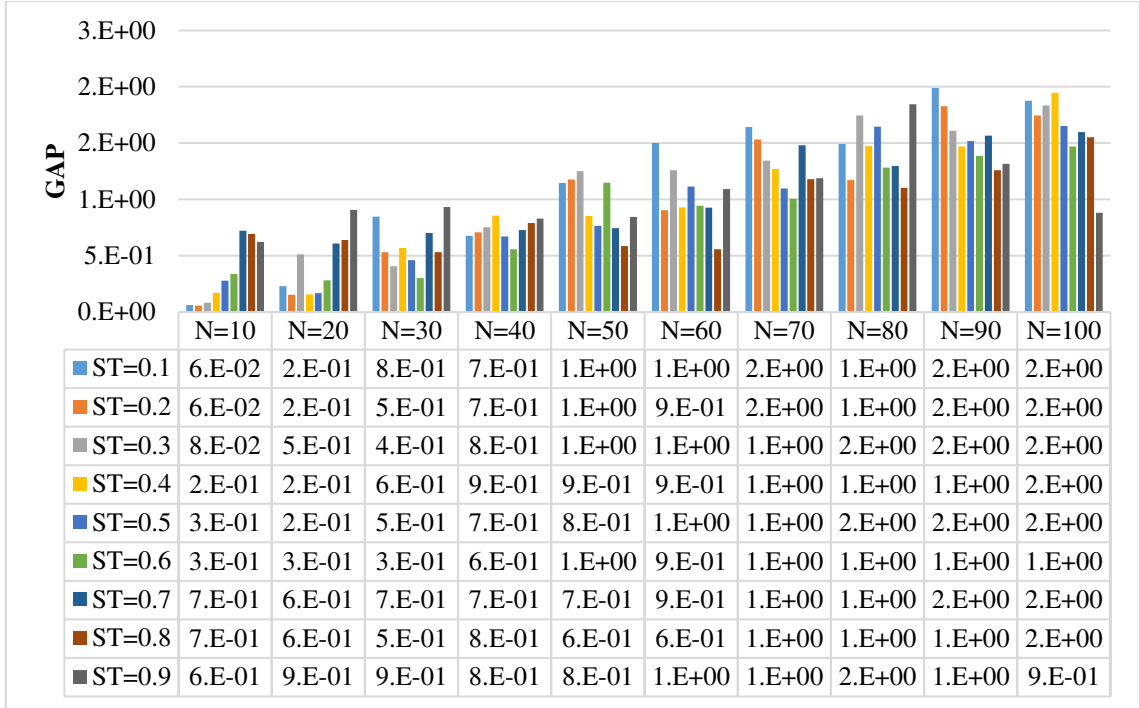


Şekil 4.26. SimTSA'nın farklı ST ve N değerleri ile Cap132 problemi için ürettiği sonuçlar

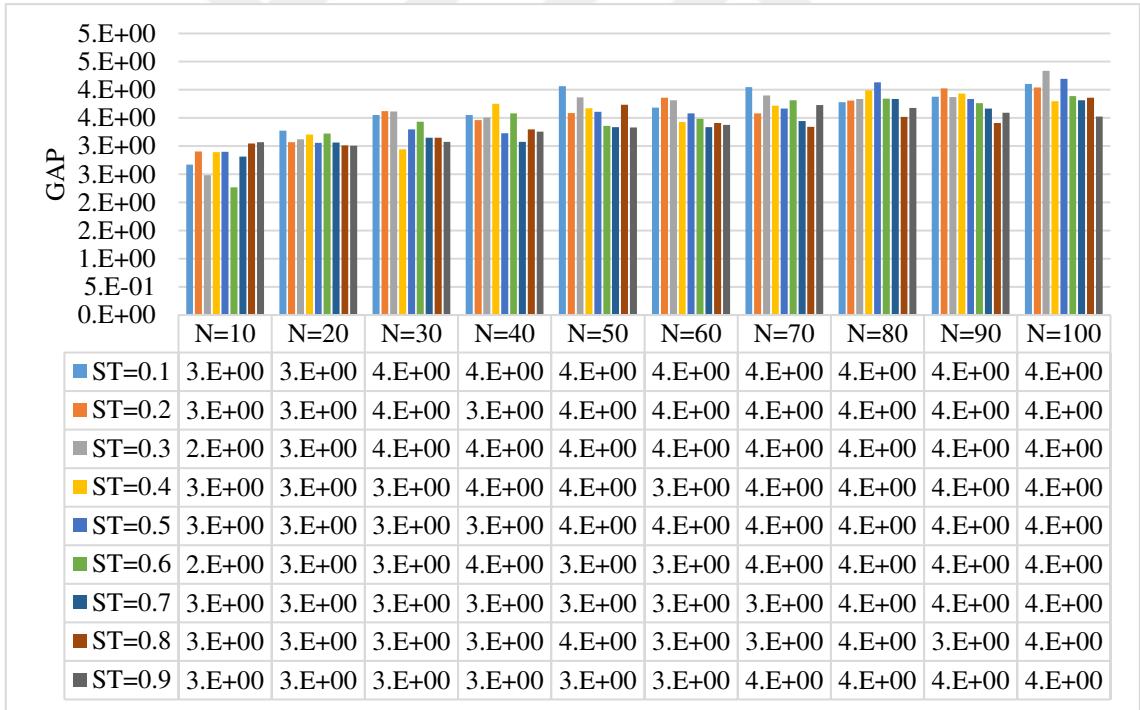


Şekil 4.27. SimTSA'nın farklı ST ve N değerleri ile Cap133 problemi için ürettiği sonuçlar

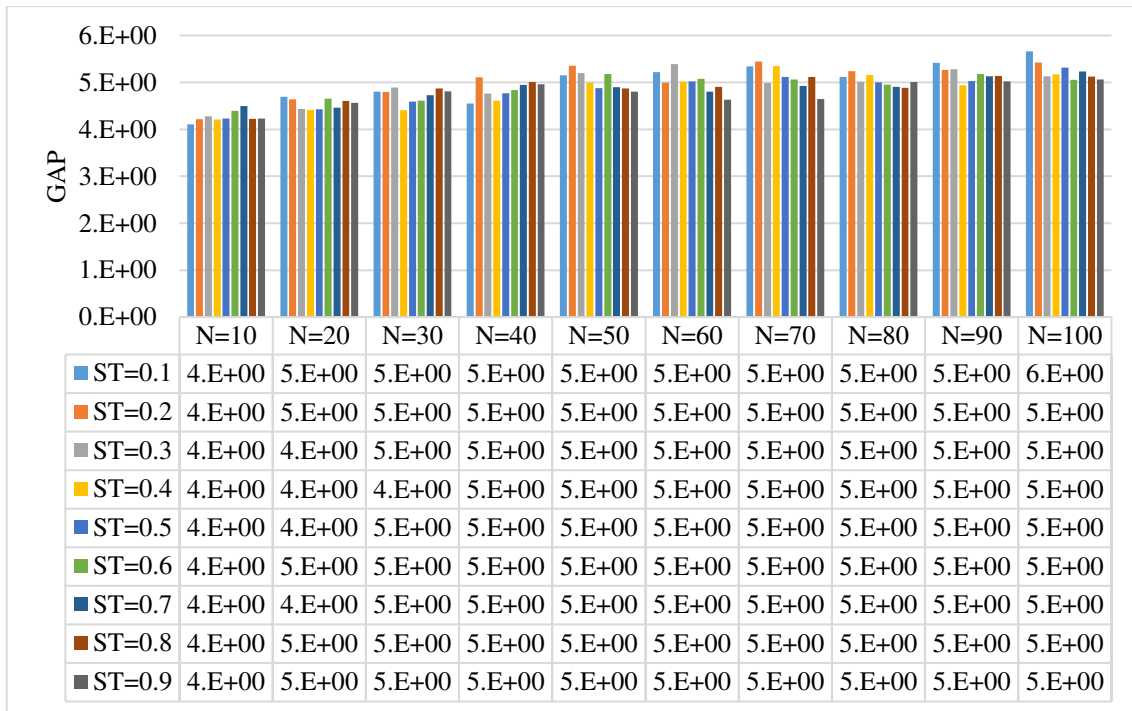
CapA (Şekil 4.28), CapB (Şekil 4.29) ve CapC (Şekil 4.30) problemlerinde de optimum çözümler elde edilememiştir. Optimizasyon probleminin boyutunun artması SimTSA'nın performansını olumsuz yönde etkilemiştir.



Şekil 4.28. SimTSA'nın farklı ST ve N değerleri ile CapA problemi için ürettiği sonuçlar

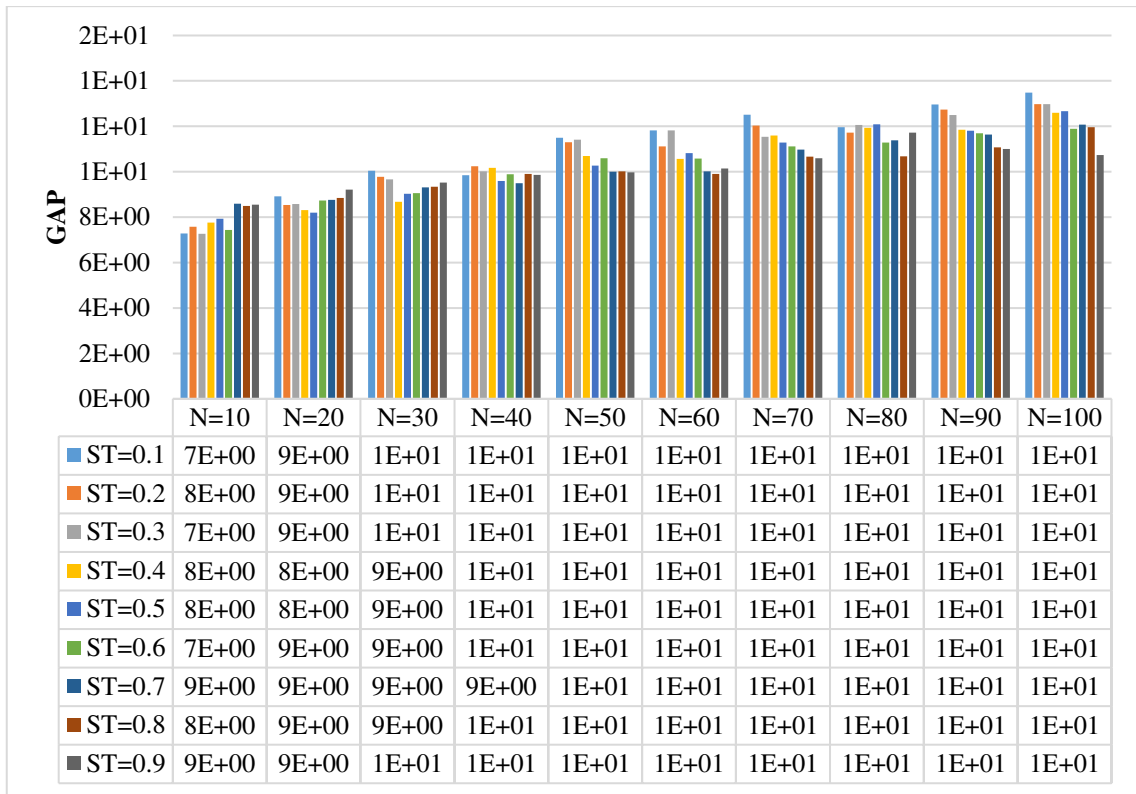


Şekil 4.29. SimTSA'nın farklı ST ve N değerleri ile CapB problemi için ürettiği sonuçlar



Şekil 4.30. SimTSA'nın farklı ST ve N değerleri ile CapC problemi için ürettiği sonuçlar

SimTSA'nın UFLP'ler üzerindeki tüm sonuçlarını analiz etmek için Şekil 4.31 ve Çizelge 4.23 hazırlanmıştır. Şekil 4.31'de SimTSA'nın farklı ST ve N değerlerindeki toplam GAP değerleri raporlanmıştır. Bu tabloya göre en iyi sonuçlar N=10 olduğu durumda elde edilmiştir. Çizelge 4.23'e bakıldığında ise 15 problemin sadece 6 tanesinde optimum çözüme ulaştığı görülmektedir. Sonuç olarak SimTSA ile $N \in [30,100]$ ve $ST \in [0.2,0.8]$ arasında iken umut verici ve kabul edilebilir (göreceli hata ≤ 0.01) çözümler üretilmiştir.



Şekil 4.31. SimTSA'nın farklı ST ve N değerlerindeki toplam GAP değerleri

Çizelge 4.23. SimTSA'da 15 UFLP problemi için toplam optimum çözüm sayısı

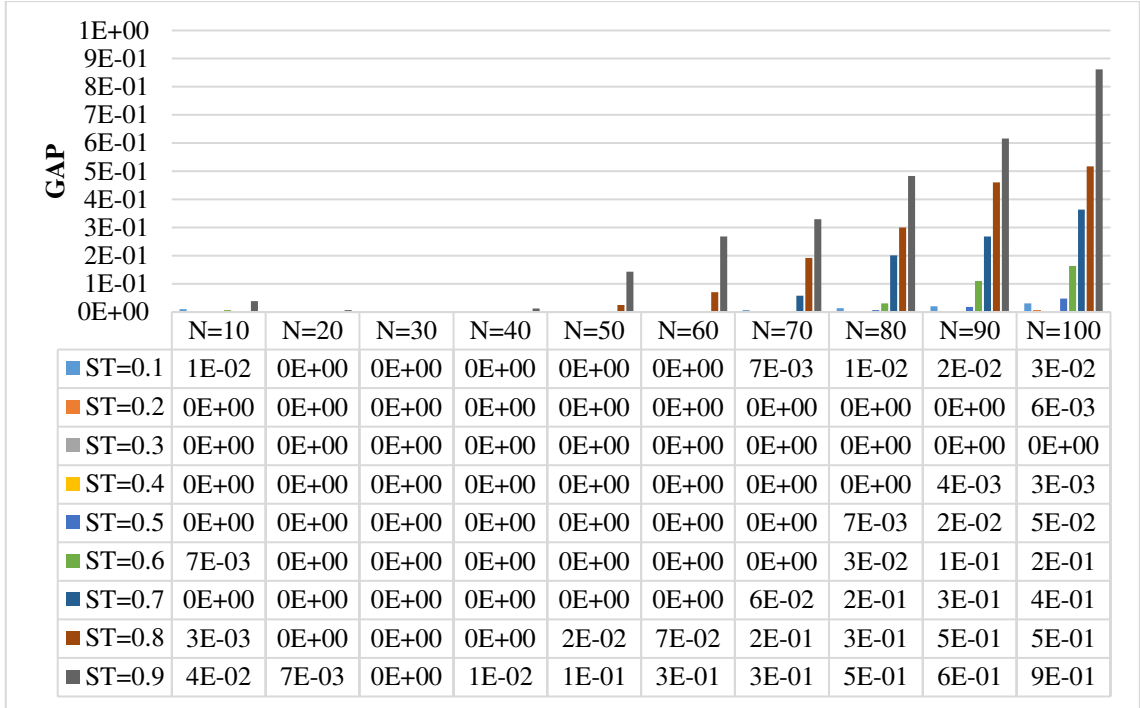
	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90	N=100	Toplam	Sıra
ST=0.1	6	9	9	8	9	9	9	9	9	9	86	3
ST=0.2	5	8	9	9	9	9	9	9	9	9	85	4
ST=0.3	7	9	9	9	9	9	9	9	9	9	88	1
ST=0.4	6	9	9	9	9	9	9	9	9	9	87	2
ST=0.5	5	8	9	9	9	9	9	9	9	9	85	4
ST=0.6	7	8	9	9	9	9	9	9	9	9	87	2
ST=0.7	4	9	8	8	9	9	9	9	9	9	83	5
ST=0.8	6	8	9	9	9	9	9	9	9	9	86	3
ST=0.9	6	8	8	9	9	8	8	9	9	9	83	5
Toplam	52	76	79	79	81	80	80	81	81	81	770	
Sıra	5	4	3	3	1	2	2	1	1	1		

4.2.3. SimLogicTSA deneysel sonuçları ve karşılaştırmaları

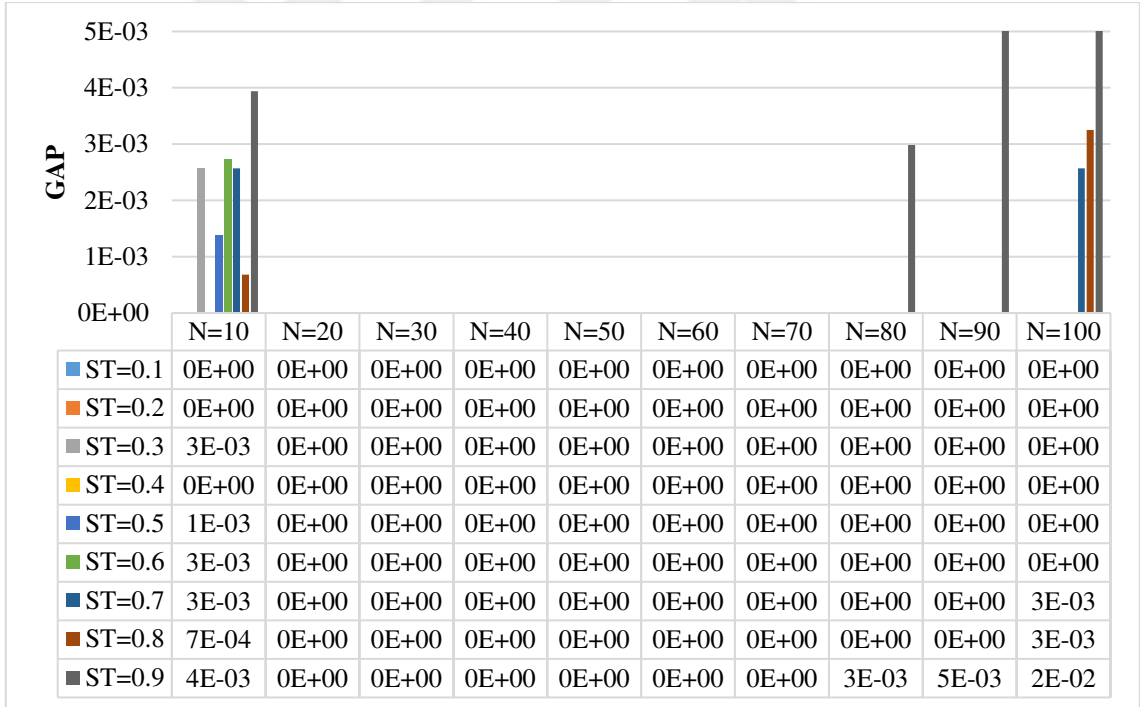
Önerilen LogicTSA ve SimTSA'nın performansını artırmak için bu metotların hibrit bir versiyonu önerilmiştir. LogicTSA ve SimTSA'ya benzer şekilde, SimLogicTSA'nın performansı N ve ST parametrelerinin farklı değerleri altında incelenmiştir.

SimLogicTSA her durum için Cap71, Cap72, Cap73 ve Cap104 problemleri üzerinde optimum çözümleri elde etmiştir. Cap74 probleminde $ST=\{0.1, 0.8, 0.9\}$ ve $N=10$ olduğunda optimum çözümler elde edilememiş diğer bütün durumlarda ise optimum durum elde edilmiştir. Cap101 probleminde $N=10$ olarak alındığında ve $ST=\{0.1, 0.2, 0.5, 0.8, 0.9\}$ olarak alındığında, Cap102 probleminde $N=10$ olarak alındığında ve $ST=\{0.1, 0.2, 0.5, 0.8, 0.9\}$ olarak alındığında ve Cap103 probleminde $N=10$ ve $ST=\{0.2,0.8\}$ olarak alındığında, algoritmanın her çalışması için optimum çözüm elde edilememiş diğer bütün durumlarda optimum çözümler elde edilmiştir.

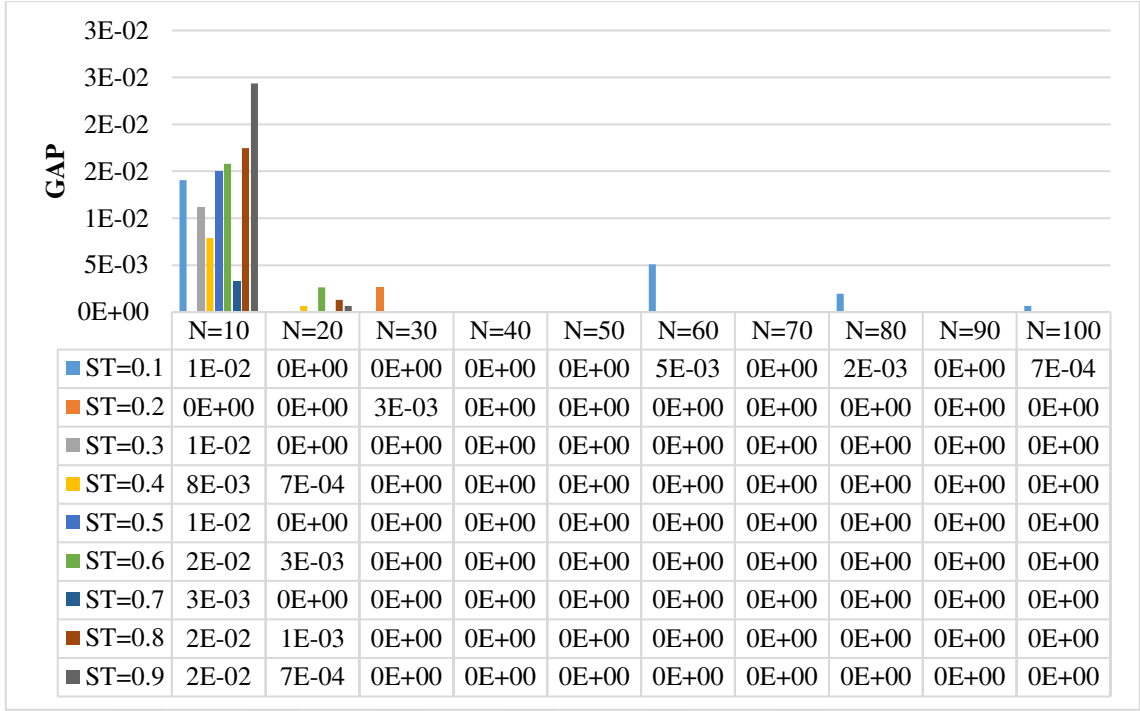
Cap 131 (Şekil 4.32) problemine ait SimLogicTSA'nın deneysel sonuçlarına göre, $N \geq 50$ ve $ST \geq 0.6$ olduğunda sonuçlar daha da kötüleşmektedir. N ve ST parametrelerinin uygun seçimi aralığı sırasıyla $N \in [10,70]$ ve $ST \in [0.2,0.5]$ şeklindedir. Cap132 (Şekil 4.33) problemi için $N \in [20,90]$ ve $ST \in [0.1,0.8]$ arasında olduğunda algoritma optimum çözüme ulaşır. Cap133 (Şekil 4.34) probleminde, algoritma parametre seçiminde Cap132 probleminden daha hassastır. $N \in [30,100]$ ve $ST \in [0.3,0.9]$ olduğunda algoritma her çalışmasında optimum çözüme ulaşmıştır.



Şekil 4.32. SimLogicTSA'nın farklı ST ve N değerleri ile Cap131 problemi için ürettiği sonuçlar

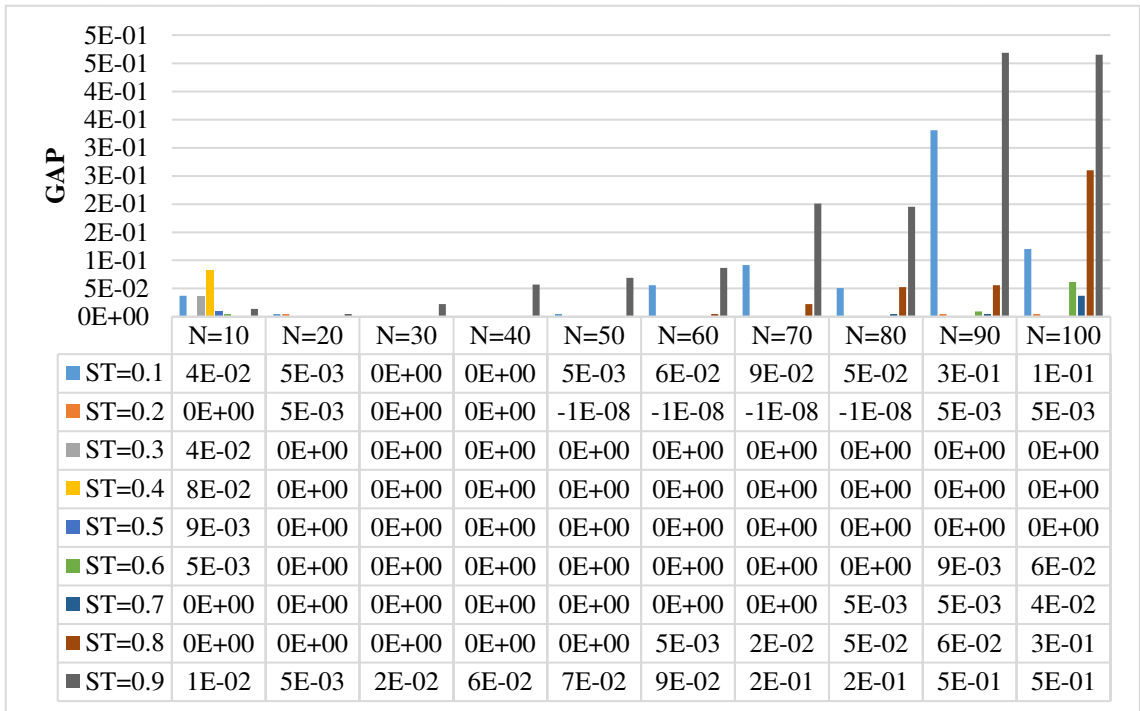


Şekil 4.33. SimLogicTSA'nın farklı ST ve N değerleri ile Cap132 problemi için ürettiği sonuçlar

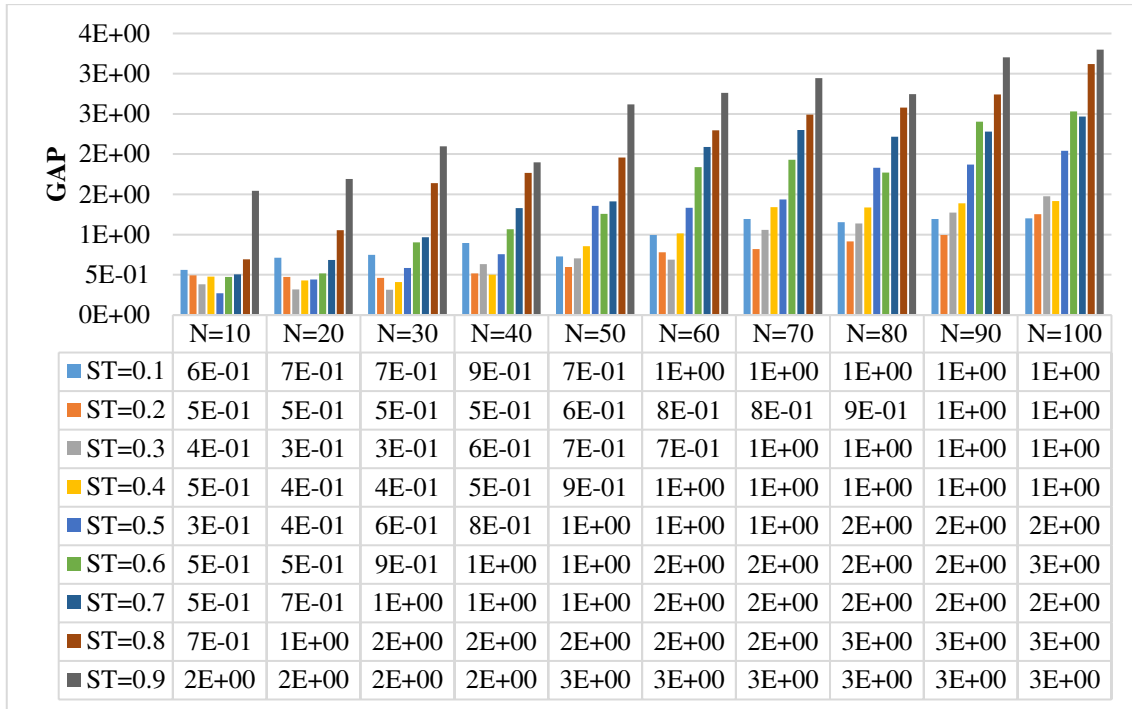


Şekil 4.34. SimLogicTSA'nın farklı ST ve N değerleri ile Cap133 problemi için ürettiği sonuçlar

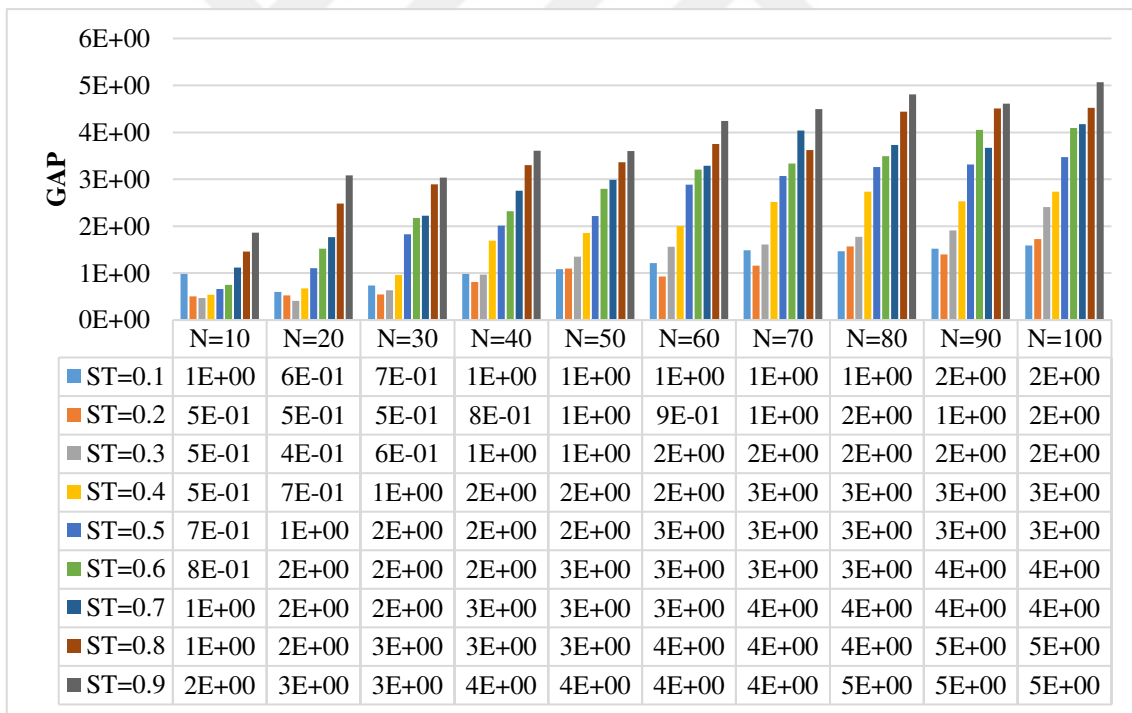
CapA (Şekil 4.35), CapB (Şekil 4.36) ve CapC (Şekil 4.37) problemlerinin sonuçları incelendiğinde düşük ST ve N değerleri seçildiğinde kabul edilebilir seviyede (göreceli hata ≤ 0.01) sonuçlar üretildiği görülmektedir.



Şekil 4.35. SimLogicTSA'nın farklı ST ve N değerleri ile CapA problemi için ürettiği sonuçlar

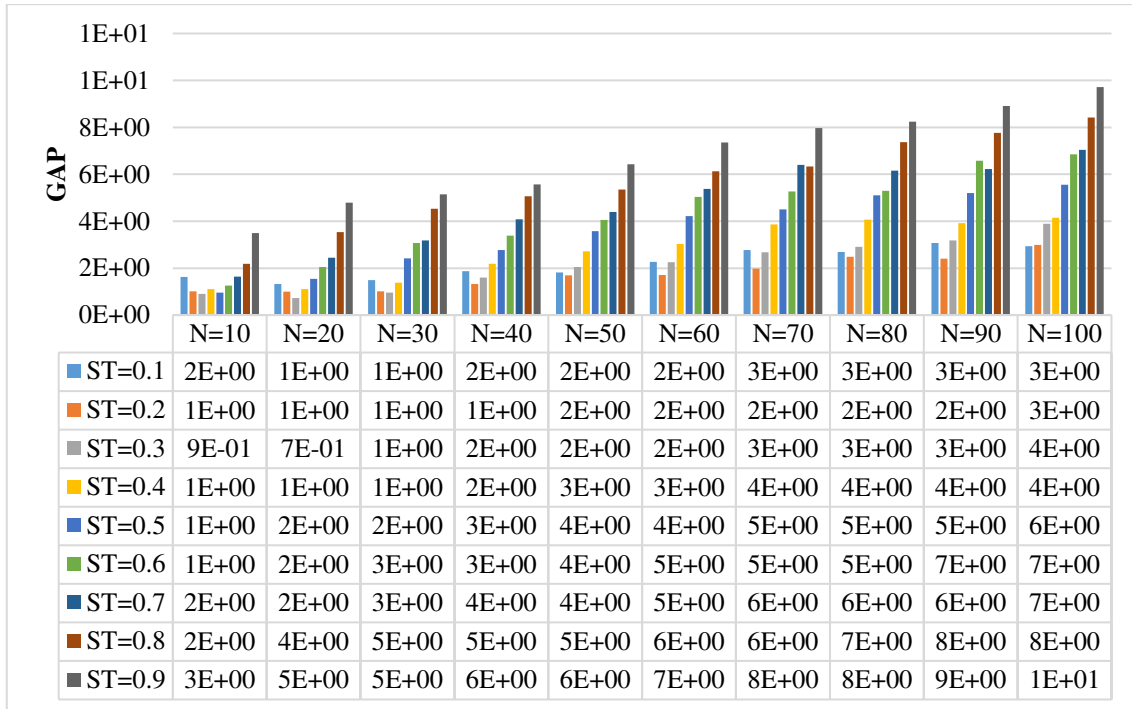


Şekil 4.36. SimLogicTSA'nın farklı ST ve N değerleri ile CapB problemi için ürettiği sonuçlar



Şekil 4.37. SimLogicTSA'nın farklı ST ve N değerleri ile CapC problemi için ürettiği sonuçlar

Tüm sonuçları kapsamlı analiz etmek için, Çizelge 4.24 ve Şekil 4.38 hazırlanmıştır. Çizelge 4.24 farklı koşullar altında algoritmanın başarılı olduğunu göstermektedir ve Şekil 4.38 ise problemlerin ortalama çözümünden elde edilen GAP değerlerini özetlemektedir.



Şekil 4.38. SimLogicTSA'nın farklı ST ve N değerlerindeki toplam GAP değerleri

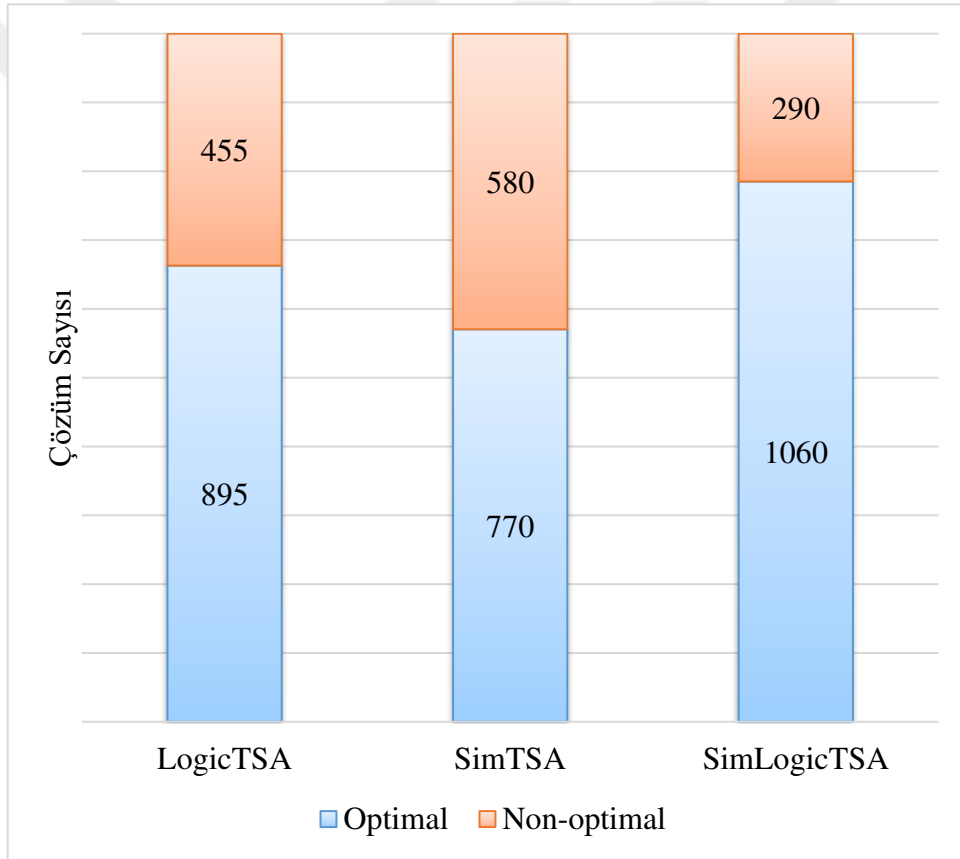
Çizelge 4.24. SimLogicTSA'da 15 UFLP problemi için toplam optimum çözüm sayısı

	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90	N=100	Toplam	Sıra
ST=0.1	8	12	13	13	12	11	11	10	11	10	111	6
ST=0.2	11	12	12	13	13	13	13	13	12	11	123	3
ST=0.3	10	13	13	13	13	13	13	13	13	13	127	1
ST=0.4	11	12	13	13	13	13	13	13	12	12	125	2
ST=0.5	9	13	13	13	13	13	13	12	12	12	123	3
ST=0.6	9	12	13	13	13	13	13	12	11	11	120	4
ST=0.7	9	13	13	13	13	13	12	11	11	10	118	5
ST=0.8	7	12	13	13	12	11	11	11	11	10	111	6
ST=0.9	6	10	12	11	11	11	11	10	10	10	102	7
Toplam	80	109	115	115	113	111	110	105	103	99	1060	
Sıra	9	5	1	1	2	3	4	6	7	8		

ST=0.3 ve N={30,40} olarak alındığında daha iyi veya eşit sonuçlar elde edilmiştir. Ancak ST=0.3 olarak alındığında algoritma popülasyon büyüklüğüne duyarlı değildir. Bu nedenle, karşılaştırmalarda ST=0.3 ve N=20 olarak seçilmiştir, çünkü problem bazlı incelemede özellikle en zor problem olan CapC'de bu koşulların daha iyi sonuç ürettiği görülmüştür.

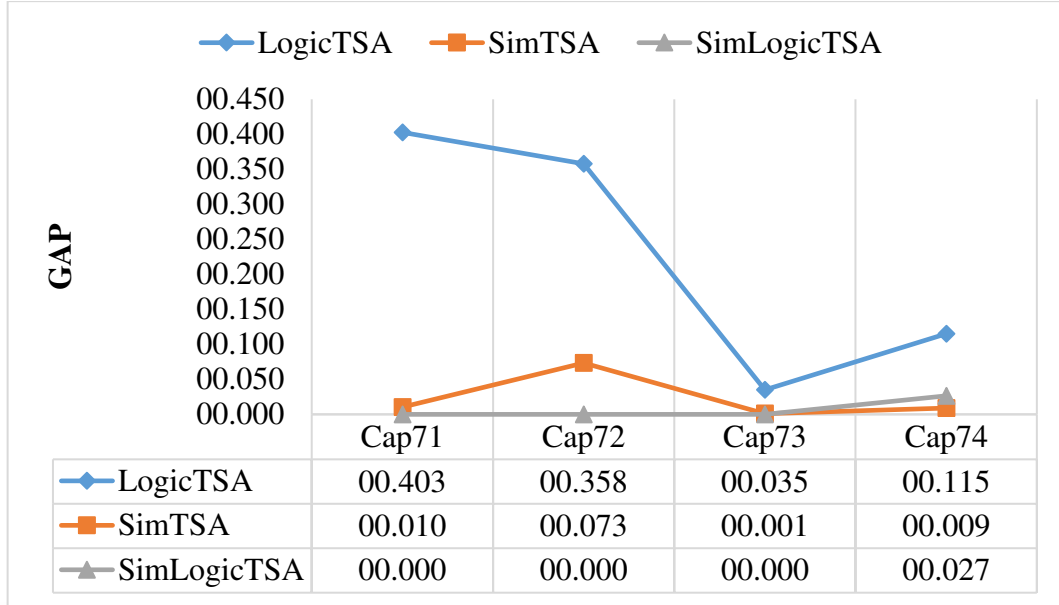
4.2.4. SimLogicTSA'nın LogicTSA ve SimTSA ile karşılaştırmaları

Düşük boyutlu problemlerde önerilen bütün algoritmalar optimum çözümleri elde ederken problemin boyutsallığı arttıkça algoritmaların performansı kötüleşmektedir. Bu bölümde SimLogicTSA'nın başarısını açıkça göstermek için LogicTSA ve SimTSA bu hibrit yaklaşımla karşılaştırılmıştır. Bu karşılaştırmada farklı bakış açıları dikkate alınmıştır. Şekil 4.39'da LogicTSA, SimTSA ve SimLogicTSA'nın optimum çözüm ürettiği ve üretmediği durumların sayısı verilmiştir. Toplam test durum sayısı 1350'dir (ST 9 farklı değer olabilir, N 10 farklı değer olabilir ve UFLP sayısı 15'tir). SimLogicTSA 1060, LogicTSA 895 ve SimTSA 770 durumda optimum çözüme ulaşmıştır. Bu sonuçlara göre yaklaşımları hibritlemek olumlu sonuçlar üretilmesini sağlamıştır.



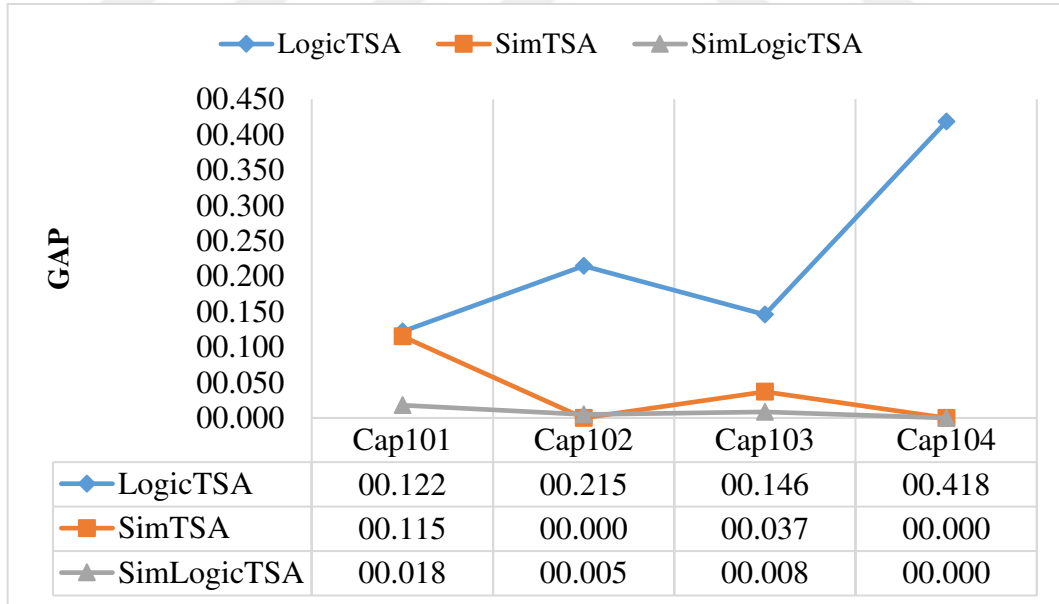
Şekil 4.39. LogicTSA, SimTSA ve SimLogicTSA'nın optimum çözüm ürettiği ve üretmediği durumların sayısı

Farklı bir bakış açısıyla sonuçları analiz etmek için Şekil 4.40-4.43 hazırlanmıştır. Şekil 4.40'da küçük boyutlu problemler için elde edilmiş sonuçlar incelendiğinde SimLogicTSA'nın SimTSA'dan, SimTSA'nın ise LogicTSA'dan daha iyi sonuçlar ürettiği görülmektedir.



Şekil 4.40. Cap71, Cap72, Cap73 ve Cap74 için optimal olmayan sonuçların toplam GAP değerleri

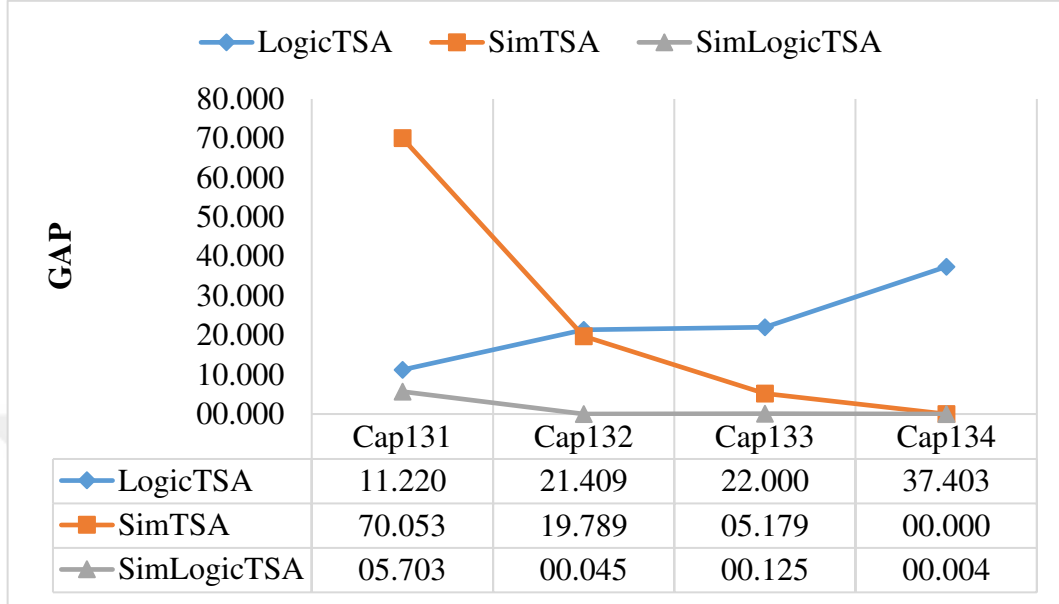
Şekil 4.41’de orta boyutlu problemler için elde edilmiş sonuçlar incelendiğinde SimLogicTSA’nın SimTSA’dan, SimTSA’nın ise LogicTSA’dan daha iyi sonuçlar ürettiği görülmektedir.



Şekil 4.41. Cap101, Cap102, Cap103 ve Cap104 için optimal olmayan sonuçların toplam GAP değerleri

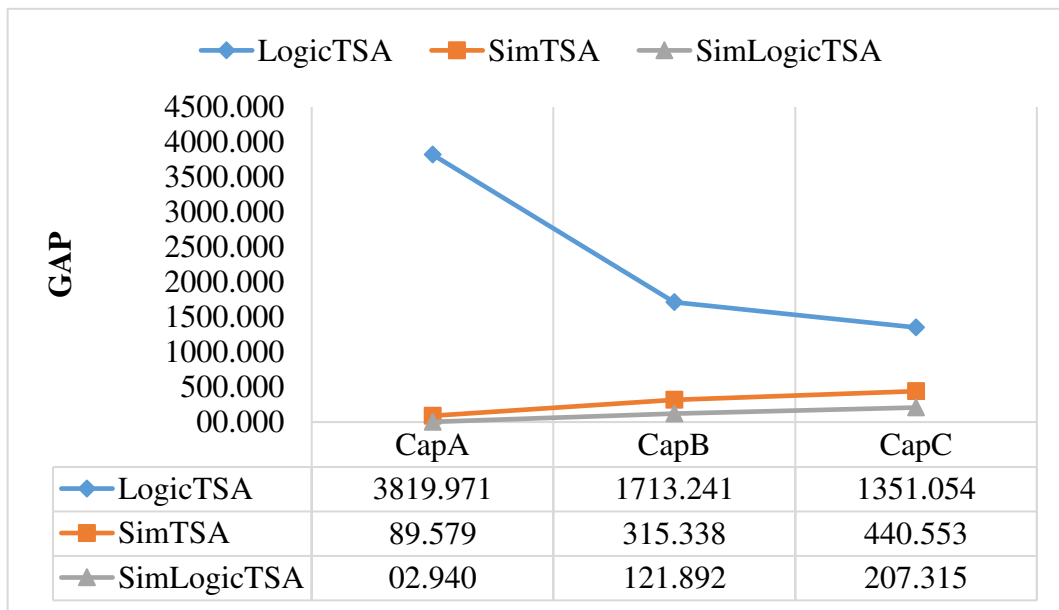
Şekil 4.42’de büyük boyutlu problemler için elde edilmiş sonuçlar incelendiğinde SimLogicTSA’nın SimTSA ve LogicTSA’dan daha iyi sonuçlar ürettiği görülmektedir. Cap131 için LogicTSA daha iyi sonuçlar ürettiği halde, Cap132, Cap133 ve Cap134 için

SimTSA daha iyi sonuçlar üretmiştir. SimTSA'nın Cap134 problemi için tüm durumlarda optimum çözüm üretmesi SimLogicTSA'nın ürettiği optimum çözümlerde etken olduğunu göstermektedir.



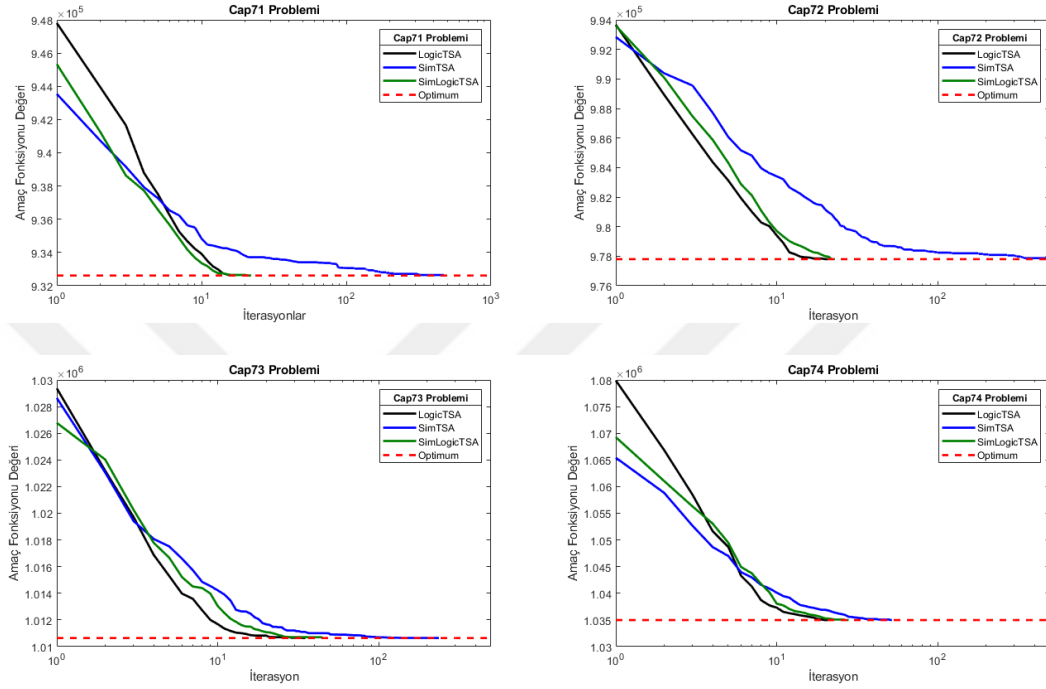
Şekil 4.42. Cap131, Cap132, Cap133 ve Cap134 için optimal olmayan sonuçların toplam GAP değerleri

Şekil 4.43'te çok büyük boyutlu problemler için elde edilmiş sonuçlar incelendiğinde, SimLogicTSA'nın SimTSA ve LogicTSA'dan daha iyi sonuçlar ürettiği gösterilmiştir. Çözümlerin iyileştirilmesine SimTSA, LogicTSA'dan daha fazla katkı sağlamıştır.



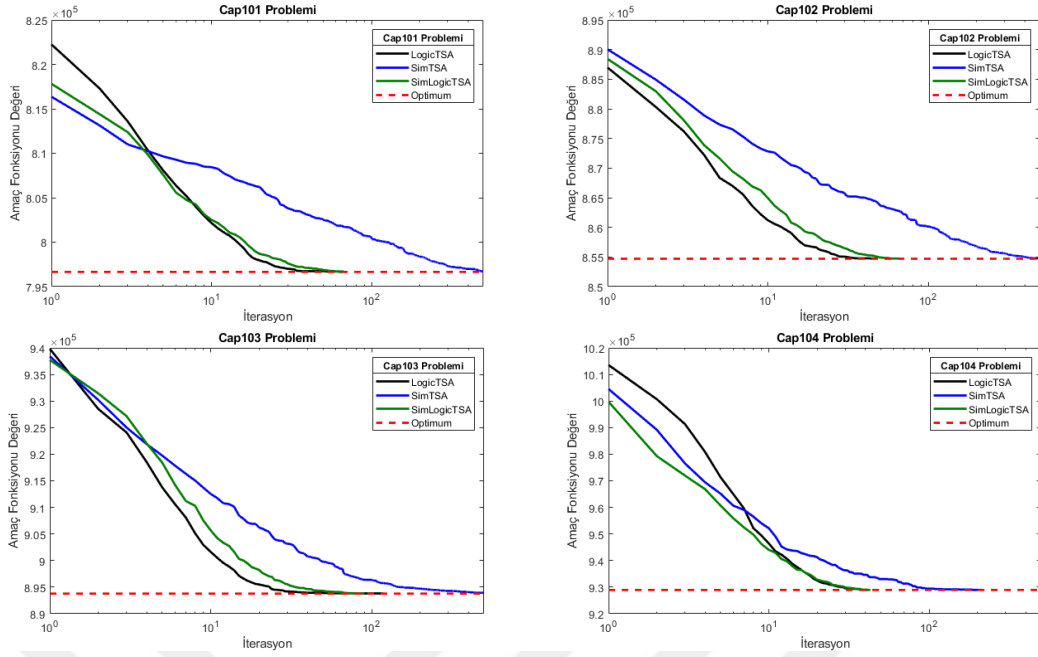
Şekil 4.43. CapA, CapB, ve CapC için optimal olmayan sonuçların toplam GAP değerleri

Karşılaştırma ve analizlerden SimLogicTSA'nın en iyi sonuç ürettiği durumun $N=20$ ve $ST=0.3$ alındığı durum olduğu tespit edilmiştir. Bu koşullar kullanılarak, SimLogicTSA, SimTSA ve LogicTSA ile yakınsama grafikleri üzerinde kıyaslanmıştır. Şekil 4.44'te yer alan küçük boyutlu problemlere ait yakınsama grafikleri incelendiğinde LogicTSA ile SimLogicTSA'nın SimTSA'dan daha hızlı yakınsadığı görülmüştür.



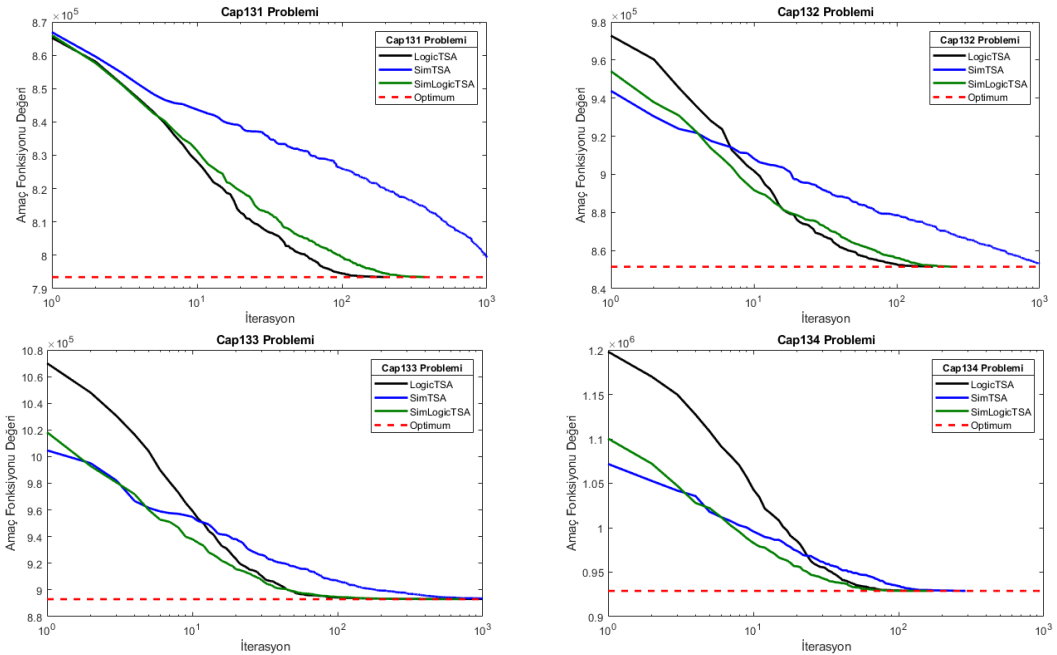
Şekil 4.44. Cap71, Cap72, Cap73, Cap74 problemleri için yakınsama grafikleri

Şekil 4.45'te yer alan orta boyutlu problemlere ait yakınsama grafikleri incelendiğinde LogicTSA ile SimLogicTSA'nın SimTSA'dan daha hızlı yakınsadığı görülmüştür.



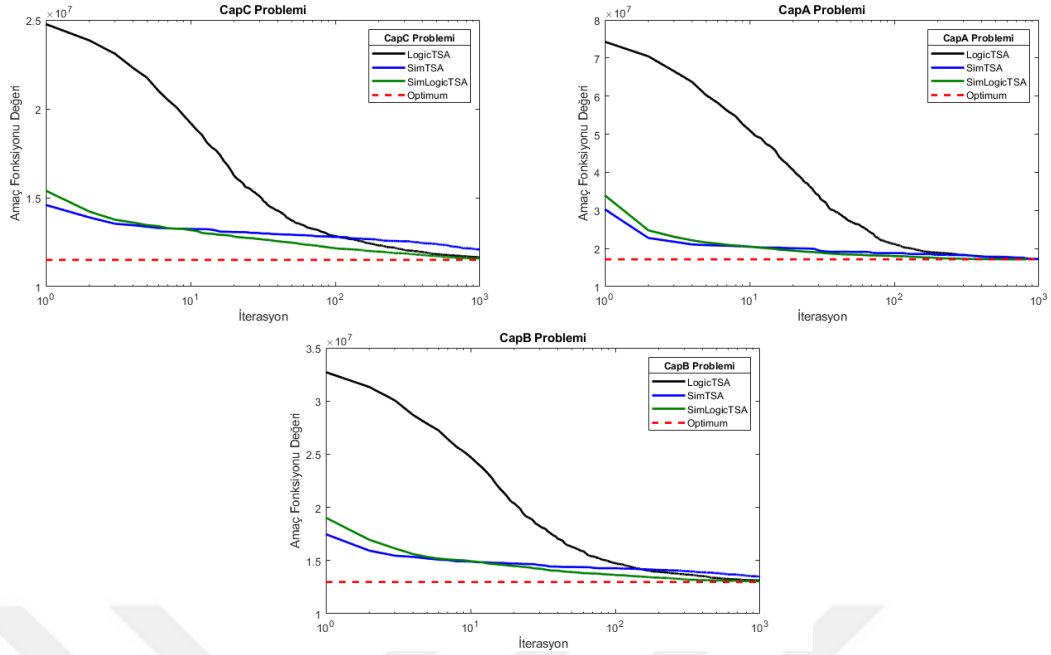
Şekil 4.45. Cap101, Cap102, Cap103, Cap104 problemleri için yakınsama grafikleri

Şekil 4.46’da yer alan büyük boyutlu problemlere ait yakınsama grafikleri incelendiğinde LogicTSA ile SimLogicTSA’nın SimTSA’dan daha hızlı yakınsadığı görülmüştür.



Şekil 4.46. Cap131, Cap132, Cap133, Cap134 problemleri için yakınsama grafikleri

Şekil 4.47’de yer alan çok büyük boyutlu problemlere ait yakınsama grafikleri incelendiğinde SimTSA ile SimLogicTSA’nın LogicTSA’dan daha hızlı yakınsadığı görülmüştür.



Şekil 4.47. CapA, CapB, CapC problemleri için yakınsama grafikleri

4.2.5. SimLogicTSA'nın diğer algoritmalarla karşılaştırılması

Adil bir karşılaştırma yapmak için sonlandırma koşulu olarak $\max\text{FES}=8\text{E}+4$ alınmıştır. Diğer algoritmaların sonuçları doğrudan (Kashan ve ark., 2013; Kiran ve Gunduz, 2013) çalışmalarından alınmıştır. SimLogicTSA için $N=20$ ve $ST=0.3$ olarak alınmıştır. Bu şartlar altında 15 UFLP üzerinde GAP değerlerine göre algoritmalar karşılaştırılmıştır. En iyi sonuçlar **kalin yazılı** olarak belirtilmiştir.

4.2.5.1. SimLogicTSA'nın BPSO ve IBPSO ile karşılaştırılması

İkili parçacık sürüsü optimizasyon (BPSO) algoritmasının sonuçları doğrudan (Kiran ve Gunduz, 2013) çalışmasından alınmıştır. Algoritmalar Çizelge 4.25'te birbirleriyle karşılaştırılmıştır. Çizelgedeki sonuçlardan da görüldüğü gibi, SimLogicTSA neredeyse tüm durumlarda BPSO'dan daha iyidir. Cap71 ve Cap72'yi çözerken algoritmalar eşit performans göstermiş diğer problemlerde ise SimLogicTSA, BPSO'ya göre daha üstün performans göstermiştir. IBPSO algoritması, mantıksal kapılar kullanılarak BPSO'nun geliştirilmiş bir versiyonudur ve (Yuan ve ark., 2009) çalışmasında IBPSO algoritmasını UC problemini çözmek için kullanmıştır. SimLogicTSA ve IBPSO'nun sonuçlarının birbirleriyle karşılaştırıldığı Çizelge 4.26'dan görüldüğü gibi, IBPSO, yüksek boyutlu UFLP'lerde kabul edilebilir bir performans

seviyesine sahip değildir. IBPSO'nun sonuçları doğrudan (Kiran ve Gunduz, 2013) çalışmasında alınmıştır.



Çizelge 4.25. SimLogicTSA ile BPSO'nun karşılaştırılması

Problem	BPSO					SimLogicTSA				
	En İyi	En Kötü	Ortalama	GAP	Std. Sap.	En İyi	En Kötü	Ortalama	GAP	Std. Sap.
Cap71	932,615.75	932,615.75	932,615.75	0.0000	0.00	932,615.7500	932,615.7500	932,615.7500	0.0000	0.0000
Cap72	977,799.4	977,799.4	977,799.4	0.0000	0.00	977,799.4000	977,799.4000	977,799.4000	0.0000	0.0000
Cap73	1,010,886.187	1,012,476.975	1,010,886.187	0.0242	634.62	1,010,641.4500	1,010,641.4500	1,010,641.4500	0.0000	0.0000
Cap74	1,035,068.312	1,037,717.075	1,035,068.312	0.0088	500.27	1,034,976.9750	1,034,976.9750	1,034,976.9750	0.0000	0.0000
Cap101	797,016.6558	799,092.1125	797,016.6558	0.0462	566.44	796,648.4375	796,648.4375	796,648.4375	0.0000	0.0000
Cap102	854,830.955	855,971.75	854,830.955	0.0148	386.76	854,704.2000	854,704.2000	854,704.2000	0.0000	0.0000
Cap103	894,159.4667	895,027.1875	894,159.4667	0.0422	485.26	893,782.1125	893,782.1125	893,782.1125	0.0000	0.0000
Cap104	929,694.4467	934,586.975	929,694.4467	0.0810	1951.81	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
Cap131	794,484.4496	797,735.5375	794,484.4496	0.1317	1207.63	793,439.5625	793,439.5625	793,439.5625	0.0000	0.0000
Cap132	852,273.2071	855,328.675	852,273.2071	0.0914	1196.19	851,495.3250	851,495.3250	851,495.3250	0.0000	0.0000
Cap133	894,072.9054	896,661.5625	894,072.9054	0.1115	821.28	893,076.7125	893,076.7125	893,076.7125	0.0000	0.0000
Cap134	930,192.0425	934,586.975	930,192.0425	0.1346	2285.42	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
CapA	17,530,210.57	18,682,895.54	17,530,210.57	2.1785	374,302.81	17,156,454.4783	17,156,454.4783	17,156,454.4783	0.0000	0.0000
CapB	13,232,039.15	13,633,079.9	13,232,039.15	1.9490	176,206.07	12,979,071.5814	13,146,170.3996	13,020,296.0958	0.3176	47,993.6815
CapC	11,676,684.07	11,871,643.26	11,676,684.07	1.4870	92,977.85	11,505,594.3288	11,690,197.0169	11,553,002.6668	0.4120	48,379.5319

Çizelge 4.26. SimLogicTSA ile IBPSO'nun karşılaştırılması

Problem	IBPSO					SimLogicTSA				
	En İyi	En Kötü	Ortalama	GAP	Std. Sap.	En İyi	En Kötü	Ortalama	GAP	Std. Sap.
Cap71	932,615.75	934,199.1375	932,964.7188	0.0374	587.49	932,615.7500	932,615.7500	932,615.7500	0.0000	0.0000
Cap72	977,799.4	983,122.2375	980,486.9238	0.2749	1844.64	977,799.4000	977,799.4000	977,799.4000	0.0000	0.0000
Cap73	1,010,641.45	1,014,917.6	1,012,642.103	0.1980	1513.78	1,010,641.4500	1,010,641.4500	1,010,641.4500	0.0000	0.0000
Cap74	1,034,976.975	1,045,383.788	1,039,149.436	0.4031	4426.67	1,034,976.9750	1,034,976.9750	1,034,976.9750	0.0000	0.0000
Cap101	796,648.4375	809,077.4875	801,403.0875	0.5968	3799.52	796,648.4375	796,648.4375	796,648.4375	0.0000	0.0000
Cap102	856,660.0125	865,438.2	860,957.6625	0.7317	3249.38	854,704.2000	854,704.2000	854,704.2000	0.0000	0.0000
Cap103	894,008.1375	909,765.25	899,511.3063	0.6410	4978.98	893,782.1125	893,782.1125	893,782.1125	0.0000	0.0000
Cap104	928,941.75	964,540.85	938,197.2625	0.9964	10,845.26	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
Cap131	806,761.2875	821,236.3625	812,669.325	2.4236	4244.29	793,439.5625	793,439.5625	793,439.5625	0.0000	0.0000
Cap132	865,407.5125	901,297.475	882,160.8613	3.6014	11,569.02	851,495.3250	851,495.3250	851,495.3250	0.0000	0.0000
Cap133	918,298.525	966,072.7125	940,076.1188	5.2626	14,905.27	893,076.7125	893,076.7125	893,076.7125	0.0000	0.0000
Cap134	973,744.7875	1,027,632.238	999,855.6225	7.6338	15,788.86	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
CapA	35,704,093	45,511,134.48	40,812,839.67	137.8862	3,357,138.19	17,156,454.4783	17,156,454.4783	17,156,454.4783	0.0000	0.0000
CapB	17,971,425.35	22,154,126.28	20,152,622.54	55.2701	1,406,575.7	12,979,071.5814	13,146,170.3996	13,020,296.0958	0.3176	47,993.6815
CapC	14,564,601.36	18,555,781.02	16,747,099.16	45.5561	1,245,252.2	11,505,594.3288	11,690,197.0169	11,553,002.6668	0.4120	48,379.5319

4.2.5.2. SimLogicTSA'nın binABC ve DisABC ile karşılaştırılması

SimLogicTSA bu bölümde ABC algoritmasının ikili varyantları ile karşılaştırılmıştır. Bu bölümdeki binABC ve DisABC algoritmalarına ait sonuçlar doğrudan (Kiran ve Gunduz, 2013) çalışmasından alınmıştır. Çizelge 4.27'deki sonuçlar incelendiğinde küçük, orta ve büyük boyutlu problemlerde, Cap133 hariç, SimLogicTSA ile binABC çözüm kalitesi ve sağlamlık açısından benzer sonuçlar üretmiştir. Cap133, CapA, CapB ve CapC problemlerinde ise SimLogicTSA, binABC algoritmasından daha iyi performans göstermiştir. Çizelge 4.28'e göre, DisABC ve SimLogicTSA küçük, orta ve Cap134 problemlerini çözmede aynı performansa sahiptir. Diğer problemlerde ise SimLogicTSA, DisABC'den üstün olduğunu göstermiştir.



Çizelge 4.27. SimLogicTSA ile binABC'nin karşılaştırılması

Problem	binABC					SimLogicTSA				
	En İyi	En Kötü	Ortalama	GAP	Std. Sap.	En İyi	En Kötü	Ortalama	GAP	Std. Sap.
Cap71	932,615.75	932,615.75	932,615.75	0.0000	0.00	932,615.7500	932,615.7500	932,615.7500	0.0000	0.0000
Cap72	977,799.4	977,799.4	977,799.4	0.0000	0.00	977,799.4000	977,799.4000	977,799.4000	0.0000	0.0000
Cap73	1,010,641.45	1,010,641.45	1,010,641.45	0.0000	0.00	1,010,641.4500	1,010,641.4500	1,010,641.4500	0.0000	0.0000
Cap74	1,034,976.975	1,034,976.975	1,034,976.975	0.0000	0.00	1,034,976.9750	1,034,976.9750	1,034,976.9750	0.0000	0.0000
Cap101	796,648.4375	796,648.4375	796,648.4375	0.0000	0.00	796,648.4375	796,648.4375	796,648.4375	0.0000	0.0000
Cap102	854,704.2	854,704.2	854,704.2	0.0000	0.00	854,704.2000	854,704.2000	854,704.2000	0.0000	0.0000
Cap103	893,782.1125	893,782.1125	893,782.1125	0.0000	0.00	893,782.1125	893,782.1125	893,782.1125	0.0000	0.0000
Cap104	928,941.75	928,941.75	928,941.75	0.0000	0.00	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
Cap131	793,439.5625	793,439.5625	793,439.5625	0.0000	0.00	793,439.5625	793,439.5625	793,439.5625	0.0000	0.0000
Cap132	851,495.325	851,495.325	851,495.325	0.0000	0.00	851,495.3250	851,495.3250	851,495.3250	0.0000	0.0000
Cap133	894,095.7625	894,752.025	894,161.3888	0.1215	200.24	893,076.7125	893,076.7125	893,076.7125	0.0000	0.0000
Cap134	928,941.75	928,941.75	928,941.75	0.0000	0.00	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
CapA	17,180,539.56	18,030,263.31	17,664,663.43	2.9622	236,833.5	17,156,454.4783	17,156,454.4783	17,156,454.4783	0.0000	0.0000
CapB	13,100,041.02	13,476,652.7	13,304,594.27	2.5081	91,430.13	12,979,071.5814	13,146,170.3996	13,020,296.0958	0.3176	47,993.6815
CapC	11,535,255.5102	11,867,887.0012	11,802,532.8641	2.5800	82,312.70	11,505,594.3288	11,690,197.0169	11,553,002.6668	0.4120	48,379.5319

Çizelge 4.28. SimLogicTSA ile disABC'nin karşılaştırılması

Problem	disABC					SimLogicTSA				
	En İyi	En Kötü	Ortalama	GAP	Std. Sap.	En İyi	En Kötü	Ortalama	GAP	Std. Sap.
Cap71	932,615.75	932,615.7500	932,615.75	0.0000	0.00	932,615.7500	932,615.7500	932,615.7500	0.0000	0.0000
Cap72	977,799.4	977,799.4	977,799.4	0.0000	0.00	977,799.4000	977,799.4000	977,799.4000	0.0000	0.0000
Cap73	1,010,641.45	1,010,641.4500	1,010,641.45	0.0000	0.00	1,010,641.4500	1,010,641.4500	1,010,641.4500	0.0000	0.0000
Cap74	1,034,976.975	1,034,976.9750	1,034,976.975	0.0000	0.00	1,034,976.9750	1,034,976.9750	1,034,976.9750	0.0000	0.0000
Cap101	796,648.4375	796,648.4375	796,648.4375	0.0000	0.00	796,648.4375	796,648.4375	796,648.4375	0.0000	0.0000
Cap102	854,704.2	854,704.2000	854,704.2	0.0000	0.00	854,704.2000	854,704.2000	854,704.2000	0.0000	0.0000
Cap103	893,782.1125	893,782.1125	893,782.1125	0.0000	0.00	893,782.1125	893,782.1125	893,782.1125	0.0000	0.0000
Cap104	928,941.75	928,941.75	928,941.75	0.0000	0.00	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
Cap131	794,299.85	802,709.225	798,355.4917	0.6196	2337.64	793,439.5625	793,439.5625	793,439.5625	0.0000	0.0000
Cap132	851,495.325	854,704.2	852,300.2575	0.0945	813.37	851,495.3250	851,495.3250	851,495.3250	0.0000	0.0000
Cap133	893,076.7125	894,095.7625	893,352.4167	0.0309	359.03	893,076.7125	893,076.7125	893,076.7125	0.0000	0.0000
Cap134	928,941.75	928,941.75	928,941.75	0.0000	0.00	928,941.7500	928,941.7500	928,941.7500	0.0000	0.0000
CapA	17,156,454.48	17,420,032.38	17,182,558.16	0.1522	74782.61	17,156,454.4783	17,156,454.4783	17,156,454.4783	0.0000	0.0000
CapB	13,205,522.04	13,683,628.78	13,407,728.05	3.3027	109,738.5	12,979,071.5814	13,146,170.3996	13,020,296.0958	0.3176	47,993.6815
CapC	11,834,640.02	12,203,264.48	12,045,991.08	4.6968	95,778.78	11,505,594.3288	11,690,197.0169	11,553,002.6668	0.4120	48,379.5319

4.2.5.3. SimLogicTSA'nın binDE ve DisDE ile karşılaştırılması

Son deneysel karşılaştırmada, SimLogicTSA, DE algoritmasının ikili varyantları ile karşılaştırılmıştır. İkili DE varyantlarına ait sonuçlar doğrudan (Kashan ve ark., 2013) çalışmasından alınmıştır. SimTSA ile DisDE ikili optimizasyon problemlerini çözmek için benzer mekanizmaları kullanmaktadır. Literatür taramasında binDE'nin çalışma prensipleri de özetlenmiştir. Kıyaslamalar GAP değerlerine ve algoritmaların her çalışmasında problemin optimum çözümüne ulaşip ulaşmadığının sayısına göre yapılmış ve Çizelge 4.29'da sunulmuştur.

Çizelge 4.29. SimLogicTSA ile binDE ve DisDE'nin karşılaştırılması

Problem	DisDE/rand		binDE		SimLogicTSA	
	Gap	Hit	Gap	Hit	Gap	Hit
Cap71	0.0000	30	0.0000	30	0.0000	30
Cap72	0.0000	30	0.0000	30	0.0000	30
Cap73	0.0000	30	0.0000	30	0.0000	30
Cap74	0.0000	30	0.0000	30	0.0000	30
Cap101	0.0036	29	0.0000	30	0.0000	30
Cap102	0.0049	29	0.0000	30	0.0000	30
Cap103	0.0055	27	0.0000	30	0.0000	30
Cap104	0.0000	30	0.0000	30	0.0000	30
Cap131	0.0036	29	0.0036	29	0.0000	30
Cap132	0.0000	30	0.0050	29	0.0000	30
Cap133	0.0138	25	0.0138	24	0.0000	30
Cap134	0.0000	30	0.0000	30	0.0000	30
CapA	0.0370	29	1.3000	8	0.0000	30
CapB	0.1890	18	1.5200	0	0.3176	0
CapC	0.0909	8	1.5500	0	0.4120	0

Çizelge 4.29'daki sonuçlar incelendiğinde DisDE'nin sonuçları orta, büyük ve çok büyük boyutlu problemlerde tutarlı değildir ve binDE'nin sonuçları da büyük ve çok büyük boyutlu problemler için tutarlı değildir. SimLogicTSA ise küçük, orta, büyük boyutlu problemlerde optimum sonuçları elde etmiştir. Çok büyük problemler olan CapB ve CapC problemlerinde DisDE/rand, özellikle CapC'de, diğer algoritmalarından daha iyi performans göstermiştir. Diğer problemlerde ise ya SimLogicTSA diğer algoritmalarından daha iyidir ya da aynı sonuçları elde etmiştir. Bu sonuçlar SimLogicTSA'nın DE algoritmalarının ikili varyantlarıyla rekabet edebildiğini göstermektedir.

4.3. DTSA deneysel sonuçları ve karşılaştırmaları

DTSA'nın performansını analiz etmek için kullanılan bütün problemler TSPLIB'de (Reinelt, 1991) bulunabilir. TSPLIB'deki problemlerin çoğu çözülmüş ve optimum değerleri sunulmuştur. Deneysel çalışmalardaki problem isimlerindeki sayılar problemdeki toplam şehir sayısını diğer bir anlamda da problemin boyutunu göstermektedir. Kullanılan problemler ve optimum değerleri Oliver30 (423.74), Eil51 (428.87), Berlin52 (7542 / 7544.37), St70 (677.11), Pr76 (108159.44), Kroa100 (21282 / 21285.44), Krob100 (22141), Kroc100 (21282 / 21285.44), Krod100 (21294), Kroe100 (22068), Eil101 (642.31), Krob150 (26130), Tsp225 (3859) ve A280 (2586.77) şeklindedir. Karşılaştırmalar bu optimum değerlerle yapılmıştır. Bu problemlerin koordinat tipleri Öklid'dir (EUC). Literatürde, bazı çalışmalarda (Chunhua ve ark., 2012; Hatamlou, 2017) Öklid olarak hesaplanan farklı tip problemler (örneğin, coğrafi-GEO) tespit edilmiştir. Bu durum karşılaştırmalarda hataya neden olmaktadır. Bu gibi durumlar alt bölümlerde açıklanmış ve belirtilmiştir. Performans kıyaslamaları Denklem 4.1'deki GAP hesaplamasına dayanarak yapılmıştır. Raporlanan sonuçlar rastgele noktalardan başlatılmış 30 farklı çalıştırmanın değerleridir.

4.3.1. TSP için komşuluk operatörlerinin karşılaştırılması

İlk test TSP için en iyi komşuluk operatörünü belirlemek için yapılmıştır. Bu deney için üç farklı duruma sahip üç komşuluk operatörü Berlin52 TSP üzerinde analiz edilmiştir. N=52 ve maxFEs=104000 (Dx2000) olarak ayarlanmış ve sonuçlar Çizelge 4.30'da gösterilmiştir. Çizelge 4.30'a göre en iyi komşuluk operatörü simetri operatörüdür. Simetri operatörü mevcut birey üzerinde daha fazla değişiklik yaptığı için belirlenen fonksiyon değerlendirme sayısında optimum değere ulaşmıştır.

Çizelge 4.30. Komşuluk operatörlerinin Berlin52 TSP'deki performans analizi

Yöntem	TSA Ort.	TSA+2Opt Ort.	En İyi	En Kötü	Std.Sap.	Ortanca	GAP
Takas (mevcut ağaç)	8133.00	7863.00	7863.00	7863.00	0.00	7863.00	4.08
Takas (rastgele ağaç)	8059.17	7858.00	7854.00	7863.00	3.22	7859.00	4.02
Takas (en iyi ağaç)	8133.00	7863.00	7863.00	7863.00	0.00	7863.00	4.08
Öteleme (mevcut ağaç)	7816.73	7678.57	7542.00	7842.00	83.39	7704.50	1.78
Öteleme (rastgele ağaç)	7903.13	7740.23	7569.00	7884.00	74.13	7741.00	2.56
Öteleme (en iyi ağaç)	7891.37	7758.43	7596.00	7816.00	65.71	7791.00	2.79
Simetri (mevcut ağaç)	7683.73	7542.00	7542.00	7542.00	0.00	7542.00	0.00
Simetri (rastgele ağaç)	7697.00	7542.00	7542.00	7542.00	0.00	7542.00	0.00
Simetri (en iyi ağaç)	7737.90	7551.03	7542.00	7813.00	49.48	7542.00	0.12

4.3.2. DTSA için N ve ST parametrelerinin belirlenmesi

DTSA için N değerinin çözülecek problemdeki toplam şehir sayısı olarak kullanılması kararlaştırılmıştır. Bu önerinin doğru olduğunu ispatlamak için, N 10'dan 300'e 10 artırılarak alınmış ve Krob150 problemi ST=0.5 ve maxFEs=800000 alınarak çözülmüştür. Çizelge 4.31'de görüldüğü gibi N parametresinin en optimuma yakın sonuçları ürettiği değerler popülasyon büyüklüğünün 120 ila 150 arasında olduğu durumlardır. Literatürdeki yaklaşımlarda (Kiran ve ark., 2013) da popülasyon büyüklüğü olarak toplam şehir sayısının alınması yaygın bir kullanımdır.

Çizelge 4.31. Krob150 TSP'de DTSA için popülasyon büyüklüğü analizi

N	Ortalama	En İyi	En Kötü	Std.Sap.	Ortanca	En İyi GAP	Ortalama GAP
10	27011.31	26457.52	27748.40	385.22	27052.20	1.25	3.37
20	26981.58	26450.82	27700.84	324.98	27071.33	1.23	3.26
30	27005.62	26600.86	27574.81	278.70	26996.37	1.80	3.35
40	26975.44	26580.77	27413.10	246.02	26939.55	1.73	3.24
50	26879.64	26580.77	27600.91	256.89	26846.01	1.73	2.87
60	26835.50	26527.84	27252.31	205.93	26845.50	1.52	2.70
70	26776.81	26446.72	27487.67	233.30	26685.50	1.21	2.48
80	26966.12	26580.77	27698.80	325.13	26925.12	1.73	3.20
90	26841.62	26580.77	27361.61	205.05	26834.09	1.73	2.72
100	26801.49	26554.42	27183.15	197.21	26738.98	1.62	2.57
110	26825.27	26472.15	27487.34	241.02	26737.77	1.31	2.66
120	26714.21	26481.24	27120.56	125.24	26680.20	1.34	2.24
130	26725.14	26492.04	27247.57	162.88	26665.66	1.39	2.28
140	26760.35	26450.82	27123.69	165.55	26714.46	1.23	2.41
150	26763.14	26482.95	27053.58	157.57	26726.64	1.35	2.42
160	26817.75	26513.03	27379.55	199.53	26782.62	1.47	2.63
170	26853.57	26507.09	27299.58	190.64	26813.50	1.44	2.77
180	26804.85	26527.84	27319.08	175.04	26770.29	1.52	2.58
190	26833.20	26580.41	27417.29	193.97	26782.62	1.72	2.69
200	26884.30	26523.75	27407.25	200.10	26834.51	1.51	2.89
210	26845.76	26637.43	27110.47	146.88	26817.82	1.94	2.74
220	26880.65	26632.48	27152.40	134.48	26865.99	1.92	2.87
230	26881.87	26564.33	27509.13	190.82	26809.66	1.66	2.88
240	26913.85	26694.28	27367.66	161.44	26876.47	2.16	3.00
250	26846.96	26637.62	27237.83	129.83	26839.47	1.94	2.74
260	26971.07	26701.00	27388.62	183.25	26917.33	2.19	3.22
270	26928.55	26610.06	27352.94	143.85	26923.70	1.84	3.06
280	26974.95	26657.74	27466.82	194.46	26956.07	2.02	3.23
290	27027.08	26743.15	27496.45	228.64	26949.01	2.35	3.43
300	26986.31	26686.64	27450.78	191.51	26921.35	2.13	3.28

ST parametresini analiz etmek için, ST değeri 0.1'den 0.9'a kadar 0.1'er artırılarak kullanılmıştır. Elde edilen sonuçlar Çizelge 4.32'de sunulmuştur. Sonuçlar incelendiğinde ortalama hatalar birbirine çok yakın olduğundan en iyi ve rastgele ağaçlara eşit şans vermek için diğer deneysel çalışmalarda ST=0.5 olarak kullanılmıştır.

Çizelge 4.32. Krob150 TSP'de DTSA için ST analizi

ST	Ortalama	En İyi	En Kötü	Std.Sap.	Ortanca	En İyi GAP	Ortalama GAP
0.1	26865.36	26519.81	27269.62	194.90	26869.17	1.49	2.81
0.2	26854.99	26523.75	27567.44	208.38	26786.06	1.51	2.77
0.3	26830.46	26469.12	27151.50	161.16	26831.27	1.30	2.68
0.4	26794.43	26580.77	27286.30	174.89	26748.47	1.73	2.54
0.5	26792.15	26446.72	27289.85	218.49	26719.28	1.21	2.53
0.6	26719.09	26466.54	27174.57	134.27	26714.27	1.29	2.25
0.7	26795.28	26580.77	27161.28	160.74	26782.62	1.73	2.55
0.8	26760.29	26580.77	27470.60	184.34	26714.27	1.73	2.41
0.9	26717.41	26446.72	27278.69	151.97	26714.27	1.21	2.25

Elde edilen bilgiler ışığında, sonraki deneylerde popülasyon büyüklüğü ilgili problemin toplam şehir sayısı, ST 0.5 ve tohum sayısı ise önerilen yöntemden dolayı 6 olarak kullanılmıştır.

4.3.3. DTSA'nın SA, ACO ve STA ile karşılaştırılması

Bu bölümde DTSA, yine komşuluk operatörleri ile çözüm üreten STA ile karşılaştırılmıştır. Chunhua ve ark. (2012) önerdikleri STA'yı SA ve ACO ile karşılaştırmıştır. İlgili çalışmada maxFEs=4000 olarak belirlenmiştir. Sonuçlar Çizelge 4.33'te raporlanmıştır. DTSA, Berlin52 problemi üzerinde daha iyi performans göstermiştir. Chunhua ve ark. (2012) çalışmalarında Att48 ve Ulysses16 problemlerini hatalı koordinat sistemi ile hesapladıklarından bu sonuçlar karşılaştırmalara dahil edilmemiştir.

Çizelge 4.33. DTSA'nın SA, ACO ve STA ile karşılaştırılması

Problem		SA	ACO	STA	DTSA
Berlin52	En İyi	8,186.40	8,240.40	7,544.40	7,542.00
	Ortalama	8,983.80	8,777.60	8,247.20	7,689.17
	En Kötü	9,585.80	9,151.30	8,630.50	7,929.00
	Std.Sap.	380.10	267.11	273.45	108.40

4.3.4. DTSA'nın SA ve DSTA varyantları ile karşılaştırılması

Zhou ve ark. (2016) çalışmalarında Kroa100, Krob100, Kroc100, Krod100 ve Kroe100 problemlerini DSTA ve varyantları ile çözmüş, elde ettikleri sonuçları SA ile karşılaştırmıştır. DTSA adil bir karşılaştırma için aynı parametrelerle çalıştırılmıştır. Tüm algoritmalar için maxFEs=90000 olarak belirlenmiştir. DTSA'nın SA ve DSTA varyantları ile karşılaştırılmasına ait sonuçlar Çizelge 4.34'te verilmiştir. Sonuçlar incelendiğinde, DTSA'nın Kroa100 ve Kroe100 problemleri için en iyi çözücü olduğu, diğer üç problem için ise en iyi ikinci çözücü olduğu gösterilmiştir.

Çizelge 4.34. DTSA'nın SA ve DSTA varyantları ile karşılaştırılması

Problem	Optimum	Algoritma	En İyi	Ortalama	Std.Sap.	En İyi GAP	Ortalama GAP
Kroa100	21,282	SA	21729.00	22635.00	778.72	2.10	6.36
		DSTA0	21853.00	23213.00	906.11	2.68	9.07
		DSTAI	21782.00	22835.00	715.85	2.35	7.30
		DSTAI	21294.00	21767.00	221.64	0.06	2.28
		DTSA	21294.40	21506.78	260.55	0.06	1.06
Krob100	22,141	SA	23032.00	23657.00	445.78	4.02	6.85
		DSTA0	23230.00	23794.00	517.05	4.92	7.47
		DSTAI	23012.00	23734.00	507.38	3.93	7.19
		DSTAI	22345.00	22880.00	302.14	0.92	3.34
		DTSA	22530.70	23139.26	181.74	1.76	4.51
Kroc100	20,749	SA	21417.00	22223.00	522.20	3.22	7.10
		DSTA0	21275.00	22877.00	709.87	2.54	10.26
		DSTAI	21038.00	21891.00	536.88	1.39	5.50
		DSTAI	20907.00	21378.00	246.34	0.76	3.03
		DTSA	21424.04	21817.08	217.77	3.25	5.15
Krod100	21,294	SA	21896.00	22911.00	483.01	2.83	7.59
		DSTA0	21945.00	23043.00	565.80	3.06	8.21
		DSTAI	21867.00	22665.00	592.53	2.69	6.44
		DSTAI	21380.00	21991.00	315.32	0.40	3.27
		DTSA	22342.16	22972.26	390.50	4.92	7.88
Kroe100	22,068	SA	22523.00	23125.00	389.42	1.73	4.44
		DSTA0	22692.00	23738.00	450.82	2.49	7.21
		DSTAI	22419.00	23371.00	678.69	1.26	5.56
		DSTAI	22311.00	22637.00	166.82	0.77	2.24
		DTSA	22261.00	22547.00	121.96	0.54	1.83

4.3.5. DTSA'nın ABC varyantları ile karşılaştırılması

Kiran ve ark. (2013) ABC algoritmasını TSP'lerin çözümü için komşuluk operatörleriyle birleştirmiştir. RS, RSS ve RRSS, takas operatörüne, RI ve RIS, öteleme operatörüne, RRS ve RRIS operatörleri ise simetri operatörüne benzerlik gösterir. Bahsi geçen çalışmada bu operatörler Kombine 1 (RS, RSS ve RRSS) ve Kombine 2 (RI, RIS ve RRIS) olarak adlandırılmış ve o şekilde de analizler yapılmıştır. Oliver30 (Çizelge 4.35), Eil51 (Çizelge 4.36), Berlin52 (Çizelge 4.37), St70 (Çizelge 4.38), Pr76 (Çizelge 4.39), Kroa100 (Çizelge 4.40), Eil101 (Çizelge 4.41), Tsp225 (Çizelge 4.42), A280 (Çizelge 4.43) problemleri kıyaslama işlemlerinde kullanılmıştır. Elde edilen sonuçlar Çizelge 4.35-4.43 arasında raporlanmıştır. Adil bir karşılaştırma yapmak için maxFES ilgili çalışmadaki gibi $D \times 100,000$ olarak ayarlanmıştır. D problemdeki şehir sayısıdır. Tablolardaki ABC+2-OPT, 2-opt uygulanmış ABC, ABC+3-OPT ise 3-opt uygulanmış ABC anlamına gelmektedir. Deneylerde, DTSA problemlerin boyutu arttığında daha başarılı sonuçlar üretmiştir. Berlin52, Kroa100, Tsp225 ve A280 problemleri için DTSA, diğer yaklaşımlardan daha iyi sonuçlar vermiştir. Diğer altı problem için DTSA rekabetçi çözümler üretmiştir.

Çizelge 4.35. DTSA'nın ABC varyantları ile Oliver30 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	439.83	439.83	439.83	423.74	510.94	510.94	503.59	435.07	477.86	477.86	476.37	426.74
RSS	423.74	423.74	423.74	423.74	423.74	423.74	423.74	435.07	423.74	423.74	423.74	426.74
RI	424.69	424.69	424.57	423.74	480.69	480.69	480.69	435.07	447.36	444.43	444.58	426.74
RIS	423.74	423.74	423.74	423.74	424.69	424.69	424.69	435.07	423.88	423.88	423.88	426.74
RR	423.74	423.74	423.74	423.74	429.83	429.83	429.38	435.07	425.16	425.16	424.91	426.74
RRIS	423.74	423.74	423.74	423.74	423.74	423.74	423.74	435.07	423.74	423.74	423.74	426.74
RRSS	423.74	423.74	423.74	423.74	423.74	423.74	423.74	435.07	423.74	423.74	423.74	426.74
Kombine 1	423.74	423.74	423.74	423.74	423.74	423.74	423.74	435.07	423.74	423.74	423.74	426.74
Kombine 2	423.74	423.74	423.74	423.74	423.74	423.74	423.74	435.07	423.74	423.74	423.74	426.74

Çizelge 4.36. DTSA'nın ABC varyantları ile Eil51 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	478.13	478.13	478.13	432.80	536.55	536.55	530.27	441.75	506.32	506.32	504.45	438.25
RSS	428.87	428.87	428.87	432.80	434.62	434.62	434.62	441.75	431.11	431.03	431.11	438.25
RI	452.21	449.71	449.64	432.80	486.17	486.17	486.01	441.75	467.62	463.28	465.03	438.25
RIS	431.17	431.17	431.17	432.80	439.33	439.33	439.33	441.75	435.78	435.77	435.78	438.25
RR	432.57	432.57	432.57	432.80	447.34	447.34	447.34	441.75	440.00	440.00	439.92	438.25
RRIS	428.87	428.87	428.87	432.80	433.93	433.93	433.93	441.75	430.41	430.37	430.41	438.25
RRSS	428.98	428.98	428.98	432.80	433.49	433.49	433.49	441.75	430.25	430.25	430.25	438.25
Kombine 1	428.87	428.87	428.87	432.80	433.60	433.60	433.60	441.75	430.08	430.08	430.08	438.25
Kombine 2	428.87	428.87	428.87	432.80	433.71	433.71	433.71	441.75	430.55	430.55	430.55	438.25

Çizelge 4.37. DTSA'nın ABC varyantları ile Berlin52 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	8,194.67	8,194.67	8,194.67	7,544.37	9,648.29	9,648.29	9,550.59	7,544.37	9,177.02	9,177.02	9,117.92	7,544.37
RSS	7,544.37	7,544.37	7,544.37	7,544.37	7,716.69	7,716.69	7,716.69	7,544.37	7,562.80	7,562.80	7,562.80	7,544.37
RI	7,898.21	7,898.21	7,898.21	7,544.37	8,850.17	8,850.17	8,850.17	7,544.37	8,470.14	8,395.98	8,401.40	7,544.37
RIS	7,544.37	7,544.37	7,544.37	7,544.37	7,746.86	7,746.86	7,746.86	7,544.37	7,591.43	7,591.43	7,591.43	7,544.37
RR	7,544.37	7,544.37	7,544.37	7,544.37	7,972.17	7,972.17	7,966.49	7,544.37	7,774.77	7,774.77	7,763.02	7,544.37
RRIS	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37
RRSS	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37
Kombine 1	7,544.37	7,544.37	7,544.37	7,544.37	7,598.44	7,598.44	7,598.44	7,544.37	7,548.47	7,547.07	7,548.47	7,544.37
Kombine 2	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37	7,544.37

Çizelge 4.38. DTSA'nın ABC varyantları ile St70 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	860.67	860.67	860.67	686.78	979.69	979.69	979.69	693.76	927.69	927.69	922.84	688.77
RSS	678.51	678.51	677.91	686.78	698.97	698.97	698.97	693.76	688.59	687.15	688.46	688.77
RI	739.29	728.66	736.17	686.78	831.12	820.76	826.70	693.76	785.14	773.34	777.03	688.77
RIS	681.87	677.11	681.87	686.78	700.37	697.10	700.37	693.76	690.04	687.87	690.03	688.77
RR	684.54	684.54	684.54	686.78	705.71	705.71	705.07	693.76	695.17	695.17	693.84	688.77
RRIS	677.11	677.11	677.11	686.78	688.70	688.70	688.70	693.76	681.52	680.85	681.32	688.77
RRSS	677.11	677.11	677.11	686.78	688.47	688.47	688.47	693.76	683.18	682.55	683.18	688.77
Kombine 1	678.51	678.51	678.51	686.78	691.33	691.33	691.33	693.76	684.17	684.17	684.05	688.77
Kombine 2	677.11	677.11	677.11	686.78	687.62	687.62	687.62	693.76	681.60	680.88	681.56	688.77

Çizelge 4.39. DTSA'nın ABC varyantları ile Pr76 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	137607.01	137607.01	137504.39	109653.04	155064.55	155064.55	154988.63	116301.17	147206.79	147206.79	146809.12	112747.33
RSS	108501.34	108304.51	108501.34	109653.04	111889.32	111005.89	111889.32	116301.17	110106.91	109871.01	110106.91	112747.33
RI	121188.99	118774.03	119910.98	109653.04	133682.89	131665.70	131665.70	116301.17	126341.76	124429.93	124738.89	112747.33
RIS	108428.00	108425.53	108428.00	109653.04	112039.72	111969.42	112039.72	116301.17	110397.12	110005.95	110356.60	112747.33
RR	108633.72	108633.72	108633.72	109653.04	110872.90	110872.90	110872.90	116301.17	109817.41	109817.41	109787.82	112747.33
RRIS	108159.44	108159.44	108159.44	109653.04	110050.90	109759.39	110050.90	116301.17	109005.00	108830.04	108965.31	112747.33
RRSS	108183.42	108159.44	108183.42	109653.04	109590.17	109590.17	109590.17	116301.17	108911.08	108848.04	108895.81	112747.33
Kombine 1	108159.44	108159.44	108159.44	109653.04	109782.95	109782.95	109777.01	116301.17	109164.47	109164.47	109164.13	112747.33
Kombine 2	108159.44	108159.44	108159.44	109653.04	109556.93	109556.93	109556.93	116301.17	108668.29	108642.94	108624.77	112747.33

Çizelge 4.40. DTSA'nın ABC varyantları ile Kroa100 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	29849.71	29849.71	29287.85	21294.40	36084.80	36084.80	36058.15	21748.85	33761.06	33761.06	33437.72	21386.20
RSS	21866.36	21450.50	21864.04	21294.40	23076.54	22519.65	22989.99	21748.85	22663.10	22081.42	22560.99	21386.20
RI	23410.06	23055.07	23212.28	21294.40	27475.19	27379.34	27400.60	21748.85	26204.50	25856.45	25959.69	21386.20
RIS	21724.60	21479.20	21595.15	21294.40	23069.65	22742.56	22871.99	21748.85	22490.43	21997.55	22331.84	21386.20
RR	21474.52	21474.52	21474.52	21294.40	22340.88	22340.88	22294.82	21748.85	21845.07	21845.07	21834.45	21386.20
RRIS	21730.11	21425.26	21729.52	21294.40	22417.75	22202.61	22323.63	21748.85	22080.16	21688.40	22013.77	21386.20
RRSS	21549.84	21285.44	21549.84	21294.40	22196.76	21957.07	22150.71	21748.85	21825.11	21687.13	21800.82	21386.20
Kombine 1	21419.13	21419.13	21419.13	21294.40	22042.91	22042.91	22042.91	21748.85	21759.41	21747.45	21749.99	21386.20
Kombine 2	21285.44	21285.44	21285.44	21294.40	21728.76	21728.76	21728.76	21748.85	21521.00	21506.96	21493.58	21386.20

Çizelge 4.41. DTSA'nın ABC varyantları ile Eil101TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	792.71	792.71	789.28	656.41	867.90	867.90	855.29	678.80	837.37	837.37	829.74	670.51
RSS	673.49	660.64	668.41	656.41	690.45	688.72	690.45	678.80	683.64	675.86	681.99	670.51
RI	710.79	706.00	710.79	656.41	759.81	753.16	753.67	678.80	730.02	724.38	726.53	670.51
RIS	675.26	665.05	672.80	656.41	686.47	682.30	686.15	678.80	681.49	674.82	679.90	670.51
RR	665.36	665.36	665.36	656.41	685.03	685.03	684.21	678.80	676.34	676.34	674.95	670.51
RRIS	660.60	652.62	657.54	656.41	677.46	672.59	676.57	678.80	670.82	664.86	669.15	670.51
RRSS	659.22	657.48	658.40	656.41	675.25	673.06	675.25	678.80	668.05	665.35	666.96	670.51
Kombine 1	658.60	658.60	657.53	656.41	675.56	675.56	675.56	678.80	667.80	667.58	667.27	670.51
Kombine 2	653.30	651.94	653.30	656.41	667.35	667.35	667.35	678.80	661.25	660.96	660.92	670.51

Çizelge 4.42. DTSA'nın ABC varyantları ile Tsp225 TSP üzerinde karşılaştırılması

	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	6732.46	6728.73	6690.53	3908.11	7642.61	7577.63	7520.06	4066.11	7264.12	7219.30	7142.35	3974.49
RSS	5370.21	4656.61	4934.29	3908.11	5785.40	5060.11	5287.12	4066.11	5578.45	4866.89	5069.09	3974.49
RI	4785.27	4726.63	4697.31	3908.11	5222.84	5160.86	5180.19	4066.11	5030.64	4916.51	4952.39	3974.49
RIS	5117.24	4517.09	4791.78	3908.11	5456.12	4893.74	5065.92	4066.11	5273.92	4720.48	4903.77	3974.49
RR	4150.86	4122.50	4130.17	3908.11	4231.84	4210.13	4231.73	4066.11	4183.45	4170.36	4177.70	3974.49
RRIS	5114.12	4448.90	4735.50	3908.11	5391.70	4868.10	5019.40	4066.11	5242.72	4693.40	4875.97	3974.49
RRSS	5031.31	4470.80	4713.42	3908.11	5348.51	4790.32	5047.78	4066.11	5165.84	4666.68	4835.13	3974.49
Kombine 1	4616.82	4491.62	4521.25	3908.11	4877.50	4774.55	4817.68	4066.11	4741.70	4624.32	4672.41	3974.49
Kombine 2	4217.11	4187.58	4216.11	3908.11	4503.33	4402.15	4489.95	4066.11	4439.00	4337.75	4404.18	3974.49

Çizelge 4.43. DTSA'nın ABC varyantları ile A280 TSP üzerinde karşılaştırılması

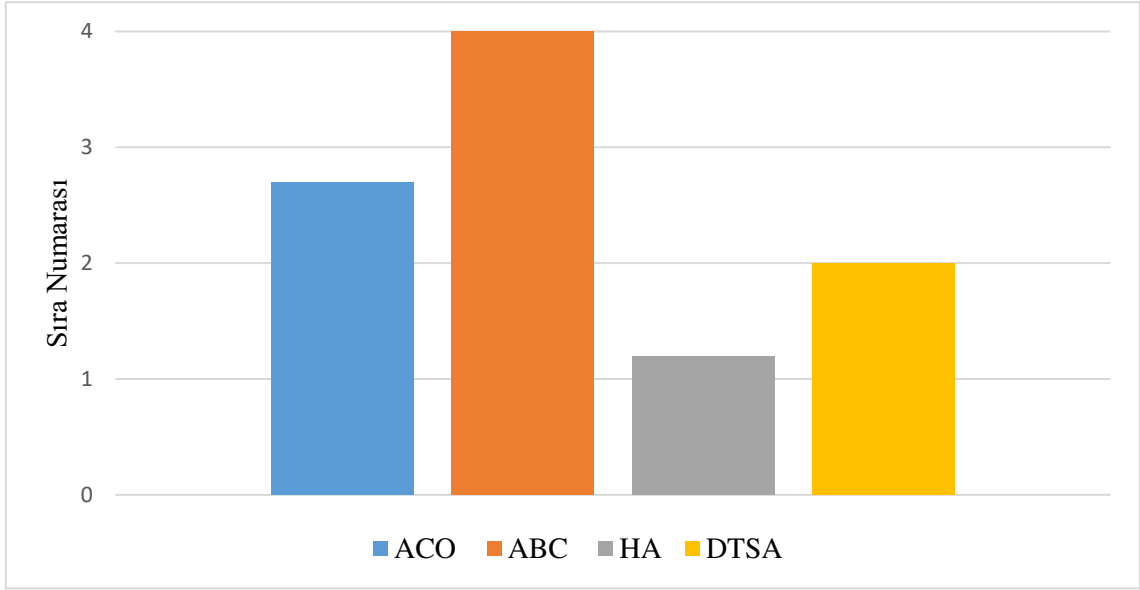
	En İyi				En Kötü				Ortalama			
	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA	ABC	ABC+2-OPT	ABC+3-OPT	DTSA
RS	5387.17	5330.41	5256.68	2627.05	5993.44	5910.20	5934.68	2724.35	5759.04	5642.34	5611.08	2686.87
RSS	4321.42	3531.69	3689.53	2627.05	4485.69	3863.26	4009.45	2724.35	4416.36	3685.15	3849.63	2686.87
RI	3450.76	3417.87	3411.70	2627.05	3772.11	3714.14	3727.61	2724.35	3674.23	3570.72	3574.11	2686.87
RIS	4003.09	3411.22	3601.74	2627.05	4212.11	3684.28	3826.07	2724.35	4136.35	3538.74	3714.04	2686.87
RR	2818.38	2812.68	2810.32	2627.05	2985.13	2953.46	2960.68	2724.35	2912.31	2878.30	2894.80	2686.87
RRIS	3973.27	3305.97	3514.56	2627.05	4171.97	3567.81	3734.22	2724.35	4082.35	3452.81	3642.13	2686.87
RRSS	4011.30	3354.50	3583.15	2627.05	4286.36	3681.99	3877.89	2724.35	4156.83	3570.00	3707.45	2686.87
Kombine 1	3476.62	3365.82	3426.27	2627.05	3670.07	3467.54	3576.73	2724.35	3584.49	3407.18	3497.76	2686.87
Kombine 2	3250.56	3151.34	3210.17	2627.05	3396.22	3275.21	3328.07	2724.35	3322.47	3205.48	3267.81	2686.87

4.3.6. DTSA'nın ACO, ABC, HA ve DPSO ile karşılaştırılması

Hiyerarşik yaklaşım (hierarchic approach, HA) Gunduz ve ark. (2015) tarafından önerilmiş ABC ve ACO'nun birlikte kullanıldığı bir algoritmadır. Bu bölümde DTSA HA, ACO, ABC ve DPSO ile karşılaştırılmıştır. Adil bir karşılaştırma için maxFEs sonuçların alındığı çalışmadaki gibi $D \times 500$ olarak ayarlanmıştır. D probleme ait toplam şehir sayısıdır. ABC, ACO ve HA'nın sonuçları doğrudan (Gunduz ve ark., 2015) çalışmasından ve DPSO algoritmasının sonuçları doğrudan (Shi ve ark., 2007) çalışmasından alınmıştır. Çizelge 4.44'te rapor edilen sonuçlar ve Şekil 4.48 incelendiğinde HA'nın daha iyi bir çözücü olduğunu görülmektedir. Bunun nedeni HA'nın problem ve popülasyon bilgisini birlikte kullanmasıdır. DTSA problem bilgisini kullanmadığından çok daha hızlı ve rekabetçi çözümler üretmektedir ancak tüm problemler için en iyi çözümü üretememektedir. Bu bilgiler çerçevesinde şu yorum yapılabilir; DTSA ACO, ABC ve DPSO'dan daha iyidir, fakat HA'dan daha kötüdür.

Çizelge 4.44. DTSA'nın ACO, ABC, HA ve DPSO ile karşılaştırılması

Problem	Yöntem	En İyi	En Kötü	Ortalama	Std.Sap.	GAP	maxFEs	Sıra
Oliver30	ACO	423.74	429.36	424.68	1.41	0.22	15000	3
	ABC	439.49	484.83	462.55	12.47	9.16	15000	4
	HA	423.74	423.74	423.74	0.00	0.00	15000	1
	DTSA	423.74	439.69	428.50	4.21	1.12	15000	2
Eil51	ACO	450.59	463.55	457.86	4.07	6.76	25500	3
	ABC	563.75	619.44	590.49	15.79	37.69	25500	4
	HA	431.74	454.97	443.39	5.25	3.39	25500	1
	DPSO	427.00	452.00	N/A	N/A	N/A	N/A	N/A
	DTSA	435.51	450.84	443.93	4.04	3.51	25500	2
Berlin52	ACO	7548.99	7681.75	7659.31	38.70	1.52	26000	3
	ABC	9479.11	11021.99	10390.26	439.69	37.72	26000	4
	HA	7544.37	7544.37	7544.37	0.00	0.00	26000	1
	DPSO	7542.00	8362.00	7952.00	N/A	5.40	N/A	N/A
	DTSA	7542.00	7657.00	7545.83	21.00	0.02	26000	2
St70	ACO	696.05	725.26	709.16	8.27	4.73	35000	3
	ABC	1162.12	1339.24	1230.49	41.79	81.73	35000	4
	HA	687.24	716.52	700.58	7.51	3.47	35000	1
	DPSO	675.00	742.00	N/A	N/A	N/A	N/A	N/A
	DTSA	690.01	725.19	708.65	6.77	4.66	35000	2
Eil76	ACO	554.46	568.62	561.98	3.50	3.04	38000	2
	ABC	877.28	971.36	931.44	24.86	70.78	38000	4
	HA	551.07	565.51	557.98	4.10	2.31	38000	1
	DPSO	546.00	579.00	N/A	N/A	N/A	N/A	N/A
	DTSA	571.55	588.15	578.58	3.93	6.09	38000	3
Pr76	ACO	115166.66	118227.41	116321.22	885.79	7.55	38000	3
	ABC	195198.90	219173.64	205119.61	7379.16	89.65	38000	4
	HA	113798.56	116353.01	115072.29	742.90	6.39	38000	2
	DPSO	108280.00	124365.00	N/A	N/A	N/A	N/A	N/A
	DTSA	111068.69	117028.63	114930.03	1545.64	6.26	38000	1
Kroa100	ACO	22455.89	23365.46	22880.12	235.18	7.49	50000	3
	ABC	49519.51	57566.05	53840.03	2198.36	152.94	50000	4
	HA	22122.75	23050.81	22435.31	231.34	5.40	50000	2
	DTSA	21294.40	22389.79	21728.40	358.13	2.08	50000	1
Eil101	ACO	678.04	705.65	693.42	6.80	7.96	50500	3
	ABC	1237.31	1392.64	1315.95	35.28	104.88	50500	4
	HA	672.71	696.04	683.39	6.56	6.40	50500	1
	DTSA	682.46	701.56	689.91	4.47	7.41	50500	2
Ch150	ACO	6648.51	6726.27	6702.87	20.73	2.61	75000	2
	ABC	20908.89	22574.99	21617.48	453.71	230.93	75000	4
	HA	6641.69	6707.86	6677.12	19.30	2.22	75000	1
	DTSA	6614.37	6803.15	6748.99	32.63	3.32	75000	3
Tsp225	ACO	4112.35	4236.85	4176.08	28.34	8.22	112500	2
	ABC	16998.41	18682.56	17955.12	387.35	365.28	112500	4
	HA	4090.54	4212.08	4157.85	26.27	7.74	112500	1
	DTSA	4102.60	4326.33	4230.45	58.76	9.63	112500	3



Şekil 4.48. ACO, ABC, HA ve DTSA için ortalama sıra değerlerinin karşılaştırılması

4.3.7. DTSA'nın ACO, PSO, GA ve BH ile karşılaştırılması

Hatamlou (2017), çalışmasında TSP'lerin çözümünde BH algoritmasını önermiş ve ACO, PSO ve GA ile karşılaştırmıştır. Ulysses22, Bays29, Bayg29, Att48, Eil51, Berlin52, St70, Eil76, Gr96 ve Eil101 problemleri bu çalışmada çözülmüştür. Bu çalışmadaki sonuçlar incelendiğinde elde edilen sonuçlarda koordinat tipi hesaplama hatası olduğu görülmüştür. Ulysses22 probleminin optimum değeri 7013 olmasına rağmen bu çalışmada 75 civarında bulunmuştur. Ulysses22 probleminin koordinatı GEO tipidir ancak bu çalışmada Bays29 (2020), Bayg29 (1610), Att48 (10628) ve Gr96 (55209) problemlerinde olduğu gibi EUC tipi olarak hesaplanmıştır. Bu nedenle, yalnızca Eil51, Berlin52, St70, Eil76 ve Eil101 problemlerine ait sonuçlar Çizelge 4.45'te raporlanarak karşılaştırma yapmak için kullanılmıştır. Adil bir karşılaştırma için DTSA ilgili çalışmada olduğu gibi, maxFEs=20000 alınarak yapılan 5 farklı çalışmanın sonuçlarıyla kıyaslanmıştır. N=100 ve ST=0.5 olarak alınmıştır. DTSA 5 problemin 4 tanesinde en iyi performansa sahip olurken, bir tanesinde ise en iyi ikinci çözücü olmuştur.

Çizelge 4.45. DTSA'nın ACO, PSO, GA ve BH ile karşılaştırılması

		En İyi	En Kötü	Ortalama	Std.Sap.
Eil51	ACO	454.3895	469.0531	461.0175	6.2974
	PSO	469.1551	737.5258	574.8022	107.2371
	GA	448.8397	462.1142	453.4773	9.4157
	BH	437.8930	526.8977	458.9252	38.6365
	DTSA	443.4276	467.6982	456.5184	8.9247
Berlin52	ACO	7757.0263	10541.1228	8522.9017	1152.2000
	PSO	9218.4682	14279.4331	11089.5286	2067.9323
	GA	8779.7559	9565.3744	9288.4483	1301.2108
	BH	8188.0714	9356.7483	8455.8304	508.9871
	DTSA	7662.0000	7815.0000	7761.6000	62.8594
St70	ACO	711.6515	855.2032	757.7540	59.6079
	PSO	1030.8484	1756.1227	1321.8137	269.2793
	GA	1112.3078	1242.2011	1158.8458	52.1734
	BH	723.2691	1081.1087	797.5745	125.2272
	DTSA	705.8918	713.2268	710.4037	2.7956
Eil76	ACO	574.2404	665.9995	594.1442	40.2152
	PSO	804.2667	1195.9021	975.6397	152.4061
	GA	619.2262	679.7864	652.0593	122.0972
	BH	566.2430	925.8417	659.1021	152.1754
	DTSA	579.9088	595.1275	588.0623	5.7296
Eil101	ACO	725.0996	868.2047	763.9207	59.9684
	PSO	1158.7040	1973.8192	1499.9911	319.7468
	GA	828.8806	854.4381	838.8307	9.9642
	BH	720.3838	1249.8684	897.3813	210.1446
	DTSA	679.5070	696.7050	689.8384	7.2994

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Bu tez kapsamında sürekli kısıtlı optimizasyon problemlerinin çözümü için CTSA, ayrık ikili optimizasyon problemlerinin çözümü için SimTSA, LogicTSA ve SimLogicTSA, ayrık tam sayılı optimizasyon problemlerinin çözümü için DTSA önerilmiştir.

CTSA, Deb'in (2000) kurallarını kullanarak kısıtlı optimizasyon problemlerini çözmek için TSA'nın değiştirilmesine odaklanmıştır. TSA yerel aramayı yeni tohumlar üretmek için gerçekleştirilmektedir. Çözüm uzayındaki farklı bölgeleri keşfetmek ve aramak için TSA iki güncelleme denklemi kullanmıştır ve bunlardan birinde rastgele komşu ağacın konumu kullanılır. Diğer denklemde ise TSA'nın yakınsama özelliğini geliştirmek için en iyi ağaç kullanılır. Tüm ağaçlar çözüm uzayında yerel bir optimuma takılı kalırsa, algoritma bu yerel optimumdan kurtulamaz, ancak bu durum çok sayıda yerel optimuma sahip bazı problemlerde ortaya çıkabilir. CTSA kısıtlı kıyas fonksiyonlarını çözerken kabul edilebilir bir performans göstermiş ve rekabetçi sonuçlar üretmiştir. TSA'nın ilk kısıtlı çözüm yaklaşımı TSA'ya bir ceza fonksiyonunun eklenmesiyle basınçlı tank tasarımı problemi'nin çözüldüğü bir yaklaşımdır. Ceza fonksiyonunun kullanımının kıyas fonksiyonlarında düşük performans göstermesinden dolayı CTSA içerisinde Deb'in kuralları kullanılmıştır. Deb'in kuralları hem kısıtların üstesinden gelir, hem de amaç fonksiyonun optimizasyonunda iyi performans gösterir. CTSA çözüm kalitesi açısından rekabetçi bir algoritma olduğunu göstermiştir. Yapılan analizler sonucunda sürekli kısıtsız optimizasyon için ST parametresi 0.1 ve N parametresi 10 olarak önerilmekteyken kısıtlı optimizasyon için N parametresi 40 ve ST parametresini 0.2 almanın daha iyi sonuçlar ürettiği gösterilmiştir. Ek olarak CTSA, yakın zamanda önerilen popülasyon temelli algoritmalarla karşılaştırıldığında, mühendislik tasarım problemlerini çözmeye karşılaştırılabilir ve gelecek vaat eden bir performans göstermektedir.

Önerilen ikili TSA varyantları (SimTSA, LogicTSA ve SimLogicTSA), ikili optimizasyon problemlerini çözmek için ikili karar değişkenleri ile çalışacak şekilde düzenlenmiştir. SimTSA ve LogicTSA'nın farklı problemleri çözme konusundaki performanslarından faydalanmak için hibrit (SimLogicTSA) bir yaklaşım önerilmiştir. Tüm ikili varyantlar için N ve ST parametreleri farklı koşullar altında analiz edilmiş ve parametrelerin algoritmaların performansı üzerinde oldukça etkili olduğu gösterilmiştir. Algoritmalarından kabul edilebilir bir performans elde etmek için $ST \in [0.2,0.6]$ ve $N \in$

[20,60] seçilmesi tavsiye edilmektedir. SimLogicTSA literatürdeki DE, ABC ve PSO algoritmalarının ikili varyantları ile kıyaslanmış, alternatif ve rekabetçi bir algoritma olduğu gösterilmiştir.

DTSA ayrık tam sayılı optimizasyon problemlerinin çözümü için önerilmiştir. Sürekli bir algoritma olan TSA'nın ayrıklaştırma sürecinde komşuluk operatörleri kullanılmıştır. Yeni çözümler oluşturmak için takas, öteleme ve simetri komşuluk operatörleri kullanılmıştır. Ek olarak, elde edilen en iyi çözümü geliştirmek için, 2-opt yerel arama algoritması kullanılmıştır. Bu yaklaşımın başarısını analiz etmek için simetrik TSP'ler kullanılmıştır. Bu tez çalışması kapsamında Ulysses16, Oliver30, Eil51, Berlin52, St70, Pr76, Kroa100, Krob100, Kroc100, Krod100, Kroe100, Eil101, Krob150, Tsp225 ve A280 problemleri çözülmüş ve sonuçlar ACO, SA, BH, GA, STA, ABC, PSO ve bazı varyantlarıyla karşılaştırılmıştır. Deneysel sonuçlar, DTSA'nın ayrık tam sayılı optimizasyon problemleri için alternatif ve rekabetçi bir çözücü olduğunu doğrulamıştır.

5.2 Öneriler ve Gelecekteki Çalışmalar

Kısıtlı optimizasyon yöntemlerini çözmek için önerilmiş diğer yaklaşımlarla TSA entegre edilerek başarısı gözlemlenebilir. Arama süreci sırasında çözüm uzayındaki yerel optimum bölgelere takıldığında TSA'nın yerel optimum noktalardan kurtulması için yeni mekanizmalar üzerinde çalışılabilir.

İkili optimizasyon problemlerinin çözümü için gerçekleştirilen deneysel sonuçlar karar değişkenlerinin sayısının önemli bir sorun olduğunu göstermektedir. Çünkü kıyaslanan algoritmaların hiçbiri her şart altında 100 boyutlu CapB ve CapC problemlerinin optimum çözümünü elde edememiştir. Bu sorunun üstesinden gelmek için bazı yerel arama yöntemleri, algoritmalar tarafından elde edilen çözümlere uygulanabilir. Farklı ikili optimizasyon problemlerinin (örneğin 0-1 sırt çantası problemi) çözümü için önerilmiş olan yaklaşımlar uygulanabilir.

Ayrık tam sayılı optimizasyon problemleri için önerilmiş DTSA yalnızca simetrik TSP'lerin çözümü için kullanılmıştır. Gelecekte, bu yaklaşım asimetric ve genelleştirilmiş TSP'leri çözmek için kullanılabilir. Ayrıca, ayrık gerçek dünya problemlerinden atölye planlama problemi (job shop scheduling problem) gibi problemler bu yaklaşımla çözülebilir.

KAYNAKLAR

- Ahmed, Z. H., 2010, Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, *International Journal of Biometrics & Bioinformatics (IJBB)*, 3 (6), 96.
- Akay, B., 2009, Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi, *Fen Bilimleri Enstitüsü, Doktora Tezi*.
- Akay, B. ve Karaboga, D., 2012, A modified artificial bee colony algorithm for real-parameter optimization, *Information Sciences*, 192, 120-142.
- Akay, B. B. ve Karaboga, D., 2017, Artificial bee colony algorithm variants on constrained optimization, *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 7 (1), 98-111.
- Al-Sultan, K. ve Al-Fawzan, M., 1999, A tabu search approach to the uncapacitated facility location problem, *Annals of Operations Research*, 86, 91-103.
- Albayrak, M. ve Allahverdi, N., 2011, Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms, *Expert Systems with Applications*, 38 (3), 1313-1320.
- Alves, M. ve Almeida, M., 1992, Simulated annealing algorithm for the simple plant location problem: A computational study, *Revista Investigacao Operacional*, 12, 1-31.
- Arora, S., 1998, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, *Journal of the ACM (JACM)*, 45 (5), 753-782.
- Aydin, M. E. ve Fogarty, T. C., 2004, A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems, *Journal of Heuristics*, 10 (3), 269-292.
- Azad, M., Kalam, A., Rocha, A. M. A. ve Fernandes, E. M. d. G., 2013, A simplified binary artificial fish swarm algorithm for uncapacitated facility location problems, *World Congress on Engineering 2013, WCE 2013*, 31-36.
- Azad, M. A. K., Rocha, A. M. A. ve Fernandes, E. M., 2012, Solving multidimensional 0-1 knapsack problem with an artificial fish swarm algorithm, *International Conference on Computational Science and Its Applications*, 72-86.
- Babalik, A., Cinar, A. C. ve Kiran, M. S., 2018a, A modification of tree-seed algorithm using Deb's rules for constrained optimization, *Applied Soft Computing*, 63 (Supplement C), 289-305.
- Babalik, A., Cinar, A. C. ve Kiran, M. S., 2018b, A modification of tree-seed algorithm using Deb's rules for constrained optimization, *Applied Soft Computing*, 63, 289-305.
- Babaoğlu, İ., 2016, Utilization of Bat Algorithm for Solving Uncapacitated Facility Location Problem, In: *Intelligent and Evolutionary Systems*, Eds: Springer, p. 199-208.
- Banitalebi, A., Aziz, M. I. A. ve Aziz, Z. A., 2016, A self-adaptive binary differential evolution algorithm for large scale binary optimization problems, *Information Sciences*, 367, 487-511.
- Bansal, J. C. ve Deep, K., 2012, A modified binary particle swarm optimization for knapsack problems, *Applied Mathematics and Computation*, 218 (22), 11042-11061.
- Banzhaf, W., 1990, The "molecular" traveling salesman, *Biological Cybernetics*, 64 (1), 7-14.

- Barcelo, J., Hallefjord, Å., Fernandez, E. ve Jörnsten, K., 1990, Lagrangean relaxation and constraint generation procedures for capacitated plant location problems with single sourcing, *Operations-Research-Spektrum*, 12 (2), 79-88.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. ve Vance, P. H., 1998, Branch-and-price: Column generation for solving huge integer programs, *Operations Research*, 46 (3), 316-329.
- Baykasoglu, A., Saltabas, A., Tasan, A. S. ve Subulan, K., 2013, Yapay Bağışıklık Sisteminin Çoklu Etmen Benzetim Ortamında Realize Edilmesi Ve Gezgin Satıcı Problemine Uygulanması, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 27 (4).
- Beasley, J. E., 1990, Or-Library - Distributing Test Problems by Electronic Mail, *Journal of the Operational Research Society*, 41 (11), 1069-1072.
- Beheshti, Z., Shamsuddin, S. M. ve Hasan, S., 2015, Memetic binary particle swarm optimization for discrete optimization problems, *Information Sciences*, 299, 58-84.
- Beşkirli, A., Özdemir, D. ve Temurtaş, H., 2019, A comparison of modified tree-seed algorithm for high-dimensional numerical functions, *Neural Computing and Applications*, 1-35.
- Beşkirli, M., 2019, Yüksek Boyutlu Test Fonksiyonlarında Ağaç Tohum Algoritmasının Performans Analizi, *Avrupa Bilim ve Teknoloji Dergisi*, 93-101.
- Brady, R., 1985, Optimization strategies gleaned from biological evolution, *Nature*, 317 (6040), 804-806.
- Braun, H., 1990, On solving travelling salesman problems by genetic algorithms, *International Conference on Parallel Problem Solving from Nature*, 129-133.
- Bryant, K. ve Benjamin, A., 2000, Genetic algorithms and the traveling salesman problem, *Department of Mathematics, Harvey Mudd College*, 10-12.
- Cagnina, L. C., Esquivel, S. C. ve Coello, C. A. C., 2006, A particle swarm optimizer for constrained numerical optimization, In: *Parallel problem solving from nature-PPSN IX*, Eds: Springer, p. 910-919.
- Chen, F., Ye, Z., Wang, C., Yan, L. ve Wang, R., 2018, A Feature Selection Approach for Network Intrusion Detection Based on Tree-Seed Algorithm and K-Nearest Neighbor, *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 68-72.
- Chen, M.-H., Chen, S.-H. ve Chang, P.-C., 2017a, Imperial competitive algorithm with policy learning for the traveling salesman problem, *Soft Computing*, 21 (7), 1863-1875.
- Chen, S.-M. ve Chien, C.-Y., 2011, Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques, *Expert Systems with Applications*, 38 (12), 14439-14450.
- Chen, W., Tan, X. ve Cai, M., 2017b, Parameter Identification of Equivalent Circuit Models for Li-ion Batteries Based on Tree Seeds Algorithm, *IOP Conference Series: Earth and Environmental Science*, 012024.
- Chen, W., Cai, M., Tan, X. ve Wei, B., 2019, Parameter Identification and State-of-Charge Estimation for Li-Ion Batteries Using an Improved Tree Seed Algorithm, *IEICE TRANSACTIONS on Information and Systems*, 102 (8), 1489-1497.
- Chen, Y., Xie, W. ve Zou, X., 2015, A binary differential evolution algorithm learning from explored solutions, *Neurocomputing*, 149, 1038-1047.
- Choi, S.-S., Cha, S.-H. ve Tappert, C. C., 2010, A survey of binary similarity and distance measures, *Journal of Systemics, Cybernetics and Informatics*, 8 (1), 43-48.

- Chunhua, Y., Xiaolin, T., Xiaojun, Z. ve Weihua, G., 2012, State transition algorithm for traveling salesman problem, *Control Conference (CCC), 2012 31st Chinese*, 2481-2485.
- Cinar, A. ve Kiran, M., 2016, A Parallel Version of Tree-Seed Algorithm (TSA) within CUDA Platform, *Selçuk International Scientific Conference On Applied Sciences*.
- Cinar, A. C., 2016, Ağaç-Tohum Algoritması için CUDA Tabanlı Bir Paralel Programlama Yaklaşımı, Yüksek lisans tezi, *Selçuk Üniversitesi*, Konya, Turkey, 102.
- Cinar, 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, *2017 International Conference on Computer Science and Engineering (UBMK)*, 571-576.
- Cinar, A. C., Iscan, H. ve Kiran, M. S., 2018, Tree-Seed algorithm for large-scale binary optimization, *KnE Social Sciences*, 3 (1), 48-64.
- Cinar, A. C. ve Kiran, M. S., 2018a, A parallel implementation of Tree-Seed Algorithm on CUDA-supported graphical processing unit, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33 (4), 1397-1409.
- Cinar, A. C. ve Kiran, M. S., 2018b, Similarity and logic gate-based tree-seed algorithms for binary optimization, *Computers & Industrial Engineering*, 115, 631-646.
- Cinar, A. C., Korkmaz, S. ve Kiran, M. S., 2019, A discrete tree-seed algorithm for solving symmetric traveling salesman problem, *Engineering Science and Technology, an International Journal*.
- Coello, C. A. C., 2002, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer methods in applied mechanics and engineering*, 191 (11-12), 1245-1287.
- Cornuéjols, G., Nemhauser, G. ve Wolsey, L., 1983, The Uncapacitated Facility Location Problem, *Cornell University Operations Research and Industrial Engineering*.
- Croes, G. A., 1958, A method for solving traveling-salesman problems, *Operations Research*, 6 (6), 791-812.
- Davis, L., 1985, Applying adaptive algorithms to epistatic domains, *IJCAI*, 162-164.
- Deb, K., 1991, Optimal design of a welded beam via genetic algorithms, *AIAA journal*, 29 (11), 2013-2015.
- Deb, K., 2000, An efficient constraint handling method for genetic algorithms, *Computer methods in applied mechanics and engineering*, 186 (2-4), 311-338.
- Deng, C., Zhao, B., Yang, Y., Peng, H. ve Wei, Q., 2011, Novel binary encoding differential evolution algorithm, *International Conference in Swarm Intelligence*, 416-423.
- Deng, W., Chen, R., He, B., Liu, Y., Yin, L. ve Guo, J., 2012, A novel two-stage hybrid swarm intelligence optimization algorithm and application, *Soft Computing*, 16 (10), 1707-1722.
- Deperlioglu, O. ve Kose, U., 2019, An Intelligent Medical Diagnosis System with Tree-Seed Algorithm Trained Support Vector Machines, *2nd International Conference on Data Science and Applications (ICONDATA'19)*, Balıkesir, Turkey.
- Ding, Z., Yao, R., Li, J. ve Lu, Z., 2018, Structural damage identification based on modified Artificial Bee Colony algorithm using modal data, *Inverse Problems in Science and Engineering*, 26 (3), 422-442.
- Ding, Z., Li, J., Hao, H. ve Lu, Z.-R., 2019a, Nonlinear hysteretic parameter identification using an improved tree-seed algorithm, *Swarm and Evolutionary Computation*, 46, 69-83.

- Ding, Z., Li, J., Hao, H. ve Lu, Z.-R., 2019b, Structural damage identification with uncertain modelling error and measurement noise by clustering based tree seeds algorithm, *Engineering Structures*, 185, 301-314.
- Ding, Z., Zhao, Y. ve Lu, Z., 2019c, Simultaneous identification of structural stiffness and mass parameters based on Bare-bones Gaussian Tree Seeds Algorithm using time-domain data, *Applied Soft Computing*, 83, 105602.
- Dorigo, M. ve Gambardella, L. M., 1997, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE transactions on evolutionary computation*, 1 (1), 53-66.
- Durmuş, B., 2019, Kaotik Harita Temelli Ağaç Tohum Algoritması, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 23 (2), 601-610.
- Eberhart, R. ve Kennedy, J., 1995, A new optimizer using particle swarm theory, *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43.
- El-Abd, M., 2013, An improved global-best harmony search algorithm, *Applied Mathematics and Computation*, 222, 94-106.
- El-Abd, M., 2017, Global-best brain storm optimization algorithm, *Swarm and Evolutionary Computation*, 37, 27-44.
- El-Fergany, A. A. ve Hasanien, H. M., 2018, Tree-seed algorithm for solving optimal power flow problem in large-scale power systems incorporating validations and comparisons, *Applied Soft Computing*, 64, 307-316.
- Eligüznel, İ. M., Özceylan, E. ve Çetinkaya, C., 2019, Testing Of Tree-Seed Algorithm On P-Median Benchmark Problems, *Sigma: Journal of Engineering & Natural Sciences/Mühendislik ve Fen Bilimleri Dergisi*, 37 (4).
- Engelbrecht, A. P. ve Pampara, G., 2007, Binary differential evolution strategies, *2007 IEEE Congress on Evolutionary Computation*, 1942-1947.
- Erlenkotter, D., 1978, A dual-based procedure for uncapacitated facility location, *Operations Research*, 26 (6), 992-1009.
- Fang, L., Zuo, H., Pang, L., Yang, Z., Zhang, X. ve Zhu, J., 2018, Image reconstruction through thin scattering media by simulated annealing algorithm, *Optics and Lasers in Engineering*, 106, 105-110.
- Feng, X., Liu, Y., Yu, H. ve Luo, F., 2017, Physarum-energy optimization algorithm, *Soft Computing*, 1-18.
- Fogel, D. B., 1988, An evolutionary approach to the traveling salesman problem, *Biological Cybernetics*, 60 (2), 139-144.
- Fogel, D. B., 1993, Applying evolutionary programming to selected traveling salesman problems, *Cybernetics and systems*, 24 (1), 27-36.
- Gandomi, A. H., Yang, X.-S. ve Alavi, A. H., 2011, Mixed variable structural optimization using firefly algorithm, *Computers & Structures*, 89 (23), 2325-2336.
- Gandomi, A. H., Yang, X.-S. ve Alavi, A. H., 2013, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering with computers*, 29 (1), 17-35.
- Geem, Z. W., Kim, J. H. ve Loganathan, G. V., 2001, A new heuristic optimization algorithm: Harmony search, *Simulation*, 76 (2), 60-68.
- Ghosh, D., 2003, Neighborhood search heuristics for the uncapacitated facility location problem, *European Journal of Operational Research*, 150 (1), 150-162.
- Goldberg, D. E. ve Lingle, R., 1985, Alleles, loci, and the traveling salesman problem, *Proceedings of an international conference on genetic algorithms and their applications*, 154-159.

- Gortázar, F., Duarte, A., Laguna, M. ve Martí, R., 2010, Black box scatter search for general classes of binary optimization problems, *Computers & Operations Research*, 37 (11), 1977-1986.
- Görkemli, B. ve Karaboga, D., 2013, Quick Combinatorial Artificial Bee Colony-qCABC-Optimization Algorithm for TSP, *The Second International Symposium on Computing in Informatics and Mathematics (ISCIM 2013)*, Tiran, Albania, 97-101.
- Grefenstette, J., Gopal, R., Rosmaita, B. ve Van Gucht, D., 1985, Genetic algorithms for the traveling salesman problem, *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, 160-168.
- Grefenstette, J. J., 1987, Incorporating problem specific knowledge in genetic algorithms, *Genetic algorithms and simulated annealing*, 42-60.
- Greistorfer, P. ve Rego, C., 2006, A simple filter-and-fan approach to the facility location problem, *Computers & Operations Research*, 33 (9), 2590-2601.
- Gulcu, Ş., Mahi, M., Baykan, Ö. K. ve Kodaz, H., 2016, A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem, *Soft Computing*, 1-17.
- Gunduz, M., Kiran, M. S. ve Ozceylan, E., 2015, A hierarchic approach based on swarm intelligence to solve the traveling salesman problem, *Turkish Journal of Electrical Engineering & Computer Sciences*, 23 (1), 103-117.
- Guner, A. R. ve Sevkli, M., 2008, A discrete particle swarm optimization algorithm for uncapacitated facility location problem, *Journal of Artificial Evolution and Applications*, 2008.
- Gungor, I., Emiroglu, B. G., Cinar, A. C. ve Kiran, M. S., 2019, Integration search strategies in tree seed algorithm for high dimensional function optimization, *International Journal of Machine Learning and Cybernetics*, 1-19.
- Hamida, S. B. ve Schoenauer, M., 2002, ASCHEA: new results using adaptive segregational constraint handling, *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 884-889.
- Hancer, E., Xue, B., Karaboga, D. ve Zhang, M., 2015, A binary ABC algorithm based on advanced similarity scheme for feature selection, *Applied Soft Computing*, 36, 334-348.
- Hansen, N. ve Ostermeier, A., 2001, Completely derandomized self-adaptation in evolution strategies, *Evolutionary computation*, 9 (2), 159-195.
- Hasançebi, O. ve Azad, S. K., 2012, An efficient metaheuristic algorithm for engineering optimization: SOPT, *int. j. optim. civil eng*, 2 (4), 479-487.
- Hatamlou, A., 2017, Solving travelling salesman problem using black hole algorithm, *Soft Computing*, 1-9.
- Holland, J. H., 1975, Adaptation in natural and artificial systems Ann Arbor, *The University of Michigan Press*, 1, 975.
- Holland, J. H., 1992, Genetic algorithms, *Scientific american*, 267 (1), 66-73.
- Holmberg, K., 1999, Exact solution methods for uncapacitated location problems with convex transportation costs, *European journal of operational research*, 114 (1), 127-140.
- Hornig, S.-C. ve Lin, S.-S., 2018, Embedding Ordinal Optimization into Tree-Seed Algorithm for Solving the Probabilistic Constrained Simulation Optimization Problems, *Applied Sciences*, 8 (11), 2153.
- Hussain, A., Muhammad, Y. S., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A. ve Gani, S., 2017, Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator, *Computational Intelligence and Neuroscience*, 2017.

- Hussain, S. F. ve Haris, M., 2019, A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data, *Expert Systems with Applications*, 118, 20-34.
- Iscan, H. ve Gunduz, M., 2017, An application of fruit fly optimization algorithm for traveling salesman problem, *Procedia Computer Science*, 111 (Supplement C), 58-63.
- Jaccard, P., 1901, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bull Soc Vaudoise Sci Nat*, 37, 547-579.
- Jakob, K. ve Pruzan, P. M., 1983, The simple plant location problem: survey and synthesis, *European journal of operational research*, 12, 36-81.
- Jaramillo, J. H., Bhadury, J. ve Batta, R., 2002, On the use of genetic algorithms to solve location problems, *Computers & Operations Research*, 29 (6), 761-779.
- Jia, D., Duan, X. ve Khan, M. K., 2014, Binary Artificial Bee Colony optimization using bitwise operation, *Computers & Industrial Engineering*, 76, 360-365.
- 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.
- Jiang, J., Xu, M., Meng, X. ve Li, K., 2020, STSA: A sine Tree-Seed Algorithm for complex continuous optimization problems, *Physica A: Statistical Mechanics and its Applications*, 537, 122802.
- Jiang Xue-ying, Shi Hui-yuan, Su Cheng-li ve Ping, L., 2018, Nonlinear Neural Network Predictive Control Based on Tree and Seed Algorithm, *JOURNAL OF APPLIED SCIENCES — Electronics and Information Engineering*, 36 (5).
- Jubeir, M. B., Almazrooie, M. ve Abdullah, R., 2017, Enhanced selection method for genetic algorithm to solve traveling salesman problem, *Proceedings of the 6th International Conference of Computing & Informatics*, Kuala Lumpur, 69-76.
- Karaboga, D. ve Basturk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization*, 39 (3), 459-471.
- Karaboga, D. ve Basturk, B., 2008, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 8 (1), 687-697.
- Karaboga, D. ve Akay, B., 2009, A comparative study of Artificial Bee Colony algorithm, *Applied mathematics and computation*, 214 (1), 108-132.
- Karaboga, D. ve Gorkemli, B., 2011, A combinatorial artificial bee colony algorithm for traveling salesman problem, *Innovations in intelligent systems and applications (inista), 2011 international symposium on*, 50-53.
- Karatas, S. K., 2011, Diferensiyel (farksal) gelişim algoritması kullanılarak kısıt yönetimi metotlarının sonuçlarının ve performanslarının karşılaştırılması, Yüksek lisans tezi, *Erciyes University*.
- Kashan, M. H., Nahavandi, N. ve Kashan, A. H., 2012, DisABC: a new artificial bee colony algorithm for binary optimization, *Applied Soft Computing*, 12 (1), 342-352.
- Kashan, M. H., Kashan, A. H. ve Nahavandi, N., 2013, A novel differential evolution algorithm for binary optimization, *Computational Optimization and Applications*, 55 (2), 481-513.
- Kaveh, A. ve Khayatazad, M., 2012, A new meta-heuristic method: ray optimization, *Computers & Structures*, 112, 283-294.
- Kazemzadeh Azad, S., Hasançebi, O. ve Erol, O., 2011, Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems, *Iran University of Science & Technology*, 1 (3), 495-505.

- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6*, 1942-1948.
- Kennedy, J. ve Eberhart, R. C., 1997, A discrete binary version of the particle swarm algorithm, *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, 4104-4108.
- Kiran, M. S. ve Gunduz, M., 2013, XOR-based artificial bee colony algorithm for binary optimization, *Turkish Journal of Electrical Engineering & Computer Sciences*, 21 (Sup. 2), 2307-2328.
- Kiran, M. S., Iscan, H. ve Gunduz, M., 2013, The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem, *Neural Computing and Applications*, 23 (1), 9-21.
- Kiran, M. S., 2015a, TSA: Tree-seed algorithm for continuous optimization, *Expert Systems with Applications*, 42 (19), 6686-6698.
- Kiran, M. S., 2015b, The continuous artificial bee colony algorithm for binary optimization, *Applied Soft Computing*, 33, 15-23.
- Kiran, M. S., 2016, An implementation of tree-seed algorithm (TSA) for constrained optimization, In: *Intelligent and Evolutionary Systems*, Eds: Springer, p. 189-197.
- Kiran, M. S., 2017a, Withering process for tree-seed algorithm, *Procedia computer science*, 111, 46-51.
- Kiran, M. S., 2017b, Withering process for tree-seed algorithm, *Procedia Computer Science*, 111 (Supplement C), 46-51.
- Knox, J., 1994, Tabu search performance on the symmetric traveling salesman problem, *Computers & Operations Research*, 21 (8), 867-876.
- Kong, M., Tian, P. ve Kao, Y., 2008, A new ant colony optimization algorithm for the multidimensional knapsack problem, *Computers & Operations Research*, 35 (8), 2672-2683.
- Kong, X., Gao, L., Ouyang, H. ve Li, S., 2015a, A simplified binary harmony search algorithm for large scale 0–1 knapsack problems, *Expert Systems with Applications*, 42 (12), 5337-5355.
- Kong, X., Gao, L., Ouyang, H. ve Li, S., 2015b, Solving large-scale multidimensional knapsack problems with a new binary harmony search algorithm, *Computers & Operations Research*, 63, 7-22.
- Korkmaz, S., Babalik, A. ve Kiran, M. S., 2018, An artificial algae algorithm for solving binary optimization problems, *International Journal of Machine Learning and Cybernetics*, 9 (7), 1233-1247.
- Korkmaz, S. ve Kiran, M. S., 2018, An artificial algae algorithm with stigmergic behavior for binary optimization, *Applied Soft Computing*, 64, 627-640.
- Koziel, S. ve Michalewicz, Z., 1999, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, *Evolutionary computation*, 7 (1), 19-44.
- Kratica, J., Tošić, D., Filipović, V. ve Ljubić, I., 2001, Solving the simple plant location problem by genetic algorithm, *RAIRO-Operations Research*, 35 (1), 127-142.
- Laalaoui, Y. ve M'Hallah, R., 2016, A binary multiple knapsack model for single machine scheduling with machine unavailability, *Computers & Operations Research*, 72, 71-82.
- Laporte, G. ve Nobert, Y., 1980, A cutting planes algorithm for the m-salesmen problem, *Journal of the operational research society*, 1017-1023.
- Larranaga, P., Kuijpers, C. M., Murga, R. H., Inza, I. ve Dizdarevic, S., 1999, Genetic algorithms for the travelling salesman problem: A review of representations and operators, *Artificial Intelligence Review*, 13 (2), 129-170.

- Larrañaga, P., Kuijpers, C. M., Poza, M. ve Murga, R. H., 1997, Decomposing Bayesian networks: triangulation of the moral graph with genetic algorithms, *Statistics and Computing*, 7 (1), 19-34.
- Lawler, E. L. ve Wood, D. E., 1966, Branch-and-bound methods: A survey, *Operations Research*, 14 (4), 699-719.
- Lee, K. S. ve Geem, Z. W., 2005, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer methods in applied mechanics and engineering*, 194 (36-38), 3902-3933.
- Li, L., Cheng, Y., Tan, L. ve Niu, B., 2011, A discrete artificial bee colony algorithm for TSP problem, *International Conference on Intelligent Computing*, 566-573.
- Li, L., Muneeswaran, V., Ramkumar, S., Emayavaramban, G. ve Gonzalez, G. R., 2019, Metaheuristic FIR filter with Game theory based Compression Technique-A Reliable Medical Image Compression Technique for online applications, *Pattern Recognition Letters*.
- Liang, J., Qu, B. ve Suganthan, P., 2013, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*.
- Lin, S. ve Kernighan, B. W., 1973, An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, 21 (2), 498-516.
- Mahdavi, M., Fesanghary, M. ve Damangir, E., 2007, An improved harmony search algorithm for solving optimization problems, *Applied mathematics and computation*, 188 (2), 1567-1579.
- Mahi, M., Baykan, Ö. K. ve Kodaz, H., 2015, A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem, *Applied Soft Computing*, 30, 484-490.
- Mallipeddi, R. ve Suganthan, P. N., 2010, Ensemble of constraint handling techniques, *IEEE transactions on evolutionary computation*, 14 (4), 561-579.
- Martin, B., Marot, J. ve Bourennane, S., 2019, Mixed grey wolf optimizer for the joint denoising and unmixing of multispectral images, *Applied Soft Computing*, 74, 385-410.
- Martin, O., Otto, S. W. ve Felten, E. W., 1991, Large-step Markov chains for the traveling salesman problem.
- Mezura-Montes, E. ve Coello, C. A. C., 2005, A simple multimembered evolution strategy to solve constrained optimization problems, *IEEE transactions on evolutionary computation*, 9 (1), 1-17.
- Mezura-Montes, E., Coello Coello, C., Velázquez-Reyes, J. ve Muñoz-Dávila, L., 2007, Multiple trial vectors in differential evolution for engineering design, *Engineering Optimization*, 39 (5), 567-589.
- Michalewicz, Z., 1992, *Genetic algorithms+ data structures= evolution programs*, Springer Science & Business Media, p.
- Michalewicz, Z., 1995, A Survey of Constraint Handling Techniques in Evolutionary Computation Methods, *Evolutionary Programming*, 4, 135-155.
- Michel, L. ve Van Hentenryck, P., 2004, A simple tabu search for warehouse location, *European journal of operational research*, 157 (3), 576-591.
- Mirjalili, S., Mirjalili, S. M. ve Lewis, A., 2014a, Grey wolf optimizer, *Advances in Engineering Software*, 69, 46-61.

- Mirjalili, S., Wang, G.-G. ve Coelho, L. d. S., 2014b, Binary optimization using hybrid particle swarm optimization and gravitational search algorithm, *Neural Computing and Applications*, 25 (6), 1423-1435.
- Mirjalili, S., 2015, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, 89, 228-249.
- Mirjalili, S. ve Lewis, A., 2016, The whale optimization algorithm, *Advances in Engineering Software*, 95, 51-67.
- Mirjalili, S., Mirjalili, S. M. ve Hatamlou, A., 2016, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, 27 (2), 495-513.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. ve Mirjalili, S. M., 2017, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, 114, 163-191.
- Mittal, N., Singh, U. ve Sohi, B. S., 2016, Modified grey wolf optimizer for global engineering optimization, *Applied Computational Intelligence and Soft Computing*, 2016, 8.
- Monabbati, E. ve Kakhki, H. T., 2015, On a class of subadditive duals for the uncapacitated facility location problem, *Applied Mathematics and Computation*, 251, 118-131.
- Muneeswaran, V. ve Rajasekaran, M. P., 2016, Performance evaluation of radial basis function networks based on tree seed algorithm, *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 1-4.
- Muneeswaran, V. ve Rajasekaran, M. P., 2017, Beltrami-regularized denoising filter based on tree seed optimization algorithm: an ultrasound image application, *International Conference on Information and Communication Technology for Intelligent Systems*, 449-457.
- Muneeswaran, V. ve Rajasekaran, M. P., 2018, Gallbladder shape estimation using tree-seed optimization tuned radial basis function network for assessment of acute cholecystitis, In: *Intelligent Engineering Informatics*, Eds: Springer, p. 229-239.
- Muneeswaran, V. ve Rajasekaran, M. P., 2019, Local Contrast Regularized Contrast Limited Adaptive Histogram Equalization Using Tree Seed Algorithm—An Aid for Mammogram Images Enhancement, In: *Smart Intelligent Computing and Applications*, Eds: Springer, p. 693-701.
- Muñoz Zavala, A. E., Aguirre, A. H. ve Villa Diharce, E. R., 2005, Constrained optimization via particle evolutionary swarm optimization algorithm (PESO), *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 209-216.
- Mühlenbein, H., Gorges-Schleuter, M. ve Krämer, O., 1988, Evolution algorithms in combinatorial optimization, *Parallel Computing*, 7 (1), 65-85.
- Mühlenbein, H., 1989, Parallel genetic algorithms, population genetics and combinatorial optimization, *Workshop on Parallel Processing: Logic, Organization, and Technology*, 398-406.
- Nezamabadi-pour, H., Rostami-Shahrbabaki, M. ve Maghfoori-Farsangi, M., 2008, Binary particle swarm optimization: challenges and new solutions, *CSI J Comput Sci Eng*, 6 (1), 21-32.
- Nezamabadi-pour, H., 2015, A quantum-inspired gravitational search algorithm for binary encoded optimization problems, *Engineering Applications of Artificial Intelligence*, 40, 62-75.

- Oliva, D., Elaziz, M. A. ve Hinojosa, S., 2019, Otsu's Between Class Variance and the Tree Seed Algorithm, In: *Metaheuristic Algorithms for Image Segmentation: Theory and Applications*, Eds: Springer, p. 71-83.
- Oliver, I., Smith, D. ve Holland, J. R., 1987, Study of permutation crossover operators on the traveling salesman problem, *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*.
- Or, I., 1976, Traveling Salesman Type Combinatorial Problems And Their Relation To The Logistics Of Regional Blood Banking, Doktora tezi, *Northwestern University*.
- Ouaarab, A., Ahiod, B. ve Yang, X.-S., 2014, Discrete cuckoo search algorithm for the travelling salesman problem, *Neural Computing and Applications*, 24 (7-8), 1659-1669.
- Ouaarab, A., Ahiod, B. ve Yang, X.-S., 2015, Random-key cuckoo search for the travelling salesman problem, *Soft Computing*, 19 (4), 1099-1106.
- Ozturk, C., Hancer, E. ve Karaboga, D., 2015, A novel binary artificial bee colony algorithm based on genetic operators, *Information Sciences*, 297, 154-170.
- Padberg, M. ve Rinaldi, G., 1987, Optimization of a 532-city symmetric traveling salesman problem by branch and cut, *Operations Research Letters*, 6 (1), 1-7.
- Pampara, G., Engelbrecht, A. P. ve Franken, N., 2006, Binary differential evolution, *2006 IEEE International Conference on Evolutionary Computation*, 1873-1879.
- Pampará, G. ve Engelbrecht, A. P., 2011, Binary artificial bee colony optimization, *2011 IEEE Symposium on Swarm Intelligence*, 1-8.
- Pang, W., Wang, K.-p., Zhou, C.-g. ve Dong, L.-j., 2004, Fuzzy discrete particle swarm optimization for solving traveling salesman problem, *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, 796-800.
- Parsopoulos, K. E. ve Vrahatis, M. N., 2002, Particle swarm optimization method for constrained optimization problems, *Intelligent Technologies—Theory and Application: New Trends in Intelligent Technologies*, 76 (1), 214-220.
- Potvin, J.-Y., 1993, State-of-the-art survey—the traveling salesman problem: A neural network perspective, *ORSA Journal on Computing*, 5 (4), 328-348.
- Potvin, J.-Y., 1996, Genetic algorithms for the traveling salesman problem, *Annals of Operations Research*, 63 (3), 337-370.
- Pulido, G. T. ve Coello, C. A. C., 2004, A constraint-handling mechanism for particle swarm optimization, *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, 1396-1403.
- Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S., 2009, GSA: a gravitational search algorithm, *Information Sciences*, 179 (13), 2232-2248.
- Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S., 2010, BGSA: binary gravitational search algorithm, *Natural Computing*, 9 (3), 727-745.
- Ravikumar, C., 1992, Parallel techniques for solving large scale travelling salesperson problems, *Microprocessors and Microsystems*, 16 (3), 149-158.
- Ray, T. ve Liew, K.-M., 2003a, Society and civilization: An optimization algorithm based on the simulation of social behavior, *IEEE transactions on evolutionary computation*, 7 (4), 386-396.
- Ray, T. ve Liew, K. M., 2003b, Society and civilization: An optimization algorithm based on the simulation of social behavior, *IEEE Transactions on Evolutionary Computation*, 7 (4), 386-396.

- Rechenberg, I., 1973, Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Information, *Stuttgart-Bad Cannstatt: Friedrich Frommann Verlag*.
- Reinelt, G., 1991, TSPLIB—A traveling salesman problem library, *ORSA Journal on Computing*, 3 (4), 376-384.
- Runarsson, T. P. ve Yao, X., 2000, Stochastic ranking for constrained evolutionary optimization, *IEEE transactions on evolutionary computation*, 4 (3), 284-294.
- Saha, S., Kole, A. ve Dey, K., 2011, A Modified Continuous Particle Swarm Optimization Algorithm for Uncapacitated Facility Location Problem, *International Conference on Advances in Information Technology and Mobile Communication*, 305-311.
- Sahman, M., Cinar, A., Saritas, I. ve Yasar, A., 2019, Tree-seed algorithm in solving real-life optimization problems, *IOP Conference Series: Materials Science and Engineering*, 012030.
- Sahman, M. A. ve Cinar, A. C., 2019, Binary tree-seed algorithms with S-shaped and V-shaped transfer functions, *International Journal of Intelligent Systems and Applications in Engineering*, 7 (2), 111-117.
- Salman, A. A., Omran, M. G. ve Ahmad, I., 2015, Adaptive probabilistic harmony search for binary optimization problems, *Memetic Computing*, 7 (4), 291-316.
- Saremi, S., Mirjalili, S. ve Lewis, A., 2017, Grasshopper optimisation algorithm: theory and application, *Advances in Engineering Software*, 105, 30-47.
- Sevkli, M. ve Guner, A. R., 2006, A continuous particle swarm optimization algorithm for uncapacitated facility location problem, *International Workshop on Ant Colony Optimization and Swarm Intelligence*, 316-323.
- Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C. ve Wang, Q., 2007, Particle swarm optimization-based algorithms for TSP and generalized TSP, *Information Processing Letters*, 103 (5), 169-176.
- Starkweather, T., McDaniel, S., Mathias, K. E., Whitley, L. D. ve Whitley, C., 1991, A Comparison of Genetic Sequencing Operators, *ICGA*, 69-76.
- Storn, R. ve Price, K., 1997, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, 11 (4), 341-359.
- Sun, M., 2006, Solving the uncapacitated facility location problem using tabu search, *Computers & Operations Research*, 33 (9), 2563-2589.
- Syswerda, G., 1991, Scheduling optimization using genetic algorithms, *Handbook of genetic algorithms*, 322-349.
- Tanabe, R. ve Fukunaga, A. S., 2014, Improving the search performance of SHADE using linear population size reduction, *2014 IEEE congress on evolutionary computation (CEC)*, 1658-1665.
- Tohyama, H., Ida, K. ve Matsueda, J., 2011, A genetic algorithm for the uncapacitated facility location problem, *Electronics and Communications in Japan*, 94 (5), 47-54.
- Tsai, H.-K., Yang, J.-M., Tsai, Y.-F. ve Kao, C.-Y., 2004, Some issues of designing genetic algorithms for traveling salesman problems, *Soft Computing*, 8 (10), 689-697.
- Ulder, N. L., Aarts, E. H., Bandelt, H.-J., Van Laarhoven, P. J. ve Pesch, E., 1990, Genetic local search algorithms for the traveling salesman problem, *International Conference on Parallel Problem Solving from Nature*, 109-116.
- Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.

- Üçoluk, G., 2002, Genetic algorithm solution of the TSP avoiding special crossover and mutation, *Intelligent Automation & Soft Computing*, 8 (3), 265-272.
- Wang, K.-P., Huang, L., Zhou, C.-G. ve Pang, W., 2003, Particle swarm optimization for traveling salesman problem, *Machine Learning and Cybernetics, 2003 International Conference on*, 1583-1585.
- Wang, L., Fu, X., Menhas, M. I. ve Fei, M., 2010a, A modified binary differential evolution algorithm, In: *Life System Modeling and Intelligent Computing*, Eds: Springer, p. 49-57.
- Wang, L., Xu, Y., Mao, Y. ve Fei, M., 2010b, A discrete harmony search algorithm, In: *Life System Modeling and Intelligent Computing*, Eds: Springer, p. 37-43.
- Whitley, D., Starkweather, T. ve Fuquay, D., 1989, Scheduling problems and traveling salesman: The genetic edge recombination operator, *Proc. 3rd Int. Conf. Genetic Algorithms*, 133-140.
- Whitley, D., Starkweather, T. ve Shaner, D., 1991, The traveling salesman and sequence scheduling: Quality solutions using genetic edge recombination, Colorado State University, Department of Computer Science, p.
- Wolpert, D. H. ve Macready, W. G., 1997, No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, 1 (1), 67-82.
- Wu, S.-J. ve Chow, P.-T., 1995, Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, *Engineering Optimization*, 24 (2), 137-159.
- Xiao, J.-h., Huang, Y.-f. ve Cheng, Z., 2013, A bio-inspired algorithm based on membrane computing for engineering design problem, *Int. J. Comput. Sci*, 10, 580-588.
- Yan, X., Li, W., Chen, W., Luo, W., Zhang, C. ve Liu, H., 2012a, Cultural algorithm for engineering design problems, *International Journal of Computer Science Issues (IJCSI)*, 9 (6), 53.
- Yan, X., Li, W., Chen, W., Luo, W., Zhang, C. ve Liu, H., 2012b, Cultural algorithm for engineering design problems, *International Journal of Computer Science Issues*, 9 (6), 53-61.
- Yang, Q., 2008, A comparative study of discrete differential evolution on binary constraint satisfaction problems, *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 330-335.
- Yang, X. S., 2010a, Firefly algorithm, stochastic test functions and design optimisation, *International Journal of Bio-Inspired Computation*, 2 (2), 78-84.
- Yang, X. S., 2010b, A New Metaheuristic Bat-Inspired Algorithm, *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization*, 284, 65-74.
- Yildirim, A. E. ve Karci, A., 2017, Applications of artificial atom algorithm to small-scale traveling salesman problems, *Soft Computing*, 1-13.
- Yong, W., 2015, Hybrid Max–Min ant system with four vertices and three lines inequality for traveling salesman problem, *Soft Computing*, 19 (3), 585-596.
- Yuan, X., Nie, H., Su, A., Wang, L. ve Yuan, Y., 2009, An improved binary particle swarm optimization for unit commitment problem, *Expert Systems with Applications*, 36 (4), 8049-8055.
- Zambrano-Bigiarini, M., Clerc, M. ve Rojas, R., 2013, Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements, *2013 IEEE Congress on Evolutionary Computation*, 2337-2344.
- Zhang, M., Luo, W. ve Wang, X., 2008, Differential evolution with dynamic stochastic selection for constrained optimization, *Information Sciences*, 178 (15), 3043-3074.

- Zhang, W.-J. ve Xie, X.-F., 2003, DEPSO: hybrid particle swarm with differential evolution operator, *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, 3816-3821.
- Zhang, X., Wu, C., Li, J., Wang, X., Yang, Z., Lee, J.-M. ve Jung, K.-H., 2016, Binary artificial algae algorithm for multidimensional knapsack problems, *Applied Soft Computing*, 43, 583-595.
- Zhao, S., Wang, N. ve Liu, X., 2019, Artificial bee colony algorithm with tree-seed searching for modeling multivariable systems using GRNN, *2019 Chinese Control And Decision Conference (CCDC)*, 4702-4707.
- Zheng, Y., Zhou, J., Zhu, W., Zhang, C., Li, C. ve Fu, W., 2016, Design of a multi-mode intelligent model predictive control strategy for hydroelectric generating unit, *Neurocomputing*, 207, 287-299.
- Zhou, J., Zheng, Y., Xu, Y., Liu, H. ve Chen, D., 2018, A heuristic TS fuzzy model for the pumped-storage generator-motor using variable-length tree-seed algorithm-based competitive agglomeration, *Energies*, 11 (4), 944.
- Zhou, X., Gao, D. Y., Yang, C. ve Gui, W., 2016, Discrete state transition algorithm for unconstrained integer optimization problems, *Neurocomputing*, 173, 864-874.
- Zhu, G. ve Kwong, S., 2010, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, 217 (7), 3166-3173.

EKLER

EK-1 CTSA için kullanılan sürekli kısıtlı optimizasyon problemleri

G01 Problemi

$$\begin{aligned} \min f(x) &= 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - 5 \sum_{i=5}^{13} x_i \\ g_1(x) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\ g_2(x) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\ g_3(x) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\ g_4(x) &= -8x_1 + x_{10} \leq 0 \\ g_5(x) &= -8x_2 + x_{11} \leq 0 \\ g_6(x) &= -8x_3 + x_{12} \leq 0 \\ g_7(x) &= -2x_4 - x_5 + x_{10} \leq 0 \\ g_8(x) &= -2x_6 - x_7 + x_{11} \leq 0 \\ g_9(x) &= -2x_8 - x_9 + x_{12} \leq 0 \end{aligned}$$

G01 probleminde 13 karar değişkeni ve 9 kısıt fonksiyonu tanımlanmıştır. Global minimum -15 ve optimum değere ait parametreler (1,1,1,1,1,1,1,1,3,3,3,1) şeklindedir. Arama uzayı $0 \leq x_i \leq 1, i = 1,2,3,4,5,6,7,8,9,13$ ve $0 \leq x_i \leq 100, i = 10,11,12$ şeklinde tanımlanmıştır.

G02 Problemi

$$\begin{aligned} \max f(x) &= \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right| \\ g_1(x) &= 0.75 - \prod_{i=1}^n x_i \leq 0 \\ g_2(x) &= \sum_{i=1}^n x_i - 7.5n \leq 0 \end{aligned}$$

G02 probleminde 20 karar değişkeni ve 2 kısıt fonksiyonu tanımlanmıştır. Global maksimum bilinmemekle beraber şimdiye kadarki en iyi çözüm 0.803619 olarak rapor edilmiştir. g_1 kısıtı ($g_1 = -10^{-8}$) değerine yakın olduğu zaman aktif olmaktadır. Arama uzayı $0 \leq x_i \leq 10$ aralığındadır.

G03 Problemi

$$\begin{aligned} \max f(x) &= (\sqrt{n}^n) \prod_{i=1}^n x_i \\ h_1(x) &= \sum_{i=1}^n x_i^2 - 1 = 0 \end{aligned}$$

G03 probleminde 10 karar değişkeni ve 1 kısıt fonksiyonu tanımlanmıştır. Global maksimum 1 değeri olup, optimum karar değişkenleri $x_i^* = \frac{1}{\sqrt{n}}$ ($i=1, \dots, n$) şeklindedir. Arama uzayı $0 \leq x_i \leq 1$ aralığındadır.

G04 Problemi

$$\begin{aligned} \min f(x) &= 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\ g_1(x) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \\ g_2(x) &= -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\ g_3(x) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\ g_4(x) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\ g_5(x) &= 9.300961 + 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 - 25 \leq 0 \\ g_6(x) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0 \end{aligned}$$

G04 probleminde 5 karar değişkeni ve 6 kısıt fonksiyonu tanımlanmıştır. Global minimum değeri -30665.539 olup, optimum karar değişkenleri (78, 33, 29.995256025682, 45 ve 36.775812905788) şeklindedir. Arama uzayı $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_3 \leq 45$, $27 \leq x_4 \leq 45$, $27 \leq x_5 \leq 45$ aralığındadır.

G05 Problemi

$$\begin{aligned} \min f(x) &= 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_3^2 \\ g_1(x) &= -x_4 + x_3 - 0.55 \leq 0 \\ g_2(x) &= -x_3 + x_4 - 0.55 \leq 0 \\ h_3(x) &= 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ h_4(x) &= 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ h_5(x) &= 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0 \end{aligned}$$

G05 probleminde 4 karar değişkeni ve 5 kısıt fonksiyonu tanımlanmıştır. Bilinen en iyi sonuç 5126.4981 olup, optimum karar değişkenleri (679.9453, 1026.067, 0.1188764, -0.3962336) şeklindedir. Arama uzayı $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, $-0.55 \leq x_4 \leq 0.55$ aralığındadır.

G06 Problemi

$$\begin{aligned} \min f(x) &= (x_1 - 10)^3 + (x_2 - 20)^3 \\ g_1(x) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\ g_2(x) &= -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \leq 0 \end{aligned}$$

G06 probleminde 2 karar değişkeni ve 2 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç -6961.81388 olup, optimum karar değişkenleri (14.095, 0.84296) şeklindedir. Arama uzayı $13 \leq x_1 \leq 100$, $0 \leq x_2 \leq 100$ aralığındadır.

G07 Problemi

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ &\quad + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\ g_1(x) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\ g_2(x) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \end{aligned}$$

$$\begin{aligned}
g_4(x) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\
g_5(x) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
g_6(x) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\
g_7(x) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\
g_8(x) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0
\end{aligned}$$

G07 probleminde 10 karar değişkeni ve 8 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 24.3062091 olup, optimum karar değişkenleri (2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, ve 8.375927) şeklindedir. Arama uzayı $-10 \leq x_i \leq 10$, ($i = 1, \dots, 10$) aralığındadır.

G08 Problemi

$$\begin{aligned}
\min f(x) &= \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \\
g_1(x) &= x_1^2 - x_2 + 1 \leq 0 \\
g_2(x) &= 1 - x_1 + (x_2 - 4)^2 \leq 0
\end{aligned}$$

G08 probleminde 2 karar değişkeni ve 2 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 0.095825 olup, optimum karar değişkenleri (1.2279713, 4.2453733) şeklindedir. Arama uzayı $0 \leq x_i \leq 10$, ($i = 1, 2$) aralığındadır.

G09 Problemi

$$\begin{aligned}
\min f(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 \\
&\quad - 4x_6x_7 - 10x_6 - 8x_7 \\
g_1(x) &= -127 + 2x_1^2 + 3x_2^2 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\
g_2(x) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\
g_3(x) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
g_4(x) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0
\end{aligned}$$

G09 probleminde 7 karar değişkeni ve 4 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 680.6300573 olup, optimum karar değişkenleri (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, ve 1.594227) şeklindedir. Arama uzayı $-10 \leq x_i \leq 10$, ($i = 1, \dots, 7$) aralığındadır.

G10 Problemi

$$\begin{aligned}
\min f(x) &= x_1 + x_2 + x_3 \\
g_1(x) &= -1 + 0.0025(x_4 + x_6) \leq 0 \\
g_2(x) &= -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\
g_3(x) &= -1 + 0.01(x_8 - x_5) \leq 0 \\
g_4(x) &= -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0 \\
g_5(x) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\
g_6(x) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0
\end{aligned}$$

G10 probleminde 8 karar değişkeni ve 6 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 7049.3307 olup, optimum karar değişkenleri (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, ve 395.5979) şeklindedir. Arama

uzayı $100 \leq x_1 \leq 10000, 1000 \leq x_i \leq 10000 (i = 2,3), 10 \leq x_i \leq 1000 (i = 4, \dots, 8)$ aralığındadır.

G11 Problemi

$$\min f(x) = x_1^2 + (x_2 - 1)^2$$

$$h(x) = x_2 - x_1^2 = 0$$

G11 probleminde 2 karar değişkeni ve 1 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 0.75 olup, optimum karar değişkenleri $(\mp \frac{1}{\sqrt{2}}, \frac{1}{2})$ şeklindedir. Arama uzayı $-1 \leq x_i \leq 1, (i = 1,2)$ aralığındadır.

G12 Problemi

$$\max f(x) = \frac{(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)}{100}$$

$$g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$$

G12 probleminde 3 karar değişkeni ve 1 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 1 olup, optimum karar değişkenleri (5,5,5) şeklindedir. Arama uzayı $0 \leq x_i \leq 10, (i = 1,2,3)$ aralığındadır ve p, q, r = 1, 2, ..., 9 şeklindedir.

G13 Problemi

$$\min f(x) = e^{x_1 x_2 x_3 x_4 x_5}$$

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$h_2(x) = x_2 x_3 - 5 x_4 x_5 = 0$$

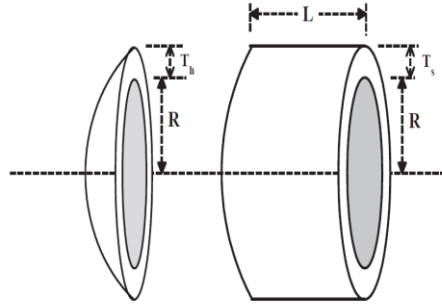
$$h_3(x) = x_1^3 + x_2^3 + 1 = 0$$

G13 probleminde 5 karar değişkeni ve 3 kısıt fonksiyonu tanımlanmıştır. Optimum sonuç 0.0539498 1 olup, optimum karar değişkenleri (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645) şeklindedir. Arama uzayı $-2.3 \leq x_i \leq 2.3 (i = 1,2), -3.2 \leq x_i \leq 3.2 (i = 3,4,5)$ şeklindedir.

EK-2 CTSA için kullanılan sürekli kısıtlı mühendislik tasarım problemleri

Basınçlı Tank Tasarımı Problemi Versiyon 1 (Pressure Vessel Design Problem Version 1, PVDP-V1)

Bu problemin dört tasarım değişkeni vardır. Basınçlı tankın kalınlığı (thickness of the pressure vessel), denklemlerde x_1 şeklinde T_s ; başlığın kalınlığı (thickness of the head) denklemlerde x_2 şeklinde T_h , tankın iç çapı (the inner radius of the vessel) denklemlerde x_3 şeklinde R ve tankın başlıksız uzunluğu (the length of the vessel without heads) denklemlerde x_4 şeklinde L şeklinde temsil edilmektedir. Örnek bir basınçlı tank tasarım problemi Şekil EK-2.1’de verilmiştir.



Şekil EK-2.1. Basınçlı tank tasarım problemi

Problemin matematiksel modeli kısaca aşağıdaki şekildedir:

$$\min f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

Arama uzayı alt ve üst sınırları : $0.0625 \leq x_1, x_2 \leq 6.1875$; $10 \leq x_3, x_4 \leq 200$

Basınçlı Tank Tasarımı Problemi Versiyon 2 (Pressure Vessel Design Problem Version 2, PVDP-V2)

Version 2, PVDP-V2)

Bu problemin PVDP-V1’e göre üç temel farkı vardır. Birincisi 3.1661 olan katsayı 3.1611 olarak alınmıştır. Böylece amaç fonksiyonu aşağıdaki şekli almıştır:

$$\min f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2x_3$$

İkinci fark arama uzayının sınırlarıdır. Arama uzayının alt ve üst sınırları aşağıdaki şekilde kullanılmıştır:

$$1.125 \leq x_1 \leq 12.5 ; 0.625 \leq x_2 \leq 12.5 ; 0 \leq x_3 \leq 240 ; 0 \leq x_4 \leq 240$$

Üçüncü fark ise x_1 ve x_2 tasarım değişkenlerini etkileyecek iki yeni kısıt aşağıdaki eklenmiştir:

$$g_5(X) = 1.1 - x_1 \leq 0$$

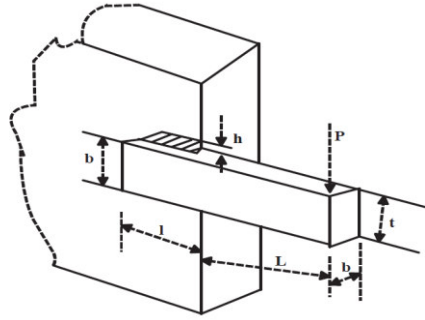
$$g_6(X) = 0.6 - x_2 \leq 0$$

Kaynaklı Kiriş Tasarım Problemi Versiyon 1 (Welded Beam Design Problem Version 1, WBDP-V1)

Version 1, WBDP-V1)

Kaynaklı kiriş tasarım problemi dört tasarım değişkenine sahiptir. Bunlar kaynak kalınlığı (thickness of the weld, h), kaynaklı kısmın uzunluğu (length of the welded joint, l), kirişin genişliği (width of the beam, t) ve kirişin kalınlığıdır (thickness of the beam,

b). Problemin şematik gösterimi Şekil EK-2.2’de ve matematiksel modeli aşağıda verilmiştir.



Şekil EK-2.2. Kaynaklı kiriş tasarım problemi

Problemin matematiksel modelindeki, x_1 , x_2 , x_3 ve x_4 değişkenleri şekil üzerindeki h , l , t ve b değişkenlerine karşılık gelmektedir.

$$\min f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$g_1(X) = \tau(X) - \tau_{max} \leq 0$$

$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0$$

$$g_3(X) = x_1 - x_4 \leq 0$$

$$g_4(X) = 0.125 - x_1 \leq 0$$

$$g_5(X) = \delta(X) - 0.25 \leq 0$$

$$g_6(X) = P - P_c(X) \leq 0$$

$$0.1 \leq x_1 \leq 2; 0.1 \leq x_2 \leq 10; 0.1 \leq x_3 \leq 10; 0.1 \leq x_4 \leq 2$$

Burada, τ kaynağtaki kayma gerilmesini (the shear stress in the weld), τ_{max} kaynağtaki maksimum kayma gerilmesini (the allowable shear stress of the weld) (13600 psi), σ kirişteki normal gerilmeyi (the normal stress in the beam), σ_{max} kiriş malzemesi için izin verilen normal gerilmeyi (the allowable normal stress for the beam material) (30000 psi), P_c çubuğun burkulma yükünü (the bar buckling load), P yükü (the load) (6000lb), ve δ kiriş ucu sapmasını (the beam end deflection) temsil etmektedir.

Kayma gerilmesi (τ) birincil gerilme (τ_1) ve ikincil gerilme (τ_2) olarak adlandırılan iki parçadan oluşur:

$$\tau(X) = \frac{\sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2}}{P}$$

$$\tau_1 = \frac{MR}{\sqrt{2}x_1x_2}$$

$$\tau_2 = \frac{MR}{J}$$

Burada,

$$M = P\left(L + \frac{x_2}{2}\right)$$

$$J(X) = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

M ve J sırasıyla moment ve polar atalet momenti olarak bilinirken, modele ilişkin diğer terimler aşağıdaki gibidir:

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$\sigma(X) = \frac{6PL}{x_4x_3^2}$$

$$\delta(X) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(X) = \frac{4.013\sqrt{\frac{EGx_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right);$$

$$G = 12 \times 10^6 \text{psi}, E = 30 \times 10^6 \text{psi}, P = 6000 \text{lb}, L = 14 \text{in}$$

Kaynaklı Kiriş Tasarım Problemi Versiyon 2 (Welded Beam Design Problem)

Version 2, WBDP-V2)

Bu problemin WBDP-V2'den farklı olan kısımları aşağıda izah edilmiştir. Bu versiyonda sapma (deflection) $\delta(X)$, burkulma yükü (buckling load) $P_c(X)$ ve polar atalet momenti (polar moment of inertia) $J(X)$ denklemleri değişmiş ve aşağıda verilen $g_7(X)$ kısıtı eklenmiştir:

$$g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0$$

$$\delta(X) = \frac{6PL^3}{Ex_3^3x_4}$$

$$P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$J(X) = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\}$$

Matematiksel modelin geri kalanı WBDP-V1 ile aynıdır.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ahmet Cevahir ÇINAR
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Çumra, 07.10.1986
Telefon : +90.506.229.73.39
Faks :
E-Posta : ahmetcevahircinar@gmail.com
accinar@selcuk.edu.tr

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: A. Meteoroloji Mes. Lis., Keçiören, Ankara			2004
Üniversite	: Selçuk Üni., Bilgisayar Müh., Selçuklu, Konya			2009
Yüksek Lisans	: Selçuk Üni., Bilgisayar Müh. ABD, Selçuklu, Konya			2016
Doktora	: Konya Teknik Üni., Bilgisayar Müh. ABD, Selçuklu, Konya			-

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2005-2020	Meteoroloji 8.Bölge Müdürlüğü	Meteorolojist
2020-	Selçuk Üniversitesi Teknoloji Fakültesi	Öğretim Görevlisi

UZMANLIK ALANI

Sürü zekası, evrimsel hesaplama, paralel programlama, yapay zeka

YABANCI DİLLER

İngilizce

YAYINLAR

MAKALELER

[1] Cinar, A. C., Korkmaz, S., & Kiran, M. S. (2019). A discrete tree-seed algorithm for solving symmetric traveling salesman problem. Engineering Science and Technology, an International Journal. (SCI-Expanded) (Doktora tezinden üretilmiştir.)

[2] Gungor, I., Emiroglu, B. G., Cinar, A. C., & Kiran, M. S. (2019). Integration search strategies in tree seed algorithm for high dimensional function optimization. International Journal of Machine Learning and Cybernetics, 1-19. (SCI-Expanded)

[3] Cinar A.C., Kiran M.S., “Similarity and logic gate-based tree-seed algorithms for binary optimization”, Computers & Industrial Engineering 115, 631-646, (SCI-Expanded) 2018. (Doktora tezinden üretilmiştir.)

[4] Babalik A., **Cinar A.C.**, Kiran M.S., “A modification of tree-seed algorithm using Deb’s rules for constrained optimization”, *Applied Soft Computing* 63, 289-305, (SCI-Expanded), 2018. (Doktora tezinden üretilmiştir.)

[5] **Cinar A.C.**, Kiran M.S., “A parallel implementation of Tree-Seed Algorithm on CUDA-supported graphical processing unit”, *Journal of the Faculty of Engineering and Architecture of Gazi University* 33(4), 1397-1409, (SCI-Expanded), 2018. (Yüksek lisans tezinden üretilmiştir.)

[6] Sahman, M. A., & **Cinar, A. C.** (2019). Binary tree-seed algorithms with S-shaped and V-shaped transfer functions. *International Journal of Intelligent Systems and Applications in Engineering*, 7(2), 111-117.

[7] Yasar, A., Saritas, I., Sahman, M. A., & **Cinar, A. C.** (2019). Classification of Parkinson disease data with artificial neural networks. In *IOP Conference Series: Materials Science and Engineering* (Vol. 675, No. 1, p. 012031). IOP Publishing.

[8] Sahman, M. A., **Cinar, A. C.**, Saritas, I., & Yasar, A. (2019). Tree-seed algorithm in solving real-life optimization problems. In *IOP Conference Series: Materials Science and Engineering* (Vol. 675, No. 1, p. 012030). IOP Publishing.

SEMPOZYUM ve KONFERANSLAR

[1] **Cinar A.C.**, Kiran M.S., “A parallel version of Tree-Seed Algorithm within CUDA platform”, *International Scientific Conference on Applied Sciences*, September 2016, Antalya, Turkey, 174-178.

[2] **Cinar A.C.**, Iscan, H., Kiran M.S., “Tree-Seed algorithm for large-scale binary optimization”, *The 9th International Conference on Advances in Information Technology (IAIT2017)*, November 2017, Bangkok, Thailand.

[3] **Cinar A.C.**, Kiran M.S., “Boundary Conditions in Tree-Seed Algorithm”, *2ND INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND ENGINEERING*, October 2017, Antalya, Turkey.

[4] Korkmaz S., **Cinar A.C.**, Seyfi G., Kiran M.S., “A DISCRETE VARIANT OF TREE SEED ALGORITHM”, *International Conference on Engineering Technologies ICENTE’17*, December 2017, Konya, Turkey.

[5] Korkmaz S., **Cinar A.C.**, Kiran M.S., “AN ANALYSIS OF POPULATION SIZE FOR BINARY ARTIFICIAL ALGAE ALGORITHMS”, *7th International*

Conference on Advanced Technologies (ICAT'18), Antalya, Turkey, 28 April – 1 May, 2018.

[6] **Cinar A.C.**, Korkmaz S., Kiran M.S., “A BINARY HYBRID APPROACH BASED ON TREE SEED ALGORITHM FOR SOLVING 0-1 KNAPSACK PROBLEMS”, 7th International Conference on Advanced Technologies (ICAT'18), Antalya, Turkey, 28 April – 1 May, 2018.

[7] Kaya E, **Cinar A.C.**, Uymaz O., Korkmaz S., Kiran M.S., “The Binary Salp Swarm Algorithm with Using Transfer Functions”, International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018), 11-13 May 2018, Safranbolu, Turkey, 2018.

[8] Tasdemir S., **Cinar A.C.**, “Application of Artificial Neural Network Forecasting of Daily Maximum Temperature in Konya”, 17th International Conference on Soft Computing MENDEL'2011, 236-243, Brno, Czech Republic, Haziran 2011.

BİLİMSEL ARAŞTIRMA PROJELERİ

Selçuk/Konya Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından desteklenen “Ağaç-tohum algoritmasının performansının optimizasyon problemleri üzerinde iyileştirilmesi” projesinde araştırmacı, 2020.