

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

**EN FAZLA KAZANÇ SAĞLAYAN ROTA
BULMA PROBLEMLERİNİN SEZGİSEL
YÖNTEMLERLE İNCELENMESİ**

Aişe Zülal ŞEVKLİ
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2010

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

**EN FAZLA KAZANÇ SAĞLAYAN ROTA
BULMA PROBLEMLERİNİN SEZGİSEL
YÖNTEMLERLE İNCELENMESİ**

Aişe Zülal ŞEVKLİ
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
Yrd. Doç. Dr. Fatih Erdoğan SEVİLGİN

GEBZE
2010

ÖZET

TEZİN BAŞLIĞI: EN FAZLA KAZANÇ SAĞLAYAN ROTA BULMA PROBLEMLERİNİN SEZGİSEL YÖNTEMLERLE İNCELENMESİ

YAZAR ADI : AİŞE ZÜLAL ŞEVKLİ

En Fazla Kazanç Sağlayan Rota Bulma Problemleri, kontrol noktaları ve bu kontrol noktaları arasındaki bağlantıları içeren bir çizge üzerinde çalışır. Problemin amacı, bir başlangıç noktasından başlayıp, bitiş noktasında sonlanacak ve verilen maliyet kısıtını (mesafe, zaman, vb.) aşmayacak biçimde araç sayısı kadar ve toplamda en fazla kazancı sağlayacak rotaları belirlemektir. Literatürde bu problemlerin tek araçlı şekli Oryantiring Problemi (OP), çok araçlı şekli ise Takım Oryantiring Problemi (TOP) olarak adlandırılmıştır.

Bu tez çalışmasında, OP ve TOP çözümü için Değişken Komşu Arama (DKA) ve Parçacık Sürü Optimizasyonu (PSO) tabanlı iki yeni model geliştirilmiştir. Birinci model, DKA'nın bir versiyonu olan İndirgenmiş-DKA(İ-DKA) yöntemini kullanır. Geliştirilen modelde, İ-DKA geliştirilme amacı olan küresel aramanın yanında yerel arama için de kullanılır.

İkinci model ise PSO tabanlıdır. PSO çözüm uzayını, popülasyonu oluşturan parçacıkların sistematik dolaşmasıyla tarar. PSO'nun doğrudan uygulanmasında iyi yerlerde yeterince detaylı arama yapamama ve erken yakınsama gibi zayıflıklarla karşılaşılır. Bu zayıflıklar sırayla, iyi yerler bulan parçacıklar için İ-DKA yerel aramasının çalıştırılması ve sonrasında o parçacıkların aramaya rasgele bir yerden devam ettirilmesi ile giderilmeye çalışılmıştır. Önerilen bu yeni PSO versiyonu, Güçlendirilmiş-PSO (G-PSO) olarak adlandırılmıştır. G-PSO üzerine, çözüme ihtiyaç duyacağı anda ayrık duruma getirilen ve kullanılan operatörler dahil tüm algoritmanın ayrık olarak ifade edildiği, iki yeni Ayrık G-PSO modeli geliştirilmiştir. Geliştirilen bu yeni modeller OP ve TOP için toplam 453 problemde test edilmiştir. Sonuçlar çözüm kalitesi ve performans açısından literatürdeki sonuçlarla karşılaştırıldığında, 12 problem için literatüre yeni sonuçlar kazandırılmış, diğer problemler için ise aynı yada rekabetçi sonuç üretildiği görülmüştür.

SUMMARY

**THESIS TITLE : A STUDY ON HEURISTICS METHODS FOR
THE VEHICLE ROUTING PROBLEMS WITH
PROFIT**

AUTHOR NAME : AİŞE ZÜLAL ŞEVKLİ

The Vehicle Routing Problems with Profit are defined on a graph including control points and connections between them. Objective of the problem is to find paths, starting at an origin and ending at a destination that maximizes total profit without violating prescribed cost function (total distance, time and etc.). A path is determined for each vehicle defined by the problem. If there is only one vehicle in the problem, the problem is named as Orienteering Problem (OP), if there are more than one vehicles, it is named as Team Orienteering Problem (TOP).

In this thesis, two new models based on Particle Swarm Optimization (PSO) and Variable Neighborhood Search (VNS) are proposed for solving OP and TOP. First model employs Reduced VNS (RVNS) which is a version of VNS. Providing a fast but sketchy exploration through the solution space is the strongest asset of RVNS. By using two nested RVNS, the proposed model accomplishes not only global search but also detailed local search at the same time.

The second model is based on PSO. PSO is a population based metaheuristic that takes advantage of individual memory and social cooperation in a swarm. Straightforward application of PSO suffers from premature convergence and lack of intensification around the local best locations. To rectify these problems, a RVNS based local search around the best particle in the swarm is performed and a random moving strategy for this particle is employed. The proposed method is named as Strengthened PSO (StPSO). Partial and full discrete versions of this method are used for OP and TOP. The proposed models are tested against 453 OP and TOP benchmark problems in literature. It is observed that they generate competitive and promising results compared to the other similar methods in literature in terms of solutions quality and performance. Furthermore, improvements are achieved for 12 benchmark problems.

TEŞEKKÜR

Doktora öğrenimim sırasında ve tez çalışmam boyunca gösterdiği her türlü destek, ilgi ve yardımdan dolayı değerli danışman hocam Dr. Fatih Erdoğan Sevilgen’e en içten dileklerimle teşekkür ederim.

Tez izleme komitemde bulunmayı kabul edip çalışmamın ilerlemesinde katkıda bulunan Dr. Yusuf Sinan Akgül ve Dr. İlyas Kandemir’e şükranlarımı sunarım.

Yoğun akademik çalışmaları arasında zaman ayırarak tezimin savunma komitesinde bulunmayı kabul edip çalışmama verdiği desteklerinden dolayı Dr. Ercan Öztemel ve Dr. Haluk Rahmi Topçuoğlu’na teşekkürü borç bilirim.

Bu çalışma boyunca yardımlarını esirgemeyen çalışma arkadaşlarıma ve rahat bir çalışma ortamı sunan GYTE Bilgisayar Mühendisliği Bölümü hocalarıma ve araştırma görevlisi arkadaşlarıma teşekkür ederim.

Çalışmayı gerçekleştirdiğim dönem içerisinde maddi manevi desteklerini her zaman hissettiğim sevgili eşim Mehmet Şevkli’ye ve her zaman yanımda olan ve beni her konuda yüreklendiren, destek olan anne ve babama teşekkürlerimi sunarım.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
SİMGELER VE KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xiii
ÇİZELGELER DİZİNİ	xv
1 GİRİŞ	1
1.1 Problemin Önemi	1
1.2 Sezgisel Yöntemler	2
1.3 Tezin Amacı, Kapsamı ve Katkıları	3
1.4 Tezin Düzeni	6
2 EN FAZLA KAZANÇ SAĞLAYAN ROTA BULMA PROBLEMLERİ	7
2.1 Problem Tanımı	7
2.2 Problemin Matematiksel İfadesi	8
2.3 Mevcut Çözüm Yaklaşımları	10
2.3.1 Kesin Çözümler	10
2.3.2 Sezgisel Çözümler	11
2.3.3 Meta-Sezgisel Çözümler	12
2.4 Karşılaştırmalı Test Problemleri	16
3 META-SEZGİSEL YÖNTEMLER	19
3.1 Değişken Komşu Arama Yöntemi	21
3.1.1 DKA Yönteminin Versiyonları	22
3.2 Parçacık Sürü Optimizasyon Yöntemi	24

3.2.1	PSO Yönteminin Versiyonları	26
4	DEĞİŞKEN KOMŞU ARAMA TABANLI ÖNERİLEN MODELLER	35
4.1	Oryantiring Problemi için Değişken Komşu Arama Modeli	35
4.1.1	Problem Gösterimi	35
4.1.2	Komşuluk Yapıları	36
4.1.3	OP için İççe İndirgenmiş-DKA Modeli ve İşleyişi	38
4.1.4	İİ-DKA İçin Yan Modeller	40
4.2	Takım Oryantiring Problemi için Değişken Komşu Arama Modeli	41
4.2.1	Problem Gösterimi	41
4.2.2	Komşuluk Yapıları	41
4.2.3	Modelin İşleyişi	43
4.3	Deneysel Sonuçlar	44
4.3.1	OP için İİ-DKA Modelinin Deneysel Sonuçları	45
4.3.2	TOP için İİ-DKA Modelinin Deneysel Sonuçları	53
5	PARÇACIK SÜRÜ OPTİMİZASYONU TABANLI ÖNERİLEN MODELLER	56
5.1	Yarı Ayrık Güçlendirilmiş-PSO Modeli	56
5.1.1	Problem Gösterimi	56
5.1.2	Güçlendirilmiş-PSO	57
5.1.3	G-PSO İçin Yan Modeller	61
5.2	Ayrık Güçlendirilmiş PSO Modeli	61
5.2.1	Problem Gösterimi	61
5.2.2	Ayrık Operatörler	62
5.3	Deneysel Sonuçlar	64
5.3.1	OP için PSO Tabanlı Modellerin Deneysel Sonuçları	64
5.3.2	TOP için PSO Tabanlı Modellerin Deneysel Sonuçları	72

5.3.3	Önerilen DKA ve PSO Tabanlı Modellerin Karşılaştırılması	75
6	SONUÇ VE ÖNERİLER	77
	KAYNAKLAR DİZİNİ	79
	ÖZGEÇMİŞ	85
	EKLER	86

SİMGELER VE KISALTMALAR DİZİNİ

ARP	: Araç Rotalama Problemi
GSP	:Gezgin Satıcı Problemi
DKA	: Değişken Komşu Arama
PSO	: Parçacık Sürü Optimizasyonu
KKO	: Karınca Koloni Optimizasyonu
GA	: Genetik Algoritma
TA	: Tabu Arama
KYA	: Kılavuzlu Yerel Arama
TB	: Tavlama Benzetimi
CGW	: Chao ve arkadaşlarının OP ve TOP için geliştirdiği sezgisel algoritma
PK	: Problem kümesi
İ-DKA	: İndirgenmiş Değişken Komşu Arama
DKYA	: Değişken Komşulu Yerel Arama
İİ-DKA	: İç içe İndirgenmiş Değişken Komşu Arama
İİ-DKA-B	: İç içe İndirgenmiş Değişken Komşu Arama-Basit
OP	: Oryantiring Problemi
TOP	: Takım Oryantiring Problemi
G-PSO	: Güçlendirilmiş Parçacık Sürü Optimizasyonu
EKS-RBP	: En Fazla Kazanç Sağlayan Rota Bulma Problemleri
IwPSO	: Atalet ağırlıklı PSO
ConPSO	: Daralma faktörlü PSO
GCPSO	: Garantili yakınsayan PSO
FIPSO	: Tam bilgilendirilmiş PSO

MPSO	: Mutasyon Operatörlü PSO
GBSET	: Parçacıkların tümüyle birbirine bağlı olduğu topoloji
LBEST	: Parçacığın k tane komşusuyla oluşturduğu topoloji
UPSO	: Birleştirilmiş PSO
CLPSO	: Kapsamlı öğrenen PSO
YAG-PSO	: Yerel Araması Güçlendirilmiş PSO
KAG-PSO	: Küresel Araması Güçlendirilmiş PSO
N	: Kontrol Noktaları Kümesi
E	: Kontrol Noktaları Arasındaki Bağlantı Kümesi
G	: Elemanları (N,E) oluşan bir çizge
n_i	: i . kontrol noktası
d	: Problem boyutu
s_i	: i . kontrol noktasının kazanç miktarı
c_{ij}	: n_i ve n_j kontrol noktaları arasındaki kenar maliyeti
T_{maks}	: Rotanın maliyet kısıtı
m	: Araç sayısı
p_i^t	: Sürü içerisinde t . iterasyondaki i . parçacığın pozisyon vektörü
v_i^t	: t . iterasyondaki i . parçacığın hız vektörü
p_{ij}^t	: t . iterasyondaki i . parçacığın j . pozisyon değeri
v_{ij}^t	: t . iterasyondaki i . parçacığın j . hız değeri
$pbest_i^t$	: i . parçacığın t . iterasyonundaki yerel en iyi komşu vektörü
$gbest_i^t$	: Küresel en iyi pozisyon vektörü
w^t	: t . iterasyondaki atalet ağırlığı
$-V_{max}$	: Hız Vektörünün Alacağı Minimum Değer

V_{max}	: Hız Vektörünün Alacağı Maksimum Değer
$rand_1$	: 0 ile 1 arasında Düzgün Dağılmış Rassal Değer
$rand_2$	: 0 ile 1 arasında Düzgün Dağılmış Rassal Değer
c_1	: Sosyal parametre Değeri
c_2	: Kavramsal Parametre Değeri
χ	: Daralma Faktörü
$dö$	: Düşük öncelik
$yö$	: Yüksek öncelik
$ö-yok$	: Öncelik yok

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1 OP ve TOP için kullanılan karşılaştırmalı problem kümelerinin 2 boyutlu düzlemde gösterimi	18
3.1 DKA algoritmasının benzetimsel kodu	22
3.2 Değişken Komşuluklu Yerel Arama benzetimsel kodu	23
3.3 Esnek-DKA , DKA'dan farklı olan kod kısmı	24
3.4 PSO algoritmasının benzetimsel kodu	25
3.5 Komşuluk topojileri	30
3.6 $pbestf_i$ vektörünü oluşturan prosedür	31
3.7 Model 1'e göre pozisyon ve hız güncelleme örneği	33
4.1 OP için İİ-DKA modelinin benzetimsel kodu	38
4.2 İİ-DKA modelinin İ-DKA Yerel Arama benzetimsel kodu	39
4.3 TOP için İççe-İndirgenmiş-DKA benzetimsel kodu	43
4.4 TOP için İİ-DKA modelinin İ-DKA Yerel Arama benzetimsel kodu	44
4.5 PK 5'te $T_{maks}=20$ için İİ-DKA ile elde edilen yeni çözüm	50
4.6 PK 5'te $T_{maks}=95$ için İİ-DKA ile elde edilen yeni çözüm	50
4.7 PK 5'te $T_{maks}=100$ için İİ-DKA ile elde edilen yeni çözüm	51
4.8 PK 5'te $T_{maks}=105$ için İİ-DKA ile elde edilen yeni çözüm	51
4.9 PK 5'te $T_{maks}=125$ için İİ-DKA ile elde edilen yeni çözüm	52
4.10 PK 6'da $T_{maks}=35$ için İİ-DKA ile elde edilen yeni çözüm	52
4.11 PK 6'da $T_{maks}=80$ için İİ-DKA ile elde edilen yeni çözüm	53
5.1 En küçük pozisyon değeri kuralı için bir örnek	57
5.2 Güçlendirilmiş-PSO modelinin benzetimsel kodu	58
5.3 OP için İ-DKA Yerel Araması Benzetimsel Kodu	59
5.4 Ayırık G-PSO'da hız güncelleme örneği	64
5.5 PK 5'te $T_{maks}=95$ için Ayırık G-PSO ile elde edilen yeni çözüm	70

5.6	PK 5'te $T_{maks}=100$ için Ayrık G-PSO ile elde edilen yeni çözüm	70
5.7	PK 5'te $T_{maks}=105$ için Ayrık G-PSO ile elde edilen yeni çözüm	71
5.8	PK 6'da $T_{maks}=80$ için Ayrık G-PSO ile elde edilen yeni çözüm	71

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
2.1 OP ve TOP için kesin çözümlü yöntemler	10
2.2 OP ve TOP için sezgisel yöntemler	11
2.3 OP ve TOP için kullanılan meta-sezgisel yöntemler	12
2.4 OP için kullanılan problem kümeleri ve özellikleri	16
2.5 TOP için kullanılan problem kümeleri ve özellikleri	17
4.1 OP için DKA modelleri ve kullanılan komşuluk yapıları	41
4.2 OP için geliştirilen DKA tabanlı modellerin PK (1,2,3 ve 1_1) için CPU(saniye), yüzdelik ve ortalama yüzdelik bağıl hata kriterlerine göre karşılaştırılması.	46
4.3 Önerilen İİ- DKA modelinin PK 1,2,3 ve 1_1 için literatür sonuçlarıyla karşılaştırılması	47
4.4 Önerilen İİ-DKA modelinin PK 5 için literatür sonuçlarıyla karşılaştırılması	48
4.5 Önerilen İİ-DKA tabanlı modelinin PK 6 için literatür sonuçlarıyla karşılaştırılması	48
4.6 OP için İİ-DKA modelinin geliştirdiği yeni sonuçlar	49
4.7 İİ-DKA modelinin toplam 7 problem kümesi için performans analizi	54
4.8 İİ-DKA modelinin performansının önceki çalışmalarla karşılaştırılması	55
5.1 PK 1,2,3 ve 1_1 için PSO tabanlı modellerin CPU(saniye), Yüzdelik ve Ortalama Yüzdelik Bağıl Hata kriterlerine göre karşılaştırılması	66
5.2 PSO ve DKA tabanlı modellerin PK 1,2,3 ve 1_1 için literatür sonuçlarıyla karşılaştırılması	67
5.3 Ayırık G-PSO modelinin PK 5 için literatür sonuçlarıyla karşılaştırılması	68
5.4 Ayırık G-PSO modelinin PK 6 için literatür sonuçlarıyla karşılaştırılması	68
5.5 OP için Ayırık G-PSO modelinin geliştirdiği yeni sonuçlar	69
5.6 TOP için geliştirilen Ayırık PSO tabanlı modellerin CPU(saniye), Yüzdelik Bağıl Hata ve Standart Sapma kriterlerine göre karşılaştırılması	72
5.7 Ayırık G-PSO modelinin toplam 7 problem kümesi için performans analizi	73
5.8 Ayırık G-PSO performanslarının önceki çalışmalarla karşılaştırılması	75

1 GİRİŞ

Bu bölümde ulaşım/dağıtım problemlerinden biri olan En Fazla Kazanç Sağlayan Rota Bulma problemlerinin kısaca tanımı, önemi ve ulaşım ve dağıtım sektöründeki yeri anlatılacaktır. Bu problemlere genel çözüm yaklaşımlarının neler olduğundan bahsedilip, tezin amacı, kapsamı ve katkıları ortaya konulacaktır.

1.1 Problemin Önemi

Küresel piyasada yoğun rekabet, hızlı ürün tüketimi ve müşterilerin artan beklentileri, üreticileri ulaşım ve dağıtım sistemlerine önem vermeye ve yatırım yapmaya zorlamıştır. Dağıtım sistemlerinde yapılacak olan iyileştirmeler, firmanın lojistik maliyetlerini önemli ölçüde azaltır. Amerika Birleşik Devletleri'ndeki lojistik maliyetler hakkında 2005 yılında yapılan bir araştırmada [Rushton et al., 2006] ulaşım maliyetlerinin % 45 pay ile en önemli unsur olduğu, bunu % 23 ile stok bulundurma maliyetlerinin, % 22 ile depolama ve % 10 ile yönetimin takip ettiği ortaya konulmuştur. Ulaşım maliyetlerini azaltma yönünde yapılan çalışmalar insan ve malzeme kaynaklarının geniş kullanımını da sağlayacağından ulusal ekonomiyi de etkiler.

Araç Rotalama Problemi (ARP) ve Gezgin Satıcı Problemi (GSP) dağıtım/ulaşım alanında en çok çalışılan optimizasyon problemlerindendir. GSP, gezgin satıcının kendi şehriden başlayıp, diğer tüm şehirleri ziyaret ederek, tekrar kendi şehrine dönmesi için gereken en kısa yolu bulma problemidir. ARP ise, başlangıç noktasından gezmeye başlayan birden fazla aracın her birinin farklı müşterilerin talebini karşılayarak başlangıç noktasına geri dönecekleri en az maliyetli rotaları tespit etme problemidir. NP-zor sınıfına ait olmasına rağmen kolay formüle edilmesi ve çok sayıda uygulama alanı olması, bu iki problemin de popüleritesinin artmasını sağlamıştır. Literatürde araç rotalama ve gezgin satıcı problemlerinin çok sayıda çeşitleri ve genellemeleri vardır.

En Fazla Kazanç Sağlayan Rota Bulma Problemleri adı altında topladığımız GSP tabanlı Oryantiring Problemi ve ARP tabanlı Takım Oryantiring Problemi de bu çeşitler arasındadır. Problemlerin üretildiği kaynak problemlerden de anlaşılacağı

üzere, Oryantiring tek araçlı, Takım Oryantiring ise çok araçlı problemlerdir. Bu problemler kısıtlı dolaşma maliyetine (zamanına) sahip olmalarından dolayı müşterilerden gelen tüm talepler karşılanamaz. Kısıtı aşmayacak şekilde en yüksek getiriyi sağlayacak (memnuniyet, kazanç) müşteriler seçilir. Kısıtlı dolaşma maliyeti göz önüne alındığında ziyaret edilecek müşterileri ve ziyaret sırasını iyi belirlemek daha fazla müşteriye hizmet götürmeyi, bunun sonucu olarak daha fazla kazanç elde etmeyi sağlayacaktır.

Problemin adı harita ve pusula yardımıyla açık alanda oynanan Oryantiring sporundan gelmektedir. Oyun arazisinin belli noktalarına puanlı hedefler konur ve haritada bu hedefler işaretlenir. Oyunun amacı belli bir oyun süresinde haritada bulunan hedefler içerisinde en fazla puanı toplamaktır. Bu spor insana, kavrama, düşünme, karar verme, konsantrasyon, harita okuma gibi bedensel ve zihinsel anlamda bir çok katkı sağlandığından dolayı uluslararası alanda hızla yaygınlaşmakta, il il ve ülke ülke müsabakalar düzenlenmektedir. Türkiye’de de bu sporu Türkiye Oryantiring Federasyonu (www.oryantiring.org) organize etmektedir. Problemin endüstriyel alanda da müşteri hizmet servisi, turist rehberliği, gaz dağıtımı gibi bir çok alanda uygulamaları vardır [Balas, 1989], [Kantor and Rosenwein, 1992], [Vansteenwegen and Oudheusden, 2007].

1.2 Sezgisel Yöntemler

Karar değişkenleri kullanılarak tanımlanan bir problemde, değişkenlerin uygun değerler alarak ulaşabileceği en iyi çözümü arayan problem tiplerine optimizasyon problemleri denir. Bu problemleri çözmek için pek çok çeşitli algoritma geliştirilmiştir. Bu algoritmaları kesin çözümü veren ve yaklaşık çözümü veren algoritmalar olarak sınıflandırabiliriz. Kesin çözümü veren algoritmalar en iyi çözümü bulmayı garanti ederler. NP-zor sınıfında bulunan optimizasyon problemleri için bilinen tüm kesin çözümlü algoritmalar problem büyüklüğü ile üssel olarak artan bir zamana ihtiyaç duyarlar. Bu ise elde edilmek istenen sonuca çok zaman harcayarak ulaşmak demektir. Bundan dolayı son 30 senede optimizasyon problemlerin çözümünde çok daha kısa sürede yaklaşık çözüm veren algoritmalar daha çok rağbet görmektedir [Blum and Roli, 2003].

Yaklaşık çözümlü algoritmalar, tüm çözümlerin değerlendirilemeyeceği kadar büyük çözüm kümeli problemlerin sadece “bazı çözümlerini” değerlendirerek en iyiye yakın çözümleri makul bir sürede bulurlar. Çözüm kümesinin hangi “bazı çözümlere” ulaşacağını deneme-yanılmaya dayalı tecrübelerle eğitilmiş bir gezme stratejisi belirleyerek öngören yöntemlere genel olarak “sezgisel yöntemler” denir. En basit sezgisel yöntemler *yapıcı* (constructive) metodlar ve *yerel arama* metodlarıdır. Yapıcı metodlar çözümü, boş bir çözümünden başlayarak ve belli bir amaç (genellikle hırslı) gözeterek çözümün bileşenlerini teker teker eklemekle oluşturur. Yerel arama ise başlangıç bir çözümünden başlayarak ve tekrarlı olarak, çözümün belirli bir komşuluğunda daha iyi çözümü arama metodudur.

Son 20 yılda, optimizasyon problemlerinin çözümünde basit sezgisel metodların birleştirilerek daha soyut, üst seviyeden çalışma yöntemi sunan *meta-sezgisel* yöntemler kullanılmaktadır [Blum and Roli, 2003]. Günümüze kadar bir çok meta-sezgisel yöntem geliştirilmiştir. Jones ve arkadaşlarının yaptığı çalışmada 1990-2000 yılları arasında problemlere %70 oranında Genetik Algoritma, %24 oranında Tavlama Benzetimi ve %6 oranında Tabu Arama metasezgisel yöntemleri kullanılarak çözümlerin araştırıldığı ortaya konulmuştur [Jones et al., 2002]. Bununla beraber son 10 yılda Parçacık Sürü Optimizasyonu ve Değişken Komşu Arama yöntemleri de elde edilen etkin sonuçlarla optimizasyon problemlerinin çözümünde literatürdeki yerini almıştır.

1.3 Tezin Amacı, Kapsamı ve Katkıları

Bu tezin birinci amacı, Oryantiring ve Takım Oryantirig Problemleri için daha önce denenmemiş Değişken Komşu Arama (DKA) ve Parçacık Sürü Optimizasyon (PSO) meta-sezgisel yöntemlerini kullanan modeller geliştirmek ve yöntemlerin bu problemlere uygulanabilirliğini göstermektir. İkinci amaç ise, bu iki meta sezgisel yöntemin güçlü yönlerini ön plana çıkartan, zayıf yanlarını kapatıcı ve iyileştirici problemenden bağımsız mekanizmalar geliştirmektir.

Önerilen modellerde kullanılan DKA yöntemi tanımlanan komşuluk alanlarını sistematik olarak gezerek çözümü bulmaya çalışan çok genel bir meta-sezgisel yöntemdir. Komşuluk yapıları ve komşuluk sırası bu yöntemin performansını

etkileyen en önemli faktörlerdir. PSO ise kuş ve balık sürülerinin iki boyutlu hareketlerinin modellenmesiyle oluşturulmuştur. PSO algoritmasında, sürüyü oluşturan parçacıklar çok boyutlu çözüm kümesinde dolaşarak en iyi çözümü bulmaya çalışırlar. Sürüdeki her bir parçacığın uçuş yönü ise kendi tecrübesi, sürünün tecrübesi ve o parçacığın o anki yönelimi kullanılarak hesaplanır ve böylece yeni çözümlere ulaşılır.

Tezin amacı doğrultusunda, Oryantiring ve Takım Oryantiring Problemi için DKA tabanlı İççe İndirgenmiş DKA modeli (İİ-DKA) geliştirilmiştir. Bu model DKA'nın bir versiyonu olan İndirgenmiş DKA (İ-DKA) yöntemini kullanır. İ-DKA yöntemi genellikle büyük çözüm uzaylı problemlerde çözüm uzayını büyük adımlarla hızlı bir şekilde taramak için, bir başka ifade ile küresel arama yapmak için kullanılır [Mladenovic and Hansen, 1997] Önerilen İİ-DKA modeliyle İ-DKA'nın, küresel arama yöntemi yanında, doğru komşuluklar seçimi ile yerel arama yöntemi olarak da kullanılabileceği gösterilmiştir.

Oryantiring ve Takım Oryantiring Problemi için geliştirilen ikinci model ise PSO tabanlıdır. PSO çözüm uzayını, popülasyonu oluşturan parçacıkların sistematik dolaşmasıyla tarar. PSO'nun karmaşık yapılı çözüm uzayına sahip problemlerde iyi yerlerde yeterince detaylı arama yapamama ve erken yakınsama gibi sorunları ortaya çıkmıştır. Önerilen PSO modelinde öncelikle bu sorunların giderilmesi amaçlanmıştır.

Bu amacı gerçekleştirmek için PSO algoritmasında bir takım değişiklikler yapılmıştır. Değişikliklerin odak noktası sürü içinde iyi yerleri bulan öncü-parçacıklardır. Bu parçacıklar için İİ-DKA modeli için geliştirilen İ-DKA yerel araması çalıştırılır. Böylece öncü parçacığın yakın çevresinde alabileceği en iyi değer tespit edilmiş olur. Sonrasında ise, öncü-parçacık çözüm uzayında rasgele bir yere yönlendirilerek erken yakınsama geciktirilmiş yada engellenmiş olur. Böylece PSO'nun hem yerel alanda arama kabiliyeti artırılmış, hem de erken yakınsama problemi giderilerek alan tarama kabiliyeti artırılmıştır. Önerilen PSO'nun bu yeni versiyonuna bundan dolayı Güçlendirilmiş-PSO (G-PSO) adı verilmiştir.

PSO, sürekli optimizasyon problemlerinin çözümü için geliştirildiğinden, G-PSO modeli ayırık problem olan OP ve TOP'a doğrudan uygulanamamıştır. Bundan dolayı algoritma akışı olarak G-PSO'yu takip eden yarı ayırık ve ayırık iki G-PSO modeli geliştirilmiştir. Yarı ayırık G-PSO modelinde sürekli değerlerden oluşan PSO çözüm bileşeni geçici olarak permütasyona çevrilerek ayırık hale getirilir. Problem çözümü elde edildikten sonra PSO bileşenleri sürekli değerle devam ederler. Ayırık G-PSO modelinde ise PSO bileşenleri ve bu bileşenlerle kullanılan tüm operatörler ayırık işlem yapabilecek şekilde tanımlanmıştır.

DKA ve PSO tabanlı geliştirilen modeller Oryantiring ve Takım Oryantiring problemleri için probleme uygun komşuluk yapıları geliştirilerek toplamda 453 karşılaştırmalı test problemi ile test edilmiştir. Yapılan deneylerde modellerin hem çözüm kalitesi (bağlı yüzde hata, ortalama bağlı yüzde hata, standart sapma, en iyi, en kötü, ortalama) hem de performans (CPU zamanı) açısından literatürdeki sonuçlarla rekabetçi sonuçlar üretildiği görülmüştür. Toplamda 12 problem için ise daha iyi sonuçlar literatüre kazandırılmıştır.

Bu tez çalışmasının katkıları şöyle özetlenebilir:

- OP ve TOP için PSO ve DKA tabanlı yeni çözüm modellerini ilk olarak geliştirmek,
- İ-DKA yönteminin küresel arama yöntemi olarak kullanımı yanında, yerel arama yöntemi olarak da kullanılabileceğini göstermek,
- PSO'nun erken yakınsama ve iyi yerlerde yoğunlaşmama zayıflıklarını iyileştirici mekanizmalar geliştirmek,
- Kombinatoriyel optimizasyon problemlerinde kullanılabilecek yeni bir ayırık PSO modeli geliştirmek,
- OP ve TOP için uygun komşuluk yapıları tespit etmektir.

1.4 Tezin Düzeni

Tez çalışması altı bölümden oluşmaktadır. Birinci bölüm giriş bölümü olup, problemin kısaca tanımı, problemin ulaştırma/dağıtım sektöründeki yeri ve önemi ve bu tür problemlere sunulan çözüm yöntemlerinden bahsedilmiştir. Sonrasında ise tez çalışmasının amacı, kapsamı ve katkılarına yer verilmiştir.

İkinci bölümde, En Fazla Kazanç Sağlayan Rota Bulma Problemlerinin tanımı, mevcut çözüm yaklaşımları ve deneylerde kullanılan karşılaştırmalı test problemlerinin özelliklerine detaylı olarak yer verilmiştir.

Üçüncü bölümde meta-sezgisel yöntemlerin özellikleri, çeşitleri hakkında bilgi verilmiştir. Meta-sezgisel yöntemlerden OP ve TOP için önerilen modellerde kullanılan Değişken Komşu Arama ve Parçacık Sürü Optimizasyon yöntemleri, çalışma şekli ve çeşitli versiyonları ile beraber detaylıca açıklanmıştır.

Dördüncü ve beşinci bölümlerde Oryantiring ve Takım Oryantiring Problemleri için önerilen modeller, problem gösterim şekilleri, komşuluk yapıları, modellerin işleyiş biçimi ve deneysel sonuçlar başlıkları altında açıklanmıştır.

Altıncı bölüm sonuç bölümüdür. Bu bölümde elde edilen sonuçlar özetlenmiştir. Bu konu çerçevesinde gelecekte yapılabilecek çalışmalar için önerilerde bulunulmuştur.

2 EN FAZLA KAZANÇ SAĞLAYAN ROTA BULMA PROBLEMLERİ

2.1 Problem Tanımı

En Fazla Kazanç Sağlayan Rota Bulma Problemleri (EKS-RBP) iki ana problemden türetilmiştir: Gezgin Satıcı Problemi-GSP (*Travelling Salesman Problem*) ve Araç Rotalama Problemi-ARP (*Vehicle Routing Problem*). Bilindiği üzere GSP’de amaç, verilen çizge üzerindeki tüm kontrol noktalarını en az maliyetle bir defada dolaşacak rotayı bulmaktır. ARP’de ise çizge üzerindeki bir yada birden fazla kontrol noktası, diğer kontrol noktalarına dağıtım yapacak araçların başlangıç noktası yani deposu olarak kabul edilir. ARP’nin amacı, araçların kapasitelerini göz önünde bulundurularak tüm kontrol noktalarının taleplerini karşılayacak en az maliyetli rotaları bulmaktır.

Problemlere daha gerçekçi yaklaşıldığında günlük hayatta karşılaşılan bazı kısıtlardan dolayı çizge üzerindeki tüm kontrol noktalarını gezmenin her zaman mümkün olmadığı görülecektir. Bu kısıtlar zaman, kapasite, mesai saatleri, kazanç ya da yol kısıtları olabilir. Çizge üzerindeki kontrol noktalarını kısıtlardan dolayı seçmek zorunda kalındığı bu tip problemler kümesi *Kazançlı-GSP* (TSP with profits) ve *Kazançlı-ARP* (VRP with profits) adları altında incelenmiştir [Feillet et al., 2005]. *Kazançlı-GSP* ve *Kazançlı-ARP* kümesindeki problemler amaç fonksiyonlarına göre üç gruba ayrılırlar [Feillet et al., 2005]:

- a. Sabit bir gezme maliyetini aşmadan en fazla kazanç sağlayan rotayı bulma problemleri
- b. Sabit bir alt limit kazanç elde edebilecek en az maliyetli rotayı bulma problemleri
- c. Yukarıda sayılan her iki amacın amaç fonksiyonunda birleştiği yani, gezme maliyetinden kazancın çıkarılması ile oluşan amaç fonksiyon değerinin en aza ineceği rotayı bulma problemleri

Bu tez kapsamında birinci grupta yer alan EKS-RBP üzerine çalışmalar yapılmıştır. Literatürde tek araçlı EKS-RBP, *Oryantiring Problemi (Orienteering Problem)* olarak adlandırılmıştır. Bu problemin çok araçlı versiyonu ise *Takım Oryantiring Problemi (Team Orienteering Problem)* dir. Bu problem isimini açık alanda oynanan Oryantiring sporundan almıştır. Türkiye’de de bu spor Türkiye Oryantiring Federasyonu (<http://www.oryantiring.org>) tarafından organize edilmektedir.

EKS-RBP genel olarak şu şekilde ifade edilebilir: Kontrol noktalarını içinde barındıran N kümesi ve kontrol noktaları arasındaki bağlantıları içinde barındıran E kümesinden oluşan bir çizgemiz $G=(N,E)$ olsun. N kümesinin içindeki her bir kontrol noktasının $(n_i, 0 \leq i \leq (d+1))$ bir kazanç miktarı, $s_i \geq 0$ vardır ($s_0 = s_{d+1}=0$). Kenar kümesi, E içindeki her bir kenar bir maliyete, $c_{ij} > 0$ (n_i ve n_j kontrol noktaları arasında) sahiptir. Problemde amaç, başlangıç kontrol noktasından (n_0) başlayıp bitiş noktasında (n_{d+1}) sonlanacak ve verilen maliyet kısıtını, T_{maks} aşmayacak biçimde maksimum kazanç elde edilen araç sayısı kadar (m) rotanın bulunmasıdır. Var olan kısıttan dolayı bulunan rotaların çizgedeki tüm kontrol noktalarını içermesi her zaman mümkün olmayacaktır. EKS-RBP kümesinden Oryantiring problemi (OP) NP-zor sınıfına ait bir problemdir [Golden et al.,1987]. Takım Oryantiring Problemi (TOP), OP’nin araç sayısı bakımından genelleştirilmiş hali olduğundan bu problem de NP-zor sınıfına aittir.

EKS-RB problemleri, Oryantiring sporu yanında stok dağıtım, üretim planlama [Balas, 1989], turist rehberi [Vansteenwegen and Van Oudheusden, 2007] gibi birçok gerçek alan problemlerine de uygulanmıştır. Ayrıca posta dağıtım ve okul servisi [Kantor and Rosenwein, 1992] uygulamalarında varolan T_{maks} kısıtına ek olarak, kontrol noktalarını belli zaman aralıklarında gezme kısıtı eklenerek problem çözülmeye çalışılmıştır.

2.2 Problemin Matematiksel İfadesi

EKS-RBP, 0-1 tamsayı programlama problemi olarak aşağıdaki gibi ifade edilmiştir:

y_{ik} : Ziyaret değişkenleri: Eğer k aracı tarafından i kontrol noktası ziyaret edilmiş ise değeri 1, ziyaret edilmemiş ise değeri 0 dır.

x_{ijk} : Rota değişkenleri: Eğer (i,j) kenarı k aracının bulunduğu rota içerisinde ise değeri 1, değilse değeri 0 dır.

U : Kontrol noktaları kümesi N 'nin alt kümesidir.

$$\text{Amaç:} \quad \text{En büyük} \sum_y \sum_{k=1}^m \sum_{i=1}^d s_i y_{ik} \quad (2.1)$$

$$\text{Kısıtlar:} \quad \sum_{k=1}^m \sum_{j=1}^{d+1} x_{0jk} = \sum_{k=1}^m \sum_{i=0}^d x_{i(d+1)k} = m, \quad (2.2)$$

$$\sum_{k=1}^m y_{ik} \leq 1; \quad \forall i = 1, \dots, d, \quad (2.3)$$

$$\sum_{i < t} x_{itk} = \sum_{i > t} x_{itk} = y_{tk}; \quad \forall t = 1, \dots, d; \quad \forall k = 1, \dots, m, \quad (2.4)$$

$$\sum_{i=0}^d \sum_{j>i} c_{ij} x_{ijk} \leq T_{max}, \quad \forall k = 1, \dots, m, \quad (2.5)$$

$$\sum_{\substack{i \in U \\ j \in U}} x_{ijk} \leq |U| - 1, (U \subset N; 2 \leq |U| \leq d; k = 1, \dots, m) \quad (2.6)$$

$$x_{ijk} \in \{0, 1\}, \quad (0 \leq i, j \leq d+1; k = 1, \dots, m) \quad (2.7)$$

$$y_{0k} = y_{(d+1)k} = 1, \quad y_{ik} \in \{0, 1\} \quad (i = 1, \dots, d; k = 1, \dots, m) \quad (2.8)$$

EKS-RBP'nin amacı, formül (2.1)'de belirtildiği gibi m araç için bulunan m tane rotanın toplam kazancını en fazla hale getirmektir. Kısıt (2.2) her rotanın n_0 kontrol noktasından başlayıp n_{d+1} kontrol noktasında bitmesini garantiler. Kısıt (2.3) ise başlangıç ve bitiş dışındaki diğer tüm kontrol noktalarının en fazla bir kez ziyaret edileceğini belirtir. Kısıt (2.4) her bir rotadaki kontrol noktalarının birbirine bağlanır olduğunu garantiler. Her bir rotanın maliyetinin sınırlı olduğunu kısıt (2.5), alt rotaların engellediğini ise kısıt (2.6) gösterir. Son olarak kısıt (2.7, 2.8) formüllerde geçen karar değişkenlerinin alabilecekleri değerleri açıklar. Bu tamsayı programlama probleminde $m=1$ yapılırsa OP, $m > 1$ olursa, TOP'nin matematiksel formülü elde edilir.

2.3 Mevcut Çözüm Yaklaşımları

Bu bölümde, Oryantiring ve Takım Oryantiring problemlerinin literatürdeki mevcut çözüm yöntemleri, kesin çözümlü yöntemler, probleme özgü sezgisel yöntemler ve meta-sezgisel yöntemler başlıkları altında incelenmiştir.

2.3.1 Kesin Çözümler

Yapılan literatür taramalarında OP ve benzeri problemler için altı kesin çözüm yöntemi geliştirilirken, TOP için iki kesin çözüm yöntemi uygulandığı görülmüştür. OP ve TOP için uygulanan kesin çözümlü yöntemler Çizelge 2.1’de özetlenmiştir. Boussier ve arkadaşlarının yapmış olduğu çalışmada [Boussier et al., 2007] TOP problem kümesinden karşılaştırmalı değerlendirmelerde kullanılan 387 problemten 270’i için üst sınır (upper bound) sonuçlar bulunmuştur. En fazla 2 saatlik süre için çalıştırma yapıldığından dolayı geriye kalan 117 problem için sonuçlar üretilmemiştir.

Kesin çözüm yöntemleri küçük boyutlu problemlerde kabul edilebilir zaman aralığında çözüm sunarken, problem boyutu büyüdükçe hesaplama zamanı üssel olarak arttığından, bu yöntemleri kullanmak elverişsiz olmaktadır. Bundan dolayı çoğu araştırmacı sezgisel çözüm yöntemlerine odaklanmıştır.

Çizelge 2.1 OP ve TOP için kesin çözümlü yöntemler

Referans	Problem	Çözüm stratejisi
[Hayes and Norman,1984]	OP	Dinamik Programlama
[Kataoka and Morito, 1988]	OP	Dallan ve Sınırla
[Laporte and Martello, 1990]	OP	Dallan ve Sınırla
[Ramesh et al.,1992]	OP	Dallan ve Sınırla
[Leifer and Rosenwein, 1994]	OP	0-1 Tamsayı Programlama
[Fischetti et al.,1998]	OP	Dallandır ve Kes
[Butt and Ryan, 1999]	TOP	Kolon oluşturma tabanlı algoritma
[Boussier et al.,2007]	TOP	Dallan ve Ödüllendir

2.3.2 Sezgisel Çözümler

OP ve TOP problemlerine özgü geliştirilen sezgisel çözümler Çizelge 2.2’de listelenmiştir. Genellikle geliştirilen sezgisel yöntemlerde problem iki safhada çözülmeye çalışılmıştır: rota oluşturma ve rota iyileştirme. Rota oluşturma adımında en çok kullanılan yöntemler rasgele ve hırslı yöntemdir. Rasgele yöntem ile rota, kontrol noktaları rasgele seçilip uzunluk kısıtını geçmeyecek şekilde ardı ardına eklenerek oluşturulur. Hırslı çözümde ise her bir kontrol noktası için kendisinde mevcut olan kazanç miktarı ve/veya yol maliyeti değerleri kullanılarak bir ağırlık değeri hesaplanır. Daha sonra kontrol noktalarının bu ağırlık değerleri kullanılarak rotaya eklenme sırası oluşturulur. Kontrol noktaları sırayla rotaya eklenerek yeni rota oluşur. Rota oluşturma adımından sonra rota iyileştirme adımı başlar. Bu adımda yol maliyetini azaltıcı ve toplam kazancı artırıcı kontrol noktası ekleme, çıkartma ve yer değiştirme gibi yerel arama yöntemleri kullanılır.

Çizelge 2.2 OP ve TOP için sezgisel yöntemler

Referans	Problem	Çözüm stratejisi
[Tsiligirides, 1984]	OP	hırslı seçimle rota oluşturma
[Golden et al.,1988]	OP	hırslı seçimle rota oluşturma, rota iyileştirme
[Keller, 1989]	OP	rasgele seçimle rota oluşturma, rota iyileştirme
[Keller, 1989]	OP	hırslı seçimle rota oluşturma, rota iyileştirme
[Ramesh and Brown, 1991]	OP	rasgele seçimle rota oluşturma, rota iyileştirme
[Butt and Cavalier, 1994]	TOP	hırslı seçimle rota oluşturma
[Chao et al., 1996]	OP,TOP	rasgele seçimle rota oluşturma, rota iyileştirme

Geliştirilen sezgisel yöntemlerden en sonuncusu 1996 yılında Chao ve arkadaşlarının geliştirdiği 5 adımlık sezgisel yöntemdir. Geliştirilen bu sezgisel yöntemin yanında OP için yeni karşılaştırmalı problem kümeleri de literatüre kazandırılmıştır (Bkz. Bölüm 2.4). Chao ve arkadaşları 5 adımlık sezgisel yöntemlerini Tsiligirides’in geliştirdiği çözüm yöntemi ile karşılaştırır [Tsiligirides, 1984]. Toplam 353 test problemi içinde, Chao ve arkadaşları Tsiligirides’in sonuçlarına göre problemlerin %60’ında daha iyi sonuç alırken , %7 sinde daha kötü %33’ ünde ise aynı sonuçları elde etmişlerdir.

2.3.3 Meta-Sezgisel Çözümler

Oryantiring ve Takım Oryantiring problemlerine sezgisel yöntemlerin yanında, problemten bağımsız arama stratejisi sunan meta-sezgisel yöntemler de uygulanmıştır. Geliştirilen meta-sezgisel tabanlı çalışmalar Çizelge 2.3’de listelenmiştir. OP için geliştirilen meta-sezgisel yöntemlerden ilki Wang ve arkadaşlarının geliştirdiği yapay sinir ağı uygulamasıdır [Wang et al., 1995]. Tasgetiren ve Smith genetik algoritma kullanarak OP için çözüm önermişlerdir [Tasgetiren and Smith, 2000]. Geliştirilen bu modelde değişken uzunluklu permutasyon gösterimi seçilmiştir. Çaprazlama yöntemi ile farklı bireyler oluşturulurken ve uyarlanabilen ceza fonksiyonu kullanarak kısıtlara uymayan çözümler de popülasyon içinde kalma şansı elde etmiştir. OP için bir başka çalışma da Karınca Koloni Optimizasyonu (KKO) ve Tabu Arama (TA) yöntemleri kullanılarak gerçekleştirilmiştir [Liang et al., 2002]. Geliştirilen KKO modelinde çevrim içi ve çevrim dışı feromon güncellemesi kullanılarak karıncaların oluşturacakları çözümler koloni bilgisine göre yönlendirilmiştir. Ayrıca her bir çözüm için yerel arama çalıştırılarak çözüm kalitesinin artırılması hedeflenmiştir. Tabu Arama tabanlı modelde ise 5 komşuluk yapısı ve dinamik bir tabu listesi kullanılmıştır. Genetik Algoritma tabanlı modelde olduğu gibi bu modelde de ceza fonksiyonu kullanılarak kısıtı aşan çözümlerde de arama yapılmasına izin verilmiştir. OP için geliştirilen modeller arasında genetik algoritma tabanlı modelin hesaplama süresi bakımından diğer modellere göre çok daha zaman harcadığı görülmüştür. Buna rağmen elde edilen kazanç yönünden sonuçlar bilinen en iyi sonuçlarla rekabet eder niteliktedir.

Çizelge 2.3 OP ve TOP için kullanılan meta-sezgisel yöntemler

Referans	Problem	Çözüm stratejisi
[Wang et al., 1995]	OP	Yapay Sinir Ağları
[Tasgetiren and Smith, 2000]	OP	Genetik Algoritma
[Liang et al., 2002]	OP	Karınca Kolonileri, Tabu Arama
[Tang and Miller-Hooks, 2005]	TOP	Tabu Arama
[Archetti et al., 2007]	TOP	Tabu Arama, Değişken Komşu Arama
[Bouly et al., 2008]	TOP	Memetik Algoritma
[Ke et al., 2008]	TOP	Karınca Kolonileri
[Vansteenwegen et al., 2009]	OP,TOP	Kılavuzlu yerel arama metodu
[Souffriau et al., 2009]	TOP	Yol Bağlama

TOP için geliştirilen meta-sezgisel tabanlı çözüm yöntemlerinden ilki, Tang ve Miller-Hooks'un Tabu Arama tabanlı modelidir [Tang and Miller-Hooks, 2005]. Geliştirilen yöntem 3 ana safhadan oluşur. Birinci safhada, kısıtı aşmayan rotalardan oluşan bir çözüm havuzu oluşturulur. Aday çözüm, araç sayısı kadar havuzdan rasgele rota seçilerek oluşturulur. İkinci safhada ise aday çözüm tabu arama yöntemi ile çözüm kalitesi artırılmaya çalışılır. Üçüncü safhada, en iyi tabu olmayan çözüm yeni aday çözüm olarak atanır ve başlangıçta oluşturulan çözüm havuzuna eklenir. Havuzdan eklenen rota sayısı kadar en kötü amaç fonksiyonuna sahip rotalar çıkartılır. Algoritma birinci safhaya geri döner.

Tabu Arama tabanlı model, Chao ve arkadaşlarının [Chao et al., 1996] geliştirdiği karşılaştırmalı problemlerle test edilmiştir. Elde edilen sonuçlar, Chao ve arkadaşlarının geliştirdiği 5 adımlık sezgisel yöntemin sonuçları ile karşılaştırılmıştır. Buna göre 18 problem kategorisinin 14'ünde tabu arama algoritması en iyi ortalama sonuçları, 5-adım sezgisel metot ise 4'ünde en iyi ortalama sonuç verebilmiştir. Ayrıca 64 kontrol noktasından büyük olan problem kümelerinde tabu arama 60 problemde 55'inde en iyi sonuçlar vermiştir. Bir başka analiz, TA'da kullanılan komşuluk sayısı üzerine yapılmıştır. Geliştirilen model, bir komşulukla çalıştırılmıştır. Sonuçlara bakıldığında komşuluk alanlarında geçiş yapmak çözüm kalitesini çok değiştirmese de, CPU açısından ortalama %40 kadar hız kazanımı sağladığı görülmüştür.

Archetti ve arkadaşları TOP için bir önceki çalışmadan ilham alarak Tabu Arama tabanlı iki model ve Değişken Komşu Arama (DKA) tabanlı bir model önermişlerdir [Archetti et al., 2007]. Önerilen üç modelden DKA tabanlı ve TA tabanlı bir model uygun çözümlerle arama yaparken diğer TA tabanlı model ise kısıdı aşan çözümlere izin vererek arama yapar. Karşılaştırmalı problemlerle yapılan deney sonuçlarında yerel arama olarak TA'yı kullanan DKA modeli, diğer iki TA modeline göre daha iyi sonuçlar bulduğu görülmüştür. 320 problemde 125'inde DKA sonuçları iyileştirmiştir. Kısıdı aşmaya izin veren TA modelinde ise 101 , diğer TA modelinde ise 115 problemde iyileştirme yapmıştır. DKA tabanlı model, çalışma zamanı bakımından 20 dakikaya kadar çıksa bile çoğunlukla 10 dakikanın altında sonuçları üretmiştir.

TOP için bir başka çalışma ise Genetik Algoritma ve Yerel Aramanın birlikte kullanıldığı Memetik tabanlı algoritmadır [Bouly et al., 2008]. Kromozom olarak adlandırılan her bir aday çözüm kontrol noktaların permutasyonu olarak oluşturulur. Kromozomların bir kısmı rasgele permutasyon ile bir kısmı ise özel bir sezgisel yöntem kullanılarak oluşturulur.

Kromozom değerinden amaç fonksiyonu hesaplanırken Optimal Bölme (Optimal Splitting) yöntemi kullanılmıştır. Bu bölme yöntemi kromozomdan elde edilebilecek en iyi amaç fonksiyon değerine göre bölmeyi sağlar. Kromozomlar LOX çaprazlanma ile yeni bireyler üretirken, aynı bireyler belli bir olasılık değerine göre mutasyona da uğrarlar. Mutasyon aslında yerel arama amaçlı yapılmıştır.

Deneyler karşılaştırmalı problemlerle yapılmış olup bu test problemlerinden 11'nin sonucu iyileştirilmiştir. Diğerler problemlerde ise varolan en iyi sonuçlar bulunmuştur. Sonuçlara ortalama 10 dakikadan az sürede ulaşılmıştır.

TOP için başka bir çalışma ise Karınca Kolonileri ile yapılmıştır [Ke et al., 2008]. Bu modelde, KKA'nın genel çalışma prensibi korunmuştur. Modelde, karıncalar çözümü oluştururken hangi araca hangi kontrol noktasını eklemek gerektiğine karar verirler. Yapılan çalışmada kontrol noktaları 4 farklı şekilde araçlara atanarak denenmiştir.

- a. Birinci araçtan başlanır, kısıdı dolduğu zaman diğer araca geçilir.
- b. Önce araç tespiti yapılır daha sonra kontrol noktası araca eklenir.
- c. Araç sırası rasgele seçilir.
- d. Araca ve kontrol noktasına bağlanabilecek kenar seçilir. Böylece aynı anda hem aracı hemde kontrol noktasını seçmiş olunur.

Performans analizlerinde birinci yöntem en iyi sonuç vermiştir. Performans analizleri karşılaştırmalı test problemleri ile yapılmış olup, sonuçlar aynı dönemde çıkan memetik algoritma dışındaki diğer yöntemlerle karşılaştırılmıştır. Buna göre 12 problemin sonucunda iyileşme olduğu görülmüştür.

Son zamanlarda TOP için geliştirilen yöntemlerden biri de kılavuzlu yerel arama tabanlı çözüm yöntemidir [Vansteenwegen et al., 2009]. Yöntem üç ana kısımdan oluşmaktadır. Birinci kısmı *başlangıç çözümleri üretme* kısmıdır. Chao ve arkadaşlarının kullandıkları aynı yöntemi [Chao et al., 1996] kullanarak başlangıç çözümleri üretilir. Yani başlangıç ve bitiş kontrol noktaları odak noktası, T_{maks} değeri büyük eksen olan bir elips alanının içinde kalan kontrol noktaları arasında başlangıç ve bitiş noktasına en uzak 5 kontrol noktası seçilir ve en ucuz ekleme yöntemi ile eklenir. Bu işlem probleme ait tüm kontrol noktaları bir rotaya ait olana kadar devam eder. Bundan sonra en iyi kazancı sağlayan araç sayısı kadarki rota başlangıç çözümü oluşturmuş olur. İkinci kısım ise *kılavuzlu yerel arama* kısmıdır. Yerel arama prosedürü içinde tanımlanan beş komşuluk yapısı sırayla uygulanır. Her bir komşuluk kendi içinde yerel en iyiye ulaşıncaya kadar çalıştırılır. Kılavuzlu yerel arama yöntemi yerel arama sırasında çözümlerin istenmeyen bileşenlerini cezalandırarak veya ödüllendirerek çözümün yerel en iyiye takılmasını engeller. Böylece arama işlemine kılavuzluk eder. Üçüncü safhası ise *çözümü bozma* işlemidir. Çözümü sağlayan turların her birini yüzdelik bir oranda çözümünden çıkarır. Bu yüzdelik oran makalede %75 olarak belirlenmiştir. Bu işlem yerel arama işlemi içinde ve algoritmanın yarı iterasyona geldiği zamanda çalıştırılmıştır. Yerel arama da sonuç hep aynı en iyi sonuca ulaşıyorsa bozma uygulanır. İyileşme olursa bozma uygulanmaz. Geliştirilen yöntem hem OP hem de TOP karşılaştırmalı problemlerle test edilmiştir. Elde edilen sonuçlar [Archetti et al., 2007] ve [Tang and Miller-Hooks, 2005] sonuçlarıyla karşılaştırılmıştır. Sonuçları, elde edilen kazanç bakımından karşılaştırıldığında sırasıyla %0.84 ve %1.29 ortalama hata ile çözümlere yaklaşıldığı, çalışma zamanı bakımından karşılaştırıldığında ise diğer iki modele göre çok daha kısa sürede bu çözümlere ulaştığı görülmüştür.

TOP için geliştirilen en son modellerden bir diğeri ise Yol Bağlama tabanlı çözüm yöntemidir [Souffriau et al., 2009]. Bu yöntemde başlangıç çözümü, $c_{baş}$ oluşturulduktan sonra çözüm, [Vansteenwegen et al., 2009] çalışmasında kullanılan yerel arama metoduna sokulur. Burdan çıkan sonuç $c_{yerelarama}$ ile modelin en iyi sonucu c_{eni} arasında bir bağ kurulur. Bu bağ c_{eni} çözümünde olup $c_{yerelarama}$ çözümünde olmayan kontrol noktalarından oluşur. Tespit edilen bu kontrol noktaları $c_{yerelarama}$ çözümüne eklenmeye çalışılarak çözümün iyileştirilmesi sağlanır. Bu model farklı tekrarlama sayısı parametresine göre yavaş ve hızlı iki versiyonla

performans analizi yapılmıştır. Buna göre yavaş versiyonu ortalama 272.8 saniye çalışarak literatüre 5 yeni sonuç kazandırmıştır. Hızlı olanı ise ortalama 5 saniye çalışarak %0.39 hata ile makul sonuçlar üretmiştir.

2.4 Karşılaştırmalı Test Problemleri

OP ve TOP için çoğunlukla kullanılan karşılaştırmalı problemler, belli sayıdaki kontrol noktalarının Öklid düzlemine yerleştirilmesi ile oluşan 8 farklı problem kümesinden türetilmiştir. OP ve TOP için kullanılan problem kümeleri, kontrol noktalarının koordinat bilgileri ve her bir kontrol noktasının kazanç miktarı ile birlikte EK-A’da verilmiştir.

OP için ilk karşılaştırmalı test problemleri literatüre Tsiligirides tarafından kazandırılmıştır [Tsiligirides, 1984]. Bu test problemleri sırayla PK 1, PK 2 ve PK 3 problem kümelerine farklı T_{maks} değerlerinin verilmesiyle oluşturulmuştur. Daha sonra Chao ve arkadaşları OP için varolan problem kümeleri üzerine araç sayısını artıtarak ve her bir aracı aynı T_{maks} değerleri ile kısıtlayarak TOP için test problemleri üretmişlerdir [Chao et al., 1996]. Ayrıca varolan problem kümelerine ek olarak daha büyük problem kümeleri (PK 4 ve PK 7) ve PK1’in bir kontrol noktasının koordinat bilgilerini değiştirerek elde ettikleri PK1_1 problem kümesini de eklemiştir.

OP ve TOP için deneylerde kullanılan karşılaştırmalı test problemleri ve özellikleri sırayla Çizelge 2.4 ve Çizelge 2.5’te verilmiştir.

Çizelge 2.4 OP için kullanılan problem kümeleri ve özellikleri

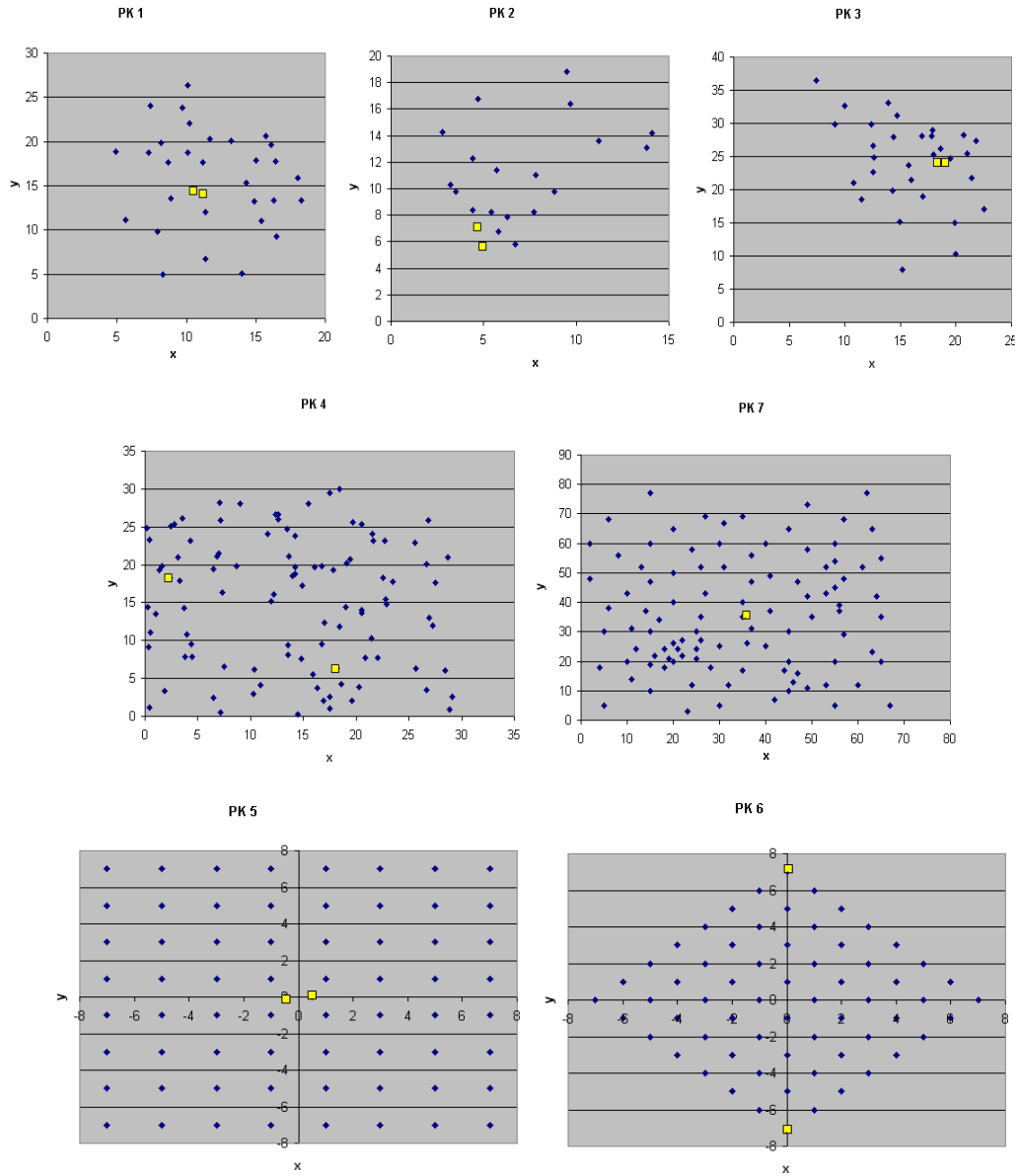
Prb. Küme Adı	Kont. Nok. Sayısı	Problem Sayısı	En Düşük T_{maks}	En Yüksek T_{maks}
PK1	32	16	15	85
PK2	21	11	15	45
PK3	33	20	15	110
PK1_1	32	16	15	85
PK5	66	26	5	130
PK6	64	14	15	80
Toplam		103		

Çizelge 2.5 TOP için kullanılan problem kümeleri ve özellikleri

Adı	Kont.Nok. Sayısı	Araç Sayısı	Problem Sayısı	En Düşük T_{maks}	En Yüksek T_{maks}
PK1	32				
		2	17	5	42.5
		3	16	5	28.5
		4	15	5	21.2
PK2	21				
		2	11	7.5	22.5
		3	11	5	15
		4	11	3.8	11.2
PK3	33				
		2	20	7.5	55
		3	19	5	36.7
		4	18	3.8	27.5
PK4	100				
		2	20	25	120
		3	20	16.7	80
		4	17	20	60
PK5	66				
		2	25	15	40
		3	25	15	26.7
		4	24	15	20
PK6	64				
		2	11	15	40
		3	8	15	26.7
		4	5	15	20
PK7	102				
		2	20	10	200
		3	19	13.3	133.3
		4	19	10	100
Toplam			350		

Kontrol noktaları köşe ve aralarındaki bağlantıları kenar kabul edildiğinden bu problem kümeleri birer çizge olarak da gösterilebilirler. Bu problem kümelerinden oluşacak çizgeler yönsüz ve tam çizge özelliğindedirler.

TOP ve OP'un kullandığı problem kümelerinin 2 boyutlu düzlemdeki kontrol noktalarının dağılımı ise Şekil 2.1'de verilmiştir. Şekiller üzerindeki kare noktalar başlangıç ve bitiş kontrol noktalarını gösterir. Şekillerden anlaşılacağı üzere PK 5 ve PK 6 da kontrol noktaları düzenli bir şekilde konumlandırılırken diğer problem kümelerinde kontrol noktaları rasgele yerlere konumlandırılarak oluşturulmuştur.



Şekil 2.1 OP ve TOP için kullanılan karşılaştırmalı problem kümelerinin 2 boyutlu düzlemde gösterimi

3 META-SEZGİSEL YÖNTEMLER

Problemden bağımsız olarak arama alanını etkili ve verimli bir şekilde dolaşıp, çözüm sunan sezgisel yöntemler *meta-sezgisel* yöntemler olarak adlandırılır. “Ötesi, daha üst düzey” manalarına gelen *meta* ve “öngörü ile arama” manasına gelen sezgisel kelimelerinin birleşiminden oluşan *meta-sezgisel* adı olarak ilk defa [Glover, 1986] çalışmasında geçmiştir. Bu zamana kadar meta-sezgisel yöntemin genel kabul görmüş kesin bir tanımı yoktur. Bu dalda uzman birkaç araştırmacının sunduğu tanımlamalar aşağıda verilmiştir:

“Meta-sezgisel yöntem, arama alanında gerektiği zamanlarda yoğunlaşma (exploiting) ve yayılma (exploring) yapabilmek için altında çalışan sezgiselleri zekice birleştirerek arama stratejisine kılavuzluk eden tekrarlı işlemler olarak tanımlanır.” [Osman and Laporte, 1996]

“Meta-sezgisel yöntemler yüksek kalitede çözüm elde edebilmek için kullandığı sezgiselleri değiştiren ve kılavuzluk eden tekrarlı bir işlemdir. Her tekrarda tek bir çözümü yada çözüm popülasyonunu değişikliğe uğratarak yeni çözümlere ulaşır. Emrinde çalışan sezgiseller basit yerel arama yada yapıcı metotlar olabileceği gibi yüksek seviyeden sezgiseller de olabilir” [Voß et al., 1999]

“Meta-sezgisel yöntemler, probleme özel sezgisellerin performansını artırmak için kılavuzluk yapan yüksek-seviyeli stratejilerdir. Başlıca amacı tekrarlı çalışarak iyileşme sağlama stratejisinin dezavantajlarından biri olan yerel en iyiden kurtulamamayı engellemektir. Bunun için daha kötü çözümleri kabul etme ya da yerel arama için yeni bir başlangıç çözümünden başlama gibi daha “akılcıca” bir yol takip eder. Kaliteli bir sonuca hızlı ulaşmak için bu yöntemler amaç fonksiyon değerlerine göre, kazandıkları tecrübelerle

göre ya da bir önceki performansına göre dolaşım stratejileri belirleyebilir. Meta-sezgisel yöntemleri, rasgele arama yönteminden ayıran en önemli fark rasgeleliği gözü kapalı bir şekilde değil daha akıllıca uygulamalarıdır. [Stützle, 1999].”

Yukarıda verilen tanımlamalardan elde edilen çıkarımlara göre meta-sezgisel yöntemler, basit sezgisel metotların birleştirilerek daha üst seviyeden bir çalışma yöntemi sunar. Bu yöntemlerin kesin metot yöntemlerine göre en büyük avantajı daha kısa sürede çözüm sunmalarıdır. Ayrıca yöntemin probleminden bağımsız sunulması sezgisel metotlara göre daha yaygın bir kullanımı sağlamıştır. Dezavantajları ise, en iyi çözümü garanti etmemesi ve iyi çözüm verebilmesi için bir çok parametrenin uygun şekilde ayarlanması gerekliliğidir. Bu işlem de çoğunlukla deneysel olarak yapılmaktadır. Ayrıca meta-sezgisel yöntemler, arama alanını dolaşırken yerel iyilerde takılıp kalma riskini aza indirgeyici stratejiler barındırmalıdır. Arama alanına gerektiği zaman iyi yayılabilmeyi (diversification), gerektiği zaman ise yerel iyilerde iyi yoğunlaşabilmeyi (intensification) sağlamalıdır. Bu iki unsurun, kullanılan meta-sezgisel yöntem tarafından yeterince dengelenebilmesi, daha kaliteli sonuçların üretilmesini sağlayacaktır. Bu açıdan meta-sezgisel yöntemleri kullanırken rasgeleliğin yanında sistematik bir dengenin de geliştirilmesi gerekmektedir.

Bu zamana kadar değişik arama stratejisine sahip birçok meta-sezgisel yöntem geliştirilmiştir. Bunlardan bazıları yerel arama metotlarını daha “akıllı” hale getirici mekanizmalarla geliştirerek yeni meta-sezgisel yöntem oluşturmuşlardır. Tabu Arama, Tavlama Benzetimi, Değişken Komşu Arama, Kılavuzlu Yerel Arama bu şekilde geliştirilmiş yöntemlerdir. Tamamen farklı arama felsefelerini içinde barındıran meta-sezgisel yöntemler de mevcuttur. Genetik Algoritma, Karınca Kolonileri Algoritması, Parçacık Sürü Optimizasyonu bu tip meta-sezgisel yöntemlere örnektir. Varolan meta-sezgisel yöntemleri [Blum and Roli, 2003]’de önerildiği gibi aşağıda sıralanan yaklaşımlarla da sınıflandırmak mümkündür :

- Popülasyon Tabanlı / Tek Nokta ile Aramalı
- Bir / Birden Fazla Komşuluk Yapıları Kullanan

- Hafızayı Az / Çok Kullanan

Meta-sezgisel yöntemler kullanarak problem çözülebilmesi için *problemin gösterim şekli* ve *amaç fonksiyonun* tanımlı olması gerekir. Bu iki bileşenden problemin gösterim şekli ile aday çözümler üretilir, amaç fonksiyon değeri ile de yöntemin elde ettiği çözümlerin kalitesi karşılaştırılır. Meta-sezgisel bir yöntemden çıkan sonuçlar çoğunlukla aşağıda listelenen kriterlerle analiz edilir:

1. Çözümün kalitesi: Oluşan çözümün amaç fonksiyon değeri
2. Çalışma süresi: Çalışma zamanı veya amaç fonksiyon hesaplama sayısı
3. Algoritmanın sağlamlığı: Algoritmanın farklı başlangıç çözümlerinden başlayarak bir kaç kere çalıştırılması sonucu elde edilen çözümlerin standart sapma değeri, bilinen en iyi çözüme değerine göre ortalama yüzdelik bağıl hata değeri

Bu tezde, En Fazla Kazanç Sağlayan Rota Bulma Problemlerine Parçacık Sürü Optimasyonu ve Değişken Komşuluk Arama yöntemleri kullanarak çözüm önerildiğinden alt bölümlerde bu iki meta-sezgisel yöntem daha detaylı anlatılacaktır.

3.1 Değişken Komşu Arama Yöntemi

Değişken Komşu Arama (DKA) yöntemi 1997 yılında Hansen ve Mladenovic tarafından geliştirilmiştir [Mladenovic and Hansen, 1997]. Meta-sezgisel yöntemlerin sınıflandırılmasında DKA, tek nokta ile arama yapan, birden fazla komşuluk kullanan ve hafızayı az kullanan yöntemler arasında sayılır. Buna göre DKA arama uzayını, sistematik bir şekilde komşuluk yapılarının değiştirilmesine göre tarayan bir meta-sezgisel yöntemdir. Algoritma oldukça genel bir çerçevede verilmiş olup, ele alınan probleme göre ayarlamalar yapılabilir. DKA yönteminin benzetimsel kodu Şekil 3.1’de verilmiştir.

Bu algoritmaya göre ilk olarak komşuluk yapıları tanımlanmalıdır. N_k , k . komşuluk yapısını gösterir. S mümkün olan tüm çözüm kümesini temsil ederken, $N_k(s)$ ise s çözümünün k . komşuluk yapısındaki tüm çözümleri temsil eder. Komşuluk yapılarını hem seçmek ve hem de sırasını belirlemek algoritmanın

performansı açısından önemli bir karardır. Genellikle komşuluk sıralaması, yakın alandan uzak alana doğru arama yapılabilecek şekilde belirlenir. Algoritma S kümesinden bir çözümü başlangıç çözümü yaparak çalışmaya başlar. İç içe döngü sayesinde algoritma S içindeki çözümleri dolaşarak en iyi çözümü bulmaya çalışır. Algoritmada *Yerel Arama* ve *Sarsma* (Shaking) fonksiyonları çözüm kümesini sistematik olarak dolaşmayı sağlar. *Sarsma*, aramayı farklı yerlere konumlandırmayı sağlarken, *Yerel Arama* ise yerel en iyilere yoğunlaşmayı sağlar. Diğer bir deyişle *Sarsma* yerel en iyilerden kurtulmayı, *Yerel Arama* ise *Sarsma* fonksiyonu sonucu oluşan s' çözümünün yakın komşularını arayarak o bölgedeki en iyi çözümü bulmaya çalışır. Yerel aramadan çıkan sonuç s'' başlangıç çözüm s ile karşılaştırılır. Eğer s'' , s başlangıç çözümünden daha iyi bir sonuç verirse algoritma birinci komşuluk yapısından ve s'' çözümü ile çalışmaya devam eder. Yerel aramadan çıkan sonuç iyi değilse algoritma s çözümünün bir sonraki komşuluk yapısından çalışmaya devam eder. İçteki döngü tanımlanan tüm komşuluk yapılarını dolaştıktan sonra sonlanır. Dıştaki döngü ise durdurma kriteri gerçekleşene kadar çalışmaya devam eder. Durdurma kriteri, belli bir CPU zamanı, belli bir tekrar miktarı veya iki iyileşme arasındaki en fazla tekrar miktarı olabilir.

```

Prosedür DKA
  Komşuluk yapılarını tanımla  $N_k$  ( $k=1, \dots, k_{\max}$ )
  Başlangıç çözümleri oluştur  $s \in S$ 
  Yap (durdurma şartı gerçekleşmediği sürece)
     $k \leftarrow 1$ 
    Yap ( $k \leq k_{\max}$ )
       $s' \leftarrow \text{Sarsma}(s), s' \in N_k(s)$ 
       $s'' \leftarrow \text{YerelArama}(s'), s'' \in S$ 
      Eğer ( $\text{AmaçFonk}(s'') < \text{AmaçFonk}(s)$ )
         $s \leftarrow s''$ 
       $k \leftarrow 1$ 
    Değilse
       $k \leftarrow k+1$ 
    Yap-Bitiş
  Yap-Bitiş
Prosedür-Bitiş

```

Şekil 3.1- DKA algoritmasının benzetimsel kodu

3.1.1 DKA Yönteminin Versiyonları

İndirgenmiş-DKA (Reduced-VNS): Bu yöntemde, DKA algoritmasının Yerel Arama kısmı yer almaz.. Yani N_k komşuluğunda rasgele bir çözüm seçildikten sonra doğrudan başlangıç çözümle karşılaştırma safhasına geçilir. Böylece çok hızlı bir

şekilde sonuç elde edilmiş olur. Bu yöntem daha çok yerel arama maliyetinin fazla olduğu büyük problemlerin çözümünde önerilmiştir.

Değişken Komşulu Yerel Arama (Variable Neighborhood Descent): Bu yöntemde, komşu yapıları yerel arama için tanımlanır. Başlangıç çözümünün (s) tanımlanan birinci komşuluk alanının içindeki en iyi çözümü $s' \leftarrow \text{Enİyi } N_1(s)$ bulunur. Eğer s' çözümü daha iyi ise algoritma s' için yeniden birinci komşuluk alanı içinde en iyi çözümü araştırır. Çözüm kötü ise bir sonraki komşuluk alanından çalışmaya devam eder. Bu versiyon daha çok yerel arama yöntemi olarak kullanılır. Değişken Komşulu Yerel Arama (DKYA) yönteminin benzetimsel kodu Şekil 3.2 'de verilmiştir.

```

Prosedür DK-Yerel Arama
  Komşuluk yapılarını tanımla  $N_k$  ( $k=1, \dots, k_{\max}$ )
  Başlangıç çözümleri oluştur  $s \in S$ 
  Yap (durdurma şartı gerçekleşmediği sürece)
     $k \leftarrow 1$ 
    Yap ( $k \leq k_{\max}$ )
       $s' \leftarrow \text{YerelArama}(s), s' \in N_k(s)$ 
      Eğer ( $\text{AmaçFonk}(s') < \text{AmaçFonk}(s)$ )
         $s \leftarrow s'$ 
         $k \leftarrow 1$ 
      Değilse
         $k \leftarrow k+1$ 
    Yap-Bitiş
  Yap-Bitiş
Prosedür-Bitiş

```

Şekil 3.2 Değişken Komşuluklu Yerel Arama benzetimsel kodu

Esnek-DKA (Skewed VNS): Bu versiyon DKA'da yer alan Sarsma ve Yerel Arama aynen korunur. Bununla beraber Yerel Aramadan çıkan s'' çözümünü kabul etme şartı DKA'ya göre esnetilmiştir. s'' çözümünün s çözümünden uzaklığı $\alpha\rho(s, s'')$ fonksiyonundan çıkan değerden küçük ise çözüm kabul edilir. $\rho(s, s'')$ fonksiyonu s ile s'' arasındaki uzaklığı geri döndürürken, α iki çözüm arasındaki uzaklığın önem katsayısını belirler. Esnek-DKA algoritmasında değişen kısmı olan içteki döngü Şekil 3.3'te verilmiştir.

```

Eğer (AmaçFonk (s'') -  $\alpha p(s, s'')$  < AmaçFonk (s))
    s ← s''
    k ← 1
Değilse
    k ← k+1

```

Şekil 3.3 Esnek-DKA , DKA'dan farklı olan kod kısmı

3.2 Parçacık Sürü Optimizasyon Yöntemi

Parçacık Sürü Optimizasyonu (PSO) Kennedy ve Eberhart tarafından lineer olmayan sürekli fonksiyonların optimizasyonu için geliştirilmiştir [Kennedy and Eberhart, 1995]. PSO'nun esin kaynağı, kuş ve balık sürülerinin sosyal paylaşımlı hareketleridir.

PSO kolay uygulanabilirliği, yerel arama ve küresel arama arasında kurduğu denge ve en iyiye yakın çözümlere hızlı bir şekilde yakınsaması ile optimizasyon problemlerin çözümü için çok kullanılır bir yöntem olmuştur. Algoritmanın doğası itibari ile önceleri sürekli optimizasyon problemlerinin çözümü için kullanılırken, sonrasında yapılan çalışmalar göstermiştir ki PSO ayrık problemlerde de başarılı bir şekilde uygulanabilmektedir.

PSO popülasyon tabanlı bir yöntemdir. Popülasyondaki her bir bireye *parçacık* denir. Parçacıklardan oluşan popülasyona da *sürü* ismi verilir. PSO'da arama işlemi popülasyondaki parçacıklar tarafından ve belirlenen iterasyon sayısınca yapılır. Sürüdeki her bir parçacık, pozisyon (p) ve hız (v) vektörüne sahiptir. Parçacığın pozisyonu aday çözümü temsil eder. Hız vektörü ise bir sonraki adımda parçacığın çözüm alanında nereye konumlanacağına yardım eder. Her bir iterasyonda parçacıkların aldıkları en iyi değer (*yerel tecrübeleri*) ve her iterasyonda sürünün elde ettiği en iyi değer (*sürü tecrübesi*) algoritma çalıştığı sürece hafızada saklanır. Parçacığın ne kadar kaliteli çözüm sunduğu, amaç fonksiyonun parçacığının pozisyon vektörünü kullanarak elde ettiği değerden ortaya çıkar. İterasyonlar süresince her bir parçacık yerel tecrübesini, sürü tecrübesini ve hızını kullanarak daha iyi çözümlere ulaşmaya çalışır.

```

Prosedür PSO
Başlangıç Popülasyonunu Oluştur
Herbir Parçacık İçin{
    Amaç Fonksiyon Değerlerini Bul
    Yerel En İyiyi Bul
}
Global En İyiyi Bul
Tekrarla{
    Herbir Parçacık İçin{
        Hız Vektörünü Güncelle
        Pozisyon Vektörünü Güncelle
        Amaç Fonksiyon Değerlerini Bul
        Yerel En İyiyi Güncelle
    }
    Global En İyiyi Güncelle
} Durdurma Kriteri
Prosedür Bitiş

```

Şekil 3.4 PSO algoritmasının benzetimsel kodu

PSO algoritmasının benzetimsel kodu Şekil 3.4'te gösterilmiştir. Bu algortmada ilk olarak sürüdeki parçacıkların pozisyon vektörleri çözüm alanı içinde rasgele veya önceden belirlenen değerlerle oluşturulur. Sonrasında pozisyon değerlerine karşılık gelen amaç fonksiyon değeri hesaplanır. Başlangıçta her bir parçacığın amaç fonksiyon değeri kendisinin yerel tecrübesi (*pbest*) olarak saklanır. Parçacıklar arasında amaç fonksiyon değeri en iyi olan parçacık ise sürü tecrübesi (*gbest*) olarak saklanır. Her bir parçacığın hız vektörü elemanları ise $[V_{min}, V_{max}]$ değerleri arasında rasgele değerlerden oluşur. Bir sonraki iterasyonda parçacıkların pozisyon ve hız vektörleri formül (3.1) ve (3.2) kullanılarak güncellenir. Bu işlem bir durdurma kriterine kadar devam eder.

$$v_{i,j}^t = v_{i,j}^{t-1} + c_1 rand_1(pbest_{i,j}^{t-1} - p_{i,j}^{t-1}) + c_2 rand_2(gbest_j^{t-1} - p_{i,j}^{t-1}) \quad (3.1)$$

$$p_{i,j}^t = p_{i,j}^{t-1} + v_{i,j}^t \quad (3.2)$$

Bu formüllerde, $v_{i,j}^t$ ve $p_{i,j}^t$ sırasıyla t. iterasyondaki i. parçacığın j. boyuttaki ($1 \leq j \leq d$) hız ve pozisyonunu temsil ederler. Parametre c_1 ve c_2 ise sürü içerisindeki sosyal etkileşim ve parçacığın yerel tecrübe değerlerinin ağırlık katsayılarıdır. PSO'nun stokastik davranışını (0,1) arasında rasgele oluşturulan *rand1* ve *rand2* katsayıları belirler. İşlemler sonrasında hız ve pozisyon değerlerinin sınır değerleri aşıp aşmadığı kontrol edilir ve belirlenen arama stratejisine göre işlem yapılır.

3.2.1 PSO Yönteminin Versiyonları

PSO algoritması geliştirildiğinden itibaren bir çok optimizasyon problemlerine uygulanmıştır. Bununla beraber çözülmek istenen problemin alanı ve/veya amaç fonksiyonu için PSO 'nun yetersiz kaldığı durumlar tespit edilmiştir. Bu noksanlıkları gidermek için PSO'ya farklı düzeylerde eklentiler yapılmış ve böylece birçok versiyonu ortaya çıkmıştır. Bu bölümde PSO'nun eklentileri ve değişikliklerini ihtiva eden versiyonları, *Sürekli problemler için PSO versiyonları* ve *Ayrık problemler için PSO versiyonları* adında iki ana başlıkta anlatılacaktır.

3.2.1.1 Sürekli problemler için PSO versiyonları

- Atalet ağırlıklı PSO (Inertia weighted PSO-IwPSO)

Shi ve Eberhart 1998 yılında yaptığı çalışmada PSO algoritması için atalet ağırlığı (inertia weight) parametresini tanıtmıştır [Shi and Eberhart, 1998]. Bu parametreyi eklemadaki amaç, PSO'nun küresel ve yerel arama mekanizmalarını kontrol edebilmektir . Buna göre PSO algoritmasında (3.1) formülü aşağıdaki gibi değişir.

$$v_{i,j}^t = wv_{i,j}^{t-1} + c_1 rand_1(pbest_{i,j}^{t-1} - p_{i,j}^{t-1}) + c_2 rand_2(gbest_j^{t-1} - p_{i,j}^{t-1}) \quad (3.3)$$

Atalet ağırlığının (w) büyük olması parçacığın daha fazla hız vektörüne göre yönelmesini bir başka ifade ile parçacığın arama alanını genişletmesini sağlarken, atalet ağırlığının küçük olması parçacığın yerel en iyi noktasına doğru yakınsamasını sağlar. Bundan dolayı da PSO'nun düzgün yakınsama yapabilmesi için atalet ağırlığını başlangıçta büyük bir değer (0.9) ile başlatıp lineer olarak küçültülmesi önerilmiştir [Shi and Eberhart, 1998].

- Daralma faktörlü PSO (Constriction PSO- ConPSO)

PSO'nun yakınsama kabiliyeti ilk olarak Clerc ve Kennedy tarafından yapılan çalışmada incelenir [Clerc and Kennedy, 2002]. Bu çalışmada PSO'nun daha iyi yakınsama yapabilmesi için hız vektörünün değerini kısıtlayan bir daralma faktörü (χ) parametresinin (constriction factor) eklenmesi önerilir ve yapılan deneylerle c_1 ve c_2 'nin 2.05 ve χ değerinin 0.729 olarak alındığında yakınsama kabiliyetinin arttığı gözlemlenir.

$$v_{i,j}^t = \chi(v_{i,j}^{t-1} + c_1 rand_1(pbest_{i,j}^{t-1} - p_{i,j}^{t-1}) + c_2 rand_2(gbest_j^{t-1} - p_{i,j}^{t-1})) \quad (3.4)$$

$$\chi = \frac{2}{2 - \beta - \sqrt{\beta^2 - 4\beta}}, \quad \beta = c_1 + c_2 \quad (3.5)$$

- Amaç Fonksiyon Değeri ile Uzaklık Orantısına Dayalı PSO (Fitness-distance-ratio-based PSO-FDRPSO)

Standart PSO, parçacık yakınsamasını *pbest* ve *gbest* konumlarına doğru yapar. Parçacığın tüm boyut değerleri iki vektörün değerleri kullanılarak değişir. Parçacık bir boyutunda iyi bir değer elde etse de vektör güncellemesinde bu değeri kaybetmesi bundan dolayı mümkündür. Peram ve arkadaşları parçacıkların hız güncellemesi işlemine yeni bir öneri getirir; Parçacığın her bir hız vektör elemanı güncellenirken, daha iyi bir amaç fonksiyon değerine sahip ve güncellenecek parçacığa o boyutta daha yakın olan parçacığın yerel tecrübesinden de yararlanılması sağlanır [Peram et al., 2003]. Buna göre hız güncelleme formülü aşağıdaki gibi değişir:

$$v_{i,j}^t = wv_{i,j}^{t-1} + c_1 rand_1(pbest_{i,j}^{t-1} - p_{i,j}^{t-1}) + c_2 rand_2(gbest_j^{t-1} - p_{i,j}^{t-1}) + c_3 rand_3(pbest_{n,j}^{t-1} - p_{i,j}^{t-1}) \quad (3.6)$$

Elde edilen yeni formülde $pbest_{n,j}^{t-1}$ i. parçacığa amaç fonksiyon değeri-uzaklık oranını en yüksek değerle sağlayan j. boyuttaki n.parçacığın yerel tecrübesidir. Parçacıklar arasındaki oran aşağıdaki gibi hesaplanır:

$$MAX_{n \in parçacık_sayısı} \left\{ \frac{Fitness(pbest_n) - Fitness(p_i)}{|pbest_{n,j} - p_{i,j}|} \right\} \quad (3.7)$$

- Garantili Yakınsayan PSO: (Guaranteed Convergence PSO -GCPSO)

Standart PSO algoritmasında global en iyi olan parçacığın hızı güncellenirken hız formülündeki diğer bileşenlerden çıkacak değer 0 olacağından sadece atalet ağırlık parametresi ve hızına bağlı olarak güncellenme işlemi gerçekleşir. Zamanla global en iyinin hızı beslenemediğinden 0 'a doğru gider ve parçacık olduğu noktada kalır ve diğer parçacıklar da global en iyi tarafına doğru yönleneceği için, sürü bu noktaya yakınsar. Eğer bu yakınsama durumu istenilen en iyi noktaya varmadan olursa sürü erken yakınsamış olur ki bu istenmeyen bir durumdur. Bu durumu aşmak

için yapılan çalışmalardan biri de Van den Berg ve arkadaşlarının yapmış olduğu garantili yakınsayan PSO çalışmasıdır [Van den Berg et al., 2002]. Bu çalışmada da hız güncelleme formülü sadece global en iyi parçacık için aşağıdaki gibi değiştirilir.

$$v_{i,j}^t = \omega v_{i,j}^{t-1} + (pbest_{i,j}^{t-1} - p_{i,j}^{t-1}) + k^{t-1} r_j \quad (3.8)$$

r_j [-1,1] arasında düzgün dağılımlı rastgele sayılardan oluşan vektördür. k^t ölçekleme faktörüdür. İlk değeri $k^0=1.0$ verilir. İterasyon ilerledikçe değeri aşağıdaki bağıntıya göre hesaplanır.

$$k^{(t+1)} = \begin{cases} 2k^{(t)} & \text{basari_sayisi} > s_c \\ 0.5k^{(t)} & \text{basarisizlik_sayisi} > f_c \\ k^{(t)} & \text{diğer} \end{cases} \quad (3.9)$$

s_c ve f_c ayarlanabilen eşik parametreleridir. Eğer $gbest$ parçacığı daha iyi bir pozisyon bulursa $basari_sayisi$ değeri artırılırken $basarisizlik_sayisi$ değeri 0'lanır. Eğer $gbest$ parçacığı daha kötü bir pozisyon bulursa tam ters işlem yapılır.

Bu güncelleme sayesinde standart PSO 'da global en iyi parçacık sadece azalan hız vektörüne bağımlı kalmayıp, en iyi parçacığın çevresine yönlendirilerek, yerel en iyiyi bulana kadar aramaya devam eder. Bu yöntem, tek biçimli (arama uzayında tek en iyi değer olması) fonksiyonlarda iyi çalışmasına rağmen, çok biçimli (arama uzayında birden fazla yerel en iyi olan) fonksiyonlarda erken yerel en iyiye takılmasından dolayı zayıf sonuçlar verir.

- Tam Bilgilendirilmiş PSO (Fully Informed PSO -FIPSO)

Mendes ve arkadaşları tarafından sunulan bu çalışma, parçacığın hızını güncellerken sadece $pbest$, $gbest$ pozisyonlarından faydalanmak yerine o parçacığın tüm komşularının tecrübelerinden faydalanmayı önerir [Mendes et al., 2004]. Hızı yenilenecek parçacığın komşuları amaç fonksiyon değeri ve komşuluk genişliğine göre ağırlıklık katsayı değerleri hesaplanır. Aslında bu algoritma conPSO'nun geliştirilmiş halidir. conPSO'da c katsayısı $gbest$ ve $pbest$ vektörlerine eşit paylaştırılırken FIPSO'da c , komşusu olan tüm parçacıklara bölünerek eşit dağıtılır. Hız güncellenme formülü aşağıdaki gibi değişir. Bu formülde $W(s)$, s . parçacığın

ağırlık katsayısını, $pbest_s$, s . parçacığın yerel tecrübesini, N ise s .parçacığın komşu parçacıklar kümesini gösterir.

$$v_i^{t+1} = \mathcal{X} \left[v_{i,j}^t + \sum_{s \in N} c_s W(s) U_s(0,1) * (pbest_{s,j} - p_{i,j}^t) \right], \quad c_s = c / |N| \forall s \in N \quad (3.10)$$

- Mutasyon Operatörlü PSO (MutatedPSO-MPSO)

Bazı çalışmalarda evrimsel algoritmaların bileşenleri olan mutasyon, seçim, çaprazlama gibi operatörler PSO'da da kullanılmıştır. Bu çalışmalardan biri, mutasyon operatörünü PSO içinde kullanan çalışmadır [Esquivel et. al., 2003]. Mutasyon operatörü parçacığın dolaşım alanını artırmak amaçlı uygulanır. Parçacığın pozisyon vektörü olasılık değerine göre mutasyona uğratılır. Pozisyon vektörünün mutasyona uğrama olasılığı iterasyon sayısına bağlanmış olup, iterasyon ilerledikçe pozisyon vektörüne mutasyon yapılma ihtimali azalır.

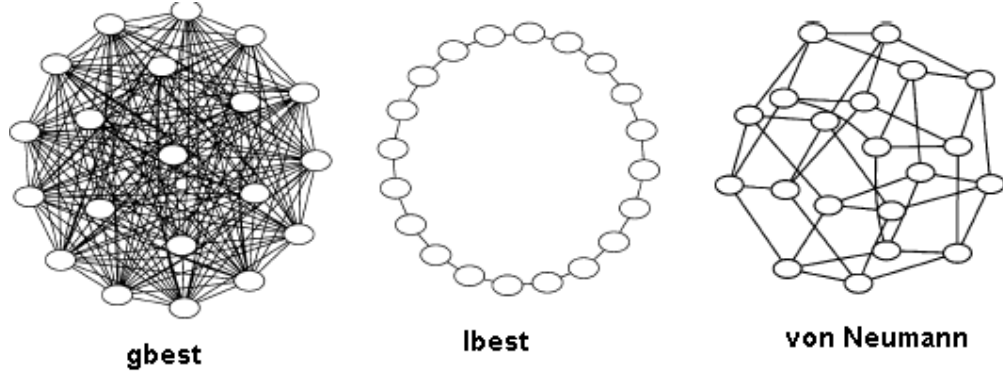
- Komşuluk Topoloji Çalışmaları

Komşuluk topoloji çalışmaları PSO'nun ortaya çıktığı andan itibaren devam edegelmiştir. Kompleks problemlerin daha küçük komşuluk kullanılarak, basit problemlerin ise geniş komşuluk kullanılarak daha iyi performans vereceği gösterilmiştir [Kennedy and Mendes, 2002],[Kennedy, 1999]. Sugantan iterasyon başlarında daha az komşu ile iterasyon sonuna doğru tüm parçacıkları komşuluk alanına dahil edecek şekilde dinamik bir komşuluk sistemi önermiştir [Suganthan, 1999]. Peer ve arkadaşları ise GCPSO'nun performansını değişik komşuluk yapıları oluşturarak denemişlerdir [Peer et al., 2003]. Çalışmada üç farklı komşuluk alanı tanıtılır :

1. Tümüyle birbirine bağlı ağ yapısı (fully connected network structure): Bu topoloji algoritmalarında GBEST olarak veya algoritma isminin yanına $-g$ gösterimi (örn: GCPSO-g) ile belirtilir. Bu topolojide her iterasyondaki global en iyi, tüm parçacıklar tarafından seçilir.
2. Yakın olan k tane komşunun oluşturduğu topoloji: Bu topoloji algoritmalarında LBEST olarak veya algoritma isminin yanına $-l$ gösterimi (örn: GCPSO-l) ile belirtilir. Genellikle parçacık bir index öncesi ve sonrasını komşusu olarak atar.

3. Von Neumann topolojisi: Bu topolojide ise her parçacık ızgara şeklinde 4 komşusu ile birbirine bağlanır. Algoritma isminin yanına $-v$ gösterimi (örn: GCPSO- v) ile belirtilir.

Şekil 3.5' de topoloji yapıları gösterilmiştir.



Şekil 3.5 Komşuluk topojileri [Peer et al., 2003]

- Birleştirilmiş PSO (Unified PSO-UPSO)

Bu versiyon hız vektörünün güncelleme işleminde GBEST ve LBEST topolojilerini birleştirilir [Parsopoulos and Vrahatis, 2004]. Formül 3.11'deki G vektörü, parçacığın GBEST topolijisine göre yeni hız değerini, L vektörü (formül 3.12) ise, parçacığın LBEST'e göre yeni hız değerini gösterir. $gbest_{pi}$, p_i komşuluğundaki en iyi vektördür. Pozisyon güncellenirken (0,1) arasındaki rasgele sayının çarpımlarından elde edilen sonuç (formül 3.13), parçacığın bulunduğu pozisyona eklenerek yeni pozisyon değeri üretilir (formül 3.14).

$$G_{i,j}^{t+1} = \chi \left[v_{i,j}^t + c_1 rand_1 \cdot (pbest_{i,j}^t - p_{i,j}^t) + c_2 rand_2 \cdot (gbest_j^t - p_{i,j}^t) \right] \quad (3.11)$$

$$L_{i,j}^{t+1} = \chi \left[v_{i,j}^t + c_1 rand_1 \cdot (pbest_{i,j}^t - p_{i,j}^t) + c_2 rand_2 \cdot (gbest_{p_i,j}^t - p_{i,j}^t) \right] \quad (3.12)$$

$$U_i^{t+1} = u G_i^{t+1} + (1-u) L_i^{t+1} \quad (3.13)$$

$$p_i^{t+1} = p_i^t + U_i^{t+1} \quad (3.14)$$

- Kapsamlı Öğrenen PSO (Comprehensive Learning Particle Swarm Optimizer-CLPSO)

Bu çalışmada [Liang et al., 2006] amaç, PSO'nun erken yakınsamasını engelleyici bir mekanizma geliştirmektir. Bu mekanizma diğer çalışmalarda olduğu

gibi hız vektörünün güncellenmesi kısmında çalışır ve bir parçacığın hız vektörü güncellenirken diğer parçacıkların kazandığı tecrübelerin de kullanılması sağlanır.

Standart PSO'da her bir parçacığın hız vektörünü güncellerken o parçacığın hız, yerel iyi (*pbest*) ve sürü en iyisi (*gbest*) kullanılır. CLPSO'da doğrudan *pbest* ve *gbest* kullanmak yerine, her bir parçacığa özel ve bütün parçacıkların *pbest* değerleri kullanılabilecek olasılıkta bir *pbestfi* ($i=1,2,...,n$; n : parçacık sayısı) vektörü oluşturulur. Bu vektör ve hız vektörü kullanılarak parçacığın pozisyon vektörü güncellenir. Böylece parçacıkların arama alanı artırıldığı gibi, sürünün en iyi vektörünün sürüyü erkenden sürükleyeceği yerel en iyiye takılma önlenmiş olunur. Her parçacığa ait olarak oluşturulan *pbestfi* vektörü belli bir iterasyon sayısı kadar pozisyon vektörünü hesaplanmakta kullanılıp daha sonra yeniden *pbestfi* vektörü oluşturulur. *pbestfi* vektörünün oluşturulması Şekil 3.6 verilmiştir.

```

Prosedür VektorPbestf (i:parçacık indeksi)
  Tekrarla (d=1,...,D, D:vektörün boyutu )
    Eğer (rand()<Pci)
      popülasyondan hız vektörü güncellenmemiş iki parçacık
      seç (p1,p2)
      Eğer (AmaçFonk (pbest (p1)<AmaçFonk (pbest (p2) )
        pbestfid = pbestp1d
      Değilse
        pbestfid = pbestp2d

      Değilse
        pbestfid = pbestid
    Tekrarla-Bitiş
  Prosedür-Bitiş

```

Şekil 3.6 pbestf_i vektörünü oluşturan prosedür

Pc öğrenme olasılık değeridir. Her bir parçacığa ayrı bir öğrenme olasılık değeri atanır. Böylece her parçacık farklı seviyelerde yakınsama kabiliyetine sahip olur. Çalışmada *Pc*'ye 0.05 ile 0.5 arasında değerler verilmiştir. Buna göre parçacığın hız vektörü formül (3.15)'e göre güncellenir.

$$v_{i,j}^t = \omega v_{i,j}^{t-1} + c_1 \text{rand}_1(pbestf_{i,j}^{t-1} - p_{i,j}^{t-1}) \quad (3.15)$$

3.2.1.2 Ayırık problemler için PSO versiyonları

Bu bölümde, ayırık problemlerde PSO algoritmasının kullanımı için literatürde sunulan PSO modellerinden birkaçı anlatılacaktır.

▪ Model 1

Bu ayırık modelde çözümler parçacığın pozisyon vektöründe permutasyon şeklinde saklanır [Rameshkumar et al., 2005]. Parçacığın hız vektörü ise $(v_i = ((ind_{1q}, ind_{2q})), q=1, \dots, L)$ L uzunluğunda pozisyon vektörünün 2'şerli indeks bilgisini tutar. Standart PSO'da hız vektörünün amacı parçacığı bulunduğu yerden arama alanı sınırlarında başka yerlere götürebilmektir. Ayırık modelde hız vektörü bileşenleri sırayla permütasyona yer değiştirme operatörü olarak yani (ind_{1q}, ind_{2q}) indekslerinde bulunan permütasyon değerlerinin yerlerinin değiştirilmesi şeklinde yapılır. Bu işlem, hız vektörünün uzunluğunca devam eder. Böylece çözüm arama alanında başka yere gitmiş olur. Bu yüzden standart PSO formüllerinin içindeki tüm operatörler, ayırık arama uzayında bir çözümünden diğerine geçebilmesi için aşağıdaki gibi yeniden tanımlanır.

pozisyon-pozisyon operatörü $(pbest_i^t - p_i^t)$: p_i pozisyonun $pbest$ pozisyonundan çıkarılmasıyla yeni hız vektör tipinde ikili yer değiştirme çiftleri meydana gelir.

pozisyon+hız operatörü $(p_i^t + v_i^{t+1})$: p_i pozisyonuna hız vektöründeki çiftler yukarıda açıklandığı gibi sırayla uygulanır.

hız+hız operatörü: hız vektörü güncelleme formülünde üç bileşen bu operatörü kullanarak yeni hız vektörünü oluşturur. Hız vektöründe ikili indeks çiftleri barındırdığından + operatörü solundaki hız vektörünün sonuna sağ taraftaki ikili indeks çiftlerini ekler.

katsayı x hız operatörü $(\omega^t \times v_i^t)$: Bu operatör sağındaki hız vektöründen ne kadar yararlanacağını sonucunu verir. Yani hız vektör elemanlarından kaç tanesinin kullanılacağına karar verir.

Aşağıda PSO'nun tanımlanan operatörleri kullanımını ve güncelleme işlemi gösteren bir örnek Şekil 3.7'de verilmiştir:

$$\begin{aligned}
 p_i^t &= (2,3,1,4) \\
 pbest_i^t &= (2,3,1,4) \\
 gbest^t &= (3,4,2,1) \\
 v_i^t &= ((2,4),(4,1)) \\
 v_i^{t+1} &= 0.5 \times [(2,4),(4,1)] + 0.5 \times [(2,3,1,4) - (2,3,1,4)] + 0.5 \times [(3,4,2,1) - (2,3,1,4)] \\
 &= [(2,4),(1,2),(2,4)] \\
 p_i^{t+1} &= p_i^t + v_i^{t+1} \\
 &= (2,3,1,4) + [(2,4),(2,3),(2,4)] \\
 &= (2,3,4,1)
 \end{aligned}$$

Şekil 3.7 Model 1'e göre pozisyon ve hız güncelleme örneği

Hız formülünde birinci bileşenden 0.5 ile x operatörüne sokulduğunda [(2,4)] sonucu çıkar. İkinci bileşende $pbest$ ve p vektörleri aynı olmasından dolayı hız uzunluğunu oluşturacak eleman yoktur. Bu yüzden bu bileşenden bir ikili indeks çifti gelmez. Üçüncü bileşenin $[(3,4,2,1) - (2,3,1,4)]$ sonucu ise [(1,2),(2,4),(3,4)] çıkar. 0.5 katsayısı ile çarpım sonucu indeks çiftleri ikiye tam olarak bölünemeyeceğinden $3 \times 0.5 = 2$ kabul edilir ve [(1,2)(2,4)] sonucu elde edilir. Bundan sonra tüm hız parçacıkları + operatörü ile birleştirilip bir sonraki hız vektörü oluşturulmuş olur.

▪ Model 2

Lio ve arkadaşlarının önerdiği bu modelde hız vektörü, pozisyon vektörünün {0,1} değerleri almasına karar verecek olan olasılık değerini tutar [Liao et al., 2007]. Hızın [0,1] arasında değer almasını (3.16)'da gösterilen sigmoid fonksiyonu ile sağlanır.

$$s(v_{ij}^t) = \frac{1}{1 + \exp(-v_{ij}^t)} \quad (3.16)$$

Parçacık pozisyonu, oluşturulan (0,1) arasında rastgele sayının v_{ij} 'den büyük olması durumunda $p_{ij} = 1$, aksi halde ise $p_{ij} = 0$ değerini alır.

▪ Model 3

Geliştirilen bir başka ayrık PSO modeli ise [Tasgetiren et al., 2007a] sunmuştur. Bu modelde PSO operatörlerinin yerini evrimsel operatörler alır. Önerilen modele göre i . parçacığın t . iterasyondaki pozisyonunu güncelleme işlemi aşağıdaki formüle göre yapılır:

$$p_i^t = c2 \oplus F3(c1 \oplus F2(w \oplus F1(p_i^{t-1}), pbest_i^{t-1}), gbest^{t-1}) \quad (3.17)$$

λ_i^t ve δ_i^t geçici pozisyon vektörü olduğu düşünüldüğünde, ilk işlem parantezin en içinden başlar. $\lambda_i^t = w \oplus F1(p_i^{t-1})$ parçacığın hız vektörü ile başka yere yönelmesini temsil eden “düğüm ekleme” işlemi F1 operatörü w ihtimali ile gerçekleştirilir. Bir başka ifade ile rasgele oluşturulan sayı w değerinden küçük ise parçacığın pozisyon vektörü F1 operatörü ile işleme girip pozisyon değişikliğe uğratılır. Eğer w değerinden büyük değer oluştu ise pozisyon vektörü değişmez ($\lambda_i^t = p_i^{t-1}$). Bu işlem sona erdikten sonra sıradaki işlem $\delta_i^t = c1 \oplus F2(\lambda_i^t, pbest_i^{t-1})$ dir. Bu işlem ise parçacığın kendi tecrübesinden yararlanmasını sağlar. Birinci işlem sonucu λ_i^t ile $pbest$ vektörlerini $c1$ ihtimali ile çaprazlama (F2) işlemine sokulur. Bu işlemden çıkan sonuç ise $c2$ ihtimalle $gbest$ vektörü ile çaprazlama (F3) işlemine sokulur. ($p_i^t = c2 \oplus F3(\delta_i^t, gbest^{t-1})$) Makalede, F1, F2 ve F3 operatörlerinin sık çalışabilmesi için $w=0.9$, $c1$ ve $c2$ ‘ye ise 0.8 değerleri verilmiştir.

▪ Model 4

Bu modelde parçacığın pozisyon ve hız vektörleri standart PSO’daki gibi sürekli değer üzerinden güncellenir. Pozisyon vektörü ara bir işlemle permutasyona çevrilip amaç fonksiyon değeri hesaplanır. Bu yöntemin avantajı kolay uygulanabilir olmasıdır. Değişik ayrık problemlere bu model kullanılarak iyi sonuçlar alınmıştır. [Zhu et al., 2006], [Tasgetiren et al., 2004].

4 DEĞİŞKEN KOMŞU ARAMA TABANLI ÖNERİLEN MODELLER

Bu bölümde OP ve TOP için geliştirilen İççe-İndirgenmiş Değişken Komşu Arama modeli anlatılacaktır. Modelin etkinliğini ortaya çıkarmak için bu modelden türetilmiş iki adet de yan model sunulacaktır. Geliştirilen bu modellerin detayları OP ve TOP için ayrı alt başlıklar altında açıklanacaktır.

4.1 Oryantiring Problemi için Değişken Komşu Arama Modeli

Bu bölümde OP için geliştirilen İççe-İndirgenmiş Değişken Komşu Arama modeli, problem gösterim şekli, komşuluk yapıları ve modelin işleyişi başlıkları altında detaylı olarak anlatılacaktır.

4.1.1 Problem Gösterimi

Problemin gösterim şekli, başlangıç ve bitiş kontrol noktaları dışındaki diğer noktaların permütasyonu şeklinde düşünülmüştür. Bu permütasyondan OP çözümü, permütasyondaki ilk elemandan başlayarak her bir kontrol noktasının başlangıç ve bitiş kontrol noktaları arasına sırayla maliyet kısıdını (T_{maks}) aşmayacak şekilde eklenmesiyle elde edilir. Maliyet kısıtı geçildiği anda eklenme işi sona erer ve böylece aday çözüme ulaşılmış olunur. Başka bir ifade ile, $(n_{i_1}, n_{i_2}, \dots, n_{i_d})$ kontrol noktalarının permütasyonu olsun. Bu permütasyondan elde edilecek OP çözümü n_0 başlangıç noktasından başlayan ve n_{d+1} bitiş noktasında sonlanan $n_0, n_{i_1}, n_{i_2}, \dots, n_{i_d}, n_{d+1}$ rotası $(q \leq d)$ ise bu rota maliyet kısıdını sağlar $(c_{0i_1} + c_{i_1i_2} + c_{i_2i_3} + \dots + c_{i_{q-1}i_q} + c_{i_qd+1} \leq T_{maks})$ fakat bir sonraki n_{q+1} noktasının eklenmesi maliyet kısıtının aşılmasına sebep olur $(c_{0i_1} + c_{i_1i_2} + c_{i_2i_3} + \dots + c_{i_{q-1}i_q} + c_{i_qi_{q+1}} + c_{i_{q+1}d+1} > T_{maks})$.

Bu çözümün amaç fonksiyonu $\sum_{j=1}^q s_{i_j}$ değerine sahiptir. Belirtmek gerekir ki bu yolla oluşturulan tüm aday çözümler uygunluk (feasibility) şartını sağlar.

4.1.2 Komşuluk Yapıları

DKA algoritmasında komşuluk yapılarını tespit etmek en önemli işlemlerden biridir. Komşuluk yapılarını belirlemek ve sıralamak genel performansı doğrudan etkiler. OP için, aşağıda detayları verilen beş komşuluk yapısı kullanılmıştır:

Düğüm Ekleme (Node Insert): Düğüm ekle yapısı çözümün yakınında bir komşu çözüm elde etmek için sıklıkla kullanılan bir yapıdır. Bu yapıda, permütasyonda rasgele seçilen bir kontrol noktası rasgele seçilen diğer bir kontrol noktasının önüne eklenir. Permütasyonda bu şekilde bir ekleme işlemi gerçekleştiğinde OP için şu durumlar ortaya çıkabilir:

- Çözüm içindeki bir kontrol noktası silinmiş olabilir.
- Çözüm dışındaki bir kontrol noktası çözüme dahil olabilir.
- Çözüm içindeki bir noktanın yeri değişebilir.
- Çözüm dışındaki bir noktanın yeri değişebilir. Bu varolan çözüm üzerinde etki göstermez ama sonraki operasyonlar için etkili olur.

Yukarıda açıklanan problem gösterim şekli kullanılarak bu komşuluk yapısı daha kararlı hale getirilebilir. Başka bir ifade ile çözümün geçerlilik kriterinin aşıldığı ilk kontrol noktasının permütasyon içindeki konumu permütasyonu iki kısma ayırır: çözüme dahil olanlar, çözüme dahil olmayanlar. Düğüm ekleme işlemi bu iki kısım arasında yapılmaya zorlandığı zaman *Kararlı-Düğüm Ekle* yapısı elde edilmiş olur ve şu durumlar gözlenir:

- Çözüm içinden bir kontrol noktası çözüm dışına eklenirse, çözümün dışında ilk sıralarda kalan kontrol noktalarının çözüme girme olasılığı diğerlerine göre daha fazla olur.
- Çözüm dışından çözüm içine ekleme yapılırsa, çözüm dışında kalan her kontrol noktasının çözüme eklenme olasılığı ise eşit olur.

Düğüm Yer Değiştirme (Node Exchange): Ekle yapısından biraz daha uzak alanda arama yapmayı sağlayan düğüm yer değiştirme komşuluk yapısında permütasyon içinde rastgele iki kontrol noktası belirlenir ve daha sonra bu noktalar

birbirleri ile yer değiştirir. Bu komşuluk yapısında da değişen noktalar çözümün içinde olup olmama durumuna göre şu işlemleri oluşur:

- Çözüm içerisindeki bir kontrol noktası, çözüm dışındaki bir nokta ile yer değiştirebilir. Bu işlem aslında silme ardından aynı pozisyona ekleme yapma manasına gelir.
- Çözüm içindeki iki kontrol noktası yer değiştirebilir.
- Çözüm dışındaki iki kontrol noktası yer değiştirebilir.

Bununla beraber yukarıda listelenen işlemlerin ilk ikisini yaptırmaya zorlayarak *Kararlı-Düğüm Yer Değiştirme* yapısı oluşturulabilir.

Yol Ekleme (Path Insert): Yol-ekleme yapısı aslında ekleme yapısına benzerlik gösterir. Buradaki tek farklılık permütasyon üzerinde rasgele iki kontrol noktası seçmek yerine üç tane seçmektir: ilk ikisi yolun başlangıç ve bitiş noktasını gösterir sonuncusu ise yolun nereye ekleneceğini gösterir. Eğer bu yolun başlangıç noktası çözüm içinden, eklenme yeri ise çözüm dışından seçilirse *Kararlı-Yol Ekleme* yapısı oluşturulmuş olur. Yol-ekleme ile oluşan yeni çözüm yukarıda açıkladığımız ekleme ve yer değiştirme komşuluk yapılarından daha uzak yerlerde çözümü arar.

Yol Yer Değiştirme (Path Exchange): Bu komşuluk yapısında, rasgele seçilen aynı uzunlukta iki yol yer değiştirir. Bu durumda çözüm dışında kalan iki yolda yerdeğiştirebilir. *Kararlı-Yol Yer Değiştirme* yapısı ise çözüm içinden iki yol ya da çözüm içinden ve dışından olmak üzere iki yolun yerdeğiştirmesine izin verir. Yol yer değiştirme ile oluşan yeni çözüm, diğer üç komşuluk yapısından çok daha uzak alanda çözümü arar.

Yolu Tersine Çevirme (Path Invert) : Bu komşu oluşturma işlemi 2-opt hareketi (2-opt move) olarak da adlandırılır. Gezgin satıcı türü problemlerde, özellikle toplam rotanın dolaşma maliyetini azaltmak için 2-opt hareketlerinden oluşan 2-opt yerel arama algoritması sıklıkla kullanılır. 2-opt hareketinin temelini permütasyon üzerinde rasgele iki kenarın seçilmesi, bu kenarların permütasyondan silinmesi ve arada kalan yolun ters çevrilerek yeniden varolan yollara bağlanması

oluşturur. Bu işlemin maliyeti azaltıcı olabilmesi için silinen kenarların kenar uzunluk toplamı yeni oluşan iki kenarın toplam uzunluğundan büyük olmalıdır. Örnek vererek açıklamak gerekirse; ABCDEFG permütasyon değerleri ise ve $|BC|+|EF| > |BE|+|CF|$ ise yeni permütasyon yol ter çevirme yapısı kullanılarak ABEDCFG olur.

Modellerimizde bu yapı, permütasyonun çözümü içeren kısmına uygulanmıştır. Çözümün başından başlanarak, ilk başarılı 2-opt hareketi uygulanarak komşu bir çözüm elde edilmiştir.

4.1.3 OP için İççe İndirgenmiş-DKA Modeli ve İşleyişi

DKA algoritmaları Bölüm 3.1’de de açıklandığı gibi iki temel kısımdan oluşur: Sarsma ve Yerel Arama. Yerel Arama kısmının olmadığı DKA algoritmaları İndirgenmiş-DKA (İ-DKA) olarak adlandırılmıştır. OP için önerdiğimiz DKA tabanlı modelimiz İ-DKA algoritmasının iççe kullanılması şeklinde tasarlanmıştır. Başka bir ifade ile İ-DKA, DKA’nın içinde yerel arama olarak kullanılmıştır.

```

Prosedür İççe İndirgenmiş-DKA()
  Rasgele bir başlangıç çözümü s oluştur
  Tekrarla (durdurma şartına rastlanmadığı sürece)
    k ← 1
    yol_uzunluğu = d / 4
    yenilenme ← yok
    Tekrarla (k ≤ 2 sürece)
      Eğer (k eşitse 1)
        s' ← Kararlı_Yol_Ekle(s, yol_uzunluğu)
      Değilse Eğer (k eşitse 2)
        s' ← Kararlı_Yer_Değiştirme(s, yol_uzunluğu)
      s'' ← İ-DKA-YerelAramasi(s')
      Eğer (AmaçFonksiyon(s'') >= AmaçFonksiyon(s))
        s ← s''
        k ← 1
        yenilenme ← var
      Değilse
        k ← k+1

    Tekrarla-bitiş

    Eğer (yenilenme var ise)
      yol_uzunluğu ← yol_uzunluğu - 1
    Değilse
      yol_uzunluğu ← yol_uzunluğu + 1

  Tekrarla -bitiş
Prosedür-bitiş

```

Şekil 4.1 OP için İİ-DKA modelinin benzetimsel kodu

Böylece arama, aslında iç içe iki sarsma fonksiyonu - uzak alanda sarsma, yakın alanda sarsma- ile gerçekleştirilir. Geliştirilen İççe İ-DKA (İİ-DKA) modelinin benzetimsel kodu Şekil 4.1’de verilmiştir. Aşağıda, bu modelin Sarsma ve Yerel Arama kısımlarında kullanılan İ-DKA yönteminin detayları anlatılmıştır.

Yerel Arama : Yerel aramada üç komşuluk yapısı kullanılmıştır: Kararlı-Düğüm Ekleme, Kararlı-Düğüm Yer Değiştirme ve Yol-Tersine Çevirme. Yol Tersine Çevir yapısı diğer iki yapının arka arkaya EŞİK_DEĞER kadar çözümü güncellemediği zaman uygulanır. Maliyeti azaltıcı rol oynayarak Kararlı-Düğüm Ekleme yapısının çalışmasına imkan tanır. İİ-DKA modelinin kullandığı yerel arama metodunun benzetimsel kodu Şekil 4.2’de verilmiştir.

```

Prosedür İ-DKA-YerelAramasi (permutasyon p)
p permütasyonunu s başlangıç çözümü olarak kullan
Tekrarla durdurma şartına rastlanmadığı sürece
    sayac_yenilenme  $\leftarrow$  0
    k  $\leftarrow$  1
    Tekrarla k  $\leq$  3 sürece
        Eğer (k = 1)
            s'  $\leftarrow$  Kararlı- Düğüm Ekle(s)
        Değilse Eğer (k = 2)
            s'  $\leftarrow$  Kararlı- Düğüm Yer Değiştirme(s)
        Değilse Eğer (k = 3)
            s'  $\leftarrow$  Yol Tersine Çevir(s)
        Eğer (AmaçFonksiyon(s')  $\geq$  AmaçFonksiyon (s))
            s  $\leftarrow$  s'
            k  $\leftarrow$  1
            sayac_yenilenme  $\leftarrow$  0
        Değilse
            sayac_yenilenme  $\leftarrow$  sayac_yenilenme +1
            Eğer (k eşit 2) ve (sayac_yenilenme  $\leq$  EŞİK_DEĞER)
                k  $\leftarrow$  1
            Değilse
                k  $\leftarrow$  k+1
    Tekrarla-bitiş
Tekrarla-bitiş
Prosedür - bitiş

```

Şekil 4.2- İİ-DKA modelinin İ-DKA Yerel Arama benzetimsel kodu

Sarsma: Genel olarak sarsma işlemi çözümü bozmak amaçlı kullanılır. İİ-DKA modelinde Kararlı Yol Ekleme ve Kararlı Yol Yer Değiştirme yapılarıyla çözüm, belli bir başlangıç uzunluğundan başlayan yolların eklenmesi veya yer değiştirmesiyle bulunduğu yerden uzak yerlere gönderilir. Kararlı yapıları kullanarak çözümü bozmak garanti altına alınır. Yol uzunluğu başlangıçta çözüm boyutunun dörtte biri (d/4) büyüklüğü olarak atanır. Sarsma fonksiyonundan sonra gelen Yerel

Arama'daki sonuca göre Sarsma fonksiyonun kullanacağı yol uzunluğu değişir. Eğer Yerel Arama başlangıçtan daha iyi bir yer buldu ise yol uzunluğu bir eksilerek o yerin çevresini araştırmak için bir fırsat daha verir. Yerel Aramadan çıkan sonuç başlangıçtan daha iyi yer bulamadı ise yol uzunluğunu bir artırarak çözümü daha uzak yerlerde araması sağlanır. Yol uzunluklarının alabilecekleri sınır değerler $[2, d/4]$ dir.

4.1.4 İİ-DKA İçin Yan Modeller

OP problemi için önerilen modelin avantajlarını/dezavantajlarını daha iyi görebilmek için aşağıda detayları verilen iki yan-model geliştirilmiştir. Bu modeller:

Değişken Komşu Yerel Aramalı Değişken Komşu Arama Modeli (YA-DKA) : Bu modelde DKYA, DKA algoritmasında yerel arama metodu olarak kullanılmıştır. DKYA kendine gelen çözümün tanımladığı komşuluk yapısı içerisindeki tüm komşu çözümlerine bakarak en iyi çözümü geri döndürür. Önerdiğimiz model olan İİ-DKA'da yerel arama işlemi komşuluk alanındaki tek bir çözüme bakarak arama yapar. YA-DKA modeli ile tüm komşulara bakacak bir yerel aramanın gerekli olup olmadığı gözlemlenmek istenmiştir. Modelimizde DKYA için Düğüm Ekle ve Düğüm Yer Değiştir komşuluk yapıları kullanılırken, Sarsma fonksiyonu için Yol Ekle ve Yol Yer Değiştir komşuluk yapılarını kullanılmıştır.

İndirgenmiş-DKA Modeli (İ-DKA): Bu model İndirgenmiş-DKA, tanımlanan komşuluk yapıları içinde yerel arama yapmadan çözüm kümesini tarar Bu modelin tasarlanmasındaki amaç, İççe İ-DKA kullanımının gerekli olup olmadığını gözlemlemektir. Diğer modellerdeki dört komşuluk yapısı (Düğüm Ekle, Düğüm Yer Değiştir, Yol Ekle, Yol Yer Değiştir) bu model için de kullanılmıştır.

Geliştirilen bu modellerde kullanılan komşuluk yapıları ve sırası Çizelge 4.1'de özetlenmiştir.

Çizelge 4.1 OP için DKA modelleri ve kullanılan komşuluk yapıları

	Sarsma	Yerel Arama
İ-DKA	1.Düğüm Ekleme 2.Düğüm Yer Değiştirme 3.Yol Ekleme 4.Yol Yer Değiştirme	-
YA-DKA	1. Yol Ekleme 2. Yol Yer Değiştirme	1.Düğüm Ekleme 2 Düğüm Yer Değiştirme
İİ-DKA-B	1.Yol Ekleme 2.Yol Yer Değiştirme	1. Düğüm Ekleme 2. Düğüm Yer Değiştirme
İİ-DKA	1.Kararlı-Yol Ekleme 2.Kararlı-Yol Yer Değiştirme 3.Yol-Ters Çevirme	1. Kararlı-Düğüm Ekleme 2. Kararlı-Düğüm Yer Değiştirme

4.2 Takım Oryantiring Problemi için Değişken Komşu Arama Modeli

Bu bölümde TOP için geliştirilen İççe-İndirgenmiş Değişken Komşu Arama modeli, problem gösterim şekli, komşuluk yapıları ve modelin işleyişi başlıkları altında anlatılacaktır

4.2.1 Problem Gösterimi

Problemin gösterim şekli problemde belirtilen araç sayısının bir fazlası, $(m+1)$ adet kontrol noktalarından oluşan listedir. İlk m liste aday çözümü ifade eder, yani listenin birinci elemanı başlangıç kontrol noktası, sonucu elemanı ise bitiş kontrol noktasını içerir. Her bir listenin maliyeti T_{maks} değerinin altındadır. Sonuncu, $(m+1)$. liste ise çözüm dışında kalan kontrol noktalarını içinde barındırır.

4.2.2 Komşuluk Yapıları

OP’de olduğu gibi TOP’da da komşuluk yapıları düğüm ekleme ve düğüm yerdeğiştirmeye dayanır. TOP’ da ekleme ve yerdeğiştirme yerleri bulunurken iki yöntem kullanılmıştır.

- *En ucuz maliyetle ekleme*: Bu yöntemde bir kontrol noktası bir rotaya eklenirken maliyet artışını en aza indirecek şekilde kontrol noktasının ekleme yeri tespit edilir.

- *En az kayıpla silme*: Fazla maliyetten dolayı rotadan bir kontrol noktasının silinmesi gerektiği zaman, maliyeti geçerli duruma getirecek ve rota kazancını en az kayba uğratacak kontrol noktasının silinmesi gerçekleştirilir.

Çözüm birden fazla rotadan oluştuğundan dolayı düğüm ekleme/yerdeğiştirme aşağıda açıklandığı gibi iki şekilde yapılır:

Dışarıdan Düğüm Ekle Yerdeğiştir: Bu komşuluk yapısı rota kazancını artıracak komşu çözümü bulmayı hedefler. Çözüm dışında kalan kontrol noktalarından oluşan listeden rasgele bir kontrol noktası seçilir. Bu kontrol noktası, rasgele seçilen bir çözüm listesine en ucuz maliyetli eklenme yöntemi kullanılarak eklenmeye çalışılır. Kontrol noktası eklenmiş yeni rotada, maliyet kısıtı aşılmamış ise kazancı artırıcı şekilde komşu çözüm bulunduğundan arama sonlanır. Eğer maliyet kısıtı aşılmışsa en az kayıpla silme yöntemi çalıştırılarak rotadan silinecek en uygun kontrol noktası aranır. Uygun kontrol noktası bulunursa işlem başarı ile sonlanır. Eğer bulunamazsa rota ilk haline geri döner.

Rotalar Arası Düğüm Ekle Yerdeğiştir: Bu komşuluk yapısı ile çözümün toplam kazancı aynı kalarak, seçilen rotaların maliyetlerinin azaltılması hedeflenmiştir. Bu yapıda komşu çözüm oluşması için, çözümü oluşturan iki rota ($rota_1$, $rota_2$) rasgele seçilir. Böyle bir seçimde aynı rota iki kere seçilebileceği gibi, farklı iki rota da seçilebilir. Aynı rotalar seçilmiş ise o rota içinden seçilen rasgele bir kontrol noktasının en ucuz maliyetle eklenme yeri tespit edilerek rotanın maliyetinin azalması hedeflenir. Eğer seçilen rotalar farklı ise, ilk seçilen rota olan $rota_1$ 'den rasgele bir kontrol noktası seçilir ve $rota$ 'ye en ucuz maliyetle eklenir. Eklenme, maliyet kısıtı aşmadan gerçekleşmiş ise, $rota_1$ 'in maliyeti düştüğünden dolayı komşu çözüm oluşmuş kabul edilir. Eğer maliyet kısıtı geçilerek düğüm $rota_2$ 'ye eklenmiş ise en az kayıpla silme yöntemi çalıştırılarak $rota_2$ 'den en uygun silinecek kontrol noktası belirlenir ve $rota_1$ 'e en ucuz maliyetle ve maliyet kısıtı aşmaksızın eklenme gerçekleştirilmeye çalışılır. Bu durumda rotalar arası düğüm değiş-tokuşu

yapılmış olur. İşlemin başarılı olabilmesi için en az bir rotanın maliyetinin azalmış olması gerekir.

Yolu Tersine Çevirme: Bu yapı OP için önerilen İİ-DKA modelinde kullanılan komşuluk yapısının aynısıdır (Bkz: Bölüm 4.1.2) Burada çözümü oluşturan rotalar birden fazla olduğu için yolu tersine çevirme yapısı her birine ayrı ayrı uygulanır.

4.2.3 Modelin İşleyişi

OP için önerilen İİ-DKA modelini TOP için yeniden uyarlanmıştır. Model iç içe yerleştirilmiş İ-DKA algoritmasından oluşmaktadır. TOP için önerilen İİ-DKA modelinin benzetimsel kodu Şekil 4.3'te verilmiştir. Benzetimsel kodda yer alan Sarsma ve İ-DKA Yerel Arama yöntemleri ve işleyişleri aşağıda açıklanmıştır.

Sarsma : Bu yöntemin amacı, var olan çözümü bozarak çözüm alanının farklı bir noktasından yerel arama metodunu başlatmaktır. OP için geliştirilen Sarsma metodundan farklı olarak, önceden belli olan bir yıkım oranı nisbetinde çözümüden rasgele seçilen kontrol noktaları çözümüden silinir. Yıkım oranı içindeki İ-DKA yerel arama yönteminden çıkan sonuca göre iki farklı değer alır. Yerel aramadan çıkan sonuç iyi ise yıkım KÜÇÜK_ORAN' da yapılır. BÜYÜK_ORAN ise yerel aramanın sonucu iyileştirememesi durumunda kullanılır.

```

Prosedür İÇİÇE_İndirgenmiş-DKA()
  Rasgele üretilmiş bir çözümü başlangıç çözümü s olarak seç
  Tekrarla (durdurma şartına rastlanmadığı sürece)
    yikim_orani ← BÜYÜK_ORAN
    s' ← Sarsma(s,yikim_orani)
    s'' ← İ-DKA-YerelAramasi(s')
    Eğer (AmaçFonksiyon(s'') >= AmaçFonksiyon(s))
      s ← s''
      yikim_orani ← KÜÇÜK_ORAN
    Değilse
      yikim_orani ← BÜYÜK_ORAN
  Tekrarla -bitiş
Prosedür-bitiş

```

Şekil 4.3 TOP için İÇİÇE-İndirgenmiş-DKA benzetimsel kodu

Yerel Arama: Yerel aramada üç komşuluk yapısı kullanılmıştır: Dışardan Düğüm Ekle Yerdeğiştir, Rotalar Arası Düğüm EkleYerdeğiştir ve Yolu Tersine Çevirme. İlk iki komşuluk yapısı TEKRAR_SAYISI kadar komşu ürettikten sonra, bir sonraki komşuluk alanına geçip geçmemeye karar verir. Yolu Tersine Çevir yapısı ise diğer iki yapının arka arkaya EŞİK_DEĞER kadar çözümü güncellemediği zaman uygulanır. İİ-DKA modelinin kullandığı yerel arama metodunun benzetimsel kodu Şekil 4.4’te verilmiştir.

```

Prosedür İ-DKA-YerelAramasi (permutasyon p)
  p permütasyonunu s başlangıç çözümü olarak kullan
  Tekrarla durdurma şartına rastlanmadığı sürece
    sayac_yenilenme ← 0
    k ← 1

    Tekrarla k ≤ 3 sürece
      Eğer (k = 1)
        s' ← DışarıdanDüğümEkleYerdegistir(s,TEKRAR_SAYISI)
      Değilse Eğer (k = 2)
        s' ← RotalarArasıDüğümEkleYerdeğiştir(s,TEKRAR_SAYISI)
      Değilse Eğer (k = 3)
        s' ← YoluTersineÇevir(s)
      Eğer (AmaçFonksiyon(s') >= AmaçFonksiyon (s))
        s ← s'
        k ← 1
        sayac_yenilenme ← 0
      Değilse
        sayac_yenilenme ← sayac_yenilenme +1
        Eğer (k = 2) ve (sayac_yenilenme ≤ EŞİK_DEĞER)
          k ← 1
        Değilse
          k ← k + 1

    Tekrarla-bitiş
  Tekrarla-bitiş
Prosedür - bitiş

```

Şekil 4.4 TOP için İİ-DKA modelinin İ-DKA Yerel Arama benzetimsel kodu

4.3 Deneysel Sonuçlar

Bu bölümde önerilen İİ-DKA modelinin OP ve TOP karşılaştırmalı test problemleri ile yapılan deneylerinin sonuçları sunulacaktır.

Deneylerden çıkan sonuçlar, *en iyi çözümünü bulma süresine* (CPU zamanı - saniye), *en iyi çözümü bulana kadar hesaplanan amaç fonksiyon sayısına*, *en iyi çözümün bağıl yüzde hatası* ve *ortalama bağıl yüzde hatası* kriterlerine göre analiz

edilmiştir. Bağlı yüzde hata, algoritmanın tekrarlı çalıştırma sırasında bilinen en iyi sonucu bulup bulamadığını ve bilinen en iyi çözüme göre eldeki sonucun yüzdelik hata oranını gösterir. Bağlı yüzde hatasının 0 olması tekrarlı çalıştırmalarda en az bir defa bilinen en iyi sonucu bulduğunu, sıfırdan küçük olması durumunda ise bilinen en iyi sonucu daha iyileştirdiğini gösterir. Tekrarlı çalıştırmaların her biri için bilinen en iyi sonuca ne kadar yüzdelik hata ile yaklaştığı bulunup, bu sonuçların ortalaması hesaplanarak ortalama bağlı yüzde hata bulunmuş olunur. Ortalama bağlı yüzde hata algoritmanın sağlamlık derecesini gösterir. Her bir çalıştırmada rasgele bir yerden problem çözümüne başlandığından, ortalama bağlı yüzde hatasının küçük değere çıkması algoritmanın her şekilde bilinen en iyi sonucu çok az hata ya da hatasız bulduğunu gösterir. Amaç fonksiyon sayısı ise arama alanında en iyi çözümü oluşturana kadar amaç fonksiyonu kaç defa hesaplandığını gösterir. Bu kriter algoritma hakkında yazılım/donanımdan bağımsız olarak bilgi verdiğinden önemli bir kriterdir. Ayrıca farklı makinelerde test edilmiş problem sonuçlarını karşılaştırma adına süre kriterinden daha kesin bir sonuç elde edilebilir.

Belirtilen kriterlere göre İİ-DKA modelinin OP ve TOP için deneysel sonuçları iki bölümde sunulmuştur.

4.3.1 OP için İİ-DKA Modelinin Deneysel Sonuçları

OP için geliştirilen İİ-DKA modeli Bölüm 2.4’de açıklanan 103 problem üzerinde test edilmiştir. İİ-DKA modeli yan modelleri ile beraber Intel P4 2.6 GHz işlemci ve 512 MB bellek içeren bir bilgisayar üzerinde 10 defa çalıştırılmıştır. Modellerin durdurma şartı 100.000 defa amaç fonksiyonunun hesaplanması olarak ayarlanmıştır.

DKA tabanlı geliştirdiğimiz dört modelin yukarıda belirttiğimiz kriterlere göre karşılaştırılması Çizelge 4.2’de verilmiştir. İ-DKA, YA-DKA ve İİ-DKA-B aynı yapıdaki komşulukları kullandığından karşılatırmayı önce bu üçü arasında yapmak doğru olur. Sonuçlara bakıldığında İ-DKA modelinde yerel minimumlar üzerinde yoğunlaşma olmadığından hızlı bir yöntem olmasına rağmen istenilen kaliteli sonuçlar bulanamamıştır. Ayrıca YA-DKA ve İİ-DKA-B en iyi sonuçları bulmakla beraber, İİ-DKA-B yapılan işi işlem süresi (CPU) bakımından daha az zamanda ve tekrarlı çalıştırmalarda daha az hata ile bulduğu görülmektedir. Bu yüzden İİ-DKA-B

modeli diğer iki modelle göre daha iyi modeldir. Algoritma yapısı aynı olan İİ-DKA modelinde ise İİ-DKA-B modelinden tek farkı kullandığı komşuluk yapısının çözüme odaklı olmasıdır. Kararlı yapılar dediğimiz bu yapılarla aynı algoritma çalıştırıldığında aynı kalitedeki sonuçları ortalama olarak yarı yarıya daha kısa sürede bulduğu gözlemlenmiştir. Buna göre DKA tabanlı modellerimiz arasından en iyi sonucu İİ-DKA modelimiz ve modelle birlikte kullanılan kararlı komşuluk yapıları vermiştir. Toplam 103 OP problemi için İİ-DKA modelinden elde edilen sonuçlar EK-B’de her bir problem için listelenmiştir.

Çizelge 4.2 OP için geliştirilen DKA tabanlı modellerin PK (1,2,3 ve 1_1) için CPU(saniye), yüzdelik ve ortalama yüzdelik bağıl hata kriterlerine göre karşılaştırılması.

	CPU				Yüzdelik Bağıl Hata				Ort Yüzdelik Bağıl Hata			
	i-DKA	YA-DKA	ii-DKA -B	ii-DKA	i-DKA	YA-DKA	ii-DKA -B	ii-DKA	i-DKA	YA-DKA	ii-DKA -B	ii-DKA
PK 1	0.91	6.86	2.79	0.54	1.16	0.00	0.00	0.00	9.06	0.50	0.12	0.16
PK 2	1.04	0.69	0.15	0.76	1.02	0.00	0.00	0.00	4.65	0.52	0.00	0.00
PK 3	1.05	9.99	0.89	0.64	0.86	0.00	0.00	0.00	6.50	0.22	0.04	0.06
PK 1_1	0.85	6.58	0.62	0.63	1.05	0.00	0.00	0.00	9.39	0.52	0.05	0.06
Ortalama	0.96	6.03	1.11	0.64	1.02	0.00	0.00	0.00	7.40	0.44	0.05	0.07

İİ-DKA modellerinden elde edilen en iyi sonuçlar aşağıda listelenen modeller kullanılarak elde edilen en iyi sonuçlarla da Çizelge 4.3’de karşılaştırılmıştır.

- CGW: Sezgisel algoritma [Chao et al., 1996] (PK 1, 2, 3, 1_1, 5 ve 6 üzerinde deneyler yapılmıştır)
- GA : Genetik algoritma [Tasgetiren and Smith, 2000] (PK 1, 2, 3, 1_1 üzerinde deneyler yapılmıştır)
- KKO: Karınca kolonileri optimizasyonu [Liang et al., 2002] (PK 1, 2, 3, 1_1 üzerinde deneyler yapılmıştır)
- KYA: Kılavuzlu yerel arama [Vansteenkeweg et al., 2009] (PK 1, 2, 3, 5 ve 6 üzerinde deneyler yapılmıştır)

Çizelge 4.3 Önerilen İİ-DKA modelinin PK 1,2,3 ve 1_1 için literatür sonuçlarıyla karşılaştırılması

	Modeller	PK 1	PK 2	PK 3	PK 1-1	Ort
Yüzdelik Bağlı Hata	CGW	0.00	0.00	0.00	0.11	0.03
	GA	0.00	0.00	0.00	0.00	0.00
	KKO	0.00	0.00	0.00	0.00	0.00
	KYA	3.17	1.36	0.65	-	1.73
	İİ-DKA	0.00	0.00	0.00	0.00	0.00
Std. Sap.	CGW	-	-	-	-	-
	GA	1.29	2.63	3.38	1.36	2.17
	KKO	0.59	1.66	0.72	0.48	0.86
	KYA	-	-	-	-	-
	İİ-DKA	0.29	0.00	0.45	0.22	0.24

Problem kümesi 1, 1_1, 2 ve 3 için modeller yüzdelik bağlı hatası ve tekrarlı çalışmalar sonunda amaç fonksiyon değerinin standart sapması kriterlerine göre karşılaştırılmıştır. İİ-DKA modeli, GA ve KKO modeli gibi, bilinen en iyi sonuçları bulmuştur. KYA her problem kümesinde iyi bilinen sonuçlara belli bir hata miktarı ile yaklaşırken CGW sadece problem kümesi 1_1 içinde bazı problemlerde iyi bilinen çözümü bulamadığı görülür. Tekrarlı çalıştırma sonucu elde edilen amaç fonksiyonun standart sapma değeri bakımından ise İİ-DKA'nın en düşük değeri olarak çözüm kalitesi yönünden test edilen problem kümesi için diğer modellerden daha iyi performans sergilediği söylenebilir.

İİ-DKA modelinin problem kümesi 5 ve 6 için literatürdeki sonuçlarla karşılaştırma listesi Çizelge 4.4 ve Çizelge 4.5'te verilmiştir. Modelimiz literatürdeki bütün en iyi sonuçları bulmakla beraber, CGW'nin toplam 12 sonucunda, KYA'nın ise 20 sonucunda iyileştirme yapmıştır. Literatüre kazandırılan yeni sonuçlar Çizelge 4.6'da çözüm yollarıyla beraber verilmiştir.

Çizelge 4.4 İİ-DKA modelinin PK 5 için literatür sonuçlarıyla karşılaştırılması

T _{maks}	CGW		KYA		İİ-DKA		CGW	KYA
	Kazanç	CPU	Kazanç	CPU	Kazanç	CPU		
5	10	1.1	10	0	10	0.2		
10	40	0.5	40	0.2	40	0.2		
15	120	4.3	120	1.3	120	1.0		
20	195	6.2	175	1.7	205	1.2	+	+
25	290	73.4	290	1.6	290	1.1		
30	400	54.8	400	2	400	1.7		
35	460	32.4	465	2.8	465	2.3	+	
40	575	98.9	575	3.2	575	3.4		
45	650	58.1	640	2.9	650	5.8		+
50	730	68.1	710	3	730	9.8		+
55	825	65.2	825	3.7	825	15.9		
60	915	84.6	905	4.1	915	14.3		+
65	980	82.2	935	3.3	980	22.0		+
70	1070	119	1070	3.5	1070	21.0		
75	1140	116.7	1140	4.4	1140	22.6		
80	1215	108.9	1195	3.1	1215	24.3		+
85	1270	132.5	1265	2.3	1270	34.1		+
90	1340	502.4	1300	2.3	1340	37.3		+
95	1380	467.1	1385	2.2	1395	31.8	+	+
100	1435	128.6	1445	2.1	1465	41.5	+	+
105	1510	316.3	1505	2.2	1520	48.2	+	+
110	1550	469.9	1560	2	1560	37.5	+	
115	1595	474.6	1580	1.7	1595	38.3		+
120	1635	358	1635	1.2	1635	51.4		
125	1655	268.9	1665	1.3	1670	30.0	+	+
130	1680	32.1	1680	0.7	1680	20.4		
Önceki çalışmaların İİ-DKA modeli ile karşılaştırma sonucu							7+ 0-	13+ 0-

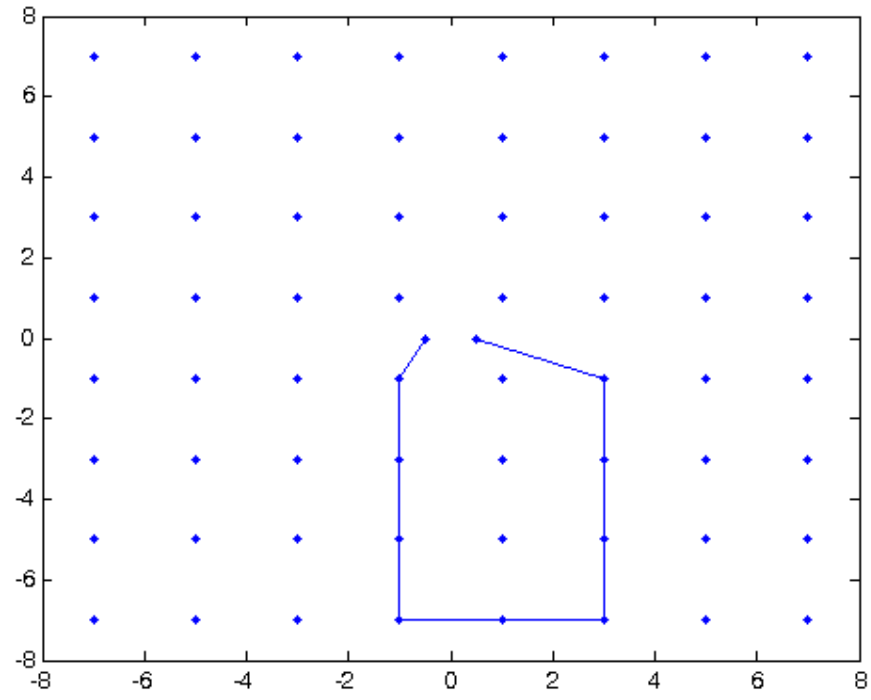
Çizelge 4.5 İİ-DKA modelinin PK 6 için literatür sonuçlarıyla karşılaştırılması

T _{maks}	CGW		KYA		İİ-DKA		CGW	KYA
	Kazanç	CPU	Kazanç	CPU	Kazanç	CPU		
15	96	13	96	0.8	96	0.4		
20	294	27.9	294	2	294	0.4		
25	390	238.9	390	2.5	390	1.2		
30	474	74.5	474	4.5	474	3.9		
35	570	139.8	552	2.3	576	7.9	+	+
40	714	137.9	702	2.5	714	13.8		+
45	816	205	780	3.2	816	18.2		+
50	900	231.6	888	2.1	900	25.1		+
55	984	246.2	972	1.4	984	16.5		+
60	1044	264.8	1062	2.3	1062	27.5	+	
65	1116	232.6	1110	1.8	1116	38.1		+
70	1176	231	1188	1.5	1188	25.8	+	
75	1224	223.1	1236	2.1	1236	40.7	+	
80	1272	212.3	1260	1.5	1278	26.4	+	+
Önceki çalışmaların İİ-DKA modeli ile karşılaştırma sonucu							5+ 0-	7+ 0-

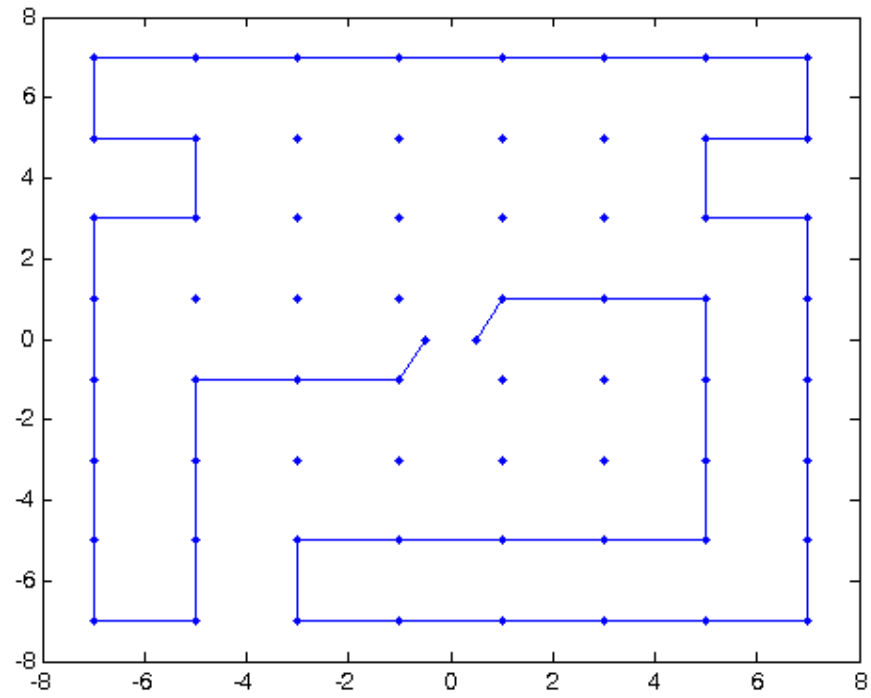
Çizelge 4.6 OP için İİ-DKA modelinin geliştirdiği yeni sonuçlar

PK	T _{maks}	T _d	Kazanç	Bulunan Rota
5	20	19.81	205	1 29 28 27 26 34 42 43 44 45 66
5	95	94.24	1395	1 29 21 13 12 11 10 2 3 4 5 6 7 15 16 8 9 17 25 33 41 49 57 65 64 56 55 63 62 61 60 59 58 50 42 34 26 18 19 27 35 43 51 52 53 54 46 38 66
5	100	99.81	1465	1 29 28 27 19 11 3 2 10 18 26 34 42 50 58 59 51 52 60 61 62 63 64 65 57 49 41 33 25 17 9 8 7 6 5 4 12 13 14 15 16 24 32 40 48 56 55 54 53 45 66
5	105	104.92	1520	1 30 31 32 24 16 15 14 13 12 11 10 2 3 4 5 6 7 8 9 17 25 33 41 40 48 49 57 65 64 56 55 63 62 54 53 61 60 52 51 59 58 50 42 43 35 34 26 18 19 27 28 29 66
5	125	124.92	1670	1 38 39 31 32 33 25 24 16 17 9 8 7 15 23 22 21 13 14 6 5 4 12 11 3 2 10 18 19 20 28 36 35 27 26 34 42 50 58 59 60 61 62 63 64 65 57 49 41 40 48 56 55 47 46 54 53 52 51 43 44 45 37 66
6	35	34.86	576	1 3 6 10 15 21 28 36 43 35 27 34 26 33 32 24 31 23 30 38 45 51 56 64
6	80	79.2	1278	1 3 6 9 14 10 15 21 28 36 43 49 54 58 53 48 42 35 27 20 26 34 41 33 40 32 25 19 13 18 24 17 12 8 4 7 11 16 22 29 37 44 50 55 51 45 38 30 23 31 39 46 52 57 61 63 64

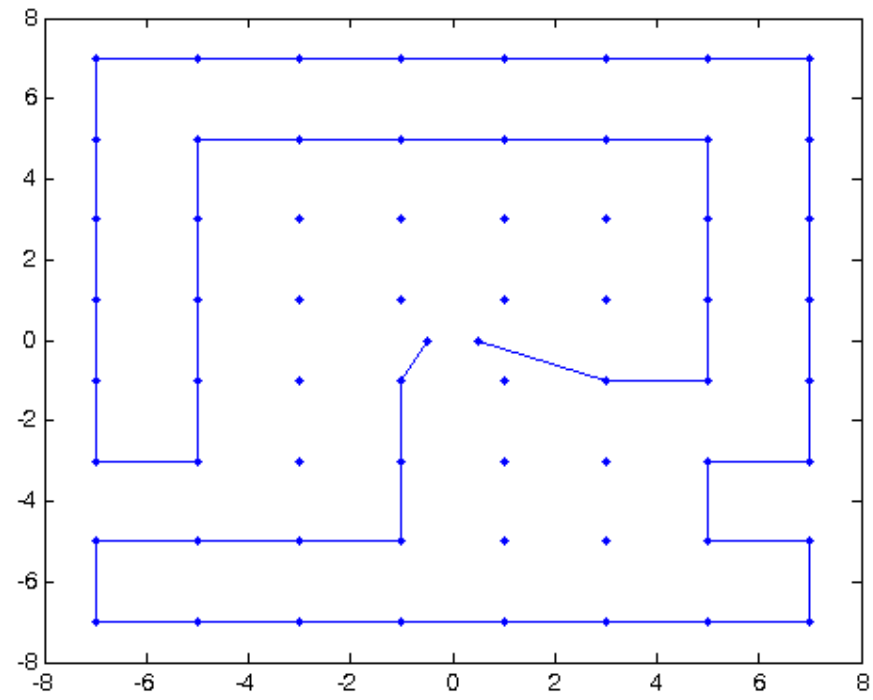
Bulunan yedi yeni çözümün grafiksel olarak gösterimi ise Şekil 4.5- Şekil 4.11 arasında verilmiştir.



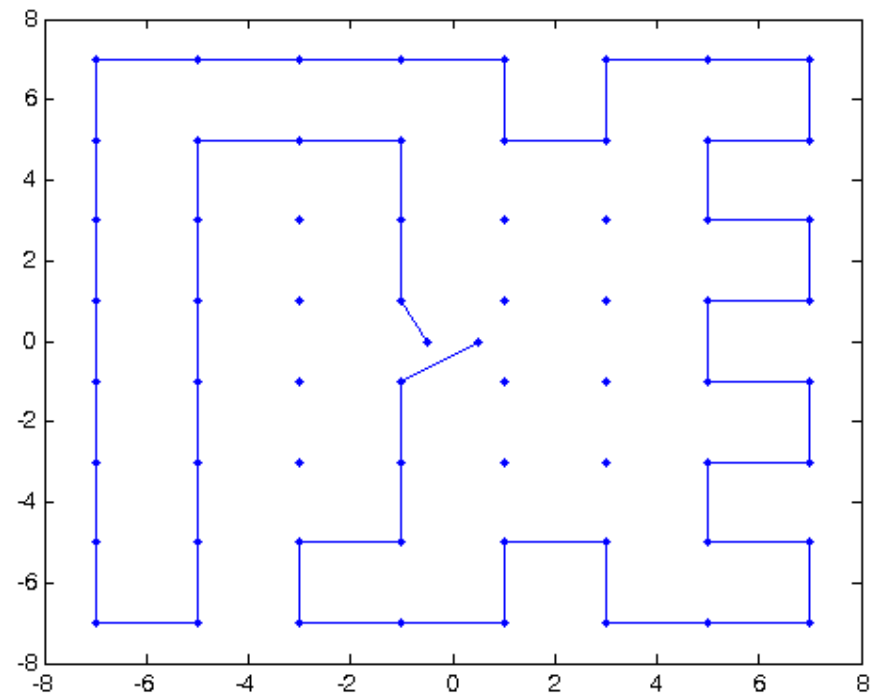
Şekil 4.5 PK 5’te $T_{maks} = 20$ için İİ-DKA ile elde edilen yeni çözüm



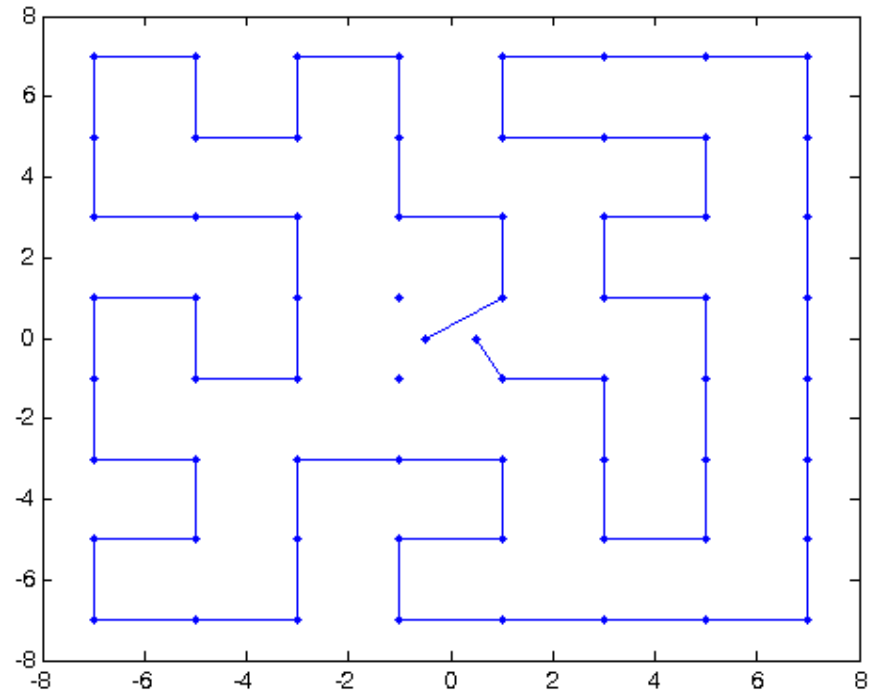
Şekil 4.6 PK 5 ‘te $T_{maks} = 95$ için İİ-DKA ile elde edilen yeni çözüm



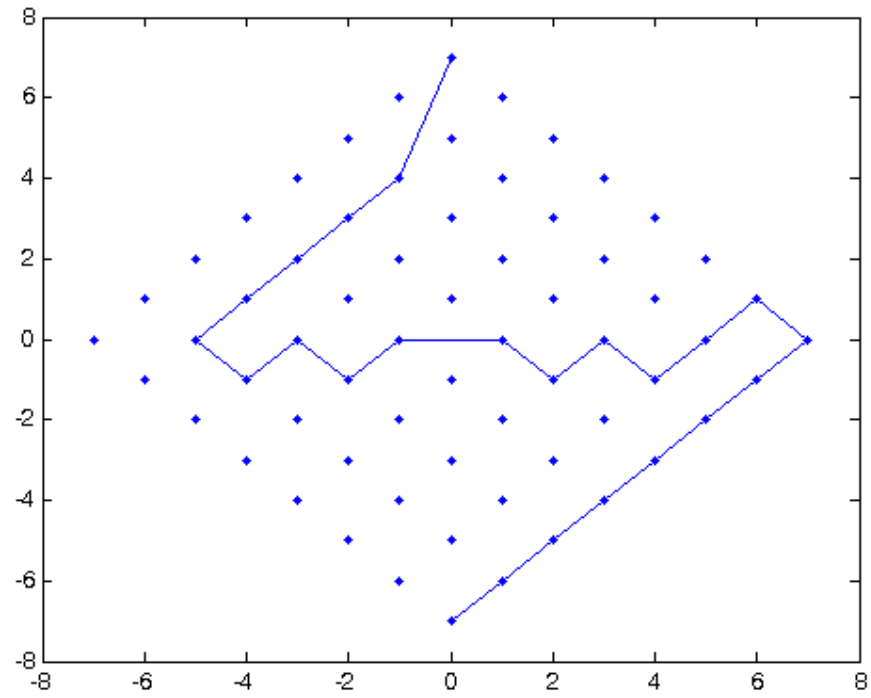
Şekil 4.7 PK 5 'te $T_{maks}=100$ için İİ-DKA ile elde edilen yeni çözüm



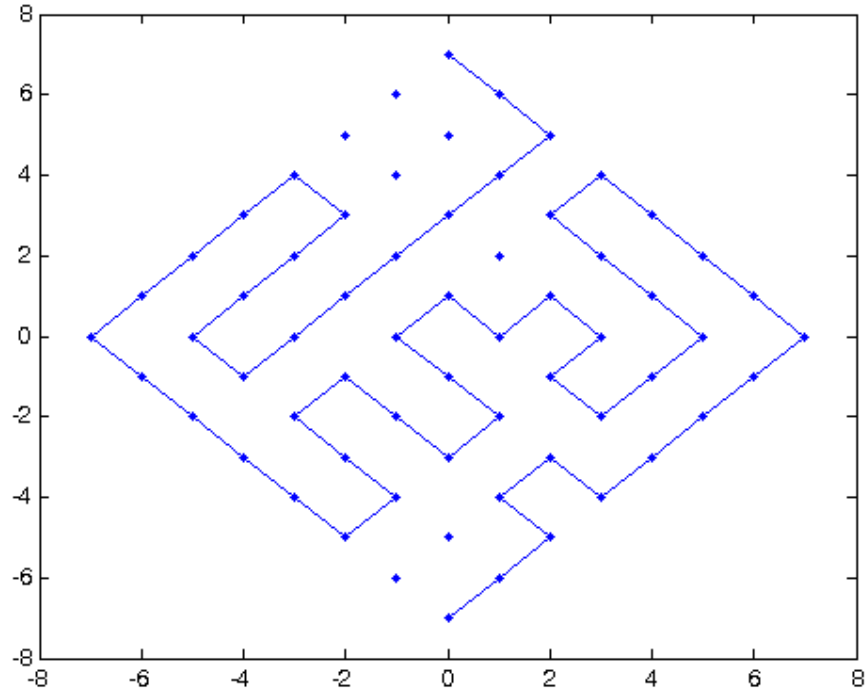
Şekil 4.8 PK 5 'te $T_{maks}=105$ için İİ-DKA ile elde edilen yeni çözüm



Şekil 4.9 PK 5 'te $T_{maks} = 125$ için İİ-DKA ile elde edilen yeni çözüm



Şekil 4.10 PK 6' da $T_{maks} = 35$ için İİ-DKA ile elde edilen yeni çözüm



Şekil 4.11 PK 6 ‘da $T_{maks}=80$ için İİ-DKA ile elde edilen yeni çözüm

4.3.2 TOP için İİ-DKA Modelinin DeneySEL Sonuçları

TOP için önerdiğimiz İİ-DKA modeli Bölüm 2.4’de açıklanan 350 karşılaştırmalı TOP problemi üzerinde (PK 1, 2, 3, 4, 5, 6 ve 7) test edilmiştir. Model, Intel P4 2.6 GHz işlemci ve 1 GB bellek içeren bir bilgisayar üzerinde 10 defa çalıştırılmış, modelin bulduğu en iyi çözüm kendini arka arkaya 300 defa yenilemediği takdirde algoritma sonlanmıştır.

Toplam 350 TOP problemi için İİ-DKA modelinden elde edilen sonuçlar EK-C’de her bir problem için listelenmiş, Çizelge 4.7’de ise yüzdelik bağıl hata, standart sapma ve CPU kriterlerine göre özetlenmiştir. Çizelgedeki özet sonuçlar her bir problem kümesi/araç sayısı için ayrı ayrı verilmiş olup, problem kümeleri de içerdiği kontrol sayılarına göre sıralanmıştır. Buna göre PK 1, 2 ve 3’de yer alan 138 problemin bilinen en iyi sonuçları İİ-DKA modeli tarafından bulunmuştur (Bkz % bağıl hata ort bakınız). PK 5 ve 6 için ortalamada %0.1 gibi çok küçük bağıl hata oranları ile 98 problemten 6’sını bulamamıştır. PK 4 ve 7’de ise %0.3’lere çıkan bağıl hatalarla 114 problemin ancak 77’si için bilinen en iyi sonuçlara ulaşabilmiştir. Performans açısından en fazla zamanı İİ-DKA PK4 için harcamıştır.

Çizelge 4.7 İİ-DKA modelinin toplam 7 problem kümesi için performans analizi

Problem Kümesi	Araç Sayısı	Toplam Problem Sayısı	İİ-DKA			
			% Bağıl Hata Ort.	Std. Sapma Ort.	CPU Ort.	Bilinen En iyi Değeri Bulma Sayısı
1	2	17	0.00	0.72	6.73	17
	3	16	0.00	0.00	3.65	16
	4	15	0.00	0.00	1.65	15
		48	0.00	0.25	4.12	48
2	2	11	0.00	0.00	1.06	11
	3	11	0.00	0.00	0.44	11
	4	11	0.00	0.00	0.11	11
		33	0.00	0.00	0.54	33
3	2	20	0.00	1.98	11.94	20
	3	19	0.00	0.33	4.80	19
	4	18	0.00	0.00	3.01	18
		57	0.00	0.81	6.74	57
5	2	25	0.09	4.53	31.48	22
	3	25	0.01	1.73	24.55	24
	4	24	0.00	2.31	18	24
		74	0.03	2.87	24.77	70
6	2	11	0.14	7.73	28.89	9
	3	8	0.00	1.44	29.15	8
	4	5	0.00	0.00	11.32	5
		24	0.07	4.02	25.32	22
4	2	20	0.33	17.41	97.53	9
	3	19	0.19	17.34	101.84	11
	4	17	0.21	16.2	52.17	13
		56	0.25	17.02	85.22	33
7	2	20	0.03	18.1	34.49	17
	3	19	0.22	9.58	30.56	13
	4	19	0.16	9.66	20.41	14
		58	0.13	12.54	28.59	44
Toplam		350	307			

İİ-DKA modelinden elde edilen en iyi sonuçlar aşağıda listelenen modellerin elde ettikleri en iyi sonuçlarla Çizelge 4.8’de karşılaştırılmıştır.

- CGW: Sezgisel algoritma [Chao et al., 1996]
- TMH: Tabu Arama Tabanlı Çözüm Yöntemi [Tang and Miller-Hooks, 2005]
- GTP: Cezalı Tabu Arama [Archetti et al., 2007]
- GTF: Cezasız Tabu Arama [Archetti et al., 2007]
- HT-DKA: Hızlı Tabulu Değişken Komşu Arama [Archetti et al., 2007]
- YT-DKA: Yavaş Tabulu Değişken Komşu Arama [Archetti et al., 2007]

- Ase KKO: Karınca Kolonileri Optimizasyonu [Ke et al., 2008]
- MA: Memetik Algoritma [Bouly et al., 2008]
- KYA: Kılavuzlu Yerel Arama [Vansteenwegen et al., 2009]

Daha önceki çalışmalarla yapılan karşılaştırma, problem kümeleri bazında iki grupta özetlenmiştir. Kontrol nokta sayısı 40'tan küçük olan problem kümesi 1,2 ve 3 A grubunda, geri kalan problem kümeleri ise B grubunda özetlenmiştir. A grubunda yer alan 138 problem için CGW, TMH ve GTP dışındaki tüm modeller her problem için en iyi sonucu bulmuşlardır. B grubunda ise İİ-DKA modelimiz en iyi sonucu bulma sayısı bakımından MA, Ase KKO ve YT-DKA'dan sonra en iyi modeldir. CPU bakımından YT-DKA ve MA elde ettiği iyi sonuçları İİ-DKA'ya göre yaklaşık yedi kat ve iki kat zaman harcayarak bulmuşlardır.

Sonuç olarak İİ-DKA çözüm kalitesi ve performans bakımından önceki çalışmalarla karşılaştırılabilir çözümler bulmuştur.

Çizelge 4.8 İİ-DKA modelinin performansının önceki çalışmalarla karşılaştırılması

		CGW	TMH	GTP	GTF	HT-DKA	YT-DKA	Ase KKO	MA	KYA	İİ-DKA
en iyi sonucu bulma sayısı	A	112	121	136	138	138	138	138	138	-	138
ort % bağıl hata		0.97	0.42	0.05	0.00	0.00	0.00	0.00	0.00	-	0.00
ort CPU		11.91	-	4.12	1.22	0.11	6.92	5.45	1.13		4.34
en iyisonucu bulma sayısı	B	88	89	124	148	150	181	182	185	68	169
ort % bağıl hata		1.98	1.10	0.41	0.19	0.17	0.05	0.56	0.47	2.00	0.12
ort CPU		565.65	361.85	103.40	157.83	21.82	279.18	26.26	68.41	10.37	42.00

5 PARÇACIK SÜRÜ OPTİMİZASYONU TABANLI ÖNERİLEN MODELLER

Bu bölümde OP ve TOP için geliştirilen Parçacık Sürü Optimizasyon (PSO) tabanlı iki model anlatılacaktır. Geliştirilen bu modeller, bildiğimiz kadarı ile OP ve TOP için geliştirilen ilk Parçacık Sürü Optimizasyon modelidir. Modellerin etkinliğini ortaya çıkarmak için modeller iki yan modelle desteklenmiştir. Geliştirilen bu ana ve yan modellerin detayları iki alt başlık altında açıklanacaktır.

5.1 Yarı Ayrık Güçlendirilmiş-PSO Modeli

Bu bölümde OP için geliştirilen Yarı Ayrık Güçlendirilmiş Parçacık Sürü Optimizasyon modeli, problem gösterim şekli ve modelin işleyişi başlıkları altında detaylı olarak anlatılacaktır.

5.1.1 Problem Gösterimi

Yarı Ayrık Güçlendirilmiş-PSO modeli ayrık problemler için geliştirilen PSO modellerinden Model 4'ü temel alır (Bkz. Bölüm 3.2.1.2). Genel olarak PSO tabanlı modellerde parçacıkların pozisyonları aday çözümü gösterir. Model 4 tabanlı bir Ayrık-PSO modellemesinde PSO'nun her iterasyonunda reel değerlerden oluşan pozisyon vektörü permütasyona çevrilerek amaç fonksiyon hesaplanır.

Yarı Ayrık Güçlendirilmiş-PSO modelinde, problem çözüm boyutu uzunluğunda reel değerlerden oluşan hız ve pozisyon vektörleri kullanılmıştır. Pozisyon vektöründen amaç fonksiyon değeri bulunmak istendiğinde Pozisyon vektörü için *en küçük pozisyon değer kuralını* [Tasgetiren et al., 2004] kullanılarak permütasyon elde edilir. Model ihtiyaç hissettiği zaman ayrık duruma dönüştürüldüğünden 'Yarı Ayrık' olarak adlandırılmıştır. En küçük pozisyon değer kuralı şu şekilde çalışır: Farz edelim, orjinal vektörümüzdeki i . indekte reel değerimiz p_i bulunuyor. Orjinal vektörümüzü sıraladıktan sonra p_i değerimiz j . indekte yer alsın. Orjinal vektörün permütasyon gösterimi i . indeksteki değere j

atanarak elde edilir. Bu kuralın işleyişi Şekil 5.1’de örnek bir pozisyon vektörü için verilmiştir.

indeks	1	2	3	4
pozisyon vektörü	2.5	5.7	0.7	1.3

indeks	1	2	3	4
sıralanmış pozisyon vektörü	0.7	1.3	2.5	5.7

indeks	1	2	3	4
Permütasyon vektörü	3	4	1	2

Şekil 5.1 En küçük pozisyon değer kuralı için bir örnek

Permütasyon değeri elde edilmiş parçacığın aday OP çözümü, DKA tabanlı uygulamada olduğu gibi bulunur (Bkz: Bölüm 4.1.1) Yani permütasyondaki kontrol noktaları birinci kontrol noktasından başlayarak başlangıç ve bitiş kontrol noktaları arasına maliyet kısıdını aşmamak şartı ile birer birer eklenir. Maliyet kısıtı aşıldığı anda ekleme işlemi durdurulur ve aday çözüme ulaşılmış olunur.

5.1.2 Güçlendirilmiş-PSO

PSO kolay kodlanabilir oluşu, az sayıda parametreye sahip olması ve hızlı sonuç üretmesinden dolayı meta-sezgisel yöntemler arasında popülerliğini korumaktadır. Bununla beraber, kompleks arama uzayına sahip problemlerde PSO’nun bazı noktalarda yetersiz kaldığı görülmüştür [Liang et al.,2006], [Peer et al., 2003]. PSO’nun bilinen en önemli problemlerinden bir tanesi erken yakınsamadır. Standart PSO yönteminde, eğer sürünün en iyi parçacığı optimal noktadan çok uzakta bir yerel en iyide bulunuyor ve hızı da onu başka yerlere götürmeye yetmeyecek kadar azalmışsa parçacık o noktada takılır ve hatta sürüyü bizzat yönlendirdiği için diğer parçacıklar da o noktaya doğru gelmeye çalışır. Böylece erken yakınsama gerçekleşmiş olur. Bununla beraber popülasyondaki parçacıklar iyi yakınsama yapmazlarsa yerel arama da tam gerçekleşmemiş olur ve algoritma iyi sonuçlar üretemez. Bu problemleri kısmen giderici farklı yöntemler geliştirilmiştir (Bkz: Bölüm 3.2.1)

```

Prosedür Güçlendirilmiş-PSO
  Parametre değerlerini belirle
  Popülasyonun ilk değerlerini belirle
  Tekrarla
    Her bir parçacık için
      Amaç fonksiyonunu bul
      Yerel_tecrübeyi belirle
  Tekrarla -Bitiş
  Sürü_tecrübelerini belirle
  Tekrarla (durdurma şartına rastlanmadığı sürece )
    Her bir parçacık için {
      Eğer (amaç_fonk_değeri = sürü_tecrübesi)
        pozisyon ← YerelArama(pozisyon)
        Yerel_tecrübeyi güncelle
        Hızı yenile
      Değilse
        Hızı güncelle
        Pozisyon güncelle
        Amaç fonksiyonunu bul
        Yerel_tecrübeyi güncelle
        Global_en_iyi güncelle
    Tekrarla-Bitiş
  Prosedür Bitis

```

Şekil 5.2 Güçlendirilmiş-PSO modelinin benzetimsel kodu

Önerdiğimiz Güçlendirilmiş-PSO (G-PSO) modelinin benzetimsel kodu Şekil 5.2’de verilmiştir. Bu model, Atalet Ağırlıklı PSO (IwPSO) modelini (Bkz. Bölüm 3.2.1.1) temel alır. Değişikliklerin odak noktası sürünün küresel tecrübesine ulaşan ya da aşan *öncü-parçacıklardır*. Bu parçacıklar yakınsamış yada yakınsama potansiyeli olan parçacıklardır. G-PSO modelinde bu parçacıkların çözümü arama yolları normal süreçten farklı olarak iki aşamalı değişikliğe uğratılır: İlk aşamada, her bir öncü-parçacık için harici bir yerel arama yöntemi çalıştırılır. Bu aşama PSO’nun yakınsama mekanizmasını kuvvetlendirir. İkinci aşamada ise yerel aramadan çıkan öncü-parçacıklara rasgele bir hız vektörü atanır. Böylece parçacık bulunduğu yerden daha uzak bir noktaya konumlandırılarak çözüm alanında dolaşımına devam ettirilir. Bu yolla da PSO’nun küresel arama özelliği kuvvetlendirilir ve erken yakınsama problemi büyük ölçüde giderilmiş olur. Yerel aramadan çıkan çözüm, parçacığın sadece yerel tecrübesine yansıtılır. Sürüdeki diğer parçacıklarının öncü-parçacık olma şansını koruyabilmesi için yerel aramadan çıkan sonucun sürünün küresel tecrübesini etkilemesi engellenmiştir.

5.1.2.1 Yerel Arama : İ-DKA

G-PSO modelinde harici yerel arama metodu olarak İndirgenmiş-DKA metodu (Bkz. Bölüm 3.1.1) kullanılmıştır. PSO’dan gelen öncü-parçağın pozisyonundan elde edilen pemütasyon vektörü İ-DKA yerel aramasına başlangıç çözüm olarak atanır. İ-

DKA yerel araması iki komşuluk kullanılarak uygulanmıştır: Düğüm Ekleme, Düğüm Yer Değiştirme. (Bu iki komşuluk yapısı DKA tabanlı modelleri açıklarken detaylıca anlatılmıştır Bkz. Bölüm 4.1.2) Önceden tanımlanan durdurma kriterine kadar döngü içinde tanımlanan komşuluklar taranarak bulunan çözüm, parçacığın yerel tecrübesini güncellemesi için geri döndürülür. Yerel aramadan dönen amaç fonksiyon değeri ile sürü tecrübesi güncellenmez.

PSO içinde kullanılan İ-DKA Yerel Arama metodunun benzetimsel kodu Şekil 5.3'te verilmiştir.

```

Prosedür İ-DKA_YerelAramasi
  PSO'dan gelen s çözümünü yerel aramanın başlangıç çözümü yap

  Tekrarla (durdurma kriterine rastlamadığı müddetçe)
    k ← 1
    Tekrarla (k ≤ 2 olduğu sürece)
      Eğer (k = 1)
        s' ← Ekle(s), s' ∈ Düğüm_Ekle(s)
      Eğer (k = 2)
        s' ← Yerdeğiştir(s), s' ∈ Düğüm_Yerdeğiştir (s)

      Eğer (AmaçFonksiyon(s') > AmaçFonksiyon(s))
        s ← s'
        k ← 1
      Değilse
        k ← k+1
    Tekrar-bitiş
  Tekrar-bitiş
Prosedür-Bitiş

```

Şekil 5.3 OP için İ-DKA Yerel Araması Benzetimsel Kodu

Yerel arama metodlarının PSO algoritmasıyla işbirliği içinde çalıştığı birçok çalışma vardır. Örneğin, Tavlama Benzetim (TB) ile PSO'nun melez kullanıldığı çalışmada [Ai-ling et al., 2006] yerel arama her iterasyonda tüm parçacıklar için çalıştırılmıştır. Ayrımsal Gelişim (Differential Evolution) ile PSO'nun melez olarak çalışıldığı bir başka çalışma da ise [Zhan et al., 2009] her dört iterasyonda bir yerel arama tüm parçacıklara uygulanmıştır. DKA'nın PSO ile beraber çalıştırıldığı çalışma da ise her iterasyonda küresel en iyi pozisyona yerel arama uygulanmıştır.

TB ile PSO ve AY ile PSO modellerinde, TB ve AY modelin merkezindedir. PSO'nun görevi iyi bir başlangıç çözümü oluşturmak olmuştur. Bununla beraber G-

PSO'da ise PSO pasif değildir, varolan en iyi çözümü aktif bir şekilde iyileştirir. G-PSO'da İ-DKA hızlı yerel arama yapması için çalıştırılmıştır. Ayrıca her parçacık için yerel arama çalıştırmak büyük problemler için çok maliyetlidir.

Maliyeti azaltmak için DKA'lı PSO'da [Tasgetiren et al., 2007b] olduğu gibi her iterasyonda sürü tecrübesine yerel arama uygulanabilir. Burda eğer PSO sürü tecrübesini iterasyondan iterasyona değiştiremezde aynı yerde defalarca yerel arama yapma durumu doğabilir. Çünkü PSO'nun yerel aramadan çıkan bir sürü tecrübesini iyileştirme olasılığı oldukça düşüktür. Bu yüzden DKA'lı PSO çalışmasında da DKA'nın ağırlığı büyük orandadır.

G-PSO'da ise yerel arama (İ-DKA) sürü tecrübesini geçen yada yakalayan öncü parçacıklara uygulanır. Bir iterasyonda birden fazla öncü parçacık olabileceği gibi hiç öncü parçacık bulunmayabilir. Bu yüzden, yerel arama gerekli olduğu durumlarda ancak çalışır. Ayrıca G-PSO'da yerel aramadan çıkan sonuç sürü tecrübesini doğrudan etkilemediğinden iyi yerlerin bilgisi sürü içinde yavaşça dağılır. Sürü içindeki parçacıkların öncü parçacık olma olasılığı böylece hiç bir zaman azalmaz.

5.1.2.2 Uzaklaştırma Stratejisi:

Standart PSO akışına yapılan ikinci değişiklik ise öncü-parçacıklara ayrı bir hız güncelleme yönteminin uygulanmasıdır. Öncü-parçacıklara başlangıçta belirlenen hız sınırları içinde rasgele üretilmiş hız vektörü atanır. Böylece bu parçacıklar normal pozisyon güncelleme sonucu elde ettiği tecrübe ile birlikte farklı bir yerden aramaya devam edecek şekilde güncellenmiş olurlar.

Önceki çalışmalarda bir çok rasgele hareket etme stratejisi kullanılmıştır. Örneğin GCP PSO'da (Bkz. Bölüm 3.2.1), hız güncelleme formülüne rasgele hız terimi eklenmiştir. Bu yeni hız formülü küresel en iyiyi temsil eden parçacık için kullanılmıştır. Ayrımsal Gelişimli PSO içinde benzer olarak küresel en iyi pozisyonun yakın çevresine bazı parçacıkların pozisyonları rasgele olarak atanmıştır.

Literatürde bu tür rasgele hareketlendirme stratejisi daha çok küresel en iyi pozisyon etrafında yoğunlaşma yapabilmek için geliştirilirken G-PSO'da kullanılan rasgele hareketlendirme stratejisi küresel en iyi pozisyondan uzaklaştırmak yani yayılmayı artırmak amaçlı kullanılmıştır.

5.1.3 G-PSO İçin Yan Modeller

OP problemi için önerilen G-PSO modelinde yapılan değişikliklerin etkilerini daha net görebilmek için aşağıda detayları verilen iki yan-model geliştirilmiştir. Bu modeller:

Yerel Araması Güçlendirilmiş-PSO (YAG-PSO): Bu modelde sadece öncü parçacıklara İ-DKA yerel arama metodu uygulanır ve parçacık hızı yenilenmeden varolan hızı ile aramaya devam eder.

Küresel Araması Güçlendirilmiş-PSO (KAG-PSO): Bu modelde ise öncü parçacık yerel aramaya sokulmadan doğrudan hız vektörü rasgele değerlerle yenilenir. Bu modelde yerel en iyi çözüme yakınsama geciktirilmiş olup küresel arama kuvvetlendirmiştir.

5.2 Ayırık Güçlendirilmiş PSO Modeli

OP ve TOP için önerilen ikinci PSO tabanlı model Ayırık Güçlendirilmiş Parçacık Sürü Optimizasyon modelidir. Model, Bölüm 5.1.2'de anlatılan G-PSO algoritma akışını takip etmekle birlikte parçacığın pozisyon, hız ve bu bileşenleri güncellemek için kullanılan tüm operatörleri ayırık olarak ifade edilmiştir. Modelin problem gösterim şekli ve ayırık operatörleri hakkındaki detaylar alt bölümlerde açıklanmıştır.

5.2.1 Problem Gösterimi

Önerilen Ayırık G-PSO modeli ayırık problemler için geliştirilen PSO modellerinden Model 1'i temel alır (Bkz. 3.2.1.2). Model 1, PSO'nun pozisyon, hız vektörleri ve tanımlı tüm operatörlerinin tamamen ayırık olarak ifade edildiği bir modeldir. Buna göre Ayırık G-PSO modelinde, parçacığın pozisyon ve hızı kontrol noktalarını içeren değişebilen uzunluklu listeler şeklinde tutulur. Pozisyon listesi aday OP/TOP çözümünü gösterir. Pozisyon listesi, problemde belirtilen araç sayısı kadar

alt listeden oluşur. Her bir alt listenin ilk elemanı başlangıç kontrol noktası son elemanı ise bitiş kontrol noktasıdır. Listenin geri kalan elemanları ise problemde her bir araç rotası için verilen maliyet kısıtını aşmayacak şekilde sıralanmış diğer kontrol noktalarıdır. Hız listesi ise aynı parçacığın pozisyon listesine maliyet kısıtından dolayı giremeyen diğer kontrol noktalarını içerir.

5.2.2 Ayrık Operatörler

Ayrık PSO modelinde, standart PSO’da kullanılan formül (3.1) ve formül (3.2), deki operatörler, ayrık arama uzayında bir çözümünden diğerine geçebilmesi için aşağıda verilen yeni hız ve pozisyon güncelleme formüllerindeki gibi yeniden tanımlanmıştır.

$$v_i^t = \left[w_i \otimes v_i^{t-1} \right] \oplus \left[(c_1 \otimes (pbest_i^{t-1} \ominus p_i^{t-1})) \oplus (c_2 \otimes (gbest^{t-1} \ominus p_i^{t-1})) \right] \quad (5.1)$$

$$p_i^t = p_i^{t-1} \circ v_i^t \quad (5.2)$$

$$v_i^t = p_{tüm} \ominus p_i^t \quad (5.3)$$

Formül (5.1)’de, parçacığın o anki hızı, kendi tecrübesi ve sürü tecrübesi yeni operatörlerle kullanılarak parçacığın hız listesi yeniden hesaplanır. Hesaplanan hız listesi parçacığın yeni pozisyonunu belirlemek için formül (5.2)’de kullanılır. Bir sonraki iterasyon için, parçacığın hızı listesi $p_{tüm}$ ve yeni pozisyon listesi kullanılarak hesaplanır. $p_{tüm}$ problem kümesinde bulunan tüm kontrol noktalarının rasgele sırada bir listesidir. Yeni formüllerde kullanılan operatörlerin işlevleri ise aşağıda açıklanmıştır.

$\ominus$ operatörü: Standart PSO’da i . parçacığın hız vektörü üç bileşenden çıkan sonuçlarla kendini günceller. Bu bileşenlerden ikisini hesaplarken parçacığın yerel tecrübesi ile ($pbest_i$) veya sürünün tecrübesi ile ($gbest_i$) parçacığın o andaki pozisyon vektörü (p_i) arasındaki mesafeyi her boyutta hesaplayan çıkarma (-) operatörü kullanılır. Ayrık G-PSO’da ise iki pozisyon listesinin $\ominus$ operatörü ile işleme girmesi ile ($p_1 \ominus p_2$), p_1 listesinde olup da p_2 listesinde olmayan kontrol noktalarını içeren yeni bir hız listesi oluşur. Eğer p_1 listesi $gbest$ ise oluşan hız listesindeki kontrol noktaları liste sıralamasında ön tarafta olmayı sağlayan *yüksek-önceliğe (yö)* sahip

olur. Eğer p_1 listesi p_{best} ise oluşan hız listesindeki kontrol noktaları *düşük-önceliğe* (*dö*) sahip olur. Diğer tüm durumlarda ise yeni oluşan hız listesine ait kontrol noktalarına öncelik atanmaz (ö-yok)

$\otimes$ operatörü : Bu operatörün sol yanı bir katsayı ($0 < c < 1$), sağ yanı ise hız listesi (v) alır. İşlemin ($c \otimes v$) sonunda ise yeni bir hız listesi oluşur. Katsayı, hız listesi içindeki bir kontrol noktasının yeni hız listesinde olma olasılığını gösterir. Buna göre sonuç hız listesi şu şekilde oluşur: Hız listesi v içindeki her bir kontrol noktası için $[0,1]$ arasında düzgün dağılımlı rasgele bir sayı üretilir. Bu rasgele sayı c 'den küçük ise yeni hız listesine eklenir. Bu işlem kontrol noktalarının sahip oldukları öncelik özelliğini değiştirmez.

$\oplus$ operatörü : Bu operatör, iki hız listesini birleştirme işlemini yapar. Birleştirme işlemi yaparken hız listelerindeki kontrol noktalarının sahip olduğu yüksek öncelik, düşük öncelik yada önceliksizlik özellikleri dikkate alınır. Eğer düşük ve yüksek önceliğe sahip kontrol noktalarından oluşan hız listelerinin birleşimi yapılacaksa ($v_{yö} \oplus v_{dö}$), yeni oluşan hız listesindeki kontrol nokta sırasında öncelik kesişimde olan kontrol noktalarına verilir. Daha sonra $v_{yö}$ 'den kalan diğer kontrol noktaları ve arkasına ise $v_{dö}$ 'den kalan kontrol noktaları eklenir. Eğer birleşime girecek iki hız listesinden birinin kontrol noktaları önceliksiz ise ($v_{yö/dö} \oplus v_{ö-yok}$), yeni oluşan hız listesindeki kontrol nokta sırası öncelikle $v_{yö/dö}$ içindeki kontrol noktalarına verilir. Sonrasında ise $v_{ö-yok}$ 'den kalan kontrol noktaları hız listesine eklenir.

◦ **operatörü**: Bu operatör, parçacığın hız ve pozisyon listesiyle işleme girer. Hız listesi pozisyon listesine eklenecek kontrol noktalarını içerir. Hız listesindeki kontrol noktaları sırayla *Dışarıdan Düğüm Ekle Yerdegistir* komşuluk yapısına (Bkz. Bölüm 4.2.2) göre eklenir.

Yukarıda anlatılan operatörlerin formül (5.1) için çalışma örneği Şekil 5.4'te sunulmuştur.

Örnek: 9 kontrol noktasından oluşan bir problem kümesinde 1 başlangıç kontrol noktası 9 ise bitiş noktası olduğunu düşünelim. Buna göre Tam Ayrık G-PSO'da bir parçacığın t. iterasyondaki bileşenleri aşağıdaki gibi hesaplanır

$$\begin{aligned}
 p_{tüm} &= (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9) \\
 p_i^t &= (1\ 4\ 7\ 9) \\
 v_i^t &= (2\ 3\ 5\ 6\ 8) \text{ (ö - yok)} \\
 pbest_i^t &= (1\ 3\ 7\ 6\ 9) \\
 gbest^t &= (1\ 8\ 6\ 4\ 2\ 9) \\
 c_1 &= 0.5 \\
 c_2 &= 0.5 \\
 w &= 0.7 \\
 v_{pbest}^{t+1} &= (pbest_i^{t-1} \ominus p_i^{t-1}) \\
 v_{pbest}^{t+1} &= (1\ 3\ 7\ 6\ 9) \ominus (1\ 4\ 7\ 9) = (3\ 6) \text{ (dö)} \\
 v_{gbest}^{t+1} &= (gbest_i^{t-1} \ominus p_i^{t-1}) \\
 v_{gbest}^{t+1} &= (1\ 8\ 6\ 4\ 2\ 9) \ominus (1\ 4\ 7\ 9) = (8\ 6\ 2) \text{ (yö)} \\
 v_i^{t+1} &= \left[0.7 \otimes v_i^t \right] \oplus \left[(0.5 \otimes v_{pbest}^{t+1}) \oplus (0.5 \otimes v_{gbest}^{t+1}) \right] \\
 v_i^{t+1} &= (3) \oplus [(3\ 6) (8\ 6)] \\
 v_i^{t+1} &= (3) \oplus (6\ 8\ 3) \\
 v_i^{t+1} &= (6\ 8\ 3)
 \end{aligned}$$

Şekil 5.4 Ayrık G-PSO'da hız güncelleme örneği

5.3 Deneyisel Sonuçlar

Bu bölümde önerilen PSO tabanlı modellerin karşılaştırmalı OP ve TOP test problemleri için yapılan deneylerinin sonuçları sunulacaktır.

5.3.1 OP için PSO Tabanlı Modellerin Deneyisel Sonuçları

Yarı Ayrık ve Ayrık G-PSO modelleri 103 karşılaştırmalı OP üzerinde test edilmiştir. Bu iki model yan modelleri ile beraber Intel P4 2.6 GHz işlemci ve 1 GB bellek içeren bir bilgisayar üzerinde 10 defa çalıştırılmıştır. Yarı Ayrık modellerin durdurma şartı amaç fonksiyonunun 100.000 defa hesaplanması, Ayrık modelin ise 300 defa arka arkaya sürü tecrübesini yenileyememek olarak tanımlanmıştır.

Yarı Ayrık G-PSO modellerinin denemelerinde sürü 20 parçacıktan oluşmuş olup c_1 , c_2 ve w parametrelerine sırasıyla 2.0, 2.0 ve 0.9 değerleri verilmiştir. Ayrıca her iterasyondan sonra atalet ağırlığı %2 azaltılmıştır. Sürüdeki parçacıkların başlangıç pozisyonları DKA tabanlı modellerdeki gibi rasgele permütasyonla oluşturulmuştur. PSO tabanlı modellerde başlangıç hız değerleri ise problemdeki kontrol sayısı d ile gösterildiği takdirde, $[-d, d]$ arasında düzenli dağılım kullanılarak rasgele üretilmiştir.

Ayrık G-PSO modeli ise sürüdeki parçacık sayısı 10 alınmıştır. Parametre c_1 ve c_2 değeri 0.5 verilmiştir. Atalet ağırlığı parametresi ise 0.7 ile başlar ve her iterasyon %2 azaltılarak 0.4'e kadar düşmesi sağlanır. Parçacığın başlangıç pozisyon ve hız liste değerleri, problem kümesini oluşturan başlangıç ve bitiş kontrol noktaları dışındaki kontrol noktalarının rasgele permütasyonundan yararlanarak hesaplanır. Pozisyon listesinin başlangıç ve bitiş kontrol noktaları arasına permütasyonun başından başlanarak ve maliyet kısıtı geçilmeyene kadar kontrol noktası aynı sırada eklenir. Bu işlem araç sayısı kadar liste oluşana kadar devam eder. Böylece parçacığın pozisyonu belirlenmiş olur. Permütasyondan geriye kalan kontrol noktaları ise aynı sırada hız listesine eklenir.

PSO tabanlı geliştirdiğimiz dört modelle birlikte Iw-PSO modelinin yukarıda açıklanan kriterlere göre 67 test problemi üzerinde karşılaştırma sonuçları Çizelge 5.1'de verilmiştir. Sonuçlara bakıldığında birinci model olan atalet ağırlıklı-PSO (IwPSO) modelinin en iyi değerini, çok kısa sürede (çok az amaç fonksiyon üreterek) fakat bilinen en iyi sonuca çok büyük hata oranında yaklaşıp bulduğu görülür. Iw-PSO modeline hız yenileme prosedürü ekleyerek elde edilen KAG-PSO'nun ise çözümleri Iw-PSO'ya göre daha az hata ile ama çok fazla CPU zamanı dolayısıyla fazla amaç fonksiyon hesaplayarak elde ettiği anlaşılır. Yarı Ayrık G-PSO modeli ise çözümü IwPSO ve KAG-PSO'ya göre hem kalite hem de hız yönünden iyileştirmiş, 67 problemde sadece 3 tanesini hata ile bulmuştur. YAG-PSO modeli ise yarı ayrık modeller arasında her üç kritere göre de en iyi sonuçları vermiştir. Ayrık G-PSO ise problemin gerçek çözüm uzayı ile algoritmanın oluşturduğu çözüm uzayının birebir eşleşmesinden dolayı Yarı Ayrık G-PSO'nun bulamadığı 3 problem içinde bilinen en iyi sonuçları CPU, yüzdelik bağıl hata ve ortalama yüzdelik bağıl hata kriterlerine göre en iyi değerlerle bulduğu tespit edilmiştir.

Çizelge 5.1 PK 1,2,3 ve 1_1 için PSO tabanlı modellerin CPU(saniye), Yüzdelik ve Ortalama Yüzdelik Bağlı Hata kriterlerine göre karşılaştırılması.

	Modeller	Problem Kümeleri				Ort
		PK1	PK2	PK3	PK1_1	
CPU	Yarı Ayrık Iw-PSO	1.52	1.04	1.17	1.57	1.33
	Yarı Ayrık KAG-PSO	19.86	17.71	31	18.57	21.79
	Yarı Ayrık YAG-PSO	4.77	1.7	4.82	4.6	3.97
	Yarı Ayrık G-PSO	4.72	2.12	5.22	4.6	4.16
	Ayrık G-PSO	4.00	0.92	6.69	4.13	3.94
Yüzdelik Bağlı Hata	Yarı Ayrık Iw-PSO	42.22	23.83	37.74	40.38	36.04
	Yarı Ayrık KAG-PSO	20.39	10.71	22.72	19.74	18.39
	Yarı Ayrık YAG-PSO	0.00	0.00	0.00	0.00	0.00
	Yarı Ayrık G-PSO	0.14	0.00	0.15	0.00	0.07
	Ayrık G-PSO	0.00	0.00	0.00	0.00	0.00
Ort Yüzdelik Bağlı Hata	Yarı Ayrık Iw-PSO	53.85	38.65	48.02	55.06	48.89
	Yarı Ayrık KAG-PSO	28.05	19.05	30.34	25.29	25.68
	Yarı Ayrık YAG-PSO	0.70	0.62	0.79	0.86	0.74
	Yarı Ayrık G-PSO	0.86	0.89	0.73	0.86	0.84
	Ayrık G-PSO	0.00	0.04	0.00	0.00	0.01

Yarı Ayrık ve Ayrık G-PSO modellerinden elde edilen en iyi sonuçlar aşağıda listelenen modellerin elde ettikleri en iyi sonuçlarla da karşılaştırılmıştır.

- CGW: Sezgisel algoritma [Chao et al., 1996] (prob. kümesi 1, 2, 3, 1_1, 5 ve 6 üzerinde deneyler yapılmıştır)
- GA : Genetik algoritma [Tasgetiren and Smith, 2000] (prob. kümesi 1, 2, 3, 1_1 üzerinde deneyler yapılmıştır)
- KKO: Karınca kolonileri optimizasyonu [Liang et al., 2002] (prob. kümesi 1, 2, 3, 1_1 üzerinde deneyler yapılmıştır)
- KYA: Kılavuzlu yerel arama [Vansteenkoven et al., 2009] (prob. kümesi 1, 2, 3, 5 ve 6 üzerinde deneyler yapılmıştır)

Problem kümesi 1, 1_1, 2 ve 3 için modellerin yüzdelik bağlı hatası ve tekrarlı çalışmalar sonunda amaç fonksiyon değerinin standart sapması kriterlerine göre karşılaştırma değerleri Çizelge 5.2’de verilmiştir. Belirtilen problem kümesi için GA, KKO ve Ayrık G-PSO’nun bilinen en iyi sonuçları bulunduğu görülür (yüzdelik bağlı hata oranına bakınız). Bunun yanında KYA modelinin CGW ve Yarı Ayrık G-PSO’ya göre daha fazla hata oranı ile bilinen en iyi çözümlere yaklaşmıştır. Tekrarlı çalıştırma sonucu elde edilen amaç fonksiyonunun standart sapma değeri bakımından ise

Ayrık G-PSO'nun en düşük değeri olarak çözüm kalitesi yönünden test edilen problem kümesi için diğer modellerden daha iyi performans sergilediği söylenebilir.

Çizelge 5.2 PSO ve DKA tabanlı modellerin PK 1,2,3 ve 1_1 için literatür sonuçlarıyla karşılaştırılması

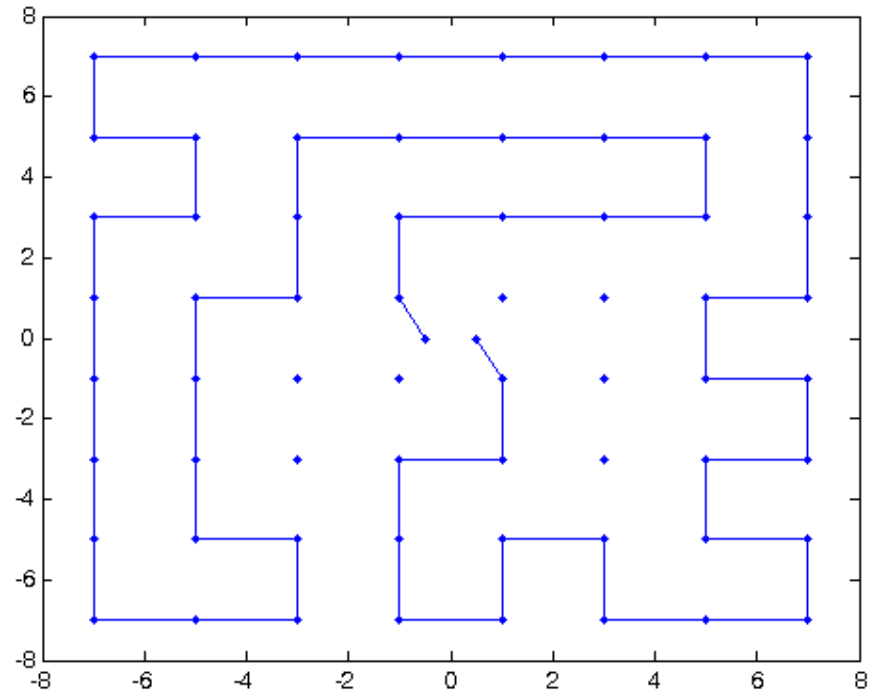
	Modeller	PK 1	PK 2	PK 3	PK 1-1	Ort
Yüzdelik Bağıl Hata	CGW	0.00	0.00	0.00	0.11	0.03
	GA	0.00	0.00	0.00	0.00	0.00
	KKO	0.00	0.00	0.00	0.00	0.00
	KYA	3.17	1.36	0.65	-	1.73
	Yarı Ayrık G-PSO	0.14	0.00	0.15	0.00	0.07
	Ayrık G-PSO	0.00	0.00	0.00	0.00	0.00
Std. Sap.	CGW	-	-	-	-	-
	GA	1.29	2.63	3.38	1.36	2.17
	KKO	0.59	1.66	0.72	0.48	0.86
	KYA	-	-	-	-	-
	Yarı Ayrık G-PSO	2.13	3.78	4.43	1.94	3.10
	Ayrık G-PSO	0.00	0.34	0.00	0.13	0.12

Ayrık G-PSO modelinin problem kümesi 5 ve 6 için literatürdeki sonuçlarla karşılaştırma listesi Çizelge 5.3 ve Çizelge 5.4'te verilmiştir. Modelimiz literatürdeki bütün en iyi sonuçları bulmakla beraber, CGW'nin toplam 12 sonucunda, KYA'nın ise 19 sonucunda iyileştirme yapmıştır. Literatüre kazandırılan yeni sonuçlar Çizelge 5.5'te çözüm yollarıyla beraber verilmiştir. Ayrıca toplam 103 OP problemi için Ayrık G-PSO modelinden elde edilen sonuçlar EK-D'de her bir problem için listelenmiştir.

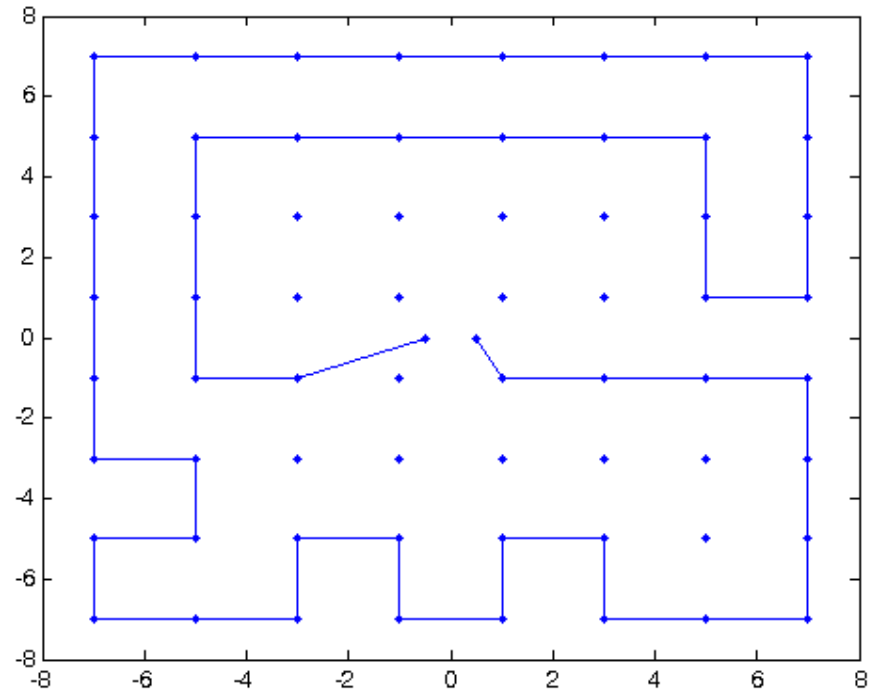
Çizelge 5.5 OP için Ayrık G-PSO modelinin geliştirdiği yeni sonuçlar

PK	T _{maks}	T _d	Kazanç	Bulunan Rota
5	20	19.81	205	1 29 28 27 26 34 42 43 44 45 66
5	95	94.24	1395	1 30 31 32 40 48 56 55 54 62 63 64 65 57 49 41 33 25 17 9 8 7 15 14 6 5 13 12 4 3 2 10 11 19 18 26 34 42 43 51 50 58 59 60 61 53 45 37 66
5	100	99.81	1465	1 21 13 14 15 16 24 32 40 48 56 55 54 62 63 64 65 57 49 41 33 25 17 9 8 7 6 5 4 12 11 3 2 10 18 19 27 26 34 35 43 42 50 58 59 60 61 53 45 37 66
5	105	104.24	1520	1 29 28 27 19 11 12 13 14 15 16 24 32 33 25 17 9 8 7 6 5 4 3 2 10 18 26 34 42 50 58 59 60 61 62 63 64 65 57 49 41 40 48 56 55 54 53 52 51 43 35 36 37 66
6	35	34.86	576	1 3 6 10 15 21 28 36 43 35 27 34 26 33 32 24 31 23 30 38 45 51 56 64
6	80	79.19	1284	1 2 4 7 12 17 11 16 22 29 37 44 50 55 59 56 51 45 38 30 23 31 24 18 13 19 14 10 15 21 28 36 43 35 42 49 54 48 41 34 27 20 26 33 25 32 39 46 40 47 52 57 53 58 61 63 64

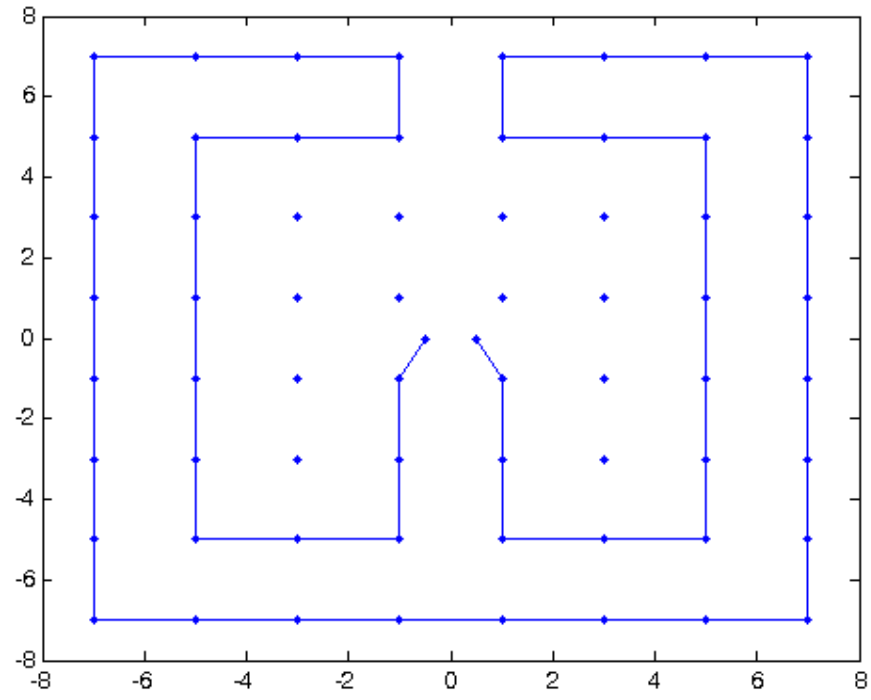
İİ-DKA farklı sıralamada Ayrık G-PSO ile elde edilmiş yeni çözümlerin grafiksel gösterimi ise Şekil 5.5-Şekil 5.8 arasında verilmiştir.



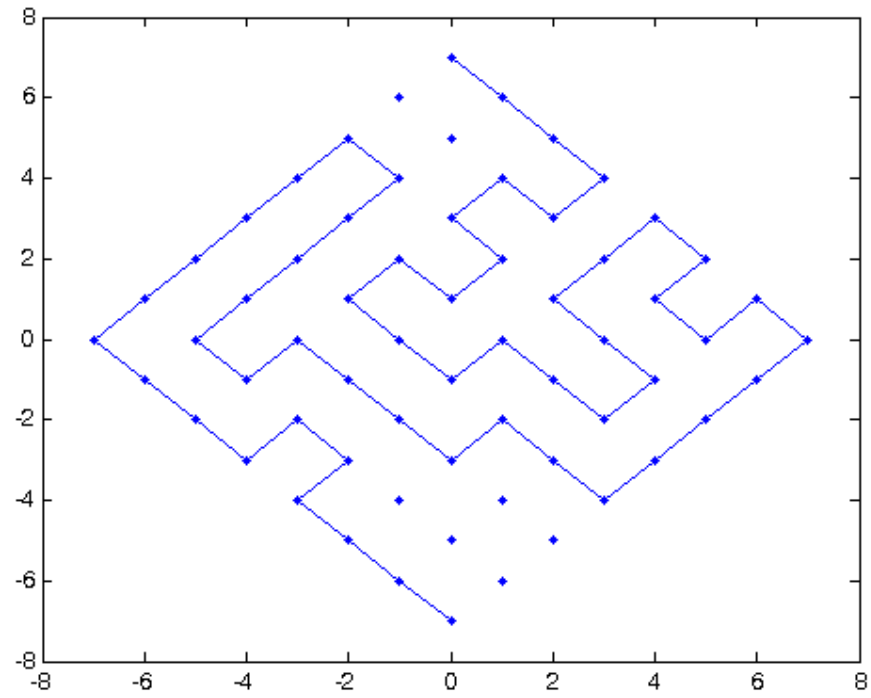
Şekil 5.5 PK 5'te $T_{maks} = 95$ için Ayrık G-PSO ile elde edilen yeni çözüm



Şekil 5.6 PK 5'te $T_{maks} = 100$ için Ayrık G-PSO ile elde edilen yeni çözüm



Şekil 5.7 PK 5'te $T_{maks} = 105$ için Ayrık G-PSO ile elde edilen yeni çözüm



Şekil 5.8 PK 6'te $T_{maks} = 80$ için Ayrık G-PSO ile elde edilen yeni çözüm

5.3.2 TOP için PSO Tabanlı Modellerin Deneysel Sonuçları

TOP için önerdiğimiz Ayırık G-PSO modeli Bölüm 2.4’de açıklanan 350 TOP problemi üzerinde (problem kümesi 1, 2, 3, 4, 5, 6 ve 7) test edilmiştir.

Modelin TOP için parametre değerleri, OP için kullanılan parametre değerleri ile aynıdır (Bkz. Bölüm 5.3.1). Deneylerden çıkan sonuçlar, algoritmanın en iyi bulma süresine (CPU zamanı saniye), bağlı yüzde hatasına ve tekrarlı çalıştırmalar arasındaki standart sapma miktarına göre karşılaştırılmıştır.

Çizelge 5.6 TOP için geliştirilen Ayırık PSO tabanlı modellerin CPU(saniye), Yüzdelik Bağlı Hata ve Standart Sapma kriterlerine göre karşılaştırılması.

		CPU				Yüzdelik Bağlı Hata				Standart Sapma			
		IwPSO	KAG-PSO	YAG-PSO	G-PSO	IwPSO	KAG-PSO	YAG-PSO	G-PSO	IwPSO	KAG-PSO	YAG-PSO	G-PSO
PK 1	2 araçlı	1.10	4.26	3.45	1.28	0.34	2.27	0.00	0.00	1.85	2.27	0.90	0.00
	3 araçlı	1.33	4.00	1.72	0.73	0.53	2.62	0.00	0.00	2.03	2.62	1.29	0.73
	4 araçlı	0.88	2.57	0.81	0.35	0.33	1.70	0.00	0.00	1.14	1.70	0.63	0.35
	Ort	1.11	3.65	2.05	0.90	0.40	1.31	0.00	0.00	1.69	2.21	0.94	0.25
PK 2	2 araçlı	0.15	0.32	0.19	0.15	0.17	0.51	0.00	0.00	2.81	3.08	0.00	0.00
	3 araçlı	0.11	0.16	0.06	0.05	0.00	0.00	0.00	0.00	0.14	0.57	0.00	0.00
	4 araçlı	0.02	0.07	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Ort	0.09	0.18	0.09	0.07	0.06	0.17	0.00	0.00	0.98	1.22	0.00	0.00
PK 3	2 araçlı	2.58	3.59	4.58	3.13	1.22	2.48	0.12	0.00	6.42	6.81	4.45	1.05
	3 araçlı	2.18	3.20	1.70	1.13	2.00	3.68	0.00	0.00	7.36	7.60	5.99	0.83
	4 araçlı	1.61	2.64	0.83	0.62	2.42	4.16	0.00	0.00	8.68	12.42	4.39	0.27
	Ort	2.14	3.16	2.44	1.67	1.86	3.41	0.04	0.00	7.45	8.84	4.94	0.73

Ayrık G-PSO, IwPSO modelinin üzerine yapmış olduğu değişiklikleri görebilmek için iki yan model olan KAG-PSO ve YAG-PSO ile beraber 138 problem üzerinde test edilmiştir. Karşılaştırmanın daha anlamlı olabilmesi için IwPSO dahil diğer iki yan model için de Ayrık G-PSO’nun kullandığı ayırık operatörler kullanılmıştır. Deney sonuçlarının problem kümesi/araç sayısı bazında özeti Çizelge 5.6’da verilmiştir. Ayrık KAG-PSO dışındaki diğer üç Ayrık modelin (IwPSO, YAG-PSO, G-PSO) kalıtsal veya dışardan eklenmiş bir detaylı arama mekanizması olduğundan, 138 problemin hepsinde elde edilen sonuçlar Ayrık KAG-PSO’ya göre daha iyidir. Kalıtsal olarak gelen yerel aramanın IwPSO’da yeterli

kalmadığını yüzdelik bağıl hata değeri göstermektedir. Ayrık YAG-PSO, problemlerin çoğunda ve Ayrık G-PSO ise hepsinde en iyi sonuçları bulmuştur. Bununla beraber Ayrık G-PSO hem standart sapma değeri bakımından hemde CPU bakımından Ayrık YAG-PSO'dan daha iyidir.

Toplam 350 TOP problemi için Ayrık G-PSO modellerinden elde edilen sonuçlar yüzdelik bağıl hata, standart sapma ve CPU kriterlerine göre Çizelge 5.7'de özetlenmiştir.

Çizelge 5.7 Ayrık G-PSO modelinin toplam yedi PK için performans analizi

Problem Kümesi (PK)	Araç Sayısı	Toplam Problem Sayısı	Ayrık G-PSO			
			% Bağıl Hata Ort.	Std. Sap. Ort.	CPU Ort.	En iyi Değeri Bulma Sayısı
1	2	17	0.00	0.00	1.28	17
	3	16	0.00	0.73	0.73	16
	4	15	0.00	0.35	0.35	15
		48	0.00	0.25	0.90	48
2	2	11	0.00	0.00	0.15	11
	3	11	0.00	0.00	0.05	11
	4	11	0.00	0.00	0.01	11
		33	0.00	0.00	0.07	33
3	2	20	0.00	1.05	3.13	20
	3	19	0.00	0.83	1.13	19
	4	18	0.00	0.27	0.62	18
		57	0.00	0.73	1.67	57
5	2	25	0.29	5.30	11.88	18
	3	25	0.29	5.98	8.27	16
	4	24	0.47	7.76	9.34	16
		74	0.35	6.33	9.84	50
6	2	11	0.28	3.29	5.86	8
	3	8	0.16	2.86	5.00	6
	4	5	0.00	3.70	4.13	5
		24	0.18	3.23	5.21	19
4	2	20	0.88	6.99	34.61	4
	3	19	1.78	11.01	35.37	2
	4	17	2.89	12.27	18.69	4
		56	1.93	10.38	29.38	10
7	2	20	0.78	7.24	15.73	7
	3	19	1.61	8.33	10.97	7
	4	19	2.38	8.65	6.45	7
		58	1.58	8.06	11.74	21
Toplam		350				238

Çizelgedeki özet sonuçlar her bir problem kümesi/ araç sayısı için ayrı ayrı verilmiş olup, problem kümeleri de içerdiği kontrol sayılarına göre sıralanmıştır. Buna göre

problem kümesi 1, 2 ve 3 içindeki 138 problemin bilinen en iyi sonuçları Ayırık G-PSO modeli tarafından bulunmuştur (Bakınız % bağıl hata ort). Problem Kümesi 5 ve 6 için ortalama %0.5'den daha küçük hata oranı ile çözümleri bulurken, problem kümesi 4 ve 7'de ise hata oranı %3'e kadar çıkmıştır. Her bir problemin detaylı sonuçları EK-E'de listelenmiştir.

Ayrık G-PSO modellerinden elde edilen en iyi sonuçlar aşağıda listelenen modellerin elde ettikleri en iyi sonuçlarla Çizelge 5.8'de karşılaştırılmıştır.

- CGW: Sezgisel algoritma [Chao et al., 1996]
- TMH: Tabu Arama Tabanlı Çözüm Yöntemi [Tang and Miller Hooks, 2005]
- GTP: Cezalı Tabu Arama [Archetti et al., 2007]
- GTF: Cezasız Tabu Arama [Archetti et al., 2007]
- HT-DKA: Hızlı Tabulu Değişken Komşu Arama [Archetti et al., 2007]
- YT-DKA: Yavaş Tabulu Değişken Komşu Arama [Archetti et al., 2007]
- Ase KKO: Karınca Kolonileri Optimizasyonu [Ke et al., 2008]
- MA: Memetik Algoritma [Bouly et al., 2008]
- KYA: Kılavuzlu Yerel Arama [Vansteenkoven et al., 2009]

Daha önceki çalışmalarla yapılan karşılaştırma, problem kümeleri bazında iki grupta özetlenmiştir. Kontrol nokta sayısı 40'tan küçük olan problem kümesi 1,2 ve 3 A grubunda geri kalan problem kümeleri ise B grubunda özetlenmiştir. A grubunda yer alan 138 problem için CGW, TMH ve GTP dışındaki tüm modeller her problem için en iyi sonucu bulmuşlardır. Bu grupta Ayırık G-PSO, son yıllarda geliştirilen HT-DKA dışındaki YT-DKA, Ase KKO ve MA'den ortalama olarak daha kısa sürede sonuçları üretmiştir. B grubunda ise Ayırık G-PSO, son zamanlarda yayımlanan ve hızlı bir model olduğunu ön plana çıkaran KYA modeline göre daha fazla en iyi sonucu eş değer süre zarfında daha az hata ile bulmuştur.

Sonuç olarak Ayırık G-PSO hızlı çözüm üretmesi bakımından TOP için ümit verici bir modeldir.

Çizelge 5.8 Ayırık G-PSO performanslarının önceki çalışmalarla karşılaştırılması

		CGW	TMH	GTP	GTF	HT-DKA	YT-DKA	Ase KKO	MA	KYA	Tam AyırıkG-PSO
en iyi sonucu bulma sayısı	A	112	121	136	138	138	138	138	138	-	138
ort % bağıl hata		0.97	0.42	0.05	0.00	0.00	0.00	0.00	0.00	-	0.00
ort CPU		11.91	-	4.12	1.22	0.11	6.92	5.45	1.13	-	1.02
en iyisonucu bulma sayısı	B	88	89	124	148	150	181	182	185	68	100
ort % bağıl hata		1.98	1.10	0.41	0.19	0.17	0.05	0.56	0.47	2.00	1.09
ort CPU		565.65	361.85	103.40	157.83	21.82	279.18	26.26	68.41	10.37	15.09

5.3.3 Önerilen DKA ve PSO Tabanlı Modellerin Karşılaştırılması

Önerilen modeller test problemlerinin içerdiği kontrol nokta sayısı (*küçük, orta, büyük*) ve kontrol noktalarının koordinat düzlemindeki dağılımına (*düzenli, düzensiz*) göre gruplanabilir 403 problemle test edilmiştir.

Düzensiz-küçük problem grubu (PK1, PK2, PK3, PK1_1) üzerinde yapılan OP deneylerinde Ayırık G-PSO ve İİ-DKA modelleri gruba ait tüm problemler için bilinen en iyi sonuca ulaşırken Yarı Ayırık G-PSO modeli 63 problemde 3'ünü yaklaşık %0.14 ortalama bağıl hata ile bulmuş, geri kalan problemlerde ise bilinen en iyi sonuçları elde etmiştir. Bu sonuç geliştirilen modelin oluşturduğu çözüm uzayı ile problemin çözüm uzayının birebir eşlenik olmasının önemini göstermektedir.

Düzenli-orta büyüklük grubu (PK5, PK6) için yapılan OP deneylerinde İİ-DKA ve Ayırık G-PSO modellerinin her ikisi de bilinen en iyi sonuçları bulmakla beraber 12 sonucu iyileştirerek literatüre kazandırmıştır. Bu iki modelin performans sonuçları birbiri ile karşılaştırıldığında T_{maks} değeri 55'in altındaki problemler için yaklaşık aynı zaman dilimlerinde sonuçlar elde edilirken T_{maks} değeri arttıkça Ayırık G-PSO, İİ-DKA'ya göre daha fazla zaman harcadığı gözlenmiştir. Tekrarlı çalıştırmalardaki çözüm kalitesine bakıldığında ise ortalama yüzdelik bağıl hata ve standart sapma değerleri bakımından Ayırık G-PSO'nun İİ-DKA'ya göre daha iyi çözüm kalitesinde sonuçlar elde ettiği görülmüştür.

İİ-DKA ve Ayrık G-PSO'nun *düzensiz-küçük* problem grubunda yapmış olduğu TOP deney sonuçları her iki modelin grupta yer alan 138 problemin bilinen en iyi sonuçlarını bulduklarını göstermiştir. Ayrık G-PSO ve İİ-DKA, yerel arama olarak aynı komşuluk yapılarında İ-DKA yöntemini çalıştırmalarına rağmen İİ-DKA'nın ortalama beş kat daha fazla süre harcayarak sonuçları bulması, Ayrık G-PSO'nun ayrık operatörleriyle yönlendirilen parçacıklarının yerel arama için İİ-DKA'ya göre daha kısa sürede iyi yerler bulunduğunu gösterir.

Düzenli-orta büyüklükteki TOP problem grubunda ise İİ-DKA 98 problemin 92'sinde bilinen en iyi sonuçları ortalama 25 saniyede bulurken, Ayrık G-PSO ise, İİ-DKA'ya göre yaklaşık dörtte bir zaman harcayarak 69 problem için bilinen en iyi sonuçları bulabilmiştir. Benzer sonuçlar *düzensiz-büyük* (PK4, PK7) problem grubunda da görülmüş olup, toplam 114 problemde İİ-DKA 77'sinde, Ayrık G-PSO ise 31'inde bilinen en iyi sonuçları bulabilmiştir. Bununla beraber İİ-DKA'nın bilinen en iyi sonuçları bulamadığı problemler için ortalama %0.12'lik hata yüzdesi ile 42 saniyelik makul bir sürede sonuçları elde ederken, Ayrık G-PSO ise %1.09'luk hata yüzdesi ile 15 saniyelik kısa bir sürede sonuçları elde etmiştir.

Bu sonuçlar, problemdeki kontrol noktalarının düzenli olup olmamasında çok problemdeki kontrol nokta sayısının fazlalığına ve T_{maks} değerinin büyüklüğüne bağlı olarak çözüm kalitesinin ve performansın değiştiğini gösterir. Çözüm kalitesi açısından bu değişim, İİ-DKA için daha az hissedilirken, Ayrık G-PSO'da kendini daha fazla göstermiştir. Bununla beraber TOP için Ayrık G-PSO'nun performans açısından İİ-DKA'ya göre daha iyi olması, bu modelin ümit verici ve geliştirmeye açık olduğunu gösterir.

6 SONUÇ VE ÖNERİLER

Bu tez kapsamında, en fazla kazanç sağlayan rotayı bulmayı amaçlayan Oryantiring (OP) ve Takım Oryantiring (TOP) Problemleri için Değişken Komşu Arama (DKA) ve Parçacık Sürü Optimizasyon (PSO) tabanlı yeni çözüm modelleri geliştirilmiştir.

Önerilen modeller, OP ve TOP için DKA ve PSO'nun ilk uygulamalarıdır. Literatürde TOP için Tabu Arama gibi farklı meta-sezgisellerle beraber kullanılarak geliştirilen DKA tabanlı modeller vardır. Önerdiğimiz DKA tabanlı modelin sadece DKA'nın uygulaması olması açısından ilk olma özelliğini korumuştur.

DKA tabanlı önerilen model, DKA yönteminin bir versiyonu olan İndirgenmiş-DKA (İ-DKA) yönteminin içiçe yerleştirilmesiyle (İİ-DKA) oluşturulmuştur. PSO tabanlı ise iki model geliştirilmiştir. Bunlardan biri Yarı Ayrık Güçlendirilmiş-PSO, diğeri ise Ayrık Güçlendirilmiş-PSO'dur.

Önerilen modeller, kontrol nokta sayısı 20 ile 102 arasında değişen 453 OP/TOP karşılaştırmalı problem ile test edilmiştir. Sonuçlar, modellerin OP için tüm karşılaştırmalı problemlerde bilinen en iyi sonucu elde ettiğini aynı zamanda 12 sonucu da iyileştirdiğini göstermiştir. TOP için ise modellerin küçük ve orta büyüklükteki problemler için bilinen en iyi sonuçlara küçük hatalarla yada hatasız yaklaştığını, problemdeki kontrol nokta sayısı arttıkça önerilen modellerin bilinen en iyi sonuçlara daha fazla hata oranı ile yaklaştığını göstermiştir. Literatürdeki hiç bir TOP modelinin tüm problemler için en iyi sonucunu bulamadığı göz önüne alındığında önerilen model sonuçlarının literatürdeki sonuçlarla karşılaştırılabilir ve ümit verici nitelikte olduğu söylenebilir.

PSO tabanlı model geliştirilirken ortaya çıkan Güçlendirilmiş-PSO versiyonu, PSO'nun erken yakınsama ve öncü parçacık etrafında detaylı arama yapamama olarak bilinen iki büyük sorununu önemli ölçüde çözmüştür. [Sevklı and Sevilgen, 2010]. OP'un 103 ve TOP'un 138 karşılaştırmalı problem sonucu bunu açıkça gösterirken 100 kontrol noktaya sahip problem kümeleri için erken yakınsama sorunu kısmen devam etmiştir. G-PSO'nun büyük problemler için de sorunsuz çalışabilir

hale getirilmesi gelecek çalışmalarda devam ettirilmesi gereken çalışma konularından biridir.

G-PSO algoritma akışı takip edilerek oluşturulan Yarı Ayrık G-PSO modeli kodlama ve modelin kullanım kolaylığı sağlamasına rağmen problem uzayı ile modelin arama uzayının örtüşmemesinden dolayı deneysel sonuçlarda istenilen verimi verememiştir. Tüm operatörleri ve PSO bileşenlerinin ayrık olarak tanıtıldığı Ayrık G-PSO modeli ise Yarı Ayrık G-PSO'ya göre daha iyi sonuçlar üretmiştir. Ayrıca önerilen Ayrık G-PSO modeli, problemde tanımlı nesnelerin bir kısmının seçilmesi ile çözümün oluşturulabileceği başka kombinatoriyel problemler içinde kullanılabilir bir model olmuştur. Bundan dolayı bir başka çalışma konusu olarak benzer problemler üzerinde Ayrık G-PSO'nun performans incelemesi yapılabilir.

Bu tez çalışmasının bir başka katkısı ise DKA ve PSO tabanlı modellerde kullanılan İ-DKA yönteminin ilk defa bu çalışmada yerel arama için kullanılmış olmasıdır [Sevklı and Sevilgen, 2006], [Sevklı and Sevilgen, 2007]. Çıkış noktası hızlı bir şekilde çözüm alanında küresel arama yapmak olan İ-DKA için iyi sonuç elde ettikçe dinamik olarak küçülen komşuluklar seçilerek bu çalışmada İ-DKA'ya yerel arama özelliği de kazandırılmıştır. Bu yöntem DKA ve PSO içinde başarılı bir şekilde uygulanmıştır. Gelecek çalışma için farklı meta-sezgisel yöntemlerle İ-DKA yerel aramasının işbirliği içinde çalıştırılma yolları araştırılabilir.

Gelecek çalışma konularından bir başkası da geliştirilen modellerin OP veya TOP tabanlı gerçek hayat problemine uygulaması olabilir. Problemdeki kontrol noktalarına belli ziyaret saati kısıtları verilerek problem genişletilebilir. Modellerin gerçek hayat problemleri üzerindeki performansı incelenebilir.

Bir başka çalışma konusu da karşılaştırmalı yeni problem kümeleri oluşturulma ihtiyacıdır. Araç Rotalama Problemlerinin farklı versiyonlarından esinlenerek değişik OP/TOP versiyonları oluşturulabilir.

Diğer çalışma konusu ise modellerde parametre değerlerinin etkisini araştırmaktır. Hangi parametre aralığında ve komşuluk yapılarında modelin daha iyi çalıştığı analizi yapılabilir.

KAYNAKLAR DİZİNİ

- Ai-ling C., Gen-ke, Y., Zhi-ming W. (2006), “Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem”, *Journal of Zhejiang University Science A*, Vol. 7(4), 607-614.
- Archetti, C., Hertz, A., Speranza, M.G. (2007), “Metaheuristics for the team orienteering problem”. *Journal of Heuristics*, Vol.13, 49–76.
- Balas, E. (1989), “The prize collecting traveling salesman problem”, *Networks*, Vol. 19, 621- 636.
- Blum, C., Roli, A. (2003), “Metaheuristic in Combinatorial optimization:Overview and Conceptual Comparison”, *ACM Computing Surveys*, Vol. 35(3), 268-308.
- Bouly, H., Dang, D. C., Moukrim, A. (2008), “A Memetic Algorithm for the Team Orienteering Problem”, *Lecture Notes in Computer Science*, Vol. 4974, 649-658.
- Boussier, S., Feillet, D., Gendreau, M. (2007), “An exact algorithm for team orienteering problems”. *4OR*, Vol. 5, 211-230.
- Butt, S.E., Cavalier, T.M. (1994), “A Heuristic for the multiple path maximum collection problem”, *Computers and Operations Research*, Vol. 21, 101–111.
- Butt, S., Ryan, D. (1999), “An optimal solution procedure for the multiple path maximum collection problem using column generation”, *Computer and Operations Research*, Vol. 26, 427–441.
- Chao, I. M., Golden, B.L., Wasil, E. A.(1996) , “A fast and effective heuristic for the orienteering problem”, *European Journal Of Operational Research.*, Vol. 88(3), 475- 489.
- Clerc M., Kennedy J. (2002), “The particle swarm explosion, stability, and convergence in a multidimensional complex space”, *IEEE Transaction on Evolutionary Computation.*, Vol. 6(1), 58-73.
- Esquivel, S.C., Coello Coello, C.A. (2003), “On the Use of Particle Swarm Optimization with Multimodal Functions”, *Proceeding of IEEE Congress on Evolutionary Computation (CEC’2003)*, Vol. 2, 1130-1136.

- Feillet, D., Dejax, P., Gendreau, M. (2005), "Traveling Salesman Problems with Profits", *Transportation Science*, Vol.39, 188-205.
- Fischetti, M., Gonzalez, J.S., Toth, P. (1998), "Solving the orienteering problem through branch and cut", *INFORMS J. Comput.*, Vol.10(2), 133-148.
- Glover, F. (1986), "Future paths for integer programming and links to artificial intelligence", *Computer Operation Research*, Vol. 13, 533-549.
- Golden, B., Levy, L., Vohra, R. (1987), "The Orienteering Problem". *Naval Res. Logist*, Vol.34(3), 307-318.
- Golden, B.L., Wang, Q., Liu, L. (1988), "A Multifaced heuristic for the orienteering problem", *Naval Res. Logist*, Vol. 35, 359-366.
- Hayes, M., Norman., J.M. (1984), "Dynamic Programming in Orienteering:Route Choice and the Siting of Controls". *J. Oper. Res. Soc.*, Vol. 35(9), 791-796.
- Jones, D.F., Mirrazavi, S.K., Tamiz M. (2002), "Multi objective meta heuristics:An overview of the current state of the art", *European Journal of Operational Research*, Vol. 137, 1-9.
- Kantor, M.G., Rosenwein, M.B. (1992), "The orienteering problem with time windows", *J. Oper. Res. Soc.*, Vol: 43(6) , 629-635.
- Kataoka, S., Morito, S. (1988), "An algorithm for the single constraint maximum collection problem", *J. Oper. Res. Soc.*, Vol. 31(4), 515- 530.
- Ke, L., Archetti, C., Feng, Z. (2008), "Ants can solve team orienteering problem", *Journal Of Computers and Industrial Engineering*, Vol. 54, 648 665.
- Keller, C.P. (1989), "Algorithms to solve orienteering problem: A Comparision", *European Journal of Operational Research*, Vol. 41, 224-231.
- Kennedy, J., Mendes, R. (2002), "Population structure and particle swarm performance", *Proceeding of IEEE Congress on Evolutionary Computation* (Honolulu, Hawaii), Piscataway, NJ., May, Vol. 2, 1671-1676.
- Kennedy, J., Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, 1942—1948.

- Kennedy, J. (1999), "Small worlds and mega minds: Effect of neighborhood topology on particle swarm performance", *Proceeding of IEEE Congress on Evolutionary Computation*, 1931-1938.
- Laporte, G., Martello, S. (1990), "The selective traveling salesman problem". *Discrete Applied Math*, Vol. 26, 193- 207.
- Leifer, A.C., Rosenwein, M. B. (1994) , "Strong Linear Programming relaxations for orienteering problem", *European Journal of Operational Research*., Vol.73, 517- 523.
- Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S. (2006) , "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions" , *IEEE Transaction on Evolutionary Computation*, Vol. 10(3), 281-295.
- Liang, Y.C., Kulturel Konak, S., Smith, A.E. (2002), "Meta Heuristic for the Orienteering Problem", *Proceeding of the Congress on Evolutionary Computation (CEC'02)*. Hawaii, 384-389.
- Liao, C. J., Tseng, C.T., Luarn, P. (2007), "A discrete version of particle swarm optimization for flowshop scheduling problems", *Computers & Operations Research*, Vol. 34(10), 3099- 3111.
- Mendes, R., Kennedy, J., Neves, J. (2004), "The fully informed particle swarm: Simpler, maybe better", *IEEE Transaction on Evolutionary Computation*, Vol. 8, 204-210.
- Mladenovic, N., Hansen, P. (1997), "Variable Neighborhood Search", *Computers & Operations Research*, Vol.24, 1097-1100.
- Osman, I.H., Laporte, G. (1996), "Metaheuristics: A bibliography", *Annual Operation Research*, Vol.63, 513-623.
- Parsopoulos, K.E., Vrahatis, M.N. (2004), "UPSO—A unified particle swarm optimization scheme", *Lecture Series on Computational Sciences*, 868- 873.
- Peer, E.S., Van den Bergh, F., Engelbrecht, A. P. (2003), "Using neighbourhoods with the guaranteed convergence PSO", *Proceeding of Swarm Intelligence Symposium* (Indianapolis, Indiana), 235- 242, Piscataway, NJ., IEEE Service Center

- Peram T., Veeramachaneni K., Mohan C.K. (2003), "Fitness distance ratio based particle swarm optimization", *Proceeding of. Swarm Intelligence Symposium*, 174-181.
- Ramesh, R., Brown, K.M. (1991), "An efficient four phase heuristic for the generalized orienteering problem", *Computers & Operations Research*, Vol.18(2), 151-165.
- Ramesh, R., Yoon, Y.S., Karwan, M.H. (1992), "An optimal algorithm for the orienteering problem", *ORSA J. Comput.*, Vol.4(42), 155-165.
- Rameshkumar, K., Suresh, R.K., Mohanasundaram, K.M. (2005) "Discrete Particle Swarm Optimization (DPSO) Algorithm for Permutation Flowshop Scheduling to Minimize Makespan", *ICNC*, Vol.3, 572-581.
- Rushton A., Croucher P., Baker P. (2006), "*Handbook of Logistics and Distribution Management*", 3rd Edition, Kogan Page, Limited
- Sevкли Z., Sevilgen E. (2006) "Variable Neighborhood Search for the Orienteering Problem", *ISCIS'06 , Lecture Notes in Computer Science*, Vol. 4263, 134- 143
- Sevкли Z., Sevilgen E, (2007), "Particle Swarm Optimization for the Orienteering Problem", *Proceedings of International Symposium on Innovation in Intelligent Systems and Applications (INISTA'07)*. Istanbul,Turkiye, 185-190.
- Sevкли Z., Sevilgen E, (2010) "StPSO: Strengthened Particle Swarm Optimization", *Turkish Journal Of Electrical Engineering & Computer Sciences*, Vol.3 (basımda)
- Shi, Y., Eberhart, R.C. (1998), "A modified particle swarm optimizer", *Proceeding of the Congress on Evolutionary Computation*, 69-73.
- Souffriau, W, Vansteenwegen, P., Berghe G.V., Oudheusden D.V. (2009), "A Path Relinking approach for the Team Orienteering Problem", *Computers & Operations Research* , doi: 10.1016/j.cor. 2009.05.02.
- Stützle, T. (1999) "Local Search Algorithms for Combinatorial Problems Analysis, Algorithms and New Applications", *DISKI Dissertationenzur Künstliken Intelligenz*. Infix, Sankt Augustin, Germany.

- Suganthan, P. N., (1999), "Particle swarm optimizer with neighborhood operator", *Proceeding of the Congress on Evolutionary Computation*, Washington, DC, 1958-1962.
- Tang, H., Miller Hooks, E. (2005), "A tabu search heuristic for the team orienteering problem", *Computers & Operations Research*, Vol.32, 1379–1407.
- Tasgetiren, F.M., Smith, A.E. (2000), "A genetic algorithm for the orienteering problem", *Proceeding of the Congress on Evolutionary Computation*, San Diego, CA. Vol.2, 910-915.
- Tasgetiren M.F., Sevkli, M., Liang Y.C., and Gencyilmaz, G. (2004), "Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem", *Lecture Notes in Computer Science*, Vol. 3172 , 382-389.
- Tasgetiren, M.F., Suganthan, P.N., Pan, Q.Q., (2007a), "A discrete particle swarm optimization algorithm for the generalized traveling salesman problem", *Proceedings of the 9th Annual Conference on Genetic and Evolutionary computation*, GECCO '07, 158-167.
- Tasgetiren M.F., Liang Y.C., Sevkli M., Gencyilmaz G. (2007b), "Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in Permutation Flowshop Sequencing Problem", *European Journal of Operational Research*, Vol. 177(3), 1930-1947
- Tsiligirides, T. (1984) , "Heuristic methods applied to orienteering". *J. Oper. Res. Soc.*, Vol. 35(9), 797-809.
- Van den Berg, F. and Engelbrecht, A. (2002), "A new locally convergent particle swarm optimizer", *Proceeding of IEEE Conference on System, Man and Cybernetics*, Oct. , 96-101.
- Vansteenwegen, P., Souffriau, W., Berghe G.V., Oudheusden D.V. (2009), "A guided local search metaheuristic for team orienteering problem", *European Journal of Operational Research*, Vol. 196, 118-127.
- Vansteenwegen, P., Van Oudheusden, D. (2007), "The mobile tourist guide". *An OR opportunity. OR Insight*, Vol:20 (7), 21 7.

- Voß, S., Martello, S., Osman, I. H., Roucairol, C. (1999), “*Meta Heuristics Advances and Trends in Local Search Paradigms for Optimization*”, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Wang, Q., Sun, X., Golden, B. L. , Jia, J . (1995), “Using artificial neural network to solve orienteering problem”, *Ann. Oper. Res.*, Vol. 61 ,111-120.
- Zhan, C., Ning J., Lu S., Ouyang D., Ding T. (2009), “A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization”, *Operations Research Letters*, Vol.37, 117-122.
- Zhu, Q., Qian, L., Li, Y., Zhu, S. (2006), “An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows”, *Proceedings of the Congress on Evolutionary Computation*, CEC 2006, 1386 – 1390.

ÖZGEÇMİŞ

1978 yılında Sivas'ta, doğdu. İlk, orta ve lise tahsilini Samsun'da tamamladı. 2000 yılında İstanbul Üniversitesi Bilgisayar Mühendisliği bölümünden mezun oldu. 2003 yılında Fatih Üniversitesi Bilgisayar Mühendisliği bölümünden yüksek lisansını tamamladı.

Halen Fatih Üniversitesi Bilgisayar Mühendisliği bölümünde Öğretim Görevlisi olarak çalışmaktadır.

EKLER

EK A: OP ve TOP İÇİN KULLANILAN PROBLEM KÜMELERİ

Çizelge A.1 Problem kümeleri 1,2,3 ve 1_1'in kontrol noktalarının koordinat bilgileri

PK 1 N:32			PK 2 N: 21			PK 3 N: 33			PK 1_1 N:32		
X	Y	Skor	X	Y	Skor	X	Y	Skor	X	Y	Skor
10.5	14.4	0	4.6	7.1	0	19.1	24.3	0	10.5	14.4	0
18	15.9	10	5.7	11.4	20	12.6	24.9	20	18	15.9	10
18.3	13.3	10	4.4	12.3	20	14.4	28	20	18.3	13.3	10
16.5	9.3	10	2.8	14.3	30	16.9	28.1	20	16.5	9.3	10
15.4	11	10	3.2	10.3	15	20.7	28.2	20	15.4	11	10
14.9	13.2	5	3.5	9.8	15	12.5	26.6	20	14.9	13.2	5
16.3	13.3	5	4.4	8.4	10	21.8	27.3	20	16.3	13.3	5
16.4	17.8	5	7.8	11	20	12.5	22.6	20	16.4	17.8	5
15	17.9	5	8.8	9.8	20	22.5	17	30	15	17.9	5
16.1	19.6	10	7.7	8.2	20	19.9	15	30	16.1	19.6	10
15.7	20.6	10	6.3	7.9	15	14.9	15.1	30	15.7	20.6	10
13.2	20.1	10	5.4	8.2	10	11.5	18.6	30	13.2	20.1	10
14.3	15.3	5	5.8	6.8	10	12.4	29.8	30	14.3	15.3	5
14	5.1	10	6.7	5.8	25	17.8	28.1	30	14	5.1	10
11.4	6.7	15	13.8	13.1	40	9.1	29.8	40	11.4	6.7	15
8.3	5	15	14.1	14.2	40	10	32.6	40	8.3	5	15
7.9	9.8	10	11.2	13.6	30	13.9	33.1	40	7.9	9.8	10
11.4	12	5	9.7	16.4	30	19.95	10.3	40	11.4	12	5
11.2	17.6	5	9.5	18.8	50	15.2	8	40	11.2	17.6	5
10.1	18.7	5	4.7	16.8	30	14.7	31.2	50	10.1	18.7	5
11.7	20.3	10	5	5.6	0	7.4	36.5	50	11.7	20.3	10
10.2	22.1	10				21	25.5	50	10.2	22.1	10
9.7	23.8	10				18	25.3	10	9.7	23.8	10
10.1	26.4	15				19.5	24.7	10	10.1	26.4	15
7.4	24	15				21.4	21.8	10	7.4	24	15
8.2	19.9	15				16	21.4	10	8.2	19.9	15
8.7	17.7	10				18.65	26.2	10	8.7	17.7	10
8.9	13.6	10				17.9	28.9	10	8.9	13.6	10
5.6	11.1	10				14.3	19.9	20	5.6	11.1	10
4.9	18.9	10				17	19	20	4.9	14.9	10
7.3	18.8	10				10.8	21	20	7.3	18.8	10
11.2	14.1	0				15.7	23.7	10	11.2	14.1	0
						18.2	24	0			

Çizelge A.2 Problem kümesi 4'ün kontrol noktalarının koordinat bilgileri

PK 4 N:100								
X	Y	Skor	X	Y	Skor	X	Y	Skor
18.19	6.32	0	0.43	1.12	9	20.30	3.86	2
15.52	28.03	7	13.57	9.41	11	17.02	12.33	17
9.00	28.01	5	0.30	9.16	7	0.20	24.80	3
16.93	2.09	24	0.45	23.35	10	13.97	18.49	4
8.64	19.85	3	22.69	23.13	13	1.88	3.33	13
18.43	29.93	23	19.12	20.20	13	14.25	23.83	1
3.94	10.77	3	7.00	21.44	8	27.22	11.98	16
14.78	7.61	26	28.88	0.86	24	22.05	7.74	10
2.41	25.06	24	19.60	2.11	14	3.08	21.03	13
18.41	11.88	16	22.78	15.42	27	12.66	26.02	3
14.91	17.24	21	13.51	8.05	1	3.69	14.33	21
7.14	0.48	6	0.51	11.06	5	26.93	13.06	24
13.44	24.76	17	4.31	23.12	20	1.57	19.84	19
17.54	29.43	27	19.68	25.62	5	19.01	14.36	13
16.71	9.50	27	16.32	3.73	3	14.24	18.80	19
12.38	26.66	1	7.14	25.90	13	20.49	13.59	8
25.69	6.34	6	1.04	13.47	6	3.30	17.86	1
11.62	24.00	27	17.46	2.56	28	28.72	20.92	29
10.24	3.02	23	23.49	17.75	5	7.49	6.59	24
3.54	26.16	18	11.96	15.15	17	12.19	16.11	1
17.86	19.36	13	22.85	14.85	2	17.52	0.97	10
3.77	7.84	1	28.47	6.08	23	10.35	6.18	11
14.20	19.64	13	6.52	19.42	17	13.61	21.12	17
15.88	5.54	21	26.85	25.82	6	0.26	14.37	13
1.32	19.32	23	21.48	10.30	21	12.58	26.64	16
20.50	25.33	16	29.08	2.62	18	21.57	24.07	2
16.76	19.84	8	22.54	18.26	14	14.50	0.30	22
27.50	17.66	24	4.46	7.85	7	25.60	22.91	2
19.41	20.67	15	21.61	23.16	4	10.92	4.13	8
20.49	14.03	24	6.82	21.10	13	26.65	20.04	28
20.90	7.67	8	6.48	2.46	25	18.63	4.30	24
2.75	25.40	5	26.70	3.44	27	7.29	16.28	12
16.10	19.74	3	7.03	28.12	1	4.34	9.51	5
						2.38	18.26	0

Çizelge A.3 Problem kümesi 5 ve 6'nın kontrol noktalarının koordinat bilgileri

PK 5 N:66						PK 6 N:64					
X	Y	Skor	X	Y	Skor	X	Y	Skor	X	Y	Skor
0.5	0	0	1	7	35	0	7	0	1	0	24
7	7	35	1	5	25	1	6	6	3	0	30
7	5	35	1	3	15	1	6	6	5	0	36
7	3	35	1	1	5	2	5	12	7	0	42
7	1	35	1	1	5	0	5	6	6	1	36
7	1	35	1	3	15	2	5	12	4	1	30
7	3	35	1	5	25	3	4	18	2	1	24
7	5	35	1	7	35	1	4	12	0	1	18
7	7	35	3	7	35	1	4	12	2	1	24
5	7	35	3	5	25	3	4	18	4	1	30
5	5	25	3	3	15	4	3	24	6	1	36
5	3	25	3	1	15	2	3	18	5	2	30
5	1	25	3	1	15	0	3	12	3	2	24
5	1	25	3	3	15	2	3	18	1	2	18
5	3	25	3	5	25	4	3	24	1	2	18
5	5	25	3	7	35	5	2	30	3	2	24
5	7	35	5	7	35	3	2	24	5	2	30
3	7	35	5	5	25	1	2	18	4	3	24
3	5	25	5	3	25	1	2	18	2	3	18
3	3	15	5	1	25	3	2	24	0	3	12
3	1	15	5	1	25	5	2	30	2	3	18
3	1	15	5	3	25	6	1	36	4	3	24
3	3	15	5	5	25	4	1	30	3	4	18
3	5	25	5	7	35	2	1	24	1	4	12
3	7	35	7	7	35	0	1	18	1	4	12
1	7	35	7	5	35	2	1	24	3	4	18
1	5	25	7	3	35	4	1	30	2	5	12
1	3	15	7	1	35	6	1	36	0	5	6
1	1	5	7	1	35	7	0	42	2	5	12
1	1	5	7	3	35	5	0	36	1	6	6
1	3	15	7	5	35	3	0	30	1	6	6
1	5	25	7	7	35	1	0	24	0	7	0
1	7	35	0.5	0	0						

Çizelge A. 4 Problem kümesi 7'nin kontrol noktalarının koordinat bilgileri

PK 7 N:102								
X	Y	Skor	X	Y	Skor	X	Y	Skor
35	35	0	65	55	14	45	65	9
15	30	26	5	5	16	65	35	3
55	5	29	11	14	18	65	20	6
31	52	27	6	38	16	64	42	9
60	12	31	47	47	13	63	65	8
8	56	27	37	31	14	2	60	5
13	52	36	57	29	18	20	20	8
6	68	30	36	26	18	40	25	9
21	24	28	12	24	13	42	7	5
56	39	36	24	58	19	24	12	5
55	54	26	62	77	20	23	3	7
16	22	41	49	73	25	2	48	1
4	18	35	67	5	25	49	58	10
28	18	26	57	68	15	27	43	9
26	27	27	47	16	25	63	23	2
55	45	13	49	11	18	53	12	6
55	20	19	49	42	13	32	12	7
55	60	16	53	43	14	17	34	3
30	60	16	57	48	23	27	69	10
20	65	12	15	47	16	15	77	9
50	35	19	14	37	11	37	47	6
30	25	23	26	35	15	37	56	5
15	10	20	18	24	22	44	17	9
10	20	19	25	24	20	46	13	8
20	40	12	22	27	11	61	52	3
15	60	17	25	21	12	56	37	6
45	20	11	18	18	17	11	31	7
45	10	18	41	49	10	26	52	9
45	30	17	35	17	7	31	67	3
35	40	16	25	30	3	15	19	1
41	37	16	20	50	5	22	22	2
40	60	21	10	43	9	19	21	10
35	69	23	30	5	8	20	26	9
53	52	11	5	30	2	35	35	0

EK B: OP İÇİN İİ-DKA MODELİ İLE ELDE EDİLEN SONUÇLAR

Çizelge B.1 PK 1 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmaks	Optimum	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
		En iyi	En kötü	Ort	StDev	Ort	Ort
5	10	10	10	10.0	0.00	6254.8	0.05
10	15	15	15	15.0	0.00	4153.7	0.05
15	45	45	45	45.0	0.00	5207.7	0.08
20	65	65	65	65.0	0.00	10153.5	0.16
25	90	90	90	90.0	0.00	23919.9	0.42
30	110	110	110	110.0	0.00	6913.8	0.15
35	135	135	135	135.0	0.00	11038.8	0.25
40	155	155	150	154.5	1.58	34637.7	0.96
46	175	175	175	175.0	0.00	14605.2	0.53
50	190	190	190	190.0	0.00	22704.2	0.85
55	205	205	200	200.5	1.58	12717.3	0.52
60	225	225	220	224.0	2.11	44353.1	2.09
65	240	240	240	240.0	0.00	18547.4	0.96
70	260	260	260	260.0	0.00	16003.3	0.90
73	265	265	265	265.0	0.00	5493.8	0.31
75	270	270	270	270.0	0.00	7153.7	0.45
80	280	280	280	280.0	0.00	9322.7	0.64
85	285	285	285	285.0	0.00	4653.9	0.32
167.78		167.78	166.94	167.44	0.29	14324.14	0.54

Çizelge B.2 PK 2 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmax	Optimum	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
		En iyi	En kötü	Ort	StDev	Ort	Ort
15	120	120	120	120.0	0.00	2165.2	0.06
20	200	200	200	200.0	0.00	5950.8	0.21
23	210	210	210	210.0	0.00	10843.2	0.40
25	230	230	230	230.0	0.00	5118.6	0.19
27	230	230	230	230.0	0.00	2086.9	0.07
30	265	265	265	265.0	0.00	45642.0	1.81
32	300	300	300	300.0	0.00	14733.8	0.65
35	320	320	320	320.0	0.00	48859.4	2.08
38	360	360	360	360.0	0.00	30315.8	1.28
40	395	395	395	395.0	0.00	19735.4	0.90
45	450	450	450	450.0	0.00	14802.7	0.71
280.00		280.00	280.00	280.00	0.00	18204.89	0.76

Çizelge B.3 PK 3 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmax	Optimum	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
		En iyi	En kötü	Ort	StDev	Ort	Ort
15	170	170	170	170.0	0.00	5000.2	0.08
20	200	200	200	200.0	0.00	9528.7	0.16
25	260	260	260	260.0	0.00	13575.2	0.27
30	320	320	320	320.0	0.00	7683.1	0.17
35	390	390	390	390.0	0.00	8286.0	0.21
40	430	430	430	430.0	0.00	9920.9	0.28
45	470	470	470	470.0	0.00	10910.6	0.36
50	520	520	520	520.0	0.00	13668.0	0.49
55	550	550	550	550.0	0.00	12584.8	0.50
60	580	580	580	580.0	0.00	11251.3	0.49
65	610	610	610	610.0	0.00	9492.0	0.45
70	640	640	640	640.0	0.00	23013.1	1.21
75	670	670	670	670.0	0.00	29586.0	1.73
80	710	710	700	703.0	4.83	20844.2	1.27
85	740	740	730	738.0	4.22	21515.4	1.42
90	770	770	770	770.0	0.00	18350.5	1.28
95	790	790	790	790.0	0.00	6405.1	0.44
100	800	800	800	800.0	0.00	7724.7	0.56
105	800	800	800	800.0	0.00	3606.0	0.25
110	800	800	800	800.0	0.00	4191.3	0.30
561.00		561.00	560.00	560.55	0.45	12356.86	0.60

Çizelge B.4 PK 1_1 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmax	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
	En iyi	En kötü	Ort	StDev	Ort	Ort
5	10	10	10.0	0.00	6254.8	0.05
10	15	15	15.0	0.00	4153.7	0.05
15	45	45	45.0	0.00	5218	0.08
20	65	65	65.0	0.00	7929.6	0.13
25	90	90	90.0	0.00	24708.3	0.52
30	110	110	110.0	0.00	6926.3	0.14
35	135	135	135.0	0.00	11642.4	0.28
40	155	155	154.5	1.58	32596.2	0.91
46	175	175	175.0	0.00	11773.4	0.38
50	190	190	190.0	0.00	11699.8	0.42
55	205	205	203.5	2.42	30568.6	1.33
60	225	225	225.0	0.00	52547.3	2.42
65	240	240	240.0	0.00	22599.4	1.22
70	260	260	260.0	0.00	10818.1	0.69
73	265	265	265.0	0.00	9696.1	0.55
75	275	275	275.0	0.00	14253.5	0.95
80	280	280	280.0	0.00	6126.3	0.39
85	285	285	285.0	0.00	5435	0.38
168.06		168.06	167.94	0.22	15274.82	0.60

Çizelge B.5 PK 5 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmax	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
	En iyi	En kötü	Ort	StDev	Ort	Ort
5	10	10	10.0	0.00	18092.7	0.16
10	40	40	40.0	0.00	13105.3	0.16
15	120	120	120.0	0.00	124028.1	1.03
20	205	205	205.0	0.00	107377.4	1.17
25	290	290	290.0	0.00	80486.1	1.11
30	400	400	400.0	0.00	100903.8	1.74
35	465	460	463.0	2.58	110619.7	2.28
40	575	545	572.0	9.49	134818.5	3.45
45	650	635	648.0	4.83	190835.3	5.81
50	730	710	724.0	8.43	264372.9	9.76
55	825	795	808.5	12.26	342030.2	15.93
60	915	875	902.0	11.60	275133.1	14.30
65	980	950	971.0	9.66	359027.3	21.95
70	1070	1030	1053.0	15.67	285751.3	20.99
75	1140	1100	1129.0	11.25	280477.6	22.55
80	1215	1155	1196.5	16.34	252767.7	24.29
85	1270	1255	1261.5	6.26	314934.8	34.07
90	1340	1310	1321.5	8.18	310685.1	37.32
95	1395	1370	1378.5	8.51	238940.8	31.81
100	1465	1425	1440.0	13.54	274456.2	41.46
105	1520	1485	1506.5	11.32	289772.9	48.19
110	1560	1530	1548.5	8.83	208142.3	37.49
115	1595	1575	1587.5	5.89	193032.5	38.28
120	1635	1615	1626.0	5.68	231305.8	51.35
125	1670	1655	1662.0	6.32	128036.9	30.03
130	1680	1680	1680.0	0.00	75094.8	20.35
	952.31	931.54	944.00	6.79	200162.66	19.89

Çizelge B.6 PK 6 için İİ-DKA modeli ile elde edilen OP sonuçları

Tmax	Amaç Fonksiyon Değeri				Amaç Fonk.Sayısı	CPU
	En iyi	En kötü	Ort	StDev	Ort	Ort
15	96	96	96.0	0.00	50180.8	0.42
20	294	294	294.0	0.00	23426.9	0.39
25	390	390	390.0	0.00	57439.0	1.20
30	474	468	472.8	2.53	143650.8	3.87
35	576	558	571.2	5.51	225618.3	7.90
40	714	702	707.4	5.25	281985.6	13.75
45	816	798	806.4	8.10	282507.2	18.24
50	900	882	891.0	6.48	341505.1	25.10
55	984	966	977.4	4.43	184193.0	16.49
60	1062	1044	1050.0	7.48	253932.0	27.53
65	1116	1098	1112.4	6.45	284348.1	38.07
70	1188	1182	1185.0	3.16	175840.8	25.82
75	1236	1224	1231.8	4.05	235082.3	40.69
80	1278	1272	1272.6	1.90	136191.5	26.39
	794.57	783.86	789.86	3.95	191135.81	17.56

**EK C: TOP İÇİN İİ-DKA MODELİ İLE ELDE EDİLEN
SONUÇLAR**

Çizelge C.1 PK 1 için İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p1.2.b	5	15	15	15	0.00	0.00
p1.2.c	7.5	20	20	20	0.00	0.00
p1.2.d	10	30	30	30	0.00	0.78
p1.2.e	12.5	45	45	45	0.00	1.80
p1.2.f	15	80	80	80	0.00	2.43
p1.2.g	17.5	90	90	90	0.00	2.84
p1.2.h	20	110	110	110	0.00	4.43
p1.2.i	23	135	130	131.5	2.42	13.21
p1.2.j	25	155	155	155	0.00	8.10
p1.2.k	27.5	175	170	171.5	2.42	9.79
p1.2.l	30	195	190	192	2.58	9.13
p1.2.m	32.5	215	205	214	3.16	8.58
p1.2.n	35	235	235	235	0.00	7.18
p1.2.o	36.5	240	240	240	0.00	5.08
p1.2.p	37.5	250	250	250	0.00	7.56
p1.2.q	40	265	265	265	0.00	2.82
p1.2.r	42.5	280	275	279.5	1.58	30.72
p1.3.c	5	15	15	15	0.00	0.00
p1.3.d	6.7	15	15	15	0.00	0.00
p1.3.e	8.3	30	30	30	0.00	0.00
p1.3.f	10	40	40	40	0.00	0.31
p1.3.g	11.7	50	50	50	0.00	1.23
p1.3.h	13.3	70	70	70	0.00	1.40
p1.3.i	15.3	105	105	105	0.00	3.50
p1.3.j	16.7	115	115	115	0.00	2.20
p1.3.k	18.3	135	135	135	0.00	12.08
p1.3.l	20	155	155	155	0.00	3.70
p1.3.m	21.7	175	175	175	0.00	4.13
p1.3.n	23.3	190	190	190	0.00	3.41
p1.3.o	24.3	205	205	205	0.00	7.31
p1.3.p	25	220	220	220	0.00	4.02
p1.3.q	26.7	230	230	230	0.00	11.14
p1.3.r	28.3	250	250	250	0.00	4.03
p1.4.d	5	15	15	15	0.00	0.00
p1.4.e	6.2	15	15	15	0.00	0.00
p1.4.f	7.5	25	25	25	0.00	0.00
p1.4.g	8.8	35	35	35	0.00	0.00
p1.4.h	10	45	45	45	0.00	0.00
p1.4.i	11.5	60	60	60	0.00	1.34
p1.4.j	12.5	75	75	75	0.00	1.50
p1.4.k	13.8	100	100	100	0.00	1.58
p1.4.l	15	120	120	120	0.00	1.62
p1.4.m	16.2	130	130	130	0.00	2.36
p1.4.n	17.5	155	155	155	0.00	2.67
p1.4.o	18.2	165	165	165	0.00	2.86
p1.4.p	18.8	175	175	175	0.00	1.87
p1.4.q	20	190	190	190	0.00	2.58
p1.4.r	21.2	210	210	210	0.00	6.37

Çizelge C.2 PK 2 için İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p2.2.a	7.5	90	90	90	0.00	0.00
p2.2.b	10	120	120	120	0.00	0.00
p2.2.c	11.5	140	140	140	0.00	1.11
p2.2.d	12.5	160	160	160	0.00	1.13
p2.2.e	13.5	190	190	190	0.00	1.24
p2.2.f	15	200	200	200	0.00	1.16
p2.2.g	16	200	200	200	0.00	1.02
p2.2.h	17.5	230	230	230	0.00	1.19
p2.2.i	19	230	230	230	0.00	1.19
p2.2.j	20	260	260	260	0.00	1.64
p2.2.k	22.5	275	275	275	0.00	2.01
p2.3.a	5	70	70	70	0.00	0.00
p2.3.b	6.7	70	70	70	0.00	0.00
p2.3.c	7.7	105	105	105	0.00	0.00
p2.3.d	8.3	105	105	105	0.00	0.00
p2.3.e	9	120	120	120	0.00	0.00
p2.3.f	10	120	120	120	0.00	0.00
p2.3.g	10.7	145	145	145	0.00	1.13
p2.3.h	11.7	165	165	165	0.00	1.56
p2.3.i	12.7	200	200	200	0.00	1.29
p2.3.j	13.3	200	200	200	0.00	0.90
p2.3.k	15	200	200	200	0.00	0.00
p2.4.a	3.8	10	10	10	0.00	0.00
p2.4.b	5	70	70	70	0.00	0.00
p2.4.c	5.8	70	70	70	0.00	0.00
p2.4.d	6.2	70	70	70	0.00	0.00
p2.4.e	6.8	70	70	70	0.00	0.00
p2.4.f	7.5	105	105	105	0.00	0.00
p2.4.g	8	105	105	105	0.00	0.00
p2.4.h	8.8	120	120	120	0.00	0.00
p2.4.i	9.5	120	120	120	0.00	0.00
p2.4.j	10	120	120	120	0.00	0.00
p2.4.k	11.2	180	180	180	0.00	1.26

Çizelge C.3 PK 3'te 2 ve 3 araçlı problemler için İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p3.2.a	7.5	90	90	90	0.00	0.00
p3.2.b	10	150	150	150	0.00	1.06
p3.2.c	12.5	180	180	180	0.00	1.26
p3.2.d	15	220	220	220	0.00	1.45
p3.2.e	17.5	260	260	260	0.00	12.83
p3.2.f	20	300	290	299	3.16	5.12
p3.2.g	22.5	360	360	360	0.00	11.90
p3.2.h	25	410	390	405	8.50	35.94
p3.2.i	27.5	460	460	460	0.00	17.59
p3.2.j	30	510	510	510	0.00	17.41
p3.2.k	32.5	550	550	550	0.00	13.24
p3.2.l	35	590	590	590	0.00	2.83
p3.2.m	37.5	620	610	619	3.16	22.89
p3.2.n	40	660	630	654	12.65	8.10
p3.2.o	42.5	690	680	689	3.16	17.92
p3.2.p	45	720	720	720	0.00	10.45
p3.2.q	47.5	760	750	758	4.22	36.76
p3.2.r	50	790	780	787	4.83	16.61
p3.2.s	52.5	800	800	800	0.00	3.31
p3.2.t	55	800	800	800	0.00	2.13
p3.3.b	6.7	90	90	90	0.00	0.00
p3.3.c	8.3	120	120	120	0.00	0.00
p3.3.d	10	170	170	170	0.00	0.56
p3.3.e	11.7	200	200	200	0.00	1.34
p3.3.f	13.3	230	230	230	0.00	1.45
p3.3.g	15	270	270	270	0.00	1.66
p3.3.h	16.7	300	300	300	0.00	1.74
p3.3.i	18.3	330	330	330	0.00	2.51
p3.3.j	20	380	380	380	0.00	2.01
p3.3.k	21.7	440	440	440	0.00	3.01
p3.3.l	23.3	480	480	480	0.00	3.41
p3.3.m	25	520	520	520	0.00	13.29
p3.3.n	26.7	570	560	569	3.16	24.18
p3.3.o	28.3	590	580	589	3.16	17.33
p3.3.p	30	640	640	640	0.00	4.38
p3.3.q	31.7	680	680	680	0.00	3.67
p3.3.r	33.3	710	710	710	0.00	2.03
p3.3.s	35	720	720	720	0.00	5.23
p3.3.t	36.7	760	760	760	0.00	3.33

Çizelge C.4 PK 3 için 4 araçlı problemlerinde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p3.4.c	6.2	90	90	90	0.00	0.00
p3.4.d	7.5	100	100	100	0.00	0.00
p3.4.e	8.8	140	140	140	0.00	0.00
p3.4.f	10	190	190	190	0.00	1.22
p3.4.g	11.2	220	220	220	0.00	1.42
p3.4.h	12.5	240	240	240	0.00	1.55
p3.4.i	13.8	270	270	270	0.00	1.60
p3.4.j	15	310	310	310	0.00	1.75
p3.4.k	16.2	350	350	350	0.00	1.84
p3.4.l	17.5	380	380	380	0.00	2.90
p3.4.m	18.8	390	390	390	0.00	2.38
p3.4.n	20	440	440	440	0.00	5.45
p3.4.o	21.2	500	500	500	0.00	18.07
p3.4.p	22.5	560	560	560	0.00	2.25
p3.4.q	23.8	560	560	560	0.00	3.58
p3.4.r	25	600	600	600	0.00	3.43
p3.4.s	26.2	670	670	670	0.00	4.09
p3.4.t	27.5	670	670	670	0.00	2.58

Çizelge C.5 PK 4 içinde 2 araçlı problemler için İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p4.2.a	25	206	206	206	0.00	11.15
p4.2.b	30	341	321	339	6.32	10.82
p4.2.c	35	452	440	447.2	5.27	16.82
p4.2.d	40	531	506	524.7	9.76	132.31
p4.2.e	45	618	572	600.8	14.60	114.85
p4.2.f	50	679	665	674.5	4.09	46.06
p4.2.g	55	751	698	734.1	14.70	63.73
p4.2.h	60	835	759	799.6	22.41	91.69
p4.2.i	65	918	833	872.6	25.53	51.12
p4.2.j	70	962	913	931.3	17.08	77.73
p4.2.k	75	1022	957	984.7	18.29	108.51
p4.2.l	80	1073	993	1035.1	25.13	91.65
p4.2.m	85	1132	1020	1095.3	36.25	86.54
p4.2.n	90	1170	1089	1126.8	28.14	129.88
p4.2.o	95	1202	1139	1173.4	26.88	90.26
p4.2.p	100	1234	1154	1197.9	24.47	156.45
p4.2.q	105	1265	1224	1247.7	12.58	151.05
p4.2.r	110	1280	1176	1254.2	28.85	122.96
p4.2.s	115	1296	1268	1277.8	9.59	217.52
p4.2.t	120	1304	1241	1290.7	18.33	179.60

Çizelge C.6 PK 4 için 3 ve 4 araçlı problemlerde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p4.3.b	20	38	38	38	0.00	0.00
p4.3.c	23.3	193	193	193	0.00	2.27
p4.3.d	26.7	335	323	332.6	3.69	12.43
p4.3.e	30	468	444	454.9	11.44	16.01
p4.3.f	33.3	579	556	569	9.42	64.88
p4.3.g	36.7	653	618	644	11.76	40.76
p4.3.h	40	729	654	719.8	23.39	71.37
p4.3.i	43.3	809	744	789	27.83	80.39
p4.3.j	46.7	861	820	846.2	14.41	54.82
p4.3.k	50	919	826	867.7	24.44	63.79
p4.3.l	53.3	977	899	950.4	27.60	80.69
p4.3.m	56.7	1051	989	1011.5	17.39	95.75
p4.3.n	60	1109	1034	1068.8	30.21	84.39
p4.3.o	63.3	1170	1090	1133.5	28.31	188.47
p4.3.p	66.7	1222	1133	1171.4	26.51	315.32
p4.3.q	70	1245	1197	1217.6	13.62	156.23
p4.3.r	73.3	1270	1195	1242.6	21.08	211.68
p4.3.s	76.7	1294	1209	1267.7	24.70	188.20
p4.3.t	80	1303	1262	1286	13.70	207.42
p4.4.d	20	38	38	38	0.00	0.00
p4.4.e	22.5	183	183	183	0.00	1.17
p4.4.f	25	324	324	324	0.00	3.36
p4.4.g	27.5	461	460	460.8	0.42	12.15
p4.4.h	30	571	541	560.6	11.77	29.16
p4.4.i	32.5	657	635	644.5	7.35	34.00
p4.4.j	35	732	698	720.4	11.59	48.85
p4.4.k	37.5	821	765	801.3	19.40	42.18
p4.4.l	40	880	831	866.4	17.58	74.53
p4.4.m	42.5	919	876	908	12.12	68.38
p4.4.n	45	968	894	946.7	22.85	68.28
p4.4.o	47.5	1061	957	1009.6	29.32	102.10
p4.4.p	50	1124	1048	1091.8	28.12	100.76
p4.4.q	52.5	1161	1060	1116.6	41.33	62.58
p4.4.r	55	1195	1092	1170.4	31.80	43.05
p4.4.s	57.5	1250	1180	1209.3	22.40	52.07
p4.4.t	60	1282	1208	1249.4	19.43	144.32

Çizelge C.7 PK 5 için 2 ve 3 araçlı problemlerde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p5.2.b	5	20	20	20	0.00	0.00
p5.2.c	7.5	50	50	50	0.00	1.14
p5.2.d	10	80	80	80	0.00	1.27
p5.2.e	12.5	180	180	180	0.00	14.92
p5.2.f	15	240	240	240	0.00	3.39
p5.2.g	17.5	320	315	317	2.58	41.82
p5.2.h	20	410	395	398	6.32	14.88
p5.2.i	22.5	480	480	480	0.00	24.21
p5.2.j	25	580	580	580	0.00	6.81
p5.2.k	27.5	670	665	668.5	2.42	4.20
p5.2.l	30	800	755	795.5	14.23	37.27
p5.2.m	32.5	860	850	859	3.16	4.98
p5.2.n	35	925	925	925	0.00	12.54
p5.2.o	37.5	1020	1010	1017	4.83	47.90
p5.2.p	40	1150	1090	1126	30.98	6.37
p5.2.q	42.5	1195	1190	1191	2.11	12.01
p5.2.r	45	1260	1260	1260	0.00	30.59
p5.2.s	47.5	1340	1300	1326	17.13	91.95
p5.2.t	50	1400	1400	1400	0.00	51.87
p5.2.u	52.5	1460	1425	1456.5	11.07	53.85
p5.2.v	55	1505	1500	1501.5	2.42	58.25
p5.2.w	57.5	1565	1560	1561	2.11	75.84
p5.2.x	60	1595	1590	1591	2.11	39.84
p5.2.y	62.5	1635	1620	1633.5	4.74	68.26
p5.2.z	65	1670	1650	1663	7.15	82.75
p5.3.b	3.3	15	15	15	0.00	0.00
p5.3.c	5	20	20	20	0.00	0.00
p5.3.d	6.7	60	60	60	0.00	0.55
p5.3.e	8.3	95	95	95	0.00	1.24
p5.3.f	10	110	110	110	0.00	1.39
p5.3.g	11.7	185	185	185	0.00	1.85
p5.3.h	13.3	260	260	260	0.00	2.54
p5.3.i	15	335	335	335	0.00	3.55
p5.3.j	16.7	470	470	470	0.00	5.06
p5.3.k	18.3	495	495	495	0.00	9.21
p5.3.l	20	595	595	595	0.00	16.24
p5.3.m	21.7	650	650	650	0.00	17.04
p5.3.n	23.3	755	755	755	0.00	4.84
p5.3.o	25	870	870	870	0.00	14.01
p5.3.p	26.7	990	990	990	0.00	4.18
p5.3.q	28.3	1070	1065	1067	2.58	39.51
p5.3.r	30	1125	1125	1125	0.00	9.19
p5.3.s	31.7	1190	1190	1190	0.00	30.14
p5.3.t	33.3	1260	1255	1259.5	1.58	69.66
p5.3.u	35	1345	1345	1345	0.00	24.01
p5.3.v	36.7	1425	1415	1419.5	4.97	91.79
p5.3.w	38.3	1485	1465	1477.5	7.91	73.96
p5.3.x	40	1555	1530	1547.5	8.25	77.36
p5.3.y	41.7	1590	1565	1582	9.78	68.75
p5.3.z	43.3	1635	1615	1627.5	8.25	47.82

Çizelge C.8 PK 4 için 5 araçlı problemlerde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p5.4.c	3.8	20	20	20	0.00	0.00
p5.4.d	5	20	20	20	0.00	0.00
p5.4.e	6.2	20	20	20	0.00	0.00
p5.4.f	7.5	80	80	80	0.00	0.00
p5.4.g	8.8	140	140	140	0.00	1.52
p5.4.h	10	140	140	140	0.00	1.40
p5.4.i	11.2	240	240	240	0.00	1.77
p5.4.j	12.5	340	340	340	0.00	2.25
p5.4.k	13.8	340	340	340	0.00	2.19
p5.4.l	15	430	430	430	0.00	5.39
p5.4.m	16.2	555	555	555	0.00	30.22
p5.4.n	17.5	620	620	620	0.00	6.20
p5.4.o	18.8	690	690	690	0.00	44.94
p5.4.p	20	765	760	763.5	2.42	30.71
p5.4.q	21.2	860	840	856	8.43	42.13
p5.4.r	22.5	960	910	950	21.08	14.91
p5.4.s	23.8	1030	1030	1030	0.00	46.49
p5.4.t	25	1160	1160	1160	0.00	7.86
p5.4.u	26.2	1300	1300	1300	0.00	11.48
p5.4.v	27.5	1320	1320	1320	0.00	7.70
p5.4.w	28.8	1390	1380	1385.5	2.84	32.84
p5.4.x	30	1450	1440	1447	4.83	62.20
p5.4.y	31.2	1520	1520	1520	0.00	26.33
p5.4.z	32.5	1620	1570	1615	15.81	53.36

Çizelge C.9 PK 6 için İİ-DKA modeli ile elde edilen TOP sonuçları

Tmax	Amaç Fonksiyon Değeri				CPU
	En İyi	En Kötü	Ort	Std. Sap.	Ort
15	192	186	191.4	1.90	13.20
17.5	360	354	358.8	2.53	3.81
20	588	534	577.2	22.77	5.01
22.5	660	600	648	25.30	11.65
25	780	780	780	0.00	15.74
27.5	888	888	888	0.00	23.86
30	942	936	938.4	3.10	38.37
32.5	1032	1008	1029.6	7.59	22.04
35	1116	1092	1101	11.75	40.86
37.5	1188	1170	1171.8	5.69	62.60
40	1248	1230	1241.4	4.43	80.63
15	282	276	281.4	1.90	25.66
16.7	444	438	443.4	1.90	24.34
18.3	642	624	636.6	7.72	39.40
20	828	828	828	0.00	4.77
21.7	894	894	894	0.00	19.02
23.3	1002	1002	1002	0.00	22.21
25	1080	1080	1080	0.00	35.41
26.7	1170	1170	1170	0.00	62.42
15	366	366	366	0.00	2.94
16.2	528	528	528	0.00	4.11
17.5	696	696	696	0.00	22.53
18.8	912	912	912	0.00	21.90
20	1068	1068	1068	0.00	5.12

Çizelge C.10 PK 7 için 2 ve 3 araçlı problemlerde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p7.2.a	10	30	30	30	0.00	0.00
p7.2.b	20	64	64	64	0.00	0.00
p7.2.c	30	101	101	101	0.00	0.73
p7.2.d	40	190	190	190	0.00	6.98
p7.2.e	50	290	279	287.8	4.64	7.66
p7.2.f	60	387	343	378.2	18.55	5.78
p7.2.g	70	459	459	459	0.00	17.13
p7.2.h	80	521	451	500.6	27.33	35.09
p7.2.i	90	579	543	572.6	12.38	24.63
p7.2.j	100	646	584	624.5	21.55	16.99
p7.2.k	110	704	631	684.5	29.49	25.06
p7.2.l	120	767	683	746	27.28	33.11
p7.2.m	130	827	766	802.1	23.17	43.50
p7.2.n	140	888	795	857.3	29.51	61.18
p7.2.o	150	945	859	908.2	32.21	47.76
p7.2.p	160	1000	912	975.4	29.17	80.61
p7.2.q	170	1044	964	1023.6	29.25	54.57
p7.2.r	180	1094	1012	1061.1	30.25	85.58
p7.2.s	190	1136	1046	1109.8	25.93	58.73
p7.2.t	200	1179	1118	1145.2	21.30	84.70
p7.3.b	13.3	46	46	46	0.00	0.00
p7.3.c	20	79	79	79	0.00	0.00
p7.3.d	26.7	117	117	117	0.00	0.57
p7.3.e	33.3	175	175	175	0.00	1.61
p7.3.f	40	247	247	247	0.00	1.18
p7.3.g	46.7	344	344	344	0.00	1.65
p7.3.h	53.3	425	425	425	0.00	4.34
p7.3.i	60	487	480	485.4	2.91	19.52
p7.3.j	66.7	564	541	548.6	9.06	33.49
p7.3.k	73.3	633	609	625.9	10.00	31.94
p7.3.l	80	683	653	670.7	12.17	26.57
p7.3.m	86.7	762	695	732.8	20.18	37.13
p7.3.n	93.3	804	782	797.6	6.42	48.63
p7.3.o	100	874	831	854.9	14.02	61.54
p7.3.p	106.7	927	850	907.2	24.58	52.87
p7.3.q	113.3	976	895	951.2	26.02	68.02
p7.3.r	120	1021	982	1008.5	16.89	55.72
p7.3.s	126.7	1081	1031	1059.8	16.56	67.08
p7.3.t	133.3	1116	1052	1084.5	23.17	68.69

Çizelge C.11 PK 7 için 4 araçlı problemlerde İİ-DKA modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p7.4.b	10	30	30	30	0.00	0.00
p7.4.c	15	46	46	46	0.00	0.00
p7.4.d	20	79	79	79	0.00	0.00
p7.4.e	25	123	123	123	0.00	0.24
p7.4.f	30	164	164	164	0.00	2.47
p7.4.g	35	217	217	217	0.00	1.62
p7.4.h	40	285	285	285	0.00	3.69
p7.4.i	45	366	366	366	0.00	5.06
p7.4.j	50	462	462	462	0.00	5.29
p7.4.k	55	520	516	518.2	1.40	33.52
p7.4.l	60	590	577	586.5	5.66	27.59
p7.4.m	65	646	620	637.9	7.96	40.17
p7.4.n	70	726	680	701.3	14.89	20.99
p7.4.o	75	780	736	760.3	16.69	37.27
p7.4.p	80	833	762	814.3	19.13	29.26
p7.4.q	85	903	840	872	24.13	50.84
p7.4.r	90	970	898	946	26.90	33.71
p7.4.s	95	1021	933	975.3	33.53	49.52
p7.4.t	100	1077	986	1047.8	33.25	46.59

**EK D: OP İÇİN AYRIK G-PSO MODELİ İLE ELDE EDİLEN
SONUÇLAR**

Çizelge C.1 PK 1 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmaks	Optimum	Amaç Fonksiyon Değeri				CPU
		En iyi	En kötü	Ort	StDev	Ort
5	10	10	10	10	0.00	0.0
10	15	15	15	15	0.00	0.1
15	45	45	45	45	0.00	0.2
20	65	65	65	65	0.00	1.1
25	90	90	90	90	0.00	0.5
30	110	110	110	110	0.00	0.5
35	135	135	135	135	0.00	1.8
40	155	155	155	155	0.00	4.4
46	175	175	175	175	0.00	1.5
50	190	190	190	190	0.00	1.7
55	205	205	205	205	0.00	4.0
60	225	225	225	225	0.00	3.0
65	240	240	240	240	0.00	4.1
70	260	260	260	260	0.00	8.3
73	265	265	265	265	0.00	6.1
75	270	270	270	270	0.00	12.2
80	280	280	280	280	0.00	13.7
85	285	285	285	285	0.00	8.8
		167.78	167.78	167.78	0.00	4.00

Çizelge C.2 PK 2 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmax	Optimum	Amaç Fonksiyon Değeri				CPU
		En iyi	En kötü	Ort	StDev	Ort
15	120	120	120	120	0.00	0.202
20	200	200	200	200	0.00	0.184
23	210	210	210	210	0.00	0.498
25	230	230	230	230	0.00	0.208
27	230	230	230	230	0.00	0.238
30	265	265	265	265	0.00	0.869
32	300	300	300	300	0.00	0.306
35	320	320	320	320	0.00	1.745
38	360	360	355	359	2.11	2.004
40	395	395	390	394.5	1.58	1.74
45	450	450	450	450	0.00	2.086
		280.00	280.00	279.09	0.34	0.92

Çizelge C.3 PK 3 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmax	Optimum	Amaç Fonksiyon Değeri				CPU
		En iyi	En kötü	Ort	StDev	Ort
15	170	170	170	170	0.00	0.44
20	200	200	200	200	0.00	0.88
25	260	260	260	260	0.00	1.39
30	320	320	320	320	0.00	0.83
35	390	390	390	390	0.00	0.38
40	430	430	430	430	0.00	1.58
45	470	470	470	470	0.00	2.76
50	520	520	520	520	0.00	3.10
55	550	550	550	550	0.00	4.53
60	580	580	580	580	0.00	5.28
65	610	610	610	610	0.00	12.77
70	640	640	640	640	0.00	9.51
75	670	670	670	670	0.00	8.08
80	710	710	710	710	0.00	23.28
85	740	740	740	740	0.00	28.84
90	770	770	770	770	0.00	11.96
95	790	790	790	790	0.00	6.50
100	800	800	800	800	0.00	7.37
105	800	800	800	800	0.00	3.53
110	800	800	800	800	0.00	0.73
561.00		561.00	561.00	561.00	0.00	6.69

Çizelge C.4 PK 1_1 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmaks	Optimum	Amaç Fonksiyon Değeri				CPU
		En iyi	En kötü	Ort	StDev	Ort
5	10	10	10	10	0.00	0.0
10	15	15	15	15	0.00	0.1
15	45	45	45	45	0.00	0.2
20	65	65	65	65	0.00	1.0
25	90	90	90	90	0.00	0.5
30	110	110	110	110	0.00	0.6
35	135	135	135	135	0.00	1.7
40	155	155	155	155	0.00	3.9
46	175	175	175	175	0.00	1.6
50	190	190	190	190	0.00	2.3
55	205	205	205	205	0.00	7.2
60	225	225	225	225	0.00	2.4
65	240	240	240	240	0.00	4.8
70	260	260	260	260	0.00	6.3
73	265	265	265	265	0.00	7.3
75	270	275	270	271.5	0.00	9.4
80	280	280	280	280	0.00	12.3
85	285	285	285	285	0.00	12.9
168.06		167.78	167.86	167.86	0.00	4.13

Çizelge C.5 PK 5 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmax	Amaç Fonksiyon Değeri				CPU
	En iyi	En kötü	Ort	StDev	Ort
20	205	195	197.0	4.22	0.45
25	290	290	290.0	0.00	0.62
30	400	400	400.0	0.00	0.56
35	465	465	465.0	0.00	0.59
40	575	575	575.0	0.00	14.61
45	650	650	650.0	0.00	1.10
50	730	730	730.0	0.00	2.10
55	825	825	825.0	0.00	3.00
60	915	915	915.0	0.00	1.83
65	980	980	980.0	0.00	4.62
70	1070	1070	1070.0	0.00	6.06
75	1140	1140	1140.0	0.00	8.05
80	1215	1215	1215.0	0.00	24.72
85	1270	1270	1270.0	0.00	7.15
90	1340	1330	1339.0	3.16	16.63
95	1395	1390	1391.0	2.11	22.05
100	1465	1455	1457.0	4.22	37.57
105	1520	1515	1516.5	2.42	89.11
110	1550	1550	1550.0	0.00	45.13
115	1595	1585	1591.5	3.37	122.99
120	1635	1625	1631.0	5.16	145.75
125	1655	1655	1655.0	0.00	153.03
130	1680	1665	1677.0	4.83	186.16
	1068.04	1064.78	1066.52	1.28	38.86

Çizelge C.6 PK 6 için Ayrık G-PSO modeli ile elde edilen OP sonuçları

Tmax	Amaç Fonksiyon Değeri				CPU
	En iyi	En kötü	Ort	StDev	Ort
15	96	96	96.0	0.00	0.84
20	294	294	294.0	0.00	0.51
25	390	390	390.0	0.00	0.74
30	474	468	470.4	3.10	16.19
35	576	576	576.0	0.00	26.63
40	714	708	713.4	1.90	21.61
45	816	816	816.0	0.00	19.13
50	900	900	900.0	0.00	24.78
55	984	984	984.0	0.00	15.81
60	1062	1062	1062.0	0.00	92.43
65	1116	1116	1116.0	0.00	49.75
70	1188	1188	1188.0	0.00	53.09
75	1236	1236	1236.0	0.00	57.30
80	1284	1272	1274.4	5.06	81.83
	795.00	793.29	794.01	0.72	32.90

**EK E: TOP İÇİN AYRIK G-PSO MODELİ İLE ELDE EDİLEN
SONUÇLAR**

Çizelge E.1 PK 1 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p1.2.b	5	15	15	15	0.00	0.00
p1.2.c	7.5	20	20	20	0.00	0.02
p1.2.d	10	30	30	30	0.00	0.09
p1.2.e	12.5	45	45	45	0.00	0.20
p1.2.f	15	80	80	80	0.00	0.41
p1.2.g	17.5	90	90	90	0.00	0.56
p1.2.h	20	110	110	110	0.00	0.44
p1.2.i	23	135	135	135	0.00	2.31
p1.2.j	25	155	155	155	0.00	0.64
p1.2.k	27.5	175	175	175	0.00	1.70
p1.2.l	30	195	195	195	0.00	2.56
p1.2.m	32.5	215	215	215	0.00	1.95
p1.2.n	35	235	235	235	0.00	0.57
p1.2.o	36.5	240	240	240	0.00	1.58
p1.2.p	37.5	250	250	250	0.00	1.25
p1.2.q	40	265	265	265	0.00	1.51
p1.2.r	42.5	280	280	280	0.00	5.92
p1.3.c	5	15	15	15	0.00	0.00
p1.3.d	6.7	15	15	15	0.00	0.00
p1.3.e	8.3	30	30	30	0.00	0.01
p1.3.f	10	40	40	40	0.00	0.06
p1.3.g	11.7	50	50	50	0.00	0.09
p1.3.h	13.3	70	70	70	0.00	0.16
p1.3.i	15.3	105	100	104	2.11	1.31
p1.3.j	16.7	115	110	113.5	2.42	1.15
p1.3.k	18.3	135	130	133.5	2.42	1.16
p1.3.l	20	155	155	155	0.00	0.91
p1.3.m	21.7	175	175	175	0.00	2.30
p1.3.n	23.3	190	190	190	0.00	1.24
p1.3.o	24.3	205	205	205	0.00	0.58
p1.3.p	25	220	220	220	0.00	0.73
p1.3.q	26.7	230	230	230	0.00	1.38
p1.3.r	28.3	250	250	250	0.00	0.58
p1.4.d	5	15	15	15	0.00	0.00
p1.4.e	6.2	15	15	15	0.00	0.00
p1.4.f	7.5	25	25	25	0.00	0.00
p1.4.g	8.8	35	35	35	0.00	0.01
p1.4.h	10	45	45	45	0.00	0.04
p1.4.i	11.5	60	60	60	0.00	0.10
p1.4.j	12.5	75	75	75	0.00	0.11
p1.4.k	13.8	100	100	100	0.00	0.28
p1.4.l	15	120	120	120	0.00	0.23
p1.4.m	16.2	130	130	130	0.00	1.28
p1.4.n	17.5	155	150	154.5	1.58	1.68
p1.4.o	18.2	165	165	165	0.00	0.48
p1.4.p	18.8	175	175	175	0.00	0.59
p1.4.q	20	190	185	189.5	1.58	1.99
p1.4.r	21.2	210	205	209	2.11	3.25

Çizelge E.2 PK 2 için Ayırık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p2.2.a	7.5	90	90	90	0.00	0.08
p2.2.b	10	120	120	120	0.00	0.05
p2.2.c	11.5	140	140	140	0.00	0.20
p2.2.d	12.5	160	160	160	0.00	0.10
p2.2.e	13.5	190	190	190	0.00	0.13
p2.2.f	15	200	200	200	0.00	0.07
p2.2.g	16	200	200	200	0.00	0.06
p2.2.h	17.5	230	230	230	0.00	0.08
p2.2.i	19	230	230	230	0.00	0.06
p2.2.j	20	260	260	260	0.00	0.25
p2.2.k	22.5	275	275	275	0.00	0.54
p2.3.a	5	70	70	70	0.00	0.00
p2.3.b	6.7	70	70	70	0.00	0.00
p2.3.c	7.7	105	105	105	0.00	0.01
p2.3.d	8.3	105	105	105	0.00	0.01
p2.3.e	9	120	120	120	0.00	0.02
p2.3.f	10	120	120	120	0.00	0.00
p2.3.g	10.7	145	145	145	0.00	0.17
p2.3.h	11.7	165	165	165	0.00	0.20
p2.3.i	12.7	200	200	200	0.00	0.07
p2.3.j	13.3	200	200	200	0.00	0.06
p2.3.k	15	200	200	200	0.00	0.05
p2.4.a	3.8	10	10	10	0.00	0.00
p2.4.b	5	70	70	70	0.00	0.00
p2.4.c	5.8	70	70	70	0.00	0.00
p2.4.d	6.2	70	70	70	0.00	0.00
p2.4.e	6.8	70	70	70	0.00	0.00
p2.4.f	7.5	105	105	105	0.00	0.00
p2.4.g	8	105	105	105	0.00	0.00
p2.4.h	8.8	120	120	120	0.00	0.00
p2.4.i	9.5	120	120	120	0.00	0.00
p2.4.j	10	120	120	120	0.00	0.00
p2.4.k	11.2	180	180	180	0.00	0.14

Çizelge E.3 2 ve 3 araçlı PK 3 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p3.2.a	7.5	90	90	90	0.00	0.05
p3.2.b	10	150	150	150	0.00	0.09
p3.2.c	12.5	180	180	180	0.00	0.13
p3.2.d	15	220	220	220	0.00	0.20
p3.2.e	17.5	260	260	260	0.00	0.34
p3.2.f	20	300	300	300	0.00	0.30
p3.2.g	22.5	360	360	360	0.00	1.03
p3.2.h	25	410	400	404	5.16	2.61
p3.2.i	27.5	460	450	459	3.16	1.70
p3.2.j	30	510	500	509	3.16	1.72
p3.2.k	32.5	550	550	550	0.00	0.55
p3.2.l	35	590	590	590	0.00	1.97
p3.2.m	37.5	620	610	619	3.16	2.54
p3.2.n	40	660	660	660	0.00	11.68
p3.2.o	42.5	690	690	690	0.00	4.39
p3.2.p	45	720	720	720	0.00	3.44
p3.2.q	47.5	760	750	759	3.16	9.70
p3.2.r	50	790	780	789	3.16	19.28
p3.2.s	52.5	800	800	800	0.00	0.70
p3.2.t	55	800	800	800	0.00	0.21
p3.3.b	6.7	90	90	90	0.00	0.00
p3.3.c	8.3	120	120	120	0.00	0.06
p3.3.d	10	170	170	170	0.00	0.07
p3.3.e	11.7	200	200	200	0.00	0.12
p3.3.f	13.3	230	230	230	0.00	0.12
p3.3.g	15	270	270	270	0.00	0.29
p3.3.h	16.7	300	300	300	0.00	0.20
p3.3.i	18.3	330	330	330	0.00	0.40
p3.3.j	20	380	380	380	0.00	0.51
p3.3.k	21.7	440	440	440	0.00	1.20
p3.3.l	23.3	480	480	480	0.00	1.39
p3.3.m	25	520	520	520	0.00	3.52
p3.3.n	26.7	570	550	566	8.43	2.82
p3.3.o	28.3	590	580	589	3.16	2.48
p3.3.p	30	640	640	640	0.00	1.28
p3.3.q	31.7	680	680	680	0.00	1.85
p3.3.r	33.3	710	710	710	0.00	0.40
p3.3.s	35	720	710	718	4.22	2.73
p3.3.t	36.7	760	760	760	0.00	2.08

Çizelge E 4 4 araçlı PK 3 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p3.4.c	6.2	90	90	90.00	0.00	0.00
p3.4.d	7.5	100	100	100.00	0.00	0.00
p3.4.e	8.8	140	140	140.00	0.00	0.04
p3.4.f	10	190	190	190.00	0.00	0.09
p3.4.g	11.2	220	220	220.00	0.00	0.09
p3.4.h	12.5	240	240	240.00	0.00	0.22
p3.4.i	13.8	270	270	270.00	0.00	0.19
p3.4.j	15	310	310	310.00	0.00	0.56
p3.4.k	16.2	350	350	350.00	0.00	0.18
p3.4.l	17.5	380	380	380.00	0.00	0.87
p3.4.m	18.8	390	390	390.00	0.00	0.55
p3.4.n	20	440	440	440.00	0.00	0.58
p3.4.o	21.2	500	490	497.00	4.83	1.61
p3.4.p	22.5	560	560	560.00	0.00	0.73
p3.4.q	23.8	560	560	560.00	0.00	0.79
p3.4.r	25	600	600	600.00	0.00	1.27
p3.4.s	26.2	670	670	670.00	0.00	3.02
p3.4.t	27.5	670	670	670.00	0.00	0.36

Çizelge E.5 2 araçlı PK 4 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p4.2.a	25	206	190	200.70	5.48	0.33
p4.2.b	30	341	307	322.10	11.71	0.70
p4.2.c	35	452	443	448.50	3.27	15.76
p4.2.d	40	531	517	524.80	4.69	35.20
p4.2.e	45	601	588	593.10	5.04	33.39
p4.2.f	50	680	661	671.20	6.25	22.17
p4.2.g	55	748	727	740.40	6.11	24.94
p4.2.h	60	817	799	809.90	6.49	31.85
p4.2.i	65	909	853	884.10	17.08	55.37
p4.2.j	70	962	934	949.29	10.09	71.37
p4.2.r	110	1269	1255	1261.80	5.94	42.27
p4.2.s	115	1291	1272	1281.70	6.09	49.29
p4.2.t	120	1302	1294	1298.30	2.63	67.33

Çizelge E.6 3 ve 4 araçlı PK 4 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p4.3.b	20	38	38	38.00	0.00	0.00
p4.3.c	23.3	193	187	192.20	1.93	0.46
p4.3.d	26.7	332	310	323.80	7.87	4.27
p4.3.e	30	464	445	453.70	5.68	16.54
p4.3.f	33.3	563	529	545.20	11.72	36.61
p4.3.g	36.7	641	617	626.60	8.50	45.62
p4.3.h	40	720	683	701.50	12.36	53.71
p4.3.i	43.3	790	722	767.10	20.16	27.71
p4.3.j	46.7	841	809	823.00	11.04	33.06
p4.3.k	50	902	856	879.20	14.40	24.46
p4.3.l	53.3	972	907	934.90	17.08	40.86
p4.3.m	56.7	1032	979	999.60	15.51	41.80
p4.3.n	60	1085	1052	1068.00	11.12	69.44
p4.3.o	63.3	1146	1096	1125.40	15.39	65.13
p4.3.p	66.7	1182	1126	1158.90	20.78	47.72
p4.3.q	70	1227	1192	1206.40	9.29	43.83
p4.3.r	73.3	1254	1222	1236.90	10.59	46.88
p4.3.s	76.7	1275	1242	1257.90	11.09	42.31
p4.3.t	80	1276	1259	1267.80	4.76	31.69
p4.4.d	20	38	38	38.00	0.00	0.00
p4.4.e	22.5	183	183	183.00	0.00	0.29
p4.4.f	25	324	309	317.50	5.08	1.25
p4.4.g	27.5	461	441	450.00	6.60	12.80
p4.4.h	30	556	504	528.00	14.13	7.72
p4.4.i	32.5	633	604	616.90	9.71	14.37
p4.4.j	35	705	672	688.60	11.83	23.01
p4.4.k	37.5	786	742	760.20	12.87	19.71
p4.4.l	40	829	791	808.60	13.11	13.94
p4.4.m	42.5	888	856	875.70	11.12	19.14
p4.4.n	45	926	896	910.40	9.40	12.63
p4.4.o	47.5	1000	958	976.50	14.94	41.31
p4.4.p	50	1079	1024	1050.20	18.98	40.85
p4.4.q	52.5	1137	1025	1092.40	29.39	35.88
p4.4.r	55	1188	1124	1147.60	16.98	22.70
p4.4.s	57.5	1218	1162	1196.10	18.88	30.26
p4.4.t	60	1245	1197	1222.40	15.59	21.88

Çizelge E.7 2 ve 3 araçlı PK 5 için Ayırık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p5.2.b	5	20	20	20	0.00	0.00
p5.2.c	7.5	50	50	50	0.00	0.12
p5.2.d	10	80	80	80	0.00	0.17
p5.2.e	12.5	180	170	175	3.33	0.61
p5.2.f	15	240	215	233.5	10.55	0.93
p5.2.g	17.5	320	310	315.5	2.84	2.35
p5.2.h	20	395	385	392.5	4.25	2.99
p5.2.i	22.5	480	455	468.5	7.47	4.22
p5.2.j	25	580	580	580	0.00	4.30
p5.2.k	27.5	670	665	669.5	1.58	1.72
p5.2.l	30	800	755	779	14.10	14.55
p5.2.m	32.5	860	845	851.5	4.12	11.27
p5.2.n	35	925	920	923	2.58	6.54
p5.2.o	37.5	1020	1010	1012	4.22	9.69
p5.2.p	40	1150	1095	1143	17.51	18.89
p5.2.q	42.5	1195	1185	1190	2.36	25.27
p5.2.r	45	1260	1245	1255	5.27	13.44
p5.2.s	47.5	1330	1310	1321.5	8.18	12.67
p5.2.t	50	1400	1370	1387.5	9.20	15.36
p5.2.u	52.5	1450	1435	1445.5	5.50	24.69
p5.2.v	55	1505	1485	1496.5	7.84	29.00
p5.2.w	57.5	1560	1545	1551	5.16	11.59
p5.2.x	60	1600	1580	1591.5	5.30	15.67
p5.2.y	62.5	1635	1620	1630.5	5.99	22.96
p5.2.z	65	1670	1660	1664	5.16	47.98
p5.3.b	3.3	15	15	15	0.00	0.00
p5.3.c	5	20	20	20	0.00	0.00
p5.3.d	6.7	60	60	60	0.00	0.07
p5.3.e	8.3	95	95	95	0.00	0.12
p5.3.f	10	110	110	110	0.00	0.16
p5.3.g	11.7	185	185	185	0.00	0.29
p5.3.h	13.3	260	250	257.5	4.25	1.83
p5.3.i	15	335	305	326.5	10.81	2.63
p5.3.j	16.7	470	395	442	24.29	3.53
p5.3.k	18.3	495	475	485	8.16	16.74
p5.3.l	20	585	575	580	4.71	10.00
p5.3.m	21.7	645	640	644	2.11	5.50
p5.3.n	23.3	755	730	746	10.49	14.76
p5.3.o	25	870	850	864	8.43	13.41
p5.3.p	26.7	990	970	986.5	6.26	12.35
p5.3.q	28.3	1070	1045	1060	7.45	10.96
p5.3.r	30	1125	1100	1114.5	8.32	6.88
p5.3.s	31.7	1185	1180	1184	2.11	8.22
p5.3.t	33.3	1255	1240	1248.5	5.80	20.96
p5.3.u	35	1340	1315	1330.5	7.25	12.43
p5.3.v	36.7	1415	1395	1404.5	6.43	12.52
p5.3.w	38.3	1470	1450	1461.5	6.26	9.13
p5.3.x	40	1535	1510	1522.5	9.20	15.41
p5.3.y	41.7	1590	1565	1570	8.50	8.47
p5.3.z	43.3	1635	1605	1618	8.56	20.25

Çizelge E. 8 4 araçlı PK 5 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p5.4.c	3.8	20	20	20	0.00	0.00
p5.4.d	5	20	20	20	0.00	0.00
p5.4.e	6.2	20	20	20	0.00	0.00
p5.4.f	7.5	80	80	80	0.00	0.13
p5.4.g	8.8	140	140	140	0.00	0.09
p5.4.h	10	140	140	140	0.00	0.18
p5.4.i	11.2	240	240	240	0.00	0.38
p5.4.j	12.5	340	315	337.5	7.91	1.11
p5.4.k	13.8	340	340	340	0.00	0.71
p5.4.l	15	430	415	424.5	4.97	11.82
p5.4.m	16.2	550	515	533	15.31	18.56
p5.4.n	17.5	620	575	599.5	15.71	7.27
p5.4.o	18.8	675	640	655.5	9.56	10.44
p5.4.p	20	760	735	756.5	8.18	13.84
p5.4.q	21.2	860	810	831.5	15.10	16.17
p5.4.r	22.5	935	895	905	11.55	12.41
p5.4.s	23.8	1025	1000	1018	8.56	20.65
p5.4.t	25	1160	1110	1151	15.95	22.87
p5.4.u	26.2	1300	1220	1264.5	25.11	28.17
p5.4.v	27.5	1320	1290	1315	10.80	12.03
p5.4.w	28.8	1375	1365	1369	3.16	9.23
p5.4.x	30	1440	1420	1431.5	7.47	14.18
p5.4.y	31.2	1520	1485	1502	12.74	15.00
p5.4.z	32.5	1570	1525	1543	14.18	8.95

Çizelge E.9 PK 6 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p6.2.d	15	192	186	189.6	3.10	0.66
p6.2.e	17.5	360	360	360	0.00	1.21
p6.2.f	20	588	564	578.4	8.58	11.01
p6.2.g	22.5	660	660	660	0.00	1.62
p6.2.h	25	780	780	780	0.00	2.61
p6.2.i	27.5	888	888	888	0.00	3.22
p6.2.j	30	942	936	939.6	3.10	17.56
p6.2.k	32.5	1032	1032	1032	0.00	5.48
p6.2.l	35	1110	1098	1103.4	4.43	9.44
p6.2.m	37.5	1188	1152	1161	12.08	9.77
p6.2.n	40	1236	1224	1228.2	4.94	1.85
p6.3.g	15	282	276	280.2	2.90	0.99
p6.3.h	16.7	444	444	444	0.00	1.63
p6.3.i	18.3	642	642	642	0.00	1.39
p6.3.j	20	828	828	828	0.00	8.44
p6.3.k	21.7	888	870	883.2	6.20	4.54
p6.3.l	23.3	996	972	985.8	8.51	8.18
p6.3.m	25	1080	1080	1080	0.00	7.37
p6.3.n	26.7	1170	1158	1163.4	5.25	7.48
p6.4.j	15	366	366	366	0.00	0.60
p6.4.k	16.2	528	522	526.2	2.90	1.51
p6.4.l	17.5	696	696	696	0.00	5.15
p6.4.m	18.8	912	870	888.6	15.61	10.16
p6.4.n	20	1068	1068	1068	0.00	3.25

Çizelge E.10 2 ve 3 araçlı PK 7 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p7.2.a	10	30	30	30	0.00	0.00
p7.2.b	20	64	64	64	0.00	0.02
p7.2.c	30	101	101	101	0.00	0.07
p7.2.d	40	190	179	181.6	4.60	0.24
p7.2.e	50	289	268	283.2	6.94	0.78
p7.2.f	60	387	369	378.3	5.03	1.12
p7.2.g	70	454	440	447.9	4.98	5.43
p7.2.h	80	521	499	514.7	7.09	24.20
p7.2.i	90	573	561	568.5	3.72	15.54
p7.2.j	100	639	613	627.1	8.21	22.92
p7.2.k	110	698	681	690.6	4.45	26.07
p7.2.l	120	764	730	748.3	12.13	34.95
p7.2.m	130	807	777	795.2	9.65	26.34
p7.2.n	140	870	837	853.9	9.57	25.91
p7.2.o	150	945	883	910.3	16.71	18.07
p7.2.p	160	991	952	974.3	15.57	26.04
p7.2.q	170	1033	1001	1016.5	9.25	19.37
p7.2.r	180	1075	1049	1060.7	7.92	17.55
p7.2.s	190	1127	1094	1107.7	9.39	25.70
p7.2.t	200	1162	1134	1146.7	9.56	24.36
p7.3.b	13.3	46	46	46	0.00	0.00
p7.3.c	20	79	79	79	0.00	0.01
p7.3.d	26.7	117	117	117	0.00	0.09
p7.3.e	33.3	175	170	173.5	2.42	0.39
p7.3.f	40	247	236	242.4	5.38	0.53
p7.3.g	46.7	344	317	337.5	10.29	1.50
p7.3.h	53.3	425	409	414.6	4.81	3.97
p7.3.i	60	482	461	472.2	6.41	5.34
p7.3.j	66.7	562	529	541	10.10	12.78
p7.3.k	73.3	626	595	604.1	9.69	16.50
p7.3.l	80	681	623	653.3	16.53	15.57
p7.3.m	86.7	732	691	716.1	12.83	21.60
p7.3.n	93.3	792	762	773.6	8.46	25.46
p7.3.o	100	840	820	830.4	7.01	19.18
p7.3.p	106.7	905	858	888.2	15.82	16.65
p7.3.q	113.3	937	897	916.8	12.52	15.03
p7.3.r	120	1010	959	981.1	13.36	19.94
p7.3.s	126.7	1040	1008	1025.4	9.55	19.18
p7.3.t	133.3	1082	1046	1064.4	13.06	14.74

Çizelge E.11 4 araçlı PK 7 için Ayrık G-PSO modeli ile elde edilen TOP sonuçları

Prb Adı	Tmax	Amaç Fonksiyon Değeri				CPU
		En İyi	En Kötü	Ort	Std. Sap.	Ort
p7.4.b	10	30	30	30	0.00	0.00
p7.4.c	15	46	46	46	0.00	0.00
p7.4.d	20	79	79	79	0.00	0.00
p7.4.e	25	123	123	123	0.00	0.07
p7.4.f	30	164	140	157	6.68	0.44
p7.4.g	35	217	201	210.8	5.09	0.28
p7.4.h	40	285	280	282.7	1.34	0.47
p7.4.i	45	354	338	347.1	5.11	1.50
p7.4.j	50	454	434	441.6	5.93	3.44
p7.4.k	55	514	489	501.4	7.86	2.59
p7.4.l	60	571	548	558.3	8.04	9.61
p7.4.m	65	633	588	611.8	14.25	8.94
p7.4.n	70	690	652	674.7	12.26	23.59
p7.4.o	75	748	710	727.8	13.49	10.58
p7.4.p	80	808	761	788.1	15.81	11.30
p7.4.q	85	879	812	836.2	22.84	13.52
p7.4.r	90	909	870	890.1	13.71	14.17
p7.4.s	95	971	912	948.5	17.53	9.08
p7.4.t	100	1023	975	996.5	14.33	13.04