

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**İKİ KRİTERLİ KARESEL ATAMA PROBLEMLERİNİN ÇOK-
AMAÇLI KARINCA KOLONİLERİ ALGORİTMALARI İLE
OPTİMİZASYONU**

DOKTORA TEZİ

Y. Müh. Celal ÖZKALE

**Anabilim Dalı: Endüstri Mühendisliği
Danışman: Prof. Dr. Alpaslan FIĞLALI**

KOCAELİ, 2010

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

İKİ KRİTERLİ KARESEL ATAMA PROBLEMLERİNİN ÇOK-
AMAÇLI KARINCA KOLONİLERİ ALGORİTMALARI İLE
OPTİMİZASYONU

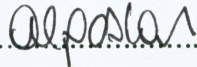
DOKTORA TEZİ

Y.Müh. Celal ÖZKALE

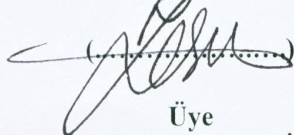
Tezin Enstitüye Verildiği Tarih: 17 ARALIK 2009

Tezin Savunulduğu Tarih: 02 ŞUBAT 2010

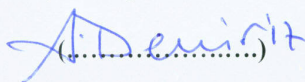
Tez Danışmanı
Prof.Dr. Alpaslan FİĞLALI

()

Üye
Prof.Dr. Zerrin ALADAĞ

()

Üye
Doç.Dr. Ayhan DEMİRİZ

()

Üye
Prof.Dr. Coşkun ÖZKAN

()

Üye
Yrd.Doç.Dr. Aybek KORUGAN

()

KOCAELİ, 2010

ÖNSÖZ ve TEŞEKKÜR

Birleşimsel optimizasyon problemlerinden çoğunun tamamen çözülebilmesi makul sürelerde gerçekleştirilememektedir. Bundan dolayı optimal ya da optimale yakın çözümleri, makul zamanlarda üretebilmek için ilk dönemlerde sezgisel, son dönemlerde metasezgisel optimizasyon yöntemleri geliştirilmiştir.

Çok-amaçlı birleşimsel optimizasyon problemlerini de geleneksel yöntemlerle çözme zorluğu, araştırmacıları daha iyi performans veren yaklaşımları araştırmaya sevk etmiştir. Son zamanlarda Karınca Kolonileri metasezgiselini esas alan, çok-amaçlı problemleri çözmeye yönelik bazı algoritmalar önerilmiştir. Bu tür algoritmalar özellikle çok-amaçlı üretim çizelgeleme, lojistik, portföy seçimi, su dağıtımı vb. problemleri çözmekte faydalanılmıştır.

Bu çalışmada Çok-Amaçlı Karınca Kolonileri metasezgiselleri ile İki Kriterli Karesel Atama Problemlerinin ilk kez çözümlenerek karşılaştırılmasında ve yeni bir değerlendirme yöntemi ile algoritma sunmamda bana fırsat tanıyan ve çalışmamda desteğini esirgemeyen hocam Prof. Dr. Alpaslan FİĞLALI'ya teşekkür ederim. Ayrıca, göstermiş olduğu sabır ve sonsuz desteği için sevgili eşim Neslihan ÖZKALE'ye, yaşamım boyunca ilgi ve emeklerini esirgemeyen annem Nermin ÖZKALE ve babam Ekrem ÖZKALE'ye de sonsuz şükran duygularımı sunarım. Çalışmamın son evrelerinde dünyaya gelen ve yoğun çalışma tempomdan dolayı yeteri kadar vakit ayıramadığım, buna rağmen görüştüğümüz kısa zamanlarda bana farkında olmadan yaşattığı mutluluktan dolayı oğlum Poyraz ÖZKALE'ye en kocaman sevgi ve şükranlarımı sunarım.

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	iv
TABLolar DİZİNİ	v
SİMGELER	vii
ÖZET	x
İNGİLİZCE ÖZET	xi
1. GİRİŞ	1
2. KARESEL ATAMA PROBLEMİ	4
2.1. Karesel Atama Problemi'ne Giriş	4
2.2. KAP ve KAP ile İlişkili Problemlerin Formülasyonları	8
2.2.1. Seçilmiş KAP formülasyonları	8
2.2.1.1. Tamsayılı doğrusal programlama formülasyonları	8
2.2.1.2. Karışık tamsayılı doğrusal programlama formülasyonları	9
2.2.1.3. Permütasyonlar yoluyla formülasyonlar	11
2.2.1.4. İz formülasyonu	12
2.2.1.5. Çizge formülasyonu	13
2.2.2. KAP ile ilişkili problemler	14
2.2.2.1. Karesel darboğaz atama problemi	15
2.2.2.2. Bikaresel atama problemi	16
2.2.2.3. Karesel 3-boyutlu atama problemi	16
2.2.2.4. Karesel yarı-atama problemi	16
2.2.2.5. Çok-amaçlı KAP	17
2.3. Alt Sınırlar	18
2.3.1. Karışık tamsayılı doğrusal programlama gevşetmelerini esas alan sınırlar.....	19
2.3.2. GLB yeniden-formülasyonları esas alan sınırlar	20
2.3.3. İç noktalar yöntemini esas alan sınırlar	20
2.3.4. Değişkenlik indirgeme sınırları	21
2.3.5. Çizge formülasyonunu esas alan sınırlar	21
2.3.6. Spektral sınırlar	21
2.3.7. Yarı-kesin programlama ve yeniden-formülasyon doğrusallaştırma sınırları.....	22
2.4. Çözüm Yöntemleri	22
2.4.1. Kesin algoritmalar	22
2.4.2. Sezgisel algoritmaları	24
2.4.2.1. Yapısal yöntemler	24
2.4.2.2. Sıralamalı yöntemler	25
2.4.2.3. Geliştirme yöntemleri	25
2.4.3. Metasezgiseller	25
2.4.3.1. Doğal süreç metaforlarını esas alan metasezgiseller	26
2.4.3.2. Doğrudan kuramsal ve deneysel düşünceleri esas alan metasezgiseller..	28
2.5. Ana Araştırma Eğilimleri ve Yösemeler	31

3. ÇOK-AMAÇLI KARINCA KOLONİLERİ OPTİMİZASYONU	32
3.1. Karınca Koloni Optimizasyonu	33
3.1.1. Karınca sistemi	34
3.1.2. Karınca koloni sistemi	36
3.2. Çok-Amaçlı Optimizasyon	38
3.3. Çok-Amaçlı Karınca Koloni Optimizasyonu Algoritmaları	40
3.3.1. Çok-amaçlı Ant-Q Algoritması (ÇAAQA)	40
3.3.2. İki-kriterli optimizasyon problemleri için karınca algoritması	42
3.3.3. İki-kriterli optimizasyon problemleri için çoklu koloni	43
3.3.4. Pareto karınca koloni optimizasyonu	44
3.3.5. Teslim süreleri ile araç rotalama problemi için çoklu karınca koloni sistemi	45
3.3.6. Çoklu karınca koloni sistemi	46
3.3.7. Karınca kolonileri optimizasyonu çok-amaçlı şebekesi	47
3.3.8. Ehil karıncalar çok-amaçlı sistemi	49
3.3.9. Çok-amaçlı karınca kolonileri optimizasyonu metasezgiseli	50
3.3.10. Çok-amaçlılara karınca kolonileri optimizasyonu yaklaşımı	51
3.3.11. SACO	52
4. İKİ KRİTERLİ KARESEL ATAMA PROBLEMLERİNİN ÇOK-AMAÇLI KARINCA KOLONİLERİ ALGORİTMALARI İLE OPTİMİZASYONU.....	54
5. SONUÇLAR VE ÖNERİLER	105
KAYNAKLAR	107
ÖZGEÇMİŞ	136

ŞEKİLLER DİZİNİ

Şekil 2.1:	K_D üzerine K_F kliklerinin $\pi = (4, 2, 1, 3)$ permütasyonunda paylaşılması.....	14
Şekil 4.1:	İKKA parametre eniyileme çalışması, D8 (solda D3, D4, D5 ve D7 ile aynı) ve D2 (sağda) pareto-önyüzleri.....	60
Şekil 4.2:	İKKA-M parametre eniyileme çalışması D1 pareto-önyüzü.....	62
Şekil 4.3:	BİKA parametre eniyileme çalışması D4(solda) ve D7 (sağda) pareto-önyüzü.....	64
Şekil 4.4:	BİKA-M parametre eniyileme çalışması D9 pareto-önyüzü.....	66
Şekil 4.5:	İKÇKA parametre eniyileme çalışması D1 pareto-önyüzü.....	68
Şekil 4.6:	İKÇKA-M parametre eniyileme çalışması D9 (solda) ve D1 (sağda) pareto-önyüzü.....	70
Şekil 4.7:	EKÇAS parametre eniyileme çalışması D3 pareto-önyüzü.....	72
Şekil 4.8:	KKOÇAŞ parametre eniyileme çalışması D3 pareto-önyüzü.....	74
Şekil 4.9:	ÇKKS parametre eniyileme çalışması D2 pareto-önyüzü.....	76
Şekil 4.10:	PKKOA parametre eniyileme çalışması D7 (solda) ve D5 (sağda) pareto-önyüzü.....	78
Şekil 4.11:	PKKOA-M parametre eniyileme çalışması D2 pareto-önyüzü.....	79
Şekil 4.12:	ÇAAQA parametre eniyileme çalışması D3 (solda) ve D6 (sağda) pareto-önyüzü.....	81
Şekil 4.13:	30 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü...	89
Şekil 4.14:	40 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü...	90
Şekil 4.15:	50 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü...	91
Şekil 4.16:	30 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü.....	101
Şekil 4.17:	40 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü.....	102
Şekil 4.18:	50 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü.....	103

TABLolar DİZİNİ

Tablo 4.1:	İKKA (BicriterionAnt) deney seviyeleri.....	56
Tablo 4.2:	İKKA (BicriterionAnt) $2^{4-1} + 1$ kesirli deney tasarımı.....	56
Tablo 4.3:	İKKA (BicriterionAnt) parametre düzen seçimi.....	58
Tablo 4.4:	İKKA-M (BicriterionAnt-C) deney seviyeleri.....	60
Tablo 4.5:	İKKA-M (BicriterionAnt-C) $2^{4-1} + 1$ kesirli deney tasarımı.....	61
Tablo 4.6:	İKKA-M (BicriterionAnt-C) parametre düzen seçimi.....	61
Tablo 4.7:	BİKA (UnsortBicriterion) deney seviyeleri.....	62
Tablo 4.8:	BİKA (UnsortBicriterion) $2^{4-1} + 1$ kesirli deney tasarımı.....	63
Tablo 4.9:	BİKA (UnsortBicriterion) parametre düzen seçimi.....	63
Tablo 4.10:	BİKA-M (UnsortBicriterion-C) deney seviyeleri.....	64
Tablo 4.11:	BİKA-M (UnsortBicriterion-C) $2^{4-1} + 1$ kesirli deney tasarımı...	65
Tablo 4.12:	BİKA-M (UnsortBicriterion-C) parametre düzen seçimi.....	65
Tablo 4.13:	İKÇKA (BicriterionMC) deney seviyeleri.....	67
Tablo 4.14:	İKÇKA (BicriterionMC) $2^{4-1} + 1$ kesirli deney tasarımı.....	67
Tablo 4.15:	İKÇKA (BicriterionMC) parametre düzen seçimi.....	68
Tablo 4.16:	İKÇKA-M (BicriterionMC-C) deney seviyeleri.....	69
Tablo 4.17:	İKÇKA-M (BicriterionMC-C) $2^{4-1} + 1$ kesirli deney tasarımı....	69
Tablo 4.18:	İKÇKA-M (BicriterionMC-M) parametre düzen seçimi.....	69
Tablo 4.19:	EKÇAS (CompetAnts) deney seviyeleri.....	71
Tablo 4.20:	EKÇAS (CompetAnts) $2^{4-1} + 1$ kesirli deney tasarımı.....	71
Tablo 4.21:	EKÇAS (CompetAnts) parametre düzen seçimi.....	71
Tablo 4.22:	KKOÇAŞ (MONACO) deney seviyeleri.....	72
Tablo 4.23:	KKOÇAŞ (MONACO) $2^{4-1} + 1$ kesirli deney tasarımı.....	73
Tablo 4.24:	KKOÇAŞ (MONACO) parametre düzen seçimi.....	73
Tablo 4.25:	ÇKKS (MOACS) deney seviyeleri.....	74
Tablo 4.26:	ÇKKS (MOACS) $2^{4-1} + 1$ kesirli deney tasarımı.....	75
Tablo 4.27:	ÇKKS (MOACS) parametre düzen seçimi.....	75
Tablo 4.28:	PKKOA (P-ACO) deney seviyeleri.....	77
Tablo 4.29:	PKKOA (P-ACO) 2^{5-2} kesirli deney tasarımı.....	77
Tablo 4.30:	PKKOA (P-ACO) parametre düzen seçimi.....	77
Tablo 4.31:	PKKOA-M (P-ACO-C) deney seviyeleri.....	78
Tablo 4.32:	PKKOA-M (P-ACO-C) 2^{5-2} kesirli deney tasarımı.....	78
Tablo 4.33:	PKKOA-M (P-ACO-C) parametre düzen seçimi.....	79
Tablo 4.34:	ÇAAQA (MOAQ) deney seviyeleri.....	80
Tablo 4.35:	ÇAAQA (MOAQ) $2^{6-3} + 1$ kesirli deney tasarımı.....	80
Tablo 4.36:	ÇAAQA (MOAQ) parametre düzen seçimi.....	81
Tablo 4.37:	ÇAAQA (MOAQ) parametre düzen seçimi (düzeltilmiş).....	82
Tablo 4.38:	ÇAKKO algoritmaları kod göstergesi.....	82
Tablo 4.39:	KC10-2fl-1uni literatür probleminde ÇAKKO algoritmaları performansları.....	83
Tablo 4.40:	KC10-2fl-3uni literatür probleminde ÇAKKO algoritmaları performansları.....	84

Tablo 4.41:	KC20-2fl-1uni literatür probleminde ÇAKKO algoritmaları performansları.....	86
Tablo 4.42:	KC20-2fl-3uni literatür probleminde ÇAKKO algoritmaları performansları.....	87
Tablo 4.43:	30 boyutlu rassal örnekte ÇAKKO algoritmaları performansları.	88
Tablo 4.44:	40 boyutlu rassal örnekte ÇAKKO algoritmaları performansları.	89
Tablo 4.45:	50 boyutlu rassal örnekte ÇAKKO algoritmaları performansları.	91
Tablo 4.46:	Global pareto-önyüzde buldukları karar seçeneği sayısına göre ÇAKKO algoritmaları performansları.....	92
Tablo 4.47:	ÇİKKA- <i>poyraz</i> ve TİKKA- <i>poyraz</i> (MBicriterionAnt- <i>boreas</i> ve SBicriterionAnt- <i>boreas</i>) deney seviyeleri.....	96
Tablo 4.48:	ÇİKKA- <i>poyraz</i> ve TİKKA- <i>poyraz</i> (MBicriterionAnt- <i>boreas</i> ve SBicriterionAnt- <i>boreas</i>) $2^{4-1} + 1$ kesirli deney tasarımı.....	96
Tablo 4.49:	ÇİKKA- <i>poyraz</i> (MBicriterionAnt- <i>boreas</i>) parametre düzen seçimi.....	96
Tablo 4.50:	TİKKA- <i>poyraz</i> (SBicriterionAnt- <i>boreas</i>) parametre düzen seçimi.....	97
Tablo 4.51:	Önerilen ÇAKKO algoritmalarının performanslarının KC10-2fl-1uni literatür probleminde karşılaştırılması.....	98
Tablo 4.52:	Önerilen ÇAKKO algoritmalarının performanslarının KC10-2fl-3uni literatür probleminde karşılaştırılması.....	99
Tablo 4.53:	Önerilen ÇAKKO algoritmalarının performanslarının KC20-2fl-1uni literatür probleminde karşılaştırılması.....	99
Tablo 4.54:	Önerilen ÇAKKO algoritmalarının performanslarının KC20-2fl-3uni literatür probleminde karşılaştırılması.....	100
Tablo 4.55:	Global pareto-önyüzde buldukları karar seçeneği sayısına göre ÇAKKO algoritmaları ve önerilen algoritmaların performansları (12 replikasyon sonucu).....	104

SİMGELER

- A : bir durumdaki tüm mümkün hareketlerin kümesi
 $\text{Aut}(L(G))$: G 'nin tüm otomorfizmalarının kümesi
 B : maliyet matrisi
 E : enerji
 G : Bir çizge (graph)
 $HE(s, a)$: sezgisel bilgi
 L : karıncalar tarafından inşa edilen çözüm (yerel hafıza)
 $L(G)$: G 'nin çizgi-grafiği
 N : komşu tesis numarası
 P : pareto-optimal küme
 P_G : birleşik pareto-önyüzdeki global pareto-önyüze ait karar seçeneği sayısı
 P_L : birleşik pareto-önyüzdeki yerel pareto-önyüze ait karar seçeneği sayısı
 Q : feromon bırakma sabiti
 $Q(s, a)$: feromon bilgisi
 S : tesislerin yerleşimlere tüm dizilimlerinin (permütasyonlarının) kümesi
 X : permütasyon matrisi
 a : uygun hareketler
 b : tesisleri yerleşimlere konumlandırma maliyeti
 c : maliyet
 d : yerleşimler arası mesafe
 e : her koordinatı 1'e eşit olan vektör
 f : tesisler arası akış
 h : karınca numarası
 k : tesisler arasında kaçınıcı akış (amaç sayısı ile ilişkili)
 l : pareto-önyüzde çözüm üreten karınca sayısı
 m : toplam karınca sayısı
 n : tesis sayısı, yerleşim sayısı
 p : koloni sayısı
 q : araştırma-devam katsayısı
 s : şimdiki durum (gerçekleştirilmiş hareketler)
 Γ : popülasyonun en iyi karıncalarının sayısı
 Ω : karıncanın mevcut uygun komşuluğu
 α : feromon bilgisinin önem ağırlığı
 β : sezgisel bilginin önem ağırlığı
 γ : ÇAAQA ödül değeri
 η : sezgisel bilgi
 π : tesislerin yerleşimlere bir dizilimi
 ρ : buharlaştırma katsayısı
 τ : feromon bilgisi

Alt indisler

i, j : tesis numarası

k, p : yerleşim numarası

Kısaltmalar

- 2-YFDT : Seviye-2 Yeniden Formülasyon Doğrusallaştırma Tekniği
3AP : 3-boyutlu Atama Problemi
BİKA : Bölümlendirilmemiş İki Kriter (UnsortBicriterion) Algoritması
BİKA-M : Bölümlendirilmemiş İki Kriter (UnsortBicriterion) Maliyet Fonksiyonlu Algoritması
BiKAP : Bikaresel Atama Problemi (BiQAP: Biquadratic Assignment Problem)
ÇAAQA : Çok-Amaçlı Ant-Q (MOAQ: Multiple Objective Ant-Q) Algoritması
ÇAKAP : Çok-Amaçlı Karesel Atama Problemi (MOQAP: Multiple Objective Quadratic Assignment Problem)
ÇAKKO : Çok-Amaçlı Karınca Koloni Optimizasyonu (Multiobjective Ant Colony Optimization)
ÇAKKOM : Çok-Amaçlı Karınca Koloni Optimizasyonu Metasezgiseli (MOACOM: Multiple Objective Ant Colony Optimization Metaheuristic)
ÇAKKOY : Çok-Amaçlılara Karınca Koloni Optimizasyonu Yaklaşımı (ACOAMO: Ant Colony Optimization Approach with Multiple Objectives)
Çİ : Çift İzlek (Dual Procedure:DP)
ÇİKKA-*poyraz* : Çarpımlı İki Kriterli Karınca Algoritması-*poyraz* (MBicriterionAnt-*boreas* : Multiplied BicriterionAnt-*boreas*)
ÇKKS-TSARP: Çoklu Karınca Koloni Sistemi-Teslim Süreleri ile Araç Rotalama Problemi (MACS-VRPTW: Multi Ant Colony System-Vehicle Routing Problem with Time Windows)
DAP : Doğrusal Atama Problemi
DKA : Değişken Komşuluk Arama (VNS: Variable Neighborhood Search)
EKÇAS : Ehil Karıncalar Çok-Amaçlı Sistemi (COMPETants)
GLB : Gilmore ve Lawler Alt Sınırı
GRIBB : Büyük Uluslararası Dal-Sınır Araştırması (Great International Branch-and-Bound Search)
GTTA : Geliştirilmiş Tekrarlı Tabu Arama (IITS: Improved Iterated Tabu Search)
HTAAP : Hırslı Tesadüfsele Adaptif Arama Prosedürü (GRASP: Greedy Random Adaptive Search Procedure)
İAKAP : İki Amaçlı Karesel Atama Problemi (Biobjective Quadratic Assignment)
İKÇKA : İki Kriterli Çoklu Koloni (BicriterionMC) Algoritması
İKÇKA-M : İki Kriterli Çoklu Koloni (BicriterionMC) Maliyet Fonksiyonlu Algoritması
İKKA : İki Kriterli Karınca (BicriterionAnt) Algoritması
İKKA-M : İki Kriterli Karınca (BicriterionAnt) Maliyet Fonksiyonlu Algoritması
JIT : Just-in-time (tam zamanında)
K3AP : Karesel 3-boyutlu Atama Problemi

KAP : Karesel Atama Problemi (Quadratic Assignment Problem)
KDAP : Karesel Darboğaz Atama Problemi (QBAP: Quadratic Bottleneck Assignment Problem)
KKO : Karınca Kolonileri Optimizasyonu (Ant Colony Optimization)
KKOÇAŞ : Karınca Kolonileri Optimizasyonu Çok-Amaçlı Şebekesi (MONACO: Multi-Objective Network ACO)
KKS : Karınca Koloni Sistemi
KKS-A : Karınca Koloni Sistemi – Araç (ACS-VEI)
KKS-Z : Karınca Koloni Sistemi – Zaman (ACS-TIME)
KS : Karınca Sistemi
KTDP : Karışık Tamsayılı Doğrusal Programlama (MILP: Mixed Integer Linear Programming)
KYAP : Karesel Yarı-Atama Problemi
NP-zor : Belirli Olmayan Polinomsal-Zaman Zor (Nondeterministic polynomial-time hard)
PKKOA : Pareto Karınca Koloni Optimizasyonu Algoritması (P-ACO: Pareto-Ant Colony Optimization)
SDRMS : Semi-Definite Relaxation-Matrix Splitting (yarı-kesin gevşetme-matrisi bölmelendirme)
TİKKA-*poyraz* : Toplamlı İki Kriterli Karınca Algoritması-*poyraz* (SBicriterionAnt-*boreas* : Summed BicriterionAnt-*boreas*)
TTA : Tekrarlı Tabu Arama (ITS: Iterated Tabu Search)
TYA : Tekrarlı Yerel Arama (ILS: Iterated Local Search)
WIP : Work-in-Process (yarı mamül)
YFDT : Yeniden Formülasyon Doğrusallaştırma Tekniği
YKP : Yarı-kesin programlama (Semi-definite programming)

İKİ KRİTERLİ KARESEL ATAMA PROBLEMLERİNİN ÇOK-AMAÇLI KARINCA KOLONİLERİ ALGORİTMALARI İLE OPTİMİZASYONU

Celal ÖZKALE

Anahtar Kelimeler: Karınca Kolonileri Algoritması, Çok-Amaçlı Optimizasyon, Karesel Atama Problemi, Metasezgiseller, *poyraz* Algoritması.

Özet: Çok-amaçlı birleşimsel optimizasyon problemlerini geleneksel yöntemlerle çözmeye zorluğu, araştırmacıları daha iyi performansa sahip yaklaşımları araştırmaya yöneltmiştir. Son yıllarda Karınca Kolonisi Optimizasyonu metasezgiselini esas alan, çok-amaçlı problemleri çözmeye yönelik bazı algoritmalar önerilmektedir. Bu çalışmada, bu tür algoritmalar incelenmiş ve İki Kriterli Karesel Atama Problemlerini çözmek ve algoritmaların performanslarını değerlendirmek üzere ilk kez programlanmıştır. 12 adet Çok-Amaçlı Karınca Kolonisi Optimizasyonu algoritmasına ait gürbüz parametre düzenleri bulunmuş, İki Kriterli Karesel Atama Problemleri bu parametre düzenleri ile çözülmüştür. Performansları, bu algoritmalarla elde edilen pareto-önyüzlerin karşılaştırılması yoluyla değerlendirilmiştir. Değerlendirme aşamasında, pareto-önyüze odaklanan yeni bir yaklaşım ortaya konmaktadır. Bu çalışmada, İKKAP'ları çözmek üzere *poyraz* adı verilen yeni bir algoritma önerilmekte ve İKKA algoritmasına uygulanmasıyla global pareto-önyüzde bulunan toplam karar seçeneği sayısı %94,34 arttırılmaktadır.

THE OPTIMIZATION OF BIOBJECTIVE QUADRATIC ASSIGNMENT PROBLEMS BY MULTI-OBJECTIVE ANT COLONY ALGORITHMS

Celal ÖZKALE

Keywords: Ant Colony Algorithm, Multi-Objective Optimization, Quadratic Assignment Problem, Metaheuristics, *boreas* Algorithm.

Abstract: The difficulty of resolving the multi-objective combinatorial optimization problems with traditional methods have directed the researchers to investigate new approaches which have better performance. In the last years some algorithms based on Ant Colony Optimization metaheuristic have been suggested to solve these multi-objective problems. In this study these algorithms have been reported and firstly programmed both to solve the Biobjective Quadratic Assignment Problems and to evaluate the performances of these algorithms. The robust parameter sets for each of 12 Multi-Objective Ant Colony Optimization algorithms have been calculated, Biobjective Quadratic Assignment Problems have been solved with these parameter sets. As a result the performances of the algorithms have been evaluated by means of comparing the pareto-fronts obtained from these algorithms. In the evaluation step, a new approach which focuses on the pareto-front is exposed. In the thesis, a new algorithm called *boreas* is suggested to solve BiQAPs, and the total number of decision alternatives found in global pareto-front is 94.34 percent increased by applying onto the Bicriterion Ant Algorithm.

1. GİRİŞ

Karesel Atama Problemi (KAP), NP-zor sınıfındaki en zor problemlerden biridir. Tesis yerleşimi, paralel ve dağıtık hesaplama ve birleşimsel veri analizi gibi alanlardaki çok sayıda gerçek hayat problemlerini modellemekte kullanılabilir. Birleşimsel optimizasyon problemlerinden gezgin satıcı probleminde olduğu gibi, en çok klik ve çizge bölmelendirme problemleri de bir KAP olarak formülize edilebilir.

Her tesisin net olarak bir yerleşime atanması problemini ele alalım. Yerleşimler arasındaki mesafeler, tesisler arasındaki talepler (akışlar) ve genel olarak yerleşime karşılık tesis atama maliyetleri bilinmekte olsun. Uluslararası literatür KAP'ı mesafe-akış sonuçlarının toplamı olarak alınan maliyetler çerçevesinde, tesislerin yerleşimlere paylaştırılmasının bir minimum maliyetini bulma problemi olarak tanımlamaktadır. Bu konuda 1999'dan beri 100'ün üzerinde makale yayınlanmıştır.

İlk formülasyonu Koopmans ve Beckman (1957) tarafından ortaya atıldığından beri KAP, sadece pratik ve kuramsal öneminden dolayı değil, aynı zamanda karmaşıklığından da dolayı dünya çapında araştırmacıların dikkatini çekmiştir. KAP en zor birleşimsel optimizasyon problemlerinden biridir. Genelde $n > 30$ büyüklüğündeki örnekler makul süreler içerisinde çözülememektedirler. Sahni ve Gonzales (1976) KAP'ın NP-zor olduğunu ve $P=NP$ olmadıkça, sabit bir f için bir f -tahmin (yakınsama) algoritması ortaya koymanın mümkün olmadığını göstermişlerdir. Akışlar ve mesafeler, simetrik katsayı matrisleri olarak görüldüğünde bile bu geçerlidir. Yüksek hesaplama karmaşıklığına rağmen KAP, GRIBB projesi (great international branch-and-bound search) için ilk büyük test uygulaması olarak seçilmiştir. Bu proje ile internete bağlı dünyadaki belirli sayıda bilgisayarın kullanımı sayesinde büyük boyutlardaki paralel araştırma problemlerini çözmek için bir yazılım kütüphanesi oluşturmaya çalışılmıştır. Testlerin ilk sonuçları Moe (2003) tarafından sunulmuştur.

KAP örnekleri için yeni yapısal özellikleri tanımlamada, çeşitli bakış açılarını esas alan çok sayıda formülasyon görülmektedir. Bunlar tamsayılı programlama, pozitif yarı-kesin programlama, kesikli ve birleşimsel matematik, çizge ve grup kuramı vb. şekilde sınıflandırılabilir. Çoğu eşit olan bu formülasyonlar KAP'ın zorluğunu düşündüğümüzde yeni çözüm tekniklerinin gelişimi için matematiksel kaynakların artmasına cesaret vermektedir.

Çok-amaçlı optimizasyon problemleri, eş zamanlı olarak optimize edilmiş olmak zorunda olunan, özellikle bundan dolayı problem çözmeyi zor hale getiren birkaç amaç tarafından karakterize edilmektedirler. Bu problemler için metasezgisellerin kullanımı son on yıl içerisinde artan bir ilgi görmektedir. Gerçek dünyadaki çoğu çok-amaçlı problemlerin varlığı, onların gerçekte karmaşıklığı ve metasezgisel prosedürlerinin avantajları, bu problemlerle uğraşmak, son birkaç yılda bu araştırma alanını güçlendirmiştir (Gandibleux ve diğ., 2004).

Çok-amaçlı birleşimsel optimizasyon problemlerini geleneksel yöntemlerle çözme zorluğu, araştırmacıları, alternatifler ve daha iyi performans veren yaklaşımlar araştırmaya sevk etmiştir. Son zamanlarda Karınca Kolonisi Optimizasyonu (KKO) metasezgiselini esas alan çok-amaçlı problemleri çözmeye yönelik bazı algoritmalar önerilmiştir.

KKO çeşitli karınca türlerinin en kısa yolu aramasından esinlenmiş olan bir metasezgiseldir. Dorigo ve diğ. (1996)'nin ilk KKO algoritması (Karınca Sistemi) üzerine çalışmaya başlamalarından beri, birkaç araştırmacı, gezgin satıcı problemi, karesel atama problemi, sıralı atama problemi, üretim çizelgeleme, programlama, proje çizelgeleme, araç rotalama, uziletişim (telecommunication) rotalama, yatırım planlama, personel çizelgeleme içinde olmak üzere, Cordon ve diğ (2002), Dorigo ve Stütze (2003) çalışmalarında da raporlanan bu gibi kombinasyonel problemleri tam anlamıyla çözmek üzere yapılmış farklı KKO algoritmaları geliştirmişlerdir

Son zamanlarda, bazı araştırmacılar çok-amaçlı problemlerle uğraşmak için KKO algoritmaları tasarlamışlardır. ÇAKKO (Çok-amaçlı Karınca Koloni Optimizasyonu) olarak isimlendirilen bu algoritmaların çoğu çizelgeleme, araç rotalama ve portföy

seçiminin de içinde bulunduğu çeşitli çok-amaçlı problemleri çözmek için özel önerilerdir. Bütün bunlara rağmen henüz çok amaçlı KAP 'ları çözmek üzere ÇAKKO algoritmalarının kullanılmamış olması bu çalışmanın başlangıç noktası olmuştur. Literatürde mevcut ve çok sayıda atıf almış ÇAKKO algoritmaları iki kriterli KAP'ları çözmek üzere programlanarak kendi içinde karşılaştırmaları, literatürdeki yöntemlerden farklı bir kriter ve yöntemle yapılmıştır. Ayrıca *poyraz* adını verdiğimiz ve ilk kez bu tezde önerdiğimiz bir algoritma, başarılı sonuçlar veren İKKA algoritmasına uygulanarak şaşırtıcı sonuçlar elde edilmiştir.

2. bölümde tez ile ilgili genel konulardan bahsedilmiştir. 3. bölümde mevcut ÇAKKO algoritmaları detaylı olarak incelenmektedir. 4. bölümde iki kriterli KAP'lar incelenmiş ÇAKKO algoritmaları ile çözümlenerek algoritmaların performansları yeni bir yaklaşımla değerlendirilmiştir. *poyraz* eklenti algoritması tanıtılarak, iyi sonuçlar veren İKKA algoritmasına uygulanmış ve elde edilen sonuçlar paylaşılmıştır. 5. bölümde tez çalışmalarında elde edilen sonuçlar ve öneriler üzerinde bir değerlendirme yapılmaktadır.

2. KARESEL ATAMA PROBLEMİ

2.1 Karesel Atama Problemi'ne Giriş

Karesel Atama Problemi (KAP)'nin NP-zor sınıfındaki en zor problemlerden biri olduğu ve çok sayıda gerçek hayat problemlerini modellemekte kullanılabildiği önceki bölümde belirtilmişti.

Koopmans ve Beckmann (1957) KAP'ı ilk olarak ekonomik aktivitelere uyarlayan bir matematiksel model olarak önerdiğinden bu yana bu problem üzerine çok sayıda çalışma yapılmıştır. Özellikle aşağıdaki çalışmaları pratik uygulama alanında gözlemlemek mümkündür: Steinberg (1961) KAP'ı bir baskılı devre bağlantılama (backboard wiring) problemindeki parçalar arasındaki bağlantı sayısını minimize etmek için kullandı; Heffley (1972, 1980) ekonomik problemlere uyguladı; Francis ve White (1974) yeni bir tesisin (polis karakolu, süpermarket, okul vb.), müşterilerin verilen bir kümesine hizmet etmek amacıyla atanması için bir karar sistemi geliştirdiler; Geoffrion ve Graves (1976) çizelgeleme problemi üzerine odaklandılar; Pollatschek ve diğ. (1976) klavye ve kontrol panelleri için en iyi tasarımı belirlemek üzere KAP'ı kullandılar; Krarup ve Pruzan (1978) onu arkeolojiye uyguladı; Hubert (1987) istatistiksel analizde; Forsberg ve diğ. (1994) reaksiyon kimyası analizinde faydalandı ve Brusco ve Stahl (2000) sayısal analizde kullandı.

Bütün bu çalışmalar içerisinde, yerleşim düzenleme problemi, KAP için en popüler uygulamadır: Dickey ve Hopkins (1972) KAP'ı bir üniversite yerleşkesinde binaların atanmasına uyguladılar, Elshafei (1977) bir hastanenin planlamasında ve Bos (1993) orman parkları ile ilişkili bir probleme uyguladılar. Benjaafar (2002) yarı mamulleri (WIP: work-in-process) minimize etmek amacıyla bir tesis yerleşim düzeni tasarımı problemini formüle etti. Çalışmasında, bir WIP tabanlı formülasyon kullanılarak elde edilen yerleşim düzeninin, geleneksel KAP formülasyonu kullanılarak elde edilenden çok farklı olabileceğini göstermektedir. Örneğin bir KAP-en iyi yerleşim düzeni

WIP-olursuz olabilmektedir. Rabak ve Sichman (2003), Miranda ve diğ. (2005) ve Duman ve İlhan (2007) elektronik parçaların yerleşimi üzerinde çalışmışlardır.

Ben-David ve Malah (2005) indeks atama problemini haberleşmede hata kontrolünde uygulamışlardır ve bu problem KAP'ın özel bir durumu olarak gösterilmiştir. Wess ve Zeitlhofer (2004) sinyal işlemcilerdeki bellek yerleşim düzeni optimizasyonu problemi üzerinde çalıştılar. Diğer uygulamalardan bazıları Scriabin ve Vergin (1975), Hubert ve Schulz (1976), Heffley (1977), Los (1978), Khare ve diğ. (1988a, 1988b), Krackhardt (1988), Bland ve Dawson (1991), Balakrishnan ve diğ. (1992), Lacksonen ve Enscoe (1993), Medova (1994), Phillips ve Rosen (1994), Gouveia ve Voß (1995), Bozer ve Suk-Chul (1996), Talbot ve Cawley (1996), White (1996), Mason ve Rönnqvist (1997), Ostrowski ve Ruoppila (1997), Ball ve diğ. (1998), Haghani ve Chen (1998), Kochhar ve diğ. (1998), Martin (1998), Sarker ve diğ. (1998), Spiliopoulos ve Sofianopoulou (1998), Tansel ve Bilen (1998), Tavakkoli-Moghaddain ve Shayan (1998), Urban (1998), Gong ve diğ. (1999), Rossin ve diğ. (1999), Bartolomei-Suarez ve Egbelu (2000), Ho ve Moodie (2000), Urban ve diğ. (2000), Hahn ve Krarup (2001), Pitsoulis ve diğ. (2001), Takagi (2001), Siu ve Chang (2002), Wang ve Sarker (2002), Youssef ve diğ. (2003), Yu ve Sarker (2003), Ciriani ve diğ. (2004), Solimanpur ve diğ. (2004) ve Abbiw-Jackson ve diğ. (2006) içerisinde bulunabilir.

KAP'ın öneminden önceki bölümde bahsedilmişti. Ayrıca öneminin bir kısmının gezgin satıcı problemi gibi NP-zor birleşimsel optimizasyon problemleri olan kutu paketleme problemi ve en çok klik problemi vb. problemlerin KAP gibi modellenebilir olmasından geldiğini belirtebiliriz. Algoritmaların performanslarının karşılaştırılması için kabul edilen bir eğilim de optimum bilinmediği durumda veya kesin algoritmaların bu örneklerde kullanımı mümkün olduğunda dahi internet-erişimli (literatür) örneklerde yerel optimumu araştırmaktır (Burkard ve diğ., 1996, Çela, 1998). KAP için de bu tür çok sayıda performansa internet ve basılı yayınlar üzerinden ulaşmak mümkündür. KAP vakasında, son zamanlarda kanıtlanan optimal çözümleri ile şu örneklerden bahsedebiliriz (QAPLIB, 2009): Bur26 (b'den h'ye kadar) Hahn (2004) tarafından ve Tai25a Hahn (2003) tarafından; Ste36a Brixius ve Anstreicher (2001) tarafından; Bur26a Hahn (2001) tarafından; Kra30a Hahn ve diğ.

(2000) tarafından; Kra30b, Kra32 ve Tho30 Anstreicher ve diğ. (2000) tarafından; Nug27, Nug28 ve Nug30 Anstreicher ve diğ. (2000) tarafından; Ste36b ve Ste36c Nyström (1999) tarafından; Nug25 Brünger ve Marzetta (1997) tarafından; Tai25b Giovanetti (1997) tarafından; Tai20b Hahn (1997) tarafından; Esc32g Bruenger ve diğ. (1996) tarafından; Nug24 Clausen ve diğ. (1996) tarafından; Esc32e, Esc32f, Had18, Had20, Nug21, Nug22, Rou20, Tai17a ve Tai20a Bruenger ve diğ. (1996) tarafından; Had16 Hahn ve diğ. (1996) tarafından. Zhang (2009) KAP için pivotlama tabanlı, etkili bir algoritma geliştirdi (QAPLIB, 2009). Nug, Tai, Chr, Had, Rou, Bur, Esc ve Tho $n = 12$ 'den 40'a kadar örnekler üzerinde bu algoritmayı denedi. Bu yöntemle en iyi problemler (Nug30 dahil) için birkaç saniyeden birkaç dakikaya kadarlık süreler içerisinde optimumları bulmayı başarmıştır. Daha önce literatürde yayınlanmamış 80 civarındaki optimum çözümü bu çalışmasında sunmuştur. Tabii ki son yıllardaki işlemcilerdeki ve belleklerdeki gelişmelerin de bu başarıda payı olduğunu göz ardı etmemek gerekir.

Mittleman ve Peng 2007 yılında SDRMS (Semi-Definite Relaxation-Matrix Splitting) adını verdikleri yeni bir sınırı geliştirdiklerini duyurdular. Bu sınırı kullanarak Taixb'nin birkaç büyük örneği için alt sınırları geliştirdiler. Bu konudaki yayınları hazırlanmaktadır (QAPLIB, 2009). Klerk ve Sotirov (2007) Escxx örneklerinden birkaçının sınırını geliştirdi. Esc64 örneği için YKP (Yarı-kesin programlama) sınırı ile çok önemli sonuçlar aldılar. 2007 yılında Misievicius Tai100a için 2003, 2004, 2005 yıllarında bulmuş olduğu alt sınırı geliştirdi. Yeni en iyi bilinen çözüm GTTA (Geliştirilmiş Tekrarlı Tabu Arama) algoritması kullanılarak bulunmuştur ve makalesi hazırlanma aşamasındadır. Hahn 2005 yılında Tai50b için yeni bir en iyi alt sınırı duyurdu. Bu alt sınır Hahn ve Grant (1998)'in Çİ (Çift İzlek) algoritması kullanılarak bulunmuştu. Misievicius 2002, 2003, 2004'te geliştirmiş olduğu Tai80a için alt sınırı son olarak 2005'te TTA (Tekrarlı Tabu Arama) algoritması kullanarak geliştirmiştir (Misievicius, 2008). Burer and Vandebussche (2004) Esc32a-d,h ve Tai35a,b problemleri için yeni sınırlar elde ettiğini duyurdu. Hahn 2004 yılında Hahn ve diğ. (2001) tarafından geliştirilmiş Seviye-2 Yeniden Formülasyon Doğrusallaştırma Tekniği'ni kullanarak Tai30b için daha iyi bir alt sınır buldu. 2003'te Misievicius Tai50a ve Tai60a için bilinen en iyi çözümü bir değiştirilmiş tabu arama kullanarak ortaya koydu. 2002'de Tho150 için

bilinen en iyi çözümleri bir tavlama benzetimi algoritması kullanarak buldu. Karisch ve diğ. (1999) Bur26x, Kra30a ve Tai25a örnekleri için alt sınırları geliştirdiler. Esc32a, Esc32b, Esc32c, Esc32d ve Esc32h örnekleri için Kaibel (1997a, 1997b) alt sınırlar geliştirdi. Stütze(1997) Tai256c için bilinen en iyi çözümleri geliştirdi. Taillard ve Gambardella (1997) Tai150b için en iyi çözümleri ürettir. Hahn ve diğ. (1998) Tai20b, Tai25a, and Tai25b örneklerinin yeni alt sınırlarını 1996'da duyurmuşlardı.

KAP için literatür örneklerinin optimal sonuçlarının veya en iyi alt sınırlarının bulunması çabalarının yanında bir de örnek üretmede de rekabet söz konusudur. Burkard ve diğ. (1991, 1997), Li ve Pardalos (1992) ve QAPLIB'de yeni örnekleri erişilebilir hale getirdiler. Aynı zamanda algoritmaları test için kullanılabilen, bilinen optimum değerler ile örnek üreticileri mevcuttur (Çela, 1998). Son olarak, Palubeckis (1999, 2000), Drezner ve diğ. (2004) ve Stütze ve Fernandes (2004) metasezgiseller için zor olduğu rapor edilen yeni örnek düzenleri sunmaktadırlar.

Birleşimsel optimizasyon üzerinde çalışan uzmanlar, NP-zor problemlerin kısmen polinomsal olarak çözülebilir versiyonları için ve problem örneklerinin zorluğunu ölçmek üzere araştırma mekanizmalarını araştırmaya yönelmişlerdir. KAP için baktığımızda, Christofides ve Gerard (1981) KAP'ın bazı özel örnekleri üzerinde çalışmışlardır. Sylla ve Babu (1987) bir sıralı karesel atama problemi için bir metodoloji geliştirmişlerdir; Chen (1995) farklı bir KAP vakası sunmuştur, Çela (1998) birkaç polinomsal olarak çözülebilir örnek sunarak takip etmiştir, Herroeven ve Vangils (1985), Cyganski ve diğ. (1994), Mautor ve Roucairol (1994b) Palubetski'nin KAP örneklerinin yozlaşmış olduğunu gösterdiler; Angel ve Zissimopoulos (1998, 2000, 2001, 2002) akış ve uzaklık kümelerinin değişkenliği üzerine kurulmuş diğer KAP örneklerinin zorluğunu tartıştı; Abreu ve diğ. (2002) çözüm maliyetlerinin değişkenliği için polinomsal bir açıklama çıkardılar ve örneklerin zorluğunun bir ölçüsünü tanımladılar ve Barvinok ve Stephen (2003) KAP çözüm değerlerinin bir dağılımını oluşturdular.

KAP örnekleri için yeni yapısal özellikleri tanımlama rekabetinde, çeşitli bakış açılarını esas alan çok sayıda formülasyon görülmektedir. Bunlar tamsayılı programlama, pozitif yarı-kesin programlama, kesikli ve birleşimsel matematik,

çizge ve grup kuramı, veya spektral kuram yoluyla lineer cebir gibi sınıflandırılabilir. Daha genel problemleri tanımlayanları dışında tutarsak, bu formülasyonların çoğu eşittir. Bu formülasyonlar KAP'ın zorluğunu düşündüğümüzde yeni çözüm tekniklerinin gelişimi için matematiksel kaynakların artmasına cesaret vermektedir.

2.2. KAP ve KAP ile İlişkili Problemlerin Formülasyonları

Bu bölümde, literatürde bilinen en önemli KAP formülasyonlarına ve her formülasyon için uygulanmış çözüm yaklaşım türlerine değinilmektedir.

2.2.1. Seçilmiş KAP formülasyonları

2.2.1.1. Tamsayılı doğrusal programlama formülasyonları

İlk olarak KAP'ı ikili kısıtların gevşetildiği bir 0-1 tamsayılı doğrusal programlama problemi olarak görmekteyiz. Boolean formülasyonu başlangıçta Koopmans ve Beckmann (1957) tarafından önerilmişti ve daha sonra şu çalışmalar gibi bazı çalışmalarda da kullanıldı: Steinberg (1961), Lawler (1963), Gavett ve Plyter (1966), Elshafei (1977), Bazaraa ve Sherali (1979), Bazaraa ve Kirca (1983), Christofides ve Benavent (1989), Bos (1993), Mans ve diğ. (1995), Jünger ve Kaibel (2000, 2001a, 2001b), Liang (1996), Torke ve diğ. (1996), Tsuchiya ve diğ. (1996, 2001), Ball ve diğ. (1998), Ishii ve Sato (1998), Kaibel (1998), Kochhar ve diğ. (1998), Martin (1998), Spiliopoulos ve Sofianopoulou (1998), Siu ve Chang (2002), Yu ve Sarker (2003) ve Fedjki ve Duffuaa (2004).

f_{ij} i ve j tesisleri arasındaki akış, ve d_{kp} k ve p yerleşimleri arasındaki mesafe olsun. Hesaplamak için hedefimiz:

$$\min \sum_{i,j=1k,p=1}^n \sum_{i,j=1k,p=1}^n f_{ij} d_{kp} x_{ik} x_{jp} \quad (2.1)$$

$$\text{aka} \quad \sum_{i=1}^n x_{ij} = 1 \quad 1 \leq j \leq n, \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad 1 \leq i \leq n, \quad (2.3)$$

$$x_{ij} \in \{0,1\} \quad 1 \leq i, j \leq n. \quad (2.4)$$

Yerleşimlere aktivitelerin atanmasının maliyetini ele alalım. Yerleşimlere tesisleri konumlandırma maliyeti b_{ik} olduğunda, n adeti için bir KAP örneğinde genel şekliyle üç matris $F = [f_{ij}]$, $D = [d_{kp}]$ ve $B = [b_{ik}]$ ifadeleriyle açıklanır. İlk iki matris tesisler arası akışları ve yerleşimler arası mesafeleri tanımlamaktadır. Bu problem şöyle tanımlanabilir:

$$\begin{aligned} \min \quad & \sum_{i,j=1k,p=1}^n \sum_{i,k=1}^n f_{ij} d_{kp} x_{ik} x_{jp} + \sum_{i,k=1}^n b_{ik} x_{ik} \\ \text{aka} \quad & (2.2), (2.3) \text{ ve } (2.4). \end{aligned} \quad (2.5)$$

Akışlar ve mesafelerin sonuçlarını ele almaya gerek duymayan, c_{ijkp} maliyetlerini içeren daha genel bir KAP versiyonu Lawler (1963) tarafından önerildi. Lawler formülasyonu şöyledir:

$$\begin{aligned} \min \quad & \sum_{i,j=1k,p=1}^n \sum_{i,k=1}^n c_{ijkp} x_{ik} x_{jp} \\ \text{aka} \quad & (2.2), (2.3) \text{ ve } (2.4). \end{aligned} \quad (2.6)$$

Bu model aynı zamanda, Bazaraa ve Elshafei (1979), Drezner (1995), Sarker ve diğ. (1995, 1998), Brüngger ve diğ. (1997, 1998), Chiang ve Chiang (1998), Hahn ve Grant (1998), Hahn ve diğ. (1998), Gong ve diğ. (1999) ve Rossin ve diğ. (1999)nde de kullanıldı.

2.2.1.2. Karışık tamsayılı doğrusal programlama (KTDP) formülasyonları

Bir KTDP formülasyonu olarak KAP, literatürde farklı biçimlerde bulunmaktadır. Bunların hepsinde karesel ifadeler doğrusal ifadelerle terk edilmiştir. Örneğin, Lawler (1963) n^4 değişken kullandı: $c_{ijkp} = f_{ij} d_{kp}$ ve $y_{ijkp} = x_{ik} x_{jp}$, $1 \leq i, j, k, p \leq n$.

Diğer formülasyonlar orijinal problemin gevşetmelerini kullanmaktadır. Bu kategoride, Love ve Wong (1976a, 1976b), Kaufman ve Broeckx (1978), Bazaraa ve Sherali (1980), Christofides ve diğ. (1980), Burkard ve Bonniger (1983), Frieze ve Yadegar (1983), Assad ve Xu (1985), Adams ve Sherali (1986), Christofides ve

Benavent (1989), Adams ve Johnson (1994), Drezner (1995), Gouveia ve VoB (1995), Milis ve Magirou (1995), Padberg ve Rijal (1996), White (1996), Ramachandran ve Pekny (1998), Karisch ve diğ. (1999) ve Ramakrishnan ve diğ. (2002)nin yayınlarını da saymak mümkündür.

Genelde, KTDP modelleri üzerine temellenmiş KAP doğrusallaştırmaları, gerçekte popüler olmayan bir hale getiren bu yaklaşımdan dolayı, çok sayıda kısıt ve değişken sunmaktadır. Bu doğrusallaştırmalar, bazı kısıt gevşetmeleri ile birlikte, optimal çözüm için çok sayıda alt sınır geliştirmekte başarılı olmuşlardır. Buna örnek olarak, Kaufman ve Broeckx (1978), Bazaraa ve Sherali (1980), Frieze ve Yadegar (1983), Adams ve Sherali (1986), Adams ve Johnson (1994) ve Padberg ve Rijal (1996) çalışmalarını sayabiliriz. Çela (1998) şu 3 adet KAP doğrusallaştırmasından bahsetmektedir: sınırlamaların daha az sayıda olmasını sağlayan Kaufman ve Broeckx (1978); Lagrangean gevşetmesi yoluyla en iyi alt sınırları başarmak için Frieze ve Yadegar (1983) ve politop (sınırlı çokyüzlü-sınırlı polihedral) tanımlaması sebebiyle Padberg ve Rijal (1996)in çalışması. Frieze ve Yadegar (1983) tarafından sunulan formülasyon, n^4 gerçek değişken, n^2 Boolean değişken ve $n^4 + 4n^3 + n^2 + 2n$ kısıt kullanarak doğrusal bir form içinde KAP'ı tanımlamaktadır. Yazarlar (2.1)-(2.4) eşitliklerine eşit olan aşağıdaki (2.7)-(2.16) denklemlerinde verilen formülasyonu göstermektedir (Çela, 1998).

$$\min \sum_{i,j=1k,p=1}^n \sum_{i,j=1k,p=1}^n f_{ij} d_{kp} \cdot y_{ijkp} \quad (2.7)$$

$$\text{aka} \quad \sum_{i=1}^n x_{ik} = 1 \quad 1 \leq k \leq n, \quad (2.8)$$

$$\sum_{k=1}^n x_{ik} = 1 \quad 1 \leq i \leq n, \quad (2.9)$$

$$\sum_{i=1}^n y_{ijkp} = x_{jp} \quad 1 \leq j, k, p \leq n, \quad (2.10)$$

$$\sum_{j=1}^n y_{ijkp} = x_{ik} \quad 1 \leq i, k, p \leq n, \quad (2.11)$$

$$\sum_{k=1}^n y_{ijkp} = x_{jp} \quad 1 \leq i, j, p \leq n, \quad (2.12)$$

$$\sum_{p=1}^n y_{ijkp} = x_{ik} \quad 1 \leq i, j, k \leq n, \quad (2.13)$$

$$y_{iik} = x_{iik} \quad 1 \leq i, k \leq n, \quad (2.14)$$

$$x_{ik} \in \{0,1\} \quad 1 \leq i, k \leq n, \quad (2.15)$$

$$0 \leq y_{ijkp} \leq 1 \quad 1 \leq i, j, k, p \leq n. \quad (2.16)$$

2.2.1.3. Permütasyonlar yoluyla formülasyonlar

Komşu yerleşimlere tesis maliyetlerinin atanması, tesisler arasındaki akışlara ve yerleşimler arasındaki mesafelerle orantılıdır. Bu orantılılıktan ortaya çıkan ve permütasyon kavramını kullanan KAP formülasyonu Hillier ve Michael (1966), Graves ve Whinston (1970), Pierce ve Crowston (1971), Burkard ve Stratman (1978), Roucairol (1979, 1987), Burkard (1984), Frenk ve diğ. (1985), Bland ve Dawson (1991, 1994), Battiti ve Tecchiolli (1994), Bui ve Moon (1994), Chakrapani ve Skorin-Kapov (1994), Fleurent ve Ferland (1994), Li ve diğ. (1994b), Mautor ve Roucairol (1994a, 1994b), Li ve Smith (1995), Taillard (1995), Bozer ve Suk-Chul (1996), Colorni ve diğ. (1996), Huntley ve Brown (1996), Peng ve diğ. (1996), Tian ve diğ. (1996, 1999), Cung ve diğ. (1997), Mavridou ve Pardalos (1997), Merz ve Freisleben (1997), Nissen (1997), Pardalos ve diğ. (1997), Angel ve Zissimopoulos (1998), Deineko ve Woeginger (1998), Talbi ve diğ. (1998a, 1998b, 2001), Tansel ve Bilen (1998), Abreu ve diğ. (1999), Fleurent ve Glover (1999), Gambardella ve diğ. (1999) ve Maniezzo ve Colorni (1999)de bulunabilir. Yakın geçmişte şu makaleler çıkmıştır: Ahuja ve diğ. (2000), Angel ve Zissimopoulos (2000, 2001, 2002), Stützle ve Hoos (2000), Arkin ve diğ. (2001), Pitsoulis ve diğ. (2001), Abreu ve diğ. (2002), Gutin ve Yeo (2002), Hasegawa ve diğ. (2002) ve Boaventura-Netto (2003). Costa ve Boaventura-Netto (1994) simetrik olmayan KAP'ı, yönlü çizge (directed graph) formülasyonu aracılığıyla çalıştılar.

n elemanlı tüm permütasyonların kümesi S_n ve $\pi \in S_n$ olsun. f_{ij} i ve j tesisleri arasındaki akışlar ve $d_{\pi(i)\pi(j)}$ $\pi(i)$ ve $\pi(j)$ yerleşimleri arasındaki mesafeler olsun. Her permütasyon π , tesislerin yerleşimlere bir paylaşımını temsil eder ve problem ifadesi şöyle olmaktadır:

$$\min_{\pi \in S_n} \sum_{i,j=1}^n f_{ij} d_{\pi(i)\pi(j)} \quad (2.17)$$

Bu formülasyon (2.1)-(2.4)te sunulan ilk formülasyona eşittir. (2.17)deki gibi, tüm $1 \leq i, j \leq n$ için aşağıdaki koşullar altında,

$$x_{ij} = \begin{cases} 1, & \pi(i) = j \text{ ise;} \\ 0, & \pi(i) \neq j \text{ ise.} \end{cases} \quad (2.18)$$

(2.2) ve (2.3) kısıtları S_n elemanları ile ilişkili $X = [x_{ij}]$ permütasyon matrislerini ifade eder.

2.2.1.4. İz (Trace) formülasyonu

Bu formülasyon lineer cebir tarafından desteklenmektedir ve maliyeti için KAP alt sınırlarını belirlemek amacı ile iz fonksiyonunu (matris esas köşegen elemanlarının toplamı) işletmektedir. Bu yaklaşımla KAP'a YKP'nın kullanımını mümkün kılan spektral kuramı uygulanabilir. İz formülasyonu Edwards (1980) tarafından şu şekilde ifade edilmektedir:

$$\min_{X \in S_n} \text{tr}(F.X.D.X^t). \quad (2.19)$$

Daha sonra bu yaklaşım şu birkaç çalışmada kullanıldı: Finke ve diğ. (1987), Hadley ve diğ. (1990, 1992a, 1992b, 1992c), Hadley (1994), Karisch ve diğ. (1994), Karisch ve Rendl (1995), Zhao ve diğ. (1998), Anstreicher ve diğ. (1999).

Aşağıdaki formülasyonlar Karisch ve diğ. (1994), Zhao ve diğ. (1998), Wolkowicz (2000), Wolkowicz ve diğ. (2000)de bulacağımız gibi Lagrangean dualinin duali doğrultusunda KAP gevşetmelerini açıklamaktadır. e her koordinatı 1'e eşit olan bir vektör olsun. X bir permütasyon matrisi ve B bir maliyet matrisi olursa, YKP formülasyonu şöyle olmaktadır:

$$\min \text{tr}(F.X.D - 2B)X^t \quad (2.20)$$

$$\text{aka } Xe = e, \quad (2.21)$$

$$X^t e = e, \quad (2.22)$$

$$X_{ij} \in \{0, 1\} \quad \forall i, j. \quad (2.23)$$

Bu yaklaşımı takiben Zhao ve diğ. (1998) tarafından ileri sürülen başka bir formülasyon şöyledir:

$$\min \quad \text{tr } F.X.D.X^t - 2BX^t \quad (2.24)$$

$$\text{aka} \quad XX^t = X^tX = I, \quad (2.25)$$

$$Xe = X^te = e, \quad (2.26)$$

$$X_{ij}^2 - X_{ij} = 0 \quad \forall i, j. \quad (2.27)$$

(2.20)-(2.23) ve (2.24)-(2.27) formülasyonlarından herhangi biri KAP için yarı-kesin gevşetmeleri açıklayabilir. Roupin (2004), ele alınan birleşimsel problemin bir mevcut doğrusal gevşetmesinden başlayarak, iki değerli değişkenler ile herhangi bir karesel veya doğrusal program için YKP gevşetmeleri elde etmek için bir basit algoritma sunmaktadır. Bu algoritma üç klasik birleşimsel problem için yarı-kesin gevşetmeler ortaya koymak üzere uygulanmaktadır: k-Küme Problemi, Karesel Atama Problemi ve Kısıtlı-Bellek Ayırtma Problemi.

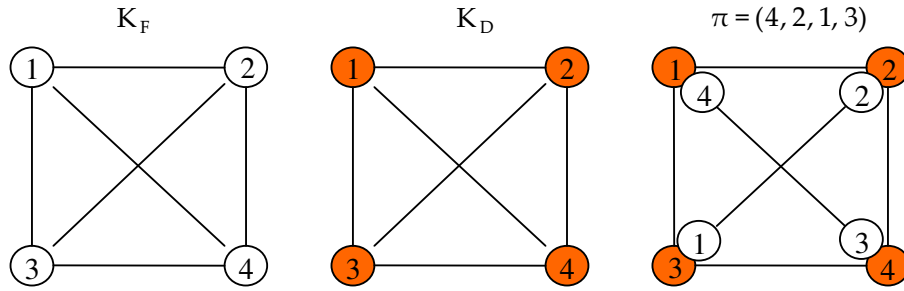
2.2.1.5 Çizge formülasyonu

İki yönsüz ağırlıklandırılmış, ilkinin kenarları akışlarla ikincisinininki mesafelerle ilişkili olan tam çizgeyi ele alalım. KAP, bir çizgenin düğümlerinin diğerinin düğümleri üzerine bir optimal paylaşılmasını bulma problemi olarak düşünülebilir. Şekil 2.1'de gösterilen bu formülasyonda çözüm maliyetleri, kenar ağırlıklarıyla ilişkili sonuçların toplamı olarak verilir.

Abreu ve diğ. (1999) tarafından uyarlanan cebirsel ve birleşimsel yaklaşım, Marins ve diğ. (2004)ne çizgi-grafik otomorfizmalarını barındıran yeni bir cebirsel çizge-kuramsal yaklaşımı tanımlaması için yüreklendirici olmuştur. Verilen bir çizge G 'nin çizgi-grafiği $L(G)$ ile temsil edilir. $L(G)$ 'nin bir kenarı G içinde komşu olan kenarların bir çifti olarak tanımlandığında, $L(G)$ 'nin bir düğümü olarak G 'nin her kenarı alınarak belirlenir. Bir çizge otomorfizması, kenarları içine alan düğümlerin bir permütasyonudur. Permütasyon kompozisyonu ile birlikte G 'nin tüm otomorfizmalarının kümesi $\text{Aut}(L(G))$ ile temsil edilen bir gruptur (Kreher ve Stinson, 1998).

Whitney (1932)'den bir teoremden, eğer $G = K_n$, $n \neq 2$ ve 4 ise, $\text{Aut}(G)$ ve $\text{Aut}(L(G))$ izomorfik grupları oluşturur. Bu sonuç üzerine, Marins ve diğ. (2004) KAP'ı çözenin bir $C_{n,2}$ permütasyonu olan, aşağıdaki ifadeyi minimize eden bir $\pi \in S_n$ permütasyonu veya bir $L(K_n)$ otomorfizması bulmak anlamına geldiğine dikkat çekmişlerdir:

$$\min_{\pi \in \text{Aut}(L(K_n))} \sum_{i=1}^N f_i d_{\pi(i)}. \quad (2.28)$$



Şekil 2.1: K_D üzerine K_F kliklerinin $\pi = (4, 2, 1, 3)$ permütasyonunda paylaşılması

KAP temsillerin birçoğunun dışbükeylik ve içbükeyliklerinin tartışıldığı White (1995) ve aynı zamanda bir m boyutlu grid üzerinde KAP için bir formülasyon sunan Yamada (1992) çalışmaları da önemlidir.

2.2.2. KAP ile ilişkili problemler

En klasik olarak KAP ile ilişkili problem, Macar Yöntemi ile kolaylıkla çözülen ve polinomsal olan Doğrusal Atama Problemi (DAP)'dir. Literatürde (örneğin, Burkard, 2002) bu problemin birkaç temsiline bulunması mümkündür.

İlk olarak Pierskalla (1967a, 1967b, 1968) tarafından önerilen 3-indeks Atama Problemi (3-boyutlu AP veya 3AP), π ve $\varphi \in S_n$ permütasyonlarını incelemekte ve bu sayede aşağıdaki ifadeyi enküçüklemetedir:

$$\min_{\pi, \varphi \in S_n} \sum_{i=1}^n c_{i\pi(i)\varphi(i)}. \quad (2.29)$$

Burkard ve Fröhlich (1980) 3AP'ı çözmek için bir dal-sınır algoritması önerdiler. Emelichev ve diğ. (1984) bu formülasyonu esas alan çoklu indeksler ile transportasyon modelleri tanımladılar. 3AP konusu bazı KAP uzmanları tarafından çalışılmıştır: Vlach (1967), Frieze (1974, 1983), Frieze ve Yadegar (1981), Burkard ve diğ. (1986, 1996a, 1996b), Euler (1987), Balas ve Saltzman (1989, 1991), Bandelt ve diğ. (1991), Crama ve Spieksma (1992), Balas ve Qi (1993), Burkard ve Rudolf (1993), Qi ve diğ. (1994), Magos ve Miliotis (1994), Poore (1994a, 1994b, 1995), Burkard ve Çela (1996), Magos (1996), Poore ve Robertson (1997) ve Burkard (2002).

KAP kuramsal çalışmalarının sahası çok geniştir. Karesel Darboğaz Atama Problemi, Bi-karesel Atama Problemi, 3-boyutlu KAP, Karesel Yarı-Atama Problemi ve Çok-amaçlı KAP olmak üzere bazı problem türleri de bu alan içerisine girmektedir. Bu problemlerden bazılarını Burkard (2002) rapor etmiştir.

2.2.2.1. Karesel darboğaz atama problemi (QBAP-KDAP)

Steinberg (1961) baskılı devre bağlantılamaya uygulamalar ile KAP'ın bir türü KDAP'ı ele aldı. O çalışmada, iki elemanı bağlamak için ihtiyaç duyulan kablonun uzunluğunu en küçükmek üzere, tekil yerleşimlerdeki n elemanın optimal bağlantısı için bir yerleştirme algoritması sunuldu. Yayının ana konusu: maksimum-kablo-uzunluğu normları arasında en azını optimal ağırlıklandırılmış-kablo-uzunluğuna eşitlemek. Bu kavram bir problemdeki tüm maliyeti en küçüklemeden sonra en büyük maliyeti en azlamak için daha iyi olabilen bir prensipten ortaya çıkmaktadır.

Darboğaz fonksiyonu terimi, önerilen KDAP genel programı, toplamlar için amaç fonksiyonundaki en büyük operasyonla değış tokuş edilerek KAP formülasyonundan elde edilir:

$$\min_{\pi \in S_n} \max \{f_{ij} d_{\pi(i)\pi(j)} : 1 \leq i, j \leq n\} \quad (2.30)$$

(2.30)'a ilişkin genel bir formülasyon Burkard ve Finke (1982), Burkard ve Zimmermann (1982), Kellerer ve Wirsching (1998) ve Burkard (2002) referanslarından ulaşılabilir.

2.2.2.2. Bikaresel atama problemi (BiQAP-BiKAP)

Burkard ve diğ. (1994)nde önerilen bu problem Burkard ve Çela (1995), Mavridou ve diğ. (1998) ve Burkard (2002) gibi diğer çalışmalarda da bulunabilir. Akış ve mesafe matrisleri n^4 sıraya sahiptir ve BiKAP formülasyonu şöyledir:

$$\min_{\pi \in S_n} \sum_{i, j, k, l=1}^n f_{ijkl} d_{\pi(i)\pi(j)\pi(k)\pi(l)} \quad (2.31)$$

2.2.2.3. Karesel 3-boyutlu atama problemi (K3AP)

Pierskalla (1967b) bu problemi bir teknik çalışma raporunda açıkladı. Çalışma açık literatürde hiç yayınlanmadı. Bundan dolayı yayın veritabanlarında bu konu bulunmamaktadır. Hahn ve diğ. (2004) K3AP'ı veri iletme sistem tasarımında ortaya çıkan bir problem üzerine çalışırken yeniden keşfetti. Çalışmanın esası daha yüksek seviyede işaret yıldız kümesi kullanarak çoklu iletmeler için haritalandırma çiftlerini ortaklaşa optimize etmektedir. Problem formülasyonu sonuçta şöyledir:

$$\min \left\{ \sum_{i=1}^N \sum_{j=1}^N \sum_{p=1}^N b_{ijp} u_{ij} w_{ip} + \sum_{i=1}^N \sum_{j=1}^N \sum_{p=1}^N \sum_{k=1}^N \sum_{n=1}^N \sum_{q=1}^N C_{ijpknq} u_{ij} u_{kn} w_{ip} w_{kq} \right\}, \quad (2.32)$$

: $u \in X, w \in X; u, w$ binary

aka

$$x \in X \equiv \left\{ x \geq 0 : j = 1, \dots, N \text{ için } \sum_{i=1}^N x_{ij} = 1; i = 1, \dots, N \text{ için } \sum_{j=1}^N x_{ij} = 1 \right\}. \quad (2.33)$$

2.2.2.4. Karesel yarı-atama problemi (KYAP)

Hansen ve Lih (1992) tarafından önerilmiş bu model kümeleme ve bölmelendirme problemlerini çözmek için özel bir yapıdır. Şöyle yazılabilir:

$$\min \sum_{k=1}^m \sum_{j=1}^n c_{ij} x_{ik} x_{jk} \quad (2.34)$$

$$\text{aka } \sum_{k=1}^m x_{ik} = 1 \quad 1 \leq i \leq n, \quad (2.35)$$

$$x_{ij} \in \{0, 1\} \quad 1 \leq i, j \leq n. \quad (2.36)$$

Diğer uygulamalar Simeone (1986a, 1986b) ve Bullnheimer (1998)'de bulunabilir. Freeman ve diğ. (1966), Magirou ve Milis (1989), Carraresi ve Malucelli (1994) ve Billionnet ve Elloumi (2001)'de çokterimli sezgiseller ve alt sınırlar için referanslar bulunmaktadır. Samra ve diğ. (2005) faz kaydırmalı kiplenim (phase-shift keying) ve dördün genlik kiplenimi (quadrature amplitude modulation) gibi ikili olmayan doğrusal kiplenimler üzerinde çalıştıkları sistemlerde, çoklu paket göndermede farklılığı çoğaltma ve açıklamanın basit fakat etkili bir yöntemini sundular.

2.2.2.5. Çok-amaçlı KAP (MOQAP-ÇAKAP)

Knowles ve Corne (2002) birkaç akış ve mesafe matrisini dikkate alan başka KAP varyasyonunu sundular. Bu problem çok-amaçlı metasezgiseller ve çok-amaçlı gelişimsel algoritmalar için bir kıyaslama vakasıdır. Yazarlara göre bu model, hastanelerdeki tesislerin doktorlar ve hastalar arasında ve benzer şekilde hemşireler ve tıbbi ekipman arasında mesafe sonuçlarının akışlarını minimize etmek amacıyla tahsisi gibi bazı yerleşim düzeni problemleri için uygundur. Matematiksel ifade şöyledir:

$$C^k(\pi) = \sum_{i,j=1}^n f_{ij}^k d_{\pi(i)\pi(j)}, \quad 1 \leq k \leq m \text{ olduğunda}$$

$$\min_{\pi \in S_n} \bar{C}(\pi) = \{C^1(\pi), C^2(\pi), \dots, C^m(\pi)\} \quad (2.37)$$

Son kısıtda, f_{ij}^k i ve j tesisleri arasındaki k. akışı ifade etmektedir. Daha sonraları Knowles ve Corne (2003) KAP'ın çok-amaçlı versiyonu için vaka üreticiler sundular. Lopez-Ibanez ve diğ. (2004) Karınca Kolonileri Optimizasyonu (KKO) algoritmalarının tasarımını tartıştılar. Paquete ve Stützle (2004) farklı derecelerdeki korelasyona sahip akış matrisleri ile iki amaçlı KAP için bir stokastik yerel arama algoritma çalışması geliştirdiler. Kleman ve diğ. (2004) bir paralel teknikle analiz ettiler ve Day ve Lamont (2005) özel bir algoritma sundular.

Smith ve Li (2001) tarafından KAP'ın kuyruk modelli bir kombinasyonu trafik şebekelerinin bağlamında tartışıldı.

2.3. Alt Sınırlar

Alt sınırlar çalışması matematiksel programlama ve birleşimsel optimizasyon problemlerini çözmek üzere algoritmaların geliştirilmesi için çok önemlidir. Genel olarak kesin yöntemler optimumu garanti etmek için bir girişimde tam bir sayım işletirler. Bu yöntemlerin performansı hesapsal kalite ve alt sınırların verimine bağlıdır.

Alt sınırlar, dal-sınır yöntemleri için ve bazı sezgisel algoritmalarından elde edilmiş olan çözümlerin kalitesini geliştirmek için temel araçlardır. Bir alt sınırın kalitesi, alt sınır değeri ve optimal çözümü arasındaki fark ile ölçülebilir. İyi alt sınırların optimuma yakın olması gerekir. Alt sınırlar sadece çabukça çözülebilir oldukları zaman kesin çözüm yöntemleri içinde kullanılırlar. Sezgisel yöntemlerde kullanıldığında alt sınırların kalitesi hesaplama hızından daha önemlidir.

Gilmore (1962) ve Lawler (1963) tarafından sunulmuş olan KAP alt sınırı, en iyi bilinenlerden biridir. Bu alt sınırın önemi basitliği ve düşük hesapsal maliyetinden gelmektedir. Daha büyük örnekler için onu bir zayıf sınır yapan özelliği, problemin boyutu ile hızla büyümekte olmasından dolayı önemli bir dezavantaj göstermektedir. Bu dezavantajın ortaya çıkmasından sonra, yarı-kesin programlama (semi-definite programming), yenidenformülasyon-doğrusallaştırma (reformulation-linearization) ve kaldır-ve-yansıt (lift-and-project) yöntemleri üzerine temellenmiş olması, bu araştırmalar için umut verici olmuştur. Buna rağmen bu yöntemler genelde fazladan hesapsal çabaya ihtiyaç duymaktadırlar. Anstreicher ve Brixius (2001) kalite ve maliyet arasında iyi ilişki ile yarı-kesin ve dışbükey karesel programlama kullanarak yeni bir KAP sınırı bildirdiler. White (1994b) hesaplama yönünden kolay işlenir sınırlar için güncel verileri özel bir sınıf atama problemlerinin verilerine bağlantılayan bir veri dekompozisyon (decomposition) yöntemi kullandı.

Gilmore ve Lawler alt sınırı (GLB) aşağıdaki doğrusal atama probleminin çözümü yoluyla hesaplanır:

$$\min \sum_{i,j=1}^n (b_{ij} + l_{ij}) \cdot x_{ij} \quad (2.38)$$

$$\text{aka} \quad \sum_{i=1}^n x_{ij} = 1 \quad 1 \leq j \leq n; \quad (2.39)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad 1 \leq i \leq n; \quad (2.40)$$

$$x_{ij} \in \{0,1\} \quad 1 \leq i, j \leq n. \quad (2.41)$$

(2.38)-(2.41) çözmek amacı ile, aşağıdaki gibi l_{ij} katsayılarını bulmak gereklidir:

$$l_{ij} = \min \sum_{k,p=1}^n c_{ijkp} \cdot y_{ijkp} \quad k \neq i, p \neq j \quad (2.42)$$

$$\text{aka} \quad \sum_{k=1}^n y_{ijkp} = 1 \quad 1 \leq i, j, p \leq n, \quad (2.43)$$

$$\sum_{p=1}^n y_{ijkp} = 1 \quad 1 \leq i, j, k \leq n, \quad (2.44)$$

$$y_{ijkp} \in \{0,1\} \quad 1 \leq i, j, k, p \leq n. \quad (2.45)$$

Roucairol (1979, 1987), Edwards (1980), Frieze ve Yadegar (1983), Finke ve diğ. (1987), White (1994a), Burkard (1991), Brüngger ve diğ. (1997, 1998), ve Spiliopoulos ve Sofianopoulou (1998) GLB ve GLB'nin KAP çözmek için kullanılan algoritmalara uygulanması için geliştirme yöntemleri sunmaktadırlar.

2.3.1. Karışık tamsayılı doğrusal programlama (KTDP-MILP) gevşetmelerini esas alan sınırlar

Bir KTDP formülasyonu için optimal çözüm, ele alınan KAP ve aynı zamanda KAP için bir alt sınır olan doğrusal programlamanın her dual çözümü için daha düşük bir sınırdır. Frieze ve Yadegar (1983), Assad ve Xu (1985), Adams ve Johnson (1994), Ramachandran ve Pekny (1998) ve Karisch ve diğ.(1999)nde olduğu gibi bazı araştırmacılar bu prensibi kullandılar. Lagrangean gevşetmesi KAP'a da uygulanmıştır (Michelon ve Maculan, 1991). Drezner (1995) doğrusal programlama gevşetmesinin GLB sınırından daha iyi veya eşit olduğunu kanıtladı. Adams ve diğ. (2007), Hahn ve diğ. (2001b)nden dolayı bir seviye-2 yeniden-formülasyon doğrusallaştırma tekniği (2-YFDT) kullanarak sınırları hesapladılar. YFDT, karmaşık

0-1 doğrusal ve polinomsal programlardan elde edilen çözümlerin dışbükey kabuğunun sıkı polihedral dışsal-yakınsamalarla, daha yüksek değişken aralıklarında yeniden formüle etmek için bir hareket tarzı olan genel bir kuramdır (Adams ve Sherali, 1986, 1990). Sherali ve Adams (1999a, 1999b)da KAP için bir ve iki seviyeli yapıların her ikisi de genel metodolojinin bir açıklaması olarak sunuldu.

2.3.2. GLB yeniden-formülasyonları esas alan sınırlar

Bu sınırlar Frieze ve Yadegar (1983), Assad ve Xu (1985), Carraresi ve Malucelli (1992, 1994) ve Adams ve Johnson (1994)ın da içinde bulunduğu birkaç yazar tarafından uyarlanmıştı. Bir dual formülasyon üzerine temellenmiş sınır, Hahn ve Grant (1998) ve Hahn ve diğ. (1998)nde önerildi. Assad ve Xu (1985) tarafından ve Carraresi ve Malucelli (1992, 1994) tarafından verilen sınırların daha az hesaplama süresi gerektirmeleri avantajı ile kalitenin benzerliği yönünden Frieze ve Yadegar (1983) tarafından elde edilenler karşılaştırılabilir. Ancak yakınsamasına dair kuramsal ispat yoktur. Hahn ve Grant (1998)deki hesapsal sonuçlar, kalite aralıklarında, en iyi sınırların bazılarıyla karşılaştırıldığında bu sınırların rekabet edebilir olduğunu göstermiştir ve hala hesaplama süresinde daha iyi olduğunu söylemek mümkündür. Sergeev (2004) Adams-Johnson tekniğinin bir sürekli gevşetmesi ile çalışmıştır.

2.3.3. İç noktalar yöntemini esas alan sınırlar

Resende ve diğ. (1995), Drezner (1995) kuramını kullandı ve bir iç noktalar algoritması (Karmarkar ve Ramakrishnan, 1991) kullanarak bir KTDP doğrusal gevşetmesi çözdü. Bu yöntem, Adams ve Johnson (1994) tarafından elde edilenden daha kaliteli alt sınırlar verdi. Yine de bu sınırlar çok hesapsal çaba gerektiriyordu ve dal-sınır algoritmaları için önerilmiyorlardı. Bu durumda, Hahn ve Grant (1998)'ın dual ilerleme alt sınırını kullanmak daha iyi olabilir.

2.3.4. Değişkenlik indirgeme sınırları

İlk olarak Li ve diğ. (1994a) tarafından önerildi, bu sınırlar indirgeme planları üzerine temellenmektedir ve akış ve mesafe matrislerinin değişkenliğinden tanımlanmıştır. Bu sınırlar bir dal-sınır algoritmasında kullanıldığında daha az hesapsal süre almaktadır ve genellikle GLB'den daha iyi performans elde etmektedir. Akış ve mesafe matrisleri yüksek değişkenliğe sahip olduğunda daha etkililik gösterirler.

2.3.5. Çizge formülasyonunu esas alan sınırlar

Önceki bölümlerde ifade edildiği gibi, söz konusu herhangi bir KAP örneği ile ilişkili olan iki $n \times n$ matrisi F ve D , iki ağırlıklandırılmış tam çizge K_F ve K_D 'nin komşuluk matrisleri olarak ele alınsın. $\pi \in S_n$ K_F ve K_D arasındaki bir izomorfizmayı ifade etmektedir. Z_π 'nin minimum olduğu bir $\pi \in S_n$ izomorfizması bulmak, söz konusu KAP örneğinin çözülmesi anlamına gelmektedir. Gavett ve Plyter (1966) ve Christofides ve Gerrard (1981) bu kavramı, bir DAP gevşetmesi yoluyla daha düşük sınır bulmak üzere izomorfik yayılan alt çizgeler içinde K_F ve K_D 'yi çözerken kullanmışlardır.

2.3.6. Spektral sınırlar

Bu kapsamda matris özdeğerlerinin hesaplamasında kullanmak üzere iz formülasyonundan elde edilen sınırları ele alınmaktadır. Önceleri sonuçların kalitesi hesaplamaların hesapsal zorluğunu dengelemekteydi; ancak son zamanlarda bu sınırların bazılarının yerini yenidenformülasyon-doğrusallaştırma ve YKP sınırları almıştır. Spektral sınırlar hakkında bazı referanslar: Finke ve diğ. (1987), Rendl (1985), Hadley ve diğ. (1990, 1992a, 1992b), Rendl ve Wolkowicz (1992) ve Karisch ve diğ. (1994).

2.3.7. Yarı-kesin programlama ve yenidenformülasyon-doğrusallaştırma sınırları

Bu eğilim KAP'ın doğrusal programlama temsillerini elde etmek için birçok kuramsal araç kullanmaktadır. Zhao ve diğ. (1998) yarı-kesin programlama (YKP) gevşetmelerine çalışmışlardır. Anstreicher (2001) YKP gevşetmelerini özdeğer sınırları ile karşılaştırmaktadır; Anstreicher ve Brixius (2001) bir temel özdeğer sınırının YKP temsilini önermektedir. Burer ve Vandebussche (2004), Lova'sz ve Schrijver (1991)'in çalışmalarından aldıkları fikirlerle, çok sıkı YKP sınırı elde etmek için bir kaldır-ve-yansıt (lift-and-project) KAP gevşetmesi üzerinde Lagrangean gevşetmesi uygulamıştır.

2.4. Çözüm Yöntemleri

Birleşimsel optimizasyon problemlerinde faydalanılan yöntemleri kesin, sezgisel veya metasezgisel olarak sınıflandırabiliriz. Kesin yöntemlerden en sık kullanılan stratejiler dal-sınır veya dinamik programlama genel metodlarıdır. Diğer yandan farklı kavramları kullanan çok sayıda sezgisel ve metasezgisel teknikler vardır.

2.4.1. Kesin algoritmalar

KAP için global optimumu başarmakta faydalanılan farklı kesin yöntemler vardır. Bu yöntemler dal-sınır, kesme düzlemi veya dal-kesme gibi bu yöntemlerin kombinasyonları ve dinamik programlama içermektedir. Dal-sınır algoritmaları en bilinen ve en çok kullanılan algoritmalarıdır ve problem için alt sınırların tanımlandığı bölüştürme ve kesme kurallarıyla tanımlanır. İstenmeyen çözümleri elemek için alt sınırlar kullanan ilk sıralamalı tasarımlar şunlardır: Gilmore (1962), Land (1963) ve Lawler (1963). KAP dal-sınır algoritmaları içeren bazı referanslar: Gavett ve Plyter (1966), Nugent ve diğ. (1968), Graves ve Whinston (1970), Pierce ve Crowston (1971), Burkard ve Stratman (1978), Bazaraa ve Elshafei (1979), Mirchandani ve Obata (1979), Roucairol (1979), Burkard ve Derigs (1980), Edwards (1980), Bazaraa ve Kirca (1983), Kaku ve Thompson (1986), Pardalos ve Crouse (1989), Burkard (1991), Laursen (1993), Mans ve diğ. (1995), Bozer ve Suk-Chul

(1996), Pardalos ve diğ. (1997), Brüngger ve diğ. (1998), Ball ve diğ. (1998), Spiliopoulos ve Sofianopoulou (1998), Brixius ve Anstreicher (2001) ve Hahn ve diğ. (2001a, 2001b). Son yıllarda paralel uyarlamalar ile dal-sınır tekniklerini birleştiren prosedürler geniş olarak kullanılır olmuştur. Bunlar sayesinde KAP için en iyi sonuçlar başarılmaktadır. Ancak aynı zamanda büyük boyutlu örnekler için başarıyı elde etmek de önemlidir ve bu başarı donanımsal teknolojik gelişmelerle de ilişkilidir: Roucairol (1987), Pardalos ve Crouse (1989), Mautor ve Roucairol (1994a), Brüngger ve diğ. (1997), Clausen ve Perregaard (1997).

Dinamik programlama, akış matrisi bir ağacın komşuluk matrisi olarak düşünüldüğünde KAP özel durumları için kullanılabilir. Christofides ve Benavent (1989) gevşetilmiş probleme KTDP yaklaşımı kullanmışlardır. Bu problem daha sonra bir dinamik programlama algoritması ile çözüldü. Bu teknik Urban (1998) tarafından da kullanıldı.

Kesme düzlemi yöntemleri ilk olarak Bazaraa ve Sherali (1980) tarafından ortaya atıldı ve memnuniyet verici sonuçlar sunmadı. Yine de, KTDP ve Bender'in dekompozisyonunu kullanan bazı sezgisellerin formülasyonu bu yöneme katkıda bulunmuşlardır. Şimdiye kadar bu teknik geniş olarak kullanılmamıştır. Ancak KAP vakaları için iyi kalitede çözümler sunmaktadır. Bu yöntemin yavaş yakınsaması onu sadece küçük örnekler için uygun yapmaktadır (Kaufman ve Broeckx (1978), Bazaraa ve Sherali (1980, 1982) ve Burkard ve Bonniger (1983)). Son zamanlarda Miranda ve diğ. (2005), formülasyonunda doğrusal maliyetleri de barındıran bir anakart tasarım problemi ile uğraşırken Bender'in dekompozisyon algoritmasını kullanmaktadır.

Dal-kesme tekniğinin Padberg ve Rinaldi (1991) tarafından önerilen varyasyonu, problemin uygun çözümleri tarafından tanımlanmış politopu açıklayan bir alternatif kesme stratejisi olarak görülmektedir. Bunun asıl avantajı kesme düzlemleri üzerinden politopun yüzleri ile kesmeleri ilişkilendirmesidir. Yüzlerle ilişkili kesmeler, kesme düzlemleri tarafından üretilenlerden daha etkilidir, bundan dolayı optimal bir çözüme yakınsama hızlandırılmaktadır. Polihedral kesme düzlemlerinin bu problemler için yaygın olarak kullanılmamasından dolayı KAP politopu hakkında

bilgi kıtlığı vardır. Şu makalelerde, bazı araştırmacılar ilerideki algoritma gelişmeleri ile katkıda bulunabilecek politopun temel özelliklerini açıklamıştır: Jünger ve Kaibel (2000, 2001a, 2001b), Padberg ve Rijal (1996), Kaibel (1998) ve Blanchard ve diğ. (2003).

2.4.2. Sezgisel algoritmalar

Sezgisel algoritmalar ile elde edilen en iyi çözümlerin optimalliğinin garantisi yoktur. Yaklaşık yöntemler bu sınıfta yer alır ve en kötü durum çözümlerine ek olma niteliğine sahiptirler. Birleşimsel Optimizasyon literatüründe sezgisel algoritmalar diye adlandırılan Osman ve Laporte (1996)'daki gibi yakınsama algoritmaları az sayıda değildir. Sezgisel prosedürler şu şekilde sınıflandırılabilir: Yapısal, sınır sıralama ve geliştirme yöntemleri. Metasezgiseller olarak adlandırılan yöntemler sezgisel algoritmalarından sonra ortaya atılmış ve en son teknikler olarak sonraki bölümlerde incelenmektedir.

2.4.2.1. Yapısal yöntemler

Gilmore (1962) algoritmanın her iterasyonunda bir permütasyon (uygun çözüm) tamamlayan bir yapısal yöntem sundu. Bu yöntemde ilki bölüştürülen tesislerle ilgilenen ve ikincisi atanmış yerleşimler olan A ve L kümeleri ele alınmaktadır. Bir permütasyon π 'nin inşası bir sezgiselin ortalamaları tarafından yapılmaktadır ve her adımda yeni bir (i, j) bölümü seçilmektedir. Bundan dolayı $i \in A$, $j \in L$ ve $\pi(i) = j$ olmaktadır. n boyutlu bir örnek için proses, bir tam permütasyon problem sırası üzerinden başarılıncaya kadar tekrarlanmaktadır. Yapısal yöntemler Armour ve Buffa (1963), Buffa ve diğ. (1964), Sarker ve diğ. (1995, 1998), Tansel ve Bilen (1998), Burkard (1991), Arkin ve diğ. (2001), Gutin ve Yeo (2002) ve Yu ve Sarker (2003)'de kullanıldı. 1990'ların sonunda, çoklu başlama teknikleri, sezgisel veya metasezgisel yöntemlerde de kullanılmaya başlandı. Bu kategoride Misevicius (1997), Fleurent ve Glover (1999) ve Misevicius ve Riskus (1999) çalışmaları kaynak gösterilebilir.

2.4.2.2. Sıralamalı yöntemler

Sıralama, sıralamalı prosesin sonuna gidilebilirse ancak elde edilen çözümün optimum olduğunu garanti edebilir. Bunun dışında iyi bir çözüm veya sıralamanın başlangıcında bulunan bir optimal çözüm dahi elde etmek mümkündür. Genellikle optimalliği garanti etmek çok daha uzun zaman alır. Bu sayının çalıştırma zamanı sınırlandırılırsa şu gibi durdurma koşulları belirlenmektedir: tam uygulama için maksimum sayıda döngüler veya birbirini izleyen gelişmeler arasındaki uygulama süresi için bir limit ve buna benzer olabilir. Bu durdurma kriterlerinden herhangi birinin optimum çözümü bertaraf edebileceği açıkça görülmektedir; bir gerçek de şudur: sınırlandırılmış sıralama yöntemleri, kullanılırken biraz daha dikkat gerektirmektedir (Burkard ve Bonniger, 1983; West, 1983). Bundan dolayı Nissen ve Paul (1995) KAP'a eşik kabul tekniği uygulamıştır.

2.4.2.3. Geliştirme yöntemleri

Geliştirme yöntemleri yerel arama algoritmalarına karşılık gelir. KAP sezgisellerinin çoğu bu kategoridedir. Bir geliştirme yöntemi uygun bir çözümle başlar ve onu geliştirmeye çalışır, komşuluğundaki diğer çözümleri aramak yoluyla onu geliştirmeye çaba göstermektedir. Bu süreç geliştirme bulunamayınca kadar tekrar edilmektedir. Bu yöntem için temel elemanlar komşuluk ve analiz edilen komşuların sırasını tanımlayan seçim kriteridir: Heider (1973), Mirchandani ve Obata (1979), Bruijs (1984), Pardalos ve diğ. (1993), Burkard ve Çela (1995), Li ve Smith (1995), Anderson (1996), Talbi ve diğ. (1998a), Deineko ve Woeginger (2000), Misevicius (2000a), Mills ve diğ. (2003). Geliştirme yöntemleri metasezgisellerde sıklıkla kullanılmaktadır.

2.4.3. Metasezgiseller

1980'lerin sonlarına doğru, birleşimsel optimizasyon problemleri için önerilen sezgisel yöntemlerin çoğu özeldi ve belli bir probleme ithaf edilmişti. O dönemden sonra, bu model değişti. Metasezgiseller olarak bilinen daha genel teknikler görülmeye başlanmıştır. Bunlar problem yapısına adapte edilmiş önsel (a priori)

stratejilerin tanımlanması yoluyla niteliklendirilmektedirler. Bu yöntemlerin birkaçı bilginin farklı bir alanı olan metaforların çalışılmış bir doğal prosesin simülasyonunun bazı biçimleri üzerine temellenmiştir. Metasezgisellerin gelişi ile KAP araştırması yeni ve artan bir ilgiye sahip oldu (Moe, 2003).

2.4.3.1. Doğal süreç metaforlarını esas alan metasezgiseller

Tavlama benzetimi, birleşimsel optimizasyon algoritmaları ve istatistiksel mekanik arasındaki benzerliği açıklayan bir yerel arama algoritmasıdır (Kirkpatrick ve diğ., 1983). E_i ve E_{i+1} iki komşu çözümüne karşılık gelen iki enerji ardışık ifadeleri ve $\Delta E = E_{i+1} - E_i$ olsun. Şu durumlar meydana gelebilir: Eğer $\Delta E < 0$ ise, bir enerji azalması vardır ve proses devam eder, diğer bir deyişle problem maliyet fonksiyonunda bir azalma vardır ve yeni bölüştürme kabul edilebilir; eğer $\Delta E = 0$ ise, bir kararlılık durumu vardır ve enerji derecesinde değişme yoktur, bu, problem maliyet fonksiyonu değişmedi anlamına gelir; eğer $\Delta E > 0$ ise enerji üzerinde bir artış gözlemlenmiştir ve parçacık yerleştirmeye izin vermek için fiziksel proses kullanışlıdır, problem maliyet fonksiyonu artmaktadır. Zayıf yerel minimum içinde yakınsamayı önlemek için bu paylaşımı bertaraf etmek yerine, bir olasılık fonksiyonunun altında kullanımı önerilmektedir. Burkard ve Rendl (1984) KAP'a tavlama benzetiminin ilk uygulamalarından birini önerdi. Ondan sonra, Wilhelm ve Ward (1987) onun için yeni denge elemanlarını sundu. Connolly (1990) değerli sonuçlar veren bir optimal sıcaklık kavramı ortaya koydu. Daha sonra Abreu ve diğ. (1999) maliyet azaltma ile birlikte problem çözümüne eşlik eden ters çevirmelerin sayısını azaltmaya çalışmak yoluyla yeni bir tekniği uyguladı. KAP'a uygulanan tavlama benzetimi için diğer yaklaşımlar Bos (1993), Yip ve Pao (1994), Burkard ve Çela (1995), Peng ve diğ. (1996), Tian ve diğ. (1996, 1999), Mavridou ve Pardalos (1997), Chiang ve Chiang (1998), Misevicius (2000b, 2003c), Tsuchiya ve diğ. (2001), Siu ve Chang (2002) ve Baykasoglu (2004)dür.

Genetik Algoritmalar doğada bulunan doğal seleksiyon ve adaptasyonu simüle eden tekniklerdir. Bu algoritmalar uygun gelen bireylerin bir alt kümesi sayesinde şekillendirilen bir popülasyonu, KAP vakasında uygun permütasyonları, onların permütasyon maliyeti ile ilişkilendirilmiş uygunluk değerleri ile saklarlar. Genetik

operatörlerin ve seçim kriterlerinin hesabı ile algoritma bir popülasyonunun en iyi elde edilebilir uygunluk değeri olan başka biriyle yer değiştirmektedir. Plan doğada üreyen biyolojik yaratıklarda olduğu gibi benzer bir şekilde en iyi bireylerin hayatta kalması ve onların genetik özelliklerini taşıyan nesillerin üretilmesini esas almıştır.

Genetik algoritmalar genel olarak tesadüf sel üretilmiş başlangıç bireylerinin (örnek olarak, çözümlerin) bir popülasyonu ile başlamaktadır. Maliyetleri hesaplanıp değerlendirilmektedir ve en düşük maliyetli çözümler ile bir alt küme seçilmektedir. Genetik operasyonlar bu alt kümeye uygulanmaktadır, bu yolla yeni bir çözüm kümesi (yeni bir popülasyon) üretilmektedir. Proses bazı durma kriterleri ile karşılaşınca kadar devam eder (Davis, 1987, Goldberg, 1989). KAP üzerine genetik algoritmaların kullanımı için çeşitli fikirler, Bui ve Moon (1994), Tate ve Smith (1995), Mavridou ve Pardalos (1997), Kochhar ve diğ. (1998), Tavakkoli-Moghaddain ve Shayan (1998), Gong ve diğ. (1999), Drezner ve Marcoulides (2003), El-Baz (2004), ve Wang ve Okazaki (2005)'de bulunmaktadır. Drezner (2005a) bir iki-fazlı genetik algoritma sunmuştur. İlk fazı faz2 için bir başlama popülasyonu inşa etmek için her iterasyonda en iyi çözümü koruyacak birkaç kez koşulan hızlı bir genetik stratejidir. Bu algoritmaların KAP kullanımı ufak örnekler için bile optimal çözümü elde etmede bazı zorluklarla karşılaşmaktadır. Bunun yerine genetik algoritmalar kullanan bazı melez fikirlerin daha etkili olduğu görülmüştür.

Dağıtık arama, tamsayılı doğrusal programlama problemlerinin bir sezgisel çalışma içinde kullanıldığı, Glover (1977) tarafından ortaya atılmış bir tekniktir. Ard arda üretimlerde yeni çözüm vektörleri üretmek amacı ile çözüm vektörlerinin doğrusal kombinasyonlarını alan bir evrimsel yöntemdir. Bu metasezgisel başlangıç ve geliştirme fazlarının birleştirilmesinden oluşur. Başlangıç fazında, iyi çözümlerin bir referans kümesi oluşturulur. Her yeni çözüm alt sıralama tüm stratejik olarak seçilen kombinasyonları kullanılarak faz içinde üretilir. Daha sonra en iyi yeni olarak üretilmiş çözümlerin bir kümesi referans kümesi içine hareket ettirilir. Evrimsel aşama prosedürü bir durma kriteri tatmin edilinceye kadar tekrar etmektedir. KAP'a dağıtık aramanın uygulaması Cung ve diğ. (1997)'de bulunabilir.

Karınca Koloni Optimizasyonu (KKO), en önemli göze çarpan özelliği olarak düşünülen birkaç basit ajanın (karıncalar) etkileşimi içinde özelliklerin tanımlanması yoluyla çalışır ve dağıtık algoritmaların bir sınıfını temsil eder. Temel prensibi karıncaların bir yiyecek kaynağına koloniden yol bulabilmek için uğraşmasıdır. Karınca sistemini oluşturan karıncaların kümesi bir problemi çözmek için sıra içinde işbirliği yaparlar. Bu yöntemin ana özelliği bu ajanların ürettikleri bir sinerjik etkinin etkileşimidir, çünkü bu ajanlar kendi aralarında etkileşerek birlikte çalıştıkları zaman elde edilen çözümün kalitesini artırmaktadırlar. KAP için sayısal sonuçlar Maniezzo ve Colorni (1995, 1999), Colorni ve diğ. (1996) ve Dorigo ve diğ. (1996)nde sunulmuştur. Gambardella ve diğ. (1999) karınca kolonisini rekabet edebilir bir metasezgisel olarak göstermektedir. Diğer referanslar Stützle ve Dorigo (1999), Stützle ve Hoos (2000), Talbi ve diğ. (2001), Middendorf ve diğ. (2002), Solimanpur ve diğ. (2004), Randall (2004) ve Ying ve Liao (2004)'dedir. Ortalama üstü kalitede bireylerden (örneğin, karıncalar) çözüm parçalarının kısmi olarak tam sıralamalarının kullanımı üzerine temellendirilmiş, alışılmamış bir harici hafıza uyarlaması ortaya konmaktadır. Bu gibi değişken-boyutlu kısmi permütasyon dizilerinin elemanları temel bireylerin tesadüf sel seçilen yerleşimlerinden alınmaktadır ve kısmi permütasyon hafızası olarak adlandırılan harici bir hafızada yerleştirilmektedir (Acan, 2005).

Sinir ağları ve Markov zincirleri metasezgisellerden yapısal olarak farklı olmalarına karşın, onlar da bir doğal metaforu esas almaktadır ve KAP'a uygulanmışlardır, Bousonocalzon ve Manning (1995), Liang (1996), Obuchi ve diğ. (1996), Tsuchiya ve diğ. (1996), Ishii ve Sato (1998, 2001), Rossin ve diğ. (1999), Shin ve Niitsuma (2000), Nishiyama ve diğ. (2001), Hasegawa ve diğ. (2002) ve Uwate ve diğ. (2004).

2.4.3.2. Doğrudan kuramsal ve deneysel düşünceleri esas alan metasezgiseller

Tabu arama tamsayı programlama problemleri için iyi kalitede çözümler bulmak üzere Glover (1989a, 1989b) tarafından sunulan bir yerel arama algoritmasıdır. Temel özelliği araştırma prosesi içinde bulunan en iyi çözümlerin güncelleştirilen listesinin olmasıdır. Her çözüm bir öncelik değeri veya bir beklenti kriteri alır. Ana içerikleri: araştırma prosesinin geliştirilmesini tutan bir tabu listesi, tabu listesi

bilgisini ve onun önceliklerini esas alan komşuluktaki yeni bir paylaşırmanın kabulü veya reddini sağlayan bir mekanizma, komşu çeşitlendirme ve yoğunlaştırma stratejileri arasında sıralı değişimi kabul eden bir mekanizma. KAP için uyarlamalar Skorin-Kapov (1990, 1994), Taillard (1991), Bland ve Dawson (1991), Rogger ve diğ. (1992), Chakrapani ve Skorin-Kapov (1993), Misevicius (2003a, 2005) ve Drezner (2005b)'de bulunabilir. Tabu listesinin boyutuna ve bu listenin yönetim yoluna bağlı güçlüklerle rağmen Taillard (1991) ve Battiti ve Tecchiolli (1994)'de analiz edildiği gibi bu algoritmaların performansları KAP için çok etkili stratejiler olduklarını göstermektedir. Taillard (1995) tabu araştırma ve genetik algoritma kullanarak KAP'a uygulandığında aralarında bir karşılaştırma sunmaktadır.

Hırslı tesadüf sel adaptif arama prosedürü (GRASP-HTAAP) her adımda problem için bir yaklaşık çözüm elde edildiği noktada bir tesadüf sel ve iteratif tekniktir. Son çözüm tüm iterasyonlara göre en iyi sonuç veren çözümdür. Her adımda, ilk çözüm bir tesadüf sel hırslı fonksiyon doğrultusunda inşa edilmektedir ve takip eden çözümler bir önceki çözüme ilişkin yeni bir en iyi çözüm veren (önceki çözüm üzerine uygulanan) bir yerel arama algoritması sayesinde elde edilir. Tüm iterasyonların sonunda sonuç çözüm en iyi üretilmiş olanıdır. HTAAP çözümlerinin bir yerel optimuma saptanmaması garanti edilemez, bu nedenle onları geliştirmek için genel arama aşamasına uygulamak önemlidir. Uygun veri yapıları ve dikkatli uyarlamalar kullanmak etkili bir yerel arama olarak kabul edilecektir. Bu yöntem KAP'a bazı araştırmacılar tarafından uygulandı: Li ve diğ. (1994b), Feo ve Resende (1995), Resende ve diğ. (1996), Fleurent ve Glover (1999), Ahuja ve diğ. (2000) ve Pitsoulis ve diğ. (2001). Oliveira ve diğ. (2004) yol yeniden linkleme stratejisi kullanarak iyi çözüm çiftlerine katılan yollar boyunca geliştirmeler için araştıran bir HTAAP inşa etti. HTAAP aynı zamanda bir KAP varyasyonuna uygulandı: BiKAP, Mavridou ve diğ. (1998).

Değişken komşuluk arama (VNS-DKA) Mladenovic ve Hansen (1997) tarafından sunuldu. Uygun bir şekilde tanımlanmış komşulukların bir kümesi içinde yapılan sistematik hareketleri esas almaktadır. Mevcut komşuluğun daha iyi bir çözüm üretmediği görüldüğünde bir değişim uygulanmaktadır. DKA bu çalışmada özellikle büyük birleşimsel problem örneklerine uygulanmıştır. Taillard ve Gambardella

(1999)da, KAP için üç DKA stratejisi önermiştir. Onlardan biri, temel paradigmaya göre değişken komşuluğu üzerinde bir araştırmadır. Diğer ikisi önceden açıklanmış yöntemlerin bazılarının kombinasyonu ile ortaya çıkan melezlerdir.

Stützle (2006) durgunluğu önlemek için algoritmanın çözümlerini genişleten daha zayıf yerel optimayı kaldıran kabul kriteri kullanarak basit ve güçlü bir stokastik yerel arama olan bir tekrarlı yerel arama (iterated local search-ILS-TYA) uyguladı. Makale popülasyon tabanlı TYA genişletmelerini önermektedir.

KAP için bazı diğer melez algoritmalar da mevcuttur. Bölte ve Thonemann (1996) tavlama benzetimi ve genetik algoritmanın bir birleşimini sunmuşlardır. Battiti ve Tecchiolli (1994), Bland ve Dawson (1994), Chiang ve Chiang (1998) ve Misevicius (2001, 2004a) tavlama benzetimi ile tabu aramayı kullanmaktadırlar. Talbi ve diğ. (1998b) ve Hasegawa ve diğ. (2002) bir sinir ağı ile tabu aramayı kullanıyorken, Youssef ve diğ. (2003) tabu arama, tavlama benzetimi ve bulanık mantığı birlikte kullanmaktadırlar. Fleurent ve Ferland (1994), Drezner (2003)deki gibi bazı melez algoritmalar tabu arama ile genetik algoritmayı bazılarında ise Ahuja ve diğ. (2000)de olduğu gibi genetik algoritmayı bir hırslı algoritma ile birleştirmektedirler. Bütün bu bahsedilen yöntemlerin genetik algoritmanın yalnız kullanımından daha başarılı sonuçlar verdiği kanıtlandı.

Daha sonraları, bu sınıfta Lim ve diğ. (2000, 2002) gibi yerel arama k-gen değişimini esas alan melez genetik algoritma ile çalışan daha karmaşık prosedürler ortaya çıkmıştır ve Misevicius (2004b) bir geliştirilmiş genetik prosedür kullanarak, karesel atama problemi için yeni sonuçlar ortaya koymuştur. Dunker ve diğ. (2004) bir dinamik tesis yerleşim problemini çözmek için evrimsel hesaplama ile dinamik programlamayı birleştirdi. Balakrishnan ve diğ. (2003) tabu arama ile genetiği birleştiren GATS'ı kullandı.

Melez genetik algoritmaların bazıları memetik algoritmalar veya evrimsel algoritmalar olarak bilinmektedirler ve bu bağlamda şu çalışmalarını gösterebiliriz: Brown ve Huntley (1991), Nissen (1994), Huntley ve Brown (1996), Merz ve Freisleben (1997, 1999, 2000), Nissen (1997), Ostrowski ve Ruoppila (1997).

KAP üzerine bazı metasezgisellerin ve melez önerilerin faydası Maniezzo ve Colomi (1995) ve Taillard ve diğ. (2001)de tartışıldı, sonuçları karşılaştırıldı ve analiz edildi. Kelly ve diğ. (1994) KAP için çeşitlendirme stratejileri çalıştı. Fedijiki ve Duffua (2004) KAP'ı çözmek üzere bir araştırma algoritmasındaki uç noktaları kullanarak bir çalışma geliştirdi. Son olarak, paralel ve büyük hesaplama kullanan bazı teknikler şunlardır: Roucairol (1987), Pardalos ve Crouse (1989), Brown ve Huntley (1991, 1996), Taillard (1991), Chakrapani ve Skorin-Kapov (1993), Laursen (1993), Mans ve diğ. (1995), Obuchi ve diğ. (1996), Brüngger ve diğ. (1997, 1998), Clausen ve diğ. (1998), Talbi ve diğ. (1998a, 1998b, 2001), Anstreicher ve diğ. (2002) ve Moe (2003).

2.5. Ana Araştırma Eğilimleri ve Yönsemeler

KAP'ın 70'lerin ortasına kadar az ilgiye sahip olduğu görülmektedir. 80'lerde birkaç kuramsal gelişme meydana gelmiştir, 80'lerin sonunda metasezgisellerin çıkması ile, problem daha dikkat çekmiştir. Bu dikkat çekicilik, kısmen bir kıyaslama olarak düşünülmekten ileri gelmektedir. Metasezgisel yöntemleri birbirleriyle rekabetçi olarak düşündüğümüzde ve KAP'a uygulandığında, hali hazırda bilinen sonuçlardan daha iyileri başarabilirdi. 90'ların sonlarında artan araştırmalar hem donanım ve hem de yazılımda daha büyük geçici bellekler ve elde edilebilen kapasite yönetim alternatifleri sayesinde bilgisayar teknolojisinin gelişmesinden kazanıldı. Paralel hesaplama (paralel computing) ve metahesaplama (metacomputing) gibi kavramlar gelişti. Bu, yapılabilen kesin yöntemler ile birleştirildiğinde, büyük örnekler (n=30dan büyük) için optimal çözümleri bulmak ve aynı zamanda bazı daha büyük örnekler için daha iyi çözümleri elde etmek de mümkün hale geldi, QAPLIB (2009). Son yıllarda algoritmalarındaki ilginin çok güçlü olarak devam ettiğini söyleyebiliriz. Kuramsal gelişmeler ise daha az bir derecede de olsa, uygulamalardan dolayı devam etmektedir. Son zamanlarda iletişim, link tasarımı ve haberleşme şebekelerinin optimizasyonuna uygulamalarda artan bir ilginin olması dikkate değerdir.

3. ÇOK-AMAÇLI KARINCA KOLONİLERİ OPTİMİZASYONU

Çok-kriterli optimizasyon problemleri, eş zamanlı olarak optimize edilmek zorunda olan, bundan dolayı özellikle problem çözmeyi zor hale getiren birkaç amaç vasıtasıyla karakterize edilmektedirler. Bu problemler için metasezgisellerin kullanımı son on yıl içerisinde artan bir ilgi görmektedir. Gerçek dünyadaki çok-amaçlı problemlerin çokluğu, bunların gerçekte karmaşıklığı ve metasezgisel prosedürlerinin avantajları son birkaç yılda bu araştırma alanını güçlendirmiştir (Gandibleux ve diğ., 2004, Jaszkiwicz, 2001).

KKO çeşitli karınca türlerinin en kısa yolu aramasından esinlenmiş olan bir metasezgiseldir. Dorigo ve diğ. (1996)nin ilk KKO algoritması (Karınca Sistemi) üzerine çalışmaya başlamalarından beri, çeşitli araştırmacılar, gezgin satıcı problemi, karesel atama problemi, sıralı atama problemi, üretim çizelgeleme, programlama, proje çizelgeleme, araç rotalama, uziletişim (telecommunication) rotalama, yatırım planlama, personel çizelgeleme, içinde olmak üzere bu gibi kombinasyonel problemleri tam anlamıyla çözmek üzere yapılmış farklı KKO algoritmaları geliştirmişlerdir (Cordon ve diğ., 2002, Dorigo ve Stützle, 2003).

Son zamanlarda, bazı araştırmacılar çok-amaçlı problemlerle uğraşmak için şu KKO (ÇAKKO) algoritmaları tasarlamışlardır: Mariano ve Morales (1999), Gambardella ve diğ. (1999), McMullen (2001), İredi ve diğ. (2001), Gravel ve diğ. (2002), T'kindt ve diğ. (2002), Barán ve Schaerer (2003), Cardoso ve diğ. (2003), Doerner ve diğ. (2003, 2004). Bunların çoğu çizelgeleme, araç rotalama, veya portföy seçiminin de içinde bulunduğu somut birer çok amaçlı problem olan problemleri çözmek için spesifik önerilerdir.

3.1 Karınca Koloni Optimizasyonu

Karıncalar koloniler halinde yaşayan sosyal böceklerdir ve birlikte çalışma etkileşimleri sebebiyle bu koloniler karmaşık davranışlar gösterme ve bir karıncanın yerel olarak zor görevleri yerine getirmesi yeteneğine sahiptirler. Birkaç karınca türünün davranışının çok ilginç bir yönü, karıncaların yuvaları ile yiyecek kaynakları arasındaki en kısa yolu bulma yetenekleridir. Bu gerçek, çoğu karınca türlerinde görsel bilginin avantajlı duruma geçmesini önlemek için karıncaların hemen hemen kör olduğunu düşündüğümüzde özellikle dikkat çekicidir.

Yuvaları ve yiyecek kaynakları arasında yürürken, bazı karınca türleri feromon olarak adlandırılan (güzel kokulu) bir kimyasal bırakırlar. Eğer feromon izleri yoksa karıncalar esas olarak tesadüf hareket ederler, fakat feromonun varlığında izi takip etme özelliğine sahiptirler. Pratikte farklı yolların arasından seçimler birkaç yol kesimlerinde meydana gelir. Daha sonra, karıncalar feromon miktarı, daha güçlü feromon izi, daha yüksek kalitesi ile etkilendirilmiş bir olasılıksal karar sonucu ortaya çıkan yolu seçerler. Çünkü çevrim içindeki karıncalar, takip ediyor oldukları yol üzerine feromon bırakırlar. Bu davranış, yüksek feromon yoğunluğu ile işaretlenmiş yolların, bir kendi kendini güçlendirme sürecine önderlik etmektedir. Bu davranış aynı zamanda karıncaların yuvaları ile yiyecek kaynakları arasındaki en kısa yolu belirlemelerini sağlamaktadır.

İkinci prosedür doğal çevre içinde bazen feromonun buharlaştırılması ile tamamlanır. Bu yol, her seferinde daha az karınca tarafından ziyaret edilmiş olma yoluyla feromon azalır.

KKO algoritmaları birleşimsel optimizasyon problemlerini çözmek için gerçek karınca kolonilerinin davranışlarından esinlenmişlerdir. Onlar, bir koloni yapay karınca üzerine temellendirilmiştir, öyle ki bu karıncalar birlikte çalışan ve yapay feromon izleri aracılığıyla iletişim kurabilen basit hesapsal ajanlardırlar (Dorigo ve Stützle, 2004).

KKO algoritmaları temel olarak yapısal algoritmalardır: Her algoritma iterasyonu içinde, her karınca bir yapılaşma grafiği üzerinde seyahat etmek yoluyla probleme bir çözüm inşa eder. Karıncanın yapabileceği mümkün adımları gösteren grafiğin her hattı, karınca hareketlerine kılavuzluk eden iki tür bilgi ile donatılmıştır:

- Sezgisel bilgi, i düğümünden j düğümüne hareket etmenin sezgisel önceliğidir. Örneğin, a_{ij} kenarına seyahat etmek η_{ij} ile gösterilmektedir. Bu bilgi, algoritma çalıştırılması sırasında karıncalar tarafından değiştirilmemektedir.
- (Yapay) feromon iz bilgisi, hareketin “öğrenilmiş çekiciliği”ni ölçmektedir ve doğal karıncaların biriktirdiği gerçek feromonu taklit eder. Bu bilgi algoritmanın çalışma esnasında karıncalar tarafından bulunan çözümlere bağlı olarak değiştirilir ve τ_{ij} ile gösterilir.

KKO metasezgiselinin içinde bulunduğu önemli bazı KKO algoritmaları (Dorigo ve Di Caro, 1999): Karınca Sistemi (Ant System) (Dorigo ve diğ., 1996), Karınca Kolonisi Sistemi (Dorigo ve Gambardella, 1997), Max-Min Karınca Sistemi (Stützle ve Hoos, 2000), Sıralama-Esaslı Karınca Sistemi (Bullnheimer ve diğ., 1999) ve En iyi-En kötü Karınca Sistemi (Cordon ve diğ. 2000, 2002a, 2002b). ÇAKKO literatürüne bakıldığında temelde bu algoritmalarından sadece Karınca Sistemi ve Karınca Kolonisi Sistemi kullanılmıştır.

3.1.1 Karınca sistemi

Dorigo ve diğ. (1996) tarafından geliştirilen Karınca Sistemi (KS), ilk KKO algoritmasıdır. KS’de, feromon güncelleme bir kez tetiklenir tüm karıncalar çözümlerini tamamlar ve şu şekilde gerçekleşir: İlk olarak tüm feromon izleri sabit bir faktör tarafından azaltılır, bu yolla feromon buharlaştırma gerçekleştirilir. İkinci olarak, koloninin her karıncası çözümünün kalitesinin bir fonksiyonu olan, yolundaki bir miktar feromonu biriktirir (yola bırakır). Başlangıçta, KS herhangi merkezleştirilmiş deamon faaliyetlerini (Karıncalar tarafından gerçekleştirilmeyen fakat bir dış etki tarafından gerçekleştirilen, herhangi doğal bir tamamlayıcılığı olmamış, fakat sadece metasezgisel performansını geliştirmek amacıyla eklenmiş

prosedürlerdir) kullanmamıştır. Fakat daha sonraları bunun yapılmasıyla daha iyi çözümler üretilebilmiştir.

$$p_{ij}^h = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{u \in N_i^h} [\tau_{iu}]^\alpha \cdot [\eta_{iu}]^\beta}, & \text{eğer } j \in N_h(i) \\ 0, & \text{aksi halde} \end{cases} \quad (3.1)$$

KS'deki çözümler şu şekilde oluşturulur. Her adımda, KS'deki bir karınca h , diğer bir düğüme gitmek için denklem (3.1)de görüldüğü şekilde hesaplanan bir olasılık ile hareket eder.

$N_h(i)$, i düğümünde yerleşik olan karınca h 'nin hareket etmesi ihtimal dahilinde olan komşusudur, ve $\alpha, \beta \in \mathbb{R}$ feromon iz ve sezgisel bilginin görelî önem ağırlığı olan iki parametredir. Her karınca h şimdiye kadar takip ettiği diziyi kaydeder ve bu hafıza her L_h inşa adımı içinde $N_h(i)$ belirlemek için kullanılır.

Belirtildiği üzere, feromon bırakmak, çözümlerini inşa etmeyi bitirmiş olan tüm karıncalar tarafından bir kez yapılır. İlk olarak, feromon iz her kenar için sabit bir faktör tarafından, $\rho \in (0, 1]$ buharlaşma oranı olduğunda, tüm feromonların azaltılması yoluyla şu formülle buharlaştırılır:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} \quad (3.2)$$

Daha sonra her karınca, takip ettiği yolu izleyerek geriye gider (bu, yerel hafıza L_h 'de depolanmıştır) ve bir miktar feromon $\Delta\tau_{ij}^h$ 'ı her içinden geçtiği bağlantı üzerine aşağıdaki formülle bırakır [çevrim içi deneyden sonra (a posteriori) güncelleme veya global güncelleme]:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^h, \quad \forall a_{ij} \in S_h, \quad (3.3)$$

Karınca h nin çözümü S_h 'nin kalitesi $C(S_h)$ 'a bağılı olarak $\Delta\tau_{ij}^h = f(C(S_h))$ kadar feromon serbest bırakılır.

KS'nin yaratıcılarının, aynı zamanda tipik olarak daha iyi performans gösteren, bu algoritmanın Elitist KS (Dorigo ve diğ., 1996) olarak adlandırılan genişletilmiş sürümünü önerdiklerine dikkat edilmelidir. Elitist KS'nde, bir kez karıncalar üretilmiş çözümlerine ilişkilendirilen bağlantılar üzerine feromon bırakmıştır, deamon, araştırma prosesi içindeki bulunan en iyi çözüme ait kenarlar üzerine ilave bir feromon birikimi yapar (Bu çözüm global en iyi çözüm olarak adlandırılır).

$$\tau_{ij} \leftarrow \tau_{ij} + e \cdot f \left(C \left(S_{global-eniyi} \right) \right), \quad \forall a_{ij} \in S_{global-eniyi} \quad (3.4)$$

Biriktirilmiş feromon miktarı, o global en iyi çözümün kalitesine bağlı olan, ele alınmış elitist karıncaların sayısı e tarafından denklem (3.4)teki gibi ağırlıklandırılmıştır.

3.1.2 Karınca koloni sistemi

Dorigo ve diğ. (1997) tarafından önerilmiş olan karınca koloni sistemi (KKS), KS'nin ilk halelerinden biridir. KS'ne üç büyük değişiklik getirir:

1. KKS *sözde-tesadüfsel orantılı kural* olarak adlandırılan farklı bir geçiş kuralı kullanır: h düğüm i 'de yerleşik bir karınca olsun, $q_0 \in [0, 1]$ olan bir parametre, ve $q [0, 1]$ aralığında bir tesadüfsel değer olsun.

Eğer $q \leq q_0$:

$$p_{ij}^h = \begin{cases} 1, & \text{if } j = \arg \max_{u \in N_h(i)} \{ \tau_{iu}^\alpha \cdot \eta_{iu}^\beta \} \\ 0, & \text{aksi halde} \end{cases}$$

değilse ($q > q_0$) :

$$p_{ij}^h = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{u \in N_h^i} [\tau_{iu}]^\alpha \cdot [\eta_{iu}]^\beta}, & \text{eğer } j \in N_h(i) \\ 0, & \text{aksi halde} \end{cases} \quad (3.5)$$

Sonraki ziyaret edilecek düğüm j , denklem (3.5)e göre tesadüfsel olarak seçilir. Görülebileceği üzere, kural bir çift amaca sahiptir: $q \leq q_0$ olduğunda, sezgisel bilgi ve feromon izine göre en iyi seçeneği seçerek elde edilebilir bilgiyi kullanır.

Bununla birlikte, eğer $q > q_0$ ise, KS'de yapıldığı gibi, bu kural bir kontrollü araştırma uygular. Özet olarak, kural yeni bağlantıların keşfedilmesi ve şimdiki elde edilebilir bilginin kullanılması arasında bir değişim meydana getirir.

2. Sadece daemon (ve tekil karıncalar değil) global feromon güncellemesini tetikler. Örneğin, bir çevrimdışı feromon iz güncellemesi yapılabilir. Böyle yapmak için, KKS sadece global en iyi çözümü ($S_{global-eniyi}$) üretmiş olan tek bir karıncayı ele alır. Feromon güncelleme, global-en iyi karınca (Bu KKS'de dikkat edilecek önemli bir noktadır, feromon buharlaştırma yalnızca feromon bırakılacak olan çözümün bağlantılarına uygulanır) tarafından gezilen tüm bağlantılar üzerindeki ilk feromon izleri şu denklem tarafından yapılır:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \quad \forall a_{ij} \in S_{global-eniyi}. \quad (3.6)$$

Sonra daemon şu kurala göre feromon bırakır:

$$\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot f(C(S_{global-eniyi})), \quad \forall a_{ij} \in S_{global-eniyi}. \quad (3.7)$$

Ek olarak, daemon, karıncaların çözümlerini feromon izlerini güncelleştirmesinden önce geliştirmeleri için bir yerel arama algoritması uygulayabilir.

3. Karıncalar yeni bulunan çözümlere karşı farklı çözümler üretmeyi cesaretlendiren bir çevrimiçi adım adım feromon iz güncelleme (yerel güncelleme) uygular. Bir karınca her seferinde bir a_{ij} kenarında seyahat ettiğinde, $\varphi \in (0,1]$ bir ikinci feromon azaltma parametresi olduğunda, şu kuralı uygular:

$$\tau_{ij} \leftarrow (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (3.8)$$

Görülebileceği üzere çevrimiçi adım adım güncelleme kuralı feromon buharlaştırmayı ve biriktirmeyi de içerir. Biriktirilen feromon miktarı çok az olduğundan (aslında, τ_0 başlangıç feromon iz değeri buna benzer bir yolla belirlenmiş olan, pratikte daha düşük bir feromon iz limitine denk gelir. Yani,

KKS feromon güncelleme kurallarının seçimi yoluyla, feromon iz değeri τ_0 'ın altına düşmez), bu kuralın uygulaması bir karınca tarafından geçtiği bağlantılar üzerinde feromon izinin az olmasını sağlar. Bundan dolayı KKS'nin bir ek keşif tekniğindeki bu yolla karıncaları takip etmek daha az çekici kılınmıştır ve her karıncanın aynı yolu takip etmesini önlemeye yardımcı olmuştur.

3.2 Çok-Amaçlı Optimizasyon

Çok-amaçlı optimizasyon problemleri birkaç amacın eş zamanlı optimize edilmiş olmak zorunda olmasından ortaya çıkmaktadır. Bundan dolayı genellikle problemi çözmek için en iyi bir tek çözüm yoktur. Bir tipik çok-amaçlı optimizasyon probleminde tüm amaçlar ele alındığında, pareto-önyüz denilen, arta kalan karar seçeneklerinden daha üstün bir çözümler kümesi vardır. Arta kalan (remainder) karar seçenekleri basılgın çözüm olarak biliniyorken, bu çözümler baskın çözümler olarak bilinmektedir (Chankong ve Haimes, 1983). Madem ki pareto-önyüz çözümlerinin herhangi biri diğer basılgın çözümlerden kesinlikle daha iyidir, o halde bunların hepsi tüm amaçların memnuniyeti hususunda eşit olarak kabul edilebilmektedir.

Baskınlık kavramının resmi tanımı şöyledir: n parametrelili (karar değişkenli) ve K amaçlı şu şekilde bir çok-amaçlı minimizasyon problemini ele alalım:

$$\text{Min } f(x) = (f_1(x), f_2(x), \dots, f_K(x)), \quad x = (x_1, x_2, \dots, x_n) \in X \quad (3.9)$$

Bir karar vektörü $a \in X$, $b \in X (a > b)$ 'e baskındır eğer, ve sadece eğer:

$$\forall i \in 1, 2, \dots, K | f_i(a) \leq f_i(b) \quad \wedge \quad \exists j \in 1, 2, \dots, K | f_j(a) < f_j(b) \text{ ise.} \quad (3.10)$$

Belirtildiği üzere, herhangi diğeri tarafından basılgın olmayan herhangi bir vektör Pareto-optimal veya baskın olarak söylenebilir.

Çok amaçlı optimizasyonun tek amaçlı optimizasyondan temel farklılığı amaçlar arasında çatışmanın mevcudiyetidir. Genelde uygun çözümlerin hiçbiri tüm amaçlar için eşanlı optimal çözümlere razı olmaz (Hans, 1988). Matematiksel olarak, bunu yapmanın en iyi yolu en az amaç çelişkisi ile o çözümleri korumaktır. Bu çözümler,

her amacın bireysel optimumundan optimal olarak yerleştirilmiş araştırma uzayındaki noktalar olarak gözlenebilir. Ancak, bu gibi çözümler bir karar-vericiyi memnun etmeyebilir çünkü o, bazı amaçların ilişkilendirilmiş önceliklerini memnun eden bir çözüm isteyebilir. Bu gibi noktaları bulmak için, tüm klasik yöntemler amaç vektörünü bir skaler (yönsüz ve sayısal) optimizasyon problemine düşürmek için skalarlaştırır. Gerçekten bu durumda belirlenmiş kısıtlar altında bir uzlaşık çözüm bulunmaktadır.

Coello ve diğ. (2002) çok-optimalliği çözmek için 3 en temel klasik yöntemden bahsetmektedirler: amaç ağırlıklandırma, mesafe fonksiyonları ve min-max formülasyonu. Bunların tümü çok-amaçlı optimizasyon problemini bir tek amaca düşürmek üzerine yapılandırılmıştır: Birincisinde (amaç ağırlıklandırmada), çok-amaçlı fonksiyonlar bir toplam amaç fonksiyonu içinde aşağıda görüldüğü gibi birleştirilmiştir:

$$F = \sum_{i=1}^K w_i \cdot f_i(x), \quad \sum_{i=1}^K w_i = 1, \quad 0 \leq w_i \leq 1 \quad (3.11)$$

Mesafe fonksiyonlarında, optimize edilmek için tek-amaç F fonksiyonu bir talep-seviye vektörü kullanılarak hesaplanır, \tilde{f} , bir karar-verici tarafından belirlenmiş olmak zorunda olunan, şu şekilde hesaplanan bir amaçtır:

$$F = \left[\sum_{i=1}^K |f_i(x) - \tilde{f}_i|^r \right]^{1/r}, \quad 1 \leq r < \infty \quad (3.12)$$

Son yöntem, min-max formülasyonu, tek-amaçlı fonksiyonların ilgili tekil optimumlarından görece sapmalarını minimize etmeye çaba gösterir. Bu yüzden, şu denklemi minimize etmeye çalışır:

$$f(x) = \max[Z_i(x)], \quad i = 1, 2, \dots, K \quad (3.13)$$

$Z_i(x)$, bir negatif olmayan hedef optimal değer $\tilde{f}_i > 0$ için şöyle hesaplanır:

$$Z_i(x) = \frac{f_i - \tilde{f}_i}{\tilde{f}_i} \quad (3.14)$$

Coello ve diğ. (2002) çok-amaçlı problemi çözmeye kullanılan tüm klasik tekniklerin zorluklarını şu şekilde açıklamışlardır:

- Amaçlar bir tek amaç fonksiyonu biçiminde birleştirildiğinde, bir yalnızca Pareto-optimal çözüm eşanlı olarak elde edilebilir. Gerçek-dünya problemlerinde, karar verici karar vermek için sık sık farklı alternatiflere gereksinme duyar, fakat bu teknikler bunları sunamaz.
- Bundan başka, amaçlar kümesinden tek amaca dönüştürülmeden önce, karar-verici bir zor problemde kendiliğinden var olan her amacın önceliğinin tam bir bilgisine genellikle sahip olmalıdır.
- Eğer bazı amaçlar gürültülü veya bir süreksiz değişken uzayına sahip ise, bu yöntemler uygun olarak çalışmayabilir.
- Eğer amaç fonksiyonları deterministik değilse, bir ağırlık vektörünün veya talep seviyesinin tanımlanması çok daha zor olabilir.
- Bu teknikler, ağırlıklar veya talep seviyelerine bağımlı ve çok duyarlılığa sahiptirler.

Bu problemleri çözmek için, son birkaç yılda bazı ileri çok-amaçlı optimizasyon teknikleri önerilmiştir. Bunlar tavlama benzetimi, geliştirme algoritmaları, ve KKO algoritmaları gibi, Gandibleux ve diğ. (2004)te görüldüğü gibi temel olarak iyi kurulmuş metasezgiseller üzerine yapılandırılmaktadır. Sonraki bölümde literatürde mevcut çok-amaçlı karınca koloni optimizasyonu algoritmaları incelenmektedir.

3.3. Çok-Amaçlı Karınca Koloni Optimizasyonu Algoritmaları

Bu bölümde ÇAKKO (Çok-Amaçlı Karınca Koloni Optimizasyonu) algoritmaları için literatürde var olan farklı öneriler, çözmekte oldukları problemler ve sahip oldukları ana nitelikleri rapor edilerek gözden geçirilmektedir.

3.3.1. Çok-amaçlı Ant-Q algoritması (CAAQA)

Çok-amaçlı Ant-Q algoritması (MOAQ) Mariano ve Morales (1999) tarafından sulama şebekeleri su dağıtımına uygulanarak önerilmiş bir ÇAKKO

algoritmasıdır. Bu algoritma, Dorigo ve Gambardella (1997)'nin geliştirdiği klasik KKS'nin bir türü olan, Ant-Q (Gambardella ve Dorigo, 1995) olarak adlandırılan bir dağıtılmış pekiştirme öğrenme algoritmasına dayanmaktaydı.

$$a = \begin{cases} \arg \max_{a \in A} (HE^\alpha(s, a) \cdot Q^\beta(s, a)), & \text{eğer } q > q_0 \\ P, & \text{diğer} \end{cases} \quad (3.15)$$

Ant-Q'da A bir durumdaki tüm mümkün hareketlerin kümesi ve S durumların kümesi olduğunda, bazı bağımsız ajanlar optimal bir politika olan $\pi: S \rightarrow A$ 'yı öğrenirler. Bu sonuçlarda $a \in A$ uygun hareketleri ve $s \in S$ olan şimdiki durumu göstermektedir. Ajan için elde edilebilir bilgi, tüm mümkün hareketler için $r(s_i, a_i)$ anlık ödülünün dizisi ve $i = 0, 1, 2, \dots, Q(s, a)$ durumlarıdır ve var olan (s) durumu içinde var olan belirli bir (a) hareketi daha iyi nasıl seçiliri gösteren sezgisel bilgiye bağlı bir HE(s, a) etki alanı vardır. Her ajan mevcut durum s'ye göre a hareketini seçmek için denklem (3.15)teki geçiş kuralını uygular.

α ve β feromon iz ve sezgisel bilginin görelî önem ağırlığı parametreleridir, q [0, 1] aralığında bir tesadüf sel değ erdir, ve q_0 ($0 \leq q \leq 1$) her iterasyonda şu formülle hesaplanır:

$$q_0 = \frac{q_0 \cdot \lambda}{q_{\max}}, \quad (3.16)$$

$q_{\max} \in [0, 1]$, ve P aşağıdaki olasılık dağılımına göre seçilmiş bir tesadüf sel harekettir:

$$p(a) = \frac{HE^\alpha(s, a) \cdot Q^\beta(s, a)}{\sum_{b \in A} HE^\alpha(s, b) \cdot Q^\beta(s, b)} \quad (3.17)$$

s şimdiki durumu göstermektedir.

CAAQA ardındaki temel düşünce, her amaç için bir ajan (karınca) ailesi ile bir optimizasyon algoritması ortaya koymaktır. Her k ailesi, diğ er amaçlar ve onların ele alınan HE^k fonksiyonu için bulunmuş çözümleri ele alarak bir amacı optimize etmeye çalışır. Bu yolla, farklı ailelerdeki tüm ajanlar hareketler düşünerek ve bir γ

ödül değeri (hareketlerinin ajanların kalanları arasında ödünleşme çözümleri bulmaya yardım edeceğine bağlı olan) bekleyerek aynı çevrede hareket ederler. Geciktirilmiş takviye şöyle hesaplanır:

$$Q(s, a) = (1 - \rho) \cdot Q(s, a) + \rho \cdot [r(s, a) + \gamma \cdot Q(s', a')], \quad (3.18)$$

ρ öğrenme basamağıdır, s' ve a' sonraki algoritma adımındaki durum ve harekettir, ve γ bir indirim faktörüdür.

Son olarak, ÇAAQA diğer üç ayırt edici özelliği sunar. İlki, i . ailedeki j . karınca çözümünü inşa ederken $i-1$. ailenin j . karıncası tarafından bulunan çözümü kullanır. İkincisi, bir çözüm olarak uygun bulunmadığında, algoritma Q değerleri üzerindeki elemanlarına bir cezalandırma uygular. Ve üçüncüsü, süreç boyunca, baskın çözümler bir dışsal dizide düzenlenir.

3.3.2 İki-kriterli optimizasyon problemleri için karınca algoritması

İkikriterli Karınca algoritması (İKKA-BicriterionAnt) Iredi ve diğ. (2001) tarafından özel olarak bir iki-kriterli araç rotalama problemini çözmek için tasarlanmıştır. Bu hesaplama için iki farklı feromon iz matrisi kullanır. τ ve τ' , her biri ele alınan kriter içindir.

$$p(j) = \begin{cases} \frac{\tau_{ij}^{\lambda a} \cdot \tau'_{ij}{}^{(1-\lambda)\alpha} \cdot \eta_{ij}^{\lambda\alpha} \cdot \eta'_{ij}{}^{(1-\lambda)\alpha}}{\sum_{u \in \Omega} \tau_{iu}^{\lambda a} \cdot \tau'_{iu}{}^{(1-\lambda)\alpha} \cdot \eta_{iu}^{\lambda\alpha} \cdot \eta'_{iu}{}^{(1-\lambda)\alpha}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde} \end{cases} \quad (3.19)$$

Her üretimde, kolonideki m karıncanın her biri probleme bir çözüm üretir. İnşa gezisi sırasında, karınca sonraki j düğümünü (3.19)daki olasılık dağılımının ortalamasına göre ziyaret edilmek üzere seçer.

$$\lambda_h = \frac{h-1}{m-1}. \quad (3.20)$$

α ve β önceki bölümlerde bahsedilen ağırlıklandırma parametreleridir. η_{ij} ve η'_{ij} , sırasıyla, birinci ve ikinci amaca göre a_{ij} kenarına ilişkilendirilmiş sezgisel değerlerdir. Ω karıncanın mevcut uygun komşuluğudur ve λ her h karıncası ($h \in \{1, \dots, m\}$) için (3.20)deki formülle hesaplanmaktadır.

Tüm karıncalar çözümlerini ürettikten sonra, feromon izleri için her a_{ij} kenarı üzerine genel olarak aşağıdaki denklemle buharlaştırma uygulanır:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} \quad (3.21)$$

$\rho \in [0, 1]$ feromon buharlaştırma oranıdır.

Daha sonra pareto-önyüzde bir çözüm üretmiş olan karınca sayısı l olduğunda, güncelleme miktarı $1/l$ 'ye eşit bir miktarda hazırlanarak, ilgili iterasyon içinde τ ve τ' feromon matrislerinin her ikisini de günceller. Algoritma çalıştırılması boyunca üretilmiş baskın çözümler ÇAAQA'da olduğu gibi bir dışsal kümede korunurlar.

3.3.3. İki-kriterli optimizasyon problemleri için çoklu koloni

Iredi ve diğ. (2001) iki farklı ÇAKKO algoritması daha önermişlerdir. İki-kriterliÇK (BicriterionMC) algoritması (İKÇKA), İKKA'ya çok benzerdir. Aralarındaki ana farklılık her karınca yeni öneride sadece bir feromon matrisini günceller. Yazarlar p adet feromon iz matrisi kullanan algoritma için genel bir tanımlama ortaya koymuşlardır ve daha sonra iki-kriterli problemleri çözmek için $p = 2$ yapmışlardır. Bu çalışmada, feromon iz güncellemesi için iki farklı yöntem ele almaktadırlar:

- Yöntem 1 – Başlangıç(orjin) tarafından güncelleme: Bir karınca yalnızca kendi kolonisindeki feromon izleri günceller. Yöntem, baskın hattın farklı bölgelerini araştırmak için her iki koloniyi de zorlamaktadır. Yöntem 1'i kullanan algoritma Bölümlendirilmemiş İki Kriter Algoritması (BİKA-UnsortBicriterion) olarak adlandırılmaktadır.

- Yöntem 2 – Bölge tarafından güncelleme: Baskın hat boyunca çözümlerin dizisi eşit boyutta p parçaya ayrılmaktadır. i . parçadaki çözümleri bulunmuş olan karıncalar i kolonisinde feromon izleri güncellemektedir ($i \in [1, p]$). Amaç her biri bir bölgede olan Pareto hattının farklı bölgelerini araştırmak için açıkça klavuzluk etmektir. Yöntem 2'yi kullanan algoritma İKÇKA olarak adlandırılıyor.

Diğer farklılık karınca t için λ geçiş parametresinin değerini hesaplamak yoludur. Yazarlar 3 farklı kural tanımlamakta ve daha iyi sonuçlar veren 3.sünü kullanmayı önermektedirler. Bu kural koloni $i-1$ ve koloni $i+1$ 'in λ aralığı ile bir %50 içindeki i kolonisinin λ aralığını üst üste getirmektedir. i kolonisi karınca sayısı, p

koloni sayısını gösterdiğinde $\left[\frac{i-1}{p+1}, \frac{i+1}{p+1} \right]$ aralığındaki λ değerleri ile hesaplanır.

3.3.4. Pareto karınca koloni optimizasyonu

Doerner ve diğ. (2004) tarafından önerilmiş olan Pareto Karınca Koloni Optimizasyonu Algoritması (PKKOA, P-ACO), çok-amaçlı portföy seçim problemini özgün şekilde çözmek için uygulanmıştır. Bu algoritma, klasik KKS'yi temel almaktadır. Fakat global feromon güncelleme, en iyi ve ikinci en iyi çözümleri her k amacı için mevcut iterasyon içinde yarattığı iki farklı karınca kullanarak gerçekleştirilmektedir.

PKKOA, her amaç k için bir tane olmak üzere birkaç feromon matrisi τ^k 'yi dikkate almaktadır. Her algoritma iterasyonunda, her karınca $p = (p_1, \dots, p_k)$ ağırlıklarının bir kümesini hesaplar ve feromon iz ile sezgisel bilgiyi birleştirmek için kullanır. Bir karınca ziyaret edilecek diğer düğümü seçmek zorunda kaldığında k feromon matrislerini dikkate alarak şu KKS geçiş kuralını kullanır:

$$j = \begin{cases} \arg \max_{j \in \Omega} \left(\left[\sum_{k=1}^K p_k \cdot \tau_{ij}^k \right]^\alpha \cdot \eta_{ij}^\beta \right), & \text{eğer } q \leq q_0 \\ \hat{i}, & \text{aksi halde.} \end{cases} \quad (3.22)$$

K amaçların sayısı, η_{ij} sezgisel bilgi olarak kullanılan a_{ij} kenarının cazibesinin toplam değeri ve \hat{i} şu olasılık dağılımına göre seçilmiş bir düğümdür:

$$p(j) = \begin{cases} \frac{\left[\sum_{k=1}^K p_k \cdot \tau_{ij}^k \right]^\alpha \cdot \eta_{ij}^\beta}{\sum_{u \in \Omega} \left[\sum_{k=1}^K p_k \cdot \tau_{iu}^k \right]^\alpha \cdot \eta_{iu}^\beta}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde.} \end{cases} \quad (3.23)$$

Bir karınca a_{ij} kenarında her geçişinde, her feromon iz matrisinde yerel feromon güncelleme, örneğin her amaç k için, aşağıdaki gibi uygular:

$$\tau_{ij}^k = (1 - \rho) \cdot \tau_{ij}^k + \rho \cdot \tau_0 \quad (3.24)$$

$\Delta \tau_{ij}^k$ aşağıdaki formüle göre değer alır:

$$\Delta \tau_{ij}^k = \begin{cases} 15, & \text{eğer } a_{ij} \text{ kenarı} \in \text{en iyi ve ikinci en iyi çözümler} \\ 10, & \text{eğer } a_{ij} \text{ kenarı} \in \text{en iyi çözüm} \\ 5, & \text{eğer } a_{ij} \text{ kenarı} \in \text{ikinci en iyi çözüm} \\ 0, & \text{aksi halde.} \end{cases} \quad (3.25)$$

Önceki ÇAKKO algoritmalarında olduğu gibi, süreç boyunca bulunan baskın çözümler bir dışsal kümede depolanmaktadır.

3.3.5. Teslim süreleri ile araç rotalama problemi için çoklu karınca koloni sistemi

Adından da anlaşılacağı üzere, Gambardella ve diğ. (1999) tarafından ortaya konan ÇKKS-TSARP (MACS-VRPTW) algoritması araç rotalama problemlerinin bu spesifik türünü çözmek için düşünülmüştür. PKKOA'da olduğu gibi, klasik KKS'den başlamaktadır. ÇKKS-TSARP seyahat zamanından çok tur sayısını minimize etmek için bir öncelik düzenlemektedir. Her iki amacın eş zamanlı optimize edildiği global ÇKKS-TSARP algoritması tarafından koordine edilen aktivitelerde, KKS-A (ACS-VEI) ve KKS-Z (ACS-TIME) olmak üzere iki farklı koloni çalışmaktadır. Sonraki koloni önceki tarafından elde edilen uygun çözümleri optimize etmeye çalışırken önceki koloni kullanılan araç sayısını azaltmaya çalışır. Her biri spesifik amacı için

bir bağımsız feromon iz matrisi kullanır ve birleştirme hareketi ile bulunan en iyi çözümü paylaşarak işbirliği yaparlar. Global algoritma, bir yeni en az araç içeren en iyi çözüm bir öncekine göre elde edildikten sonra, her seferinde iki koloniyi durdurur ve çalıştırır.

3.3.6. Çoklu karınca koloni sistemi

Barán ve Schaerer (2003) önceki bölümde anlatılmış olan ÇKKS-TSARP algoritmasının bir çeşidi olarak Çoklu Karınca Koloni Sistemi (ÇKKS)ni önermiştir. KKS üzerine temellerini kurmuş bu algoritma, selefinin aksine, bir tekli feromon matrisi τ ve iki sezgisel bilgi matrisi η^0 ve η^1 kullanır. Bu yolla, bir karınca düğüm i 'den düğüm j 'ye şu kuralı uygulayarak hareket eder:

$$j = \begin{cases} \arg \max_{j \in \Omega} (\tau_{ij} \cdot [\eta_{ij}^0]^{\lambda\beta} \cdot [\eta_{ij}^1]^{(1-\lambda)\beta}), & \text{eğer } q \leq q_0 \\ \hat{i}, & \text{aksi halde.} \end{cases} \quad (3.26)$$

β sezgisel bilginin önemidir. m karınca sayısı olduğunda λ her karınca h için $\lambda = h/m$ olarak hesaplanmaktadır ve amaçların göreceli önemini ağırlıklandırır. \hat{i} aşağıdaki olasılık dağılımına göre seçilmiş bir düğümdür:

$$p(j) = \begin{cases} \frac{\tau_{ij} \cdot [\eta_{ij}^0]^{\lambda\beta} \cdot [\eta_{ij}^1]^{(1-\lambda)\beta}}{\sum_{u \in \Omega} \tau_{iu} \cdot [\eta_{iu}^0]^{\lambda\beta} \cdot [\eta_{iu}^1]^{(1-\lambda)\beta}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde.} \end{cases} \quad (3.27)$$

Bir karınca a_{ij} kenarını her geçişinde, aşağıdaki yerel feromon güncelleme kuralını uygular:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (3.28)$$

Başlangıçta τ_0 her iki amaç fonksiyonu içindeki ortalama maliyetleri alınarak bir sezgisel çözümler kümesinden hesaplanmaktadır, bu maliyetler f^0 ve f^1 ile gösterilir ve aşağıdaki denklem ile uygulanır:

$$\tau_0 = \frac{1}{\hat{f}^0 \cdot \hat{f}^1} \quad (3.29)$$

Buna rağmen, τ_0 değeri, algoritma çalıştırılması sırasında genel KKS'deki gibi sabit kalmaz. Bir h karıncası her tam bir çözüm oluşturduğunda kontrol için şimdiye kadar üretilmiş Pareto kümesi P ile eğer önceki çözüm bir baskın çözüm ise karşılaştırılır. Her iterasyon sonunda τ'_0 Pareto kümesini içeren anlık çözümler alan her amaç fonksiyonunun ortalama değerleri ile önceki formül uygulanarak hesaplanmaktadır.

Daha sonra, eğer iterasyondaki feromon değeri $\tau'_0 > \tau_0$ ise, feromon izleri yeni değere atanarak ($\tau_0 \leftarrow \tau'_0$) yeniden başlatılır.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \frac{\rho}{f^0(S) \cdot f^1(S)} \quad (3.30)$$

Öte yandan, global güncelleme, iterasyondaki Pareto optimal küme P 'nin her çözümü S ile a_{ij} kenarları üzerinde oluşturulduğu denklem (3.30)a göre uygulanır.

3.3.7. Karınca kolonileri optimizasyonu çok-amaçlı şebekesi

Cardoso ve diğ. (2003) Karınca kolonileri optimizasyonu çok-amaçlı şebekesi (KKOÇAŞ, Multi-Objective Network ACO) adıyla, bir şebekedeki mesaj trafiği optimizasyonu gibi bir dinamik probleme uygulanmak için tasarlanmış olan ve bundan dolayı gözden geçirilmiş diğer algoritmalarından oldukça farklı yapıda yeni bir algoritma önermişlerdir. Bundan dolayı, şebekenin davranış biçimi algoritmanın adımlarına göre değişir ve algoritma sonlanıncaya kadar beklemes. KKOÇAŞ'ı statik problemlerde kullanmak için bir adaptasyona ihtiyaç vardır. Karıncaların feromon izlerini güncellemeden önce döngü sonlandırılıncaya kadar beklemek zorunda olduğu gibi bazı modifikasyonlar içermektedir.

$$p(j) = \begin{cases} \frac{\eta_{ij}^\beta \cdot \prod_{k=1}^K (\tau_{ij}^k)^{\alpha_k}}{\sum_{u \in \Omega} \eta_{iu}^\beta \cdot \prod_{k=1}^K (\tau_{iu}^k)^{\alpha_k}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde} \end{cases} \quad (3.31)$$

Orijinal algoritma klasik KS'ni (Dorigo ve diğ., 1996) temel olarak ele alır, fakat birkaç feromon iz matrisi (τ^k) kullanır. Bir mesaj olarak tanımlanan her karınca, çoklu-feromon iz ve ziyaret edilecek sonraki düğümü seçmek için denklem (3.31)deki olasılık dağılımına göre bir tekli sezgisel bilgi kullanır. Bu formülde η_{ij} , a_{ij} kenarı için sezgisel bilgidir. β ve farklı α_k 'nın her ikisi her sezgisel bilgi ve feromon matris değerlerinin önemi ağırlığı, K amaç fonksiyonlarının sayısı ve Ω bu aşamadaki karıncanın uygun komşusudur.

$$\tau_{ij}^k = (1 - \rho_k) \cdot \tau_{ij}^k, \quad (3.32)$$

Her çevrimden sonra, feromon izleri genel yolla buharlaştırılmış şimdiki iterasyon içindeki en az bir karınca tarafından ziyaret edilen her kenara denklem (3.32) ile ilişkilendirilir. ρ_k k amacı için feromon buharlaştırma oranıdır (her feromon iz matrisi için farklı bir buharlaştırma oranı ele alındığına dikkat edelim). Daha sonra, bu kenarların feromon izleri güncellenir.

Her h karıncası, her k hedefi için ve her karınca tarafından kullanılan her a_{ij} kenarı içinde aşağıdaki denklemden hesaplanan feromon miktarı kadar feromon bırakır:

$$\Delta \tau_{ij}^k = \frac{Q}{f^k(S_h)} \quad (3.33)$$

Q karıncalar tarafından feromon miktarı ile ilişkili bir sabit, f^k k 'nın amaç fonksiyonu ve S_h karınca tarafından üretilen çözümdür. Belirtildiği üzere, orijinal algoritmanın amacı baskın çözümlerin iyi bir kümesini bulmak değil şebekeyi etkili olarak çalışır hale getirmektir. Bunu statik optimizasyon problemlerine uygulamak için, bir dış kümedeki her komşunun içinde üretilmiş baskın çözümleri depolamak zorundayız.

3.3.8. Ehil karıncalar çok-amaçlı sistemi (EKÇAS-COMPETants)

Doerner ve diğ. (2003) iki-amaçlı transportasyon problemleri üzerinde çalışmak için EKÇAS'ı ortaya koydular. Algoritma, Bullheimer ve diğ. (1999)'nin ilk olarak yayınladığı sıralama-temelli KS üzerine temellenmiş olan, iki karınca kolonisi kullanan, her birinin kendi feromon iz matrisi τ^0 ve τ^1 olan, η^0 ve η^1 sezgisel bilgisi olan bir algoritmadır. Diğer algoritmalarından farklı ve ilginç olarak her popülasyondaki karıncaların sayısı sabit değildir, ancak adaptasyona uğramaktadır.

Her karınca çözümünü inşa etmiş olduğunda daha iyi çözümler inşa etmiş olan koloni sonraki iterasyon için daha çok karınca alır. Karıncalar, sonraki ziyaret edilmiş olacak düğümü seçmek için aşağıdaki olasılık dağılımını kullanarak kenarlar boyunca hareket ederler (çoklu sezgisel ve feromon iz değerlerinin durumunun KS geçiş kuralına bir uyarlama olduğuna dikkat edelim):

$$p(j) = \begin{cases} \frac{\tau_{ij}^{k^\alpha} \cdot \eta_{ij}^{k^\beta}}{\sum_{u \in \Omega} \tau_{iu}^{k^\alpha} \cdot \eta_{iu}^{k^\beta}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde.} \end{cases} \quad (3.34)$$

Her karınca k , kolonisinin τ ve η değerlerini kullanır. Ardından, casus olarak adlandırılan her popülasyondaki bazı karıncalar, her iki feromon izlerin bilgilerini birleştirmek için aşağıdaki farklı bir kuralı uygularlar:

$$p(j) = \begin{cases} \frac{[0.5 \cdot \tau_{ij} + 0.5 \cdot \tau'_{ij}]^\alpha \cdot \eta_{ij}^{k^\beta}}{\sum_{u \in \Omega} [0.5 \cdot \tau_{iu} + 0.5 \cdot \tau'_{iu}]^\alpha \cdot \eta_{iu}^{k^\beta}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde.} \end{cases} \quad (3.35)$$

Bu yeni kural içerisinde, τ karıncanın feromon izi ve τ' diğer koloninin feromon izidir. η^k karıncanın sezgisel bilgisidir. Her karıncanın çözümünü oluşturduğu, her a_{ij} kenarındaki feromon izleri şu formülle buharlaştırılmaktadır:

$$\tau_{ij}^k = (1 - \rho) \cdot \tau_{ij}^k. \quad (3.36)$$

Daha sonra popülasyonun en iyi karıncaları Γ , kendi feromon iz matrisi ve aşağıda verilen kuralı kullanarak, ziyaret edilmiş kenarlar üzerine feromon bırakırlar:

$$\Delta\tau_{ij}^\lambda = 1 - \frac{\lambda-1}{\Gamma} \quad (3.37)$$

λ , Γ en iyi karıncalar listesindeki karıncanın pozisyonudur. Çevrimin sonuna doğru her karınca şu denklemdeki olasılık ile ilk koloniye atanır:

$$\frac{\hat{f}^1}{\hat{f}^0 + \hat{f}^1} \quad (3.38)$$

\hat{f}^1 ikinci kolonideki çözümlerinin (ikinci amaç fonksiyonu ile ilişkili) maliyetlerinin ortalaması ve \hat{f}^0 ilk koloninin (ilk amaç fonksiyonuyla ilişkili) maliyetlerinin ortalamasıdır. Arta kalan karıncalar ikinci koloniye atanmaktadır.

$$\frac{f(best)}{4 \cdot f'(best') + f(best)} \quad (3.39)$$

Son olarak, her iki kolonideki casusların sayısı (3.39)daki olasılık ile tesadüf sel olarak hesaplanmaktadır. f mevcut koloni ile ilgili amaç fonksiyonu, f' diğer koloni ile ilgili amaç fonksiyonu, $best$ f 'ye göre mevcut koloninin en iyi karıncası ve $best'$ diğer f' 'ne göre en iyi karınca olmaktadır.

3.3.9 Çok-amaçlı karınca koloni optimizasyonu metasezgiseli (ÇAKKOM)

Gravel ve diğ. (2002) Çok-amaçlı karınca koloni optimizasyonu metasezgiseli (ÇAKKOM-MOACOM) olarak adlandırdıkları, alüminyum üretim sanayinde bir çizelgeleme problemi için bir ÇAKKO algoritmasını önerdiler. ÇAKKOM karar verici tarafından a priori (önsel) olarak düzenlenmiş, bir sözlük sırası içerisinde çoklu amaçlarla uğraşan bir KS algoritması üzerine temellenmiştir.

Bir çevrimin sonunda ele alınan sözlük sırasına göre sadece bir çözüm hesaba katılır. Bu sebeple ÇAKKOM'un amacı tam olarak iyi bir baskın çözüm kümesi bulmak değildir. Bundan sonra algoritma bir tek sezgisel bilgi ve feromon iz matrisi ile

çalışır. Her değer her amaçla ilişkilendirilmiş bilginin birleştirmesinin sonucudur. Yazarlar λ sezgisel matrisini inşa etmek için birkaç kural ileri sürmüşlerdir.

$$\Delta\tau_{ij} = \frac{Q}{f^0(S_h)} \quad (3.40)$$

Daha sonraki iki matris ele alındığında, KS geçiş kuralı karıncaların çözümlerini inşa etmek için uygulanmaktadır. Bir çevrimin sonunda, feromon iz yoğunluğu yalıtılmış ilk öncelikli amacın değerlendirilmesi esaslı olarak güncelleştirilmektedir. Her h karıncası kullanılmış kenarlar üzerine (3.40)daki miktarda feromon bırakmaktadır. f^0 ilk öncelikli amaç fonksiyonu, S_h karınca tarafından bulunan çözüm ve Q bir sabittir.

3.3.10 Çok-amaçlılara karınca koloni optimizasyon yaklaşımı (ÇAKKOY)

McMullen (2001) bir çok-amaçlı tam zamanında (JIT) sıralama problemini çözmek için ÇAKKOY algoritmasını önerdi. Bu algoritma önceki algoritmadaki olduğu gibi her iterasyondaki baskın bir küme bulmak yerine tek bir en iyi çözüm bulmaya uğraşmak esasına dayanmaktadır. Bu çözüm amaç üzerinde tanımlanmış bir sözlük sıralamasının ilkidir.

ÇAKKOY çözümleri parçalara ayırır ve araştırma uzayındaki bu tesadüsel parçaları tesadüsel dağıtır. Sezgisel bilgi parçalar arası mesafelerin tersidir ve karıncalar şu olasılık dağılımına göre hareket ederler:

$$p(j) = \begin{cases} \frac{\tau_{ij} \cdot \eta_{ij}}{\sum_{u \in \Omega} \tau_{iu} \cdot \eta_{iu}}, & \text{eğer } j \in \Omega \\ 0, & \text{aksi halde.} \end{cases} \quad (3.41)$$

τ feromon iz matrisi, η sezgisel bilgiye ait matris ve Ω karıncanın mevcut uygun komşuluğudur.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (3.42)$$

Bir karınca a_{ij} kenarı doğrultusunda hareket ettiğinde denklem (3.42)deki KKS yerel güncelleme kuralını uygular.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot C^{-1} \quad (3.43)$$

Son olarak bir çözümü tamamlayan her karınca geçtiği her kenarın mesafelerini ve öklidyen toplam olarak maliyetini (C) hesaplar. Eğer bu değer, C 'nin o iterasyona kadar bulunan en küçük değerinden daha az ise, global güncelleme denklem (3.43)e göre bu çözümdeki her kenara uygulanır.

3.3.11. SACO

SACO iki makine iki kriterli akış tipi çizelgeleme problemleri için T'kindt ve diğ. (2002) tarafından önerilmiş özel bir ÇAKKO algoritmasıdır. Adı tavlama benzetimi ve bazı yerel arama algoritmalarının uygulanması ile ortaya çıkan bir araştırma metodolojisinden gelmektedir.

Yolculuğun başlangıcında güçlü bir çeşitlendirme geliştirmek ve sonraki safhalarda güçlü bir yoğunlaşma ortaya koymak esasına dayalıdır. Baskın çözümlerin bir kümesi üzerinde çalışma ile değil her amacın en iyi maliyetine sahip olan bir en iyi çözüm ile uğraşma üzerine temellidir.

$$p_0 = \frac{\log(it)}{\log(N)} \quad (3.44)$$

$$\tau_{ij} = \tau_{ij} + \frac{1}{f^0(S)} \quad (3.45)$$

Her çevrimde, her karınca feromon iz bilgisi kullanarak iki farklı yol kullanabilen bir uygun çözüm inşa etmektedir: a) Yoğunlaşma modu, her karıncanın en yüksek τ_{ij} değerine göre en uygun işi seçer ve b) Çeşitlendirme modu, her karıncanın sonraki düğümünü bir tesadüfsel-oran kuralına göre seçer. Yoğunlaşma modunu seçim olasılığı p_0 ile temsil edilmektedir, denklem (3.44)teki gibi her iterasyon it de hesaplanan bir parametredir.

N toplam iterasyon sayısıdır. Çevrimin sonunda her kenarda oluşturulan buharlaştırma ve elde edilen en iyi çözüm, kenarların feromon izlerini güncellemede kullanılır. S bulunan en iyi çözüm ve f^0 amaçlardan biridir.

4. İKİ KRİTERLİ KARESEL ATAMA PROBLEMLERİNİN ÇOK-AMAÇLI KARINCA KOLONİLERİ ALGORİTMALARI İLE OPTİMİZASYONU

Önceki bölümlerde bahsedilen ÇAKKO algoritmalarını kısaltmalarıyla özetlersek: Mariano ve Morales (1999), ÇAAQA; Gambardella ve diğ. (1999), ÇKKS-TSARP; McMullen (2001), ÇAKKOY; İredi ve diğ. (2001), İKKA, BİKA ve İKÇKA; Gravel ve diğ. (2002), ÇAKKOM; T'kindt ve diğ. (2002), SACO; Barán ve Schaerer (2003), ÇKKS; Cardoso ve diğ. (2003), KKOÇAŞ; Doerner ve diğ. (2003, 2004), EKÇAS, PKKOA. Bunların çoğu çizelgeleme, araç rotalama, veya portföy seçiminin de içinde bulunduğu somut birer çok amaçlı problem olan problemleri çözmek için spesifik önerilerdir.

Bu algoritmalarından ÇKKS, ÇKKS-TSARP'ın daha geliştirilmiş, genelleştirilmiş ve iyi sonuçlar veren bir versiyonu olduğu için karşılaştırmalar içerisine alınmıştır. ÇKKS-TSARP daha çoğunlukla araç rotalama problemlerine yönelik önerilmiş özel bir algoritmadır. Bunun yanı sıra ÇAKKOY ve ÇAKKOM tek amaçlı KKO algoritmasının bir versiyonu gibi düşünülebildiğinden ve pareto-önyüze yönelik olmadığı açık olduğundan karşılaştırmalarda dikkate alınmamıştır. SACO da bunlara benzer özellikte ve ayrıca çok özel bir problem türüne yönelik hazırlanmış melez bir algoritma olduğundan karşılaştırmalar içinde yer almamıştır. Geriye kalan ÇAAQA, İKKA, BİKA, İKÇKA, ÇKKS, KKOÇAŞ, EKÇAS ve PKKOA algoritmaları iki kriterli karesel atama problemlerini çözmek üzere MATLAB programlama dilinde kodlanmıştır. Karşılaştırmalar içerisine bazı algoritmaların feromon güncelleme matrislerinin amaç fonksiyonu değerleri (maliyetler) ile formülize edilmesiyle ortaya çıkan farklı versiyonları da eklenmiştir. Bunlar İKKA-M, BİKA-M, İKÇKA-M ve PKKOA-M olarak isimlendirilmiştir. Bu özelliğe doğasından sahip olan algoritmalar için bu ekleme yapılmasına gerek kalmamıştır. Böylelikle toplam 12 çeşit algoritma değerlendirilmeye alınmıştır.

Birçok çalışmada görülebileceği gibi algoritmalar karşılaştırılırken sabit parametre düzenleri kullanılmaktadır. Oysa bazı değişkenleri ortak olsa da, yapıları ve hesaplamaları birbirinden çok farklı olan bu algoritmaları sabit parametre düzeni ile karşılaştırmanın çok adaletli olmayacağı açıktır. Bundan dolayı, algoritmaları değerlendirme çalışmaları yapılmaya başlamadan önce, her algoritmanın kendi içerisinde en iyi sonuçları veren parametre düzenleri bulunmaya çalışılmıştır.

Her algoritmanın iyi sonuç veren parametre düzenlerini bulmak maksadıyla yapılmış olan deney tasarımlarında, Fıđlalı ve diđ. (2009)da ortaya konmuş sonuçlar dikkate alınmıştır. Bu çalışmada yazarlar KS algoritmalarını çizelgeleme problemleri üzerinde denerken parametreler arasında etkileşim olmadığını, bundan dolayı bazı kesirli deney tasarımlarının, tam faktöryel deney tasarımlarından çok farklı sonuçlar ortaya koymadığını göstermişlerdir. Bu çalışmada özellikle KS kullanan sistemlerin mümkün olduğunca çok sayıda karıncayla koşturulmasının iyi sonuçlar vereceđi görölmüştür.

Bu yönseme dikkate alınarak yapılmış parametre eniyileme çalışmalarında görölmüştür ki; KKO temelli algoritmalarda benzer kaniya ulaşmak çok mümkün görölmemektedir. Özellikle feromon güncelleme hesaplamalarındaki farklılıklarından ve literatür çalışmalarında KKO üzerine parametre eniyileme çalışmalarının yapılmamış olmasından dolayı, KKO temelli algoritmaların parametre düzenlerinin belirlenmesinde KS temelli algoritmalarda kullanılan parametre düzenlerine benzer düzenler kullanmak deney sonuçlarını yanlış yönlendirebilirdi. Bütün bunların dışında KS temelli olarak sözü edilen algoritmalarından bazılarının bazı hesapsal yapıların klasik KS'den çok farklı oluşu, KKO temelli ÇAKKO algoritmaları için de benzer durumun söz konusu olması da parametre eniyileme çalışmalarında karşımıza bir zorluk olarak çıkmaktadır.

İki kriterli karesel atama problemlerini çözmek üzere programlanmış ÇAKKO algoritmalarına ait gürbüz parametre düzenlerini bulmak maksadıyla, sözü geçen çalışmada önerilen deney düzenleri ve sözü edilen düşünceler ele alınarak kesirli deney tasarımları ortaya konulmuştur.

Tablo 4.1: İKKA (BicriterionAnt) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	80	100
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

Tablo 4.2: İKKA (BicriterionAnt) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	100	1	1	0,02

Bütün tasarımlar (PKKOA ve PKKOA-M hariç) adaletli olunması açısından $2^3 + 1$ desenli olarak tasarlanmıştır. Çok çeşitli faktörlere sahip olan algoritmaların daha detaylı incelenmesi yerine bütün algoritmaların sonuçta bu boyutta kesirli tasarımlara indirgenmesi sağlanarak, az çeşitli faktöre sahip algoritmaların rekabet şansı korunmuş, deney zamanından da tasarruf sağlanmıştır. Ayrıca bu algoritmaların kendi ilk önerdikleri bilimsel çalışmalarında koşulmuş olan parametre düzenleri de 9. düzen olarak söz konusu tasarımlarda yerini almıştır. 9. düzenlerin hepsi tablolarda seviye olarak değil ilgili faktörün değeri cinsinden gösterilmektedir. Yalnız PKKOA ve PKKOA-M algoritmalarında, Figlali N. ve diğ. (2009)da önerilmiş tasarımlar, bahsettiğimiz nedenlerden dolayı olumsuz sonuçlar vermişlerdir. Bu algoritmalar için 2^3 tasarımlar oluşturulmuş ve sadece önerdikleri makalelerdeki parametre düzenleriyle parametre eniyilemesi yapılması tasarlanmıştır.

Hem parametre eniyilemesi yapmak hem de ÇAKKO algoritmalarını karşılaştırmak için Knowles ve Corne (2003, 2009)da mevcut literatür problemlerinden faydalanılmıştır. Bu bilimsel çalışmalarda ortaya konmuş problemlerden 30 boyutlu olanları 3 kriterli, 10 ve 20 boyutlu olanları ise iki kriterli KAP'ları çözmek üzere tasarlanmıştır. Bu çalışmada gürbüz parametre düzeninin bulunması amacıyla

KC10-2fl-1uni problemi çözülmüştür. Aynı problem ve diğer 10 ile 20 boyutlu iki kriterli problemler ÇAKKO algoritmalarını karşılaştırmak amacıyla kullanılmıştır.

İki kriterli optimizasyon algoritmalarını karşılaştırmak üzere literatürde çok sayıda yöntem mevcuttur. Bunlardan Zitzler ve diğ. (2000)'nin çalışmaları en önemli ve en çok atıf alan yöntemlerdendir. Bu çalışmada algoritmalar, çözmüş oldukları iki kriterli optimizasyon problemlerinden elde edilen pareto-önyüz'lerine göre karşılaştırılmaktadır. Bu karşılaştırmaları yapmak maksadıyla M_1 , M_2 , M_3 ve C ölçüleri adı verilen metrikler ortaya atılmıştır.

Bunlardan M_1 , algoritmadan alınan pareto-önyüz ile sözde-optimal olan pareto-önyüz (veya ikinci bir algoritmanın pareto-önyüzü) arasındaki (ortalama) uzaklığı değerlendirmek amacıyla hesaplanmaktadır. M_2 , algoritmadan alınan pareto-önyüzün dağılımını değerlendirmektedir. M_3 , elde edilen pareto-önyüzün uç noktalarını değerlendirmektedir. C ölçüsü ise daha çok algoritmaları ikili karşılaştırmaya yarar. Bu ölçüyle algoritmalar elde ettikleri pareto-önyüzlerinin baskınlık derecelerine göre karşılaştırılmaktadır.

Bütün bunlara göre iyi sonuç veren algoritma şüphesiz ki en iyi algoritma olacaktır. Ancak çalışmamızda 12 adet algoritmayı karşılaştırmayı düşünmekteyiz. Çabamız, dağılımı nasıl olursa olsun, orjinden veya en iyi pareto-önyüzden ne kadar uzakta olursa olsun ve uç noktalarda ne kadar başarılı olursa olsun, en iyi pareto-önyüzü elde eden algoritmayı seçebilmektir. Bahsedilen ilk 3 ölçünün dezavantajı üçünün birden değerlendirilmesindeki zorluktan kaynaklanmaktadır. Adı geçen çalışmada ortaya atılmış C ölçüsü ise tamamen pareto-önyüze odaklanmaktadır. İlk üç ölçüyü bir anlamda birleştirici nitelik taşımakta, ancak algoritmaları ikili karşılaştırırken sadece baskınlık ölçüsünü dikkate almaktadır. Bu yöntemin dezavantajı ikili karşılaştırmalar sırasında basılgın bölümdeki çözümleri de hesaba katmasıdır ve ikili karşılaştırmalardan yine bir tümevarım yapmanın zor olmasıdır.

Bu konuda yazarlar, performans metrikleri tanımlamanın zor ve tek bir metrik üzerinden tüm kriterleri anlamlı olarak değerlendirmenin de imkansız olabildiğini ifade etmişlerdir. Özellikle performans karşılaştırmalarını istatistiksel bakış açısıyla

değerlendirmenin zor olduğunu, çoklu önem testlerine ve varyans analiziyle yapılacak çalışmalara hala ihtiyaç olduğunu belirtmektedirler.

Çalışmamızda bu ihtiyaçlar dikkate alınarak algoritmaları pareto-önyüzlerinin kalitesine göre karşılaştırabileceğimiz ve bu karşılaştırmaların tutarlılıklarını ölçebileceğimiz bir çoklu önem testi uygulaması gerçekleştirilmiştir. Bir yandan da ilk kez iki kriterli ÇAKAP problemleri çözülerek ÇAKKO algoritmaları karşılaştırılmaktadır. Ayrıca yine ilk kez bu karşılaştırmalar (İKKAP'ları ÇAKKO'larla çözerken) yapılmadan önce parametre eniyileme çalışmaları yapılarak (sabit düzenler almak yerine) daha adil bir performans değerlendirme yapılmaktadır.

İKKA algoritması için parametre düzeni belirleme çalışması yapılırken kullanılacak deney tasarımı tablo 4.1 ve tablo 4.2'de verilmişti. Diğer ÇAKKO'lar için de benzer tablolar oluşturulmuştur. En iyi parametre düzenini belirlemek maksadıyla M_1 , M_2 , M_3 ve C performans kriterleri devreye sokulabilirdi. Ancak daha kesin, güvenilir ve tutarlı sonuçlar elde etmek adına tablo 4.3'de görülen öncelik vektörleri hesabını (çoklu önem testi) kullanmaktayız.

Tablo 4.3: İKKA (BicriterionAnt) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	0,6667	0,6667	0,6667	0,6667	0,8333	0,6667	0,6667	1,1111	0,0831
D2	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D3	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D4	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D5	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D6	1,2	0,9167	0,9167	0,9167	0,9167	1	0,9167	0,9167	1,3333	0,1092
D7	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D8	1,5	1	1	1	1	1,0909	1	1	1,5	0,1213
D9	0,9	0,6667	0,6667	0,6667	0,6667	0,75	0,6667	0,6667	1	0,0802
Tutarlılık oranı: 0,0002										

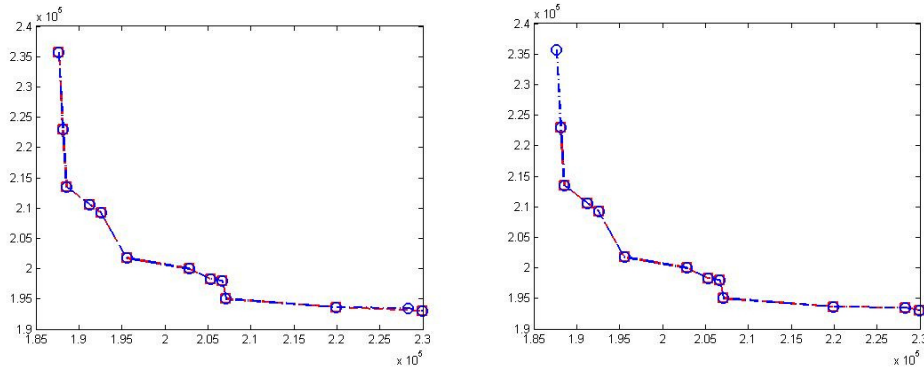
Tablo 4.3'de ve diğer algoritmalar için de oluşturulmuş benzer tablolarda deney düzenleri ikili olarak karşılaştırılmış, deney düzenlerinin önem dereceleri hesaplanmıştır. Önem derecesi belirlenmesinde ilgili satır elemanına ait deney düzeniyle elde edilen pareto-önyüzün, ilgili sütun elemanına ait deney düzeniyle elde

edilen pareto-önyüzle birleştirilmesi sonucu elde edilen pareto-önyüzler dikkate alınmıştır. İkili olarak birleştirilip ortaya çıkan pareto-önyüzde satır elemanına ait düzenden kaç karar seçeneği olduğu ve sütun elemanına ait düzenden kaç karar seçeneği olduğu hesaplatılmıştır. Daha sonra ilgili satıra ait deney düzeninin, birleştirilmiş pareto-önyüzdeki karar seçeneği sayısı ilgili sütuna ait deney düzenininkine bölünmüş ve satırı temsil eden deney düzeninin sütunu temsil eden deney düzenine önemi bu şekilde hesaplatılmıştır. Her ikili karşılaştırma tablosu için öncelik vektörleri ve tutarlılık oranları hesaplanmıştır. Benzer hesaplama diğer algoritmaların deney düzenlerinin seçiminde ve algoritmaların performanslarının karşılaştırılmasında da yapılmıştır.

Bu yöntemin en önemli avantajı ikili karşılaştırmalar yaparken ortaya çıkan basılgin karar seçeneklerini dikkate almaması, tamamen pareto-önyüzlerin birleştirilmesi sonucu ortaya çıkmış yeni pareto-önyüze odaklanmasıdır. Bu da önem derecesi belirlemede daha anlamlı olmaktadır. Bu yöntemin en büyük dezavantajı bir düzen diğerine tamamen baskın olduğunda, önem derecesinin 0'a bölme hatasından dolayı bu şekilde hesaplanamamasıdır. Dolayısıyla bunun için pareto-önyüzde ortalama en çok bulunan karar seçeneği sayısına göre varsayımsal bir değer alınmaktadır. Parametre düzen seçiminde kullanılan bu literatür problemi için pareto-önyüzde 10 civarı karar seçeneği ortalama olarak bulunduğundan, tamamen baskın olan düzenin önem derecesi 10, tamamen basılgin olan düzenin önem derecesi de 0 olarak alınmıştır. Hesaplamalar sonucu tutarsızlık tespit edilen matrislerde bu şekilde belirlenmiş olan önem derecesi tekrar ele alınarak bu dezavantaj ortadan kaldırılabilir. Yine de bu konu ilerideki çalışmalarda ele alınmaya değer bir konu olarak görülebilir.

İKKA için gürbüz parametre düzeninin belirlenmesi çalışmalarında ortaya çıkan tablo 4.3'e baktığımızda D2, D3, D4, D5, D7 ve D8 seçeneklerinin öncelik değerinin eşit ve en yüksek çıktığını görmekteyiz. Bunun nedeni bu düzenlerin hepsinin bu problem için pareto-optimumu tamamen bulmuş veya bir karar seçeneği eksik olarak çok yaklaşmış olmasındandır. Bu algoritmanın ilk ortaya atıldığı bilimsel çalışmasında önerilmiş olan D9 deney düzeninin iyi sonuç vermediği görülmektedir. Bunun sebebi buharlaştırma katsayısının feromon hesaplama yapısına göre çok

düşük verilmiş olmasıdır. Çünkü klasik KS'den farklı olarak İKKA'da, pareto-önyüzde sonuç bulan karınca sayısına göre (maliyete göre değil) bir hesaplama yapılmaktadır. Bundan dolayı zaten biraz yüksek olan feromon değerleri, buharlaştırma katsayısı da düşük verilmiş olduğunda makul sınırlarda kalamamakta, çözüm yerel optimumda tıkanmaktadır. Şekil 4.1'de görüldüğü üzere D2, uç noktalardan birini bulamadığı için tercih edilmemektedir. Ayrıca tutarlılık oranının çok iyi olması, deney düzenlerinin çoğunun aynı sonucu bulmasından kaynaklanmaktadır.



Şekil 4.1: İKKA parametre eniyileme çalışması, D8 (solda D3, D4, D5 ve D7 ile aynı) ve D2 (sağda) pareto-önyüzleri

İKKA-M algoritması, İKKA algoritmasının feromon güncelleme ifadesinin, pareto-önyüze çıkan karıncalara göre hesaplanan maliyetin bir fonksiyonu olarak hesaplanması sonucu ortaya çıkmıştır. Bu algortmada da diğerleri gibi, parametre eniyilemesi için her deneme 1 saat sürmek üzere toplam 3 deneme yapılmıştır. 3 deneme sonucu ortaya çıkan pareto-önyüzler birleştirilerek her deney düzenine ait pareto-önyüz hesaplatılmıştır. Karşılaştırmalar yine bu pareto-önyüzler üzerinden yukarıda ifade edildiği şekilde yapılmıştır. İKKA-M parametre eniyileştirilmesi için oluşturulmuş deney tasarımı tablo 4.4 ve tablo 4.5'te verildiği gibidir.

Tablo 4.4: İKKA-M (BicriterionAnt-C) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	80	100
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

Tablo 4.5: İKKA-M (BicriterionAnt-C) $2^{4-1} + 1$ kesirli deney tasarımı

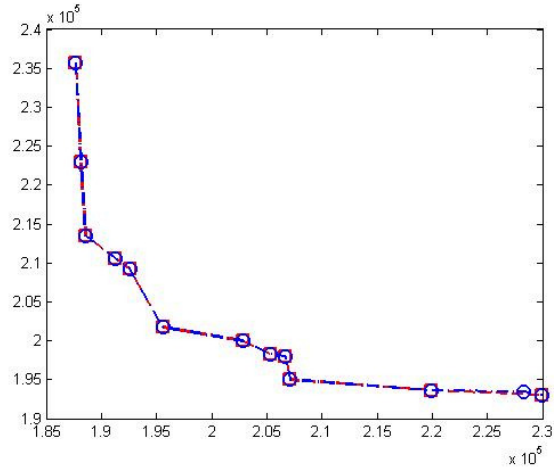
Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	100	1	1	0,02

İKKA-M deney tasarımı yapı itibariyle İKKA'nınkilerle aynıdır. Ancak hesapsal yapıları farklı olduğu için deney düzenlerinin öncelikleri biraz daha farklılaşmaktadır. Tablo 4.6'da görüldüğü üzere D1, diğer deney düzenlerinin önüne geçmiştir. Oysa İKKA için oldukça kötü sonuçlar vermekteydi. Bunun sebebi maliyetin bir fonksiyonu olarak ifade edilmiş olan feromon güncelleme ifadelerinin İKKA'ya göre daha az değerler alması, bununla birlikte buharlaştırma katsayısı, karınca sayısı ve feromonun öneminden en az birinin yüksek olmasının sonucu olumsuz etkilemesidir. Tutarlılık oranının biraz daha yükseldiğini ama hala oldukça iyi bir seviyede olduğunu söyleyebiliriz.

Tablo 4.6: İKKA-M (BicriterionAnt-C) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1,2	1,2	1,7143	1,7143	1,5	1,3333	1,5	1,5	0,1514
D2	0,8333	1	1	1,4286	1,4286	1,25	1,1111	1,25	1,1667	0,1252
D3	0,8333	1	1	1,5	1,625	1,3333	1,2	1,25	1,3333	0,1317
D4	0,5833	0,7	0,6667	1	1	1	0,6923	0,7273	0,8333	0,0859
D5	0,5833	0,7	0,6154	1	1	0,8182	0,6923	0,7273	0,8333	0,0832
D6	0,6667	0,8	0,75	1	1,2222	1	0,9	1	0,9	0,0986
D7	0,75	0,9	0,8333	1,4444	1,4444	1,1111	1	1,0909	1,0909	0,1146
D8	0,6667	0,8	0,8	1,375	1,375	1	0,9167	1	1	0,1057
D9	0,6667	0,8571	0,75	1,2	1,2	1,1111	0,9167	1	1	0,1037
Tutarlılık oranı: 0,0008										

İKKA-M algoritması için D1 deney düzeni seçilmesi uygun görülmektedir. Bu deneye ait pareto-önyüz şekil 4.2'de verilmektedir. Şekilden de görüleceği üzere tek karar seçeneği hariç tüm pareto-optimum küme D1 düzeniyle bulunmuştur.



Şekil 4.2: İKKA-M parametre eniyileme çalışması D1 pareto-önyüzü

BİKA'da da benzer şekilde parametre eniyilemesi yapılmıştır. BİKA algoritması temel olarak İKKA'yla aynı yapıdadır. İKKA'dan farklı olarak tüm karıncaları iki amaç ağırlıkları üzerinde homojen şekilde dağıtırken kolonilere ayırmaktadır. Her koloni kendi kolonisine ait feromonu (pareto-önyüzde sonuç bulan karıncalar tarafından) güncellemekte, böylelikle orjine göre güncelleme yapılması sağlanmaya çalışılmaktadır. Bölümlendirilmemiş diye adlandırılmasının sebebi ise bu güncellemenin koşullar sırasında ortaya çıkan pareto-önyüzdeki bölümleri dikkate almamasıdır. Bu durum dikkate alınarak aynı yazarlar tarafından İKÇKA önerilmiştir. BİKA için oluşturulan deney tasarım modeli tablo 4.7 ve 4.8'de verilmektedir.

Tablo 4.7: BİKA (UnsortBicriterion) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (her kolonideki karınca sayısı, koloni sayısı)	10, 10	20, 5
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

BİKA deneyleri için toplam 100 karınca alınmıştır. 1. seviyede 100 karınca 10 koloniye 10'arlı paylaştırılmış, 2. seviyede ise 100 karınca 5 koloniye 20'şerli paylaştırılmıştır.

Tablo 4.8: BİKA (UnsortBicriterion) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	10, 10	1	1	0,02

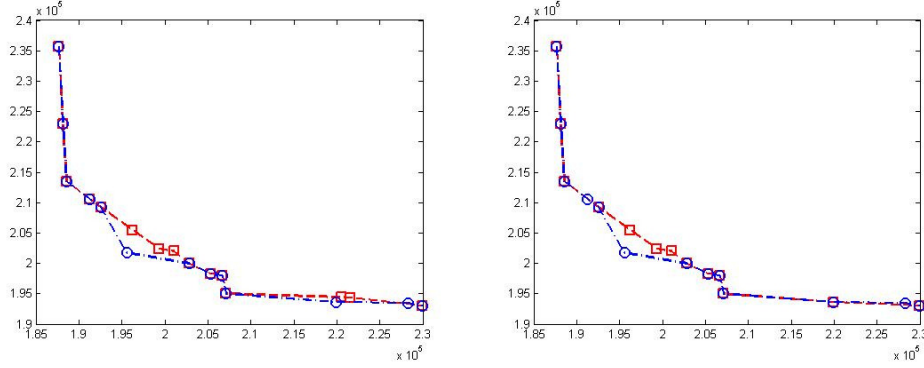
BİKA için deney sonuçları tablo 4.9’da verilmektedir. Bu sonuçlara göre en iyi deney düzeni D4’tür. D7 düzeni de D4’e çok yakın ancak ikinci öncelikli düzen olarak karşımıza çıkmaktadır. Bu sonuçlardan da anlayacağımız üzere, BİKA algoritması için de bilimsel çalışmasında belirlenmiş parametre düzeni İKKAP’lar için yetersizdir.

Tablo 4.9: BİKA (UnsortBicriterion) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	0,875	0,75	0,4615	1,3	0,75	0,4615	1	0,7143	0,0824
D2	1,1429	1	0,9	0,7	1,2222	0,8889	0,6	1,8	0,7273	0,1013
D3	1,3333	1,1111	1	0,6667	1,6364	1,2	0,6923	1,3077	0,9333	0,1126
D4	2,1667	1,4286	1,5	1	2,1429	1,4444	1	2,5	1	0,1588
D5	0,7692	0,8182	0,6111	0,4667	1	0,6667	0,4615	0,8	0,5333	0,0703
D6	1,3333	1,125	0,8333	0,6923	1,5	1	0,6923	1,5	0,7273	0,1061
D7	2,1667	1,6667	1,4444	1	2,1667	1,4444	1	2,1667	1	0,1582
D8	1	0,5556	0,7647	0,4	1,25	0,6667	0,4615	1	0,7143	0,0766
D9	1,4	1,375	1,0714	1	1,875	1,375	1	1,4	1	0,1339
Tutarlılık oranı: 0,0043										

BİKA için yapılmış parametre eniyileme çalışmalarından anlaşılacağı üzere, koloni sayısı az, kolonideki karınca sayısı çok tutuluyorsa feromonun önem parametresi yüksek, sezgisel bilginin önemi düşük ve buharlaştırma katsayısı da düşük (ancak D9’daki kadar da düşük olmamak kaydıyla) tutulsa iyi olur. Bu durumda en iyi sonucu alabilmekteyiz. Eğer koloni sayısı biraz daha artırılıp her kolonideki karınca sayısı azaltılıyorsa, yine feromonun öneminin yüksek ve buharlaştırma katsayısının düşük olmasında fayda var; ancak bu durumda sezgisel bilginin öneminin biraz

arttırılması gerekli olmaktadır. Bu deneyler için de tablo 4.8'den görüleceği üzere tutarlılık oranı oldukça kabul edilebilir seviyededir.



Şekil 4.3: BİKA parametre eniyileme çalışması D4(solda) ve D7 (sağda) pareto-önyüzü

Ancak D4 ve D7 öncelik değerlerinin yakınlığı ve tutarlılık oranının biraz daha artmış olmasından dolayı bu iki deney düzeni sonuçlarını şekil 4.3 üzerinden de son olarak değerlendirmekte fayda vardır. Görüldüğü üzere iki düzenin de birbirinden pek farkı yoktur. Pareto-optimum'da ikisi de aynı sayıda karar seçeneği bulmuştur. D4'ün biraz daha önemli olmasının sebebi ise D7'nin bulduğu pareto-optimum karar seçeneklerinden birisini bulamayıp 2 adet pareto-optimum olmayan (ama pareto-optimuma çok yakın) nokta bulması ve bu noktaların diğer düzenlere göre baskın sonuçlar vermesiyle bu düzenin öneminin biraz daha artmış (ki sadece 0,0006 kadar) olmasıdır. Sonuç olarak iki deney düzeni de kullanılabilir gözükmektedir. D7'nin dağılımı daha iyi olabileceğinden performans karşılaştırma deneylerinde bu düzen alınmasına karar verilmiştir.

BİKA-M da (İKKA'ya benzer şekilde) BİKA'nın feromon hesaplamalarında maliyet fonksiyonu kullanılmış şeklidir. Bundan dolayı parametre eniyileme çalışmalarında aynı deney tasarımı kullanılmıştır. Bu tasarım tablo 4.10 ve 4.11'de verilmektedir.

Tablo 4.10: BİKA-M (UnsortBicriterion-C) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (her kolonideki karınca sayısı, koloni sayısı)	10, 10	20, 5
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

Tablo 4.11: BİKA-M (UnsortBicriterion-C) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	10, 10	1	1	0,02

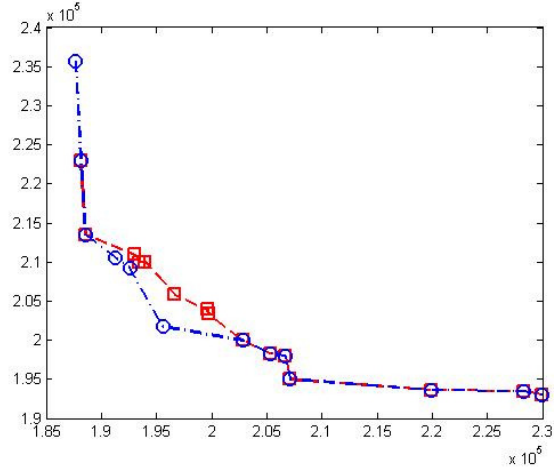
BİKA-M için parametre eniyileme sonuçları tablo 4.12’de verilmiştir. Bu tabloda da görüldüğü üzere yine oldukça iyi bir tutarlılık oranıyla öncelik vektörü hesaplanmıştır. Özellikle D9’un ona en yakın olan D1 ve D2’den ayrıldığını görmekteyiz. Bu da bize bu algoritmanın önerilen çalışmadaki parametre seviyeleriyle iyi sonuçlar verdiğini ifade etmektedir.

Tablo 4.12: BİKA-M (UnsortBicriterion-C) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1,1111	2,5	1,375	1,5	1,5	1,6667	1,5	1	0,1494
D2	0,9	1	1,8333	1,2857	1,1111	2,2	1,5	1,8333	1	0,1440
D3	0,4	0,5455	1	0,625	0,8182	0,6364	0,7	0,8	0,4545	0,0681
D4	0,7273	0,7778	1,6	1	1,25	1,6667	1,2857	1,25	0,6429	0,1142
D5	0,6667	0,9	1,2222	0,8	1	0,8462	0,8333	0,9091	0,5	0,0891
D6	0,6667	0,4545	1,5714	0,6	1,1818	1	1,2	1	0,4286	0,0887
D7	0,6	0,6667	1,4286	0,7778	1,2	0,8333	1	0,8182	0,5833	0,0897
D8	0,6667	0,5455	1,25	0,8	1,1	1	1,2222	1	0,5	0,0909
D9	1	1	2,2	1,5556	2	2,3333	1,7143	2	1	0,1658
Tutarlılık oranı: 0,0069										

BİKA’dan farklı olarak BİKA-M’da feromon değerlerinin daha küçük değerlerle artırılması bize bu sonucu getirmiştir. Ayrıca İKKA-M’dan da farklı olması her karıncanın kendi kolonisinde feromon güncellemesinden dolayıdır. Yani İKKA’ya göre tüm koloni kolonilere bölünmüş ve feromon güncellemesini de azaltmış durumdadır. Bu durumda feromon önemi düşük olsa da buharlaştırmanın çok az olması gerekmekte olduğunu sonuçlardan anlamaktayız. Çünkü bu algorithmada aslında tüm kolonilerin birleşiminden (sinerjisinden) oluşan bir feromon güncelleme

yoktur; karıncalar sadece görece kendi küçük kolonilerinde (ve görece küçük miktarlarda) güncelleme yapmakta ve dolayısıyla fazla feromon azaltılmaması pareto-önyüzde daha iyi sonuç bulunmasına neden olmaktadır. Ayrıca D9'a en yakın öncelik değerine sahip D1'in D9'a göre feromonun öneminin iki kat artırılmasıyla feromon buharlaştırmanın da artırılmış olması yukarıdaki fikirleri destekler niteliktedir. Şekil 4.4'te D9 düzenine ait pareto-önyüz verilmiştir.



Şekil 4.4: BİKA-M parametre eniyileme çalışması D9 pareto-önyüzü

BİKA ve BİKA-M algoritmalarıyla bulunan çözümlerde pareto-önyüzdeki boşluklar dikkat çekicidir. Bu algoritmayı ileri süren yazarlar da bunu fark etmiş ve basılgın olmayan önyüzdeki bölgelere göre feromon güncelleyen bir algoritma tasarlamışlardır. İKÇKA (BicriterionMC) adını verdikleri algoritma BİKA'dan farklı bir amaç ağırlıklandırma sistemine sahiptir. Bu sistemle karıncalar yine kolonilere ayrılmakta ancak her koloni pareto-önyüzün belli bir bölgesinde çalışmaktadır. Pareto-önyüz kolonilere eşit olarak bölünmektedir ve yine her kolonideki karınca kendi kolonisinde feromon güncellemektedir. Yazarlar bu algoritmanın BİKA ve İKKA'dan daha iyi sonuç verdiğini ifade etmişlerdir. Ancak yapılan deneyler belirli bir atama sayısı üzerinden sınırlandırılmış, algoritmaların çalışma süreleri dikkate alınmamıştır. Şu bir gerçektir ki İKÇKA'daki hesap yükü diğerlerine göre daha ağırdır ve pareto-önyüzde diğerlerine göre çok sıkı bir tarama gerçekleştirmektedir. Bundan dolayı belirli bir sürede yapabileceği atama sayısı BİKA ve İKKA'dan daha azdır. Ancak atama sayısı ile denemeleri sınırlandırdığımızda ise her koşumun diğerlerinden daha uzun süreli olacağı kesindir. Bu çok adaletli görünmemekte ve

son zamanlardaki yönsemeleri de dikkate alarak (diğer algoritmalarda da yaptığımız gibi) deneyleri süre ile kısıtlandırmayı uygun görmekteyiz. İKÇKA'nın ağır ama iyi sonuç vermesinden ziyade belki de İKKA hafif hesap yüküyle daha çeşitli sonuçlar bulma şansına sahip olduğu için zaman kısıtlı çalıştırıldığında İKÇKA'dan daha iyi sonuç verebilir.

Tablo 4.13: İKÇKA (BicriterionMC) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (her kolonideki karınca sayısı, koloni sayısı)	10, 10	20, 5
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

İKÇKA parametreleri BİKA'ninkilerle aynıdır. Çünkü aralarındaki tek farklılık kolonilerin pareo-önyüzde görevlendirilme ve feromon güncelleme hesapsal yapılarıdır. İKÇKA parametre eniyileme çalışmasında kullanılan deney tasarım modeli tablo 4.13 ve tablo 4.14'te verilmiştir ve BİKA ile aynıdır.

Tablo 4.14: İKÇKA (BicriterionMC) $2^{4-1} + 1$ kesirli deney tasarımı

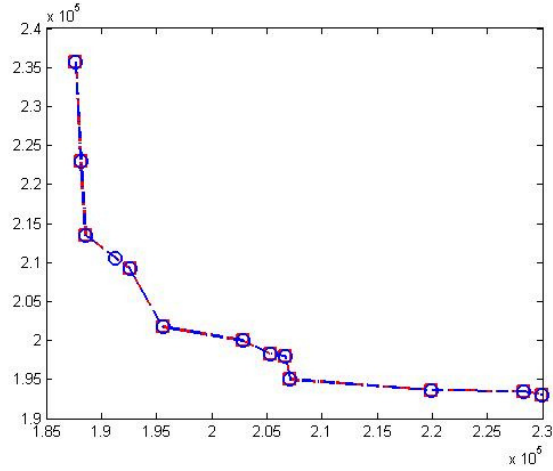
Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	10, 10	1	1	0,02

İKÇKA parametre eniyileme deneyleri sonuçları tablo 4.15'te verilmiştir. D1 deney düzeninin diğerlerinden önemli bir farkla önde olduğu açıktır. D9 düzeni ikinci olarak BİKA-M'dakine benzer bir yapı ortaya çıkmıştır. Tutarlılık oranı oldukça makul seviyededir. Buharlaştırma katsayısının çok düşük alınmasına gerek kalmamıştır. Dikkate değer bir nokta da yüksek buharlaştırma katsayısının iyi sonuç vermemesinin yanında, yüksek feromon önemi ve sezgisel bilgi önemi ile de iyi sonuçlar elde edilememiştir.

Tablo 4.15: İKÇKA (BicriterionMC) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1,7143	2	2	4	1,5	1,3333	6	1,2	0,1937
D2	0,5833	1	1	0,8889	1,8571	0,8182	0,75	1,5	0,7	0,0953
D3	0,5	1	1	0,7857	1,7273	0,9091	0,7692	2,7143	0,5833	0,0965
D4	0,5	1,125	1,2727	1	3,25	1	0,8462	4,3333	0,6667	0,1220
D5	0,25	0,5385	0,5789	0,3077	1	0,3333	0,2857	0,9357	0,3077	0,0444
D6	0,6667	1,2222	1,1	1	3	1	0,8571	2,4	0,75	0,1165
D7	0,75	1,3333	1,3	1,1818	3,5	1,1667	1	2,8	0,9167	0,1354
D8	0,1667	0,6667	0,3684	0,2308	1,0667	0,4167	0,3571	1	0,3077	0,0437
D9	0,8333	1,4286	1,7143	1,5	3,25	1,3333	1,0909	3,25	1	0,1524
Tutarlılık oranı: 0,0082										

Şekil 4.5'te D1'e ait pareto-önyüzü görmekteyiz. Bir karar seçeneği hariç diğer tüm pareto-optimum karar seçenekleri bulunmuştur. Bulunan seçenekler arasında göz ardı edilemeyecek kadar çok boşluk bulunmamaktadır. Bu algoritma için D1 düzenini kullanmakta fayda görülmektedir.



Şekil 4.5: İKÇKA parametre eniyileme çalışması D1 pareto-önyüzü

İKKA'da ve BİKA'da olduğu gibi İKÇKA'da da feromon güncellemesinde maliyet fonksiyonu kullansaydık sonuçlar nasıl olurdu ve hangi deney düzeninde deneyler yapmaya karar verirdik? Bu sorunun cevabını araştırmak için İKÇKA-M ismini verdiğimiz algoritmayı kodladık. Bu algoritmanın deneylerinin tasarımı tablo 4.16 ve tablo 4.17'de verilmiştir.

Tablo 4.16: İKÇKA-M (BicriterionMC-C) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (her kolonideki karınca sayısı, koloni sayısı)	10, 10	20, 5
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1.5
F4 (buharlaştırma katsayısı)	0,25	0,75

İKÇKA-M'ın İKÇKA'dan farkı feromon güncelleme hesabındadır. Feromon güncellerken pareto-önyüzde sonuç bulan karınca sayısı yerine bu karıncaların buldukları çözümlerin maliyetinin dikkate alınmaktadır.

Tablo 4.17: İKÇKA-M (BicriterionMC-C) $2^{4-1} + 1$ kesirli deney tasarımı

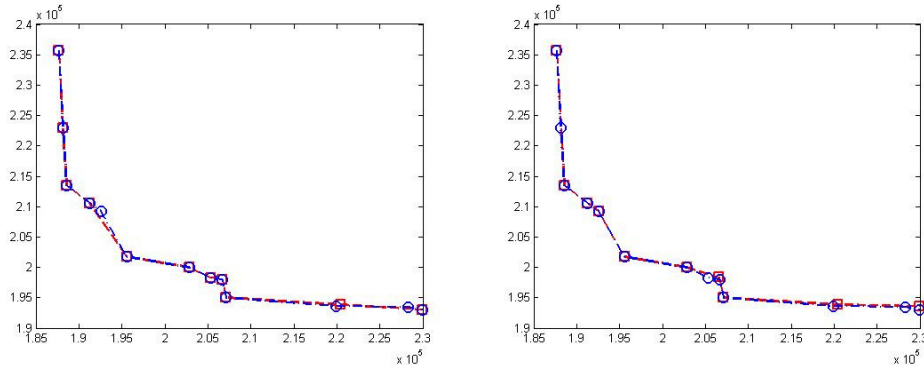
Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	10, 10	1	1	0,02

İKÇKA-M parametre eniyileme deney sonuçları tablo 4.18'de rapor edilmiştir. Deney sonuçlarından anlaşılacağı üzere en iyi düzen D9'dur. D9'dan sonra yine D1 ve ondan sonra da D2 öncelikli olarak görülmektedir.

Tablo 4.18: İKÇKA-M (BicriterionMC-C) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1	1,6	2,5	2,5	1,6667	10	1,75	0,7273	0,1706
D2	1	1	1,625	2,1667	4,3333	1,125	7	1,625	0,8	0,1625
D3	0,625	0,6154	1	0,9	0,9375	0,7778	1	0,6	0,6	0,0778
D4	0,4	0,4615	1,1111	1	1,6667	0,5556	2,75	0,75	0,2727	0,0725
D5	0,4	0,2308	1,0667	0,6	1	0,5	1,3077	0,2143	0,2727	0,0503
D6	0,6	0,8889	1,2857	1,8	2	1	5	1,1429	0,6364	0,1185
D7	0	0,1429	1	0,3636	0,7647	0,2	1	0,2667	0	0,0272
D8	0,5714	0,6154	1,6667	1,3333	4,6667	0,875	3,75	1	0,6	0,1191
D9	1,375	1,25	1,6667	3,6667	3,6667	1,5714	10	1,6667	1	0,2016
Tutarlılık oranı: 0,0197										

Bu algoritma için de feromonun önemi, sezgisel bilginin önemi ve buharlaştırma katsayısı parametrelerinin yüksek olmaması gerektiği kanısına varılmaktadır. Tutarlılık oranı İKKA ve BİKA'lara göre biraz daha yükselmiştir, ancak hala iyi bir düzeydedir. D1'in D7'ye ve D9'un da D7'ye tamamen baskın olduğu görülmektedir. Tamamen baskın olduğunda 10 kat önemliymiş gibi değerlendirilecekti. D7'nin kötü bir çözüm olması, çok sayıda koloni üretilerek her kolonide olması gerekenden daha az karınca görevlendirildiğinde, feromonun önemi ve sezgisel bilginin öneminin yüksek verilmesi, buharlaştırma katsayısının da düşük verilmiş olmasının karıncalarımızı yerel optimuma saptamasından kaynaklanmaktadır. Tutarlılık biraz daha düştüğünden dolayı şekil 4.6'da hem D1 hem de D9'a ait pareto-önyüzlere yer verilmiştir. Açıkça anlaşılacağı üzere D9 pareto-optimumda 10 karar seçeneği bulurken D1 sadece 7 adet bulabilmiştir. D9 düzeninde deneyler yapmak oldukça mantıklı görülmektedir.



Şekil 4.6: İKÇKA-M parametre eniyileme çalışması D9 (solda) ve D1 (sağda) pareto-önyüzü

EKÇAS (CompetAnts) algoritması klasik KS üzerinde temellenmiş olmakla birlikte en önemli özelliği karıncaları gruplama şeklidir. Karıncaları ilk olarak iki ayrı gruba ayırmakta ve atamalar yapıldıkça daha başarılı olan koloniye daha çok karınca vermektedir. Her grubun içerisinde ise diğer gruptan bilgi sızdıran casuslar mevcuttur. Casus karıncalar sayesinde pareto-önyüzün özellikle orta noktasına odaklanılmış olmakta, casus olmayanlar ise ilgili amacı eniyilemeye çalışarak uçlardaki karar seçeneklerini bulmaya çalışmaktadırlar. EKÇAS parametre eniyileme çalışmaları için tasarlanmış olan deney tasarımı tablo 4.19 ve 4.20'de verilmektedir. Buharlaştırma katsayısı yüksek değerleri için algoritma çok verimsiz çalıştığından ötürü 2. seviyesi 0,5 alınmıştır.

Tablo 4.19: EKÇAS (CompetAnts) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı toplamı)	80	100
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,5

EKÇAS'ı öneren yazarlar makalede 10 boyutlu problem için 160 gibi oldukça yüksek sayıda karınca kullanmaktadırlar. D9'da bu deney düzeni test edilmektedir.

Tablo 4.20: EKÇAS (CompetAnts) $2^{4-1} + 1$ kesirli deney tasarımı

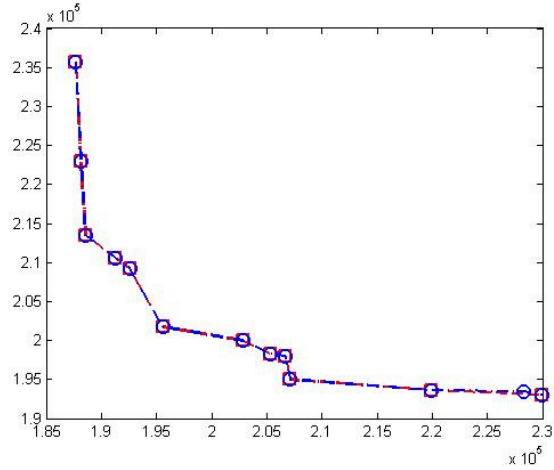
Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	160	1	1	0,5

EKÇAS parametre eniyileme çalışmalarının sonuçları tablo 4.21'de verilmektedir. D3 deney düzeninin diğerlerinden farklı şekilde iyi sonuç verdiği görülmektedir. Bu sonuçlar da göstermektedir ki, D9'da alınmış olan yüksek miktarda karınca sayısı çözüme olumlu yansımamıştır. Tutarlılık oranının da çok iyi sonuç vermiş olması deney düzen seçiminde tamamen D3'e yönelinmesini sağlamaktadır.

Tablo 4.21: EKÇAS (CompetAnts) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	0,7	0,5833	0,7	1,1111	0,9167	0,7	0,8333	1	0,0894
D2	1,4286	1	0,8333	1,1111	1,6667	1,1111	1	1	1,5714	0,1262
D3	1,7143	1,2	1	1,3333	2	1,3333	1,2	1,2	2	0,1525
D4	1,4286	0,9	0,75	1	1,5	1	0,9	0,9	1,25	0,1135
D5	0,9	0,6	0,5	0,6667	1	1,0909	0,6	0,6667	1,1111	0,0835
D6	1,0909	0,9	0,75	1	0,9167	1	0,9	0,8462	1,0833	0,1026
D7	1,4286	1	0,8333	1,1111	1,6667	1,1111	1	1	1,6667	0,1271
D8	1,2	1	0,8333	1,1111	1,5	1,1818	1	1	1,4444	0,1221
D9	1	0,6364	0,5	0,8	0,9	0,9231	0,6	0,6923	1	0,0832
Tutarlılık oranı: 0,0025										

EKÇAS algoritmasının D3 deney düzeniyle elde edilmiş olan, diğer düzenlere göre en iyi pareto-önyüzü şekil 4.7’de verilmiştir. Görüldüğü üzere bir nokta hariç pareto-optimumdaki tüm noktalar bu düzenle bulunmuştur.



Şekil 4.7: EKÇAS parametre eniyileme çalışması D3 pareto-önyüzü

KKOÇAŞ (MONACO) algoritması da EKÇAS, İKKA, İKKA-M, BİKA, BİKA-M, İKÇKA ve İKÇKA-M algoritmaları gibi klasik KS temellidir. Bunlardan en önemli farkları feromon güncelleme hesaplamalarının da klasik KS’deki gibi olması, tekleştirilmiş sezgisel bilgiye sahip olması ve iki amaç için ayrı ayrı hesaplanan feromonların öneminin amaç ağırlıklandırılmasında sabit olarak kullanılmasıdır. Yani iki amaç için iki farklı feromon matrisi vardır ve ağırlıkları algoritma çalıştırılmadan belirlenmektedir. KKOÇAŞ parametrelerinin eniyilenmesi amacıyla hazırlanmış deney tasarım modeli tablo 4.22 ve tablo 4.23’te verilmektedir.

Tablo 4.22: KKOÇAŞ (MONACO) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı toplamı)	80	100
F2 (birinci, ikinci amaç için feromonun önemi)	2, 2	4, 4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

KKOÇAŞ algoritması çalıştırılırken feromon güncelleme sabiti 10 olarak alınmış ve değiştirilmemiştir. Literatür problemlerimiz içinde amaç ağırlıklandırmaları farklı olan örnekler olmadığı için (iki amaç da aynı ağırlıkta ele alınması gerektiği için) F2 faktörü iki amaç için de aynı olmak üzere tek bir faktör olarak ele alınmıştır.

Tablo 4.23: KKOÇAŞ (MONACO) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	100	1, 1	1	0,5

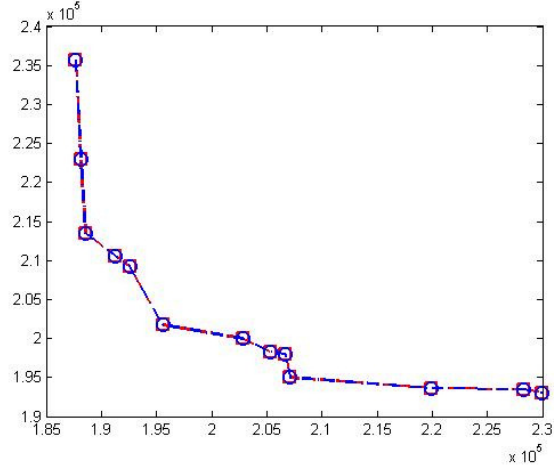
KKOÇAŞ algoritmasının deney düzen seçimi için yapılan koşullar sonucunda ortaya çıkan öncelik vektörü ve tutarlılık oranı tablo 4.24'te verilmektedir. D3 deney düzeni EKÇAS'takine benzer olarak en iyi çözümü veren seçenek olmuştur. 80 karınca ile çözüm yaparken feromonun öneminin sezgisel bilgininkinden 4 kat fazla olması ve yüksek buharlaştırma katsayısı ile farklı çözümler üretebilme yeteneğinin artmış olmasından dolayı iyi sonuç vermektedir. Az karınca sayısı, düşük feromon önemi ile yüksek sezgisel bilgi ve yüksek buharlaştırma katsayısının oluşturduğu deney düzeninin (D5) kötü sonuç vermesi oldukça olağan karşılanabilir.

Tablo 4.24: KKOÇAŞ (MONACO) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1	0,9231	1,0909	2	1	1,3333	1,5	1,5	0,1321
D2	1	1	0,9231	1,0909	1,625	1	1,3333	1,4444	1,3	0,1267
D3	1,0833	1,0833	1	1,1818	2,1667	1,0833	1,4444	1,625	1,625	0,1431
D4	0,9167	0,9167	0,8462	1	1,8333	0,9167	1,2222	1,3333	1,375	0,1207
D5	0,5	0,6154	0,4615	0,5455	1	0,5	0,6667	1	0,8	0,0706
D6	1	1	0,9231	1,0909	2	1	1,3333	1,5	1,5	0,1321
D7	0,75	0,75	0,6923	0,8182	1,5	0,75	1	1,125	1,25	0,1003
D8	0,6667	0,6923	0,6154	0,75	1	0,6667	0,8889	1	0,9	0,0852
D9	0,6667	0,7692	0,6154	0,7273	1,25	0,6667	0,8	1,1111	1	0,0890
Tutarlılık oranı: 0,0011										

KKOÇAŞ algoritması ile yapılan bu deneylerde, bundan önce bahsedilmiş olan algoritmalarından farklı olarak, pareto-optimum'un karar seçeneklerinin 12'si de şekil 4.8'de görüldüğü üzere D3 deney düzeni ile bulunmuştur. D1 ve D6 alternatifleri ise

sadece bir noktayı bulamadıkları için D3'ün öneminden biraz daha düşük öncelik değerine sahip olmuşlardır.



Şekil 4.8: KKOÇAŞ parametre eniyileme çalışması D3 pareto-önyüzü

ÇKKS (MOACS) buraya kadar deneyleri analiz edilmiş tüm algoritmalarından farklı olarak klasik KKS temelli bir algoritmadır. Tek feromon matrisi ve iki sezgisel bilgi matrisi kullanmaktadır. Bu sezgisel bilgi matrisleri arasındaki amaç ağırlıklandırması İKKA'dakine benzer şekilde (ancak İKKA'da feromon matrislerinde yapılıyordu) hesaplanmaktadır. Bu algorithmada ilginç olan bir nokta da feromonun önemi parametresinin bulunmayışı (ya da 1 olması olarak da düşünülebilir), tamamen sezgisel bilginin önemi dikkate alınarak pareto-önyüzde amaç ağırlıklandırması yapılmasıdır. Bundan dolayı feromonun önemi deney tasarımlarımızda yer almamakta, bunun yerini araştırma-devam katsayısı diye ifade edilen klasik KKS'ye özgü bir parametre almaktadır. ÇKKS için parametre eniyileme çalışmalarında kullanılan deney seviyeleri ve deseni tablo 4.25 ve tablo 4.26'da verilmektedir.

Tablo 4.25: ÇKKS (MOACS) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı toplamı)	80	100
F2 (sezgisel bilginin önemi)	1	1,5
F3 (buharlaştırma katsayısı)	0,25	0,75
F4 (araştırma-devam katsayısı)	0,25	0,75

ÇKKS algoritması için dikkat edilmesi gerekli bir nokta da önerildiği çalışmada verilen parametrelerin bizim denemeyi önerdiklerimizden oldukça farklı oluşudur. Deneysel çalışma sonuçları tablo 4.27'de verilmektedir.

Tablo 4.26: ÇKKS (MOACS) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	10	1	0,1	0,9

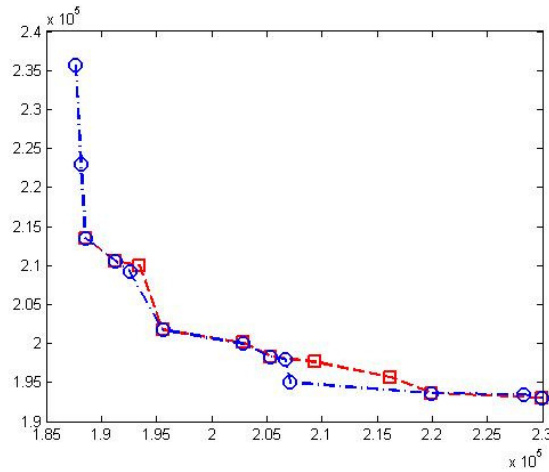
ÇKKS parametre eniyileme çalışması sonuçlarında görülmektedir ki, bu algoritma için en iyi deney düzeni D2'dir ve diğerlerinden oldukça ayrılmış durumdadır. D2 düzeninin D1 ve D7'ye tamamen baskın, D4 ve D6 düzenlerinden de çok önemli olduğu gözlemlenmiştir. Tutarlılık oranı da oldukça iyi olduğundan dolayı koşulsuz en iyi parametre düzeni olarak D2 seçilebilir.

Tablo 4.27: ÇKKS (MOACS) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	0	0,2667	1,25	0,25	0,9167	0,9091	0,2222	0,2857	0,0309
D2	10	1	1,25	10	1,6667	10	10	1,2857	1,125	0,2420
D3	3,75	0,8	1	4,6667	1,5	3	3,75	0,9167	0,8571	0,1418
D4	0,8	0,1	0,2143	1	0,1111	0,8333	0,7692	0,1765	0,1538	0,0249
D5	4	0,6	0,6667	9	1	9	10	0,6667	0,6667	0,1615
D6	1,0909	0,1	0,3333	1,2	0,1111	1	0,9286	0,1667	0,2143	0,0306
D7	1,1	0	0,2667	1,3	0	1,0769	1	0,1111	0,1429	0,0234
D8	4,5	0,7778	1,0909	5,6667	1,5	6	9	1	1,1667	0,1788
D9	3,5	0,8889	1,1667	6,5	1,5	4,6667	7	0,8571	1	0,1661
Tutarlılık oranı: 0,0019										

ÇKKS'nin klasik KKS tabanlı yapısından dolayı global feromon güncellemenin yanında bir de yerel feromon güncelleme yapmasının, sonuç üzerinde farklı çözümleri üretmek maksatlı bir eğilim olduğu açıktır. D2 deney düzeni sayesinde düşük buharlaştırma katsayısı ve yüksek araştırma-devam katsayısı ile elde edilen çözümlerin korunmaktadır. Algoritma bütün ÇAKKO'lardan farklı bir feromon güncelleme hesapsal yapısına sahiptir.. Algoritmanın global feromon güncelleme mantığı, ortalamada daha iyi bir pareto-önyüz bulduğunda feromon değerlerini

başlangıç değerlerinden başlatarak pareto-önyüzün farklı noktalarının araştırılmasını desteklemektedir. Daha iyi bir pareto-önyüz bulamadığında ise karıncalar klasik KKS'de de bulunan global feromon güncellemeyi yapmaktadır. Özetle ÇKKS algoritması bu feromon güncelleme mantığından ve ayrıca feromon formülasyonunun çarpımlı yapısından dolayı (üstelik yerel feromon güncelleme de var olmasına rağmen) farklı noktaları araştırmaya yönelmekte, bir de buharlaştırma katsayısı veya araştırma-devam katsayısı sonuçları çeşitlendirmeyi arttıracak yönde belirlendiğinde, elde edilen iyi çözümler üzerinde devamlılık sağlanması yeterince mümkün olamamaktadır. Dolayısıyla bu düşüncelere en uygun olan D2 düzeni, şekil 4.9'da görüldüğü üzere en iyi deney düzeni olarak karşımıza çıkmaktadır. Diğer algoritmalara göre oldukça kötü sayılabilir bir sonuç görülmektedir. Pareto-optimumda sadece 6 nokta bulabilmiştir.



Şekil 4.9: ÇKKS parametre eniyileme çalışması D2 pareto-önyüzü

PKKOA algoritması da ÇKKS'ye benzer şekilde klasik KKS temelli bir algoritmadır. İki amacın birleştirilmesiyle oluşan tek bir sezgisel bilgi matrisi ve iki feromon matrisi bulunmaktadır. Amaç ağırlıklandırma her karınca için feromon matrislerinde ve rassal olarak yapılmaktadır. Klasik KKS'deki gibi yerel ve global feromon güncellemelere sahiptir, ancak kendine özgü bir feromon ekleme yapısı vardır. Bu yapı sayısal değer itibariyle yüksektir ve önceki bölümde detaylı olarak açıklanmaktadır. Bu yapının olumsuz etkileyip etkilemediğini test etmek amacıyla PKKOA-M algoritması geliştirilmiştir. PKKOA algoritması yukarıdaki algoritmalara benzer olarak koşulduğunda çok yüksek feromon değerlerine ulaşıldığından dolayı

program hesapsal sınırlarını aşmaktadır. Bundan dolayı deneysel tasarım sadece ilgili bilimsel çalışmada önerilen sınırlar içinde yapılmıştır. 5 faktöre sahip PKKOA algoritması parametre eniyilemesi için deney tasarımı tablo 4.28 ve tablo 4.29’da verilmiştir.

Tablo 4.28: PKKOA (P-ACO) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	10	15
F2 (feromonun önemi)	1	1,5
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,05	0,1
F5 (araştırma-devam katsayısı)	0,2	0,4

PKKOA algoritması koşullarında 5 faktör 8 deney düzeninde test edilmiştir. Çalışma sonuçları tablo 4.30’da verilmektedir.

Tablo 4.29: PKKOA (P-ACO) 2^{5-2} kesirli deney tasarımı

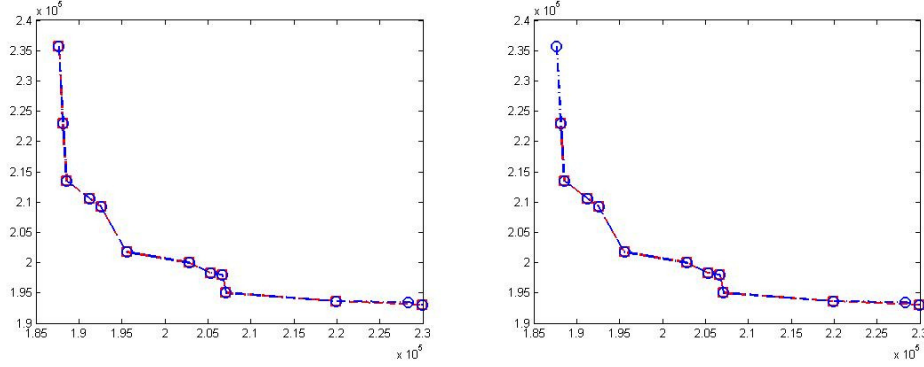
Seviye	F1	F2	F3	F4	F5
D1	1	1	1	2	2
D2	2	1	1	1	1
D3	1	2	1	1	2
D4	2	2	1	2	1
D5	1	1	2	2	1
D6	2	1	2	1	2
D7	1	2	2	1	1
D8	2	2	2	2	2

Tablo 4.30’da da görüldüğü üzere en iyi deney düzeni D7 olmuş, ikinci en iyi düzen ise D5 olmuştur. İki deney düzeninde de az sayıda karınca olması dikkat çekicidir.

Tablo 4.30: PKKOA (P-ACO) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	Ö.V.
D1	1	1	1	2,75	0,9091	1,8333	0,8333	2,75	0,1529
D2	1	1	1	2,75	0,9091	1,8333	0,8333	2,75	0,1529
D3	1	1	1	2,5	0,9091	1,6667	0,8333	3,4	0,1541
D4	0,3636	0,3636	0,4	1	0,3636	0,8333	0,3333	0,625	0,0567
D5	1,1	1,1	1,1	2,75	1	1,8333	0,9167	2,75	0,1623
D6	0,5455	0,5455	0,6	1,2	0,5455	1	0,5	1	0,0807
D7	1,2	1,2	1,2	3	1,0909	2	1	3	0,1771
D8	0,3636	0,3636	0,2941	1,6	0,3636	1	0,3333	1	0,0634
Tutarlılık oranı: 0,0053									

Tutarlılık oranı iyi çıkmasına rağmen, diğer algoritmalarından oldukça farklı olarak birbirine çok yakın seviyeler seçildiğinden dolayı D7 ve D5 deney düzenlerinin pareto-önyüzlerini incelemekte fayda vardır. Şekil 4.10'da D7 ve D5 düzenlerine ait birleştirilmiş pareto-önyüzler pareto optimumla birlikte görülmektedir. D7 düzeniyle D5 düzeninden bir nokta daha fazla bulma imkanı olduğu görülmektedir. Dolayısıyla PKKOA için D7 düzenini seçmekte fayda vardır.



Şekil 4.10: PKKOA parametre eniyileme çalışması D7 (solda) ve D5 (sağda) pareto-önyüzü
PKKOA-M algoritması PKKOA'daki yüksek feromon değerlerinin oluşturabileceği dezavantajları yok etmek amacıyla, global feromon güncelleme ifadesinin, amaçların maliyetlerinin birer fonksiyonu olarak hesaplanmasından ibarettir.

Tablo 4.31: PKKOA-M (P-ACO-C) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	10	15
F2 (feromonun önemi)	1	1,5
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,05	0,1
F5 (araştırma-devam katsayısı)	0,2	0,4

Tablo 4.32: PKKOA-M (P-ACO-C) 2^{5-2} kesirli deney tasarımı

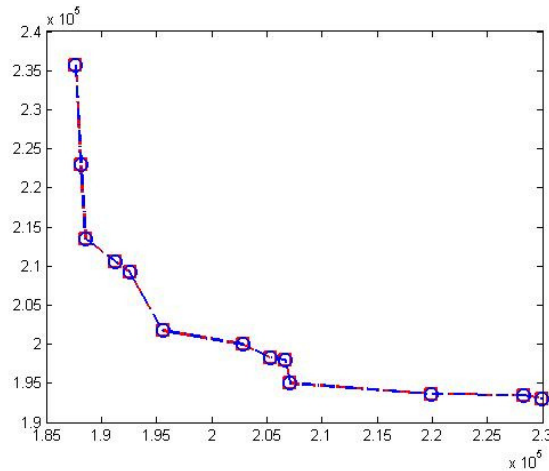
Seviye	F1	F2	F3	F4	F5
D1	1	1	1	2	2
D2	2	1	1	1	1
D3	1	2	1	1	2
D4	2	2	1	2	1
D5	1	1	2	2	1
D6	2	1	2	1	2
D7	1	2	2	1	1
D8	2	2	2	2	2

Deney düzeni belirleme maksadıyla oluşturulan tasarımlar tablo 4.31 ve 4.32’de verilmektedir. Deneyler sonucunda ortaya çıkan öncelik değerleri tablo 4.33’de gösterilmiştir.

Tablo 4.33: PKKOA-M (P-ACO-C) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	Ö.V.
D1	1	0,9231	1,2	2,4	1	4	1	1,7143	0,1619
D2	1,0833	1	1,3	2,6	1,0833	4,3333	1,0833	1,8571	0,1754
D3	0,8333	0,7692	1	2	0,8333	3,6667	0,8333	1,5714	0,1383
D4	0,4167	0,3846	0,5	1	0,4167	1,4	0,4167	0,7143	0,0661
D5	1	0,9231	1,2	2,4	1	4	1	1,7143	0,1619
D6	0,25	0,2308	0,2727	0,7143	0,25	1	0,25	0,4286	0,0410
D7	1	0,9231	1,2	2,4	1	4	1	1,7143	0,1619
D8	0,5833	0,5385	0,6364	1,4	0,5833	2,3333	0,5833	1	0,0934
Tutarlılık oranı: 0,0005									

D2 düzeni kendisine en yakın önceliğe sahip D1, D5 ve D7 desenlerinden az da olsa ayrılmaktadır. Tutarlılık oranının da oldukça düşük olması, bu düzenlerin pareto-optimumda oldukça başarılı olduğunu göstermektedir. Şekil 4.11’de de görüldüğü üzere D2 düzeniyle tüm pareto-optimum karar seçenekleri bulunmuştur. D1, D5 ve D7 ise D2’den sadece bir adet eksik olarak karar seçeneklerini bulmuştur. Genel anlamda bakarsak PKKOA’daki olağandan yüksek feromon güncelleme değerleri yerine, PKKOA-M’da feromonların maliyetin fonksiyonu ile güncelleştirilmesi bu örnek üzerinde olumlu sonuç vermiştir.



Şekil 4.11: PKKOA-M parametre eniyileme çalışması D2 pareto-önyüzü

ÇAKKO algoritmalarının parametre eniyileme çalışmalarında son olarak ÇAAQA'yı (MOAQ) ele almaktayız. Bu klasik KKS temelli algoritmanın en önemli özelliği global feromon güncelleme kuralıdır. Bu kurala, Q-öğrenme adıyla literatürde anılan algoritmayı uyarlamışlardır. Bunun sonucunda feromon bilgisinin (geciktirilmesiyle) iyi çözümlerin daha iyi öğrenilmesi sağlanmaktadır. ÇAAQA'da her amaç için bir karınca ailesi atanır ve klasik KKS kuralları çalıştırılmaktadır. Ancak klasik KKS'den tek farkı araştırma-devam katsayısının sabit olmamasıdır. Algoritmanın bu özelliği, koşumlar sırasında başlarda iyi bir araştırmayı, sonlara doğru ise iyi çözümlerde devamlılığı sağlamaya çalışmaktadır. ÇAAQA'ya ait parametre eniyilemesi amaçlı deneylerin tasarımı tablo 4.34 ve 4.35'te verilmektedir.

Tablo 4.34: ÇAAQA (MOAQ) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	80	100
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75
F5 (araştırma-devam katsayısı başlangıç değeri; araştırma devam katsayısı azaltma sabiti)	0,99; 0,9	0,999; 0,99
F6 (Q-öğrenme indirim faktörü)	0,3	0,4

Tablo 4.35: ÇAAQA (MOAQ) $2^{6-3} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4	F5	F6
D1	1	1	1	2	2	2
D2	2	1	1	1	1	2
D3	1	2	1	1	2	1
D4	2	2	1	2	1	1
D5	1	1	2	2	1	1
D6	2	1	2	1	2	1
D7	1	2	2	1	1	2
D8	2	2	2	2	2	2
D9	10	1	1	0,1	0,99; 0,9	0,3

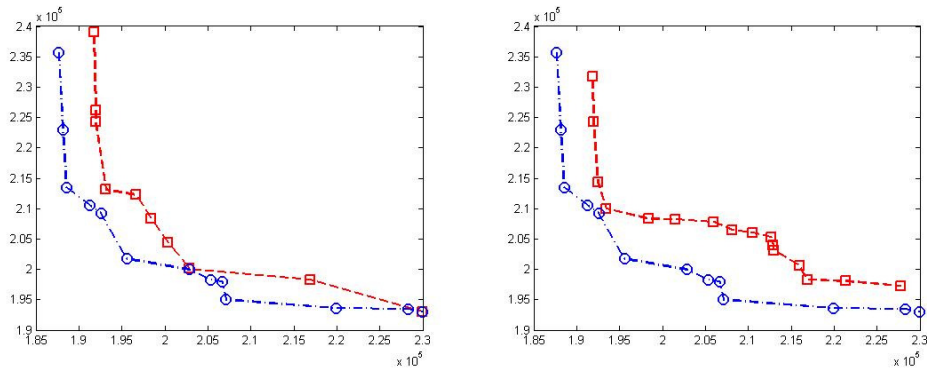
ÇAAQA'da diğer algoritmalarından farklı olarak 6 faktör bulunmaktadır ve özellikle F5 çok tartışılabilir. Bunun nedeni ise bu faktörün sadece belirli bir oranda değişmesidir. Bu değişimin atama sayısına bağlı olmaması veya bias bir değişken olmaması itibarıyla, algoritmanın koşumunun büyük kısmında çok küçük bir değer almış olacak ve sonuçlar hep devamlılık yönünde yerel optimuma saplanmış şekilde

çıkacaktır. Tablo 4.36’da verilen sonuçlarda F5 için 2.seviye sonuçlarının iyi çıkması bu düşüncelerimizi doğrular niteliktedir.

Tablo 4.36: ÇAAQA (MOAQ) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	3	0,8	7	10	1	10	1	10	0,1994
D2	0,3333	1	0,2	1,6667	2	0,1875	1,6	0,25	10	0,0592
D3	1,25	5	1	5	10	1,125	10	1,25	10	0,2221
D4	0,1429	0,6	0,2	1	2	0,1538	1	0,25	7	0,0430
D5	0,1	0,5	0,1	0,5	1	0,0625	0,8333	0,1111	6	0,0272
D6	1	5,3333	0,8889	6,5	16	1	10	1,125	10	0,2285
D7	0	0,625	0,1	1	1,2	0	1	0	10	0,0304
D8	1	4	0,8	4	9	0,8889	10	1	10	0,1876
D9	0	0	0	0,1429	0,1667	0	0,1	0	1	0,0027
Tutarlılık oranı:										

D2, D4, D5, D7 ve D9 düzenlerinin diğer düzenlere göre önem derecelerinin çok düşük olmasının temel sebebi tamamen F5 değerinin bu düzenlerde yeterince iyi belirlenmemiş olmasıdır. Aslında bu konudaki temel çukur F5 hesapsal yapısından kaynaklanmaktadır. F5 için daha iyi değerleri araştırmak çok uzun süre alıcı ve zahmetli olacağından, ilerideki çalışmalar kapsamına bu görevi görececek bir formülasyon arayışının yerleştirilmesi daha uygun görülmektedir.



Şekil 4.12: ÇAAQA parametre eniyileme çalışması D3 (solda) ve D6 (sağda) pareto-ön yüzü

Tablo 4.36’deki tutarlılık oranı anlamsız çıkmaktadır, çünkü diğerlerine hemen hemen tamamen basılgın olan seçeneklerden 5 tanesi, diğer 4’ünden oldukça kötüdür. Bu yüzden aslında karar verme seçeneği olarak değerlendirilmemesi gereken D2, D4, D5, D7 ve D9 değerleri tablodan çıkarılarak, öncelik vektörü ve

tutarlılık oranının daha anlamlı olması için geriye kalan 4 düzen dikkate alınarak tablo 4.37 oluşturulmuştur. Bu düzeltilmiş tablodan görüldüğü üzere, tablo 4.36'daki D3 düzeni ile D6 düzeni arasındaki küçük fark, D3 lehine daha büyüterek karşımıza çıkmaktadır. Şekil 4.12'ye bakıldığında D3 düzeninin iyi bir seçenek olduğu görülmektedir. Ancak değişken atama-devam katsayısı hesapsal yapısından ve pareto-önyüzü dikkate almayan feromon güncelleme kuralından ötürü diğer ÇAKKO'lara göre en kötü sonucu elde etmiş olduğu açıktır.

Tablo 4.37: ÇAAQA (MOAQ) parametre düzen seçimi (düzeltilmiş)

Ö.D.	D1	D3	D6	D8	Ö.V.
D1	1	0,8	1	1	0,2356
D3	1,25	1	1,125	1,25	0,2867
D6	1	0,8889	1	1,125	0,2491
D8	1	0,8	0,8889	1	0,2287
Tutarlılık oranı: 0,0004					

Bu çalışmada ele alınan algoritmaların 12'sinin de parametre eniyileme çalışmaları tamamlandıktan sonra, algoritmalar kendi aralarında karşılaştırılmıştır. Karşılaştırmalar deney düzen seçiminde de yapıldığı gibi öncelik vektörü ve tutarlılık oranı hesaplarına dayanmaktadır. KC10-2fl-1uni literatür probleminde yapılan çalışmalar ışığında, yine 10 büyüklüğündeki KC10-2fl-3uni örneği ile 20 büyüklükteki KC20-2fl-1uni ve KC20-2fl-3uni literatür örnekleri çözümlenerek ÇAKKO algoritmalarının performansları karşılaştırılmaktadır. Bunun için oluşturulacak tabloların büyüklüklerinin makul boyutlarda olması amacıyla tablo 4.38'deki kodlama sistemi gösterge olarak alınmaktadır.

Tablo 4.38: ÇAKKO algoritmaları kod göstergesi

ÇAKKO KISALTMASI	KOD	ÇAKKO KISALTMASI	KOD
İKKA	A1	EKÇAS	A7
İKKA-M	A2	KKOÇAŞ	A8
BİKA	A3	ÇKKS	A9
BİKA-M	A4	PKKOA	A10
İKÇKA	A5	PKKOA-M	A11
İKÇKA-M	A6	ÇAAQA	A12

KC10-2fl-1uni örneği için algoritmaların elde ettikleri performans sonuçları tablo 4.39’da karşılaştırılmaktadır. Tablodaki değerlerin ilgili hücrelere sığması için virgülden sonra iki basamak alınmıştır. Tutarlılık oranının da oldukça iyi olduğu görülmektedir.

Tablo 4.39: KC10-2fl-1uni literatür probleminde ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	1	1,2	1,33	1	1,2	1	0,92	2	1	0,92	12	0,0978
A2	1	1	1,2	1,33	1	1,2	1	0,92	2	1	0,92	12	0,0978
A3	0,83	0,83	1	1,44	0,83	1	0,83	0,77	1,67	0,83	0,77	13	0,0854
A4	0,75	0,75	0,69	1	0,75	1,1	0,75	0,69	1,57	0,75	0,69	15	0,0773
A5	1	1	1,2	1,33	1	1,2	1	0,92	2	1	0,92	12	0,0978
A6	0,83	0,83	1	0,91	0,83	1	0,83	0,77	1,43	0,83	0,77	11	0,0799
A7	1	1	1,2	1,33	1	1,2	1	0,92	2	1	0,92	12	0,0978
A8	1,08	1,08	1,3	1,44	1,08	1,3	1,08	1	2,17	1,08	1	13	0,1059
A9	0,5	0,5	0,6	0,64	0,5	0,7	0,5	0,46	1	0,5	0,46	5	0,0487
A10	1	1	1,2	1,33	1	1,2	1	0,92	2	1	0,92	12	0,0978
A11	1,08	1,08	1,3	1,44	1,08	1,3	1,08	1	2,17	1,08	1	13	0,1059
A12	0,08	0,08	0,08	0,07	0,08	0,09	0,08	0,08	0,2	0,08	0,08	1	0,0078
Tutarlılık oranı: 0,0022													

Tablo 4.39’den da görülmektedir ki bu örnekte en iyi sonuç veren algoritmalar A8 (KKOÇAS) ve A11 (PKKOA-M)’dir. Bu algoritmalarla şekil 4.8 ve şekil 4.11’de görüldüğü gibi, pareto optimumdaki tüm karar seçenekleri gürbüz parametre düzeni ile bulunabilmiştir. A1(İKKA), A2(İKKA-M), A5 (İKÇKA), A7(EKÇAS) ve A10 (PKKOA) algoritmaları ise 2.sırada görülmektedir. Bu algoritmaların hepsi pareto-optimumda sadece birer karar seçeneğini bulamamışlardır. Sonra A3 (BİKA), A6 (İKÇKA-M), A4 (BİKA-M), A9 (ÇKKS) ve A12 (ÇAAQA) algoritmaları sıralanmaktadır. ÇAAQA’nın değişken atama-devam katsayısı hesaplaması gerçekten bu algoritmayı oldukça dezavantajlı duruma düşürmektedir. ÇKKS’de ise amaç ağırlıklandırmasının sezgisel bilgi üzerinden yapılması ve feromonun önem parametresinin dikkate alınmayışından dolayı son sıralarda yer aldığı düşünülebilir. BİKA ve BİKA-M için ise bu örneğin çok parçalı oluşunun dezavantaj yarattığı açıktır. Çünkü bu algoritmalar (adlarında da belirtildiği gibi) kolonileri pareto-önyüzde bölümlendirmezler, her koloni feromon bilgisini kendi içinde değerlendirir.

Koloni sayısı belirli bir miktarda tutulacağı ve karıncaların arasında var olan haberleşme, koloniler arasında olmadığı için, bu koloniler çok parçalı büyük bir pareto-önyüzde bir anlamda başarılı olmak için çaresiz kalmış durumdadırlar. Ama ürettikleri sonucun çok kötü olduğunu söylemek mümkün değildir. Özellikle BİKA bu çaresizliği, feromon miktarını daha yüksek düzeyde (BİKA-M'a göre) ekleyerek bir anlamda aşmaya çalışmaktadır. BİKA sonuçları bundan dolayıdır ki bu örnekte BİKA-M'dan daha iyi çıkmaktadır. İKÇKA-M da İKÇKA'ya göre daha kötü sonuç vermiştir. Benzer şeyler bu durum için de söylenebilir. Yalnız bu algoritmalarda karıncalar bölümlendirildikleri ve hesapsal yapı bazı bölümlerde oldukça yoğun arama sağladığı için BİKA'lara göre biraz daha iyi sonuç vermişlerdir.

KC10-2fl-3uni literatür örneği KC10-2fl-1uni'ye göre oldukça farklı bir yapıdadır. Pareto-optimumunda 130 sıkı dağılmış karar seçeneği vardır ve KC10-2fl-1uni'deki parçalı yapı bu örnekte yoktur. Belirlediğimiz gürbüz parametre düzenleriyle, algoritmaları, KC10-2fl-3uni örneğinde, her biri 180 sn. olmak üzere 3 tekrarlı olarak çalıştırdık. Elde edilen sonuçlar ikili karşılaştırmalar yoluyla değerlendirilmiş, algoritmaların ikili önem değerleri, öncelik vektörü ve tutarlılık oranı tablo 4.40'da verilmiştir.

Tablo 4.40: KC10-2fl-3uni literatür probleminde ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	1,46	1,89	1,26	1,56	1,32	0,94	1,16	1,5	0,96	1,04	2,52	0,1068
A2	0,69	1	1,36	1,03	1,1	1,09	0,65	0,89	1,15	0,72	0,89	1,75	0,0796
A3	0,53	0,74	1	0,63	0,74	0,89	0,6	0,72	0,94	0,59	0,58	1,55	0,0612
A4	0,79	0,97	1,58	1	1,08	1,15	0,73	0,93	1,33	0,61	0,78	1,6	0,0808
A5	0,65	0,91	1,34	0,93	1	0,91	0,58	1,02	1,11	0,52	0,59	1,71	0,0716
A6	0,75	0,92	1,13	0,87	1,1	1	0,91	0,8	1,21	0,65	0,82	1,7	0,0773
A7	1,06	1,55	1,68	1,38	1,72	1,1	1	1,15	1,54	0,72	1,04	2,19	0,1040
A8	0,87	1,12	1,38	1,08	0,98	1,25	0,87	1	1,15	0,79	0,97	1,38	0,0853
A9	0,67	0,87	1,07	0,75	0,9	0,83	0,65	0,87	1	0,53	0,61	1,63	0,0667
A10	1,04	1,39	1,69	1,65	1,91	1,53	1,39	1,27	1,87	1	1,29	2,48	0,1205
A11	0,97	1,13	1,72	1,29	1,68	1,22	0,97	1,03	1,64	0,78	1	2,2	0,1002
A12	0,4	0,57	0,64	0,63	0,59	0,59	0,46	0,72	0,61	0,4	0,45	1	0,0461
Tutarlılık oranı: 0,0031													

Pareto-optimum'unda çok karar seçeneği bulunan KC10-2fl-3uni literatür örneği için en iyi algoritma, tablo 4.40'da görüldüğü üzere A10 (PKKOA) algoritmasıdır. Amaç ağırlıklandırmasının tesadüf sel ve feromon matrisi üzerinden olması, diğer algoritmalar dan daha yoğun feromon stratejisine sahip olması sayesinde birinci olduğu düşünülebilir. İkinci, üçüncü ve dördüncü sırayı alan algoritmalar (aralarında öncelik değeri olarak çok büyük fark olmamasına rağmen) sırasıyla A1 (İKKA), A7 (EKÇAS) ve A11 (PKKOA-M)'dir. Bu algoritmalar önceki örnekte de başarılı olan algoritmalar dır. PKKOA-M, KC10-2fl-1uni'de birinci olmuşken, bu örnekte dördüncü; EKÇAS ve İKKA ise KC10-2fl-1uni'de ikinciliği paylaşmışken, bu örnekte ikinci ve üçüncü olmuşlardır. Diğer algoritmalar dan öncelik değeri itibariyle biraz daha ayrı olarak başarılı görülmektedirler. Sonraki algoritmalar öncelik sırasıyla, A8 (KKOÇAS), A4 (BİKA-M), A2(İKKA-M), A6 (İKÇKA-M), A5 (İKÇKA), A9 (ÇKKS), A3 (BİKA) ve A12 (ÇAAQA)'dir. Özellikle ÇKKS ve ÇAAQA'nın yine son sıraları alması dikkat çekicidir. Bu örnekte de ÇKKS'nin özellikle pareto-önyüzde çözüm bulunduğunda feromonları başlangıç değerinden başlatma stratejisi ve ÇAAQA'nın önceki örnekte de bahsettiğimiz hesapsal dezavantajları, bu algoritmaları en başarısızlar arasına tekrar koymuştur. BİKA ve BİKA-M algoritmaları yine başarısız algoritmalar olarak karşımıza çıkmıştır. İKÇKA ve İKÇKA-M algoritmaları bu örnekte önceki örneğe göre biraz daha başarısız görülmektedir. Bunun sebebi bu algoritmaların amaç ağırlıklandırma formülasyonlarının özellikle pareto-önyüzün orta noktasına odaklanmış olmasıdır. Bu örnekte çok sayıda karar seçeneği düzgün dağılmış olduğundan bu algoritmaların orta noktadaki başarısı genel başarıya yansımamıştır.

KC20-2fl-1uni literatür örneği de KC10-2fl-1uni gibi çok parçalı bir pareto-önyüze sahip olmakla birlikte, KC10-2fl-1uni'den farklı olarak 20 boyutlu bir problemdir ve henüz pareto-optimum'u bilinmemektedir. KC20-2fl-1uni literatür örneği de bu çalışmada belirlenmiş gürbüz parametre düzenleriyle ve 10 boyutlu örneklerden farklı olarak her tekrar 360 sn. çalıştırılmıştır. Koşumlar sonucu oluşan önem değerleri, öncelik vektörü ve tutarlılık oranı tablo 4.41'de verilmektedir. Tutarlılık oranının 10 boyutlu örneklere göre kabul edilebilir bir seviyede olmasına rağmen biraz daha yükselmiş olması, koşumlar sonucu ortaya çıkan pareto-önyüzlerin, pareto-optimum gibi sabit ve belirli bir ön-yüze ulaşamayacak olması ile her

algoritmanın 2 amaçlı uzayda birbirinden çok farklı karar seçenekleri bulabilmesinden kaynaklandığı düşünülmektedir.

Tablo 4.41: KC20-2fl-1uni literatür probleminde ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	4,67	5	8,5	7,5	1,86	3	4	5,33	1,33	1,5	5	0,2118
A2	0,21	1	1,43	1	1,29	0,4	0,9	1,25	1,44	0,23	0,7	1,6	0,0502
A3	0,2	0,7	1	0,71	1,5	0,4	0,8	0,88	0,67	0,14	0,38	1,5	0,0388
A4	0,12	1	1,4	1	1,13	0,4	0,6	0,7	0,6	0,15	0,4	4	0,0438
A5	0,13	0,78	0,67	0,89	1	0,36	0,56	1	0,5	0,23	0,3	1,5	0,0350
A6	0,54	2,5	2,5	2,5	2,75	1	0,86	2	3,67	0,46	1,11	3,67	0,1003
A7	0,33	1,11	1,25	1,67	1,8	1,17	1	1,22	2,25	0,38	0,86	1,67	0,0691
A8	0,25	0,8	1,14	1,43	1	0,5	0,82	1	0,9	0,14	0,53	1,83	0,0452
A9	0,19	0,69	1,5	1,67	2	0,27	0,44	1,11	1	0,08	0,77	1,5	0,0445
A10	0,75	4,33	7	6,5	4,33	2,17	2,6	7	13	1	3,25	7	0,2407
A11	0,67	1,43	2,6	2,5	3,33	0,9	1,17	1,88	1,3	0,31	1	4,5	0,0929
A12	0,2	0,63	0,67	0,25	0,67	0,27	0,6	0,55	0,67	0,14	0,22	1	0,0276
Tutarlılık oranı: 0,0286													

KC20-2fl-1uni örneğinde yapılan koşumlar sonucu (KC10-2fl-3uni'de de olduğu gibi) en başarılı algoritma A10 (PKKOA) olmuş, ikinci sırayı A1 (İKKA) almıştır. KC10-2fl-3uni örneğinden farklı olarak bu örnekte bu ilk iki en iyi algoritma diğer algoritmalarından oldukça ayrılmıştır. A6 (İKÇKA-M) algoritmasının üçüncü olması, göreceli olarak diğerlerine göre pareto-önyüzün ortasında daha iyi bir arama gerçekleştirmesi ve bu örnekte bir pareto-optimum olmamasından dolayı orjine yakın, kaliteli karar seçenekleri bulmasına bağlanabilir; ancak bu başarı en iyi algoritma olması için yeterli olmamıştır. A11 (PKKOA-M) algoritması üçüncüye çok yakın olarak dördüncü olmuştur. Bu algoritma bu örnekte de iyi bir konum elde etmiştir. Sonra sırasıyla A7 (EKÇAS), A2 (İKKA-M), A8 (KKOÇAS), A9 (ÇKKS), A4 (BİKA-M), A3 (BİKA), A5 (İKÇKA) ve A12 (ÇAAQA) gelmektedir. Görüldüğü üzere ÇAAQA, ÇKKS, BİKA ve BİKA-M algoritmaları diğer örneklerde olduğu gibi son sıraları paylaşmaktadır.

KC20-2fl-3uni literatür örneği, KC10-2fl-3uni'ye benzer şekilde, pareto-önyüzde çok sayıda karar seçeneği bulunabilen ve pareto-önyüzü parçalı olmayan yapıda bir

örneklerdir. KC20-2fl-1uni'de olduğu gibi bu örnekte de pareto-optimum bilinmemektedir. 20 boyutlu olan bu örnek de 360 sn. ve 3 tekrarlı olarak çözülmüştür. Tablo 4.42'de KC20-2fl-3uni örneğinin ÇAKKO algoritmalarıyla çözülmesi sonucu hesaplanan algoritmaların ikili önem değerleri, öncelik vektörü ve tutarlılık oranı verilmektedir. Bu matrisin de oldukça tutarlı olduğu görülmektedir.

Tablo 4.42: KC20-2fl-3uni literatür probleminde ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	1,33	1,7	1	1,42	1,79	0,77	1,09	1,88	0,91	1,03	1,33	0,1001
A2	0,75	1	1,32	0,7	1	1,15	0,82	1,22	1	0,68	0,73	1,43	0,0776
A3	0,59	0,76	1	1	0,8	0,88	0,59	1,04	1,07	0,52	0,58	1,32	0,0661
A4	1	1,42	1	1	1,23	1,3	0,51	0,97	0,97	0,65	0,74	1,38	0,0800
A5	0,7	1	1,25	0,81	1	1,07	0,77	1,12	1,04	0,77	0,73	1,52	0,0777
A6	0,56	0,87	1,14	0,77	0,94	1	0,57	0,93	1,06	0,49	0,57	1,03	0,0646
A7	1,3	1,22	1,68	1,95	1,3	1,76	1	1,26	1,42	0,58	1	1,43	0,1044
A8	0,92	0,82	0,96	1,03	0,9	1,07	0,79	1	1,33	0,79	0,78	1,29	0,0781
A9	0,53	1	0,93	1,03	0,96	0,94	0,7	0,75	1	0,72	0,59	1,32	0,0691
A10	1,1	1,48	1,94	1,55	1,29	2,05	1,73	1,26	1,38	1	1,14	1,71	0,1173
A11	0,97	1,36	1,74	1,35	1,36	1,75	1	1,28	1,7	0,88	1	1,63	0,1050
A12	0,75	0,7	0,76	0,73	0,66	0,97	0,7	0,78	0,76	0,58	0,61	1	0,0602
Tutarlılık oranı: 0,0068													

Tablo 4.42'den de görüldüğü üzere bu örnekte de A10 (PKKOA) en başarılı algoritmadır. A11 (PKKOA-M), A7 (EKÇAS) ve A1 (İKKA), sırasıyla ikinci, üçüncü ve dördüncü olmuş, birbirlerine çok yakın öncelik değerleri almışlardır. Sonraki algoritmalar öncelik sırasına göre A4 (BİKA-M), A8 (KKOÇAŞ), A5 (İKÇKA), A2 (İKKA-M), A9 (ÇKKS), A3 (BİKA), A6 (İKÇKA-M) ve A12 (ÇAAQA) olarak sayılabilir. Bu örnekle birlikte, EKÇAS algoritmasının bu tür örneklerde daha başarılı olduğu, çok parçalı pareto-önyüzlü yapılarda daha az başarılı olduğunu söyleyebiliriz. BİKA ve İKÇKA-M algoritmaları için de benzer şeyleri söylemek mümkündür. ÇAAQA ve ÇKKS yine en son sıralarda yer almaktadır.

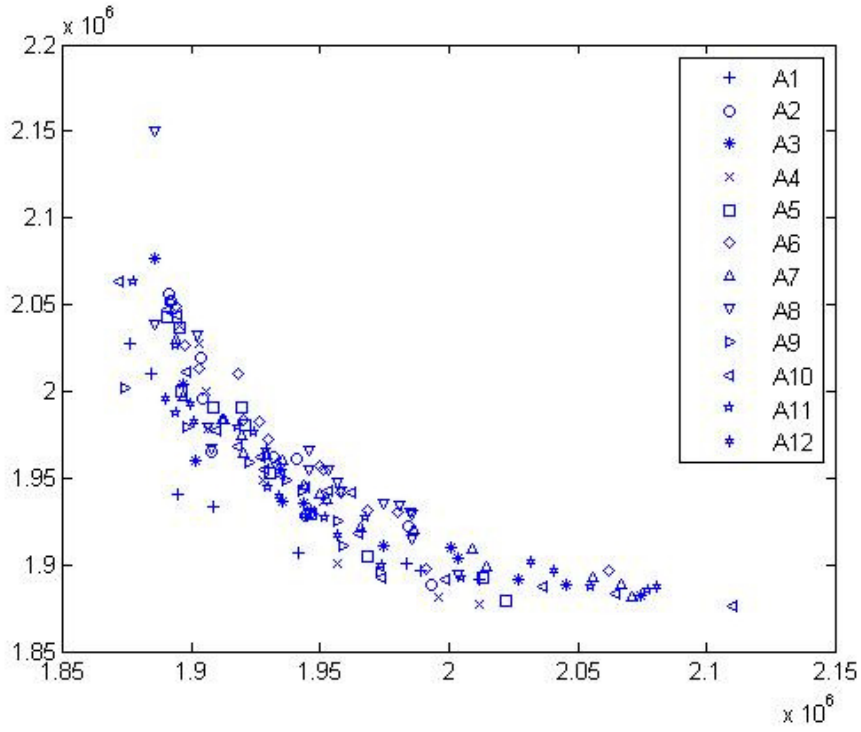
Literatür problemlerinin dışında, rassal üretilmiş 30, 40 ve 50 boyutlu toplam 3 problem daha ÇAKKO algoritmaları ile çözümlenip, çalışmada ortaya konulmuş kriterle

değerlendirilmiştir. 30 boyutlu örnek için her algoritma, 3 tekrarlı olarak ve her tekrar 540 sn. çalıştırılmıştır. 30 boyutlu rassal üretilmiş iki kriterli örnek probleme ait sonuçlar tablo 4.43'te görülmektedir.

Tablo 4.43: 30 boyutlu rassal örnekte ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	7	2,67	2	4	10	4	8	2	0,83	1,4	4	0,2020
A2	0,14	1	0,5	1	0,67	1,6	0,6	1,6	0,38	0,31	0,56	0,67	0,0430
A3	0,38	2	1	1,2	0,75	3,67	1,71	6	1,14	0,64	0,6	2,2	0,0918
A4	0,5	1	0,83	1	1,25	2	0,86	1,8	0,83	0,8	0,63	1,5	0,0700
A5	0,25	1,5	1,33	0,8	1	3,67	0,89	1,4	0,75	0,67	0,33	1,2	0,0638
A6	0	0,63	0,27	0,5	0,27	1	0,13	0,7	0,11	0,11	0,07	0,57	0,0176
A7	0,25	1,67	0,58	1,17	1,13	8	1	2	0,67	0,5	0,9	1,5	0,0737
A8	0,13	0,63	0,17	0,56	0,71	1,43	0,5	1	0,33	0,21	0,33	1	0,0321
A9	0,5	2,67	0,88	1,2	1,33	9	1,5	3	1	0,9	0,88	1,5	0,0981
A10	1,2	3,25	1,57	1,25	1,5	9	2	4,67	1,11	1	1,71	2	0,1361
A11	0,71	1,8	1,67	1,6	3	14	1,11	3	1,14	0,58	1	3	0,1241
A12	0,25	1,5	0,45	0,67	0,83	1,75	0,67	1	0,67	0,5	0,33	1	0,0478
Tutarlılık oranı: 0,0266													

Bu örnekte A1 (İKKA) en iyi sonucu veren algoritma olmuştur. Sonra sırasıyla A10 (PKKOA), A11 (PKKOA-M), A9 (ÇKKS), A3 (BİKA), A7 (EKÇAS), A4 (BİKA-M), A5 (İKÇKA), A12 (ÇAAQA), A2 (İKKA-M), A8 (KKOÇAŞ) ve A6 (İKÇKA-M) görülmektedir. Tutarlılık oranının arttığını (tutarsızlığın arttığını) ancak kabul edilir seviyelerde olduğunu görmekteyiz. Grafik üzerinden sezgisel olarak sıralama yapmanın zorluğu Şekil 4.13'te de görülmektedir. Orjine en yakın noktaları İKKA'nın bulduğu şekilden de görülmektedir. ÇKKS, ÇAAQA ve BİKA algoritmalarının başarısının artması bu örnek için dikkat çekicidir. İlk dörtte yine İKKA, PKKOA ve PKKOA-M algoritmalarının yerlerini almaları dikkat çekicidir.



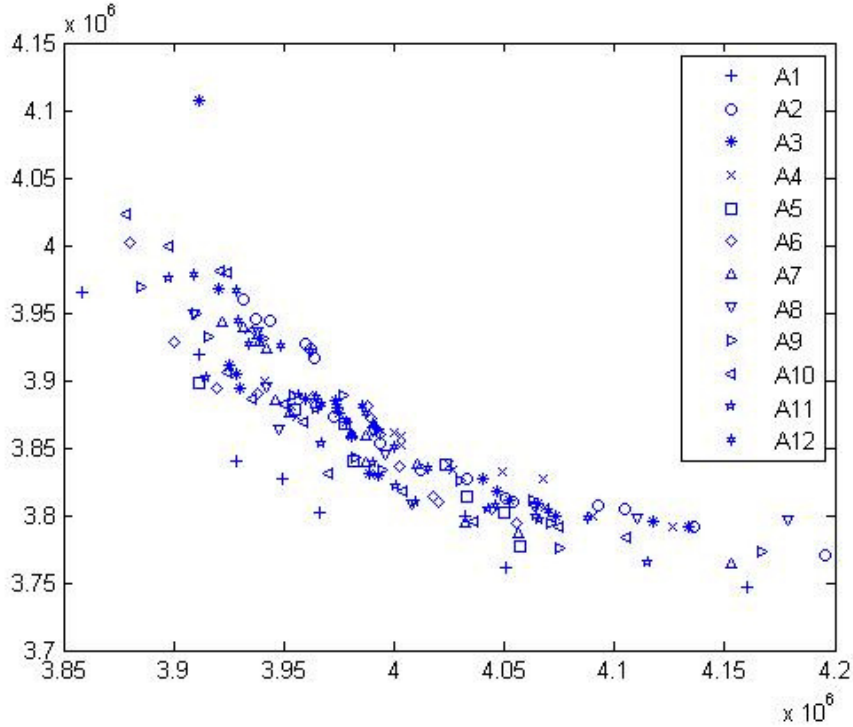
Şekil 4.13: 30 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü

40 boyutlu rassal örnek için her algoritma, 3 tekrarlı olarak ve her tekrar 720 sn. çalıştırılmıştır. 40 boyutlu rassal üretilmiş iki kriterli örnek probleme ait sonuçlar tablo 4.44'te görülmektedir. Tutarlılık oranı kabul edilebilir seviyededir.

Tablo 4.44: 40 boyutlu rassal örnekte ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	10	4	10	7	4	8	8	8	4	8	10	0,3478
A2	0	1	0,38	1,13	0,43	0,58	0,14	0,5	0,55	0,08	0	0,75	0,0179
A3	0,25	2,67	1	3,75	0,86	1,25	1,43	1,8	1,4	0,36	0,36	3	0,0668
A4	0	0,89	0,27	1	0,5	0,27	0,31	0,5	0,45	0,15	0,3	0,6	0,0180
A5	0,14	2,33	1,17	2	1	0,67	0,5	1	1	0,5	0,71	2,33	0,0501
A6	0,25	1,71	0,8	3,67	1,5	1	1	1,6	1,5	0,6	0,6	1,71	0,0644
A7	0,13	7	0,7	3,25	2	1	1	2,75	1,29	0,6	0,86	4	0,0773
A8	0,13	2	0,56	2	1	0,63	0,36	1	0,56	0,33	0,44	1,6	0,0385
A9	0,13	1,83	0,71	2,2	1	0,67	0,78	1,8	1	0,67	0,64	1,67	0,0507
A10	0,25	13	2,75	6,5	2	1,67	1,67	3	1,5	1	1,5	11	0,1369
A11	0,13	10	2,75	3,33	1,4	1,67	1,17	2,25	1,57	0,67	1	12	0,1098
A12	0	1,33	0,33	1,67	0,43	0,58	0,25	0,63	0,6	0,09	0,08	1	0,0216
Tutarlılık oranı: 0,017													

Tablo 4.44'ten de anlaşılacağı üzere en iyi algoritma A1 (İKKA) olarak görülmektedir. Sonrasında sırasıyla A10 (PKKOA), A11 (PKKOA-M), A7 (EKÇAS), A3 (BİKA), A6 (İKÇKA-M), A9 (ÇKKS), A5 (İKÇKA), A8 (KKOÇAS), A12 (ÇAAQA), A4 (BİKA-M) ve A2 (İKKA-M) gelmektedir. Şekil 4.14'te İKKA'nın başarısı açıkça görülmektedir.



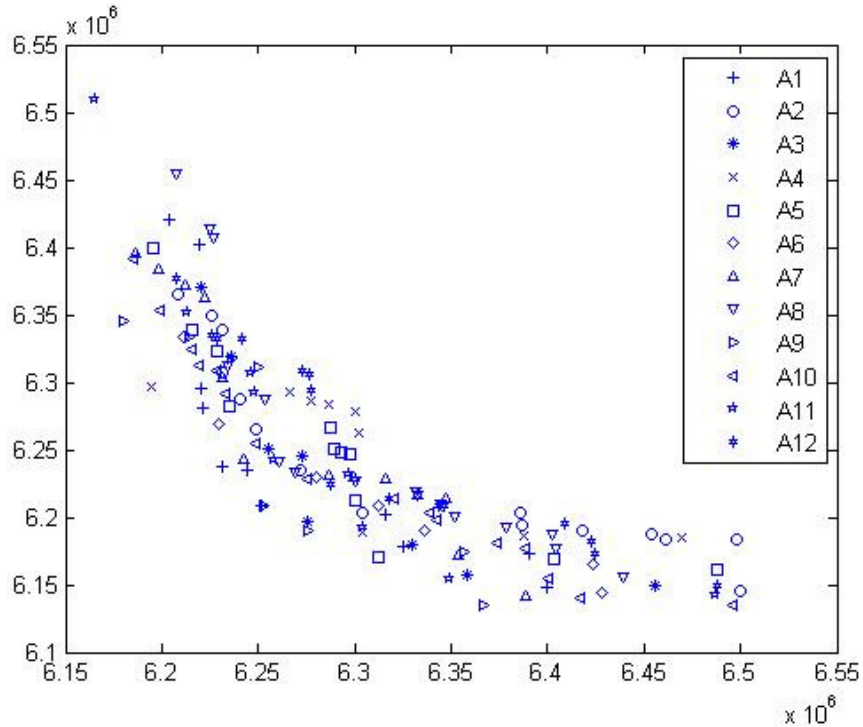
Şekil 4.14: 40 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü

50 boyutlu rassal örnek için her algoritma, 3 tekrarlı olarak ve her tekrar 900 sn. çalıştırılmıştır. 50 boyutlu rassal üretilmiş iki kriterli örnek probleme ait sonuçlar tablo 4.45'te görülmektedir. Tutarsızlık daha da artmış ancak tutarlılık oranı kabul edilebilir düzeydedir. Tablodan görüldüğü üzere A1 (İKKA) ele alınan kriter yönünden yine en iyi algoritma olarak karşımıza çıkmaktadır. Sonra sırasıyla A10 (PKKOA), A9 (ÇKKS), A7 (EKÇAS), A11 (PKKOA-M), A5 (İKÇKA), A3 (BİKA), A2 (İKKA-M), A6 (İKÇKA-M), A8 (KKOÇAS), A4 (BİKA-M) ve A12 (ÇAAQA) gelmektedir. Görüldüğü üzere PKKOA-M algoritması bu örnekte ilk defa ilk dört algoritma arasında yerini almamıştır. İKKA ve PKKOA aynı özelliğini korumuştur. ÇKKS küçük örneklerdeki performansından farklı olarak iyi bir performans ortaya koymaktadır.

Tablo 4.45: 50 boyutlu rassal örnekte ÇAKKO algoritmaları performansları

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Ö.V.
A1	1	3,33	3	4	2	3,33	1,4	10	1,2	1,29	1,6	10	0,1794
A2	0,3	1	1,6	2	0,75	0,83	0,56	0,89	0,5	0,27	1	1,57	0,0553
A3	0,33	0,63	1	3	1	1,67	0,5	1	0,38	0,45	0,63	2,67	0,0598
A4	0,25	0,5	0,33	1	0,22	0,75	0,33	0,5	0,4	0,22	0,33	1,5	0,0316
A5	0,5	1,33	1	4,5	1	0,83	0,63	1,75	0,6	0,27	0,86	2,25	0,0694
A6	0,3	1,2	0,6	1,33	1,2	1	0,83	1,17	0,33	0,33	1	1,75	0,0549
A7	0,71	1,8	2	3	1,6	1,2	1	1,25	0,6	0,42	2	1,29	0,0889
A8	0	1,13	1	2	0,57	0,86	0,8	1	0,14	0,19	0,67	1,22	0,0416
A9	0,83	2	2,67	2,5	1,67	3	1,67	7	1	1,2	1,5	8	0,1479
A10	0,78	3,75	2,2	4,5	3,67	3	2,4	5,33	0,83	1	2,2	8	0,1697
A11	0,63	1	1,6	3	1,17	1	0,5	1,5	0,67	0,45	1	1,6	0,0715
A12	0,1	0,64	0,38	0,67	0,44	0,57	0,78	0,82	0,13	0,13	0,63	1	0,0299
Tutarlılık oranı: 0,0369													

Şekil 4.15'te bu örnek için pareto-önyüzler görülmektedir.



Şekil 4.15: 50 boyutlu rassal örnekte ÇAKKO algoritmaları pareto-önyüzü

Bu çalışmada ortaya koyduğumuz kriter, algoritmaları karşılaştırırken algoritmanın hesaplamış olduğu tüm pareto-önyüzü ele almaktadır. Öncelik vektörü hesabında

birinci sırada yer alan algoritma, nadir olarak da olsa global pareto-önyüzde en çok sayıda karar seçeneğine sahip olamayabilmektedir. Bu durum özellikle 30 boyutlu rassal örnekte belirgin olarak karşımıza çıkmaktadır. Tablo 4.43'te A1'in A10'a göre önemi 0,83 çıkmasına rağmen öncelik vektör değerlerine bakıldığında A10 açık ara ikinci sıradadır. Yine aynı örnekte A4 (BİKA-M) algoritması öncelik vektörüne göre 7. sırada yer alırken şekil 4.13'e bakıldığında global pareto-önyüzde iyi sayıda karar seçeneği bulunduğunu görmekteyiz. Bu durum, A4'ün global pareto-önyüzde bulunduğu iyi karar seçeneklerinin dışındaki seçeneklerin diğer algoritmalara görece çok daha kötü olabileceğini göstermektedir.

Tablo 4.46: Global pareto-önyüzde buldukları karar seçeneği sayısına göre ÇAKKO algoritmaları performansları

Ö.D.	KC10-2fl -1uni	KC10-2fl -3uni	KC20-2fl -1uni	KC20-2f -3uni	r30	r40	r50	T	Oran
A1	12	24	9	5	3	7	5	65	0,24
A2	12	15	2	6	1	0	0	36	0,13
A3	10	12	1	5	0	0	1	29	0,11
A4	9	17	1	3	3	0	2	35	0,13
A5	12	12	1	5	0	1	1	32	0,12
A6	10	15	3	4	0	2	1	35	0,13
A7	12	18	2	9	0	1	0	42	0,15
A8	13	17	0	6	0	0	0	36	0,13
A9	6	13	1	5	2	0	3	30	0,11
A10	12	28	6	13	3	0	0	62	0,23
A11	13	10	1	10	1	0	2	37	0,13
A12	1	14	2	6	1	0	0	24	0,09
T	13	116	29	77	14	11	15	275	1

İkili karşılaştırmalara dayalı performans değerlendirme yönteminde algoritmalar tüm pareto-önyüzüyle dikkate alınmaktadır. Karar verici tek algoritma çalıştırarak sonuca gitmek istediğinde bu kriter oldukça anlamlı ve genelleştiricidir. Ancak amaç, global pareto-önyüzde çok sayıda karar seçeneği bulmaya yönelik ise ve karar verici tek bir algoritma değil, birkaç algoritma çalıştırarak karar verecek ise, her algoritmanın (birleştirilmesiyle oluşan) global pareto-önyüzde bulunduğu karar seçeneği sayısı da dikkate alınsa iyi olur. Bundan dolayı her örnek için 12 algoritmanın birleştirilmesiyle oluşturduğu pareto-önyüzde ilgili algoritmaya ait kaç karar

seçeneđi bulunduđunu gösteren tablo 4.46 oluşturulmuştur. Bu tablodan da görüldüğü üzere tüm çalışma boyunca global pareto-önyüzde bulunan karar seçeneklerinden %24'ü İKKA'ya, %23'ü ise PKKOA algoritmasına aittir. Dolayısıyla global pareto-önyüze yönelik önerdiğimiz bu kritere göre de bu iki algoritma çok başarılı görülmektedir. Ancak diđer algoritmalara baktığımızda hemen hemen aynı oranda global pareto-önyüzde sonuç bulmakla birlikte, ÇAAQA algoritması sonuncu konumunu korumaktadır.

Özellikle bu çalışmada ortaya çıkan bir nokta da ÇAKKO algoritmalarından, feromon güncelleme stratejilerini hem klasik tek amaçlı sistemlerdekinden farklı hem de özellikle pareto-önyüze yönelik olarak oluşturan ve bu stratejileri oluştururken de pareto-önyüzü homojen olarak arayabilen yöntemlerin İKKAP problemlerini çözerken oldukça başarılı olacaktır. Bundan dolayı PKKOA ve İKKA algoritmaları en başarılı algoritmalar olarak yer almaktadırlar. Çözüm üretirken pareto-önyüzü dikkate almayan yöntemlerin çok başarılı olamadığı ve hatta bazı algoritmaların özellikle kötü hesapsal yapılarından ötürü oldukça başarısız oldukları tespit edilmiştir. Bu konuda özellikle ÇAAQA algoritması dikkat çekici bir başarısızlığa sahip olmuştur.

Homojen ve kaliteli feromon stratejilerine sahip başarılı algoritmaların halen farklı yöntemlerle geliştirilebilir yapıları ilgi çekicidir. Bu algoritmalarda tekil amaçlı ve stabil kullanılmış araştırma-devam katsayısı vb. stratejileri de pareto-önyüze yönlendirilerek geliştirmek mümkündür. Bundan ziyade kaliteli feromon stratejilerinin daha önemli sonuçlar ortaya koyabileceđi sezgisi, tüm bu algoritmalarla yapılan denemeler sonucunda ortaya çıkmıştır.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \frac{1}{l} \cdot poyraz \quad (4.1)$$

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \frac{1}{l} + poyraz \quad (4.2)$$

$$poyraz = \frac{P_L + 1}{P_G + 1} \quad (4.3)$$

Her ne kadar başarılı sonuçlar elde edilse de, İKKA algoritmasının pareto-önyüze yönelik olduđu düşünülerek tasarlanmış feromon stratejileri, tam anlamıyla pareto-

önyüze odaklanamamaktadır. Her iterasyonda yerel pareto-önyüzdeki karınca sayısına göre feromon güncelleme yapmak, pareto-önyüzle ilgili gözüke ve aritmetik olarak da deneylerde başarılı sonuçlar ortaya çıkarsa da daha da geliştirilebilir bir mantıktır. Özellikle algoritmaların çoğunun bu stratejilerini geliştirirken, Zitzler ve diğ. (2000)'nin ileri sürmüş olduğu pareto-önyüz değerlendirme kriterleriyle ilgisi olmayan, yetersiz veya bilimsel altyapısı olmayan formülasyonlar kullandıkları çokça görülmüştür. Bundan dolayı bu çalışmada öne sürdüğümüz karşılaştırmalı pareto-önyüz değerlendirme kriterini İKKA'ya uyarladık. İKKA'da her yerel pareto-önyüz oluşturulduğunda, pareto-önyüzdeki karınca sayısı dikkate alınarak hesaplanan $1/l$ değerinin iki amaca ait feromon matrisine de eklendiğini önceki bölümlerde anlatmıştık. Bunun yerine denklem (4.1)'de ifade edilen feromon güncelleme stratejisini ve onun bir toplamlı versiyonu olan denklem (4.2)'yi denemeye karar verdik. Bu denklemlerde söz konusu olan *poyraz* faktörü denklem (4.3)'de görüldüğü gibi hesaplanmaktadır.

Her iterasyonda denklem (4.3), yerel pareto-önyüz ile global pareto-önyüzün birleştirilmesiyle ortaya çıkan birleşik pareto-önyüz üzerinden hesaplanır. P_L birleşik pareto-önyüzdeki ilgili iterasyonda ortaya çıkan yerel pareto-önyüze ait karar seçeneği sayısı ve P_G birleşik pareto-önyüzdeki global pareto-önyüze ait karar seçeneği sayısını ifade etmektedir. *poyraz* değeri hesaplanırken, algoritma başlangıcında *poyraz* değerinin olması gereken değere eşit olması (yani 1'e eşit olması için paydaya 1 eklenmesi) veya algoritma çalıştırılması sırasında yerel pareto-önyüzün birleşik pareto-önyüzde karar seçeneği çıkartamaması durumunda ifadenin sıfırlanmaması (bu yolla eklenmesi gereken $1/l$ değerinin eklenebilmesi) için formülasyonda paya ve paydaya 1 eklenmiştir. *poyraz* değerinin yerel pareto-önyüzün karar seçeneği sayısının globalinkine oranı olmadığına özellikle dikkat etmek gerekir.

Tanımlandığı şekliyle *poyraz*, denklem (4.1)'de, eklenen feromon miktarını her iterasyonda koloninin bulunduğu pareto-önyüzün kalitesi kadar kat artırmakta veya kalitesizliği kadar kat feromonu azaltmaktadır. Sadece iyi çözümlere daha çok feromon eklenmesi değil, kötü çözümlerin feromon miktarını daha çok azaltması

sayesinde daha başarılı sonuçlar elde edebileceği düşünülmektedir. Ayrıca İKKA gibi başarılı sonuçlar veren bir algoritmanın, karıncaların feromonlarını pareto-önyüzle daha gerçekçi şekilde ilişkilendirilmesi, bu sayede algoritma sonuçlarının orjine doğru daha çok mesafe kat edebilecek nitelikte bir stratejiyle donatılması sayesinde daha iyi sonuçlar ortaya koyacağı sezgisi oluşmuştur.

İKKA algoritmasının orijinal şekliyle başarılı olmasındaki etkenlerden birisinin iyi belirlenmiş bir aritmetiğe sahip feromon güncellemesi olduğu düşünülmektedir. Denklem (4.1)den farklı olarak *poyraz* değerini her iterasyonda ekleyen denklem (4.2) de aritmetiksel olarak iyi tasarlanmış olabileceği düşünüldüğünden yeni denemelere alınmaya karar verilmiştir. Toplamı olan bu ifade sayesinde her iterasyonda koloninin bulduğu sonucun iyiliği ölçüsünde feromon eklenmesi sözkonusudur. Bu ifadede pareto-önyüzle ilişkili feromon azaltma mantığı olmadığına dikkat edelim.

poyraz faktörü bir algoritma olarak düşünüldüğünde diğer tüm İKKAP ve birleşimsel iki-kriterli problem çözen iteratif iki-kriterli ÇAKKO'lara uygulanabilir. Fakat bütün 12 algoritmaya uygulamak ve deneylerini yapmak çok zaman gerektireceğinden sadece en başarılı olan algoritmalarından biri olan İKKA'ya uygulanması şimdilik uygun görülmüştür. Çarpımlı olan denklem (4.1)i kullanan algoritmaya ÇİKKA-*poyraz* algoritması (bu çalışmada 13. algoritma olduğu için A13 diye de adlandırılabilir), toplamı olan denklem (4.2)yi kullanan algoritmaya TİKKA-*poyraz* algoritması (bu çalışmada 14. algoritma olduğu için A14 diye de adlandırılabilir) ismi verilmesi uygun görülmüştür. Bu algoritmalara yapılan eklenti algoritmasına *poyraz* isminin verilmesi sebebi, ilerideki sayfalarda sonuçlar ortaya konulduktan sonra açıklanacaktır.

Önerdiğimiz yeni algoritmalarla birlikte toplam 14 algoritma, başlangıç denemelerdeki 3 yerine 12 deneme sayısı ile her örnek için (toplamda 7 örnek bulunmaktadır) süreler önceki sayfalardakilerle aynı alınmak suretiyle test edilecektir. Bu performans karşılaştırmaları yapılmaya başlanmadan önce, yine adaletli olunması açısından ÇİKKA-*poyraz* ve TİKKA-*poyraz* algoritmaları için

parametre eniyileme çalışmaları, diğer algoritmalara yapıldığıyla benzer şekilde yapılmıştır. Tablo 4.47 ve 4.48 bu çalışmalar için oluşturulmuş deney tasarımlarını göstermektedir.

Tablo 4.47: ÇİKKA- *poyraz* ve TİKKA- *poyraz* (MBicriterionAnt-*boreas* ve SBicriterionAnt-*boreas*) deney seviyeleri

FAKTÖR	Seviye 1	Seviye 2
F1 (karınca sayısı)	80	100
F2 (feromonun önemi)	2	4
F3 (sezgisel bilginin önemi)	1	1,5
F4 (buharlaştırma katsayısı)	0,25	0,75

Tablo 4.48: ÇİKKA- *poyraz* ve TİKKA- *poyraz* (MBicriterionAnt-*boreas* ve SBicriterionAnt-*boreas*) $2^{4-1} + 1$ kesirli deney tasarımı

Seviye	F1	F2	F3	F4
D1	1	1	1	1
D2	2	1	1	2
D3	1	2	1	2
D4	2	2	1	1
D5	1	1	2	2
D6	2	1	2	1
D7	1	2	2	1
D8	2	2	2	2
D9	100	1	1	0,02

Tablo 4.49: ÇİKKA- *poyraz* (MBicriterionAnt-*boreas*) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D2	0,8333	1	0,8333	0,8333	0,8333	0,8333	0,9091	0,8333	0,9091	0,0962
D3	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D4	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D5	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D6	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D7	0,9167	1,1	0,9167	0,9167	0,9167	0,9167	1	0,9167	1	0,1058
D8	1	1,2	1	1	1	1	1,0909	1	1,0909	0,1154
D9	0,9167	1,1	0,9167	0,9167	0,9167	0,9167	1	0,9167	1	0,1058
Tutarlılık oranı: 0,000001										

ÇİKKA- *poyraz* algoritması için parametre düzen seçimi tablo 4.49'da, TİKKA- *poyraz* algoritması için seçim ise tablo 4.50'de görülmektedir.

Tablo 4.50: TİKKA-*poyraz* (SBicriterionAnt-*boreas*) parametre düzen seçimi

Ö.D.	D1	D2	D3	D4	D5	D6	D7	D8	D9	Ö.V.
D1	1	0,6667	0,6667	0,6667	0,6667	0,8333	0,6667	0,6667	1,1111	0,0834
D2	1,5	1	0,8333	0,8333	0,8333	0,8333	0,9091	0,8333	0,9091	0,1018
D3	1,5	1,2	1	1	1	1,0909	1	1	1,5	0,1236
D4	1,5	1,2	1	1	1	1	1,0909	1	1,0909	0,1194
D5	1,5	1,2	1	1	1	1,0909	1	0,6667	1,5	0,1190
D6	1,2	1,2	0,9167	1	0,9167	1	1,0909	1	1,0909	0,1145
D7	1,5	1,1	1	0,9167	1	0,9167	1	0,8333	1,5	0,1168
D8	1,5	1,2	1	1	1,5	1	1,2	1	1,0909	0,1273
D9	0,9	1,1	0,6667	0,9167	0,6667	0,9167	0,6667	0,9167	1	0,0942
Tutarlılık oranı: 0,0047										

İKKA'da iyi sonuçlar verdiğini gördüğümüz D8 deney düzeni, önerdiğimiz iki algoritma için de iyi sonuçlar vermiştir. Bundan dolayı söz konusu iki algoritma için de D8 deney düzeni ile denemeler yapmaya karar verilmiştir. Performans değerlendirme deneyleri aynı örnekler üzerinde aynı sürelerde fakat 12 deneme sayısı ile gerçekleştirilmiştir.

KC10-2fl-1uni örneği için 14 algoritmanın 12 denemeye elde ettiği performans sonuçları tablo 4.51'de karşılaştırılmaktadır. Performans karşılaştırma tablolarının sayfaya sığması için her hücre değeri virgülden sonra iki haneye yuvarlatılmıştır. Biri diğerine tamamen baskın olan algoritmanın öncelik değeri belirlenirken tutarlılık oranı dikkate alınmıştır. Tutarlılık oranını negatif yapmayı pozitif enküçükleyen herhangi bir baskın algoritmaya ait en küçük pozitif tamsayı öncelik değeri deneme yoluyla bulunarak tabloya konulmuştur. Bu tablo için baskın algoritmanın öncelik değeri bu yolla 18 olarak belirlenmiştir. Tablodan da görüldüğü üzere PKKOA (A10) ve TİKKA-*poyraz* (A14) en başarılı algoritmalar olmuştur. İKKA (A1) ikinci olurken PKKOA-M (A11) üçüncü olmuştur. Bundan sonra sırasıyla ÇİKKA-*poyraz* (A13), EKÇAS (A7), BİKA (A3), İKÇKA (A5), KKOÇAS (A8), İKÇKA-M (A6), BİKA-M (A4), İKKA-M (A2), ÇKKS (A9), ÇAAQA (A12) algoritmaları gelmektedir.

Tablo 4.51: Önerilen ÇAKKO algoritmalarının performanslarının KC10-2fl-1uni literatür probleminde karşılaştırılması

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Ö.V.
A1	1	10	2,00	3,33	2,00	5,00	1,43	2,50	10	0,91	1,11	18	1,25	0,91	0,1332
A2	0,10	1	0,50	1,71	0,45	0,75	0,50	0,86	1,50	0,09	0,15	2,33	0,22	0,09	0,0259
A3	0,50	2,00	1	2,00	1,60	2,00	0,89	1,43	4,00	0,45	0,50	14	0,88	0,45	0,0692
A4	0,30	0,58	0,50	1	0,60	0,75	0,60	0,64	1,29	0,27	0,25	3,25	0,50	0,27	0,0325
A5	0,50	2,20	0,63	1,67	1	1,83	0,90	1,20	2,75	0,45	0,56	6,00	0,67	0,45	0,0569
A6	0,20	1,33	0,50	1,33	0,55	1	0,55	1	1,67	0,18	0,30	5,00	0,44	0,18	0,0341
A7	0,70	2,00	1,13	1,67	1,11	1,83	1	1,40	3,67	0,64	0,70	18	0,89	0,64	0,0759
A8	0,40	1,17	0,70	1,57	0,83	1	0,71	1	4,00	0,36	0,38	6,00	0,56	0,36	0,0477
A9	0,10	0,67	0,25	0,78	0,36	0,60	0,27	0,25	1	0,09	0,10	2,20	0,22	0,09	0,0167
A10	1,10	11	2,20	3,67	2,20	5,50	1,57	2,75	11	1	1,22	18	1,38	1	0,1457
A11	0,90	6,50	2,00	4,00	1,80	3,33	1,43	2,60	10	0,82	1	18	1,11	0,82	0,1213
A12	0	0,43	0,07	0,31	0,17	0,20	0	0,17	0,45	0	0	1	0	0	0,0044
A13	0,80	4,50	1,14	2,00	1,50	2,25	1,13	1,80	4,50	0,73	0,90	18	1	0,73	0,0910
A14	1,10	11	2,20	3,67	2,20	5,50	1,57	2,75	11	1	1,22	18	1,38	1	0,1457
Tutarlılık oranı: 0,0002															

KC10-2fl-3uni örneği için yapılan 12 tekrarlı çalıştırmalarda 14 algoritmanın performans sonuçları tablo 4.52’de gösterilmiştir. PKKOA (A10) algoritması birinci, ÇİKKA-*poyraz* (A13) algoritması ikinci ve PKKOA-M (A11) algoritması üçüncü olmuştur. Sonra gelen algoritmalar sırasıyla KKOÇAŞ (A8), EKÇAS (A7), İKKA (A1), İKKA-M (A2), TİKKA-*poyraz* (A14), İKÇKA-M (A6), BİKA-M (A4), ÇKKS (A9), BİKA (A3), İKÇKA (A5), ÇAAQA (A12) algoritmalarıdır. İki 10 boyutlu örnekte de PKKOA algoritmasının üstünlüğünü pekiştirmesi, algoritmaların hemen hemen hepsinin orjine belli mesafeye (belli bir sınıra) gelmesi ve bu sınıra ulaştıktan sonra pareto-önyüzde kuzeybatı-güneydoğu yönündeki arayışın, bu algoritmaya ait rassal amaç ağırlıklandırması sayesinde yeterli ölçüde çeşitlendirilmesi sayesinde olduğu düşünülebilir. Çok sayıda tekrar yapıldığında bu çeşitlendirme stratejisi daha da güçlü çözümler üretmeye sebep olabilmektedir. ÇİKKA-*poyraz* algoritmasının TİKKA-*poyraz*’a göre başarısı da bu gibi çok karar seçenekli çözümün olduğu küçük örnekte bu çeşitlendirmeyi feromon stratejisi (azaltılması) sayesinde başarıyor olabilmektedir.

Tablo 4.52: Önerilen ÇAKKO algoritmalarının performanslarının KC10-2fl-3uni literatür probleminde karşılaştırılması

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Ö.V.
A1	1	0,97	1,31	1,15	1,67	1,06	1,04	0,91	1,36	0,75	0,95	2,37	0,93	1	0,0776
A2	1,03	1	1,25	1	1,61	1,14	0,84	0,98	1,27	0,71	0,98	2,06	0,91	1,06	0,0754
A3	0,76	0,80	1	0,73	1,11	0,78	0,74	0,68	0,95	0,51	0,62	1,75	0,64	0,75	0,0556
A4	0,87	1	1,37	1	1,56	1,02	0,86	0,96	1,25	0,65	0,95	2,35	0,81	0,88	0,0725
A5	0,60	0,62	0,90	0,64	1	0,55	0,59	0,62	0,73	0,46	0,56	1,47	0,55	0,59	0,0464
A6	0,94	0,88	1,29	0,98	1,80	1	0,92	0,84	1,23	0,69	0,81	2,63	0,75	1,06	0,0728
A7	0,96	1,18	1,35	1,16	1,68	1,09	1	1	1,11	0,76	0,94	2,39	0,96	1,04	0,0786
A8	1,10	1,02	1,47	1,05	1,60	1,20	1	1	1,31	0,80	0,85	2,29	0,97	1,03	0,0790
A9	0,74	0,79	1,05	0,80	1,37	0,82	0,90	0,76	1	0,59	0,79	1,97	0,70	0,82	0,0614
A10	1,34	1,41	1,96	1,54	2,19	1,45	1,32	1,25	1,70	1	1,42	2,87	1,23	1,58	0,1058
A11	1,05	1,02	1,62	1,05	1,79	1,24	1,06	1,17	1,27	0,70	1	2,34	0,93	1	0,0811
A12	0,42	0,49	0,57	0,43	0,68	0,38	0,42	0,44	0,51	0,35	0,43	1	0,41	0,44	0,0330
A13	1,08	1,09	1,57	1,23	1,83	1,33	1,05	1,03	1,43	0,81	1,08	2,46	1	1,11	0,0855
A14	1	0,94	1,34	1,14	1,70	0,94	0,96	0,97	1,21	0,63	1	2,26	0,90	1	0,0752
Tutarlılık oranı: 0,0012															

Tablo 4.53: Önerilen ÇAKKO algoritmalarının performanslarının KC20-2fl-1uni literatür probleminde karşılaştırılması

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Ö.V.
A1	1	13	13	13	15	8,00	3,75	6,50	13	2,00	6,00	15	0,18	0,78	0,1759
A2	0,00	1	1,40	1,18	1,18	0,35	0,44	0,90	1,56	0,15	0,64	3,67	0,09	0,13	0,0204
A3	0,00	0,71	1	1,40	0,90	0,38	0,39	1	1,30	0,24	0,71	1,75	0,00	0,06	0,0145
A4	0,00	0,85	0,71	1	0,92	0,69	0,11	0,64	0,92	0,10	0,55	3,67	0,00	0,00	0,0127
A5	0,07	0,85	1,11	1,09	1	0,64	0,41	0,64	1,36	0,20	0,33	2,60	0,00	0,11	0,0155
A6	0,13	2,83	2,60	1,44	1,57	1	0,82	1,25	2,67	0,30	1,33	4,25	0,09	0,20	0,0338
A7	0,27	2,29	2,57	9,50	2,43	1,22	1	2,00	2,00	0,50	1,14	5,33	0,00	0,13	0,0427
A8	0,15	1,11	1	1,57	1,57	0,80	0,50	1	1,10	0,36	1,14	4,67	0,09	0,12	0,0258
A9	0,00	0,64	0,77	1,08	0,73	0,38	0,50	0,91	1	0,10	0,50	1,75	0,09	0,06	0,0152
A10	0,50	6,67	4,25	10	5,00	3,33	2,00	2,80	10,5	1	2,50	6,33	0,20	0,31	0,0889
A11	0,17	1,57	1,40	1,83	3,00	0,75	0,88	0,88	2,00	0,40	1	10	0,11	0,13	0,0355
A12	0,07	0,27	0,57	0,27	0,38	0,24	0,19	0,21	0,57	0,16	0,10	1	0,00	0,06	0,0068
A13	5,50	11	13	13	13	11	13	11	11	5,00	9,00	13	1	3,00	0,3124
A14	1,29	8,00	18	13	9,00	5,00	8,00	8,50	17	3,20	7,50	18	0,33	1	0,1999
Tutarlılık oranı: 0,0012															

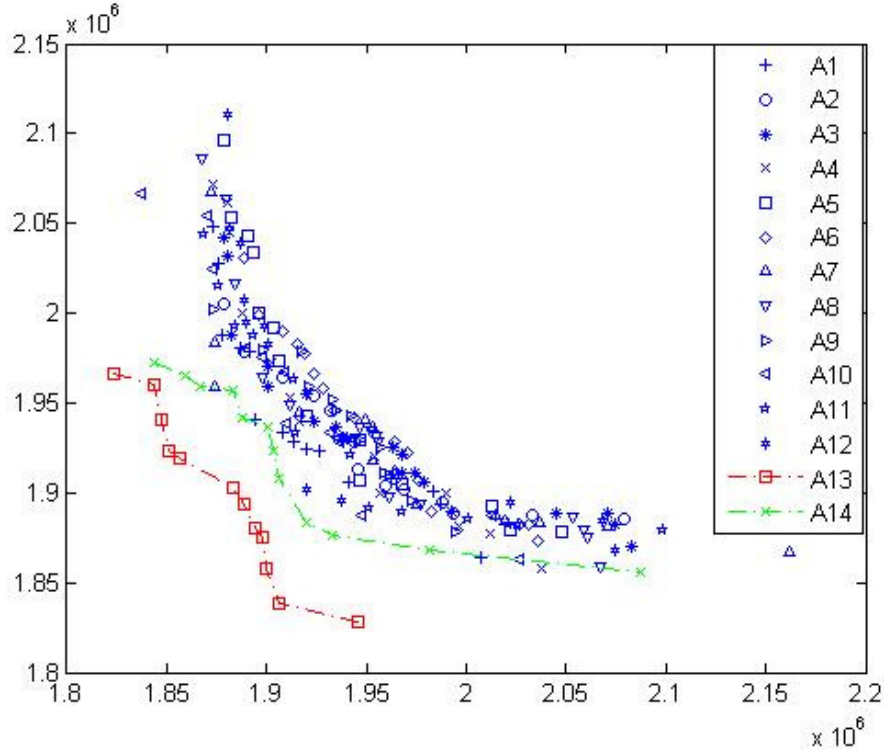
KC20-2fl-1uni örneği için yapılan 12 tekrarlı çalıştırmalarda 14 algoritmanın performans sonuçları tablo 4.53’de gösterilmiştir. Bu tablo için baskın algoritmanın öncelik değeri 13 olarak belirlenmiştir. Tutarlılık oranı oldukça makul bir seviyededir. ÇİKKA-*poyraz* (A13) algoritmasının açık ara birinci olduğu görülmektedir. TİKKA-*poyraz* (A14) algoritması ikinci ve İKKA (A1) algoritması üçüncü olmuştur. Diğer algoritmalar öncelik vektörü sırasına göre, PKKOA (A10), EKÇAS (A7), PKKOA-M (A11), İKÇKA-M (A6), KKOÇAŞ (A8), İKKA-M (A2), İKÇKA (A5), ÇKKS (A9), BİKA (A3), BİKA-M (A4), ÇAAQA (A12)dir.

Tablo 4.54: Önerilen ÇAKKO algoritmalarının performanslarının KC20-2fl-3uni literatür probleminde karşılaştırılması

Ö.D.	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Ö.V.
A1	1	1,10	1,33	0,51	1,16	0,98	0,57	0,62	1,39	0,47	0,42	1,28	0,67	0,65	0,0552
A2	0,91	1	1,11	0,78	1,17	1,16	0,68	0,76	1	0,40	0,59	1,67	0,61	0,78	0,0581
A3	0,75	0,90	1	0,78	1,22	0,97	0,68	0,74	0,88	0,40	0,61	1,15	0,49	0,65	0,0527
A4	1,96	1,29	1,28	1	1,39	1,26	0,79	1,13	1,44	0,71	0,65	1,72	0,91	0,79	0,0757
A5	0,86	0,86	0,82	0,72	1	1	0,57	1	0,73	0,58	0,50	1,61	0,58	0,63	0,0537
A6	1,03	0,86	1,03	0,79	1	1	0,42	0,90	1,35	0,58	0,72	1,43	0,61	0,76	0,0582
A7	1,75	1,48	1,47	1,26	1,75	2,37	1	1,27	1,83	0,92	0,94	1,96	1,19	1,07	0,0951
A8	1,62	1,31	1,36	0,88	1	1,11	0,79	1	1,52	0,57	0,74	2,00	0,77	0,77	0,0710
A9	0,72	1	1,13	0,69	1,37	0,74	0,55	0,66	1	0,49	0,55	1,06	0,56	0,53	0,0514
A10	2,15	2,50	2,47	1,40	1,73	1,73	1,09	1,76	2,05	1	1,33	3,36	1,36	1,08	0,1152
A11	2,37	1,71	1,64	1,55	2,00	1,39	1,07	1,34	1,81	0,75	1	1,86	0,94	1,07	0,0954
A12	0,78	0,60	0,87	0,58	0,62	0,70	0,51	0,50	0,94	0,30	0,54	1	0,52	0,70	0,0431
A13	1,50	1,63	2,06	1,10	1,73	1,64	0,84	1,29	1,79	0,73	1,07	1,94	1	1,15	0,0905
A14	1,55	1,28	1,55	1,26	1,60	1,32	0,94	1,29	1,89	0,92	0,93	1,43	0,87	1	0,0848
Tutarlılık oranı: 0,0068															

KC20-2fl-3uni örneği için yapılan 12 tekrarlı çalıştırmalarda 14 algoritmanın performans sonuçları tablo 4.54’te gösterilmiştir. Bu tablo için baskın algoritmanın öncelik değeri belirlenmesine gerek kalmamıştır. Tutarlılık oranı kabul edilebilir bir seviyededir. Bu örnekte birinci olan algoritma PKKOA (A10)’dur. PKKOA-M (A11) ikinci, EKÇAS (A7) üçüncü, ÇİKKA-*poyraz* (A13) dördüncü ve TİKKA-*poyraz* (A14) beşinci algoritmalar olmuşlardır. Geriye kalan algoritmalar öncelik sırasına göre şöyledir: BİKA-M (A4), KKOÇAŞ (A8), İKÇKA-M (A6), İKKA-M (A2), İKKA (A1), İKÇKA (A5), BİKA (A3), ÇKKS (A9), ÇAAQA (A12). Özellikle bu

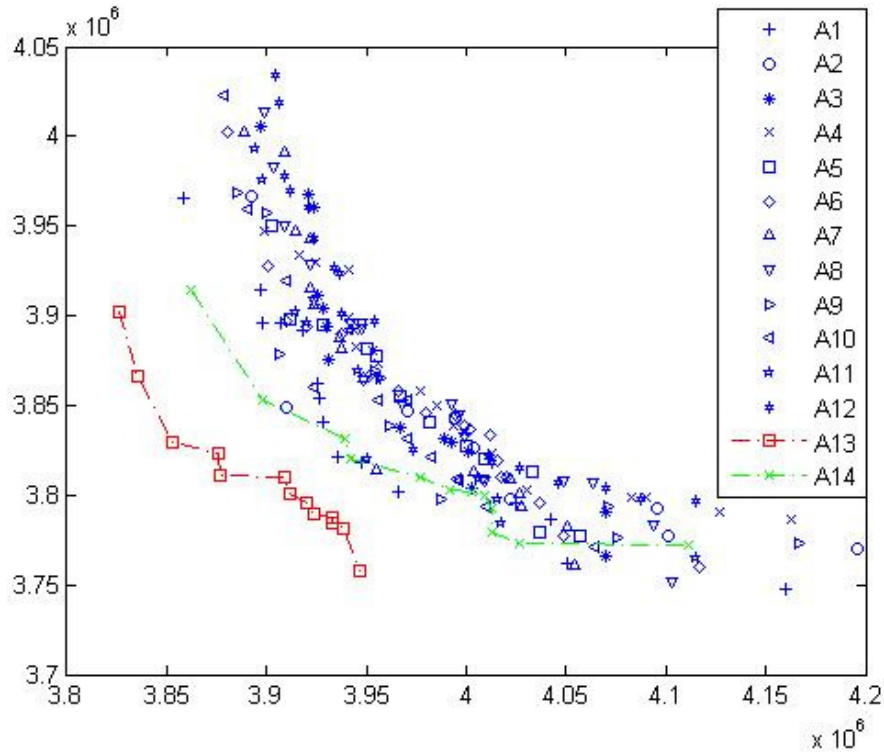
örnekte ortaya çıkan bir durum da İKKA algoritmasına yapılan *poyraz* eklentisinin bu algoritmayı ne kadar üst sıralara taşıyabildiğidir. Sonraki daha büyük örneklerde bu farklılık daha da belirginleşecektir.



Şekil 4.16: 30 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü

30 boyutlu rassal örnekte yapılan denemeler sonucu şekil 4.16'da görülmektedir. Literatürdeki algoritmaların hemen hemen hepsi belirli bölgede çözüm üretebiliyorken, TİKKA-*poyraz* (A14) bu çözüm grubunun hemen altında, ÇİKKA-*poyraz* (A13) ise bu gruptan açık ara önde bir şekilde orijine daha yakın çözüm üretmiştir. Bütün bu 14 algoritmanın pareto-önyüzünü birleştirdiğimizde, karar seçeneklerinin hepsinin ÇİKKA-*poyraz* (A13)'a ait olacağı açıktır.

40 boyutlu rassal örnekte yapılan denemeler sonucu şekil 4.17'de görülmektedir. Bu örnekte de görülmektedir ki literatürdeki algoritmaların hemen hemen hepsi belirli bölgede çözüm üretebiliyorken önerdiğimiz algoritmalar çok iyi konumlarda çözüm üretebiliyorlar. ÇİKKA-*poyraz* (A13) tartışmasız üstünlüğünü bu örnekte de ortaya koymaktadır.

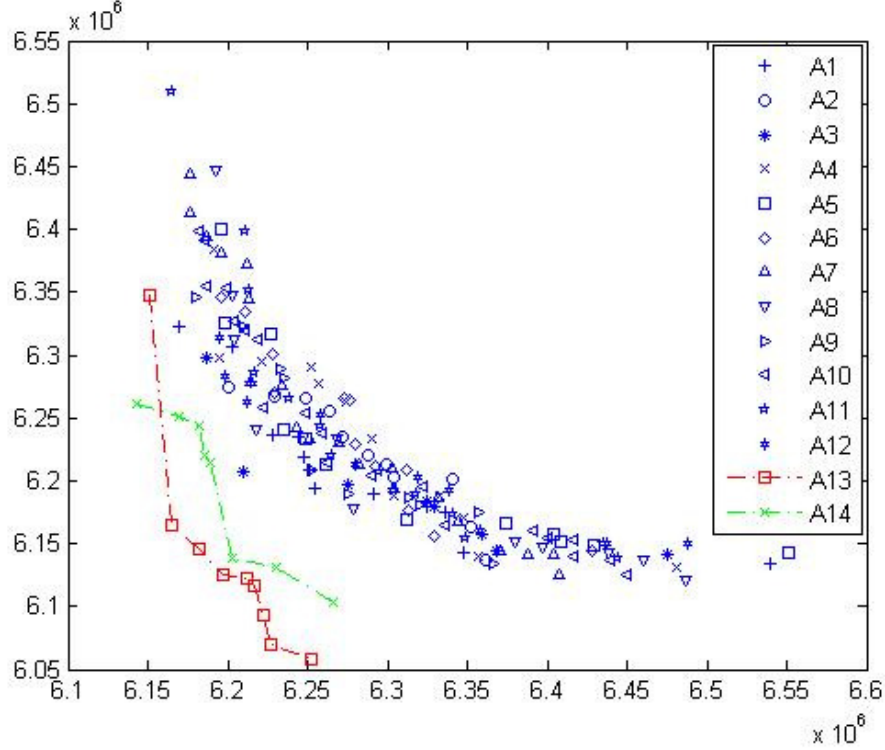


Şekil 4.17: 40 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü

50 boyutlu rassal örnekte yapılan denemeler sonucu şekil 4.18’de görülmektedir. Bu örnekte de görülmektedir ki literatürdeki algoritmaların hemen hemen hepsi belirli bölgede çözüm üretebiliyorken önerdiğimiz algoritmalar açık ara orjine yakın çözüm üretebilmektedirler. ÇİKKA-*poyraz* (A13) tartışmasız üstünlüğünü bu örnekte de ortaya koymaktadır. Bu örnekte TİKKA-*poyraz* (A14) algoritmasının biraz daha diğer algoritmalarından ayrışıp ÇİKKA-*poyraz* (A13) algoritmasına yakınlaştığını açıkça görebilmekteyiz.

Bütün bu denemeler de göstermektedir ki; İKKA algoritmasına *poyraz* algoritması (faktörü) eklendiğinde, oldukça rekabetçi ve büyük örneklerde literatürde mevcut algoritmaların hepsine baskın sonuçlar üretebilen bir algoritma ortaya çıkmıştır. Eklenti algoritmasının adının *poyraz* konulmuş olmasının sebebi, bu eklenti sayesinde algoritmanın İKKAP minimizasyon problemlerinde pareto-önyüzü, kuzeydoğu yönünden güneybatı yönüne doğru (deyim yerindeyse) savurmuş olmasıdır. İyi temsil edilen pareto-önyüz değerlendirme yönteminin feromon

güncelleme stratejisine ustaca uygulanması ile oluşan bu algoritmanın, diğerlerine göre orjine daha hırslı şekilde yakınsamaması için hiçbir sebep görülmemektedir. Sonuçlar da bunu göstermiştir.



Şekil 4.18: 50 boyutlu rassal örnekte 12 replikasyonda oluşan ÇAKKO algoritmaları ve önerilen algoritmaların pareto-önyüzü

Global pareto-önyüzde buldukları karar seçeneği sayısına göre ÇAKKO algoritmaları performansları tablo 4.46'da değerlendirilmiştir. Bu değerlendirme önerilecek algoritmaya ön hazırlık niteliğinde 3 replikasyon sonucu elde edilen çözümler üzerinden yapılmıştır. Tablo 4.55'te, önerdiğimiz algoritmalarla birlikte toplam 14 algoritmadan 12 replikasyon sonucunda elde edilen pareto-önyüzlerden oluşturulmuş değerlendirme ortaya konulmuştur. Bu tablodan da görüldüğü üzere ÇİKKA-*poyraz* (A13) algoritması genel anlamda açıkça birinci olmuştur. Tüm çalışma boyunca global pareto-önyüzde bulunan karar seçeneklerinden %41'i bu algoritmaya aittir ve global pareto-önyüzde bulunan toplam karar seçeneği sayısında %94,34'lük artış elde edilmiştir. PKKOA (A10) algoritması ise küçük örneklerden bazılarındaki başarısını büyük örneklerde ortaya koyamamış ve ikinci olmuştur. ÇAAQA algoritması sonuculuğunu yine korumuştur.

Tablo 4.55: Global pareto-önyüzde buldukları karar seçeneği sayısına göre ÇAKKO algoritmaları ve önerilen algoritmaların performansları (12 replikasyon sonucu)

Ö.D.	KC10-2f -1uni	KC10-2f -3uni	KC20-2f -1uni	KC20-2f -3uni	r30	r40	r50	T	Oran
A1	10	40	0	2	0	1	0	53	0,21
A2	1	43	0	4	0	0	0	48	0,19
A3	5	33	0	4	0	0	0	42	0,17
A4	3	34	0	4	0	0	0	41	0,16
A5	5	23	0	3	0	0	0	31	0,12
A6	2	36	0	1	0	0	0	39	0,15
A7	7	44	0	6	0	0	0	57	0,23
A8	4	42	1	3	0	1	0	51	0,20
A9	1	33	0	2	0	0	0	36	0,14
A10	11	60	0	7	0	0	0	78	0,31
A11	9	45	1	5	0	0	0	60	0,24
A12	0	21	0	2	0	0	0	23	0,09
A13	8	44	8	10	12	13	8	103	0,41
A14	11	38	2	12	0	0	1	64	0,25
T	13	127	12	65	12	15	9	253	1

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada ÇAKKO algoritmaları İKKAP'ları çözmek üzere kodlanmış ve performansları değerlendirilmiştir. Algoritmaları değerlendirme çalışmaları yapılmaya başlamadan önce, her algoritmanın kendi içerisinde en iyi sonuçları veren parametre düzenleri bulunmaya çalışılmıştır. Hem performansların değerlendirmesinde hem de sözü geçen bu parametre düzenlerinin belirlenmesinde, literatürde bilinen değerlendirme kriterlerinden ve yöntemlerinden farklı bir kriter ve yöntem önerilerek bilimsel literatüre bir yenilik kazandırılmıştır. Önerdiğimiz kriterle, algoritmaları orjine veya pareto-optimuma göre, pareto-önyüzün düzgün dağılmasına veya uç noktaları bulmadaki başarısına göre ayrı ayrı karşılaştırmaktan ziyade, doğrudan pareto-önyüzlere odaklanarak ikili karşılaştırmada direkt amaca yönelmeye çalışılmaktadır. Ayrıca pareto-önyüzlerin literatür kriterleri ile değerlendirilmesinde ortaya çıkan ayrı bir çok kriterli yapıda algoritmalar arasında nasıl bir öncelik sıralaması yapılabileceğine dair yeni bir yaklaşım sunulmaktadır.

Bu çalışmada görülmüştür ki, algoritmaların önerildiği parametre düzenleri, bu algoritmalar için en iyi (gürbüz) parametre düzeni olamayabilmektedir. Ayrıca önerildiği çalışmadan farklı düzenlerle çalıştırıldığında sonuçlar çok başarısız da olabilmektedir. Literatürde ÇAKKO algoritmalarının İKKAP çözmek üzere çalıştırılmasına ait bir öneri veya uygulama mevcut olmaması da dikkate alındığında, ÇAKKO'ların İKKAP örneklerini çözerek performanslarını karşılaştırmaya başlamadan önce deneysel tasarımlar yapmanın ve her algoritma için gürbüz parametre düzeni bulmaya çalışmanın, daha adaletli bir değerlendirme yapmayı sağlayacağı açıkça görülmektedir.

Değerleme yöntemi olarak önerilen yöntemlerin avantajı doğrudan pareto-önyüzleri ele almasıdır. Yapılan ikili karşılaştırmalarda ortaya çıkan birleştirilmiş pareto-önyüzlerde iki algoritmanın da karar seçeneklerinin yer alması, yani bir algoritmanın diğerine baskın olmaması durumunda oldukça adaletli bir karşılaştırma yapılabilir.

Ancak bir algoritmanın diğerine baskın olduğu durumlarda yapılacak ikili karşılaştırma ve önem değeri hesabının nasıl olması gerektiğine dair konularda çalışmakta fayda olduğu açıktır. Belki de özellikle bu durumlar için sözü geçen 3 metriği birleştirmeye çalışmak yollarını aramak daha avantajlı olacaktır. İleri sürülen ikili karşılaştırma yoluyla performans değerlendirme yönteminin en önemli dezavantajı bu olmakla birlikte, tutarlılık oranlarına baktığımızda bu örnekler için bu dezavantajın karşılaştırma yapamayacak oranda meydana gelmediğini söylemek mümkündür. Bunun sebebi bu tür baskın durumlar için pareto-önyüzde bulunan karar seçeneği ortalama sayısını dikkate almamız ve bu karar seçeneklerinin sayısının algoritmadan ziyade problem özellikli (çözümün uzlaşık, çok parçalı veya sıkı dağılımlı olup olmadığı) nedenlerden belirleniyor olmasıdır. Özellikle performans karşılaştırmalarını istatistiksel bakış açısıyla değerlendirmenin zor olduğu, çoklu önem testlerine ve varyans analiziyle yapılacak çalışmalara hala ihtiyaç olduğu çalışma içerisinde de ifade edilmişti. İlerideki çalışmalarda bu konular üzerinde durmakta fayda olduğu görülmektedir.

Özellikle bu çalışmada ortaya çıkan bir konu da ÇAKKO algoritmalarından, feromon güncelleme stratejilerini hem klasik tek amaçlı sistemlerdekinden farklı hem de özellikle pareto-önyüze yönelik olarak oluşturan ve bu stratejileri oluştururken de pareto-önyüzü homojen olarak arayabilen yöntemlerin İKKAP problemlerini çözerken oldukça başarılı olacaktır. Bundan dolayı özellikle ilk denemelerde PKKO ve İKKA algoritmalarını en başarılı algoritmalar olarak yer aldığını gördük. İKKA'ya çalışmamızda önerdiğimiz *poyraz* eklentisiyle ortaya çıkan ÇIKKA-*poyraz* algoritmasının bu çalışmada açıkça görülen başarısındaki temel sebep, pareto-önyüz için önerdiğimiz değerlendirme kriterinin feromon güncelleme kuralına en gerçekçi ve akılcı şekilde uyarlanabilmiş olmasıdır. Bu net başarısına rağmen, bu algoritmadaki *poyraz* değerinin, oldukça nadiren de olsa 1'den büyük değer alması konusunda çalışılması durumunda, algoritma az da olsa daha iyi sonuçlar ortaya koyabilir. Çözüm üretirken pareto-önyüzü dikkate almayan yöntemlerin çok başarılı olamadığı ve hatta bazı algoritmaların özellikle kötü hesapsal yapılarından ötürü oldukça başarısız oldukları tespit edilmiştir. Bu konuda özellikle ÇAAQA algoritması dikkat çekici bir başarısızlığa sahip olmuştur.

KAYNAKLAR

Abbiw-Jackson, R., Golden, B., Raghavan, S., Wasil, E., “A divide-and-conquer local search heuristic for data visualization”, *Computers and Operations Research*, 33:11, 3070-3087, (2006).

Abreu, N., M., M., Boaventura-Netto, P., O., Querido, T., M., Gouvea, E., F., “Classes of quadratic assignment problem instances: Isomorphism and difficulty measure using a statistical approach”, *Discrete Applied Mathematics*, 124:1-3, 103-116, (2002).

Abreu, N., M., M., Querido, T., M., Boaventura-Netto, P., O., “RedInv-SA: A simulated annealing for the quadratic assignment problem”, *RAIRO Operations Research*, 33:3, 249-273, (1999).

Acan, A., “An external partial permutations memory for ant colony optimization”, *Lecture Notes in Computer Science*, 3448, 1-11, (2005).

Adams, W., P., Guignard, M., Hahn, P., M., Hightower, W.L., “A level-2 reformulation-linearization technique bound for the quadratic assignment problem”, *European Journal of Operational Research*, 180:3, 983-996, (2007).

Adams, W., P., Johnson, T., A., “Improved linear programming-based lower bounds for the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 43-75, (1994).

Adams, W., P., Sherali, H., D., “A tight linearization and an algorithm for zero-one quadratic programming problems”, *Management Science*, 32:10, 1274-1290, (1986).

Adams, W., P., Sherali, H., D., “Linearization strategies for a class of zero-one mixed integer programming problems”, *Operations Research*, 38:2, 217-226, (1990).

Ahuja, R., Orlin, J., B., Tiwari, A., “A greedy genetic algorithm for the quadratic assignment problem”, *Computers and Operations Research*, 27:10, 917-934, (2000).

Anderson, E., J., “Theory and methodology: Mechanisms for local search”, *European Journal of Operational Research*, 88, 139-151, (1996).

Angel, E., Zissimopoulos, V., “On the quality of local search for the quadratic assignment problem”, *Discrete Applied Mathematics*, 82:1-3, 15-25, (1998).

Angel, E., Zissimopoulos, V., “On the classification of NP-complete problems in terms of their correlation coefficient”, *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 99:1–3, 261–277, (2000).

Angel, E., Zissimopoulos, V., “On the landscape ruggedness of the quadratic assignment problem”, *Theoretical Computer Science*, 263:1–2, 159–172, (2001).

Angel, E., Zissimopoulos, V., “On the hardness of the quadratic assignment problem with metaheuristics”, *Journal of Heuristics*, 8:4, 399–414, (2002).

Anstreicher, K., M., “Eigenvalue bounds versus semidefinite relaxations for the quadratic assignment problem”, *SIAM Journal on Optimization*, 11:1, 254–265, (2001).

Anstreicher, K., M., Brixius, N., W., “A new bound for the quadratic assignment problem based on convex quadratic programming”, *Mathematical Programming*, 89:3, 341–357, (2001).

Anstreicher, K., M., Brixius, N., W., Goux, J., P., Linderoth, J., “Solving large quadratic assignment problems on computational grids”, *Mathematical Programming*, 91:3, 563–588, (2002).

Anstreicher, K., M., Chen, X., Wolkowicz, H., Yuan, Y., “Strong duality for a trust-region type relaxation of the quadratic assignment problem”, *Linear Algebra and its Applications*, 301:1–3, 121–136, (1999).

Assad, A., A., Xu, W., “On lower bounds for a class of quadratic $\{0,1\}$ programs”, *Operations Research Letters*, 4:4, 175–180, (1985).

Arkin, E., M., Hassin, R., Sviridenko, M., “Approximating the maximum quadratic assignment problem”, *Information Processing Letters*, 77:1, 13–16, (2001).

Armour, G., C., Buffa, E., S., “Heuristic algorithm and simulation approach to relative location of facilities”, *Management Science*, 9:2, 294–309, (1963).

Balakrishnan, J., Cheng, C., H., Conway, D., G., Lau, C., M., “A hybrid genetic algorithm for the dynamic plant layout problem”, *International Journal of Production Economics*, 86:2, 107–120, (2003).

Balakrishnan, J., Jacobs, F., R., Venkataramanan, M., A., “Solutions for the constrained dynamic facility layout problem”, *European Journal of Operational Research*, 15, 280–286, (1992).

Balas, E., Saltzman, M., J., “Facets of the three-index assignment polytope”, *Discrete Applied Mathematics*, 23, 201–229, (1989).

Balas, E., Saltzman, M., J., “An algorithm for the three-index assignment problem”, *Operations Research*, 39:1, 150–161, (1991).

- Balas, E., Qi, L., “Linear-time separation algorithms for the three-index assignment polytope”, *Discrete Applied Mathematics*, 43, 1–12, (1993).
- Ball, M., O., Kaku, B., K., Vakhutinsky, A., “Network-based formulations of the quadratic assignment problem”, *European Journal of Operational Research*, 104:1, 241–249, (1998).
- Bandelt, H., J., Crama, Y., Spieksma, F., C., R., “Approximation algorithms for multi-dimensional assignment problems with decomposable costs”, *Discrete Applied Mathematics*, 49, 25–50, (1991).
- Barán, B., Schaerer, M., “A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows”, *Proc. Twenty first IASTED International Conference on Applied Informatics*, Innsbruck, Austria, February 10-13, 97-102, (2003).
- Bartolomei-Suarez, S., M., Egbelu, P., J., “Quadratic assignment problem QAP with adaptable material handling devices”, *International Journal of Production Research*, 38:4, 855–873, (2000).
- Barvinok, A., Stephen, T., “The distribution of values in the quadratic assignment problem”, *Mathematics of Operations Research*, 28, 64–91, (2003).
- Battiti, R., Tecchiolli, G., “Simulated annealing and tabu search in the long run: A comparison on QAP tasks”, *Computer and Mathematics with Applications*, 28:6, 1-8, (1994).
- Bazaraa, M., S., Elshafei, A., N., “An exact branch-and-bound procedure for the quadratic assignment problem”, *Naval Research Logistics Quarterly*, 26, 109–121, (1979).
- Bazaraa, M., S., Kirca, O., “A branch-and-bound based heuristic for solving the quadratic assignment problem”, *Naval Research Logistics Quarterly*, 30, 287–304, (1983).
- Bazaraa, M., S., Sherali, H., D., “New approaches for solving the quadratic assignment problem”, *Operations Research Verfahren*, 32, 29–46, (1979).
- Bazaraa, M., S., Sherali, H., D., “Benders’ partitioning scheme applied to a new formulation of the quadratic assignment problem”, *Naval Research Logistics Quarterly*, 27, 29–41, (1980).
- Bazaraa, M., S., Sherali, H., D., “On the use of exact and heuristic cutting plane methods for the quadratic assignment problem”, *Journal of the Operational Research Society*, 33, 991–1003, (1982).

- Ben-David, G., Malah, D., “Bounds on the performance of vector-quantizers under channel errors”, *IEEE Transactions on Information Theory*, 51:6, 2227–2235, (2005).
- Benjaafar, S., “Modeling and analysis of congestion in the design of facility layouts”, *Management Science*, 48:5, 679–704, (2002).
- Billionnet, A., Elloumi, S., “Best reduction of the quadratic semi-assignment problem”, *Discrete Applied Mathematics*, 109:3, 197–213, (2001).
- Bland, J., A., Dawson, G., P., “Tabu search and design optimization”, *Computer Aided Design*, 23:3, 195–201, (1991).
- Bland, J., A., Dawson, G., P., “Large-scale layout of facilities using a heuristic hybrid algorithm”, *Applied Mathematical Modeling*, 18:9, 500–503, (1994).
- Boaventura-Netto, P., O., “Combinatorial instruments in the design of a heuristic for the quadratic assignment problems”, *Pesquisa Operacional*, 23:3, 383–402, (2003).
- Bos, J., “A quadratic assignment problem solved by simulated annealing”, *Journal of Environmental Management*, 37:2, 127–145, (1993).
- Bousonocalzon, C., Manning, M., R., W., “The Hopfield neural-network applied to the quadratic assignment problem”, *Neural Computing and Applications*, 3:2, 64–72, (1995).
- Bozer, Y., A., Suk-Chul, R., “A branch and bound method for solving the bidirectional circular layout problem”, *Applied Mathematical Modeling*, 20:5, 342–351, (1996).
- Bölte, A., Thonemann, U., W., “Optimizing simulated annealing schedules with genetic programming”, *European Journal of Operational Research*, 92:2, 402–416, (1996).
- Brixius, N., W., Anstreicher, K., M., “Solving quadratic assignment problems using convex quadratic programming relaxations”, *Optimization Methods and Software*, 16, 49–68, (2001).
- Brown, D., E., Huntley, C., L., “A parallel heuristic for the quadratic assignment problem”, *Computers and Operations Research*, 18, 275–289, (1991).
- Brujjs, P., A., “On the quality of heuristic solutions to a $19 \cdot 19$ quadratic assignment problem”, *European Journal of Operational Research*, 17, 21–30, (1984).
- Brusco, M., J., Stahl, S., “Using quadratic assignment methods to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices”, *Journal of Classification*, 17:2, 197–223, (2000).

Brünger, A., Marzetta, A., Clausen, J., Perregaard, M., “Joining forces in solving large-scale quadratic assignment problems”, In: Proceedings of the 11th International Parallel Processing Symposium IPPS, *IEEE Computer Society Press*, 418–427, (1997).

Brünger, A., Marzetta, A., Clausen, J., Perregaard, M., “Solving large-scale QAP problems in parallel with the search library ZRAM”, *Journal of Parallel and Distributed Computing*, 50:1–2, 157–169, (1998).

Buffa, E., S., Armour, G., C., Vollmann, T., E., “Allocating facilities with CRAFT”, *Harvard Business Review*, 42:2, 136–158, (1964).

Bui, T., N., Moon, B., R., “A genetic algorithm for a special class of the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 99–116, (1994).

Bullnheimer, B., “An examination-scheduling model to maximize students’ study time”, *Lecture Notes in Computer Science*, 1408, 78–91, (1998).

Bullnheimer, B., Kotsis, G., Strauss, C., “A New Rank-based Version of the Ant System: A Computational Study”, *Central European Journal for Operations Research and Economics*, 7:1, 25–38, (1999).

Burer, S., Vandebussche, D., 2004, Solving lift-and-project relaxations of binary integer programs [online], http://www.optimization-online.org/DB_HTML/2004/06/890.html (**Ziyaret tarihi: 3 Nisan 2009**).

Burkard, R., E., “ Quadratic assignment problems”, *European Journal of Operational Research*, 15, 283–289, (1984).

Burkard, R., E., “Locations with spatial interactions: The quadratic assignment problem”, Mirchandani, P., B., Francis, R., L. (Eds.), Discrete Location Theory, *John Wiley and Sons*, 387–437, (1991).

Burkard, R., E., “Selected topics on assignment problems”, *Discrete Applied Mathematics*, 123:1–3, 257–302, (2002).

Burkard, R., E., Bonniger, T., “A heuristic for quadratic Boolean programs with applications to quadratic assignment problems”, *European Journal of Operation Research*, 13, 374–386, (1983).

Burkard, R., E., Çela, E., “Heuristics for biquadratic assignment problems and their computational comparison”, *European Journal of Operational Research*, 83:2, 283–300, (1995).

Burkard, R.E., Çela, E., “Quadratic and three-dimensional assignment problems: An annotated bibliography”, In: Dell’Amico, M., Maffioli, F., Martello, S. (Eds.),

Annotated Bibliographies in Combinatorial Optimization, **Wiley**, Chichester, 373–392, (1996).

Burkard, R., E., Çela, E., Klinz, B., “On the biquadratic assignment problem”, Pardalos, P.M., Wolkowicz, H. (Eds.), *Quadratic Assignment and Related Problems*, **DIMACS Series in Discrete Mathematics and Theoretical Computer Science**, 16, AMS, Rhode Island, 117–146, (1994).

Burkard, R., E., Çela, E., Rote, G., Woeginger, G., J., “The quadratic assignment problem with a monotone Anti-Monge and asymmetric Toeplitz matrix: Easy and hard cases”, *Lecture Notes in Computer Science*, 1084, 204–218, (1996a).

Burkard, R., E., Derigs, U., “Assignment and matching problems: Solutions methods with Fortran programs”, *Lectures Notes in Economics and Mathematical Systems*, 184, **Springer-Verlag**, New York, Secaucus, (1980).

Burkard, R., E., Euler, R., Grommes, R., “On Latin squares and the facial structure of related polytopes”, *Discrete Mathematics*, 62, 155–181, (1986).

Burkard, R., E., Finke, G., “On random quadratic bottleneck assignment problems”, *Mathematical Programming*, 23, 227–232, (1982).

Burkard, R., E., Karisch, S., Rendl, F., “QAPLIB—A quadratic assignment problem library”, *European Journal of Operational Research*, 55, 115–119, (1991).

Burkard, R., E., Karisch, S., Rendl, F., “QAPLIB—A quadratic assignment problem library”, *Journal of Global Optimization*, 10, 391–403, (1997).

Burkard, R., E., Rendl, F., “A thermodynamically motivated simulation procedure for combinatorial optimization problems”, *European Journal of Operational Research*, 17:2, 169–174, (1984).

Burkard, R., E., Rudolf, R., “Computational investigations on 3-dimensional axial assignment problems”, *Belgian Journal of Operations Research, Statistics and Computer Science*, 32, 85–98, (1993).

Burkard, R., E., Rudolf, R., Woeginger, G., J., “Three-dimensional axial assignment problems with decomposable cost coefficients”, *Discrete Applied Mathematics*, 65, 123–139, (1996b).

Burkard, R., E., Stratman, R., H., “Numerical investigations on quadratic assignment problem”, *Naval Research Logistics Quarterly*, 25, 129–140, (1987).

Burkard, R., E., Zimmermann, U., “Combinatorial optimization in linearly ordered semimodules: A survey”, Korte, B. (Ed.), “Modern Applied Mathematics: Optimization and Operations Research”, North Holland, Amsterdam, 392–436, (1982).

Burer, S., Vandenbussche, D., “Solving lift-and-project relaxations of binary integer programs”, *Optimization Online*, http://www.optimization-online.org/DB_HTML/2004/06/890.html (**Ziyaret tarihi: 22 Mart 2009**).

Cardoso, P., Jesús, M., Márquez, A., “MONACO - Multi-Objective Network Optimisation Based on an ACO”, *Proc.. X Encuentros de Geometría Computacional*, Seville, Spain, June 16-17, (2003).

Carraresi, P., Malucelli, F., “A new lower bound for the quadratic assignment problem”, *Operations Research*, 40:1, 22–27, (1992).

Carraresi, P., Malucelli, F., “A reformulation scheme and new lower bounds for the QAP”, Pardalos, P.M., Wolkowicz, H. (Eds.), *Quadratic Assignment and Related Problems*, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 147–160, (1994).

Chakrapani, J., Skorin-Kapov, J., “Massively parallel tabu search for the quadratic assignment problem”, *Annals of Operations Research*, 41:1–4, 327–342, (1993).

Chakrapani, J., Skorin-Kapov, J., “A constructive method to improve lower bounds for the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), *Quadratic Assignment and Related Problems*, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 161–171, (1994).

Chankong, V., Haimes, Y., Y., “Multiobjective Decision Making Theory and Methodology”, *North-Holland*, (1983).

Chen, B., “Special cases of the quadratic assignment problem”, *European Journal of Operational Research*, 81:2, 410–419, (1995).

Chiang, W., C., Chiang, C., “Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation”, *European Journal of Operational Research*, 106:2–3, 457–488, (1998).

Christofides, N., Benavent, E., “An exact algorithm for the quadratic assignment problem”, *Operations Research*, 37:5, 760–768, (1989).

Christofides, N., Gerrard, M., “A graph theoretic analysis of bounds for the quadratic assignment problem”, In: Hansen, P. (Ed.), *Studies on Graphs and Discrete Programming*, North-Holland, 61–68, (1981).

Christofides, N., Mingozzi, A., Toth, P., “Contributions to the quadratic assignment problem”, *European Journal of Operational Research*, 4, 243–247, (1980).

Ciriani, V., Pisanti, N., Bernasconi, A., “Room allocation: A polynomial subcase of the quadratic assignment problem”, *Discrete Applied Mathematics*, 144:3, 263–269, (2004).

- Clausen, J., Perregaard, M., “Solving large quadratic assignment problems in paralel”, *Computational Optimization and Applications*, 8, 111–127, (1997).
- Clausen, J., Karisch, S., E., Perregaard, M., Rendl, F., “On the applicability of lower bounds for solving rectilinear quadratic assignment problems in parallel”, *Computational Optimization and Applications*, 10:2, 127–147, (1998).
- Coello, C., A., Van Veldhuizen, D., A., Lamant, G., B., “Evolutionary Algorithms for Solving Multiobjective Problems”, *Kluwer*, (2002).
- Coloni, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G., Trubian, M., “Heuristics from nature for hard combinatorial optimization problems”, *International Transactions in Operational Research*, 3:1, 1–21, (1996).
- Connolly, D., T., “An improved annealing scheme for the QAP”, *European Journal of Operational Research*, 46, 93–100, (1990).
- Cordón, O., Fernández de Viana, I., Herrera, F., Moreno, L., “A New ACO model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System”, Dorigo, M., Middendorf, M., Stützle, T., *Proc. of ANTS2000 - From Ant Colonies to Artificial Ants*, Brussels, Belgium, September, 22-29, (2000).
- Cordón, O., Fernández de Viana, I., Herrera, F., “Analysis of the Best-Worst Ant System and its Variants on the TSP”, *Mathware & Soft Computing*, 9:2-3, 177-192, (2002a).
- Cordón, O., Fernández de Viana, I., Herrera, F., “Analysis of the Best-Worst Ant System and its Variants on the QAP”, Dorigo, M., Di Caro, G., Sampels, M. (Eds.), “Ant Algorithms”, Proc. of ANTS2002, Lecture Notes in Computer Science, 2463, *Springer Verlag*, Berlin, Germany, 228-234, (2002b).
- Cordón, O., Herrera, F., Stützle, T., “A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends”, *Mathware & Soft Computing*, 9:2-3, 141-175, (2002).
- Costa, C., S., Boaventura-Netto, P., O., “An algebraic-combinatorial description for the asymmetric quadratic assignment problem”, *Advances in Modeling and Analysis A*, 22:2, 1–11, (1994).
- Crama, Y., Spieksma, F., C., R., “Approximation algorithms for three-dimensional assignment problems with triangle inequalities”, *European Journal of Operational Research*, 60, 273–279, (1992).
- Cung, V., D., Mautor, T., Michelon, P., Tavares, A., “A scatter search based approach for the quadratic assignment problem”, *Proceedings of IEEE International Conference on Evolutionary Computation*, 165–169, (1997).

Cyganski, D., Vaz, R., F., Virball, V., G., “Quadratic assignment problems generated with the Palubetskis algorithm are degenerate”, *IEEE Transactions on Circuits and Systems I—Fundamental Theory and Applications*, 41:7, 481–484, (1994).

Çela, E., “The quadratic assignment problem: Theory and algorithms”, Du, D., Z., Pardalos, P. (Eds.), *Combinatorial Optimization*, **Kluwer Academic Publishers**, Dordrecht, (1998).

Davis, L., “Genetic Algorithms and Simulated Annealing”, **Morgan Kaufmann Publishers**, (1987).

Day, R., O., Lamont, G., B., “Multiobjective quadratic assignment problem solved by an explicit building block search algorithm— MOMGA-IIa”, *Lecture Notes in Computer Science*, 3448, 91–100, (2005).

Deineko, V., G., Woeginger, G., J., “A solvable case of the quadratic assignment problem”, *Operations Research Letters*, 22:1, 13–17, (1998).

Deineko, V., G., Woeginger, G., J., “A study of exponential neighborhoods for the traveling salesman problem and for the quadratic assignment problem”, *Mathematical Programming Series A*, 78, 519–542, (2000).

Dickey, J., W., Hopkins, J., W., “Campus building arrangement using Topaz”, *Transportation Research*, 6, 59–68, 1972

Doerner, K., Gutjahr, W., J., Hartl, R., F., Strauss, C., Stummer, C., “Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection”, *Annals of Operations Research*, 131:1-4, 79-99, (2004).

Doerner, K., Hartl, R., F., Teimann, M., “Are COMPETants More Competent for Problem Solving? - The Case of Full Truckload Transportation”, *Central European Journal of Operations Research*, 11:2, 115-141, (2003).

Dorigo, M., Di Caro, G., “The Ant Colony Optimization Meta-heuristic”, Corne, D., Dorigo, M., Glover, F., *New Ideas in Optimization*, **McGraw Hill**, London, UK, 11-32, (1999).

Dorigo, M., Gambardella, L., “Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem”, *IEEE Transactions on Evolutionary Computation*, 1:1, 53-66, (1997).

Dorigo, M., Maniezzo, V., Colorni, A., “The Ant System: Optimization by a Colony of Cooperating Agents”, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26:1, 29-41, (1996).

Dorigo, M., Stützle, T., “The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances”, Glover, F., Kochenberger, G., A. (Eds.), *Handbook of Metaheuristics*, **Kluwer**, 251-258, (2003).

- Dorigo, M., Stützle, T., “Ant Colony Optimization”, *MIT Press*, Cambridge, MA, (2004).
- Drezner, Z., “Lower bounds based on linear programming for the quadratic assignment problem”, *Computational Optimization and Applications*, 4:2, 159–165, (1995).
- Drezner, Z., “A new genetic algorithm for the quadratic assignment problem”, *Inform Journal on Computing*, 15:3, 320–330, (2003).
- Drezner, Z., “Compounded genetic algorithms for the quadratic assignment problem”, *Operations Research Letters*, 33:5, 475–480, (2005a).
- Drezner, Z., “The extended concentric tabu for the quadratic assignment problem”, *European Journal of Operational Research*, 160, 416–422, (2005b).
- Drezner, Z., Hahn, P., Taillard, E., “A study of quadratic assignment problem instances that are difficult for meta-heuristic methods”, In: Guignard-Spielberg, M., Spielberg, K. (Eds.), *Annals of Operations Research, Special issue devoted to the State-of-the-Art in Integer Programming*, (2004).
- Duman, E., Ilhan, O., “The quadratic assignment problem in the context of the printed circuit board assembly process”, *Computers and Operations Research*, 34:1, 163-179, (2007).
- Dunker, T., Radons, G., Westkamper, E., “Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem”, *European Journal of Operational Research*, 165:1, 55–69, (2004).
- Edwards, C., S., “A branch and bound algorithm for the Koopmans–Beckmann quadratic assignment problem”, *Mathematical Programming Study*, 13, 35–52, (1980).
- El-Baz, M., A., “A genetic algorithm for facility layout problems of different manufacturing environments”, *Computers and Industrial Engineering*, 47:2–3, 233–246, (2004).
- Elshafei, A., N., “Hospital layout as a quadratic assignment problem”, *Operations Research Quarterly*, 28:1, 167–179, (1977).
- Emelichev, V., A., Kovalev, M., N., Kravtsov, M., K., “Polytopes”, Graphs and Optimization, *Cambridge University Pres*, (1984).
- Euler, R., “Odd cycles and a class of facets of the axial 3-index assignment polytope”, *Applicationes Mathematicae (Zastosowania Matematyki)*, 19, 375–386, (1987).

- Fedjki, C., A., Duffuaa, S., O., “An extreme point algorithm for a local minimum solution to the quadratic assignment problem”, *European Journal of Operational Research*, 156:3, 566–578, (2004).
- Feo, T., A., Resende, M., G., C., “Greedy randomized adaptive search procedures”, *Journal of Global Optimization*, 6, 109–133, (1995).
- Fıđlalı, N., Özkale, C., Engin, O., Fıđlalı, A., “Investigation of Ant System parameter interactions by using design of experiments for job-shop scheduling problems”, *Computers & Industrial Engineering*, 56:2, 538-559, (2009).
- Finke, G., Burkard, R., E., Rendl, F., “Quadratic assignment problems”, *Annals of Discrete Mathematics*, 31, 61–82, (1987).
- Fleurent, C., Ferland, J., A., “Genetic hybrids for the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 173–187, (1994).
- Fleurent, C., Glover, F., “Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory”, *INFORMS Journal on Computing*, 11, 189–203, (1999).
- Forsberg, J., H., Delaney, R., M., Zhao, Q., Harakas, G., Chandran, R., “Analyzing lanthanide-included shifts in the NMR spectra of lanthanide (III) complexes derived from 1,4,7,10-tetrakis (N,N-diethylacetamido)-1,4,7,10-tetraazacyclododecane”, *Inorganic Chemistry* 34, 3705–3715, (1994).
- Francis, R., L., White, J., A., “Facility Layout and Location: An Analytical Approach”, *Prentice-Hall Englewood Cliffs*, (1974).
- Freeman, R., J., Gogerty, D., C., Graves, G., W., Brooks, R., B., S., “A mathematical model of supply for space operations”, *Operations Research*, 14, 1–15, (1966).
- Frenk, J., B., G., Houweninge, M., V., Kan, A., H., G., R., “Asymptotic properties of the quadratic assignment problem”, *Mathematics of Operations Research*, 10:1, 100–116, (1985).
- Frieze, A., M., “A bilinear programming formulation of the 3-dimensional assignment problems”, *Mathematical Programming*, 7, 376–379, (1974).
- Frieze, A., M., “Complexity of a 3-dimensional assignment problem”, *European Journal of Operational Research*, 13, 161–164, (1983).
- Frieze, A., M., Yadegar, J., “An algorithm for solving 3-dimensional assignment problems with applications to scheduling a teaching practice”, *Operations Research*, 32, 989–995, (1981).

- Frieze, A., M., Yadegar, J., “On the quadratic assignment problem”, *Discrete Applied Mathematics*, 5, 89–98, (1983).
- Gambardella, L., Dorigo, M., “Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem”, *Proc. Twelfth International Conference on Machine Learning (ML-95)*, Tahoe City, CA, USA, 252-260, (1995).
- Gambardella, L., Taillard, E., Agazzi, G., “MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows”, Corne, D., Dorigo, M., Glover, F., (Eds.), *New Ideas in Optimization, McGraw-Hill*, 73-76, (1999),
- Gambardella, L., M., Taillard, D., Dorigo, M., “Ant colonies for the QAP”, *Journal of the Operational Research Society*, 50, 167–176, (1999).
- Gandibleux, X., Sevaux, M., Sorensen, K., T'kindt, V. (Eds.) “Metaheuristics for Multiobjective Optimisation”, *Lecture Notes in Economics and Mathematical Systems, vol:535*, Springer Verlag, (2004).
- Gavett, J., W., Plyter, N., V., “The optimal assignment of facilities to locations by branch-and-bound”, *Operations Research*, 14, 210–232, (1966).
- Geoffrion, A., M., Graves, G., W., “Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach”, *Operations Research*, 24, 595–610, (1976).
- Gilmore, P., C., “Optimal and suboptimal algorithms for the quadratic assignment problem”, *SIAM Journal on Applied Mathematics*, 10, 305–313, (1962).
- Glover, F., “Heuristics for integer programming using surrogate constraints”, *Decision Science*, 8, 156–166, (1977).
- Glover, F., “Tabu search—Part I”, *ORSA Journal on Computing*, 1, 190–206, (1989a).
- Glover, F., “Tabu search—Part II”, *ORSA Journal on Computing*, 2, 4–32, (1989b).
- Goldberg, D.E., “Genetic Algorithms in Search, Optimization and Machine Learning”, *Addison-Wesley*, Wokingham, England, (1989).
- Gong, D., Yamazaki, G., Gen, M., Xu, W., “A genetic algorithm method for one-dimensional machine location problems”, *International Journal of Production Economics*, 60:1, 337–342, (1999).
- Gouveia, L., VoB, S., “A classification of formulations for the (time-dependent) traveling salesman problem”, *European Journal of Operational Research*, 83:1, 69–82, (1995).

Gutin, G., Yeo, A., “Polynomial approximation algorithms for TSP and QAP with a factorial domination number”, *Discrete Applied Mathematics*, 119:1–2, 107–116, (2002).

Gravel, M., Price, W., L., Gagné, C., “Scheduling Continuous Casting of Aluminium using a Multiple Objective Ant Colony Optimization Metaheuristic”, *European Journal of Operational Research*, 143:1, 218–229, (2002).

Graves, G., W., Whinston, A., B., “An algorithm for the quadratic assignment problem”, *Management Science*, 17:7, 453–471, (1970).

Hadley, S., W., “Domination and separation applied to the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), *Quadratic Assignment and Related Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 189–196, (1994).

Hadley, S., W., Rendl, F., Wolkowicz, H., “Bounds for the quadratic assignment problem using continuous optimization techniques”, *Integer Programming and Combinatorial Optimization, University of Waterloo Press*, 237–248, (1990).

Hadley, S., W., Rendl, F., Wolkowicz, H., “Nonsymmetric quadratic assignment problems and the Hoffman–Wielandt inequality”, *Linear Algebra and its Applications*, 58, 109–124, (1992a).

Hadley, S., W., Rendl, F., Wolkowicz, H., “A new lower bound via projection for the quadratic assignment problem”, *Mathematics of Operations Research*, 17, 727–739, (1992b).

Hadley, S., W., Rendl, F., Wolkowicz, H., “Symmetrization of nonsymmetric quadratic assignment problems and the Hoffman–Wielandt inequality”, *Linear Algebra and its Applications*, 167, 53–64, (1992c).

Haghani, A., Chen, M., C., “Optimizing gate assignments at airport terminals”, *Transportation Research A*, 32:6, 437–454, (1998).

Hahn, P., M., 2000, Progress in solving the Nugent instances of the quadratic assignment problem, <http://www-unix.mcs.anl.gov/metaneos/nug30/nug30.pdf> (Ziyaret tarihi 23 Kasım 2005).

Hahn, P., Grant, T., “Lower bounds for the quadratic assignment problem based upon a dual formulation”, *Operations Research*, 46, 912–922, (1998).

Hahn, P., M., Grant, T., Hall, N., “A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method”, *European Journal of Operational Research*, 108, 629–640, (1998).

Hahn, P., M., Hightower, W., L., Johnson, T., A., Guignard-Spielberg, M., Roucairol, C., “Tree elaboration strategies in branch and bound algorithms for

solving the quadratic assignment problem”, *Yugoslavian Journal of Operational Research*, 11:1, 41–60, (2001a).

Hahn, P., M., Hightower, W., L., Johnson, T., A., Guignard-Spielberg, M., Roucairol, C. “A level-2 reformulation-linearization technique bound for the Quadratic Assignment Problem”, Working Paper 01-04, *Systems Engineering Department, University of Pennsylvania*, (2001b).

Hahn, P., M., Krarup, J., “A hospital facility layout problem finally solved”, *Journal of Intelligent Manufacturing*, 12, 487–496, (2001).

Hans, A., E., “Multicriteria Optimization for Highly Accurate Systems”, Stadler, W. (Ed.), *Multicriteria Optimization in Engineering and Sciences, Mathematical Concepts and Methods in Science and Engineering*, 19, *Plenum Press*, 309-352, (1988).

Hansen, P., Lih, K. W., “Improved algorithms for partitioning problems in parallel, pipelined, and distributed computing”, *IEEE Transactions on Computers*, 41:6, 769–771, (1992).

Hasegawa, M., Ikeguchi, T., Aihara, K., Itoh, K., “A novel chaotic search for quadratic assignment problems”, *European Journal of Operational Research*, 139:3, 543–556, (2002).

Heffley, D., R., “The quadratic assignment problem: A note”, *Econometrica*, 40:6, 1155–1163, (1972).

Heffley, D.R., “Assigning runners to a relay team.”, *In: Ladany, S.P., Machol, R.E. (Eds.), Optimal Strategies in Sports*, North Holland, Amsterdam, 169–171, (1977).

Heffley, D., R., “Decomposition of the Koopmans–Beckmann problem”, *Regional Science and Urban Economics*, 10:4, 571–580, (1980).

Heider, C., H., “An N-step, 2-variable search algorithm for the component placement problem”, *Naval Research Logistics Quarterly*, 20, 699–724, (1973).

Herroeleven, W., Vangils, A., “On the use of flow dominance in complexity measures for facility layout problems”, *International Journal of Production Research*, 23, 97–108, (1985).

Hillier, F., S., Michael, M., C., “Quadratic assignment problem algorithms and the location of indivisible facilities”, *Management Science*, 13, 44–57, (1966).

Ho, Y., C., Moodie, C., L., “A hybrid approach for concurrent layout design of cells and their flow paths in a tree configuration”, *International Journal of Production Research*, 38:4, 895–928, (2000).

Hubert, L., “Assignment methods in combinatorial data analysis”, *Statistics: Textbooks and Monographs Series, vol:73*, Marcel Dekker, (1987).

- Hubert, L., Schulz, J., “Quadratic assignment as a general data analysis strategy” *British Journal of Mathematical Psychology*, 29, 190–241, (1976).
- Huntley, C., L., Brown, D., E., “Parallel genetic algorithms with local search”, *Computers and Operations Research*, 23:6, 559–571, (1996).
- Ishii, S., Sato, M., “Constrained neural approaches to quadratic assignment problems”, *Neural Networks*, 11:6, 1073–1082, (1998).
- Ishii, S., Sato, M., “Doubly constrained network for combinatorial optimization”, *Neurocomputing*, 43:1–4, 239–257, (2001).
- Iredi, S., Merkle, D., Middendorf, M., “Bi-Criterion Optimization with Multi Colony Ant Algorithms”, Proc. First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), *Lecture Notes in Computer Science*, 1993, 359–372, (2001).
- Jaszkiewicz, A., “Multiple Objective Metaheuristic Algorithms for Combinatorial Optimization”, Habilitation thesis, 360, *Poznan University of Technology*, (2001).
- Jünger, M., Kaibel, V., “On the SQAP-polytope”, *SIAM Journal on Optimization*, 11:2, 444–463, (2000).
- Jünger, M., Kaibel, V., “The QAP-polytope and the star transformation”, *Discrete Applied Mathematics*, 111:3, 283–306, (2001a).
- Jünger, M., Kaibel, V., “Box-inequalities for quadratic assignment polytopes”, *Mathematical Programming*, 91:1, 175–197, (2001b).
- Kaibel, V., “Polyhedral combinatorics of the quadratic assignment problem”, Doktora tezi, *University of Cologne*, Cologne, Germany, (1997a).
- Kaibel, V., “Polyhedral combinatorics of QAPs with less objects than locations”, *Technical Report Nr. 97-297*, Angewandte Mathematik und Informatik, Universitaet zu Koeln, Cologne, Germany, (1997b).
- Kaibel, V., “Polyhedral combinatorics of quadratic assignment problems with less objects than locations”, *Lecture Notes in Computer Science*, 1412, 409–422, (1998).
- Kaku, B., K., Thompson, G., L., “An exact algorithm for the general quadratic assignment problem”, *European Journal of Operational Research*, 2, 382–390, (1986).
- Karisch, S., E., Çela, E., Clausen, J., Espersen, T., “A dual framework for lower bounds of the quadratic assignment problem based on linearization”, *Computing*, 63, 351–403, (1999).

Karisch, S., E., Rendl, F., “Lower bounds for the quadratic assignment problem via triangle decompositions”, *Mathematical Programming*, 71, 137–152, (1995).

Karisch, S., E., Rendl, F., Wolkowicz, H., “Trust regions and relaxations for the quadratic assignment problem”, Pardalos, H., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 199–219, (1994).

Karmarkar, N., K., Ramakrishnan, K., G., “Computational results of an interior point algorithm for large-scale linear programming”, *Mathematical Programming*, 52, 555–586, (1991).

Kaufman, L., Broeckx, F., “An algorithm for the quadratic assignment problem using Bender’s decomposition”, *European Journal of Operation Research*, 2, 204–211, (1978).

Kellerer, H., Wirsching, G., “Bottleneck quadratic assignment problems and the bandwidth problem”, *Asia-Pacific Journal of Operational Research*, 15, 169–177, (1998).

Kelly, J., P., Laguna, M., Glover, F., “A study of diversification strategies for the quadratic assignment problem”, *Computers and Operations Research*, 21:8, 885–893, (1994).

Khare, V., K., Khare, M., K., Neema, M., L., “Estimation of distribution parameters associated with facilities design problems involving forward and backtracking of materials”, *Computers and Industrial Engineering*, 14:1, 63–75, (1988a).

Khare, V., K., Khare, M., K., Neema, M., L., “Combined computer-aided approach for the facilities design problem and estimation of the distribution parameter in the case of multigoal optimization”, *Computers and Industrial Engineering*, 14:4, 465–476, (1988b).

Kleeman, M., P., Day, R., O., Lamont, G., B., “Analysis of a parallel MOEA solving the multi-objective quadratic assignment problem”, *Lecture Notes in Computer Science*, 3103, 402–403, (2004).

Klerk, E., D., Sotirov, R., “Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem”, Optimization Online, http://www.optimization-online.org/DB_HTML/2007/06/1684.html (**Ziyaret tarihi: 22 Mart 2009**).

Knowles, J., D., Corne, D., W., “Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem”, Abraham, A., Ruiz-del-Solar, J., Koppen, M. (Eds.), Soft Computing Systems: Design, Management and Applications, *IOS Press*, Amsterdam, 271–279, (2002).

Knowles, J., D., Corne, D., W., “Instance generators and test suites for the multiobjective quadratic assignment problem”, *Lecture Notes in Computer Science*, 2632, 295–310, (2003).

Knowles, J., D., Corne, D., W., The Multiobjective Quadratic Assignment Problem (mQAP), 2009, <http://dbkgroup.org/knowles/mQAP/>, (**Ziyaret tarihi: 15 Mayıs 2009**).

Kochhar, J., S., Foster, B., T., Heragu, S., S., “Hope: A genetic algorithm for the unequal area facility layout problem”, *Computers and Operations Research*, 25:7–8, 583–594, (1998).

Koopmans, T., C., Beckmann, M., J., “Assignment problems and the location of economic activities”, *Econometrica*, 25, 53–76, (1957).

Krackhardt, D., “Predicting with networks: Nonparametric multiple regression analysis of dyadic data”, *Social Networks*, 10:4, 359–381, (1988).

Krarup, J., Pruzan, P., M., “Computer-aided layout design”, *Mathematical Programming Study*, 9, 75–94, (1978).

Kreher, D., L., Stinson, D., R., “Combinatorial algorithms: Generation, enumeration, and search”, DIMACS Discrete Mathematics and its Applications, *CRC Pres*, (1998).

Lacksonen, T., A., Ensore Jr., E., E., “Quadratic assignment algorithms for the dynamic layout”, *International Journal of Production Research*, 31:3, 503–517, (1993).

Land, A., M., “A problem of assignment with interrelated costs”, *Operational Research Quarterly*, 14, 185–198, (1963).

Laursen, P., S., “Simple approaches to parallel branch-and-bound”, *Parallel Computing*, 19, 143–152, (1993).

Lawler, E., L., “The quadratic assignment problem”, *Management Science*, 9, 586–599, (1963).

Li, Y., Pardalos, P., M., “Generating quadratic assignment test problems with known optimal permutations”, *Computational Optimization and Applications*, 1, 163–184, (1992).

Li, Y., Pardalos, P., M., Ramakrishnan, K., G., Resende, M., G., C., “Lower bounds for the quadratic assignment problem”, *Operations Research*, 50, 387–410, (1994a).

Li, Y., Pardalos, P., M., Resende, M., G., C., “A greedy randomized adaptive search procedure for the quadratic assignment problem”, Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete*

Mathematics and Theoretical Computer Science, 16, AMS, Rhode Island, 237–261, (1994b).

Li, W., J., Smith, M., “An algorithm for quadratic assignment problems”, *European Journal of Operational Research*, 81, 205–216, (1995).

Liang, Y., “Combinatorial optimization by Hopfield networks using adjusting neurons”, *Information Sciences*, 94:1–4, 261–276, (1996).

Lim, M., H., Yuan, Y., Omatu, S., “Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem”, *Computational Optimization and Applications*, 15:3, 248–268, (2000).

Lim, M., H., Yuan, Y., Omatu, S., “Extensive testing of a hybrid genetic algorithm for solving quadratic assignment problems”, *Computational Optimization and Applications*, 23:1, 47–64, (2002).

Lopez-Ibanez, M., Paquete, L., Stutzle, T., “On the design of ACO for the biobjective quadratic assignment problem”, *Lecture Notes in Computer Science*, 3172, 214–225, (2004).

Los, M., “Simultaneous optimization of land use and transportation: A synthesis of the quadratic assignment problem and the optimal network problem”, *Regional Science and Urban Economics*, 8:1, 21–42, (1978).

Lova’sz, L., Schrijver, A., “Cones of matrices and set-functions, and 0–1 optimization”, *SIAM Journal on Optimization*, 1, 166–190, (1991).

Love, R., F., Wong, J., Y., “Solving quadratic assignment problems with rectangular distances and integer programming”, *Naval Research Logistics Quarterly*, 23, 623–627, (1976a).

Love, R., F., Wong, J., Y., “On solving a one-dimensional space allocation problem with integer programming”, *INFOR*, 14, 139–143, (1976b).

Magirou, V., F., Milis, J., Z., “An algorithm for the multiprocessor assignment problem”, *Operations Research Letters*, 8, 351–356, (1989).

Magos, D., Miliotis, P., “An algorithm for the planar three-index assignment problem”, *European Journal of Operational Research*, 77, 141–153, (1994).

Maniezzo, V., Colorni, A., “AlgoDesk: An experimental comparison of eight evolutionary heuristics applied to the quadratic assignment problem”, *European Journal of Operational Research*, 81:1, 188–204, (1995).

Maniezzo, V., Colorni, A., “The ant system applied to the quadratic assignment problem”, *Knowledge and Data Engineering*, 11:5, 769–778, (1999).

Mans, B., Mautor, T., Roucairol, C., “A parallel depth first search branch and bound algorithm for the quadratic assignment problem”, *European Journal of Operational Research*, 81, 617–628, (1995).

Mariano, C., E., Morales, E., “A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks”, *Technical Report HC-9904*, Instituto Mexicano de Tecnología del Agua, Mexico, June, (1999).

Marins, M., T., A, Abreu, N., M., M., Jurkiewicz, S., “Automorphism of groups and quadratic assignment problem”, *Annals of XII Congreso Latino-Iberoamericano de Investigacion de Operaciones y Sistemas (CLAIO 2004)*, La Habana, Cuba, (2004).

Martin, W., “Fast equi-partitioning of rectangular domains using stripe decomposition”, *Discrete Applied Mathematics*, 82:1-3, 193–207, (1998).

Mason, A., Rönnqvist, M., “Solution methods for the balancing of jet turbines”, *Computers and Operations Research*, 24:2, 153–167, (1997).

Mautor, T., Roucairol, C., “A new exact algorithm for the solution of quadratic assignment problems”, *Discrete Applied Mathematics*, 55, 281–293, (1994a).

Mautor, T., Roucairol, C., “Difficulties of exact methods for solving the QAP”, In: Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 263–274, (1994b).

Mavridou, T., Pardalos, P., M., “Simulated annealing and genetic algorithms for the facility layout problem: A survey”, *Computational Optimization and Applications*, 7, 111–126, (1997).

Mavridou, T., Pardalos, P., M., Pitsoulis, L., S., Resende, M., G., C., “A GRASP for the biquadratic assignment problem”, *European Journal of Operational Research*, 105:3, 613–621, (1998).

McMullen, P., R., “An Ant Colony Optimization Approach to Addressing a JIT Sequencing Problem with Multiple Objectives”, *Artificial Intelligence in Engineering*, 15:3, 309-317, (2001).

Medova, E., “Using QAP bounds for the circulant TSP to design reconfigurable Networks”, In: Pardalos, P.M., Wolkowicz, H. (Eds.), *Quadratic Assignment and Related Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, 275–292, (1994).

Merz, P., Freisleben, B., “A genetic local search approach to the quadratic assignment problem”, Back, T. (Ed.), *Proceedings of the 7th International Conference on Genetic Algorithms*, 465–472, (1997).

Merz, P., Freisleben, B., “A comparison of mimetic algorithms, tabu search, and ant colonies for the quadratic assignment problem”, *Proceedings of the 1999*

International Congress of Evolutionary Computation (CEC'99), IEEE Press, 2063–2070, (1999).

Merz, P., Freisleben, B., “Fitness landscape analysis and mimetic algorithms for the quadratic assignment problem”, *IEEE Transactions on Evolutionary Computation*, 4:4, 337–352, (2000).

Michelon, P., Maculan, N., “Lagrangean decomposition for integer nonlinear programming with linear constraints”, *Mathematical Programming*, 52:2, 303–313, (1991).

Middendorf, M., Reischle, F., Schmeck, H., “Multi colony ant algorithms”, *Journal of Heuristics*, 8:3, 305–320, (2002).

Milis, I. Z., Magirou, V., F., “A Lagrangean-relaxation algorithm for sparse quadratic assignment problems”, *Operations Research Letters*, 17:2, 69–76, (1995).

Mills, P., Tsang, E., Ford, J., “Applying an extended guided local search to the quadratic assignment problem”, *Annals of Operations Research*, 118:1–4, 121–135, (2003).

Miranda, G., Luna, H., P., L., Mateus, G., R., Ferreira, R., P., M., “A performance guarantee heuristic for electronic components placement problems including thermal effects”, *Computers and Operations Research*, 32, 2937–2957, (2005).

Mirchandani, P., B., Obata, T., “Algorithms for a class of quadratic assignment problems”, *ORSA/TIMS National Meeting*, New Orleans, (1979).

Misevicius, A., “A new algorithm for the quadratic assignment problem”, *Information Technology and Control*, 5, 39–44, (1997).

Misevicius, A., “An intensive search algorithm for the quadratic assignment problem”, *Informatica*, 11, 145–162, (2000a).

Misevicius, A., “A new improved simulated annealing algorithm for the quadratic assignment problem”, *Information Technology and Control*, 17, 29–38, (2000b).

Misevicius, A., “Combining simulated annealing and tabu search for the quadratic assignment problem”, *Information Technology and Control*, 20, 37–50, (2001).

Misevicius, A., “A modification of tabu search and its applications to the quadratic assignment problem”, *Information Technology and Control*, 27, 12–20, (2003a).

Misevicius, A., “Genetic algorithm hybridized with ruin and recreate procedure: Application to the quadratic assignment problem”, *Knowledge-Based Systems*, 16:5–6, 261–268, (2003b).

Misevicius, A., “A modified simulated annealing algorithm for the quadratic assignment problem”, *Informatica*, 14:4, 497–514, (2003c).

- Misevicius, A., “Ruin and recreate principle-based approach for the quadratic assignment problem”, *Lecture Notes in Computer Science*, 2723, 598–609, (2003d).
- Misevicius, A., “An improved hybrid optimization algorithm for the quadratic assignment problem”, *Mathematical Modelling and Analysis*, 9:2, 149–168, (2004a).
- Misevicius, A., “An improved hybrid genetic algorithm: New results for the quadratic assignment problem”, *Knowledge-Based Systems*, 17:2–4, 65–73, (2004b).
- Misevicius, A., “A tabu search algorithm for the quadratic assignment problem”, *Computational Optimization and Applications*, 30:1, 95–111, (2005).
- Misevicius, A., “An implementation of the iterated tabu search algorithm for the quadratic assignment problem”, Working Paper, *Kaunas University of Technology*, Kaunas, Lithuania, (2008).
- Misevicius, A., Riskus, A., “Multistart threshold accepting: Experiments with the quadratic assignment problem”, *Information Technology and Control*, 12, 31–39, (1999).
- Misevicius, A., Rubliauskas, D., Smolinskas, J., “Reconstruct and improve principle-based algorithm for the quadratic assignment problem”, *Information Technology and Control*, 23, 7–17, (2002).
- Mladenovic, N., Hansen, P., “Variable neighborhood search”, *Computers and Operations Research*, 24, 1097–1100, (1997).
- Moe, R., “GRIBB—Branch-and-bound methods on the Internet. In: Parallel Processing and Applied Mathematics”, *Lecture Notes in Computer Science*, 3019, 1020–1027, (2003).
- Nishiyama, T., Tsuchiya, K., Tsujita, K., “A Markov chain Monte Carlo algorithm for the quadratic assignment problem based on replicator equations”, *Lecture Notes in Computer Science*, 2130, 148–155, (2001).
- Nissen, V., “Solving the quadratic assignment problem with clues from nature”, *IEEE Transactions on Neural Networks*, 5:1, 66–72, (1994).
- Nissen, V., “Quadratic assignment”, Back, T., Fogel, D., B., Michalewicz, Z., Baeck, T. (Eds.), *Handbook of Evolutionary Computation*, G9:10, *Institute of Physics Publishing and Oxford University Press*, 1–8, (1997).
- Nissen, V., Paul, H., “A modification of threshold accepting and its application to the quadratic assignment problem”, *OR Spektrum*, 17:2–3, 205–210, (1995).

- Nugent, C., E., Vollmann, T., E., Ruml, J., “An experimental comparison of techniques for the assignment of facilities to locations”, *Operations Research*, 16, 150–173, (1968).
- Obuchi, Y., Masui, H., Ohki, M., “Weighted parallel problem solving by optimization networks”, *Neural Networks*, 9:2, 357–366, (1996).
- Oliveira, C., A., S., Pardalos, M., P., Resende, M., G., G., “GRASP with path relinking for the quadratic assignment problem”, Experimental and Efficient Algorithms, *Third International Workshop (WEA 2004)*, BrazilLNCS 3059, Springer, 356–368, (2004).
- Osman, I., H., Laporte, G., “Metaheuristics: A bibliography”, *Annals of Operations Research*, 63, 513–623, (1996).
- Ostrowski, T., Ruoppila, V., T., “Genetic annealing search for index assignment in vector quantization”, *Pattern Recognition Letters*, 18:4, 311–318, (1997).
- Padberg, W., Rijal, P., “Location, Scheduling, Design and Integer Programming”, *Kluwer Academic Publishers*, Boston, (1996).
- Padberg, M., W., Rinaldi, G., “A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems”, *SIAM Review*, 33, 60–100, (1991).
- Palubeckis, G., “Generating hard test instances with known optimal solution for the rectilinear quadratic assignment problem”, *Journal of Global Optimization*, 15:2, 127–156, (1999).
- Palubeckis, G., “An algorithm for construction of test cases for the quadratic assignment problem”, *Informatica*, 11:3, 281–296, (2000).
- Paquete, L., Stützle, T., “A study of stochastic local search algorithms for biobjective QAP with correlated flow matrices”, *European Journal of Operational Research*, 169:3, 943–959, (2004).
- Pardalos, P., Crouse, J., “A parallel algorithm for the quadratic assignment problem”, *Proceedings of the Supercomputing Conference*, ACM Press, 351–360, (1989).
- Pardalos, P., M., Murthy, K., A., Harrison, T., P., “A computational comparison of local search heuristics for solving quadratic assignment problems”, *Informatica*, 4:1–2, 172–187, (1993).
- Pardalos, P., M., Ramakrishnan, K., G., Resende, M., G., C., Li, Y., “Implementation of a variance reduction-based lower bound in a branch-and-bound algorithm for the quadratic assignment problem”, *SIAM Journal on Optimization*, 7:1, 280–294, (1997).

Peng, T., Huanchen, W., Dongme, Z., “Simulated annealing for the quadratic assignment problem: A further study”, *Computers and Industrial Engineering*, 31:3–4, 925–928, (1996).

Phillips, A., T., Rosen, J., B., “A quadratic assignment formulation of the molecular-conformation problem”, *Journal of Global Optimization*, 4:2, 229–241, (1994).

Pierce, J., F., Crowston, W., B., “Tree-search algorithms for quadratic assignment problems”, *Naval Research Logistics Quarterly*, 18, 136, (1971).

Pierskalla, W., P., “The tri-substitution method for the three-multidimensional assignment problem”, *Canadian Operational Research Society Journal*, 5:2, 71–81, (1967a).

Pierskalla, W., P., “The Multi-Dimensional Assignment Problem”, *Technical Memorandum No. 93*, Operations Research Department, CASE Institute of Technology, (1967b).

Pierskalla, W., P., “The multidimensional assignment problem”, *Operations Research*, 16, 422–431, (1968).

Pitsoulis, L., S., Pardalos, P., M., Hearn, D., W., “Approximate solutions to the turbine balancing problem”, *European Journal of Operational Research*, 130:1, 147–155, (2001).

Pollatschek, M., A., Gershoni, N., Radday, Y., T., “Optimization of the typewriter keyboard by simulation.”, *Angewandte Informatik*, 17, 438–439, (1976).

Poore, A., “Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking”, *Computational Optimization and Applications*, 3, 27–57, (1994a).

Poore, A., “Partitioning multiple data sets: Multidimensional assignment and Lagrangean relaxation”, Pardalos, P., M., Wolkowicz, H. (Eds.), Quadratic Assignment and Related Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, AMS, Rhode Island, (1994b).

Poore, A., “Multidimensional assignment and multitarget tracking”, In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 19, 169–196, (1995).

Poore, A., Robertson, A., “A new Lagrangean relaxation based algorithm for a class of multidimensional assignment problems”, *Computational Optimization and Applications*, 8, 129–150, (1997).

QAPLIB-A Quadratic Assignment Problem Library, 2009, QAPLIB Home Page, <http://www.seas.upenn.edu/qaplib/> (**Ziyaret tarihi: 22 Mart 2009**).

- Qi, L., Balas, E., Gwan, G., “A new facet class and a polyhedral method for the three-index assignment problem”, Du, D., Z., Sun, J. (Eds.), *Advances in Optimization and Approximation*, **Kluwer Academic**, 256–274, (1994).
- Rabak, C., S., Sichman, J., S., “Using A-Teams to optimize automatic insertion of electronic components”, *Advanced Engineering Informatics*, 17:2, 95–106, (2003).
- Ramachandran, B., Pekny, J., F., “Lower bounds for nonlinear assignment problems using many body interactions”, *European Journal of Operational Research*, 105:1, 202–215, (1998).
- Ramakrishnan, K., G., Resende, M., G., C., Ramachandran, B., Pekny, J., F., “Tight QAP bounds via linear programming”, Pardalos, P., M., Migdalas, A., Burkard, R., E. (Eds.), *Combinatorial and Global Optimization*, **World Scientific Publishing**, Singapore, 297–303, (2002).
- Randall, M., “Near parameter free ant colony optimization”, *Lecture Notes in Computer Science*, 3172, 374–381, (2004).
- Rendl, F., Wolkowicz, H., “Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem”, *Mathematical Programming*, 53, 63–78, (1992).
- Resende, M., G., C., Pardalos, P., M., Li, Y., “Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using GRASP”, *ACM Transactions on Mathematical Software*, 22:1, 104–118, (1996).
- Resende, M., G., C., Ramakrishnan, K., G., Drezner, Z., “Computing lower bounds for the quadratic assignment with an interior point algorithm for linear programming”, *Operations Research*, 43, 781–791, (1995).
- Rogger, A., Fiechter, C., N., Werra, D., “Basic ideas of tabu search with an application to traveling salesman and quadratic assignment”, *Ricerca Operativa*, 62, 5–28, (1992).
- Rossin, D., F., Springer, M., C., Klein, B., D., “New complexity measures for the facility layout problem: An empirical study using traditional and neural network analysis”, *Computers and Industrial Engineering*, 36:3, 585–602, (1999).
- Roucairol, C., “A reduction method for quadratic assignment problem”, *Methods of Operations Research*, 32, 185–187, (1979).
- Roucairol, C., “A parallel branch and bound algorithm for the quadratic assignment problem”, *Discrete Applied Mathematics*, 18, 211–225, (1987).
- Roupin, F., “From linear to semidefinite programming: An algorithm to obtain semidefinite relaxations for bivalent quadratic problems”, *Journal of Combinatorial Optimization*, 8:4, 469–493, (2004).

- Sahni, S., Gonzales, T., “P-complete approximation problems”, *Journal of the Association for Computing Machinery*, 23, 555–565, (1976).
- Samra, H., Ding, Z., Hahn, P., M., “Symbol mapping diversity design for multiple packet transmissions”, *IEEE Transactions on Communications*, 53:5, 810–817, (2005).
- Sarker, B., R., Wilhelm, W., E., Hogg, G., L., “One-dimensional machine location problems in a multi-product flowline with equidistant locations”, *European Journal of Operational Research*, 105:3, 401–426, (1998).
- Sarker, B., R., Wilhelm, W., E., Hogg, G., L., Han, M., H., “Backtracking of jobs in one-dimensional machine location problems”, *European Journal of Operational Research*, 85:3, 593–609, (1995).
- Scriabin, M., Vergin, R., C., “Comparison of computer algorithms and visual based methods for plant layout”, *Management Science*, 22, 172–187, (1975).
- Sergeev, S., I., “Improved lower bounds for the quadratic assignment problem”, *Automation and Remote Control*, 65:11, 1733–1746, (2004).
- Sherali, H., D., Adams, W., P., “Reformulation–linearization techniques for discrete optimization problems”, Pardalos, P., M., Du, D., Z. (Eds.), Handbook of Combinatorial Optimization, *Kluwer Academic Publishers*, 479–532, (1999a).
- Sherali, H., D., Adams, W., P., “A Reformulation–linearization Technique for Solving Discrete and Continuous Nonconvex Optimization Problems”, *Kluwer Academic Publishers*, Dordrecht, The Netherlands, (1999b).
- Shin, I., Niitsuma, H., “Lambda-opt neural approaches to quadratic assignment problems”, *Neural Computation*, 12, 2209–2225, (2000).
- Simeone, B., “An asymptotically exact polynomial time algorithm for equipartition problems”, *Discrete Applied Mathematics*, 14, 283–293, (1986a).
- Simeone, B., “Topological network synthesis”, Simeone, B. (Ed.), Combinatorial Optimization, Lecture Notes in Mathematics, 1403, *Springer-Verlag*, 282–303, (1986b).
- Siu, F., Chang, R., K., C., “Effectiveness of optimal node assignments in wavelength division multiplexing networks with fixed regular virtual topologies”, *Computer Networks*, 38:1, 61–74, (2002).
- Skorin-Kapov, J., “Tabu search applied to the quadratic assignment problem”, *ORSA Journal on Computing*, 2:1, 33–45, (1990).
- Skorin-Kapov, J., “Extensions of a tabu search adaptation to the quadratic assignment problem”, *Journal of Computers and Operations Research*, 21:8, 855–865, (1994).

- Smith, J., M., Li, W., J., “Quadratic assignment problems and M/G/C/C state dependent network flows”, *Journal of Combinatorial Optimization*, 5:4, 421–443, (2001).
- Solimanpur, M., Vrat, P., Shankar, R., “Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing”, *European Journal of Operational Research*, 157:3, 592–606, (2004).
- Spiliopoulos, K., Sofianopoulou, S., “An optimal tree search method for the manufacturing systems cell formation problem”, *European Journal of Operational Research*, 105:3, 537–551, (1998).
- Steinberg, L., “The backboard wiring problem: A placement algorithm”, *SIAM Review*, 3, 37–50, (1961).
- Stützle, T., “MAX-MIN ant system for quadratic assignment problems”, *Research Report AIDA-97-04*, Department of Computer Science, Darmstadt University of Technology, Germany, (1997).
- Stützle, T., “Iterated local search for the quadratic assignment problem”, *European Journal of Operational Research*, 174:3, 1519–1539, (2006).
- Stützle, T., Dorigo, M., “ACO algorithms for the quadratic assignment problem”, Corne, D., Dorigo, M., Glover, F. (Eds.), *New Ideas in Optimization*, **McGraw-Hill**, 33–55, (1999).
- Stützle, T., Fernandes, S., “New benchmark instances for the QAP and the experimental analysis of algorithms”, *Lecture Notes in Computer Science*, 3004, 199–209, (2004).
- Stützle, T., Hoos, H., H., “MAX-MIN Ant System”, *Future Generation Computer Systems*, 16:8, 889–914, (2000).
- Sylla, C., Babu, A., J., G., “Methodology for an orderly quadratic assignment problem”, *Computers and Industrial Engineering*, 13:1–4, 281–284, (1987).
- T'kindt, V., Monmarché, N., Tercinet, F., Laügt, D., “An Ant Colony Optimization Algorithm to Solve a 2-machine Bicriteria Flowshop Scheduling Problem”, *European Journal of Operational Research*, 142:2, 250–257, (2002).
- Taillard, E., “Robust taboo search for the quadratic assignment problem”, *Parallel Computing*, 17, 443–455, (1991).
- Taillard, E., “Comparison of iterative searches for the quadratic assignment problem”, *Location Science*, 3:2, 87–105, (1995).
- Taillard, E., D., Gambardella, L., M., “Adaptive memories for the quadratic assignment problem”, *Research report, IDSIA*, Lugano, Switzerland, (1997).

- Taillard, E., Gambardella, L., “Adaptive memories for the quadratic assignment problem”, *Technical Report I-87-97*, IDSIA, Lugano, (1999).
- Taillard, E., Gambardella, L., M., Gendreau, M., Potvin, J., Y., “Adaptive memory programming: A unified view of metaheuristics”, *European Journal of Operational Research*, 135:1, 1–16, (2001).
- Takagi, M., “Optimization of placement by candidate sieving”, *IEEE Transactions on Electronics Packaging Manufacturing*, 24:3, 178–187, (2001).
- Talbi, E., G., Hafidi, Z., Kebbal, D., Geib, J., M., “A fault-tolerant parallel heuristic for assignment problems”, *Future Generation Computer Systems*, 14:5–6, 425–438, (1998a).
- Talbi, E., G., Hafidi, Z., Geib, J., M., “A parallel adaptive tabu search approach”, *Parallel Computing*, 24:14, 2003–2019, (1998b).
- Talbi, E., G., Roux, O., Fonlupt, C., Robillard, D., “Parallel ant colonies for the quadratic assignment problem”, *Future Generation Computer Systems*, 17:4, 441–449, (2001).
- Talbot, N., L., C., Cawley, G., C., “A quadratic index assignment algorithm for vector quantization over noisy transmission channels”, *Proceedings of the Institute of Acoustics Autumn Conference*, 18, 195–199, (1996).
- Tansel, B., C., Bilen, C., “Move based heuristics for the unidirectional loop network layout problem”, *European Journal of Operational Research*, 108:1, 36–48, (1998).
- Tate, D., E., Smith, A., E., “A genetic approach to the quadratic assignment problem”, *Computers and Operations Research*, 22, 73–83, (1995).
- Tavakkoli-Moghaddain, R., Shayan, E., “Facilities layout design by genetic algorithms”, *Computers and Industrial Engineering*, 35:3–4, 527–530, (1998).
- Tian, P., Wang, H., C., Zhang, D., M., “Simulated annealing for the quadratic assignment problem: A further study”, *Computers and Industrial Engineering*, 31:3–4, 925–928, (1996).
- Tian, P., Ma, J., Zhang, D., M., “Application of the simulated annealing algorithm to the combinatorial optimization problem with permutation property: An investigation of generation mechanism”, *European Journal of Operational Research*, 118:1, 81–94, (1999).
- Torki, A., Yajima, Y., Enkawa, T., “A low-rank bilinear programming approach for sub-optimal solution of the quadratic assignment problem”, *European Journal of Operational Research*, 94:2, 384–391, (1996).

- Tsuchiya, K., Bharitkar, S., Takefuji, Y., “A neural network approach to facility layout problems”, *European Journal of Operational Research*, 89:3, 556–563, (1996).
- Tsuchiya, K., Nishiyama, T., Tsujita, K., “A deterministic annealing algorithm for a combinatorial optimization problem using replicator equations”, *Physica D: Nonlinear Phenomena*, 149:3, 161–173, (2001).
- Urban, T., L., “Solution procedures for the dynamic facility layout problem”, *Annals of Operations Research*, 76, 323–342, (1998).
- Urban, T., L., Chiang, W., C., Russell, R., A., “The integrated machine allocation and layout problem”, *International Journal of Production Research*, 38:13, 2911–2930, (2000).
- Uwate, Y., Nishio, Y., Ushida, A., “Markov chain modeling of intermittency chaos and its application to Hopfield NN”, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E87A:4, 774–779, (2004).
- Vlach, M., “A branch-and-bound method for the three-index assignment problem”, *Ekonomicko-Matematicky Obzor*, 3, 181–191, (1967).
- Wang, R., L., Okazaki, K., “Solving facility layout problem using an improved genetic algorithm”, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E88A:2, 606–610, (2005).
- Wang, S., J., Sarker, B., R., “Locating cells with bottleneck machines in cellular manufacturing systems”, *International Journal of Production Research*, 40:2, 403–424, (2002).
- Wess, B., Zeitlhofer, T., “On the phase coupling problem between data memory layout generation and address pointer assignment”, *Lecture Notes in Computer Science*, 3199, 152–166, (2004).
- West, D., H., “Algorithm 608: Approximate solution of the quadratic assignment problem”, *ACM Transactions on Mathematical Software*, 9, 461–466, (1983).
- White, D., J., “Strengthening Gilmore’s bound for the quadratic assignment problem”, *European Journal of Operational Research*, 77:1, 126–140, (1994a).
- White, D., J., “The use of specially structured models for obtaining bounds in the quadratic assignment problem”, *Journal of the Operational Research Society*, 45:4, 451–462, (1994b).
- White, D., J., “Some concave–convex representations of the quadratic assignment problem”, *European Journal of Operational Research*, 80:2, 418–424, (1995).

- White, D., J., “A Lagrangean relaxation approach for a turbine design quadratic assignment problem”, *Journal of the Operational Research Society*, 47:6, 766–775, (1996).
- Whitney, H., “Congruent graphs and the connectivity of graphs”, *American Journal of Mathematics*, 54, 150–168, (1932).
- Wilhelm, M., R., Ward, T., L., “Solving quadratic assignment problems by simulated annealing”, *IEEE Transactions*, 19, 107–119, (1987).
- Wolkowicz, H., “Semidefinite programming approaches to the quadratic assignment problem”, Pardalos, P., M., Pitsoulis, L. (Eds.), *Nonlinear Assignment Problems: Algorithms and Applications*, Combinatorial Optimization Series, 7, *Kluwer Academic Publishers*, 143–174, (2000).
- Wolkowicz, H., Saigal, R., Vandenberghe, L., *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, International Series in Operations Research, 27, *Kluwer*, (2000).
- Yamada, S., “A new formulation of the quadratic assignment problem on r-dimensional grid”, *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications*, 39:10, 791–797, (1992).
- Ying, K., C., Liao, C., J., “An ant colony system for permutation flow-shop sequencing”, *Computers and Operations Research*, 31:5, 791–801, (2004).
- Yip, P., P., C., Pao, Y., H., “A guided evolutionary simulated annealing approach to the quadratic assignment problem”, *IEEE Transactions on Systems Man and Cybernetics*, 24:9, 1383–1387, (1994).
- Youssef, H., Sait, S., M., Ali, H., “Fuzzy simulated evolution algorithm for VLSI cell placement”, *Computers and Industrial Engineering*, 44:2, 227–247, (2003).
- Yu, J., Sarker, B., R., “Directional decomposition heuristic for a linear machine-cell location problem”, *European Journal of Operational Research*, 149:1, 142–184, (2003).
- Zhao, Q., Karisch, S., E., Rendl, F., Wolkowicz, H., “Semidefinite programming relaxations for the quadratic assignment problem”, *Journal of Combinatorial Optimization*, 2, 71–109, (1998).
- Zitzler, E., Deb, K., Thiele, L., “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results”, *Evolutionary Computation*, 8:2, 173–195, (2000).

ÖZGEÇMİŞ

1978 yılında Kocaeli’nde doğdu. İlk, orta ve lise öğrenimini Kocaeli’nde tamamladı. 1995 yılında lisans öğrenimine başladığı Kocaeli Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü’nden 1999 yılında Endüstri Mühendisi olarak mezun oldu. 1999-2002 yılları arasında, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda Yüksek Lisans öğrenimini tamamladı. 1999 yılından beri Kocaeli Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü’nde Araştırma Görevlisi olarak görev yapmaktadır. Bu görevinin yanında, 2004-2005 yıllarında Avrupa Birliği İş Geliştirme Merkezi-Kocaeli’nde danışman olarak görev yapmış olan End. Yük. Müh. Celal ÖZKALE, sistem simülasyonu, norm kadro, reorganizasyon, kalite yönetimi vb. konularda halen üniversite-sanayi işbirliğine katkıda bulunmaktadır. End. Yük. Müh. Celal ÖZKALE, 2001 yılında eşi Neslihan ÖZKALE ile evlenmiş ve bu mutlu birlikteliğinden 2009 yılında oğlu Poyraz ÖZKALE dünyaya gelmiştir.