

154265

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KARINCA ALGORİTMASI VE TESİS YERLEŞİMİ
PROBLEMLERİNE UYGULANMASI**

End.Müh. M.Duran TOKSARI

FBE Endüstri Mühendisliği Anabilim Dalı Sistem Mühendisliği Programında
Hazırlanan

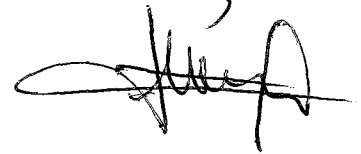
YÜKSEK LİSANS TEZİ

Tez Danışmanı :Yrd.Doç.Dr. Nihan ÇETİN DEMİREL

Prof.Dr. Cengiz Kahraman

Yrd.Doç.Dr. Nihan Çetin Demirel

Prof.Dr. Mesut ÖZGÜLLER



İSTANBUL, 2004

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vii
ŞEKİL LİSTESİ	viii
ÇİZELGE LİSTESİ	x
ÖNSÖZ	xi
ÖZET	xii
ABSTRACT	xiii
1 GİRİŞ	1
2 LİTERATÜRÜN İNCELENEMESİ	3
3 TESİS YERLEŞİMİ	6
3.1 Tesis Yerleşimi Problemleri	6
3.2 Problemin Formülizasyonu	7
3.3 Kareli Atama Problemi (KAP)	8
3.3.1 Kombinatoriyal Optimizasyon Problemi Formülizasyonu	9
3.3.2 Koopmans-Beckmann Formülizasyonu	10
3.4 KAP için Çözüm Yaklaşımları	11
4 KARINCA ALGORİTMASI	14
4.1 Karınca Sistemi	14
4.2 Karınca Algoritması ve GSP Uygulaması	17
4.2.1 Karınca-Yoğunluğu Modeli	24
4.2.2 Karınca-Niceliği Modeli	24
4.2.3 Deneysel Çalışmalar	25
4.2.4 Algoritmanın Performansını Etkileyen Faktörler	26
4.2.4.1 Sinerjik Etkiler	26
4.2.4.2 Başlangıç Noktası	27
4.2.4.3 Problem Boyutu	27
4.3 Karınca Algoritması Uygulamaları	27

5	TESİS YERLEŞİMİ PROBLEMLERİ İÇİN KARINCA ALGORİTMASI	30
5.1	Kareli Atama Problemi için Karınca Algoritması Uygulamaları	30
5.1.1	Problemin Tanımlanması.....	30
5.1.2	Kareli Atama Problemi için Karınca Algoritması	31
5.1.2.1	Her Karınca için Tabu Listelerinin Oluşturulması	32
5.1.2.2	Mesafe Potansiyelleri ve Akış Potansiyelleri Vektörleri.....	32
5.1.2.3	Atama Olasılıklarının Kullanımı	34
5.1.3	Kareli Atama Problemi için Melez Karınca Algoritması	37
5.1.3.1	Çözümlerin Başlangıcı	37
5.1.3.2	Fenomen İzi Matrisi Başlangıcı.....	38
5.1.3.3	Çözümlerin Geliştirilmesi.....	38
5.1.3.3.1	Fenomen İz Tabanlı Gelişme.....	38
5.1.3.3.2	Yerel Arama Prosedürü ile Gelişme	39
5.1.3.4	Yoğunlaşma	40
5.1.3.5	Fenomen İz Matrisinin Güncellenmesi.....	41
5.1.3.6	Çeşitlendirme Mekanizması	41
5.1.4	Kareli Atama Problemi için Paralel Uygulama	43
5.1.4.1	Algoritmanın Paralel Yapısı	44
5.1.4.2	Yerel Arama Prosedürü	45
5.1.4.3	Fenomen Matrisinin Güncellenmesi.....	45
5.1.4.4	Çeşitlendirme Mekanizması	45
5.2	Esnek İmalat Sisteminde Tek Sıralı Tesis Yerleşimi Problemi için Karınca Algoritması	46
5.2.1	Matematiksel Modelin Geliştirilmesi	46
5.2.2	Problemin Çözümü için Karınca Algoritması	47
5.2.2.1	Çözümün Geliştirilmesi.....	48
5.2.2.2	Sezgisel Bilgi	48
5.2.2.3	Fenomen İzini Güncelleme.....	49
5.2.2.4	Seçme Olasılığı.....	49
5.2.2.5	Yerel Arama Algoritması	50
5.2.2.6	Durdurma Kriteri	50
6	UYGULAMA: TESİS YERLEŞİMİ PROBLEMİ İÇİN TAVLAMALIKARINCA ALGORİTMASI.....	52
6.1	Problemin Tanımlanması.....	52
6.2	Algoritmanın Genel Yapısı.....	53
6.3	Karıncaların Başlangıç Çözümleri.....	54
6.4	Fenomen İzi Matrisi Başlangıcı.....	55
6.5	Çözümlerin Geliştirilmesi.....	55
6.5.1	Fenomen İzine Bağlı Gelişme	55
6.5.2	Yerel Arama Prosedürü ile Gelişme	56
6.5.2.1	Tavlama Benzetimi ve Genel Algoritmik Yapısı	56
6.5.2.2	TavlamalıKarınca Algoritmasında Tavlama Benzetiminin Kullanımı.....	58
6.5.2.2.1	Çiftli Yer Değiştirme Mekanizması	60
6.5.2.2.2	En iyi Çiftli Yer Değiştirme Mekanizması.....	60
6.5.3	Yoğunlaşma	60
6.5.4	Fenomen İz Matrisinin Güncellenmesi.....	61
6.5.5	Çeşitlendirme Mekanizması	61
6.5.6	KAP için TavlamalıKarınca Algoritmasının Adımları.....	61

6.5.7	Parametrelerin Belirlenmesi	64
7	DENEY TASARIMI	65
7.1	Dikkate Alınan Etkinlik Ölçütü	65
7.2	Test Problemleri	65
7.3	Deney Tasarımı	66
7.3.1	Uygun Parametre Setinin Belirlenmesi	66
7.3.2	Analiz Sonuçları	67
7.3.2.1	NUG20 Tasarımı	68
7.3.2.2	NUG30 Tasarımı	73
7.3.2.3	TAI30a Tasarımı	77
7.3.2.4	TAI60a Tasarımı	82
7.3.2.5	SKO64 Tasarımı	88
7.3.2.6	BUR26a Tasarımı	91
7.3.2.7	BUR26h Tasarımı	97
7.3.2.8	TAI30b Tasarımı	102
7.3.2.9	TAI40b Tasarımı	108
7.3.2.10	TAI80b Tasarımı	114
7.4	Çözümlerin Değerlendirilmesi	119
7.5	Diğer Meta-Sezgisellerle Karşılaştırılması	120
8	SONUÇLAR VE ÖNERİLER	122
	KAYNAKLAR	125
	EKLER	130
Ek 1	Düzenli Test Problemleri için Tavlamalı Karınca Algoritması Sonuçları	131
Ek 2	Düzensiz Test Problemleri için Tavlamalı Karınca Algoritması Sonuçları	137
	ÖZGEÇMİŞ	143

SİMGE LİSTESİ

c_{ij}	i departmanından j departmanına bir birim uzaklığa bir birim malzemenin taşıma maliyeti
C	Atama maliyeti matrisi
d_i	i düğümünün en küçük mesafe potansiyeli
d_{ij}	Departmanlar arası uzaklık
D	Mesafe matrisi
D	Mesafe potansiyelleri vektörü
f_i	i düğümünün en büyük akış potansiyeli
f_{ij}	Departmansal iç akışlar
$f_{\pi_{ij}}$	π çözümünde i ve j tesisleri arası akış miktarı
$f(\pi)$	π çözümü için amaç fonksiyonun değeri
$f(\pi^*)$	O ana kadar elde edilen en iyi çözümün amaç fonksiyonu değeri
F	Akış matrisi
F	Akış potansiyelleri vektörü
F^T	Akış matrisinin transpozu
I	Yerel arama prosedürünün iterasyon sayısı
$I^{\max}$	Algoritmanın iterasyon sayısı
L_k	k'nci karıncanın tur uzunluğu
m	Toplam karınca sayısı
$m_i(t)$	t zamanında i'de bulunan karınca sayısı
n	Problemin boyutu
p	İzin buharlaşmadan kalan oranı
p_{ij}^k	k'nci karıncanın i kasabasından sonra j kasabasına gitme olasılığı
Q	Karıncalar tarafından yayılan izin miktarı ile ilişkili bir sabit
r	Soğuma oranı
S	Çeşitlendirme kararının verileceği iterasyon
t	Zaman birim
T	Takas sayısı
T_0	İlk sıcaklık değeri
T_k	Mevcut sıcaklık
T_{son}	Son sıcaklık
$Tabu_k$	k'nci karıncanın tabu listesi
V_k	(i,l) taşımalarının saklı tutulduğu hafıza
Y_{ij}	Departmanlar arası yakınlık oranı
$\Pi(n)$	n elemanın permütasyonu
π	Çözüm permütasyonu
π^k	Fenomen iz tabanlı gelişme ile elde edilen çözüm
$\pi^{\infty k}$	Yerel arama prosedüründe geliştirilerek elde edilen çözüm
π^*	O ana kadar elde edilen en iyi çözüm
π^-	O ana kadar elde edilen en kötü çözüm
η	Görüş derecesi
α	İzin önem ilişkisi
β	Görünürlüğün önem ilişkisi
λ	Hesaplama sürecinin dinamiklerini kontrol eden ölçü faktörü
τ_{ij}	(i,j) kenarı/yerleşimi üzerindeki fenomen izi miktarı
$\tau_{\min}$	Minimum fenomen miktarı
τ_{maksimum}	Maksimum fenomen miktarı

$\tau_{ij}(t)$	t zamanda i yerleşimine j tesisinin atanmasındaki fenomen iz yoğunluğu
$\Delta\tau_{ij}$	i yerleşimine j tesisinin atanmasındaki toplam fenomen iz miktarı
$\Delta\tau_{il}^k$	i yerleşimine j tesisinin atanmasında k karıncasının bıraktığı iz miktarı
τ_0	Fenomen izleri başlangıç değeri
$\emptyset$	Boş küme



KISALTMA LİSTESİ

ÇAF	Çözümler Arası Fark
EİS	Esnek İmalat Sistemleri
GA	Genetik Algoritmalar
GMM	Genetik Melez Metot
GSP	Gezgin Satıcı Problemi
KA	Karınca Algoritması
KAP	Kareli Atama Problemi
KKO	Karınca Kolonisi Optimizasyonu
MKKS	Melez Karınca Kolonisi Sistemi
NP	Non-Polinomial
PKA	Paralel Karınca Algoritması
RTA	Reaktif Tabu Arama
TA	Tabu Arama
TB	Tavlama Benzetimi
TYP	Tesis Yerleşimi Problemi



ŞEKİL LİSTESİ

Şekil 4.1	Gerçek karıncaların davranışları.....	15
Şekil 4.2	Yapay karıncaların davranışları.....	16
Şekil 4.3	Karıncalar algoritmasının genel akış şeması	28
Şekil 5.1	Algoritmanın paralel yapısı	44
Şekil 6.1	TavlamalıKarıncalar algoritmasının genel akış şeması	53
Şekil 7.1	NUG20 problemi için normal olasılık eğrisi	69
Şekil 7.2	NUG20 problemi için pareto grafiği	70
Şekil 7.3	NUG20 problemi için etkin parametrelerle normal olasılık eğrisi	71
Şekil 7.4	NUG20 problemi için etkin parametrelerle histogram grafiği	72
Şekil 7.5	NUG20 probleminde çözüm kalitesi için ana etkiler	72
Şekil 7.6	NUG30 problemi için normal olasılık eğrisi	75
Şekil 7.7	NUG30 problemi için pareto grafiği	75
Şekil 7.8	NUG30 probleminde çözüm kalitesi için ana etkiler	76
Şekil 7.9	TAI30a problemi için normal olasılık eğrisi	78
Şekil 7.10	TAI30a problemi için pareto grafiği.....	79
Şekil 7.11	TAI30a problemi için etkin parametrelerle normal olasılık eğrisi	80
Şekil 7.12	TAI30a problemi için etkin parametrelerle histogram grafiği.....	81
Şekil 7.13	TAI30a probleminde çözüm kalitesi için ana etkiler.....	81
Şekil 7.14	TAI60a problemi için normal olasılık eğrisi	84
Şekil 7.15	TAI60a problemi için pareto grafiği.....	84
Şekil 7.16	TAI60a problemi için etkin parametrelerle normal olasılık eğrisi	86
Şekil 7.17	TAI60a problemi için etkin parametrelerle histogram grafiği.....	86
Şekil 7.18	TAI60a probleminde çözüm kalitesi için ana etkiler.....	87
Şekil 7.19	SKO64 problemi için normal olasılık eğrisi	89
Şekil 7.20	SKO64 problemi için pareto grafiği	90
Şekil 7.21	SKO64 probleminde çözüm kalitesi için ana etkiler	90
Şekil 7.22	BUR26a problemi için normal olasılık eğrisi	93
Şekil 7.23	BUR26a problemi için pareto grafiği	93
Şekil 7.24	BUR26a problemi için etkin parametrelerle normal olasılık eğrisi.....	95
Şekil 7.25	BUR26a problemi için etkin parametrelerle histogram grafiği	95
Şekil 7.26	BUR26a probleminde çözüm kalitesi için ana etkiler	96
Şekil 7.27	BUR26h problemi için normal olasılık eğrisi.....	98
Şekil 7.28	BUR26h problemi için pareto grafiği	99
Şekil 7.29	BUR26h problemi için etkin parametrelerle normal olasılık eğrisi	100
Şekil 7.30	BUR26h problemi için etkin parametrelerle histogram grafiği.....	101
Şekil 7.31	BUR26h probleminde çözüm kalitesi için ana etkiler.....	101
Şekil 7.32	TAI30b problemi için normal olasılık eğrisi	104
Şekil 7.33	TAI30b problemi için pareto grafiği	104
Şekil 7.34	TAI30b problemi için etkin parametrelerle normal olasılık eğrisi	106
Şekil 7.35	TAI30b problemi için etkin parametrelerle histogram grafiği	107
Şekil 7.36	TAI30b probleminde çözüm kalitesi için ana etkiler	107
Şekil 7.37	TAI40b problemi için normal olasılık eğrisi	110
Şekil 7.38	TAI40b problemi için pareto grafiği	110
Şekil 7.39	TAI40b problemi için etkin parametrelerle normal olasılık eğrisi	112
Şekil 7.40	TAI40b problemi için etkin parametrelerle histogram grafiği	112
Şekil 7.41	TAI40b probleminde çözüm kalitesi için ana etkiler	113
Şekil 7.42	TAI80b problemi için normal olasılık eğrisi	115
Şekil 7.43	TAI80b problemi için pareto grafiği	115
Şekil 7.44	TAI80b problemi için etkin parametrelerle normal olasılık eğrisi	117

Şekil 7.45 TAI80b problemi için etkin parametrelerle histogram grafiği	117
Şekil 7.46 TAI80b probleminde çözüm kalitesi için ana etkiler	118



ÇİZELGE LİSTESİ

Çizelge 4.1	Üç farklı model için parametre ayarlaması.....	26
Çizelge 4.2	Problem boyutunun fonksiyonu olarak optimumu bulmak için gereken zaman	27
Çizelge 5.1	Parametrelerin ölçüm değerleri	36
Çizelge 5.2	Parametrelerin ölçüm değerleri	43
Çizelge 5.3	Parametrelerin ölçüm değerleri	46
Çizelge 5.4	Parametrelerin ölçüm değerleri	50
Çizelge 7.1	Düzenli sınıfına dahil test problemleri	65
Çizelge 7.2	Düzensiz sınıfına dahil test problemleri	66
Çizelge 7.3	Test problemleri için parametre düzeyleri.....	67
Çizelge 7.4	NUG20 tasarımı.....	68
Çizelge 7.5	NUG20 indirgenmiş tasarımı.....	70
Çizelge 7.6	NUG20 problemi için parametre düzeyleri	73
Çizelge 7.7	NUG30 tasarımı.....	74
Çizelge 7.8	NUG30 problemi için parametre düzeyleri	77
Çizelge 7.9	TAI30a tasarımı	77
Çizelge 7.10	TAI30a indirgenmiş tasarımı	79
Çizelge 7.11	TAI30a problemi için parametre düzeyleri	82
Çizelge 7.12	TAI 60a tasarımı.....	83
Çizelge 7.13	TAI60a indirgenmiş tasarımı	85
Çizelge 7.14	Tai60a problemi için parametre düzeyleri	88
Çizelge 7.15	SKO64 tasarımı	88
Çizelge 7.16	Sko64 problemi için parametre düzeyleri	91
Çizelge 7.17	BUR26a tasarımı	92
Çizelge 7.18	BUR26a indirgenmiş tasarımı.....	94
Çizelge 7.19	Bur26a problemi için parametre düzeyleri.....	97
Çizelge 7.20	BUR26h tasarımı.....	97
Çizelge 7.21	BUR26h indirgenmiş tasarımı.....	99
Çizelge 7.22	Bur26h problemi için parametre düzeyleri.....	102
Çizelge 7.23	TAI30b tasarımı	103
Çizelge 7.24	TAI30b indirgenmiş tasarımı	105
Çizelge 7.25	Tai30b problemi için parametre düzeyleri	108
Çizelge 7.26	TAI40b tasarımı	109
Çizelge 7.27	TAI40b indirgenmiş tasarımı	111
Çizelge 7.28	Tai40b problemi için parametre düzeyleri	114
Çizelge 7.29	TAI80b tasarımı	114
Çizelge 7.30	TAI80b indirgenmiş tasarımı	116
Çizelge 7.31	Tai80b problemi için parametre düzeyleri	119
Çizelge 7.32	Düzenli sınıfına dahil test problemleri	119
Çizelge 7.33	Düzensiz sınıfına dahil test problemleri.....	120
Çizelge 7.34	Diğer meta-sezgisellerle karşılaştırma	120
Çizelge 7.35	Diğer meta-sezgisellerle karşılaştırma	121

ÖNSÖZ

Bu çalışma son yıllarda bir çok kombinatorial optimizasyon problemine çözüm üreten Karınca Algoritması (KA) hakkında detaylı bilgi vermekte ve NP-zor yapıları tesis yerleşimi problemi (TYP) olarak formüle edilen kareli atama problemine (KAP) çözüm üretmek için geliştirilen yerel arama prosedürü Tavlama Benzetimi (TB) tabanlı KA'yı (TavlamalıKarınca) sunmaktadır.

Çalışmada KA'nın gelişimi, uygulama alanları ve TYP'ye uygulanması üzerine bir literatür araştırması yapılmış, tesis yerleşimi problemleri, problemin formülizasyonu, kareli atama problemi ve kareli atama problemine çözüm yaklaşımları incelenmiştir. Genel doğal karınca sistemi incelendikten sonra KA ve KA'nın ilk uygulaması olan gezgin satıcı problemine uygulanması detaylı bir biçimde anlatılarak KA uygulamalarından bahsedilmiştir. Ayrıca günümüze kadar KA'nın tesis yerleşimi problemlerine uygulanması detaylı biçimde anlatılmış ve kareli atama problemine çözüm üretmek için geliştirilen yerel arama prosedürü TB tabanlı olan "TavlamalıKarınca" algoritması detaylı biçimde tanıtılmıştır. C++ kodlama diliyle programlanan algoritma literatürde bilinen KAP örnekleriyle denenmiş ve her bir örnek için MINITAB istatistik programında deney tasarımı yapılarak uygun parametre setleri belirlenmiştir. Elde edilen "TavlamalıKarınca" algoritması çözüm sonuçları, aynı KAP problemlerine çözüm üreten diğer sezgisellerin çözüm değerleri ile karşılaştırılmıştır.

Çalışmalarım boyunca, değerli görüş ve katkılarıyla bana her konuda yardımcı olan Sayın Hocam Yrd.Doç.Dr. Nihan Çetin Demirel'e ve çalışmamdaki desteğinden dolayı meslektaşım Sn.Arş.Gör. Sinem Kulluk'a sonsuz teşekkürlerimi sunarım. Bu günlere gelmemde maddi ve manevi açıdan bana her konuda destek olan değerli büyüklerim Prof.Dr. Ali Toksarı ve Gülen Toksarı'ya teşekkürü bir borç bilirim.

Ayrıca çalışma sürecim boyunca manevi desteğini ve hoşgörüsünü eksik etmeyen sevgili Elif Gürpınar'a teşekkür ederim.

ÖZET

Karınca Algoritması optimizasyon problemlerinin çözümü için gerçek karınca kolonilerinin davranışlarından esinlenilerek geliştirilmiş sezgisel bir algoritmadır. Arama alanının araştırılmasında karıncalar tarafından yayılan fenomen kokusu seviyesinin değişimi kullanılmıştır.

Tesis yerleşimi problemleri (TYP), toplam malzeme taşıma maliyetlerini minimize edecek şekilde, m tesis n yerleşime atamakla ilgilenir. İlk defa 1957 yılında ortaya atılan Kareli Atama Problemi (KAP), eşit alanlı tesisler söz konusu olduğunda TYP olarak formüle edilebilir.

KAP uygulamalarının NP-Zor yapıda olmasından dolayı sınırlı kullanımı, sezgisel algoritmaların gelişmesine sebep olmuştur. Çünkü optimal arama prosedürleri en fazla 15-20 tesis gibi küçük boyutlu problemlerle sınırlıdır. Bu sebepten dolayı, çalışmada Kareli Atama Probleminin çözümü için lokal arama prosesi Tavlama Benzetimi olan Karınca Algoritması sunulmuştur. C++ programlama dili ile kodlanan Tavlama Karınca Algoritması literatürdeki mevcut problemler kullanılarak analiz edilmiş ve mevcut problemler için denenmiş sezgisellerin bazılarıyla karşılaştırılmıştır.

Anahtar Sözcükler: Kareli Atama Problemleri, Karınca Algoritması, Tavlama Benzetimi, Meta-Sezgiseller

ABSTRACT

Ant Algorithm is a heuristics algorithm which is developed from the inspiration of the behaviors of real ant colonies to solve optimization problems. In the exploration of the search space, the evolution of pheromones which are laid on the ground by ants is used.

Facility Layout Problems (FLPs) deal with assigning m facilities to n locations, in such a way that the sum of the material handling costs is minimised. Quadratic Assignment Problem (QAP) was first formulated in 1957 which can be formulated as FLP when equal area facilities are considered.

The limited usage of QAP applications due to its NP-Hard structure, has caused the development of heuristics. Because optimal seeking procedures have been restricted to small sizes of the problem say maximum 15-20 facilities. Because of this, in this study, we presents the ant colony to solve the quadratic assignment problem (QAP) of which local search process is Simulated Annealing. AnnealingAnt Algorithm that is coded by C++ programming language is analysed by using current problems in the literature and is compared with other some heuristics.

Keywords: Quadratic Assignment Problem, Ant Algorithm, Simulated Annealing, Metaheuristics

1. GİRİŞ

Günümüzde kaynakların azalmasına karşın tüketim bilincinin ve rekabetin artması, gerek kamu gerek özel sektördeki işletmelerin ayakta kalabilme ve gelişebilmeleri için kendilerini sürekli iyileştirmeleri gereğini ön plana çıkarmıştır. Bu süreç içerisinde işletmeler oluşabilecek problemlerden dolayı bir takım maliyetlerle karşılaşacaktır. Bu problemler arasında tesis yerleşimi ve malzeme taşımaya bağlı problemler önemli bir yere sahiptir. İşletmelerin meydana gelecek bu maliyetleri en aza indirmeleri için etkin bir tesis planlaması yapması gerekmektedir.

Tesis yerleşimi problemi, malzeme taşıma maliyetlerinin ve oluşan sabit maliyetlerin minimize edilmeye çalışıldığı m tesisin n yerleşime ($m \leq n$) etkin bir şekilde atanmasıyla ilgilenir (Tavakkoli-Moghaddain ve Shayan, 1998).

Üretim maliyetinin %60'ını oluşturan malzeme taşıma maliyetlerini minimize etmeyi amaçlayan tesis yerleşimi problemi için geliştirilen çözüm, sistemin genel verimini artırıcı katkılar da sağlayacaktır.

Literatüre tesis yerleşim problemleri veya fabrika yerleşim problemleri olarak geçen yerleşim karar problemleri, kareli atama problemi, sınırsız kapasiteli tesis yerleşimi, p -medyan ve p -center olmak üzere dörde ayrılır (Yiğit ve Türkbey, 2003).

Tesis yerleşimi problemi (TYP), çözüm zamanının, problem boyutuna bağlı olarak üstel artış göstermesi nedeniyle NP (non-polinomial) problemlerdendir (Mirchandani ve Francis, 1990). Son yıllarda bilgisayar teknolojisinde gerçekleşen büyük gelişmeler ve araştırmacıların kabiliyetlerine rağmen, büyük boyutlu tesis yerleşimi problemleri optimizasyon metotlarıyla çözülememekte bunun yerine problemlere sistematik yaklaşımlarda bulunan ve çözüm zamanı polinom sınırlar içinde kalan sezgisellerle çözüm aranmaktadır.

Bu sezgisellerin büyük bir çoğunluğu belirli problemlere çözüm getirmek için geliştirilmiştir, bu nedenden dolayı bir problem için kullanılan başka bir problem için kullanılamamaktadır. Özellikle 1970'li yıllarda başlayan ve bugüne kadar geliştirilerek gelen Tavlama Benzetimi (TB), Tabu Arama (TA), Genetik Algoritmalar (GA), Yapay Sinir Ağları, Karınca Algoritması (KA) ve bunları temel alarak geliştirilen bazı algoritmalar, NP-Zor problemlerinde optimum çözüme yaklaşımda başarılı sonuçlar vermektedir.

Bu sezgisellerden KA, bir çok NP-zor problemleri için başarı ile uygulanabilen popülasyon tabanlı bir algoritmadır (McMullen, 2001). KA ismini, gerçek karıncaların yemlerini arama

davranışlarından esinlenerek oluşturulmasından almaktadır. Algoritmanın ana fikirlerinden bir tanesi koloninin bireyleri (karıncalar) arasında direkt bir iletişimin olmamasıdır (Dorigo vd., 1996). İletişim için gerçek karıncalar kimyasal bir madde olan fenomen izlerini kullanmaktadır (Gutjahr, 2000).

KA'nın temeli, bütün alternatif çözümler için istenebilirlik olasılığının belirlenmesi ve en yüksek olasılıkta istenebilirliğe sahip çözümün seçilmesine dayanır.

Bu çalışmada, Karınca Algoritması ve tesis yerleşimi problemlerine uygulanması incelenmiştir. Eşit alanlı büyük boyutlu problemlerin çözümü için, lokal arama prosedürü TB tabanlı "TavlamalıKarınca" algoritması geliştirilmiştir.

Çalışmanın birinci bölümünde TYP ve KA konusunda genel bir giriş yapıldıktan sonra, ikinci bölümde KA'nın gelişimi, uygulama alanları ve TYP'ye uygulanması üzerine bir literatür araştırması yapılmıştır.

Üçüncü bölümde, tesis yerleşimi problemleri, problemin formülizasyonu, kareli atama problemleri ve kareli atama problemlerine çözüm yaklaşımları incelenmiştir.

Dördüncü bölümde, öncelikle kısaca doğal karınca sistemi incelendikten sonra KA ve KA'nın ilk uygulaması olan gezgin satıcı problemine (GSP) uygulanması detaylı bir biçimde anlatılarak KA uygulamalarından bahsedilecektir.

Beşinci bölümde, bugüne kadar KA'nın tesis yerleşimi problemlerine uygulanması geniş bir biçimde incelenecektir.

Altıncı bölümde, bu çalışmada tesis yerleşimi problemi olarak ele alınan kareli atama problemi tanımlanacak ve bu probleme çözüm üretmek için geliştirilen yerel arama prosedürü TB tabanlı olan "TavlamalıKarınca" algoritması detaylı biçimde tanıtılacaktır.

Yedinci bölümde, diğer meta-sezgiseller tarafından denenilen kareli atama problem örnekleri geliştirilen "TavlamalıKarınca" algoritmasıyla denenecek ve algoritmanın uygun parametre setinin belirlenmesi için deney tasarımı düzenlenecektir. Algoritmanın performansını belirlemek için çözüm sonuçları "çözümler arası fark" kriteri göz önüne alınarak diğer meta-sezgiseller ile karşılaştırılacaktır.

Sonuç ve öneriler bölümünde, yapılan çalışmadan elde edilen sonuçlar değerlendirilecek ve gelecekteki çalışmalara yön verecek öneriler sunulacaktır.

2. LİTERATÜRÜN İNCELENMESİ

Karınca Algoritması bilinen diğer sezgisel metotlara bakarak oldukça yeni bir sezgisel metot olmasına rağmen literatürde bir çok alanda başarı ile uygulandığı görülmektedir.

KA ilk kez Dorigo (1992) tarafından ortaya atılmıştır. Dorigo, "Optimization, Learning and Natural Algorithms" isimli doktora tezinde bir karınca kolonisinin iki nokta arasındaki en kısa yolu bulmada kullandığı temel haberleşme mekanizmasının kombinatoryal optimizasyon problemlerine çözüm bulmada yararlı olacağını belirtmiştir.

Algoritmanın ilk uygulamasını Dorigo vd. (1992) Gezgin Satıcı Problemlerine (GSP) uygulamışlardır. Yapılan deneysel çalışmalar fenomen izinin güncellenmesi ve atama olasılıklarında kullanılan parametrelerin en iyi kombinasyonu saptamıştır. Bu çalışmanın sonucunda algoritma küçük boyutlu GSP'de optimuma yakın sonuçlar elde etmiştir.

1994 yılında, Colorni vd. KA'yı atölye tipi iş çizelgeleme problemleri için denemişlerdir. Küçük boyutlu problemlerde başarılı sonuçlar elde edilmesine rağmen problem boyutu büyüdükçe algoritmanın performansı düşmüştür.

Dorigo vd. (1996 ve 1997) simetrik ve asimetrik GSP için karınca algoritmasını kademeli olarak geliştirmişler ve büyük boyutlu problemler için de yüksek çözüm performanslarına ulaşmışlardır.

1997 yılında Gambardella ve Dorigo KA'yı ardışık sıralama problemine çözüm geliştirmede denemişler ve başarılı sonuçlar elde etmişlerdir, yine aynı yıl Kuntz vd. (1997) KA'yı bölme problemi için kullanırken, Schoonderwoerd vd.(1997) telekomünikasyon ağı problemleri için denemişler ve başarılı sonuçlar elde etmişlerdir.

Farklı boyut ve kısıtlar altında, telekomünikasyon ağı problemi çözümü için KA uygulamasına DiCaro ve Dorigo (1998) ve Navarro ve Sinclair (1999) çalışmışlardır.

Forsyth ve Wren (1997) ilki Dorigo vd. (1993) tarafından yapılan KA'nın atölye tipi iş çizelgeleme problemi üzerine uygulanmasını geliştirerek büyük boyutlu problemlerde yüksek performans elde etmişlerdir.

Bullnheimer vd. (1999) kombinatoryal optimizasyon problemleri içinde önemli bir yere sahip araç sıralama problemine KA'yı başarılı şekilde uygulamışlardır.

Stützle ve Hoos (2000) mevcut KA'da her bir iterasyon sonunda yapılan fenomen miktarının

güncellenmesinde ve fenomen miktarının belirli limitler içinde tutulmasında mevcut algoritmaya yenilikler getiren MaksimumMinimum Karınca Sistemi (MaksimumMinimum Ant Systems-MMAS) isimli yeni bir algoritma geliştirmişlerdir. MaksimumMinimum Karınca Sistemi, GSP için uygulanmış ve daha iyi sonuçlar elde edilmiştir.

McMullen (2001) amaçları ayar zamanlarını minimize eden ve malzeme kullanma oranlarının durağanlığını optimize eden tam zamanında üretim sıralama problemi için KA yaklaşımı üzerine çalışmıştır.

Randal ve Lewis (2002) KA'nın paralel uygulamasını geliştirmişlerdir. Uygulamalarını GSP'de denemişler ve algoritmanın etkinliğinin arttığını göstermişlerdir.

T'kindt vd. (2002) KA'nın iki ölçütlü iki makineli akış tipi çizelgeleme problemine uygulayarak yeni bir algoritma geliştirmişlerdir. Yaptıkları deneysel çalışmalar sonucunda iki ölçütlü iki makineli akış tipi çizelgeleme problemlerinde en etkili sonuçları veren algoritmalarından daha iyi sonuçlar elde etmişlerdir.

Lee vd. (2002) farklı bir uygulama alanı olarak Silah-Hedef atama problemine KA'yı uygulamışlardır. Yine farklı bir uygulama alanı olan klavye düzenleme probleminin optimizasyonu için KA uygulamasına Eggers vd. (2003) çalışmışlardır.

KA'nın tesis yerleşimi problemlerine ilk uygulaması Maniezzo vd. (1994) tarafından kareli atama problemine (KAP) uygulanması ile gerçekleştirilmiştir. Uygulamada kullanılan algoritma, GSP uygulamasından esinlenerek, her bir çözüm çifti için istenebilirlik olasılığını hesaplayıp, en yüksek olasılığa sahip olanı atama prensibine dayandırılmıştır. Bu uygulama, küçük boyutlu problemlerde yüksek çözüm performansı elde ederken büyük boyutlu problemlerde başarılı sonuçlar verememiştir. Bütün alternatifler için olasılıkların hesaplanması da algoritmayı yavaşlatan bir etken olmuştur.

Gambardella vd. (1997, 1999) temelleri Taillard ve Gambardella (1997) tarafından atılan KAP için Melez Karınca Kolonisi Sistemini (MKKS) geliştirmişlerdir. Fenomen izine bağlı gelişimi standart KA'dan farklı olan bu uygulamada algoritmanın performansını artıracak yerel arama prosedürü de kullanılmıştır. Bu özelliklere sahip algoritma büyük problemlerin çözümünde de başarılı sonuçlar vermiştir.

KAP için KA uygulamasını son olarak Talbi vd. (2001) geliştirmiştir. Bu uygulamada temel farklılıklar algoritmanın paralel yapısı ve yerel arama prosedürü olarak TA'nın kullanılmasıdır. "AnTabu" olarak adlandırılan bu algoritma ile MKKS'den daha yüksek

özüm performansları elde edilmiştir.

KA'nın tesis yerleşimi problemine KAP'den farklı uygulaması, Solimanpur vd. (2003) tarafından esnek imalat sistemlerinde (EİS) tek sıralı tesis yerleşimi problemine uygulanması ile yapılmıştır. Solimanpur uygulamasında, bilinen problemler için mevcut sezgisellerden daha iyi sonuçlar elde etmiştir.



3. TESIS YERLEŞİMİ

Üretim araçlarının, yardımcı tesislerin veya iş istasyonlarının fiziksel konumları açısından bir bütün olarak koordinasyonuna tesis yerleşimi denir.

İyi planlanan bir tesis yerleşiminin sistemin genel verimini artırması ve toplam imalat maliyetinin önemli bir kısmını oluşturan malzeme taşıma maliyetlerini azaltması tesis yerleşiminin önemini ortaya koymaktadır.

3.1 Tesis Yerleşimi Problemleri

Tesis yerleşimin planlanması esnasında ürün veya hizmet üretiminin verimliliğini maksimuma çıkarmak ve kaynak kullanımında en yüksek etkinliğe ulaşmak amaçlanmalıdır. Tesis yerleşimi problemi, bu amaca ulaşmada maliyetleri minimize eden tesisin yerleşimi etkin bir biçimde atamakla ilgilenir.

Tesis yerleşimi konusunda yapılan çalışmalar 1960'lı yıllarda Armour ve Buffa (1963) ve Nugent (1968)'in yaptığı çalışmalar ile başlamıştır ve günümüzde hala bir çok araştırmacı tarafından çalışılmaktadır. Bu alanda yapılan çalışmalar, daha az hesaplama zamanlarına ihtiyaç duyulduğu için genellikle küçük boyutlu problemler üzerinedir (Kulluk, 2003).

Tesis yerleşimi problemlerinde genellikle iki amaç optimize edilmeye çalışılır. Bu amaçlardan birincisi, iki tesis yeri arasındaki personel veya malzeme tarafından kat edilen mesafeyi minimize edecek yerleşim planının yapılması, ikincisi ise önceden belirlenen yakınlık oranları ölçümlemlerini maksimize edecek yerleşim planının yapılmasıdır (Rosenblatt, 1986).

Heregua ve Kusiak (1991), tesis yerleşiminde temel amacı, makineler/istasyonlar arasında yapılan insan gücü ve malzeme taşımalarının yanında istasyonlar arasındaki mesafenin de minimize edilmesi olarak tanımlamışlardır.

Tesis yerleşimi problemleri, toplam imalat maliyetlerinin % 30-75'ini malzeme taşıma maliyetleri oluşturduğu için birçok imalat işlemlerinin başarısı için stratejik bir öneme sahiptir (Chiang ve Chiang, 1998). İyi planlanmayan bir yerleşim, proses içi stokun yığılmasına, malzeme taşıma sisteminin aşırı yüklenmesine, yetersiz setuplara ve uzun kuyruklara sebep olur (Jajodia vd., 1992).

Tesis yerleşimi problemi maliyet, uzun süreli stok ve yapılması çok pahalıya mal olan ve kolayca yapılamayan değişiklikler gösterir. Tesisin yeniden yerleşimi ise sadece zaman

kaybına sebep olmaz aynı zamanda malzemelerin akışını ve çalışanların faaliyetlerini bozar (Sule, 1988). Tesis yerleşimi tasarımı, üzerinde önemle durulması gereken bir konudur.

3.2 Problemin Formülizasyonu

Tesis yerleşimi problemlerinde genellikle amaç, ya malzeme taşıma miktarını minimize etmektir veya yakınlık oranlarını maksimize etmektir. Bu nedenle tesis yerleşimi problemleri güdülen bu iki temel amaca göre formülize edilir.

Tesis yerleşimi probleminde, malzeme taşıma miktarı minimize edilmesi amaçlanırsa problem;

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n (f_{ij} \cdot c_{ij}) d_{ij} \quad (3.1)$$

Notasyonlar:

f_{ij} = i departmanından j departmanına akış miktarı,

c_{ij} = i departmanından j departmanına bir birim uzaklığa bir birim malzemenin taşıma maliyeti,

d_{ij} = i departmanı ile j departmanı arası uzaklık,

şeklinde formülize edilir.

Departmanlar arası uzaklığın ölçülmesinde, giriş/çıkış noktalarının belirlenmesindeki zorluk sebebiyle ilk departmanın çıkış noktasıyla ikinci departmanın giriş noktası arasındaki mesafenin kullanılması yerine genellikle merkezden merkeze (Center To Center) metodu kullanılır. Giriş/çıkış noktaları veya merkezler arasındaki mesafenin hesaplanmasında genel olarak öklit bağıntısı veya rektilineer uzaklık kullanılır.

Tesis yerleşimi probleminde amaç yakınlık derecelerinin maksimizasyonu ise problem;

$$\text{Max} \sum_{i=1}^n \sum_{j=1}^n Y_{ij} \cdot m_{ij} \quad (3.2)$$

Notasyonlar:

Y_{ij} = i departmanı ile j departmanı arasındaki yakınlık oranı,

m_{ij} = 1 i departmanı j departmanına yakınsa; 0 i departmanı j departmanına yakınsa.

şeklinde formülize edilir.

Bu iki formülizasyonun eksik yönlerini azaltmak maksadıyla Rosenblatt (1979) bu iki formülizasyonu tek bir formülde birleştirmiştir. Elde edilen yeni formülizasyon, malzeme akış uzaklığı ile yakınlık oranı arasındaki ağırlandırılmış farkın minimize edilmesiyle elde edilir.

3.3 Kareli Atama Problemi (KAP)

Bir çok araştırmacı tesis yerleşimi problemlerinin çözümü için farklı formüller kullanmalarına rağmen, genel olarak potansiyel iki tesis arasındaki mesafenin değerlendirmesinde referans noktası olarak yerleşimin merkezini kullanan KAP kullanılan en popüler formüldür (Chiang ve Chiang, 1998).

“Kareli Atama Problemi” isminin seçilme sebebi, amaç fonksiyonunun, değişkenlerin ikinci dereceden polinomial fonksiyonu olması ve kısıtlarının atama probleminin kısıtlarına benzemesidir. KAP’nin amaç fonksiyonu, alınan mesafeyi ve iş akışını ifade eden malzeme taşıma maliyetini minimize etmek amacıyla n yerleşime n tesisin (departman, makine istasyonu v.b.) optimum atamasını bulur. KAP NP-Tam problemler sınıfına dahil olması sebebiyle çözümü için sezgisel metotlar kullanılmıştır ve sezgisel yöntemler KAP için genelde iyi sonuçlar vermiştir.

KAP amaç fonksiyonunun kareli olmasının yanında konveks olmayan bir yapıya sahiptir. KAP çözüm uzayında $(n!)$ kadar çözüm noktası ve birden çok yerel optimum çözüm vardır. Bilinen hiçbir metot 20’den daha fazla tesis sayısı olan KAP için mümkün hesaplama zamanında optimum çözümü bulamaz (Sha ve Chen, 2001).

Problemin tanımlanması esnasında, tesislerin her bir çifti arasında meydana gelen akışlar ve yerleşimlerin her bir çifte arasında bir birim malzemenin taşıma maliyeti bilinir. 6 tesisin 6 yerleşime atanmasıyla ilgili örnek ele alınırsa gerekli olan akışlar ve maliyetler 6×6 ’lık bir matriste gösterilir (Nugent vd., 1968).

$$\text{Maliyet} = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 & 1 & 2 \\ 2 & 1 & 0 & 3 & 2 & 1 \\ 1 & 2 & 3 & 0 & 1 & 2 \\ 2 & 1 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 2 & 1 & 0 \end{bmatrix} \quad \text{ve} \quad \text{Akış} = \begin{bmatrix} 0 & 5 & 2 & 4 & 1 & 0 \\ 5 & 0 & 3 & 0 & 2 & 2 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 5 & 2 \\ 1 & 2 & 0 & 5 & 0 & 9 \\ 0 & 2 & 0 & 2 & 9 & 0 \end{bmatrix}$$

Bundan sonra belirlenmesi gereken, toplam taşıma maliyetinin minimum olması için hangi yerleşime hangi tesisin atanması gerektiğidir.

Literatürde, TYP'nin formülizasyonu için üç ana yaklaşım mevcuttur.

1. Kombinatorial optimizasyon problemi formülizasyonu (Burkard, 1990),
2. Tamsayılı programlama formülizasyonları
 - a. Genel tamsayılı lineer olmayan programlama,
 - b. Koopmans-Beckmann formülizasyonu (Koopmans ve Beckman, 1957),
 - c. Tamsayılı lineer programlama,
 - d. Karma tamsayılı lineer programlama,
3. İzleme formu (Finke vd. 1987).

Bu yaklaşımlar içinde tesis yerleşimi problemleri için en çok kullanılanları, kombinatorial optimizasyon problemi formülizasyonu ve Koopmans-Beckmann formülizasyonudur.

3.3.1 Kombinatorial Optimizasyon Problemi Formülizasyonu

Kombinatorial optimizasyon problemi formülizasyonunda, n yerleşime atanması gerek n tesis düşünülmüştür. Boyutu n olan bir problemde $n \times n$ 'lik $A = (a_{ij})$ ve $B = (b_{ij})$ matrisleri verilir ve n elemanın permütasyonu ($\Pi(n)$) içinden (3.3)'de tanımlanan formülü minimize eden π^* çözümü bulunur.

$$\min_{\pi \in \Pi} f(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi(i)j} \quad (3.3)$$

Notasyonlar:

$f(\pi)$ = π çözümü için amaç fonksiyonun değeri,

a_{ij} , i yerleşimi ile j yerleşimi arasındaki mesafe,

b_{ij} , i tesisi ile j tesisi arasındaki akış miktarı,

$b_{\pi(i)j}$, π çözümünde i tesisi ile j tesisi arasındaki akış miktarını ifade eder.

Bu formülizasyonda iki önemli kısıt vardır.

1. Bir tesis en fazla bir yerleşime atanmalıdır,
2. Bütün tesisler atanmalıdır.

3.3.2 Koopmans-Beckmann Formülizasyonu

Kopmans ve Beckman (1957) tarafından yapılan ve literatüre KAP üzerine yapılan ilk çalışma olarak geçen bu formülizasyonun amaç fonksiyonu (3.4)'deki gibi tanımlanmıştır.

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n A_{ijkl} x_{ij} x_{kl} \quad (3.4)$$

Formülizasyonun kısıtları;

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j,$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i,$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \text{ şeklindedir.}$$

$$x_{ij} = \begin{cases} 1, & i \text{ tesisi } j \text{ ye atandı ise,} \\ 0, & \text{aksi taktirde,} \end{cases}$$

$$A_{ijkl} = \begin{cases} f_{ik} d_{jl} & i \neq k \text{ veya } j \neq l \\ F_{ij} & i = k \text{ ve } j = l \\ \infty & i = k \text{ ve } j \neq l \end{cases}$$

Notasyonlar:

f_{ik} , i ve k tesisleri arasındaki iş akışı, $f_{ii} = 0$,

d_{jl} , j ve l yerleşimleri arasındaki mesafe, $d_{jj} = 0$ 'dır.

Bu formülizasyon iki bölümden oluşmaktadır. Birinci bölüm tesislerin yerleşimlere atama maliyetlerini, ikinci bölüm ise tesislerin aralarındaki iş akışına bağlı olan değişken maliyetleri gösterir.

KAP'nin formülizasyonu ile ilgili olarak bir çok araştırmacı çalışmalar yapmıştır. Bu çalışmalar arasında Francis ve White (1974); Kusiak ve Heragu (1987), Kusiak (1990) ve Francis vd. (1992) en önemlileridir.

3.4 KAP için Çözüm Yaklaşımları

KAP NP-Tam'dır (Sahni ve Gonzalez, 1976). Gilmore (1963) ve Lawler (1963) KAP çözümü için optimal prosedürler geliştirmişlerdir. Diğer optimal prosedürleri ise Burkard ve Stratmann (1978); Bazaraa ve Elshafei (1979), ve Bazaraa ve Sherali (1980) çalışmalarıyla literatüre kazandırmışlardır. Fakat bu prosedürler 20'den daha büyük olan KAP'ler için çok büyük hesaplama zamanları gerektirmiştir (Resende vd., 1995).

Optimal algoritmaların büyük boyutlu problemler için ortaya çıkan bu dezavantajı araştırmacıları problemin çözümü için sezgisel algoritmalar geliştirmeye yöneltmiştir. Bu sezgiseller genel olarak yapılandırma metotları (ALDEP ve CORELAP) ve iyileştirme metotları (CRAFT) olarak sınıflandırılmaktadır. Yapılandırma metotları, departmanları zemin alanına yerleştirme sırası olan seçim sırasını belirleme ve bu departmanları amaçlar doğrultusunda alanlara yerleştirme aşamalarından oluşur. İyileştirme metotları çözüm kalitesini artırmak maksadıyla rasgele veya belirli bir iyileştirme tekniği ile hiçbir gelişme sağlanamayana kadar departmanların yerlerini değiştirir. İyileştirme metodu yapılandırma metoduna göre daha iyi sonuçlar verir (Tam, 1992). Bu iki metodun en önemli dezavantajı, çözümün başlangıç çözümüne bağlı olmasından kaynaklanan yerel optimum çözümlere takılma olasılığının yüksek olmasıdır.

Optimal prosedürlerin 20'den daha büyük olan KAP'ler için uygulanamaması sebebiyle araştırmacılar sezgisel metotlar üzerine çalışmışlardır. Bu bağlamda Kusiak (1990) ve Burkard (1990) KAP için sezgisel algoritmalar geliştirmişlerdir.

Son zamanlarda, araştırmacıların bir çoğu KAP için zeki sezgisel yaklaşımların uygulamaları üzerine çalışmalar yapmışlardır. Bu yaklaşımlar içinde en popüler olanları;

- Tavlama Benzetimi,
- Tabu Arama,

- Genetik Algoritmalar ve
- Karınca Algoritmasıdır.

Burkard ve Rendl (1984) KAP için Tavlama Benzetimi ilk kez uygulayanlardır. Çalıştıkları KAP'ler için bilinen en iyi çözümlerinden sadece %1-2'lik sapmalar elde etmişlerdir. TB tekniğinin kötü çözümleri de saklayarak geriye doğru hareketi de sağlar.

Sharpe ve (1985) ve Sharpe ve Marksjo (1985) etkileşimli tesis yerleşimi kararları için TB'yi kullanmışlardır. Bonomi ve Lutton (1986) KAP'lerin maliyet fonksiyonlarının optimum değerlerinin asimptotik davranışlarını incelemişlerdir. Wilhelm ve Ward (1987) KAP'ler için TB'nin sonuçları üzerine deneysel sonuçların büyük bir setini hazırlamışlardır. Jajodia vd. (1992) hücresel imalatla tesis yerleşimi problemi için TB tekniklerini kullanmıştır. Kouvelis ve Chiang (1992) esnek imalat sistemleri için KAP'nin özel durumlarının uygulanmasında problem çözümü için TB kullanmışlardır. Kouvelis vd. (1992) iş akış matrislerinin doğal yapısını araştırmada TB ve penaltı metodunu kullanmışlar ve yüksek performanslı sonuçlar elde etmişlerdir.

Skorin-Kapov (1990) KAP için Tabu Aramayı ilk uygulayan araştırmacıdır. Tabu Arama algoritması yerel optimumlara yakalanmamak için yapılan değişiklikleri tabu listesine aktarır. TA'nın TB'den üstün olan tarafı daha az hesaplama zamanı gerektirmesidir. Bunun yanında sadece iyi çözümleri kullanması nedeniyle yerel optimuma yakalanma riskinin fazla olması TB'den zayıf olan yönüdür. Kelly vd. (1994) çeşitlendirme stratejilerini uygulamışlar ve büyük boyutlu KAP örnekleri için daha kaliteli çözümler elde etmişlerdir.

Tam (1992), tesis yerleşimi problemi için Genetik Algoritmaları uygulamıştır. Küçük bir çözümden başlayarak bazı olasılıkları göz önüne alarak yeni nesiller üretilip iteratif olarak optimumu arayan GA'nın KAP için uygulamasını Tavakkoli-Moghaddin ve Shayan (1998) çalışmıştır. Çalışmada eşit ve farklı alanlı tesis yerleşimi problemleri için kareli atama formülizasyonunu çözen GA, TB ve TA'dan dahi iyi sonuçlar elde edilmiştir.

Dorigo vd. (1994) Karınca Algoritmasını KAP için denemişler ve küçük boyutlu problemlerde başarılı sonuçlar elde etmişlerdir. Büyük boyutlu problemlerde diğer zeki sezgisellerden daha iyi sonuçlar elde edebilecek melez yapılı bir algoritma olan MKKS'yi Gamberdella vd. (1999) geliştirmiştir. Talbi vd. (2001) yerel arama prosedürü TA olan AnTabu algoritmasının paralel uygulamasını geliştirmişler ve literatürde önceki zeki sezgiseller tarafından çalışılan bir çok KAP için en iyi sonucu elde etmişlerdir. Solimanpur vd. (2003) esnek imalat sistemlerinde tek sıralı tesis yerleşimi probleminin çözümü için KA

uygulamasına çalışmışlardır.

Son yıllarda literatürde, MKKS ve AnTabu gibi melez algoritmalara sıkça rastlanmaktadır. Melez algoritmaların yapısı genellikle iki zeki sezgisel metottan oluşmaktadır. Bu iki zeki sezgisel metottan bir tanesi yerel çözümleri araştırırken ikincisi bu çözümden optimum çözüme ulaşacak ilerlemeyi sağlar. Melez algoritmaların geliştirilmesindeki amaç kullanılan metotların birbirlerinin eksikliklerini kapatmasıdır. Melez algoritmalar, yapısını oluşturan zeki sezgisellere göre daha iyi sonuçlar vermektedir. Lee vd. (2002) melez genetik algoritmalar üzerine çalışmışlardır. Çalışmalarında genetik algoritmaların zayıflıklarını, yerel aramada hızlı çözümler bulan TA ve TB algoritmalarının kuvvetli noktalarını kullanarak ortadan kaldırmışlardır.



4. KARINCA ALGORİTMASI

Bir çok kombinatorial optimizasyon problemlerinin çözümü için kullanılabilen, KA ilk kez Dorigo (1992) tarafından ortaya atılmıştır. Dorigo, "Optimization, Learning and Natural Algorithms" isimli doktora tezinde bir karınca kolonisinin iki nokta arasındaki en kısa yolu bulmada kullandığı haberleşme mekanizmasını kombinatorial optimizasyon problemlerine çözüm üretmekte kullanmıştır.

Algoritmanın ilk uygulamasını Dorigo vd. (1992) GSP'ye uygulamışlardır. Algoritmanın doğruluğunu ispatlayan bu çalışmada yapılan deneysel çalışmalarda fenomen izinin güncellenmesi ve atama olasılıklarında kullanılan parametrelerin en iyi kombinasyonu saptanmıştır.

Karınca Kolonisi Optimizasyonu (Ant Colony Optimization-ACO) bir çok NP-Zor kombinatorial optimizasyon problemleri için başarı ile uygulanabilen populasyon tabanlı bir algoritmadır (Gambardella ve Dorigo, 1997). Bu bölümde öncelikle karınca sisteminden bahsedilecektir. Daha sonra karınca sisteminin kombinatorial optimizasyon problemlerine adaptasyonu sonucu oluşan karınca algoritması incelenecek ve GSP uygulamasıyla algoritmanın genel yapısı tanıtılacaktır.

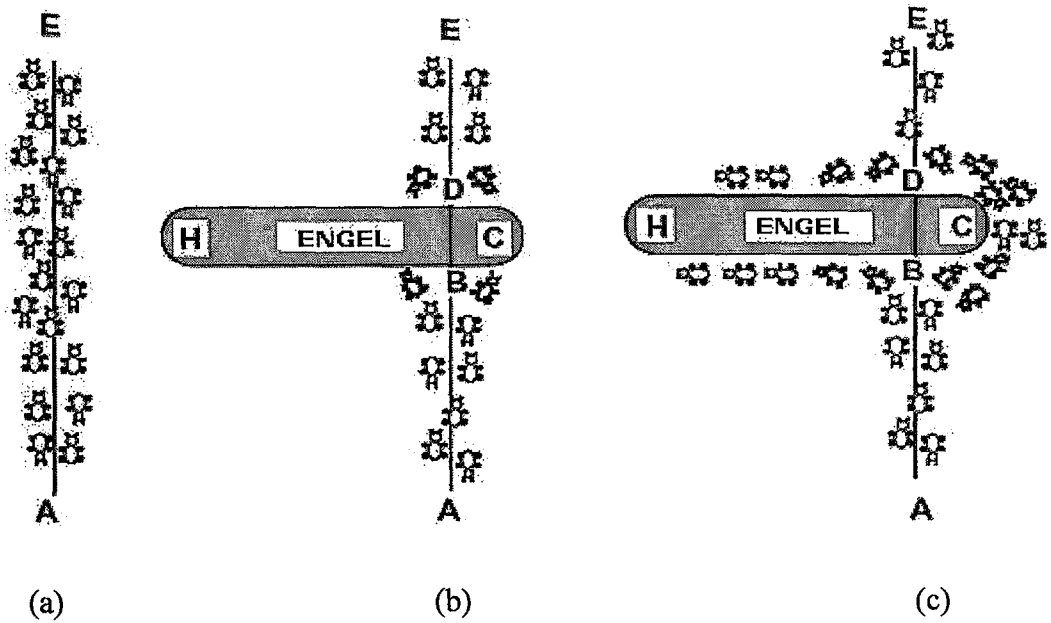
4.1 Karınca Sistemi

KA ismini, yapısının gerçek karıncaların yemlerini arama davranışlarından esinlenerek oluşturulmasından almaktadır. Algoritmanın ana fikirlerinden bir tanesi koloninin bireyleri (karıncalar) arasında direkt bir iletişimin olmamasıdır. İletişim için gerçek karıncalar kimyasal bir madde olan fenomen izlerini kullanmaktadır.

Karıncalar gibi neredeyse kör hayvanların kolonisinin beslenme kaynaklarına ulaşmada en kısa yolu bulmayı nasıl yönettikleri, günümüzde biyoloji bilimi tarafından da çalışılan bir problemdir. Koloniler yönlere ilişkin bireyler arasında bilgi iletişimini sağlamada ve nereye gideceklerine karar vermede fenomen izlerini bir araç olarak kullanır. Hareket halindeki bir karınca yerin üzerine değişik miktarlarda fenomenler yayar, ve gittiği yolu bu madde vasıtasıyla işaretler. Aslında bir tek karınca rasgele hareket ederken önceden başka karıncalar tarafından yayılan izle karşılaşabilir ve büyük olasılıkla onu takip eder, böylece kendi fenomenini de yola yayarak yol üzerindeki izi güçlendirir. Ortak davranış, izi takip eden ne kadar çok karınca varsa izlenilen yolun istenebilirlik oranının o kadar artmasına neden olur. Böyle bir davranış kendisini güçlendiren bir proses olan otokatalitik davranışın bir çeşidini

ortaya koyar. Süreç, bir karıncanın önceden başka karıncalar tarafından seçilen yolu seçmesi olasılığının yüksek olduğu yerde pozitif bir geri besleme mekanizması ile tanımlanmıştır.

Şekil 4.1a'da gösterilen deneysel görünümde, A yuvayı E ise yemek kaynağını göstermektedir. A-E yolu üzerinde hareket eden karıncaların yollarına Şekil 4.1b'deki gibi bir engel konduğunda A noktasından gelen karıncalar B noktasına geldiklerinde sağa veya sola dönüp dönmeye karar vermek zorundadır. Karıncalar bu durum karşısında seçim yaparken önceki karıncalar tarafından yola bırakılan fenomen izinin yoğunluğundan etkilenirler. İlk durumda B veya D noktasına varmış ilk karınca sağa veya sola dönmek için aynı olasılığa sahiptir, bu nedenle karıncalar yaklaşık oranda ikiye bölünürler. Zamanla BCD yolu üzerindeki fenomen seviyesinin daha fazla olması karıncaya daha güçlü bir uyarıcı verir. Bu nedenle karıncanın BCD yolunu tercih etmesi daha büyük bir olasılıktır. Çünkü Şekil 4.1c'de görüldüğü gibi BCD yolu BHD'den daha kısadır ve BCD yolunu izleyen birinci karınca BHD yolunu takip eden birinci karıncadan daha önce D'ye varacaktır. E'den D'ye dönen bir karıncanın DCB yolu üzerinde daha güçlü bir iz bulacağı olasılığı yüksektir. Sonuç olarak birim zamanda BCD yolunu takip eden karınca sayısı BHD yolunu takip eden karınca sayısından daha fazladır. Bu nedenle, daha kısa yol üzerindeki fenomen miktarı daha uzun olan yol üzerindeki daha hızlı artar ve bu nedenle takip için bazı karıncaların daha kısa olan yol yönünde eğilim göstermesi olasıdır. Zamanla da bütün karıncalar daha kısa yolu seçeceklerdir.



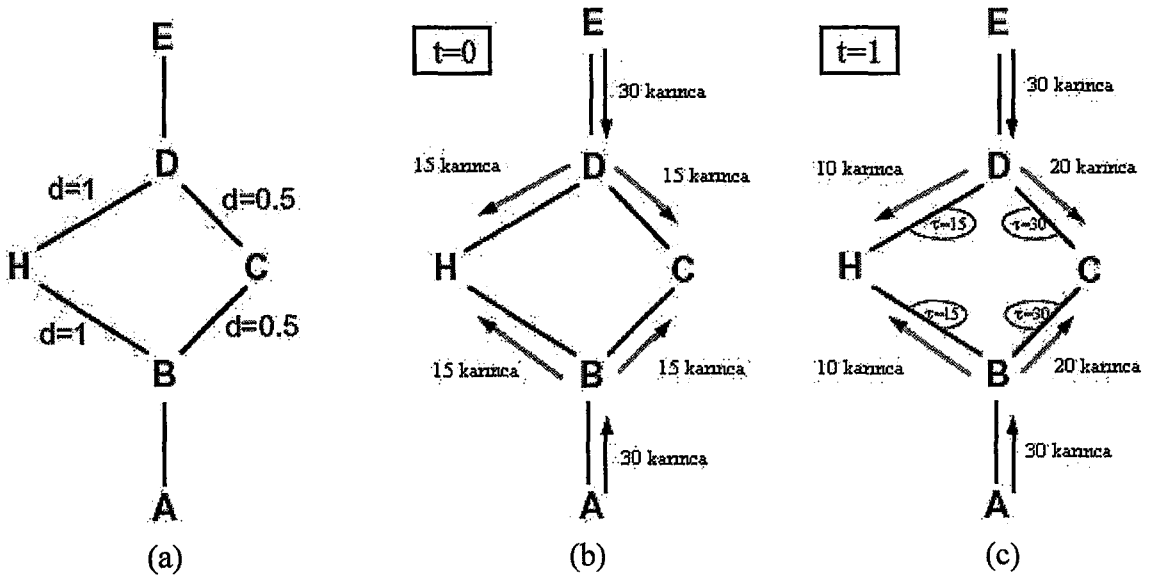
Şekil 4.1 Gerçek karıncaların davranışları

Sistemin sayısal veriler ile izahı Şekil 4.2’de gösterilmiştir. Şekil 4.2a grafiği, şekil 4.1b durumunun mümkün bir Karınca Sistemi açıklamasını ele almaktadır. Bu grafikte D ve H arasındaki mesafe B ve H arasındaki mesafeye eşittir ve C üzerinden B ve D arasındaki mesafeye denktir. D ve B arasındaki yolun ortasına C yerleştirilmiştir. $t = 0, 1, 2, \dots$ zaman aralıklarının düzenli bir biçimde artırıldığında nelerin meydana geleceği incelendiğinde, her zaman biriminde 30 yeni karınca A’dan B’ye 30’u da E’den D’ye gelir. Her bir karıncanın her zaman biriminde 1 hızında yürüdüğü ve bir karınca yürürken 1 yoğunlukta fenomen izini bir birim zaman içinde yaydığı varsayılırsa;

$t = 0$ anında henüz iz yoktur, ama B’de ve D’de 30 karınca vardır. Gidecekleri yolların seçimleri tamamen rassaldır. Bu nedenle, her iki düğümde ortalama 15 karınca H yönünde ve 15 karınca C yönünde gidecektir (Şekil 4.2b).

$t=1$ ’de A’dan B’ye gelen 30 yeni karınca H’a götüren yol üzerinde B’den bu yola gelen 15 karıncadan yayılan fenomenlerden dolayı 15 yoğunluğunda bir iz bulur ve C yolu üzerindeki, C vasıtasıyla D’den gelen C’ye ulaşan 15 karınca ve B’den bu yola gelen 15 karınca vasıtasıyla yayılan izlerin toplamı olarak elde edilen 30 yoğunluğunda bir iz bulunur (Şekil 4.2c). Bir yolun seçilmesi olasılığı bu nedenle eğilim gösterir, bu yüzden H’a doğru giden karıncaların beklenen sayısı 10’a karşılık C’ye giden bunun iki katı olacaktır. E’den gelen D’deki 30 yeni karınca için aynısı geçerlidir.

Bu süreç, neticede bütün karıncaların en kısa yolu seçmesine kadar devam eder.



Şekil 4.2 Yapay karıncaların davranışları

Kabul edilen yargı; verilen bir noktadaki bir karınca farklı yollardan birisini seçmek zorunda ise, daha büyük bir olasılıkla önceki karıncalar tarafından seçilmiş olan yolu seçer. Bunun yanında yüksek iz seviyeleri kısa yol ile eş anlamlıdır.

Karıncalarının yem bulmada kullandıkları bu sisteme Karınca Sistemi denilmiştir. Bu sistemin kombinatoriyal optimizasyon problemlerine adaptasyonu sonucu oluşan algorithmaya ise Karınca Algoritması ismi verilmiştir.

Karınca sistemi gerçekte bazı farklılıklar içerir. Örneğin;

- Yapay karıncaların biraz hafızası vardır
- Tamamıyla kör değildirler
- Her karınca bir birim zamanda bir birim yol kat etmektedir

İlk kez 1992 yılında ortaya atılan karınca sisteminin kombinatoriyal optimizasyon problemlerine çözüm üretme yaklaşımı günümüze kadar bir çok problem türü üzerinde denenmiştir.

Kombinatoriyal optimizasyon problemleri için uygulanan KA'nın algoritmik çatısı şu şekilde oluşturulabilir.

ADIM-1 (Başlangıç)

- İlk fenomen miktarının hesaplanması

ADIM-2 (İterasyonlar)

- Her bir karınca için tekrarla
 - Fenomen izini kullanarak çözümün geliştirilmesi
 - Fenomen miktarının güncellenmesi
- Durdurma kriteri sağlanana kadar

KA'nın günümüze kadar, gezgin satıcı problemleri, tesis yerleşimi problemleri, atölye tipi iş çizelgeleme problemleri, araç rotalama problemleri vb. bir çok kombinatoriyal optimizasyon problemi üzerine başarılı uygulamaları yapılmıştır.

4.2 Karınca Algoritması ve GSP Uygulaması

Gerçek karıncaların doğal davranışlarından esinlenerek geliştirilen bu algoritma aşağıda ki

verilen karakteristiklere sahiptir (Dorigo vd, 1996):

- *Çok yönlülük*; Bazı problemlerin benzer versiyonları için uygulanabilir, örneğin; gezgin satıcı probleminden asimetrik gezgin satıcı problemine geniş bir uzama vardır.
- *Güçlülük*; Diğer kombinatoriyal optimizasyon problemleri için sadece çok küçük değişikliklerle uygulanabilir, örneğin: kareli atama problemleri ve atölye tipi iş çizelgeleme problemi
- *Populasyon merkezlilik*; Arama mekanizması olarak pozitif geri beslemenin kullanımına izin verir. Paralel uygulamalar için sistem uygunluğu da oluşturulur.

Bu sistem ile karşılaştırdığımız TB ve TA gibi diğer popüler yaklaşımlar tarafından paylaşırlan bir problemdir.

Algoritmanın değerlendirilmesi hususunda diğer sezgisel yaklaşımlarla daha kolay karşılaştırması için ilk olarak iyi bilinen gezgin satıcı problemine uygulanmıştır (Dorigo vd.,1996). Bu problemlerden elde edilen başarılı sonuçlar algoritmanın doğruluğunu ispatlamıştır.

GSP, n kasabanın a seti için her bir kasabayı bir kez ziyaret eden minimum uzunluktaki kapalı turu bulma problemini ifade eder.

d_{ij} , i ve j kasabaları arasındaki uzunluk $(i, j) \in A$

Öklit gezgin satıcı problemine göre d_{ij} , i ve j kasabaları arasındaki öklit mesafedir.

Bir t anında toplam karınca sayısı (4.1)' deki gibi bulunur.

$$M = \sum_{i=1}^n m_i(t) \quad (4.1)$$

Notasyonlar:

$m_i(t)$, t zamanında i kasabasında bulunan karınca sayısı ($i=1, 2, 3, \dots, n$).

Algoritmada karıncalar optimum çözüme ulaşmada kullanılan araçları ifade eder ve her bir karıncanın aşağıda verilen özelliklere sahip olduğu varsayılır.

- Gidilecek kasabanın seçilme olasılığı, kasaba mesafesi ve izlerin miktarının bir fonksiyonudur.
- Tam bir tur yapmak için karınca mecburdur, bir tur tamamlanana kadar önceden ziyaret edilen kasabalara geçişler engellenir.
- Bir tur tamamlanırken ziyaret edilen her bir kenar üzerinde iz olarak bilinen bir madde yayılır.

Her bir karınca t anında sonraki kasabayı seçerken $t+1$ zamanı olacaktır. Bu nedenle $(t, t+1)$ arasında m karınca tarafından meydana getirilen m taşıma KA'nın bir iterasyonudur. n boyutlu bir problemde, n iterasyonda her bir karınca bir tur tamamlar.

Bu durumda fenomen iz yoğunluğu (4.2) eşitliğinden faydalanılarak bulunur.

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (4.2)$$

Notasyonlar:

τ_{ij} , t zamanında (i,j) kenarı üzerindeki fenomen izi yoğunluğu,

ρ , sabit bir katsayıdır, t ve $t+n$ zaman aralığında yok olmayan iz miktarıdır. Bu nedenle, $\rho < 1$ kullanılmalıdır.

$\Delta \tau_{ij}$, m adet karıncanın (i,j) kenarı üzerindeki fenomen izi yoğunluğunun toplamı (4.3) kullanılarak elde edilir.

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (4.3)$$

Notasyonlar:

$\Delta \tau_{ij}^k$, t ve $t+n$ zaman aralığında k 'ıncı karınca tarafından (i,j) kenarı üzerinde yayılan iz maddesinin uzunluğunun her birimindeki miktarı (4.4).

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{karınca } (i, j) \text{ yolunu kullanırsa,} \\ 0 & \text{aksi takdirde,} \end{cases} \quad (4.4)$$

Notasyonlar:

Q , sabit bir sayı,

L_k , k 'inci karıncanın tur uzunluğudur.

$t=0$ zamanında izin yoğunluğu ($\tau_{ij}(0)$) sıfıra yakın çok küçük bir değerdedir. Bir karıncanın bütün n farklı kasabaya uğramasını sağlamak için her bir karınca t zamanında önceden uğradığı kasabaları *tabu list* adı verilen bir veri deposuna kayıt eder. Tam bir tur yani n iterasyon tamamlanmadan önce karıncanın listedeki kasabalara tekrardan uğraması yasaklanır. Bir tur tamamlandıktan sonra tabu listesi boşaltılır ve karınca seçim için tekrardan bütün alternatifleri değerlendirebilir.

$Tabu_k$, k 'inci karıncanın tabu listesini içeren büyüme vektörünü tanımlarken, $Tabu_k(s)$, listenin s 'inci elementini gösterir. Burada $Tabu_k(s)$, k 'inci karıncanın ziyaret ettiği s 'inci kasabayı gösterir.

k 'inci karıncanın i kasabasından sonra j kasabasına gitme olasılığı (p_{ij}^k) yolun üzerindeki fenomen izi yoğunluğuna bağlı olduğundan bahsedilmişti. Bu olasılığın hesaplanmasında kullanılan matematiksel ifade (4.5)' de gösterilmiştir.

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (4.5)$$

Not : j ve l kasabaları tabu listesinin dışında olan kasabalardır.

Notasyonlar:

α ve β , görüş derecesine karşı izin nispi (görelî) önemini kontrol eden parametreler,

η_{ij} , görüş derecesidir.

Görüş derecesi (4.6) eşitliğinde görüldüğü gibi kenarın uzunluğu ile ters orantılıdır

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4.6)$$

(4.5) eşitliğinden de anlaşılacağı gibi geçiş olasılığı, görüş derecesi ve t zamanındaki iz

yoğunluğu arasında bir değişimdir. Yani, eğer (i,j) kenarı üzerinde çok fazla yoğunluk varsa (i,j) kenarının istenirliği yüksek derecededir.

Bu ifadeler ışığında algoritmanın biçimsel gösterimi şu şekilde yapılabilir.

1. Başlangıç

$t=0$,

$\{t: \text{zaman sayacı}\}$

Set $T=0$,

$\{T: \text{tur sayacı}\}$

Her (i,j) kenarı için iz yoğunluğu

$\tau_{ij}(t) = \tau_0$,

Karıncaların (i,j) kenarı üzerine yaydığı toplam iz miktarı

$\Delta\tau_{ij} = 0$,

n düğüm üzerine m adet karıncayı yerleştir.

2. Tabu listesinin başlangıcı

Set $s:=1$,

$\{s: \text{tabu listesi indeksi}\}$

For $k:=1$ to m do

$\text{tabu}_k(s)$ de k 'inci karıncanın başlama kasabasını tabu listesinin ilk elemanı olarak kaydet.

3. Bir turun tamamlanması

Set $s:=s+1$,

For $k:=1$ to m do

(4.5)' e göre kasaba j' yi seç,

k' inci karıncayı kasaba j' ye taşı,

$\text{tabu}_k(s)$ ' ye kasaba j' yi ekle,

k 'inci karınca tarafından tanımlanan turun L_k uzunluğunu hesapla,

bulunan en kısa turu güncelle,

Elde elden bilgiler doğrultusunda, her kenar (i, j) için,

For $k := 1$ to m do

Denklem (4.4)' i uygulayarak p_{ij}^k olasılıklarını hesapla,

Her (i,j) kenarı üzerindeki toplam fenomen miktarını güncelle

$$\Delta\tau_{ij} := \Delta\tau_{ij} + \Delta\tau_{ij}^k,$$

4. Tur sonunda her (i,j) kenarı üzerindeki fenomen miktarını güncelle

Her (i, j) kenarı için,

$$\Delta\tau_{ij}(t+n) := \rho \cdot \Delta\tau_{ij}(t) + \Delta\tau_{ij} \text{ denklemine göre } \Delta\tau_{ij}(t+n) \text{ ' leri hesapla,}$$

$$t := t+n,$$

$$T := T+1,$$

Her (i, j) kenarı için set $\Delta\tau_{ij} := 0$

5. If $(T < T_{\text{maksimum}})$ ve (durgunlaşma davranışı yoksa) then

Bütün tabu listesini boşalt,

Adım 3' e git,

Else

En kısa yolu yazdır,

Dur.

- Başlangıç

Kenarlar üzerindeki iz yoğunluğu için ilk değer ($\tau_{ij} = \tau_0$) belirlenir ve karıncalar farklı kasabalar üzerine yerleştirilir.

- Tabu listesinin başlangıcı

Başlangıçta bütün karıncaların tabu listesi boştur. Her bir karıncanın tabu listesinin ilk elemanı başlama kasabasını içerir.

-Bir turun tamamlanması

Bu aşamada her karınca i kasabasından istenilen iki ölçümün bir fonksiyonu olan (p_{ij}^k) olasılık ile belirlenen j kasabasına gider. Burada,

$\tau_{ij}(t)$ izi, aynı (i,j) kenarını kaç karıncanın seçtiği hakkında bilgi verirken,

η_{ij} görüş derecesi, bir kasaba ne kadar yakınsa o kadar istenilebilirliğinin ölçüsünü gösterir. Her bir k karınca için L_k nın değeri hesaplanır. Bunların yanında çeşitli başlama noktalarıyla stokastik bir algoritma elde edilir.

-Tur sonunda her (i,j) kenarı üzerindeki fenomen miktarını güncelle

n iterasyondan sonra bütün karıncaların bir tur tamamlamasıyla tabu listeleri dolacaktır. Bu noktada ve $\Delta\tau_{ij}^k$ değeri (4.2)' deki eşitliğe göre güncellenir. Aynı zamanda karıncalar tarafından bulunan en kısa yol kaydedilir ve bütün tabu listeleri boşaltılır.

Her bir seçim noktasında, alternatif tercihler arasından seçimin fenomen izi yoğunluğuna bağlı olması durgunlaşma davranışına sebep olabilir. (4.5)' deki eşitlik tarafından belirlenen seçim olasılığı nedeniyle, bir karınca en yüksek olasılıktaki kasabayı tercih edecektir. Bu seçim sonunda seçilen kasabanın güçlendirilmesi yapılacaktır. Bunun sonucunda karınca defalarca aynı çözümü bulacaktır.

Açıkçası, böyle bir durgunlaşma durumu kaçınılmazdır. Bunu önlemenin bir yolu direkt olarak fenomen izine bağlı olan seçim olasılığını etkilemektir. Bu fenomen izini sınırlamakla sağlanabilir (Stützle ve Dorigo, 1999).

Bütün fenomen izleri için,

$$\tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max} \quad (4.7)$$

sınırlaması getirilmesi durgunlaşmayı önleyici bir etki yapacaktır.

Bu durumun sağlanması için birtakım varsayımların sağlanması gerekir. Bunlar,

1. Eğer $\tau_{ij}(t) < \tau_{\min}$ ise $\tau_{ij}(t) = \tau_{\min}$ ayarlamasını yap,
2. Eğer $\tau_{ij}(t) > \tau_{\max}$ ise $\tau_{ij}(t) = \tau_{\max}$ ayarlamasını yap,
3. Her zaman için $\tau_{\min} > 0$ ve $\eta_{ij} < \infty$ dur.

- If $(T < T_{\text{maksimum}})$ ve (durgunlaşma davranışı yoksa) then

Bu süreç, tur sayacı turların maksimum sayısına (T_{maksimum}) ulaşınca kadar veya aynı turu bütün karıncalar yapana kadar devam eder. Aynı turu bütün karıncaların yapması durumunda durgunlaşma davranışı gözlenir. Bu durumda alternatif çözümler için algoritmanın aramayı

durdurması gerekir.

Dörigo vd. (1996), karınca algoritmasında izdeki artışın hesaplanması açısından farklı yaklaşımlar ortaya koyarak iki yeni algoritma türetmişlerdir. Mevcut algoritmaya *karınca-turu* algoritması ismini verirken yeni türetilen algoritmalara *karınca-yoğunluğu* ve *karınca-niceliği* olarak adlandırmışlardır. Bu iki modelde de her bir karınca, her bir adımda turun sonunu beklemeksizin izlerini yayar.

4.2.1 Karınca-Yoğunluğu Modeli

Bu modelde (i,j) kenarı üzerinde i' den j' ye giden bir karınca Q miktar iz bırakır. Bu nedenle karınca yoğunluğu modelinde (i,j) kenarı üzerindeki iz yoğunluğu (4.8) kullanılarak hesaplanır.

$$\Delta\tau_{ij}^k = \begin{cases} Q & \text{karınca (i, j) yolunu kullanırsa,} \\ 0 & \text{aksi taktirde,} \end{cases} \quad (4.8)$$

Formülden de anlaşılacağı gibi (i,j) kenarı üzerindeki iz yoğunluğu hesaplanırken dij' den bağımsız hareket edilir.

4.2.2 Karınca-Niceliği Modeli

Karınca-niceliği modelinde i' den j' ye gidilirken her zaman i' den j' ye giden karınca (i,j) kenarı üzerinde Q/d_{ij} miktar iz bırakır. Bu nedenle de karınca niceliği modelinde (i,j) kenarı üzerindeki iz yoğunluğu hesaplanırken,

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}} & \text{karınca (i, j) yolunu kullanırsa,} \\ 0 & \text{aksi taktirde,} \end{cases} \quad (4.9)$$

formülü kullanılır.

Bu modelde, daha kısa yolların karıncalar tarafından daha çok isteneceği görülmektedir.

4.2.3 Deneysel Çalışmalar

Sezgisel metotlar bir çok değişken parametre içerirler. Bu parametrelerin optimum değerlerinin belirlenmesi, diğerleri sabitken bir veya birkaçının değiştirilmesi sonucu daha iyi sonuçların araştırılması için deneysel çalışmaların yapılması gerekir. Bunun yanında kurulu olan mevcut modeldeki yapısal küçük değişimlerde daha iyi sonuçlar verebilir. Bunun araştırılması içinde yine deneysel çalışmaların yapılması gerekir.

Colorni vd. (1996) karınca algoritmasında türetilen üç algoritmanın (*karınca turu*, *karınca-yoğunluğu*, *karınca-niceliği*) deneysel çalışmalarını yaparak güçlü ve zayıf ilişkilerini araştırmışlardır. Deneysel çalışması yapılacak parametreler;

- α : İzin önem ilişkisi, $\alpha \geq 0$;
- β : Görünürlüğün önem ilişkisi, $\beta \geq 0$;
- p : İzin buharlaşmadan kalan oranı, $0 \leq p < 1$ ($1-p$ iz buharlaşması gibi yorumlanabilir);
- Q : Karıncalar tarafından yayılan izin miktarı ile ilişkili bir sabit

Bu parametreler içinde yer almayan karınca sayısı (m) problemin boyutuna (n) eşit kabul edilmiştir.

Her bir değişken parametre için 5 adet ölçü düzeyi tespit edilmiştir. Bunlar,

α için $\{0, 0.5, 1, 2, 5\}$,

β için $\{0, 1, 2, 3, 5\}$,

p için $\{0.3, 0.5, 0.7, 0.9, 0.999\}$,

ve Q için $\{1, 10, 100, 1000, 10000\}$ 'dir.

Deneysel çalışmanın yapıldığı bütün problemler üzerinde 4 değişken parametrenin 5 ölçü düzeyi için, diğerleri sabitken bir parametrenin değiştirilmesiyle elde edilen (5^4)' den 625 adet deney yapılmıştır. İlk sonuçlar küçük ölçekli problemlerde bulunurken, (Colorni vd., 1992; Dorigo, 1992; Dorigo vd., 1991) diğer bütün testler Whitley vd. (1989) tarafından yapılan çalışmada tanımlanan 30 şehirli bir problem olan OLIVER30 problemi üzerine denenmiştir.

Üç modeli karşılaştırmak için başta deneysel olarak her bir algoritma için parametrelerin en iyi değerleri belirlenmiş ve daha sonra en iyi parametreler kullanılarak her bir algoritma on

kere çalıştırılmıştır. Sonuçlar Çizelge (4.1)' de gösterilmiştir.

Çizelge 4.1 Üç farklı model için parameter ayarlaması (Dorigo vd., 1992)

	En iyi parameter ayarı	Ortalama sonuç	En iyi sonuç
<i>Karınca-Yoğunluğu</i>	$\alpha=1$ $\beta=5$ $p=0.99$	426.740	424.635
<i>Karınca-Niceliği</i>	$\alpha=1$ $\beta=5$ $p=0.99$	427.315	426.255
<i>Karınca-Turu</i>	$\alpha=1$ $\beta=5$ $p=0.5$	424.250	423.741

Deneysel çalışmalar sonucu Q parametresinin etkisi önemsiz bulunduğu için gösterilmemiştir. Hem *karınca-yoğunluğu* hem de *karınca-niceliği* modelleri *karınca-turu* modeli ile bulunanlardan daha kötü sonuçlar vermiştir. Bunun nedeni arama prosesinde direkt olarak kullanılan geri besleme bilgisinin çeşidinden kaynaklanır. *Karınca-turu* modeli global bilgi kullanır. Bu bilgi, üretilen çözümün iyi olmasının karıncaların yaydığı izin miktarıyla orantılı olmasındandır. Diğer yandan, hem *karınca-yoğunluğu* hem de *karınca-niceliği* yerel bilgi kullanır. Aramaları son neticeye ulaşmanın bazı ölçüleri tarafından yönetilmez. Bu nedenle, daha kötü performans sonuçları vermesi sürpriz değildir.

4.2.4 Algoritmanın Performansını Etkileyen Faktörler

Algoritmanın performansını etkileyen faktörler; karınca sayısına bağlı olan sinerjik etkiler, karıncaların başlama noktası ve problem boyutudur.

4.2.4.1 Sinerjik Etkiler

Çözüm sürecinin etkinliği üzerinde, hem karınca sayısının etkisini hem de iz yoluyla haberleşmenin önemini belirlemek için deneyler seti çalıştırılmıştır. Bu deneysel çalışma için boyutu 4 olan bir problem kullanılmıştır. İz yoluyla haberleşmenin öneminin tespiti için problem $\alpha=0$ ve $\alpha=1$ değerleri için çalıştırılmıştır. Karınca sayısının etkisi için ise farklı düzeylerde alınan karınca sayıları kullanılmıştır. Sonuçta, en iyi turun bulunmasında karıncalar arasında bir haberleşme mekanizmasının olmasının önemi tespit edilmiştir. Bunun yanında karınca sayısının problem boyutuna yakın değerleri iyi sonuçlar vermiştir ($m \approx n$).

4.2.4.2 Başlangıç Noktası

Bu deney ilk karınca dağılımının karınca sistemi performansını etkileyip etkilemeyeceğini incelemek için çalıştırılmıştır. Bütün karıncaların $t=0$ zamanında aynı şehirde olduğu durum ve düzenli olarak dağıtıldıkları durum arasında fark olup olmadığını 30 boyutlu Oliver30 problemiyle test edilmiştir. Çalışmalar sonucunda karıncaları bir noktadan başlatmanın performansı kötü yönde etkileyeceği, bu nedenle başlangıçta karıncaların dağıtılması gerektiği tespit edilmiştir.

4.2.4.3 Problem Boyutu

Bu deney ise, problem boyutunun algoritma performansına etkisinin araştırılması için yapılmıştır. Benzer problemlerin boyutların artırılmasıyla elde edilen (kenar uzunluğu seti 10 için $r*r$ şebekesi) problemlerin sonuçları Çizelge 4.2' de verilmiştir.

Çizelge 4.2 Problem boyutunun fonksiyonu olarak optimumu bulmak için gereken zaman (Dorigo vd., 1992)

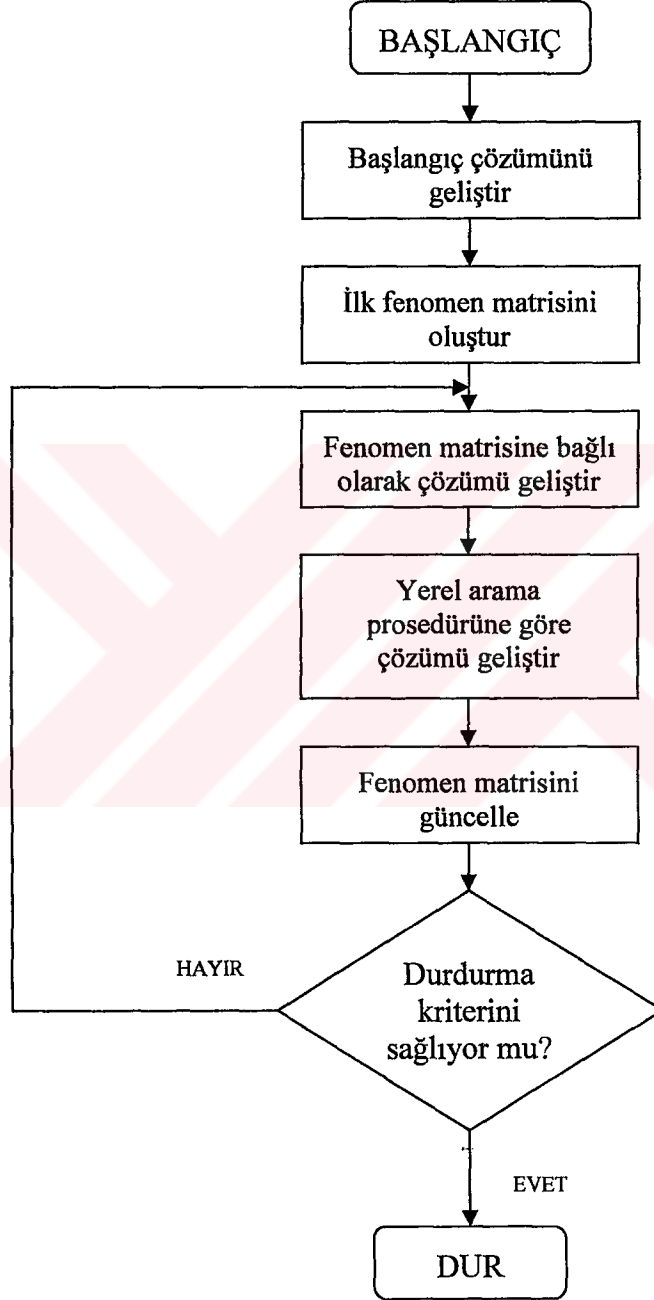
Problem	Boyut	Optimum çözüm	En iyi çözüm	Optimumu bulmak için çevrimlerin ortalama sayısı	Optimumu bulmak için gereken zaman (saniye)
4*4	16	160	160	5.6	8
5*5	25	254.1	254.1	13.6	75
6*6	36	360	360	60	1020
7*7	49	494.1	494.1	320	13440
8*8	64	640	640	970	97000

Çizelge 4.2'de de görüldüğü gibi problem boyutunu büyümesine rağmen algoritma optimum çözümleri bulabilmektedir.

4.3 Karınca Algoritması Uygulamaları

İlk kez Dorigo (1992) tarafından doktora tezi çalışmalarında geliştirilen KA günümüzü kadar bir çok kombinatoriyal optimizasyon problemlerinin çözümü için denenmiş ve başarılı sonuçlar elde edilmiştir.

KA'nın özellikle son yapılan çalışmalarında, iyi sonuçların elde edilmesi için tırmanıcı etki yapan yerel arama prosedürleri kullanılmıştır. Yerel arama prosedürleri bazen araştırmacılar tarafından geliştirilirken bazen de bilinen zeki sezgiseller bu amaç için kullanılmıştır. Algoritmanın genel akış şeması Şekil 4.3'deki gibi verilebilir.



Şekil 4.3 Karınca algoritmasının genel akış şeması

KA'nın uygulandığı optimizasyon problemleri;

- Simetrik ve asimetrik GSP (Dorigo vd., 1996, 1997, 1999),
- Ardışık sıralama problemi (Gambardella ve Dorigo, 1997),
- KAP (Maniezzo vd., 1994; Taillard ve Gambardella, 1997; Gambardella vd., 1997; Stützle ve Dorigo, 1999),
- Araç sıralama problemi (Bullnheimer vd., 1999),
- Atölye tipi iş çizelgeleme problemi (Forsyth ve Wren, 1997; T'kindt vd., 2001),
- Bölme problemi (Kuntz vd., 1997),
- Telekomünikasyon ağı problemi (Schoonderwoerd vd., 1997; Di Caro ve Dorigo, 1998; Navarro ve Sinclair, 1999),
- Klavye düzenleme problemi (Eggers vd., 2003),
- Silah-Hedef atama problemi (Lee vd., 2002),
- Çok amaçlı JIT çizelgeleme problemi (McMullen, 2001),
- Esnek İmalat Sistemlerinde (EİS) tek sıralı tesis yerleşimi problemi (Solimanpur, 2003),
- Paralel uygulamalar (Bullnheimer vd., 1998; Randall ve Lewis, 2002),

olarak gösterilebilir.

5. TESİS YERLEŞİMİ PROBLEMLERİ İÇİN KARINCA ALGORİTMASI

Tesis yerleşimi problemleri için karınca algoritması ilk olarak Maniezzo vd. (1994) tarafından çalışılmıştır. Bu çalışmada geliştirilen algoritma eşit alanlı tesisler için kareli atama problemi üzerinde denenmiş ve küçük boyutlu problemler için iyi sonuçlar elde edilmiştir. Gambardella vd. (1997), karınca algoritmasına kendilerinin geliştirdiği yerel arama prosedürünü ekleyerek Melez Karınca Algoritmasını (MKA) oluşturmuşlar ve yine kareli atama problemi için deneysel çalışmalar yapmışlardır. MKA, küçük boyutlu problemler için KA'dan daha iyi sonuçlar vermesinin yanında büyük boyutlu problemler için de yüksek performans göstermiştir. Kareli atama problemi için KA'nın son uygulamasını Talbi vd. (2001) KAP için KA'nın paralel uygulamasını geliştirerek yapmışlardır. Bu algoritmanın paralel olmasının dışında önemli bir farklılığı da yerel arama prosedürü olarak TA'nın kullanılmasıdır.

KAP dışında karınca algoritmasının tesis yerleşimi uygulaması, Solimanpur (2003) tarafından EİS'de tek sıralı tesis yerleşimi problemi için çalışılmıştır.

5.1 Kareli Atama Problemi için Karınca Algoritması Uygulamaları

Bütün departmanlar eşit alanlıysa veya departmanların fiziksel yerleri, diğer kalan departmanlarla arasındaki yakınlık veya uzaklık ilişkilerini değiştirmeden değiştirilebiliyorsa departmanların işgal edeceği potansiyel yerleri belirlemek kolaydır. Böyle bir durumda, yerleşim problemi, maliyetin akış miktarı ve kat edilen uzaklık bakımından doğrusal olduğu farz edilen Kareli Atama Problemi (KAP) olarak modellenebilir (Tate ve Smith, 1994).

5.1.1 Problemin Tanımlanması

KAP daha önceden de tanımlandığı gibi n tesisin n yerleşime atanmasıdır. İlk kez Koopmans ve Beckman (1957) tarafından formülize edilen KAP'nin uygulamaları arasında üniversite kampüsündeki binaların planlanması, hastanelerde departmanların düzenlenmesi, elektronik devrelerde toplam tel uzunluğunun minimizasyonu örnek olarak gösterilebilir (Burkard, 1984).

KAP'nin KA uygulamasında problemin matematiksel ifadesi için $n \times n$ boyutlu üç matris tanımlanır.

$D = [d_{ij}]$, i yerleşimi ile j yerleşimi arasındaki mesafeler matrisi,

$F = [f_{ij}]$, i tesis ile j tesisi arasındaki akışların matrisi,

$C = [c_{ij}]$, i yerleşimine j tesisinin atama maliyeti,

Bu matrislerden D ve F tamsayı değerli simetrik matrislerdir. C matrisi ise genellikle 1 olarak kabul edilir.

$\Pi : i \rightarrow \pi(i)$, i yerleşimine $j = \pi(i)$ tesisinin atanması olarak yorumlanabilir. ($i = 1, 2, 3, \dots, n$)

İki tesis arasında veri transferinin maliyeti yerleşimler arasındaki uzaklıkla, yerleşimlere atanan tesisler arasındaki akış miktarının çarpımına eşittir.

KAP'yi çözmek için toplam atama maliyetinin Π permütasyon için bulunması gerekir. Bu nedenle KAP'nin matematiksel formülü (5.1)'deki gibidir.

$$\min z = \sum_{i,j=1}^n d_{ij} \cdot f_{\pi(i)\pi(j)} \quad (5.1)$$

Amaç fonksiyonunun karesel özelliğini göstermek için formül yeniden düzenlenirse;

$$\min z = \sum_{i,h=1}^n \sum_{j,k=1}^n d_{ij} \cdot f_{hk} \cdot x_{ih} \cdot x_{jk} \quad (5.2)$$

şeklinde gösterilir. Burada;

$$x_{ij} = \begin{cases} 1, & i \text{ tesisi } j \text{ ye atandı ise,} \\ 0, & \text{aksi takdirde,} \end{cases}, \quad x_{ij} \in \{0,1\} \forall_{i,j} \text{ ' dir.}$$

5.1.2 Kareli Atama Problemi için Karınca Algoritması

Gerçek karıncaların davranışlarından esinlenerek geliştirilen KAP'nin KA'ya uygulamasında yapay karıncalar sistemi tanımlanır. Bu karıncalar;

1. i yerleşimine j tesisi atanırken (i,j) çiftine fenomen yayar,
2. i yerleşimine j tesisi atanırken olasılıklar göz önünde bulundurulur,
3. Bütün faaliyetler atanana kadar atanan yerleşim ve tesisler bir listede tutulur, yani $n \times n$ 'lik bir problemde her tesis sadece bir yerleşime atanır.

Literatürde KAP için geliştirilen KA ve türevlerinden ilki Maniezzo vd. (1994) tarafından

geliştirilmiştir. Tesislerin yerleşimlere atanması diğer algoritmalara göre farklılık göstermektedir. Bu farklılıklar;

1. Her bir karınca için tabu listelerinin oluşturulması,
2. Mesafe potansiyelleri ile akış potansiyelleri vektörlerinin ve birleştirme matrisinin oluşturulması,
3. i yerleşimine j tesisinin atanmasında olasılıkların kullanılmasıdır.

5.1.2.1 Her Karınca için Tabu Listelerinin Oluşturulması

i yerleşimine j tesisi atandı ise (i,j) çifti $\text{tabu}_k(s)$ listesine alınır. n boyutlu problemde $\text{tabu}_k(n)$ olana kadar permütasyon devam eder. Listenin eleman sayısı problemin boyutuna eşit olduğunda liste boşaltılır.

$\text{tabu}_k(s)$, k 'ıncı karıncanın s . atama işlemini göstermektedir.

5.1.2.2 Mesafe Potansiyelleri ve Akış Potansiyelleri Vektörleri

Bir atamada “potansiyel fayda” değerinin hesaplanması için mesafe ve akış potansiyelleri vektörlerinin hesaplanması gerekir.

D , mesafelerin matrisini, F ise akışların matrisini gösterirse,

D , mesafe potansiyelleri vektörünü, F ise akış potansiyelleri vektörünü gösterebilir.

D ve F vektörleri D ve F matrislerinin satır değerleri toplamına eşittir. D vektörü, bir düğüm ile diğerleri arasındaki mesafelerin toplamını gösterirken, F vektörü, bir tesis ile diğerleri arasındaki toplam akışı göstermektedir.

i düğümünün en küçük mesafe potansiyeli (d_i) ağdaki en önemli yerleşimi gösterirken, j tesisinin en büyük akış potansiyeli (f_j) en önemli tesisi göstermektedir.

Boyutu 6 olan bir problem (Nugent vd.,1968) için düşünülürse, D ve F matrisleri,

$$D = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 & 1 & 2 \\ 2 & 1 & 0 & 3 & 2 & 1 \\ 1 & 2 & 3 & 0 & 1 & 2 \\ 2 & 1 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 2 & 1 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 & 5 & 2 & 4 & 1 & 0 \\ 5 & 0 & 3 & 0 & 2 & 2 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 5 & 2 \\ 1 & 2 & 0 & 5 & 0 & 9 \\ 0 & 2 & 0 & 2 & 9 & 0 \end{bmatrix}$$

ise, D ve F potansiyeller vektörleri,

$$D = \begin{bmatrix} 9 \\ 7 \\ 9 \\ 9 \\ 7 \\ 9 \end{bmatrix} \quad F = \begin{bmatrix} 12 \\ 12 \\ 5 \\ 11 \\ 17 \\ 13 \end{bmatrix}$$

olarak belirlenir.

Bu problemde, yerleşim 2 ve 5 merkezi pozisyona yerleştirilmelidir. Diğerleri ile en çok ilişkide olan tesis 5 bu merkezi yerleşimlere yerleştirilmelidir. Bu düşünceden de anlaşılacağı gibi iyi bir permütasyon “minimum-maksimum” ikilisiyle elde edilir.

Bundan sonra yapılacak işlem atamaların belirlenmesi için birleştirme matrisi A 'nın elde edilmesidir.

$A = D \cdot F^T$ formülü ile bulunurken,

$a_{ij} = d_i \cdot f_j$ ile belirlenir. Örneğimiz için birleştirme matrisi,

$$A = \begin{bmatrix} 108 & 108 & 45 & 99 & 63 & 117 \\ 84 & 84 & 35 & 77 & 49 & 117 \\ 108 & 108 & 45 & 99 & 63 & 117 \\ 108 & 108 & 45 & 99 & 63 & 117 \\ 84 & 84 & 35 & 77 & 49 & 91 \\ 108 & 108 & 45 & 99 & 63 & 117 \end{bmatrix}, \text{ dir.}$$

Çözümün yapılandırılmasında, A birleştirme matrisinin tek tek bütün sütunları ele alınır. “Minimum-maksimum” kuralına göre en kısa mesafede en çok akışı sağlayan atamayı belirlemek için sütunların minimum değerleri seçilir. Seçilen değerin bulunduğu satır ve sütunlar kaldırılır. Yeni oluşan duruma göre ilgili sütunun minimum değeri seçilir bu atama mekanizması bütün yerleşimlere tesisler atanıncaya kadar devam eder.

$$A = \begin{bmatrix} 108 & 108 & 45 & 99 & 63 & \underline{117} \\ \underline{84} & 84 & 35 & 77 & 49 & 117 \\ 108 & 108 & 45 & 99 & \underline{63} & 117 \\ 108 & 108 & \underline{45} & 99 & 63 & 117 \\ 84 & \underline{84} & 35 & 77 & 49 & 91 \\ 108 & 108 & 45 & \underline{99} & 63 & 117 \end{bmatrix}$$

A matrisinden de anlaşılacağı gibi, 1. tesis 2. yerleşime, 2. tesis 5. yerleşime, 3. tesis 4. yerleşime, 4. tesis 6. yerleşime, 5. tesis 3. yerleşime, 6. tesis 1. yerleşime atanır. Bütün tesisler atandığında permütasyon tamamlanır.

Permütasyon (2,5,4,6,3,1)'in maliyeti,

$$a_{12} + a_{25} + a_{34} + a_{46} + a_{53} + a_{61} = 492 \text{ dir.}$$

5.1.2.3 Atama Olasılıklarının Kullanımı

KA'da permütasyon olasılıkla yapılandırılır. m karıncalı bir popülasyonda i yerleşimine j tesisinin atanması için $\tau_{ij}(t+1)$ fenomen iz yoğunluğu hesaplanır.

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (5.3)$$

Notasyonlar:

$\tau_{ij}(t+1)$, t+1 zamanda i yerleşimine j tesisinin atanmasındaki fenomen iz yoğunluğu,

ρ , buharlaşmadan kalan fenomen izinin yüzdesini gösteren parametre, ($0 \leq \rho \leq 1$),

$\tau_{ij}(t)$, t zamanda i yerleşimine j tesisinin atanmasındaki fenomen iz yoğunluğu,

$\Delta \tau_{ij}$, i yerleşimine j tesisinin atanmasındaki toplam fenomen iz değişimidir ve (5.4)'e göre hesaplanır.

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (5.4)$$

Notasyonlar:

$\Delta \tau_{ij}^k$, k'ıncı karıncanın i yerleşimine j tesisinin atanmasını yapması durumunda fenomen iz miktarına yapacağı değişim miktarıdır ve (5.5)'deki eşitlikle hesaplanır.

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{karınca } (i, j) \text{ yolunu kullanırsa,} \\ 0 & \text{aksi takdirde,} \end{cases} \quad (5.5)$$

Notasyonlar:

Q , sabit bir sayı,

L_k , atama sonucunda k 'inci karıncanın elde ettiği amaç fonksiyonu değeridir.

k 'inci karıncanın i yerleşimine j tesisini atama olasılığı (p_{ij}^k),

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{r \notin tabu_k} [\tau_{ir}(t)]^\alpha [\eta_{ir}]^\beta} \quad (5.6)$$

formülü ile hesaplanır.

Not : j ve r tesisleri tabu listesinin dışında olan tesislerdir.

Notasyonlar:

α ve β , görüş derecesine karşı izin nispi (görelî) önemini kontrol eden parametreler,

η_{ij} , görüş derecesidir.

Görüş derecesi (5.7) eşitliğinde görüldüğü gibi birleştirme matrisi elamanları (a_{ij}) ile ters orantılıdır.

$$\eta_{ij} = \frac{1}{a_{ij}} \quad (5.7)$$

Verilen bu bilgiler ışığında, KAP için KA'nın algoritmik ifadesi şu şekilde ifade edilebilir.

1. $t := 0$

Fenomen iz matrisinin başlangıç değerleri,

- Mesafe ve akış potansiyelleri ile A birleşme matrisi değerlerini hesapla,

- i yerleşimine j tesisinin atanmasının istenilebilirliğini hesapla (5.7),

2. For $k := 1$ to m

Repeat

(5.6)'a göre atamaları yap,

Until (Tabu listesi dolana kadar) {ilgili karınca bütün tesisleri atayana kadar}

End,

For $k:=1$ to m

L_k değerlerini hesapla,

En iyi permütasyonu güncelle,

End,

3. Her bir birleşme (i,j) için $\Delta\tau_{ij}$ 'i hesapla,

Fenomen matrisini (5.3)' e göre güncelle,

4. If (durdurma kriteri sağlanmıyorsa)

Her bir birleşme (i,j) için $\Delta\tau_{ij} = 0$ yap,

Bütün karıncaların tabu listesini boşalt,

Adım 2' ye git,

Else

DURDUR ve en iyi permütasyonu yazdır.

Çizelge 5.1 Parametrelerin ölçüm değerleri

Parametre	Belirlenen ölçüm değeri
ρ (iz ısrarının katsayısı)	0,9
α (izin önemi)	1
β (potansiyellere bağlı istenebilirliğin önemi)	1
Q (her bir iterasyonda iz bırakmanın önemi)	10
m (karınca sayısı)	n (problem boyutu)

Algoritmanın performansı ρ (iz ısrarının katsayısı), α (izin önemi), β (potansiyellere bağlı istenebilirliğin önemi), Q (her bir iterasyonda iz bırakmanın önemi) ve m (karınca sayısı) değişken parametrelerine bağlıdır. Bu parametrelerin değerlerinin hesaplanması için 5, 7, 12,

15, 19, 20 ve boyutlu bir problem seti kullanılarak deney tasarımı yapılmıştır. Algoritmayı geliştiren Maniezzo vd. (1994) tarafından yapılan deney tasarımında Çizelge 5.1'deki değerler elde edilmiştir (Maniezzo vd., 1994).

5.1.3 Kareli Atama Problemi için Melez Karınca Algoritması

KAP'nin çözümü için Maniezzo vd. (1994) tarafından geliştirilen KA'nın büyük boyutlu problemler için iyi sonuçlar vermemesi sonucu Gambardella vd. (1997), mevcut karınca algoritmasına kendilerinin geliştirdiği yerel arama prosedürü eklemişler ve fenomen izine bağlı çözüm geliştirme yapısında değişiklikler yapmışlardır. Melez Karınca Algoritması (MKA) olarak literatüre geçen bu yeni algoritma küçük boyutlu problemlerin yanında büyük boyutlu problemler için de iyi sonuçlar vermiştir.

MKA'nın en basit algoritmik ifadesi şu şekilde gösterilebilir.

m karınca için başlangıç çözümü ata,

İlk fenomen matrisini oluştur,

For $I_{maksimum}$ iterasyon tekrarla

For her bir karınca $k = 1, 2, 3, \dots, m$ do

Fenomen izi matrisinin kullanarak k 'inci karıncanın çözümünü geliştir,

Lokal arama prosedürünü kullanarak k 'inci karıncanın çözümünü geliştir,

k 'inci karınca için yoğunlaşma mekanizması ile yeni bir başlangıç çözümü ata,

End,

Fenomen matrisini güncelle,

Eğer gelişme yoksa çeşitlendirme mekanizmasını uygula,

End.

5.1.3.1 Çözümlerin Başlangıcı

Başlangıçta m adet karıncaya rassal olarak seçilen bir permütasyon verilir. Bu permütasyonlar başlangıçta, daha sonra detaylı olarak tanımlanacak olan yerel arama prosedürü kullanımıyla

optimize edilir.

5.1.3.2 Fenomen İzi Matrisi Başlangıcı

Başlangıçta fenomen izi matrisi hiç bilgi içermez. Bütün τ_{ij} fenomen izleri aynı τ_0 değerine atanır. Bu değer hesaplanmasında m adet karıncaya önce rassal olarak atanıp daha sonrada yerel arama prosedüründe geliştirilen permütasyonlar içinden en iyi permütasyon seçilir ve fenomen izi matrisinin başlangıç değeri (5.8) eşitliğinden faydalanılarak belirlenir

$$\tau_0 = \frac{1}{Q(f(\pi^*))} \quad (5.8)$$

Notasyonlar:

τ_0 , fenomen izleri başlangıç değeri,

$f(\pi^*)$, o ana kadar elde edilen en iyi çözüm,

Q , değişken bir parametredir.

5.1.3.3 Çözümlerin Geliştirilmesi

KAP problemi için MKA'nın, bilinen KA'dan en önemli farkı çözümlerin geliştirilmesi aşamasıdır. Bu aşama iki kısımdan oluşmaktadır.

1. Fenomen izini bağlı gelişme,
2. Yerel arama prosedürüne bağlı gelişme.

5.1.3.3.1 Fenomen İz Tabanlı Gelişme

Fenomen iz tabanlı gelişme, her bir karınca için kendi permütasyonu π^k 'ya T (takas sayısı) kadar uygulanır. Gelişme prosesi en basit anlatımı ile iki parametrenin (r,s) belirlenip takası ile olur. Öncelikle $(r = 1, 2, 3, \dots, n)$ olmak üzere rassal olarak r parametresini seçilir. $s \neq r$ koşulunu sağlayan s parametresi, rassal olarak seçilen iki strateji ile seçilir.

Strateji 1 (exploiting)

q olasılıkla (5.9)'u maksimum yapan s parametresi belirlenir.

$$\tau_{r\pi_s}^k + \tau_{s\pi_r}^k \quad (5.9)$$

Strateji 2 (exploring)

(1-q) olasılığı ile (5.10)'u maksimum yapan s parametresi belirlenir.

$$\sum_{j \neq r} \frac{\tau_{r\pi_s}^k + \tau_{s\pi_r}^k}{(\tau_{r\pi_j}^k + \tau_{j\pi_r}^k)} \quad (5.10)$$

r ve s parametreleri belirlendikten sonra π^k çözümünde π_r^k ve π_s^k değerleri yer değiştirir.

Elde edilen geliştirilmiş çözüm $\hat{\pi}^k$ ile simgelenir.

5.1.3.3.2 Yerel Arama Prosedürü ile Gelişme

Fenomen iz tabanlı gelişme ile elde edilen $\hat{\pi}^k$ çözümü yerel arama prosedüründe geliştirilerek $\pi^{\infty k}$ çözümü elde edilir. Yerel arama prosedürü şu şekilde gösterilebilir.

$I = \emptyset$

Repeat

rassal olarak i parametresini seç, (i = 1, 2, 3, ..., n)

$J = \{i\}$

Repeat

i parametresinden farklı j parametresini rassal olarak seç, (j = 1, 2, 3, ..., n)

If ($\Delta(\pi, i, j) < 0$, (5.11))

π çözümünde π_i ve π_j değerlerinin değiştir, $J = J \cup \{j\}$

Until ($|J| < n$)

$I = I \cup \{i\}$

Until ($|I| < n$)

Burada $\Delta(\pi, i, j)$ ifadesi (5.11)'deki eşitliğe göre belirlenir.

$$\Delta(\pi, i, j) = (a_{ii} - a_{jj})(b_{\pi_j \pi_j} - b_{\pi_i \pi_i}) + (a_{ij} - a_{ji})(b_{\pi_j \pi_i} - b_{\pi_i \pi_j}) + \sum_{k \neq i, j} (a_{ki} - a_{kj})(b_{\pi_k \pi_j} - b_{\pi_k \pi_i}) + (a_{ik} - a_{jk})(b_{\pi_j \pi_k} - b_{\pi_i \pi_k}) \quad (5.11)$$

Notasyonlar:

$\Delta(\pi, i, j)$, π çözümünde π_i ve π_j elementlerinin değişimi ile amaç fonksiyonu değerindeki değişim,

a_{ij} , akış matrisi elementi,

$b_{\pi_i \pi_j}$, mesafe matrisi elementi,

Bu prosedür sistematik olarak olanaklı bütün alternatifleri dener ve en uygun değişimi yapar. Yerel arama prosedüründe $i = j$ iken $\Delta(\pi, i, j)$ hesaplamaya gerek yoktur. Bu da algoritmanın hızını artıran bir yoldur.

5.1.3.4 Yoğunlaşma

Yoğunlaşma fonksiyonu daha çok iyi çözümlerin komşuluklarını araştırmaktır. Yoğunlaşma mekanizması o ana kadar yapılan aramada elde edilen en iyi çözüm üretilirken çalıştırılır.

Yoğunlaşma aktifken her bir karınca $\pi^{\infty k}$ ve π^k arasında en iyi permütasyon ile sonraki iterasyona başlar. Yoğunlaşma, bir iterasyon esnasında çözümü geliştirmede en az bir karınca başarıyla aktif kalır. Yoğunlaşmanın temel mantığı, yeni bulunan en iyi çözümün komşuluğunda aramaya devam etmektir. Yoğunlaşma mekanizmasına göre bir iterasyonda genel bir k karıncasının davranışı üç arama adımından oluşur. Bunlar; ilk çözüm (π^*), fenomen izine bağlı gelişme ($\hat{\pi}$) ve yerel aramaya bağlı gelişmedir (π^{∞}).

i 'inci iterasyonun başlangıcında, yoğunlaşmanın aktif olmadığı varsayılır. Bu nedenle,

$$\pi^k(i+1) \leftarrow \pi^{\infty k}(i) \text{ 'dir.}$$

$(i+1)$ 'inci iterasyonun sonunda yoğunlaşma aktif hale getirilir çünkü bulunan en iyi çözüm geliştirilmiştir. Bu nedenle,

$\pi^k(i+2) \leftarrow \pi^{\alpha^k}(i+1)$ 'dir.

(i+2)'nci iterasyonun sonunda yoğunlaşma aktiftir çünkü $\pi^k(i+2)$, $\pi^{\alpha^k}(i+1)$ 'den daha iyi sonuç vermiştir. Bu nedenle,

$\pi^k(i+3) \leftarrow \pi^k(i+2)$ ataması yapılır.

5.1.3.5 Fenomen İz Matrisinin Güncellenmesi

Fenomen izlerinin güncellenmesi standart karınca sisteminden farklı yapılıdır. Fenomen izlerinin güncellenmesi standart karınca sisteminde çok yavaş işlemektedir (Dorigo ve Gambardella, 1997). Hızlandırmak için, fenomen izleri üretilen en iyi çözümün hesaplanmasıyla güncellenir. Başta bütün fenomen izleri fenomenin buharlaşmasını simüle etmek için (5.12) kullanılarak azaltılır.

$$\tau_{ij} = (1 - \alpha_1) \tau_{ij} \quad 1 \leq i, j \leq n \quad \text{ve} \quad 0 < \alpha_1 < 1 \quad (5.12)$$

$0 < \alpha_1 < 1$ fenomen izlerinin buharlaşmasını kontrol eden bir parametredir. 0' a yakın α_1 değeri, uzun bir zaman aktif kalan fenomen izi anlamına gelirken, 1' e yakın α_1 değeri, buharlaşmanın yüksek seviyesi anlamına gelir. Daha sonra, fenomen izleri iterasyon sonunda, (5.13)' deki gibi elde edilen en iyi çözüm (π^*) hesaba katılmasıyla güçlendirilir.

$$\tau_{i\pi^*} = \tau_{i\pi^*} + \frac{\alpha_2}{f(\pi^*)} \quad (5.13)$$

Notasyonlar:

α_2 , tıpkı α_1 gibi fenomen izlerinin buharlaşmasını kontrol eden bir parametredir.

$f(\pi^*)$, elde edilen en iyi çözümün amaç fonksiyonu değeridir.

5.1.3.6 Çeşitlendirme Mekanizması

Çeşitlendirmenin amacı, farklı yapı ile çözümler üzerinde çalışmak için algoritmayı kısıtlamaktır. S iterasyon sonunda en iyi çözümde ilerleme bulunamamışsa çeşitlendirme mekanizması aktif hale gelir. Çeşitlendirme fenomen iz matrisinin yeniden başlaması sayesinde fenomen izlerinde tutulan bütün bilgileri silmeye ve bütün karıncalar için yeni

genel bir çözümü rassal olarak oluşturmaya bağlıdır. Sadece bütün karıncalar için o ana kadarki üretilen en iyi çözüm saklanır.

Bu bilgiler doğrultusunda KAP için MKA'nın en kapsamlı algoritmik ifadesi şu şekilde gösterilebilir (Gambardella vd, 1997).

m karıncanın her birisi için *m* adet rassal olarak başlangıç çözümü üret, $\pi^1(1), \dots, \pi^m(1)$,

Yerel arama prosedürü ile çözümleri geliştir, $\pi^1(1), \dots, \pi^m(1)$,

En iyi çözümü (π^*) seç,

Fenomen matrisini (*D*) bu çözüme göre oluştur,

For (*i* = 1 to I^{maksimum}) repeat $\{I^{\text{maksimum}}, \text{iterasyon sayısı}\}$

For (her bir karınca çözümü ($\pi^k(i)$)) do

$\pi^k(i)$ çözümünden $\hat{\pi}^k(i)$ çözümünü elde eden *T* adet takası uygula,

$\hat{\pi}^k(i)$ çözümünden $\pi^{\alpha k}(i)$ çözümünü elde eden yerel arama prosedürünü uygula,

End,

For (her bir karınca) do

If (yoğunlaşma aktif) then $\pi^k(i+1) \leftarrow \pi^k(i)$ ve $\pi^{\alpha k}(i)$ çözümleri arasındaki en iyi değer,

Else,

$\pi^k(i+1) \leftarrow \pi^{\alpha k}(i)$,

End,

If ($\pi^k(i+1) = \pi^k(i)$) then

Yoğunlaşmayı pasifleştir,

If ($\pi^{\alpha k}(i) < f(\pi^*)$) then

O ana kadar elde edilen en iyi çözümü (π^*) güncelle,

Yoğunlaşmayı aktifleştir,

Fenomen iz matrisini güncelle,

If (S iterasyon sonunda gelişme sağlanmıyor) then

Çeşitlendirme mekanizmasını çalıştır.

Algoritmada yer alan bir çok değişken parametrenin değerlerinin belirlenmesinde deney analizi yapılmıştır. Yapılan çalışma sonucunda değişken parametreler için Çizelge 5.2' deki değerler elde edilmiştir. Bu parametre değerleri kullanılarak algoritma , 20, 25, 26, 30, 35, 40, 50, 60 ve 80 boyutlu problemler (QAP Library) için denenmiş ve başarılı sonuçlar elde edilmiştir.

Çizelge 5.2 Parametrelerin ölçüm değerleri

Parametre	Belirlenen ölçüm değeri
T (takas sayısı)	$n/3$ (n, problemin boyutu)
α_1 ve α_2 (izin buharlaşmadan kalan kısmı)	0,1
S (çeşitlendirme kararının verileceği iterasyon)	$n/2$
q (1. takas stratejisinin belirlenme olasılığı)	%90
m (karınca sayısı)	10
Q (her bir iterasyonda iz bırakmanın önemi)	100

5.1.4 Kareli Atama Problemi için Paralel Uygulama

Talbi vd. (2001) KAP'nin çözümü için, paralel karınca algoritmasını (PKA) uygulamışlardır. Gambardella vd. (1997) tarafından geliştirilen MKA'yı temel alan çalışmalarında, yerel arama prosedürü, fenomen izi matrisinin güncellenmesi ve çeşitlendirme aşamalarında bazı değişiklikler yapmışlardır. Bu algoritmanın MKA'dan en önemli farkı paralel yapısıdır. PKA, MKA ile aynı problemler üzerinde denenmiş ve çok daha iyi sonuçlar elde edilmiştir (Talbi vd., 2001).

PKA'nın algoritmik yapısı MKA'nın algoritmik yapısına benzer olması sebebiyle detaylı bir biçimde bahsedilmeyecek sadece farklılıkları üzerinde durulacaktır.

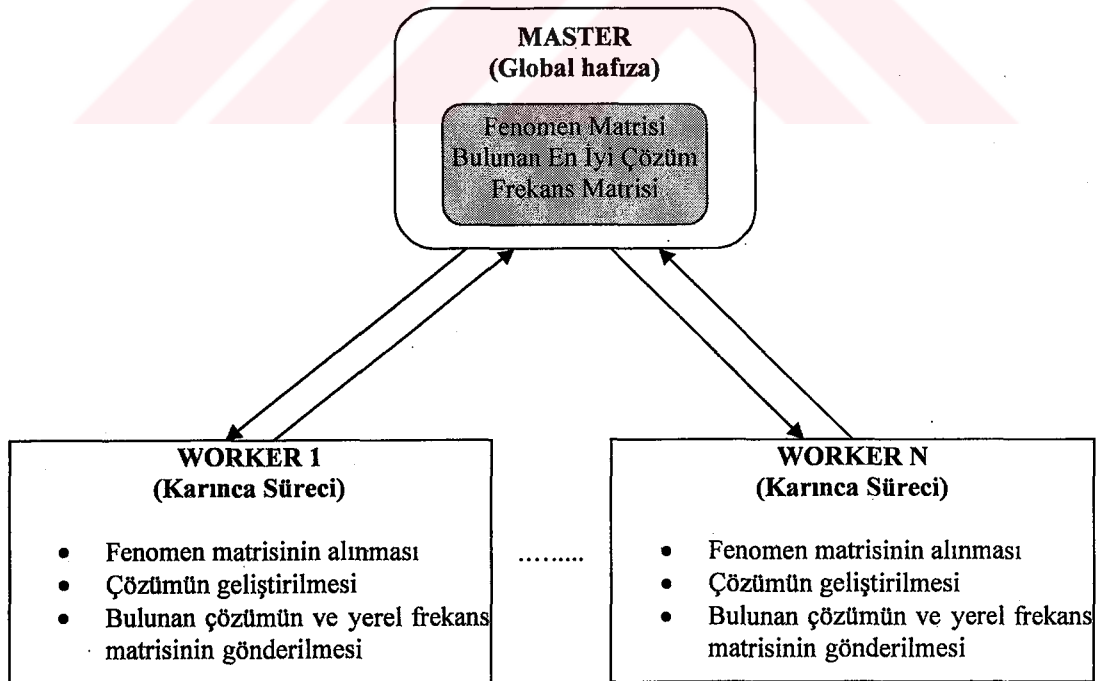
PKA ile MKA arasında, algoritmanın paralel yapısı, yerel arama prosedürü, fenomen izi matrisinin güncellenmesi ve çeşitlendirme mekanizması olmak üzere 4 önemli farklılığı vardır.

5.1.4.1 Algoritmanın Paralel Yapısı

Daha büyük problemlerin çözüm etkinliğini artırmak için algoritmanın paralel modeli geliştirilmiştir. Programlama stili olarak *master/workers* paradigması kullanılmıştır. *Master*, bütün haberleşmelerin içinden geçtiği merkezi bir hafızayı kullanır ve arama esnasında elde edilen bütün global bilgileri tutar. *Worker*, arama sürecini yerine getirir.

Talbi vd. (2001) tarafından yapılan çalışmada fenomen matrisi ve bulunan en iyi çözüm *master* tarafından yönetilir. Her bir iterasyonda *master* bütün *workerlar* için fenomen izi matrisini bildirir. Her *worker*, bir karıncayı kontrol eder. *Worker*, fenomen iz matrisini alır, çözümü geliştirir, bulunan çözümü ve fenomen matrisinin güncellenmesiyle oluşan yerel frekans matrisini *master*'a gönderir. Böylece süreç itere edilmiş olur.

Senkronize *master/workers* modeli Şekil 5.1'deki gibi gösterilebilir.



Şekil 5.1 Algoritmanın paralel yapısı (Talbi vd., 2001)

5.1.4.2 Yerel Arama Prosedürü

PKA'nın yerel arama prosedüründe tabu arama (Glover, 1989) metodu kullanılmıştır. Tabu listesinde sadece değiştirilen (i,j) eleman çifti tutulmuştur. Algoritmanın etkinliği tabu listesinin büyüklüğü ile doğrudan ilişkili olması sebebiyle en uygun tabu listesinin belirlenmesinde bazı deneysel çalışmalardan faydalanılmıştır. n boyutlu problem için $n/2$ ve $(3n)/2$ rassal olarak elde edilen tabu listesi boyutları için iyi sonuçlar elde edilmiştir. Aspirasyon kriteri olarak bulunan en iyi çözümden daha iyi bir sonucun bulunması tanımlanmıştır.

5.1.4.3 Fenomen Matrisinin Güncellenmesi

KAP için MKA'da fenomen iz matrisinin güncellenmesi iki aşamadan oluşmaktaydı. Birincisi, (5.12)'deki eşitliğe göre bütün fenomen izleri fenomenin buharlaşmasını simüle etmek için azaltılmıştı. Bu PKA içinde aynen kullanılmıştır. İkinci kısım olan, elde edilen en iyi çözüm (π^*) hesaba katılmasıyla iz matrisinin güçlendirilmesi, (5.14)'deki eşitlikte görüldüğü gibi elde edilen en kötü çözümünde (π^-) hesaba katılmasıyla geliştirilmiştir.

$$\tau_{i\pi_i} = (1-\alpha) \cdot \tau_{i\pi_i} + \frac{\alpha}{f(\pi)} \frac{f(\pi^-) - f(\pi)}{f(\pi^*)} \quad (5.14)$$

5.1.4.4 Çeşitlendirme Mekanizması

PKA'da tıpkı MKA olduğu gibi, $n/2$ iterasyon sonunda bir gelişme sağlanamıyorsa çeşitlendirme mekanizması aktif hale getirilmiştir. Bu iki algoritmanın çeşitlendirme aşamasındaki farkı mekanizmanın işleme yapısındadır. PKA'nın çeşitlendirme mekanizmasında frekans matrisi kullanılır. Çünkü frekans matrisi $t=0$ anından itibaren atanan bütün değerleri saklar. Çeşitlenme mekanizmasının aktifleşmesine sebep olan çözümün gelişmemesi durumu ortaya çıktığında frekans matrisinin tuttuğu en iyi değerler kullanılarak algoritma çalışmaya devam eder.

PKA'da yer alan bir çok değişken parametrenin değerlerinin belirlenmesinde deneysel çalışmalar yapılarak Çizelge 5.3'de verilen değerler elde edilmiştir. Bu parametre değerleri kullanılarak algoritma MKA ile aynı problemler için denenmiş ve bir çok problem için bilinen en iyi değerler elde edilmiştir.

Çizelge 5.3 Parametrelerin ölçüm değerleri

Parametre	Belirlenen ölçüm değeri
T (takas sayısı)	n/3 (n, problemin boyutu)
α (izin buharlaşmadan kalan kısmı)	0,1
S (çeşitlendirme kararının verileceği iterasyon)	n/2
q (1. takas stratejisinin belirlenme olasılığı)	%90
m (karınca sayısı)	10
Q (her bir iterasyonda iz bırakmanın önemi)	100

5.2 Esnek İmalat Sisteminde Tek Sıralı Tesis Yerleşimi Problemi için Karınca Algoritması

Esnek imalat sistemi (EİS), merkezi bir bilgisayar tarafından kontrol edilen ve bağlanan robot, otomatik kılavuzlu araç v.b. esnek malzeme taşıma sisteminin ve makinelerin bir setinden oluşan üretim sistemidir. EİS, otomasyonun yüksek seviyeli olması, makine sayısının daha az olması, sık sık yapılan set-up, bilgi akışının ve miktarının daha büyük olması açısından klasik sistemlerden farklıdır (Kusiak, 1990).

EİS'nin uygulamalarında ve tasarımında karşılaşılan en önemli problemlerden birisi imalat hücrelerine makinelerin yerleştirilmesidir. Solimanpur vd. (2003) EİS'nin tek sıralı yerleşim problemini çözmek için KA'yı uygulamışlardır.

5.2.1 Matematiksel Modelin Geliştirilmesi

Tek sıralı makinelerin yerleşimi için matematiksel modelde malzeme taşıma aracı tarafından alınan toplam mesafeyi belirten amaç fonksiyonu şu şekilde tanımlanmıştır.

$$\min \sum_{i=1}^M \sum_{j=1}^M \sum_{h=1}^{M-1} \sum_{l=h+1}^M (f_{ij} + f_{ji}) d_{hl} x_{ih} x_{jl} \quad (5.15)$$

Notasyonlar:

M, makinelerin ve yerleşimlerin toplam sayısı,

f_{ij} , i makinesinden j makinesine yapılan gezinti miktarı,

d_{hl} , h ve k yerleşimlerinin merkezleri arasındaki mesafe,

$$x_{ih}^* = \begin{cases} 1, & i \text{ makinesi } h \text{ yerlesimine yerlestirilirse} \\ 0, & \text{aksi takdirde} \end{cases},$$

Bu problemin kısıtları ise (5.16)'nın kullanımıyla elde edilen h ve k yerleşimlerinin merkezleri arasındaki mesafenin hesaplanması ve (5.17)'deki eşitlikle elde edilen bir makinenin sadece bir yerleşime, bir yerleşime de sadece bir makinenin atanmasını sağlayan kısıttır

$$d_{hl} = \sum_{b=1}^M \sum_{t=h+1}^{l-1} L_b x_{bt} + \sum_{b=1}^M \sum_{r=1}^M \sum_{t=h}^{l-1} s_{br} x_{bt} x_{r,t+1} + \frac{1}{2} \sum_{b=1}^M L_b x_{bh} + \frac{1}{2} \sum_{b=1}^M L_b x_{bl} \quad (l \leq h < k \leq M) \quad (5.16)$$

$$\sum_{i=1}^M x_{ih} = 1, \quad \sum_{h=1}^M x_{ih} = 1, \quad (h, i = 1, 2, 3, \dots, M), \quad (5.17)$$

L_i , i makinesinin uzunluğunu verirken, s_{ij} , i ve j makineleri arasındaki boşluğu verir.

Makinelerin permütasyonunun π ve permütasyonlar setinin Π olduğu bir yerde amaç fonksiyonunun 0-1 non-lineer matematiksel modeli (5.18)'deki eşitlik gibi yeniden formülize edilmiştir.

$$\min \sum_{h=1}^{M-1} \sum_{l=h+1}^M \left(f_{\pi(h)\pi(l)} + f_{\pi(l)\pi(h)} \right) \left(\sum_{t=h+1}^{l-1} L_{\pi(t)} + \sum_{t=h}^{l-1} S_{\pi(t)\pi(t+1)} + \frac{L_{\pi(h)}}{2} + \frac{L_{\pi(l)}}{2} \right) \quad (5.18)$$

$\pi(h)$, π permütasyonunda h'inci makineyi göstermektedir.

5.2.2 Problemin Çözümü için Karınca Algoritması

Solimanpur vd. (2003) amaç fonksiyonu bir önceki bölümdeki gibi tanımlanmış olan EİS'de tek sıralı tesis yerleşimi probleminin çözümü için KA uygulamasını yapmışlardır.

Geliştirilen algoritma 6 kısımdan oluşmaktadır.

1. Çözümlerin geliştirilmesi,
2. Sezgisel bilgi,

3. Fenomenin güncellenmesi,
4. Seçim olasılığı,
5. Yerel arama algoritması,
6. Durdurma kriteri.

5.2.2.1 Çözümün Geliştirilmesi

Problemin çözümü gerçek hayat karıncalarında yem kaynaklarından yuvalarına giden yola benzemektedir. Tabu arama algoritmasına benzer bir biçimde, *taşıma* olarak adlandırılan arama mekanizması tam bir çözüm yapılandırması için karınca algoritmasında kullanılır. Tek sıralı tesis yerleşimi probleminin formülizasyonu için uygun ve tam bir çözüm bütün makinelerin permütasyonu olarak düşünülür. Geliştirilen karınca algoritmasında her bir karınca başlangıçta yerleşim 1'e bir makine atar, daha sonra yerleşim 2'ye bir makine atar ve tam bir çözüm elde edilene kadar bu işlem devam eder.

$v=(i,l)$, h yerleşimine i makinesinin atanmasını gösterir. k 'ıncı karınca $v=(i,l)$ taşımasını seçerse, i indeksi Tabu_k listesine eklenir. Makinelerin tamamı atanınca tabu_k listesi de tamamen dolmuş olur. Bu taşıma işlemi V^k hafızasında tutulur. V^k iterasyonun sonunda iz miktarını güncellemek için kullanılır.

5.2.2.2 Sezgisel Bilgi

Yapay karıncalar yerleşimlere makinelerin atanmasında bazı sezgisel bilgiler tutarlar. $v=(i,l)$ taşımasının istenebilirlik derecesi (η_{il}) ile ilgi olan sezgisel bilgi sezgisel bir yaklaşım kullanılarak hesaplanır.

Geliştirilen algorithmada sezgisel bilgi $v=(i,l)$ taşımasının kısmi katkısını ifade eder. Buna göre geliştirilen sezgisel bilgi (5.19)'daki gibi formülize edilir:

$$\eta_{il} = \frac{1}{1 + \sum_{h=1}^{l-1} (f_{\pi(h)l} + f_{l\pi(h)})d_{hl}} \quad (5.19)$$

(5.19)'daki hiperbolik fonksiyondan da anlaşılacağı gibi amaç fonksiyona ne kadar küçük katkı yapılırsa o taşımanın istenirliği o kadar yüksektir.

5.2.2.3 Fenomen İzini Güncelleme

Bir taşımanın karıncalar tarafından ne kadar sık kullanıldığını gösteren taşımaların iz seviyesi miktarıdır. k karıncası tarafından yapılan (i,l) taşıması sonucunda taşımanın iz miktarında meydana getirdiği değişimi hesaplayan (5.20) eşitliği şu şekilde gösterilebilir:

$$\Delta \tau_{il}^k = \lambda \frac{z^*}{z_k} \quad (5.20)$$

Notasyonlar:

$\Delta \tau_{il}^k$, (i,l) taşımasının iz miktarında meydana gelen değişim,

z^* , o ana kadar elde edilen en iyi amaç fonksiyonu değeri,

z_k , k karıncasının elde ettiği amaç fonksiyonu değeri,

λ , hesaplama sürecinin dinamiklerini kontrol eden ölçü faktörüdür.

Buna göre (i,l) taşımasının toplam iz seviyesi (5.21)'deki formül ile hesaplanır.

$$\tau_{il}(t) = \tau_{il}(t-1) + \sum_{k=1}^m \Delta \tau_{il}^k \quad (5.21)$$

Notasyonlar:

$\tau_{il}(t)$, t iterasyonundaki (i,l) taşımasının toplam iz miktarı,

m, karınca sayısıdır.

5.2.2.4 Seçme Olasılığı

k karıncasının (i,l) taşımasını seçmesi iki ana faktöre bağlıdır.

1. l yerleşimine i makinesinin yerleştirilme istenebilirliği (η_{il}),
2. (i,l) taşımasının iz seviyesidir.

Seçme olasılığı tanımlanırken bu iki faktörün kombinasyonu kullanılır. Bu bilgiler ışığında olasılık, (5.22) kullanılarak hesaplanır.

$$p_{il}^k = \frac{\alpha \times \tau_{il} + (1 - \alpha) \times \eta_{il}}{\sum_{l \in Tabu_k} (\alpha \times \tau_{il} + (1 - \alpha) \times \eta_{il})} \quad (5.22)$$

α , η_{il} 'e karşılık τ_{il} 'nin önemini ifade eden bir katsayıdır.

5.2.2.5 Yerel Arama Algoritması

KA tarafından elde edilen çözümlerin geliştirilmesi için algoritma yerel arama algoritması ile bütünleştirilir. Öncelikle karınca algoritması tarafından bir çözüm elde edilir ve daha sonra bu çözüm yerel arama algoritmasıyla geliştirilir.

Bu yerel arama algoritmasında bütün makine çiftleri yer değiştirir ve bu değişimler sonucunda amaç fonksiyonundaki maksimum düşüşü sağlayan çift seçilir. Elde edilen yeni çözüme göre fenomen izleri güncellenir ve yeni iterasyon başlar. Bu süreç, gelişmede istenilmeyen noktaya varıldığı anda bitirilir.

5.2.2.6 Durdurma Kriteri

Her bir iterasyonun en iyi amaç fonksiyonu değeri kaydedilir ve o ana kadar elde edilen en iyi çözüm ile karşılaştırılır. Önceden kontrol parametresi olarak belirlenen *maksimum-iter* iterasyonuna kadar gelişme sağlanamamışsa algoritma durdurulur.

Solimanpur vd. (2003) yapmış oldukları deneysel çalışma sonucunda algoritmanın 4 değişken parametresi için Çizelge 5.4'deki uygun değerleri elde etmişlerdir.

Çizelge 5.4 Parametrelerin ölçüm değerleri

Parametre	Belirlenen ölçüm değeri
m (karınca sayısı)	M/3 (M, makine sayısı)
maksimum-iter (kontrol parametresi)	M/3
α (η_{il} 'e karşılık τ_{il} 'nin önem derecesi)	0,8
λ (ölçü faktörü)	0,005

4 makineden 30 makineye kadar 16 problem üzerinde algoritmayı deneyen Solimanpur vd. (2003) önceden aynı problemler için çözüm üreten 4 algoritma ile karşılaştırmışlardır. Geliştirilen algoritma 11 problem için elde edilen en iyi çözümleri bulurken, kalan 5 problem için daha iyi çözüm performansı elde etmiştir.



6. UYGULAMA: TESİS YERLEŞİMİ PROBLEMİ İÇİN TAVLAMALIKARINCA ALGORİTMASI

Bu bölümde tesis yerleşimi probleminin çözümü için geliştirilen karınca algoritması tabanlı “TavlamalıKarınca” algoritması tanıtılacaktır. Bu çalışmada, bütün departmanların eşit alana sahip olduğu tesis yerleşimi problemi incelenmiştir. Daha önceki bölümlerde de değinildiği gibi departmanların eşit alanlara sahip olduğu tesis yerleşimi problemine Kareli Atama Problemi (KAP) denilmektedir. NP-Zor yapıya sahip KAP için büyük boyutlu problemlerde optimum çözümü bulmak zordur. Bu nedenle çalışmada KAP için TavlamalıKarınca algoritması geliştirilmiştir. Algoritma yerel arama prosedürü tavlama benzetiminden (TB) oluşan melez yapılı bir algoritmadır. Yerel arama prosedür olarak TB’nin seçilme sebebi, diğer zeki sezgisellere göre yerel optimuma yakalanma olasılığını azaltan bir geri besleme mekanizmasına sahip olmasıdır.

6.1 Problemin Tanımlanması

KAP’de önceki bölümlerde de değinildiği gibi, malzeme taşıma maliyeti minimize eden, n yerleşime n tesisinin atanması amaçlanmaktadır. Problemin tanımlanmasında 3 varsayım kabul edilir.

1. Bütün dörtgen yerleşimlerin alanları eşittir,
2. Her tesis her alana yerleştirilebilir,
3. Yerleşim sayısı departman sayısına eşittir.

n boyutlu bir problemin amaç fonksiyonu (6.1)’deki eşitlik gibidir;

$$\min z = \sum_{i,j=1}^n d_{ij} \cdot f_{\pi(i)\pi(j)} \quad (6.1)$$

$\Pi : i \rightarrow \pi(i)$, i yerleşimine $j = \pi(i)$ tesisinin atanması olarak yorumlanabilir. ($i = 1,2,3, \dots, n$)

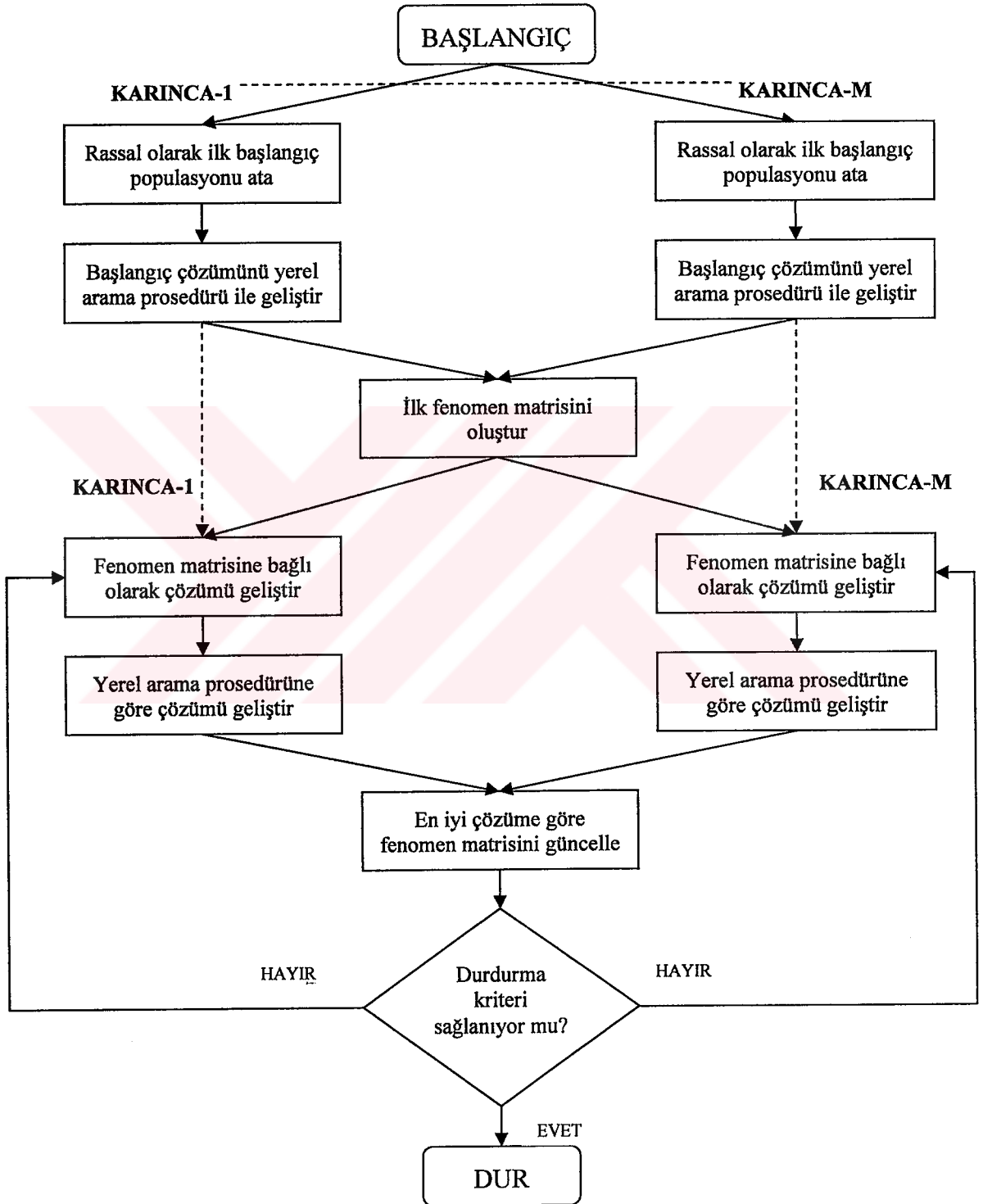
d_{ij} , i tesisi ile j tesisi arasındaki mesafe,

$f_{\pi(i)\pi(j)}$, π permütasyonunda i ’inci eleman ile j ’ninci eleman arasındaki akış miktarıdır.

İki tesis arasında veri transferinin maliyeti yerleşimler arasındaki uzaklıkla, yerleşimlere atanan tesisler arasındaki akış miktarının çarpımına eşittir.

6.2 Algoritmanın Genel Yapısı

TavlamalıKarıncı algoritması büyük boyutlu KAP'lerin çözüm kalitesini geliştirmek amacıyla geliştirilmiş bir algoritmadır. Algoritmanın temel yapısı bu çalışmadan önce geliştirilen KA ve türevlerinden oluşan sezgisellerle benzerlik göstermektedir.



Şekil 6.1 TavlamalıKarıncı algoritmasının genel akış şeması

Genel akış şeması Şekil 6.1’de gösterilen TavlamalıKarınca algoritmasının en basit algoritmik ifadesi ise şu şekilde ifade edilebilir.

BAŞLANGIÇ

m karınca için başlangıç çözümü ata,

m çözümü yerel arama prosedüründe geliştir ve en iyi çözümü seç,

En iyi çözüme göre ilk fenomen matrisini oluştur,

For $I_{maksimum}$ iterasyon tekrarla

ÇÖZÜMLERİN GELİŞTİRİLMESİ

For her bir karınca $k = 1, 2, 3, \dots, m$ do

Fenomen izi matrisinin kullanarak k ’ıncı karıncanın çözümünü geliştir,

Tavlama benzetimi tabanlı yerel arama prosedürünü kullanarak k ’ıncı karıncanın çözümünü geliştir,

k ’ıncı karınca için yoğunlaşma mekanizması ile yeni bir başlangıç çözümü ata,

End,

FENOMEZ İZİ MATRİSİNİN GÜNCELLENMESİ

Fenomen matrisini o ana kadar elde edilen en iyi üç çözümü hesaba katarak güncelle,

ÇEŞİTLENDİRME

Eğer gelişme yoksa çeşitlendirme mekanizmasını uygula,

End.

6.3 Karıncaların Başlangıç Çözümleri

Başlangıçta m adet karıncaya rassal olarak seçilen bir permütasyon verilir. Elde edilen $m \times n$ adet çözüm, daha sonra detaylı olarak tanımlanacak olan TB yerel arama prosedürü kullanımıyla optimize edilir.

6.4 Fenomen İzi Matrisi Başlangıcı

Başlangıçta fenomen izi matrisi hiç bilgi içermez. Bütün τ_{ij} fenomen izleri aynı τ_0 değerine atanır. Bu değer hesaplanmasında, (6.2)'deki eşitlikte olduğu gibi o ana kadar elde edilen en iyi permütasyonun amaç fonksiyonu değeri kullanılır.

$$\tau_0 = \frac{1}{100 \cdot (f(\pi^*))} \quad (6.2)$$

Notasyonlar:

τ_0 , fenomen izleri başlangıç değeri,

$f(\pi^*)$, o ana kadar elde edilen en iyi çözüm,

KAP için geliştirilen daha önceki KA ve türevi olan algoritmalarda ilk fenomen izi matrisinin oluşturulmasında kullanılan Q değişken parametresi bu çalışmada kullanılmamıştır. Q parametresinin değeri belirlenirken önceki çalışmalar (Gamberdalla vd., 1997; Talbi vd., 2001) referans alınarak sabit bir değer atanmıştır.

6.5 Çözümlerin Geliştirilmesi

KAP problemi için TavlamalıKarınca algoritmasının bundan önceki uygulamalardan ayıran özelliği bu aşamada geliştirilmiştir. Bu aşama, fenomen izine bağlı gelişme ve yerel arama prosedürüne bağlı gelişme olmak üzere iki kısımdan oluşmaktadır.

6.5.1 Fenomen İzine Bağlı Gelişme

Fenomen izine bağlı gelişmede, m adet karınca kendi permütasyonuna (π^k) takas sayısı (T) kadar gelişme prosesini uygular.

Gelişme Prosesi

Rassal olarak r parametresini seç, ($r = 1, 2, 3, \dots, n$),

0,9 olasılığı ile $\tau_{r\pi_s}^k + \tau_{s\pi_r}^k$ ifadesini maksimum yapan, 0,1 olasılığı ile

$\sum_{j \neq r} \left(\tau_{r\pi_s}^k + \tau_{s\pi_r}^k \right)$ ifadesini maksimum yapan s parametresini belirle,

π^k çözümünde π_r^k ve π_s^k değerleri yer değiştirilerek $\hat{\pi}^k$ geliştirilmiş permütasyonunu belirle.

6.5.2 Yerel Arama Prosedürü ile Gelişme

Her bir karınca fenomen izine bağlı gelişme ile elde ettiği $\hat{\pi}^k$ çözümünü yerel arama prosedüründe bir kez daha geliştirilerek $\pi^{\infty k}$ çözümlerini elde edilir.

Bu çalışmada yerel arama prosedür olarak tavlama benzetimi (TB) kullanılmıştır. Bu nedenle çalışmanın bu kısmında öncelikle TB algoritmasından kısaca bahsedilecek daha sonra Tavlama Karınca algoritmasındaki kullanımı anlatılacaktır.

6.5.2.1 Tavlama Benzetimi ve Genel Algoritmik Yapısı

Tavlama Benzetimi (TB), kombinatoriyal optimizasyon problemlerinin çözümü için kullanılan ve katıların fiziksel tavlama sürecini taklit eden bir stokastik arama yöntemidir. İlk kez Kirkpatrick vd. (1983) ve Cerny (1985) katıların tavlama benzetimini optimizasyon problemlerinin çözümünde nasıl kullanılabileceğini göstermişlerdir. Günümüze kadar TB, bugüne kadar moleküler fizik ve kimya, tesis yerleşimi, montaj hattı dengeleme, çizelgeleme, şebeke tasarımı vb. çeşitli alanlarda optimizasyon problemlerinin çözümünde başarıyla uygulanmıştır.

Kombinatoriyal optimizasyon problemi için araştırmacılara yön veren fiziksel tavlama, bir katının düşük enerjili durumlarının elde edilmesi prosesidir. Eritilen katının ısısının çok yavaş düşürülmesi ile katının düşük enerjili durumuna ulaşması sağlanır.

Tabu Arama sezgiselinde olduğu gibi yerel arama yöntemleri yerel optimuma takılıp global optimum için daha fazla arama yapmamalarından kaynaklanan bir eksikliğe sahiptir. TB, yerel arama yöntemlerinin bu eksikliğini gidermeye çalışan bir yöntemdir. TB'nin diğer yerel arama yöntemlerinden ayrılan en temel özelliği, bazen yeni çözümün (j) amaç fonksiyon değeri (f(j)) mevcut çözümün (i) amaç fonksiyon değerinden büyük olmasına rağmen yeni çözümü mevcut çözüm olarak kabul edip arama işlemine devam eder. Yani, mevcut çözümün

amaç fonksiyonu değeri ile yeni çözümün amaç fonksiyonu değeri arasındaki fark $\Delta = f(j) - f(i)$ ile ifade edilirse,

$\Delta \leq 0$ ise, j mevcut çözüm olarak kabul edilirken,

$\Delta > 0$ ise, j $e^{-\Delta/T}$ olasılığı ile mevcut çözüm olarak kabul edilir. Burada T , sıcaklık değeridir.

Katıların tavlama sıcaklığında olduğu gibi yüksek sıcaklık ile başlanır ve arama işlemi sırasında yavaş yavaş düşürülür. Bu şekilde, aramanın başlangıç aşamalarında amaç fonksiyonunda büyük artışların olduğu yeni çözümler kabul edilir. Böylece kötü çözümler de kabul edilerek yerel optimuma takılma riski azaltılır. Aramanın sonuna yaklaşıldığında ise sıcaklık çok düşeceğinden sadece amaç fonksiyonunda iyileşme sağlayan çözümler kabul edilecektir. TB algoritmasının genel yapısı şu şekilde ifade edilebilir.

BAŞLANGIÇ

$t=0$;

Bir başlangıç çözümü üret;

Bir başlangıç sıcaklığı belirle $T_0 > 0$;

Sıcaklık değişim sayacını sıfırla $t=0$;

ÇÖZÜMÜN GELİŞTİRLMESİ

Repeat

i mevcut çözümünden komşu üretme mekanizmasıyla j yeni çözümünü üret;

$\Delta = f(j) - f(i)$;

If $(\Delta < 0)$ then

$i = j$;

else

If $(\text{random}(0,1) < \exp(-(\Delta / T)))$ then

$i = j$;

ITERASYONUN BİTİŞİ

$t := t + 1;$

$T := T(t);$

Until (durdurma koşulu)

TB algoritmasında, başlangıç ve bitiş sıcaklıkları (T_0, T_{son}), iterasyon sayısı (I), kötü çözümlerin kabul edilme olasılığını belirleyen sıcaklık düşürme fonksiyonunun ve algoritma durdurma kriterinin belirlenmesi gerekir.

Bunların yanında mevcut çözümden yeni çözümü üreten komşuluk geliştirme mekanizmasının belirlenmesi gerekir.

6.5.2.2 TavlamalıKarınca Algoritmasında Tavlama Benzetiminin Kullanımı

Diğer yerel arama algoritmalarının eksikliği olan yerel optimuma takılma riskini azaltan bir geri besleme yapısına sahip olan TB bu çalışmada TavlamalıKarınca algoritmasının yerel arama prosedürü olarak kullanılmıştır.

Bu çalışmada, soğutma çizelgesi olarak (6.3)'teki eşitlik kullanılmıştır (Lundy ve Mees, 1986).

$$T_{k+1} = T_k / 1 + (r \times T_k) \quad (6.3)$$

Notasyonlar:

T_k , mevcut sıcaklık,

r , soğuma oranıdır.

Soğuma oranı,

$$r = [(T_0 - T_{son}) / (I - 1) T_0 T_{son}] \quad (6.4)$$

olarak tanımlanır.

TavlamalıKarınca algoritmasında kullanılan yerel arama prosedürünün genel yapısı şu şekilde oluşturulmuştur.

BAŞLANGIÇ

$t=0$;

Rassal olarak başlangıç çözümü üret;

Başlangıç ve bitiş sıcaklıklarını belirle $T_0, T_{son} > 0$;

ÇÖZÜMÜN GELİŞTİRLMESİ

Repeat

i mevcut çözümünden komşu üretme mekanizmasıyla j yeni çözümünü üret;

$\Delta = f(j) - f(i)$;

If ($\Delta < 0$) then

$i = j$;

else

If ($\text{random}(0,1) < \exp(-(\Delta / T))$) then

$i = j$;

else

$i = i$;

ITERASYONUN BİTİŞİ

$T_{t+1} = T_t / 1 + (r \times T_t)$

$t = t + 1$;

Until (durdurma koşulu)

Uygulamada T_0 ve T_{son} değerleri daha önce iyi sonuçlar elde eden çalışmalar referans alınarak belirlenmiştir. Buna göre;

$T_0 = 100$ ve $T_{son} = 0,01$ değerlerini almıştır.

İterasyon sayısı Tavlamalı Karınca algoritmasının değişken parametresi olarak kabul edilmiş ve deney tasarımı sonucunda ölçüm değeri belirlenmiştir.

Komşuluk üretme mekanizması olarak, rassal seçilen çiftli yer değiştirme mekanizması ile en iyi çiftli yer değiştirme mekanizması kullanılmıştır.

6.5.2.2.1 Çiftli Yer Değiştirme Mekanizması

Bu mekanizmanın çalışma prensibi sıraları rassal seçilen iki π permütasyon elemanının yer değiştirmesidir. Örneğin; boyutu 5 olan bir KAP için rassal seçilen 3. ve 5. sıradaki elemanların yer değiştirmesi.

$$\begin{array}{ccc} \underline{1-2-3-4-5} & \longrightarrow & \underline{1-2-3-4-5} \\ 3-2-1-5-4 & & 3-2-4-5-1 \end{array}$$

6.5.2.2.2 En iyi Çiftli Yer Değiştirme Mekanizması

Bu mekanizmada, n boyutlu bir problem için mevcut çözümden çiftli yer değiştirme mekanizması ile n adet çözüm bulunur. Bulunan bu çözümler içinden en iyi sonuç veren seçilir.

6.5.3 Yoğunlaşma

İterasyon sonunda her bir karınca $\pi^{\infty k}$ ve π^k arasında en iyi permütasyon ile sonraki iterasyona başlar. Tavlamalı Karınca algoritmasında da yoğunlaşma mekanizmasına göre bir iterasyonda genel bir k karıncasının davranışı üç arama adımından oluşur. Bunlar, ilk çözüm (π^*), fenomen izine bağlı gelişme ($\hat{\pi}$) ve yerel aramaya bağlı gelişmedir (π^{∞}).

i 'inci iterasyonun başlangıcında, yoğunlaşmanın aktif olmadığı varsayılmıştır. Bu nedenle,

$$\pi^k(i+1) \leftarrow \pi^{\infty k}(i) \text{ 'dir.}$$

$(i+1)$ 'inci iterasyonun sonunda yoğunlaşma aktif hale getirilir çünkü bulunan en iyi çözüm geliştirilmiştir. Bu nedenle,

$$\pi^k(i+2) \leftarrow \pi^{\infty k}(i+1) \text{ 'dir.}$$

$(i+2)$ 'nci iterasyonun sonunda yoğunlaşma aktiftir çünkü $\pi^k(i+2)$, $\pi^{\infty k}(i+1)$ 'den daha iyi sonuç vermiştir. Bu nedenle,

$\pi^k(i+3) \leftarrow \pi^k(i+2)$ ataması yapılır.

6.5.4 Fenomen İz Matrisinin Güncellenmesi

Fenomen izlerinin güncellenmesi bundan önce yapılan KA uygulamalarından farklı yapılır. Fenomen izleri üretilen en iyi üç çözümün amaç fonksiyonu değerlerinin ortalaması hesaplanarak güncellenir. Başta bütün fenomen izleri fenomenin buharlaşmasını simüle etmek için (6.5) kullanılarak azaltılır.

$$\tau_{ij} = (1 - \alpha) \tau_{ij} \quad 1 \leq i, j \leq n \quad \text{ve} \quad 0 < \alpha < 1 \quad (6.5)$$

Daha sonra, fenomen izleri iterasyon sonunda elde edilen en iyi üç çözümün $(\pi^*, \pi^{*-1}, \pi^{*-2})$ ortalama değeri hesaba katılmasıyla (6.6)'daki eşitlik kullanılarak güçlendirilir.

$$\tau_{i\pi^*} = \tau_{i\pi^*} + \frac{\alpha}{f\left(\frac{\pi^* + \pi^{*-1} + \pi^{*-2}}{3}\right)} \quad (6.6)$$

$0 < \alpha < 1$, fenomen izlerinin buharlaşmasını kontrol eden bir parametredir.

0'a yakın α değeri, uzun bir zaman aktif kalan fenomen izi anlamına gelir,

1' e yakın α değeri, buharlaşmanın yüksek seviyesi anlamına gelir.

6.5.5 Çeşitlendirme Mekanizması

TavlamalıKarınca algoritmasında $n/2$ iterasyon sonunda en iyi çözümde ilerleme bulunamamışsa çeşitlendirme mekanizması aktif hale getirilmiştir. Çeşitlendirme mekanizması devreye girdiğinde fenomen iz matrisindeki bütün değerler silinir. Sadece bütün karıncalar için o ana kadarki üretilen en iyi çözüm saklanır. Bu çözümler karıncaların ilk çözümü olarak kabul edilir ve yeniden algoritma çalıştırılmaya başlanır.

6.5.6 KAP için TavlamalıKarınca Algoritmasının Adımları

Yukarıda işlemleri detaylı bir şekilde anlatılan TavlamalıKarınca'nın algoritmik adımları şu şekilde ifade edilebilir;

ADIM 1. BAŞLANGIÇ

m karıncanın her birisi için m adet rassal olarak başlangıç çözümü üret, $\pi^1(1), \dots, \pi^m(1)$,

ADIM 1.1 Her Bir Karıncanın Başlangıç Değerlerini Yerel Arama Prosedüründe Geliştir

Başlangıç ve bitiş sıcaklıklarını belirle $T_0, T_{son} > 0$;

$t=0$;

Repeat

$\pi^k(1)$ mevcut çözümünden komşu üretme mekanizmasıyla $\pi^k(1')$ yeni çözümünü üret;

$$\Delta = f(\pi^k(1')) - f(\pi^k(1)) ;$$

If ($\Delta < 0$) then

$$\pi^k(1) = \pi^k(1') ;$$

else

If ($\text{random}(0,1) < \exp(-(\Delta / T))$) then

$$\pi^k(1) = \pi^k(1') ;$$

else

$$\pi^k(1) = \pi^k(1) ;$$

$$T_{t+1} = T_t / 1 + (r \times T_t)$$

$t = t + 1$;

Until (durdurma koşulu)

ADIM 1.3 En İyi Çözümlerin Belirlenmesi

Elde edilen çözümler içinden en iyi çözümü (π^*) seç,

ADIM 1.2 İlk Fenomen İz matrisini Oluştur

İlk fenomen matrisini (D) en iyi çözüme göre (π^*) oluştur,

For ($i = 1$ to I^{maksimum}) repeat

$\{I^{\text{maksimum}}, \text{iterasyon sayısı}\}$

ADIM 2. ÇÖZÜMLERİN GELİŞTİRİLMESİ

For (her bir karınca çözümü ($\pi^k(i)$)) do

ADIM 2.1 Takasların Uygulanması

$\pi^k(i)$ çözümünden $\hat{\pi}^k(i)$ çözümünü elde eden T adet takası uygula,

ADIM 2.2 Yerel Arama Prosedürünün Uygulanması

$\hat{\pi}^k(i)$ çözümünden $\pi^{\infty k}(i)$ çözümünü elde eden yerel arama prosedürünü uygula,

End,

For (her bir karınca) do

ADIM 3. YOĞUNLAŞMA MEKANİZMASI

If (yoğunlaşma aktif) then

$\pi^k(i+1) \leftarrow \pi^k(i)$ ve $\pi^{\infty k}(i)$ çözümleri arasındaki en iyi değer,

Else,

$\pi^k(i+1) \leftarrow \pi^{\infty k}(i),$

End,

If ($\pi^k(i+1) = \pi^k(i)$) then

Yoğunlaşmayı pasifleştir,

If ($\pi^{\infty k}(i) < f(\pi^*)$) then

O ana kadar elde edilen en iyi çözümü (π^*) güncelle,

Yoğunlaşmayı aktifleştir,

ADIM 4. FENOMEN MATRİSİNİN GÜNCELLENMESİ

ADIM 4.1 Fenomen İz Miktarlarının Azaltılması

Fenomen iz matrisin değerlerinin buharlaşmayı simüle etmek için azalt ,

ADIM 4.2 Fenomen İz Miktarlarını Elde Edilen En İyi 3 Çözüme Göre Geliştir

Fenomen iz matrisin değerlerinin elde edilen en iyi 3 değer in ortalamasına göre artır,

ADIM 5. ÇEŞİTLENDİRME MEKANİZMASI

If (n/2 iterasyon sonunda gelişme sağlanmıyor) then

Çeşitlendirme mekanizmasını çalıştır.

6.5.7 Parametrelerin Belirlenmesi

TavlamalıKarıncı Algoritmasında değişken parametreler olarak; takas sayısı (T), karınca sayısı (m), yerel arama prosedürünün iterasyon sayısı (I), algoritmanın iterasyon sayısı (I^{maksimum}) ve fenomen iz matrisinde kullanılan α parametreleri belirlenmiştir. Bu güne kadar KAP için geliştirilen KA ve türevi olan algoritmalarda kullanılan Q (her bir iterasyonda iz bırakmanın önemi) ve S (çeşitlendirme kararının verileceği iterasyon) değişken parametrelerine, bu çalışmada önceden yapılan çalışmalar (Gambardella vd., 1997; Talbi vd., 2001) referans alınarak sabit değerler (Q için 10 ve S için n/2) atanmıştır.

Bu çalışmada, değişken parametrelerin (T, m, I, I^{maksimum} ve α) uygun düzeylerinin belirlenmesi için deney tasarımı yapılmıştır.

7. DENEY TASARIMI

Kareli Atama Problemine (eşit alanlı tesis yerleşim problemi) çözüm üretmesi için geliştirilen “TavlamalıKarınca” algoritmasının parametre setlerini ve etkinliğini belirlemek için deney tasarımı düzenlenmiştir. Bu bölümde, uygulanan deney tasarımı açıklanarak algoritmanın etkinliği araştırılmıştır.

7.1 Dikkate Alınan Etkinlik Ölçütü

Geliştirilen “TavlamalıKarınca” algoritmasının etkinliğinin belirlenmesi ve karşılaştırılması için, bilinen en iyi çözüm ile algoritmanın elde ettiği çözüm arasındaki farkın bilinen en iyi çözüme oranı ile elde edilen “çözümler arası fark” dikkate alınmıştır. Deney tasarımında kullanılacak 5 adet düzenli 5 adet düzensiz test problemi, karşılaştırma yapılacak algoritmaların denendiği problemler içinden seçilmiştir. Bu problemler ve optimum çözümleri KAP Kütüphanesinden (QAP Library) alınmıştır.

Dikkate alınan etkinlik ölçütü olan “çözümler arası fark (ÇAF)” şu formüle göre hesaplanmıştır:

$$\text{ÇAF} = (\text{Bulunan değer} - \text{Bilinen en iyi değer}) / \text{Bilinen en iyi değer} \quad (7.1)$$

7.2 Test Problemleri

Bu çalışmada, düzenli ve düzensiz sınıfına dahil problemden eşit sayıda seçilerek oluşturulan problemler seti kullanılmıştır. KAP Kütüphanesinden alınan, bu problemlerin isimleri, boyutları ve bilinen en iyi çözümleri Çizelge 7.1 ve 7.2’de gösterilmiştir.

Çizelge 7.1 Düzenli sınıfına dahil test problemleri

PROBLEM	BOYUTU	BİLİNE EN İYİ ÇÖZÜMÜ
Nug 20	20	2570
Nug 30	30	6124
Tai 30a	30	1818146
Tai 60a	60	7208572
Sko 64	64	48498

Çizelge 7.2 Düzensiz sınıfına dahil test problemleri

PROBLEM	BOYUTU	BİLİLEN EN İYİ ÇÖZÜMÜ
Bur26a	26	5426670
Bur26h	26	7098658
Tai30b	30	637117113
Tai40b	40	637250948
Tai80b	80	818415043

Bu problemlerin her biri için departmanlar arası uzaklık matrisi ve departmanlar arası ilişki matrisleri mevcuttur.

Geliştirilen TavlamalıKarıncı algoritması C++ programlama dili kullanılarak kodlanmıştır. Parametrelerin uygun düzeylerinin belirlenmesi amacı ile yapılan deney tasarımı ise MINITAB istatistik programı kullanılarak yapılmıştır.

7.3 Deney Tasarımı

TavlamalıKarıncı algoritması için öncelikle daha önceki bölümde de tanımlanan uygun parametre seti belirlenmiştir. Bu parametrelerin tasarımda kullanılacak düzeyler belirlendikten sonra MINITAB istatistik programının Deney Tasarımı (DOE-Design of Experiment) bölümü kullanılarak tasarım gerçekleştirilmiştir.

Çalışmanın deney tasarımı kısmında 10 test problemi üzerinde 2 düzeyli tam faktöryel tasarımı kullanılmıştır. TavlamalıKarıncı algoritmasına göre kodlanan programda her bir problem 96 kez denenmiştir. Diğer bir ifadeyle, her bir problem her bir parametrenin iki düzeyi (2^5) için 3 kez denenmiştir.

7.3.1 Uygun Parametre Setinin Belirlenmesi

Daha önceden de belirtildiği gibi geliştirilen algoritmanın parametreleri olarak; takas sayısı (T), karınca sayısı (m), yerel arama prosedürünün iterasyon sayısı (I), algoritmanın iterasyon

sayısı ($I^{\max}$) ve fenomen iz matrisinde kullanılan α parametreleri belirlenmiştir.

Bu beş faktöre göre, 2^k ($k=5$) iki faktörlü deney tasarımı dikkate alınan faktörler ve düzeyleri çizelge 7.3’de verilmiştir.

Çizelge 7.3 Test problemleri için parametre düzeyleri

Faktörler	Düzeyler
Takas Sayısı (T)	10 – 12
Karınca Sayısı (m)	10 – 13
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100 – 200
Algoritmanın İterasyon sayısı ($I^{\max}$)	8 – 12
İzin Buharlaşmadan Kalan Kısmını İfade Eden Parametre (α)	0,1 – 0,2

Algoritma değişik test problemleri için pek çok kez çalıştırılarak tasarımda kullanılacak uygun parametre setleri belirlenmiştir. İki faktörlü deney tasarımı kullanılarak, faktörlerin performans ölçütü üzerinde etkin olup olmadığı araştırılmıştır. Bu etkiler incelenirken ana etkiler (main effects), ikili etkiler ve üçlü etkiler (interaction effect) dikkate alınmıştır.

7.3.2 Analiz Sonuçları

Aşağıda, düzenli sınıfından NUG20, NUG30, TAI30a, TAI60a, SKO64 ve düzensiz sınıfından BUR26a, Bur26h, TAI30b, TAI40b, TAI80b test problemlerinde, geliştirilen TavlamalıKarınca algoritması için analiz sonuçları verilmektedir.

Problemlerin düzenli veya düzensiz olmasına ve problem boyutuna bağlı olarak parametrelerin etkinliğinin değişmesi sebebiyle her problem için ayrı ayrı tasarım yapılmıştır.

Tasarımda hangi ana etki ve ortak etkilerin çözüm kalitesi üzerinde anlamlı olduğunu belirlerken tablodaki P değerleri incelenir. Eğer parametrelerin P değerleri 0.1’den küçük ise bu parametreler performans ölçütü üzerinde etkin demektir.

7.3.2.1 NUG20 Tasarımı

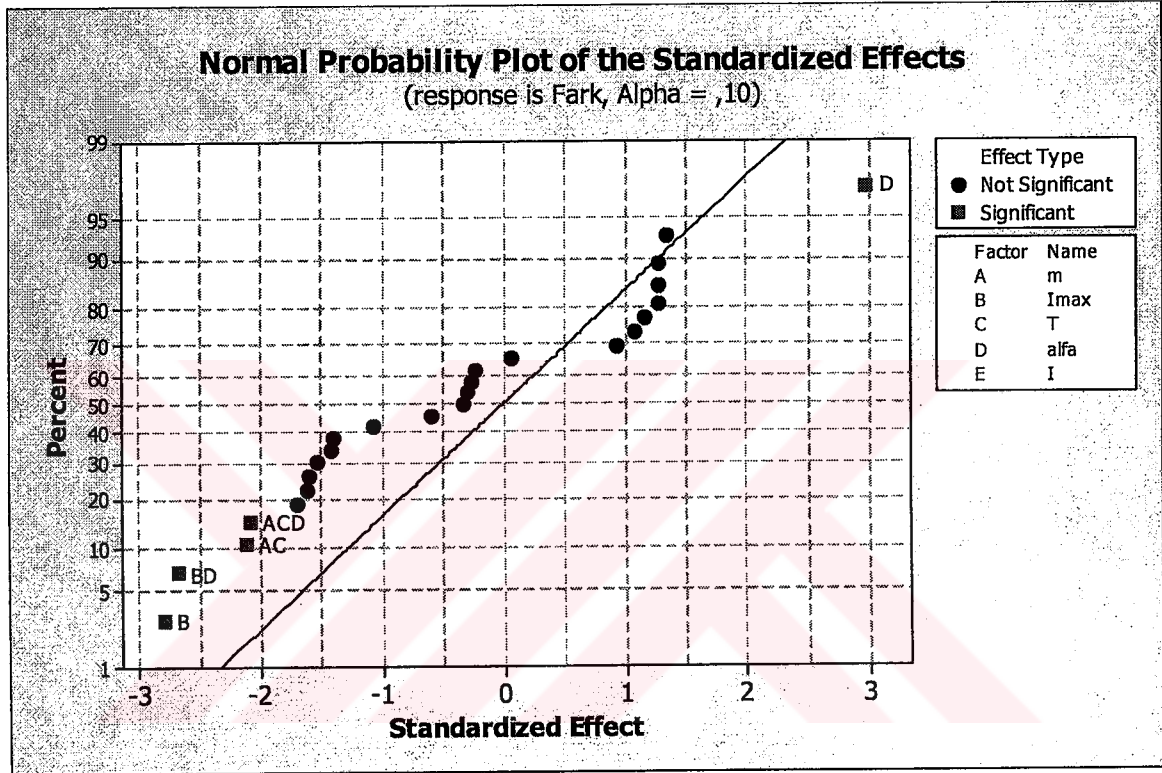
NUG20 problemi için yapılan tasarımda Çizelge 7.4'deki istatistik değerleri incelendiğinde, Algoritmanın iterasyon sayısı " $I^{\max}$ " ve " α " parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olan " $I^{\max}$ " ve " α " parametreleri bir araya geldiğinde de çözüm kalitesi performansı üzerinde etkili olmaktadır. Bunun yanında ana etkileri tek başına o kadar etkin olmayan takas sayısı " T " ve karınca sayısı " m " bir araya geldiğinde çözüm kalitesinin artmasında etkin olmaktadır. Üçlü ortak etkilerden ise yalnızca tek başlarına iken etkin olmayan fakat bir araya geldiğinde etkin olan takas sayısı " T " ve karınca sayısı " m " parametreleri " α " parametresi ile bir araya geldiklerinde oluşturdukları ortak etki anlamlı bulunmuştur.

Çizelge 7.4 NUG20 tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		0,08709	0,028120	3,10	0,021	
m	-0,09019	-0,04509	0,028120	-1,60	0,160	
$I^{\max}$	-0,15689	-0,07834	0,028120	-2,78	0,032	
T	0,09694	0,04847	0,037726	1,28	0,246	
α	0,06187	0,03094	0,010408	2,97	0,025	
I	-0,09544	-0,04772	0,028120	-1,70	0,141	
$m \cdot I^{\max}$	0,05875	0,02937	0,025155	1,17	0,287	
$m \cdot T$	-0,15994	-0,07997	0,037726	-2,12	0,078	
$m \cdot \alpha$	-0,03188	-0,01594	0,010408	-1,53	0,177	
$m \cdot I$	-0,01556	-0,00778	0,025155	-0,31	0,768	
$I^{\max} \cdot T$	-0,10744	-0,05372	0,037726	-1,42	0,204	
$I^{\max} \cdot \alpha$	-0,05562	-0,02781	0,010408	-2,67	0,037	
$I^{\max} \cdot I$	0,06781	0,03391	0,025155	1,35	0,226	
$T \cdot \alpha$	0,03562	0,01781	0,013963	1,28	0,249	
$T \cdot I$	-0,01819	-0,00909	0,037726	-0,24	0,818	
$\alpha \cdot I$	-0,03375	-0,01688	0,010408	-1,62	0,156	
$m \cdot I^{\max} \cdot T$	0,00262	0,00131	0,001396	0,94	0,384	
$m \cdot I^{\max} \cdot \alpha$	0,02000	0,01000	0,009309	1,07	0,324	
$m \cdot I^{\max} \cdot I$	-0,00200	-0,00100	0,000931	-1,07	0,324	
$m \cdot T \cdot \alpha$	-0,05813	-0,02906	0,013963	-2,08	0,083	
$m \cdot T \cdot I$	0,00019	0,00009	0,001396	0,07	0,949	
$m \cdot \alpha \cdot I$	-0,00625	-0,00313	0,009309	-0,34	0,749	
$I^{\max} \cdot T \cdot \alpha$	-0,03938	-0,01969	0,013963	-1,41	0,208	
$I^{\max} \cdot T \cdot I$	-0,00169	-0,00084	0,001396	-0,60	0,568	
$I^{\max} \cdot \alpha \cdot I$	0,02375	0,01187	0,009309	1,28	0,249	
$T \cdot \alpha \cdot I$	-0,00750	-0,00375	0,013963	-0,27	0,797	

Çözüm İçin Varyans Analizi

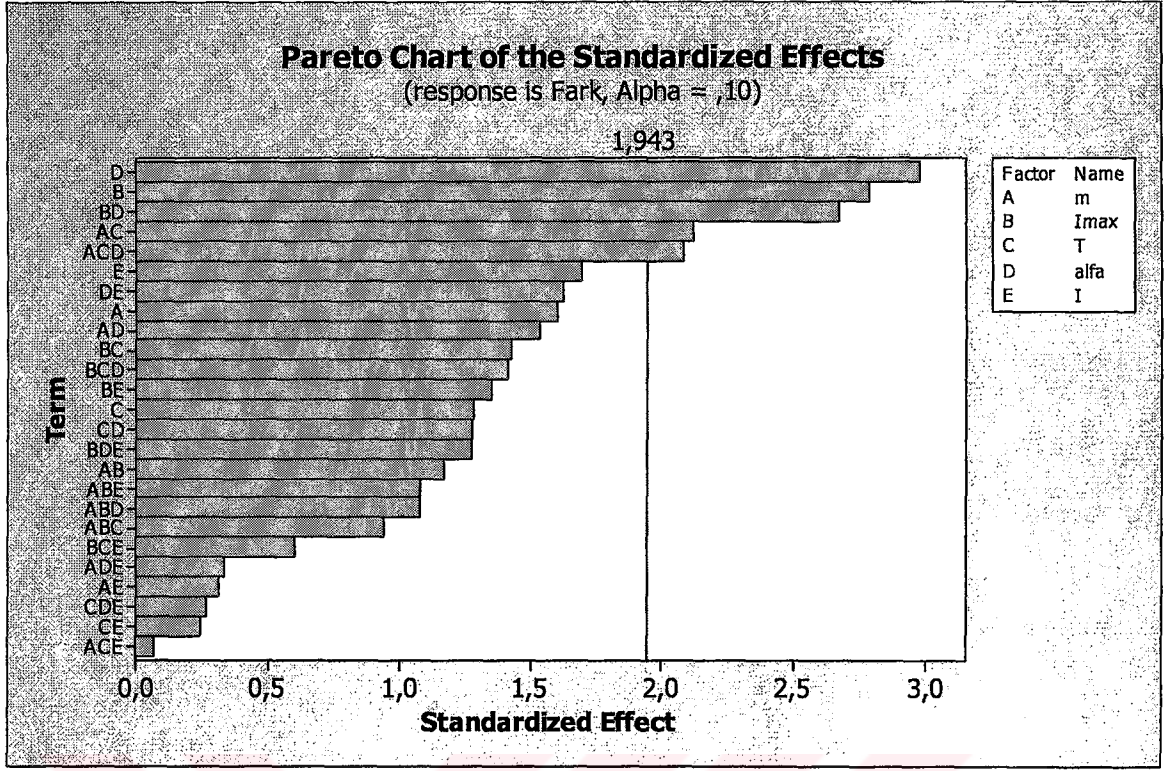
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,0007803	0,0006115	0,00012230	4,41	0,049
İkili Etkileşimler	10	0,0006090	0,0005532	0,00005532	2,00	0,206
Üçlü Etkileşimler	10	0,0003242	0,0003242	0,00003242	1,17	0,443
Kalan Hata	6	0,0001664	0,0001664	0,00002773		
Toplam	31	0,0018799				



Şekil 7.1 NUG20 problemi için normal olasılık eğrisi

Şekil 7.1’de görüldüğü gibi “ $T^{\max}$ ”, “ α ”, “ $T^{\max} - \alpha$ ”, “ $m-T$ ” ve “ $m-T - \alpha$ ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.2’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “ $T^{\max}$ ”, “ α ”, “ $T^{\max} - \alpha$ ”, “ $m-T$ ” ve “ $m-T - \alpha$ ” etkileşimleri olduğu görülmektedir.

Bu problem için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.



Şekil 7.2 NUG20 problemi için pareto grafiği

Çizelge 7.5 NUG20 indirgenmiş tasarımı

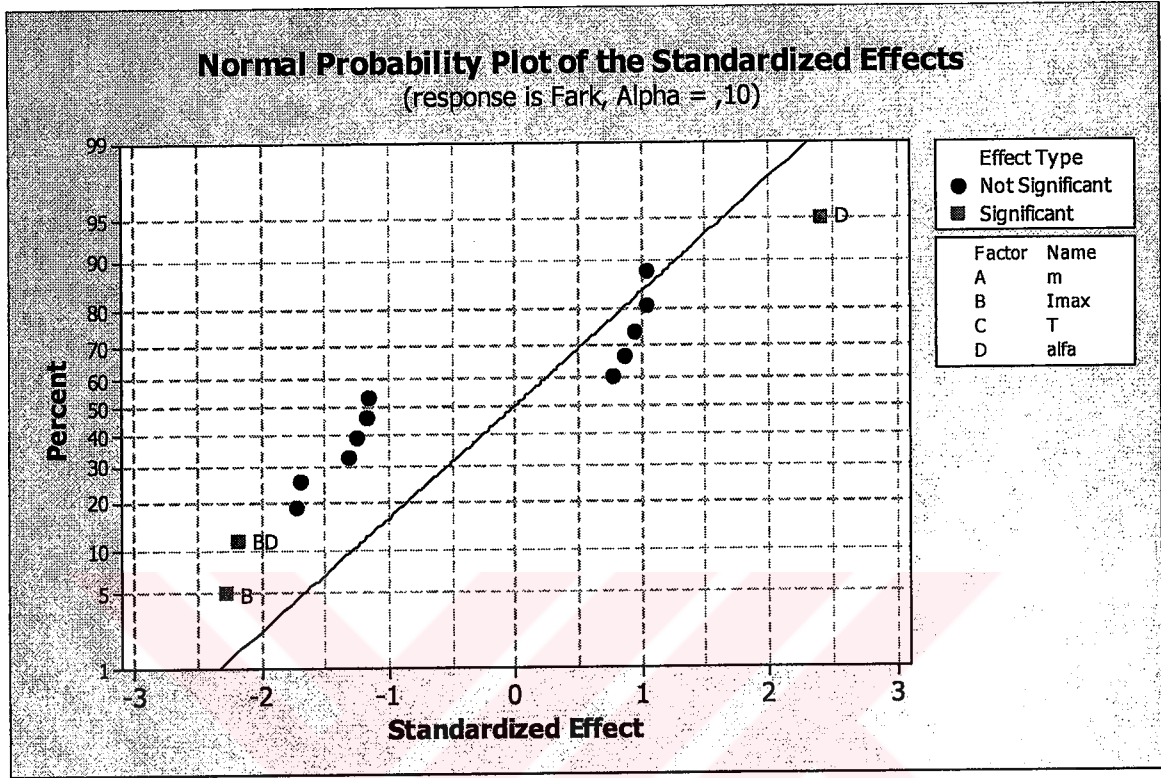
Fark İçin Tahmin Edilen Etki ve Katsayılar

Terim	Etki	Katsayı	SE Katsayı	T	P
Sabit		0,08709	0,034489	2,53	0,022
m	-0,09019	-0,04509	0,034489	-1,31	0,208
Imax	-0,15669	-0,07834	0,034489	-2,27	0,036
T	0,09694	0,04847	0,046272	1,05	0,310
alfa	0,06187	0,03094	0,012765	2,42	0,027
m*Imax	0,05875	0,02938	0,030853	0,95	0,354
m*T	-0,15994	-0,07997	0,046272	-1,73	0,102
m*alfa	-0,03188	-0,01594	0,012765	-1,25	0,229
Imax*T	-0,10744	-0,05372	0,046272	-1,16	0,262
Imax*alfa	-0,05562	-0,02781	0,012765	-2,18	0,044
T*alfa	0,03562	0,01781	0,017126	1,04	0,313
m*Imax*T	0,00262	0,00131	0,001713	0,77	0,454
m*Imax*alfa	0,02000	0,01000	0,011417	0,88	0,393
m*T*alfa	-0,05812	-0,02906	0,017126	-1,70	0,108
Imax*T*alfa	-0,03938	-0,01969	0,017126	-1,15	0,266

Çözüm için Varyans Analizi

Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	4	0,0005802	0,0005316	0,00013291	3,19	0,040
İkili Etkileşimler	6	0,0003588	0,0004192	0,00006987	1,67	0,188
Üçlü Etkileşimler	4	0,0002317	0,0002317	0,00005794	1,39	0,280
Kalan Hata	17	0,0007091	0,0007091	0,00004171		
Toplam	31	0,0018799				

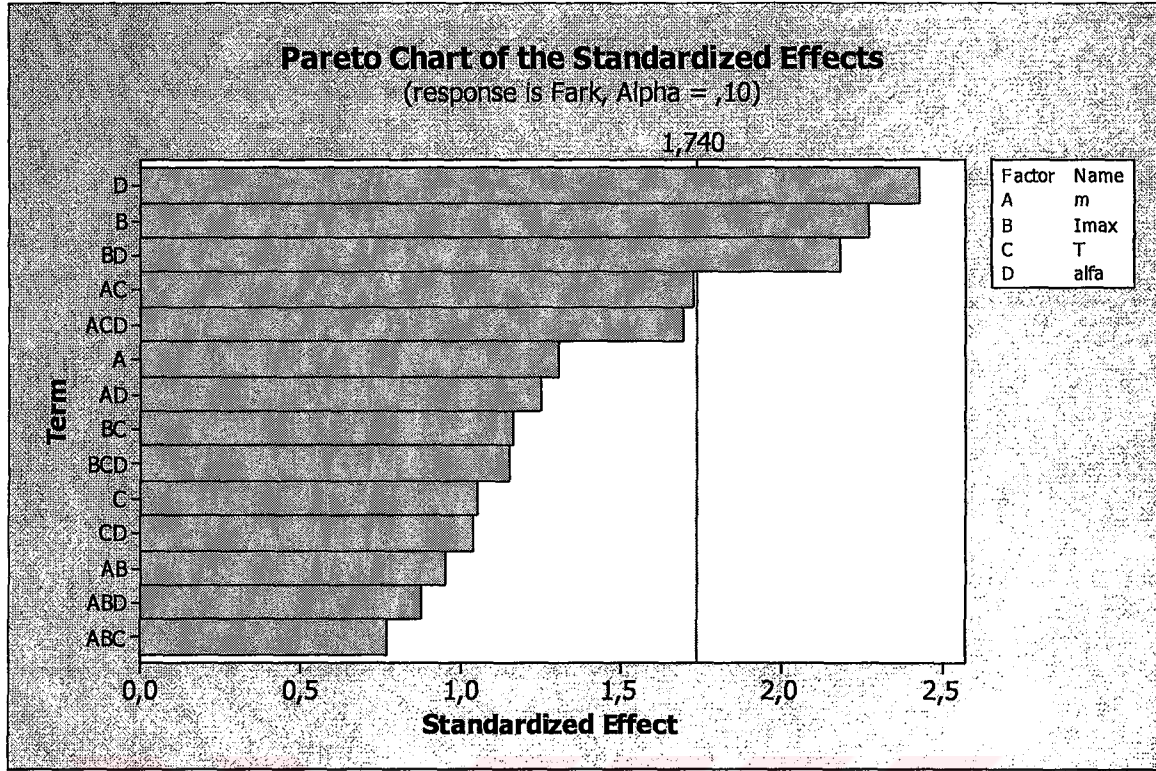
Çizelge 7.5’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla model ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



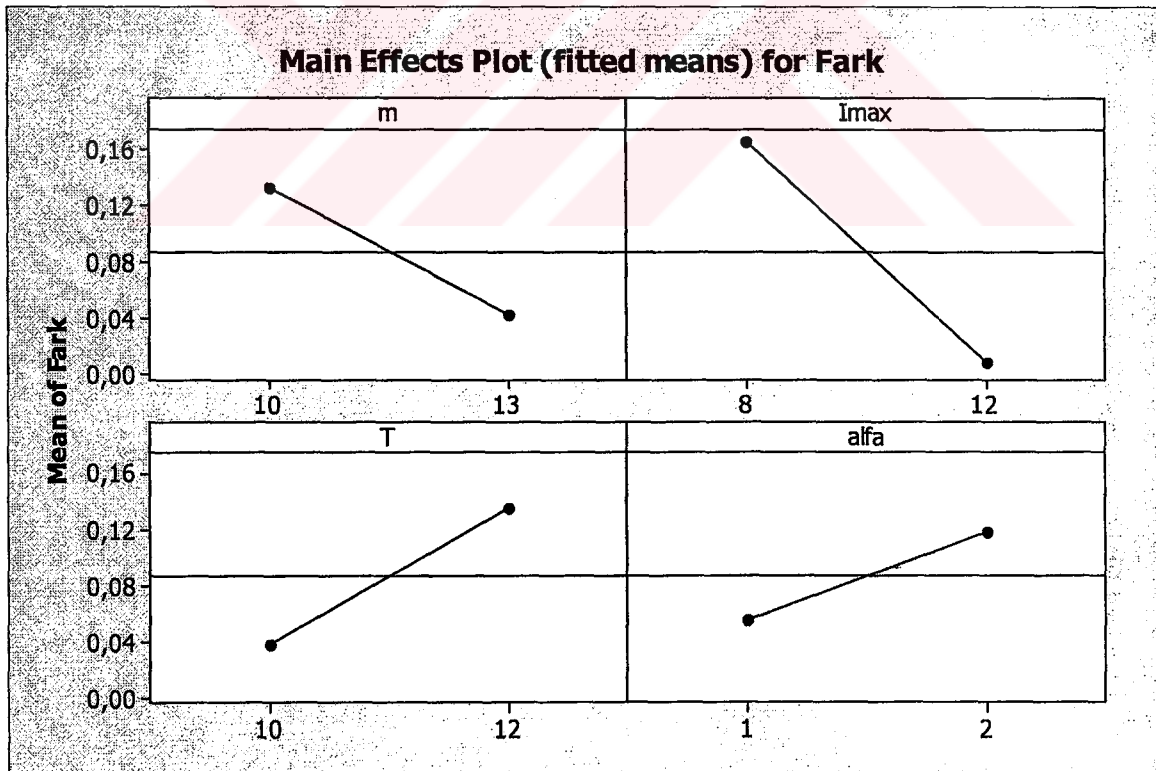
Şekil 7.3 NUG20 problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.3’deki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.

Yine Şekil 7.4’de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.4 NUG20 problemi için etkin parametrelerle histogram grafiği



Şekil 7.5 NUG20 probleminde çözüm kalitesi için ana etkiler

Şekil 7.5 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana

etkiye sahip olan parametre iterasyon sayısı $I^{\max}$ 'dır. Çünkü iterasyon sayısının iki düzeyini yani 8 ve 12 değerlerini birleştiren eğri en dik eğridir. Bu diklik ise ana etkinin ne kadar büyük olduğunu ifade etmektedir. Diğer parametreler bu parametre kadar büyük ana etkiye sahip değildir.

Sonuç olarak, karınca sayısı "m" 13, iterasyon sayısı " $I^{\max}$ " 12, takas sayısı "T" 10 ve " α " 0,1 olarak ele alınırsa çözüm kalitesi artacaktır. Yerel arama prosedürünün iterasyon sayısı "I" 100 ve 200 seçilmesinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu sebepten dolayı en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Nug20 problemi için parametre düzeyleri Çizelge 7.6'da verilmiştir.

Çizelge 7.6 NUG20 problemi için parametre düzeyleri

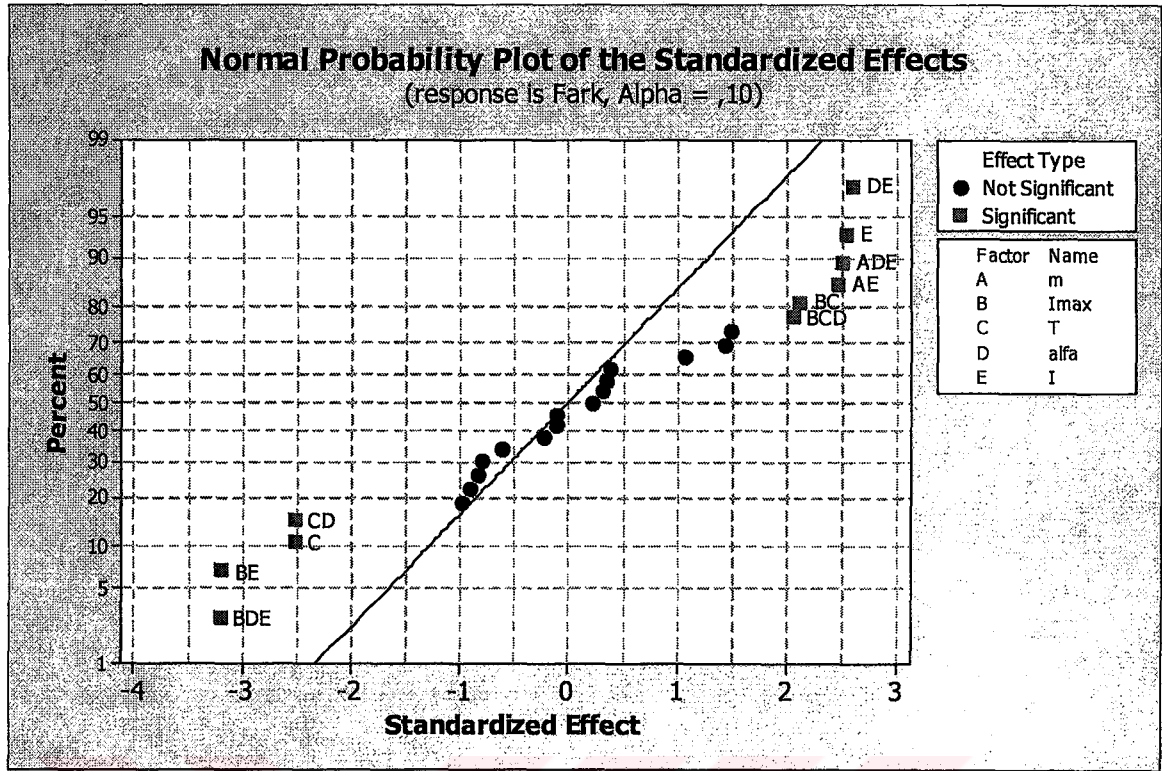
Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	13
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($I^{\max}$)	12
İzin Buharlaştırmadan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.2 NUG30 Tasarımı

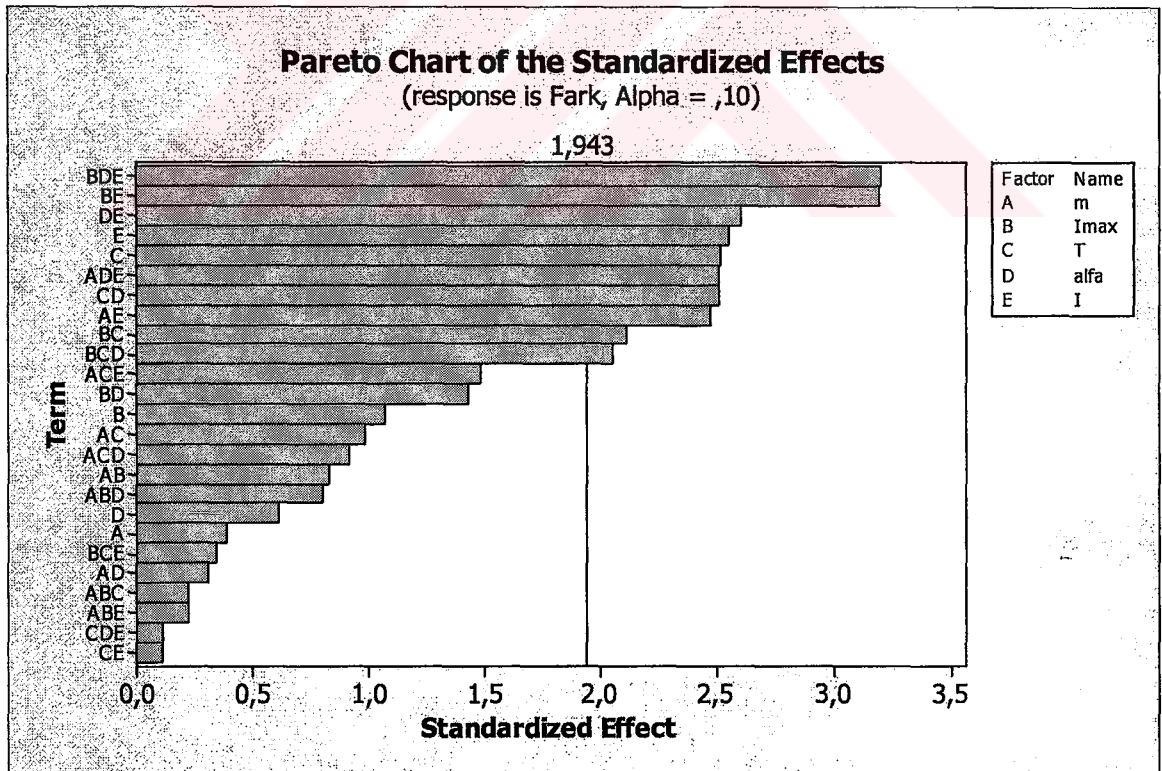
Anlamlılık düzeyi $\alpha = 0.1$ dikkate alınarak yapılan tasarım için Çizelge 7.7'deki istatistik değerleri incelendiğinde, yerel arama prosedürün iterasyon sayısı "I" ve takas sayısı "T" parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkisi tek başına o kadar etkin olmayan parametreler, ana etkisi tek başına etkili olan parametrelerle bir araya geldiğinde çözüm kalitesini arttırdıkları görülmektedir. Bu ikili etkileşimler " $m-I$ ", " $I^{\max}-T$ ", " $I^{\max}-I$ ", " $T-\alpha$ " ve " $\alpha-I$ " dir. Üçlü ortak etkilerden ise " $m-\alpha-I$ ", " $I^{\max}-T-\alpha$ " ve " $I^{\max}-I-\alpha$ "nın çözüm üzerindeki etkisi anlamlı bulunmuştur.

Çizelge 7.7 NUG30 tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		0,00000	0,016531	0,00	1,000	
m	0,01275	0,00638	0,016531	0,39	0,713	
Imax	0,03550	0,01775	0,016531	1,07	0,324	
T	0,11175	-0,05587	0,022179	-2,52	0,045	
alfa	-0,00750	-0,00375	0,006118	-0,61	0,562	
m*Imax	-0,02437	-0,01219	0,014788	-0,82	0,441	
m*T	-0,04350	-0,02175	0,022179	-0,98	0,365	
m*alfa	0,00375	0,00188	0,006118	0,31	0,770	
Imax*T	0,09375	0,04687	0,022179	2,11	0,079	
Imax*alfa	0,01750	0,00875	0,006118	1,43	0,203	
T*Imax	-0,09431	-0,04716	0,014788	-3,19	0,019	
T*alfa	-0,04125	-0,02062	0,008209	-2,51	0,046	
T*I	-0,00487	-0,00244	0,000821	-0,11	0,916	
alfa*I	0,03188	0,01594	0,006118	2,60	0,040	
m*Imax*T	-0,00037	-0,00019	0,000821	-0,23	0,827	
m*Imax*alfa	-0,00875	-0,00437	0,005472	-0,80	0,455	
m*Imax*I	0,00025	0,00012	0,000547	0,23	0,827	
m*T*alfa	-0,01500	-0,00750	0,008209	-0,91	0,396	
m*T*I	0,00244	0,00122	0,000821	1,48	0,188	
T*Imax*alfa	0,02750	0,01375	0,005472	2,51	0,046	
Imax*T*alfa	0,03375	0,01687	0,008209	2,06	0,086	
Imax*T*I	0,00056	0,00028	0,000821	0,34	0,744	
Imax*alfa*I	-0,03500	-0,01750	0,005472	-3,20	0,019	
T*alfa*I	-0,00187	-0,00094	0,0008209	-0,11	0,913	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,00140838	0,00013886	0,00002777	2,90	0,114
İkili Etkileşimler	10	0,00023850	0,00036929	0,00003693	3,85	0,056
Üçlü Etkileşimler	10	0,00023650	0,00023650	0,00002365	2,47	0,141
Kalan Hata	6	0,00005750	0,00005750	0,00000958		
Toplam	31	0,00194088				



Şekil 7.6 NUG30 problemi için normal olasılık eğrisi

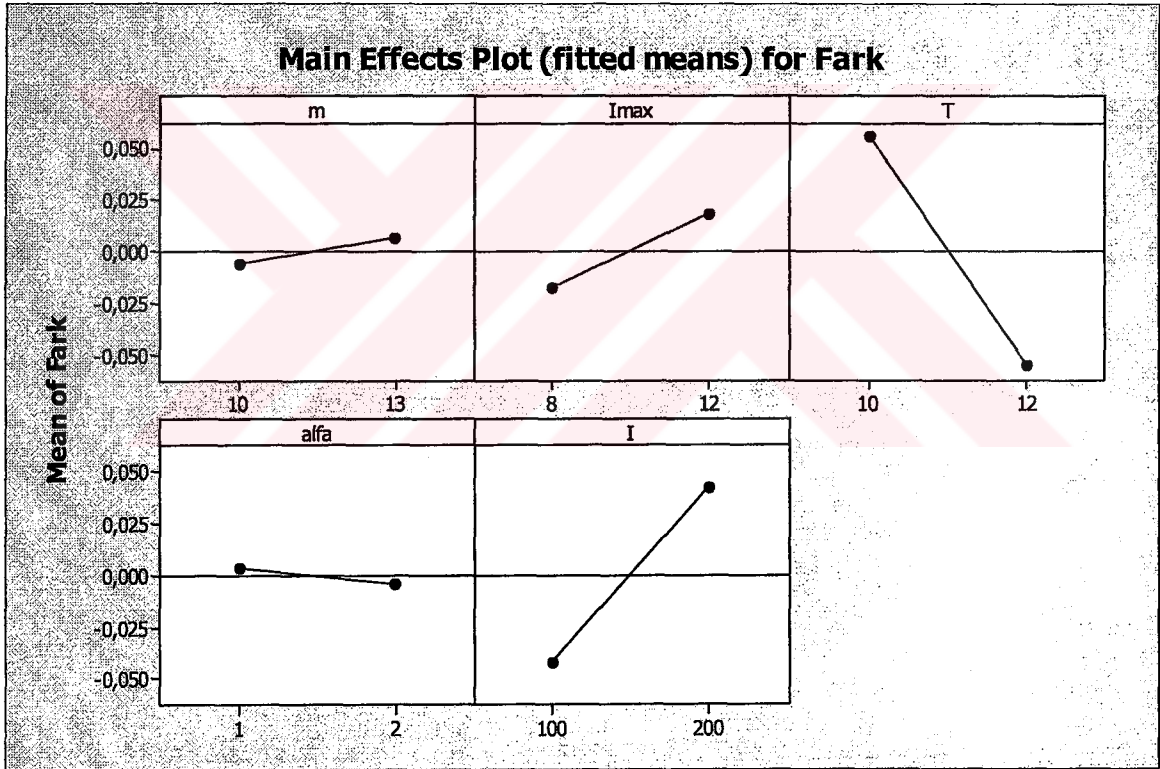


Şekil 7.7 NUG30 problemi için pareto grafiği

Şekil 7.6’da görüldüğü gibi “I”, “T” ana etkilerinin, “m-I”, “I^{max}-T”, “I^{max}-I”, “T-α” ve “α-I” ikili ortak etkilerin ve “m-α-I”, “I^{max}-T-α” ve “I^{max}-I-α” normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.7’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “I”, “T”, “m-I”, “I^{max}-T”, “I^{max}-I”, “T-α”, “α-I”, “m-α-I”, “I^{max}-T-α” ve “I^{max}-I-α” etkileşimleri olduğu görülmektedir.

NUG30 için bütün parametrelerin etkin olması nedeniyle indirgenmiş tasarım yapılmamıştır.

Şekil 7.8 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametre yerel arama prosedürünün iterasyon sayısı (I) ve takas sayısıdır (T). Çünkü yerel arama prosedürünün iterasyon sayısının (I) iki düzeyini ve takas sayısının (T) iki düzeyini birleştiren eğriler en dik eğrilerdir. Diğer parametreler bu parametreler kadar büyük ana etkiye sahip değildir.



Şekil 7.8 NUG30 probleminde çözüm kalitesi için ana etkiler

Sonuç olarak, karınca sayısı “m” 10, iterayon sayısı “I^{max}” 8, takas sayısı “T” 12, “α” 0,2 ve yerel arama prosedürünün iterasyon sayısı “I” 100 olarak ele alınırsa çözüm kalitesi artacaktır.

Nug30 problemi için parametre düzeyleri Çizelge 7.8’de verilmiştir.

Çizelge 7.8 NUG30 problemi için parametre düzeyleri

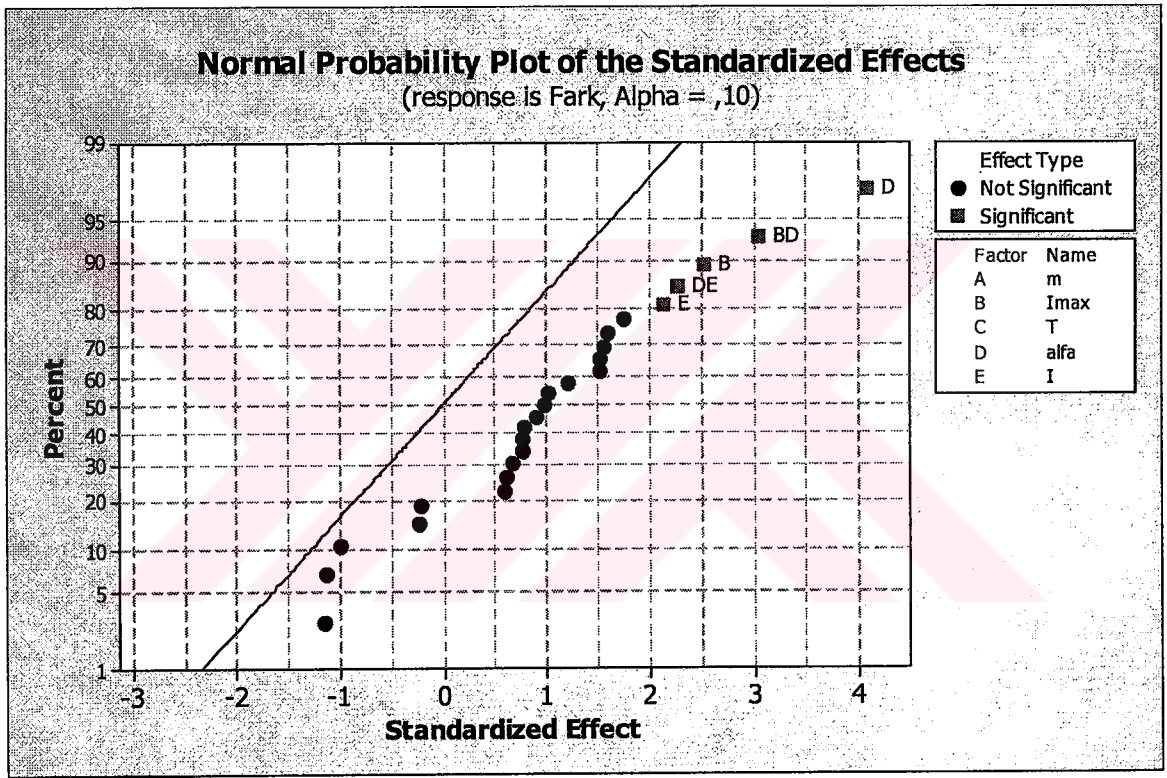
Parametre	Düzeyi
Takas Sayısı (T)	12
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($I^{\max}$)	8
İzin Buharlaştırmadan Kalan Kısmını İfade Eden Parametre (α)	0,2

7.3.2.3 TAI30a Tasarımı

Çizelge 7.9 TAI30a tasarımı

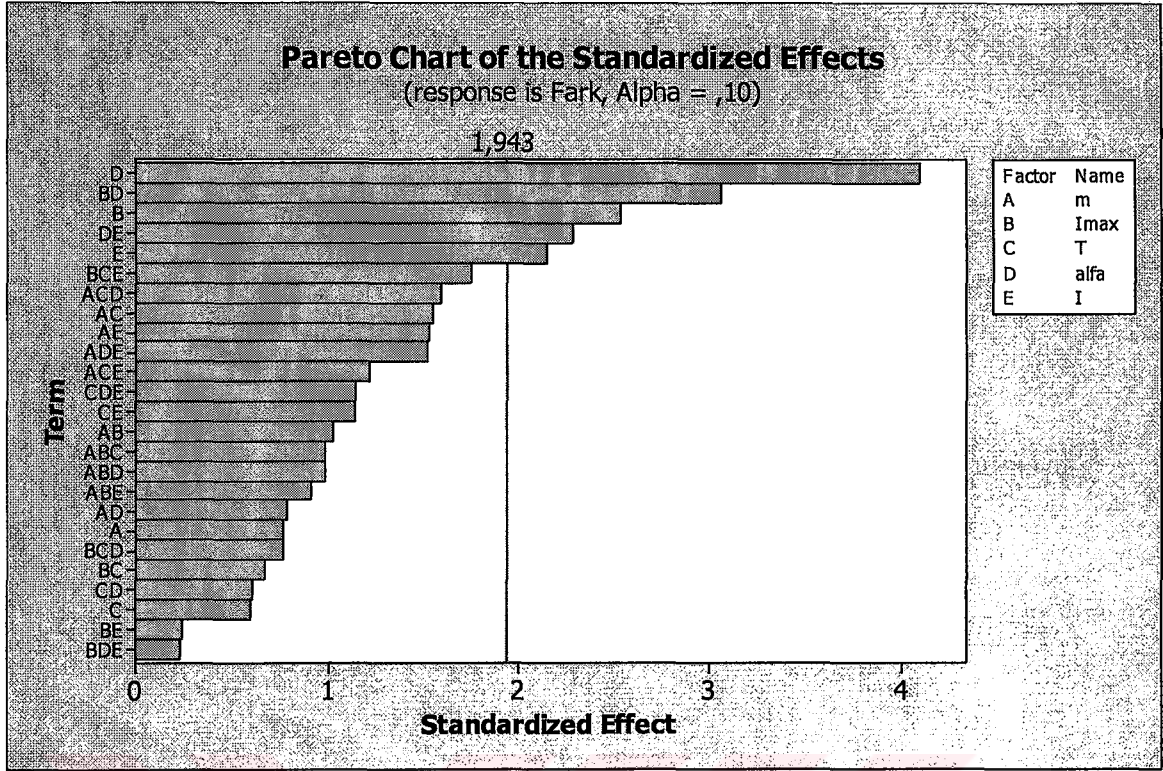
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Kaynak		0,12562	0,024785	5,07	0,002	
m	0,03800	0,01900	0,024785	0,77	0,472	
$I^{\max}$	0,12600	0,03200	0,024785	2,54	0,014	
T	0,03975	0,01988	0,033252	0,60	0,572	
α	0,07500	0,03750	0,009173	4,09	0,006	
I	0,10675	0,05338	0,024785	2,15	0,075	
m* $I^{\max}$	0,04569	0,02284	0,022172	1,03	0,343	
m*T	0,10350	0,05175	0,033252	1,56	0,171	
m* α	0,01438	0,00719	0,009173	0,78	0,463	
m*I	0,06800	0,03400	0,022172	1,53	0,176	
$I^{\max}$ *T	0,04500	0,02250	0,033252	0,68	0,524	
$I^{\max}$ * α	0,05625	0,02812	0,009173	3,07	0,022	
$I^{\max}$ *I	-0,01056	-0,00528	0,022172	-0,24	0,820	
T* α	0,01500	0,00750	0,012307	0,61	0,565	
T*I	-0,07575	-0,03788	0,033252	-1,14	0,298	
α *I	0,04188	0,02094	0,009173	2,28	0,063	
m* $I^{\max}$ *T	-0,00244	-0,00122	0,001231	-0,99	0,360	
m* $I^{\max}$ * α	0,01625	0,00812	0,008205	0,99	0,360	
m* $I^{\max}$ *I	0,00150	0,00075	0,000820	0,91	0,396	
m*T* α	0,03938	0,01969	0,012307	1,60	0,161	
m*T*I	0,00300	0,00150	0,001231	1,22	0,269	
$I^{\max}$ *T*I	0,00431	0,00216	0,001231	1,75	0,130	
T* α *I	-0,02813	-0,01406	0,012307	-1,14	0,297	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,0053273	0,0006524	0,00013049	6,06	0,024
İkili Etkileşimler	10	0,0006443	0,0005883	0,00005883	2,73	0,116
Üçlü Etkileşimler	10	0,0003052	0,0003052	0,00003052	1,42	0,347
Kalan Hata	6	0,0001293	0,0001293	0,00002154		
Toplam	31	0,0064060				

Anlamlılık düzeyi $\alpha = 0.1$ dikkate alınarak yapılan tasarım için Çizelge 7.9'daki istatistik değerleri incelendiğinde, Algoritmanın iterasyon sayısı " $I^{\max}$ ", yerel arama prosedürünün iterasyon sayısı " I " ve " α " parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olan " $I^{\max}$ " ve " α " parametreleri bir araya geldiğinde de çözüm kalitesi performansı üzerinde etkili olmaktadır. Bunun yanında ana etkisi tek başına o kadar etkin olmayan takas sayısı " T ", ana etkisi çözüm kalitesi üzerinde etkili olan " α " bir araya geldiğinde çözüm kalitesinin artmasında etkin olmaktadır. Üçlü ortak etkilerden ise hiçbir ortak etki anlamlı değildir.



Şekil 7.9 TAI30a problemi için normal olasılık eğrisi

Şekil 7.9'da görüldüğü gibi " $I^{\max}$ ", " α ", " I ", " $I^{\max} - \alpha$ " ve " $I - \alpha$ " etkili değerlerin, normal olasılık eğrisinin dışın kaldığı görülmektedir. Şekil 7.10'da ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, " $I^{\max}$ ", " α ", " I ", " $I^{\max} - \alpha$ " ve " $I - \alpha$ " etkileşimleri olduğu görülmektedir.



Şekil 7.10 TAI30a problemi için pareto grafiği

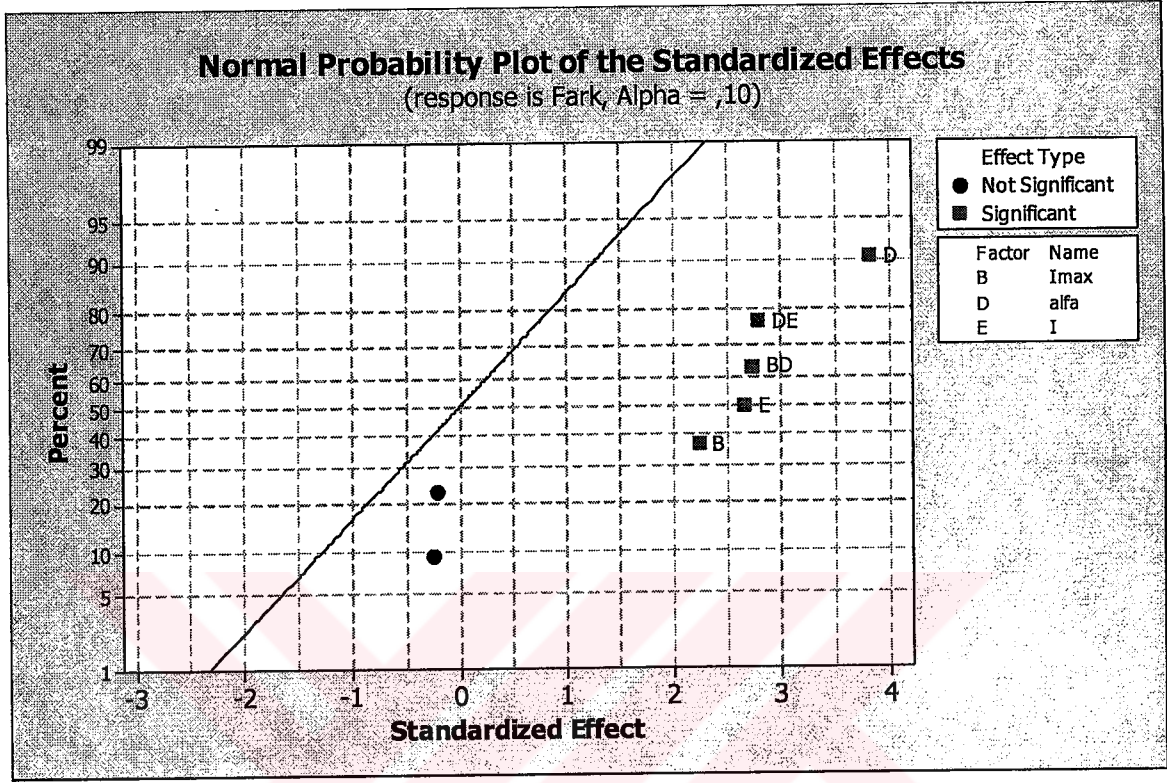
Bu problem için etkin parametreler belirlendikten sonra modelden etkin olmayan “m” ve “T” parametreleri çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Çizelge 7.10 TAI30a indirgenmiş tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE	Katsayı	T	P
Sabit		0,119000	0,024693		4,82	0,000
Imax	0,111000	0,055500	0,024693		2,25	0,034
alfa	0,070000	0,035000	0,009139		3,83	0,001
I	0,132000	0,066000	0,024693		2,67	0,013
Imax*alfa	0,050000	0,025000	0,009139		2,74	0,012
Imax*I	-0,012000	-0,006000	0,024693		-0,24	0,810
alfa*I	0,051250	0,025625	0,009139		2,80	0,010
Imax*alfa*I	-0,003750	-0,001875	0,009139		-0,21	0,839

Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	3	0,00532512	0,00071797	0,00023932	8,95	0,000
İkili Etkileşimler	3	0,00043825	0,00041170	0,00013723	5,13	0,007
Üçlü Etkileşimler	1	0,00000113	0,00000113	0,00000113	0,04	0,839
Kalan Hata	24	0,00064150	0,00064150	0,00002673		
Toplam	31	0,00640600				

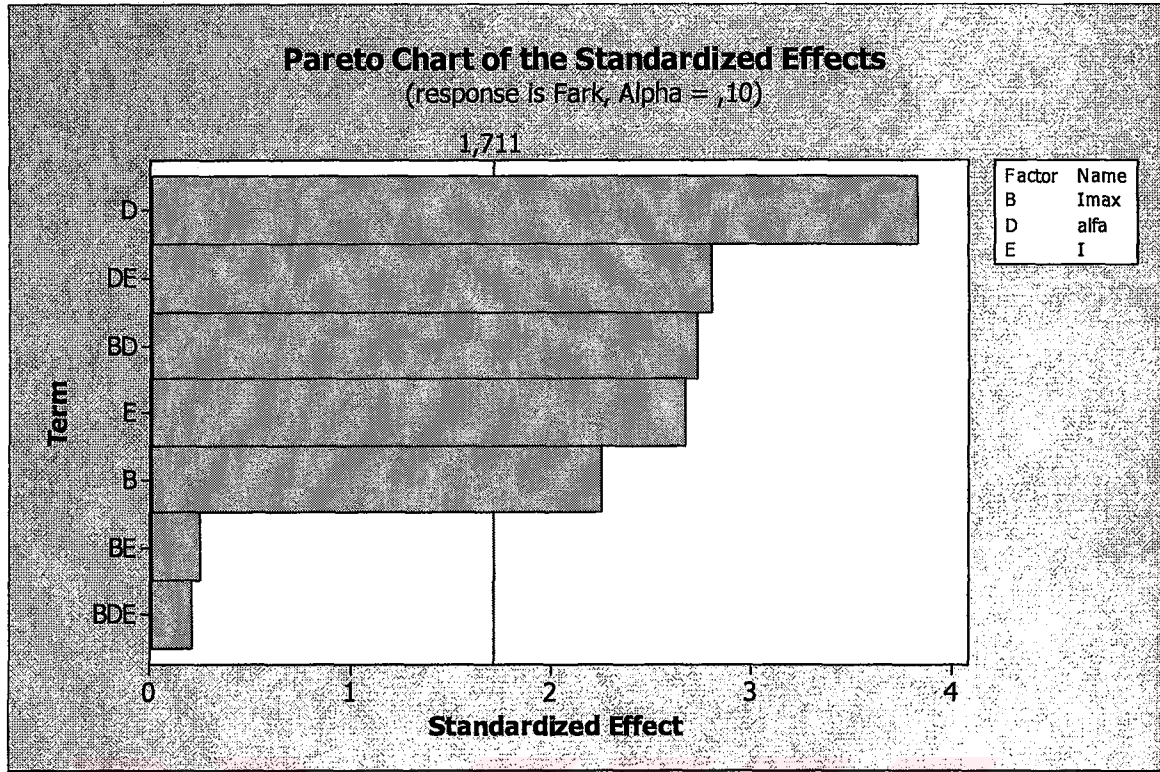
Çizelge 7.10'da indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Yani model ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



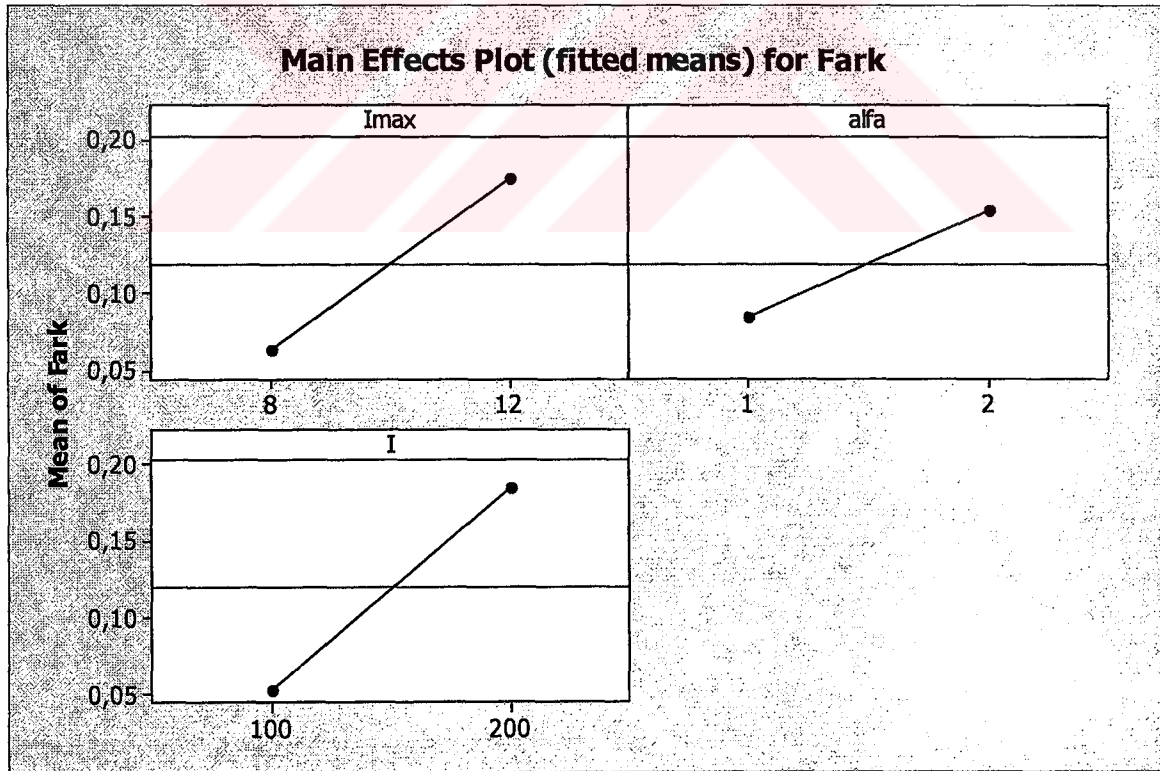
Şekil 7.11 TAI30a problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.11'deki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.

Yine Şekil 7.12'de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.12 TAI30a problemi için etkin parametrelerle histogram grafiği



Şekil 7.13 TAI30a probleminde çözüm kalitesi için ana etkiler

Şekil 7.13 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi bütün parametrelerin ana etkisi birbirine yakındır . Çünkü üç parametrenin düzeylerini birleştiren

eğrilerin diklikleri yaklaşık olarak aynıdır. Diğer parametreler bu parametre kadar büyük ana etkiye sahip değildir.

Sonuç olarak, karınca sayısı iterasyon sayısı " I^{max} " 8, yerel arama prosedürünün iterasyon sayısı " I " 100 ve " α " 0,1 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı " m " ve takas sayısı parametrelerinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu nedenle en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Tai30a problemi için parametre düzeyleri Çizelge 7.11'de verilmiştir.

Çizelge 7.11 TAI30a problemi için parametre düzeyleri

Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı (I^{max})	8
İzin Buharlaşımdan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.4 TAI60a Tasarımı

Çizelge 7.12'e bakıldığında takas sayısı " T " ve " α " parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olan " T " ve " α " parametreleri bir araya geldiğinde de çözüm kalitesi performansı üzerinde etkili olmaktadır. Bunun yanında ana etkisi tek başına o kadar etkin olmayan algoritmanın iterasyon sayısı " I^{max} " " α " parametresi ile bir araya geldiğinde çözüm kalitesinin artmasında etkin olmaktadır. Üçlü ortak etkilerden hiçbiri anlamlı bulunmamıştır.

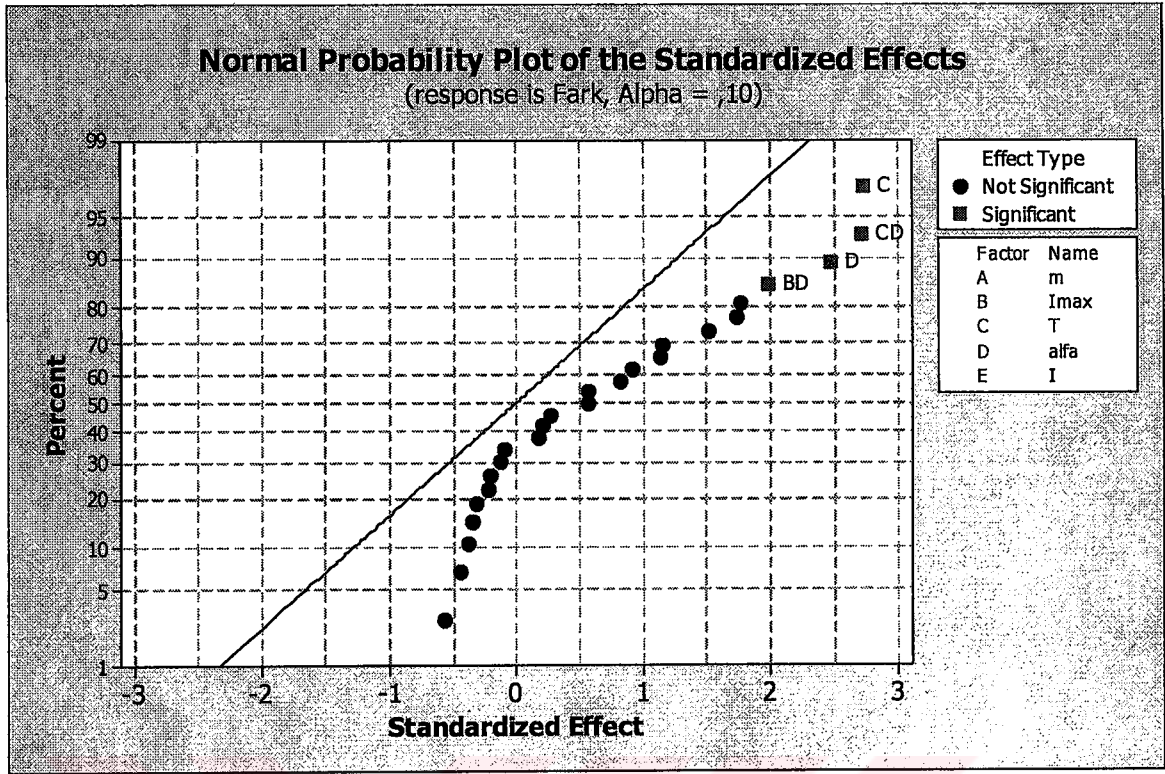
Çizelge 7.12 TAI 60a tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar

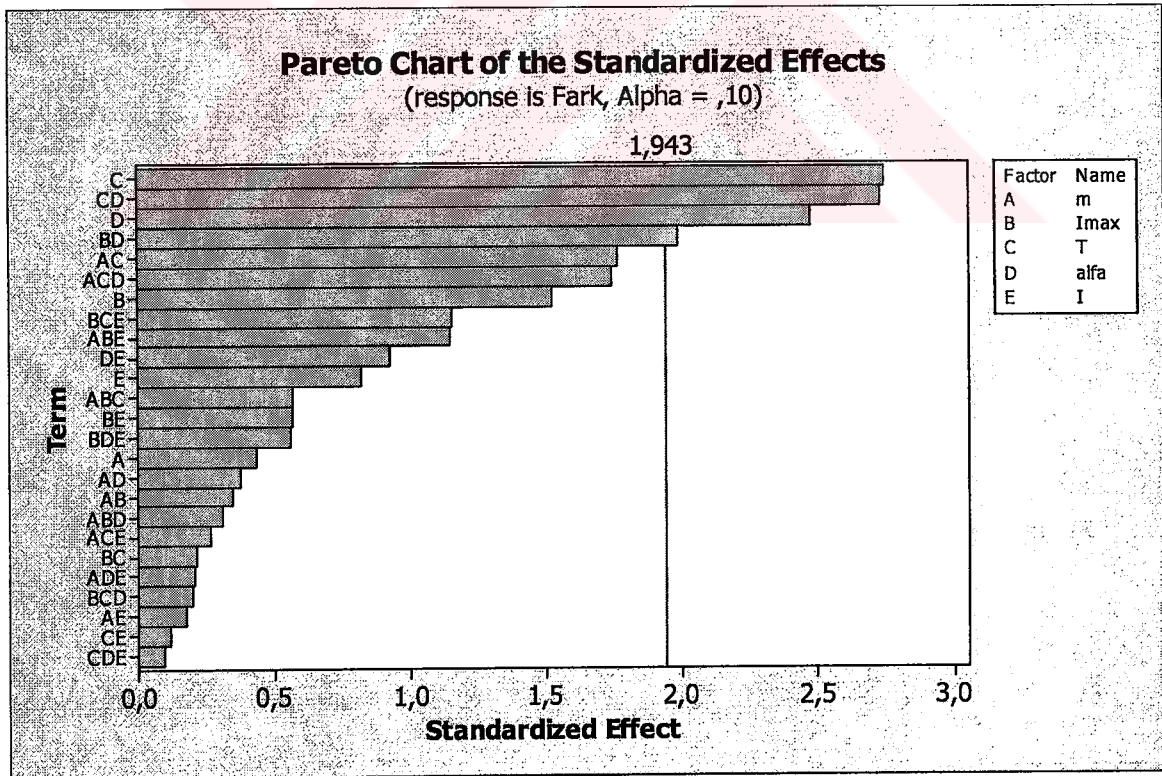
Terim	Etki	Katsayı	SE	Katsayı	T	P
Sabit		4,5434	1,39348		3,26	0,017
m	-1,2155	-0,6078	1,39348		-0,44	0,678
Imax	4,2544	2,1272	1,39348		1,53	0,178
I	2,2905	1,1452	1,39348		0,82	0,443
m*Imax	-0,8619	-0,4310	1,24658		-0,35	0,741
m*T	6,6068	3,3034	1,86955		1,77	0,128
m*alfa	-0,3894	-0,1947	0,51575		-0,38	0,719
m*I	0,4411	0,2206	1,24658		0,18	0,865
Imax*T	-0,8029	-0,4014	1,86955		-0,21	0,837
Imax*alfa	2,0512	1,0256	0,51575		1,99	0,094
Imax*I	1,4141	0,7070	1,24658		0,57	0,591
T*alfa	3,7763	1,8881	0,69195		2,73	0,034
T*I	-0,4537	-0,2269	1,86955		-0,12	0,907
alfa*I	0,9506	0,4753	0,51575		0,92	0,392
m*Imax*T	-0,0793	-0,0397	0,06920		-0,57	0,587
m*Imax*alfa	-0,2888	-0,1444	0,46130		-0,31	0,765
m*Imax*I	0,1058	0,0529	0,04613		1,15	0,295
m*T*alfa	2,4131	1,2066	0,69195		1,74	0,132
m*T*I	0,0371	0,0186	0,06920		0,27	0,797
m*alfa*I	0,1925	0,0962	0,46130		0,21	0,842
Imax*T*alfa	-0,2812	-0,1406	0,69195		-0,20	0,846
Imax*T*I	0,1592	0,0796	0,06920		1,15	0,294
Imax*alfa*I	0,5187	0,2594	0,46130		0,56	0,594
T*alfa*I	-0,1294	-0,0647	0,69195		-0,09	0,929

Çözüm İçin Varyans Analizi

Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	13,8070	0,8633	0,17267	2,54	0,144
İkili Etkileşimler	10	1,2471	1,3226	0,13226	1,94	0,215
Üçlü Etkileşimler	10	0,4484	0,4484	0,04484	0,66	0,734
Kalan Hata	6	0,4086	0,4086	0,06810		
Toplam	31	15,9111				



Şekil 7.14 TAI60a problemi için normal olasılık eğrisi



Şekil 7.15 TAI60a problemi için pareto grafiği

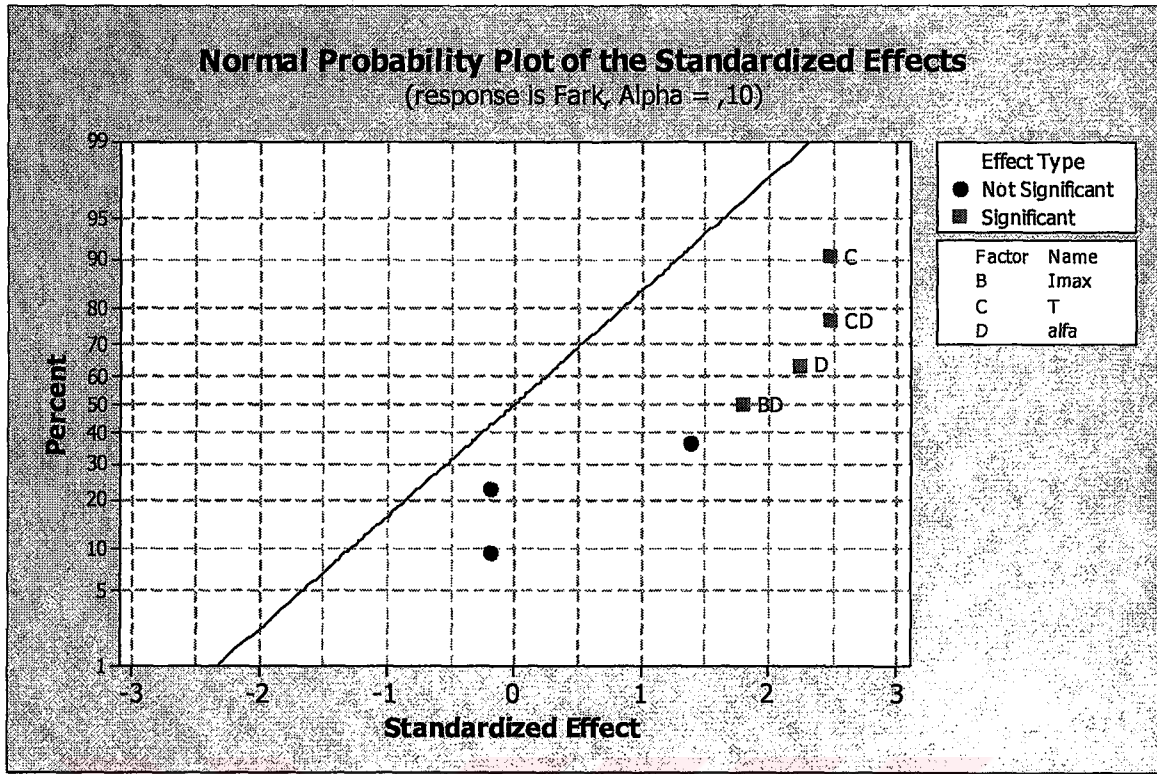
Şekil 7.14’de görüldüğü gibi “ $I^{\max}$ ”, “ α ”, “ $I^{\max} - \alpha$ ”, ve “ $T - \alpha$ ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.15’de ise pareto grafiğinde, çizginin dışında kalan birli ve ikili etkileşimlerin tabloda verilen, “ $I^{\max}$ ”, “ α ”, “ $I^{\max} - \alpha$ ”, ve “ $T - \alpha$ ” etkileşimleri olduğu görülmektedir.

Tai60a için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Çizelge 7.13 TAI60a indirgenmiş tasarımı

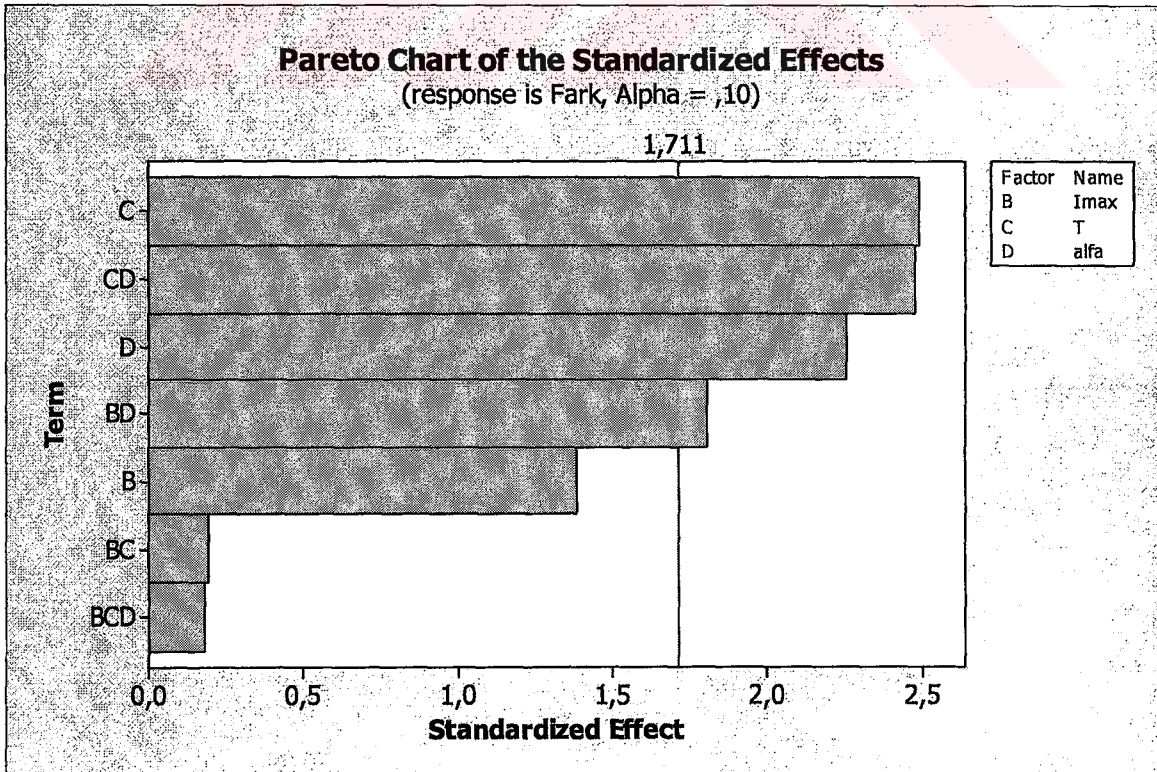
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		4,5434	1,5371	2,96	0,007	
Imax	4,2544	2,1272	1,5371	1,38	0,179	
α	0,2480	0,1240	0,0622	2,48	0,020	
$I^{\max} - \alpha$	2,5587	1,0794	0,5689	2,25	0,034	
Imax*T	-0,8029	-0,4014	0,0622	-0,19	0,847	
Imax*alfa	2,0513	1,0256	0,5689	1,80	0,084	
T*alfa	3,7752	1,8881	0,7633	2,47	0,021	
Imax*T*alfa	-0,2812	-0,1406	0,7633	-0,18	0,855	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	3	13,0379	0,80438	0,268127	3,24	0,040
İkili Etkileşimler	3	0,8818	0,87992	0,293306	3,54	0,030
Üçlü Etkileşimler	1	0,0028	0,00281	0,002812	0,03	0,855
Kalan Hata	24	1,9886	1,98855	0,082856		
Toplam	31	15,9111				

Çizelge 7.13’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla bu model ilerideki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



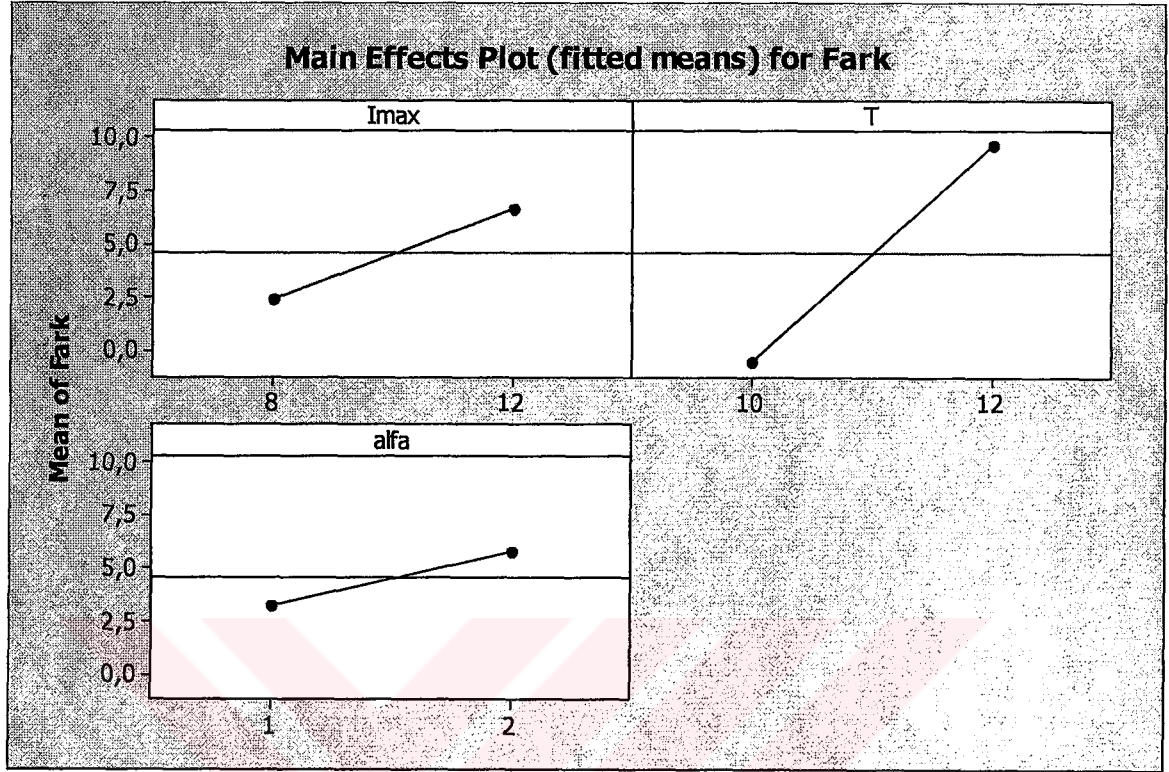
Şekil 7.16 TAI60a problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.16'daki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.



Şekil 7.17 TAI60a problemi için etkin parametrelerle histogram grafiği

Yine Şekil 7.17’de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.18 TAI60a probleminde çözüm kalitesi için ana etkiler

Şekil 7.18 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametre takas sayısı T’dir. Çünkü takas sayısının iki düzeyini yani 10 ve 12 değerlerini birleştiren eğri en dik egridir. Diğer parametreler bu parametre kadar büyük ana etkiye sahip değildir.

Sonuç olarak, algoritmanın iterasyon sayısı “ $I^{\max}$ ” 8, takas sayısı “T” 10 ve “ α ” 0,1 olarak ele alınırsa çözüm kalitesi artacaktır. Yerel arama prosedürünün iterasyon sayısı “I” ve karınca sayısı “m” parametrelerinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu sebepten dolayı yine en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Tai60a problemi için parametre düzeyleri Çizelge 7.14’de verilmiştir.

Çizelge 7.14 Tai60a problemi için parametre düzeyleri

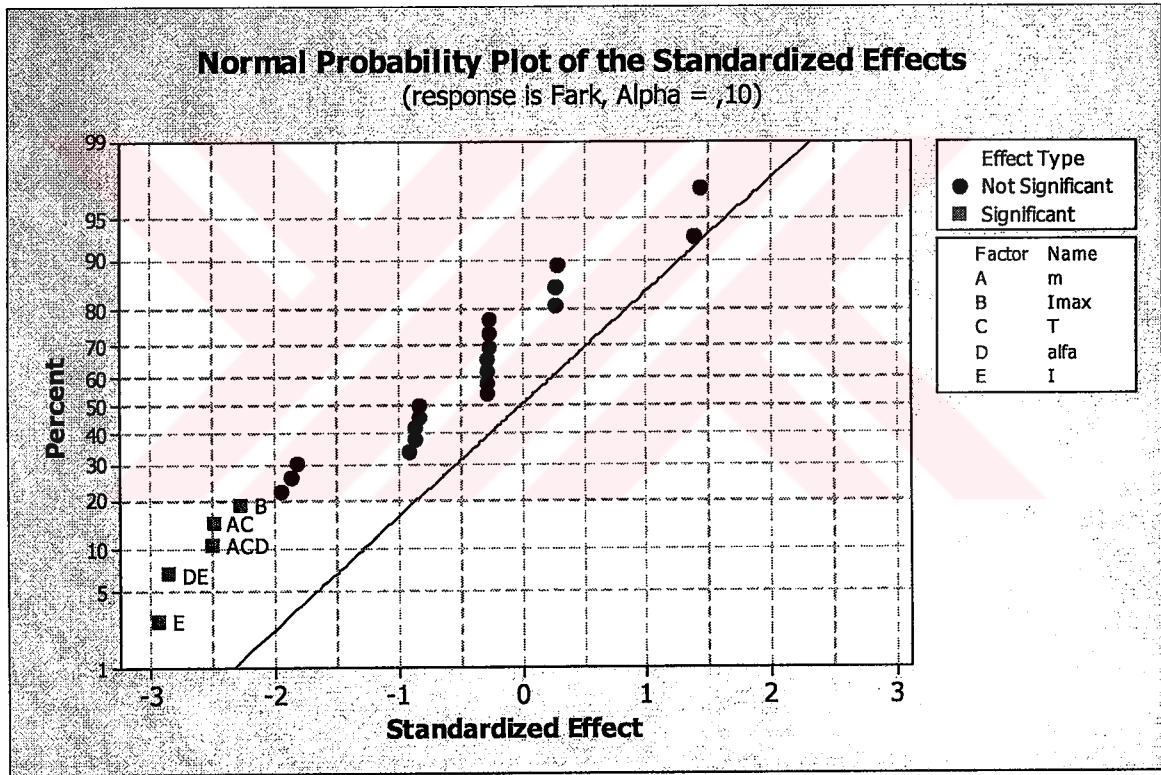
Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($I^{\max}$)	8
İzin Buharlaşmadan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.5 SKO64 Tasarımı

Çizelge 7.15 SKO64 tasarımı

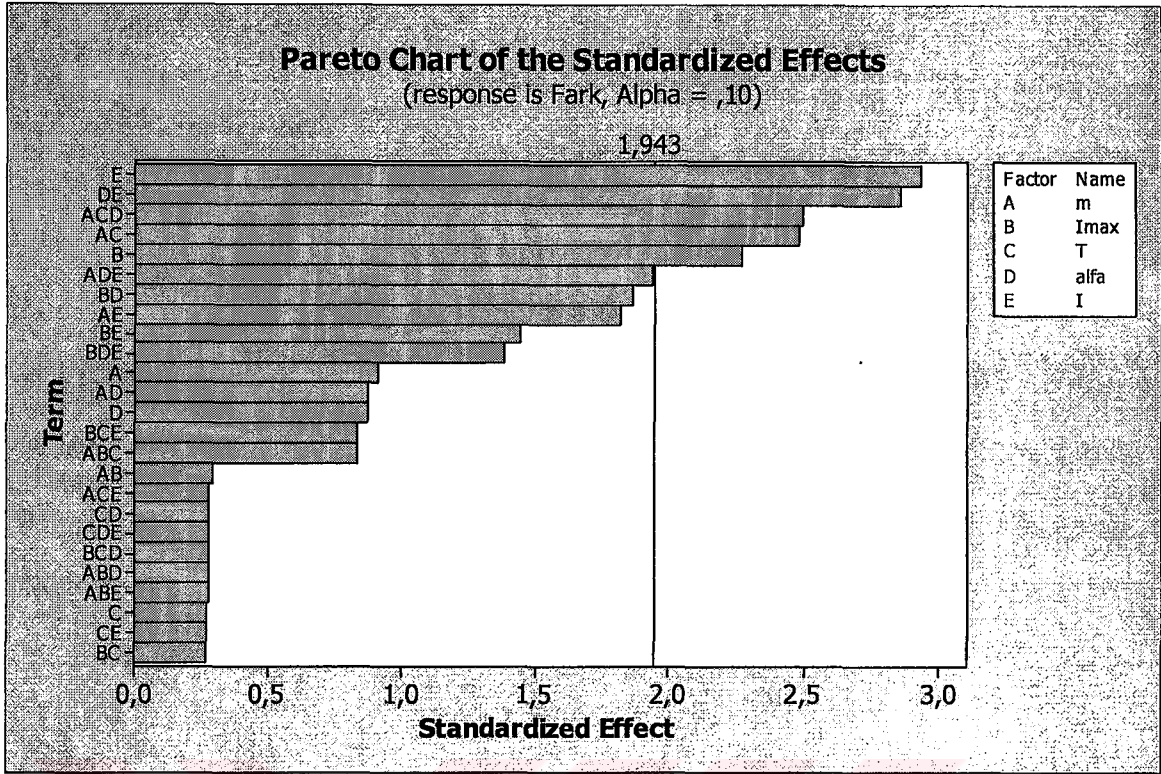
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE	Katsayı	T	P
Sabit		-0,00047	0,003404		-0,14	0,895
m	-0,00619	-0,00309	0,003404		-0,91	0,398
Imax	-0,01544	-0,00772	0,003404		-2,27	0,064
T	-0,00244	-0,00122	0,004566		-0,27	0,798
alfa	-0,00219	-0,00109	0,001260		-0,87	0,419
I	-0,01994	-0,00997	0,003404		-2,93	0,026
m*Imax	0,00178	0,00089	0,003045		0,29	0,780
m*T	-0,00269	-0,00134	0,004566		-2,48	0,048
m*alfa	-0,00219	-0,00109	0,001260		-0,87	0,419
m*I	-0,01109	-0,00555	0,003045		-1,82	0,118
Imax*T	-0,00244	-0,00122	0,004566		-0,27	0,798
Imax*I	0,00878	0,00439	0,003045		1,44	0,199
T*alfa	-0,00094	-0,00047	0,001690		-0,28	0,791
T*I	-0,00244	-0,00122	0,004566		-0,27	0,798
alfa*I	-0,00719	-0,00359	0,001260		-2,85	0,029
m*Imax*T	-0,00028	-0,00014	0,000169		-0,83	0,437
m*Imax*alfa	0,00062	0,00031	0,001127		0,28	0,791
m*Imax*I	-0,00006	-0,00003	0,000113		-0,28	0,791
m*T*alfa	-0,00344	-0,00172	0,001690		-2,50	0,047
m*T*I	0,00009	0,00005	0,000169		0,28	0,791
m*alfa*I	-0,00438	-0,00219	0,001127		-1,94	0,100
Imax*T*I	-0,00028	-0,00014	0,000169		-0,83	0,437
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,00006691	0,00000622	0,00000124	3,06	0,103
İkili Etkileşimler	10	0,00001106	0,00001019	0,00000102	2,51	0,136
Üçlü Etkileşimler	10	0,00000556	0,00000556	0,00000056	1,37	0,364
Kalan Hata	6	0,00000244	0,00000244	0,00000041		
Toplam	31	0,00008597				

Çizelge 7.15’ de görüldüğü gibi algoritmanın iterasyon sayısı “ $T^{\max}$ ” ve yerel arama prosedürü iterasyon sayısı “ T ” parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olmayan takas sayısı “ T ” ve karınca sayısı “ m ” parametrelerinin bir araya geldiğinde çözüm kalitesi performansı üzerinde etkili olmaktadır. Bunun yanında ana etkisi tek başına o kadar etkin olmayan algoritmanın iterasyon sayısı “ α ” parametresi etkin “ T ” parametresi ile bir araya geldiğinde çözüm kalitesinin artmasında etkin olmaktadır. Üçlü ortak etkilerden ise ikili ortak etkileri çözüm kalitesi performansı üzerinde etkili olan “ m ” ve “ T ” parametreleri “ α ” parametresi ile bir araya geldiğinde oluşan üçlü ortak etki anlamlı olmaktadır.



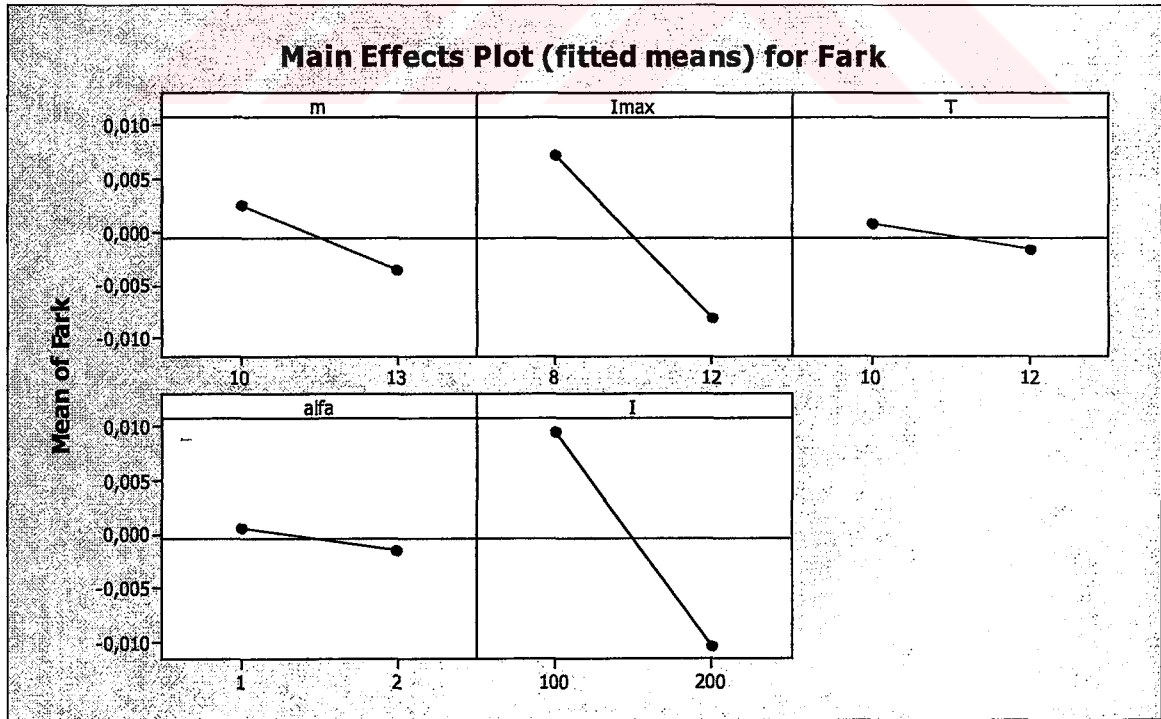
Şekil 7.19 SKO64 problemi için normal olasılık eğrisi

Şekil 7.19’da görüldüğü gibi “ $T^{\max}$ ”, “ T ”, “ $m-T$ ”, “ $I-\alpha$ ”, ve “ $m-T-\alpha$ ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.20’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “ $T^{\max}$ ”, “ T ”, “ $m-T$ ”, “ $I-\alpha$ ”, ve “ $m-T-\alpha$ ” etkileşimleri olduğu görülmektedir.



Şekil 7.20 SKO64 problemi için pareto grafiği

Sko64 için etkin parametrelerin hepsinin etkin olmasından dolayı indirgenmiş tasarım yapılmasına gerek kalmamıştır.



Şekil 7.21 SKO64 probleminde çözüm kalitesi için ana etkiler

Şekil 7.21 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametreler algoritmanın iterasyon sayısı ($I^{\max}$) ve yerel arama prosedürü iterasyon sayısı (I)'dır (algoritmanın iterasyon sayısı ($I^{\max}$) ve yerel arama prosedürü iterasyon sayısının (I) iki düzeyini birleştiren eğriler en dik eğrilerdir). Diğer parametreler bu parametreler kadar büyük ana etkiye sahip değildir.

Sonuç olarak, algoritmanın iterasyon sayısı " $I^{\max}$ " 12, takas sayısı "T" 12, yerel arama prosedürünün iterasyon sayısı "I" 200, karınca sayısı "m" 13 ve " α " 0,2 olarak ele alınırsa çözüm kalitesi artacaktır.

Sko64 problemi için parametre düzeyleri Çizelge 7.16'da verilmiştir.

Çizelge 7.16 Sko64 problemi için parametre düzeyleri

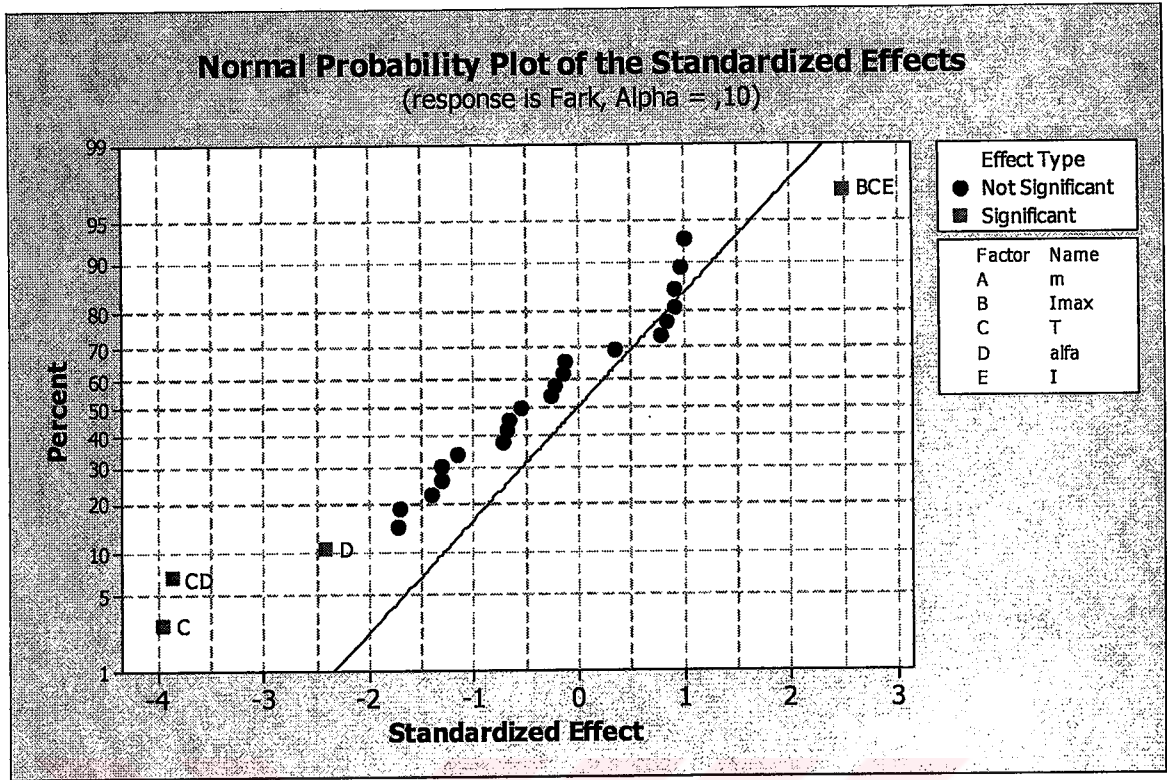
Parametre	Düzeyi
Takas Sayısı (T)	12
Karınca Sayısı (m)	13
Yerel Arama Prosedürünün İterasyon Sayısı (I)	200
Algoritmanın İterasyon sayısı ($I^{\max}$)	10
İzin Buharlaşmadan Kalan Kısmını İfade Eden Parametre (α)	0,2

7.3.2.6 BUR26a Tasarımı

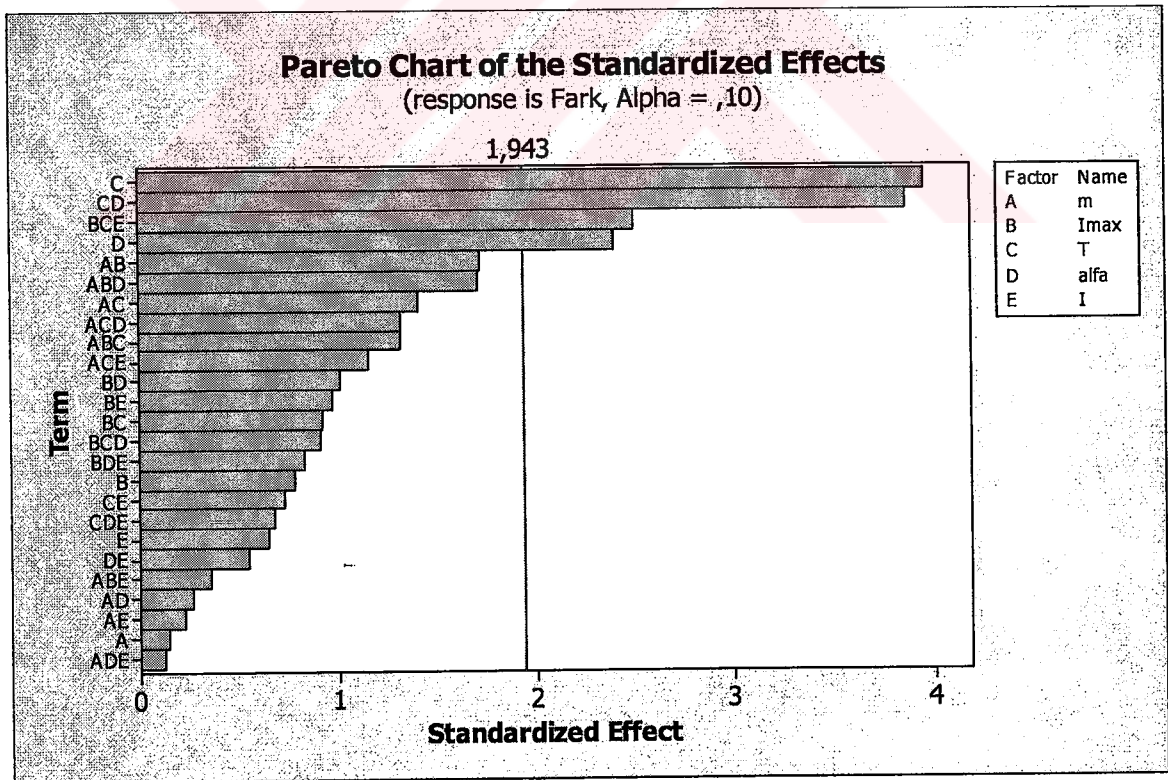
Çizelge 7.17'de görüldüğü gibi algoritmanın takas sayısı "T" ve " α " parametrelerinin ana etkileri çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olan "T" ve " α " parametreleri bir araya geldiğinde de çözüm kalitesi performansı üzerinde etkili olmaktadır. Bunun dışındaki ikili ortak etkiler çözüm kalitesi performansı üzerinde etkili değildir. Üçlü ortak etkiler arasından ise birli ve ikili etkilerde çözüm kalitesi performansı üzerinde etkili olmayan yerel arama prosedürünün iterasyon sayısı "I" " $I^{\max}$ " ve "T" parametreleri ile bir araya geldiğinde anlamlı olmaktadır.

Çizelge 7.17 BUR26a tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		-0,0452	0,023725	-1,91	0,105	
m	-0,0069	-0,0034	0,023725	-0,14	0,890	
Imax	0,0373	0,0186	0,023725	0,79	0,462	
T	0,2513	-0,1256	0,031831	-3,95	0,008	
alfa	-0,0422	-0,0211	0,008781	-2,40	0,053	
I	-0,0309	-0,0154	0,023725	-0,65	0,539	
m*Imax	-0,0730	-0,0365	0,021224	-1,72	0,136	
m*T	-0,0896	-0,0448	0,031831	-1,41	0,209	
m*alfa	-0,0047	-0,0023	0,008781	-0,27	0,798	
m*I	-0,0097	-0,0048	0,021224	-0,23	0,828	
Imax*T	0,0585	0,0292	0,031831	0,92	0,394	
Imax*alfa	0,0178	0,0089	0,008781	1,01	0,350	
Imax*I	0,0415	0,0207	0,021224	0,98	0,366	
T*alfa	-0,0909	-0,0455	0,011781	-3,86	0,008	
T*I	-0,0461	-0,0231	0,031831	-0,72	0,496	
alfa*I	-0,0097	-0,0048	0,008781	-0,55	0,601	
m*Imax*T	-0,0031	-0,0015	0,001178	-1,31	0,237	
m*Imax*alfa	-0,0269	-0,0134	0,007854	-1,71	0,138	
m*Imax*I	0,0006	0,0003	0,000785	0,36	0,733	
m*T*alfa	-0,0309	-0,0155	0,011781	-1,31	0,237	
m*T*I	-0,0027	-0,0014	0,001178	-1,15	0,292	
m*alfa*I	-0,0019	-0,0009	0,007854	-0,12	0,909	
Imax*T*alfa	0,0216	0,0108	0,011781	0,92	0,395	
Imax*T*I	0,0089	0,0030	0,001178	2,51	0,046	
Imax*alfa*I	0,0131	0,0066	0,007854	0,84	0,435	
T*alfa*I	-0,0159	-0,0080	0,011781	-0,68	0,524	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,0016817	0,0003385	0,00006769	3,43	0,083
İkili Etkileşimler	10	0,0007168	0,0004520	0,00004520	2,29	0,161
Üçlü Etkileşimler	10	0,0003183	0,0003183	0,00003183	1,61	0,289
Kalan Hata	6	0,0001184	0,0001184	0,00001974		
Toplam	31	0,0028352				



Şekil 7.22 BUR26a problemi için normal olasılık eğrisi



Şekil 7.23 BUR26a problemi için pareto grafiği

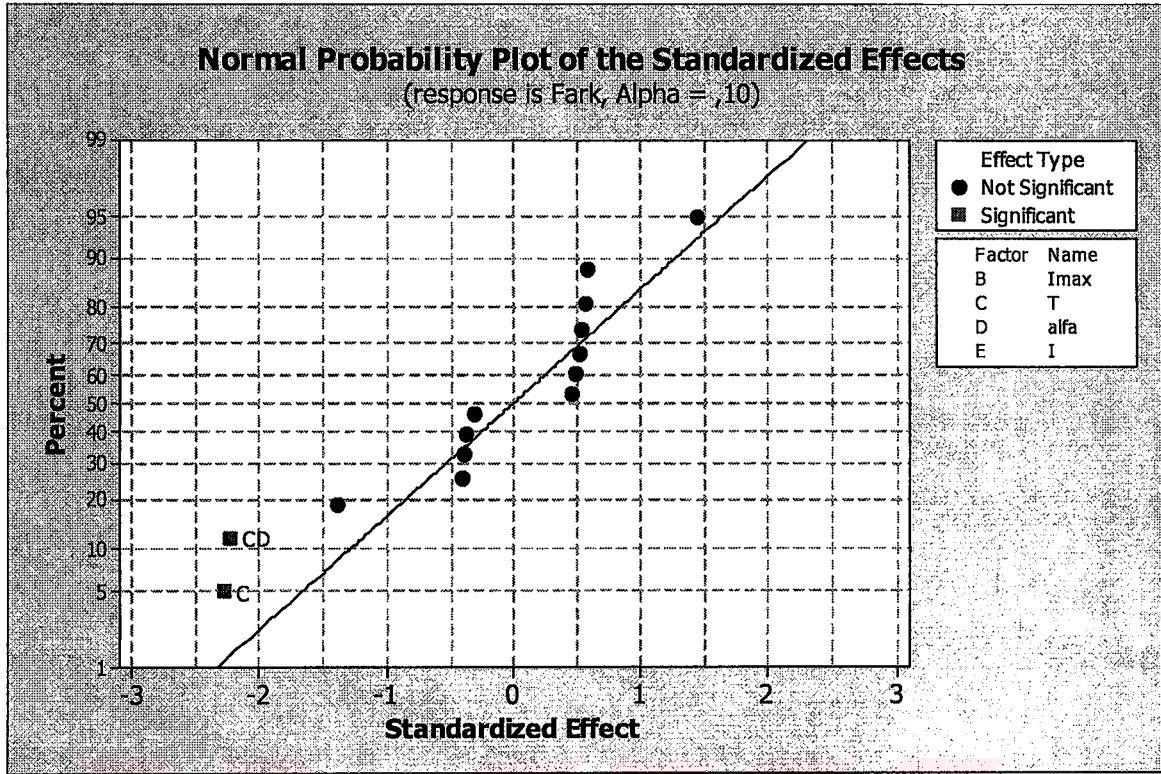
Şekil 7.22’de görüldüğü gibi “T”, “ α ”, “T- α ”, ve “ $I^{\max}$ -T-I” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.23’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “T”, “ α ”, “T- α ”, ve “ $I^{\max}$ -T-I” etkileşimleri olduğu görülmektedir.

Bur26a için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Çizelge 7.18 BUR26a indirgenmiş tasarımı

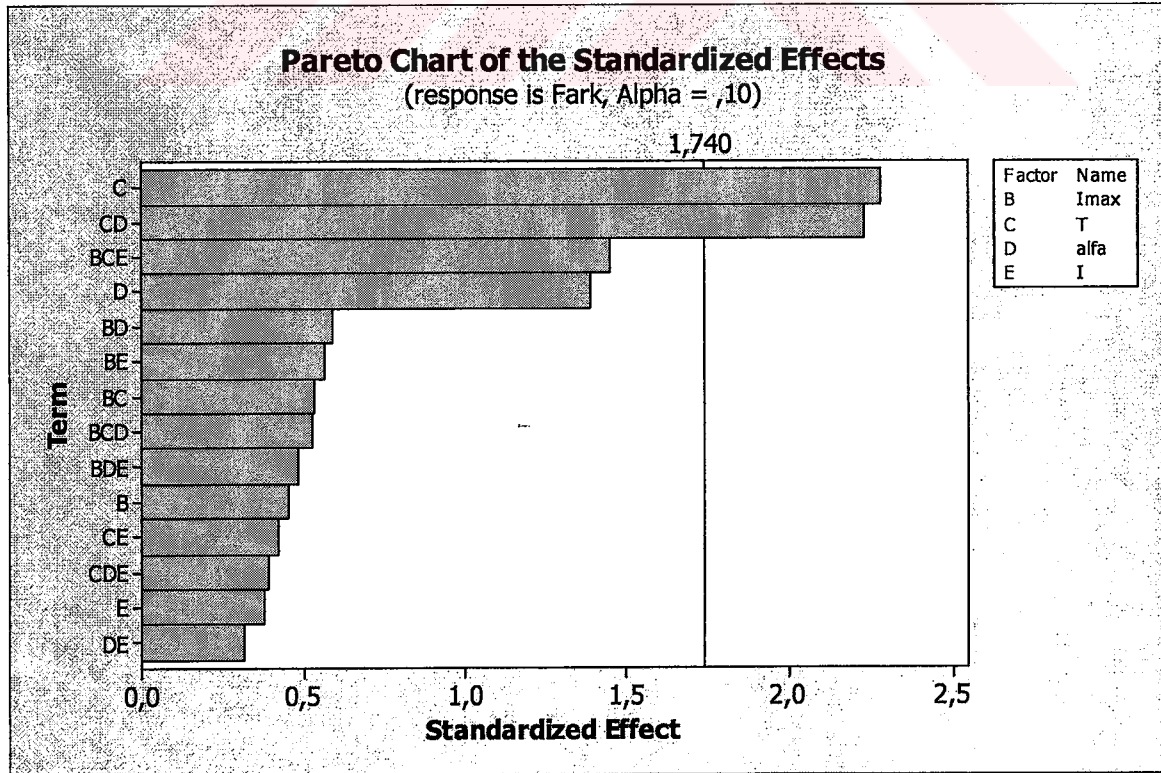
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		-0,0452	0,030421	-1,49	0,156	
Imax	0,0372	0,0186	0,030421	0,61	0,549	
T	-0,02513	-0,0256	0,040814	-3,08	0,007	
alfa	-0,0422	-0,0211	0,011259	-1,87	0,079	
I	-0,0309	-0,0154	0,030421	-0,51	0,619	
Imax*T	0,0585	0,0292	0,040814	0,72	0,484	
Imax*alfa	0,0178	0,0089	0,011259	0,79	0,441	
Imax*I	0,0415	0,0207	0,027214	0,76	0,457	
T*alfa	-0,0909	-0,0455	0,015106	-3,01	0,008	
T*I	-0,0461	-0,0231	0,040814	-0,57	0,580	
alfa*I	-0,0097	-0,0048	0,011259	-0,43	0,673	
Imax*T*alfa	0,0216	0,0108	0,015106	0,71	0,486	
Imax*T*I	0,0059	0,0030	0,001511	1,95	0,068	
Imax*alfa*I	0,0131	0,0066	0,010071	0,65	0,524	
T*alfa*I	-0,0159	-0,0080	0,015106	-0,53	0,605	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,0016817	0,0008263	0,00016526	5,09	0,006
İkili Etkileşimler	6	0,0004709	0,0003502	0,00005836	1,80	0,163
Üçlü Etkileşimler	4	0,0001634	0,0001634	0,00004084	1,26	0,327
Kalan Hata	16	0,0005192	0,0005192	0,00003245		
Toplam	31	0,0028352				

Çizelge 7.18’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla bu model için de ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



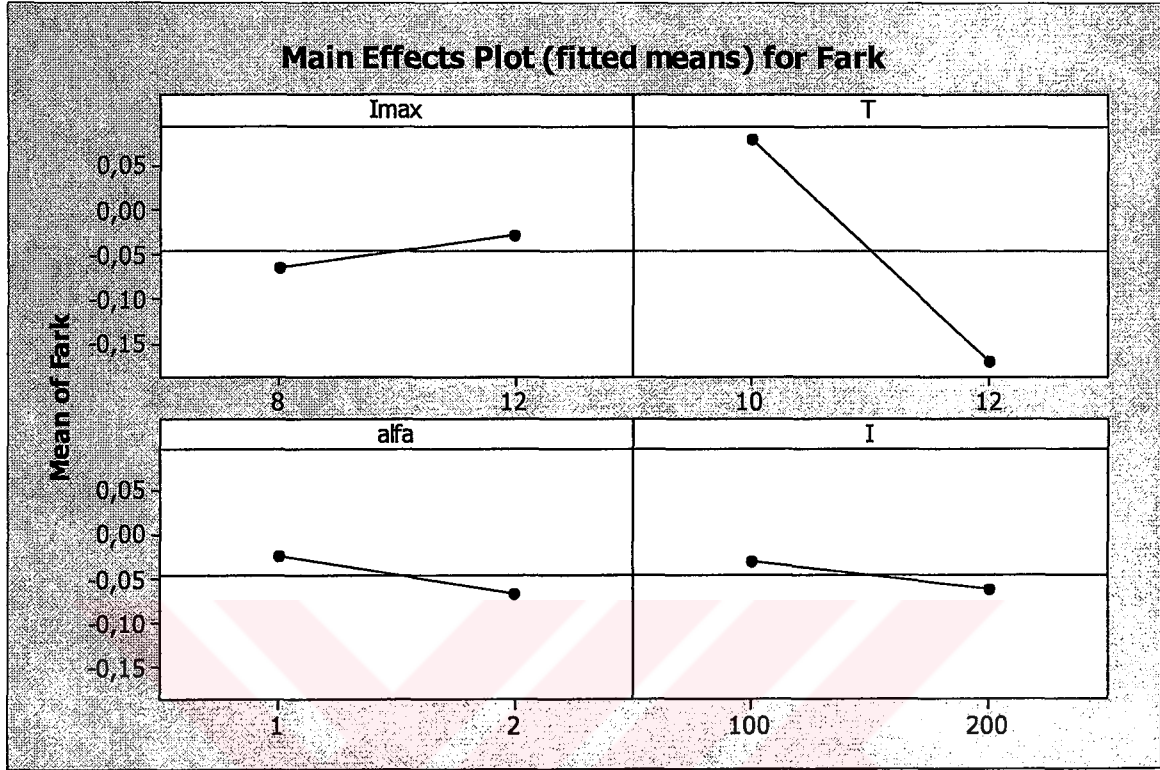
Şekil 7.24 BUR26a problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.24'deki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.



Şekil 7.25 BUR26a problemi için etkin parametrelerle histogram grafiği

Yine Şekil 7.25’de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.26 BUR26a probleminde çözüm kalitesi için ana etkiler

Şekil 7.26 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametre takas sayısı T’dir. Çünkü takas sayısının iki düzeyini yani 10 ve 12 değerlerini birleştiren eğri en dik eğridir. Diğer parametreler bu parametre kadar büyük ana etkiye sahip değildir.

Sonuç olarak, algoritmanın iterasyon sayısı “ $I^{\max}$ ” 8, takas sayısı “T” 12, “ α ” 0,2 ve yerel arama prosedürünün iterasyon sayısı “I” 200 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı “m” parametresinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu nedenle yine en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Bur26a problemi için parametre düzeyleri Çizelge 7.19’da verilmiştir.

Çizelge 7.19 Bur26a problemi için parametre düzeyleri

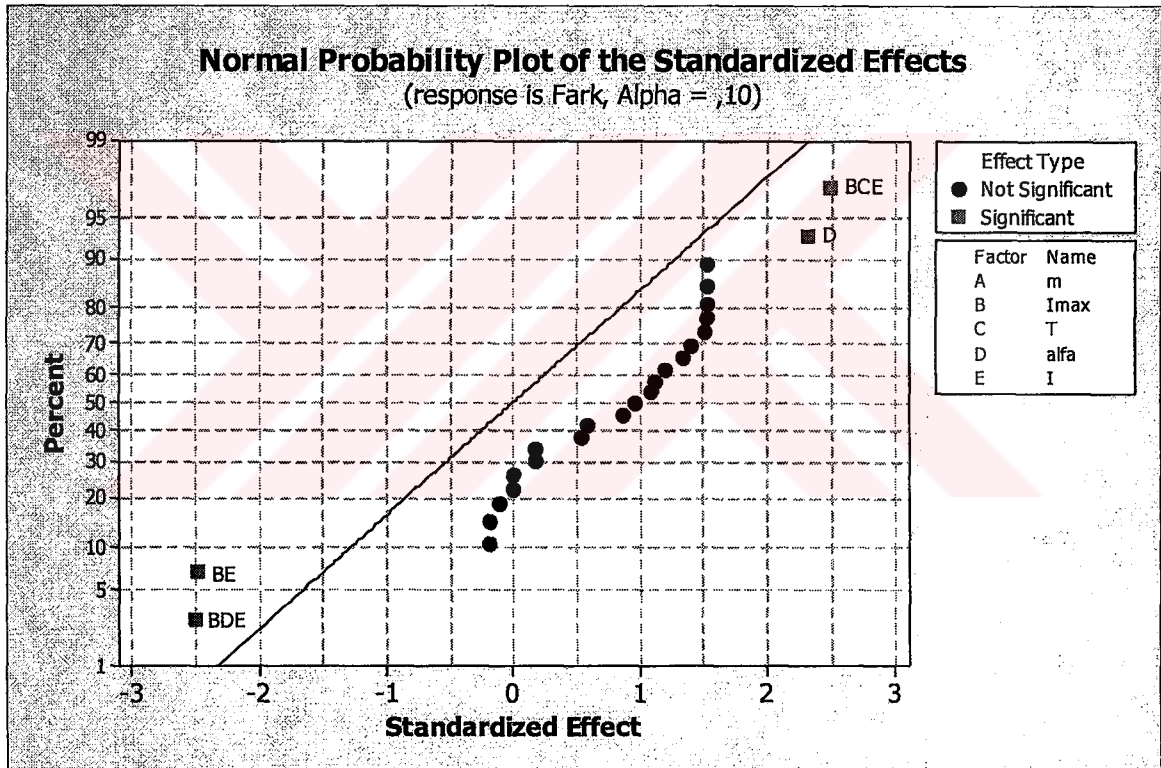
Parametre	Düzeyi
Takas Sayısı (T)	12
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	200
Algoritmanın İterasyon sayısı ($I^{\max}$)	8
İzin Buharlaştırmadan Kalan Kısmını İfade Eden Parametre (α)	0,2

7.3.2.7 BUR26h Tasarımı

Çizelge 7.20 BUR26h tasarımı

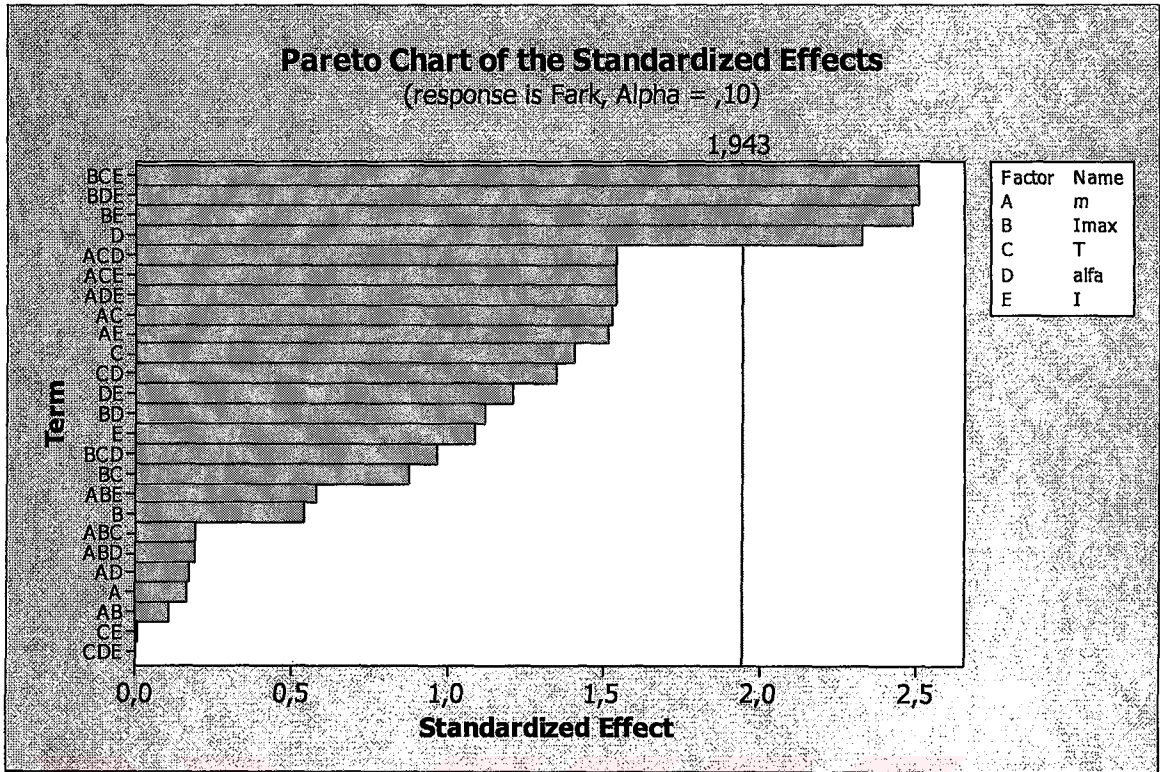
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE	Katsayı	T	P
Sabit		0,03022	0,009810		3,08	0,022
m	0,00331	0,00166	0,009810		0,17	0,871
Imax	0,01056	0,00528	0,009810		0,54	0,610
T	0,03694	0,01847	0,013162		1,40	0,210
alfa	0,01687	0,00844	0,003631		2,32	0,059
I	0,02131	0,01066	0,009810		1,09	0,319
m*Imax	-0,00194	-0,00097	0,008776		-0,11	0,916
m*T	0,04031	0,02016	0,013162		1,53	0,177
m*alfa	0,00125	0,00062	0,003631		0,17	0,869
m*I	0,02662	0,01331	0,008776		1,52	0,180
Imax*alfa	0,00812	0,00406	0,003631		1,12	0,306
Imax*I	-0,04356	-0,02178	0,008776		-2,48	0,048
T*alfa	0,01313	0,00656	0,004871		1,35	0,227
alfa*I	0,00875	0,00437	0,003631		1,20	0,274
m*T*alfa	0,01500	0,00750	0,004871		1,54	0,175
m*T*I	0,00150	0,00075	0,000487		1,54	0,175
m*alfa*I	0,01000	0,00500	0,003248		1,54	0,175
Imax*T*alfa	0,00938	0,00469	0,004871		0,96	0,373
Imax*T*I	0,00244	0,00122	0,000487		2,50	0,046
Imax*alfa*I	-0,01625	-0,00812	0,003248		-2,50	0,046
T*alfa*I	-0,00000	-0,00000	0,004871		-0,00	1,000
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,00096413	0,00002384	0,00000477	1,41	0,340
İkili Etkileşimler	10	0,00005975	0,00005472	0,00000547	1,62	0,286
Üçlü Etkileşimler	10	0,00007075	0,00007075	0,00000707	2,10	0,189
Kalan Hata	6	0,00002025	0,00002025	0,00000338		
Toplam	31	0,00111488				

BUR26h probleminin tasarımında “ α ” parametresinin ana etkisi çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olmayan “ $I^{\max}$ ” ve “I” parametreleri bir araya geldiğinde çözüm kalitesi performansı üzerinde etkili olduğu görülmektedir. Bunun dışındaki ikili ortak etkiler çözüm kalitesi performansı üzerinde etkili değildir. Üçlü ortak etkiler arasından ise ikili etkilerde çözüm kalitesi performansı üzerinde etkili olan yerel arama prosedürünün iterasyon sayısı “I” ve iterasyon sayısı “ $I^{\max}$ ” parametreleri takas sayısı “T” veya “ α ” parametresi ile bir araya geldiğinde anlamlı olmaktadır.



Şekil 7.27 BUR26h problemi için normal olasılık eğrisi

Şekil 7.27’de görüldüğü gibi “ α ”, “ $I^{\max}$ -I”, “ $I^{\max}$ -I-T” ve “ $I^{\max}$ -I- α ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.28’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “ α ”, “ $I^{\max}$ -I”, “ $I^{\max}$ -I-T” ve “ $I^{\max}$ -I- α ” etkileşimleri olduğu görülmektedir.



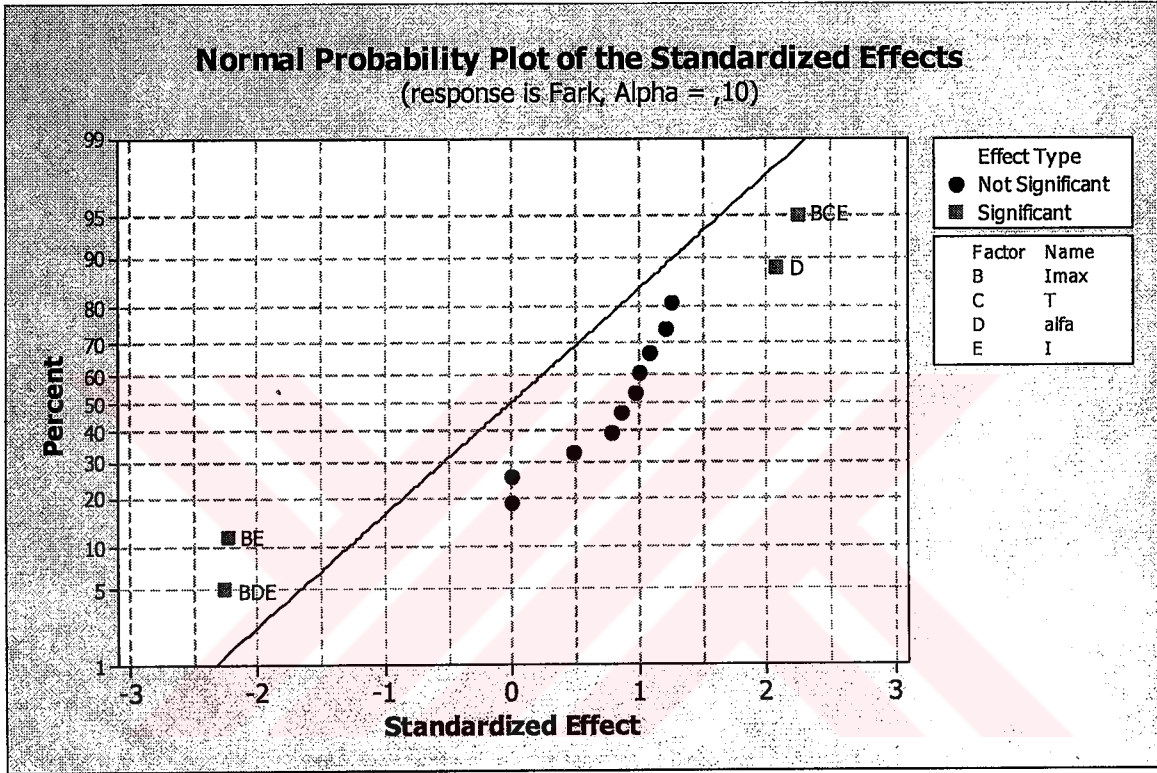
Şekil 7.28 BUR26h problemi için pareto grafiği

Çizelge 7.21 BUR26h indirgenmiş tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		0,03022	0,010913	2,77	0,013	
Imax	0,01056	0,00528	0,010913	0,48	0,635	
T	0,03694	0,01847	0,014641	1,26	0,224	
alfa	0,01687	0,00844	0,004039	2,09	0,052	
I	0,02131	0,01066	0,010913	0,98	0,343	
Imax*T	0,02306	0,01153	0,014641	0,79	0,442	
Imax*alfa	0,00812	0,00406	0,004039	1,01	0,329	
Imax*I	-0,04356	-0,02178	0,009763	-2,23	0,039	
T*alfa	0,01312	0,00656	0,005419	1,21	0,242	
T*I	-0,00019	-0,00009	0,014641	-0,01	0,995	
alfa*I	0,00875	0,00438	0,004039	1,08	0,294	
Imax*T*alfa	0,00938	0,00469	0,005419	0,87	0,399	
Imax*T*I	0,00244	0,00122	0,000542	2,25	0,038	
Imax*alfa*I	-0,01525	-0,00813	0,003613	-2,25	0,038	
T*alfa*I	0,00000	0,00000	0,005419	0,00	1,000	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	4	0,00096412	0,00002375	0,00000594	1,42	0,269
İkili Etkileşimler	6	0,00003437	0,00003789	0,00000632	1,51	0,233
Üçlü Etkileşimler	4	0,00004538	0,00004538	0,00001134	2,72	0,065
Kalan Hata	17	0,00007100	0,00007100	0,00000418		
Toplam	31	0,00111487				

BUR26h için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

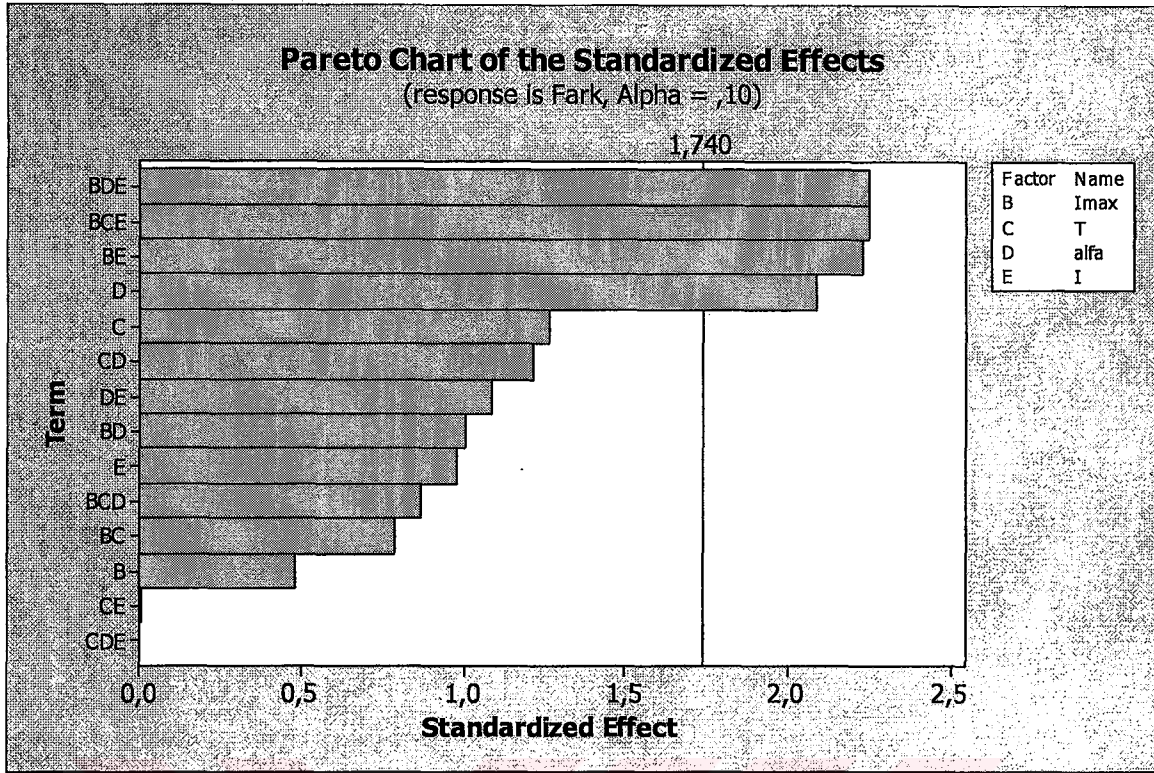
Çizelge 7.21’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla bu model için de ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



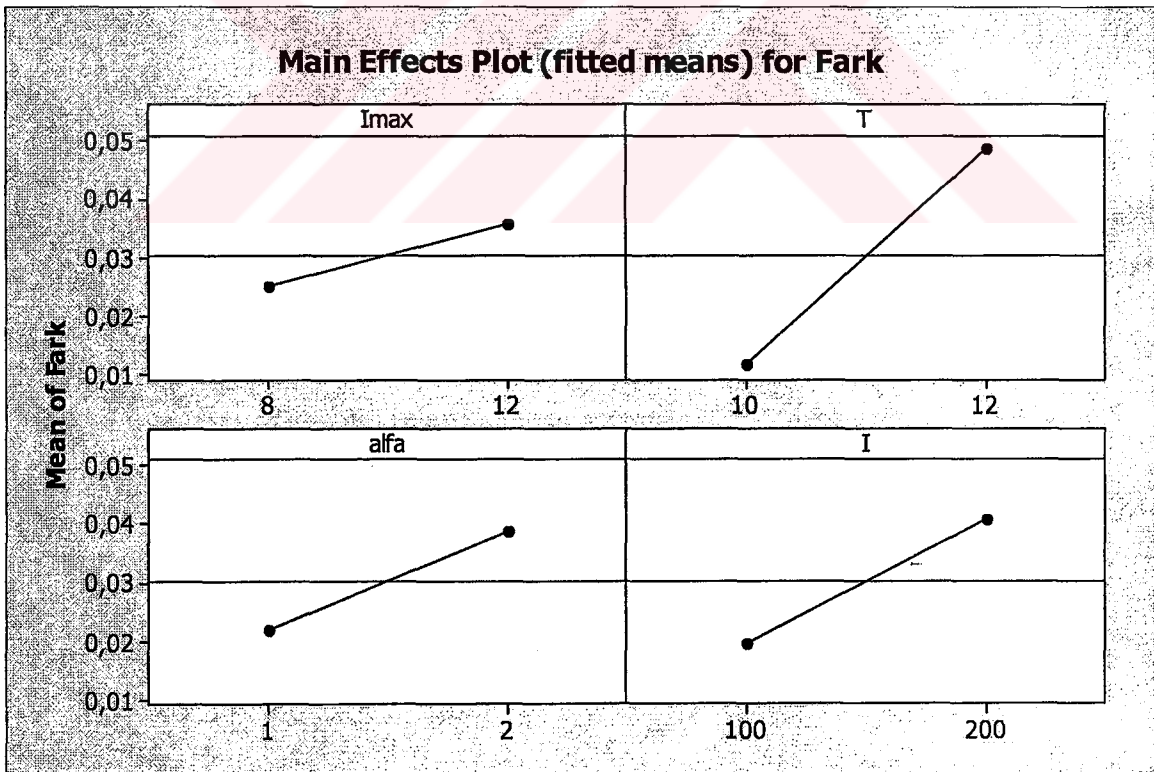
Şekil 7.29 BUR26h problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.29’daki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.

Yine Şekil 7.30’da çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.30 BUR26h problemi için etkin parametrelerle histogram grafiği



Şekil 7.31 BUR26h probleminde çözüm kalitesi için ana etkiler

Şekil 7.31 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametre takas sayısı T 'dir. Çünkü takas sayısının iki düzeyini yani 10 ve 12 değerlerini birleştiren eğri en dik eğridir. Diğer parametreler bu parametre kadar büyük ana etkiye sahip değildir.

Sonuç olarak, algoritmanın iterasyon sayısı " T^{max} " 8, takas sayısı " T " 10, " α " 0,1 ve yerel arama prosedürünün iterasyon sayısı " I " 100 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı " m " parametresinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu sebepten dolayı yine en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Bur26h problemi için parametre düzeyleri Çizelge 7.22'de verilmiştir.

Çizelge 7.22 Bur26h problemi için parametre düzeyleri

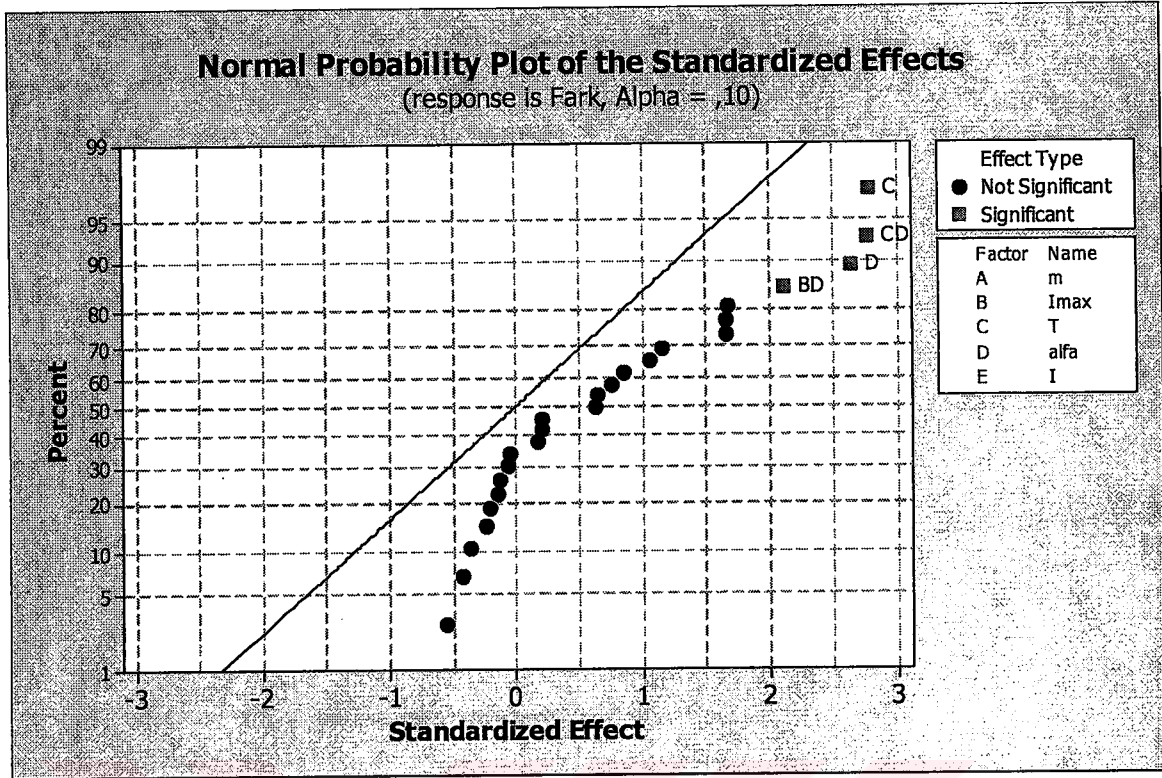
Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı (T^{max})	8
İzin Buharlaşmadan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.8 TAI30b Tasarımı

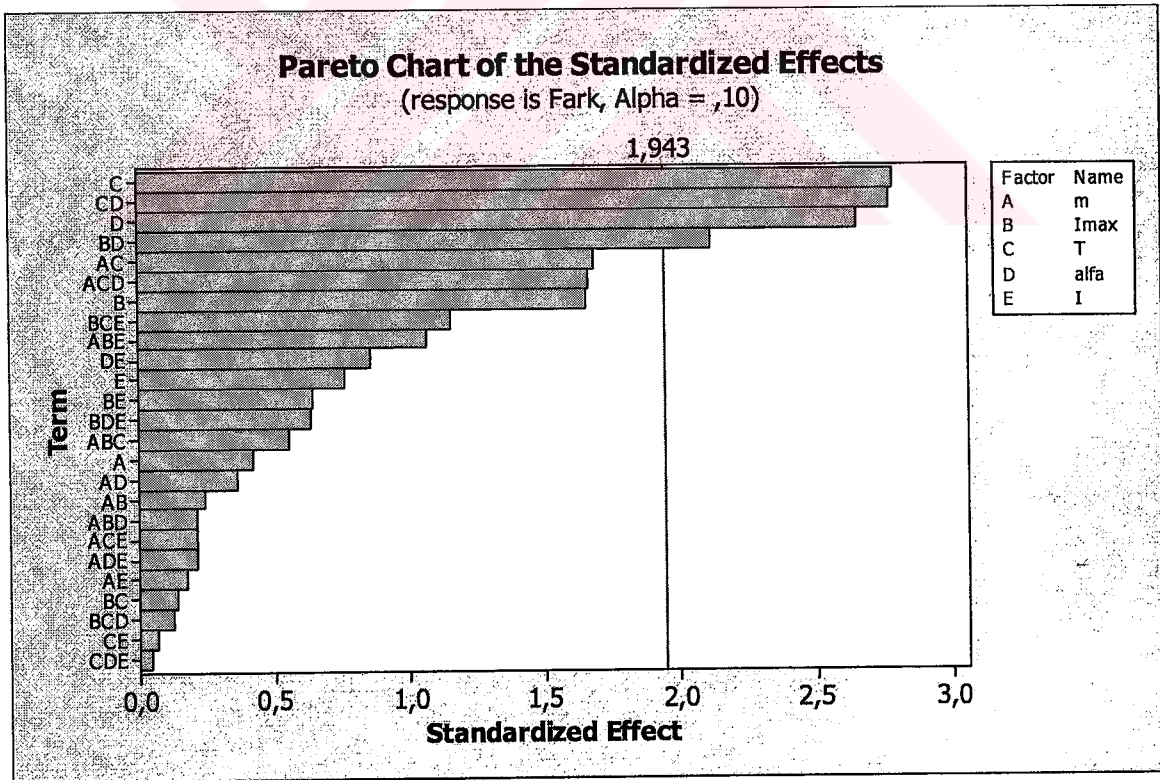
Tai30b probleminin tasarımında Çizelge 7.23'de görüldüğü gibi takas sayısı " T " ve " α " parametrelerinin ana etkisi çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkiler arasından ise ana etkileri çözüm kalitesi performansı üzerinde etkili olan " T " ve " α " parametreleri bir araya geldiğinde de çözüm kalitesi performansı üzerinde etkili olduğu görülmektedir. Bunun dışındaki ikili ortak etkiler arasından " T^{max} ", " α " etkileşimi çözüm kalitesi performansı üzerinde etkili olmaktadır. Üçlü ortak etkilerin hiç birisi anlamlı bulunmamıştır.

Çizelge 7.23 TAI30b tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE	Katsayı	T	P
Sabit		0,075812	0,022145		3,42	0,014
m	-0,018625	-0,009313	0,022145		-0,42	0,689
Imax	0,073500	0,036750	0,022145		1,66	0,148
T	0,061075	0,082688	0,029711		2,78	0,032
alfa	0,043438	0,021719	0,008196		2,65	0,038
I	0,033625	0,016813	0,022145		0,76	0,476
m*Imax	-0,009656	-0,004828	0,019811		-0,24	0,816
m*T	0,100125	0,050063	0,029711		1,68	0,143
m*alfa	-0,005938	-0,002969	0,008196		-0,36	0,730
m*I	0,007156	0,003578	0,019811		0,18	0,863
Imax*T	-0,008250	-0,004125	0,029711		-0,14	0,894
Imax*alfa	0,034687	0,017344	0,008196		2,12	0,079
Imax*I	0,025594	0,012797	0,019811		0,65	0,542
T*alfa	0,060938	0,030469	0,010996		2,77	0,032
T*I	-0,004125	-0,002062	0,029711		-0,07	0,947
alfa*I	0,014063	0,007031	0,008196		0,86	0,424
m*Imax*T	-0,001219	-0,000609	0,001100		-0,55	0,600
m*Imax*alfa	-0,003125	-0,001563	0,007331		-0,21	0,838
m*Imax*I	0,001563	0,000781	0,000733		1,07	0,328
m*T*alfa	0,036563	0,018281	0,010996		1,66	0,147
m*T*I	0,000469	0,000234	0,001100		0,21	0,838
m*alfa*I	0,003125	0,001562	0,007331		0,21	0,838
Imax*T*alfa	-0,002812	-0,001406	0,010996		-0,13	0,902
Imax*T*I	0,002531	0,001266	0,001100		1,15	0,294
Imax*alfa*I	0,009375	0,004687	0,007331		0,64	0,546
T*alfa*I	-0,000937	-0,000469	0,010996		-0,04	0,967
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,0034167	0,0002360	0,00004720	2,74	0,126
İkili Etkileşimler	10	0,0003183	0,0003357	0,00003357	1,95	0,213
Üçlü Etkileşimler	10	0,0001048	0,0001048	0,00001048	0,61	0,767
Kalan Hata	6	0,0001032	0,0001032	0,00001720		
Toplam	31	0,0039430				



Şekil 7.32 TAI30b problemi için normal olasılık eğrisi



Şekil 7.33 TAI30b problemi için pareto grafiği

Tai30b için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler

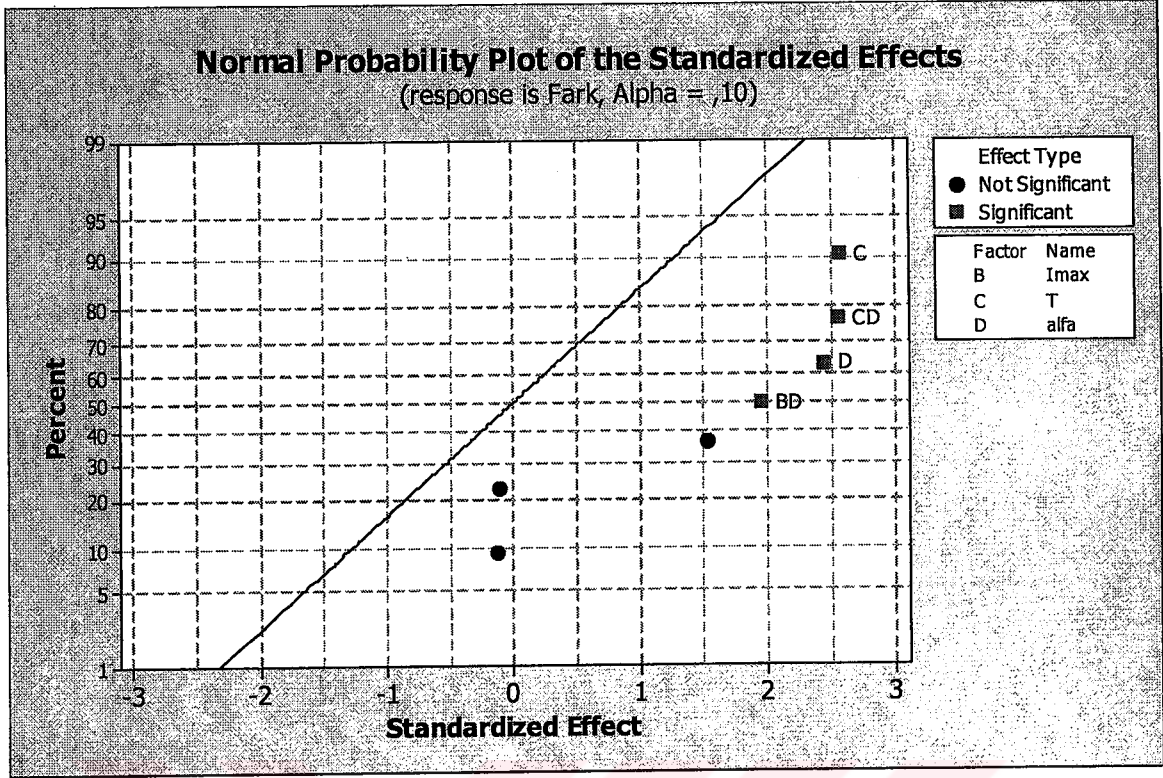
çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Şekil 7.32’de görüldüğü gibi “T”, “ α ”, “ $I^{\max}$ - α ” ve “T- α ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.33’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “T”, “ α ”, “ $I^{\max}$ - α ” ve “T- α ” etkileşimleri olduğu görülmektedir.

Çizelge 7.24 TAI30b indirgenmiş tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		0,075812	0,023937	3,17	0,004	
Imax	0,073500	0,036750	0,023937	1,54	0,138	
T	0,165377	0,082097	0,032115	2,57	0,017	
alfa	0,043437	0,021719	0,008860	2,45	0,022	
Imax*T	-0,008250	-0,004125	0,032115	-0,13	0,899	
Imax*alfa	0,034688	0,017344	0,008860	1,96	0,062	
T*alfa	0,060937	0,030469	0,011886	2,56	0,017	
Imax*T*alfa	-0,002812	-0,001406	0,011886	-0,12	0,907	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	3	0,00322534	0,00022305	0,00007435	3,70	0,026
İkili Etkileşimler	3	0,00023509	0,00023431	0,00007810	3,89	0,021
Üçlü Etkileşimler	1	0,00000028	0,00000028	0,00000028	0,01	0,907
Kalan Hata	24	0,00048225	0,00048225	0,00002009		
Toplam	24	0,00048225	0,00048225	0,00002009		

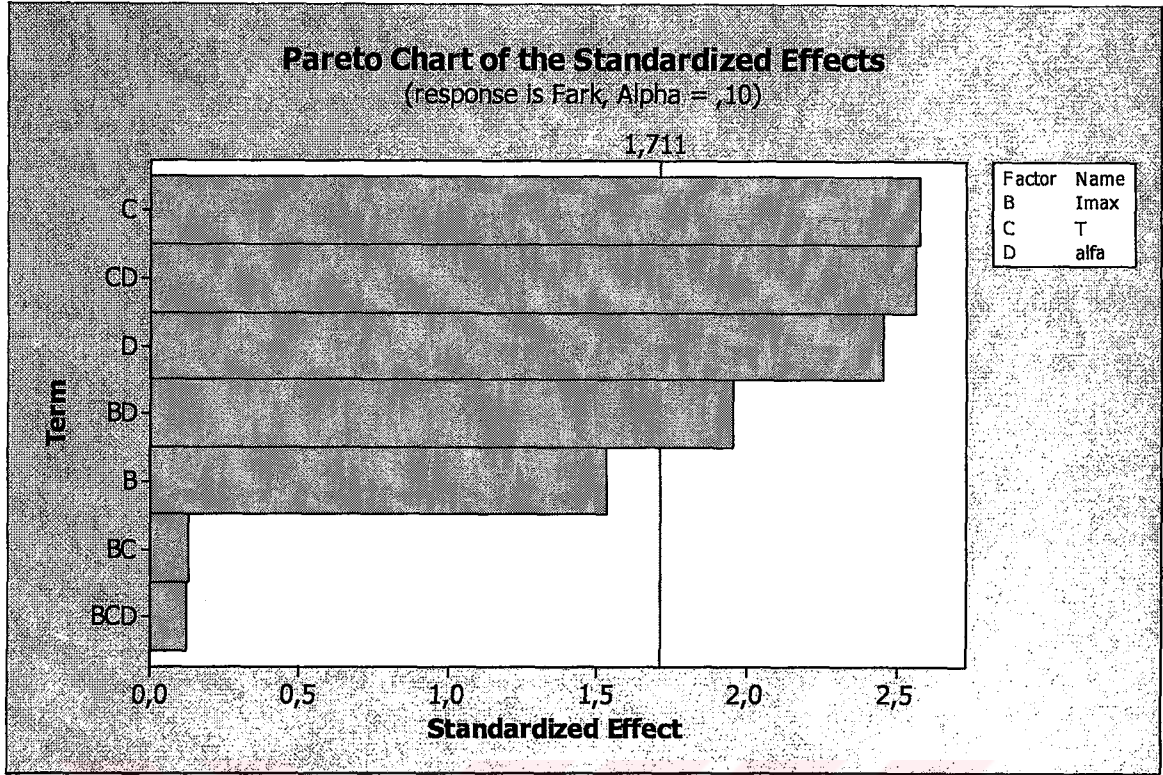
Çizelge 7.24’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla bu model için de ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



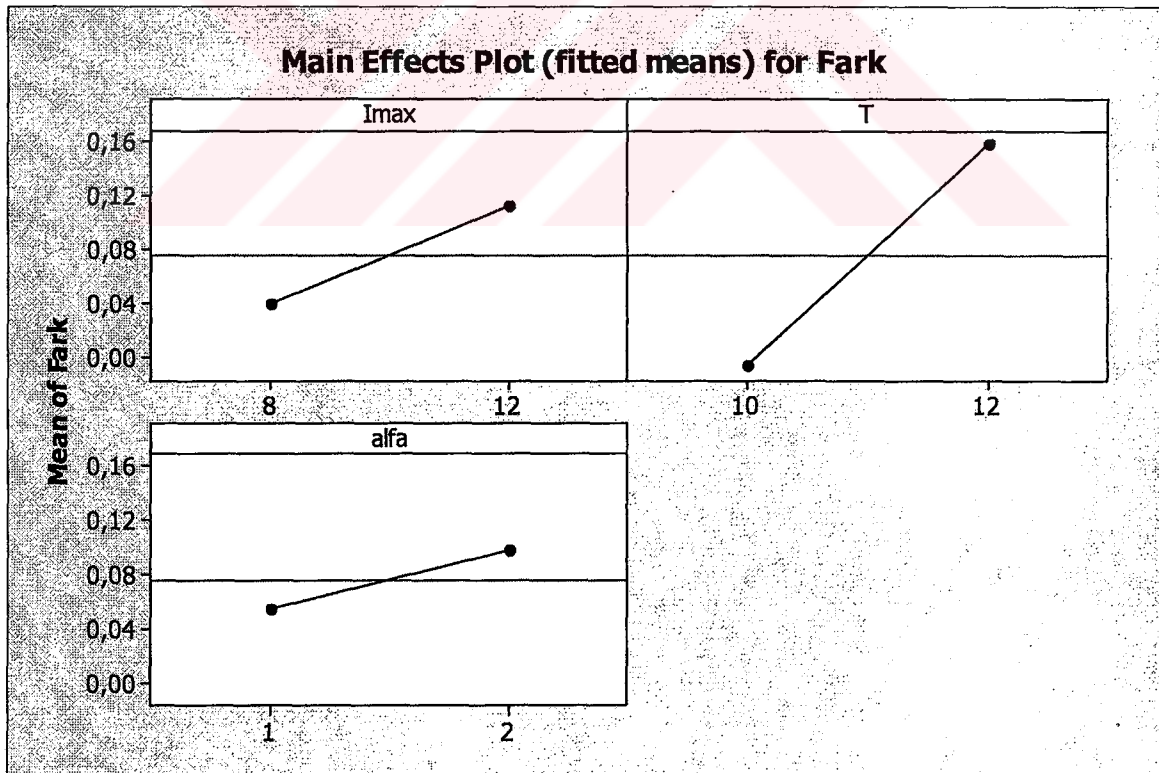
Şekil 7.34 TAI30b problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.34'deki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.

Şekil 7.35'de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.35 TAI30b problemi için etkin parametrelerle histogram grafiği



Şekil 7.36 TAI30b probleminde çözüm kalitesi için ana etkiler

Şekil 7.36 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametre “T”dir. Çünkü parametrenin iki düzeyini birleştiren eğriler en dik eğrilerdir. Diğer parametreler bu parametreler kadar büyük ana etkiye sahip değildir.

Sonuç olarak, takas sayısı “T” 10, “ α ” 0,1 ve algoritmanın iterasyon sayısı “ $T^{\max}$ ” 8 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı “m” ve yerel arama prosedürünün iterasyon sayısı “I”, parametrelerinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamışlardır, bu nedenle yine en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Tai30b problemi için parametre düzeyleri Çizelge 7.25’de verilmiştir.

Çizelge 7.25 Tai30b problemi için parametre düzeyleri

Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($T^{\max}$)	8
İzin Buharlaşımdan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.9 TAI40b Tasarımı

Tai40b probleminin tasarımında “ α ” parametresinin ana etkisi çözüm kalitesi performansı üzerinde etkili olduğu diğer parametrelerin ise tek başlarına fazla etkili olmadığı görülmektedir. İkili ortak etkilerin hiç birisi çözüm kalitesi performansı üzerinde etkili değildir. Üçlü ortak etkiler arasından ise ana etkisi çözüm kalitesi performansı üzerinde etkili olmayan iterasyon sayısı “I”, takas sayısı “T” ve algoritmanın iterasyon sayısı “ $T^{\max}$ ” parametreleri bir araya geldiğinde anlamlı olmaktadır.

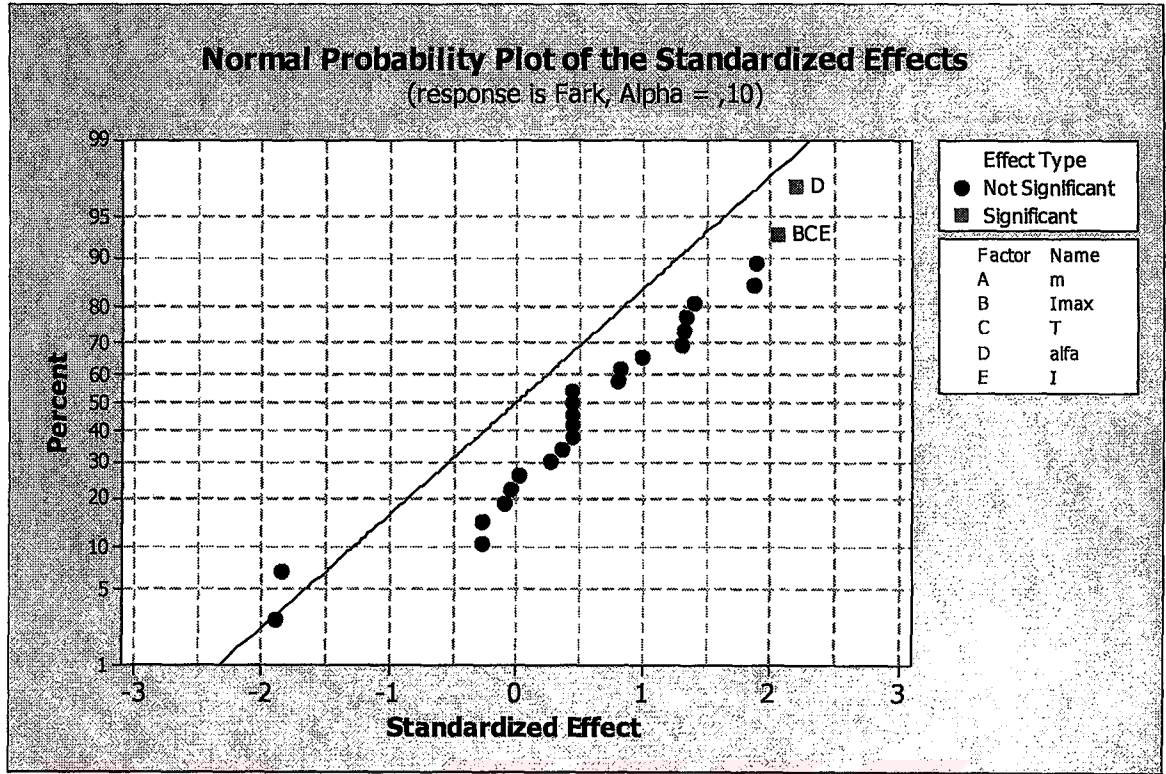
Çizelge 7.26 TAI40b tasarımı

Fark için Tahmin Edilen Etki ve Katsayılar

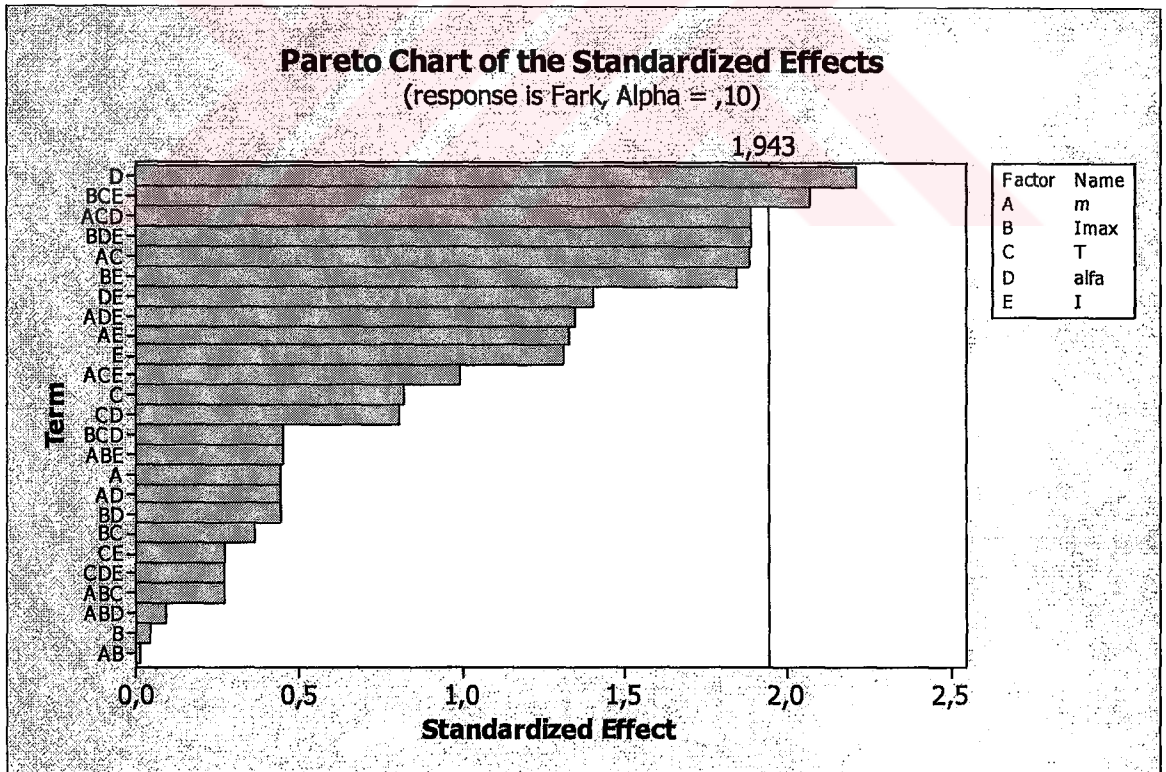
Terim	Etki	Katsayı	SE Katsayı	T	P
Sabit		0,03041	0,010498	2,90	0,027
m	0,00931	0,00466	0,010498	0,44	0,673
Imax	-0,00094	-0,00047	0,010498	-0,04	0,966
T	0,02306	0,01153	0,014084	0,82	0,444
alfa	0,01719	0,00859	0,003885	2,21	0,069
I	0,02756	0,01378	0,010498	1,31	0,237
m*Imax	0,00022	0,00011	0,009391	0,01	0,991
m*T	0,05306	0,02653	0,014084	1,88	0,109
m*alfa	0,00344	0,00172	0,003885	0,44	0,674
m*I	0,02497	0,01248	0,009391	1,33	0,232
Imax*T	0,01031	0,00516	0,014084	0,37	0,727
Imax*alfa	0,00344	0,00172	0,003885	0,44	0,674
Imax*I	-0,03466	-0,01733	0,009391	-1,85	0,115
T*alfa	0,00844	0,00422	0,005213	0,81	0,449
T*I	-0,00769	-0,00384	0,014084	-0,27	0,794
alfa*I	0,01094	0,00547	0,003885	1,41	0,209
m*Imax*T	0,00028	0,00014	0,000521	0,27	0,796
m*Imax*alfa	-0,00063	-0,00031	0,003475	-0,09	0,931
m*Imax*I	0,00031	0,00016	0,000348	0,45	0,669
m*T*alfa	0,01969	0,00984	0,005213	1,89	0,108
m*T*I	0,00103	0,00052	0,000521	0,99	0,361
m*alfa*I	0,00937	0,00469	0,003475	1,35	0,226
Imax*T*alfa	0,00469	0,00234	0,005213	0,45	0,669
Imax*T*I	0,00216	0,00108	0,000521	2,07	0,084
Imax*alfa*I	-0,01312	-0,00656	0,003475	-1,89	0,108

Çözüm için Varyans Analizi

Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	0,00075941	0,00002647	0,00000529	1,37	0,352
İkili Etkileşimler	10	0,00006456	0,00004948	0,00000495	1,28	0,397
Üçlü Etkileşimler	10	0,00005706	0,00005706	0,00000571	1,48	0,328
Kalan Hata	6	0,00002319	0,00002319	0,00000386		
Toplam	31	0,00090422				



Şekil 7.37 TAI40b problemi için normal olasılık eğrisi



Şekil 7.38 TAI40b problemi için pareto grafiği

Şekil 7.37’de görüldüğü gibi “ α ” ve “ $m-I-I^{\max}$ ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.38’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “ α ” ve “ $m-I-I^{\max}$ ” etkileşimleri olduğu görülmektedir.

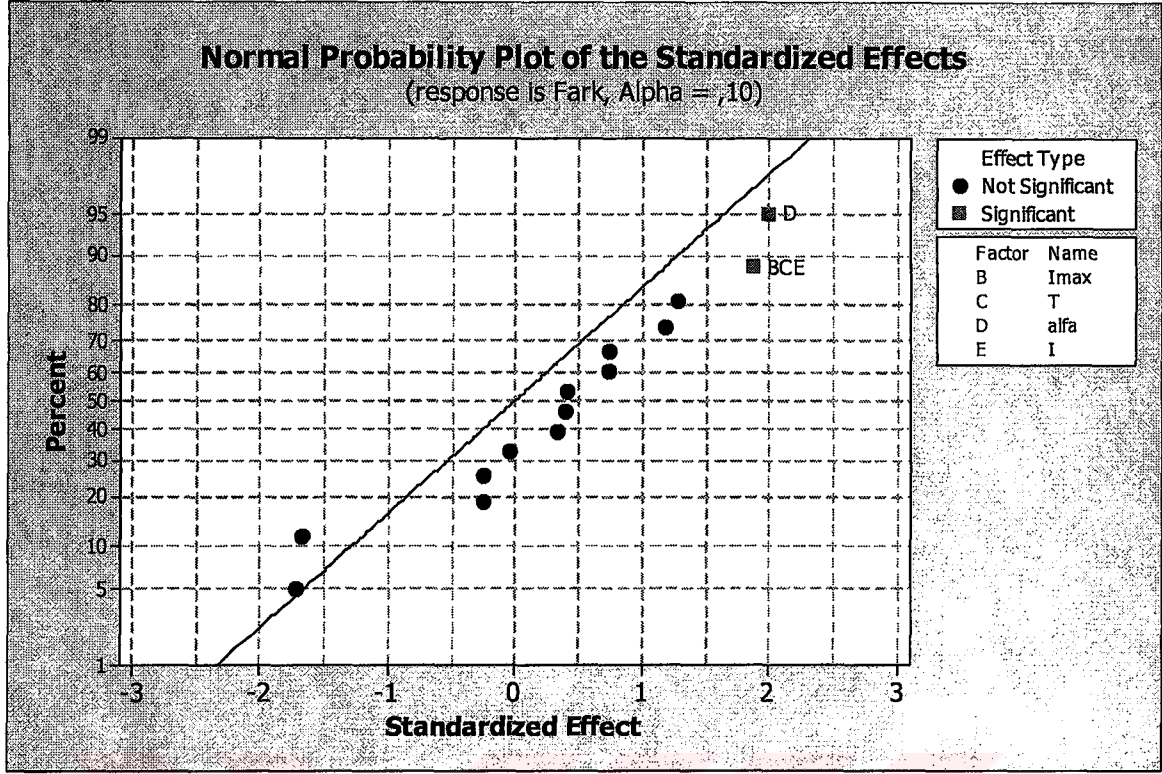
Tai40b için etkin parametreler belirlendikten sonra modelden etkin olmayan parametreler çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Çizelge 7.27 TAI40b indirgenmiş tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		0,03041	0,011568	2,63	0,018	
Imax	-0,00094	-0,00047	0,011568	-0,04	0,968	
T	0,02306	0,01153	0,015520	0,74	0,468	
alfa	0,01719	0,00859	0,004282	2,01	0,061	
I	0,02756	0,01378	0,011568	1,19	0,250	
Imax*T	0,01031	0,00516	0,015520	0,33	0,744	
Imax*alfa	0,00344	0,00172	0,004282	0,40	0,693	
Imax*I	-0,03466	-0,01733	0,010349	-1,67	0,112	
T*alfa	0,00844	0,00422	0,005744	0,73	0,473	
T*I	-0,00769	-0,00384	0,015520	-0,25	0,807	
alfa*I	0,01094	0,00547	0,004282	1,28	0,219	
Imax*T*alfa	0,00469	0,00234	0,005744	0,41	0,688	
Imax*T*I	0,00216	0,00108	0,000574	1,88	0,078	
Imax*alfa*I	-0,01313	-0,00656	0,003830	-1,71	0,105	
T*alfa*I	-0,00281	-0,00141	0,005744	-0,24	0,810	

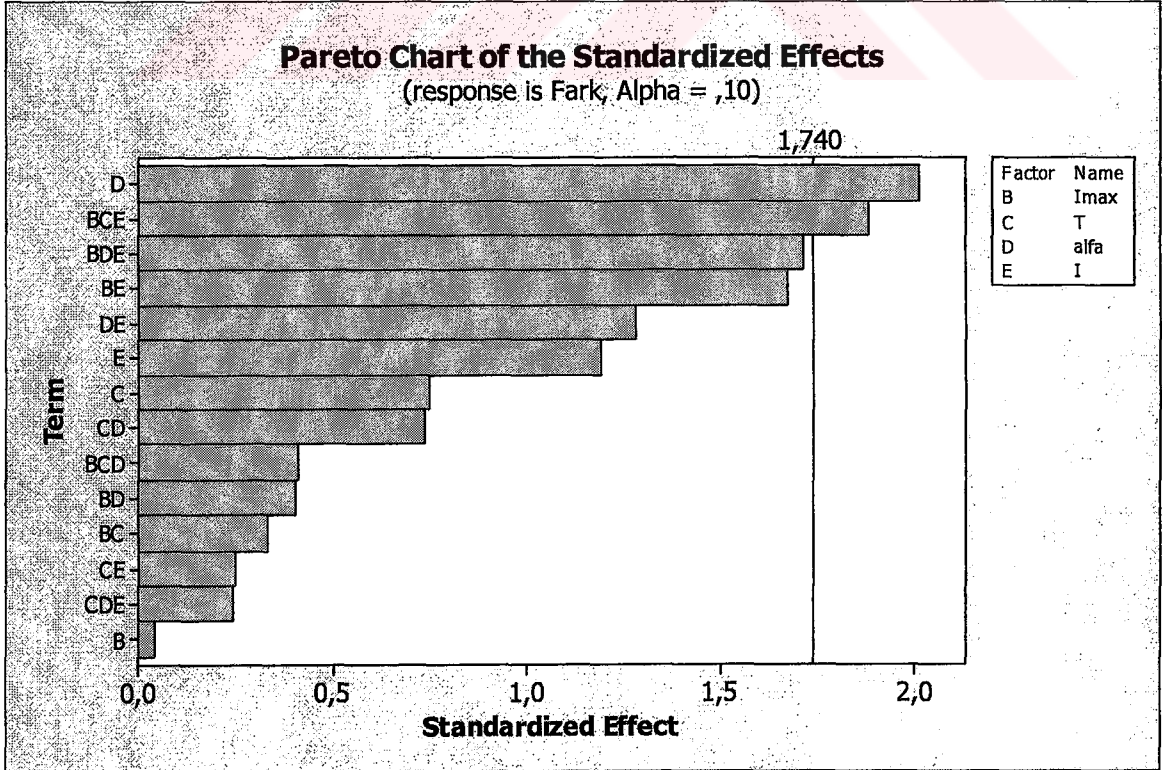
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	4	0,00075938	0,00002571	0,00000643	1,37	0,286
İkili Etkileşimler	6	0,00003369	0,00002817	0,00000469	1,00	0,457
Üçlü Etkileşimler	4	0,00003138	0,00003138	0,00000784	1,67	0,203
Kalan Hata	17	0,00007978	0,00007978	0,00000469		
Toplam	31	0,00090422				

Çizelge 7.27’de indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Dolayısıyla bu model için de ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.



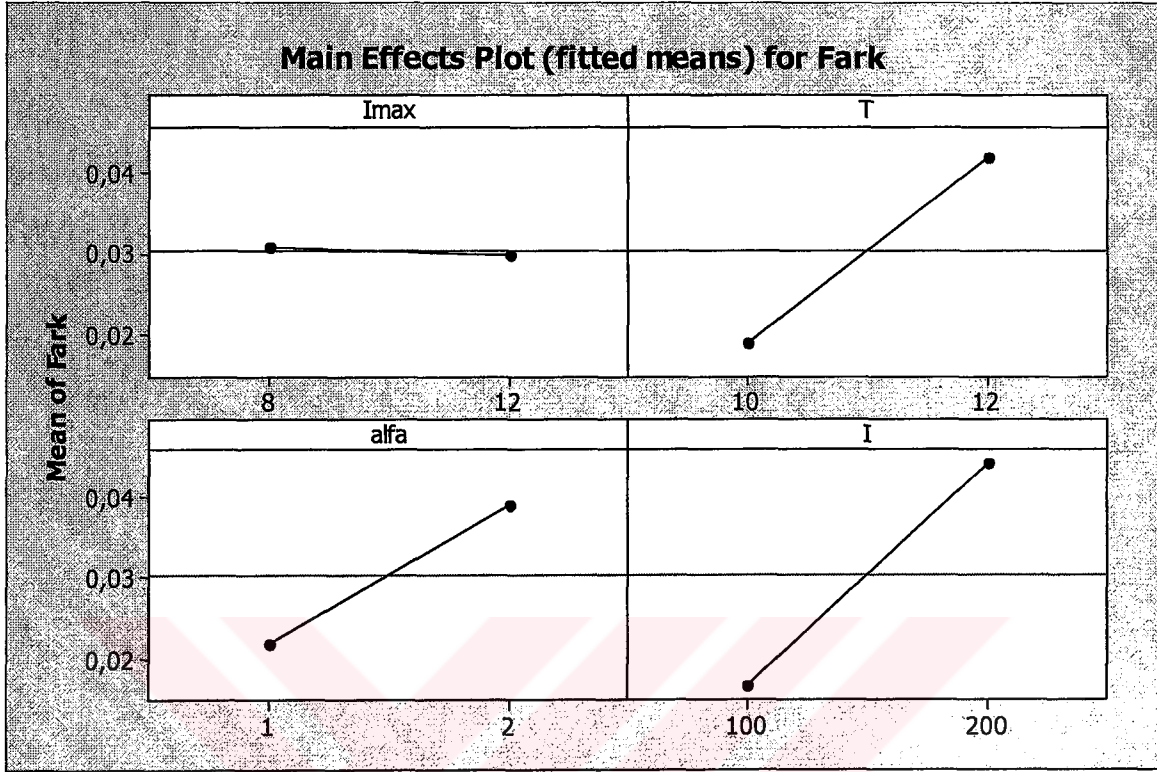
Şekil 7.39 TAI40b problemi için etkin parametrelerle normal olasılık eğrisi

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.39'daki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.



Şekil 7.40 TAI40b problemi için etkin parametrelerle histogram grafiği

Şekil 7.40'da çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.41 TAI40b probleminde çözüm kalitesi için ana etkiler

Şekil 7.41 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi en fazla ana etkiye sahip olan parametreler “ α ”, “T” ve “I”dır. Çünkü parametrelerin iki düzeyini birleştiren eğriler en dik eğrilerdir. Diğer parametreler bu parametreler kadar büyük ana etkiye sahip değildir.

Sonuç olarak, takas sayısı “T” 10, “ α ” 0,1, yerel arama prosedürünün iterasyon sayısı “I” 100 ve algoritmanın iterasyon sayısı “ $I^{\max}$ ” 12 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı “m” parametresinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu sebepten dolayı yine en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Tai40b problemi için parametre düzeyleri Çizelge 7.28’de verilmiştir.

Çizelge 7.28 Tai40b problemi için parametre düzeyleri

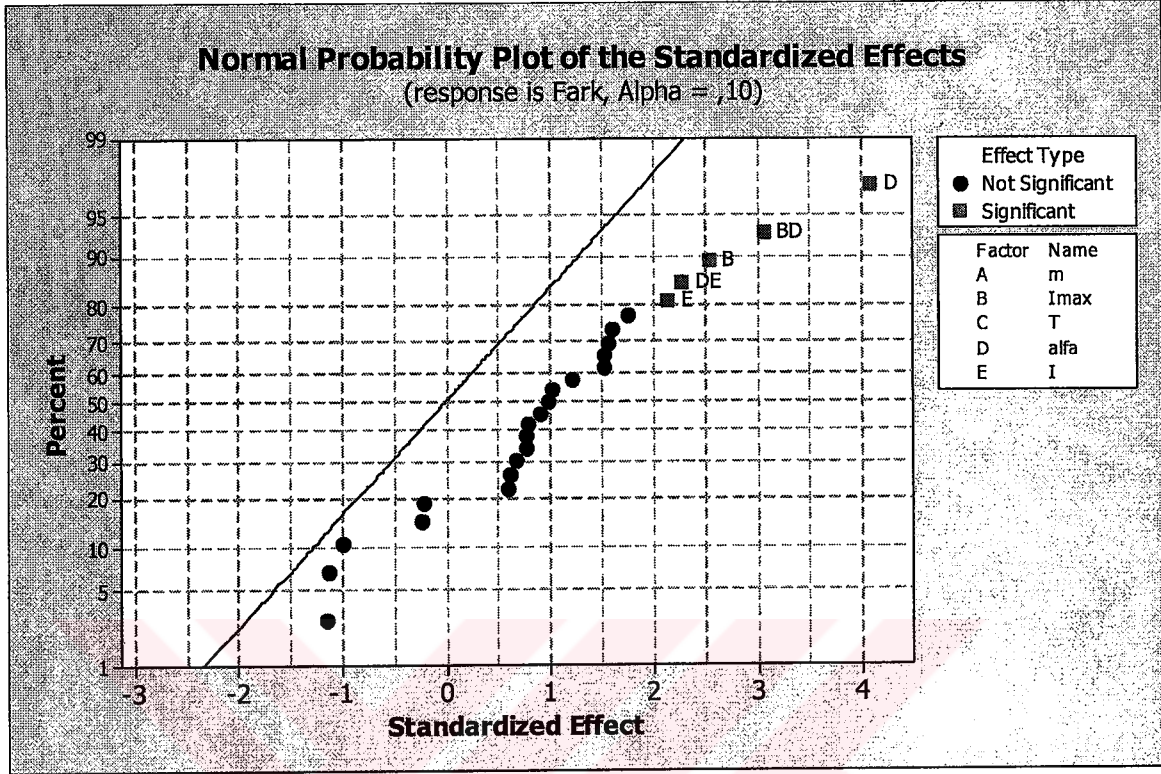
Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($I^{\max}$)	12
İzin Buharlaşmadan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.3.2.10 TAI80b Tasarımı

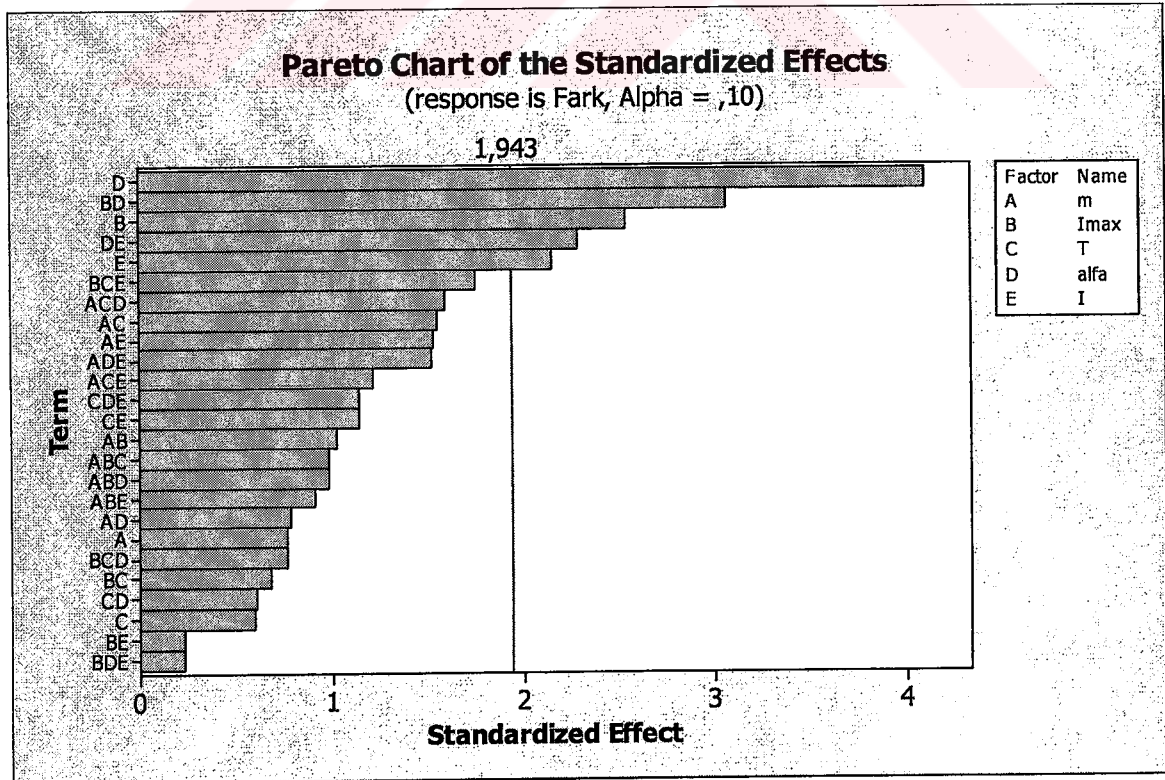
Çizelge 7.29 TAI80b tasarımı

Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		5,074	1,00097	5,07	0,002	
m	1,537	0,768	1,00097	0,77	0,472	
$I^{\max}$	5,094	2,547	1,00097	2,54	0,044	
T	1,605	0,803	1,34294	0,60	0,572	
α	5,029	1,51	0,37048	4,09	0,006	
I	4,315	2,157	1,00097	2,16	0,075	
m* $I^{\max}$	1,842	0,921	0,89545	1,03	0,343	
m*T	4,179	2,090	1,34294	1,56	0,171	
m* α	0,581	0,291	0,37048	0,78	0,463	
m*I	2,747	1,374	0,89545	1,53	0,176	
$I^{\max}$ *T	1,818	0,909	1,34294	0,68	0,524	
$I^{\max}$ * α	2,274	1,137	0,37048	3,07	0,022	
$I^{\max}$ *I	-0,422	-0,211	0,89545	-0,24	0,821	
T* α	0,606	0,303	0,49705	0,61	0,565	
T*I	-3,060	-1,530	1,34294	-1,14	0,298	
α *I	1,693	0,846	0,37048	2,28	0,062	
m* $I^{\max}$ *T	-0,098	-0,049	0,04970	-0,99	0,360	
m* $I^{\max}$ * α	0,655	0,327	0,33136	0,99	0,361	
m* $I^{\max}$ *I	0,061	0,030	0,03314	0,91	0,396	
m*T* α	1,590	0,795	0,49705	1,60	0,161	
m* α *I	1,010	0,505	0,33136	1,52	0,178	
$I^{\max}$ *T* α	0,758	0,379	0,49705	0,76	0,475	
$I^{\max}$ * α *I	-0,150	-0,075	0,33136	-0,23	0,828	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	5	8,6943	1,0651	0,21302	6,06	0,024
İkili Etkileşimler	10	1,0518	0,9605	0,09605	2,73	0,115
Üçlü Etkileşimler	10	0,4980	0,4980	0,04980	1,42	0,347
Kalan Hata	6	0,2108	0,2108	0,03514		
Toplam	31	10,4549				

Ana etkisi çözüm performansını etkileyen parametreler " $I^{\max}$ ", " α ", " I " dir. Anlamli olan ikili ortak etkiler ise " $I^{\max} - \alpha$ " ve " $I - \alpha$ " dir.



Şekil 7.42 TAI80b problemi için normal olasılık eğrisi



Şekil 7.43 TAI80b problemi için pareto grafiği

Şekil 7.42’de görüldüğü gibi “ $T^{\max}$ ”, “ α ”, “ I ”, “ $T^{\max} - \alpha$ ” ve “ $I - \alpha$ ” etkili değerlerin, normal olasılık eğrisinin dışında kaldığı görülmektedir. Şekil 7.43’de ise pareto grafiğinde, çizginin dışında kalan birli, ikili ve üçlü etkileşimlerin tabloda verilen, “ $T^{\max}$ ”, “ α ”, “ I ”, “ $T^{\max} - \alpha$ ” ve “ $I - \alpha$ ” etkileşimleri olduğu görülmektedir.

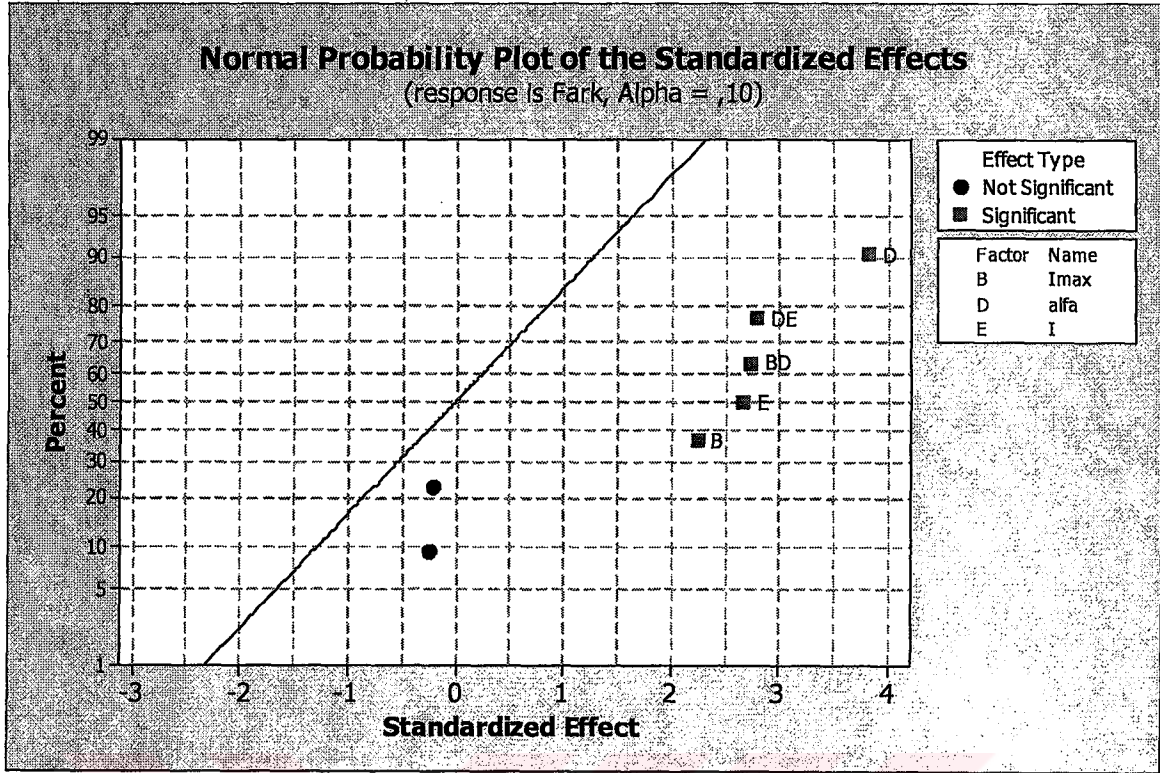
Bu problem için etkin parametreler belirlendikten sonra modelden etkin olmayan “ m ” ve “ T ” parametreleri çıkarılarak tasarım yeniden yapılır ve yine aynı parametreler etkin çıkıyorsa model doğrulanmış olur.

Çizelge 7.30 TAI80b indirgenmiş tasarımı

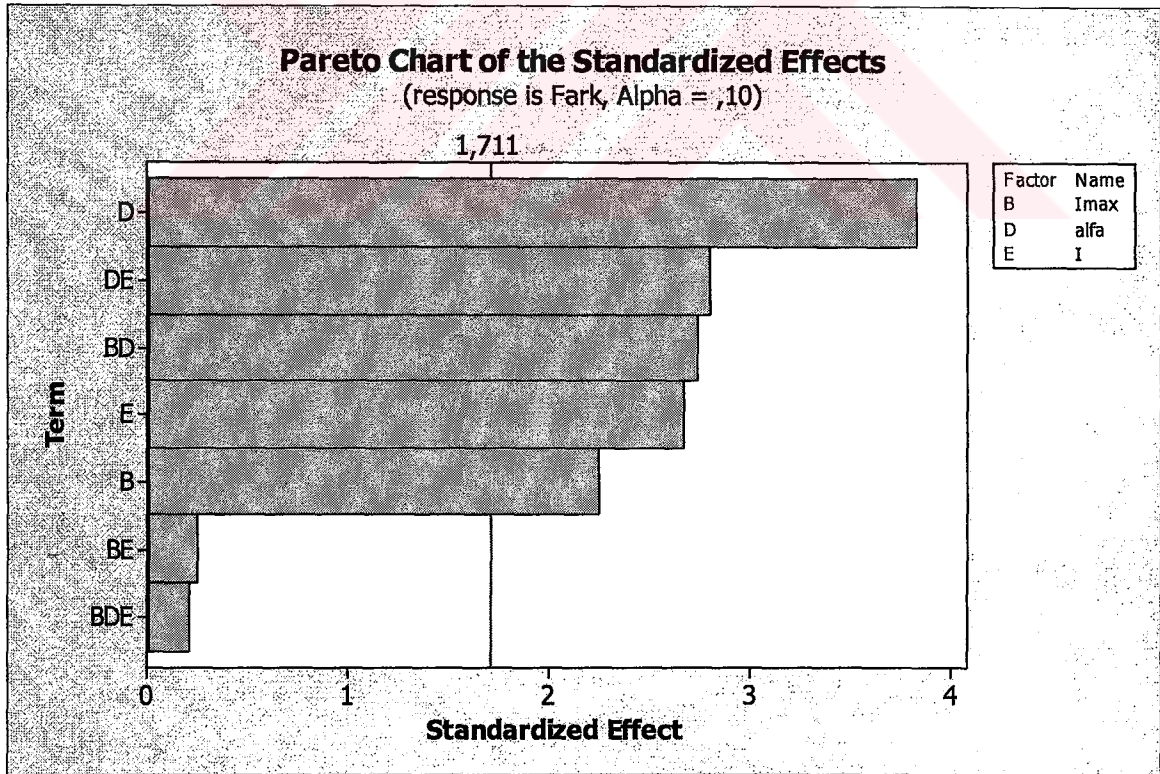
Fark İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE Katsayı	T	P	
Sabit		4,8069	0,9974	4,82	0,000	
Imax	4,4877	2,2439	0,9974	2,25	0,034	
alfa	2,8275	1,4137	0,3692	3,83	0,001	
I	5,3350	2,6675	0,9974	2,67	0,013	
Imax*alfa	2,0212	1,0106	0,3692	2,74	0,011	
Imax*I	-0,4805	-0,2403	0,9974	-0,24	0,812	
alfa*I	2,0713	1,0356	0,3692	2,81	0,010	
Imax*alfa*I	-0,1500	-0,0750	0,3692	-0,20	0,841	
Çözüm İçin Varyans Analizi						
Kaynak	DF	Seq SS	Adj SS	Adj MS	F	P
Ana Etkiler	3	8,6909	1,17221	0,390736	8,96	0,000
İkili Etkileşimler	3	0,7156	0,67257	0,224191	5,14	0,007
Üçlü Etkileşimler	1	0,0018	0,00180	0,001800	0,04	0,841
Kalan Hata	24	1,0466	1,04660	0,043608		
Toplam	31	10,4549				

Çizelge 7.30’da indirgenmiş modelin tasarımı gösterilmektedir. Çizelge incelendiğinde yine aynı ana etki ve ikili etkilerin etkin olduğu gözükmemektedir. Yani model ileriki değerlendirme ve araştırmalar için adaydır sonucuna varılabilir.

Etkin olmayan parametreler çıkarıldıktan sonra çizdirilen Şekil 7.44’deki normal olasılık eğrisi, tablodan elde edilen sonuçları doğrulamaktadır.

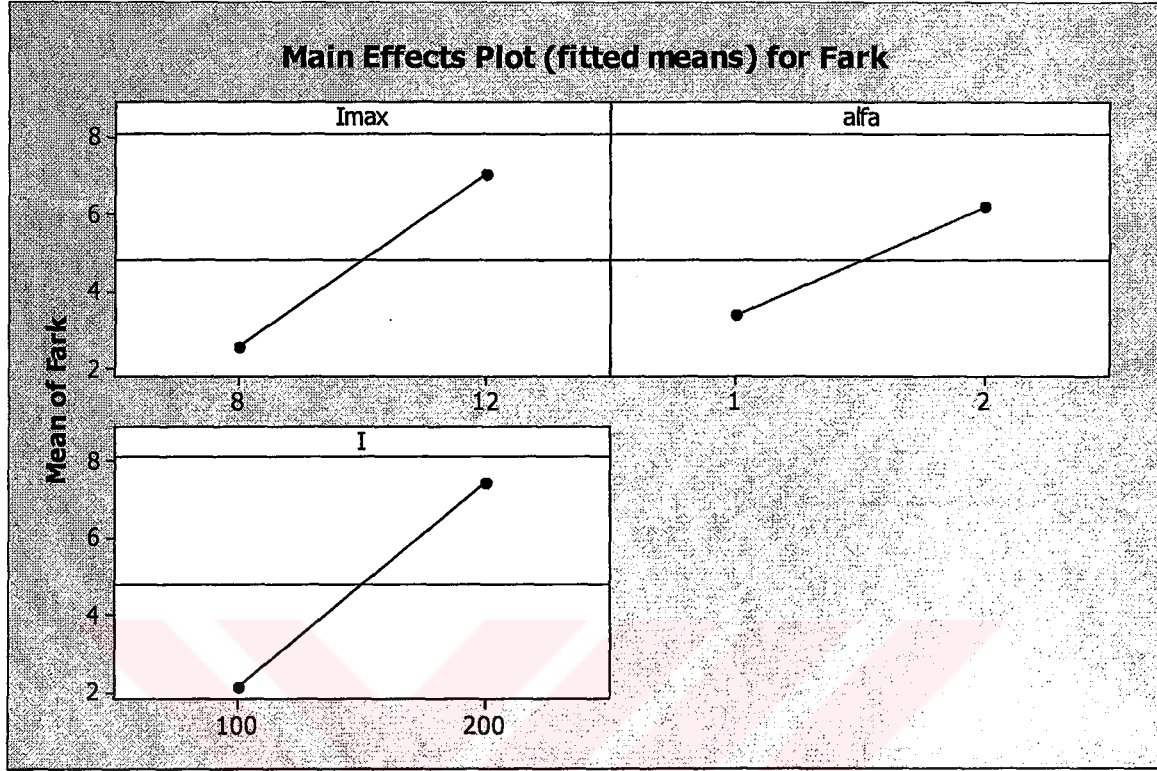


Şekil 7.44 TAI80b problemi için etkin parametrelerle normal olasılık eğrisi



Şekil 7.45 TAI80b problemi için etkin parametrelerle histogram grafiği

Şekil 7.45’de çizdirilen histogram grafiğinde, çizginin dışında kalan parametre ve etkileşimlerinin de tabloyu doğruladığı görülmektedir.



Şekil 7.46 TAI80b probleminde çözüm kalitesi için ana etkiler

Şekil 7.46 ana etki eğrilerini göstermektedir. Şekilden de anlaşılabileceği gibi bütün “I” ve “I^{max}” parametrelerin ana etkisi birbirine yakındır. Çünkü iki parametrenin düzeylerini birleştiren eğrilerin diklikleri yaklaşık olarak aynıdır. Diğer parametreler bu parametreler kadar büyük ana etkiye sahip değildir.

Sonuç olarak, iterasyon sayısı “I^{max}” 8, yerel arama prosedürünün iterasyon sayısı “I” 100 ve “α” 0,1 olarak ele alınırsa çözüm kalitesi artacaktır. Karınca sayısı “m” ve takas sayısı parametrelerinin çözüm kalitesi üzerinde fazla bir etkisi yoktur çünkü deney tasarımı sonucunda etkin parametre olarak çıkmamıştır, bu nedenle en küçük düzey etkin olmayan parametreler için parametre seçiminde kullanılmıştır.

Tai80b problemi için parametre düzeyleri Çizelge 7.31’de verilmiştir.

Çizelge 7.31 Tai80b problemi için parametre düzeyleri

Parametre	Düzeyi
Takas Sayısı (T)	10
Karınca Sayısı (m)	10
Yerel Arama Prosedürünün İterasyon Sayısı (I)	100
Algoritmanın İterasyon sayısı ($I^{\max}$)	8
İzin Buharlaştırmadan Kalan Kısmını İfade Eden Parametre (α)	0,1

7.4 Çözümlerin Değerlendirilmesi

Geliştirilmiş olan TavlamalıKarınca algoritması KAP kütüphanesinden alınan 5'i düzenli 5'i düzensiz sınıfından oluşan problemler seti üzerinde denenmiş ve en uygun parametreler seti yapılan deney tasarımı ile belirlenmiştir. Bu denemelerde her bir parametre kombinasyonu için C++ programlama dilinde yazılan program üç kere çalıştırılmış ve bunlar içinden elde edilen en iyi çözümler arası farklar Ek 1 ve Ek 2'de sunulmuştur. Tasarım esnasında yapılan deneylerden elde edilen en iyi sonuçlar Çizelge 7.32'de ve Çizelge 7.33'de verilmiştir. Her bir Çizelgeden de anlaşılacağı gibi geliştirilen TavlamalıKarınca algoritması 10 deney probleminde sadece birisi dışında problemin bilinen en iyi çözümünü bulmuştur.

Çizelge 7.32 Düzenli sınıfa dahil test problemleri

PROBLEM	BOYUTU	BİLİNE EN İYİ ÇÖZÜMÜ	BULUNAN EN İYİ ÇÖZÜM
Nug 20	20	2570	2570
Nug 30	30	6124	6124
Tai 30a	30	1818146	1818146
Tai 60a	60	7208572	7280658
Sko 64	64	48498	48498

Çizelge 7.33 Düzensiz sınıfına dahil test problemleri

PROBLEM	BOYUTU	BİLİLEN EN İYİ ÇÖZÜMÜ	BULUNAN EN İYİ ÇÖZÜM
Bur26a	26	5426670	5426670
Bur26h	26	7098658	7098658
Tai30b	30	637117113	637117113
Tai40b	40	637250948	637250948
Tai80b	80	818415043	818415043

7.5 Diğer Meta-Sezgisellerle Karşılaştırılması

TavlamalıKarinca algoritmasının performansı reaktif tabu arama (RTA) (Battiti ve Tecchiolli, 1994), TA (Taillard, 1991), genetik melez metot (GMM) (Fleurent ve Ferland, 1994), TB (Connolly, 1990), MKA (Gambardella vd., 1997) meta-sezgiselleri ile karşılaştırılmıştır.

AnfTabu algoritması paralel bir uygulama olması nedeniyle karşılaştırılmaya tabi tutulmamıştır.

Çizelge 7.34 Diğer meta-sezgisellerle karşılaştırma

Problem	Bilinen En İyi Değer	RTA	TA	GMM	TB	MKA	TavlamalıKarinca
Nug20	2570	0,911	0	0	0,070	0	0,003
Nug30	6124	0,872	0,032	0,007	0,121	0,098	0,010
Tai30a	1818146	0,286	0,340	0,439	0,907	1,311	0,024
Tai60a	7208572	0,831	1,270	1,159	1,395	3,070	1,080
Sko64	48498	0,861	0,064	0,092	0,095	0,129	0,002
Bur26a	5426670	-	0	0,012	0,141	0	0,012
Bur26h	7098658	-	0	0	0,126	0	0,007
Tai30b	637117113	-	0,054	0	4,407	0	0,017
Tai40b	637250948	-	0,208	0,210	4,564	0	0,007
Tai80b	818415043	-	1,435	0,828	1,438	0,667	0,989

Bu karşılaştırmada TavlamalıKarnca algoritmasının Ek 1 ve Ek 2’de her bir problem için verilen 32 değerin ortalaması dikkate alınmıştır.

Çizelge 7.34’de görüldüğü gibi TavlamalıKarnca algoritması üç problem için en iyi performansı sağlarken diğer problemler için de başarılı sonuçlar elde etmiştir.

Çizelge 7.35 Diğer meta-sezgisellerle karşılaştırma

Problem	Bilinen En İyi Değer	RTA	TA	GMM	TB	MKA	TavlamalıKarnca
Nug20	2570	0,911	0	0	0,070	0	0
Nug30	6124	0,872	0,032	0,007	0,121	0,098	0,002
Tai30a	1818146	0,286	0,340	0,439	0,907	1,311	0,006
Tai60a	7208572	0,831	1,270	1,159	1,395	3,070	0,365
Sko64	48498	0,861	0,064	0,092	0,095	0,129	0,001
Bur26a	5426670	-	0	0,012	0,141	0	0,005
Bur26h	7098658	-	0	0	0,126	0	0
Tai30b	637117113	-	0,054	0	4,407	0	0,005
Tai40b	637250948	-	0,208	0,210	4,564	0	0
Tai80b	818415043	-	1,435	0,828	1,438	0,667	0,274

Ek 1 ve Ek 2’de verilen deney sonuçlarından her bir problem için en iyi 10 değerinin ortalaması alınarak yapılan Çizelge 7.35’deki karşılaştırmada test edilen 10 problemin 8’inde en iyi çözüm performansı sağlayan algoritma olduğu görülmektedir.

8. SONUÇLAR VE ÖNERİLER

Son yıllarda artan rekabet ortamında, pazar payları azalan firmalar sürekliliklerini sağlamak için oluşabilecek problemlerini tespit edip çözümlemesi gerekmektedir. Teknolojideki inanılmaz gelişmeler ve bilimin hızlı ilerlemesine rağmen hala bir çok problem için optimum çözüm üreten yöntemler geliştirilememiştir. Araştırmacılar böyle problemler için sezgisel metotlar geliştirerek optimuma yakın sonuçlar elde etmeye çalışmaktadır. Bu çalışmada, bu sezgisellerden Karınca Algoritması ve Karınca Algoritmasının tesis yerleşimi problemlerine uygulanması üzerinde durulmuştur.

Malzeme taşıma maliyetlerinin minimizasyonunun amaçlandığı n tesise m yerleşimin atanması olarak tanımlanan tesis yerleşimi problemi, çözüm zamanının problem boyutuna bağlı olarak üstel artış göstermesi nedeniyle NP problemlerdendir. Bu nedenle mevcut metotlarla büyük boyutlu tesis yerleşimi problemlerinde optimum çözüm bulunamamaktadır.

Literatürde tesis yerleşimi probleminin formülizasyonu üzerine bir çok araştırmacı çalışmalar yapmıştır. Bunlar arasında en yaygın olarak kullanılan, Koopmans ve Beckman (1957) tarafından geliştirilen, eşit alanlı n yerleşime n tesisin atanması olarak tanımlanan Kareli Atama Problemidir (KAP).

Günümüzde bir çok kombinatoriyal optimizasyon problemi için kullanılan KA ismini, yapısının gerçek karıncaların yemlerini arama davranışlarından esinlenerek oluşturulmasından almaktadır. Algoritmanın temeli gerçek hayattaki karınca kolonisi bireyleri arasında direkt bir iletişimin olmaması, iletişim için bireylerin toprağın üzerine bıraktıkları kimyasal bir madde olan fenomen izlerini kullanmalarıdır.

Karıncalar gibi neredeyse kör hayvanların kolonisinin beslenme kaynaklarına ulaşmada en kısa yolu bulmayı nasıl yönettikleri algoritmayı geliştiren araştırmacılara yön vermiştir.

Bu çalışmada, mevcut metotlarla büyük boyutlu problemlerde optimum çözümü bulunamayan KAP için yerel arama prosedürü TB tabanlı “TavlamalıKarınca” algoritması geliştirilmiştir. Algoritmada yerel arama prosedürünün kullanılması algoritmanın melez yapılı bir algoritma olduğunu göstermektedir.

Algoritmada ilk olarak karınca sayısı kadar rassal ilk çözümler atanır. Bu çözümler TB tabanlı yerel arama prosedüründe geliştirilir. Bu çözümler arasından en iyisi referans alınarak ilk fenomen matrisi oluşturulur.

Karınca sayısı kadar çözüm öncelikle fenomen izine bağlı olarak geliştirilir. Elde edilen karınca sayısı kadar yeni çözüm yerel arama prosedüründe de bir geliştirme sürecine tâbi tutulur.

Her bir iterasyonun geliştirme sürecinde gelişme sağlanıyorsa yoğunlaşma mekanizması aktiftir denir. Belli bir iterasyon sonunda gelişme sağlanamamışsa çeşitlendirme mekanizması devreye girer. Bu mekanizma yeniden rassal olarak başlama olabileceği gibi, başlangıç noktası olarak ayrı bir hafızada tutulan iyi çözümlerde kullanılabilir.

TavlamalıKarınca algoritmasının, en iyi parametre setlerini belirlemek için deney tasarımı yapılmıştır. Bu tasarımda kullanılan 20, 26, 30, 40, 60, 64 ve 80 boyutlu, eşit sayıda düzenli ve düzensiz yapıli test problemleri KAP Kütüphanesinden alınmıştır. Deney tasarımında 2^k faktöryel deney tasarımı ile en iyi performansı sağlayan parametre düzeyleri belirlenmiştir. Deneyler TavlamalıKarınca algoritmasının C++ programlama dilinde kodlanmasıyla geliştirilen programda yapılmıştır.

Bu çalışma sonucunda elde edilen deney sonuçlarına göre, geliştirilen TavlamalıKarınca algoritmasının performansı iki önemli faktöre göre değerlendirilebilir. Bu faktörlerden birincisi, problemin yapısı iken ikincisi ise problemin boyutudur.

Düzenli yapıli problemlerde optimum çözüm bulma zorluğunun düzensiz yapıli problemlere göre daha fazla olduğu sonucuna varılmıştır. Bu sonuç daha önce diğer sezgisellerle yapılan çalışmalarda da görülmüştür. İkinci önemli faktör olarak test, probleminin boyutu arttıkça çözüm performansında kısmen düşmeler görülmektedir.

Her bir test problemi için deney tasarımı kısmında elde edilen 96 sonuçtan en iyiler göz önüne alındığında algoritmanın 9 problemin optimum sonucuna vardığı görülmektedir. Bu sonuçtan da anlaşılacağı üzere, geliştirilen algoritmanın farklı yapıda ve boyutta problemler için optimum sonuç bulma performansı oldukça yüksektir.

Algoritmanın performansının diğer bir değerlendirmesi, bu alanda daha önceden yapılan çalışmalar referans alınarak çözümler arası farkların ortalaması ile yapılmıştır. Elde edilen ortalamalar reaktif tabu arama (RTA) (Battiti ve Tecchiolli, 1994), TA (Taillard, 1991), genetik melez metot (GMM) (Fleurent ve Ferland, 1994), TB (Connolly, 1990) ve MKA (Gambardella vd., 1997) meta-sezgiselleri ile elde edilen sonuçlara ile karşılaştırılmıştır.

Yapılan değerlendirmede, öncelikle elde edilen bütün sonuçların ortalaması ele alınmış ve algoritmanın 3 problem için en iyi sonucu verdiği görülmüştür. Yapılan deneylerden her bir

problem için elde edilen en iyi 10 sonuç ele alındığında, algoritma 8 problem için en iyi sonucu vermiştir.

İleride yapılması düşünülen çalışmalar; öncelikle KAP için TavlamalıKarıncı paralel algoritmasının geliştirilmesidir. Geliştirilecek algorithmada frekans matrisine benzer bir hafıza modülü kullanılacak ve fenomen iz matrisi limitlerinden faydalanılacaktır. Bunun dışında “TavlamalıKarıncı” algoritmasının EİS’de tek sıralı tesis yerleşimi problemi için yeniden tasarlanması planlanmaktadır. Bugüne kadar KA ile yapılan çalışmalarda yer alamayan ve önemli bir kombinatoriyal optimizasyon problem çeşidi olan Montaj Hattı Dengeleme problemine KA uygulaması düşünülmektedir. Son olarak da stokastik optimizasyon problemleri içinde KA uygulamaları planlanmaktadır.



KAYNAKLAR

Ant Colony Optimization, <http://iridia.ulb.ac.be/~mdorigo/ACO/publications.html>.

Armour, G.C., Bufa, E.S. (1963), "A Heuristic Algorithm and Simulation Approach to Relative Allocation of Facilities", *Management Science*, 9:294-309.

Battiti, R. ve Tecchiolli, G. (1994), "The Reactive Tabu Search", *ORSA Journal on Computing*, 6:126-140.

Bazaraa, M.S. ve Elshafei, A.N. (1979), "An Exact Branch and Bound Procedure for Quadratic Assignment Problems", *Naval Research Logistics Quarterly*, 26:109-121.

Bazaraa, M.S. ve Sherali, A.N. (1980), "Bender's Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem", *Naval Research Logistics Quarterly*, 27(1):29-41.

Bonomi, E. ve Lutton, J.L. (1986), "The Asymptotic Behavior of Quadratic Sum Assignment Problems: A Statistical Mechanics Approach", *European Journal of Operational Research*, 26:295-300.

Bullnheimer, B., Kotsis, G. ve Strauss, C. (1998), "Parallelization Strategies for the Ant System", *High Performance Algorithms and Software in Nonlinear Optimization*; series: *Applied Optimization*, 24:87-100.

Bullnheimer, B., Hartl, R.F. ve Strauss, C., (1999), "An Improved Ant system Algorithm for the Vehicle Routing Problem", *Sixth Viennese workshop on Optimal Control, Dynamic Games, Nonlinear Dynamics and Adaptive Systems*, May 21-23 1997, Vienna (Austria).

Burkard, R.E. ve Stratmann, K.H. (1978), "Numerical Investigation on Quadratic Assignment Problems", *Naval Research Logistic Quarterly*, 25:129-148.

Burkard, R.E. ve Rend, F. (1984), "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems", *European Journal of Operational Research*, 17:169-174.

Burkard, R.E., Karisch, S.E. ve Rendl, F. (1991), "QAPLIB-A Quadratic Assignment Problem Library", *European Journal of Operational Research*, 55:115-119.

Cerny, V. (1985), "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, 45:41-51.

Chiang, W.-C., Chiang, C. (1998), "Intelligent Local Search Strategies for Solving Facility Layout Problems with The Quadratic Assignmet Problem Formulation", *European Journal of Operation Research*, 106:457-488.

Colorni, A., Dorigo, M. ve Maniezzo, V., (1992), "Distributed Optimization by Ant Colonies", *Proceedings of the First European Conference on Artificial Life*, Paris.

Colorni, A., Dorigo, M., Maniezzo, V. ve Trubian, M., (1994), "Ant System for Job-Shop Scheduling", *JORBEL-Belgian Journal of Operations Research*, 34(1):39-53.

- Colomi, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini ve G., Trubian, M. (1996), "Heuristics from Nature for Hard Combinatorial Problems", *International Transactions in Operational Research*, 3(1):1-21.
- Connolly D.T. (1990), "An Improved Annealing Scheme for the QAP", *European Journal of Operational Research*, 46:93-100.
- Di Caro, G. ve Dorigo, M. (1998), "AntNet: Distributed Stigmergetic Control for Communications Networks", *Journal of Artificial Intelligence Research (JAIR)*, 9:317-365.
- Dorigo, M., Maniezzo, V. ve Colomi, A. (1991), "Positive Feedback as a Search Strategy" Technical Report No. 91-016, Politecnico di Milano, Italy.
- Dorigo, M. (1992), "Optimization, Learning and Natural Algorithms", Ph.D.Thesis, Politecnico di Milano, Italy.
- Dorigo, M., Maniezzo, V. ve Colomi, A. (1996), "The Ant System: Optimization by a Colony Of Cooperating Agents", *IEEE Transaction Systems Man Cybernet B*, 26:29-42.
- Dorigo, M. ve Gambardella, L.M. (1997), "Ant Colonies for the Traveling Salesman Problem", *BioSystems*, 43:73-81.
- Dorigo, M. ve Gambardella, L.M. (1997), "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, 1(1):53-66.
- Eggers, J., Feillet, D., Kehl, S., Wagner, M. O. ve Yannou, B. (2003), "Optimization of The Keyboard Arrangement Problem Using an Ant Colony Algorithm", *European Journal of Operational Research*, 148:Pages 672-686.
- Finke, G., Burkard R.E. ve Rendl, F. (1984), "Quadratic Assignment Problem", *Annals of Discrete Mathematics*, 31:61-82.
- Fleurent, C. ve Ferland, J.A. (1994), "Genetic Hybrids for the Quadratic Assignment Problem", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16:173-188.
- Forsyth, P. ve Wren, A., (1997), "An Ant System for Bus Driver Scheduling", Presented at the 7th International Workshop on Computer-Aided Scheduling of Public Transport, August 1997, Boston.
- Francis, R.L., McGinnis, L.F. ve White, J.A., (1992), "Facility Layout and Location: An Analytical Approach", Prentice-Hall, Englewood Cliffs, NJ.
- Gambardella, L. M. ve Dorigo, M. (1997), "HAS-SOP: An Hybrid Ant System for The Sequential Ordering Problem", Tech. Rep. No. IDSIA 97-11.
- Gambardella, L. M., Taillard, E. ve Dorigo, M. (1997), "Ant Colonies for the Quadratic Assignment Problem", Technical Report 97-4, IDSIA, Lugano, Switzerland.

- Gambardella, L. M., Taillard, E. ve Dorigo, M. (1999), "Ant Colonies for the Quadratic Assignment Problem", *Journal of the Operational Research Society*, 50:167-176.
- Gilmore, P.C. (1963), "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem", *SIAM Journal*, 10(2):305-313.
- Glover, F. (1989), "Tabu Search-I", *ORSA Journal of Computing*, 1(3):190-206.
- Gutjahr, W.J. (2000), "A Graph-Based Ant System and Its Convergence", *Future Generation Computer Systems*, 16:873-888.
- Heragu, S.S., Kusiak, A. (1991), "Efficient Models for The Facility Layout Problem", *European Journal of Operational Research*, 53:1-13.
- Jajodia, S., Minis, I., Harhalakia, G. ve Proth, J.-M. (1992), "CLASS: Computerized Layout Solutions Using Simulated Annealing", *International Journal of Production Research*, 30(1):95-108.
- Kelly, J.P., Laguna, M. ve Glover, F. (1994), "A Study of Diversification Strategies for the Quadratic Assignment Problem", *Computers and operations research*, 21(8):885-893.
- Kirkpatrick, S., Gelatt, Jr. ve Vecchi, M. P. (1983), "Optimization by Simulated Annealing", *Science*, 220:671-680.
- Koopmans, T.C. ve Beckmann, M.J. (1957), "Assignment Problems and The Location of Economic Activities", *Econometrica*, 25:53-76.
- Kouvelis, P. ve Chiang, W.C. (1992), "A Simulated Annealing Procedure for Single Row Layout Problems in Flexible Manufacturing Systems", *International Journal of Production Research*, 30(4):717-732.
- Kouvelis, P., Chiang, W.C. ve Fitzsimmons, J. (1992), "Simulated Annealing for Machine Layout Problems in the Presence of Zoning Constraints", *European Journal of Operational Research*, 57:203-223.
- Kulluk, S. (2003), "Tesis Yerleşimi Problemlerinde Genetik Algoritmalar ve Bir Genetik Algoritma Uygulaması", Yüksek Lisans Tezi, G.Ü. Fen Bilimleri Enstitüsü, Ankara.
- Kuntz, P., Layzell, P. ve Snyers, D., (1997), "A Colony of Ant-like Agents for Partitioning in VLSI Technology", *Proceedings of the Fourth European Conference on Artificial Life*, 417-424, London.
- Kusiak, A. ve Herdgu, S.S. (1987), "The Facility Layout Problem", *European Journal of Operational Research*, 29 (3):229-251.
- Kusiak, A., (1990), "Intelligent Manufacturing Systems", Prentice-Hall, Englewood Cliffs. NJ.
- Lawler, E.L. (1963), "The Quadratic Assignment Problem", *Management Science*, 9 (4):586-599.

Lee, Z., Lee C. ve Su, S. (2002), "An Immunity-Based Ant Colony Optimization Algorithm for Solving Weapon-Target Assignment Problem", *Applied Soft Computing*, 2:39-47.

Lundy, M. ve Mees, A. (1986), "Convergence of an Annealing Algorithm", *Mathematical Programming*, 34:111-124.

Maniezzo, V., Colomi, A. ve Dorigo, M. (1994), "The Ant System Applied to the Quadratic Assignment Problem", Tech. Rep. IRIDIA/94-28.

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

Mirchandani, P.B. ve Francis, R.L. (1990), "Discrete Location Theory", John Wiley & Sons, New York.

Navarro Varela, G. ve Sinclair, M.C., (1999), "Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation", *Proceedings of the Congress on Evolutionary Computation (CEC'99)*, , July 1999, Washington DC.

Nugent, C.E., Vollmann, T.E. ve Ruml, J. (1968), "An Experimental Comparison of Techniques for The Assignment of Facilities to Locations", *Operations Research*, 150-173.

QAP Library, <http://www.opt.math.tu-graz.ac.at/qaplib>.

Randall, M. ve Lewis, A. (2002), "A Parallel Implementation of Ant Colony Optimization", *Journal of Parallel and Distributed Computing*, 62(9):1421-1432.

Resende, M.G.C., Ramakrishnan, K.G. ve Drezner, Z. (1995), "Computing Lower Bounds for the Quadratic Assignment Problem with an Interior Point Algorithm for Linear Programming", *Operations Research*, 43 (5):781-791.

Rosenblatt, M.J. (1979), "The Facilities Layout Problem: A Multi-goal Approach", *International Journal of Production Research*, 17: 323-332.

Rosenblatt, M.J. (1986), "The Dynamics of Plant Layout", *Management Science*, 32(1):76-86.

Sahni, S. ve Gonzalez, T. (1976), "P-Complete Approximation Problem", *Journal of the Association for Computing Machinery*, 23 (3):555-565.

Schoonderwoerd, R., Holland, O., Bruten, J. ve Rothkrantz, L. (1997), "Ant-Based Load Balancing in Telecommunications Networks", *Adaptive Behavior*, 5(2):169-207.

Sha, D.Y. ve Chen, C.-W. (2001), "A New Approach to The Multiple Objective Facility Layout Problem", *Integrated Manufacturing Systems*, 12(1):59-66.

Sharpe, R. ve Marksjo, B.S. (1985), "Facility Layout Optimization Using the Metropolis Algorithm", *Environmental and Planning*, 12:443-453.

- Sharpe, R., Marksjo, B.S., Mitchell, J.R. ve Crawford, J.R. (1985), "An Interactive Model for the Layout of Buildings", *Applied Mathematical Modeling*, 9(3):207-7214.
- Skorin-Kapov, J. (1994), "Extensions of a Tabu Search Adaptation to the Quadratic Assignment Problem", *Computers and Operations Research*, 21(8):855-865.
- Solimanpur, M., Vrat, P. ve Shankar R. (2003), "An Ant Algorithm for The Single Row Layout Problem in Flexible Manufacturing Systems", *Computers & Operations Research*, In Press, Corrected Proof.
- Stützle, T. ve Dorigo, M. (1999), "ACO Algorithms for the Quadratic Assignment Problem", *New Ideas in Optimization*, McGraw-Hill, Tech.Rep.IRIDIA/99-2.
- Stützle, T. ve Hoos, H. H. (2000), "Max-Min Ant System", *Future Generation Computer Systems*, 16:889-914.
- Sule, D.R., (1988), "Manufacturing Facilities Location", *Planning and Design*, PWS Kent MA.
- Taillard E. (1991), "Robust Tabu Search for the Quadratic Assignment Problem", *Parallel Computing*, 17:443-455.
- Taillard, E. ve Gambardella, L. M. (1997), "An Ant Approach for Structured Quadratic Assignment Problems", 2nd Metaheuristics International Conference, July 21-24 1997, Sophia-Antipolis.
- Talbi, E.G., Roux, O., Fonlupt, C. ve Robillard, D. (2001), "Parallel Ant Colonies for the quadratic assignment problem", *Future Generation Computer Systems*, 17:441-449.
- Tam, K.Y. (1992), "Genetic Algorithms, Function Optimization, and Facility Layout Design", *European Journal of Operational Research*, 63:322-346.
- Tate, D.M., Smith, A.E. (1994), "Unequal Area Facility Layout Using Genetic Search", *IIE Transactions*.
- Tavakkoli-Moghaddain, R. ve Shayan, E. (1998), "Facilities Layout Design By Genetic Algorithms", *Computers and Industrial Engineering*, 35:527-530.
- T'kindt, V., Monmarché, N., Tercinet, F. ve Laügt, D. (2002), "An Ant Colony Optimization Algorithm to Solve a 2-Machine Bicriteria Flowshop Scheduling Problem", *European Journal of Operational Research*, 142:250-257.
- Whitley, D., Starkweather, T. ve Fuquay, D., (1989) "Scheduling Problems and Travelling Salesman: the Genetic Edge Recombination Operator", *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, San Francisco.
- Wilhelm, M.R. ve Ward, T.L. (1987), "Solving Quadratic Assignment Problems by Simulated Annealing", *IIE Transactions*, 19:107-119.
- Yigit, V. ve Türkbey O. (2003), "Tesis Yerleşim Problemlerine Sezgisel Metotlarla Yaklaşım", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 18:45-56.

EKLER**EK 1 : Düzenli Test Problemleri İçin TavlamalıKarınca Algoritması Sonuçları****EK 2 : Düzensiz Test Problemleri İçin TavlamalıKarınca Algoritması Sonuçları**

EK 1

**DÜZENLİ TEST PROBLEMLERİ İÇİN TAVLAMALIKARINCA ALGORİTMASI
SONUÇLARI**

NUG20 PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	2570
2	10	12	12	0,2	100	0	2570
3	10	8	12	0,1	100	0	2570
4	10	8	10	0,1	100	0,015	2609
5	10	12	12	0,1	200	0	2570
6	10	12	10	0,2	100	0	2570
7	13	8	10	0,1	200	0	2570
8	13	12	10	0,1	100	0	2570
9	10	8	12	0,1	200	0	2570
10	10	12	12	0,1	100	0	2570
11	13	8	10	0,2	100	0,017	2614
12	10	12	12	0,2	200	0	2570
13	10	8	12	0,2	200	0,015	2609
14	13	12	10	0,1	200	0	2570
15	13	12	12	0,2	200	0	2570
16	13	12	12	0,2	100	0,002	2575
17	13	12	12	0,1	200	0	2570
18	10	8	10	0,2	200	0	2570
19	10	12	10	0,1	100	0	2570
20	13	8	12	0,2	100	0,007	2588
21	13	12	10	0,2	200	0	2570
22	13	8	12	0,1	200	0	2570
23	10	8	12	0,2	100	0,035	2660
24	13	12	10	0,2	100	0,004	2580
25	10	12	10	0,2	200	0	2570
26	13	8	12	0,2	200	0	2570
27	10	8	10	0,2	100	0,015	2609
28	13	8	12	0,1	100	0	2570
29	10	8	10	0,1	200	0	2570
30	13	8	10	0,1	100	0	2570
31	13	8	10	0,2	200	0	2570
32	10	12	10	0,1	200	0	2570

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas Sayısı

α : Fenomen güncelleme katsayı

I : Yerel Aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

NUG30 PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	Imax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0,005	6155
2	10	12	12	0,2	100	0,007	6167
3	10	8	12	0,1	100	0,019	6240
4	10	8	10	0,1	100	0,013	6204
5	10	12	12	0,1	200	0,004	6148
6	10	12	10	0,2	100	0	6124
7	13	8	10	0,1	200	0,009	6179
8	13	12	10	0,1	100	0,008	6173
9	10	8	12	0,1	200	0,015	6216
10	10	12	12	0,1	100	0,003	6142
11	13	8	10	0,2	100	0,022	6259
12	10	12	12	0,2	200	0,003	6142
13	10	8	12	0,2	200	0,012	6197
14	13	12	10	0,1	200	0,001	6130
15	13	12	12	0,2	200	0,004	6148
16	13	12	12	0,2	100	0,006	6161
17	13	12	12	0,1	200	0,005	6155
18	10	8	10	0,2	200	0,017	6228
19	10	12	10	0,1	100	0	6124
20	13	8	12	0,2	100	0,009	6179
21	13	12	10	0,2	200	0,005	6155
22	13	8	12	0,1	200	0,011	6191
23	10	8	12	0,2	100	0,011	6191
24	13	12	10	0,2	100	0,009	6179
25	10	12	10	0,2	200	0,002	6136
26	13	8	12	0,2	200	0,02	6246
27	10	8	10	0,2	100	0,014	6210
28	13	8	12	0,1	100	0,027	6289
29	10	8	10	0,1	200	0,014	6210
30	13	8	10	0,1	100	0,024	6271
31	13	8	10	0,2	200	0,027	6289
32	10	12	10	0,1	200	0	6124

m : Karınca sayısı

Imax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

TAI30a PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	1818146
2	10	12	12	0,2	100	0,018	1850873
3	10	8	12	0,1	100	0,042	1894508
4	10	8	10	0,1	100	0,035	1881781
5	10	12	12	0,1	200	0	1818146
6	10	12	10	0,2	100	0,021	1856327
7	13	8	10	0,1	200	0,028	1869054
8	13	12	10	0,1	100	0,028	1869054
9	10	8	12	0,1	200	0,035	1881781
10	10	12	12	0,1	100	0,013	1841782
11	13	8	10	0,2	100	0,032	1876327
12	10	12	12	0,2	200	0,012	1839964
13	10	8	12	0,2	200	0,038	1887236
14	13	12	10	0,1	200	0	1818146
15	13	12	12	0,2	200	0,019	1852691
16	13	12	12	0,2	100	0,021	1856327
17	13	12	12	0,1	200	0	1818146
18	10	8	10	0,2	200	0,041	1892690
19	10	12	10	0,1	100	0,011	1838146
20	13	8	12	0,2	100	0,037	1885417
21	13	12	10	0,2	200	0,022	1858145
22	13	8	12	0,1	200	0,027	1867236
23	10	8	12	0,2	100	0,043	1896326
24	13	12	10	0,2	100	0,021	1856327
25	10	12	10	0,2	200	0,014	1843600
26	13	8	12	0,2	200	0,038	1887236
27	10	8	10	0,2	100	0,036	1883599
28	13	8	12	0,1	100	0,045	1899963
29	10	8	10	0,1	200	0,032	1876327
30	13	8	10	0,1	100	0,04	1890872
31	13	8	10	0,2	200	0,035	1881781
32	10	12	10	0,1	200	0	1818146

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

TAI60a PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0,013	7302283
2	10	12	12	0,2	100	0,82	13119601
3	10	8	12	0,1	100	1,788	20097499
4	10	8	10	0,1	100	1,785	20075873
5	10	12	12	0,1	200	0,01	7280658
6	10	12	10	0,2	100	0,955	14092758
7	13	8	10	0,1	200	1,77	19967744
8	13	12	10	0,1	100	0,845	13299815
9	10	8	12	0,1	200	1,585	18634159
10	10	12	12	0,1	100	0,595	11497672
11	13	8	10	0,2	100	1,45	17661001
12	10	12	12	0,2	200	0,854	13364692
13	10	8	12	0,2	200	1,72	19607316
14	13	12	10	0,1	200	0,01	7280658
15	13	12	12	0,2	200	0,42	10236172
16	13	12	12	0,2	100	0,855	13371901
17	13	12	12	0,1	200	0,021	7359952
18	10	8	10	0,2	200	1,855	20580473
19	10	12	10	0,1	100	0,505	10848901
20	13	8	12	0,2	100	1,975	21445502
21	13	12	10	0,2	200	0,298	9356726
22	13	8	12	0,1	200	1,208	15916527
23	10	8	12	0,2	100	1,945	21229245
24	13	12	10	0,2	100	0,101	7936638
25	10	12	10	0,2	200	0,64	11822058
26	13	8	12	0,2	200	1,72	19607316
27	10	8	10	0,2	100	1,63	18958544
28	13	8	12	0,1	100	2,035	21878016
29	10	8	10	0,1	200	1,968	21395042
30	13	8	10	0,1	100	1,92	21049030
31	13	8	10	0,2	200	1,088	15051498
32	10	12	10	0,1	200	0,188	8563784

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

SKO64 PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0,001	48546
2	10	12	12	0,2	100	0,001	48546
3	10	8	12	0,1	100	0,004	48692
4	10	8	10	0,1	100	0,005	48740
5	10	12	12	0,1	200	0,001	48546
6	10	12	10	0,2	100	0,001	48546
7	13	8	10	0,1	200	0,003	48643
8	13	12	10	0,1	100	0	48498
9	10	8	12	0,1	200	0,003	48643
10	10	12	12	0,1	100	0,002	48595
11	13	8	10	0,2	100	0,005	48740
12	10	12	12	0,2	200	0,001	48546
13	10	8	12	0,2	200	0,003	48643
14	13	12	10	0,1	200	0,002	48595
15	13	12	12	0,2	200	0	48498
16	13	12	12	0,2	100	0,001	48546
17	13	12	12	0,1	200	0,002	48595
18	10	8	10	0,2	200	0,003	48643
19	10	12	10	0,1	100	0,002	48595
20	13	8	12	0,2	100	0,004	48692
21	13	12	10	0,2	200	0,001	48546
22	13	8	12	0,1	200	0,005	48740
23	10	8	12	0,2	100	0,006	48789
24	13	12	10	0,2	100	0,001	48546
25	10	12	10	0,2	200	0	48498
26	13	8	12	0,2	200	0,003	48643
27	10	8	10	0,2	100	0,005	48740
28	13	8	12	0,1	100	0,003	48643
29	10	8	10	0,1	200	0,004	48692
30	13	8	10	0,1	100	0,003	48643
31	13	8	10	0,2	200	0,003	48643
32	10	12	10	0,1	200	0,001	48546

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

EK 2

**DÜZENSİZ TEST PROBLEMLERİ İÇİN TAVLAMALIKARINCA ALGORİTMASI
SONUÇLARI**

BUR26a PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0,015	5508070
2	10	12	12	0,2	100	0	5426670
3	10	8	12	0,1	100	0,024	5556910
4	10	8	10	0,1	100	0,014	5502643
5	10	12	12	0,1	200	0,007	5464657
6	10	12	10	0,2	100	0,005	5453803
7	13	8	10	0,1	200	0,017	5518923
8	13	12	10	0,1	100	0,015	5508070
9	10	8	12	0,1	200	0,016	5513497
10	10	12	12	0,1	100	0	5426670
11	13	8	10	0,2	100	0,031	5594897
12	10	12	12	0,2	200	0,01	5480937
13	10	8	12	0,2	200	0,002	5437523
14	13	12	10	0,1	200	0,011	5486363
15	13	12	12	0,2	200	0	5426670
16	13	12	12	0,2	100	0,002	5437523
17	13	12	12	0,1	200	0,009	5475510
18	10	8	10	0,2	200	0,018	5524350
19	10	12	10	0,1	100	0	5426670
20	13	8	12	0,2	100	0,029	5584043
21	13	12	10	0,2	200	0,02	5535203
22	13	8	12	0,1	200	0,017	5518923
23	10	8	12	0,2	100	0,012	5491790
24	13	12	10	0,2	100	0,019	5529777
25	10	12	10	0,2	200	0,001	5432097
26	13	8	12	0,2	200	0,002	5437523
27	10	8	10	0,2	100	0,015	5508070
28	13	8	12	0,1	100	0,027	5573190
29	10	8	10	0,1	200	0,015	5508070
30	13	8	10	0,1	100	0,024	5556910
31	13	8	10	0,2	200	0,027	5573190
32	10	12	10	0,1	200	0,001	5432097

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

BUR26h PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	7098658
2	10	12	12	0,2	100	0,004	7127053
3	10	8	12	0,1	100	0,016	7212237
4	10	8	10	0,1	100	0,012	7183842
5	10	12	12	0,1	200	0	7098658
6	10	12	10	0,2	100	0,005	7134151
7	13	8	10	0,1	200	0,009	7162546
8	13	12	10	0,1	100	0,008	7155447
9	10	8	12	0,1	200	0,012	7183842
10	10	12	12	0,1	100	0	7098658
11	13	8	10	0,2	100	0,011	7176743
12	10	12	12	0,2	200	0	7098658
13	10	8	12	0,2	200	0,014	7198039
14	13	12	10	0,1	200	0	7098658
15	13	12	12	0,2	200	0,004	7127053
16	13	12	12	0,2	100	0,005	7134151
17	13	12	12	0,1	200	0	7098658
18	10	8	10	0,2	200	0,015	7205138
19	10	12	10	0,1	100	0	7098658
20	13	8	12	0,2	100	0,013	7190941
21	13	12	10	0,2	200	0	7098658
22	13	8	12	0,1	200	0,008	7155447
23	10	8	12	0,2	100	0,016	7212237
24	13	12	10	0,2	100	0,004	7127053
25	10	12	10	0,2	200	0	7098658
26	13	8	12	0,2	200	0,014	7198039
27	10	8	10	0,2	100	0,01	7169645
28	13	8	12	0,1	100	0,017	7219335
29	10	8	10	0,1	200	0,011	7176743
30	13	8	10	0,1	100	0,012	7183842
31	13	8	10	0,2	200	0,01	7169645
32	10	12	10	0,1	200	0	7098658

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

TAI30b PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	637117113
2	10	12	12	0,2	100	0,013	645399635
3	10	8	12	0,1	100	0,028	654956392
4	10	8	10	0,1	100	0,028	654956392
5	10	12	12	0,1	200	0	637117113
6	10	12	10	0,2	100	0,015	646673870
7	13	8	10	0,1	200	0,028	654956392
8	13	12	10	0,1	100	0,013	645399635
9	10	8	12	0,1	200	0,025	653045041
10	10	12	12	0,1	100	0,009	642851167
11	13	8	10	0,2	100	0,023	651770807
12	10	12	12	0,2	200	0,014	646036753
13	10	8	12	0,2	200	0,027	654319275
14	13	12	10	0,1	200	0	637117113
15	13	12	12	0,2	200	0,007	641576933
16	13	12	12	0,2	100	0,014	646036753
17	13	12	12	0,1	200	0	637117113
18	10	8	10	0,2	200	0,029	655593509
19	10	12	10	0,1	100	0,008	642214050
20	13	8	12	0,2	100	0,031	656867744
21	13	12	10	0,2	200	0,005	640302699
22	13	8	12	0,1	200	0,019	649222338
23	10	8	12	0,2	100	0,031	656867744
24	13	12	10	0,2	100	0,002	638391347
25	10	12	10	0,2	200	0,01	643488284
26	13	8	12	0,2	200	0,027	654319275
27	10	8	10	0,2	100	0,026	653682158
28	13	8	12	0,1	100	0,032	657504861
29	10	8	10	0,1	200	0,031	656867744
30	13	8	10	0,1	100	0,03	656230626
31	13	8	10	0,2	200	0,017	647948104
32	10	12	10	0,1	200	0,003	639028464

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

TAI40b PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	637250948
2	10	12	12	0,2	100	0,002	638525450
3	10	8	12	0,1	100	0,015	646809712
4	10	8	10	0,1	100	0,012	644897959
5	10	12	12	0,1	200	0	637250948
6	10	12	10	0,2	100	0,006	641074454
7	13	8	10	0,1	200	0,008	642348956
8	13	12	10	0,1	100	0,009	642986207
9	10	8	12	0,1	200	0,011	644260708
10	10	12	12	0,1	100	0,001	637888199
11	13	8	10	0,2	100	0,01	643623457
12	10	12	12	0,2	200	0	637250948
13	10	8	12	0,2	200	0,014	646172461
14	13	12	10	0,1	200	0,001	637888199
15	13	12	12	0,2	200	0,005	640437203
16	13	12	12	0,2	100	0,005	640437203
17	13	12	12	0,1	200	0,001	637888199
18	10	8	10	0,2	200	0,014	646172461
19	10	12	10	0,1	100	0	637250948
20	13	8	12	0,2	100	0,013	645535210
21	13	12	10	0,2	200	0,002	638525450
22	13	8	12	0,1	200	0,007	641711705
23	10	8	12	0,2	100	0,015	646809712
24	13	12	10	0,2	100	0,004	639799952
25	10	12	10	0,2	200	0,002	638525450
26	13	8	12	0,2	200	0,012	644897959
27	10	8	10	0,2	100	0,011	644260708
28	13	8	12	0,1	100	0,015	646809712
29	10	8	10	0,1	200	0,01	643623457
30	13	8	10	0,1	100	0,012	644897959
31	13	8	10	0,2	200	0,011	644260708
32	10	12	10	0,1	200	0,001	637888199

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

TAI80b PROBLEMİ İÇİN ÇÖZÜMLER ARASI FARKLAR

Sıra	m	lmax	T	alfa	I	Çözümler Arası Fark	Çözüm Değeri
1	13	12	12	0,1	100	0	818415043
2	10	12	12	0,2	100	0,727	1413402779
3	10	8	12	0,1	100	1,697	2207265371
4	10	8	10	0,1	100	1,414	1975653914
5	10	12	12	0,1	200	0	818415043
6	10	12	10	0,2	100	0,848	1512430999
7	13	8	10	0,1	200	1,131	1744042457
8	13	12	10	0,1	100	1,131	1744042457
9	10	8	12	0,1	200	1,414	1975653914
10	10	12	12	0,1	100	0,525	1248082941
11	13	8	10	0,2	100	1,293	1876625694
12	10	12	12	0,2	200	0,485	1215346339
13	10	8	12	0,2	200	1,535	2074682134
14	13	12	10	0,1	200	0	818415043
15	13	12	12	0,2	200	0,768	1446957796
16	13	12	12	0,2	100	0,848	1512430999
17	13	12	12	0,1	200	0	818415043
18	10	8	10	0,2	200	1,656	2173710354
19	10	12	10	0,1	100	0,444	1181791322
20	13	8	12	0,2	100	1,495	2041945532
21	13	12	10	0,2	200	0,889	1545986016
22	13	8	12	0,1	200	1,091	1711305855
23	10	8	12	0,2	100	1,737	2240001973
24	13	12	10	0,2	100	0,848	1512430999
25	10	12	10	0,2	200	0,566	1281637957
26	13	8	12	0,2	200	1,535	2074682134
27	10	8	10	0,2	100	1,454	2008390516
28	13	8	12	0,1	100	1,818	2306293591
29	10	8	10	0,1	200	1,293	1876625694
30	13	8	10	0,1	100	1,616	2140973752
31	13	8	10	0,2	200	1,414	1975653914
32	10	12	10	0,1	200	0	818415043

m : Karınca sayısı

lmax : Algoritmanın iterasyon sayısı

T : Takas sayısı

α : Fenomen güncelleme katsayı

I : Yerel aramanın iterasyon sayısı

Çözümler Arası Fark : (Bilinen en iyi çözüm - Bulunan çözüm) / Bilinen en iyi çözüm

ÖZGEÇMİŞ

Doğum tarihi	11.09.1979	
Doğum yeri	Kayseri	
Lise	1993-1996	Kayseri Sümer Lisesi
Lisans	1997-2002	Erciyes Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü
Yüksek Lisans	2002-2004	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Sistem Mühendisliği Programı

Çalıştığı Kurumlar

2002-Devam Ediyor Erciyes Üniversitesi Mühendislik Fakültesi
Araştırma Görevlisi