

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**HİPERSEZGİSEL YÖNTEMLERLE LOJİSTİK AĞ TASARIMI VE
OPTİMİZASYON**



DOKTORA TEZİ

Vural EROL

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

EKİM 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**HİPERSEZGİSEL YÖNTEMLERLE LOJİSTİK AĞ TASARIMI VE
OPTİMİZASYON**

DOKTORA TEZİ

**Vural EROL
(507062110)**

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

**Tez Danışmanı: Doç. Dr. Murat BASKAK
Eş Danışman: Prof. Dr. Gülgün KAYAKUTLU**

EKİM 2017

İTÜ, Fen Bilimleri Enstitüsü'nün 507062110 numaralı Doktora Öğrencisi Vural EROL, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “HİPERSEZGİSEL YÖNTEMLERLE LOJİSTİK AĞ TASARIMI VE OPTİMİZASYON” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Murat BASKAK**
İstanbul Teknik Üniversitesi

Eş Danışman : **Prof. Dr. Gülgün KAYAKUTLU**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Şeyda Serdar ASAN**
İstanbul Teknik Üniversitesi

Doç. Dr. Özgür KABAK
İstanbul Teknik Üniversitesi

Prof. Dr. Mehmet TANYAŞ
Maltepe Üniversitesi

Doç. Dr. Ahmet BEŞKESE
Bahçeşehir Üniversitesi

Prof. Dr. Gülçin BÜYÜKÖZKAN FEYZİOĞLU
Galatasaray Üniversitesi

Teslim Tarihi : 15 Eylül 2017
Savunma Tarihi : 17 Ekim 2017





Aileme,



ÖNSÖZ

Zor günler yaşadığımız bu günlerden uzun vâdede ülkemizi refaha çıkartacak tek yol nitelikli eğitimidir. Kapsamlı ve kaliteli öğretim veren kurumlar için öncelikle donanımlı öğretmenlere, bilimsel, sosyal ve özgür olanaklara, çalışkan, açık görüşlü ve araştırmacı ruhlu öğrencilere gereksinim vardır. Kaliteli eğitim alan bireyler, sorgulama ve yenilikçilik yönlerini geliştirerek topluma katkı sağlayacak seçim ve faaliyetlerde bulunurlar. Özellikle ekonomik anlamda gelişme sağlamada mutlak gerekli olan küresel markaları oluşturabilmek için de yenilikçilik kültürünün toplumda benimsenmesi, ancak nitelikli eğitim ile olanaklı olabilmektedir. Son yıllarda bilişim teknolojilerinde yaşanan ilerlemeler, yenilikçi düşüncelerin uygulamaya dönüşmesini hızlandırarak, günlük yaşamda ve iş yaşamında karşılaşılan gereksinimlerin giderilmesinde ve sorunların çözülmesinde kolaylıklar sağlamaktadır. Bu çalışmada da Tedârik Zinciri Yönetimi için bilgi teknolojilerinin kullanıldığı yenilikçi bir yöntem ele alınmıştır.

Bu tez çalışmasında literatürde en zor kombinatoriyal optimizasyon problemlerinden biri olan Lojistik Ağ Tasarımı problemi için Hipersezgisel Yöntemler kullanılarak bir teknik geliştirilmiş, sanal ve gerçek problemler üzerinde denemeler yapılarak sonuçlar karşılaştırılmıştır. Hipersezgisel Yöntemler, problem çözmede problemin topolojisinden öte kullanılan yöntemler üzerinde de soyutlama sağlayarak optimizasyon sürecinde ölçeklenebilir uygulamaların hazırlanmasına olanak tanımaktadır.

Bu tez çalışmasında kullandığım yöntemle tanışmamı sağlayan Prof. Dr. Haluk TOPÇUOĞLU'na, tez için uygulama yapmamı sağlayan Ali YILMAZ'a, lisans eğitimimden beri birlikte çalıştığım, bana yol gösterici ve destek olan saygıdeğer danışman hocam Doç. Dr. Murat BASKAK'a, bilgi ve deneyimleri ile bana sürekli katkı sağlayan, çalışmalarımı titizlikle izleyen saygıdeğer eş danışman hocam Prof. Dr. Gülgün KAYAKUTLU'ya ve şu ana kadar nitelikli eğitim almamı sağlayan değerli öğretmenlerime içten teşekkür ederim. Bu çalışma süresince manevî destekleriyle beni hiç bir zaman yalnız bırakmayan aileme de sonsuz teşekkürü borç bilirim.

Eylül 2017

Vural Erol
(Yüksek Endüstri Mühendisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET.....	xix
SUMMARY	xxiii
1. GİRİŞ.....	1
1.1 Tezin Kapsamı ve Amacı	1
2. LOJİSTİK AĞ TASARIMI PROBLEMİ.....	5
2.1 İleri Lojistik Ağ Tasarımına İlişkin Çalışmalar	9
2.2 Tersine Lojistik Ağ Tasarımına İlişkin Yapılan Çalışmalar	20
2.3 Bütünleşik Lojistik Ağ Tasarımına İlişkin Yapılan Çalışmalar	25
2.4 Lojistik Ağ Tasarımı İçin Kullanılan Yöntemler	29
2.4.1 Genetik Algoritmalar	30
2.4.2 Tavlama Benzetimi Algoritması	33
2.4.3 Tabu Arama Algoritması.....	37
2.4.4 Karınca Kolonisi Optimizasyon Algoritması.....	39
2.5 Lojistik Ağ Tasarımına İlişkin Çalışmaların Özeti	44
2.6 Lojistik Ağ Tasarımı Problemi Hakkındaki Çalışmalardan Çıkarımlar.....	60
2.7 Günümüzde Ağ İşletmecilerinin Sorun Olarak Yaşadığı Durumlar	61
3. ELE ALINAN LOJİSTİK AĞ TASARIMI PROBLEMİ VE ÇÖZÜMÜ İÇİN BİR MODEL ÖNERİSİ.....	63
3.1 Problemin Tanımı.....	63
3.2 Oluşturulan Model.....	63
3.2.1 Varsayımlar	63
3.2.2 Karar değişkenleri ve parametreler	64
3.2.3 Amaç fonksiyonu	65
3.2.4 Kısıtlar.....	65
3.2.5 Önerilen Karma Tamsayılı Programlama Modeli.....	68
4. HİPERSEZGİSEL ALGORİTMALAR.....	71
4.1 Yöntem tanımı.....	71
4.2 Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritmalar	77
4.3 Metasezgisel Yöntemlerle Melezleştirilen Hipersezgisel Algoritmalar.....	86
4.4 Öğrenme Tabanlı Hipersezgisel Algoritmalar	103
4.5 Hipersezgisel Algoritmalarla İlişkin Çalışmaların Özeti	107
4.6 Hipersezgisel Algoritmalarla ilgili basit bir uygulama	109
4.7 Hipersezgisel Algoritmalarla ilişkin çalışmalardan çıkarımlar	112

5. ELE ALINAN LOJİSTİK AĞ TASARIMI PROBLEMİ İÇİN HİPERSEZGİSEL ALGORİTMA ÖNERİSİ VE UYGULAMASI.....	115
5.1 Geliştirilen Hipersezgisel Algoritmanın Öğeleri.....	115
5.2 Yöntemin Getirdiği Yenilikler	120
5.3 Kurgulanmış Lojistik Ağ Tasarımı Uygulamaları.....	120
5.3.1 İki katmanlı lojistik ağı	120
5.3.2 Üç katmanlı lojistik ağı	123
5.3.3 Dört katmanlı lojistik ağı.....	127
5.4 Geliştirilen Hipersezgisel Algoritmanın Sanal Problemler İçin Metasezgisel Yöntemlerle Karşılaştırılması.....	130
5.4.1 Lojistik Ağ Tasarımı sanal problemleri	131
5.4.2 Sanal problemler için elde edilen sonuçlar	132
5.4.3 Sanal problemler için sonuçların ve farklılıkların değerlendirilmesi...	147
6. UYGULAMA ÇALIŞMALARI	149
6.1 Bir İmalat Firmasında Uygulama	149
6.1.1 İmalat firmasının tanıtımı.....	149
6.1.2 İmalat firmasında ele alınan LAT problemi ve bulunan sonuçlar.....	150
6.1.3 İmalat firmasında uygulama sonuçlarının değerlendirilmesi	152
6.2 Bir Lojistik Firmasında Uygulama	153
6.2.1 Lojistik firmasının tanıtımı.....	153
6.2.2 Lojistik firmasında ele alınan LAT problemi ve bulunan sonuçlar	153
6.2.3 Lojistik firmasında uygulama sonuçlarının değerlendirilmesi	157
7. SONUÇLAR.....	159
KAYNAKLAR.....	163
EKLER.....	179
ÖZGEÇMİŞ.....	311

KISALTMALAR

AHP	: Analitik Hiyerarşı Prosesi
ALE	: Adaptif Liste Tabanlı İterasyon Limitli Eşik Kabulü
BF	: Büyük Fırtına (Great Deluge) Yöntemi
DM	: Dağıtım Merkezi
GA	: Genetik Algoritmalar
GK	: Geç Kabul Ölçütü
HS	: Hipersezgisel
İD	: İyileştirme Sağlayan ve Değiştirme Yapmayan Hareketler
KDS	: Karar Destek Sistemi
KKO	: Karınca Kolonisi Optimizasyon
MC	: Monte Carlo Yöntemi
Sİ	: Çözümde Salt İyileştirme Sağlayan Hareketler
TA	: Tabu Arama
TB	: Tavlama Benzetimi
TH	: Çözümde Tüm Hareketler
TY	: Turnuva Yöntemi
TT	: Tepe Tırmanma Yöntemi
TZY	: Tedârik Zinciri Yönetimi



SEMBOLLER

b_{sk}	: s. tedârikçiden, k. fabrikaya taşınan hammadde miktarı
f_{kj}	: k. fabrikadan, j. dağıtım merkezine taşınan ürün miktarı
q_{ji}	: j. dağıtım merkezinden, i. müşteriye taşınan ürün miktarı
H_s	: s. tedârikçinin hammadde üretim kapasitesi
D_k	: k. fabrikanın kapasitesi
W_j	: j. dağıtım merkezinin kapasitesi
d_i	: i. müşterinin talebi
v_j	: j. dağıtım merkezinin sabit işletme mâliyeti
g_k	: k. fabrikanın sabit işletme mâliyeti
e_k	: k. fabrikada birim üretim mâliyeti
c_{ji}	: j. dağıtım merkezinden i. müşteriye olan taşıma mâliyeti
w_{io}	: i. müşteriden o. müşteriye olan taşıma mâliyeti
a_{kj}	: k. fabrikadan j. dağıtım merkezine birim ürün taşıma mâliyeti
t_{sk}	: s. tedârikçiden k. fabrikaya birim taşıma ve satınalma mâliyeti
α	: ürünlerdeki birim hammadde kullanım oranı
u	: ürünlerdeki birim hammadde kullanım oranı
z_j	: j. dağıtım merkezi aktif ise 1, aksi durumda 0 olan değişken
p_k	: k. fabrika aktif ise 1, aksi durumda 0 olan değişken
y_{ji}	: j. dağıtım merkezi i. müşteriye ürün sağlıyorsa 1, aksi durumda 0 olan değişken
n_{ji}	: j. dağıtım merkezinden çıkan araç ilk olarak i. müşteriye ulaşıyorsa 1, aksi durumda 0 olan değişken
r_{ji}	: i. müşteriden çıkan araç j. dağıtım merkezine dönerse 1, aksi durumda 0 olan değişken
x_{ioj}	: j. dağıtım merkezinden çıkan araç i. müşteriden sonra o. müşteriye uğrarsa 1, aksi durumda 0 olan değişken
f_1	: N_j sezgiseli için çözüm kalitesindeki iyileşme performansı
$I_n(N_j)$	: N_j sezgiselinin n. iterasyondaki çözümde yaptığı değişiklik
$T_n(N_j)$	: N_j sezgiselinin n. iterasyondaki kullanımda çalışma süresi
$I_n(N_j, N_k)$	: N_j sezgiselinin N_k sezgiselinden hemen sonra çağırılması durumunda çözümdeki değişiklik
$T_n(N_j, N_k)$	: N_j sezgiselinin N_k sezgiselinden hemen sonra çağırılması durumunda çalışma süresi
$I_{kj}(t)$	: Karınca Kolonisi Algoritmasında k. karıncanın t. iterasyonda uğradığı j. sezgiselin çözümde yaptığı değişiklik
$T_{kj}(t)$	: Karınca Kolonisi Algoritmasında k. karıncanın t. iterasyonda uğradığı j. sezgiselin çalışma süresi
m	: Karınca Kolonisi Algoritmasında çözümdeki son iyileştirmeden sonraki iterasyon sayısı
$\eta_j(t - m)$	: Karınca Kolonisi Algoritmasında çözümdeki son iyileştirme anındaki sezgisel yöntemlerin feromon miktarı
γ	: Karınca Kolonisi Algoritmasında feromon yenileme oranı



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Lojistik mâliyet yüzdeleri.....	6
Çizelge 2.2 : Tersine ve İleri Lojistik farklılıkları.....	8
Çizelge 2.3 : Lojistik Ağ Tasarımı Problemi sınıfları (Pishvae ve diğ, 2009).....	45
Çizelge 2.4 : İleri Lojistik Ağ Tasarımı literatür araştırması sonuçları.....	46
Çizelge 2.5 : Tersine Lojistik Ağ Tasarımı literatür araştırması sonuçları.....	51
Çizelge 2.6 : Bütünleşik Lojistik Ağ Tasarımı literatür araştırması sonuçları.....	56
Çizelge 4.1 : Basit Hipersezgisel Algoritmalar ile ilgili çalışmalar.....	77
Çizelge 4.2 : Seçim Fonksiyonlu Hipersezgisel Algoritmalar ile ilgili çalışmalar...	80
Çizelge 4.3 : Hipersezgisel Algoritmaların diğer yöntemlerle karşılaştırılması.....	84
Çizelge 4.4 : Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar.....	87
Çizelge 4.5 : Dolaylı Genetik Algoritma çözüm sonuçları.....	91
Çizelge 4.6 : Hiper-Genetik Algoritmalar çözüm değerleri.....	94
Çizelge 4.7 : Melez Hipersezgisel Algoritma karşılaştırma sonuçları.....	96
Çizelge 4.8 : Öğrenme Tabanlı Hipersezgisel Algoritmalar ile ilgili çalışmalar. ...	104
Çizelge 4.9 : Müşteriler arasındaki mesafeler.....	110
Çizelge 4.10 : İterasyonlarda Hipersezgisel Algoritmanın bulduğu sonuçlar.....	110
Çizelge 5.1 : Uygulamada kullanılan parametreler.....	133
Çizelge 5.2 : Sanal uygulama sonucunda bulunan özet sonuçlar.....	134
Çizelge 6.1 : Gerçek uygulamada kullanılan parametreler.....	150
Çizelge 6.2 : Gerçek uygulamada üretilen özet sonuçlar.....	151
Çizelge 6.3 : Lojistik firması uygulamasında kullanılan parametreler.....	155
Çizelge 6.4 : Lojistik firması uygulamasında üretilen özet sonuçlar.....	156
Çizelge M.1 : $2 \times 2 \times 2 \times 2_1$ problemi için bulunan sonuçlar.....	250
Çizelge N.1 : $2 \times 2 \times 2 \times 2_2$ problemi için bulunan sonuçlar.....	253
Çizelge O.1 : $2 \times 2 \times 2 \times 2_3$ problemi için bulunan sonuçlar.....	256
Çizelge P.1 : $6 \times 1 \times 4 \times 6_1$ problemi için bulunan sonuçlar.....	259
Çizelge Q.1 : $6 \times 1 \times 4 \times 6_2$ problemi için bulunan sonuçlar.....	262
Çizelge R.1 : $6 \times 1 \times 4 \times 6_3$ problemi için bulunan sonuçlar.....	265
Çizelge S.1 : $10 \times 2 \times 5 \times 10_1$ problemi için bulunan sonuçlar.....	268
Çizelge T.1 : $10 \times 2 \times 5 \times 10_2$ problemi için bulunan sonuçlar.....	271
Çizelge U.1 : $10 \times 2 \times 5 \times 10_3$ problemi için bulunan sonuçlar.....	274
Çizelge V.1 : $12 \times 2 \times 8 \times 12_1$ problemi için bulunan sonuçlar.....	277
Çizelge W.1 : $12 \times 2 \times 8 \times 12_2$ problemi için bulunan sonuçlar.....	280
Çizelge X.1 : $12 \times 2 \times 8 \times 12_3$ problemi için bulunan sonuçlar.....	283
Çizelge Y.1 : $100 \times 2 \times 20 \times 300_1$ problemi için bulunan sonuçlar.....	286
Çizelge Z.1 : $100 \times 2 \times 20 \times 300_2$ problemi için bulunan sonuçlar.....	289
Çizelge AA.1 : $100 \times 2 \times 20 \times 300_3$ problemi için bulunan sonuçlar.....	292
Çizelge BB.1 : $1000 \times 60 \times 50 \times 300_1$ problemi için bulunan sonuçlar.....	295
Çizelge CC.1 : $1000 \times 60 \times 50 \times 300_2$ problemi için bulunan sonuçlar.....	298
Çizelge DD.1 : $1000 \times 60 \times 50 \times 300_3$ problemi için bulunan sonuçlar.....	301

Çizelge EE.1 : $6 \times 4 \times 4 \times 720$ problemi için bulunan sonuçlar.	304
Çizelge FF.1 : $4 \times 2 \times 2 \times 80$ problemi için bulunan sonuçlar.....	307



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Lojistik Ağ Tasarımında karar aşamaları arasındaki etkileşimler.	6
Şekil 2.2 : Bütünleşik Lojistik Ağı Tasarımı örneği.	7
Şekil 2.3 : Bütünleşik Lojistik Ağı Tasarımı süreci.	8
Şekil 2.4 : İleri Lojistik Ağı Tasarımı örneği.	9
Şekil 2.5 : Geleneksel ve Esnek Çok Aşamalı LAT Problemi (Lin ve diğ, 2009). ...	18
Şekil 2.6 : Kapalı Döngülü Tedârik Zinciri (Du ve diğ, 2009).	28
Şekil 2.7 : Genetik Algoritma akış süreci.	33
Şekil 2.8 : Tavlama Benzetimi akış şeması.	36
Şekil 2.9 : Tabu Arama algoritması akış şeması.	39
Şekil 2.10 : Karıncaların en kısa yolu bulma yöntemi.	42
Şekil 2.11 : Karınca Kolonisi Optimizasyon algoritması akış şeması.	43
Şekil 3.1 : İleri Lojistik Ağı Tasarımı.	63
Şekil 4.1 : Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritma süreç adımları.	74
Şekil 4.2 : Hiper-Genetik Algoritma süreç adımları.	75
Şekil 4.3 : Hipersezgisel Algoritmalar için oluşturulan yapılar.	102
Şekil 4.4 : Örnek problem müşteri koordinatları.	109
Şekil 4.5 : İterasyonlar süresince bulunan çözümler.	112
Şekil 5.1 : Hipersezgisel Algoritma akış şeması.	119
Şekil 5.2 : Senaryo 1 için başlangıç durum.	121
Şekil 5.3 : Fabrikalar için parametre ve değişkenler.	121
Şekil 5.4 : Müşteriler için parametre ve değişkenler.	122
Şekil 5.5 : Fabrika-müşteri kanalı için parametre ve değişkenler.	122
Şekil 5.6 : Senaryo 1 için bulunan en iyi iki çözüm.	123
Şekil 5.7 : Üç katmanlı lojistik ağ için başlangıç durum.	124
Şekil 5.8 : Üç katmanlı lojistik ağda tedârikçiler için parametre ve değişkenler. ...	124
Şekil 5.9 : Üç katmanlı lojistik ağda fabrikalar için parametre ve değişkenler.	125
Şekil 5.10 : Üç katmanlı lojistik ağda müşteriler için parametre ve değişkenler. ...	125
Şekil 5.11 : Tedârikçi-fabrika kanalı için parametre ve değişkenler.	125
Şekil 5.12 : Fabrika-müşteri kanalı için parametre ve değişkenler.	126
Şekil 5.13 : Senaryo 2 için bulunan en iyi iki çözüm.	126
Şekil 5.14 : Dört katmanlı lojistik ağ için başlangıç durumu.	127
Şekil 5.15 : Tedârikçiler için parametre ve değişkenler.	128
Şekil 5.16 : Fabrikalar için parametre ve değişkenler.	128
Şekil 5.17 : Dağıtım merkezleri için parametre ve değişkenler.	128
Şekil 5.18 : Müşteriler için parametre ve değişkenler.	129
Şekil 5.19 : Tedârikçi-fabrika kanalı için parametre ve değişkenler.	129
Şekil 5.20 : Fabrika-dağıtım merkezi kanalı için parametre ve değişkenler.	129
Şekil 5.21 : Dağıtım merkezi-müşteri kanalı için parametre ve değişkenler.	130
Şekil 5.22 : Dört katmanlı lojistik ağ için bulunan çözümler.	130
Şekil 5.23 : Lojistik Ağ Tasarımı için hazırlanan uygulama.	132

Şekil 5.24 : Programın oluşturduğu bir çözüm.	133
Şekil 5.25 : $2 \times 2 \times 2 \times 2_1$ problemi çözüm grafikleri.	138
Şekil 5.26 : $2 \times 2 \times 2 \times 2_2$ problemi çözüm grafikleri.	138
Şekil 5.27 : $2 \times 2 \times 2 \times 2_3$ problemi çözüm grafikleri.	139
Şekil 5.28 : $6 \times 1 \times 4 \times 6_1$ problemi çözüm grafikleri.	139
Şekil 5.29 : $6 \times 1 \times 4 \times 6_2$ problemi çözüm grafikleri.	140
Şekil 5.30 : $6 \times 1 \times 4 \times 6_3$ problemi çözüm grafikleri.	140
Şekil 5.31 : $10 \times 2 \times 5 \times 10_1$ problemi çözüm grafikleri.	141
Şekil 5.32 : $10 \times 2 \times 5 \times 10_2$ problemi çözüm grafikleri.	141
Şekil 5.33 : $10 \times 2 \times 5 \times 10_3$ problemi çözüm grafikleri.	142
Şekil 5.34 : $12 \times 2 \times 8 \times 12_1$ problemi çözüm grafikleri.	142
Şekil 5.35 : $12 \times 2 \times 8 \times 12_2$ problemi çözüm grafikleri.	143
Şekil 5.36 : $12 \times 2 \times 8 \times 12_3$ problemi çözüm grafikleri.	143
Şekil 5.37 : $100 \times 2 \times 20 \times 300_1$ problemi çözüm grafikleri.	144
Şekil 5.38 : $100 \times 2 \times 20 \times 300_2$ problemi çözüm grafikleri.	144
Şekil 5.39 : $100 \times 2 \times 20 \times 300_3$ problemi çözüm grafikleri.	145
Şekil 5.40 : $1000 \times 60 \times 50 \times 300_1$ problemi çözüm grafikleri.	146
Şekil 5.41 : $1000 \times 60 \times 50 \times 300_2$ problemi çözüm grafikleri.	146
Şekil 5.42 : $1000 \times 60 \times 50 \times 300_3$ problemi çözüm grafikleri.	147
Şekil 6.1 : $6 \times 4 \times 4 \times 720$ problemi çözüm grafikleri.	152
Şekil 6.2 : Lojistik firması için ele alınan ileri lojistik ağı.	154
Şekil 6.3 : $4 \times 2 \times 2 \times 80$ problemi çözüm grafikleri.	156

HİPERSEZGİSEL YÖNTEMLERLE LOJİSTİK AĞ TASARIMI VE OPTİMİZASYON

ÖZET

Tedârik Zinciri Yönetiminde en çok karşılaşılan problemlerden biri, Lojistik Ağ Tasarımı Problemidir. Bir işletmenin lojistik ağı, tedârikçiler, depolar, dağıtım merkezleri ve perakende mağazalarından ve bunlar arasındaki hammadde, ara ürün ve bitmiş ürün akışlarından oluşmaktadır.

Lojistik Ağ Tasarımı Probleminde genel olarak amaç, müşteri gereksinimleri karşılanırken aynı zamanda sabit mâliyetler ile satınalma, üretim, depolama ve taşıma gibi değişken mâliyetleri de enküçüklemeye çalışmaktır. NP-Zor problem kapsamında ele alınan Lojistik Ağ Tasarımı Problemi için sezgisel ve metasezgisel yöntemler, statik durumlarda etkin çözümler sunmaktadır. Hipersezgisel yöntemler ise probleminden bağımsız teknikler olup, çözüm arama süreci ile problem uzayı arasında bir soyutlama sağlarlar.

Bir organizasyonun lojistik ağı, bir kuruluş için tedârikçi, üretim tesisleri, dağıtım merkezleri ve depolar arasındaki her türlü taşıma ve depolama işlemleri ile ilgilidir. Lojistik Ağ Tasarımı (LAT), işletmenin depo, dağıtım merkezi gibi noktalarının tedârikçi, üretim noktaları ve bayi, müşteri, toptancı gibi satış noktaları ile ilişkilendirilmesidir. Bu tez çalışmasında LAT problemleri İleri, Tersine ve Bütünleşik olmak üzere üç başlık altında incelenmiştir. İleri Lojistik Ağ Tasarımında hammadde ve ürünler tedârikçi, fabrika, dağıtım merkezi ve müşteri yoluyla ilerlemektedir. Tersine Lojistik Ağ Tasarımında ise ürün bu yönün tersi bir akışa sahip olup, Bütünleşik LAT’de her iki yöndeki hareket ele alınmaktadır.

Lojistik Ağ Tasarımı Problemleri dört boyutta değerlendirilebilir. Karar Düzeyi olarak stratejik, taktik ve operasyonel, İşlevler olarak lokasyon belirleme, araç rotalama, depo miktarı belirleme, araç sayısı hesaplama vb., Çözüm Yöntemleri olarak sürekli yakınlaştırma, kesin ve sezgisel yöntemler, Uygulama Alanlarına (Sektörler) göre ise otomotiv, lojistik, elektronik vb. olarak sınıflandırılabilir.

Bu tez çalışmasında Lojistik Ağ Tasarımı Problemi ve Hipersezgisel Algoritmalar ayrı başlıklar hâlinde anlatılmış ve literatür araştırmalarına ilişkin ayrıntılı sonuçlar açıklanmıştır. Çalışmada İleri Lojistik Ağ Tasarımı Problemi ile ilgili bir Karma Tamsayılı Programlama Modeli oluşturulmuştur. Lojistik Ağ Tasarımı Probleminde genel olarak amaç, müşteri gereksinimlerinin karşılanması koşuluyla tesislerin sabit mâliyetleri ve satınalma, üretim, depolama ve taşıma gibi değişken mâliyetlerin enküçüklemeye çalışılmasıdır. Problem oldukça geniş ve karmaşık olduğu için literatürde genel olarak belirli alt süreçlere ait problemler üzerinde çalışmalar yapılmıştır. Kurulan Karma Tamsayılı Programlama Modelinde temel olarak tedârikçi seçimi, tesis ve dağıtım merkezi aktifleştirilmesi, hangi fabrikanın, hangi tedârikçiden hammadde alacağı ve hangi dağıtım merkezine ürün göndereceği,

müşterilere hangi dağıtım merkezi üzerinden hizmet edileceği ve dağıtım merkezi ile müşteriler arasındaki araç rotaları belirlenmeye çalışılmaktadır.

Günümüzde, özellikle ekonomik nedenlerden dolayı, gerçek yaşamda karşılaşılan ve optimizasyon gerektiren problemler oldukça önem kazanmıştır. Bu tür problemler için kesin modellemeler yapmak karmaşık ve ayrıntılı olduğundan ve aynı zamanda çözümü uzun süreler gerektirdiği için bunlar uygulanabilir değildir. Bu nedenlerle, optimum sonucu garanti etmeyen ancak uygun süreler içerisinde yürütülmesi kolay, fazla kaynak gerektirmeyen ve uygulanabilir çözümler üreten basit sezgisel yöntemlere başvurulmaktadır.

Metasezgisel Yöntemler son yıllarda başarısını kanıtlayarak, yürütülmesi oldukça zor, maliyetli ve problem özelinde şekillendirilen tekniklerdir. Bir problem için yüksek kaliteli çözümler üreten bir Metasezgisel Algoritma, problemde birtakım değişiklikler yapıldığında başarısız olabilmektedir. Bu yüzden, gerçek yaşamda kullanılacak Metasezgisel yöntemlerin, bu değişikliğe uyum sağlamak üzere elden geçirilmesi gerekmektedir. Hipersezgisel Algoritmalar ise genelleştirilmiş, dayanıklı ve probleminden bağımsız, yeniden kullanılabilen yöntemlerdir, Metasezgisel Yöntemlerden daha üst düzeyde soyutlama sağlarlar ve algoritma sürecinde herhangi bir zamanda hangi klasik sezgisel yöntemin uygulanacağına karar veren bir çözüm yaklaşımına sahiptirler. Klasik sezgisel teknikler, optimum, çözümü zor ve uzun süreler gerektiren büyük boyutlu problemler için, kabul edilebilir süre içinde optimuma yakın çözümler üreten, kullanımı kolay ve probleme özgü yöntemlerdir. Hipersezgisel Algoritmalar ise bu basit sezgisel yöntemleri uyumlu bir şekilde organize ederek etkin sonuçlar üretmektedir. Basit bir sezgisel yöntem, her problem için iyi çözümü üretmese de, diğer basit sezgisel tekniklerle uygulandığında kaliteli sonuçlar çıkarabilmekte, diğer bir deyişle çözümün belirli aşamalarında performansı yüksek olabilmektedir. Bu nedenle, sezgisel yöntemleri tek tek kullanmak yerine bütünleşik melez bir yapı oluşturmak, çözüm kalitesini arttırmaktadır.

Hipersezgisel Algoritmalarda basit sezgisel tekniklerin nasıl seçileceğine kılavuzluk eden bir öğrenme mekanizması vardır. Hipersezgisel yöntemler süreç açısından Basit Hipersezgiseller (Sezgisel yöntemleri rassal seçen yöntemler), Seçim Fonksiyonlu Hipersezgiseller (Sezgisel yöntemlere ağırlık verip, bu ağırlığa göre seçim yapan yöntemler), Metasezgisel Yöntemlerle Melezleştirilen Hipersezgiseller (Sezgisel yöntemlerin uygulanma sırasını kodlayıp Metasezgisel kullanarak bu sırayı güncelleyen yöntemler), Öğrenme Tabanlı Hipersezgiseller (Basit sezgisel yöntemlerin daha önceki performanslarını dikkate alarak, hangi yöntemin seçileceğine karar veren uyarlanabilir (adaptif) öğrenme mekanizmasına sahip yöntemler) olmak üzere dört ayrılabilir.

Bu tez çalışmasında Lojistik Ağ Tasarımı problemi için bir Hipersezgisel Algoritma geliştirilmiş ve Microsoft .NET C# programı üzerinde kodlaması yapılmıştır. Ayrıca tasarlanan Hipersezgisel Algoritmanın farklı boyutlardaki Lojistik Ağ Tasarımı Problemleri için literatürde yer alan Genetik Algoritma ve Tavlama Benzetimi yöntemleri ile de karşılaştırılması yapılmıştır. Literatürde Lojistik Ağ Tasarımı için varolan bir problem kümesi (seti) olmadığı için, bu tezdeki problemler sanal olarak hazırlanmıştır ve problem içindeki parametrelere rassal değerler verilmiştir. Hazırlanan problemlerin düşük boyuttaki olanları için GAMS ortamında Karma Tamsayı Programlama modeli oluşturularak optimum sonuçlar bulunmuştur, büyük boyutlu problemler için ise algoritmalar süresince bulunan en iyi çözüm dikkate alınmıştır. Geliştirilen Hipersezgisel Algoritmanın karşılaştırılması, optimum sonuca

veya bulunan en iyi sonuca yakınlık, çözüm süresi ve bulunan çözümlerin dağılımları üzerinden yapılmıştır.

Bu çalışmada Hipersezgisel Yöntemlerin daha iyi açıklanabilmesi için ayrıca Gezgin Satıcı Problemi üzerinde iyileştirici sezgiselleri kullanan ve salt iyileştirme gösteren hareketleri kabul eden (Sİ) Basit Hipersezgisel bir algoritma örneği verilmiştir. Bunun yanında geliştirilen Hipersezgisel yöntem kullanılarak bir, iki ve üç rotalı (güzergâhlı) LAT kapsamında basit örnekler üzerinden denemeler yapılmıştır.

Tez kapsamında geliştirilen Hipersezgisel Algoritmanın etkinliğini ölçmek için altı sanal problem tipi oluşturulmuş olup, her tip için üç örnek problem hazırlanmıştır. Bu problemler üzerinden ise Altıparmak ve diğerleri tarafından 2006'da geliştirilen Genetik Algoritma ile Jamayaran ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi yöntemi arasında kıyaslama yapılmıştır. İlk dört problem, Karma Tamsayılı Programlama Modeli kurularak GAMS ortamında çözülmüş ve optimum sonuçlar elde edilmiştir. Diğer iki problem için, büyük olmalarından dolayı GAMS ortamında çözüm oluşturulamamış, algoritmalarca bulunan en iyi çözüm değeri ile karşılaştırmalar yapılmıştır.

Literatürde Lojistik Ağ Tasarımı için bir problem kümesi olmadığı için problem verileri tümüyle rassal olarak üretilmiştir, bu sırada kapasite ve talep kısıtları dikkate alınarak uygun çözümlerin olabilirliği gözönüne alınmıştır. Üretilen problemler için bu çalışmada kurulan Karma Tamsayılı Programlama modeli dikkate alınmış ve senaryolar bu model üzerinden gerçekleştirilmiştir. Bulunan çözüm değerlerinin ayrıca sağlaması yapılarak sürecin kontrolü gerçekleştirilmiştir. Oluşturulan GAMS modelleri, program üzerindeki DICOPT ve BARON modülü ile çözülmüş ve optimum sonuçlar elde edilmiştir.

Bu çalışmada yöntemler, her bir problemde her bir parametre kümesi için yüz kez uygulanmış olup, bulunan çözüm değerleri, en iyi çözümle arasındaki fark, buna ait sapma yüzdesi ve çözüm süreleri hesaplanmıştır. Bu noktada yer alan çözüm sürelerinden kasıt, algoritmanın çalışma süreci boyunca bulduğu en iyi çözüme ulaşma süresidir. Uygulamada Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince 100 denemede bulunan çözüm değerlerinin ve en iyi çözümün üretilme süresinin kutu grafikleri ile analizi yapılmıştır.

Tez kapsamında gerçek yaşamdan örnek olması açısından imalat sektöründe Türkiye'nin birçok yerinde tesisleri bulunan ve sektörün lider firmalarından biri seçilerek bir LAT problemi ele alınmıştır. Türkiye'de altı adet kutu fabrikası, dört adet depo, dört dağıtım merkezi ve 700'ü aşkın müşteriye sahip firmada gerçeğe uygun veriler alınarak gerçek yaşamdan bir uygulama gerçekleştirilmiştir. Tez çalışmasında ayrıca uluslararası hızlı taşımacılık alanında faaliyet gösteren bir lojistik firmasında da gerçeğe yakın veriler alınarak bir uygulama yapılmıştır. İngiltere, İtalya, Güney Kore ve Malezya'dan Türkiye'ye gönderilen 270 gönderi ve gönderim noktası olan 80 parsel üzerinde dört katmanlı LAT problemi için çözüm oluşturulmuştur.

Bu tez çalışmasının amacı; gerçek yaşamda karşılaşılan Lojistik Ağ Tasarımı Problemi için literatürde eksikliği giderecek sağlam (robust), esnek ve kaliteli çözümleri kabul edilebilir sürede üreten bir Hipersezgisel Algoritma hazırlamak ve gerçek bir süreç üzerinde uygulamasını yapmaktır. Geliştirilmesi yapılan Hipersezgisel Algoritma, uygun süreler içinde farklı problemlerde en iyi çözümü elde etmiş, sonuçların ayrıntılı analizi yapıldığında diğer algoritmalarla göre daha iyi performans göstermiştir.



LOGISTIC NETWORK DESIGN AND OPTIMIZATION USING HYPERHEURISTIC METHODS

SUMMARY

One of the most common problems in the Supply Chain Management is Logistics Network Design Problem. A company's logistics network consists of suppliers, warehouses, distribution centers and retail stores, as well as raw materials, intermediate products and finished product flows among them.

In the Logistics Network Design Problem, the objective is generally to minimize the fixed costs and the variable costs such as purchasing, production, storage and transportation while meeting the customer requirements. For the Logistic Network Design defined as the NP-hard problem, Heuristics and Metaheuristics methods provide effective solutions in static situations. However, Hyperheuristic methods are independent of the problem and provide an abstraction between the solution search process and the problem solution space.

The logistics network of an organization is concerned with all kinds of transportation and warehousing operations between the supplier, the production facility, the distribution centers and the warehouse for a company. Logistics Network Design is to associate warehouse, distribution center, etc. with the supplier, production points and sales points such as dealers, customers and wholesalers. In this thesis study, Logistics Network Design problems are described under three headings as Forward, Reverse and Integrated. In Forward Logistics Network Design, raw materials and products are progressing through supplier, factory, distribution center and customer respectively. In Reverse Logistic Network Design, the product has a contrary direction in this flow and the movement in both directions is taken into consideration in the Integrated Logistic Network Design.

Logistics Network Design Problems can be evaluated in four perspectives: As Decision dimension; Strategic, Tactic and Operational, as Functional dimension; determination of locations, vehicle routes, inventories, vehicle quantities etc., as Solution Methods dimension; exact and heuristic techniques, as Scope of Implementation; automotive, logistics, electronics sectors etc. are classifications.

In this thesis study, Logistic Network Design Problem and Hyperheuristic Algorithms are described in separate sections as a result of detailed literature research. For Forward Logistic Network Design Problem a Mixed Integer Linear Programming Model was designed and modelled in GAMS software. In Logistics Network Design Problem, the aim is generally to try to minimize the fixed costs of the facilities and to minimize the variable costs such as purchasing, production, storage and transportation provided that customer requirements are met. Since the problem is so complex, the literature studies are generally focused on the problems of certain subprocesses. In the Mixed Integer Linear Programming Model proposed, focus is given to the selection of suppliers, the activation of the factory and the

distribution center, the assignments between supplier-factory and factory-distribution centers and the vehicle routes between the distribution centers and the customers.

Nowadays, especially for economic reasons, the problems encountered in real life require optimization and have become very important. It is not feasible to make exact models for such problems because of its complexity and long periods of time for reaching solutions. Thus, simple heuristic methods that do not guarantee optimum results but that are easy to execute within reasonable times to produce feasible solutions are being used.

Although Metaheuristic methods have proven their success in recent years, they are difficult to implement, have high costs and problem-specific structure. A Metaheuristic Algorithm that produces a high-quality solution which can fail due to little changes in the state of the problem. Hyperheuristic Algorithms are generalized, robust and problem independent reusable methods, that provide a higher level of abstraction from Metaheuristics and have a solution approach that decides which classical heuristic method to apply at any time in the algorithm process. Classic heuristics are easy-to-use and problem-specific methods that produce solutions that are close to optimal in acceptable time for large-scale problems which are difficult to optimize and require long time to elapse. Hyperheuristic Algorithms organize these simple heuristic methods in a consistent way and create effective results. While a simple heuristic does not produce a good solution for every problem, it can produce quality results when applied together with other simple heuristic techniques, in other words the performance of the solution can be high at certain stages. Hence, rather than using individual heuristic methods, creating an integrated hybrid structure using a Hyperheuristic approach increases the quality of the solution.

Hyperheuristic Algorithms have a learning mechanism that guides how to choose simple heuristic techniques. Hyperheuristic methods can be classified as: Simple Hyperheuristics (random selection of heuristic methods), Hyperheuristics with choice functions (selection of heuristics according to heuristic weights), Hybrid Hyperheuristics with Metaheuristic methods (encoding the sequence of heuristic methods and selection of heuristics by this sequence using Metaheuristics), Learning based Hyperheuristics (adaptive learning mechanisms that decide which method to choose by considering their previous performance).

In this thesis, the proposed Hyperheuristic Algorithm for Logistic Network Design problem was developed on Microsoft .NET C # programming language. Moreover the suggested Hyperheuristic Algorithm was compared with the use of Genetic Algorithm and Simulated Annealing algorithms. There is no Logistic Network Design example of integrated functions is found in literature, that is why the virtually prepared problems and random parameters are used for comparisons. For the low dimensional problems, optimum solutions were produced by Mixed Integer Linear Programming model created in GAMS software and for the large problems the best solution found during the algorithms was taken into consideration. The comparison for the algorithms is based on the closeness to the best solution found, the solution time and the distributions of the solutions produced.

In order to clarify the procedure, a Simple Hyperheuristics was applied on Travelling Salesman Problem which accepts only better movements in solution. Furthermore, the proposed Hyperheuristic method was applied to simple Logistic Network Design Problems that have one, two and three stages.

The virtual Logistic Network Design Problem data are produced randomly considering feasible solutions by checking capacity and demand constraints. Since there is no problem sets for Logistic Network Design in the literature, the Mixed Integer Linear Programming model designed in this study was taken into account and the scenarios were realized through this model. Algorithm processes were checked by controlling solutions values separately. The generated GAMS models were solved with the DICOPT and BARON modules on the program and optimum results were obtained.

In this study, the methods were applied hundred times for each parameter set of problems and the solution values, the differences among the best solutions, the percentage of deviation and the solution time were recorded. The duration of solution is the execution time of finding the best solutions during the running process. Analysis of Hyperheuristic Algorithm, Genetic Algorithm and Simulated Annealing approaches are carried out with box graphs of 100 experiment values and the execution times.

As a case study of the proposed algorithm, applications of one of the leading packaging companies is chosen, as it has distributed facilities in many regions. The company has six factories, four warehouses, four distribution centers and more than 700 customers which is implemented as a real life case. The second case is taken from a global logistics firm operating in fast transportation sector. A solution for the four layered Logistic Network Design problem was created for 80 parcels, 270 shipments sent from the UK, Italy, South Korea and Malaysia to Turkey. Both of these real life cases showed better performances with the proposed Hyperheuristic Algorithm.

The aim of this study is to fill the gap in literature for a robust and flexible method that produces good quality solutions in acceptable times for the Logistic Network Design Problem using this Hyperheuristic approach. In the thesis, proposed Hyperheuristic Algorithm has obtained the best solutions for different problems within best solution durations and has performed better than Genetic and Simulated Annealing algorithms. This study will be a lead for finding fast solutions for a complex network company.



1. GİRİŞ

Günümüzün küreselleşen dünyasında tedârik zincirleri arasında yaşanan yoğun rekabet, firmaları, her türlü mâliyeti azaltarak müşterilerinin gereksinimlerini karşılamaya yönlendirmektedir. Dünyanın dört bir yanında üretilen ürün ve hizmetlerin pazarlara kolayca ulaşabilmesi, yeni akımlar ve yaşanan ekonomik krizler, kısa ve uzun vâdede tüketimde dalgalanmalara yol açmakta, şirket yöneticilerinin ileriye yönelik kararlarını olumsuz etkileyebilmektedir. Bu noktada tedârik zincirini iyi bir şekilde plânlayan ve kontrol altında tutabilen firmalar, değişken müşteri taleplerine hızlı bir şekilde uyum sağlayabilir. Böylece firmalar hem mâliyet avantajına sahip olacak, hem de müşteri hoşnutluğunu sürekli kılacaktır.

Tedârik Zinciri Yönetiminde kapsam bakımından en çok çalışılan problemlerden biri, Lojistik Ağ Tasarımı Problemidir. Lojistik Ağ Tasarımı, satınalma, üretim, depolama ve taşıma mâliyetlerini enküçüklemek için üretim tesislerinin sayısı, lokasyon ve kapasitelerinin belirlenmesi, tedârikçilerin seçimi, ürünlerin üretim tesislerine ve depolara atanması, dağıtım kanallarının ve taşıma türlerinin belirlenmesi, hammadde, yarı ürün ve ürün akışlarının saptanması gibi konuları içermektedir (Cordeau ve diğ, 2006). Bir işletmenin lojistik ağı, tedârikçiler, depolar, dağıtım merkezleri ve perakende mağazalarından ve bunlar arasındaki hammadde, ara ürün ve bitmiş ürün akışlarından oluşmaktadır. Ağ tasarımı yapılırken, tesis, depo ve perakende mağazalarına ilişkin kararlar alınır. Bu alanda, depo yeri, depo sayısı, depo kapasitesi ve aynı zamanda hangi depodan hangi müşterilere sevkiyat yapılacağı gibi konulara karar verilmeye çalışılır (Pishvae ve diğ, 2009).

1.1 Tezin Kapsamı ve Amacı

Bu tez çalışmasında ele alınan İleri Lojistik Ağ Tasarımı problemi dört katmanlı, tek ürünlü, tek dönemli, deterministik talepli ve kapasite kısıtlı olup ayrıca problemde dağıtım merkezi ve müşteriler arasında araç rotalama yapılmaktadır. Literatürde günümüzün dinamik koşullarında karşılaşılan ve bu tez kapsamında irdelenen Lojistik Ağ Tasarımı Problemi için etkin çözümler üretebilecek ve değişikliklere hızlı bir şekilde uyum sağlayabilecek esnek bir yöntemin, bilgilerimiz ışığında henüz geliştirilmediği ve uygulamasının yapılmadığı anlaşılmaktadır. Literatürde bu boşluğun doldurulması için son yıllarda sıkça araştırma

konusu olan ve özellikle esneklik konusunda ön plâna çıkan Hipersezgisel Algoritmalar, bu tez çalışmasının ana çıkış noktasını oluşturmaktadır.

Lojistikle ilgili kararlarda müşteri talepleri karşılanmaya çalışılırken aynı zamanda satınalma, üretim, depolama ve taşıma ile ilgili sabit ve değişken mâliyetler enküçüklenmeye çalışılır. NP-Zor problem kapsamında ele alınan Lojistik Ağ Tasarımı Problemi için sezgisel ve metasezgisel yöntemler, statik durumlarda etkin çözümler sunmaktadır. Bu çözümler, günümüzün gerektirdiği dinamik koşullar altında, uygulanabilir sürede çözümler üretememekte, çevresel değişikliklerden çok büyük oranda etkilenmekte ve dayanıklılık göstermemektedir (Chakhlevitch ve Cowling, 2008). Bu noktada Hipersezgisel Algoritmalar bu duruma çözüm getirmektedir. Hipersezgisel yöntemler, problemde bağımsız teknikler olup, çözüm arama süreci ile problem uzayı arasında bir soyutlama sağlarlar. Problemin topolojisi üzerinde herhangi bir değişiklik olması veya problemde farklı koşulların denenmek istenmesi durumunda algoritma süreci hızlı bir şekilde uyum sağlar (Cowling ve diğ, 2002). Bu tez çalışmasında İleri Lojistik Ağ Tasarımı Problemi için esneklik sağlayan ve kaliteli çözümleri kabul edilebilir sürede üreten bir Hipersezgisel Algoritma geliştirilerek bu eksikliğin giderilmesi hedeflenmiştir.

Lojistik Ağ Tasarımı problemleri, karar düzeyinde (stratejik, taktik ve operasyonel), işlevler açısından (lokasyon belirleme, araç rotalama, depo miktarı, vb.), çözüm yöntemleri (kesin, sezgisel yöntemler) ve uygulama alanları açısından farklı boyutlarda ele alınabilmektedir. Bu tez çalışmasının amacı, gerçek yaşamda karşılaşılan Lojistik Ağ Tasarımı Problemi için bu boyutlarda sağlam (robust), esnek ve kaliteli çözümleri kabul edilebilir bir sürede üreten bir Hipersezgisel Algoritma hazırlamak ve gerçek bir süreç üzerinde uygulamasını yapmaktır.

Hazırlanan bu tez çalışmasının ilerleyen bölümlerinde Lojistik Ağ Tasarımı Problemi Bölüm 2’de ve Hipersezgisel Algoritmalar Bölüm 4’te olmak üzere ayrı başlıklar hâlinde anlatılacak ve literatür araştırmalarına ilişkin ayrıntılı sonuçlar açıklanacaktır. Çalışmada Bölüm 3’te Lojistik Ağ Tasarımı Problemi ile ilgili bir Karma Tamsayılı Programlama Modeli oluşturulmuştur. Bölüm 5’te ise Lojistik Ağ Tasarımı için oluşturulan Hipersezgisel Algoritmanın adımları ve geliştirilen Microsoft .NET C# programının özellikleri açıklanacaktır. Ayrıca bu bölümde farklı boyutlardaki sanal LAT problemleri için tasarlanan Hipersezgisel Algoritmanın literatürde yer alan Genetik Algoritma ve Tavlama Benzetimi yöntemleri ile karşılaştırılması yapılacaktır. Hazırlanan problemlerin düşük boyutta olanları için GAMS ortamında Karma Tamsayılı Programlama modeli oluşturularak optimum sonuçlar bulunmuştur, büyük boyutlu problemler için ise algoritmaların uygulanması

süresince bulunan en iyi çözüm dikkate alınmıştır. Tasarlanan Hipersezgisel Algoritmanın karşılaştırılması, optimum sonuca veya bulunan en iyi sonuca yakınlık, çözüm süresi ve bulunan çözümlerin dağılımları üzerinden yapılacaktır. Bu tez çalışmasında ayrıca Bölüm 6'da gerçek yaşamdan alınan verilerle İmalat ve Hızlı Taşımacılık sektöründe iki uygulama yapılarak bulunan çözümler değerlendirilmiştir. Bölüm 7'de ise tez kapsamında bulunan sonuçlar irdelenerek bu çalışma konusunun literatüre getirdiği yenilikler, öneriler ve daha önce yapılan çalışmalardan hangi yönlerden ayrıldığı belirtilmiştir.





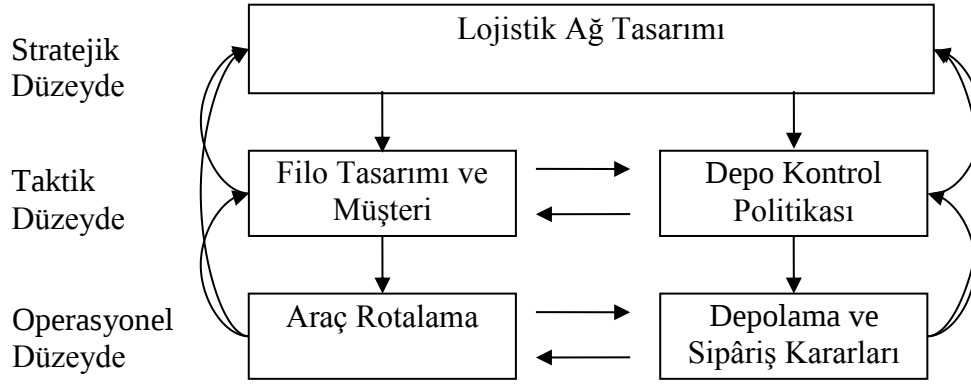
2. LOJİSTİK AĞ TASARIMI PROBLEMİ

Bir lojistik ağı, bir kuruluş için tedârikçi, üretim tesisleri, dağıtım merkezleri ve depolar arasındaki her türlü taşıma ve depolama işlemleri ile ilgilidir. Lojistik Ağ Tasarımı Problemi, genellikle bir üretim tesisi kurulurken karşılaşılan ve uygulanan çözümün orta vâdede değiştirilmesi oldukça zor olduğu bir problemidir. Bir lojistik ağ tasarlanırken aşağıdaki konuların gözönüne alınması gereklidir (Cordeau ve diğ, 2006):

- Üretim tesisleri, dağıtım merkezi ve depo sayılarının, yerlerinin ve kapasitelerinin belirlenmesi,
- Tedârikçilerin seçimi,
- Hangi ürünlerin hangi üretim tesislerinde üretileceği, hangi depolarda depolanacağı ve hangi dağıtım merkezleri tarafından dağıtılacağını belirlenmesi,
- Dağıtım kanallarının ve taşıma türlerinin seçimi,
- Hammadde, yarı ürün ve ürünlerin bu ağ içindeki akışlarının belirlenmesi.

Bu konular, önem dereceleri ve plânlama süreçlerinin uzunluklarına göre üç gruba ayrılabilirler. İlk grup Stratejik Karar olarak adlandırılan ve uzun yılları içeren, tesislerin yer, lokasyon ve kapasitelerinin belirlenmesi konusudur. Taktik grupta tedârikçilerin seçimi ve ürünlerin üretim, dağıtım ve depo yerlerinin belirlenmesi konusu yer almaktadır. Son grup olarak operasyonel aşamada ise hammadde, yarı ürün ve ürünlerin hareketleri kararlaştırılır.

Stratejik düzeyde Lojistik Ağ Tasarımı, taktik ve operasyonel aşamaları doğrudan ya da dolaylı olarak etkilemektedir. Aynı zamanda aynı düzeyde verilen kararlar birbirleriyle etkileşim hâlinindedir. Şekil 2.1’de bu etkileşimler gösterilmektedir (Miranda ve diğ, 2009).



Şekil 2.1 : Lojistik Ağ Tasarımında karar aşamaları arasındaki etkileşimler.

Lojistik Ağ Tasarımında ana konular aşağıda açıklanmaktadır (Lu ve diğ, 2008):

- Hiyerarşi yapısını belirlemek.
- Dağılım ve yoğunlaşma derecesini saptamak.
- Tesislerin yeri, sayısı ve kapasitesini belirlemek.

Bir Lojistik Ağ Tasarımı Probleminde en az iki kademe olması gerekmektedir, aksi durumda teker teker problem bir operasyonel süreç olarak görülüp, rotalama, taşıma, depolama problemi gibi ele alınabilir. Tek katmanlı problemler bir akış problemidir ve bu problemlerde ağ (şebeke) yapısı bulunmamaktadır. Ağ kavramı en az iki kademeli olan bu durumu zorunlu kılmaktadır.

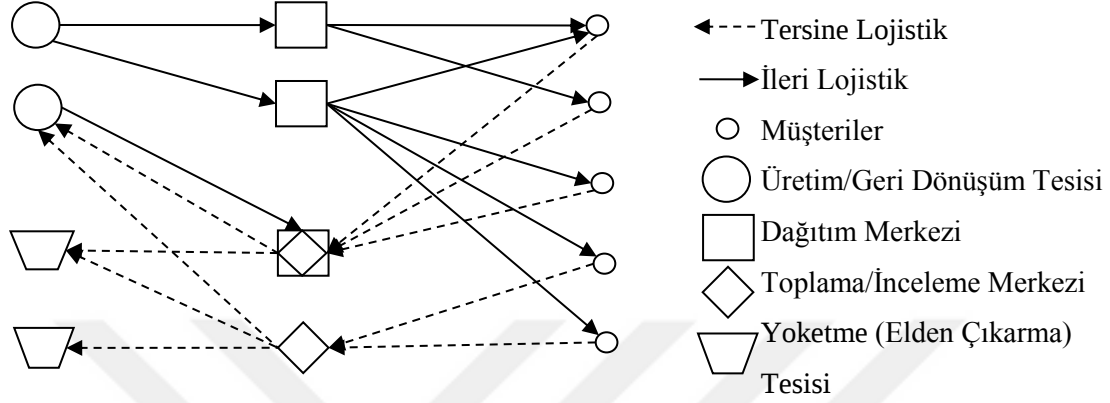
Lojistik mâliyetleri ile ilgili 2001 yılında Japonya’da yapılan White Paper on Land, Infrastructure and Transport araştırmasından elde edilen sonuçlar Çizelge 2.1’de gösterilmektedir (Lin ve diğ, 2009).

Çizelge 2.1 : Lojistik mâliyet yüzdeleri.

Lojistik Mâliyetleri	Yüzde (%)
Taşıma (Transportasyon)	52.66
Depolama	19.67
Yükleme/Boşaltma	12.33
Yönetim	8.59
Paketleme	6.75

Lojistik Ağ Tasarımı, İleri ve Tersine Lojistik’te stratejik, taktik ve operasyonel karar aşamalarını içerir. Bu karar aşamalarında yer alan İleri ve Tersine Lojistik problemlerinin birleştirilmesi ile yapılan tasarımlara, Bütünleşik Lojistik Ağ Tasarımı denir. Bu tanıma ait bir

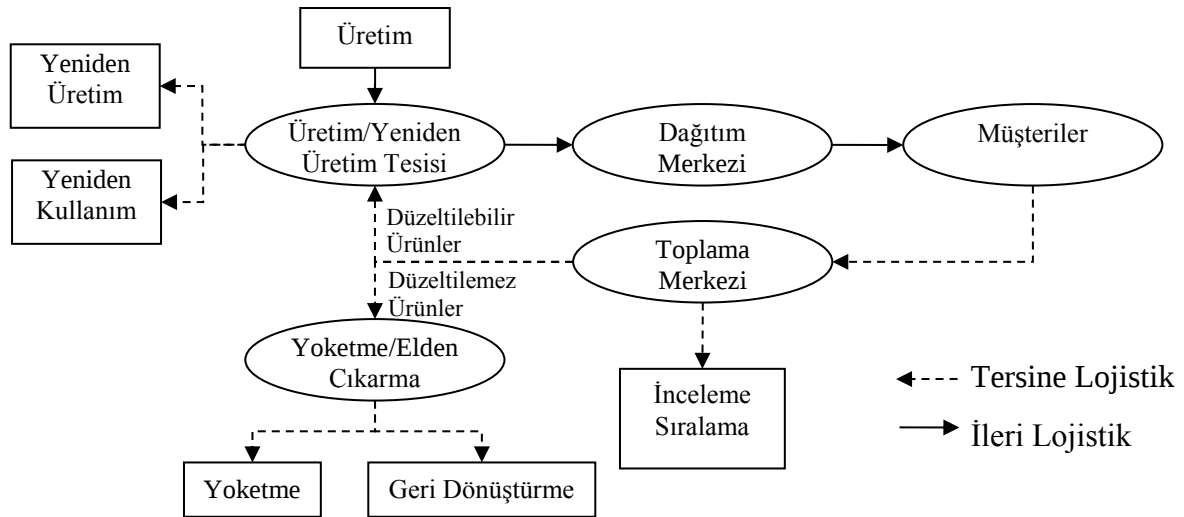
örnek Şekil 2.2’de ve süreç Şekil 2.3’te gösterilmektedir. Eğer birleştirilecek problemler aynı karar aşamasında ise yatay bütünleşme, farklı aşamalarda ise dikey bütünleşme olarak adlandırılır. Örneğin, tedârikçilerin seçilmesi ve tesis yerlerinin belirlenmesi yatay bütünleşme, taşıma türlerinin seçilmesi ve rotaların belirlenmesi dikey bütünleşme olarak adlandırılır (Pishvae ve diğ, 2009).



Şekil 2.2 : Bütünleşik Lojistik Ağ Tasarımı örneği.

Bir Lojistik Ağ Tasarımı problemi, aşağıdaki alt problemlerin birleştirilmesi ile oluşmaktadır. Bu problemler stratejik düzeyde ele alınabileceği gibi, taktik ve operasyonel düzeyde de işlenebilir.

- Yer Belirleme Problemleri
- Araç Rotalama Problemleri
- Tesis Atama Problemleri
- Yer – Atama (Location – Allocation) Problemleri
- Yer – Rotalama (Location – Routing) Problemleri



Şekil 2.3 : Bütünleşik Lojistik Ağı Tasarımı süreci.

İleri ve Tersine Lojistik Ağ Tasarımları farklı politika ve bakış açıları ile yapılmalıdır. Bunun nedeni, Çizelge 2.2’de belirtilen farklılıklardır (Min ve Ko, 2008):

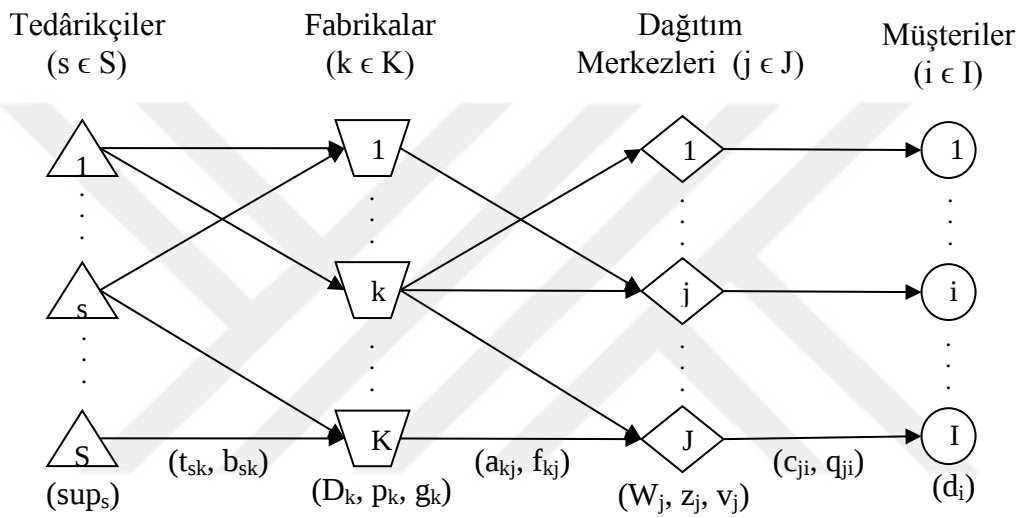
Çizelge 2.2 : Tersine ve İleri Lojistik farklılıkları.

	Tersine Lojistik	İleri Lojistik
Miktar	Küçük miktarlarda taşıma Standartlaşma yok	Büyük miktarlarda Standartlaşma var
Bilgi Takibi	Otomasyon ve Elle	Otomasyon (Bilgi Teknolojileri)
Sipariş Çevrim Süresi	Orta veya Uzun	Kısa
Ürün Değeri	Orta veya Düşük	Yüksek
Envanter Kontrol	Yapılmıyor	Odaklı
Öncelik	Düşük	Yüksek
Mâliyet Kalemleri	Öngörülemez	Belirgin
Ürün Akışı	İki yönlü (Çekme ve İtme)	Tek yönlü (İtme)
Lojistik Kanal	Karmaşık (Çok katmanlı)	Az Karmaşık (Tek veya Çok katmanlı)

Lojistik Ağ Tasarımı Probleminde genel olarak amaç, müşteri gereksinimleri karşılanırken aynı zamanda sabit mâliyetler ve satınalma, üretim, depolama ve taşıma gibi değişken mâliyetleri de enküçüklemeye çalışmaktır. Problem oldukça geniş ve karmaşık olduğu için literatürde genel olarak belirli alt süreçlere ait problemler üzerinde çalışmalar yapılmıştır.

2.1 İleri Lojistik Ağ Tasarımına İlişkin Çalışmalar

Literatürde Lojistik Ağ Tasarımı Problemi ilk olarak tesis yerlerinin belirlenmesi sürecinde Alfred Weber tarafından 1909 yılında Lokasyon Kuramında ortaya atılmıştır (Fong, 2005). Literatürde konuyla ilgili yapılmış çok sayıda çalışmaya rastlanmış olup, bu bölümde 1998'den sonra yapılan çalışmalar özetlenmiştir. Lojistik Ağı Tasarımı modellenirken genellikle Karma Tamsayılı Programlama yapısı kullanılmaktadır. Şekil 2.4'te modelleme için kullanılacak tek ünlü ve tek dönemli bir İleri Lojistik Ağ Tasarımı örneği gösterilmektedir (Altıparmak ve diğ., 2006).



Şekil 2.4 : İleri Lojistik Ağ Tasarımı örneği.

Karma Tamsayılı Programlama modeli için aşağıdaki varsayımlar yapılmıştır:

- Tedârikçilerin, müşterilerin sayıları ile müşteri talep ve tesis kapasite miktarları bilinmektedir.
- Potansiyel fabrika ve dağıtım merkezi yerleri önceden belirlenmiştir.
- Müşteriler ürünü tek bir dağıtım merkezinden temin edeceklerdir ve müşteri talepleri tam olarak karşılanacaktır.

Modelde müşteriler için $i \in I$, dağıtım merkezleri için $j \in J$, fabrikalar için $k \in K$ ve tedârikçiler için $s \in S$ indisleri kullanılmıştır. Modelde kullanılan sürekli değişkenler ve karar değişkenleri aşağıda belirtilmektedir:

$$z_j = \begin{cases} 1, & j. \text{dağıtım merkezi açıksa} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$p_k = \begin{cases} 1, & k. \text{ fabrika açıksa} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$y_{ji} = \begin{cases} 1, & j. \text{ dağıtım merkezi, } i. \text{ müşteriye hizmet sağlıyorsa} \\ 0, & \text{Aksi durumda} \end{cases}$$

b_{sk} : s. tedârikçiden, k. fabrikaya taşınan hammadde miktarı ($s \in S, k \in K$)

f_{kj} : k. fabrikadan, j. dağıtım merkezine taşınan ürün miktarı ($k \in K, j \in J$)

q_{ji} : j. dağıtım merkezinden, i. müşteriye taşınan ürün miktarı ($j \in J, i \in I$)

Karma Tamsayılı Programlama Modelinde kullanılacak parametreler ise şunlardır:

D_k : k. fabrikanın kapasitesi ($k \in K$)

W_j : j. dağıtım merkezinin kapasitesi ($j \in J$)

sup_s : s. tedârikçinin hammadde üretim kapasitesi ($s \in S$)

d_i : i. müşterinin talebi ($i \in I$)

v_j : j. dağıtım merkezinin sabit işletme mâliyeti ($j \in J$)

g_k : k. fabrikanın sabit işletme mâliyeti ($k \in K$)

c_{ji} : j. dağıtım merkezinden i. müşteriye birim ürün taşıma mâliyeti ($j \in J, i \in I$)

a_{kj} : k. fabrikadan j. dağıtım merkezine birim ürün taşıma mâliyeti ($k \in K, j \in J$)

t_{sk} : s. tedârikçiden k. fabrikaya birim taşıma ve satınalma mâliyeti ($s \in S, k \in K$)

α : ürünlerdeki birim hammadde kullanım oranı

Modeldeki amaç fonksiyonu; tesis açma, taşıma ve satınalma mâliyetlerinin toplamını enazlamak olup aşağıda belirtilmiştir:

$$\min f = \sum_k g_k p_k + \sum_j v_j z_j + \sum_s \sum_k t_{sk} b_{sk} + \sum_k \sum_j a_{kj} f_{kj} + \sum_j \sum_i c_{ji} q_{ji}$$

Modeldeki kısıtlar aşağıda yer almaktadır:

$$\sum_j y_{ji} = 1, \quad \forall i \in I \quad (\text{Her müşterinin ancak bir dağıtım merkezine bağlı olma kısıtı})$$

$$\sum_j d_i y_{ji} \leq W_j z_j, \quad \forall j \in J \quad (\text{Dağıtım merkezi kapasite kısıtı})$$

$$q_{ji} y_{ji} = d_i, \quad \forall i \in I, \forall j \in J \quad (\text{Müşteri talep karşılama kısıtı})$$

$$\sum_k f_{kj} = \sum_i q_{ji}, \quad \forall j \in J \quad (\text{Dağıtım merkezi girdi – çıktı eşitlenmesi})$$

$$\sum_k b_{sk} = \sup_s, \quad \forall s \in S \quad (\text{Tedârikçi kapasite kısıtı})$$

$$\alpha \sum_j f_{kj} \leq \sum_s b_{sk}, \quad \forall k \in K \quad (\text{Gerekli olan hammadde kısıtı})$$

$$\alpha \sum_j f_{kj} \leq D_k p_k, \quad \forall k \in K \quad (\text{Fabrika kapasite kısıtı})$$

$$z_j = \{0, 1\}, p_k = \{0, 1\}, y_{ji} = \{0, 1\} \quad (\text{Karar değişkenleri})$$

$$b_{sk} \geq 0, f_{kj} \geq 0, q_{ji} \geq 0 \quad (\text{Negatif olmama koşulu})$$

Sabri ve Beamon tarafından 2000 yılında yapılan bir çalışmada, İleri Lojistik Ağ Tasarımında çok amaçlı bir Karma Tamsayılı Programlama modeli ortaya koyulmuştur. Modelde ilk amaç mâliyet enazlaması olup sabit (tesis açma) ve değişken (satınalma, taşıma, depolama ve tesis faaliyetini sürdürme) mâliyetler dikkate alınmıştır. İkinci amaç fonksiyonunda ise, tesis ve taşıma kapasitesi kullanım miktarı enbüyüklenmeye çalışılmaktadır. Model statik (tek dönemli), kapasite kısıtlı, deterministik talepli ve çok ürünlü olup, sonuç olarak tesis yeri yerleşimi, müşterilerin tesislere atanması ve taşıma miktarlarını vermektedir. Yazarlar, Karma Tamsayılı Programlama modelini küçük bir örnek üzerinde denemişler, başarılı sonuçlar almışlardır.

Jayaraman ve Pirkul tarafından 2001 yılında yapılan bir çalışmada İleri Lojistik Ağ Tasarımı problemi için ilk olarak Karma Tamsayılı Programlama modeli kurulmuş, daha sonra Lagrange Yakınlaştırma (Relaxation) tabanlı bir sezgisel çözüm yöntemi önerilmiştir. Kurulan modelde amaç sabit ve değişken mâliyetlerin enazlanması olup, model çıktı olarak tesis yerlerini ve tek dönem içerisinde tedârikçi, üretim tesisi, depo ve müşteriler arasındaki malzeme taşıma miktarlarını vermektedir. Ayrıca tesisler kapasite kısıtlı ve çok türde ürün üretebilen özelliktedir. Lagrange Yakınlaştırma yöntemi ile bâzı kısıtlar gevşetilmiş ve modeli adım adım çözen bir sezgisel algoritma geliştirilmiştir. Bu yöntem, her iterasyonda ilk olarak müşterileri belirli depolara atamakta, daha sonra ne kadar ürün üretilmesi gerektiğini hesaplamakta ve taşıma miktarlarını belirlemektedir. Yöntem gerçek yaşamda karşılaşılan bir probleme uygulanmış ve başarılı sonuçlar üretmiştir.

Dasci ve Verter tarafından 2001 yılında yapılan bir çalışmada sürekli (çok dönemli) Lojistik Ağ Tasarımı Problemi için Sürekli Yakınlaştırma yöntemi ile optimum çözüm üreten bir yöntem geliştirilmiştir. Tek ürün için ele alınan problemde talepler deterministik olup, mâliyet enazlaması hedefiyle kapasite kısıtı olmayan üretim merkezlerinin lokasyonu ve hangi bölgeye servis yapacağı belirlenmektedir.

Syarif, Yun ve Gen tarafından 2002 yılında yapılan bir çalışmada İleri Lojistik Ağ Tasarımı problemi için Ağaç Tabanlı Genetik Algoritma geliştirilmiştir. Algoritmada çözümler Prüfer sayıları ile kodlanmış ve uygun olmayan çözümleri onaran bir yordam (prosedür) kullanılmıştır. Yapılan çalışmada problemin öncelikle 0-1 Karma Tamsayılı Programlama modeli kurulmuştur. Oluşturulan modelde amaç, en az mâliyetle açılacak fabrika ve dağıtım merkezlerinin yerlerini ve malzeme akışlarını belirlemek olup tek ürün, tek dönem ve tesis kapasite kısıtı ele alınmıştır. Algoritmada elde edilen çözümler, Matris Tabanlı Genetik Algoritma ve modelin Lindo ile çözümünden elde edilen sonuçlar ile karşılaştırılmış ve Matris Tabanlı Genetik Algoritma'dan daha iyi, optimuma yakın çözümler elde edilmiştir.

Zapfel ve Wasner tarafından 2002 yılında Avusturya'da kargo sektöründe üçüncü parti lojistik servis sağlayıcılarda Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli geliştirmişlerdir. Uygulaması yapılan modelde (Hub and Spoke Model) merkezde bir depo, çevresinde dağıtım merkezi ve müşteriler bulunmaktadır ve dağıtım merkezleri müşterilere yakın konumdadır. Amaç fonksiyonunda salt taşıma mâliyetleri gözönüne alınmaktadır. Geliştirilen model, kural tabanlı bir Uzman Sistem ile bütünleşik kullanılmış ve gerçek yaşamdaki problemler için mâliyet azaltıcı sonuçlar elde edilmiştir.

Sung ve Song tarafından 2003 yılında İleri Lojistik Ağ Tasarımı problemi için Tabu Arama tabanlı sezgisel bir yöntem geliştirilmiştir. Kurulan Karma Tamsayılı Programlama modelinde amaç sabit ve değişken mâliyetlerin enazlanması olup model, çıktı olarak tesis yerlerini ve fabrika, çapraz sevkiyat noktaları ve müşteriler arasında taşıma için ne kadar araç gerektiğini vermektedir. Çapraz Sevkiyat ile depolama süresi ve mâliyeti azaltılarak, ürünlerin müşterilere hızlı bir şekilde ulaştırılması sağlanır. Bu çalışmanın yeniliği, çapraz sevkiyat noktalarını da dikkate alarak, stratejik ve operasyonel düzeyde kararları oluşturan bir yöntem sunmasıdır.

Jayaraman ve Ross tarafından 2003 yılında yapılan bir çalışmada, bir perakende firmasında karşılaşılan Çok Ürünlü Lojistik Ağ Tasarımı problemi için Tavlama Benzetimi algoritması geliştirilmiştir. Oluşturulan Karma Tamsayılı Programlama modelinde, merkezde bir üretim tesisi çevresinde dağıtım merkezleri, çapraz sevkiyat noktaları ve müşteri bölgeleri bulunmakta olup, tek dönem için en az mâliyetle taşıma miktarları ve dağıtım merkezlerinin yerlerinin belirlenmesi amaçlanmaktadır. Algoritma sonuçları optimum çözümler ile karşılaştırılmış ve gerçek yaşamda uygulanabilir sonuçlar çıktı olarak elde edilmiştir.

Miranda ve Garrido tarafından 2004 yılında yapılan bir çalışmada Belirsiz Talepli Lojistik Ağ Tasarımı problemi için, Karma Tamsayılı Programlama modeli oluşturulup, bu modeli çözmek için Lagrange Yakınlaştırma (Relaxation) tabanlı bir sezgisel geliştirilmiştir. Model çok dönemli bir süreçte toplam mâliyeti enazlamak için Ekonomik Sipariş Miktarı ve Güvence (Emniyet) Stoğu gibi depo kontrol politikalarını dikkate alıp, çıktı olarak dağıtım merkezi yerlerini ve hangi müşteriye hangi dağıtım merkezi tarafından servis sağlanacağını vermektedir. Algoritmanın sonuçları geleneksel yöntem ile karşılaştırılmış ve elde tutma mâliyeti ve talep miktarının varyansı arttığında sezgisel yöntem daha iyi çözümler üretmiştir.

Yeh tarafından 2005 yılında yapılan bir çalışmada, İleri Lojistik Ağı Tasarımı problemi için ilk olarak Karma Tamsayılı Programlama modeli oluşturulmuş, daha sonra melez bir sezgisel yöntem geliştirilmiştir. Ele alınan problemde talepler deterministik olup, tesis kapasiteleri dikkate alınarak, tek dönem ve tek ürün için problem çözülmeye çalışılmaktadır. Melez sezgisel yöntem ilk olarak başlangıç çözümü üreten Fırsatçı (Greedy) yöntem, Doğrusal Programlama Tekniği ve üç temel sezgisel (Ekleme, Çıkarma ve Yer değiştirme) yöntemin birleştirilmesi ile oluşturulmuştur. Problemde ilk olarak Fırsatçı yöntem ile tesisler arasındaki yakınlığa göre uygun bir başlangıç çözümü oluşturulmaktadır. Bu çözümde hangi tesislerin açık olacağı belirlenmiştir. Daha sonra Doğrusal Programlama ile bu tesisler arasındaki akışlar hesaplanmakta ve son olarak Ekleme, Çıkarma ve Yer değiştirme sezgiselleri ile çözüm üzerinde değişiklikler yapılarak daha iyi bir çözüme ulaşılmaya çalışılmaktadır. Algoritma çıktı olarak en az mâliyet için tesis yerleri ve taşıma miktarlarını vermektedir.

Melachrinoudis, Messac ve Min tarafından 2005 yılında gerçek yaşamda karşılaşılan Çok Amaçlı İleri Lojistik Ağı Tasarımı problemi için Karma Tamsayılı Programlama modeli oluşturulmuştur. Model tek dönemli ve tek ürün için olup, tesis kapasiteleri dikkate alınarak üretim ve dağıtım merkezlerinin yerleri ile tesisler ve müşteriler arasındaki taşıma miktarları belirlenmeye çalışılmaktadır. Yapılan çalışmanın önceki çalışmalardan farkı, toplam mâliyetin enazlanmasının yanında, müşteri taleplerini karşılayabilme oranının enbüyüklenmesi amaçlanmaktadır. Çok amaçlı modelleme Lineer Fiziksel Programlama yöntemi ile yapılmış olup, karar verici Bulanık Mantığa benzer olarak amaçları belirli sınıflara göre ayırmaktadır.

Amiri tarafından 2006 yılında yapılan bir çalışmada, İleri Lojistik Ağı Tasarımı problemi için Karma Tamsayılı Programlama modeli kurulmuş ve modeli çözmek için Lagrange Yakınlaştırma (Relaxation) tabanlı bir sezgisel geliştirilmiştir. Lagrange Yakınlaştırma yönteminin mantığı, problemi zorlaştıran kısıtların dualize edilip bir ceza ağırlığı (Lagrange

multiplier) ile amaç fonksiyonuna eklenmesidir. Bu şekilde problem, genellikle çözülmesi ve parçalanması kolay alt problemlere bölünür. Kurulan modelde amaç, sabit ve değişken mâliyetlerin enazlanması olup, model çıktı olarak üretim ve dağıtım tesis yerlerini ve tesisler arasındaki taşıma miktarını vermektedir. Bunun yanında, yapılan çalışmanın önceki çalışmalardan farkı, kurulan modelin sonuç olarak tesis kapasitelerini de içermesidir. Bu durum, modele farklı kapasite düzeylerinde, karar değişkenlerinin eklenmesi ile elde edilmiştir.

Gen, Altıparmak ve Lin tarafından 2006 yılında yapılan bir çalışmada, İleri Lojistik Ağ Tasarımı problemi için Öncelik Tabanlı Kodlama ile Genetik Algoritma geliştirilmiştir. Öncelik Tabanlı Kodlama yönteminde müşteriler veya tesisler öncelik değerlerine göre sıralanarak, en düşük mâliyetli müşteriler veya tesisler ile ilişkilendirilmektedir. Problem kapasite kısıtlı ve deterministik talepli olup, tek ürün ve tek dönem üzerinden toplam mâliyetin enazlanmasına çalışılmaktadır. Yazarlar, Genetik Algoritma için Ağırlık Eşleme Çaprazlaması adını verdikleri yeni bir çaprazlama yöntemi geliştirmiş olup, bu yöntemde tek nokta üzerinden çaprazlama yapılmakta ve öncelik değerlerini tutarlı duruma getirmek için eşleme yapılmaktadır. Algoritma sonucunda dağıtım merkezlerinin yerleri ve tesisler arasındaki akışlar çıktı olarak verilmektedir. Geliştirilen yöntem, Syarif ve diğerleri tarafından 2002 yılında hazırlanan Genetik Algoritma ile karşılaştırılmış ve özellikle büyük problemler için daha kaliteli çözümler üretmiştir.

Altıparmak ve diğerleri tarafından 2006 yılında geliştirilen bir çalışmada gerçek yaşamda karşılaşılan Çok Amaçlı Lojistik Ağı problemi için ilk olarak Karma Tamsayılı Programlama modeli oluşturulmuş, daha sonraki aşamada bir Genetik Algoritma önerilmiştir. Algoritmada Öncelik Tabanlı Gösterim tekniği kullanılmış olup, çaprazlama ve mutasyon, lojistik düzeyler/katmanlar (tedârikçi - fabrika, fabrika - dağıtım merkezi) üzerinden yapılmıştır. Tek dönem ve tek ürün üzerinden yürütülen modelde talepler deterministik olup, tesis kapasite kısıtı dikkate alınarak üretim, dağıtım merkezlerinin yerleri ve malzeme akışları hesaplanmaktadır. Çok amaçlı problem için ilk amaç sabit ve değişken mâliyetleri enazlamak iken, diğer amaçlar, müşteri taleplerini karşılayabilme oranını ve fabrika ve dağıtım merkezleri için kapasite kullanım oranı arasındaki dengeyi enbüyüklemektir. Yazarlar geliştirdikleri Genetik Algoritma'yı Tavlama Benzetimi yöntemi ile karşılaştırmışlar ve daha iyi çözümler elde etmişlerdir.

Cordeau, Pasin ve Solomon tarafından 2006 yılında yapılan bir çalışmada tek periyotlu deterministik Lojistik Ağ Tasarımı Problemi için bütünleşik bir model önerilmiştir.

Geliştirilen modelde tesis yerleri ve kapasitelerinin belirlenmesi, tedârikçi ve taşıma türlerinin seçimi, ürün atamaları ve ürün akışları birleşik olarak çözülmeye çalışılmaktadır. Bu modeli çözmek için Simpleks tabanlı Dal Sınır Yöntemi ve Benders Parçalara Ayırma Algoritması kullanılmıştır. Amaç fonksiyonunda sabit ve değişken mâliyetler enküçüklenmeye çalışılmaktadır. Sabit mâliyetler olarak tesis yerlerinin belirlenmesi, ürünlerin tesislere atanması ve taşıma türlerinin seçimi gözönüne alınmıştır. Değişken mâliyet olarak ise taşıma, depolama ve tesis işletme mâliyetleri ele alınmıştır. Sonuçlara bakıldığında Benders Parçalara Ayırma Algoritmasının daha iyi sonuçlar ürettiği görülmüştür.

Shimizu, Matsuda ve Wada, 2006 yılında deterministik olmayan taleplerin yer aldığı esnek bir Lojistik Ağ Tasarımı için melez bir Tabu Arama algoritması geliştirmişlerdir. Esneklik analizinde kabul edilebilir düzeyde parametre sapmalarının miktarı ölçülmekte ve böylece gerçek yaşamda karşılaşılan lojistik problemlerine çözümler üretilmeye çalışılmaktadır. Önerilen algoritma iki aşamalı olup, ilk aşamada tesislerin yerleri Tabu Arama algoritması ile belirlenmekte, ikinci aşamada bir çizge (graf) algoritması ile rota seçimi yapılmaktadır. Geliştirilen algoritmadan elde edilen çözümler Karma Tamsayılı Programlama modellemesi ile elde edilen çözümlerle karşılaştırılmış ve optimuma yakın başarılı sonuçlar elde edilmiştir.

Meepetchdee ve Shah tarafından 2007 yılında yapılan bir çalışmada Çok Amaçlı Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli oluşturulmuştur. Yapılan çalışmada amaçlar; lojistik ağın etkinliğini ve dayanıklılığını arttırmak ve ağın karmaşıklığını azaltmak olarak belirlenmiştir. Yazarlar bir lojistik ağın etkinliğini mâliyet, kârlılık, müşteriye yanıt verme süresi ve talepleri karşılama oranıyla ilişkili olarak tanımlamışlar, kurguladıkları modelde mâliyet enküçüklemeyi amaçlamışlardır. Yazarlara göre bir lojistik ağın dayanıklılığı, ağ içindeki bir tesisin veya akışın kaldırılması sonucunda, toplam talebin hangi oranla karşılanabileceği ile ilgilidir. Bir lojistik ağın karmaşıklığı ise, o ağdaki varolan bağlantı sayısının, ağdaki her noktayı bağlayan en az sayıdaki bağlantı sayısına (Minimum Spanning Tree) oranıdır. Modelde tesis kapasite kısıtı dikkate alınmayıp, tesis sayısı bir kısıt olarak eklenmiştir. Tek dönem tek ürün üzerinden çalıştırılan modelde, dağıtım tesislerinin yerleri ve dağıtım ağındaki akışlar küçük örnekler üzerinden hesaplanmaktadır. Çalışmanın sonuçlarına göre eğer bir lojistik ağın dayanıklılığı arttırılmak isteniyorsa, sistemin karmaşıklığının arttırılması gerekmektedir.

Tsiakis ve Papageorgiou tarafından 2008 yılında yapılan bir çalışmada, gerçek yaşamda karşılaşılan bir İleri Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli oluşturulmuştur. Kurulan modelin daha önce hazırlanan modellerden farkı, tesisler

arasında ürün akışlarında kapasite kısıtının olması ve model sonucunda üretim ve dağıtım tesisi yerlerinin, kapasitelerinin ve tesisler arasındaki taşıma miktarlarının hesaplanmasıdır. Tek dönemli ve çok ürün için çalıştırılan model, farklı senaryolar üzerinden küçük örnekler için başarılı sonuçlar üretmiştir.

Kengpol tarafından 2008 yılında yapılan bir çalışmada, Mekong bölgesinde toplam dağıtım mâliyetlerini düşürmek amaçlı Lojistik Ağ Tasarımı Problemi için bir Karar Destek Sistemi geliştirilmiştir. Karar Destek Sisteminde ilk olarak Analitik Hiyerarşi Modeli ile hangi müşteri gereksinimlerinin önemli olduğu saptanmış, daha sonra Karma Tamsayılı Programlama ile bu taleplerin en iyi şekilde karşılandığı model kurulmuş ve Çok Ürünlü Taşıma Modeli ile en büyük müşteri hoşnutluğu ve en az taşıma mâliyeti için müşterilere ulaştırılacak optimum ürün miktarı belirlenmiştir. AHP ile taşıma işlemlerinin yarar, esneklik ve mâliyet ölçütlerine göre bir lojistik firması tarafından, üreticinin kendisi tarafından veya karışık olarak yapılması seçenekleri gözönüne alınmıştır. Hazırlanan Karma Tamsayılı Programlama Modelinde taşıma miktarlarına göre bu seçeneklerden sağlanan hoşnutluk enbüyüklenmeye çalışılmıştır.

Qin ve Qin tarafından 2008 yılında konkav mâliyet fonksiyonlu Lojistik Ağ Problemi için sezgisel bir rotalama yöntemi ile melezleştirilmiş Parçacık Sürü Algoritması önerilmiş ve bir otomotiv lojistik firmasında uygulaması yapılmıştır. Bu sezgisel rotalama yöntemi, bir müşterinin talebinin hangi üretim ya da dağıtım merkezinden karşılanacağını belirlemektedir. Amaç fonksiyonunda depolama, tesis açma, taşıma ve elde bulundurmama mâliyetleri enküçüklenmeye çalışılmaktadır. Taşıma mâliyetleri hesaplanırken Euler değişkeninin yer aldığı konkav mâliyet fonksiyonu kullanılmış ve sonuçlar Genetik Algoritma ve Saf Parçacık Sürü Algoritması ile karşılaştırılarak, geliştirilen yöntemin daha iyi sonuçlar elde ettiği ortaya konulmuştur.

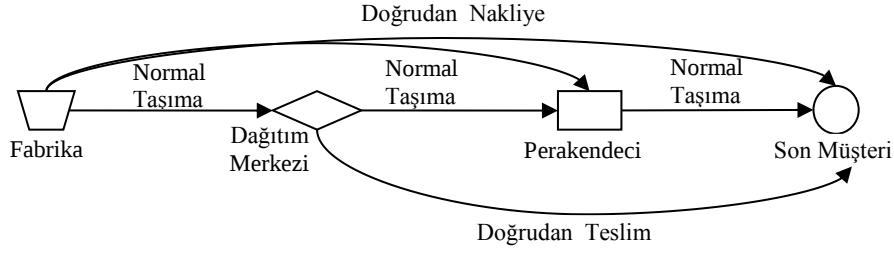
Cordeau, Laporte ve Pasin tarafından 2008 yılında Deterministik Lojistik Ağ Tasarımı Problemi için Tepe Tırmanma süreci ile birleştirilmiş Yerel Arama Sezgiseli sunulmuştur. Uygulaması yapılan problemde tedârikçilerin seçimi, fabrika ve depo yerlerinin belirlenmesi, hammaddelerin tedârikçilere, ürünlerin fabrika ve depolara atanması ve hammadde ve ürünlerin bu ağ içindeki akışları birlikte düşünülmüştür. Ayrıca her hammaddenin salt bir tedârikçi tarafından karşılandığı ve her ürünün salt bir fabrikada üretildiği ve bir depoda depolandığı varsayılmıştır. Amaç fonksiyonunda taşıma, üretim ve depolama mâliyetleri enküçüklenmeye çalışılmıştır. Yerel Arama Sezgiseli'nde çözümler, ilk olarak rassal bir

şekilde bozulmakta, daha sonra yerel arama yapılarak optimum çözüme ulaşılmaya çalışılmaktadır.

Liu tarafından 2008 yılında yapılan bir çalışmada Çok Amaçlı Lojistik Ağ Tasarımı Problemi için geliştirilen Stokastik Akış Ağı modellenli bir Genetik Algoritma uygulaması yapılmıştır. Çalışmada Deb'in 2002 yılında geliştirdiği Genetik Algoritma, çok amaçlı problem için denenmiş ve başarılı sonuçlar elde edilmiştir. Stokastik Akış Ağı, çizge modelinde her nokta için en yakın noktalar bulunarak oluşturulan bir lojistik ağıdır. Uygunluk fonksiyonunda taşıma güvenilirliği, taşıma mâliyeti ve süresi dikkate alınmıştır.

Men ve Wen tarafından 2008 yılında yapılan bir çalışmada, dört aşamalı Lojistik Ağ Tasarımı için Parçacık Sürü Algoritması geliştirilmiştir. Yapılan çalışmada ilk olarak lojistik merkezlerin yerleri belirlenmekte, daha sonra taşıma rotaları saptanmaktadır. Kurulan modelde tesis açma, taşıma ve depolama mâliyetleri, işleme kapasitesi dikkate alınmış ve amaç olarak mâliyet enküçüklenmesi belirlenmiştir. Yazarlar, Parçacık Sürü Algoritmasında iterasyonlar boyunca Atalet (Inertia) Ağırlığını değiştirmek için Doğrusal Olmayan Fonksiyon önermişler ve başarılı sonuçlar elde etmişlerdir.

Lin, Gen ve Wang'ın 2009 yılında geliştirdiği Melez Genetik Algoritma yöntemi ile Çok Aşamalı Esnek Lojistik Ağ Problemi için bir çözüm şekli önerilmiştir. Geleneksel Çok Aşamalı Lojistik Ağ Tasarımı Problemi, bir kuruluşa ait lojistik mâliyetlerini enküçüklemek veya toplam kârı enbüyüklemek için, deterministik müşteri taleplerini karşılamak amacıyla üretim, dağıtım ve perakendeci tesisler arasındaki ürün dağıtımının plânlanması problemidir. Geleneksel Lojistik Ağ Tasarımı Probleminde ürünlerin; fabrika, dağıtım merkezi, perakendeci ve müşteri sıralaması ile dağıtımı yapılmakta iken, bu çalışmada fabrikadan doğrudan perakendeciye veya müşteriye (doğrudan dağıtım) yapılan taşımalar ile dağıtım merkezlerinden müşteriye (doğrudan teslim) yapılan nakliyeler de gözönüne alınmıştır (Şekil 2.5). Ayrıca geliştirilen yöntemde dağıtım plânlamanın yanısıra stok kontrolü de yapılmaktadır. Hazırlanan Genetik Algoritma yönteminde Öncelik Tabanlı Kodlama tekniği kullanılmış ve algoritmaya arama yeteneğini arttırmak için bir Bulanık Mantık kontrol mekanizması eklenmiş ve yerel arama tekniği ile birleştirilmiştir.



Şekil 2.5 : Geleneksel ve Esnek Çok Aşamalı LAT Problemi (Lin ve diğ, 2009).

Geleneksel Lojistik Ağ Tasarımında amaç, genel olarak taşıma ve tesislerin yönetim mâliyetini enazlamaktır ve depolama mâliyetleri gözönüne alınmamaktadır. Yazarlar tarafından geliştirilen çözüm yönteminde uzun tedârik süreleri, gecikmeler ve talep miktarındaki dalgalanmalar da dikkate alınmaktadır. Daha önce geliştirilen tekniklerde, problem alt problemlere ayrılıp global optimum çözümden uzak ve gerçek yaşamda uygulanması zor çözümler bulunmuştur. Buna karşılık bu yöntemde, esnek hatlar ile tesislerin lokasyonlarının belirlenmesinde bu tesisler benzer tesisler olarak gözönüne alınmış, atama sürecinde ise Wagner Within algoritması kullanılmıştır. Amaç fonksiyonu; taşıma mâliyeti, elde bulundurma mâliyeti, sipâriş verme mâliyeti ve tesis mâliyeti toplamını enküçüklemek olarak tanımlanmıştır. Tesis mâliyeti sabit (tesis açma) ve değişken mâliyet olmak üzere ikiye ayrılmaktadır. Kısıt olarak fabrika üretim kapasitesi, dağıtım merkezi ve perakendeci depo kapasitesi, müşteri taleplerinin tam zamanında karşılanması yer almaktadır.

Miranda, Garrido ve Ceroni'nin 2009 yılında yaptıkları çalışmada, stratejik ve taktik düzeyde Lojistik Ağ Tasarımı Problemi için otomasyon tabanlı işbirlikli ardışık bir yaklaşım önermişlerdir. Sunulan yöntemde üç aşamalı ve tek üretim tesisli tedârik zinciri sistemi için tesis yerleşimi, depo kontrolü, filo tasarımı ve müşterilerin gruplanması, e-iş modeli ile aynı anda gerçekleştirilmektedir. Geliştirilen modelde bir iterasyonda ilk olarak müşteriler gruplanıp filo tasarımı yapılmakta, daha sonra depo kontrolü gözönüne alınarak dağıtım ağı tasarımı yapılmaktadır. Model değişkenleri iki kümeye ayrılmıştır: Birinci kümede filo tasarımı ve müşteri gruplama ile ilgili değişkenler tutulmaktadır. İkinci kümede ise depo kontrol kararları ile dağıtım ağ tasarımı değişkenleri yer almaktadır. Amaç fonksiyonunda depo açma, depo işletme, sipâriş verme, elde tutma, güvence stoğu ve taşıma mâliyeti gözönüne alınmıştır. Sonuçlar, müşteri gruplaması ve dağıtım ağı tasarımının ayrı ayrı yapıldığı Doğrusal Olmayan Tamsayı Karmaşık Programlama Modelleri ile karşılaştırılmış ve eşzamanlı işbirlikli ardışık yaklaşımın daha iyi sonuçlar ürettiği görülmüştür.

Chiou tarafından 2009 yılında geliştirilen yöntemde Lojistik Ağ Tasarımı Problemi için iki aşamalı programlama modeli önerilmiştir. Bu modelde kesin yöntemler kullanılmakta olup, Lojistik Ağ Probleminde türevi alınamayan akışlar ile ilgili bazı denklemler için Tepe Tırmanma yöntemine benzer iteratif bir algoritma kullanılmıştır. Üzerinde çalışılan problem, marjinal mâliyetlerin kullanıldığı değişken eşitsizlikler olarak formüle edilmiştir. Oluşturulan programlama modelinde üst aşamada gelir değişkenleri yer alırken, alt aşamada marjinal mâliyet değişkenleri yer almaktadır. Bu sâyede optimum çözüm, daha az hesaplama yapılarak bulunmuştur.

Yi ve Dong tarafından 2009 yılında otomotiv sektöründe karşılaşılan Lojistik Ağ Tasarımı Problemi için Melez Parçacık Sürüsü Algoritması önerilmiştir. Geliştirilen algoritma ile depo ve rota optimizasyonu aynı anda gerçekleştirilmekte ve taşıma, depolama, tesis kurma ve elde bulundurmama mâliyetlerinin enazlanması amaçlanmaktadır. Parçacık Sürüsü Algoritması, Genetik Algoritmalar ve Tavlama Benzetimi ile melezleştirilerek daha iyi sonuçlar elde edilmiştir.

2009 yılında Jin ve Xin tarafından hazırlanan çalışmada Lojistik Ağ Tasarımı Problemi, en az mâliyeti sağlamak için tesislerin yerlerinin belirlendiği ve müşteri taleplerinin ayrıldığı (tahsis edildiği) bir problem olarak tanımlanmıştır. Geliştirilen Birleşik Tavlama Benzetimi yöntemi, ilk olarak tesis yerlerinin belirlendiği kısım ve daha sonra müşteri taleplerinin karşılandığı kısım olmak üzere iki tabakadan oluşmaktadır. İlk kısımda komşuluk fonksiyonları olarak bir tesisin yeni bir yere taşınması, boş olan bir alanda yeni bir tesisin açılması ve varolan bir tesisin kapanması gözönüne alınmıştır. İkinci kısımda ise bir müşteri talebinin farklı bir tesis tarafından karşılanması ve iki müşteri arasında gereksinimi karşılayan tesislerin yer değiştirmesi işlemleri, komşuluk fonksiyonu olarak kullanılmıştır. Yapılan çalışmada amaç, tesis açma (aktifleştirme) ve taşıma mâliyetlerinin toplamını enküçükmektir.

2011 yılında Martinez ve Zang, birden fazla ürün ve sağlayıcının yer aldığı çok amaçlı bir Lojistik Ağ Tasarımı modeli için Pareto Karınca Kolonisi yöntemi geliştirmişlerdir. Bu çalışmada sözügeçen üretim süreçlerinde toplam mâliyet enküçüklenmeye çalışılırken, aynı zamanda ürünler müşteriye en hızlı şekilde ulaştırılmaya çalışılmaktadır. Tasarlanan algoritma örnek bir problem üzerinde farklı parametrelerle denenerek karşılaştırmalar yapılmıştır.

Kadadevaramath ve diğerleri tarafından 2012 yılında geliştirilen Parçacık Sürü Algoritması ile otomotiv sektöründe üç aşamalı İleri Lojistik Ağ Problemi için aynı model üzerinde 20

senaryo oluşturularak çözümler bulunmuştur. Kurgulanan model üzerinde amaç fonksiyonu, toplam tedârik zinciri mâliyetini enküçükmek olup, bu mâliyetle satınalma, üretim ve taşıma mâliyetleri dahil edilmiştir. Hazırlanan algoritma üzerinde parametre değerleri değiştirilerek denemeler yapılmış ve son olarak problemler üzerine Genetik Algoritma da uygulanarak sonuçlar karşılaştırılmıştır.

2.2 Tersine Lojistik Ağ Tasarımına İlişkin Yapılan Çalışmalar

Bu başlık altında, literatürde 1998 yılından bu tez çalışmasının teslim tarihine kadar salt Tersine Lojistik kapsamında yapılan Ağ Tasarımı çalışmaları incelenmiştir. Tersine Lojistik Ağ Tasarımı modelinde kullanılmış ürünlerin müşteriden üreticiye akışlarının yanında, toplama ve onarım gibi tesislerin yerleri de belirlenmektedir.

Marin ve Pelegrin tarafından 1998 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımı için ilk olarak Karma Tamsayılı Programlama modeli kurulmuş ve daha sonra modeli çözmek için Lagrange Yakınlaştırma (Relaxation) tabanlı bir sezgisel geliştirilmiştir. Model üzerinde tesis kapasiteleri dikkate alınmamış olup, salt fabrika lokasyonları ve tek dönem ve tek ürün için en az mâliyetle müşteri-fabrika arasındaki taşıma miktarları belirlenmeye çalışılmıştır.

Jayaraman, Guige ve Srivastava tarafından 1999 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli geliştirilmiştir. Tek ürün ve çok dönemde deterministik talepler için çalıştırılan modelde tesis kapasite kısıtları dikkate alınarak sabit ve değişken mâliyet enküçüklemesi hedeflenmektedir. Oluşturulan ağda, toplama merkezleri sabit tutularak, iyileştirme (onarım) ve yeniden dağıtım merkezlerinin lokasyonları ve ayrıca lojistik ağ içindeki taşıma miktarları belirlenmeye çalışılmaktadır.

Krikke, Harten ve Schuur tarafından 1999 yılında gerçek yaşamda karşılaşılan Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli geliştirilmiştir. Modelde tesis kapasitesi dikkate alınmamış olup, açılacak en fazla tesis sayısı kısıtı yer almaktadır. Tek ürün ve tek dönem üzerinden işleme, taşıma ve depolama mâliyetlerinin enazlanması için yürütülen çalışmanın diğer çalışmalardan farkı, çıktı olarak tesis yerleri ve dağıtım akışlarının yanında yedek parçaların depo miktarlarının ve ara stoklarının da gözönünde bulundurulmasıdır.

Jayaraman, Patterson ve Rolland tarafından 2003 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımı için ilk olarak Karma Tamsayılı Programlama modeli oluşturulmuş ve daha sonra bir sezgisel yöntem önerilmiştir. Kurulan model, tek dönemli ve deterministik

talepli tek ürün için olup, mâliyet enküçüklenmesi hedeflenmiştir. Model çıktı olarak, toplama ve iyileştirme tesisleri için lokasyon ve lojistik ağ içindeki akışları vermektedir. Sezgisel yöntem üç adımda çalışmakta olup, ilk önce rassal olarak toplama ve iyileştirme merkezleri seçilmekte ve modeller çözülerek sonuçlar saklanmaktadır. İkinci adımda ise en iyi sonuçları veren toplama ve iyileştirme merkezleri birleştirilerek yeni bir model oluşturulmakta ve bu problem çözülerek, önceki sonuçlarla karşılaştırılmaktadır. Son adımda ise bulunan en iyi çözüme, sıra ile seçilmeyen toplama ve iyileştirme merkezleri birer birer eklenerek çözüm geliştirilmeye çalışılmaktadır. Özetlemek gerekirse; sezgisel yöntem problemi alt problemlere ayırmakta ve daha az karar değişkeni ile alt problemler için çözümler oluşturmakta, daha sonra ise bu çözümler birleştirilerek ana problem için çözüm elde edilmektedir.

Listes ve Dekker tarafından 2005 yılında yapılan bir çalışmada, gerçek yaşamda karşılaşılan Stokastik Talepli Tersine Lojistik Ağ Tasarımı problemi için Stokastik Karma Tamsayılı Programlama modeli oluşturulmuştur. Tek ürün ve çok dönem üzerinden çalıştırılan modelde tesislerin kapasite kısıtları dikkate alınarak en yüksek kârlılık için, toplama ve geri dönüşüm merkezlerinin yerleri belirlenmekte ve geri dönüştürülecek ürünlerin bu ağ içindeki akışları hesaplanmaktadır. Stokastik programlama yaklaşımında, her senaryonun bir olasılığı bulunmakta ve bulunan çözüm, bir senaryo için optimum sonucu değil, bu senaryolar arasında dengeli bir sonucu sağlamaktadır.

Min ve diğerleri tarafından 2006 yılında yapılan bir çalışmada, Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli oluşturulmuş ve bir Genetik Algoritma geliştirilmiştir. Ele alınan problemde, tek ürünün çok dönemli hareketleri incelenmiş ve tesis kapasite kısıtları dikkate alınmamıştır. Taşıma, kiralama, depolama ve işleme mâliyetlerinin enküçüklenmesi amaçlanan modelde, ürünlerin ağ içindeki akışları hesaplanarak toplama merkezlerinin yerleri belirlenmeye çalışılmaktadır. Hazırlanan Genetik Algoritmada karar değişkenlerinin gösterildiği Matris Tabanlı kodlama kullanılmış olup, İkili Turnuva yöntemi ile çözümler seçilmiş ve Tek Noktadan Çaprazlama yöntemi ile bireyler çaprazlanmıştır. Geliştirilen yöntem sanal bir problem üzerinde denenmiş ve başarılı sonuçlar elde edilmiştir.

Üster ve diğerleri tarafından 2007 yılında otomotiv endüstrisinde karşılaşılan Kapalı Döngülü Tersine Lojistik Ağ Tasarımı Problemi için hazırlanan çalışmada Karma Tamsayılı Programlama modeli kurulmuş ve Benders Ayırıştırma Algoritması ile bir çözüm geliştirilmiştir. Kurulan modelde tesis kapasiteleri dikkate alınmamış olup, süreç en az mâliyet amacıyla çok ürün ve tek dönem üzerinden yürütülmüştür. Modelde ilk olarak toplama merkezleri tarafından ürünler müşterilerden toplatılmakta, iyileştirme (yeniden

üretim) merkezinde onarımları/iyileştirilmeleri gerçekleştirilmekte ve üretim merkezine gönderilmekte, en son olarak dağıtım merkezinden yeniden müşterilere ulaştırılmaktadır. Algoritma sonucunda, toplama ve iyileştirme tesislerinin lokasyonları ve ürünlerin akış rotası belirlenmeye çalışılmaktadır.

Lieckens ve Vandaele tarafından 2007 yılında yapılan bir çalışmada, tek ürünlü ve tek aşamalı bir dağıtım sürecinde Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama Modeli önerilmiş ve bu model kullanılarak Diferansiyel Evrim tabanlı bir Genetik Algoritma geliştirilmiştir. Model bazı dinamik yaklaşımlarla zenginleştirilerek, geri dönüşüm tesislerinde kuyruk modeli uygulanmıştır. Amaç, geri dönüşüm tesislerinin yerleşimini ve ürün akışlarını belirleyerek toplam kârı enbüyükleme. Bu da, geri dönüşümü yapılmış ürün gelirlerinden tesis açma, geri dönüştürme, depolama, taşıma, müşteriden geri dönüşümü yapılacak ürünleri toplamama, geri dönüştürülmüş ürünler için müşteri gereksinimlerini karşılamama (elde bulundurmama) ve elden çıkarma (yok etme) mâliyetlerinin toplamının çıkarılması ile bulunmaktadır. Hazırlanan GA, Kuyruk Modeli ile toplama, bu kuyruk üzerinden akış ve parçalara ayırma işlemleri ile yürütülmektedir.

Zhou, Cao, Meng ve Xu tarafından 2008 yılında Onarım Lojistiği Ağ Problemi için çift amaçlı Tamsayılı Programlama Modeli ve Genetik Algoritma yöntemi geliştirilmiştir. Tek ürün üzerinden çalıştırılan modelde onarım hizmet (servis) merkezi ve müşterilerin hangi tesisler tarafından taleplerinin karşılanacağı belirlenmektedir. Müşterilerin parça değiştirme ve bakım olmak üzere iki tip talepleri vardır. Modeldeki amaçlardan birincisi sabit işletme ve depolama mâliyetleri toplamını düşürmek iken, diğeri en fazla müşteri bekleme süresini enküçükleme. Geliştirilen Genetik Algoritmada Tek Noktalı Çaprazlama ve Mutasyon operatörü kullanılmıştır.

Demirel ve Gökçen tarafından 2008 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli ortaya konulmuştur. Çalışmada tesis kapasite kısıtları dikkate alınmış olup, tek dönem için en az mâliyetle deterministik talepli çok ürünün tersine lojistiği hedeflenmiştir. Kurulan modelde tesis açma, üretim, taşıma, toplama, satınalma, demontaj ve yoketme mâliyetleri amaç fonksiyonunda yer almaktadır. Model sonucunda toplama ve dağıtım merkezlerinin yerleri belirlenmeye çalışılmakta ve ürünlerin ağ içindeki taşıma miktarları hesaplanmaktadır. Ayrıca kurulan modelin duyarlılığını kontrol etmek için müşterilere satılan ürünlerin 3 farklı oranda geri dönüşü incelenmiş ve sonuç olarak müşterilerden ne kadar fazla kullanılmış ürün geri dönerse mâliyetlerin o kadar

azalacağı gözlemlenmiştir. Yazarlar problem büyüdükçe çözüm süresi üstel olarak arttığından sezgisel bir yöntemin daha uygun olacağını belirtmişlerdir.

Aras, Aksen ve Tanugur tarafından 2008 yılında yapılan bir çalışmada, Tersine Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli kurulmuş ve Tabu Arama tabanlı bir sezgisel geliştirilmiştir. Kurulan model tek dönemli ve çok ürünlü olup, tesis kapasitesi yerine tesis sayısı kısıtları dikkate alınmıştır. Yapılan çalışmanın diğer çalışmalardan farkı, toplama merkezi yerlerinin ve ürünlerin ağ içindeki hareketlerinin belirlenmesinin yanında, ne kadar taşıma aracının gerekeceği ve geri alınacak kullanılmış ürün fiyatlarının ne olacağı bilgisinin de problem sonucunda elde edilmesidir. Tabu Arama yöntemi, toplama merkezi yerleri belirlenirken kullanılmış ve her iterasyonda Nelder – Mead Simpleks Arama yöntemi çağırılarak araçların sayısı ve yükleme durumlarına karşılık gelen bir çözüm değeri elde edilmiştir. Geri alınacak kullanılmış ürün fiyatlarını belirlemek için ise sabit ve kaliteye bağımlı iki politika belirlenmiş ve denemelerde kaliteye bağımlı politika daha yüksek kârlılık getirmiştir.

Du ve Evans tarafından 2008 yılında yapılan bir çalışmada Çok Amaçlı Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Programlama modeli oluşturulmuş ve problem için Melez Serpilme Arama algoritması geliştirilmiştir. Problem kapasite kısıtlı, deterministik talepli, tek dönemli ve çok ürünlü olup, onarım merkezlerinin yerleri, kapasiteleri ve taşıma miktarları bulunmaya çalışılmaktadır. Üretim ve dağıtım merkezlerinin dikkate alındığı modelde ilk amaç toplam mâliyeti enküçükmek iken, diğer amaç müşteri taleplerini karşılama oranını enbüyüklemektir. Serpilme Arama algoritması, Dual Simpleks ve Sınır yöntemi ile melezleştirilmiş olup, muhtemel tesis yerlerindeki kapasiteyle ilgili karar değişkenlerine atama yapılmaktadır. Dual Simpleks yöntemi, modeldeki taşımayla ilgili sürekli değişkenler için bir çözüm değeri oluşturur. Son olarak Sınır yöntemi ile varolan çözüm iyileştirilmeye çalışılmaktadır. Algoritma sonuçları, ilk amaca göre bir çözüm oluşturulduğunda merkezî bir yapının, ikinci amaca göre bir çözüm oluşturulduğunda merkezî olmayan bir lojistik ağ tasarımının yapıldığını ortaya koymuştur.

Shanshan ve Kejing tarafından 2008 yılında elektronik sektöründe Tersine Lojistik Ağ Tasarımı için Karma Tamsayılı Doğrusal Programlama modeli geliştirilmiştir. Oluşturulan modelde toplama, parçalara ayırma, geri dönüştürme, onarım mâliyetleri ve tesis kapasiteleri dikkate alınarak tesis lokasyonları ve ürün akışları belirlenmeye çalışılmıştır. Daha önce yapılan çalışmalardan farklı olarak elektronik sektöründe geri dönüşümü yapılacak ürünler için de tesisler arasında alt ağ tasarımı yapılmıştır.

Lee ve Dong'un 2009 yılında yayınladıkları çalışmada Tersine Lojistikte belirsizlik altında (stokastik durumlarda) dinamik ağ tasarımı için iki aşamalı bir Tavlama Benzetimi yöntemi geliştirmişlerdir. Geliştirdikleri yöntem kapalı döngülü bir lojistik ağı için tesis yerleşimi ve dağıtım plânlamasını yaparken, aynı zamanda ürünlerin ve geri dönüştürülecek parçaların işlenebildiği melez tesisleri de dikkate almaktadır. Amaç fonksiyonunda taşıma ve tesis açma mâliyetleri toplamı enküçüklenmeye çalışılmıştır. Problemden zamanla değişen müşteri talepleri için o periyotta hangi tesislerin aktif olacağı ve tesislerin ürünleri mi, yoksa geri dönüşümü yapılacak parçaları mı toplayacağı dinamik olarak belirlenmektedir.

Lee, Gen ve Rhee tarafından 2009 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımında taşıma ve tesis açma mâliyetleri toplamını enazlamak için Melez Genetik Algoritma geliştirilmiştir. Uygulaması yapılan algoritmada Öncelik Tabanlı Kodlama yöntemi, Ağırlık Eşlendirmeli Çaprazlama yöntemi ve mutasyon operatörü için ise Ekleme sezgiseli kullanılmıştır.

Lu, Li ve Liang tarafından 2008 yılında Çok Amaçlı Tersine Lojistik Ağ Tasarımı için Parçacık Sürü Algoritması geliştirilmiştir. Amaç fonksiyonunda doğaya negatif etkinin ve toplam mâliyetin enküçüklenmesi yer almaktadır. Toplam mâliyet içerisinde tesis açma, taşıma, depolama ve işleme mâliyeti gözönüne alınmakta ve algoritma sonucunda tesis lokasyonları ve rotalar belirlenmektedir. Doğaya negatif etki ise lokasyon, frekans, kalite ve sayı gibi geri dönüşüm etmenleri ile ölçülmektedir. Parçacık Sürü Algoritması ile Afet (Cataclysm) Kuramı birleştirilerek oluşturulan yöntemde yazarlar, algoritma tarafından bulunan çözümün global optimum olduğunu iddia etmektedirler. Afet Kuramı ile algoritmada yoğunlaşma hızlı bir şekilde yapılmakta ve eğer belirli sayıdaki iterasyonda çözümde bir iyileşme sağlanmazsa, her parçacığın pozisyon ve hızları rassal olarak belirlenmektedir. Bu sâyede yerel optimum noktalardan kaçılmış olur.

Qin ve Ji tarafından 2010 yılında Tersine Lojistikte Ağ Tasarımı konusunda yapılan çalışmada Genetik Algoritma, Bulanık Mantık simülasyonu ile birlikte melez olarak kullanılmıştır. Hazırlanan simülasyonda yer alan değişkenler için olasılık dağılımları yerine Bulanık Mantık ile değer ataması yapılmıştır. Geliştirilen tersine lojistik modelinde müşteriler, toplama merkezi ve fabrika olmak üzere üç katman bulunmaktadır. Her toplama merkezinin bir kapasitesi bulunmakta olup, amaç fonksiyonunda toplama merkezi açma mâliyeti, taşıma mâliyeti ve müşterilerden toplamama mâliyeti gözönüne alınmıştır. Hazırlanan Genetik Algoritma da Bulanık Mantık başlangıç çözümlerinin üretilmesinde ve her kromozomun uygunluk değerlerinin belirlenmesinde kullanılmıştır.

Soleimani and Govindan tarafından 2014 yılında yapılan bir çalışmada Tersine Lojistik Ağ Tasarımı Problemi için iki aşamalı Stokastik Karma Tamsayılı Programlama modeli geliştirilmiş, geri gönderilen ürün miktarı ve ürün değerleri değişken olarak alınarak, toplam kâr enbüyüklenmeye çalışılmıştır. Çalışmada toplam kâr, satış gelirlerinden yeniden üretim, onarım, demontaj, taşıma ve depolama mâliyetlerinin toplamının çıkarılması ile elde edilmektedir. Amaç fonksiyonuna ayrıca risk etmeni de dahil edilerek koşullu değer risk parametreleri ile model stokastik duruma getirilmiştir.

2.3 Bütünleşik Lojistik Ağ Tasarımına İlişkin Yapılan Çalışmalar

Bütünleşik Lojistik Ağ Tasarımında, İleri ve Tersine Lojistik süreçler aynı anda ele alınmaktadır. Problem üzerinde üretim, dağıtım, geri dönüştürme, toplama, onarım gibi tesislerin yerleri seçilmekte ve ağ içindeki akışlar belirlenmektedir. Bu başlık altında son yıllarda, Bütünleşik Lojistik için yapılan Ağ Tasarımı çalışmaları özetlenecektir.

Fleischmann ve diğerlerinin 2001 yılında yaptığı çalışmada gerçek yaşamda karşılaşılan Bütünleşik Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli geliştirilmiştir. Model tek ürün ve tek dönem için hazırlanmış, tesis kapasite kısıtları dikkate alınmamıştır. Çalışmada mâliyet enazlanması hedeflenmiş olup, üretim, dağıtım, toplama ve demontaj merkezi lokasyonları ve talep karşılama miktarları bulunmaya çalışılmaktadır. Modelde tesis açma, taşıma, üretim, yoketme, toplama ve müşteriye hizmet vermeme mâliyetleri gözönüne alınmış, çeşitli senaryolarla elektronik ve kağıt sektöründe uygulamalar yapılmıştır.

Salema, Pova ve Novais tarafından 2006 yılında yapılan bir çalışmada, Bütünleşik Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli geliştirilmiştir. Tek dönemli modelde ilk olarak tek ürün için tesis kapasite kısıtları dikkate alınmamış, daha sonra çok ürünlü ve kapasite kısıtlarının gözönüne alındığı başka bir model oluşturulmuştur. Modellerde mâliyet enküçüklenmesi amaçlanmış olup, iki model üzerinden karşılaştırmalar yapılmıştır. Hazırlanan çalışmada üretim, dağıtım, iyileştirme ve toplama merkezlerinin lokasyonu belirlenmekte ve bu lojistik ağdaki akışlar ve talep karşılama miktarları bulunmaya çalışılmaktadır.

Lu ve Bostel tarafından 2007 yılında yapılan bir çalışmada Bütünleşik Lojistik Ağ Tasarımı için ilk olarak Karma Tamsayılı Programlama modeli hazırlanmış, daha sonra Lagrange Yakınlaştırma tabanlı bir sezgisel geliştirilmiştir. Problem deterministik talepli tek ürün ve tek

dönem üzerinden yürütülmekte ve kapasite kısıtları dikkate alınmamaktadır. Mâliyet enazlanmasının hedeflendiği modelde, üretim, dağıtım, toplama ve yeniden üretim tesislerinin yerleri belirlenmekte ve lojistik ağ içindeki akışlar hesaplanmaktadır. Çalışmada yoketme tesislerinin yer seçimi yapılmamış, lokasyonu sabit kabul edilmiştir.

Ko ve Evans tarafından 2007 yılında yapılan bir çalışmada üçüncü parti lojistik firmalarında karşılaşılan Bütünleşik Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli ve Genetik Algoritma tabanlı bir sezgisel yöntem geliştirilmiştir. Üzerinde çalışılan problem, çok dönemli, çok ürünlü, kapasite kısıtlı ve deterministik taleplidir. Tesis açma, işletme, taşıma ve tesis kapasitesi genişletme mâliyetinin enazlanmasının hedeflendiği modelde, üretim ve iyileştirme merkezleri baz alınarak, dağıtım merkezlerinin yerleri ve dağıtım ağı içindeki akışlar belirlenmektedir. Genetik Algoritmada kromozomlar için Matris gösterimi, seçim için İkili Turnuva yöntemi ve çaprazlama için İki Noktalı Çaprazlama yöntemi kullanılmış olup, GA süreci salt tesis lokasyonlarını belirlemektedir. Lojistik ağ içindeki akışlar ise Simpleks yöntemi ile hesaplanmakta ve iterasyonlar süresince her çözüm için uygunluk değerinin bulunması amacıyla kullanılmaktadır.

Salema, Pova ve Novais tarafından 2007 yılında yapılan bir çalışmada Stokastik talepli Bütünleşik Lojistik Ağ Tasarımı problemi için Stokastik Karma Tamsayılı Programlama modeli geliştirilmiştir. Çalışma daha önce Fleischmann ve diğerleri tarafından 2001 yılında yapılan çalışmanın eksikliklerini gidermek amacıyla yapılmıştır. Bu eksiklikler; üretim, depo kapasiteleri, çok ürünlü süreç ve müşteri taleplerinde ve geri dönüşlerde belirsizliktir. Model tek dönem için, belirsiz talepli çok ürün üzerinden kurgulanmakta olup, tesis kapasite kısıtları dikkate alınmış ve mâliyet enazlanması hedeflenmiştir. Problem sonucunda üretim, dağıtım, iyileştirme ve toplama merkezi lokasyonları, ağ içindeki taşıma miktarları ve talep karşılama düzeyleri bulunmaya çalışılmaktadır. Yazarlar, bir ofis dokümanı şirketinde tesis açma mâliyeti, talepleri ve geri dönüşleri karşılama ve karşılamama mâliyetlerini düşürmeye çalışmışlar ve başarılı sonuçlar almışlardır.

Hui, Zhen ve Hu tarafından 2007 yılında Belirsizlik Altında Yarı Kapalı Döngülü Lojistik Ağ Tasarımı Problemi için bir Karma Tamsayılı Doğrusal Programlama modeli ve Genetik Algoritma yöntemi geliştirilmiştir. Kurulan modelde müşteri talepleri ve kullanılan ürünlerin geri dönüş oranları belirsiz olup belirli bir periyot için dinamiktir. Amaç fonksiyonunda sabit, taşıma ve depolama mâliyetleri enküçüklenmeye çalışılmıştır. Modelin yarı kapalı döngülü olması, üretim ile geri dönüşüm tesislerinin farklı yerlerde olması ile ilgilidir. Hazırlanan Genetik Algoritmada kodlama için İkili Kodlama, seçim için Rulet Tekerleği yöntemi ve

operatör olarak Tek Noktadan Çaprazlama ve Mutasyon tekniği kullanılmış ve başarılı sonuçlar alınmıştır.

Peng ve Zhong, 2008 yılında yaptıkları çalışmada tek periyotlu, çok ürünlü ve tesisli Lojistik Ağ Tasarımı için 0-1 Karma Tamsayılı Programlama modeli ortaya koymuşlardır. Amaç fonksiyonunda depolama, yeniden işleme, taşıma ve tesis işletme mâliyetleri gözönüne alınmıştır. Üretim, dağıtım, yeniden üretim ve demontaj tesisleri ele alınmış ve ağ içindeki akışlar hesaplanmaya çalışılmıştır.

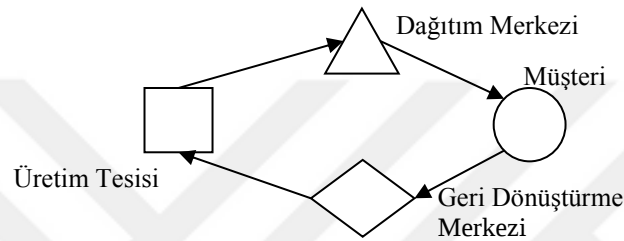
Min ve Ko tarafından 2008 yılında yapılan bir çalışmada, üçüncü parti lojistik firmalarında karşılaşılan Bütünleşik Lojistik Ağ Tasarımı problemi için ilk olarak Karma Tamsayılı Programlama modeli oluşturulmuş ve daha sonra Ko ve Evans tarafından 2007 yılındaki çalışmaya benzer Genetik Algoritma yöntemi geliştirilmiştir. Çok ürün ve çok dönem üzerinden hazırlanan modelde kapasite kısıtları dikkate alınmış olup, en az mâliyet hedefi ile toplama ve dağıtım merkezlerinin yerleri ve ağ içindeki akışlar belirlenmeye çalışılmaktadır. Geliştirilen Genetik Algoritmada kromozomlar için Matris kodlama, seçim için Rulet Tekerleği yöntemi ve çaprazlama için İki Noktalı Çaprazlama yöntemi kullanılmış olup, GA süreci ile lokasyonlar belirlenmekte, dağıtım ağı, ceza yöntemi kullanılarak farklı bir algoritma tarafından oluşturulmaktadır. Çalışma, büyük boyutlu bir problem için süperbilgisayar üzerinde denenmiş ve başarılı sonuçlar alınmıştır.

Lee ve Dong tarafından 2008 yılında bilgisayar sektöründe kiralama işlerinde karşılaşılan Bütünleşik Lojistik Ağ Tasarımı problemi için Karma Tamsayılı Programlama modeli ve Tabu Arama tabanlı bir sezgisel yöntem geliştirilmiştir. Deterministik talepli tek ürün ve tek dönem üzerinden hazırlanan modelde kapasite kısıtlarının yanında dağıtım ağındaki taşıma kapasitesi ve açılacak tesis sayısı da gözönüne alınmıştır. Algoritmada, üretim ve onarım tesisleri dikkate alınarak melez toplama ve dağıtım merkezlerinin yerleri belirlenmekte ve ağ içindeki akışlar hesaplanmaktadır. İki aşamalı sezgisel yöntem, ilk olarak Yer-Atama (LocationAllocation) problemini çözmekte, daha sonra dağıtım ağı içindeki akışları hesaplamaktadır. İlk aşamada lokasyonlar rassal olarak seçilip Simpleks yöntem ile dağıtım miktarları bulunurken, ikinci aşamada ise Tabu Arama ile çözümler iyileştirilmeye çalışılmaktadır.

Du, Wu ve Hu tarafından 2009 yılında Kapalı Döngülü Lojistik Ağ Tasarımı Probleminde yeniden üretim tesis yerleri, 0-1 Karma Tamsayılı Programlama modeli ile belirlenmiş olup, dağıtım problemi ise Genetik Algoritma ile çözülmüştür. Kapalı Döngülü bir lojistik sistemi

örneği Şekil 2.6’da gösterilmektedir. Bu sistemde amaç, mâliyetleri enazlamak için tesislerin sayılarını ve yerlerini belirlemek ve dağıtım rotalarını oluşturmaktır.

Zhao tarafından 2009 yılında yapılan bir çalışmada İleri ve Tersine Lojistikte Ağ Tasarımı Problemi için 0-1 Karma Tamsayılı Programlama modeli oluşturulmuştur. Kapalı döngülü bir lojistik sürecinde müşterilerden kullanılmış ürünler alınır ve yeniden üretim tesislerinde işlenerek yeniden müşterilere sunulur. Çalışmada tesis yerleri ve sayıları, lojistik akışlar belirlenerek toplam yatırım mâliyeti enküçüklenmeye çalışılmaktadır. Model, yeniden üretim süreci için çok ürünlü ve çok tesisli problem üzerinde denenmiş ve kapalı döngülü bir problem örneği için Lindo 8.0 kullanılarak optimum çözüm bulunmuştur.



Şekil 2.6 : Kapalı Döngülü Tedârik Zinciri (Du ve diğ, 2009).

Pishvae, Farahani ve Dullaert tarafından 2009 yılında yapılan bir çalışmada Çok Amaçlı İleri ve Tersine (Kapalı Döngülü) Lojistik Ağ Tasarımı Problemi için Karma Tamsayılı Programlama modeli ve Yerel Arama ile melezleştirilmiş Memetik Algoritma (Melez Genetik Algoritma) geliştirilmiştir. Oluşturulan modelde amaçlar üretim, dağıtım, toplama/inceleme, geri alma ve yoketme faaliyetlerini gözönüne alarak toplam mâliyeti enküçüklemek ve lojistik ağının etkinliğini enbüyüklemektir. Bu noktada etkinlik, müşteri taleplerinin hangi oranda karşılandığı ile ilgilidir. Çıktı olarak ise tesislerin lokasyonu, kapasiteleri ve taşıma miktarları algoritmadan alınmaktadır. Geliştirilen Genetik Algoritmada çözümler Matris Tabanlı Öncelik Gösterimi ile kodlanmış, Rassal Segment Tabanlı Çaprazlama yöntemi uygulanmış ve ayrıca seçim işlemi için Rulet Tekerleği yöntemi kullanılmıştır. Yerel arama sezgiseli olarak ise 3-Opt, 2-Opt ve Kapasite Ayarlama teknikleri rassal olarak sürece eklenmiştir.

Sayed, Afia ve Kharbotly tarafından 2009 yılında yapılan bir çalışmada belirsiz talep durumunda kapalı döngülü Lojistik Ağ Tasarımı Problemi için Stokastik Karma Tamsayılı Doğrusal Programlama Modeli geliştirilmiştir. Kurulan modelde tesislerin lokasyonları, her fabrikadaki üretim miktarları, taşıma miktarları ve depolarda her periyotta tutulacak ürün miktarları bulunmaya çalışılmaktadır. Amaç toplam kârı enbüyüklemektir. Sabit mâliyetler ile malzeme, üretim, kullanılmayan kapasite, elde bulundurmama, taşıma, elde bulundurma, geri

dönüşüm, yeniden üretim ve yoketme mâliyetleri gözönüne alınmıştır. Geliştirilen model Mosel dilinde kodlanmış ve başarılı sonuçlar elde edilmiştir.

2013 yılında Cardoso, Pova ve Relvas tarafından geliştirilen Karma Tamsayılı Programlama modelinde, belirsiz talepli Bütünleşik Lojistik Ağ Tasarımı problemi için üretim, dağıtım ve tersine lojistik süreçleri aynı anda kurgulanmıştır. Modelde talep belirsizliği bir senaryo ağacı yaklaşımı ile ele alınmış olup, amaç net kazancın enbüyüklenmesidir. Model de gerçek yaşamdan bir tedârik zinciri ağ problemine uygulanmış ve farklı senaryolar üzerinden etkin çözümler elde edilmiştir.

2.4 Lojistik Ağ Tasarımı İçin Kullanılan Yöntemler

Bu başlık altında Lojistik Ağ Tasarımı problemi için literatürde karşılaşılan yöntemler sıralanacak olup, bu tez çalışması kapsamında karşılaştırılması yapılacak olan Genetik Algoritma ve Tavlama Benzetimi yöntemleri ayrıntılı olarak açıklanacaktır.

Literatürde yapılan çalışmalar dikkate alındığında, Lojistik Ağ Tasarımı probleminde kullanılan yöntemler üçe ayrılabilir:

- Klasik Optimizasyon Yöntemleri (Doğrusal Programlama, Doğrusal Olmayan Programlama, Karma Tamsayılı Programlama modelleri vb.): Matematiksel olarak hedefi, kısıtları tam olarak tanımlanabilen modellerdir. Hedef ve kısıtlarda karar değişkenleri arasındaki ilişki doğrusal ve katsayılar tümüyle deterministik ise doğrusal programlama adı verilir. Karar değişkenleri arasındaki ilişkiler doğrusal olmayan bir şekilde tanımlanabiliyorsa doğrusal olmayan programlama, olasılıklı katsayılar kullanılıyorsa stokastik programlama adını alır. İkili ya da tamsayılı değer alan, karar değişkenlerinin yer aldığı doğrusal ya da doğrusal olmayan modeller karma tamsayılı modeller olarak tanımlanır (Winston ve Goldberg, 2004).
- Sezgisel Yöntemler (Lagrange Yakınlaştırma, Benders Ayrıştırma sezgiselleri): Karma Tamsayılı Programlama modellerinin tam optimizasyon değeri bulmakta zorlandığı karmaşık problemler için problemi parçalayarak çözme yöntemleridir. Benders Ayrıştırma yöntemi, tamsayılı doğrusal programlama modellerinin karşılaştığı zorlukların üstesinden gelmek için problemleri küçük parçalara ayırarak optimizasyon noktalarını bulur, Lagrange Yakınlaştırma ise doğrusal olmayan programlama modellerinin doğrusal hâle getirilmesini sağlar.

- Metasezgisel Yöntemler: İzleyen alt bölümlerde metasezgisel yöntemler açıklanmaktadır.

2.4.1 Genetik Algoritmalar

Genetik Algoritmalar (GA), en iyinin korunumu ve doğal seçim (seleksiyon) ilkesinin benzetim yolu ile bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir. Evrimsel programlamanın bir parçası olan GA, Darwin'in evrim kuramından esinlenerek ortaya çıkmıştır. GA'nın bugünkü biçimi ilk olarak Holland tarafından 1975 yılında belirtilmiş ve Goldberg'ın 1989'da yazdığı kitap ile kombinatorial problemlerdeki başarısı ortaya konulmuştur. GA, doğada canlıların değişen çevreye uyma yeteneğinden ilham alınarak geliştirilmiş evrimsel bir programlama tekniğidir. Her iterasyonda (kuşakta) seçilen bazı çözümlere, bir sonraki iterasyondaki yeni çözümleri oluşturmak için bazı genetik operatörler uygulanır. Doğal seçim kavramıyla oldukça ilişkili olarak GA'nın yönlendirici özelliği, çözümlerin uygunluk değerine, başka bir deyişle amaç fonksiyon değerine göre olasılıksal bir şekilde seçilmesine dayanır. Buna göre diğer çözümlere göre daha iyi amaç fonksiyon değerine sahip çözümlerin seçilme olasılığı daha fazladır (Blum ve Roli, 2003).

Genetik Algoritmalar, doğadaki evrimsel süreçlerini model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, GA'nın yardımıyla daha kolay ve hızlı olarak çözülebilmektedir. Genel anlamıyla GA bir araştırma konusu olup, model hâline getirilmiş neden-sonuç işleminin tersine, rassal örnekleme olgusu altında modellenmiştir. Kontrol edilerek onaylanan kod bilgisi, organizma olarak adlandırılan aday çözümler içerisinde saklanmıştır. Organizmalar ise popülasyon olarak adlandırılıp grup hâlinde yer almışlardır. Genetik Algoritmalar ile ilgili temel kavramlar aşağıda kısaca açıklanmaktadır (Eren, 2002):

Gen: Kalıtsal molekülde bulunan ve organizmanın karakterlerinin tayininde rol oynayan kalıtsal birimlere denir. Yapay sistemlerde gen, kendi başına anlamlı bilgi taşıyan en küçük birim olarak alınır.

Kromozom: Birden fazla genin belirli bir düzenle bir araya gelerek oluşturduğu diziye denir. Kromozomlar seçenek aday çözümleri gösterir.

Plân – Katar: Belirli pozisyonlarda, uygun olan alt dizileri (kromozomları) tanımlamak için kullanılan sayı katarlarına veya plânlarına denir.

Popülasyon: Kromozomlardan oluşan topluluğa denir. Popülasyon, geçerli seçenek çözüm kümesidir. Popülasyondaki birey (kromozom) sayısı genelde sabit tutulur. GA’da popülasyondaki birey sayısı ile ilgili genel bir kural yoktur. Popülasyondaki kromozom sayısı arttıkça çözüme ulaşma süresi (iterasyon sayısı) azalır.

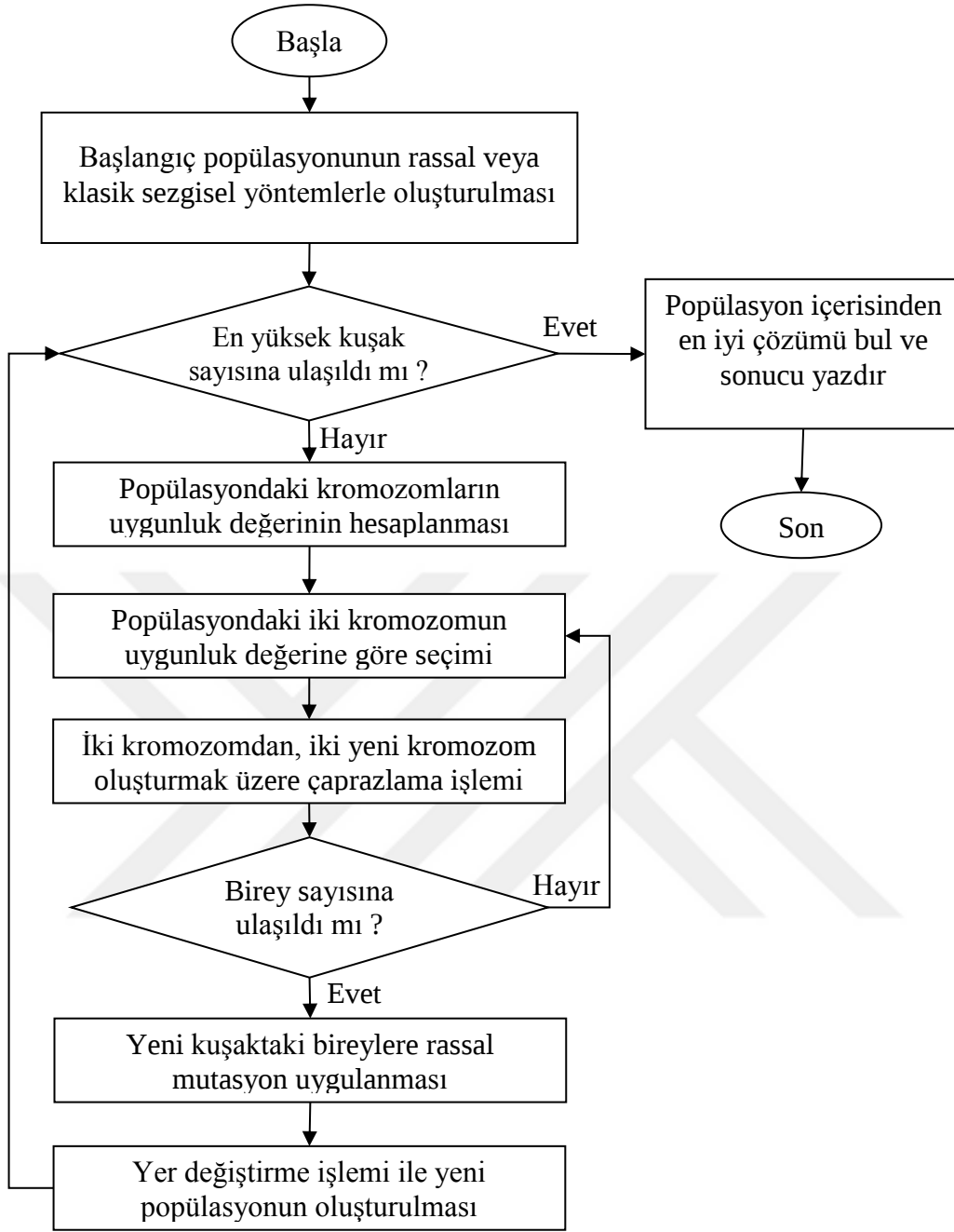
Standart bir GA yordamı aşağıdaki gibi verilebilir (Nabiyev, 2003):

1. Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur (Çözüm grubu biyolojideki benzerliği nedeniyle popülasyon, çözümlerin kodları da kromozom olarak adlandırılır). Popülasyonda bulunacak birey sayısı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde, yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Popülasyon bu işlemten sonra rassal veya klasik sezgisel yöntemlerle oluşturulur. Birey sayısı belirlendikten sonra probleme bağlı olarak kromozomların kodlanması gerekmektedir.
2. Popülasyondaki her kromozomun ne kadar iyi olduğu bulunur. Bu amaçla kullanılan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon yardımı ile kromozomların uygunluklarının bulunmasına ise evrimleşme adı verilir. Uygunluk fonksiyonu GA’nın beynini oluşturmaktadır. GA’da probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri hâline getirerek onların bir bakıma şifresini çözmekte ve sonra bu parametrelere göre hesaplamayı yaparak kromozomların uygunluğunu bulmaktadır. Çoğu zaman GA’nın başarısı, bu fonksiyonun verimli ve hassas olmasına bağlıdır.
3. Popülasyonda bulunan çözümler, uygunluk değerlerine göre olasılıksal olarak seçilip eşlenerek çaprazlama ve mutasyon operatörleri uygulanır. Sonuçta yeni bir toplum oluşturulur. Bu seçimi yapmak için rulet tekerleği seçimi, turnuva seçimi gibi seçme yöntemleri vardır. Yeniden kopyalama, genlerdeki genetik bilginin birinden diğerine geçme işlemine benzediği için çaprazlama olarak adlandırılır. Bu işlem toplumda çeşitliliği sağlar. Mutasyon ise salt bir çözüm üzerinde uygulanarak yerel optimum noktalardan kurtulmak için yapılmaktadır.
4. Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.
5. Tüm kromozomların uygunlukları yeniden hesaplanır ve yeni toplumun başarısı bulunur.
6. İşlemler yinelenerek, verilen kuşak sayısı süresince daha iyi olan yeni kuşakların oluşturulması gerçekleştirilir.

7. Sonuçta toplumların hesaplanması sırasında en iyi bireyler bulunduğunda çözüm elde edilmiş olur.

GA, çözüm grupları ile çalışarak, daha optimal, yapılabilir sonuçlara ulaşmada gerekli olan araştırmalara liderlik eder. Kromozom, genellikle problemdeki değişkenlerin belirli bir düzende sıralanmasıdır. Rassal oluşturulan başlangıç topluluğu bir kez oluşturulduktan sonra seçim, çaprazlama ve mutasyondan oluşan evrim süreci başlar. Şekil 2.7’de bu süreç gösterilmektedir. GA’nın problem çözmedeki başarısı, pek çok deneysel çalışma ile gösterilmiştir. Yöntemin çalıştığını kanıtlamak için birçok çalışma yapılmıştır. Bu çalışmaların büyük bir bölümü özel durumları içerir. GA uygulamalarının çeşitliliği ve probleme özel kullanılan yöntemler bu incelemeyi daha da zorlaştırmaktadır. GA’nın en uygun olduğu problemler, geleneksel yöntemler ile çözümü olanaklı olmayan ya da çözüm süresi problemin büyüklüğü ile üstel orantılı olarak artanlardır.

GA popülasyon tabanlı algoritma olduğundan, özellikle gerçek yaşamda çözüm üretilmesinin gerekli olduğu problem uygulamalarında önemli bir olumsuz yanı vardır. Bir algoritmanın belirli bir süre içerisinde çözüm geliştirme işlemini tamamlamış olması gerekir. Bu işlemi tamamlamak için popülasyon tabanlı algoritmalar, yerel arama algoritmaları ile karşılaştırıldığında görece daha uzun süreye gereksinim duyarlar. Süreyi kısaltmak için az popülasyonla çalışabilen ve iyi performansa sahip GA’nın geliştirilmesi önemli duruma gelmiştir. Bunun yanında GA’nın bölgesel yakınsama hızının çok iyi olmaması bir diğer sakıncalı yanıdır.



Şekil 2.7 : Genetik Algoritma akış süreci.

2.4.2 Tavlama Benzetimi Algoritması

Tavlama, metal malzemelerde katı hâlde sıcaklık değişimleri ile bir ya da birbirine bağlı birkaç işlemle amaca uygun özellik değişimlerinin sağlanması olayı şeklinde tanımlanır. Bu tanım gereği malzemenin belirli bir sıcaklığa kadar ısıtılması, bu sıcaklıkta uygun bir süre tutulması ve belirli bir stratejiye göre sıcaklığın oda sıcaklığına kadar azaltılması sayesinde üç aşamada malzemede özellik değişimleri sağlanır. Metal malzemelerde ısıl işlemle özellik

değişimleri sağlanırken, malzemenin kimyasal bileşiminde değişiklik yapılmadan kristal ya da kafes yapısında düzenlemeler yapılabilir (Karaboğa, 2004).

Tavlama Benzetimi (TB), fiziksel tavlama süreci ile kombinatorial optimizasyon problemlerinin benzeşiminden ortaya çıkarılmış bir yöntemdir. Fiziksel tavlama prosesi, ısı banyosu içerisindeki katı bir cismin düşük enerjilerini elde etmek için uygulanmaktadır. Eğer katı bir cisim erime noktasına kadar ısıtılır ve sonra hızla soğutulmaya başlanırsa, katı cismin moleküler yapısı (kristal yapı) soğutma oranına bağlı olarak değişir. TB, ilk olarak 1953 yılında Metropolis, Rosenbluth ve Teller tarafından bir ısı banyosu içindeki taneler kümesinin denge dağılımını hesaplamak amacıyla uygulanmıştır. Metropolis simülasyon yardımı ile enerjideki değişimi hesaplamıştır. Metropolis algoritmasının temel ilkesi, soğutma sürecindeki enerji değişimine göre belirlenir. Termodinamik kanuna göre, t. durumdaki sıcaklığa bağlı olarak enerjideki azalma olasılığı, denklem 2.1 ile elde edilir (Eren, 2002):

$$P(\delta E) = \exp(\delta E/k.t) \quad (2.1)$$

2.1 nolu formüldeki k fiziksel sabit olup, Boltzman sabiti olarak da bilinir. Eğer enerji azalır ise sistem yeni bir duruma geçer, enerji artar ise yeni durum kabul edilir. Bu işlemler donma noktasına kadar sürer.

Tavlama Benzetimi (TB) orijini doğa tavlama işleminden almaktadır. Bu algoritmanın optimizasyon problemlerine uygulanması ile ilgili çalışmalar 1983 yılında Kirkpatrick ve arkadaşları tarafından yapılan bir çalışma ile başlamıştır. Algoritma, metallerin tavlama işlemi ile bir optimizasyon problemine çözüm araştırma olayları arasındaki benzerlikten ilham alınarak ortaya konulmuştur. Burada metaldeki atomların durumları, optimizasyon probleminin olası çözümlerine ve durumların enerjileri de çözümlere ait amaç fonksiyon değerlerine karşılık gelmektedir. Hızlı soğutma işlemi ise bölgesel optimizasyon işlemine karşılık gelmektedir. Dış sıcaklık sıfır olduğunda daha yüksek enerjili bir duruma geçiş olanaklı olmaz. Böylece bölgesel optimizasyondaki gibi yukarı doğru hareketler yasaklanır ve araştırma bir bölgesel en aza takılı kalır. Bu işlemde sıcaklık (T) çeşitli düzeyler boyunca yavaşça düşürülür. Enerji düzeyinden uzaklaşmamayı sağlamak için varolan sıcaklığı belirli bir süre korumak sùretiyle ve sıfır dereceye yaklaşıncaya kadar bu işlemlerin yinelenmesi, bölgesel optimallikten kaçışı sağlayabilmektedir (Karaboğa, 2004).

TB’de çözüm uzayı taranırken yukarı doğru hareketler, yâni amaç değerinin aleyhine hareketler, değişen bir olasılık tabanlı fonksiyon ile kontrol edilir. TB genel olarak yinelemeli

bir geliştirme algoritmasıdır. Standart bir TB algoritmasının temel adımları aşağıda verilmektedir (Breedam, 2001):

Adım 1. Rassal olarak bir başlangıç çözümü üret ve en iyi çözüm S olarak ata. Ayrıca t iterasyon indeksini 0 olarak ata.

Adım 2. Bir başlangıç sıcaklık değeri T_B belirle ve varolan sıcaklık değeri $T_0 = T_B$ olarak ata.

Adım 3. En iyi çözümden hareketle rassal komşu bir çözüm $S^1 \in N(S)$ oluştur.

Adım 4. Üretilen S^1 çözümüyle S çözümünün amaç fonksiyonu değerleri arasındaki farkı hesapla ($\delta = C(S^1) - C(S)$)

Adım 5. Eğer S^1 , S'den daha iyi ($\delta < 0$) ise S çözümüne S^1 çözümünü ata. S^1 , S'den daha kötü ama varolan T_t sıcaklığında ($e^{\frac{-\delta}{T_t}} > \theta$) sağlanıyorsa (θ , 0 ile 1 arasında rassal üretilmiş bir sayıdır) S çözümü ile S^1 çözümünü yer değiştir. Yoksa S'i varolan çözüm olarak koru.

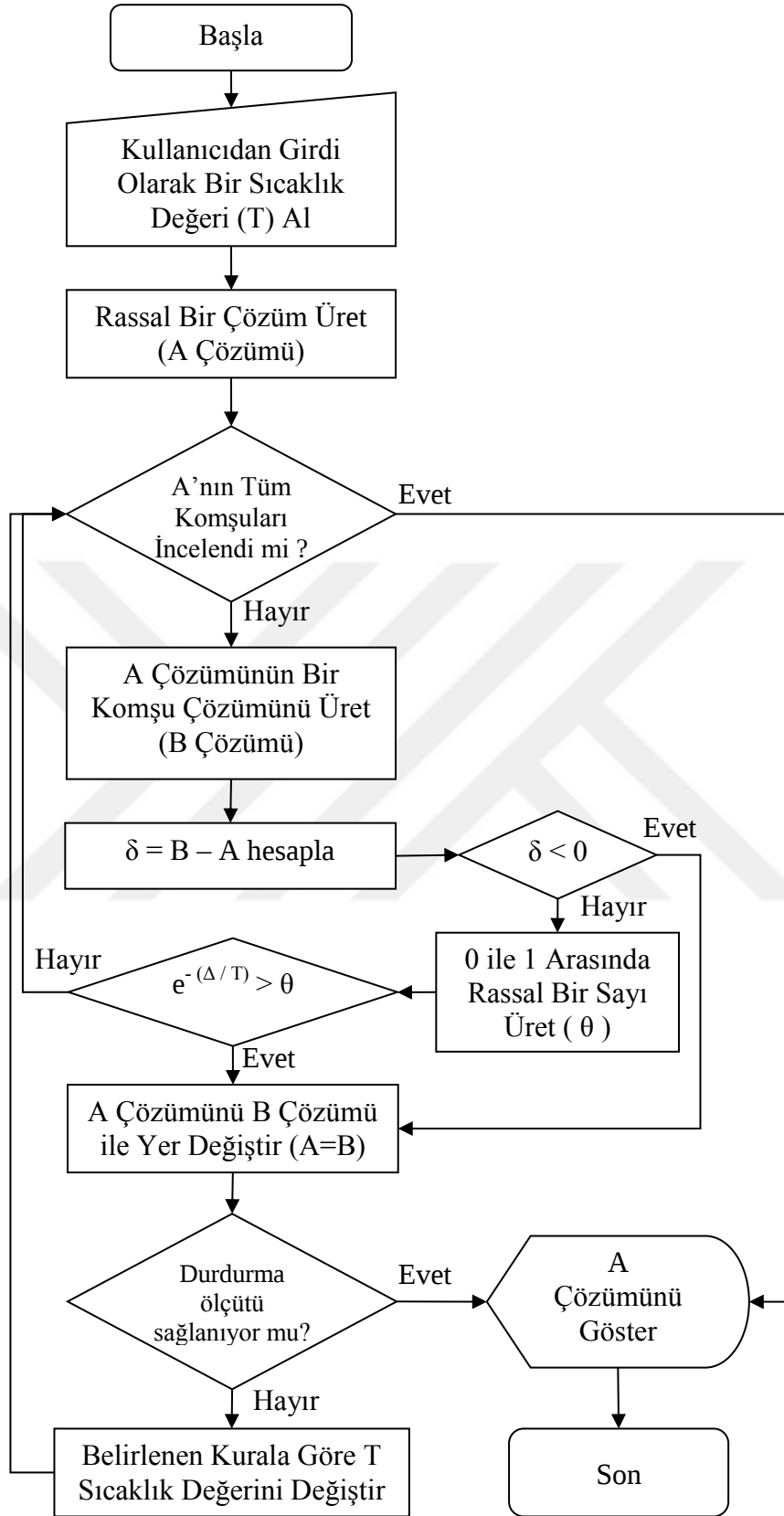
Adım 6. T sıcaklığını denklem 2.2 veya 2.3'teki formüle göre değiştir. 2.2 nolu formülde R sıfırla bir arasında rassal bir değer, 2.3 nolu formülde ise β uygun küçük bir değerdir.

$$T_t = R \cdot T_{t-1} \quad (2.2)$$

$$T_t = t / (1 + \beta t) \quad (2.3)$$

Adım 7. Durdurma ölçütü sağlanıyorsa araştırmayı durdur, aksi durumda yineleme indeksi t'yi 1 arttırarak üçüncü adıma git.

TB, bölgesel en düşüklerden uzaklaşmak için rassal çıkış hareketleri kullanan, değiştirilmiş bir yinelemeli iniş algoritmasıdır. Yukarı doğru hareketlerin kabul düzeyi, mâliyet fonksiyonundaki artışın genliğine ve kontrol parametresi olarak adlandırılan sıcaklık T'ye bağlıdır. Algoritmanın hesaplama süresi, yukarıda verilen genel adımlara değişik düşünceler eklemek sùretiyle hızlandırılabilir. Örneğin, bunlar, amaç fonksiyonundaki değişimin yaklaşık olarak hesaplanması, seçenek yöntemlerin kullanılması, araştırma yapmak için iyi çözümlerin bulunduğu umut verici araştırma bölgelerinin tanımlanması ve komşuluk büyüklüğünün belirlenmesi gibi olabilir (Karaboğa, 2004). Şekil 2.8'de standart bir TB algoritmasının akış şeması gösterilmektedir.



Şekil 2.8 : Tavlama Benzetimi akış şeması.

2.4.3 Tabu Arama Algoritması

Glover tarafından 1989’da geliştirilen Tabu Arama (TA) yönteminde yerel optimum noktalara takılmaktan kurtulmak için daha önce arama sırasında elde edilen bilgiler kullanılmaktadır. Kısa dönemli bellek denilen bu yapıda, arama sırasında henüz ziyaret edilen çözümler bir tabu listesinde tutulmakta ve algoritma süresince bu çözümlerin yeniden ele alınması engellenmektedir. TA’da TB’den farklı olarak bir iterasyonda bir çözümün tabu listesinde yer almayan tüm komşu çözümleri taranmaktadır ve en iyi komşu çözüm seçilip başlangıçtaki çözüm tabu listesine eklenerek diğer iterasyona geçilmektedir. Tabu listesine çözüm ekleme ve çıkarma işlemleri ise ilk giren son çıkar (FIFO) ilkesi ile yapılmaktadır. Bu nedenle TA yöntemi, literatürde “dinamik komşuluk arama tekniği” olarak da telaffuz edilmektedir. TA algoritması herhangi bir durdurma ölçütü sağlandığında veya bir S çözümünün tüm komşu çözümleri $N(S)$, tabu listesinde yer alıyorsa durdurulmaktadır (Blum ve Roli, 2003).

TA algoritmasının bölgesel optimalite aşmak amacıyla kullandığı temel ilke, değerlendirme fonksiyonu denilen bir fonksiyon tarafından her iterasyonda en yüksek değerlendirme değerine sahip hareketin bir sonraki çözümü oluşturmak amacıyla seçilmesine dayanmaktadır. TA’nın altında yatan temel düşünce, araştırma işlemini kontrol etmek amacıyla esnek bir bellek yapısının kullanılmasıdır. TA’nın işlem basamakları aşağıda verilmektedir (Karaboğa, 2004):

Adım 1. Bir başlangıç çözümü (S) al. Başlangıçta değer atanması gereken parametreler için (tabu listesi uzunluğu, durdurma kriteri, vs) değerlerini ata.

Adım 2. Belirlenen bir komşuluk yapısı ile S’ye ait komşu çözümler üret ve bu çözümler arasından tabu listesinde olmayan tüm $S^1 \in N(S)$ en iyi kabul edilebilir olan (S_{eni}) seç.

Adım 3. Varolan çözümü (S), S_{eni} ile yer değiştir ve tabu listesini yenile.

Adım 4. Durdurma ölçütü sağlanıncaya kadar Adım 2 ve 3’ü yinele.

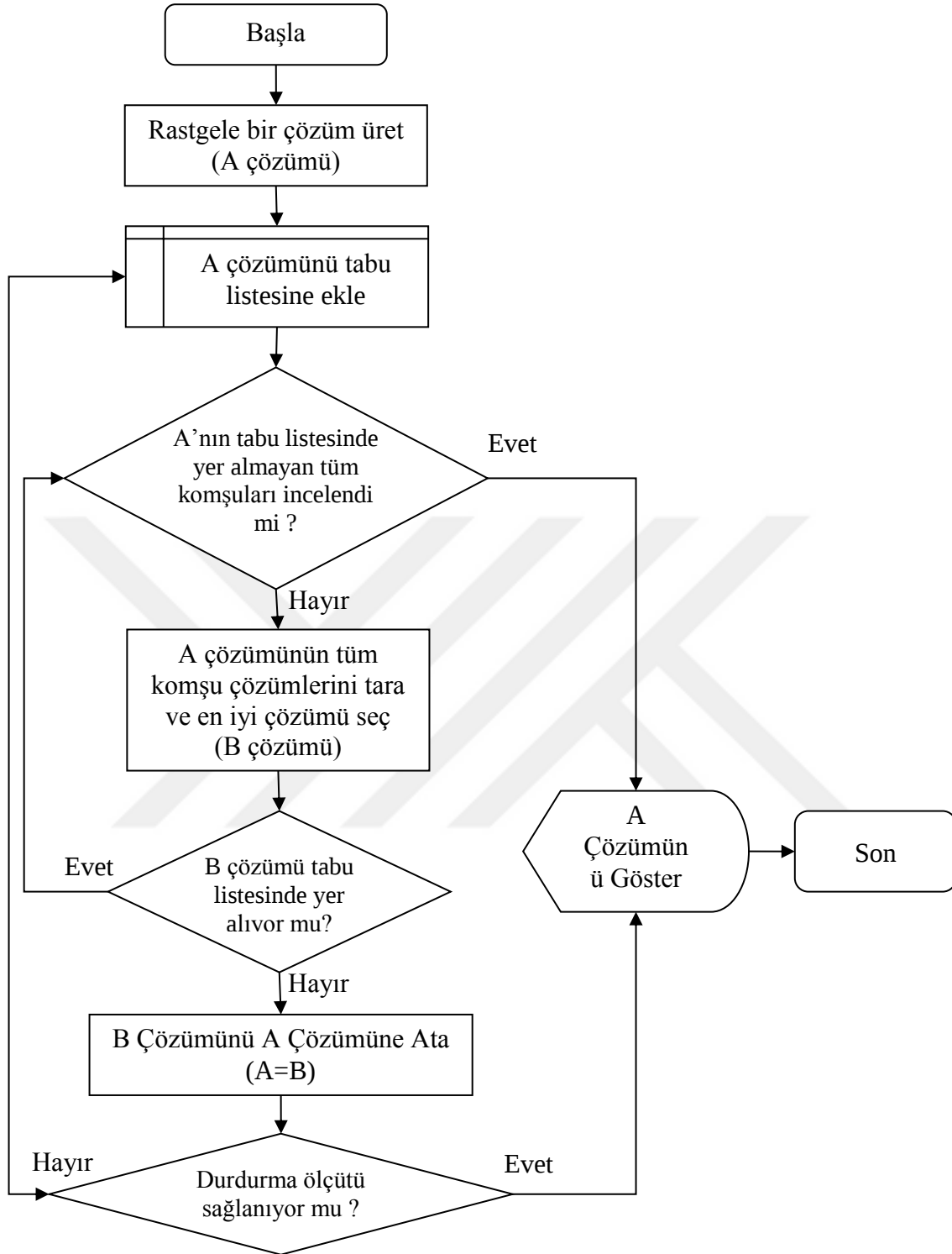
Bu araştırma işlemine göre en iyi kabul edilebilir hareket, varolan çözümün komşuluğunda bulunan amaç değeri ve tabu sınırlamaları açısından en yüksek değerlendirme değerine sahip olan harekettir. Değerlendirme fonksiyonu, amaç fonksiyonunda en az kötüleşme veya en çok iyileşme sağlayan hareketi seçer. Tabu listesinde kabul edilen hareketlerin karakteristikleri depolanmakta ve bu karakteristikler sonraki iterasyonlarda tabu olacak belirli hareketleri sınıflandırmak için kullanılmaktadır. Kötüleşme sağlayan hareketlerin kabul edilebilir olması nedeniyle daha önceden denenmiş olan, diğer bir deyişle ziyaret edilmiş çözümlere yeniden dönmek olanaklı olabilmekte ve böylece bir döngü oluşabilmektedir. Tabu listesini

kullanmanın amacı böyle bir olayın oluşmasını önlemektir. Bu nedenle, görevi salt tabu listesini kontrol etmek ve yenilemek olan bir yasaklama stratejisini kullanarak araştırmayı sınırlamak zorunludur. Yasaklama stratejisinin amacı, önceden izlenmiş araştırma yollarının yinelenmemesini sağlamak ve böylece yeni bölgelerin araştırılmasını olanaklı kılmaktır. Şekil 2.9’da standart bir TA algoritmasının akış şeması gösterilmektedir (Erol, 2006).

TA, yerel optimallığın tıkanıklığından kaçınmak ve diğer yöntemlere yol göstermek için geliştirilmiş, optimuma yakın çözüm veren sezgisel bir arama yöntemidir. Tabu araştırmaları hedefe, kısıt ve tabu kullanarak belirli bir olasılıkla ulaşır. Hedefe ulaşma olasılığının yükselmesi için kısıt ve tabu sayılarını arttırmak gerekir (Eren, 2002).

TA algoritması, katı ve özel bir algoritma olmaktan ziyade genel bir yaklaşımdır. TA algoritması genellikle bir olası başlangıç çözümü ile araştırmaya başlar ve bu çözümü daha iyi çözüme dönüştürmek için ardışık hareketler gerçekleştirir. Araştırma sırasında aşağıdaki stratejileri kullanmak sûretiyle bölgesel optimallikten kurtulmaya ve diğer bölgeleri araştırmaya çalışır (Karaboğa, 2004):

- Tabu listesine neyin girip neyin girmeyeceğine karar veren yâni kontrol eden yasaklama stratejisi.
- Tabu listesinden neyin ne zaman çıkacağını kontrol eden serbest bırakma stratejisi.
- Denenecek hareketlerin seçimi için yukarıdaki stratejiler arasında ilişkiyi kontrol eden kısa dönem stratejisi.



Şekil 2.9 : Tabu Arama algoritması akış şeması.

2.4.4 Karınca Kolonisi Optimizasyon Algoritması

Doğadaki bazı sosyal sistemler, sınırlı yetenekli basit bireyler tarafından oluşturulmalarına rağmen, toplu zekâ davranışı sergilerler. Problemler için üretilen zekî çözümler, bu bireylerin kendi içerisindeki organizasyonlarından ve dolaylı iletişimlerinden ortaya çıkar. Karıncalar

tek başlarına basit yeteneklere sahip olmalarına rağmen, koloninin bütünü yüksek bir yapıdadır; kendilerinden çok büyük cisimleri taşımak, köprüler oluşturmak veya yuva ile yiyecek arasındaki en kısa yolu bulmak için çözüm üretirler. Zekî davranış doğal olarak karıncalar arasındaki organizasyon ve dolaylı iletişim sonucunda ortaya çıkar (Nabiyev, 2003).

Karıncı Kolonisi Optimizasyon (KKO) algoritması Dorigo tarafından 1996 yılında karıncaların yiyecek bulma mekanizmalarından ilham alınarak geliştirilen bir metasezgisel yöntemdir. KKO tekniği genel olarak parametrelendirilmiş bir olasılıksal model olan feramon modeline dayanır. Çoğu uygulamada, karıncalar uygun bir çözüm oluşturmak için kullanılsa da, olasılıksal modelin bir sonucu olarak uygun olmayan çözümler de üretilebilmektedir (Blum ve Roli, 2003).

KKO, karınca kolonilerinin yiyecek toplama ilkesine göre çalışır. Doğadaki karıncalar kör olduklarından, koloniler hâlinde yiyecek toplamadaki en kısa yolu seçme mekanizmalarına göre algoritma oluşturulur. Her bir karınca, geçiş olasılıklarına bağlı olarak hareket edeceği yönü belirler. Karınca i. düğümden j. düğüme hareket ettiğinde, j. düğüm karıncanın belleğinde tabu listesine kaydedilir. Bir sonraki adımda karınca tabu listesinde bulunan bu yöne doğru hareket etmez (Eren, 2002).

Yapay karıncalar ile doğal karıncalar arasındaki benzerlikler aşağıda belirtilmektedir (Dorigo ve diğ, 1999):

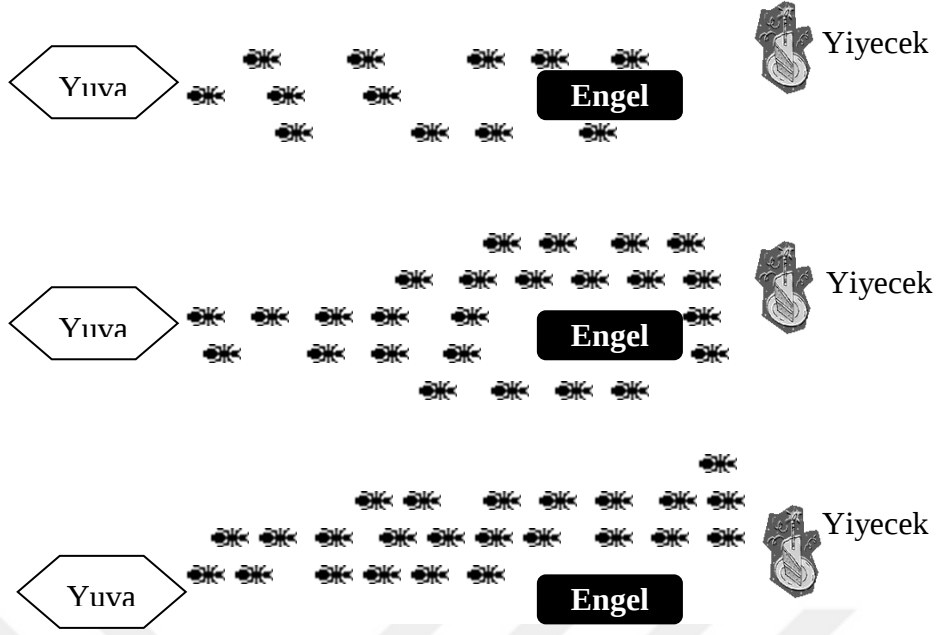
- Yapay karıncalar, gerçek karıncalar gibi belirli bir hedefe yönelik işbirliği içinde çalışan bir koloni oluşturmaktadır. Bu işbirliği sayesinde yüksek kaliteli çözümler üretilebilmektedir.
- Yapay karıncalar geçtikleri yollara, hem kendi yönlerini bulmada hem de arkadan gelen karıncalar için kılavuzluk yapması açısından feramon maddesi bırakırlar. Feramon maddesi bir bakıma koloni içinde iletişimi sağlamaktadır ve aynı gerçek karıncalar için olduğu gibi zamanla buharlaşır.
- Gerçek ve yapay karıncaların tek ortak amacı, yuvaları ve yiyecekleri arasında en kısa, başka bir deyişle en az mâliyetli olan yolları bulmaktır. Gerçek karıncalar rotalarını engeller arasından geçerek, yapay karıncalar ise düğüm noktalarını seçerek belirlerler.

- Yapay karıncalar, gerçek karıncalar gibi gidecekleri yolları, yollarda bulunan feramon miktarına göre olasılıksal olarak seçerek çözüm oluştururlar. Bu nedenle seçim tümüyle belirli bir zamandaki yerel verilere dayanmaktadır.

KKO algoritması, gerçek karınca kolonisi davranışının matematiksel modellerine dayalı bir algoritmadır. Bu davranışların tam olarak modellenmesi yerine, yapay karınca kolonilerinin bir optimizasyon aracı olarak değerlendirilmesinden dolayı önerilen algoritmalar gerçek karınca davranışından biraz farklı yapıdadır (Karaboğa, 2004). Yapay karıncalar ile doğal karıncalar arasındaki farklar aşağıda açıklanmaktadır:

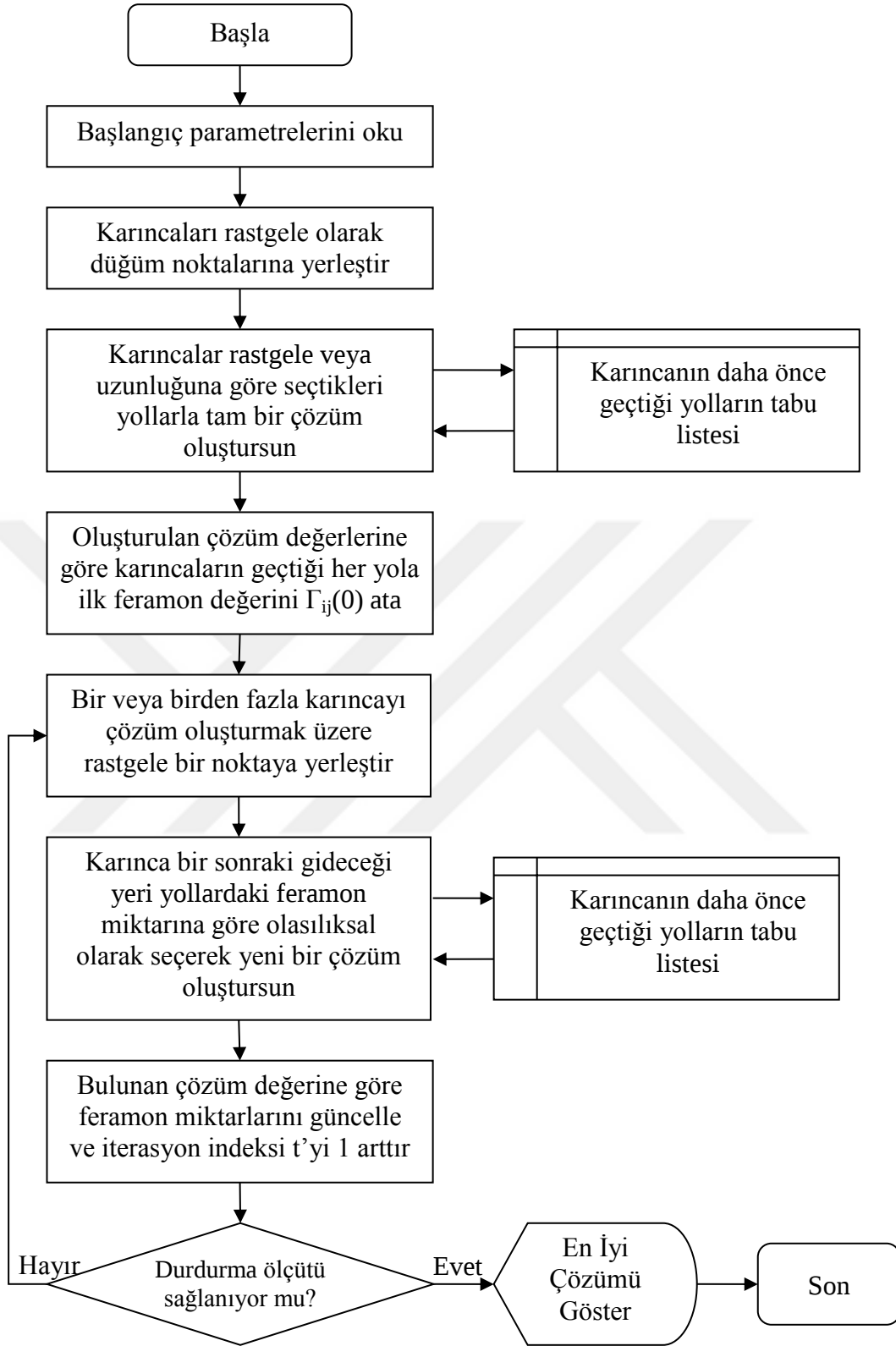
- Yapay karıncalar, ayrık bir dünyada yaşarlar ve onların hareketleri durumdan duruma değişim gösterir.
- Yapay karıncaların, daha önce yapılan hareket bilgilerinin tutulduğu bir iç duruma (belleğe) sahiptir.
- Yapay karıncalar, bulunan çözümün kalitesi ile orantılı miktarda feramon salgırlar.
- Yapay karıncaların feramon salgılaması, genellikle problem için bir çözüm üretildikten sonra yapılmaktadır ve bu da gerçek karıncaların davranışı ile farklılık göstermektedir.
- KKO algoritmasının performansını artırmak için algoritmaya ek olarak görünürlük derecesi, yerel optimizasyon, geri besleme gibi özellikler ilave edilebilmekte ve çoğu KKO uygulaması yerel arama teknikleri ile melezleştirilmektedir. Oysa bu özellikler gerçek karıncalarda bulunmamaktadır.

Karıncalar yuvaları ile yiyecek arasındaki en kısa yolu şu şekilde bulmaktadır. Her karınca yiyecek ararken geçtiği yerlere (rotasına) değişik miktar ve yoğunluklarda karın bölgesinde yer alan feramon adlı özel bir sıvı bırakır. Bu koku karıncaya yuvasına dönmesini sağladığı gibi diğer karıncalara da yiyeceğe giden yolu göstermeye yarayan bir izdir. Karıncalar ancak feramon kokularını izleyerek doğru yolu bulabilmektedir. Ancak hiç bir karınca tek başına yiyecek ile yuva arasındaki en kısa yolu bulabilme yeteneğine sahip olmamaktadır. Bu süreci başarabilecek kavrama yetenekleri veya değişik yolları değerlendirebilecek bellekleri yoktur. Ancak koloni davranışı, toplu zekâ sonucunda bu optimum çözümü düzenleyebilmektedir. Şekil 2.10'da yuvalarından çıkan karıncaların önlerine çıkan bir engelle karşı en kısa yolu nasıl buldukları gösterilmektedir.



Şekil 2.10 : Karıncaların en kısa yolu bulma yöntemi.

Karıncalar ilk olarak engelle karşılaştıklarında rastgele olarak eşit olasılıklarla yollarını seçerler. Bu durumda karıncaların yarısının sağa yarısının sola döneceğini bekleyebiliriz. Alt taraftaki yolun daha uzun olmasından ve feramonun uçuculuğundan dolayı daha fazla karınca aynı sürede üst yolu izlemiş olacaktır. Bundan dolayı bu üst koldaki feramon yoğunluğu daha da artacaktır. Belirli bir süre sonra bu kollar arasındaki fark daha da artacak ve sisteme yeni karıncalar dahil olmaya başlayacaktır. Sisteme dahil olan karıncalar feramon yoğunluğunun daha fazla olduğu üst yolu yeğleyecektir. Böylece karıncalar yiyeceğe giden en kısa yolu bulmuş olacaklardır. Her karıncanın ortalama aynı hızda ve aynı miktarda feramon bıraktığı düşünülürse, başlangıçta karıncaların en kısa yolu seçmeleri daha uzun zaman gerektirecektir. Ancak sonradan feramon yoğunluğunun artmasıyla yiyeceğe götüren süre (problemin çözümü) kısılacaktır. Standart bir KKO akış süreci Şekil 2.11’de gösterilmektedir. (Nabiyev, 2003).



Şekil 2.11 : Karınca Kolonisi Optimizasyonu algoritması akış şeması.

2.5 Lojistik Ağ Tasarımına İlişkin Çalışmaların Özeti

Lojistik Ağ Tasarımı; işletmenin depo, dağıtım merkezi gibi noktalarının, tedârikçi, üretim noktaları ve bayi, müşteri, toptancı gibi satış noktaları ile ilişkilendirilmesidir. Literatür araştırmaları genel olarak değerlendirildiğinde LAT Problemlerinin sınıflandırılması Çizelge 2.3'teki gibi yapılabilir (Pishvae ve diğ, 2009). Çizelge 2.3'e göre bir Lojistik Ağ Tasarımı Probleminin tanımlanması, ilk olarak problem tanımı, modelleme türü, amaçlar, çıktılar ve lojistik ağ aşamalarına göre yapılabilir.

Son yıllarda Lojistik Ağ Tasarımı optimizasyonu ile ilgili yapılan yayınlarda, mâliyet enküçüklemesi amaçlanmakta ve çözüm yöntemi olarak daha çok Metasezgisel teknikler kullanılmaktadır. Genellikle yöntemlerin çözüm kalitesini arttırmak ve hesaplama süresini azaltmak için basit sezgisel tekniklerle yerel arama süreci melezleştirilmiştir. Aynı zamanda hazırlanan yöntemin bulduğu çözümleri karşılaştırmak için Doğrusal veya Doğrusal Olmayan Programlama Modelleri (Kesin Yöntemler) geliştirilmiş ve probleme uygulanmıştır. Son zamanlarda Tersine Lojistik konusunda yapılan çalışmalar literatürde oldukça fazla yer almakta, İleri ve Tersine Lojistik (Bütünleşik Lojistik) ise yeni bir konu olarak karşımıza çıkmaktadır.

Şimdiye kadar yapılan çalışmalar değerlendirildiğinde Lojistik Ağ Tasarımı Problemlerinin dört boyutta oldukları görülmektedir:

- Karar Düzeyi: Stratejik, Taktik ve Operasyonel
- İşlevler: Lokasyon belirleme, araç rotalama, depo miktarı belirleme, araç sayısı hesaplama vb.
- Çözüm Yöntemleri: Sürekli yakınlaştırma, kesin ve sezgisel yöntemler
- Uygulanan Alanlar (sektörler): Otomotiv, lojistik, elektronik vb.

İncelenen akademik çalışmalara göre çözüm yöntemlerinin, Lojistik Ağ Tasarımı Problemlerinde tüm sürece değil, depolama, tesis yerleşimi veya araç rotalama gibi belirli bir bölümüne uygulandığı ve gerçek yaşama ilişkin problemler için yapılan uygulamaların oldukça az olduğu görülmektedir.

Çizelge 2.3 : Lojistik Ağ Tasarımı Problemi sınıfları (Pishvae ve diğ, 2009).

Ana Sınıflandırma Türü	Alt Sınıflandırma Türü	Sınıflar
Problem Tanımı	Dönem	Çok Dönemli, Tek Dönemli
	Açılacak Tesis Sayısı	Belirsiz, Belirli
	Ürün	Tek Ürünlü, Çok Ürünlü
	Akış Kapasitesi	Akış Kapasite Kısıtsız, Akış Kapasite Kısıtlı
	Talep	Stokastik, Deterministik
	Tesis Kapasitesi	Tesis Kapasitesi Kısıtsız, Tesis Kapasitesi Kısıtlı
Modelleme	Sürekli	Sürekli Model
	Kesikli	Kesikli Stokastik Karma Tamsayılı Programlama, Karma Tamsayılı Programlama, Karma Tamsayılı Programlama Modeli
Amaçlar		En Düşük Mâliyet, En Yüksek Kâr, Müşteri Gereksinimlerini Anında Yanıtlayabilme, Çevresel Değişikliklerden Etkilenmeme (Dayanıklılık), Tesis ve Taşıma Kapasitesi Kullanım Oranı/Dengeyi Enbüyükleme, Sistem Karmaşıklığını Enazlama
Çıktılar		Depolama, Araç sayısı, Müşteri taleplerinin karşılanması, Ürün fiyatları, Taşıma miktarı, Hizmet bölgesi, Tesis kapasitesi, Tesis yeri yerleşimi, Müşterilerin tesislere atanması
Lojistik Ağ Aşamaları	İleri Lojistik	Dağıtım Merkezi, Üretim Merkezi, Tedârik Merkezi
	Tersine Lojistik	Yeniden Dağıtım Merkezi, Elden Çıkarma Merkezi, Geri Dönüştürme Merkezi, Onarım Merkezi, Toplama-Denetim Merkezi
	Bütünleşik Lojistik	Üretim Merkezi, Dağıtım Merkezi, Yeniden Dağıtım Merkezi, Elden Çıkarma Merkezi, Geri Dönüştürme Merkezi

İleri Lojistik Ağ Tasarımı ile ilgili çalışmalarla ilgili genel özellikler (Yazarlar, Yayın Yılı, Problem Tipi, Uygulandığı Alan, Problem Özellikleri, Amaçlar, Kullanılan Yöntemler ve Eksik Yönü) Çizelge 2.4’te açıklanmaktadır. Bu çizelgede Eksik Yönü kolonu, yapılan çalışmanın bu tez çalışmasına göre eksik yönlerini göstermektedir.

Çizelge 2.4 : İleri Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problem Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Sabri ve Beamon	2000	Çok Amaçlı İleri Lojistik Ağ Tasarımı (Sanal problem)	Çalışmada amaç, sabit ve değişken mâliyetlerin enazlanması ve kapasite kullanımının enbüyüklenmesidir.	KarmaTamsayılı Programlama	Sadece programlama modelinin kurulması
Jayaraman ve Pirkul	2001	İleri Lojistik Ağ Tasarımı (Sağlık sektörü)	Amaç, toplam mâliyetlerin enazlanması olup, tesis yeri ve taşıma miktarı belirlenmektedir.	Lagrange Yakınlaştırma Tabanlı Sezgisel	Sezgisel yöntem kullanılması
Dasçi ve Verter	2001	Sürekli Lojistik Ağ Tasarımı (Uygulama yapılmamış)	Tek ürün için mâliyet enazlaması hedefiyle kapasite kısıtı olmayan üretim merkezlerinin lokasyonu belirlenmektedir.	Sürekli Yaklaştırma Yöntemi	Kapasite kısıtının olmaması
Zapfel ve Wasner	2002	İleri Lojistik Ağ Tasarımı (Paketleme Sektörü)	Amaç olarak salt taşıma mâliyetlerinin enazlanması gözönüne alınmıştır.	Karma Tamsayılı Programlama Modeli	Sadece programlama modeli kurulması
Syarif, Yun ve Gen	2002	İleri Lojistik Ağ Tasarımı (Sanal problemler)	En az mâliyetle tesislerin yerleri ve malzeme akışları belirlenerek tek ürün, tek dönemde tesis kapasite kısıtı ele alınmıştır.	Ağaç tabanlı Genetik Algoritma	Sadece atama problemi üzerinde çalışılması
Sung ve Song	2003	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Amaç mâliyet enazlanması olup, tesis yerlerini ve araç sayısını belirlenmektedir.	Tabu Arama Sezgiseli	Üç katmanlı problemin kullanılması
Jayaraman ve Ross	2003	Çok Ürünlü Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada tek dönem için en az mâliyetle taşıma miktarları ve dağıtım merkezlerinin yerleri amaçlanmaktadır.	Tavlama Benzetimi	Sadece atama problemi üzerinde çalışılması
Miranda ve Garrido	2004	Belirsiz Talepli Lojistik Ağ Tasarımı (Sanal problemler)	Çok dönemli bir süreçte toplam mâliyeti enazlamak için tesis yerleri ve atamalar ele almaktadır.	Lagrange Yakınlaştırma Tabanlı Sezgisel	Sadece sezgisel yöntem kullanılması
Yeh	2005	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Modelde kapasite kısıtlı tek dönem ve tek ürünlü ağda en az mâliyet için tesis yerleri ve taşıma miktarları hesaplanmaktadır.	Melez Sezgisel (Fırsatçı (Greedy))	Sadece atama problemi üzerinde çalışılması
Melachrinoudis, Messac ve Min	2005	Çok Amaçlı İleri Lojistik Ağ Tasarımı (Paketleme Sektörü)	Tesis yerleri belirlenerek mâliyetin enazlanması ve müşteri taleplerini karşılama oranının enbüyüklenmesi amaçlanmaktadır.	Lineer Fiziksel ve Karma Tamsayılı Programlama	Sadece modelin kurulması

Çizelge 2.4 (devam) : İleri Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problem Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Shimizu, Matsuda ve Wada	2006	Stokastik Talepli İleri Lojistik Ağ Tasarımı (Sanal problemler)	Modelde sabit ve değişken toplam taşıma mâliyetleri enazlanmaya çalışılmıştır.	Karma Tamsayılı Programlama ve Tabu Arama	Sadece atama problemi üzerinde çalışılması
Gen, Altıparmak ve Lin	2006	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Kapasite kısıtlı problemde tek ürün ve tek dönem üzerinden toplam mâliyetin enazlanmasına çalışılmaktadır.	Öncelik Tabanlı Kodlama ile Genetik Algoritma	Sadece atama problemi üzerinde çalışılması
Altıparmak ve diğerleri	2006	Çok Amaçlı Lojistik Ağ problemi (Plastik Sektörü)	Problemde mâliyetlerin enazlanması ve taleplerin karşılayabilme oranı ile kapasite kullanımını enbüyüklemek hedeflenmiştir.	Karma Tamsayılı Programlama ve Genetik Algoritma	Sadece atama problemi üzerinde çalışılması
Amiri	2006	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Kurulan modelde amaç mâliyetlerin enazlanması olup, tesis yerleri, taşıma miktarları ve tesis kapasiteleri belirlenmeye çalışılmaktadır.	Lagrange Yakınlaştırma Tabanlı Sezgisel	Sadece sezgisel yöntem kullanılması
Meepetchdee ve Shah	2007	Çok Amaçlı Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada amaçlar, lojistik ağın etkinliğini ve dayanıklılığını arttırmak ve ağın karmaşıklığını azaltmak olarak belirlenmiştir.	Karma Tamsayılı Programlama	Kapasite kısıtının dikkate alınmaması
Tsiakis ve Papageorgiou	2008	İleri Lojistik Ağ Tasarımı problemi (Kimya sektörü)	Tesis yerlerinin, kapasitelerinin ve tesisler arasındaki taşıma miktarları hesaplanarak mâliyet enazlaması hedeflenmiştir.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Kengpol	2008	İleri Lojistik Ağ Tasarımı (Şehir Lojistiği)	Modelde en yüksek müşteri hoşnutluğu ve en az taşıma mâliyeti için müşterilere ulaştırılacak ürün miktarı belirlenmiştir.	Analitik Hiyerarşi Süreci, Tamsayılı Programlama	Sadece programlama modelinin kurulması
Qin ve Qin	2008	İleri Lojistik Ağ Tasarımı (Otomotiv sektörü)	Çalışmada depolama, tesis açma, taşıma ve elde bulundurmama mâliyetleri toplamı enazlanması hedeflenmiştir.	Sezgisel Rotalama ile Melez Parçacık Sürü Algoritması	Üç katmanın ve sadece atamanın ele alınması
Cordeau, Laporte ve Pasin	2008	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Tedârikçilerin seçimi, tesis yerlerinin belirlenmesi gözönüne alınarak toplam mâliyetler enküçüklenmeye çalışılmıştır.	Tepe Tırmanma ile Melez Yerel Arama Sezgiseli	Sadece atama probleminin ele alınması

Çizelge 2.4 (devam) : İleri Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problemin Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Liu	2008	Çok Amaçlı İleri Lojistik Ağ Tasarımı (Sanal problemler)	Tesis yerleri ve kapasitelerinin belirlenerek amaç fonksiyonunda taşıma güvenilirliği, taşıma mâliyeti ve süresi dikkate alınmıştır.	Stokastik Akış Ağı Modelli bir Genetik Algoritma	Modelin üç katmanlı olması
Men ve Weng	2008	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Tesislerin yerleri belirlenip, taşıma rotaları saptanarak tesis açma, taşıma ve depolama mâliyetleri enazlanmaya çalışılmaktadır.	Parçacık Sürü Algoritması	Sadece atama probleminin ele alınması
Lin, Gen ve Wang	2009	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada taşıma mâliyeti, elde tutma mâliyeti, sipariş verme ve tesis mâliyetleri enazlanmaya çalışılmaktadır.	Bulanık Mantık Kontrollü Melez Genetik Algoritma	Sadece atama probleminin ele alınması
Miranda, Garrido ve Ceroni	2009	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada depo açma, depo işletme, sipariş verme, elde tutma, güvence stoğu ve taşıma mâliyeti enazlanmaya çalışılmaktadır.	e-İş Tabanlı Sezgisel	Sadece atama probleminin ele alınması
Jin ve Xin	2009	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada amaç, toplam tesis açma ve taşıma mâliyetlerini enküçükmektir.	Tavlama Benzetimi Yöntemi	Sadece atama probleminin ele alınması
Yi ve Dong	2009	İleri Lojistik Ağ Tasarımı (Uygulama yapılmamıştır.)	Modelde depo ve rota aynı anda düzenlenmekte ve toplam mâliyetlerin enazlanması amaçlanmaktadır.	Melez Parçacık Sürü Algoritması	Modelin üç katmanlı olması
Raad, Sinske, ve Vuuren	2010	Su Ağ Tasarımı Problemi (Sanal Problem)	Amaç su basıncı kısıtını sağlayarak talep noktalarının en az mâliyetle ve artık boru ile su ihtiyacını karşılamaktır.	Hipersezgisel Yöntemler	Katman ve rotalamanın olmaması
Martinez ve Zhang	2011	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Toplam dağıtım mâliyeti ve toplam taşıma süresi enazlanmaya çalışılmaktadır.	Pareto Karınca Kolonisi Algoritması	Sadece atama probleminin ele alınması
Kadadevaramath ve diğerleri	2012	İleri Lojistik Ağ Tasarımı (Otomotiv sektörü)	Taşıma, satınalma ve üretim mâliyetleri toplamı enazlanmaya çalışılmaktadır.	Parçacık Sürü Algoritması	Sadece atama probleminin ele alınması

Çizelge 2.4 (devam) : İleri Lojistik Ağı Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Confessore, Galiano, Liotta ve Stecca	2013	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Dört katmanlı ve kapasite kısıtlı lojistik ağ tasarımı için toplam mâliyetler enküçüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması
McClymont ve Keedwell, Savic, ve Smith	2013	Su Ağ Tasarımı (Gerçek Problem)	En az mâliyet, artık boru ve su yaşının sağlanması için talep noktalarına belirli basınçla su sağlanması hedeflenmiştir.	Hipersezgisel Yöntemler	Katman ve rotalamanın olmaması
Keizer, Haijema, Bloemhof ve Vorst	2015	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Çok ürünlü lojistik ağda işleme, taşıma mâliyetleri dikkate alınarak mâliyet enazlaması amaçlanmıştır.	Karma Tamsayılı Programlama ve Basit Sezgisel	Rotalamanın yapılmaması
Cortinhal, Lopes ve Melo	2015	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Çok katmanlı ve çok periyotlu lojistik ağ tasarımı için satınalma, üretim ve dağıtım mâliyetlerin enazlanması çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması
Kheiri, Keedwell, Gibson ve Savic	2015	Su Ağ Tasarımı (Gerçek problem)	Talep noktalarına en az mâliyet ve artık boru sağlayarak belirli basınçla su temini amaçlanmıştır.	Hipersezgisel Yöntemler	Katman ve rotalamanın olmaması
Nourbakhsh, Bai, Maia, Ouyang ve ve Rodriguez	2016	İleri Lojistik Ağ Tasarımı (Gerçek problem)	Depolama, taşıma ve işleme mâliyetleri dikkate alınarak mâliyet enazlaması yapılmaya çalışılmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Yang, Guo ve Ma	2016	İleri Lojistik Ağ Tasarımı (Gerçek problem)	Kapasite kısıtlı modelde tesis açma, işleme, taşıma ve karbon salınımlarında doğan vergi mâliyetleri enküçüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Modelin iki katmanlı olması
Keizer ve diğerleri	2017	İleri Lojistik Ağ Tasarımı (Sanal problemler)	Kapasite kısıtlı modelde sabit işleme, taşıma ve depolama mâliyetleri dikkate alınarak toplam kâr enbüyüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması
Dofour ve diğerleri	2017	İleri Lojistik Ağ Tasarımı (Gerçek problem)	Depolama, taşıma, satınalma mâliyetleri dikkate alınarak mâliyet enazlaması yapılmaya çalışılmaktadır.	Karma Tamsayılı Programlama	Modelin üç katmanlı olması

Tersine Lojistik Ağ Tasarımı ile ilgili çalışmalarla ilgili genel özellikler (Yazarlar, Yayın Yılı, Problem Tipi, Uygulandığı Alan, Problem Özellikleri, Amaçlar, Kullanılan Yöntemler ve Eksik Yönü) Çizelge 2.5'te açıklanmaktadır. Bu çizelgede Eksik Yönü kolonu, yapılan çalışmanın bu tez çalışmasına göre eksik yönlerini göstermektedir.



Çizelge 2.5 : Tersine Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problem Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Marin ve Pelegrin	1998	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	En az mâliyetle müşteri-fabrika arasındaki taşıma miktarları belirlenmeye çalışılmıştır.	Karma Tamsayılı Programlama ve Lagrange Tabanlı Sezgisel	İki katmanlı ağ üzerinde çalışma yapılması
Jayaraman, Guige ve Srivastava	1999	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	En az mâliyetle yeniden dağıtım merkezlerinin yerleri ve taşıma miktarları belirlenmeye çalışılmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Krikke, Harten ve Schuur	1999	Tersine Lojistik Ağ Tasarımı (Elektronik sektörü)	En az mâliyetle tesis yerleri ve dağıtım akışları ile birlikte yedek parçaların depo miktarları ve ara stokları ele alınmaktadır.	Karma Tamsayılı Programlama	Tesis kapasitesinin dikkate alınmaması
Jayaraman, Patterson ve Rolland	2003	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Kurulan modelde mâliyet enküçüklenmesi hedeflenmiştir.	Karma Tamsayılı Programlama ve Sezgisel Yöntem	Sadece sezgisel yöntem kullanılması
Listes ve Dekker	2005	Stokastik Talepli Tersine Lojistik Ağ Tasarımı (İnşaat sektörü)	Çalışmada en yüksek kârlılık için, tesis yerleri belirlenmekte ve geri dönüşüm ürünlerinin akışları hesaplanmaktadır.	Stokastik Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Min ve diğerleri	2006	Tersine Lojistik Ağ Tasarımı (Sanal bir örnek)	Toplam mâliyetin enküçüklenmesinin amaçlandığı modelde toplama merkezlerinin yerleri belirlenmeye çalışılmaktadır.	Karma Tamsayılı Programlama ve Genetik Algoritmalar	Sadece atama problemi üzerinde çalışılması
Üster ve diğerleri	2007	Kapalı Döngülü Tersine Lojistik Ağ Tasarımı (Otomotiv endüstrisi)	Çalışmada en az mâliyet amacıyla çok ürün ve tek dönem üzerinden toplama ve iyileştirme tesislerinin lokasyonları ve ürünlerin akış rotası belirlenmeye çalışılmaktadır.	Karma Tamsayılı Programlama ve Benders Ayrıştırma Algoritması	Kapasite kısıtının dikkate alınmaması

Çizelge 2.5 (devam) : Tersine Lojistik Ağı Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problemin Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Lieckens ve Vandaele	2007	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Amaç tesislerin yerleşimini ve rotaları belirleyerek dönüşümü yapılmış ürün gelirlerinden tesis açma, geri dönüştürme, depolama, taşıma, toplamama ve elden çıkarma mâliyetlerinin çıkarılması ile bulunan toplam kârı enbüyüklemektir.	Karma Tamsayılı Programlama Modeli ve Diferansiyel Genetik Algoritma	Modelin üç katmanlı olması
Demirel ve Gökçen	2008	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada toplama ve dağıtım merkezlerinin yerleri belirlenmeye çalışılmakta ve taşıma miktarları hesaplanarak mâliyet enazlanmaya çalışılmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Aras, Aksen ve Tanugur	2008	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Kurulan modelde toplama merkezi yerleri ve ürünlerin ağ içindeki hareketleri belirlenerek kullanılmış ürün fiyatları elde edilmektedir.	Karma Tamsayılı Programlama ve Tabu Arama tabanlı bir sezgisel	Kapasite kısıtının dikkate alınmaması
Du ve Evans	2008	Çok Amaçlı Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Onarım merkezlerinin yerleri, kapasiteleri ve taşıma miktarı belirlendiği çalışmada ilk amaç toplam mâliyeti enküçükmek iken, diğer amaç müşteri taleplerini karşılama oranını enbüyüklemektir.	Karma Tamsayılı Programlama ve Melez Serpilme Arama Algoritması	Modelin üç katmanlı olması
Zhou, Cao, Meng ve Xu	2008	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Çalışmada onarım hizmet merkezi ve müşteri tesis ataması yapılarak sabit işletme ve depolama mâliyeti ve en fazla müşteri bekleme süresi enküçülenmeye çalışılmaktadır.	Çok Amaçlı Tamsayılı Programlama Modeli ve Genetik Algoritmalar Yöntemi	Sadece atama probleminin ele alınması

Çizelge 2.5 (devam) : Tersine Lojistik Ağı Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problemin Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Shanshan ve Kejing	2008	Tersine Lojistik Ağ Tasarımı (Elektronik Sektörü)	Toplama, parçalara ayırma, geri dönüştürme, onarım mâliyetleri dikkate alınarak tesis lokasyonları ve ürün akışları belirlenmeye çalışılmıştır.	Karma Tamsayılı Doğrusal Programlama	Sadece programlama modelinin kurulması
Lee ve Dong	2009	Stokastik Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Problemde amaç, taşıma ve tesis açma mâliyetlerini enküçükmektir.	Tavlama Benzetimi Yöntemi	Sadece atama probleminin ele alınması
Lee, Gen ve Rhee	2009	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Demontaj ve işleme tesisleri için sabit tesis açma ve toplam taşıma mâliyetleri enazlanmaya çalışılmaktadır.	Melez Genetik Algoritma	Modelin üç katmanlı olması
Qin ve Ji	2010	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Toplama merkezi açma mâliyeti, taşıma mâliyeti ve müşterilerden toplamama mâliyeti enazlanmaya çalışılmaktadır.	Bulanık Mantık Simülasyonlu Genetik Algoritma	Sadece atama probleminin ele alınması
Soleimani ve Govindan	2014	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Satış gelirlerinden yeniden üretim, onarım, demontaj, taşıma ve depolama mâliyetlerinin toplamının çıkarılması ile elde edilen toplam kâr enbüyüklenmeye çalışılmaktadır.	Stokastik Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması
Roghanian ve Pazhoheshfar	2014	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	Çok ürünli lojistik ağda kullanılmış ürünlerin taşınması için toplam taşıma ve tesis açma mâliyetleri enazlanmaya çalışılmıştır.	Öncelik Tabanlı Genetik Algoritma	Rotalamanın dikkate alınmaması
Niknejad, ve Petrovic	2014	Tersine Lojistik Ağ Tasarımı (Sanal problemler)	İki farklı akışın dikkate alındığı lojistik ağda belirsiz talepler bulanıklaştırılarak toplam mâliyet enazlanmaya çalışılmaktadır.	Bulanık Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması

Çizelge 2.5 (devam) : Tersine Lojistik Ağı Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Ayvaz, Bolat, ve Aydın	2015	Tersine Lojistik Ağ Tasarımı (Gerçek problem)	Geri alınan ürünlerde miktar, kalite ve taşıma mâliyeti belirsizliği altında en düşük mâliyetle gelir enbüyüklenmesi amaçlanmıştır.	Stokastik Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması
Alshamsi ve Diabat	2015	Tersine Lojistik Ağ Tasarımı (Gerçek problem)	Satış gelirlerinden yeniden üretim, onarım, tesis açma ve depolama mâliyetlerinin toplamının çıkarılması ile elde edilen toplam kâr enbüyüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması
Zhang, Zhan, Chen ve Li	2016	Tersine Lojistik Ağ Tasarımı (Gerçek problem)	Üç katmanlı modelde toplam gelirlerin mâliyete oranı enbüyüklenmeye çalışılmaktadır.	Melez Genetik Algoritma	Rotalamanın yapılmaması
Alshamsi ve Diabat	2017	Tersine Lojistik Ağ Tasarımı (Gerçek Problem)	Üç katmanlı modelde satış gelirlerinden tesis açma, kapasite artırma, işleme, taşıma ve depolama mâliyetlerinin toplamının çıkarılması ile elde edilen toplam kâr enbüyüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli ve Genetik Algoritma	Modelin üç katmanlı olması

Bütünleşik Lojistik Ağ Tasarımı ile ilgili çalışmalarla ilgili genel özellikler (Yazarlar, Yayın Yılı, Problem Tipi, Uygulandığı Alan, Problem Özellikleri, Amaçlar, Kullanılan Yöntemler ve Eksik Yönü) Çizelge 2.6’da açıklanmaktadır. Bu çizelgede Eksik Yönü kolonu, yapılan çalışmanın bu tez çalışmasına göre eksik yönlerini göstermektedir.



Çizelge 2.6 : Bütünleşik Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problem Tipi ve Uygulandığı Alan	Problemin Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Fleischmann ve diğerleri	2001	Bütünleşik Lojistik Ağ Tasarımı (Elektronik ve kağıt sektörü)	Mâliyet enazlanması hedeflenmiş olup, üretim, dağıtım, toplama ve demontaj tesisi yerleri ve talep karşılama miktarları bulunmaya çalışılmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modeli kurulması
Salema, Pova ve Novais	2006	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Yapılan çalışmada kapasite kısıtları dikkate alınarak üretim, dağıtım, onarım ve toplama merkezlerinin yerleri belirlenmekte ve akışlar ve talep karşılama miktarları bulunmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Lu ve Bostel	2007	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Mâliyet enazlanması hedeflenen modelde, tesislerin yerleri belirlenmekte, akışlar hesaplanmaktadır.	Karma Tamsayılı Programlama ve Lagrange Tabanlı Sezgisel	Kapasite kısıtının dikkate alınmaması
Ko ve Evans	2007	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Geliştirilen modelde tesis açma, işletme, taşıma ve tesis kapasitesi genişletme mâliyetinin enazlanması amaçlanmakta ve dağıtım merkezlerinin yerleri ve dağıtım ağı içindeki akışlar belirlenmektedir.	Karma Tamsayılı Programlama ve Genetik Algoritma Tabanlı Sezgisel	Sadece atama problemi üzerinde çalışılması
Salema, Pova ve Novais	2007	Stokastik talepli Bütünleşik Lojistik Ağ Tasarımı (Ofis Dokümanı Şirketi)	Modelde mâliyet enazlanması hedeflenmiş olup problem sonucunda tesis lokasyonları, taşıma miktarları ve talep karşılama düzeyleri bulunmaya çalışılmaktadır.	Stokastik Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Hui, Zhen ve Hu	2007	Stokastik Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Üretim ile geri dönüşüm tesislerinin farklı yerlerde olması durumu için sabit mâliyetler ile taşıma ve depolama mâliyetleri enküçüklenmeye çalışılmıştır.	Karma Tamsayılı Doğrusal Programlama Modeli ve Genetik Algoritma	Üç katmanın ve sadece atama probleminin ele alınması

Çizelge 2.6 (devam) : Bütünleşik Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problemin Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Min ve Ko	2008	Bütünleşik Lojistik Ağ Tasarımı (Üçüncü Parti Lojistik Firması)	Çalışmada en az mâliyet hedefi ile toplama ve dağıtım merkezlerinin yerleri ve ağ içindeki akışlar belirlenmeye çalışılmaktadır.	Karma Tamsayılı Programlama ve Genetik Algoritma	Sadece atama problemi üzerinde çalışılması
Lee ve Dong	2008	Bütünleşik Lojistik Ağ Tasarımı (Bilgisayar Sektöründe Kiralama)	Algoritmada toplama ve dağıtım merkezlerinin yeri ve akışlar belirlenerek mâliyeti enazlanması amaçlanmıştır.	Karma Tamsayılı Programlama ve Tabu Arama Tabanlı Sezgisel	Modelin üç katmanlı olması
Peng ve Zhong	2008	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Amaç fonksiyonunda depolama, yeniden işleme, taşıma ve tesis işletme mâliyetleri gözönüne alınmıştır.	0-1 Karma Tamsayılı Programlama	Sadece modelin kurulması
Du, Wu ve Hu	2009	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Mâliyetleri enazlamak için gerekli yeniden üretim tesislerinin sayıları ve yerleri belirlenmiş ve dağıtım rotaları oluşturulmuştur.	Genetik Algoritma	Modelin üç katmanlı olması
Zhao	2009	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Tesis yerleri ve sayıları, lojistik akışlar belirlenerek toplam yatırım mâliyeti enazlanmaya çalışılmaktadır.	Karma Tamsayılı Programlama	Sadece programlama modelinin kurulması
Pishvae, Farahani ve Dullaert	2009	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Oluşturulan modelde amaçlar; üretim, dağıtım, toplama/inceleme, geri alma ve yoketme faaliyetleri ile toplam mâliyeti enazlamak ve lojistik ağının etkinliğini enbüyüklemektir.	Karma Tamsayılı Programlama Modeli ve Melez Genetik Algoritma	Sadece atama probleminin ele alınması

Çizelge 2.6 (devam) : Bütünleşik Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Sayed, Afia ve Kharbotly	2009	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Tesislerin lokasyonları, her fabrikadaki üretim ve taşıma miktarları ve depolarda her periyotta tutulacak ürün miktarları bulunarak toplam kârı enbüyüklemeye çalışılmaktadır.	Stokastik Karma Tamsayılı Doğrusal Programlama Modeli	Sadece programlama modelinin kurulması
Cardoso, Pova ve Relvas	2013	Bütünleşik Lojistik Ağ Tasarımı (Gerçek problem)	Toplam gelirden, üretim, dağıtım ve depolama mâliyetleri toplamı çıkartılarak net kazanç enbüyüklenmeye çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması
Keyvanshokoh, Fattahi, Hosseini ve Moghaddam Li	2013	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Kapasite kısıtlı çok ürünlü lojistik ağda kullanılmış ürünlerin taşınması için mâliyet enazlanmaya çalışılmıştır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması
	2013	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Depolama, taşıma, satınalma mâliyetleri dikkate alınarak mâliyet enazlaması yapılmaya çalışılmaktadır.	Karma Tamsayılı Programlama ve Lagrange Yakınlaştırma Tabanlı Sezgisel	Modelin iki katmanlı olması
Zhang ve Xu	2014	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Belirsiz müşteri talepleri altında optimum sipariş miktarı belirlenmeye çalışılarak en düşük mâliyetle gelir enbüyüklenmesi amaçlanmıştır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması ve modelin üç katmanlı olması
Hatefi ve Jolai	2014	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Tesis açma, işleme, taşıma mâliyetleri ve talep karşılamama cezasını dikkate alarak toplam mâliyet enazlanmaya çalışılmaktadır.	Karma Tamsayılı Programlama Modeli	Sadece modelin kurulması

Çizelge 2.6 (devam) : Bütünleşik Lojistik Ağ Tasarımı literatür araştırması sonuçları.

Yazarlar	Yayın Yılı	Problemi Tipi ve Uygulandığı Alan	Problem Amacı	Kullanılan Yöntemler	Çalışmanın Eksik/Yetersiz Yönü
Soysal, Ruwaard ve Vorst	2014	Bütünleşik Lojistik Ağ Tasarımı (Sanal problemler)	Toplam lojistik mâliyetleri ve taşımadan doğan emisyon gaz değerleri en azlanması amaçlanmıştır.	Çok Amaçlı Doğrusal Programlama Modeli	Sadece programlama modelinin kurulması
Kalaitzidou, Longinidis ve Georgiadis	2015	Bütünleşik Lojistik Ağ Tasarımı (Gerçek problem)	Çok periyotlu çok ürünlü modelde taleplerin karşılanabilmesi için taşıma ve operasyonel mâliyet enazlanmaya çalışılmıştır.	Karma Tamsayılı Programlama Modeli	Sadece programlama modelinin kurulması
Yi, Huang, Guo ve Shi	2016	Bütünleşik Lojistik Ağ Tasarımı (Gerçek problem)	Kapasite kısıtlı ağda tesis yerleri ve kullanılmış ürünlerin, ara ürün ve yeniden üretilen ürünlerin akışları belirlenerek taşıma, işleme ve sabit mâliyetlerin enazlanması amaçlanmıştır.	Karma Tamsayılı Programlama ve Melez Genetik Algoritma	Modelin iki katmanlı olması
Zohal ve Soleimani	2016	Bütünleşik Lojistik Ağ Tasarımı (Gerçek problem)	Çok amaçlı modelde ilk amaç satış gelirlerinden taşıma, üretim, dağıtım ve toplama mâliyetlerinin toplamının çıkarılması ile elde edilen toplam kâr enbüyüklenmeye çalışılması, ikinci amaç ise toplam emisyon değerinin enazlanmasıdır.	Karma Tamsayılı Programlama ve Karınca Kolonisi Algoritması	Rotalamanın yapılmaması
Guo, Wang, Fan ve Gen	2017	Bütünleşik Lojistik Ağ Tasarımı (Gerçek Problem)	Kapasite kısıtlı modelde tesis açma, işleme, taşıma ve karbon salınımından doğan vergi mâliyetleri enazlanmaya çalışılmaktadır.	Genetik Algoritma ve Parçacık Sürü Algoritması	Modelin üç katmanlı olması

2.6 Lojistik Ağ Tasarımı Problemi Hakkındaki Çalışmalardan Çıkarımlar

Lojistik Ağ Tasarımı Problemi için bu tez çalışmasının tamamlandığı süreye kadar yapılan literatür çalışmaları incelendiğinde aşağıdaki sonuçlara varılabilir:

- Ele alınan Lojistik Ağ Tasarımı problem tipleri dikkate alındığında en fazla İleri (Feed Forward) Lojistik Ağ Tasarımı problemlerine yönelik çalışma yapıldığı görülmektedir. Daha sonra sırasıyla Tersine ve Bütünleşik Ağ Tasarımı problemi için araştırmalar vardır.
- Literatürde Lojistik Ağ Tasarımı problemleri için bir problem kümesi olmadığından dolayı yapılan çalışmalarda geliştirilen yöntemlerin etkinliği için ortak problem kümesi üzerinden karşılaştırmalar yapılmamıştır.
- Uygulamaların daha çok sanal problemler üzerinde gerçekleştirildiği gözlemlenmiştir. Gerçek yaşamdan uygulamalara ise özellikle son yıllarda yapılan çalışmalarda rastlanmaktadır.
- Lojistik Ağ Tasarımına yönelik yapılan çalışmaların çoğunluğunda Karma Tamsayılı Programlama modeli kullanılmıştır. Belirsizliklerin artmasıyla stokastik modellerin kullanımına da rastlanmıştır. Son yıllarda ise çözüm için daha çok Metasezgisel yöntemlerin kullanıldığı görülmektedir. En çok kullanılan metasezgisel yöntem Genetik Algoritma olmuştur.
- Lojistik Ağ Tasarımı optimizasyonu ile ilgili yapılan yayınlarda genel olarak mâliyet enküçüklenmesi amaçlanmaktadır.
- İncelenen akademik çalışmalara göre Lojistik Ağ Tasarımı Problemlerinde çözüm yöntemlerinin, tüm sürece değil, depolama, tesis yerleşimi veya araç rotalama gibi sürecin belirli bir bölümüne uygulandığı görülmektedir.

Literatür araştırması, hazırlanan bu tez çalışması konusunda irdelendiğinde aşağıdaki sonuçlara ulaşılmaktadır:

- Lojistik Ağ Tasarımı için daha önce Hipersezgisel Yöntemlerin kullanıldığı bir çalışmaya rastlanmamıştır. LAT alt problemlerine bakıldığında ise Garrido ve Castro'nun 2009 yılında Kapasite Kısıtlı Araç Rotalama Problemi konusundaki yayınından başka Lojistik Ağ Tasarımı kapsamına giren bir çalışma yapılmadığı görülmektedir. Bu çalışmada da tek katman üzerinde salt

rotalama problemi çalışılmış, LAT’da yer alan çok katmanlı yapı ele alınmamıştır.

- Literatür araştırması ağ tasarımı problemleri kapsamında değerlendirildiğinde ise Hipersezgisel yöntemlerin Su Ağ Tasarımı üzerinde uygulamaları görülmektedir (Raad ve diğ., 2010), (McClymont ve diğ., 2013) ve (Kheiri ve diğ., 2015). Su Ağ Tasarımı problemlerinde temel amaç, tüm talep noktalarına suyun belirli bir basınçla, en az mâliyetle ve en az artık boru ile ulaşımını sağlamak olup, talep noktaları arasında dönecek borular ve boru çapları belirlenmektedir. Bu problem tipinde katman yapısı, rotalama, kapasite gibi Lojistik Ağ Tasarımı özellikleri bulunmadığından, problem doğası itibarıyla LAT’a göre daha basit bir problemidir.
- Daha önce yapılan LAT araştırmalarından farklı olarak bu tez çalışmasında odaklanılan LAT problemi; dört katmanlı ağ modelini ele almakta, kapasite kısıtını kullanmakta, atamanın yanında rotalama sürecini de içermekte ve kullanılan yöntem olarak salt basit sezgisel yöntem algoritma havuzu ile bulguları elde etmektedir.
- Bu tez kapsamında geliştirilen çözüm yöntemi, hem sanal hem de gerçek problemler üzerinde uygulanacaktır.

2.7 Günümüzde Ağ İşletmecilerinin Sorun Olarak Yaşadığı Durumlar

İlk tedârikçiden son müşteriye kadar uzanan ürün ve bilgi akışını içeren bir yaklaşım olan Tedârik Zinciri Yönetimi (TZY), son yıllarda doğrusal bir yönetim kavramı olarak değil bir ağ yönetimi kavramı olarak ele alınmaktadır (Lu ve diğ., 2013). Bir firmanın lojistik ağı; tedârikçi, nakliyecisi, üretici, dağıtım merkezleri, perakendeci ve tüketici ile ortaya çıkan tedârik zincirini oluşturan sistemler, alt sistemler, operasyonlar, faaliyetler ve bunların birbirleriyle olan ilişkilerini içeren karmaşık bir bütündür. Bu karmaşık bütünün tasarımı, modelinin oluşturulması ve yaşama geçirilmesi, işletmenin en yüksek etkinlik ve verimliliğe sahip olmasında oldukça belirleyici bir rol üstlenecektir (Bakoğlu ve Yılmaz, 2001).

Lojistik Ağ Tasarımı kapsamında ağ ekonomisi üzerinde aşağıdaki unsurlar ele alınmaktadır (Finger ve Kunneke, 2011):

- Atama/Rotalama

- Hammadde/Bitmiş ürün akışları
- Tedârikçi ve Müşterinin aynı anda ele alınması
- Süreçlerin tedârik, üretim, nakliye ve teslîmatı içerdiği
- Zamanın ve uzaklığın en önemli sorunları yarattığı (Mâliyet en çok etki eden etmen)

Finger ve Künneke'ye göre (2013) lojistik süreçlerin ağ olarak tanımlanabilmesi için yukarıdakilerin tek başına ya da birarada alınmasının zorluğu nedeniyle taşımacılık sektörü gibi karmaşıklık düzeyi yüksek olan sektörlerde hizmet veren firmalarda operasyonel ve stratejik kararlar iç içe analiz edilmelidir. Örneğin, bir kargo şirketinde önemli bir müşteriye hızlı ve zamanında hizmet verilememesine neden olan bir uygulama stratejik düzeyde bir uygulama değişikliğine neden olabilir. LAT bir sistem olarak ele alındığında, bu sistemin bütünleşmesi, yönetim alanındaki guruların da katkılarıyla gelişim göstermektedir. Daha önceleri yöneticiler bu bütünleşim felsefesini işletmelerde uygulamaya çalışmışlar ama bilgi teknolojisinden yoksun olan işletmeler, sisteme yönelik yaklaşımı tam anlamıyla uygulayamamışlardır. 1950'li yıllara döndüğümüzde, endüstri dinamikleri alanındaki gelişmeler, tedârik zincirinin bütünleşik bir sistem olması zorunluluğunu ortaya çıkarmıştır. Son yıllarda teknolojiye gelişmeler sâyesinde iletişimin baş döndürücü bir hızla artmasıyla birlikte TZY'deki bütünleşim felsefesi de ivme kazanmıştır. Hem yönetim hem de teknoloji alanındaki gelişmeler, üreticiden perakendeciye doğru güçlü bir ötelemeyi doğurmuştur (Johnson ve Pyke, 2001).

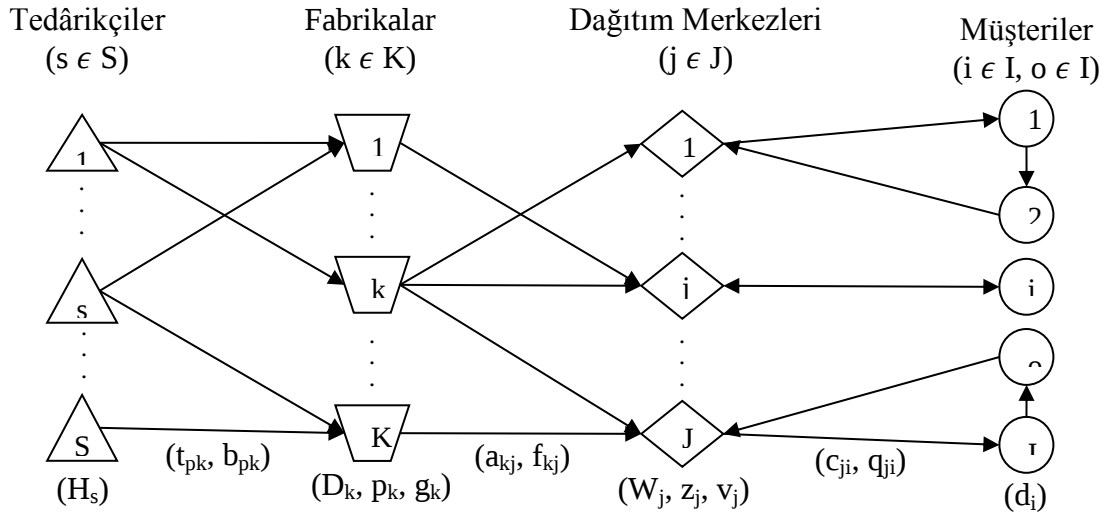
Tedârik zincirinin plânlama, tasarım ve kontrol aşamaları boyunca bir bütün olarak ele alınmasının stratejik önemi zaman içinde kavranmıştır. Bu nedenle tedârik zinciri yönetimi kapsamında büyük bir öneme sahip olan Lojistik Ağ Tasarımı, firmalara mâliyet avantajı kazandırarak rekabetçi güçlerini arttırmalarını sağlamaktadır. LAT'nın mâliyet ve müşteri hoşnutluğu üzerindeki etkileri nedeniyle çoğu firma bu konuya odaklanmış durumdadır (Min ve Zhou, 2002).

Bugüne kadar uygulanagelmiş lojistik problemlerin çözümünde üç ve daha az sayıda katman kullanılması, ağların daha karmaşık duruma gelmesi ile dört ve daha fazla katman kullanımını gerektirmektedir. Örneğin bu tez çalışması kapsamında DHL Express'in yerine getirdiği lojistik işlemleri, karmaşık ve 4 ve üzeri katmanlı uygulamalara örnek içermektedir.

3. ELE ALINAN LOJİSTİK AĞ TASARIMI PROBLEMİ VE ÇÖZÜMÜ İÇİN BİR MODEL ÖNERİSİ

3.1 Problemin Tanımı

Bu tez çalışmasında 4 katmanlı tek ürünlü İleri Lojistik Ağ Tasarımı Problemi ele alınmıştır. Lojistik Ağ Tasarımı Probleminde belirlenen amaç, müşteri gereksinimlerinin karşılanması koşuluyla tesislerin sabit mâlîyetlerini ve satınalma, üretim, depolama ve taşıma gibi değişken mâlîyetlerini enküçüklemeye çalışmaktır. Şekil 3.1’de İleri Lojistik Ağ Tasarımı Problemi için bir tasarım örneği bulunmaktadır. Bu tasarımda temel olarak tedârikçi seçimi, tesis ve dağıtım merkezi yerlerinin seçimi, hangi fabrikanın, hangi tedârikçiden hammadde alacağı ve hangi dağıtım merkezine ürün göndereceği, müşterilere hangi dağıtım merkezi üzerinden hizmet edileceği ve dağıtım merkezi ile müşteriler arasındaki araç rotalarının ne olması gerektiği belirlenmeye çalışılmaktadır.



Şekil 3.1 : İleri Lojistik Ağı Tasarımı.

3.2 Oluşturulan Model

3.2.1 Varsayımlar

Oluşturulan model için aşağıdaki varsayımlar yapılmıştır:

- Tedârikçilerin ve müşterilerin sayıları ile müşteri talep ve tesis kapasite miktarları bilinmektedir.
- Potansiyel fabrika ve dağıtım merkezi yerleri önceden belirlenmiştir.
- Müşteriler ürünü tek bir dağıtım merkezinden temin edeceklerdir ve müşteri talepleri tam olarak karşılanacaktır.
- Bir dağıtım merkezinden bir müşteriye salt bir araç çıkabilecektir.
- Problem tek ürün üzerinden bir dönem için modellenmiştir.
- Taşıma modu olarak salt karayolu taşımacılığı dikkate alınmıştır.
- Araç rotalama problemi için zaman penceresi dikkate alınmamıştır.

3.2.2 Karar değişkenleri ve parametreler

Hazırlanan modelde müşteriler için $i \in I$ ve $o \in I$, dağıtım merkezleri için $j \in J$, fabrikalar için $k \in K$ ve tedârikçiler için $s \in S$ indisleri kullanılmıştır. Modelde kullanılan karar değişkenleri ve parametreler aşağıda belirtilmektedir:

$$z_j = \begin{cases} 1, & j. \text{dağıtım merkezi aktifse} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$p_k = \begin{cases} 1, & k. \text{fabrika aktifse} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$y_{ji} = \begin{cases} 1, & j. \text{dağıtım merkezi, } i. \text{müşteriye hizmet veriyorsa} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$n_{ji} = \begin{cases} 1, & j. \text{dağıtım merkezinden çıkan araç, ilk } i \text{ müşteriye uğrarsa} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$r_{ji} = \begin{cases} 1, & i. \text{müşteriden çıkan araç } j. \text{dağıtım merkezine dönerse} \\ 0, & \text{Aksi durumda} \end{cases}$$

$$x_{ioj} = \begin{cases} 1, & \text{eğer } j. \text{dağıtım merkezinden çıkan araç, } i \text{ müşterisinden sonra} \\ & \text{ } o \text{ müşterisine uğrarsa} \\ 0, & \text{Aksi durumda} \end{cases}$$

b_{sk} : s. tedârikçiden, k. fabrikaya taşınan hammadde miktarı ($s \in S, k \in K$)

f_{kj} : k. fabrikadan, j. dağıtım merkezine taşınan ürün miktarı ($k \in K, j \in J$)

q_{ji} : j. dağıtım merkezinden, i. müşteriye taşınan ürün miktarı ($j \in J, i \in I$)

Karma Tamsayılı Programlama Modelinde kullanılacak parametreler ise şunlardır:

H_s : s. tedârikçinin hammadde üretim kapasitesi ($s \in S$)

D_k : k. fabrikanın kapasitesi ($k \in K$)

W_j : j. dağıtım merkezinin kapasitesi ($j \in J$)

d_i : i. müşterinin talebi ($i \in I$)

v_j : j. dağıtım merkezinin sabit işletme mâliyeti ($j \in J$)

g_k : k. fabrikanın sabit işletme mâliyeti ($k \in K$)

e_k : k. fabrikada birim üretim mâliyeti ($k \in K$)

c_{ji} : j. dağıtım merkezinden i. müşteriye olan taşıma mâliyeti ($j \in J, i \in I$)

w_{io} : i. müşteriden o. müşteriye olan taşıma mâliyeti ($i \in J, o \in I$)

a_{kj} : k. fabrikadan j. dağıtım merkezine birim ürün taşıma mâliyeti ($k \in K, j \in J$)

t_{sk} : s. tedârikçiden k. fabrikaya birim taşıma ve satınalma mâliyeti ($s \in S, k \in K$)

α : ürünlerdeki birim hammadde kullanım oranı

u_j : müşteri rotalarında alt turların engellenmesi için kullanılan bir değişken

3.2.3 Amaç fonksiyonu

Modeldeki amaç fonksiyonu; tesis kullanma (aktifleştirme), üretim, taşıma ve satınalma mâliyetlerinin toplamını enazlamak olup aşağıda belirtilmiştir:

$$\begin{aligned} \text{Min } f = & \sum_k g_k p_k + \sum_j v_j z_j + \sum_k \sum_j e_k f_{kj} \\ & + \sum_s \sum_k t_{sk} b_{sk} + \sum_k \sum_j a_{kj} f_{kj} \\ & + \sum_j \sum_i n_{ji} c_{ji} + \sum_j \sum_i \sum_o x_{ioj} w_{io} + \sum_j \sum_i r_{ji} c_{ji} \end{aligned} \quad (3.1)$$

3.2.4 Kısıtlar

Modeldeki kısıtlar aşağıda yer almaktadır:

1. Her müşteri için ancak bir dağıtım merkezinden ürün temin edilebilir. Bu kısıt aşağıda belirtilmiştir:

$$\forall i \in I \text{ için } \sum_j y_{ji} = 1, \quad j \in J \quad (3.2)$$

2. Müşterilere bir dağıtım merkezinden gönderilen toplam miktar, dağıtım merkezinin kapasitesini aşamaz. Bu kısıt (3.3) her dağıtım merkezi için geçerlidir.

$$\forall j \in J \text{ için } \sum_i d_i y_{ji} \leq W_j z_j, \quad j \in J \quad (3.3)$$

3. Müşterilerin talepleri dağıtım merkezleri tarafından tümüyle karşılanacaktır. Bu kısıt aşağıda kısıt 3.4'te belirtilmiştir:

$$\forall i \in I \text{ ve } \forall j \in J \text{ için } q_{ji} y_{ji} \leq d_i \quad (3.4)$$

4. Her dağıtım merkezi için, dağıtım merkezine fabrikalardan ulaşan ürün miktarı, dağıtım merkezinden müşterilere taşınan ürün miktarından büyük veya eşit olmalıdır.

$$\forall j \in J \text{ için } \sum_k f_{kj} \geq \sum_i q_{ji}, \quad i \in I, k \in K \quad (3.5)$$

5. Her tedârikçiden fabrikalara gönderilen hammadde miktarı ise, ilgili tedârikçinin kapasite kısıtını aşmamalıdır.

$$\forall s \in S \text{ için } \sum_k b_{sk} \leq H_s, \quad k \in K \quad (3.6)$$

6. Bir fabrikada ürün üretebilmek için gerekli olan hammadde miktarı, ürünündeki birim hammadde kullanım oranı (α) ile ilişkilidir. Her fabrikadan dağıtım merkezine gönderilecek ürün miktarı, o fabrikaya tedârikçi firmalardan ulaşan gerekli hammadde miktarının bu sabit oranla çarpılması ile sınırlandırılmıştır. Bu kısıt aşağıda yer almaktadır:

$$\forall k \in K \text{ için } \alpha \sum_j f_{kj} \leq \sum_s b_{sk}, \quad j \in J, s \in S \quad (3.7)$$

7. Her fabrika için, ürün üretim miktarı fabrika kapasite kısıtını aşmamalıdır. Aşağıdaki kısıtta kapasite değeri, hammadde miktarı üzerinden ele alınmıştır.

$$\forall k \in K \text{ için } \alpha \sum_j f_{kj} \leq D_k p_k, \quad j \in J \quad (3.8)$$

8. Bir dağıtım merkezinin bir müşteriye hizmet etmesi için müşterinin, dağıtım merkezinin rotası üzerinde olması gerekmektedir. Bu durum dağıtım merkezinden

müşterilere veya müşterilerin kendi arasında güzergâhın olup olmamasına bakılarak anlaşılmaktadır.

$$\forall i \in I \text{ ve } \forall j \in J \text{ için } y_{ji} = n_{ji} + \sum_o x_{ioj}, \quad o \in I \quad (3.9)$$

9. Açık olan her dağıtım merkezinden ancak ve ancak bir araç yola çıkabilecektir. Eğer dağıtım merkezi açık değil ise, o dağıtım merkezinden araç çıkamaz. Bu kısıt aşağıda belirtilmiştir:

$$\forall j \in J \text{ için } \sum_i n_{ji} = z_j, \quad i \in I \quad (3.10)$$

10. Açık olan dağıtım merkezine müşterilerden ancak bir araç geri dönebilecektir. Eğer dağıtım merkezi açık değil ise, o dağıtım merkezine araç dönemez.

$$\forall j \in J \text{ için } \sum_i r_{ji} = z_j, \quad i \in I \quad (3.11)$$

11. Müşteriler arasında rotalama sürecinde alt turların oluşması önlenmelidir. Bu da aşağıdaki kısıt ile gerçekleştirilir. Kısıtta “ $|I|$ ” ifadesi müşteri sayısını gösterir:

$$\forall i \in I \text{ ve } \forall o \in I \text{ için } u_i - u_o + |I| \sum_j x_{ioj} \leq |I| - 1, \quad j \in J \quad (3.12)$$

12. Modelde dağıtım merkezinden çıkan her araç, rotası üzerindeki müşterileri ancak bir kez ziyaret edebilmelidir. Bu durum, bir müşteriye mutlaka bir aracın ulaşması ve bir aracın terk etmesi şeklinde yorumlanabilir. Bu kısıtlar aşağıda yer almaktadır:

$$\forall o \in I \text{ için } \sum_j n_{jo} + \sum_j \sum_i x_{ioj} = 1, \quad i \in I, j \in J \quad (3.13)$$

$$\forall o \in I \text{ için } \sum_j r_{jo} + \sum_j \sum_i x_{oij} = 1, \quad i \in I, j \in J \quad (3.14)$$

13. Karar değişkenleri ve negatif olmama kısıtları aşağıda belirtilmiştir.

$$z_j = \{0, 1\}, p_k = \{0, 1\}, y_{ji} = \{0, 1\}, n_{ji} = \{0, 1\}, r_{ji} = \{0, 1\}, x_{ioj} = \{0, 1\}, \quad (3.15)$$

$$u_j = \{0, 1\} \quad (3.16)$$

$$b_{sk} \geq 0, \quad f_{kj} \geq 0, \quad q_{ji} \geq 0$$

14. Modele araç kapasite kısıtı da eklenebilir (C: Kapasite kısıtı) (Değişken durum için konulabilir ama bu varsayımlar altında gerek görülmemiştir.).

$$\forall j \in J \text{ için } \sum_i n_{ji} d_i + \sum_o \sum_i x_{oij} d_i \leq C, \quad i \in I, o \in I \quad (3.17)$$

3.2.5 Önerilen Karma Tamsayılı Programlama Modeli

Çözüm önerisi 4 katmanlı, tek ürünlü ileri LAT problemi için oluşturulan model aşağıdadır:

$$\begin{aligned} \text{Min } f = & \sum_k g_k p_k + \sum_j v_j z_j + \sum_k \sum_j e_k f_{kj} \\ & + \sum_s \sum_k t_{sk} b_{sk} + \sum_k \sum_j a_{kj} f_{kj} \\ & + \sum_j \sum_i n_{ji} c_{ji} + \sum_j \sum_i \sum_o x_{ioj} w_{io} + \sum_j \sum_i r_{ji} c_{ji} \end{aligned}$$

$$\forall i \in I \text{ için } \sum_j y_{ji} = 1, \quad j \in J$$

$$\forall j \in J \text{ için } \sum_i d_i y_{ji} \leq W_j z_j, \quad j \in J$$

$$\forall i \in I \text{ ve } \forall j \in J \text{ için } q_{ji} y_{ji} \leq d_i$$

$$\forall j \in J \text{ için } \sum_k f_{kj} \geq \sum_i q_{ji}, \quad i \in I, k \in K$$

$$\forall s \in S \text{ için } \sum_k b_{sk} \leq H_s, \quad k \in K$$

$$\forall k \in K \text{ için } \alpha \sum_j f_{kj} \leq \sum_s b_{sk}, \quad j \in J, s \in S$$

$$\forall k \in K \text{ için } \alpha \sum_j f_{kj} \leq D_k p_k, \quad j \in J$$

$$\forall i \in I \text{ ve } \forall j \in J \text{ için } y_{ji} = n_{ji} + \sum_o x_{ioj}, \quad o \in I$$

$$\forall j \in J \text{ için } \sum_i n_{ji} = z_j, \quad i \in I$$

$$\forall j \in J \text{ için } \sum_i r_{ji} = z_j, \quad i \in I$$

$$\forall i \in I \text{ ve } \forall o \in I \text{ için } u_i - u_o + |I| \sum_j x_{ioj} \leq |I| - 1, \quad j \in J$$

$$\forall o \in I \text{ için } \sum_j n_{jo} + \sum_j \sum_i x_{ioj} = 1, \quad i \in I, j \in J$$

$$\forall o \in I \text{ için } \sum_j r_{jo} + \sum_j \sum_i x_{oij} = 1, \quad i \in I, j \in J$$

$$z_j = \{0, 1\}, p_k = \{0, 1\}, y_{ji} = \{0, 1\}, n_{ji} = \{0, 1\}, r_{ji} = \{0, 1\}, x_{ioj} = \{0, 1\},$$

$$u_j = \{0, 1\}$$

$$b_{sk} \geq 0, \quad f_{kj} \geq 0, \quad q_{ji} \geq 0$$





4. HİPERSEZGİSEL ALGORİTMALAR

4.1 Yöntem tanımı

Günümüzde, özellikle ekonomik nedenlerden dolayı, gerçek yaşamda karşılaşılan ve optimizasyon gerektiren problemler oldukça önem kazanmıştır. Bu tür problemler için kesin modellemeler yapmak karmaşık ve ayrıntılı olduğundan, aynı zamanda uzun süreler gerektirdiği için uygulanabilir değildir. Bu nedenlerle, optimum sonucu garanti etmeyen ancak uygun süreler içerisinde yürütülmesi kolay, fazla kaynak gerektirmeyen ve uygulanabilir çözümler üreten basit sezgisel yöntemlere başvurulmaktadır. Metasezgisel Yöntemler son yıllarda başarısını kanıtlasa da, yürütülmesi oldukça zor, mâliyetli ve problem özelinde şekillendirilen tekniklerdir. Yöntemin performansı daha çok komşuluk yapısını sağlayan operatörlere ve modeli kuran kişinin uzmanlığına bağlıdır (Özcan ve diğ, 2008). Bir problem için yüksek kaliteli çözümler üreten bir Metasezgisel Algoritma, problemde birtakım değişiklikler yapıldığında başarısız olabilmektedir. Bu yüzden, gerçek yaşamda kullanılacak Metasezgisel yöntemlerin duruma uyumunu sağlamak üzere elden geçirilmesi gerekmektedir.

Burke ve diğerlerine göre (2003a), yöneticiler iş yaşamında uzun sürede elde edilen ve mâliyeti yüksek optimum çözümler yerine, kullanılması kolay ve dayanıklı basit sezgisel yaklaşımlarla, kısa sürede üretilen, yeterli kalitede ve düşük mâliyetteki çözümleri yeğlemektedir. Bu durum genelleştirilmiş, dayanıklı ve problemden bağımsız yeniden kullanılabilen Hipersezgisel Algoritmaların önünü açmıştır. Hipersezgisel Algoritmalarla ilgili bir diğer durum ise, basit sezgisel yöntemleri uyumlu bir şekilde organize etmesidir. Basit bir sezgisel yöntem her problem için iyi çözüm üretmese de diğer basit sezgisel tekniklerle birlikte uygulandığında kaliteli sonuçlar ortaya çıkarabilmekte, diğer bir deyişle çözümün belirli aşamalarında performansı yüksek olabilmektedir. Bu nedenle, sezgisel yöntemleri tek tek kullanmak yerine bütünleşik melez bir yapı oluşturmak, çözüm kalitesini arttırmaktadır. Hipersezgisel Algoritmalar bu organizasyonu yöneterek basit sezgisel yöntemlerin uygulanma sırasını belirlemektedir (Chakhlevitch ve Cowling, 2008).

Bu noktadan hareketle Hipersezgisel Algoritmaların ortaya çıkma nedenleri için şunları söyleyebiliriz:

- İş yaşamında optimuma yakın çözümler üreten, dayanıklı ve problemten bağımsız çözümlerin yeterli bulunması;
- Basit sezgisel yöntemlerin bütünleşik bir süreçte birlikte kullanılmasını sağlayacak yapısal model gereksinimi;
- En az parametreye dayalı optimizasyon çözümlerine duyulan gereksinim.

Hipersezgisel sürece ilişkin ilk düşünceler, Fisher ve Thompson tarafından 1961 ve 1963 yılında Atölye Tipi İş Çizelgeleme problemine uygulanmıştır. Yapılan çalışmada çizelgeleme problemlerine uygulanacak basit sezgisel yöntemlerin seçimi, olasılık tabanlı yapılmaktadır. Mockus ve diğerleri tarafından 1989, 1991 ve 1997 yıllarında yapılan çalışmalarda, sezgisel yöntemlerin seçilmesinde Bayesian yaklaşımı kullanılmıştır. 1998 yılında Hart ve Ross, Atölye Tipi İş Çizelgeleme Problemi için basit sezgisel yöntemlerin kombinasyonunu belirlemeye yönelik bir Genetik Algoritma geliştirmiştir (Chakhlevitch ve Cowling, 2008).

“Hipersezgisel” terimi ilk olarak Cowling tarafından 2000 yılında ortaya konulmuş ve yöntem satış çizelgeleme problemi için kullanılmıştır. Cowling’in Hipersezgisel Algoritmalar için yaptığı tanım aşağıda belirtilmiştir:

“Hipersezgisel Algoritmalar, Metasezgisel Yöntemlerden daha üst düzeyde soyutlama sağlayan ve algoritma sürecinde herhangi bir zamanda hangi klasik sezgisel yöntemin uygulanacağına karar veren bir çözüm yaklaşımıdır.”

Soubeiga 2003 yılında yaptığı çalışmada ise Hipersezgisel Algoritmaları, “Her karar aşamasında bir yöntem havuzundan, uygulanacak basit sezgisel yöntemi seçen bir sezgisel” olarak tanımlamıştır.

Chakhlevitch ve Cowling 2005 yılında Hipersezgisel Algoritmalar için “Problemin çözüm uzayı yerine, düşük düzeyli sezgisel yöntemlerin oluşturduğu uzayda arama yapan üst düzeydeki bir prosedür” tanımlamasını yapmışlardır.

Klasik sezgisel teknikler, optimum çözümü zor ve uzun süreler gerektiren büyük boyutlu problemler için, kabul edilebilir süre içinde optimuma yakın çözümler üreten, kullanımı kolay ve probleme özgü yöntemlerdir. Genetik Algoritmalar, Tavlama Benzetimi, Karınca Kolonisi gibi popülasyon veya komşuluk bazlı

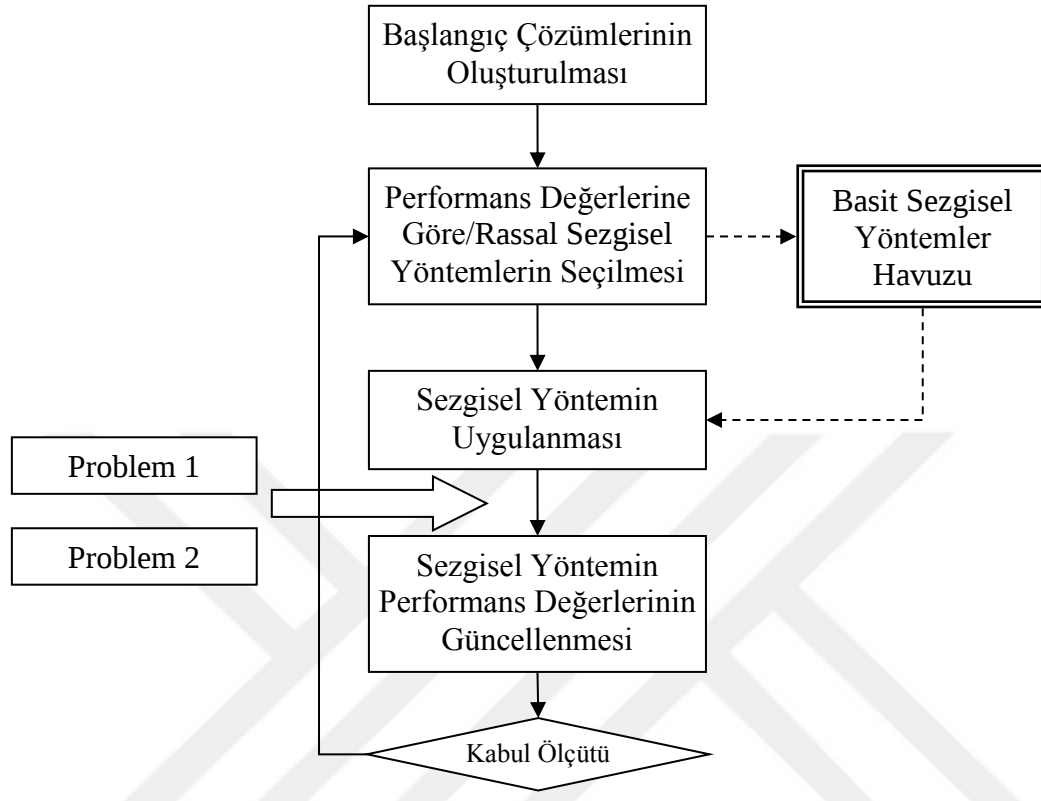
Metasezgisel yöntemler ise problemin çözüm uzayı içinde belirli bölgelerde yerel aramalar yaparak varolan çözümü iteratif bir yolla geliştirmeye çalışan ve klasik sezgisel yöntemlere göre daha kaliteli ancak daha uzun sürede çözüm üreten, doğadan esinlenerek oluşturulmuş optimizasyon yöntemleridir. Burke ve diğerlerine göre (2003a) Metasezgisel yöntemler, gerçek yaşamda ve iş dünyasında uygulamada bâzı temel olumsuzluklara sahiptirler. Bunlar, yöntemlerin esnek ve dayanıklı olmaması (çözüm kalitesinin çevresel değişikliklerden büyük oranda etkilenmesi), kolay kullanılabilir olmaması ve uygulanmasının mâliyetli olmasıdır. Hipersezgisel yöntemler bu eksikliklerden yola çıkılarak geliştirilmiş, problemden bağımsız ve genelleştirilmiş dinamik sezgisel yordamlardır.

Hipersezgisel Tekniklerde amaç, probleme özgü sezgisel yöntemlerdeki gibi optimuma çok yakın çözümler bulmak değil, hemen her tip optimizasyon problemine uygulanabilecek, iyi çözümler üreten genel bir yapı ortaya koymaktır (Chakhlevitch ve Cowling, 2008). Belirli bir probleme özgü sezgisel bir yöntem, farklı alandaki başka bir probleme uygulandığı zaman iyi sonuç vermeyebilir, fakat algoritma içinde başka yöntemler ile ardarda kullanıldığında (melezleştirildiğinde) iyi performans gösterebilir. Bu noktadan hareketle Hipersezgisel yöntemler, algoritma adımlarını sezgisel yolla tasarlayan ve Şekil 4.1’de gösterilen bir süreç barındırırlar. Bu şekilde yer alan performans değerleri, sezgisel yöntemin çözümde sağladığı iyileşme ve çözüm süresi ile belirlenir.

Hipersezgisel Algoritmalarda basit sezgisel tekniklerin nasıl seçileceğine kılavuzluk eden bir öğrenme mekanizması vardır. Hipersezgisel yöntemler süreç açısından dörde ayrılabilir:

1. Basit Hipersezgiseller (Sezgisel yöntemleri rassal seçen yöntemler)
2. Seçim Fonksiyonlu Hipersezgiseller (Sezgisel yöntemlere ağırlık verip, bu ağırlığa göre seçim yapan yöntemler)
3. Metasezgisel Yöntemlerle Melezleştirilen Hipersezgiseller (Sezgisel yöntemlerin uygulanma sırasını kodlayıp Metasezgisel kullanarak bu sırayı güncelleyen yöntemler)
4. Öğrenme Tabanlı Hipersezgiseller (Basit sezgisel yöntemlerin daha önceki performanslarını dikkate alarak, hangi yöntemin seçileceğine karar veren öğrenme süreçlerine sahiptir.)

Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritmaların akış diyagramı Şekil 4.1’de, örnek olması amacıyla bir Hiper-Genetik Algoritmanın süreç adımları ise Şekil 4.2’de gösterilmektedir.

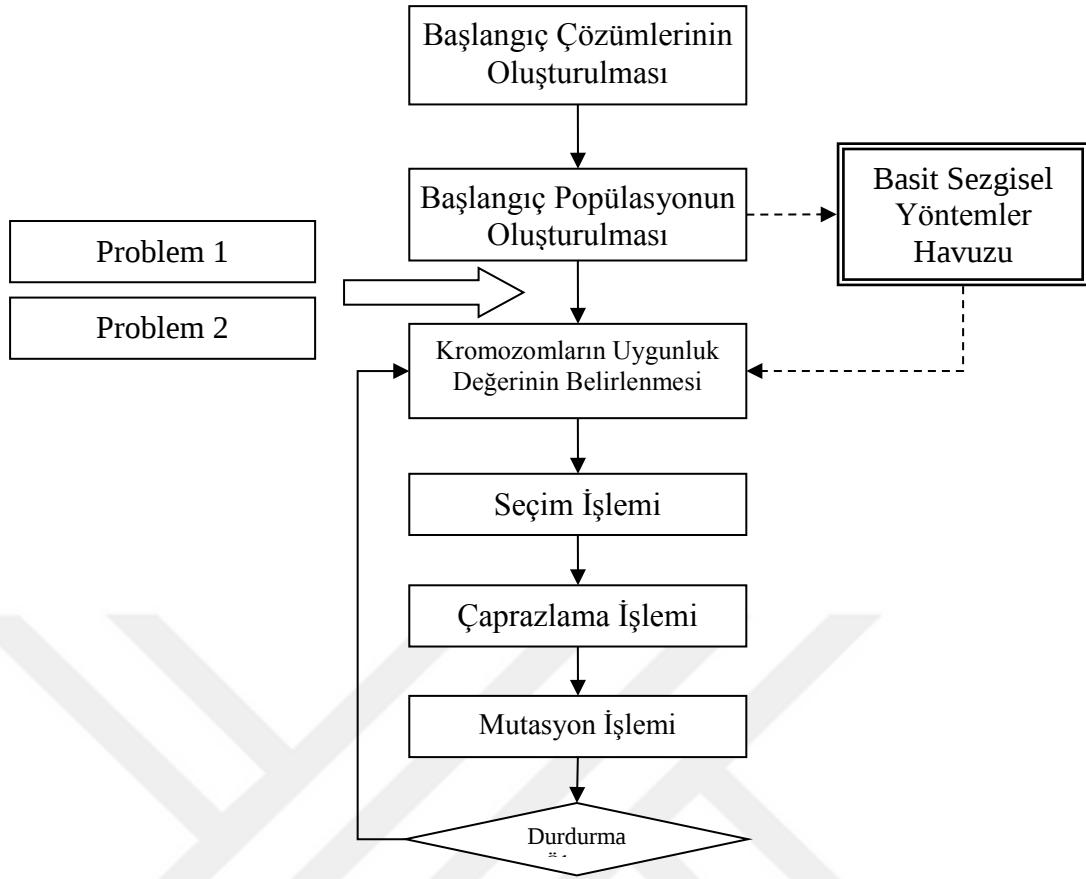


Şekil 4.1 : Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritma süreç adımları.

Hipersezgisel Algoritmalarda genellikle aşağıdaki basit sezgisel yöntemler kullanılır (Chakhlevitch ve Cowling, 2008):

- 2-opt. yer değiştirme sezgiseli
- Ekleme sezgiseli
- Çıkarma sezgiseli
- En yakın yer değiştirme sezgiseli

Bu sezgisel yöntemler problemden bağımsız olup, her sonlu kombinatoriyal probleme uygulanabilir. Bu sayede Hipersezgisel yöntemlerin bir problemin değiştirilmiş hâline uygulanması daha kolay olmaktadır. Hipersezgisel Algoritmalar, bazı problemler için Metasezgisel yöntemlerden daha kısa sürede çok daha iyi sonuçlar üretir (Cowling ve diğ, 2002).



Şekil 4.2 : Hiper-Genetik Algoritma süreç adımları.

Hipersezgisel Algoritmalar, çözüm uzayında kabul edilen ilerlemelerin türüne bağlı olarak aşağıdaki şekillerde de sınıflandırılabilir (Mısır ve diğ, 2013):

1. Salt iyileştirme gösteren hareketler kabul edilir (Sİ).
2. İyileştirme gösteren veya değişiklik yapmayan hareketler kabul edilir (İD).
3. Tüm hareketler kabul edilir (TH).
4. Monte-Carlo yöntemi (MC): Çözümde tüm iyileştirici hareketler kabul edilir, iyileşme sağlamayan hareketler ise belirli bir olasılığa göre kabul edilir.
5. Büyük Fırtına (Great Deluge) yöntemi (BF): Çözümde tüm iyileştirici hareketler kabul edilir, iyileşme sağlamayan hareketler ise kötüleşme miktarı belirli bir değer ile karşılaştırılarak kabul edilir.
6. Geç Kabul ölçütü (GK): Basit sezgisel ile bulunan yeni çözüm belirli bir durdurma ölçütüne kadar varolan çözüm üzerinde yinelemeli olarak uygulanarak iterasyonlar sonunda elde edilen en iyi çözüm alınmaktadır.

7. Turnuva yöntemi (TY): Klasik sezgisel yöntemle ilk olarak belirli sayıda komşu çözüm elde edilmeye çalışılır. Bu komşu çözümler içerisinde en iyisi seçilmekte ve varolan çözümden daha iyi ise çözüm kabul edilmektedir.
8. Tepe Tırmanma yöntemi (TT): Klasik sezgisel yöntem kullanılarak belirli bir iterasyon sayısı kısıtı altında komşu çözümler sırayla ele alınmaktadır. Daha iyi bir çözüm bulunması durumunda çözüm kabul edilmektedir.
9. Adaptif Liste Tabanlı İterasyon Limitli Eşik Kabulü (Adaptive iteration limited list-based threshold accepting) (ALE): Basit sezgisel yöntem ile çözüm arama uzayında varolan çözümden daha iyi olarak bulunan çözümler doğrudan kabul edilirken, algoritmada çeşitlendirme sağlamak için belirli bir iterasyon sonucunda daha iyi çözüm elde edilememesi durumunda kötü çözümler de seçilebilmektedir. Yöntemde ayrıca algoritma süresince bulunan en iyi çözüm değerleri bir listede tutulmakta, bu liste kötü çözümlerin seçilmesi durumunda eşik olarak kullanılmaktadır.
10. Tavlama Benzetimi (TB): Bir sıcaklık değeri ile algoritmaya başlanarak klasik yöntemlerin bulduğu çözümlerin kabul edilme durumu belirlenmektedir. Algoritma süresince bu sıcaklık değeri düşürülerek kötü çözümlerin seçilme olasılığı azaltılmaktadır.

Bunların yanında Hipersezgisel yöntemler, kullanılan basit sezgisel tekniklere göre ikiye ayrılmaktadır (Burke ve diğ, 2007):

1. İyileştirici Sezgiselleri Kullanan Hipersezgisel Algoritmalar: Çözümde belirli yerlerde değişiklik yaparak çözümü iyileştirmeye çalışan tekniklerdir.
2. Yapılandırıcı Sezgiselleri Kullanan Hipersezgisel Algoritmalar: Belirli parametrelere göre bir kerede çözüm üreten tekniklerdir.

Hipersezgisel Algoritmalar, sürecin işlediği aşamalara göre ise tek aşamalı ve çok aşamalı olmak üzere iki tiptir (Gaw ve diğ, 2004). Tek aşamalı Hipersezgisel yöntemlerde seçim işlemi, sezgisel yöntem havuzundan bir kerede yapılmakta, çok aşamalı Hipersezgisel yöntemlerde ise ilk olarak sezgisel yöntem grubu seçilmekte, daha sonra grup içerisinde uygulanacak sezgisel yöntem belirlenmektedir.

4.2 Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritmalar

Basit Hipersezgisel Algoritmalar karar aşamasında sezgisel yöntemleri rassal olarak seçen, uygulanması oldukça kolay yöntemlerdir. Rassal olarak seçilen klasik sezgisel teknikler, varolan çözümde yaptıkları değişiklikler dikkate alınarak süreç sürdürülür. Literatürde yapılan çalışmalarda Basit Hipersezgisel Algoritmalar, genellikle diğer Hipersezgisel Yöntemlerle karşılaştırılmak amacıyla kullanılmış ve daha az kaliteli sonuçlar üretmiştir. Çizelge 4.1’de, literatürde Basit Hipersezgisel Algoritmaların uygulandığı çalışmalar listelenmiştir.

Çizelge 4.1 : Basit Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Rassal Seçim Yöntemi	Yazarlar	Yıl	Uygulama Problemi	Kabul Ölçütü
Basit Rassal Seçim	Cowling, Kendall ve Soubeiga	2000	Satış Çizelgeleme	Sİ, TH
	Cowling ve diğerleri (2002b)	2002	Sunum Çizelgeleme	Sİ, TH
	Cowling ve diğerleri (2002c)	2002	Hemşire Çizelgeleme	Sİ, TH
	Bai ve Kendall	2003	Depo-Boşluk Planlama	Sİ, TH
	Ayob ve Kendall	2003	Parça Sıralama	MC
	Cowling ve Chakhlevitch	2003	Personel Çizelgeleme	Sİ, TH
	Kendall ve Mohamad	2004	Hücre – Kanal Atama	BF
	Burke, Silva ve Soubeiga	2005	Ders Programı Hazırlama	Sİ, TH
	Bilgin, Özcan ve Korkmaz	2006	Sınav Programı Hazırlama	Sİ, TH
	Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Sİ, TH, BF, MC
	Cowling, Kendall ve Soubeiga	2000	Satış Çizelgeleme	Sİ, TH

Çizelge 4.1 (devam) : Basit Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Rassal Seçim Yöntemi	Yazarlar	Yıl	Uygulama Problemi	Kabul Ölçütü
	Topcuoglu, Ucar ve Altin	2014	Genel Atama Problemi	TH, Sİ, MC
	Elhag ve Özcan	2015	Harita Renklendirme, Sınav Çizelgeleme	BF, ALE, GK
	Ahmed, Özcan ve Kheiri	2015	Ders Çizelgeleme Problemi	ALE, BF, Sİ, TB
	Alcaraz ve diğerleri	2016	Ders Çizelgeleme	Sİ, TT
	Li, Özcan ve John	2017	Çok Amaçlı Rüzgar Santrali Yerleşimi	BF, GK, Sİ, TH
	Cowling, Kendall ve Soubeiga	2001	Satış Personel Çizelgeleme	Sİ, TH
	Soubeiga	2003	Personel Çizelgeleme	Sİ, TH
	Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Sİ, TH, BF, MC
	Mısır ve diğerleri	2013	Çizelgeleme Problemleri	Sİ, TH, BF, ALE, TB, TT, TY
	Topcuoglu, Ucar ve Altin	2014	Genel Atama Problemi	TH, Sİ, MC
İyileştirici Rassal Seçim	Yahyaoui ve diğerleri	2015	İş Çizelgeleme Problemi	TH, Sİ, BF

Çizelge 4.1 (devam) : Basit Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Rassal Seçim Yöntemi	Yazarlar	Yıl	Uygulama Problemi	Kabul Ölçütü
Yinelemeli Rassal	Cowling, Kendall ve Soubeiga	2001	Satış Personel Çizelgeleme	Sİ, TH
	Bilgin, Özcan ve Korkmaz	2006	Sınav Programı Hazırlama	Sİ, TH
	Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Sİ, TH, BF, MC
	Topcuoglu, Ucar ve Altin	2014	Genel Atama Problemi	TH, Sİ, MC
Yinelemeli İyileştirici Rassal	Han, Kendall ve Cowling	2002	Eğitimci Çizelgeleme	Sİ, TH
	Bilgin, Özcan ve Korkmaz	2006	Sınav Programı Hazırlama	Sİ, TH
	Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Sİ, TH, BF, MC
	Mısır ve diğerleri	2013	Çizelgeleme Problemleri	Sİ, TH, BF, ALE, TB, TT, TY
	Topcuoglu, Ucar ve Altin	2014	Genel Atama Problemi	TH, Sİ, MC
	Kheiri ve diğerleri	2015	Su Ağ Tasarımı Problemi	Sİ, TH

Seçim Fonksiyonlu Hipersezgisel Algoritmalarla ise sezgisel yöntemler, belirli performans ölçütlerine göre seçilip probleme uygulanmaktadır. Bu performans ölçütleri, genel olarak çözüm değerinde yapılan değişiklik ve çözüm süresi ile ilişkilidir. Seçim Fonksiyonlu Hipersezgisel Algoritmalar, genel olarak Basit Hipersezgisel yöntemlerden daha kaliteli çözüm üretmektedir. Çizelge 4.2’de literatürde Seçim Fonksiyonlu Hipersezgisel Algoritmaların uygulandığı çalışmalar listelenmiştir.

Çizelge 4.2 : Seçim Fonksiyonlu Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Yazarlar	Yıl	Uygulama Problemi	Kabul Ölçütü
Cowling, Kendall ve Soubeiga	2000	Satış Çizelgeleme	Sİ, TH
Cowling, Kendall ve Soubeiga	2001	Satış Personel Çizelgeleme	Sİ, TH
Cowling ve diğerleri (2002b)	2002	Sunum Çizelgeleme	Sİ, TH
Cowling ve diğerleri (2002c)	2002	Hemşire Çizelgeleme	Sİ, TH
Kendall, Soubeiga ve Cowling	2002	Personel Çizelgeleme	Sİ, TH
Gaw, Rattadilok ve Kwan	2004	Ders Programı Hazırlama	Sİ, TH
Chakhlevitch	2006	Eğitimci Çizelgeleme	Sİ
Bilgin, Özcan ve Korkmaz	2006	Sınav Programı Hazırlama	Sİ, TH, İD, MC, BF
Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Sİ, TH, BF, MC
Kendall ve Li	2013	Rekabetçi Gezgin Satıcı Problemi	Sİ
Mısır ve diğerleri	2013	Çizelgeleme Problemleri	Sİ, TH, BF, ALE, TB, TT, TY
Topcuoglu, Ucar ve Altin	2014	Genel Atama Problemi	TH, Sİ, MC
Elhag ve Özcan	2015	Harita Renklendirme, Sınav Çizelgeleme	BF, ALE, GK
Maashi, Kendall ve Özcan	2015	Çok Amaçlı Araç Dayanıklılık Problemi	BF, GK
Yahyaoui ve diğerleri	2015	İş Çizelgeleme Problemi	TH, Sİ, BF
Ahmed, Özcan ve Kheiri	2015	Ders Çizelgeleme Problemi	ALE, BF, Sİ, TB
Sabar ve Kendall	2015	Çizelgeleme Problemleri	MC, Sİ, TT
Kheiri ve Özcan	2016	Çizelgeleme, Rotalama ve Sıralama Problemleri	ALE, Sİ
Alcaraz ve diğerleri	2016	Ders Çizelgeleme	Sİ, TT
Asta ve diğerleri	2016	Proje Çizelgeleme Problemi	Sİ, ALE, TH, MC

Çizelge 4.2 (devam) : Seçim Fonksiyonlu Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Yazarlar	Yıl	Uygulama Problemi	Kabul Ölçütü
Alcaraz ve diğerleri	2017	Ders Çizelgeleme	TH
Guizzo ve diğerleri	2017	Entegrasyon Test Sıralama Problemi	Sİ
Li, Özcan ve John	2017	Çok Amaçlı Rüzgar Santrali Yerleşim Problemi	BF, GK, Sİ, TH

Basit ve Seçim Fonksiyonlu Hipersezgisel Algoritmalar, ilk olarak Cowling, Kendall ve Soubeiga tarafından 2000 yılında Satış Çizelgeleme Problemi için yapılan çalışmada kullanılmıştır. Bu uygulamada düşük düzeyli sezgisel yöntemler seçilirken üç Hipersezgisel yaklaşım belirtilmiştir. İlk yaklaşımda sezgisel teknikler tümüyle rassal olarak yerel optimal sonuca ulaşmaya kadar seçilmektedir. Diğer seçim yönteminde (Greedy), bir iterasyonda tüm sezgisel yöntemler denenir ve bulunan en iyi sonuç uygulanır. En son yaklaşımda seçim işlemi için bir seçim fonksiyonu kullanılır. Bu seçim fonksiyonu, sezgisel yöntemlerin son çağrıldığındaki iyileştirme miktarı ve çalışma süresini dikkate alarak her sezgisel yönteme bir değer atar ve bu değere göre hangi yöntemin seçileceği belirlenir. Ayrıca algoritmada arama sürecinde yoğunlaştırma ve çeşitlendirme stratejisini uygulayabilmek için bir parametre yer almaktadır. Satış Çizelgeleme Probleminde amaç, belirli bir satış potansiyeline ulaşabilmek için en az maliyetle müşterilerin satış toplantı çizelgesini hazırlamaktır. Çalışmada seçim fonksiyonu ile oluşturulan algoritma, diğer seçeneklere göre daha iyi sonuç bulmuştur. Sezgisel yöntemler dört türlü rassal olarak seçilmektedir:

1. Basit Rassal: Rassal olarak bir sezgisel yöntem seçilir.
2. İyileştirici Rassal: Çözümde iyileştirme sağlanmayıncaya dek sezgiseller rassal olarak seçilir.
3. Yinelemeli Rassal: Başlangıçta sezgisellerin bir permütasyon sırası oluşturulup, her iterasyonda bu sıra sürdürülür.

4. Yinelemeli İyileştirici Rassal: Üçüncü yönteme benzer olarak, çözümde iyileştirme sağlanıncaya dek belirli bir sıra ile sezgiseller seçilir.

Yapılan çalışmada yazarlar üç durumu içeren parametrik özel bir seçim fonksiyonu geliştirmişler ve bu seçim fonksiyonu ile diğer fonksiyonları karşılaştırmışlardır. Bu üç durum aşağıda açıklanmaktadır:

- N_j sezgiseli için çözüm kalitesindeki iyileşme performansı (f_1) (4.2):

$$f_1(N_j) = \sum_n \alpha^{n-1} \left(\frac{I_n(N_j)}{T_n(N_j)} \right) \quad (4.2)$$

Formülde kullanılan notasyonlar aşağıda belirtilmiştir:

$I_n(N_j)$: N_j sezgiselinin n . iterasyondaki kullanımda çözümde yaptığı değişiklik.

$T_n(N_j)$: N_j sezgiselinin n . iterasyondaki kullanımda çalışma süresi.

α : 0 ile 1 arasındaki parametre olup N_j sezgiselinin performansına önceki iyileştirmelerin etkisini gösterir.

- Sezgisel çiftler için iyileşme performansı (f_2) (4.3):

$$f_2(N_j, N_k) = \sum_n \beta^{n-1} \left(\frac{I_n(N_j, N_k)}{T_n(N_j, N_k)} \right) \quad (4.3)$$

Formülde kullanılan notasyonlar aşağıda belirtilmiştir:

$I_n(N_j, N_k)$: N_j sezgiselinin N_k sezgiselinden hemen sonra çağırılması durumunda çözümdeki değişiklik.

$T_n(N_j, N_k)$: N_j sezgiselinin N_k sezgiselinden hemen sonra çağırılması durumunda çalışma süresi.

β : 0 ile 1 arasındaki parametre olup N_j sezgiselinin performansına önceki iyileştirmelerin etkisini gösterir.

- Sezgisel yöntemin son kullanıldığından şimdiye kadar geçen süre (f_3) (4.4):

$$f_3(N_j) = \text{Şimdiki zaman} - j. \text{ sezgiselin son kullanıldığı zaman} \quad (4.4)$$

Sonuçta seçim fonksiyonu aşağıdaki gibi olmaktadır (4.5):

$$f(N_k) = \alpha f_1(N_k) + \beta f_2(N_j, N_k) - \delta f_3(N_k) \quad (4.5)$$

Seçim yöntemindeki f_1 ve f_2 fonksiyonları algoritmanın yoğunlaştırma stratejisini, f_3 fonksiyonu ise çeşitlendirme stratejisini oluşturmaktadır. Seçim fonksiyonundaki α , β ve δ parametreleri ise 0 ile 1 arasında değer alıp sürecin yoğunlaştırma ve çeşitlendirme faaliyetini belirler. Sonuç olarak yazarların geliştirdiği seçim fonksiyonlu Hipersezgisel Algoritma, Satış Çizelgeleme Problemi için diğer seçim yöntemlerine göre daha iyi sonuçlar üretmiştir.

Cowling, Kendall ve Soubeiga tarafından 2001'de yapılan başka bir çalışmada Hipersezgisel Algoritmalar, Satış Personel Çizelgeleme Probleminin aynı problem örneklerinde kullanılmıştır. Daha önce hazırladıkları çalışmaya benzer olarak bir seçim fonksiyonu, sezgisel yöntemlerin daha önceki performanslarına göre seçilmesinde kullanılmaktadır. Ancak diğer çalışmadan farklı olarak bu çalışmada, parametrelere bir değer atanmasına gerek yoktur. Arama uzayında çeşitlendirme sağlamak amacıyla seçim fonksiyonunda daha önceki çalışmalarındaki parametre yerine, algoritma kendiliğinden bir ayarlama yapmaktadır. Bu sırada ilk olarak her sezgisel yöntem için yoğunlaştırma ve çeşitlendirme değeri, iterasyonlar boyunca yöntemin seçilip seçilmeme durumuna göre artırılıp azaltılmaktadır. Yazarlar, yöntemi, aynı yıl hazırladıkları bir diğer çalışma ile karşılaştırmışlar ve daha başarılı sonuçlar elde etmişlerdir.

Kendall, Soubeiga ve Cowling tarafından 2002 yılında Personel Çizelgeleme Problemi için Seçim Fonksiyonu kullanan bir Hipersezgisel algoritma geliştirilmiştir. Yapılan çalışmada ilk olarak rassal bir sezgisel fonksiyon seçilmekte ve çözümde iyileştirme sağlanamayınca dek yinelemeli bir şekilde uygulanmaktadır. İkinci uygulamada ise bir seçim fonksiyonu ile çözümde yapılan değişiklik, sezgiselin çalışma süresi ve sezgisel çiftlerin çözümde yaptığı değişiklikler dikkate alınarak bir sıralama yapılmaktadır. İki uygulama arasında sonuç karşılaştırması yapıldığında, Seçim Fonksiyonlu Hipersezgisel Algoritmanın daha iyi sonuçlar ürettiği görülmüş, bu da Hipersezgisel tekniklerin basit sezgisel yöntemlerle çözüm uzayı arasında dinamik bir öğrenme mekanizması oluşturduğunu göstermiştir.

Cowling, Kendall ve Soubeiga tarafından 2002 yılında Hipersezgisel yöntemler, İngiltere'de bir hastanede karşılaşılan Hemşire Çizelgeleme Problemine uygulanmıştır (Cowling ve diğ., 2002c). Üzerinde çalışılan problem, Personel

Çizelgeleme Probleminden farklı olarak personelin önceliklendirilmesi ve iş yükünün adil dağıtılması için çizelgenin dengeli olması durumları modele eklenmiştir. Seçim yöntemi olarak parametresiz seçim fonksiyonu kullanılmaktadır. Geliştirilen yöntem, aynı problem için Tabu Arama ve Genetik Algoritma ile karşılaştırılmış ve Çizelge 4.3'te gösterilen sonuçlar elde edilmiştir.

Çizelge 4.3 : Hipersezgisel Algoritmaların diğer yöntemlerle karşılaştırılması.

	Hipersezgisel Yöntemler	Saf Genetik Algoritma	Melez Genetik Algoritma	Tabu Arama
Uygun Çözüm Bulma Olasılığı	1,0	0,91	0,99	1,0
Ortalama Çözüm Süresi	<60 sn	15 sn	10 sn	30 sn

Cowling, Kendall ve Soubeiga tarafından 2002 yılında Hipersezgisel yöntemler, İngiltere’de bir akademik kurumda öğrencilerin sunum çizelgesi oluşturulurken uygulanmıştır (Cowling ve diğ, 2002a). Sunum çizelgeleme yapılırken sunumlara hem oda, hem de üç akademik personel atanması gerekmektedir. Problemden amaç olarak her akademik personele dengeli bir sunum plânı çıkartmak ve akademik personelin araştırma alanı ile ilgili sunumlara katılmasını sağlamak belirlenmiştir. Problem için iki tip Hipersezgisel Algoritma denenmiştir. İlkinde sezgisel yöntemler rassal olarak, ikincisinde ise daha önce kullanılan seçim fonksiyonu ile seçilmiştir. Seçim fonksiyonu kullanan algoritma ile daha iyi sonuçlar elde edilmiştir.

Ayob ve Kendall tarafından 2003 yılında Parça Sıralama Problemi için geliştirilen Hipersezgisel Algoritmada sezgisellerin kabul ölçütü olarak Monte-Carlo yöntemi kullanılmıştır. Parça sıralama probleminde amaç, en az zamanda çok başlıklı bir makina üzerinde işlenecek parça sırasını belirlemektir. Monte Carlo kabul ölçütünde çözümde iyileştirme sağlayan tüm hareketler kabul edilmekte, iyileşme sağlamayan hareketler ise belirli olasılıklara göre kabul edilmektedir. Bu olasılık doğrusal, üstel ve sayaçlı üstel olmak üzere üç farklı yöntemle hesaplanabilmektedir. Doğrusal Monte Carlo kabul ölçütünde, rassal üretilen bir değişkenin $(M-\delta)$ değerinden küçük olup olmaması dikkate alınmaktadır. Burada M değeri algoritmanın bir parametresi, δ ise üretilen çözüm ile varolan çözüm arasındaki farktır. Üstel Monte Carlo kabul ölçütünde rassal değişken $(e-\delta)$ değerinden küçük ise sezgisel yöntem kabul edilir.

Sayaçlı Üstel Monte Carlo kabul ölçütünde ise olasılıklar iterasyonlar boyunca azaltılır, eğer belli iterasyonlar süresince çözümde iyileşme sağlanmazsa arttırılır. Geliştirilen algoritma, daha önce kullanılan TH ve Sİ gibi kabul ölçütleri ve seçim fonksiyonu dahil edilerek örnekler üzerinden karşılaştırılmış ve TH kabul ölçütü en kötü, Sayaçlı Üstel Monte Carlo ise en iyi çözümleri üretmiştir.

Kendall ve Mohamad tarafından 2004 yılında yapılan bir çalışmada Hipersezgisel yöntemler, mobil iletişim sektöründe hücrelere kanal frekansı atama problemine uygulanmıştır. Hücresel iletişim sistemlerinde her hücreye farklı bir iletişim kanalı atanmalı ve özellikle komşu hücreler için atanan frekanslar arasında belirli farklılıklar olmalıdır. Yazarlar, bu çalışmaları ile (Great Deluge) Büyük Fırtına adında yeni bir kabul ölçütü geliştirmişlerdir. Bu ölçüte göre iyileştirme sağlayan tüm hareketler kabul edilmekte, çözümde kötüleşmeye yol açan sezgiseller için kabul durumuna ise kötüleşme miktarı belirli bir değerle karşılaştırılarak karar verilmektedir. Ayrıca iterasyonlar süresince bu değer düşürülmektedir. Geliştirilen algorithmada Basit Rassal Seçim yöntemi kullanılmıştır.

Gaw, Rattadilok ve Kwan'ın 2004 yılında yaptığı çalışmada Ders Programı Hazırlama ve Çizelgeleme Problemleri için dağıtık seçim fonksiyonu kullanan Hipersezgisel bir yöntem önerilmiştir. Yapılan çalışmanın amacı, yeni bir seçim yöntemi ortaya koyarak Hipersezgisel yöntemlerde paralel işlemeye olanak tanımak ve uygulanabilirliğini arttırmaktır. Yeni seçim yönteminde sezgisel yöntemler yoğunlaştırıcı, çeşitlendirici ve koşullandırıcı sezgisel olarak üç sınıfa ayrılmıştır. Koşullandırıcı sezgisel, çözümde herhangi bir iyileştirmeyi amaçlamayan ama diğer sezgiseller için potansiyel durumlar yaratan sezgisel tipidir. Bu durumda çok katmanlı hipersezgisellik durumu ortaya çıkmaktadır. Paralel bilgisayar mimarîlerinin de yardımıyla iki dağıtık Hipersezgisel Algoritma denenmiştir. İlkinde farklı işlemciler, bir problemin farklı noktalarında basit sezgisel süreçleri çalıştırmakta, diğer yöntemde ise problem, farklı işlemcilerde ayrı ayrı belleklerde çözülmeye çalışılmaktadır.

Bilgin, Özcan ve Korkmaz, 2006 yılında Hipersezgisel Algoritmelerde daha önce kullanılan kabul ölçütleri ve seçme yöntemlerini, literatürde bulunan Sınav Programı Hazırlama Problem örnekleri üzerinde denemişler ve sonuçları, farklı amaç fonksiyonları üzerinden karşılaştırmışlardır. Seçim yöntemleri olarak Basit Rassal, İyileştirici Rassal, Yinelemeli Rassal, Yinelemeli İyileştirici Rassal, Seçim

Fonksiyonu, Tabu Arama ve tüm sezgisellerin denenmesi (Greedy) ele alınmıştır. Sezgisel tekniklerin uygulanmasında kabul ölçütü olarak ise çözümde salt iyileştirme sağlayan (Sİ), çözümdeki her hareket (TH), çözümde iyileştirme sağlayan veya değişiklik yapmayan (İD), Büyük Fırtına (Great Deluge) ve Monte Carlo (MC) ölçütleri kullanılmıştır. Çözüm değerlerine göre, hiç bir sezgisel seçim yöntemi kabul ölçütü kombinasyonu, diğer kombinasyonlara açık ve belirgin bir üstünlük kuramamıştır. Seçim Fonksiyonu – Monte Carlo simülasyonu, performansı en iyi ikili olmuştur. Tek tek incelendiğinde ise, kabul ölçütü olarak çözümde iyileştirme sağlayan veya değişiklik yapmayı (İD) ve seçim yöntemi olarak ise Seçim Fonksiyonu, diğerlerine göre daha iyi sonuçlar üretmiştir.

Maashi, Kendall ve Özcan tarafından 2015 yılında geliştirilen seçim fonksiyon tabanlı Hipersezgisel Algoritma'da çok amaçlı optimizasyon problemleri için Büyük Fırtına (BF) ve Geç Kabul (GK) ölçütü denenmiştir. Geç Kabul ölçütünde basit sezgisel algoritma ile bulunan yeni çözüm belirli bir durdurma ölçütüne kadar varolan çözüm üzerinde yinelemeli olarak uygulanarak iterasyonlar sonunda elde edilen en iyi çözüm alınmaktadır. Çalışmada literatürde bulunan sanal problemler üzerinde yöntem denenmiş, aynı zamanda araç dayanıklılık problemi üzerinde gerçek yaşamdan uygulama gerçekleştirilmiştir. Karşılaştırmada BF kabul ölçütü, GK'ya göre daha iyi çözümler üretmiştir.

4.3 Metasezgisel Yöntemlerle Melezleştirilen Hipersezgisel Algoritmalar

Hipersezgisel Algoritmalarla hangi sezgisel tekniğin kullanılacağına karar verilirken; Genetik Algoritmalar, Tabu Arama ve Tavlama Benzetimi gibi Metasezgisel Yöntemler kullanılabilir. Metasezgisel Yöntemler, çözüm uzayında çalışan yerel arama tabanlı yöntemler olup, temel özelliği, yerel optimum noktalardan uzaklaşmayı sağlayan operatörlere sahip olmasıdır. Fakat literatürde “Hipersezgisel” terimi geçmeyen ama Metasezgisel Algoritmaların sezgisel yöntemleri yönettiği çalışmalar da bulunmaktadır. Bu çalışmalarda çözüm uzayında optimum çözümü aramak yerine, problemi çözecek etkili bir yöntem araştırılmaktadır. Bu nedenle algoritmik süreç Hipersezgisel bir mantık taşımaktadır. Sözügeçen bu çalışmalar da bu başlık altında incelenecektir. Çizelge 4.4'te literatürde yer alan Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar ve Hipersezgisel özellik gösteren Metasezgisel yöntemler özetlenmiştir.

Çizelge 4.4 : Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Fang, Ross ve Corne	1994	Açık İş Çizelgeleme	Dolaylı Genetik Algoritma
Storer, Wu ve Vaccari	1995	Atölye Tipi İş Çizelgeleme	Problem ve Sezgisel Uzaylı Metasezgisel Yöntemler
Dorndorf ve Pesch	1995	Atölye Tipi İş Çizelgeleme	Dolaylı Genetik Algoritma
Hart, Ross ve Nelson	1998	İş Emri Bölme ve İş Atama	Dolaylı Genetik Algoritma
Hart, Ross ve Nelson	1999	İş Çizelgeleme	Dolaylı Genetik Algoritma
Norenkov ve Goodman	1997	Çok Aşamalı Atölye Tipi İş Çizelgeleme	Sezgisel Birleştirme Yöntemi
Hart ve Ross	1998	Atölye Tipi İş Çizelgeleme	Dolaylı Genetik Algoritma
Terashima-Marin ve diğerleri	1999	Sınav Programı Hazırlama	Dolaylı Genetik Algoritma
Cowling ve diğerleri (2002b)	2002	Eğitimci Çizelgeleme	Hiper - Genetik Algoritma
Han, Kendall ve Cowling	2002	Eğitimci Çizelgeleme	Adaptif Hiper - Genetik Algoritmalar
Han ve Kendall	2003	Sunum Çizelgeleme	Klavuzlu Adaptif Hiper - Genetik Algoritma
Soubeiga	2003	Personel Çizelgeleme	Tavlama Benzetimi ile Melez Hipersezgisel Algoritma
Burke ve Soubeiga	2003	Hemşire Çizelgeleme	Tabu Arama ve Takviyeli Öğrenme ile Melez Hipersezgisel Algoritma
Burke, Kendall ve Soubeiga	2003	Ders Programı ve Hemşire Çizelgeleme	Tabu Arama ile Melez Hipersezgisel Algoritma
Cowling ve Chakhlevitch	2003	Eğitimci Çizelgeleme	Tabu Arama ile Melez Hipersezgisel Algoritma

Çizelge 4.4 (devam) : Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Kendall ve Mohd Hussin	2003	Sınav Programı Hazırlama	Tabu Arama ile Melez Hipersezgisel Algoritma
Kendall ve Mohd Hussin	2004	Sınav Programı Hazırlama	Tabu Arama ile Melez Hipersezgisel Algoritma
Dowsland, Soubeiga ve Burke	2004	Kutu Paketleme	Tabu Arama ve Tavlama Benzetimi ile Melez Hipersezgisel Algoritma
Burke ve diğerleri (2005c)	2005	Sunum Çizelgeleme	Karınca Kolonisi Yöntemi ile Melez Hipersezgisel
Burke, Dror, Petrovic ve Qu	2005	Ders Programı Hazırlama	Tabu Arama ve Durum Tabanlı Çıkarsama ile Melez Hipersezgisel Algoritma
Cuesta, Garrido ve Terashimar	2005	Kutu Paketleme	Karınca Kolonisi Yöntemi ile Melez Hipersezgisel
Qu ve Burke	2005	Sınav Programı Hazırlama	Değişken Komşuluk Arama Hipersezgisel Algoritması
Burke, Silva ve Soubeiga	2005	Ders Programı ve Oda Yerleştirme	Tabu Arama ile Melez Hipersezgisel Algoritma
Bilgin, Özcan ve Korkmaz	2006	Sınav Programı Hazırlama	Tabu Arama ile Melez Hipersezgisel Algoritma
Chakhlevitch	2006	Eğitimci Çizelgeleme	Tavlama Benzetimi ile Melez Hipersezgisel Algoritma
Burke ve diğerleri	2007	Ders Programı Hazırlama	Tabu Arama ile Melez Hipersezgisel Algoritma
Dowsland, Soubeiga ve Burke	2007	Taşıyıcı Boyutları Belirleme	Tabu Arama ve Tavlama Benzetimi ile Melez Hipersezgisel Algoritma
Cowling ve Chakhlevitch	2007	Eğitimci Çizelgeleme	Tabu Arama ile Melez Hipersezgisel Algoritma
Kumar ve diğerleri	2008	0-1 Sırt Çantası Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma

Çizelge 4.4 (devam) : Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Özcan, Bilgin ve Korkmaz	2008	Karşılaştırma Fonksiyonları	Tabu Arama ile Melez Hipersezgisel Algoritma
Garrido ve Castro	2009	Kapasite Kısıtlı Araç Rotalama	Tepe Tırmanma ile Melez Hipersezgisel Algoritma
Leon, Miranda, Segura	2009	Kutu Paketleme	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Remde ve diğerleri	2009	İşgücü Çizelgeleme	Tabu Arama ile Melez Hipersezgisel Algoritma
Burke ve diğerleri	2010	Sınav Programı Hazırlama	Değişken Komşuluk Arama ve Genetik Algoritma ile Melezleştirilmiş Hipersezgisel Algoritma
Raad, Sinske ve Vuuren	2010	Su Ağ Tasarımı Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Rafique, Kamran ve Zeeshan	2011	Uydu Fırlatma Aracı Tasarlama	Genetik Algoritma, Parçacık Sürü ve Tavlama Benzetimi ile Melezleştirilmiş Hipersezgisel Algoritma
Kampouridis, Alsheddy ve Tsang	2012	Finansal Tahmin Problemi	Genetik Algoritma ile Melezleştirilmiş Hipersezgisel Algoritma
Moreno ve diğerleri	2013	Hava Sıcaklığı ve Ozon Tahmini	Genetik Algoritma ile Melezleştirilmiş Hipersezgisel Algoritma
Koulinas ve Anagnostopoulos	2013	Proje Çizelgeleme Problemi	Tabu Arama ile Melezleştirilmiş Hipersezgisel Algoritma
Rajni ve Chana	2013	Dağıtık Bilgi İşlemde Kaynak Çizelgeleme	Bakteri Besin Arama ile Melezleştirilmiş Hipersezgisel Algoritma

Çizelge 4.4 (devam) : Metasezgisel yöntemlerle melezleştirilen Hipersezgisel Algoritmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Koulinas, Kotsikas ve Anagnostopoulos	2014	Proje Çizelgeleme Problemi	Parçacık Sürü ile Melezleştirilmiş Hipersezgisel Algoritma
Camacho ve diğerleri	2014	Kutu Paketleme	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Thomas ve Chaudhari	2014	İki Boyutlu Paketleme	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Aziz	2015	Gezgin Satıcı Problemi	Karınca Kolonisi ile Melezleştirilmiş Hipersezgisel Algoritma
Dong, Jiao ve Wu	2015	Kaynak Çizelgeleme	Harmoni Arama ile Melez Hipersezgisel Algoritma
Beyaz, Dokeroglu ve Cosar	2015	İki Boyutlu Paketleme	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Poyato ve diğerleri	2015	Enerji Çizelgeleme Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Zamli, Alkazemi ve Kendall	2016	T-Yönlü Entegrasyon Testi Problemi	Tabu Arama ile Melez Hipersezgisel Algoritma
Kumari ve Srinivas	2016	Çok Amaçlı Gruplama Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Dokeroglu ve Cosar	2016	Atama Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Asta ve diğerleri	2016	Proje Çizelgeleme Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Guizzo ve diğerleri	2017	Entegrasyon Test Sıralama Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma
Lin, Wang ve Li	2017	Atölye Tipi İş Çizelgeleme Problemi	Genetik Algoritma ile Melez Hipersezgisel Algoritma

Geleneksel olarak Metasezgisel yöntemlerde problem bileşenleri sıraya konularak, gruplanarak veya birbiriyle ilişkilendirilerek bir çözüm oluşturulmaktadır. Bu tür çözüm yapısına Doğrudan Çözüm Kodlama adı verilmektedir. Ancak gerçek yaşamda karşılaşılan problemler oldukça karmaşık olduğu için, Doğrudan Çözüm Kodlama hem zor olmakta hem de fazla kaynağa gereksinim duymaktadır. Ayrıca çözümlerin uygulanabilirliğini sağlamak için uygunluk (onarım) operatörleri, çözümleri kontrol etmelidir. Bu nedenlerden ötürü 1994 yılında Bilgisayar Destekli Tasarım plânlama süreci için Norenkov tarafından ve açık iş çizelgeleme problemi için Fang ve diğerleri tarafından Dolaylı Çözüm Kodlama yöntemi geliştirilmiştir. Bu kodlama türü, çözümleri değil de, çözümleri oluşturacak yolları içerir. Yapılan çalışmalarda kromozomlar, başlangıç çözümüne, ilgili sezgisel yöntemin hangi sırada uygulanacağı bilgisini tutmaktadır.

Fang ve diğerleri tarafından 1994 yılında geliştirilen Genetik Algoritmada, Açık (Akış Tipi) Çizelgeleme Problemi için Dolaylı Çözüm Kodlama yöntemi kullanılmıştır. Bir çözümü oluşturmak için 8 düşük düzeyli sezgisel yöntem, belirli sıralarda kromozom içerisinde kodlanmıştır. Çalışmada bir kromozom, ilki hangi sezgisel yöntemin uygulanacağı, ikincisi ise tamamlanamayan iş bilgisinin tutulduğu gen çiftlerinden oluşmaktadır. Açık Çizelgeleme Problemi için amaç, toplam yayılım süresini enazlamak olup, Çizelge 4.5'te çalışmadan elde edilen sonuçlar özetlenmiştir. Çizelgeye göre geliştirilen GA, iki problem kümesi için daha önce bilinen optimum çözümden daha iyi çözümler üretmiştir.

Çizelge 4.5 : Dolaylı Genetik Algoritma çözüm sonuçları.

Problemler	4×4	5×5	7×7	10×10	15×15	20×20
Bilinen En İyi Çözüm	193	300	438	645	937	1155
Saf Genetik Algoritma (Ortalama Değer)	194,4	308,5	454,1	690,7	968,9	1244,5
Saf Genetik Algoritma (En İyi Değer)	193	302	441	668	951	1224
Dolaylı Genetik Algoritma (Ortalama Değer)	194,1	305,3	449,8	660,1	940,1	1167,7
Dolaylı Genetik Algoritma (En İyi Değer)	193	300	(435)	(641)	937	1156

Dorndorf ve Pesch tarafından 1995 yılında yapılan bir çalışmada Dolaylı Genetik Algoritma, Atölye Tipi İş Çizelgeleme Problemi için kullanılmıştır. Belirlenen 12 öncelik, kural tabanlı sezgisel bir havuz içerisinde toplanarak seçim işlemi yapılmıştır. Bir kromozom, n operasyon sayısını vermek üzere $n-1$ genden oluşmakta olup, bir gen bir operasyonu çizelgeleyecek şekilde öncelik kuralını kodlamaktadır.

Storer ve diğerleri tarafından 1995 yılında Atölye Tipi İş Çizelgeleme problemi için uygulanan yöntemde arama işlemi, problem ve sezgisel uzay olmak üzere iki farklı uzayda yapılmaktadır. Problem uzayındaki arama işlemi, bir sezgisel yöntemin problemin operasyon süreleri farklı olan değişik versiyonları için verdiği çözüm ile gerçekleşmektedir. Bu çözümler daha sonra varolan probleme uygulanarak, çözümün kalitesi anlaşılmaktadır. Sezgisel uzayda ise Tavlama Benzetimi, Tabu Arama ve Genetik Algoritmalar, sezgisel yöntemlerin uygulama sıralaması aşamasında kullanılmaktadır.

Norenkov ve Goodman'ın 1997 yılında Sezgisel Birleştirme Yöntemi adını verdikleri teknikte Çok Aşamalı Atölye Tipi İş Çizelgeleme Problemi için Dolaylı Kodlama tekniği kullanılmıştır. Geliştirdikleri yöntemde ilk olarak iş sırası belirlenmekte, daha sonra işlerin makinalara atanması yapılmaktadır. Kromozomların uzunluğu sabit olup, bir matris olarak kodlanmıştır. Ayrıca yapılan çalışmada Genetik Algoritmaya benzer çaprazlama ve mutasyon operatörleri kullanılmıştır.

Hart ve Ross tarafından 1998 yılında yapılan bir çalışmada Atölye Tipi Dinamik İş Çizelgeleme Problemi için Sezgisel Kılavuzlu Genetik Algoritma geliştirilmiştir. Bu yöntemde Dorndorf ve Pesch tarafından 1995 yılında yapılan çalışma ilerletilerek her gen, yöntem ve sezgisel olmak üzere iki kısım hâline getirilmiştir. Yöntem kısmında her iterasyondaki çakışan operasyonları hesaplayan yöntemler temsil edilirken, sezgisel kısımda ise bu çakışan operasyonları çizelgeleyen 12 öncelik kuralından biri simgelenmektedir.

Hart ve diğerleri tarafından 1998 ve 1999 yılında geliştirilen Dolaylı Genetik Algoritma, bir tavuk işleme merkezinde iş plânlama için kullanılmıştır. İlk olarak verilen iş emirleri alt operasyonlara ayrılmakta, daha sonra bu operasyonlar takımlara atanmaktadır. Geliştirilen kodlama tekniğinde, bir kromozom 4 bölümden oluşmaktadır. İlk bölümde probleme özgü temel ölçüt bilgileri, ikinci bölümde iş emirlerinin işlenme sırası tutulmaktadır. Üçüncü ve dördüncü bölümde ise iş

emirlerinin bölünmesini ve atanmasını gerçekleştiren sezgisel yöntemlerin uygulanma sırası tutulmaktadır. Algoritma, tüm kısıtların sağlandığı, hazırlanması zor olan uygulanabilir iş çizelgelerini oluşturabilmektedir.

Terashima-Marin ve diğerleri tarafından 1999 yılında gerçek yaşamda karşılaşılan sınav çizelgeleme problemi için Dolaylı Genetik Algoritma geliştirilmiştir. Algoritmada farklı sezgisel yöntem kombinasyonları bir arada tutularak, uygulanabilir çözüm elde edilmeye çalışılmıştır. Bu çalışmada kromozom içerisinde çözüm uzayı hakkında bilgiler tutulduğu için, diğer optimizasyon problemlerine uygun olmayabilmektedir.

Cowling, Kendall ve Han, 2002 yılında Eğitimci Çizelgeleme Problemi için Genetik Algoritmalarla melezleştirdikleri Hipersezgisel Algoritmaları kullanmışlardır (Cowling ve diğerleri, 2002b). Eğitimci Çizelgeleme Probleminde belirli sayıda eğitimci, eğitim merkezi ve zaman programı bulunmaktadır. Kurulan modelde öncelikli eğitimlerin yapılması ve eğitimcilerin kendilerinden uzakta bulunan eğitim merkezlerine atanmaması amaçlanmaktadır. İlk olarak, problemi doğrudan çözen bir Genetik Algoritma geliştirilmiştir. Daha sonra hazırlanan Hiper-GA modelinde ise Genetik Algoritmalar, sezgisel yöntemlerin seçilmesinde kullanılmaktadır. Süreç içinde mutasyon ve çaprazlama oranı, iterasyonlar boyunca uygunluk fonksiyonundaki iyileşmelere göre uyarlamalı (adaptif) olarak değiştirilmekte veya sabit tutulmaktadır. Hiper-GA modelinde, her iterasyonda sabit sayıda bir grup sezgisel yöntem çözüme uygulanmakta ve kromozomlar sezgisel yöntemleri temsil eden tamsayı dizisinden oluşmaktadır. Sezgisel yöntemler, kromozomda yer alan sıra ile ele alınmakta ve daha iyi sonuç elde edilmeye çalışılmaktadır. Geliştirilen yöntem, gerçek yaşamdaki probleme uygulanmış ve uyarlamalı olmayan Hiper-GA, daha iyi sonuçlar üretmiştir.

Han, Kendall ve Cowling, 2002 yılında yaptıkları başka bir çalışmada Eğitimci Çizelgeleme Problemi için Adaptif Parametrelili Hiper-GA yönteminin, değişken uzunlukta kromozom ile uygulamasını yapmışlardır. ALC-Hiper-GA adını verdikleri yöntemin en büyük başarısı, kromozomun uzunluğunun iterasyonlar süresince kendiliğinden belirlenmesidir. Çalışmada ayrıca en iyi sezgisellerin çaprazlanması, en kötünün kaldırılması mutasyonu ve en iyi ekleme mutasyonu olmak üzere üç yeni operatör tasarlanmıştır.

Han ve Kendall tarafından 2003 yılında yapılan diğer çalışmada Genetik Algoritmalarla adaptif uzunluklu kromozomlarda çaprazlama ve mutasyon operatörleri için kılavuzluk mekanizması önermişlerdir. Sunum Çizelgeleme Problemi üzerinde uygulanan yöntemde daha önce yaptıkları çalışmadan farklı olarak kromozom uzunluğu, önceki kuşaklardaki kromozomların ortalama uzunluğundan büyük ise, en kötü sezgisel kromozomdan çıkartılır, küçük ise en iyi sezgiseller eklenir. Böylece hem algoritma daha hızlı çalışmakta, hem de bir önceki çalışmaya göre daha kaliteli çözümler üretmektedir.

Tüm bu çalışmalara ilişkin uygulama sonuçları, Çizelge 4.6'da özetlenmiştir.

Çizelge 4.6 : Hiper-Genetik Algoritmalar çözüm değerleri.

	Klavuzlu Hiper-GA	ALC- Hiper-GA	Hiper-GA	En İyi Sezgisel	Genetik Algoritma	Memetik Algoritma
Çözüm Değeri	1972	1961,64	1959,09	1556,75	1796,19	1832,14
Çözüm Süresi	1184 sn	1357 sn	1456 sn	0,16 sn	1628 sn	2064 sn

Han ve Kendall, 2003 yılında Eğitimci Çizelgeleme Problemi için Tabu Arama ile melezleştirilmiş Adaptif Parametrelili Hiper - Genetik Algoritma kullanan bir Hipersezgisel yöntem önermişlerdir. Genetik Algoritmalar kullanılarak 2002 yılında yapılan çalışmaya benzer olarak bu uygulamada sürece, Tabu Arama yordamı eklenmiştir. Basit iyileştirici sezgisellerin çözümde yaptıkları değişikliklere göre bir sezgiselin kaç iterasyonda Tabu listesinde kalacağı belirlenmektedir. Genetik Algoritmada Tek Noktalı Çaprazlama ve Mutasyon operatörü kullanılmış ve özellikle TA kullanılarak kısa sürelerde başarılı sonuçlar elde edilmiştir. Bu çalışmayla birlikte yapılan Hipersezgisel süreçler problemten bağımsız olup, diğer problem kümelerine de kolay bir şekilde uygulanabilmektedir.

Ross ve diğerleri tarafından 2003 yılında yapılan bir çalışmada, şimdiye kadar yapılan çalışmalardan farklı olarak bir kromozomun uygunluk değeri salt bir problem üzerinden değil, literatürde yer alan birkaç örnek üzerinden belirlenmektedir. Geliştirilen Dolaylı Genetik Algoritmada, genler problemin durumu ve uygulanan sezgisel yöntem ile ilgili bilgileri içermektedir. Süreçte ilk adım olarak literatürden alınan problem kümeleri üzerinden en iyi kromozomlar oluşturulmakta, daha sonra

bu kromozomlar bir havuz içerisindeki problemlere uygulanmaktadır. Yazarlara göre algoritma iyi sonuç verse de, sürecin gerçek yaşamdaki problemlere uygulanması zor olduğu için Hipersezgisel mantığa tam olarak uymamaktadır.

Burke ve Soubeiga'nın 2003 yılında yaptığı çalışmada yazarlar, daha önce üzerinde çalıştıkları Hemşire Çizelgeleme Problemi için Tabu Arama ile melezleştirilmiş bir Hipersezgisel yöntem geliştirmişlerdir. Algoritmada sezgisel yöntemlerin seçilmesinde Takviyeli Öğrenmeye dayalı dinamik bir tabu listesi kullanılmıştır. Takviyeli Öğrenmeye dayalı süreçte, başlangıçta her basit sezgisel yöntemin sıfır puanı olmakta, süreç ilerledikçe yöntemin performansına göre puan artırılıp azaltılmaktadır, böylece 9 adet sezgisel yöntem birbirleri ile yarış içerisinde. Ayrıca algoritmada yer alan negatif takviye oranı ile, arama işleminin çeşitlendirilme özelliği belirlenmektedir. Yüksek puanlı bir sezgisel yöntem sürekli seçileceğinden, seçim sonucunda iyileştirme sağlamayan bir sezgisel tabu listesine koyulmakta ve çözümde değişiklik oluncaya dek tabu olarak kalmaktadır. Geliştirilen TA bazlı algoritma ile bulunan sonuçlar, seçim fonksiyonu kullanan Hipersezgisel yöntemlerle karşılaştırılmış ve daha kısa sürelerde benzer çözümler ürettiği belirtilmiştir. Yapılan denemelerde; algoritma verimliliğinin, tabu listesi uzunluğundan ve negatif takviye oranından çok fazla etkilenmediği görülmüş ve bu da, geliştirilen algoritmanın dayanıklı olduğunu kanıtlamıştır.

Burke ve diğerlerinin 2003 yılında yaptığı çalışmada Ders Programlama ve Hemşire Çizelgeleme Problemi için Tabu Arama ile melezleştirilmiş Hipersezgisel bir algoritma geliştirilmiştir (Burke ve diğerleri, 2003b). Kullanılan tabu listesi değişken boyutta olup, belirli zamanlarda belirli sezgisellerin seçilmesini önlemiştir. Çözümde iyileşme sürdüğü sürece tabu listesinde sezgiseller tutulmakta, çözümde herhangi bir iyileşme sağlanamadığı takdirde tabu listesinden sezgiseller “ilk giren ilk çıkar” mantığı ile çıkarılmaktadır. Yazarlar bu şekilde bir çalışma yaparak, iki farklı tipteki problemi, farklı sezgisel yöntemlerle, aynı algoritma üzerinde çözme olanağını kazanmışlar ve kısa sürelerde Çizelge 4.7’de gösterilen özet sonuçları elde etmişlerdir.

Çizelge 4.7 : Melez Hipersezgisel Algoritma karşılaştırma sonuçları.

	Melez Hipersezgisel Yöntemler	Yerel Arama Sezgiseli	Karınca Kolonisi Yöntemi
Uygun Çözüm	1.0	0.95	1.0
Bulma Olasılığı			
Ortalama Çözüm Kalitesi	0,64	0	0,37

Kendall ve Mohd Hussin tarafından 2003 yılında Sınav Çizelgeleme problemi için Tabu Arama tabanlı Hipersezgisel Algoritma geliştirilmiştir. Yapılan çalışmada Hipersezgisel yapı içerisinde tabu listesi, 13 adet düşük düzeyli sezgisel yöntemin performansını izlemek ve çok fazla sayıda kullanılan sezgiselin yeniden ele alınmasını engellemek için kullanılmıştır. Eğer düşük düzeyli sezgisel bir yöntem tabu listesine girerse, belirli bir iterasyon süresince bu listede kalmaktadır. Geliştirilen algoritma diğer yöntemlerle karşılaştırılmış ve başarılı sonuçlar elde edilmiştir.

Kendall ve Mohd Hussin 2004 yılında Sınav Çizelgeleme problemi için bir önceki yılki çalışmalarını daha da geliştirerek Tabu Arama tabanlı Hipersezgisel Algoritma ile ilgili iki yeni versiyon sunmuşlardır. İlk versiyonda, varolan iterasyonda en yüksek iyileştirmeyi yapan düşük düzeyli sezgisel, tepe tırmanma sürecine benzer bir şekilde iyileşme olmayıncaya kadar yeniden çeşitli kerelerce kullanılmış ve bu işlem sonucunda tabu listesine konulmuştur. İkinci versiyonda ise, tabu listesine varolan çözüm üzerinde belirli sınırlar dahilinde en kötü etkiyi yapan sezgisel yöntem atılmıştır (çözümünden çıkarılmıştır). Ayrıca, sezgiselin yöntemin tabu listesinde ne kadar süre kalacağı rassal olarak belirlenmiştir. Yazarlar, algoritmanın bir önceki çalışmalarından daha iyi sonuçlar ürettiğini belirtmişlerdir.

Dowsland, Soubeiga ve Burke tarafından 2004 yılında Kutu Paketleme Problemi için, Tabu Arama ve Tavlama Benzetimi ile melezleştirilmiş Hipersezgisel Algoritma geliştirilmiştir. Amaç, paket kullanım oranını enbüyüklemek için paketlerin boyutlarını belirlemektir. Kurulan modelde ayrıca, ağırlık kısıtı da dikkate alınmış ve Tavlama Benzetimi mantığı ile bâzı durumlarda kötü çözümler de kabul edilmiştir. Yazarlar, yeni kısıtlar eklenmesi durumunda, modelde çok fazla değişikliğe gerek olmadığını belirtmişlerdir.

Burke ve diğerleri tarafından Sınav Programı Hazırlama problemi için 2005 yılında iki sezgisel yöntemin kullanıldığı Tabu Arama ve Durum Tabanlı Muhakemeye dayanan Hipersezgisel bir algoritma geliştirilmiştir (Burke ve diğerleri, 2005a). Durum Tabanlı Çıkarsama yaklaşımında Sınav Programı Hazırlama problemi için daha önce çözülmüş problem kümeleri, kullanılan sezgiseller ve stratejiler baz alınarak varolan problem çözülmeye çalışılmaktadır. Geliştirilen algorithmada başlangıç çözümü oluşturmak yerine, yapılandırıcı sezgisellerle çizelge adım adım oluşturulmuştur. Harita Çizelgeleme Probleminde kullanılan En Büyük ve Doyurulma Derecelendirme sezgiselleri, tabu listesinde yer alan sezgisellerin sıralanması için kullanılmıştır. Kullanılan yöntemler, problemler için daha önce hazırlanan algoritmalarla karşılaştırılmış ve ortalama sonuçlar elde edilmiştir.

Qu ve Burke tarafından 2005 yılında yapılan bir çalışmada Sınav Çizelgeleme problemi için, Değişken Komşuluk Arama yöntemi ile melezleştirilmiş Hipersezgisel Algoritma geliştirilmiştir. Değişken Komşuluk Arama yönteminde salt belirli bir komşuluk tipi benimsenmeyip, varolan çözümden komşu çözüme farklı kurallarla geçiş yapılmaktadır (Hansen ve Mladenovic, 2001). Belirli bir uzunlukta hazırlanan sezgisel yöntem uygulama dizileri ile, bu sıra içindeki yan yana duran sezgiselleri rassal olarak yer değiştirerek çözüm geliştirilmeye çalışılmaktadır. Diğer bir deyişle, çalışmada sözedilen komşuluk, çözümlerin komşuluğu değil, uygulanan basit sezgisel yöntem dizisinin komşuluğudur. Yazarlar, geliştirdikleri yöntemde diğer Hipersezgisel Algoritmalarla göre başarı kaydedememişlerdir.

Cuesta ve diğerleri tarafından 2005 yılında Öklidyen Uzayda Kutu Paketleme problemi için Karınca Kolonisi Algoritması ile melezleştirilmiş bir Hipersezgisel Algoritma geliştirilmiştir. Kutu Paketleme probleminde dikdörtgen bir kutu içine taşma olmadan yine dikdörtgen şeklindeki çeşitli boyutlardaki cisimler yerleştirilmeye çalışılmaktadır. Problemde amaç, kullanılan kutu sayısını ve kutu içindeki boşluk miktarını enazlamaktır. Yazarlara göre Karınca Kolonisi Algoritması problem büyüklüğünden bağımsız olarak sezgisel uzayda çözüm uzayından daha kolay bir şekilde arama yapabilmektedir. Geliştirilen yöntem, literatürde bulunan problemlere uygulanmış ve sezgisel yöntemlerle karşılaştırıldığında daha kaliteli çözümler üretmiştir.

Burke ve diğerleri tarafından 2005 yılında yapılan bir çalışmada, Çok Amaçlı Ders Programı ve Oda Yerleştirme problemleri için 2003 yılında Burke ve diğerleri

tarafından geliştirilen Tabu Arama tabanlı Hipersezgisel yöntem uygulanmıştır (Burke ve diğ., 2005b). Oda Yerleştirme problemi, belirli kısıtlara göre bir kuruluştaki kişilere odaların atanması problemidir. Geliştirilen algoritmada her iterasyonda salt bir amaç aktif yapılarak, süreç çok amaçlı duruma getirilmekte ve her amaç için bir tabu listesi kullanılmaktadır. Amaç, Pareto Optimizasyon yaklaşımı ile çözüm uzayında düzgün olarak dağılmış çözümler üretmektir. Hangi amaç fonksiyonunun seçileceği, rassal olarak ve Rulet Tekerleği yöntemi ile belirlenmektedir.

Burke ve diğerlerinin 2005 yılında yaptıkları çalışmada, daha önce uygulamasını yaptıkları Sunum Çizelgeleme Problemi için Karınca Kolonisi ile melezleştirilmiş Hipersezgisel yöntem geliştirilmiştir (Burke ve diğerleri, 2005c). Kurulan modelde Karınca Kolonisi basit sezgisel yöntemlerin uygulanma sırasını göstermekte olup, sezgisel yöntemler, karıncaların ziyaret edeceği noktalara karşılık gelmektedir. Her basit sezgisel yöntemin feramon miktarı (η), daha önce seçim fonksiyonlu Hipersezgisel yöntemlerdekine benzer olarak aşağıdaki formüle göre güncellenmektedir (4.6).

$$\eta_j(t) = \gamma \eta_j(t - m) + \sum_k^m \frac{I_{kj}(t)}{T_{kj}(t)} \quad (4.6)$$

Formülde $I_{kj}(t)$ değeri, k. karıncanın t. iterasyonda uğradığı j. sezgiselin çözümde yaptığı değişikliği, $T_{kj}(t)$ ise bu sezgiselin çalışma süresini göstermektedir. m parametresi çözümdeki son iyileştirmeden sonraki iterasyon sayısını, $\eta_j(t - m)$ çözümdeki son iyileştirme anındaki sezgisel yöntemlerin feramon miktarını ve γ parametresi ise feramon yenileme oranını göstermektedir. Geliştirilen algoritma, Basit Hipersezgisel ve Seçim Fonksiyonlu Hipersezgisel yöntemlerle karşılaştırılmış ve üç problem kümesinden ikisinde başarılı sonuçlar elde edilmiştir.

Chakhlevitch'in 2006 yılında yaptığı çalışmada Eğitimci Çizelgeleme Problemi için Tavlama Benzetimi ile melezleştirilmiş Hipersezgisel Algoritma geliştirilmiştir. Algoritma tarafından üretilen çözümler salt Tavlama Benzetimi sonuçları ile karşılaştırılmış ve daha iyi çözümlerin elde edildiği görülmüştür. Bunun yanında yapılan denemelerde Hipersezgisel Algoritma performansının Tavlama Benzetimine göre başlangıç çözümüne daha az bağlı olduğu anlaşılmıştır.

Burke ve diğerleri tarafından 2007 yılında Ders Programı Hazırlama Problemi için geliştirilen Tabu Arama tabanlı Hipersezgisel yöntemde Harita Çizelgeleme Probleminde kullanılan sezgiseller uygulanmıştır. Çizge tabanlı bu sezgiseller iyileştirici sezgiseller olmayıp, başlangıçtan itibaren tümüyle yeni bir çözüm oluştururlar. Tabu Arama, basit sezgisellerin seçilmesinde kullanılmaktadır. Çalışmanın sonuçlarına göre ne kadar fazla sezgisel yöntem kullanılırsa, Hipersezgisel Algoritmanın optimuma yakın çözümler üretme olasılığı o kadar artmaktadır. Geliştirilen algoritma çok aşamalı olup çizge tabanlı sezgiselden sonra iyileştirici sezgisel de uygulanmaktadır.

Dowsland, Soubeiga ve Burke'un 2007 yılında yaptıkları çalışmada, lojistikte yer kullanımının enbüyüklenmesinin amaçlandığı konteyner boyutlarının belirlenmesi problemi için, Tabu Arama ve Tavlama Benzetimi ile melezleştirilen bir Hipersezgisel yöntem geliştirilmiştir. Üzerinde uygulanan problemde farklı fiziksel özelliklere sahip n ürünü taşımak ve depolamak için boşlukların en az kalacak şekilde taşıyıcı veya konteyner boyutlarını bulmak amaçlanmıştır. Yazarlar, algoritmanın çalışma süresini azaltmak için uygunluk fonksiyonunda değişikliğe gitmişler, çözümlerin komşu çözümlerini ve çözüm uzayını daraltmışlar ve en son bulunan çözümde iyileştirme yapmışlardır. Geliştirilen Hipersezgisel Algoritmada TB yaklaşımı, bir çözümden diğer çözüme geçmek için seçilecek komşuluk yordamlarının seçilmesi için kullanılmıştır. Sürece ayrıca yoğunlaştırma ve çeşitlendirme stratejileri için bir tabu listesi eklenmiştir. Yazarlar, çalışmalarında, problem özelliklerinde ve çözüm uzayında değişiklikler oluştuğunda Hipersezgisel Algoritmaların kolay bir şekilde uyum sağladığını belirtmişlerdir.

Cowling ve Chakhlevitch tarafından 2003 ve 2007 yıllarında Eğitimci Çizelgeleme problemi için geliştirilen Tabu Arama tabanlı Hipersezgisel Algoritma, çok sayıdaki düşük düzeyli sezgiseli yönetmektedir. Yapılan çalışmalarda tabu listesi son zamanlarda çağırılan ve çözümde iyileşme sağlamayan sezgisel yöntemleri tutmaktadır. Algoritma, her iterasyonda tüm sezgiselleri tek tek deneyerek (Greedy), en iyi çözümü uygulamaktadır. Eğer en iyi çözüm tabu listesindeyse, tabu listesinden çıkartılır. Yazarlar, tabu listesini sabit ve değişken tutarak algoritmayı yürütmüşler ve başarılı sonuçlar elde etmişlerdir.

Kumar ve diğerleri tarafından 2008 yılında çok amaçlı 0-1 Sırtçantası (Knapsack) problemi için hazırlanan çalışmada Genetik Algoritmalarla melezleştirilmiş

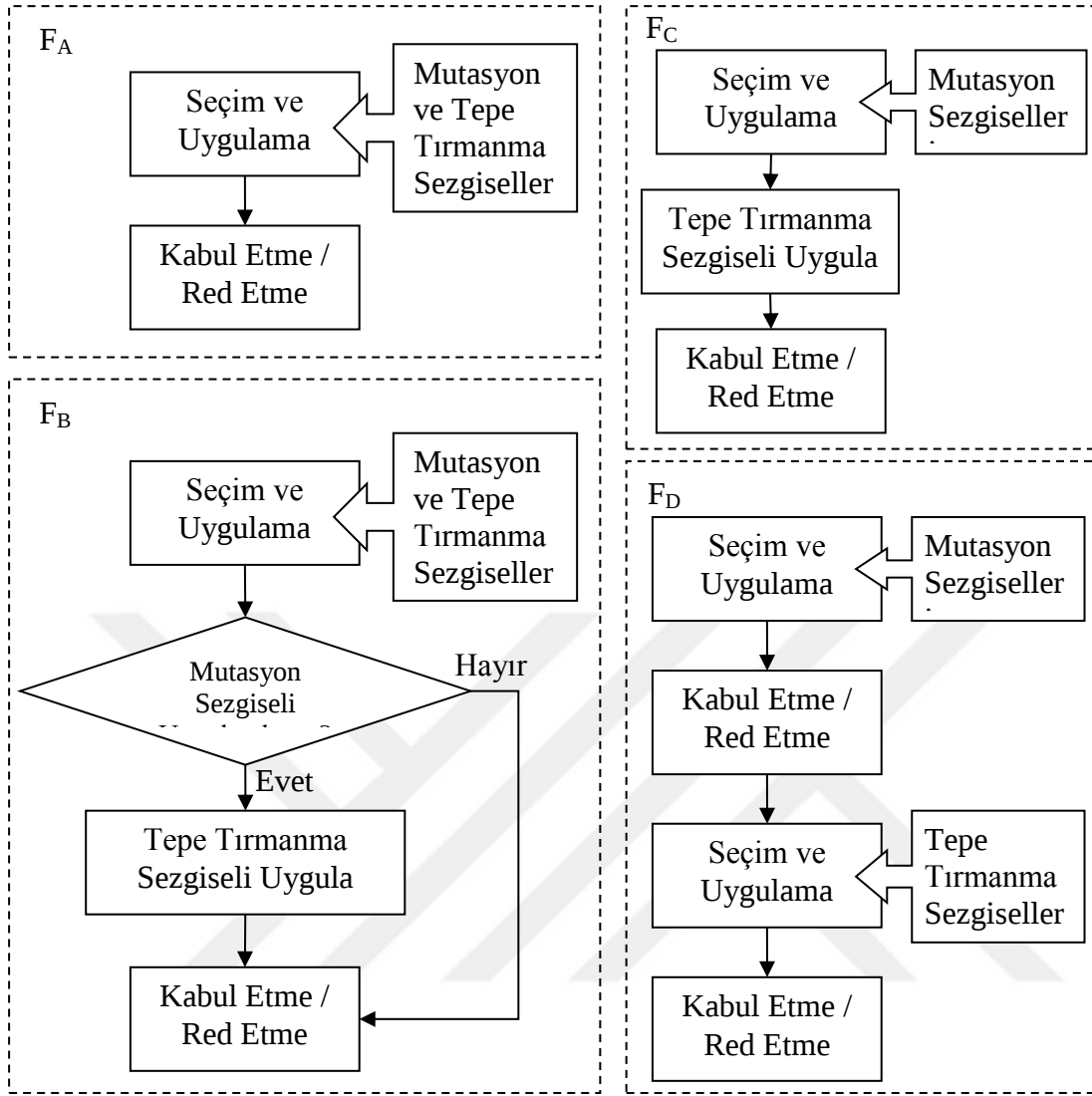
Hipersezgisel Algoritmalar kullanılmıştır. Oluşturulan modelde, belirli değerleri ve ağırlıkları olan cisimler bir sırt çantasından seçilmektedir. Problemin amacı, elde edilen toplam değeri enbüyükleyip, toplam ağırlığı enküçükmektir. Bu çalışma ile, Genetik Algoritmaların yerel optimum sonuçlar üretebilen sezgisel yöntem kombinasyonlarını geliştirebileceği görülmüştür. Burada sözedilen sezgisel yöntemler, bir cismin seçilip seçilmeyeceğini belirleyen ve Pareto çözüm üreten yordamlardır. Yazarlar, üretilen sonuçların, daha önce varolan sezgisel yöntemlerle aynı olduğunu göstermişlerdir.

Özcan, Bilgin ve Korkmaz tarafından 2008 yılında yapılan bir çalışmada, çeşitli karşılaştırma fonksiyonları kullanarak Hipersezgisel Yöntemler için 7 seçim yöntemi ve 5 kabul ölçütü test edilmiştir. Seçim yöntemleri olarak rassal, seçim fonksiyonu, Tabu Arama ve tüm sezgisellerin denenmesi; kabul ölçütleri olarak ise tüm hareketler (TH), salt iyileştirme sağlayan (Sİ), iyileştirme sağlayan veya çözümü değiştirmeyen (İD), sayaçlı üstel Monte Carlo (MC) ve Büyük Fırtına (BF) yöntemleri denenmiştir. Bunların yanında, kullanılan sezgisel yöntemlerin türlerine göre 4 farklı Hipersezgisel yapı ortaya konmuştur. Yazarlar sezgisel yöntemleri tepe tırmanıcı ve mutasyon sezgiselleri olarak ikiye ayırmışlar ve bu sezgisellerin kullanılma durumlarına göre Şekil 4.3'te gösterilen FA, FB, FC ve FD süreçlerini ortaya koymuşlardır. Tepe tırmanıcı sezgisellerde amaç, yoğunlaştırma stratejisi ile iyi bir sonuç elde etmek iken, mutasyon sezgiselleri çeşitlendirme stratejisi ile rassal hareket ederek çözümü değiştirmeye çalışmaktadır. FA geleneksel olarak kullanılan Hipersezgisel süreç olup, diğerleri yazarlar tarafından geliştirilmiştir. Problem kümelerinde en başarılısı FC süreci olup, daha sonra FB, FA ve FD gelmektedir. Seçim yöntemi-kabul ölçütü çiftine bakıldığında ise en iyi çift, seçim fonksiyonu ve iyileşme sağlayan veya çözümü değiştirmeyen kabul ölçütü olmuştur. Tabu Arama kullanan Hipersezgisel ise ancak bir problemde en iyi sonuçları üretmiştir. Ayrıca elde edilen sonuçlar Saf Genetik ve Memetik Algoritmalar ile kıyaslanmış ve Hipersezgisel kullanan Memetik Algoritmalar diğer yöntemlere göre en başarılı sonuçları kısa sürede üretmiştir.

Garrido ve Castro'nun 2009 yılında yaptıkları çalışmada Kapasite Kısıtlı Araç Rotalama Problemleri için Tepe Tırmanma yaklaşımı bir Hipersezgisel Algoritma geliştirmişlerdir. Tepe Tırmanma yaklaşımı ile belirli bir süre içerisinde varolan çözümünden iyileştirici sezgisellerle rassal bulunan komşu çözümler incelenir. Sonraki

iterasyonda yeniden yapılandırıcı sezgisellerle bir çözüm üretilip süreç sürdürülmekte ve en sonunda bulunan çözümler karşılaştırılarak en iyi çözüm saklanmaktadır. Yapılan çalışmada iyileştirici sezgisellerin uygulanma sırasını tasarlayan işbirlikli bir süreç ortaya konulmuştur. Yazarlar, problem değişse bile bu yapının çözüm uzayına kolayca uyarlanabileceği ve istikrarını sürdüreceğini iddia etmişlerdir. Geliştirilen algoritma, literatürde bulunan problem kümelerine uygulanmış, ortalamanın üzerinde iyi sonuçlar üretmiş ve aynı zamanda defalarca çalıştırılrsa dahi varyansı düşük, sürekli optimuma yakın çözüm değerleri ortaya koymuştur.

Leon ve diğerleri, 2009 yılında Öklidyen düzlemde Kutu Paketleme problemi için Genetik Algoritmalarla paralel, Hipersezgisel Algoritmaları kullanmışlardır (Leon ve diğerleri, 2009a). Yararlandıkları ada tabanlı paralelleştirme yönteminde, popülasyon alt gruplara bölünüp bu gruplar üzerinden Genetik Algoritma süreci paralel bir şekilde ilerletilmektedir. Algoritmanın Hipersezgisel mantığı ise çaprazlama ve mutasyon operatörlerinin uygulanması üzerinedir. Yapılan çalışmada çaprazlama ve mutasyon işlemleri birer sezgisel olarak görülüp, çözümleme, işleme sırası Hipersezgisel Algoritma süreci ile sürdürülmektedir. Genetik ve Evrimsel Hesaplama Konferansında yer alan yarışmada Kutu Paketleme problemi için en iyi sonuçları üretmiştir. Yazarların 2009 yılında yaptığı diğer çalışmada, algoritma parametrelerinin de kendiliğinden ayarlandığı ada tabanlı paralelleştirme yöntemi kullanılmıştır (Leon ve diğerleri, 2009b).



Şekil 4.3 : Hipersezgisel Algoritmalar için oluşturulan yapılar.

Remde ve diğerleri tarafından 2009 yılında gerçek yaşamdaki İşgücü Çizelgeleme problemi için Tabu Arama ile Melezleştirilen Hipersezgisel Algoritma geliştirilmiştir. Yapılan çalışmanın önceki çalışmalardan farkı, düşük düzeyli sezgiselleri tabu listesine dinamik olarak atan, bu listede ne kadar kalacağını dinamik olarak belirleyen ve ağ iletişimde sık sık kullanılan İkili Üstel Geri Çekme yönteminin kullanılmış olmasıdır. Algoritmanın ürettiği sonuçlar; Değişken Komşuluk Arama tekniği, tüm sezgisellerin denendiği yöntem, rassal ve Tabu Arama ile melezleştirilmiş diğer Hipersezgisel yöntemlerin çözümleri ile karşılaştırılmış ve daha başarılı olduğu görülmüştür. Gerçek yaşamdaki problem için işlemler uzun süreceğinden, paralel programlama tekniğine başvurulmuş ve 22 işlemcili bilgisayar sistemi üzerinde denemeler yapılmıştır.

Burke ve diğeri tarafından 2010 yılında Sınav Programı Hazırlama problemi için yapılan çalışmada Qu ve Burke'un 2005 yılında yaptığı çalışma ileriye itilerek Değişken Komşuluk tabanlı ve Genetik Algoritma tabanlı bir yöntem geliştirilmiştir. Çalışmada iyileştirici basit sezgisel yöntemler, komşuluk türleri olarak kullanılmış ve bu sezgiseller kodlanarak Genetik Algoritmalar ile yeni çözümler elde edilmiştir. Algoritmanın yürütülmesi uzun sürse de, oldukça başarılı çözümler üretmiştir.

Metasezgisel tabanlı Hipersezgiseller oldukça kaliteli çözümler üretse de, Metasezgisel yöntemlerde kullanılan parametrelerden dolayı, tam olarak bağımsız yöntemler değildirler. Fakat Hipersezgisel Algoritmalar sezgisel yöntem uzayında hareket ettiği için, parametre değişimlerine Metasezgisel Yöntemlerden daha az duyarlıdır (Chakhlevitch ve Cowling, 2008).

4.4 Öğrenme Tabanlı Hipersezgisel Algoritmalar

Öğrenme Tabanlı Hipersezgisel Algoritmalar, basit sezgisel yöntemlerin daha önceki performanslarını dikkate alarak, hangi yöntemin seçileceğine karar veren yöntemlerdir. Bu yöntemlerin en önemli özelliği parametreden bağımsız olmalarıdır. En fazla kullanılan öğrenme mekanizması, Takviyeli Öğrenme denilen, çözümde iyileştirme sağlayan sezgisellerin ödüllendirildiği tekniktir. Takviyeli Öğrenme süresince ödüllendirilen sezgisel yöntemlerin puanları arttırılarak seçilme olasılıkları yükseltilmektedir (Kaelbling ve diğ, 1996). Takviyeli Öğrenme, sezgisel yöntemler için ilk olarak, Fisher ve Thompson tarafından, 1963 yılında, Atölye Tipi İş Çizelgeleme Problemi için kullanılmıştır. Ayrıca Lagoudakis ve Littman tarafından 2000 yılında geliştirilen yöntem, sezgisel yöntem seçmek yerine, Takviyeli Öğrenme ile algoritma seçerek Sıralama problemleri için çözüm geliştirmektedir. Literatürde Öğrenme tabanlı Hipersezgisel Algoritmalar ile ilgili çalışmalar Çizelge 4.8'de listelenmektedir.

Ross ve diğeri tarafından 2002 yılında hazırlanan çalışmada, bir boyutlu Kutu Paketleme Problemi için bir Hipersezgisel Algoritma geliştirilmiştir. Kutu Paketleme Probleminde farklı büyüklükteki nesneler, olanaklı en az sayıdaki kutuya yerleştirilmeye çalışılmaktadır. Kullanılan basit sezgiseller çözüm iyileştirici yöntemler olmayıp, kural tabanlıdır ve bir kerede baştan itibaren yeni bir çözüm üretmektedir. Yapılan çalışmada XCS Öğrenme Sınıflandırma yöntemi kullanılarak sezgiseller gruplanmış ve literatürde bulunan problem kümelerine uygulanmıştır.

Oluşturulan algoritma, daha önce hazırlanan çalışmalardan daha iyi performans elde etmiş ve %78 oranında optimum çözümü üretmiştir.

Çizelge 4.8 : Öğrenme Tabanlı Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Fisher ve Thompson	1963	Atölye Tipi İş Çizelgeleme Problemi	Takviyeli Öğrenme
Lagoudakis ve Littman	2000	Sıralama Problemi ve İstenilen Sıradaki Elemanı Verme	Takviyeli Öğrenme
Burke ve diğerleri	2002	Sınav Programı Hazırlama	Bilgi çıkarımı ile Durum Tabanlı Çıkarsama
Petrovic ve Qu	2002	Ders Programı Hazırlama	Bilgi çıkarımı ile Durum Tabanlı Çıkarsama
Ross ve diğerleri	2002	Kutu Paketleme	XCS Öğrenme Sınıflandırma
Burke ve diğerleri (2003c)	2003	Ders Programı Hazırlama	Bilgi çıkarımı ile Durum Tabanlı Çıkarsama
Burke ve Soubeiga	2003	Hemşire Çizelgeleme	Tabu Arama ve Takviyeli Öğrenme
Nareyek	2003	Kısıt Sağlama Problemi	Takviyeli Öğrenme
Chakhlevitch ve Cowling	2005	Eğitimci Çizelgeleme Problemi	Çeşitli Öğrenme Stratejileri
Chakhlevitch	2006	Eğitimci Çizelgeleme Problemi	Çeşitli Öğrenme Stratejileri
Pham ve Ho	2007	Makina Öğrenmesi	Ağırlıklı Örtü Algoritması
Bittle ve Fox	2009	Harita Renklendirme ve Atölye Tipi İş Çizelgeleme	Kısıt Tabanlı Muhakeme
McClymont ve diğerleri	2013	Su Ağ Tasarımı	Takviyeli Öğrenme

Çizelge 4.8 (devam) : Öğrenme Tabanlı Hipersezgisel Algoritmalar ile ilgili çalışmalar.

Yazarlar	Yıl	Uygulama Problemi	Uygulama Yöntemi
Alcaraz ve diğerleri	2014	Ders Programı Hazırlama	Statik ve Dinamik Adaptif Öğrenme
Elhag ve Özcan	2015	Harita Renklendirme, Sınav Çizelgeleme	Takviyeli Öğrenme

Burke ve diğerleri tarafından 2002 yılında Sınav Programı problemi için hazırlanan Hipersezgisel Algoritma, Durum Tabanlı Muhakemeye dayanmaktadır. Süreç içerisinde yapılandırıcı sezgiseller kullanılmakta olup, çözüm adım adım oluşturulmaktadır. Her adımda uygun sezgisel çözüm bir durum tabanından alınmaktadır. Durum tabanı çözümün belirli bir parçasına veya duruma karşılık, uygun düşük düzeyli sezgiselleri barındıran bir havuzdur. Diğer bir deyişle bir öneri sistemidir. Durum tabanı, algoritma çalıştırılmadan önce belirli bir eğitim kümesi çözülerek oluşturulmaktadır, daha sonra test kümesi üzerinde uygunluk testleri yapılmaktadır. Algoritma her adımda sezgisel yöntemi seçerken, varolan yapıya en yakın durumu, durum tabanında belirler. Yazarlar, geliştirilen algoritmanın başarılı sonuçlar ürettiğini belirtmişlerdir.

Petrovic ve Qu tarafından 2002 ve Burke ve diğerleri (2003c) tarafından 2003 yılında ders programı oluşturmak için geliştirilen Hipersezgisel Algoritmada Durum Tabanlı Çıkarsama ile sezgisel yöntemlerin seçimi yapılmaktadır. Bilgi çıkarımı ile Durum Tabanlı Çıkarsama (Case Based Reasoning) daha önceki problemlerden kazanılan deneyimleri temel alarak, şu anki probleme benzer çözümler üretmeye çalışan iki aşamalı iteratif bir süreç olup, durum temsili ve durum tabanının oluşturulmasına dayanmaktadır. Çok aşamalı bir atama problemi olan ders programı oluşturma sürecinde ise öğrenci ve öğretmenler ilgili zaman aralıklarına ve dersler ise ilgili sınıflara atanmaktadır. Ders programı hazırlama ile ilgili literatürde oldukça fazla sezgisel olduğu için Durum Tabanlı Çıkarsama için oldukça iyi bir bilgi kaynağı oluşmaktadır. Durum temsiliinde özellik-değer çiftleri bulunmakta ve Tabu Arama ve Tepe Tırmanma yöntemleri, özelliklerin aranmasında kullanılmaktadır.

Nareyek tarafından 2003 yılında yapılan bir çalışmada Kısıt Sağlama Problemi (Constraint Satisfaction Problem) için sezgisel yöntemlerin seçilmesinde ağırlıklar Takviyeli Öğrenme tabanlı olarak hesaplanmaktadır. Sezgisel yöntemler her

kullanıldığında bu ağırlıklar değişmekte ve bir sonraki iterasyonda seçim olasılıkları farklılaşmaktadır. Eğer sezgisel yöntem varolan çözümde bir iyileştirme yaptıysa, sezgisel yöntemin ağırlığı arttırılmakta, aksi durumda azaltılmaktadır. Bu ağırlıkların bir alt ve bir üst limiti bulunmakta olup, çalışmada sezgisel yöntemleri seçmek için iki yöntem uygulanmıştır. İlk yöntemde, sezgisel yöntemler Rulet Tekerleği yöntemine benzer şekilde, ağırlıkları ile orantılı olarak rassal seçilmektedir. İkinci yöntemde ise, en yüksek ağırlığa sahip olan sezgisel yöntem seçilmektedir. Kısıt Sağlama Probleminde, sürekli ve kesikli değişkenlerin yer aldığı bir modelde belirli kısıtlar sağlanarak, amaç fonksiyonu enbüyüklenmeye çalışılmaktadır. Geliştirilen algoritma, gerçek yaşamda karşılaşılan problemler için başarılı sonuçlar üretmiştir.

Cowling ve Chakhlevitch'in 2005 yılında ve Chakhlevitch'in 2006 yılında yaptığı çalışmalarda, Eğitimci Çizelgeleme Problemi için çok sayıdaki düşük düzeyli sezgiselin seçimini yapan öğrenme stratejileri geliştirilmiştir. Yazarlara göre algoritmanın hızlı bir şekilde sonuç vermesi için tüm sezgiselleri denemek yerine, probleme özel sezgisellerin havuz içerisinden seçilip uygulanması gerekmektedir. Bu nedenle, sezgisel havuz içerisinden bazı sezgiselleri çıkarmak, sürecin performansını arttıracaktır. Geliştirilen algorithmada iki öğrenme tekniği denenmiştir. İlk yöntemde, ilk olarak tüm sezgisel yöntemlerin problem üzerinde performansları incelenmiş ve yüksek performans gösteren sezgiseller seçilmiştir. Daha sonra Hipersezgisel süreç, seçilen sezgiseller üzerinden yürütülmüştür. İkinci teknikte ise algoritma ilerlerken performansı düşük bazı sezgisel teknikler sürekli olarak havuzdan çıkartılmıştır. Yapılan denemelerde öğrenme stratejileri daha önceki yöntemlerden daha başarılı sonuçlar üretmiş olup ikinci teknik genellikle daha kaliteli çözümler ortaya koymuştur.

Pham ve Ho tarafından 2007 yılında hazırlanan çalışmada Ağırlıklı Örtü Algoritması Hipersezgisel yöntemlerde basit arama sezgisellerinin oluşturulması için kullanılmıştır. Ağırlıklı Örtü Algoritması, Makina Öğrenmesinde tümevarımsal bir yöntem olup, eğitim kümelerinde ayırım yaparak kural çıkarımı yapmakta ve kalan örnekler için yinelemeli bir şekilde ilerlemektedir. Bu şekilde doğrusal ve doğrusal olmayan sezgisel yöntemler oluşmaktadır. Yapılan çalışmada veri kümeleri üzerinde çalıştırılan Hipersezgisel Algoritma ile oluşturulan arama sezgiselleri ile kural çıkarımı yapılmış ve özellikle doğrusal olmayan sezgiseller için başarılı sonuçlar elde edilmiştir.

Bittle ve Fox tarafından 2009 yılında Harita Renklendirme ve Atölye Tipi İş Çizelgeleme problemi için hazırlanan çalışmada Kısıt Tabanlı Çıkarsama ile Hipersezgisel Algoritmalarla sezgisel seçim işlemi yapılmıştır. Kısıt Tabanlı Çıkarsama ile çözüm uzayında komşu çözümler, kısıtlar sağlandığı sürece sezgisel yöntemlerle incelenebilmektedir. Bu sürece Sınırlı Sezgisel Arama denilmiş olup, bu öğrenme mekanizması ile düşük düzeyli sezgisel yöntemlerin tek tek performansına değil, birbirleriyle ilişkili olan performans değerlerine de bakılabilmektedir. Bu sâyede seçim süreci daha da etkin duruma gelmekte, Hipersezgisel Algoritmaların performansı artmaktadır.

4.5 Hipersezgisel Algoritmalarla İlişkin Çalışmaların Özeti

Son yıllarda Hipersezgisel Algoritmalar, özellikle gerçek yaşamdaki optimizasyon problemlerinin çözülmesine yeni bir bakış açısı getirmektedir. Hipersezgisel Algoritmaların amacı, optimum çözüme çok yakın sonuçlar üretmekten ziyade, her çeşit optimizasyon problemi için sonuç üretebilen bir çözüm süreci oluşturmaktır (Chakhlevitch ve Cowling, 2008). Literatür araştırmasına göre Hipersezgisel Algoritmalar dört genel kategoride incelenebilir (Garrido ve Castro, 2009):

1. Hipersezgiseller basit sezgiselleri seçen sezgisel yöntemlerdir, varolan çözüm üzerinde iyileştirme adımlarını belirlerler (Burke ve diğ, 2003a).
2. Hipersezgiseller, daha önceki sezgisellerden yararlanarak baştan itibaren yapılandırıcı bir sezgisel yaratırlar.
3. Hipersezgiseller, önceki bilgilerden ve çözüm uzayının topolojisinden yararlanıp Takviyeli Öğrenme (Reinforcement Learning) ve Metasezgisel Algoritmalarla hangi basit sezgisellerin seçileceğine dinamik olarak karar verirler (Soubeiga, 2003).
4. Hipersezgiseller, Durum Tabanlı Çıkarsama (Case Base Reasoning) yöntemi ile daha önceki deneyimlerden yola çıkılarak oluşturulmuş genel kuralları temel alıp, basit sezgiselleri statik olarak seçerler (Burke ve diğ, 2005c).

Metasezgisel Algoritmalar, sezgisel yöntemleri aşağıdaki biçimlerde kullanabilirler (Zhang ve diğ, 2007):

1. Melezleştirme işlemi ile: Metasezgisel yöntemlerle bulunan çözümler, iyileştirici sezgisellerle geliştirilirler. Kanonik melezleştirme işleminde tek bir sezgisel yerel arama algoritması, Uyarlamalı melezleştirmede ise birkaç sezgisel algoritma kullanılır.
2. Hipersezgisel yöntemlerle: Bir Hipersezgisel algoritma içerisinde Metasezgisel yöntemler, basit sezgisellerin problem üzerinde uygulanma sırasını belirlerler.
3. Yapılandırıcı sezgisellerin kontrol parametrelerinin ayarlanmasında: Metasezgisel yöntemler, probleme özgü tek başına çözüm üreten Yapılandırıcı sezgisellerin kontrol parametrelerinin dinamik olarak ayarlanmasını gerçekleştirebilirler.

Hipersezgisel Yöntemler, diğer algoritmalara göre aşağıdaki üstünlüklere sahiptir (Cowling ve diğ, 2001), (Özcan ve diğ, 2008):

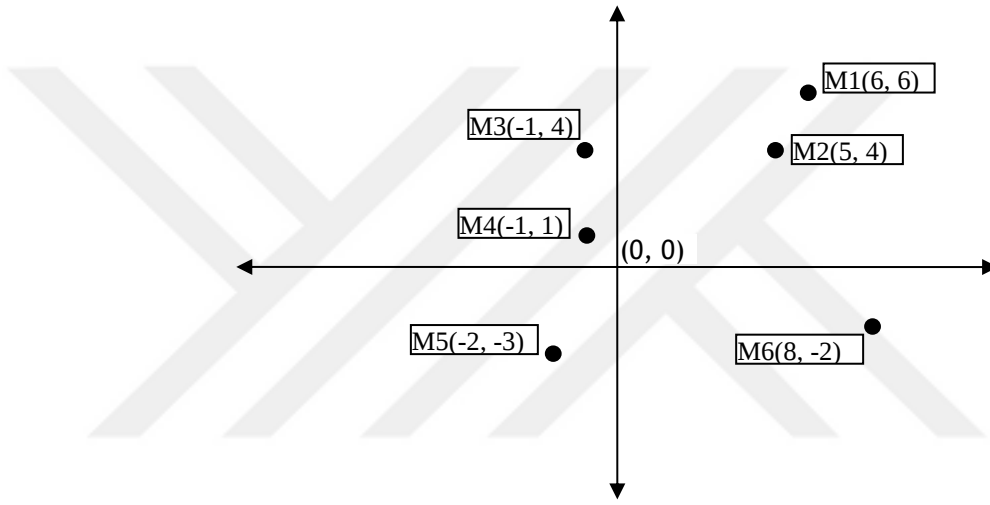
1. Basit sezgisel yöntemlerin modellenmesi ve kodlanması daha kolay olduğu için Hipersezgisel Algoritmaların kullanımı daha pratiktir.
2. Hipersezgisel Algoritmalar yapısı korunarak farklı problemlere de uygulanabilirler, problemden bağımsız yöntemlerdir.
3. Hipersezgisel Algoritmaların performansı, problemdeki değişikliklerden ve parametrelerden oldukça az etkilenir. Hipersezgisel Algoritmalar bağımsızdır ve problem hakkında uzman olmayan biri tarafından rahatlıkla kullanılabilir.
4. Sezgisel yöntemlerin çalışma süresi daha kısa olduğu için, Hipersezgisel Yöntemlerden hızlı bir şekilde uygun çözüm alınabilir.
5. Hipersezgisel Algoritmalar çözüm uzayında değil de, sezgisel yöntem uzayında arama yaptığı için çevresel değişikliklerden az etkilenir, daha dayanıklıdır.

Buna göre Hipersezgisel Algoritmaların en iyi şekilde, aşağıdaki durumlarda kullanılabileceği söylenebilir (Chakhlevitch ve Cowling, 2008):

1. Daha önce karşılaşılmamış bir problem ile çalışılması durumunda.
2. Probleme kısa sürede hızlı bir şekilde çözüm bulunması gerektiğinde.
3. Problemde kısıtlar, parametreler veya koşullar çok sık değiştiğinde.
4. Bilgisayarda çoklu kullanımlı uygulamalara olanak tanındığında.
5. Problemin bir çizge modeli ile tasarımı olanaklı olduğunda.

4.6 Hipersezgisel Algoritmalarla ilgili basit bir uygulama

Bu başlık altında iyileştirici sezgiselleri kullanan ve salt iyileştirme gösteren hareketleri kabul eden (Sİ) Basit Hipersezgisel bir algoritma, Öklidyen Düzlemde Gezgin Satıcı Problemine uygulanacaktır. Bu problemde bir satıcının tüm müşterileri dolaşarak başladığı yere yeniden dönecek şekilde bir rota plânı oluşturulmaktadır ve amaç toplam sefer mesafesini enazlamaktır. Altı müşteriden oluşan örnek problemle ilgili konumlar Şekil 4.4'te gösterilmektedir. Yöntemde çözümleri göstermek için Permutasyon Kodlama tekniği kullanılacak olup, bu kod, satıcının müşterileri ziyaret etme sırasını ifâde etmektedir.



Şekil 4.4 : Örnek problem müşteri koordinatları.

Algoritmada sezgisel yöntemler rassal seçilmekte ve varolan çözümde salt iyileşme sağlayan hareketler kabul edilmektedir. Kullanılacak basit sezgisel yöntemler aşağıda belirtilmiştir:

- Yerdeğiştirme Sezgiseli $r(a, b)$: Rota üzerinde a ve b no.'lu müşteriler yer değiştirir.
- Ters Döndürme (2 opt) Sezgiseli $f(x, y)$: Rota üzerinde $x < y$ koşuluyla x ve y arasında kalan noktalar ters çevrilerek yeni bir rota oluşturulur.
- Ekleme Sezgiseli $g(a, x)$: Rota üzerinde a no.'lu müşteri bulunduğu yerden çıkarılıp, x. sıraya konularak yeni bir rota oluşturulur.

Algoritma adımlarının daha hızlı ilerlemesi açısından basit sezgisel yöntemlerdeki ilk parametre rassal olarak belirlenecek, ikinci parametre için ise olası tüm durumlar değerlendirilerek sezgisel yöntemin en iyi sonucu bulunacaktır. Süreç ilk olarak

rassal bir çözümün oluşturulması ile başlamaktadır. Oluşturulan rassal çözüm 523641 çözümüdür. Bu çözümün verdiği değer

$$F(523641) = 9,9 + 6 + 10,82 + 9,49 + 8,6 + 12,04 = 56,85 \quad (4.1)$$

olup, Çizelge 4.9’de yer alan mesafe bilgilerinden yararlanılarak hesaplanmıştır.

Çizelge 4.9 : Müşteriler arasındaki mesafeler.

	M1	M2	M3	M4	M5	M6
M1	0,00	2,24	7,28	8,60	12,04	8,25
M2	2,24	0,00	6,00	6,71	9,90	6,71
M3	7,28	6,00	0,00	3,00	7,07	10,82
M4	8,60	6,71	3,00	0,00	4,12	9,49
M5	12,04	9,90	7,07	4,12	0,00	10,05
M6	8,25	6,71	10,82	9,49	10,05	0,00

Uygulaması yapılacak Basit Hipersezgisel Algoritma, sezgisel yöntemleri ve bu yöntemlerde kullanılacak ilk parametre değerlerini rassal olarak seçmektedir. Diğer seçenekler sırasıyla hesaplanarak bulunan en iyi çözüm, iterasyon sonucunda elde edilen çözüm olarak seçilmekte ve eğer varolan çözümden daha iyi ise varolan çözümle yer değiştirmektedir. Eğer iterasyon sırasında varolan çözümden daha iyi bir çözüm elde edilemezse herhangi bir yer değiştirme yapılmamaktadır. Çizelge 4.10’da iterasyonlar boyunca Hipersezgisel Algoritmanın bulduğu sonuçlar ve iyileşme miktarları gösterilmektedir. Çizelgeye göre 6. iterasyonda herhangi bir iyileşme sağlanamamış ve varolan çözüm korunmuştur.

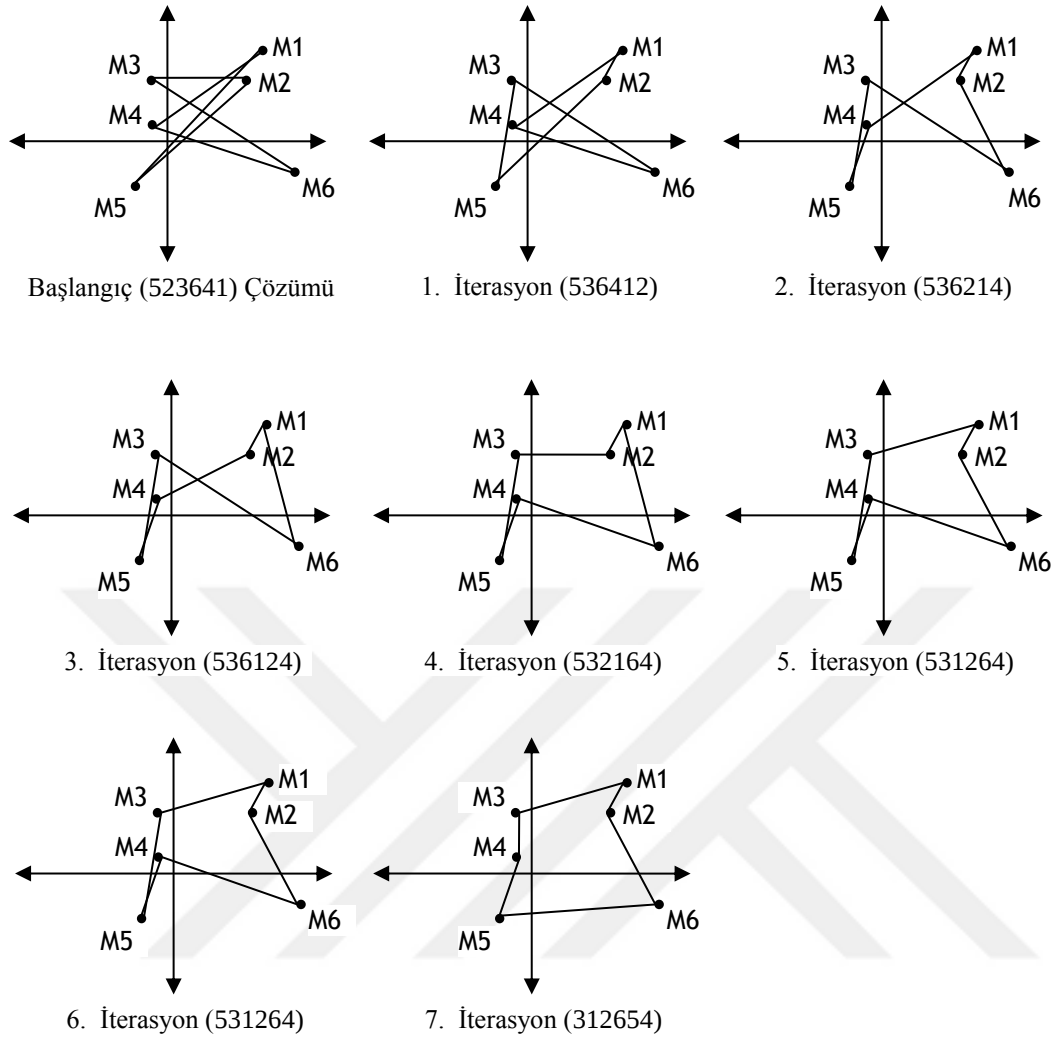
Çizelge 4.10 : İterasyonlarda Hipersezgisel Algoritmanın bulduğu sonuçlar.

İterasyon	Kullanılan Yöntem	İlk Parametre Değeri	Varolan Çözüm	Bulunan Çözümler	İyileşme Miktarı
0	Rassal			$F(523641) = 56,85$	
1	Ekleme Sezgiseli	2 No.’lu Müşteri	523641	$F(532641) = 49,91$ $F(536241) = 51,95$ $F(536421) = 48,36$ $F(536412) = 48,11$	8,73
2	Yerdeğiştirme Sezgiseli	4 No.’lu Müşteri	536412	$F(436512) = 44,85$ $F(546312) = 43,84$ $F(534612) = 39,94$ $F(536214) = 39,56$ $F(536142) = 46,87$	8,56

Çizelge 4.10 (devam) : İterasyonlarda Hipersezgisel Algoritmanın bulduğu sonuçlar.

İterasyon	Kullanılan Yöntem	İlk Parametre Değeri	Varolan Çözüm	Bulunan Çözümler	İyileşme Miktarı
3	Ekleme Sezgiseli	1 No.'lu Müşteri	536214	F(153624) = 51,95 F(513624) = 47,68 F(531624) = 40,14 F(536124) = 39,2 F(536241) = 51,95	0,36
4	Ters Döndürme	3. Konum	536124	F(531624) = 40,14 F(532164) = 37,16 F(534216) = 37,31	2,04
5	Yer Değiştirme Sezgiseli	1 No.'lu Müşteri	532164	F(132564) = 51,32 F(512364) = 44,7 F(531264) = 36,91 F(532614) = 40,75 F(532461) = 49,55	0,26
6	Ters Döndürme	3. Konum	531264	F(532164) = 37,16 F(536214) = 39,56 F(534621) = 40,54	İyileşme sağlanamadı
7	Ekleme Sezgiseli	5 No.'lu Müşteri	531264	F(351264) = 40,54 F(315264) = 48,42 F(312564) = 41,95 F(312654) = 33,4	3,51

Basit Hipersezgisel Algoritma başlangıç değerinden 23,45 birimlik iyileşme göstererek 7. iterasyonun sonunda optimum çözüme ulaşmıştır. Optimum çözümde satıcı, müşterileri 312654 rotası ile ziyaret edecek ve toplam 33,4 birim mesafe katedecektir. İterasyonlar süresince bulunan çözümler Şekil 4.5'te gösterilmektedir.



Şekil 4.5 : İterasyonlar süresince bulunan çözümler.

4.7 Hipersezgisel Algoritmalarla ilişkin çalışmalardan çıkarımlar

Hipersezgisel Yöntemler için bu tez çalışmasının tamamlandığı süreye kadar yapılan literatür çalışmaları incelendiğinde aşağıdaki sonuçlara varılabilir:

- Hipersezgisel Yöntemlerle ilgili yapılan araştırma sayısı zaman geçtikçe artmaktadır. Uygulama yapılan problem tipleri dikkate alındığında en fazla Sınav, Ders ve Satış Çizelgeleme problemlerine yönelik çalışma yapıldığı görülmektedir.
- Uygulamaların daha çok sanal problemler üzerinde gerçekleştirildiği gözlemlenmiştir. Gerçek yaşamdan uygulamalara ise özellikle son yıllarda yapılan çalışmalarda rastlanmaktadır.

- Çalışma sayıları dikkate alındığında Hipersezgisel Yöntemlerin en çok Metasezgisel Algoritmalarla melez olarak kullanıldığı görülmektedir. Basit Hipersezgiseller ise daha çok, diğer geliştirilen Hipersezgisel Yöntemle karşılaştırma amacıyla uygulanmıştır. Melezleştirmede en fazla kullanılan Metasezgisel yöntem, Genetik Algoritma olmuştur.
- Algoritma sürecinde yer alan kabul ölçütü olarak ise en çok Sİ ve TH ölçütleri kullanılmıştır. Daha sonra çalışmalarda sırasıyla BF ve MC ölçütlerinin yer aldığı gözlemlenmiştir.

Bu tez çalışması kapsamında ise Hipersezgisel Yöntemlere ilişkin literatür araştırması için aşağıdaki çıkarımlar elde edilebilir:

- Literatürde Hipersezgisel Algoritmalar daha önce, gerçek yaşamda karşılaşılan Tedârik Zinciri Yönetiminde tesislerin aktifleştirilme kararının verildiği, akışların ve rotaların hesaplandığı çok katmanlı Lojistik Ağ Tasarımı Problemine uygulanmamıştır.
- Lojistik Ağ Tasarımı alt problemleri konusunda ise Hipersezgisel Algoritmalar 2009 yılında Garrido ve Castro tarafından Kapasite Kısıtlı Araç Rotalama Problemi üzerinde uygulanmıştır. Bu çalışmada da tek katman üzerinde sadece rotalama problemi üzerinde çalışılmış, LAT'da yer alan çok katmanlı yapı ele alınmamıştır.
- Ağ tasarımı problemleri kapsamında Hipersezgisel Yöntemler, Su Ağ Tasarımı konusunda, 2010 yılında Raad ve diğerleri, 2013'te McClymont ve diğerleri ve 2015 yılında ise Kheiri ve diğerleri tarafından uygulanmıştır. Su Ağ Tasarımı probleminde katman yapısı, rotalama, kapasite gibi Lojistik Ağ Tasarımındaki unsurlar bulunmadığından problem, kapsamı itibarıyla LAT'a göre daha basit bir problemidir.



5. ELE ALINAN LOJİSTİK AĞ TASARIMI PROBLEMİ İÇİN HİPERSEZGİSEL ALGORİTMA ÖNERİSİ VE UYGULAMASI

Bu bölümde Kapasite Kısıtlı İleri Lojistik Ağ Tasarımı problemleri için geliştirilen ve önerilen bir Hipersezgisel Algoritma tanıtılacaktır. Öncelikle algoritma ayrıntılı bir şekilde açıklanacak, ardından bir, iki ve üç güzergâhlı (sırasıyla iki, üç ve dört katmanlı) LAT problem örnekleri, geliştirilen bu algoritma kullanılarak çözülecektir. Onu tâkiben Hipersezgisel Algoritma, literatürde daha önceden kullanılmış olan metasezgisel yöntemlerle karşılaştırılacaktır.

Literatürde bu tez çalışmasının tamamlanma tarihine kadar, atama ve rotalamanın birlikte yer aldığı Kapasite Kısıtlı Lojistik Ağ Tasarımı problemi için yapılmış bir Hipersezgisel Algoritma uygulaması yayınına rastlanmamıştır.

5.1 Geliştirilen Hipersezgisel Algoritmanın Öğeleri

Lojistik Ağ Tasarımı Problemi için hazırlanan Hipersezgisel Algoritma, nesne tabanlı teknikler ile geliştirilmiş olup, algoritma içerisinde yer alan nesneler ve tanımları aşağıda belirtilmektedir:

Parametre: Algoritma süresince değişmeyen değerleri tutan, bir ad ve numaraya sahip bileşendir. Örneğin; tesis kapasitesi, tedârikçi kapasitesi, sabit mâliyet ve birim değişken mâliyet, algoritma içinde sabit olarak alınır.

Değişken: Algoritma süresince karar değişkenlerini tutan, bir ad ve numaraya sahip, kesikli veya sürekli değer alabilen bileşendir. Örneğin, bir tesisin aktifleştirilmesi veya aktifleştirilmemesi ile ilgili 0-1 karar değişkeni, tesislerdeki üretim miktarı, araçların taşıdığı miktar ve stok miktarı, algoritma içinde değişken olarak ele alınır.

Düğüm: Lojistik Ağ Tasarımı içerisinde ana noktalarda bulunan; tedârikçi, fabrika, dağıtım merkezi, depo ve müşterileri ifâde eden bileşendir. Her düğüm bir veya birden fazla sabit, değişken veya kısıt barındırabilir.

Kanal: Lojistik Ağ Tasarımı Probleminde; düğümler arasındaki ürün, hammadde akışını gösteren, bir başlangıç ve bitiş düğümüne sahip bileşendir. Her kanal bir veya birden fazla sabit, değişken veya kısıta sahip olabilir.

Kısıt: Bir düğüm veya kanalda yer alan karar değişken değerlerinin birbirleriyle veya sabitlerle olan ilişkisine sınır koyan, bir ad ve numaraya sahip bileşendir. Örneğin, her düğüm ve kanal için fabrika ve dağıtım merkezi kapasitesinin geçilememesi, müşteri taleplerinin karşılanması, algoritma içinde birer kısıt olarak ele alınır.

Amaç: Problemden ulaşılması gereken hedefi, sabit ve değişkenler kullanarak ortaya koyan, döngüler ve koşullar barındırabilen algoritma bileşenidir. Örneğin, toplam maliyetin enazlanması, algoritmada genel amaç olarak yer alır.

Düğüm Grubu: Lojistik Ağ Tasarımı Probleminde tedârikçiler, fabrikalar, dağıtım merkezleri ve müşteriler gibi katmanları oluşturan ve içerisinde düğümler bulunan, bir ad ve numaraya sahip bileşendir. Bir düğüm grubunda, gruptaki düğümlerin tümüne ait kısıtlar tanımlanabilir.

Güzergâh: Düğüm grupları arasındaki ilişkileri betimleyen, bir başlangıç ve bitiş düğüm grubuna sahip ve içerisinde kanallar barındıran algoritma bileşenidir. Güzergâh, atama problemleri için kanallar arasında bir sıralama yapmaksızın gruplama yapar. Her güzergâh, bir ekleme-güncelleme ile bir kaldırma olayı içerir.

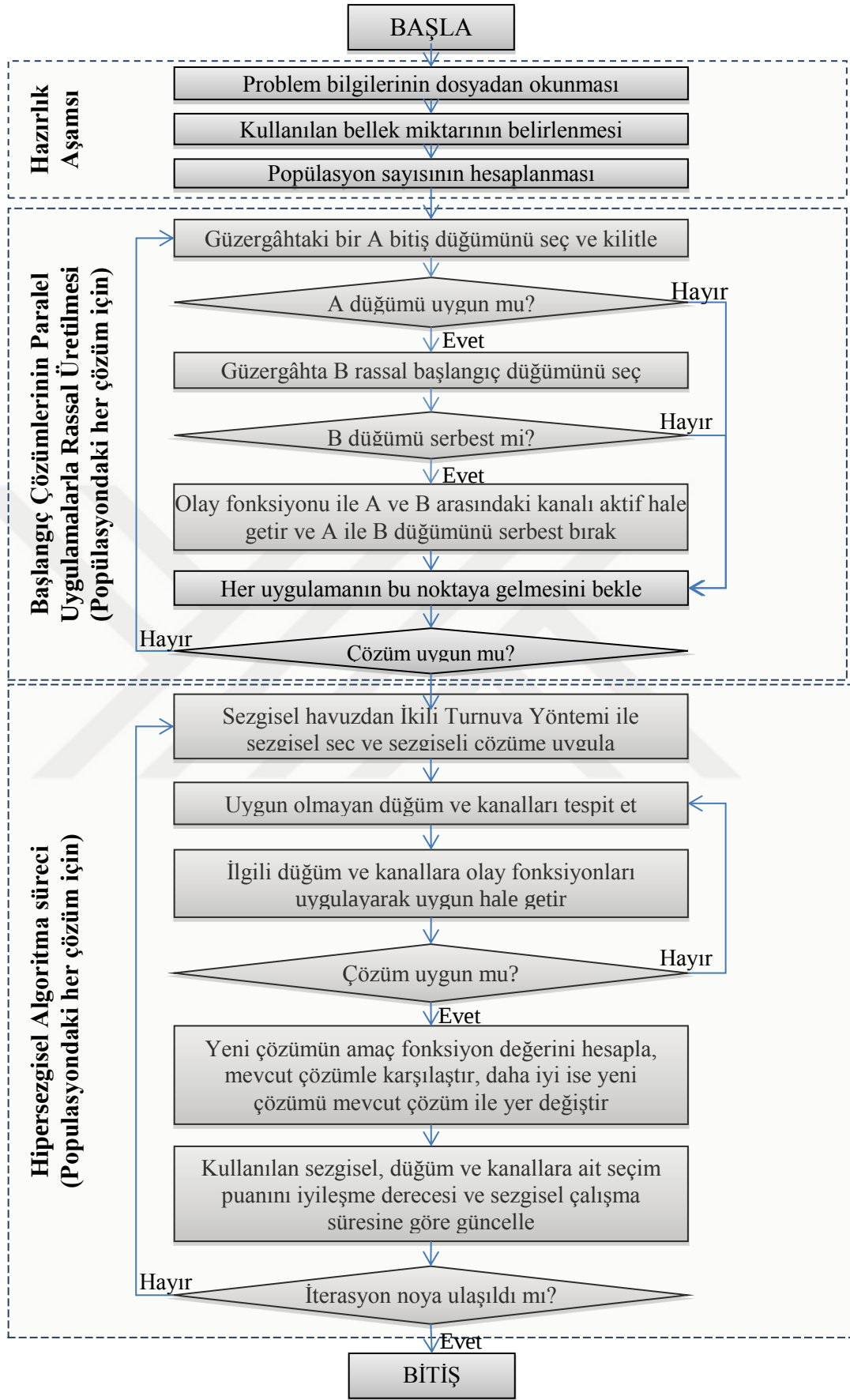
Olay: Lojistik Ağ Tasarımında bir güzergâh üzerinde uygunluğu bozmayan, ancak diğer güzergâhlar üzerinde uygunluğu bozabilen ve düğüm ile kanallar üzerindeki sabit ve değişkenler üzerinden işleyen kullanıcı tanımlı fonksiyonlardır. Olaylar, uygun olmayan çözümleri uygun duruma getirmek için kullanılırlar. Örneğin bir olay, dağıtım merkezi–müşteri arasındaki kapasite kısıtı veya talep karşılama kısıtını dikkate alarak bir atama yapabilir veya kapasite kısıtı sağlamayan bir tesis için yapılan atamayı kısıtı sağlayacak şekilde düzenleyebilir.

Basit Sezgisel: Algoritma geliştirici tarafından problem amacını gerçekleştirmek için bir konuya odaklanarak hazırlanan ve bu bakış açısı üzerinde uygunluğu (feasibility) sağlayan ancak konu dışındaki uygunluk ile ilgilenmeyen ve içerisinde olay fonksiyonu barındıran kullanıcı tanımlı fonksiyonlardır. Basit sezgisel fonksiyon, bir veya birden fazla güzergâh üzerinde çalışabilir ve genel bir fonksiyon formatına (alias) sahiptir. Lojistik Ağ Tasarımı Problemi için tasarlanan basit sezgisel fonksiyonlar aşağıda açıklanmıştır:

1. Mâliyeti en düşük düğümü ekle (CheapestAdd): Çözümde bulunan herhangi bir düğüm (Tedârikçi, tesis, müşteri) seçilir. Güzergâhlar üzerinden ataması yapılacak bu düğüm ile arasında en az taşıma mâliyeti olan ve çözümde yer alan düğüm bulunur. Bu düğüm ile diğer düğümler arasındaki taşıma kanalları kaldırılır. Başta seçilen düğüm ile ona en yakın düğüm arasında atama gerçekleştirilir.
2. En düşük üretimi yapan tesisi kaldır (LowestAmountFacilityRemove): Çözümde yer alan bir güzergâhtaki en düşük üretimi yapan tesis bulunur, bu tesis ile diğer düğümler arasındaki taşıma kanalları kaldırılır, tesis çözümden çıkartılır.
3. En büyük sabit mâliyete sahip tesisi kaldır (HighestFCFacilityRemove): Çözümde yer alan bir güzergâhtaki en yüksek sabit mâliyete sahip tesis bulunur, bu tesis ile diğer düğümler arasındaki taşıma kanalları kaldırılır, tesis çözümden çıkartılır.
4. En büyük birim değişken mâliyete sahip tesisi kaldır (HighestVCFacilityRemove): Çözümde yer alan bir güzergâhtaki en yüksek birim değişken mâliyete sahip tesis bulunur, bu tesis ile diğer düğümler arasındaki taşıma kanalları kaldırılır, tesis çözümden çıkartılır.
5. En büyük toplam mâliyete sahip tesisi kaldır (HighestTCFacilityRemove): Çözümde yer alan bir güzergâhtaki en yüksek toplam mâliyete sahip tesis bulunur, bu tesis ile diğer düğümler arasındaki taşıma kanalları kaldırılır, tesis çözümden çıkartılır.
6. Belirli bir tesisi çözümden çıkart (RemoveNode): Çözümde yer alan bir güzergâhtaki belirli olan tesis ile diğer düğümler arasındaki taşıma kanalları kaldırılır, tesis çözümden çıkartılır.
7. En yüksek mâliyete sahip kanalı çözümden çıkart (HighestCostArcRemove): Çözümde yer alan bir güzergâhtaki en yüksek mâliyete sahip kanal bulunur, bu kanal çözümden çıkartılır.
8. En az taşıma yapılan kanalı çözümden çıkart (LowestAmountArcRemove): Çözümde yer alan bir güzergâhtaki en düşük taşıma miktarına sahip kanal bulunur, bu kanal çözümden çıkartılır.
9. Belirli bir kanalı çözümden çıkart (RemoveArc): Çözümde yer alan bir güzergâhtaki belirli olan kanal çözümden çıkartılır.

Lojistik Ağ Tasarımı için geliştirilen Hipersezgisel Algoritma'nın akış şeması Şekil 5.1'de gösterilmektedir. Hipersezgisel Algoritma'da başlangıç çözümleri rassal olarak oluşturulmakta olup, basit sezgisellerin seçilmesinde çözümlerde yapılan değişiklikler gözönüne alınmaktadır.

Çözümde iyi yönde bir değişiklik olması durumunda, o basit sezgiselin seçilme olasılığı değişiklikle doğru orantılı olarak artmaktadır. Geliştirilen Hipersezgisel Yöntem'de iterasyon sayısı probleme bağlı olarak belirlenmiştir. Çözüm süresi ise algoritmanın çalışma süreci boyunca bulunan en iyi çözümün ulaşılma süresi olarak ele alınmıştır. Yöntemin ilk adımı olan başlangıç çözümlerinin oluşturulması, en son güzergâhta rassal olarak kanalların atanması ile başlar. En son güzergâhta müşteri kısıtlarının sağlanması durumunda, bir önceki güzergâhta aynı şekilde kanalların atanması ile sürer. Bu atamalar ile uygun bir çözüm oluşturulur ve amaç fonksiyon değeri hesaplanır.



Şekil 5.1 : Hipersezgisel Algoritma akış şeması.

5.2 Yöntemin Getirdiği Yenilikler

Bu tez çalışması kapsamında geliştirilen Hipersezgisel Algoritma, Bölüm 5.1’de sözedilen algoritma öğeleri üzerinden, Lojistik Ağ Tasarımı problemine uyarlanabilirlik ve esneklik özelliği sağlamaktadır. Örneğin değişken, düğüm, kanal, güzergâh ve kısıt öğeleri ile ilgili yöntem düzenlenerek farklı Lojistik Ağ Tasarımı problemleri için çözüm oluşturulabilmektedir. Bu da algoritmanın Lojistik Ağ Tasarımı probleminde örneğin katman ve ürün sayısı değişkenliğine ve farklı kısıtlara göre esneklik kazanmasını sağlar.

Bu çalışmada ele alınan problem dört katmanlı Kapasite Kısıtlı İleri Lojistik Ağ Tasarımı Problemi olup, problem içinde dağıtım merkezi ile müşteriler arasında rotalama gerçekleştirmektedir. Bunlara ek olarak, problem çözümü için kurulan modelde tedârik, sevkiyat ve dağıtım sürecine ilişkin kısıtlar da bulunmaktadır. Bu nedenle problemin karmaşıklık düzeyi artmaktadır. Geliştirilen Hipersezgisel yöntemin uyarlanabilirlik özelliği, bu karmaşıklık düzeyindeki Lojistik Ağ Tasarımı probleminin çözülmesini sağlamıştır.

5.3 Kurgulanmış Lojistik Ağ Tasarımı Uygulamaları

Bu bölümde sanal XYZ firması için farklı katman sayılarında, kurgulanmış, farklı kısıtlar barındıran Lojistik Ağ Tasarımı problemleri için tez kapsamında geliştirilen Hipersezgisel Algoritmanın uygulaması yapılmıştır. Yöntemin farklı problemlere uygulanabilmesi, algoritmanın uyarlanabilir (adaptif) olduğunu göstermektedir.

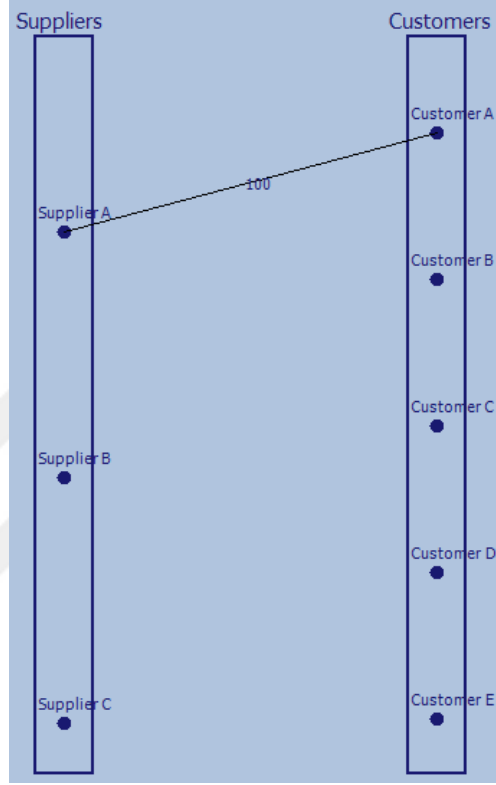
5.3.1 İki katmanlı lojistik ağı

XYZ Firması şu an salt A tesisi ile hizmet vermektedir ve artan müşteri talepleri karşısında bir veya iki tesisin daha açılması düşünülmektedir. Yeni açılması düşünülen tesisler farklı yerlerde olup, B ve C tesisleri olarak belirlenmiştir. Problemde, hangi tesislerin açılacağı ve tesisler ile müşteriler arasındaki ürün taşımalarının ne miktarlarda olacağı bulunmaya çalışılmaktadır. 3 tesis ve 5 müşteriden oluşan lojistik sistemde amaç; tesis açma, üretim ve taşıma mâliyetleri toplamının enazlanmasıdır. Problemde yer alan kısıtlar aşağıda belirtilmiştir:

- Tesis kapasiteleri
- Müşteri taleplerinin karşılanması

- Taşıma kapasiteleri
- Tesis A'nın her zaman açık olması
- Tesis A'dan müşteri A'ya en az 100 birim ürün taşınması

Problemdeki düğüm grupları, düğümler ve çözümde bulunan taşıma kanalları Şekil 5.2'de gösterilmektedir.



Şekil 5.2 : Senaryo 1 için başlangıç durum.

Problemde Fabrika düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.3'te gösterilmektedir.

Parameters				Variables			
Nodes	FixedCost	UnitProductionCost	Capacity	Nodes	ExistInSolution	Stock	ProductionAmount
Factory A	0	12	550	Factory A	1	100	0
Factory B	18000	11	450	Factory B	0	0	0
Factory C	12000	14	400	Factory C	0	0	0

Şekil 5.3 : Fabrikalar için parametre ve değişkenler.

Problemde Müşteri düğüm grubunda yer alan müşteriler için sabit ve değişken başlangıç değerleri Şekil 5.4'te gösterilmektedir.

Parameters		Variables	
Nodes	Demand	Nodes	TotalDelivery
Customer A	200	Customer A	100
Customer B	100	Customer B	0
Customer C	150	Customer C	0
Customer D	200	Customer D	0
Customer E	150	Customer E	0

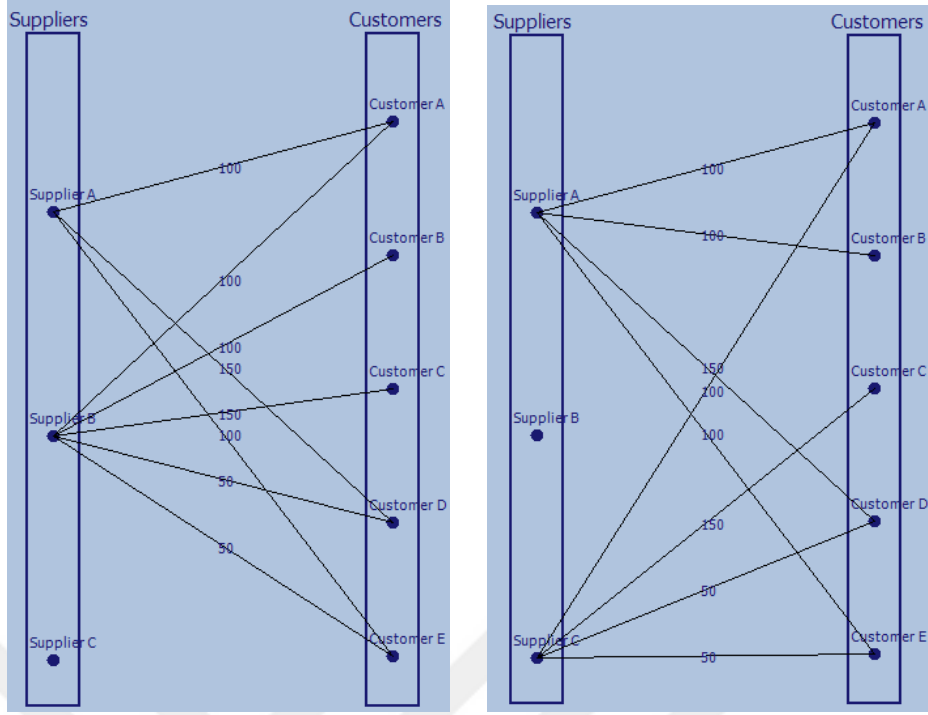
Şekil 5.4 : Müşteriler için parametre ve değişkenler.

Problemde Fabrika ile Müşteri arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.5'te gösterilmektedir.

Parameters			Variables		
Nodes	UnitTransportCost	Capacity	Nodes	ExistInSolution	TransportAmount
Factory A - Customer A	4	120	Factory A - Customer A	1	100
Factory A - Customer B	3	100	Factory A - Customer B	0	0
Factory A - Customer C	2	110	Factory A - Customer C	0	0
Factory A - Customer D	5	150	Factory A - Customer D	0	0
Factory A - Customer E	3	100	Factory A - Customer E	0	0
Factory B - Customer A	3	100	Factory B - Customer A	0	0
Factory B - Customer B	3	120	Factory B - Customer B	0	0
Factory B - Customer C	3	150	Factory B - Customer C	0	0
Factory B - Customer D	3	110	Factory B - Customer D	0	0
Factory B - Customer E	4	150	Factory B - Customer E	0	0
Factory C - Customer A	4	130	Factory C - Customer A	0	0
Factory C - Customer B	5	160	Factory C - Customer B	0	0
Factory C - Customer C	6	170	Factory C - Customer C	0	0
Factory C - Customer D	3	100	Factory C - Customer D	0	0
Factory C - Customer E	3	150	Factory C - Customer E	0	0

Şekil 5.5 : Fabrika-müşteri kanalı için parametre ve değişkenler.

Algoritma çalıştığı zaman popülasyonu 480 olarak ele almıştır. Hipersezgisel yöntem tarafından bulunan en iyi iki çözüm (27600, 23250) Şekil 5.6'da gösterilmektedir.



Şekil 5.6 : Senaryo 1 için bulunan en iyi iki çözüm.

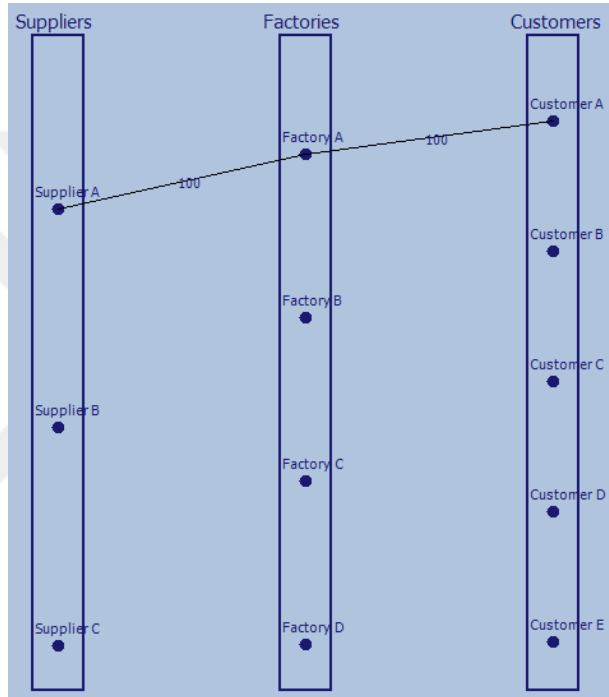
5.3.2 Üç katmanlı lojistik ağı

XYZ Firması iki katmanlı lojistik ağı benzer olarak A fabrikası ve A tedârikçisi ile işleyişini sürdürmektedir. Şirket yöneticileri firmanın farklı pazarlara açılma stratejisine yönelik, yeni fabrikalar için üç farklı yerde B, C ve D olmak üzere tesis alternatifleri belirlemişlerdir. Yöneticiler aynı zamanda hammadde gereksinimini karşılamak üzere, bölgelerdeki tedârikçi firmalarla da çalışmak istemektedir. Bu nedenle hangi fabrikanın nerede açılacağı kararının yanında, fabrikanın hangi tedârikçiden ne kadar hammadde temin edeceğinin de belirlenmesi gerekmektedir. Tedârikçi seçiminde riskin karar sürecini etkilemesini sağlamak için ise modele sabit tedârikçi risk mâliyeti eklenmiştir. 3 tedârikçi, 4 tesis ve 5 müşteriden oluşan lojistik sistemde amaç, toplam tesis açma, satınalma, tedârikçi seçimi, üretim ve taşıma mâliyetlerinin enazlanmasıdır. Problemden yer alan kısıtlar aşağıda belirtilmiştir:

- Tedârikçi kapasiteleri
- Fabrika kapasiteleri
- Müşteri taleplerinin karşılanması
- Fabrika hammadde taleplerinin karşılanması
- Tedârikçi-Fabrika arası taşıma kapasiteleri

- Fabrika-Müşteri arası taşıma kapasiteleri
- Fabrika A'nın her zaman açık olması
- Fabrika A'dan müşteri A'ya en az 100 birim ürün taşınması
- Tedârikçi A'nın her zaman seçili olması
- Tedârikçi A'dan fabrika A'ya en az 100 birim hammadde taşınması

Problemdeki düğüm grupları, düğümler ve çözümde bulunan taşıma kanalları Şekil 5.7'de gösterilmektedir.



Şekil 5.7 : Üç katmanlı lojistik ağ için başlangıç durum.

Problemde Tedârikçi düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.8'de gösterilmektedir.

Parameters				Variables			
Nodes	FixedCost	UnitSalePrice	Capacity	Nodes	ExistInSolution	Stock	SaleAmount
Supplier A	0	16	600	Supplier A	1	0	100
Supplier B	1500	11	650	Supplier B	0	0	0
Supplier C	1200	14	700	Supplier C	0	0	0

Şekil 5.8 : Üç katmanlı lojistik ağda tedârikçiler için parametre ve değişkenler.

Problemde Fabrika düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.9'da gösterilmektedir:

Parameters				Variables					
Nodes	FixedCost	Unit ProductionCost	Capacity	Nodes	Exist In Solution	Product Stock	Production Amount	Raw Material Stock	Total Raw Material Amount
Factory A	15000	12	300	Factory A	1	100	0	50	
Factory B	18000	11	350	Factory B	0	0	0	0	0
Factory C	12000	14	300	Factory C	0	0	0	0	0
Factory D	11000	15	400	Factory D	0	0	0	0	0

Şekil 5.9 : Üç katmanlı lojistik ağda fabrikalar için parametre ve değişkenler.

Problemde Müşteri düğüm grubunda yer alan müşteriler için sabit ve değişken başlangıç değerleri Şekil 5.10’da gösterilmektedir.

Parameters		Variables	
Nodes	Demand	Nodes	TotalDelivery
Customer A	200	Customer A	100
Customer B	100	Customer B	0
Customer C	150	Customer C	0
Customer D	200	Customer D	0
Customer E	150	Customer E	0

Şekil 5.10 : Üç katmanlı lojistik ağda müşteriler için parametre ve değişkenler.

Problemde Tedârikçi ile Fabrika arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.11’de gösterilmektedir.

Parameters			Variables		
Nodes	UnitTransportCost	Capacity	Nodes	ExistInSolution	TransportAmount
Supplier A - Factory A	4	200	Supplier A - Factory A	1	100
Supplier A - Factory B	3	200	Supplier A - Factory B	0	0
Supplier A - Factory C	2	200	Supplier A - Factory C	0	0
Supplier A - Factory D	3	200	Supplier A - Factory D	0	0
Supplier B - Factory A	3	200	Supplier B - Factory A	0	0
Supplier B - Factory B	3	200	Supplier B - Factory B	0	0
Supplier B - Factory C	3	200	Supplier B - Factory C	0	0
Supplier B - Factory D	3	200	Supplier B - Factory D	0	0
Supplier C - Factory A	4	200	Supplier C - Factory A	0	0
Supplier C - Factory B	3	200	Supplier C - Factory B	0	0
Supplier C - Factory C	3	200	Supplier C - Factory C	0	0
Supplier C - Factory D	3	200	Supplier C - Factory D	0	0

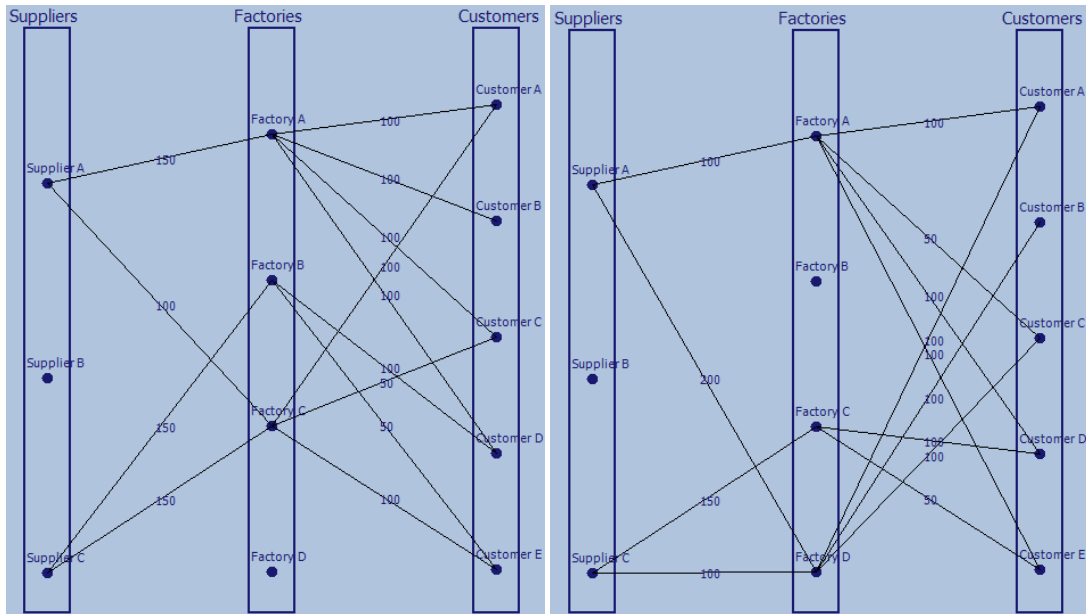
Şekil 5.11 : Tedârikçi-fabrika kanalı için parametre ve değişkenler.

Problemde Fabrika ile Müşteri arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.12’de gösterilmektedir.

Parameters			Variables		
Nodes	UnitTransportCost	Capacity	Nodes	ExistInSolution	TransportAmount
Factory A - Customer A	4	100	Factory A - Customer A	1	100
Factory A - Customer B	3	100	Factory A - Customer B	0	0
Factory A - Customer C	2	100	Factory A - Customer C	0	0
Factory A - Customer D	3	100	Factory A - Customer D	0	0
Factory A - Customer E	3	100	Factory A - Customer E	0	0
Factory B - Customer A	3	100	Factory B - Customer A	0	0
Factory B - Customer B	3	100	Factory B - Customer B	0	0
Factory B - Customer C	3	100	Factory B - Customer C	0	0
Factory B - Customer D	3	100	Factory B - Customer D	0	0
Factory B - Customer E	3	100	Factory B - Customer E	0	0
Factory C - Customer A	4	100	Factory C - Customer A	0	0
Factory C - Customer B	3	100	Factory C - Customer B	0	0
Factory C - Customer C	3	100	Factory C - Customer C	0	0
Factory C - Customer D	3	100	Factory C - Customer D	0	0
Factory C - Customer E	3	100	Factory C - Customer E	0	0
Factory D - Customer A	3	100	Factory D - Customer A	0	0
Factory D - Customer B	3	100	Factory D - Customer B	0	0
Factory D - Customer C	3	100	Factory D - Customer C	0	0
Factory D - Customer D	3	100	Factory D - Customer D	0	0
Factory D - Customer E	3	100	Factory D - Customer E	0	0

Şekil 5.12 : Fabrika-müşteri kanalı için parametre ve değişkenler.

Algoritma çalıştığı zaman popülasyon sayısını 240 olarak hesaplamıştır. Hipersezgisel yöntem tarafından bulunan en iyi iki çözüm (66150, 60010) Şekil 5.13'te gösterilmektedir.



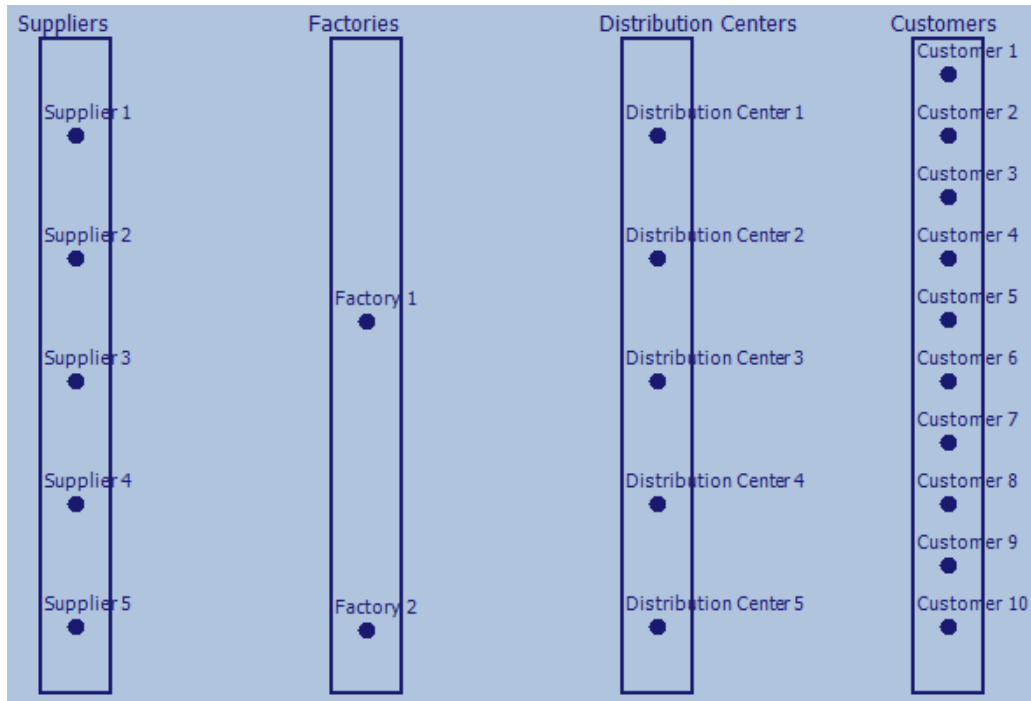
Şekil 5.13 : Senaryo 2 için bulunan en iyi iki çözüm.

5.3.3 Dört katmanlı lojistik ağı

XYZ Firması diğer senaryolardan farklı olarak bir bölgede iki fabrika ve beş dağıtım merkezi kurmak istemektedir. Şirket yöneticileri fabrika ve dağıtım merkezlerinin yerlerini en az mâliyeti gözönünde bulundurarak belirlemeyi amaçlamaktadır. Yöneticiler aynı zamanda hammadde gereksinimini karşılamak için, bölgelerdeki tedârikçi firmalarla da çalışmak istemektedir. Bu nedenle hangi fabrikanın nerede aktifleştirileceği kararının yanında, fabrikanın hangi tedârikçiden ne kadar hammadde temin edeceğinin de belirlenmesi gerekmektedir. 5 tedârikçi, 2 fabrika, 5 dağıtım merkezi ve 10 müşteriye oluşan lojistik sistemde amaç, dağıtım mâliyetlerinin enazlanmasıdır. Problemden yer alan kısıtlar aşağıda belirtilmiştir:

- Tedârikçi kapasiteleri
- Fabrika kapasiteleri
- Dağıtım Merkezi kapasiteleri
- Müşteri taleplerinin karşılanması

Problemdenki düğüm grupları, düğümler ve çözümde bulunan taşıma kanalları Şekil 5.14’te gösterilmektedir.



Şekil 5.14 : Dört katmanlı lojistik ağı için başlangıç durumu.

Problemde Tedârikçi düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.15’te gösterilmektedir.

Parameters					Variables		
Nodes	UnitSalePrice	Capacity	x	y	Nodes	ExistInSolution	SaleAmount
Supplier 1	25	500	5	2	Supplier 1	0	0
Supplier 2	30	600	4	6	Supplier 2	0	0
Supplier 3	22	400	4	3	Supplier 3	0	0
Supplier 4	20	400	7	8	Supplier 4	0	0
Supplier 5	27	550	6	4	Supplier 5	0	0

Şekil 5.15 : Tedârikçiler için parametre ve değişkenler.

Problemde Fabrika düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.16’da gösterilmektedir.

Parameters		Variables			
Nodes	Capacity	Nodes	StockAmount	x	y
Distribution Center 1	200	Distribution Center 1	0	0	0
Distribution Center 2	250	Distribution Center 2	0	0	0
Distribution Center 3	250	Distribution Center 3	0	0	0
Distribution Center 4	200	Distribution Center 4	0	0	0
Distribution Center 5	200	Distribution Center 5	0	0	0

Şekil 5.16 : Fabrikalar için parametre ve değişkenler.

Problemde Dağıtım Merkezi düğüm grubunda yer alan tesisler için sabit ve değişken başlangıç değerleri Şekil 5.17’de gösterilmektedir.

Parameters		Variables			
Nodes	Capacity	Nodes	StockAmount	x	y
Distribution Center 1	200	Distribution Center 1	0	0	0
Distribution Center 2	250	Distribution Center 2	0	0	0
Distribution Center 3	250	Distribution Center 3	0	0	0
Distribution Center 4	200	Distribution Center 4	0	0	0
Distribution Center 5	200	Distribution Center 5	0	0	0

Şekil 5.17 : Dağıtım merkezleri için parametre ve değişkenler.

Problemde Müşteri düğüm grubunda yer alan müşteriler için sabit ve değişken başlangıç değerleri Şekil 5.18’de gösterilmektedir.

Parameters				Variables	
Nodes	Demand	x	y	Nodes	TotalDelivery
Customer 1	100	20	5	Customer 1	0
Customer 2	80	22	10	Customer 2	0
Customer 3	70	24	6	Customer 3	0
Customer 4	90	18	12	Customer 4	0
Customer 5	80	16	20	Customer 5	0
Customer 6	60	20	10	Customer 6	0
Customer 7	100	21	30	Customer 7	0
Customer 8	70	23	12	Customer 8	0
Customer 9	80	26	18	Customer 9	0
Customer 10	120	19	20	Customer 10	0

Şekil 5.18 : Müşteriler için parametre ve değişkenler.

Problemde Tedârikçi ile Fabrika arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.19’da gösterilmektedir.

Parameters		Variables		
Nodes	UnitTransportCost	Nodes	ExistInSolution	TransportAmount
Supplier 1 - Factory 1	10	Supplier 1 - Factory 1	0	0
Supplier 1 - Factory 2	10	Supplier 1 - Factory 2	0	0
Supplier 2 - Factory 1	10	Supplier 2 - Factory 1	0	0
Supplier 2 - Factory 2	10	Supplier 2 - Factory 2	0	0
Supplier 3 - Factory 1	10	Supplier 3 - Factory 1	0	0
Supplier 3 - Factory 2	10	Supplier 3 - Factory 2	0	0
Supplier 4 - Factory 1	10	Supplier 4 - Factory 1	0	0
Supplier 4 - Factory 2	10	Supplier 4 - Factory 2	0	0
Supplier 5 - Factory 1	10	Supplier 5 - Factory 1	0	0
Supplier 5 - Factory 2	10	Supplier 5 - Factory 2	0	0

Şekil 5.19 : Tedârikçi-fabrika kanalı için parametre ve değişkenler.

Problemde Fabrika ile Dağıtım Merkezi arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.20’de gösterilmektedir.

Parameters		Variables		
Nodes	UnitTransportCost	Nodes	ExistInSolution	TransportAmount
Factory 1 - Distribution Center 1	10	Factory 1 - Distribution Center 1	0	0
Factory 1 - Distribution Center 2	10	Factory 1 - Distribution Center 2	0	0
Factory 1 - Distribution Center 3	10	Factory 1 - Distribution Center 3	0	0
Factory 1 - Distribution Center 4	10	Factory 1 - Distribution Center 4	0	0
Factory 1 - Distribution Center 5	10	Factory 1 - Distribution Center 5	0	0
Factory 2 - Distribution Center 1	10	Factory 2 - Distribution Center 1	0	0
Factory 2 - Distribution Center 2	10	Factory 2 - Distribution Center 2	0	0
Factory 2 - Distribution Center 3	10	Factory 2 - Distribution Center 3	0	0
Factory 2 - Distribution Center 4	10	Factory 2 - Distribution Center 4	0	0
Factory 2 - Distribution Center 5	10	Factory 2 - Distribution Center 5	0	0

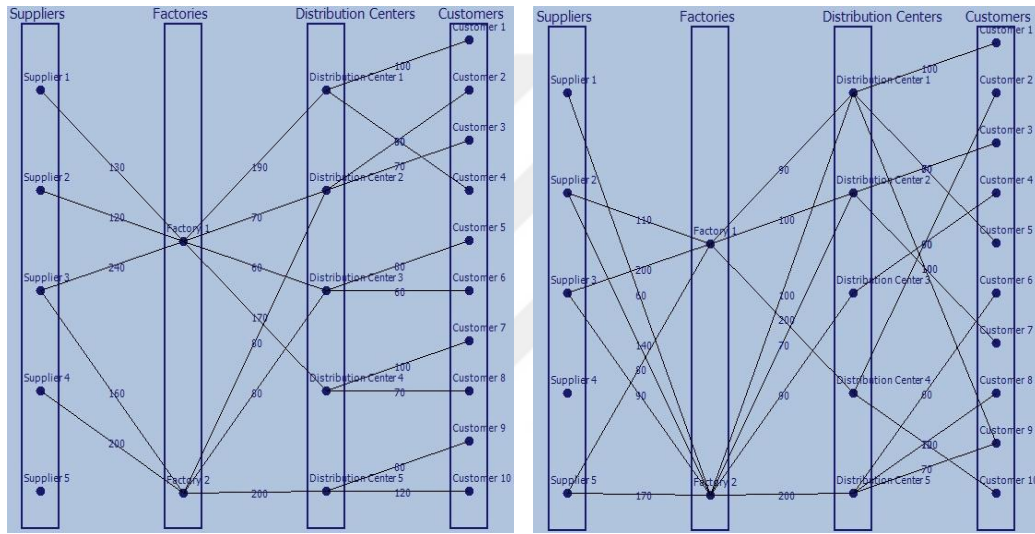
Şekil 5.20 : Fabrika-dağıtım merkezi kanalı için parametre ve değişkenler.

Problemde Dağıtım Merkezi ile Müşteri arasında yer alan kanallar için sabit ve değişken başlangıç değerleri Şekil 5.21’de özetlenmiştir. Birim taşıma mâlîyetleri, tüm kanallar için sabit 10 değeri olarak belirlenmiştir.

Parameters		Variables		
Nodes	Unit TransportCost	Nodes	ExistInSolution	TransportAmount
Distribution Center 1 - Customer 1	10	Distribution Center 1 - Customer 1	0	0
Distribution Center 1 - Customer 2	10	Distribution Center 1 - Customer 2	0	0
Distribution Center 1 - Customer 3	10	Distribution Center 1 - Customer 3	0	0
Distribution Center 1 - Customer 4	10	Distribution Center 1 - Customer 4	0	0

Şekil 5.21 : Dağıtım merkezi-müşteri kanalı için parametre ve değişkenler.

Algoritma çalıştığı zaman popülasyonu 135 olarak ele almıştır. Hipersezgisel yöntem tarafından bulunan en iyi iki çözüm (81258, 84391) Şekil 5.22’de gösterilmektedir.



Şekil 5.22 : Dört katmanlı lojistik ağ için bulunan çözümler.

5.4 Geliştirilen Hipersezgisel Algoritmanın Sanal Problemler İçin Metasezgisel Yöntemlerle Karşılaştırılması

Bu tez çalışmasında ele alınan Lojistik Ağ Tasarımı problemi için Hipersezgisel Yöntemin literatürde daha önce geliştirilen Genetik Algoritma ve Tavlama Benzetimi ile karşılaştırılması yapılacaktır.

Dört katmanlı Kapasite kısıtlı Lojistik Ağ Tasarımı problemi, ilk olarak 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma uygulaması ile çözülmüştür. Bu çalışmada öncelik tipinde kodlama yapılmakta olup, başlangıç çözümleri rassal oluşturulmaktadır. Ayrıca algorithma çaprazlama ve mutasyon, lojistik düzeyler (tedârikçi - fabrika, fabrika - dağıtım merkezi) üzerinden kanalların katmanlar ve bire-bir yer değiştirmesiyle yapılmıştır. Algorithma çaprazlama ve

mutasyon oranı 5 farklı deęerde denenmiř olup, seęim yntemi ise 2 farklı trde (Rulet ve İekli Turnuva teknięi) geręekleřtirilmiřtir.

Karřılařtırılma iin kullanılan Tavlama Benzetimi algoritması, Jayaraman ve Ross tarafından 2003 yılında LAT problemi iin tasarlanmıřtır. Bu alıřmada bařlangı zmleri rassal oluřturulmakta, komřu zmler ise yer deęiřtirme sezgiseli ile bulunmaktadır. Buna gre bir gzergh üzerinde rassal bir mřteri ve kanal seilir, her ařamada alternatif kanallar ile yer deęiřtirilerek yeni bir zme geilir. Sıcaklık parametresinin dřrlmesi ile komřu zmlerin kabul edilme olasılıęı azalmaktadır. Uygulamada 5 farklı deęerde soęutma oranı kullanılmıřtır.

Bu blmde ilk olarak, tasarlanan sanal Lojistik Aę Tasarımı Problemleri ve GAMS modelleri tanıtılacaktır. nerilen Hipersezgisel Algoritmanın sanal problemler üzerinde uygulama sonuları, Genetik Algoritma ve Tavlama Benzetimi algoritmalarının aynı rneklerde verdięi sonularla karřılařtırılarak deęerlendirilecektir.

5.4.1 Lojistik Aę Tasarımı sanal problemleri

Tez kapsamında geliřtirilen Hipersezgisel Algoritmanın etkinlięini lmek iin altı sanal problem tipi oluřturulmuř olup, bu problemler zerinden Altıparmak ve dięerleri tarafından 2006'da geliřtirilen Genetik Algoritma ile Jamayaran ve Ross tarafından 2003 yılında geliřtirilen Tavlama Benzetimi yntemi arasında karřılařtırmalar yapılmıřtır. Ayrıca bu alıřmada, her problem tipi iin 3 rassal sanal problem retilmiřtir. İlk drt problem tipi, Karma Tamsayılı Programlama Modeli kurularak GAMS ortamında zlmř ve optimum sonular elde edilmiřtir. Kalan dięer iki problem tipi byk olmasından dolayı GAMS ortamında zlememiř, algoritmalarca bulunan en iyi zm deęeri ile karřılařtırılmalar yapılmıřtır. Bu rapor kapsamında ele alınan problem trleri řunlardır:

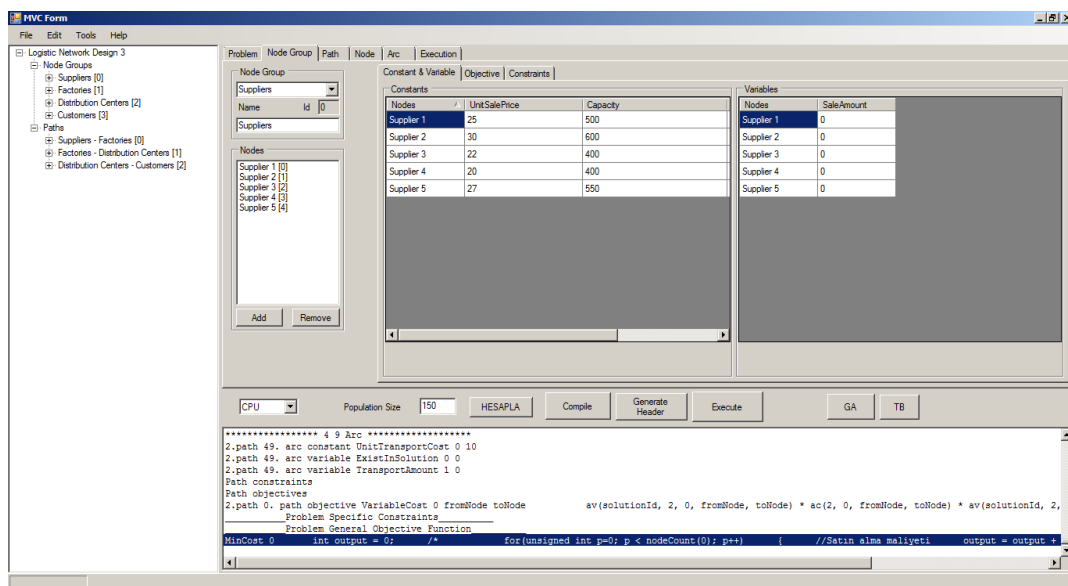
- 2×2×2×2 Problemleri: 2 tedâriki, 2 fabrika, 2 daęıtım merkezi ve 2 mřteri
- 6×1×4×6 Problemleri: 6 tedâriki, 1 fabrika, 4 daęıtım merkezi ve 6 mřteri
- 10×2×5×10 Problemleri: 10 tedâriki, 2 fabrika, 5 daęıtım merkezi ve 10 mřteri
- 12×2×8×12 Problemleri: 12 tedâriki, 2 fabrika, 8 daęıtım merkezi ve 12 mřteri
- 100×2×20×300 Problemleri: 100 tedâriki, 2 fabrika, 20 daęıtım merkezi ve 300 mřteri

- 1.000×60×50×300 Problemleri: 1.000 tedârikçi, 60 fabrika, 50 dağıtım merkezi ve 300 müşteri

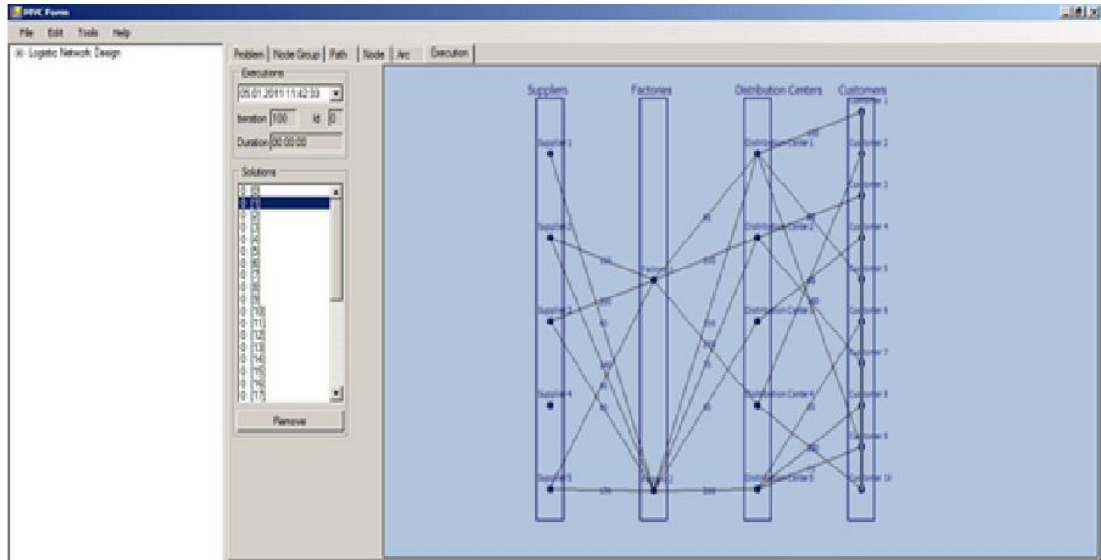
Literatürde Lojistik Ağ Tasarımı için bir problem kümesi olmadığından problem verileri tümüyle rassal olarak üretilmiştir, bu sırada kapasite ve talep kısıtları dikkate alınarak uygun çözümlerin olabilirliği gözönüne alınmıştır. Üretilen problemler için bu çalışmada Bölüm 3'te kurulan model dikkate alınmış ve senaryolar bu model üzerinden gerçekleştirilmiştir. Bulunan çözüm değerlerinin ayrıca sağlaması yapılarak sürecin kontrolü gerçekleştirilmiştir. Oluşturulan GAMS modelleri, program üzerindeki DICOPT ve BARON modülü ile çözülmüş ve optimum sonuçlar elde edilmiştir. Problem grubundaki $2 \times 2 \times 2 \times 2$, $6 \times 1 \times 4 \times 6$, $10 \times 2 \times 5 \times 10$ ve $12 \times 2 \times 8 \times 12$ problem türleri için kurulan GAMS modelleri, çalışmanın sonunda yer alan Ek A, Ek B, Ek C, Ek D, Ek E, Ek F, Ek G, Ek H, Ek I, Ek J, Ek K ve Ek L'de sunulmuştur.

5.4.2 Sanal problemler için elde edilen sonuçlar

Bu çalışmada yapılan karşılaştırmanın aynı düzeyde olabilmesi için Altıparmak ve diğerleri tarafından 2006'da geliştirilen Genetik Algoritma ile Jamayaran ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi algoritması, Hipersezgisel Algoritmalar için yazılan uygulamada olduğu gibi Microsoft .Net 2008 ortamında hazırlanmıştır. Geliştirilen yazılım Intel i7 işlemcili 8 GB Ram bellekli bilgisayarda çalıştırılmıştır. Programın ekran görüntüsü ve oluşturulan bir çözüm Şekil 5.23 ve 5.24'te gösterilmektedir.



Şekil 5.23 : Lojistik Ağ Tasarımı için hazırlanan uygulama.



Şekil 5.24 : Programın oluşturduğu bir çözüm.

Uygulaması yapılacak algoritmalar için parametreler, literatürde yapılan çalışmalar dikkate alınarak belirlenmiş olup, her bir problem tipi için kullanılan parametre değeri Çizelge 5.1’de yer almaktadır.

Çizelge 5.1 : Uygulamada kullanılan parametreler.

Yöntem	Problem Tipi	Parametre	Değer
Hipersezgisel Algoritma	2×2×2×2	İterasyon Sayısı	50
	6×1×4×6	İterasyon Sayısı	100
	10×2×5×10	İterasyon Sayısı	100
	12×2×8×12	İterasyon Sayısı	200
	100×2×20×300	İterasyon Sayısı	250
	1000×60×50×300	İterasyon Sayısı	300
Genetik Algoritma	2×2×2×2	İterasyon Sayısı	50
		Popülasyon Sayısı	40
	6×1×4×6	İterasyon Sayısı	100
		Popülasyon Sayısı	60
	10×2×5×10	İterasyon Sayısı	200
		Popülasyon Sayısı	80
	12×2×8×12	İterasyon Sayısı	300
		Popülasyon Sayısı	100
	100×2×20×300	İterasyon Sayısı	450
		Popülasyon Sayısı	100
	1000×60×50×300	İterasyon Sayısı	500
		Popülasyon Sayısı	100
Tüm Problemler		Çaprazlama Oranı	0,2, 0,3, 0,4, 0,5 ve 0,6
Tüm Problemler		Mutasyon Oranı	0,03, 0,04, 0,05, 0,06 ve 0,07

Çizelge 5.1 (Devam) : Uygulamada kullanılan parametreler.

Yöntem	Problem Tipi	Parametre	Değer
Tavlama Benzetimi	2×2×2×2	İterasyon Sayısı	50
	6×1×4×6	İterasyon Sayısı	100
	10×2×5×10	İterasyon Sayısı	100
	12×2×8×12	İterasyon Sayısı	200
	100×2×20×300	İterasyon Sayısı	250
	1000×60×50×300	İterasyon Sayısı	300
	Tüm Problemler	Sıcaklık Katsayısı	200, 300, 400, 500 ve 600
	Tüm Problemler	Soğutma Oranı	0,75, 0,8, 0,85, 0,90 ve 0,95

Bu çalışmada her problem tipi için rastgele 3 sanal problem üretilmiştir. Yöntemler, her bir problemde her bir parametre kümesi için yüz kez uygulanmış olup, bulunan çözüm değerleri, en iyi çözümle arasındaki fark, buna ait sapma yüzdesi ve çözüm süreleri hesaplanmıştır. Daha sonra yöntemler için parametre kümeleri içinde yapılan denemelerde bulunan en iyi 100 çözüm ele alınarak bunlar üzerinden karşılaştırmalar yapılmıştır. Bu raporda yer alan çözüm sürelerinden kasıt, algoritmanın çalışma süreci boyunca bulduğu en iyi çözüme ulaşma süresidir.

Tez çalışmasında geliştirilen Hipersezgisel Yöntem (HS), Altıparmak ve diğerleri tarafından 2006’da geliştirilen Genetik Algoritma (GA) ve Jamayaran ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi (TB) algoritması oluşturulan sanal problemler üzerinde farklı parametre değerleri için yüz kez çalıştırılmış ve Ekler bölümünde yer alan çözüm değerleri üretilmiştir. Çizelge 5.2’de özet sonuçlar görülmektedir. Bu tabloda, her bir problem için en iyi çözümün (ilk dört problem tipi için optimum) bulunup bulunmadığı, üretilen yüz çözüm için en iyi çözümden ortalama sapma yüzdesi, ortalama çözüm süresi, üretilen yüz çözümün standart sapması ve çözüm sürelerinin standart sapması gösterilmektedir.

Çizelge 5.2 : Sanal uygulama sonucunda bulunan özet sonuçlar.

Problem	Yöntem	Bulunan en iyi değer	En iyi çözümden ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
2×2×2×2 ₁	HS	312090	0,00%	1,45	0,00	0,50
2×2×2×2 ₁	GA	312090	0,00%	4,55	0,00	0,50
2×2×2×2 ₁	TB	312090	0,00%	2,51	0,00	0,50

Çizelge 5.2 (Devam) : Sanal uygulama sonucunda bulunan özet sonuçlar.

Problem	Yöntem	Bulunan en iyi değer	En iyi çözümde ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
$2 \times 2 \times 2 \times 2_2$	HS	300490	0,00%	2,02	0,00	0,70
$2 \times 2 \times 2 \times 2_2$	GA	300490	0,00%	5,52	0,00	0,50
$2 \times 2 \times 2 \times 2_2$	TB	300490	0,00%	3,47	0,00	0,50
$2 \times 2 \times 2 \times 2_3$	HS	372680	0,00%	2,14	0,00	0,67
$2 \times 2 \times 2 \times 2_3$	GA	372680	0,00%	5,47	0,00	0,50
$2 \times 2 \times 2 \times 2_3$	TB	372680	0,00%	3,45	0,00	0,50
$6 \times 1 \times 4 \times 6_1$	HS	35520	0,00%	8,48	2,28	0,50
$6 \times 1 \times 4 \times 6_1$	GA	35520	0,01%	17,53	3,59	1,04
$6 \times 1 \times 4 \times 6_1$	TB	35520	0,01%	11,45	3,22	0,61
$6 \times 1 \times 4 \times 6_2$	HS	26220	0,00%	9,57	1,19	0,50
$6 \times 1 \times 4 \times 6_2$	GA	26220	0,01%	17,57	2,34	1,13
$6 \times 1 \times 4 \times 6_2$	TB	26220	0,01%	11,58	2,44	0,68
$6 \times 1 \times 4 \times 6_3$	HS	38925	0,01%	9,60	2,47	0,49
$6 \times 1 \times 4 \times 6_3$	GA	38925	0,01%	17,61	3,83	0,90
$6 \times 1 \times 4 \times 6_3$	TB	38925	0,01%	11,47	3,76	0,67
$10 \times 2 \times 5 \times 10_1$	HS	460330	0,02%	28,51	52,98	1,77
$10 \times 2 \times 5 \times 10_1$	GA	460330	0,05%	40,52	127,58	2,25
$10 \times 2 \times 5 \times 10_1$	TB	460330	0,02%	33,1	64,83	1,88
$10 \times 2 \times 5 \times 10_2$	HS	725515	0,02%	28,72	83,18	1,61
$10 \times 2 \times 5 \times 10_2$	GA	725515	0,05%	40,75	220,11	2,28
$10 \times 2 \times 5 \times 10_2$	TB	725515	0,02%	33,63	104,55	1,94

Çizelge 5.2 (Devam) : Sanal uygulama sonucunda bulunan özet sonuçlar.

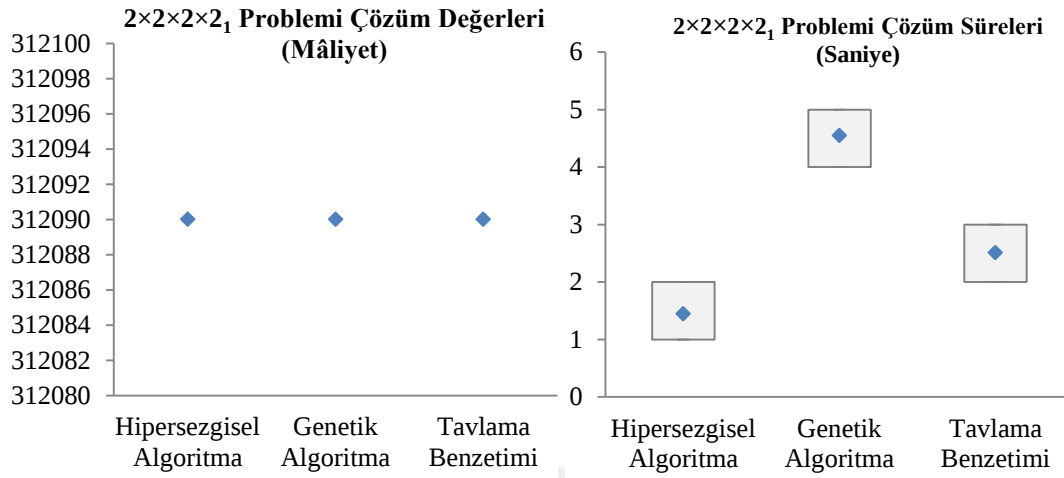
Problem	Yöntem	Bulunan en iyi değer	En iyi çözümden ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
10×2×5×10 ₃	HS	658580	0,02%	28,54	78,97	1,57
10×2×5×10 ₃	GA	658580	0,05%	40,25	177,68	2,58
10×2×5×10 ₃	TB	658580	0,02%	33,52	101,82	1,71
12×2×8×12 ₁	HS	51930	0,05%	57,61	14,80	3,61
12×2×8×12 ₁	GA	51930	0,14%	84,81	45,69	5,16
12×2×8×12 ₁	TB	51930	0,09%	46,15	31,96	3,87
12×2×8×12 ₂	HS	73105	0,05%	57,2	23,04	3,59
12×2×8×12 ₂	GA	73105	0,13%	85,27	61,60	4,89
12×2×8×12 ₂	TB	73105	0,11%	64,79	44,60	3,64
12×2×8×12 ₃	HS	98140	0,04%	57,68	27,71	3,07
12×2×8×12 ₃	GA	98140	0,15%	85,74	87,88	4,75
12×2×8×12 ₃	TB	98140	0,09%	65,36	56,94	3,92
100×2×20×300 ₁	HS	132538	0,49%	203,83	402,55	11,85
100×2×20×300 ₁	GA	132597	1,86%	358,13	1469,04	21,16
100×2×20×300 ₁	TB	132545	1,13%	236,72	934,34	13,40
100×2×20×300 ₂	HS	164718	0,50%	204,28	459,55	11,43
100×2×20×300 ₂	GA	164756	2,23%	360,84	1844,98	21,34
100×2×20×300 ₂	TB	164736	1,32%	237,71	1175,05	14,47

Çizelge 5.2 (Devam) : Sanal uygulama sonucunda bulunan özet sonuçlar.

Problem	Yöntem	Bulunan en iyi değer	En iyi çözümde ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
$100 \times 2 \times 20 \times 300_3$	HS	183052	0,55%	205,41	514,44	11,99
$100 \times 2 \times 20 \times 300_3$	GA	183088	2,33%	355,72	2148,12	20,18
$100 \times 2 \times 20 \times 300_3$	TB	183129	1,16%	236,65	1432,36	14,10
$1000 \times 60 \times 50 \times 300_1$	HS	195496	1,08%	340,45	1147,41	19,37
$1000 \times 60 \times 50 \times 300_1$	GA	195603	3,32%	584,18	4116,54	33,93
$1000 \times 60 \times 50 \times 300_1$	TB	195579	1,97%	394,14	2135,37	22,80
$1000 \times 60 \times 50 \times 300_2$	HS	170814	1,14%	314,02	1006,00	18,94
$1000 \times 60 \times 50 \times 300_2$	GA	171047	3,52%	569,9	3580,16	33,17
$1000 \times 60 \times 50 \times 300_2$	TB	170938	1,92%	362,18	1885,10	21,52
$1000 \times 60 \times 50 \times 300_3$	HS	196330	1,00%	324,19	1198,01	17,97
$1000 \times 60 \times 50 \times 300_3$	GA	197310	3,75%	583,51	3702,96	33,62
$1000 \times 60 \times 50 \times 300_3$	TB	196457	1,96%	393,7	2357,63	22,48

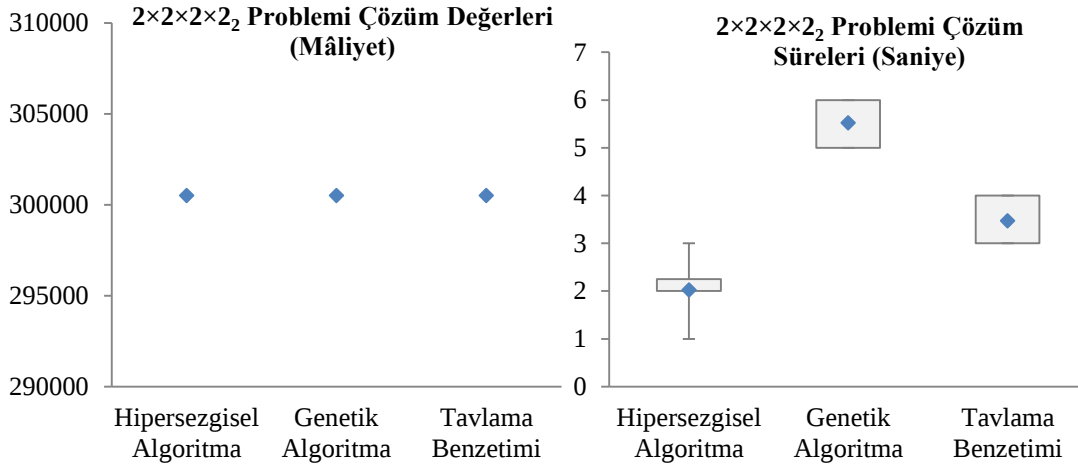
Şekil 5.25'te $2 \times 2 \times 2 \times 2_1$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerini ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri

gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 312090 olup tüm yöntemler bu sonucu elde etmiştir.



Şekil 5.25 : $2 \times 2 \times 2 \times 2_1$ problemi çözüm grafikleri.

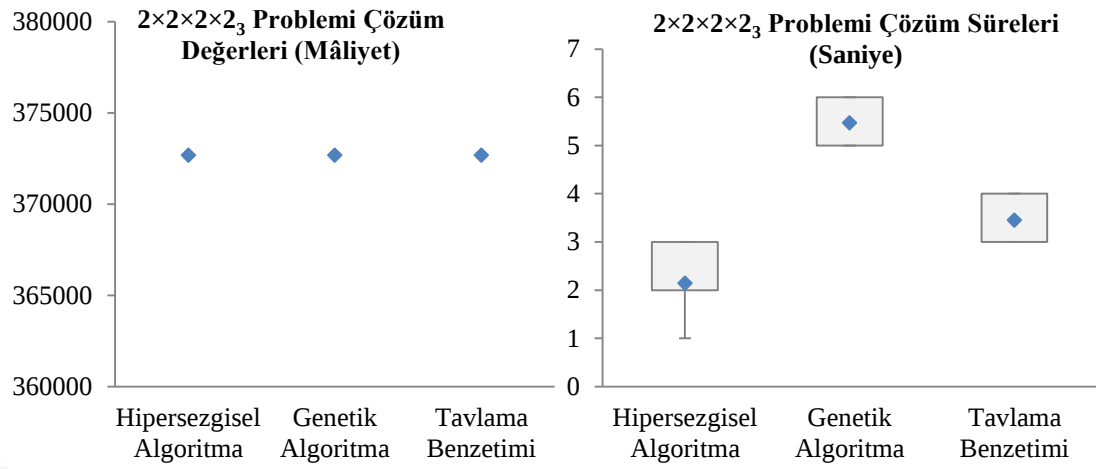
Şekil 5.26'de $2 \times 2 \times 2 \times 2_2$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerini ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 300490 olup tüm yöntemler bu sonucu elde etmiştir.



Şekil 5.26 : $2 \times 2 \times 2 \times 2_2$ problemi çözüm grafikleri.

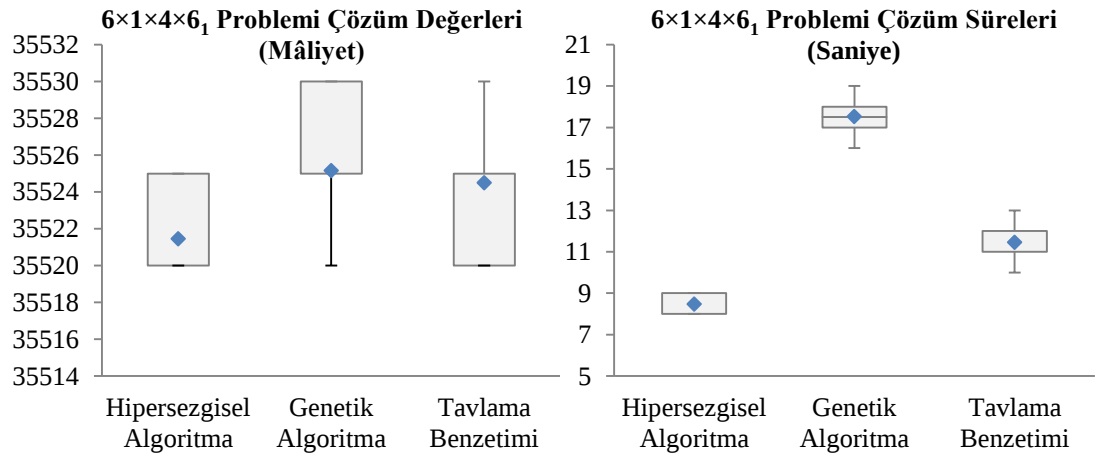
Şekil 5.27'de $2 \times 2 \times 2 \times 2_3$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerini ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri

gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 372680 olup tüm yöntemler bu sonucu elde etmiştir.



Şekil 5.27 : $2 \times 2 \times 2 \times 2_3$ problemi çözüm grafikleri.

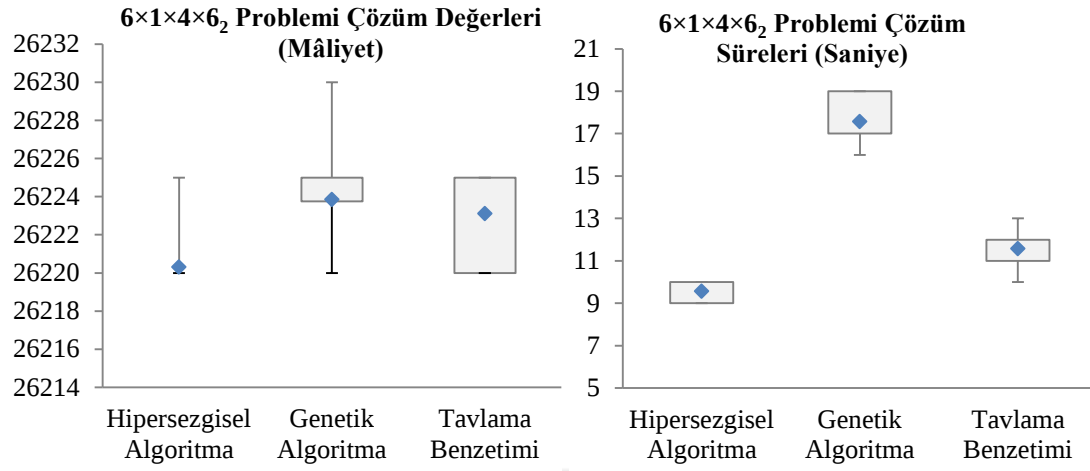
Şekil 5.28’de $6 \times 1 \times 4 \times 6_1$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 35520 olup tüm yöntemler bu çözümü üretebilmiştir.



Şekil 5.28 : $6 \times 1 \times 4 \times 6_1$ problemi çözüm grafikleri.

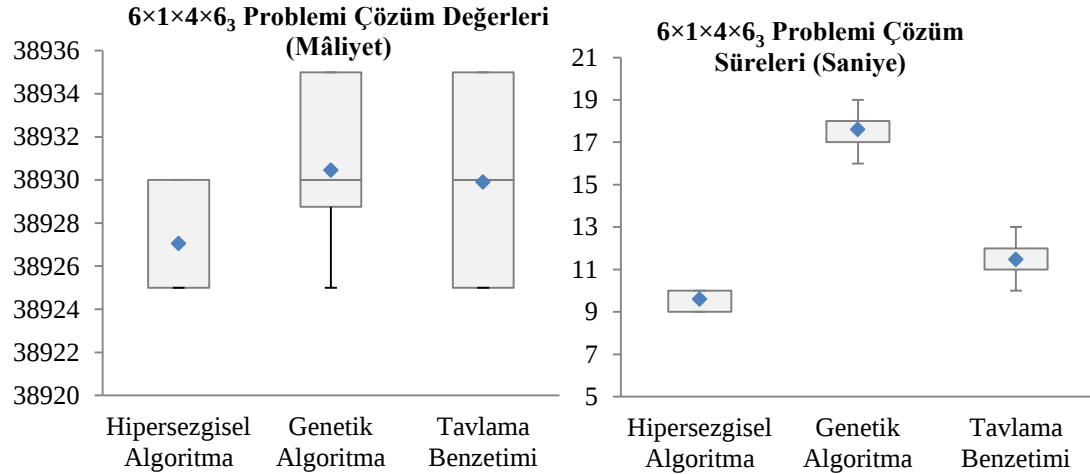
Şekil 5.29’da $6 \times 1 \times 4 \times 6_2$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri

gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 26220 olup tüm yöntemler bu çözümü üretebilmiştir.



Şekil 5.29 : $6 \times 1 \times 4 \times 6_2$ problemi çözüm grafikleri.

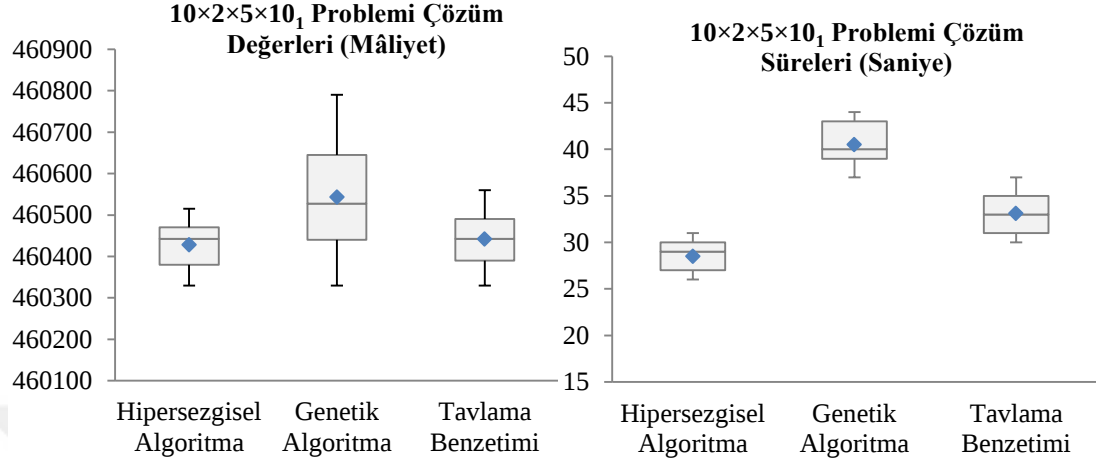
Şekil 5.30'da $6 \times 1 \times 4 \times 6_3$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 38925 olup tüm yöntemler bu çözümü üretebilmiştir.



Şekil 5.30 : $6 \times 1 \times 4 \times 6_3$ problemi çözüm grafikleri.

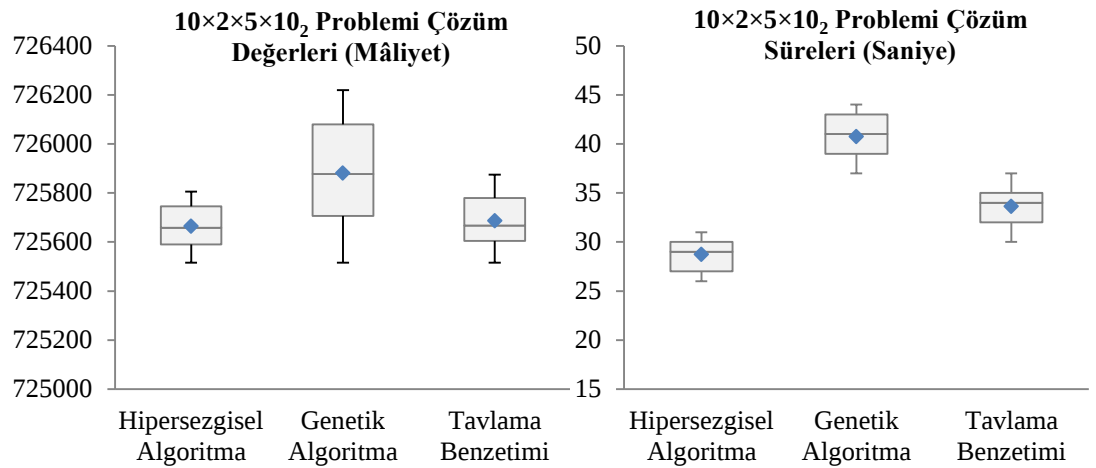
Şekil 5.31'de $10 \times 2 \times 5 \times 10_1$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu

grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 460330 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmıştır.



Şekil 5.31 : 10×2×5×10₁ problemi çözüm grafikleri.

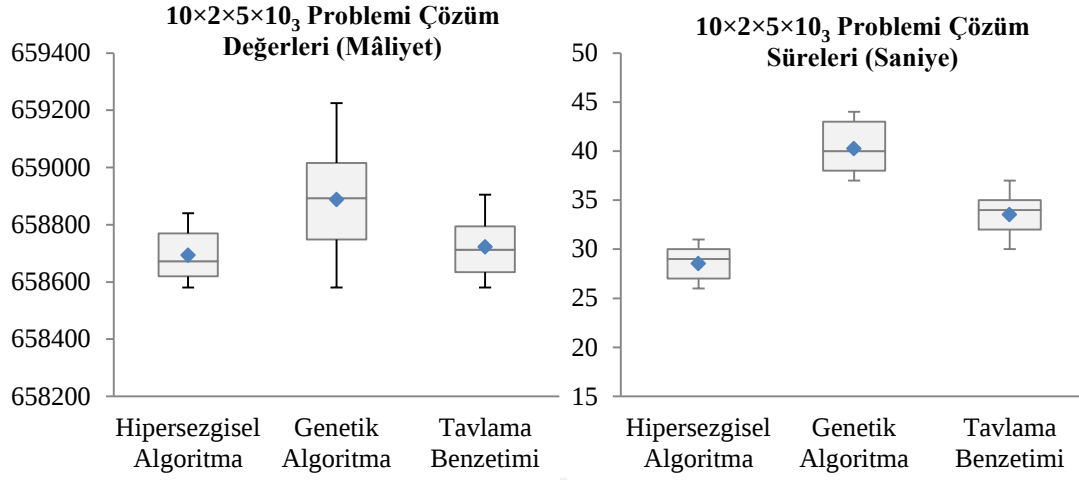
Şekil 5.32’de 10×2×5×10₂ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 725515 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmıştır.



Şekil 5.32 : 10×2×5×10₂ problemi çözüm grafikleri.

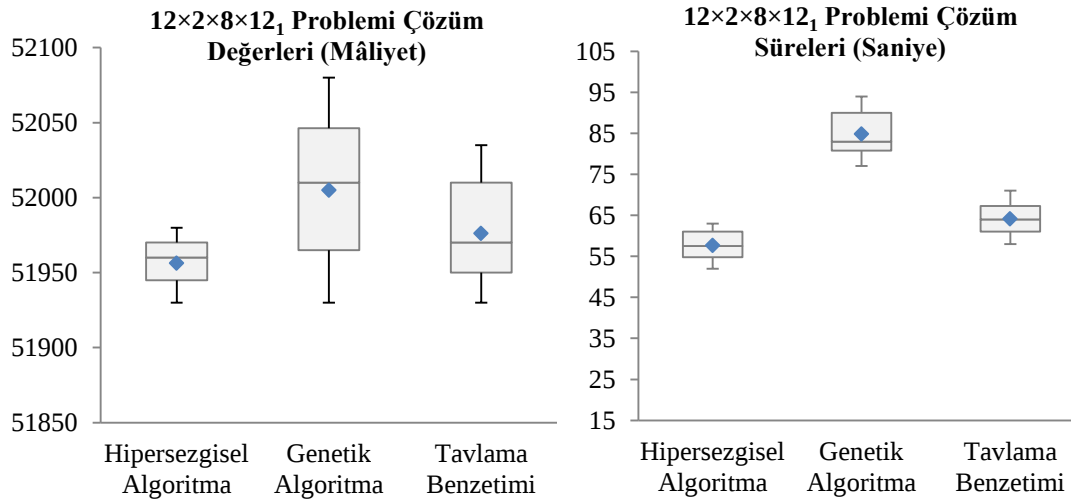
Şekil 5.33’te 10×2×5×10₃ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri

gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 658580 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmiştir.



Şekil 5.33 : 10x2x5x10₃ problemi çözüm grafikleri.

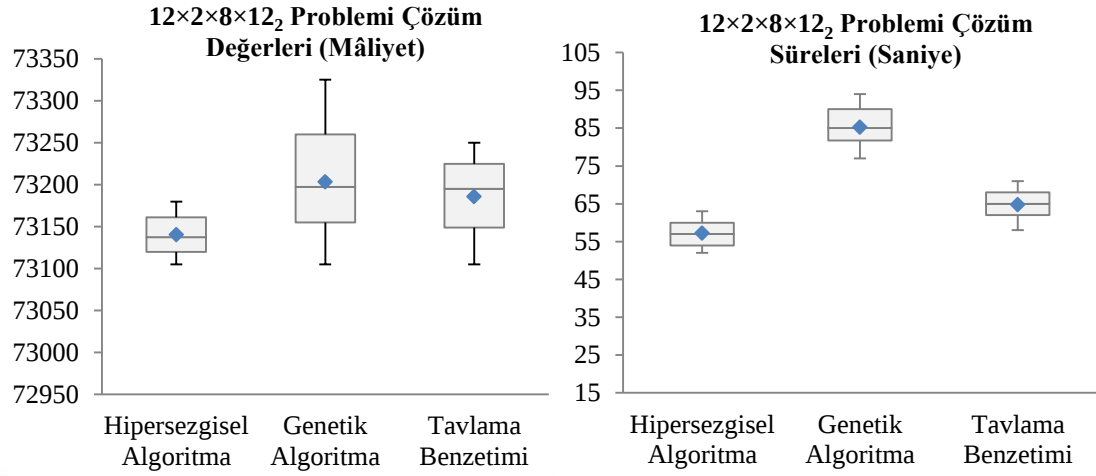
Şekil 5.34'te 12x2x8x12₁ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 51930 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmiştir.



Şekil 5.34 : 12x2x8x12₁ problemi çözüm grafikleri.

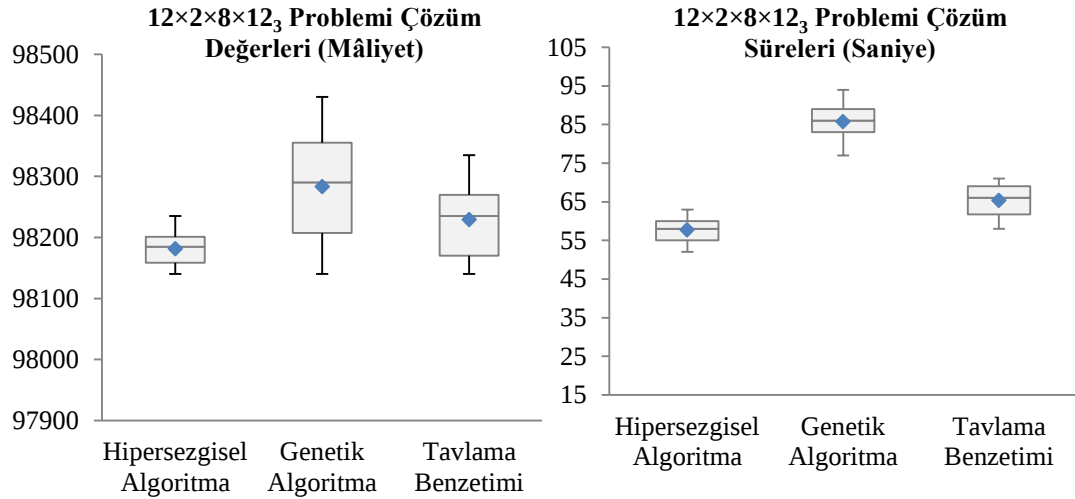
Şekil 5.35'te 12x2x8x12₂ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri

gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 73105 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmiştir.



Şekil 5.35 : 12x2x8x12₂ problemi çözüm grafikleri.

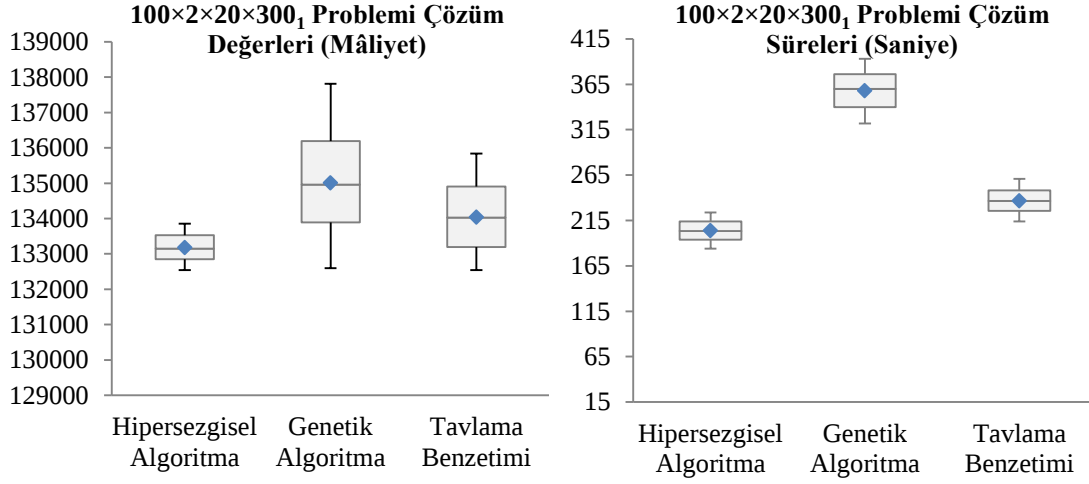
Şekil 5.36'da 12x2x8x12₃ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problem için GAMS uygulaması ile bulunan optimum çözüm 98140 olup tüm yöntemler 100 deneme içinde bu çözüme ulaşabilmiştir.



Şekil 5.36 : 12x2x8x12₃ problemi çözüm grafikleri.

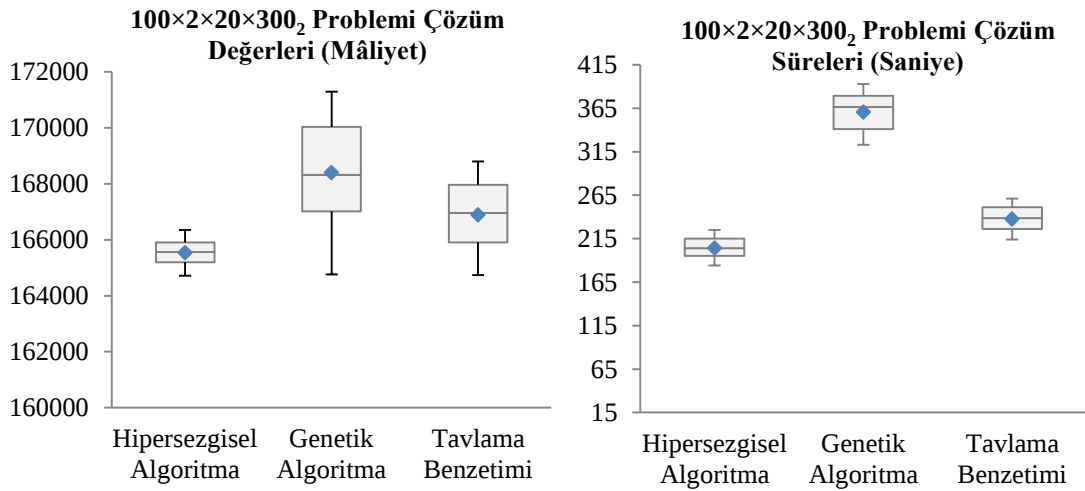
Şekil 5.37'de 100x2x20x300₁ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı

olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 132538 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



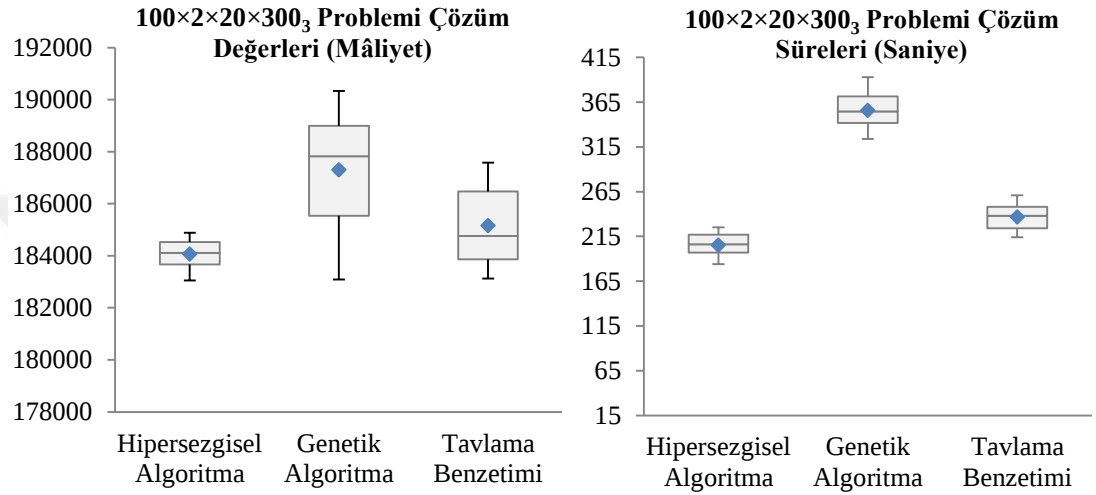
Şekil 5.37 : 100×2×20×300₁ problemi çözüm grafikleri.

Şekil 5.38’de 100×2×20×300₂ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 132538 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



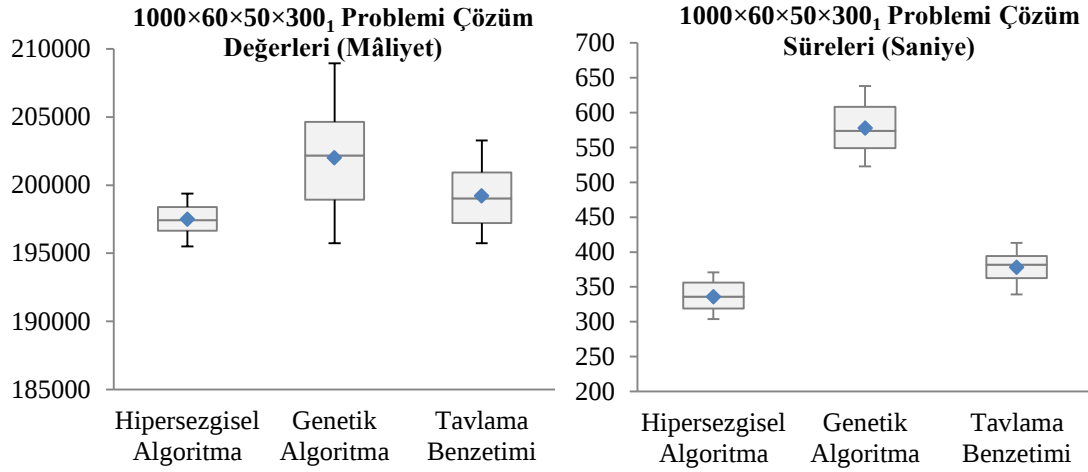
Şekil 5.38 : 100×2×20×300₂ problemi çözüm grafikleri.

Şekil 5.39’da $100 \times 2 \times 20 \times 300_3$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 183052 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



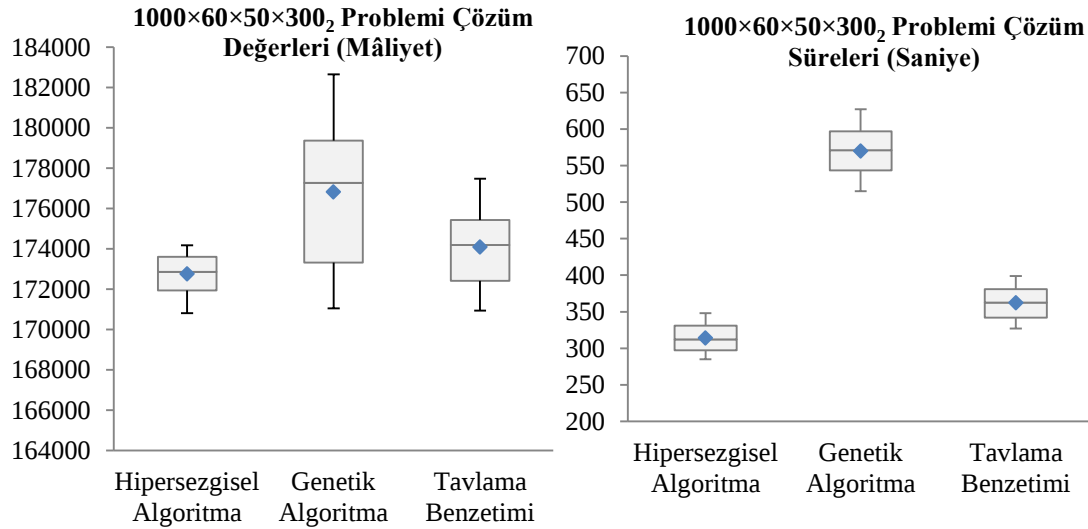
Şekil 5.39 : $100 \times 2 \times 20 \times 300_3$ problemi çözüm grafikleri.

Şekil 5.40’da $1000 \times 60 \times 50 \times 300_1$ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 195496 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



Şekil 5.40 : 1000×60×50×300₁ problemi çözüm grafikleri.

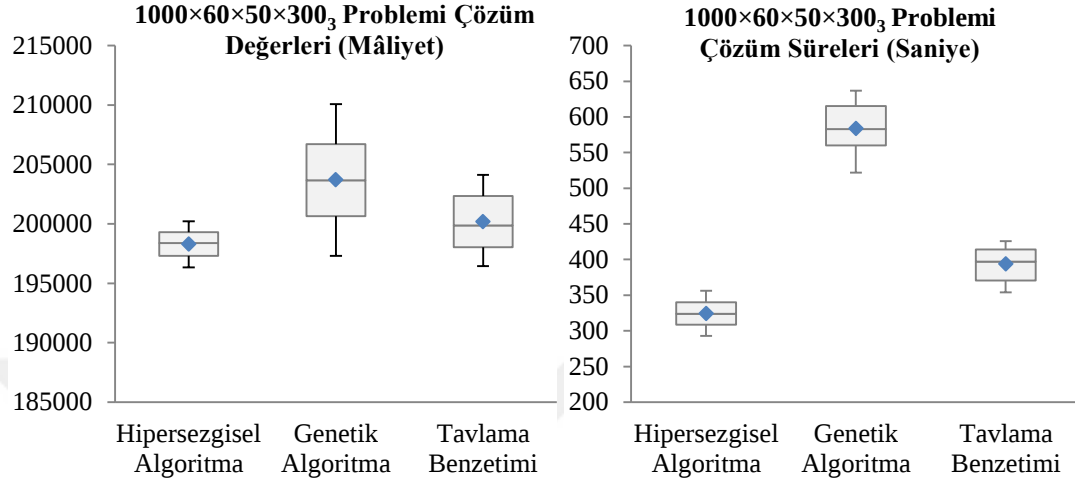
Şekil 5.41’de 1000×60×50×300₂ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 170814 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



Şekil 5.41 : 1000×60×50×300₂ problemi çözüm grafikleri.

Şekil 5.42’de 1000×60×50×300₃ problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu

grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 196330 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



Şekil 5.42 : 1000×60×50×300₃ problemi çözüm grafikleri.

5.4.3 Sanal problemler için sonuçların ve farklılıkların değerlendirilmesi

Tasarlanan sanal Lojistik Ağ Tasarımı Problemleri için yapılan uygulamalara göre aşağıdaki değerlendirmeler yapılabilir:

- Hipersezgisel Algoritma, tüm problemler için en iyi çözümü bulmuştur. Ayrıca ilk dört problem için, diğer yöntemlerle birlikte GAMS ile bulunan optimum sonuçları elde etmiştir.
- Hipersezgisel Algoritma yüz kez çalıştığında bulunan ortalama çözüm değerleri, tüm problemler için diğer yöntemlerden daha iyidir. Son iki sanal problem tipi (100×22×20×300 ve 1000×60×50×300) için diğer yöntemlerin bulamadığı çözümler üretmiştir. Bu sonuçlar ışığında geliştirilen Hipersezgisel Algoritmanın, büyük boyutlu problemler için daha iyi sonuçlar ortaya çıkardığı kanısına varılabilir.
- Hipersezgisel Algoritma'nın bulduğu çözümler incelendiğinde, bulunan değerlerin daha küçük bir aralıkta olduğu ve standart sapmasının daha küçük olduğu bulunmuştur. Bu durum da Hipersezgisel Algoritma'nın diğer yöntemlere göre üretilen sonuçlar açısından daha güçlü ve kararlı olduğunu göstermektedir.

- Ortalama çözüm süreleri dikkate alındığında geliştirilen Hipersezgisel Yöntem, diğer algoritmalara göre en iyi çözüme daha kısa sürede ulaşmıştır (Burada belirtilen çözüm süresi ile algoritmanın işleyişi içinde en iyi çözümü bulma süresi ifade edilmek istenmiştir.).
- Hipersezgisel Algoritma için çözüm sürelerinin standart sapması $-2 \times 2 \times 2 \times 2$ problemlerinkiler dışında- diğer yöntemlere göre daha düşüktür. Bu da Hipersezgisellerin diğer yöntemlere göre çözüm süresi açısından daha etkin olduğunu göstermektedir.
- Çalışmada kullanılan Genetik Algoritma ve Tavlama Benzetimi yöntemi karşılaştırıldığında, Tavlama Benzetimi yönteminin daha başarılı sonuç değerleri ürettiği görülmektedir. Ayrıca Tavlama Benzetimi ile daha kısa süre içerisinde çözümler oluşturulmakta olup, sonuç değerleri ve süre açısından yöntemin daha güçlü olduğu görülmektedir.
- Parametre kümeleri için Genetik Algoritmalarla yapılan denemelerde en iyi sonuçlar 0,4 çaprazlama oranı ve 0,03 mutasyon oranı parametreleri ile elde edilmiştir. Tavlama Benzetimi yönteminde ise en iyi sonuçlara 400 sıcaklık katsayısı ve 0,8 soğutma oranı parametreleri ile ulaşılmıştır.

6. UYGULAMA ÇALIŞMALARI

Bu bölümde iki vaka çalışması açıklanmıştır. İlk olarak ambalaj imalatı yapan bir işletmede LAT problemi ele alınarak geliştirilen Hipersezgisel Algoritma ile sonuç oluşturulmuştur. İkinci olarak da bir lojistik firmasında sektöre özgü bir problem için uygulama yapılmıştır. Her iki uygulamadan elde edilen sonuçlar sanal problemlerde olduğu gibi irdelenerek karşılaştırmalar yapılacaktır. Hazırlanan yazılım Intel i7 işlemcili 8 GB Ram bellekli bilgisayarda çalıştırılmıştır.

6.1 Bir İmalat Firmasında Uygulama

6.1.1 İmalat firmasının tanıtımı

Bu tez çalışmasında ele alınan dört katmanlı LAT probleminin gerçek yaşamda uygulamasının yapılabilmesi için firmanın aşağıdaki ölçütlere sahip olması gerekmektedir:

- Aynı ürünü üreten birden çok fabrika ve tesisin olması.
- Tesislerin aynı coğrafi bölgede bulunması.
- Tesislerin birbirlerinin eşdeğeri olabilmesi.
- Uygulamada diğer yöntemlerle sağlıklı karşılaştırma yapılabilmesi için birçok müşterisi olması.
- Alınan verilerin doğruluğu açısından kurumsal ve şeffaf olması.

Bu ölçütler doğrultusunda Türkiye'nin birçok yerinde tesisleri bulunan ve ambalaj sektörünün lider firmalarından Mondi Tire Kutsan firması seçilerek LAT için uygulama gerçekleştirilmiştir.

Oluklu mukavva ambalaj sektöründe üretim yapan Mondi Tire Kutsan şirketi Türkiye'de pazar lideri olarak faaliyetlerini sürdürmektedir. Firmanın ana faaliyet konusu, oluklu mukavva kutu kağıt ve hurda kağıt üretimi ve satışlarıdır.

Şirket 1974 yılında İzmir Tire’de kurulmuştur. 1991 yılından bu yana İstanbul Menkul Kıymetler Borsası’nda işlem görmektedir. 2007 yılının sonlarında dünya kağıt ve ambalaj sanayi sektörünün öncülerinden Mondi Group bünyesine katılmıştır. Firma şu an 1000’in üzerinde çalışan ile 700 müşteriye üretim yapmaktadır.

6.1.2 İmalat firmasında ele alınan LAT problemi ve bulunan sonuçlar

Türkiye’de 6 adet kutu fabrikası, 4 adet depo, 4 dağıtım merkezi ve 700 müşteriye sahip firmada gerçeğe uygun veriler alınarak Lojistik Ağ Tasarımı Problemi üzerinde uygulama yapılmıştır. Dört katmanlı LAT problemi üzerinde dağıtım merkezi ve müşteriler arasında rotalama da gerçekleştirilmektedir.

Uygulama probleminde, literatürde yer alan 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında tasarlanan TB algoritması üzerinde denemeler yapılarak karşılaştırmalar gerçekleştirilmiştir. Çizelge 6.1’de, uygulamada kullanılan parametre değerleri yer almaktadır.

Çizelge 6.1 : Gerçek uygulamada kullanılan parametreler.

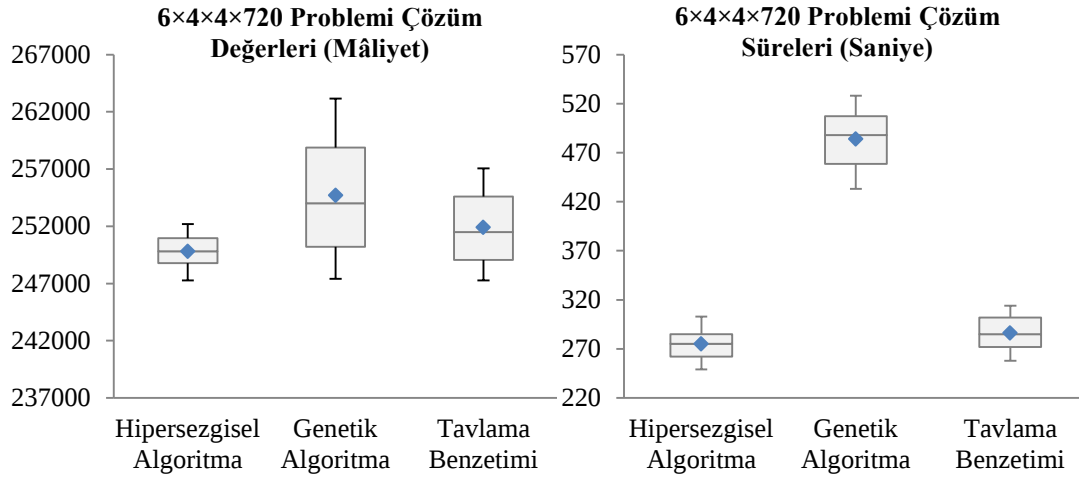
Yöntem	Problem	Parametre	Değer
Hipersezgisel Algoritma	6×4×4×720	İterasyon Sayısı	300
Genetik Algoritma	6×4×4×720	İterasyon Sayısı	500
		Popülasyon Sayısı	100
		Çaprazlama Oranı	0,2, 0,3, 0,4, 0,5 ve 0,6
		Mutasyon Oranı	0,03, 0,04, 0,05, 0,06 ve 0,07
Tavlama Benzetimi	6×4×4×720	İterasyon Sayısı	300
		Sıcaklık Katsayısı	200, 300, 400, 500 ve 600
		Soğutma Oranı	0,75, 0,8, 0,85, 0,90 ve 0,95

Bu çalışmada, yöntemler her bir problemde her bir parametre kümesi için 100 kez uygulanmış olup, bulunan çözüm değerleri, en iyi çözümle arasındaki fark, buna ait sapma yüzdesi ve çözüm süreleri hesaplanmıştır. Daha sonra yöntemler için parametre kümeleri içinde yapılan denemelerde bulunan en iyi 100 çözüm ele alınarak karşılaştırmaları yapılmıştır. Bu raporda yer alan çözüm sürelerinden kasıt, algoritmanın çalışma süreci boyunca bulduğu en iyi çözüme ulaşma süresidir. Bulunan sonuçlar Ek K bölümünde listelenmiştir. Çizelge 6.2’de ise uygulamada elde edilen sonuçlar gösterilmektedir.

Çizelge 6.2 : Gerçek uygulamada üretilen özet sonuçlar.

Problem	Yöntem	Bulunan en iyi değer	En iyi çözümden ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
Gerçek Uygulama (6×4×4×720)	HS		1,03%	274,84	1374,28	15,44
Gerçek Uygulama (6×4×4×720)	GA		3,01%	483,72	4931,75	27,99
Gerçek Uygulama (6×4×4×720)	TB		1,87%	286,02	3062,54	17,42

Şekil 6.1’de 6×4×4×720 problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 247258 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



Şekil 6.1 : 6x4x4x720 problemi çözüm grafikleri.

6.1.3 İmalat firmasında uygulama sonuçlarının değerlendirilmesi

Türkiye’de 6 adet kutu fabrikası, 4 adet depo, 4 dağıtım merkezi ve 700 müşteriye sahip firmada gerçeğe uygun veriler alınarak hazırlanan bilgisayar programı üzerinden sonuçlar elde edilmiştir. Lojistik Ağ Tasarımı Problemi için yapılan uygulamalara göre aşağıdaki değerlendirmeler yapılabilir:

- Hipersezgisel Algoritma, vaka problemi için en iyi çözümü bulmuştur.
- Hipersezgisel Algoritma yüz kez çalıştığında bulunan ortalama çözüm değerleri, tüm problemler için diğer yöntemlerden daha iyidir.
- Hipersezgisel Algoritma’nın bulduğu çözümler incelendiğinde bulunan değerler daha küçük bir aralıkta olduğu ve standart sapmasının daha küçük olduğu görülmüştür. Bu durum da Hipersezgisel Algoritma’nın diğer yöntemlere göre üretilen sonuçlar açısından daha güçlü ve kararlı olduğunu göstermektedir.
- Ortalama çözüm süreleri dikkate alındığında geliştirilen Hipersezgisel Yöntem, diğer algoritmalara göre en iyi çözüme daha kısa sürede ulaşmıştır (Burada belirtilen çözüm süresi ile algoritmanın işleyişi içinde en iyi çözümü bulma süresi ifâde edilmek istenmiştir.).
- Hipersezgisel Algoritma için çözüm sürelerinin standart sapması diğer yöntemlere göre daha düşüktür. Bu da Hipersezgisellerin diğer yöntemlere göre çözüm süresi açısından daha etkin olduğunu göstermektedir.

- Çalışmada kullanılan Genetik Algoritma ve Tavlama Benzetimi yöntemi karşılaştırıldığında, Tavlama Benzetimi yönteminin daha başarılı sonuç değerleri ürettiği görülmektedir. Ayrıca Tavlama Benzetimi ile daha kısa süre içerisinde çözümler oluşturulmakta olup, sonuç değerleri ve süre açısından yöntemin daha güçlü olduğu görülmektedir.
- Parametre kümeleri için Genetik Algoritmalarla yapılan denemelerde en iyi sonuçlar 0,4 çaprazlama oranı ve 0,03 mutasyon oranı parametreleri ile elde edilmiştir. Tavlama Benzetimi yönteminde ise en iyi sonuçlara 400 sıcaklık katsayısı ve 0,8 soğutma oranı parametreleri ile ulaşılmıştır.

6.2 Bir Lojistik Firmasında Uygulama

6.2.1 Lojistik firmasının tanıtımı

Uluslararası hızlı taşımacılık alanında faaliyet gösteren DHL Express, 220 ülkede 350.000 çalışanı ile dünya çapında önde gelen posta ve lojistik şirketi Deutsche Post DHL grubunun bir alt firmasıdır. Organizasyon dünyanın en büyük global Express ağına sahiptir ve âcil doküman ve paketleri hızlı bir şekilde kapıdan kapıya taşımaktadır.

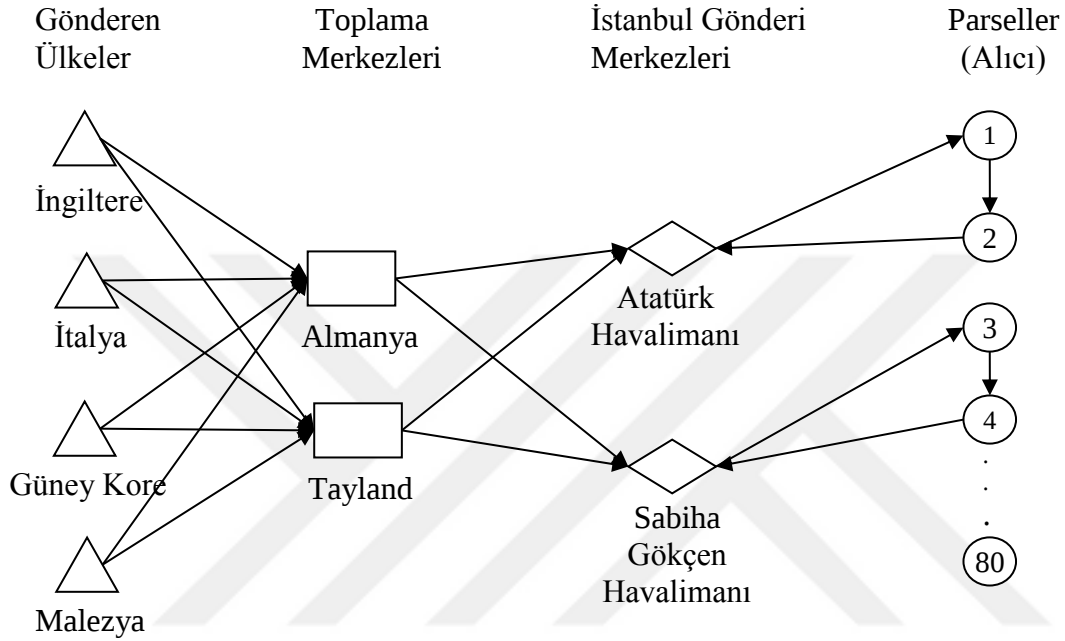
DHL Express özellikle dünyanın genişlemekte olan hareketli e-ticaret pazarlarında taşımacılık faaliyetlerini hedeflemektedir. DHL ayda 100.000'in üzerinde Express gönderiyi dünyanın dört bir yanına taşımakta ve bir o kadar gönderiyi Türkiye'ye getirmektedir. DHL, Türkiye'de 2006 yılından bugüne IATA sıralamasındaki liderliğini sürdürmektedir.

6.2.2 Lojistik firmasında ele alınan LAT problemi ve bulunan sonuçlar

Bu bölümde DHL Express için İngiltere, İtalya, Güney Kore ve Malezya'dan Türkiye'ye gönderilen 270 gönderi ve gönderim noktası olan 80 parsel için dört katmanlı İleri Lojistik Ağ Tasarımı probleminin uygulaması yapılacaktır. Uygulamada İstanbul gönderi merkezleri ile parseller arasında rotalama süreci gerçekleştirilecektir.

DHL Express'in taşımacılık ağında Türkiye, Avrupa bölgesine bağlı olup dağıtım merkezi (Hub) olarak gönderileri Almanya'da Leipzig'den ulaşmaktadır. Buna göre Avrupa bölgesinden Türkiye'ye gönderilen gönderiler ilk olarak Leipzig toplama

merkezine, daha sonra burada sıralama işlemine sokularak Türkiye'ye ulaştırılmaktadır. Aynı şekilde DHL Express'in Asya kıtasında Tayland'da Bangkok sınırları içinde bir toplama merkezi bulunmaktadır. Bu toplama merkezinden çeşitli ülkelere uçaklar kalkmaktadır. Bu uygulamada dört ülkeden Türkiye'ye gelecek gönderiler için gerçeğe yakın veriler kullanılmıştır. Aşağıda Şekil 6.2'de ele alınan ileri lojistik ağ gösterilmektedir.



Şekil 6.2 : Lojistik firması için ele alınan ileri lojistik ağı.

Bu lojistik ağında dört ülkeden taşınacak gönderiler öncelikle toplama merkezlerine ulaştırılacak, daha sonra İstanbul gönderi merkezlerine uçakla taşınarak buradan kuryeler ile dağıtımı gerçekleştirilecektir. Kurulan dört katmanlı LAT problemi üzerinde gönderi merkezi ve parseller arasında rotalama da gerçekleştirilmektedir. Modelde ayrıca araç kapasiteleri de dikkate alınmıştır. Bir kurye aracı ile en fazla 40 gönderi taşınabilmektedir.

Uygulama probleminde, literatürde yer alan 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında tasarlanan TB algoritması üzerinde denemeler yapılarak karşılaştırmalar gerçekleştirilmiştir. Çizelge 6.3'te, uygulamada kullanılan parametre değerleri yer almaktadır.

Çizelge 6.3 : Lojistik firması uygulamasında kullanılan parametreler.

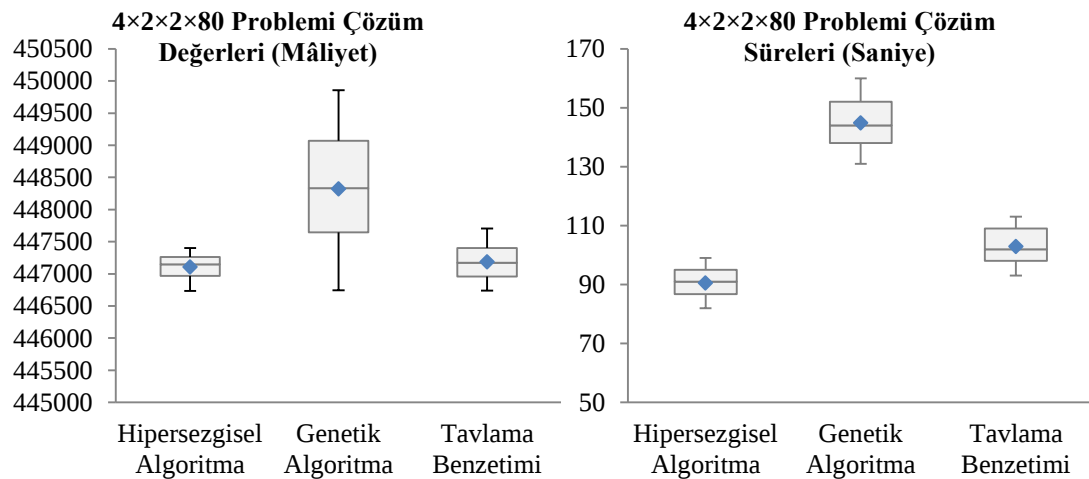
Yöntem	Problem	Parametre	Değer
Hipersezgisel Algoritma	4×2×2×80	İterasyon Sayısı	300
Genetik Algoritma	4×2×2×80	İterasyon Sayısı	500
		Popülasyon Sayısı	100
		Çaprazlama Oranı	0,2, 0,3, 0,4, 0,5 ve 0,6
		Mutasyon Oranı	0,03, 0,04, 0,05, 0,06 ve 0,07
Tavlama Benzetimi	4×2×2×80	İterasyon Sayısı	300
		Sıcaklık Katsayısı	200, 300, 400, 500 ve 600
		Soğutma Oranı	0,75, 0,8, 0,85, 0,90 ve 0,95

Bu çalışmada, yöntemler her bir problemde her bir parametre kümesi için 100 kez uygulanmış olup, bulunan çözüm değerleri, en iyi çözümle arasındaki fark, buna ait sapma yüzdesi ve çözüm süreleri hesaplanmıştır. Daha sonra yöntemler için parametre kümeleri içinde yapılan denemelerde bulunan en iyi 100 çözüm ele alınarak karşılaştırmaları yapılmıştır. Bu raporda yer alan çözüm sürelerinden kasıt, algoritmanın çalışma süreci boyunca bulduğu en iyi çözüme ulaşma süresidir. Bulunan sonuçlar Ek L bölümünde listelenmiştir. En iyi çözüm Hipersezgisel Yöntem ile bulunmuştur; en düşük mâliyet olarak 446736 değeri hesaplanmıştır. Çizelge 6.4'te ise uygulamada elde edilen en iyi çözümle farklar gösterilmektedir.

Çizelge 6.4 : Lojistik firması uygulamasında üretilen özet sonuçlar.

Problem	Yöntem	Bulunan en iyi değer	En iyi çözümden ortalama sapma (%)	Ortalama çözüm süresi (sn)	Çözüm değeri standart sapma	Çözüm süresi standart sapma
Gerçek Uygulama (4×2×2×80)	HS	446736	0,08%	90,55	195,99	5,14
Gerçek Uygulama (4×2×2×80)	GA		0,36%	144,79	914,13	8,52
Gerçek Uygulama (4×2×2×80)	TB		0,10%	102,87	279,49	5,80

Şekil 6.3'te 4×2×2×80 problemi için Hipersezgisel Algoritma, Genetik Algoritma ve Tavlama Benzetimi yöntemlerince en iyi 100 denemede bulunan çözüm değerlerinin ve deneme sırasında en iyi çözümün üretilme süresinin kutu grafikleri gösterilmektedir. Bu problemin GAMS uygulaması ile çözülmesi olanaklı olmadığı için denemeler boyunca bulunan en iyi çözüm değeri olan 446736 baz alınmıştır. Bu çözüm değerini salt bu çalışma kapsamında geliştirilen Hipersezgisel Algoritma üretebilmiştir.



Şekil 6.3 : 4×2×2×80 problemi çözüm grafikleri.

6.2.3 Lojistik firmasında uygulama sonuçlarının değerlendirilmesi

DHL Express firmasından gerçeğe uygun veriler alınarak hazırlanan bilgisayar programı üzerinden çözüm değerleri elde edilmiştir. Lojistik Ağ Tasarımı Problemi için yapılan uygulamalara göre aşağıdaki değerlendirmeler yapılabilir:

- Hipersezgisel Algoritma, vaka problemi için en iyi çözümü bulmuştur.
- Hipersezgisel Algoritma yüz kez çalıştırıldığında bulunan ortalama çözüm değerleri, tüm problemler için diğer yöntemlerden daha iyidir.
- Hipersezgisel Algoritma'nın bulduğu çözümler incelendiğinde bulunan değerler daha küçük bir aralıkta olduğu ve standart sapmasının daha küçük olduğu bulunmuştur. Bu durum da Hipersezgisel Algoritma'nın diğer yöntemlere göre üretilen sonuçlar açısından daha güçlü ve kararlı olduğunu göstermektedir.
- Ortalama çözüm süreleri dikkate alındığında geliştirilen Hipersezgisel Yöntem, diğer algoritmalara göre en iyi çözüme daha kısa sürede ulaşmıştır (Burada belirtilen çözüm süresi ile algoritmanın işleyişi içinde en iyi çözümü bulma süresi ifade edilmek istenmiştir.).
- Hipersezgisel Algoritma için çözüm sürelerinin standart sapması diğer yöntemlere göre daha düşüktür. Bu da Hipersezgisellerin diğer yöntemlere göre çözüm süresi açısından daha etkin olduğunu göstermektedir.
- Çalışmada kullanılan Genetik Algoritma ve Tavlama Benzetimi yöntemleri karşılaştırıldığında, Tavlama Benzetimi yönteminin daha başarılı sonuç değerleri ürettiği görülmektedir. Ayrıca Tavlama Benzetimi ile daha kısa süre içerisinde çözümler oluşturulmakta olup, sonuç değerleri ve süre açısından yöntemin daha güçlü olduğu görülmektedir.
- Parametre kümeleri için Genetik Algoritmalarla yapılan denemelerde en iyi sonuçlar 0,4 çaprazlama oranı ve 0,03 mutasyon oranı parametreleri ile elde edilmiştir. Tavlama Benzetimi yönteminde ise en iyi sonuçlara 400 sıcaklık katsayısı ve 0,8 soğutma oranı parametreleri ile ulaşılmıştır.



7. SONUÇLAR

Bu tez çalışmasında İleri Lojistik Ağ Tasarımı problemi için bir Karma Tamsayılı Programlama Modeli kurularak Lojistik Ağ Tasarımı süreci irdelenmiş ve ayrıntılı bir literatür incelemesi yapılmıştır. Ele alınan LAT problemi dört katmanlı, tek ürünlü, tek dönemli, deterministik talepli ve kapasite kısıtlıdır. Ayrıca problemde dağıtım merkezi ve müşteriler arasında araç rotalama yapılmaktadır. Literatürdeki çalışmalar dikkate alındığında, günümüzün dinamik koşullarında karşılaşılan ve Karma Tamsayılı Programlama Modeli kurulan dört katmanlı Lojistik Ağ Tasarımı Problemi için etkin çözümler üretebilecek ve değişikliklere hızlı bir şekilde uyum sağlayabilecek bir yöntemin, bilgilerimiz ışığında henüz geliştirilmediği ve uygulamasının yapılmadığı anlaşılmaktadır. Literatürde bu boşluğun doldurulması için son yıllarda sıkça araştırma konusu olan ve özellikle esneklik konusunda ön plâna çıkan Hipersezgisel Algoritmalar bu tez çalışmasının ana çıkış noktasını oluşturmaktadır. Hipersezgisel Algoritmalar probleminden bağımsız teknikler olup, çözüm arama süreci ile problem uzayı arasında bir soyutlama sağlamaktadır. Bu çalışmada, dinamik/uyarlanabilir (adaptif) bir Hipersezgisel Algoritma önerisi sunulmuş ve sürecin sektörel bazda uygulama tasarımları ile çözüm seçenekleri değerlendirilmiştir. Geliştirilen Hipersezgisel Algoritma'nın adaptif özelliğini göstermek için ise tez çalışmasında farklı LAT problemleri kurgulanmış ve hazırlanan yöntemin bu problemler üzerinde ayrıntılı olarak uygulaması yapılmıştır. Bu çalışma ile İleri LAT probleminde katman sayısı, kapasite, rotalama ve kısıtlar üzerinde esneklik sağlanmıştır.

Hipersezgisel yöntemler salt on beş yıllık bir geçmişe sahip, yâni yeni bir yöntem olmalarından dolayı literatürde sınırlı alanlarda uygulaması yapılmıştır. Şimdiye kadar Garrido ve Castro'nun 2009 yılında Kapasite Kısıtlı Araç Rotalama Problemi için oluşturulduğu yayından başka Lojistik Ağ Tasarımı konusu içinde yer alan çalışmaya rastlanmamıştır. Sözü geçen çalışmada ise salt araç rotalama süreci işlenmiş, tedârikçi seçimi, tesisler arası akış gibi diğer Lojistik Ağ Tasarımı konuları ele alınmamıştır. Ayrıca uygulama, literatürde bulunan problem kümeleri üzerinde

gerçekleştirilmiş, gerçek yaşamda karşılaşılan çok dönemli ağ tasarımı problemi için geçerliliği sınanmamıştır.

Bu tez çalışmasında, genel olarak literatürde bulunan bu eksikliği gidermek amacıyla çok aşamalı ve tesis kapasiteli Lojistik Ağ Tasarımı Problemine uygulanacak bir Hipersezgisel Algoritma geliştirilmiştir. Temel olarak tez çalışmasındaki çıktılar şu şekilde sıralanabilir:

- İşletme kapsamında aktifleştirilecek tesis (fabrika, dağıtım merkezi ve depo) lokasyonlarının, müşteri ve tedârikçilerin dikkate alınarak seçilmesi
- Depo ve dağıtım merkezlerindeki stok miktarlarının belirlenmesi
- Hammadde, yarı ürün ve ürün akışlarının belirlenmesi
- Uygun atamaların ve dağıtım merkezi-müşteri rotalarının saptanması

Lojistik Ağ Tasarımı problemleri, karar düzeyinde (stratejik, taktik ve operasyonel), işlevler açısından (lokasyon belirleme, araç rotalama, depo miktarı, vb.), çözüm yöntemleri (kesin, sezgisel yöntemler) ve uygulama alanları açısından farklı boyutlarda ele alınabilmektedir. Bu tez çalışmasının amacı, gerçek yaşamda karşılaşılan Lojistik Ağ Tasarımı Problemi için bu boyutlarda sağlam (robust), esnek ve kaliteli çözümleri kabul edilebilir bir sürede üreten bir Hipersezgisel Algoritma hazırlamak ve gerçek bir süreç üzerinde uygulamasını yapmaktır. Literatür araştırmasına göre bu tez çalışması daha önceki çalışmalardan aşağıdaki yönlerden ayrılmaktadır:

- Lojistik Ağ Tasarımı problemi için daha önce Hipersezgisel Yöntemlerin kullanıldığı bir çalışma yapılmamıştır. LAT alt problemlerine bakıldığında ise Garrido ve Castro'nun 2009 yılında tek katman üzerinde Kapasite Kısıtlı Araç Rotalama Problemi konusundaki yayınından başka Lojistik Ağ Tasarımı kapsamına giren bir çalışma yapılmadığı görülmektedir.
- Ağ tasarımı problemleri kapsamında değerlendirildiğinde ise Hipersezgisel yöntemlerin Su Ağ Tasarımı üzerinde uygulamaları görülmektedir. Bu problem tipinde katman yapısı, rotalama, kapasite gibi Lojistik Ağ Tasarımı özellikleri bulunmadığından, problem doğası itibarıyla LAT'a göre daha basit bir problemdir.

- Daha önce yapılan LAT arařtırmalarından farklı olarak bu tez alıřmasında odaklanılan LAT problemi; drt katmanlı ađ modelini ele almakta, kapasite kısıtını kullanmakta, atamanın yanında rotalama srecini de iermekte ve kullanılan yntem olarak salt basit sezgisel yntem algoritma havuzu ile bulguları elde etmektedir.
- Bu tez kapsamında geliřtirilen zm yntemi hem sanal hem de gerek problemler zerinde uygulanmıřtır.

Bu alıřmada, İleri Lojistik Ađ Tasarımı problemi iin geliřtirilen Hipersezgisel Algoritma literatrde bulunan Metasezgisel yntemlerle aynı problemler zerinde zm karřılařtırılmaları yapılmıřtır. Karřılařtırmalar iin, ncelikle farklı boyutlarda altı problem tipi ele alınarak toplam 18 sanal Lojistik Ađ Tasarımı Problemi hazırlanmıř ve farklı parametre deđerleri ile denemeler yapılmıřtır. Gerek yařam uygulaması olarak birincisi paketleme sektrnden alınan verilerle, ikincisi lojistik sektrnden alınan verilerle denemeler gerekleřtirilmiř ve sonular analiz edilmiřtir. Sonular incelendiđinde, bu tezde yapılandırılan Hipersezgisel Algoritmanın, uygun sreler iinde tm problemlerde en iyi zm elde ettiđi grlmektedir.

Bu tez alıřması Tek rnl ok Ařamalı İleri Lojistik Ađ Tasarımı problemi konusunda Hipersezgisel Algoritma uygulaması ile ayrıcalık gstermektedir. Tasarlanan Hipersezgisel Algoritma, gelecekte Tersine ve Btnleřik Lojistik Ađ Tasarımı problemi veya ok rnl Lojistik Ađ Tasarımı problemi zerinde zm retebilecek řekilde geliřtirilebilir. Bu tez alıřması, Lojistik Ađ Tasarımı problemleri iin gnmzn deđiřen kořullarına hızlı bir řekilde uyum sađlayabilecek, zm kalitesi ve zm sresi ile gerek yařamda uygulanabilir yntemler iin bir kaynak niteliđi tařımaktadır.

Gelecekte bu tez konusu kapsamında ařađıdaki ynde alıřmalar yapılabilir:

- Farklı rn sayıları ile Lojistik Ađ Tasarımı problemi oluřturulabilir.
- Deđerřken mřteri talepleri ile kurulacak probleme Hipersezgisel Yntem uygulanabilir.
- Yeni bir tesisin aılmasının olanaklı olabileceđi senaryolarla Lojistik Ađ Tasarımı kurgulanabilir.

- Çok dönemli Lojistik Ağ Tasarımı problemine yönelik uygulama yapılabilir.
- Karayolu taşımacılığı dışındaki farklı taşıma modları ile Lojistik Ağ Tasarımı yapılabilir.
- LAT problemi üzerinden Metasezgisel yöntemlerle Hipersezgisel Algoritmalar da denenebilir.

Bu tez çalışmasında vurgulanan kuramsal ve uygulamalı sonuçlar, hem araştırmacılar hem de lojistik ağ yöneticileri için yeni bir bakış açısı oluşturmaktadır.



KAYNAKLAR

- Ahmed, L. N., Özcan, E., Kheiri, A. (2015). Solving high school timetabling problems worldwide using selection hyper-heuristics, *Expert Systems with Applications*, Sayı: 42, Sayfa: 5463–5471.
- Alcaraz, J. A. S., Ochoa, G., Swan, J., Carpio, M., Puga, H., & Burke, E. K. (2014). Effective learning hyper-heuristics for the course timetabling problem, *European Journal of Operational Research*, Sayı: 238, Sayfa: 77-86.
- Alcaraz, J. A. S., Özcan, E., Swanc, J., Kendall, G., Carpio, M. (2016). Iterated local search using an add and delete hyper-heuristic for university course timetabling, *Applied Soft Computing*, Sayı: 40, Sayfa: 581–593.
- Alcaraz, J. A. S., Ochoa, G., Figeroa, M. A. S., Burke, E. K. (2017). A methodology for determining an effective subset of heuristics in selection hyper-heuristics, *European Journal of Operational Research*, Sayı: 260, Sayfa: 972–983.
- Alshamsi, A., Diabat, A. (2015). A reverse logistics network design, *Journal of Manufacturing Systems*, Sayı: 37, Sayfa: 589–598.
- Alshamsi, A., Diabat, A. (2017). A Genetic Algorithm for Reverse Logistics network design: A case study from the GCC, *Journal of Cleaner Production*, Sayı: 151, Sayfa: 652-669.
- Altıparmak, F., Gen, M., Lin, L., Paksoy, T. (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks, *Computers and Industrial Engineering*, Sayı: 51, Sayfa: 197–216.
- Amiri, A. (2006). Designing a distribution network in a supply chain system: Formulation and efficient solution procedure, *European Journal of Operational Research*, Sayı: 171, Sayfa: 567–76.
- Aras, N., Aksen, D., Tanugur, A. G. (2008). Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles, *European Journal of Operational Research*, Sayı: 191, Sayfa: 1223–1240.
- Asta, s., Karapetyan, D., Kheiri, A., Özcan, E., Parkes, A. J. (2016). Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem, *Information Sciences*, Sayı: 373, Sayfa: 476–498.
- Ayob, M., Kendall, G. (2003). A Monte Carlo Hyper-Heuristic To Optimise Component Placement Sequencing For Multi Head Placement Machine, *InTech'03*, Thailand.

- Ayvaz, B., Bolat, B., Aydın, N.** (2015). Stochastic reverse logistics network design for waste of electrical and electronic equipment, *Resources, Conservation and Recycling*, Sayı: 104, Sayfa: 391–404.
- Aziz, Z. A.** (2015). Ant Colony Hyper-heuristics for Travelling Salesman Problem, 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015), *Procedia Computer Science*, Sayı: 76, Sayfa: 534 – 538.
- Bai, R., Kendall, G.** (2003). An investigation of automated planograms using a simulated annealing based hyper-heuristic, *In: Proceedings of the 5th Metaheuristics, International Conference (MIC2003)*, Kyoto, Japan, August 23.
- Bakoğlu, R., Yılmaz, E.** (2001). Tedârik Zinciri Tasarımının Rekabet Avantajı Yaratması Açısından Değerlendirilmesi: Fast Food Sektörü Örneği, VI. *Ulusal Pazarlama Kongresi*, Erzurum, 28 Haziran-1 Temmuz 2001, Sayfa: 173-185.
- Beyaz, M., Dokeroglu, T., Cosar, A.** (2015). Robust hyper-heuristic algorithms for the offlineoriented/non-oriented 2D bin packing problems, *Applied Soft Computing*, Sayı: 36, Sayfa: 236–245.
- Bilgin, B., Özcan, E., Korkmaz, E. E.** (2006). An experimental study on hyper-heuristics and exam timetabling, *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling*.
- Bittle, S., A., Fox, M., S.** (2009). Learning and using hyper-heuristics for variable and value ordering in constraint satisfaction problems, *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*, Sayfa: 2209-2212.
- Blum, C., Roli, A.** (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, Sayı: 35, 3, Sayfa: 268-308.
- Breedam, A., V.** (2001). Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem, *Computers & Operations Research*, Sayı: 28, Sayfa: 289-315.
- Burke, E., Petrovic, S., Qu, R.** (2002). Case based heuristic selection for examination timetabling, *4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002)*, Sayfa 277–281.
- Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.** (2003). Hyperheuristics: An Emerging Direction In Modern Search Technology, *Handbook of Metaheuristics*, Sayfa: 457 – 474.
- Burke, E., Kendall, G., Soubeiga, E.** (2003). A Tabu-search Hyperheuristic for Timetabling and Rostering, *Journal of Heuristics*.
- Burke, E., Maccarthy, B. L., Petrovic, S., Qu, R.** (2003). Knowledge Discovery in Hyper-heuristic Using Case-Based Reasoning on Course Timetabling, *Selected Papers from the 4th International Conference on the Practice and Theory of Automated Timetabling*.

- Burke, E., Soubeiga, E.** (2003). Scheduling nurses using a tabu-search hyperheuristic, *Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003)*, sayfa 197–218.
- Burke, E., Dror, M., Petrovic, S., Qu, R.** (2005). Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems, Raghavan and E.A. Wasil (eds.), *The Next Wave in Computing, Optimization, and Decision Technologies*.
- Burke, E., Silva, A., Soubeiga, E.** (2005). Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, *Meta-heuristics: Progress as Real Problem Solvers*.
- Burke, E., Kendall, G., Silva, D., L., O'Brien, R., Soubeiga, E.,** (2005). An ant algorithm hyperheuristic for the project presentation scheduling problem, *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'05)*. Volume 3
- Burke, E., Mc-Collum, B., Meisels, A., Petrovic, S., Qu, R.** (2007). A graph-based hyper-heuristic for educational timetabling problems, *European Journal of Operational Research*, volume 176, sayfa: 177, 192.
- Burke, E., Eckersley, A.J., McCollum, B., Petrovic, S., Qu, R.** (2010). Hybrid variable neighbourhood hyperheuristics for exam timetabling problems, *European Journal of Operational Research*, doi: 10.1016/j.ejor.2010.01.044.
- Camacho, E. L., Marin, H. T., Ross, P., Ochoa, G.** (2014). A unified hyper-heuristic framework for solving bin packing problems, *Expert Systems with Applications*, Sayı: 41, Sayfa: 6876–6889.
- Cardoso, S. R., Pova, F. D. B., & Relvas, S.** (2013). Design and planning of supply chains with integration of reverse logistics activities under demand uncertainty. *European Journal of Operational Research*, Sayı: 226, Sayfa: 436-451.
- Chakhlevitch, K., Cowling, P.** (2005). Choosing the Fittest Subset of Low Level Heuristics in a Hyperheuristic Framework, *EvoCOP 2005. LNCS*, vol. 3448, sayfa 23–33. Springer, Heidelberg.
- Chakhlevitch, K.** (2006). A hyperheuristic methodology for real-world scheduling, *Doktora Tezi, Department of Computing, University of Bradford*.
- Chakhlevitch, K., Cowling, P.** (2008). Hyperheuristics: Recent Developments, *Adaptive and Multilevel Metaheuristics*, SCI 136, Sayfa: 3–29.
- Chiou, S., W.** (2009). A bi-level programming for logistics network design with system-optimized flows, *Information Sciences*, Volume 179, Issue 14, 27 Haziran 2009, Sayfa: 2434-2441.
- Confessore, G., Galiano, G., Liotta, G., Stecca, G.** (2013). A production and logistics network model with multimodal and sustainability considerations, *8th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, Sayfa: 342 – 347.

- Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F.** (2002). A Guide To Vehicle Routing Heuristics, *Journal of the Operational Research Society*, Sayı: 53, Sayfa: 512-522.
- Cordeau, J. F., Pasin, F., Solomon, M. M.** (2006). An integrated model for logistics network design, *Annals of Operations Research*, Volume 144, Number 1 / Nisan, 2006, Sayfa: 59-82.
- Cordeau, J. F., Laporte, G., Pasin, F.** (2008). An iterated local search heuristic for the logistics network design problem with single assignment, *International Journal of Production Economics*, Volume 113, Issue 2, Sayfa: 626-640.
- Cortinhal, M.J., Lopes, M.J., Melo, M.T.** (2015). Dynamic design and re-design of multi-echelon, multi-product logistics networks with outsourcing opportunities: A computational study, *Computers & Industrial Engineering*, Sayı: 90, Sayfa: 118–131.
- Cowling, P., Kendall, G., Soubeiga, E.** (2000). A Hyperheuristic Approach to Scheduling a Sales Summit, *PATAT 2000. LNCS*, vol. 2079, pp. 176–190. Springer, Heidelberg.
- Cowling, P., Kendall, G., Soubeiga, E.** (2001). A parameter-free hyperheuristic for scheduling a sales summit, *Proceedings of the 4th Metaheuristic International Conference*, MIC 2001.
- Cowling, P., Kendall, G., Soubeiga, E.** (2002). Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation, *EvoWorkShops. Lecture Notes in Computer Science*.
- Cowling, P., Kendall, G., Han, L.** (2002). An investigation of a hyperheuristic genetic algorithm applied to a trainer scheduling problem, *Proceedings of the Congress on Evolutionary Computation*.
- Cowling, P., Kendall, G., Soubeiga, E.** (2002). Hyperheuristics: A robust optimisation method applied to nurse scheduling, *PPSN 2002, LNCS*, vol. 2439, Sayfa: 851–860, Springer, Heidelberg.
- Cowling, P., Chakhlevitch, K.** (2003). Hyperheuristics for managing a large collection of low level heuristics to schedule personnel, *Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC 2003)*, sayfa 1214–1221.
- Cowling, P., Chakhlevitch, K.** (2007). Using a large set of low level heuristics in a hyperheuristic approach to personnel scheduling, *Evolutionary Scheduling*. Springer, Heidelberg.
- Crowston, W. B., Glover, F., Thompson, G. L.** (1963). Trawick, J. D., Probabilistic and parametric learning combinations of local job shop scheduling rules, *ONR Research memorandum, GSIA, Carnegie Mellon University, Pittsburgh*, 117.
- Cuesta, C., Garrido, L., Terashimar, M.** (2005). Building Hyper-heuristics Through Ant Colony Optimization for the 2D Bin Packing Problem, *LNCS 3684*, Sayfa 654–660.

- Dasci, A., Verter, V.** (2001). A continuous model for production–distribution system design, *European Journal of Operational Research*, Sayı: 129, Sayfa: 278–98.
- Deb, K., Pratap, A., Agarwal, S., Mearivan, T.** (2002). A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Sayı: 6 -2, Sayfa: 182 – 197.
- Demirel, Ö. N., Gökçen, H.** (2008). A mixed-integer programming model for remanufacturing in reverse logistics environment, *International Journal of Advanced Manufacturing Technology*, Sayı: 39(11–12), Sayfa: 1197–206.
- Dokeroglu, T., Cosar, A.** (2016). A novel multi start hyper-heuristic algorithm on the grid for the quadratic assignment problem, *Engineering Applications of Artificial Intelligence*, Sayı: 52, Sayfa: 10–25.
- Dong, B., Jiao, L., Wu, J.** (2015). A two-phase knowledge based hyper-heuristic scheduling algorithm in cellular system, *Knowledge-Based Systems*, Sayı: 88, Sayfa: 244–252.
- Dorigo, M., Caro, G. D., Gambardella, L., M.** (1999). Ant Algorithms for Discrete Optimization, *Artificial Life*, Sayı: 5, Sayfa: 137-172.
- Dowsland, K., Soubeiga, E., Burke, E.** (2004). Solving a shipper rationalisation problem with a simulated annealing based hyperheuristic, *Technical Report NOTTCSTR- 2004-1*, School of Computer Science and Information Technology, University of Nottingham.
- Dowsland, K., Soubeiga, E., Burke, E.** (2007). A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation, *European Journal of Operational Research* 179, Sayfa 759–774.
- Du, F., Evans, G. W.** (2008). A bi-objective reverse logistics network analysis for post-sale service, *Computers & Operations Research*, Sayı: 35, Sayfa: 2617–34.
- Du, L., Wu, J., Hu, F.** (2009). Logistics Network Design and Optimization of Closed-Loop Supply Chain Based on Mixed Integer Nonlinear Programming Model, *2009 ISECS International Colloquium on Computing, Communication, Control and Management*.
- Dufour, E., Laporte, G., Paquette, J., Rancourt, M.E.** (2017). Logistics service network design for humanitarian response in East Africa, *Omega*, Article in Press
- Dorndorf, U., Pesch, E.** (1995). Evolution based learning in a job shop scheduling environment, *Computers and Operations Research*, 22, 25–40.
- Elhag, A., Özcan, E.** (2015). A grouping hyper-heuristic framework: Application on graph colouring, *Expert Systems with Applications*, Sayı: 42, Sayfa: 5491–5507.
- Eren, H.** (2002). Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma(GA) İle Çözüm Performansının Artırılmasında Deney Tasarımı Uygulaması, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Yüksek Lisans Tezi*, İstanbul.

- Erol, V.** (2006). Araç Rotalama Problemleri için Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Yüksek Lisans Tezi*, İstanbul.
- Fang, H.L., Ross, P., Corne, D.** (1994). A promising hybrid GA/heuristic approach for open-shop scheduling problems, In: *Cohn, A. (ed.) Proceedings of ECAI 1994: 11th European Conference on Artificial Intelligence*, pp. 590–594. John Wiley, Chichester.
- Finger, M., Künneke, R. W.** (2011). International Handbook of Network Industries: The liberalisation of infrastructure, *Edward Elgar Publishing Limited*, Cheltenham, İngiltere.
- Fisher, H., Thompson, G. L.** (1961). Probabilistic learning combinations of local job-shop scheduling rules, In *Factory Scheduling Conference*, Carnegie Institute of Technology.
- Fisher, H., Thompson, G. L.** (1963). Probabilistic learning combinations of local job-shop scheduling rules, *Industrial Scheduling*, pages 225–251, New Jersey, Prentice-Hall, Inc.
- Fleischmann, M., Beullens, P., Bloemhof-ruwaard J. M., Wassenhove, L.** (2001). The impact of product recovery on logistics network design, *Production and Operations Management*, Sayı: 10, Sayfa: 156–73.
- Fong, C., L.** (2005). New Models in Logistics Network Design and Implications for 3PL Companies. (Doktora Tezi). Nanyang Teknik Üniversitesi, Innovation in Manufacturing and Systems Technology.
- Garrido, P., Castro, C.** (2009). Stable solving of CVRPs using hyperheuristics, *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation Archive*, sayfa 255 -162, Montreal, Canada.
- Gaw, A., Rattadilok, P., Kwan, R. S. K.** (2004). Distributed choice function hyper-heuristics for timetabling and scheduling, *Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling*, Lecture Notes in Computer Science Series
- Gen, M., Altıparmak, F., Lin, L.** (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding, *OR Spectrum*, Sayı: 28, Sayfa: 337–354.
- Guizzo, G., Vergilio, S. R., Pozo, A. T. R., Fritsche, G. M.** (2017). A multi-objective and evolutionary hyper-heuristic applied to the Integration and Test Order Problem, *Applied Soft Computing*, Sayı: 56, Sayfa: 331–344.
- Guo, J., Wang, X., Fan, S., Gen, M.** (2017). Forward and reverse logistics network and route planning under the environment of low-carbon emissions: A case study of Shanghai fresh food E-commerce enterprises, *Computers & Industrial Engineering*, Sayı: 106, Sayfa: 351–360.

- Han, L., Kendall, G., Cowling, P.** (2002). An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem, *Proceedings of the fourth Asia-Pacific Conference on Simulated Evolution And Learning*, (SEAL'02), Singapore.
- Han, L.** (2003). Guided Operators for a Hyper-Heuristic Genetic Algorithm, In *proceedings of AI-2003: Advances in Artificial Intelligence. The 16th Australian Conference on Artificial Intelligence (AI'03)* (eds Tams D Gedeon and Lance Chun Che Fung).
- Han, L., Kendall, G.** (2003). Investigation of a Tabu Assisted Hyper-Heuristic Genetic Algorithm, In *proceedings of Congress on Evolutionary Computation (CEC2003)*.
- Hansen, P., Mladenovic, N.** (2001). Variable neighbourhood search: Principles and applications, *European Journal of Operational Research* 130, Sayfa 449–467.
- Hart, E., Ross, P.** (1998). A heuristic combination method for solving job-shop scheduling problems. In: *Eiben, A.E., Back, T., Schoenauer, M., Schwefel, H.P. (eds.) PPSN 1998. LNCS*, vol. 1498, sayfa 845–854, Springer, Heidelberg.
- Hart, E., Ross, P., Nelson, J.** (1999). Scheduling chicken catching – An investigation into the success of a genetic algorithm on a real-world scheduling problem, *Annals of Operations Research*, Sayı: 92, Sayfa: 363–380.
- Hatefi, S.M., Jolai, F.** (2014). Robust and reliable forward–reverse logistics network design under demand uncertainty and facility disruptions, *Applied Mathematical Modelling*, Sayı: 38, Sayfa: 2630–2647.
- Hertz, A., Widmer, M.** (2004). Guidelines For The Use Of Meta-Heuristics In Combinatorial Optimization, *European Journal of Operation Research*, 151, 247-252.
- Hui, Q. X., Zhen, Y. H., Hu, Y., C. H.** (2007). An Optimization Design for a Half-closed-loop Logistics Network with Uncertainty, *Wireless Communications, Networking and Mobile Computing*, 2007. WiCom 2007. International Conference.
- Jayaraman, V., Guige V. D. R., Srivastava, R.** (1999). A closed-loop logistics model for remanufacturing, *Journal of the Operational Research Society*, Sayı: 50, Sayfa: 497–508.
- Jayaraman, V., Pirkul, H.** (2001). Planning and coordination of production and distribution facilities for multiple commodities, *European Journal of Operational Research*, Sayı: 133, Sayfa: 394–408.
- Jayaraman, V., Patterson, R. A., Rolland, E.** (2003). The design of reverse distribution networks: models and solution procedures, *European Journal of Operational Research*, Sayı: 150, Sayfa: 128–49.
- Jayaraman, V., Ross, A.** (2003). A simulated annealing methodology to distribution network design and management, *European Journal of Operational Research*, 144:629–45.

- Jin, Q., Xin, M., L.** (2009). Combined Simulated Annealing Algorithm for Logistics Network Design Problem, *Intelligent Systems and Applications*, ISA 2009. International Workshop on (978-1-4244-3893-8).
- Johnson, M. E., Pyke, D. F.** (2001). Supply chain management: Integration and globalization in the age of e-business, *The Amos Tuck School of Business*, Collage Research Paper, Dartmouth.
- Kadadevaramath, R. S., Chen, J. C. H., Shankar, B. L., & Rameshkumar, K.** (2012). Application of particle swarm intelligence algorithms in supply chain network architecture optimization. *Expert Systems with Applications*, 39(11), 10160-10176.
- Kaelbling, L.P., Littman, M.L., Moore, A.W.** (1996). Reinforcement learning: a survey, *Journal of Artificial Intelligence Research*, Vol 4, Sayfa 237–285.
- Kalaitzidou, M.A., Longinidis, P., Georgiadis, M.C.,** (2015). Optimal design of closed-loop supply chain networks with multifunctional nodes, *Computers and Chemical Engineering*, Sayı: 80, Sayfa: 73–91.
- Kampouridis, M., Alsheddy, A. & Tsang, E.** (2012). On the investigation of hyper-heuristics on a financial forecasting problem, *Annals of Mathematics and Artificial Intelligence*, Sayfa: 1012–2443.
- Karaboğa, D.** (2004). Yapay Zeka Optimizasyon Algoritmaları, *Atlas Yayın Dağıtım*, Çağaloğlu, İstanbul.
- Keizer, M., Haijema, M., Bloemhof, J.M., Vorst, J.** (2015). Hybrid optimization and simulation to design a logistics network for distributing perishable products, *Computers & Industrial Engineering*, Sayı: 88, Sayfa: 26–38.
- Keizer, M., Akkerman, R., Grunow, M., Bloemhof, J.M., Haijema, R., Vorst, J.** (2017). Logistics network design for perishable products with heterogeneous quality decay, *European Journal of Operational Research*, Sayı: 262, Sayfa: 535–549.
- Kendall, G., Soubeiga, E., Cowling, P.** (2002). Choice function and Random Hyperheuristics, *Proceedings of the fourth Asia-Pacific Conference on Simulated Evolution And Learning*, SEAL.
- Kendall, G., Mohd Hussin, N.** (2003). An investigation of a tabu search based hyperheuristic for examination timetabling. *Selected papers from the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003)*, sayfa 309–328.
- Kendall, G., Mohamad, M.** (2004). Channel Assignment in Cellular Communication Using a Great Deluge Hyper-heuristic, *in the Proceedings of the 2004 IEEE International Conference on Network (ICON2004)*.
- Kendall, G., Mohd Hussin, N.** (2004). Tabu search hyper-heuristic approach to the examination timetabling problem at University of Technology MARA, *PATAT 2004. LNCS*, vol. 3616, sayfa 199–217, Springer, Heidelberg.

- Kendall, G., Li, J.** (2013). Competitive travelling salesmen problem: A hyper-heuristic approach, *Journal of the Operational Research Society*, 64(2), 208-216.
- Kengpol, A.** (2008). Design of a decision support system to evaluate logistics distribution network in Greater Mekong Subregion Countries, *International Journal of Production Economics*, Sayı: 115, Bölüm: 2, Sayfa: 388-399.
- Keyvanshokoh, E., Fattahi M., Hosseini S., Moghaddam, R.** (2013). A dynamic pricing approach for returned products in integrated forward/reverse logistics network design, *Applied Mathematical Modelling*, Sayı: 37, Sayfa: 10182–10202.
- Kheiri, A., Keedwell, E., Gibson, M.J., Savic, D.** (2015). Sequence Analysis-based Hyper-heuristics for Water Distribution Network Optimisation, *13th Computer Control for Water Industry Conference, CCWI 2015*, Procedia Engineering, Sayı: 119, Sayfa: 1269 – 1277.
- Kheiri, A., Özcan, E.** (2016). An iterated multi-stage selection hyper-heuristic, *European Journal of Operational Research*, Sayı: 250, Sayfa: 77–90.
- Ko, H. J., Evans, G. W.** (2007). A genetic-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs, *Computers & Operations Research*, Sayı: 34, Sayfa: 346–66.
- Koulinas, G. K., Anagnostopoulos, K. P.** (2013). A new tabu search-based hyper-heuristic algorithm for solving construction leveling problems with limited resource availabilities, *Automation in Construction*, Sayı: 31, Sayfa: 169–175.
- Koulinas, G. K., Kotsikas, L., Anagnostopoulos, K. P.** (2014). A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, *Information Sciences*, Sayı: 277, Sayfa: 680–693.
- Krikke, H. R., Harten, A. V., Schuur P. C.** (1999). Reverse logistic network re-design for copiers, *OR Spektrum 1999*, Springer-Verlag, Sayı: 21, Sayfa: 381–409.
- Kumar, R., Joshi, A., H., Banka, K., K., Rockett, P., I.** (2008). Evolution of hyperheuristics for the biobjective 0/1 knapsack problem by multiobjective genetic programming, *10th annual conference on Genetic and evolutionary computation*, Sayfa: 1227-1234.
- Kumari, A. C., Srinivas, K.** (2016). Hyper-heuristic approach for multi-objective software module clustering, *The Journal of Systems and Software*, Sayı: 117, Sayfa: 384–401.
- Lagoudakis, M.G., Littman, M.L.** (2000). Algorithm selection using reinforcement learning, *In: Proceedings of the 17th International Conference on Machine Learning*, Sayfa: 511–518.
- Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F.** (2000). Classical And Modern Heuristics For The Vehicle Routing Problem, *International Transactions in Operation Research*, 7, 285-300.

- Lee, D. H., Dong, M.** (2008). A heuristic approach to logistics network design for end-of-lease computer products recovery, *Transportation Research Part E*, Sayı: 44, Sayfa: 455–474.
- Lee, D. H., Dong, M.** (2009). Dynamic network design for reverse logistics operations under uncertainty, *Transportation Research, Part E: Logistics and Transportation Review*, Sayı: 45, Bölüm: 1, Sayfa 61-71.
- Lee, E. J., Gen, M., Rhee, K.** (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm, *Computers & Industrial Engineering*, Sayı: 56, Bölüm: 3, Nisan 2009, Sayfa: 951-964.
- Leon, C., Miranda, G., Segura, C.** (2009). A memetic algorithm and a parallel hyperheuristic island-based model for a 2D packing problem, *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, Sayfa: 1371-1378.
- Leon, C., Miranda, G., Segura, C.** (2009). Parallel Hyperheuristic: A Self-Adaptive Island-Based Model for Multi-Objective Optimization, *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*, Sayfa: 2209-2212.
- Lieckens, K., Vandaele, N.** (2007). Reverse logistics network design with stochastic lead times, *Computers & Operations Research*, Volume 34, Issue 2, February 2007, Sayfa 395-416.
- Li, X.** (2013). An integrated modeling framework for design of logistics networks with expedited shipment services, *Transportation Research*, Bölüm: E, Sayı: 56, Sayfa: 46–63.
- Li, W., Özcan, E., John, R.** (2017). Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimisation, *Renewable Energy*, Sayı: 105, Sayfa: 473-482.
- Lin, J., Wang, Z. J., Li, X.** (2017). A backtracking search hyper-heuristic for the distributed assembly flow-shop scheduling problem, *Swarm and Evolutionary Computation*, Article In Press.
- Lin, L., Gen, M. ve Wang, X.** (2009). Integrated multistage logistics network design by using hybrid evolutionary algorithm, *Computers & Industrial Engineering*, Sayı: 56, Sayfa: 854 – 873.
- Liu, Q.** (2008). Study on multi-object optimization of logistics network based on genetic algorithm, *International Conference on Computational Intelligence and Security*.
- Listes, O., Dekker, R.** (2005). A stochastic approach to a case study for product recovery network design, *European Journal of Operational Research*, Sayı: 160, Sayfa: 268–87.
- Lu, Z., Bostel, N.** (2007). A facility location model for logistics systems including reverse flows: the case of remanufacturing activities, *Computers & Operations Research*, Sayı: 34, Sayfa: 299–323.

- Lu, Y., Li, X., Liang, L.** (2008). Multi-objective Optimization of Reverse Logistics Network Based on Improved Particle Swarm Optimization, *Proceedings of the 7th World Congress on Intelligent Control and Automation 2008*.
- Lu, T. P., Trappey, A. J. C., Chen, Y. K., Chang, Y. D.** (2013). Collaborative design and analysis of supply chain network management key processes model, *Journal of Network and Computer Applications*, Sayı: 36 - 6, Sayfa: 1503–1511.
- Maashi, M., Kendall, G., Özcan, E.** (2015). Choice function based hyper-heuristics for multi-objective optimization, *Applied Soft Computing* 28, (2015) 312–326.
- Marin, A., Pelegrin, B.** (1998). The return plant location problem: modelling and resolution, *European Journal of Operational Research*, Sayı:104(2), Sayfa: 375–392.
- Martinez, L. A. M., ve Zhang, D. Z.** (2011). Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design, *International Journal of Production Economics*, 131(1), 407-420.
- McClymont, K., Keedwell, E., Savić, D., ve Randall-Smith, M.** (2013). A general multi-objective hyper-heuristic for water distribution network design with discolouration risk, *Journal of Hydroinformatics*, Sayı: 15 Bölüm: 3, Sayfa: 700-716.
- Meepetchdee, Y., Shah, N.** (2007). Logistical network design with robustness and complexity considerations, *International Journal of Physical Distribution & Logistics Management*, Sayı 37, Sayfa 201–22.
- Melachrinoudis, E., Messac, A., Min, H.** (2005). Consolidating a warehouse network: a physical programming approach, *International Journal of Production Economics*, Sayı 97, Sayfa 1–17.
- Meng, F., Wen, L.** (2008). Optimization of Logistics Distribution Network Based on Improved Particle Swarm Optimization, *Fourth International Conference on Natural Computation*.
- Min., H., Ko, C. S., Ko, H. J.** (2006). The spatial and temporal consolidation of returned products in a closed-loop supply chain network, *Computers & Industrial Engineering*, Sayı: 51, Sayfa: 309–20.
- Min, H., Ko, H. J.** (2008). The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers, *International Journal of Production Economics*, Sayı: 113, Sayfa: 176–92.
- Min, H., Zhou, G.** (2002). Supply Chain Modeling: Past, Present and Future, *Computers & Industrial Engineering*, Sayı: 43, Sayfa: 231-249.
- Miranda, P. A., Garrido, R. A.** (2004). Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand, *Transportation Research, Part E*, Sayı:40, Sayfa: 183–207.
- Miranda, P. A., Garrido, R. A., Ceroni, J.A.** (2009). e-Work Based Collaborative optimization approach for strategic logistic network design problem, *Computers & Industrial Engineering*, Sayı 57, Issue 1, Ağustos 2009, Sayfa 3-13.

- Mısır, M., Verbeeck, K., Causmaecker, P. D., Berghe, G. V.** (2013). An investigation on the generality level of selection hyper-heuristics under different empirical conditions, *Applied Soft Computing*, Sayı: 13, Sayfa: 3335–3353.
- Mockus, J., Mockus, L.** (1991). Bayesian approach to global optimization and applications to multi-objective constrained problems, *Journal of Optimization Theory and Applications*, 70(1):155–170.
- Mockus, J., Eddy, W., Mockus, A., Mockus, L., Reklaitis, G.** (1997). Bayesian Heuristic Approach to Discrete and Global Optimization. *Kluwer Academic Publishers*.
- Moreno, J. G., Sanz, S. S., Moreno, B. S., Calvo, L. C., Figueras, A. P.** (2013). An evolutionary-based hyper-heuristic approach for optimal construction of group method of data handling networks, *Information Sciences*, Sayı: 247, Sayfa: 94–108.
- Nabiyev, V. V.** (2003). Yapay Zeka – Problemler Yöntemler Algoritmalar, *Seçkin Yayınevi*, Ankara.
- Nareyek, A.** (2003). Choosing search heuristics by non-stationary reinforcement learning, *Metaheuristics: Computer decision-making*, Sayfa 523–544, *Kluwer Academic Publishers*, Dordrecht.
- Niknejad, A., Petrovic, D.** (2014). Optimisation of integrated reverse logistics networks with different product recovery routes, *European Journal of Operational Research*, Sayı: 238, Sayfa: 143–154.
- Norenkov, I.** (1994). Scheduling and allocation for simulation and synthesis of CAD system hardware. In: *Proceedings of EWITD 1994, East-West International Conference*, Moscow, ICSTI, sayfa 20–24.
- Norenkov, I., Goodman, E.** (1997). Solving scheduling problems via evolutionary methods for rule sequence optimisation, *Second World Conference on Soft Computing (WSC2)*.
- Nourbakhsh, S.M., Bai, Y., Maia, G. Ouyang, Y., Rodriguez, L.** (2016). Grain supply chain network design and logistics planning for reducing post-harvest loss, *Biosystems Engineering*, Sayı: 151, Sayfa: 105-115.
- Özcan, E., Bilgin, B., Korkmaz, E.** (2008). A comprehensive analysis of hyper-heuristics, *Intelligent Data Analysis*, Vol. 12 Issue 1, sayfa 3-23.
- Peng, Z. Y., Zhong, D. Y.** (2008). Optimization Model for Integrated Logistics Network Design in Green Manufacturing System, *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*.
- Petrovic, S., Qu, R.** (2002). Case-Based Reasoning as a Heuristic Selector in a Hyper-Heuristic for Course Timetabling Problems, *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information Engineering Systems and Applied Technologies (KES'02)*.
- Phan, T., P., Ho, T., B.** (2007). A hyper-heuristic for descriptive rule induction, *International Journal of Data Warehousing and Mining*.

- Pishvaei, M. S., Farahani, R., Z., Dullaert, W.** (2009). A memetic algorithm for bi-objective integrated forward/reverse logistics network design, *Computers & Operations Research*, Article In Press.
- Poyato, R. M., Sanz, S. S., Fernandez, S. J., Villar, P. D.** (2015), Optimal discharge scheduling of energy storage systems in MicroGrids based on hyper-heuristics, *Renewable Energy*, Sayı: 83, Sayfa: 13-24.
- Preux, P., Talbi, E. G.** (1999). Towards Hybrid Evolutionary Algorithms, *International Transactions in Operation Research*, 6, 557-570.
- Qin, X. W., Qin, F.** (2008). Hybrid Particle Swarm Optimization Algorithm for the Logistics Network Design Problem under Concave Cost Function, *2008 International Conference on Computer Science and Software Engineering*.
- Qin, Z., Ji, X.** (2010). Logistic Network Design for Product Recovery in Fuzzy Environment, *European Journal Operational Research*, Volume 202, Sayı 2, 16 April 2010, Sayfa: 479-490.
- Qu, R., Burke, E.** (2005). Hybrid variable neighbourhood hyperheuristics for exam timetabling problems, *Proceedings of the 6th Metaheuristics International Conference (MIC 2005)*.
- Raad, D., Sinske, A., van Vuuren, J.** (2010). Multiobjective optimization for water distribution system design using a hyperheuristic, *Journal of Water Resources Planning and Management*, Sayı: 136, Bölüm: 5, Sayfa: 592-596.
- Rafique, A., F., He, L., Kamran, A., Zeeshan, Q.** (2011). Hyper Heuristic Approach for Design and Optimization of Satellite Launch Vehicle, *Chinese Journal of Aeronautics*, 24(2), 150-163.
- Rajni, I. C.** (2013). Bacterial foraging based hyper-heuristic for resource scheduling in grid computing, *Future Generation Computer Systems*, Sayı: 29, Sayfa: 751–762.
- Remde, S., Dahal, K., Cowling, P., Colledge, N.** (2009). Binary Exponential Back Off for Tabu Tenure in Hyperheuristics, *Evolutionary Computation in Combinatorial Optimization*, Springer Berlin, Sayfa 109-120.
- Roghanian, E., Pazzoheshfar, P.** (2014). An optimization model for reverse logistics network under stochastic environment by using genetic algorithm, *Journal of Manufacturing Systems*, Sayı: 33, Sayfa: 348–356.
- Ross, P., Schulenburg, S., Hart, E., Marin-Blazquez, J. G.** (2002). Hyper-heuristics: learning to combine simple heuristics in bin-packing problem, *In Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO'02, Sayfa 942–948.
- Ross, P., Marin-Blazquez, J. G., Schulenburg, S., Hart, E.** (2003). Learning a procedure that can solve hard bin-packing problems: a new GA-based approach to Hyperheuristics, *GECCO 2003. LNCS*, vol. 2723, sayfa, 1295–1306. Springer, Heidelberg.

- Sabar, N. R., Kendall, G.** (2015). Population based Monte Carlo tree search hyper-heuristic for combinatorial optimization problems, *Information Sciences*, Sayı: 314, Sayfa: 225–239.
- Sabri, E. H., Beamon, B. M.** (2000). A multi-objective approach to simultaneous strategic and operational planning in supply chain design, *Omega*, 2000, Sayı:28, Sayfa: 581–98.
- Salema, M. I., Povia, A. P. B., Novais, A. Q.** (2006). A warehouse-based design model for reverse logistics, *Journal of the Operational Research Society*, Sayı: 57(6), Sayfa: 615–629.
- Salema, M. I. G., Povia, A. P. B., Novais, A. Q.** (2007). An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty, *European Journal of Operational Research*, Volume 179, Issue 3, 16 June 2007, Sayfa 1063-1077.
- Sayed, M. E., Afia, N., Kharbotly, A., E.** (2009). A stochastic model for forward–reverse logistics network design under risk, *Computers & Industrial Engineering*, Article in Press.
- Shanshan, W., Kejing, Z.** (2008). Optimization Model of E-waste Reverse Logistics and Recycling Network, *Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering*.
- Shimizu, Y., Matsuda, S., Wada, T.** (2006). A flexible design of logistic network against uncertain demands through hybrid meta-heuristic method, *Computer Aided Chemical Engineering*, Volume 21, Part 2, 2006, Sayfa 2051 – 2056.
- Soleimani, H., & Govindan, K.** (2014). Reverse logistics network design and planning utilizing conditional value at risk, *European Journal of Operational Research*, 237(2), 487-497.
- Soubeiga, E.** (2003). *Development and Application of Hyperheuristics to Personnel Scheduling*. (Doktora Tezi), University of Nottingham, School of Computer Science and Information Technology.
- Soysal, M., Ruwaard, J., Vorst, J.** (2014). Modelling food logistics networks with emission considerations: The case of an international beef supply chain, *Int. J. Production Economics*, Sayı: 152, Sayfa: 57–70.
- Storer, R.H., Wu, S.D., Vaccari, R.** (1995). Problem and heuristic search space strategies for job shop scheduling, *ORSA Journal on Computing*, 7, Sayfa: 453–467.
- Sung C. S., Song, S. H.** (2003). Integrated service network design for a cross-docking supply chain network, *Journal of the Operational Research Society*, Sayı: 54, Sayfa: 1283–95.
- Syarif, A., Yun, Y. S., Gen, M.** (2002). Study on multi-stage logistics chain network: a spanning tree-based genetic algorithm approach, *Computers & Industrial Engineering*, Sayı: 43, Sayfa: 299–314.
- Tarantilis, C. D., Ioannou, G., Prastacos, G.** (2004). Advanced Vehicle Routing Algorithms For Complex Operations Management Problems, *Journal of Food Engineering*, Elsevier Ltd.

- Terashima-Marin, H., Ross, P., Valenzuela-Rendon, M.** (1999). Evolution of constraint satisfaction strategies in examination timetabling. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, sayfa. 635–642, Morgan Kaufmann, San Francisco.
- Thomas, J., Chaudhari, N. S.** (2014). Design of efficient packing system using genetic algorithm based on hyper heuristic approach, *Advances in Engineering Software*, Sayı: 73, Sayfa: 45–52.
- Topcuoglu, H. R., Ucar, A., Altin, L.** (2014). A hyper-heuristic based framework for dynamic optimization problems, *Applied Soft Computing*, Sayı: 19, Sayfa: 236–251.
- Tsiakis, P., Papageorgiou, L. G.** (2008). Optimal production allocation and distribution supply chain networks, *International Journal of Production Economics*, Sayı: 111, Sayfa: 468–83.
- Üster, H., Easwaran, G., Akçali, E., Çetinkaya, S.** (2007). Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model, *Naval Research Logistics*, Sayı: 54, Sayfa: 890–907.
- Winston, W. L., Goldberg, J. B.** (2004). *Operations Research: Applications and Algorithms*, Thomson Brooks/Cole, The University of California, Edition 4.
- Yayhaoui, H., Krichen, S., Derbel, B., Talbi, E. G.** (2015). A hybrid ILS-VND based hyper-heuristic for permutation flowshop scheduling problem, 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, *Procedia Computer Science*, Sayı: 60, Sayfa: 632 – 641.
- Yang, J., Guo, J., Ma, S.** (2016), Low-carbon city logistics distribution network design with resource deployment, *Journal of Cleaner Production*, Sayı: 119, Sayfa: 223-228.
- Yeh, W. C.** (2005). A hybrid heuristic algorithm for the multistage supply chain network problem, *International Journal of Advanced Manufacturing and Technology*, Sayı: 26, Sayfa: 675–85.
- Yi, G., Dong, W.** (2009). Optimum Design for Full Vehicle Logistic Network Based on Mixed Particle Swarm Algorithm, *Computer Science and Information Engineering, 2009 WRI World Congress*, (978-0-7695-3507-4).
- Yi, P., Huang, M., Guo, L., Shi, T.** (2016). A retailer oriented closed-loop supply chain network design for end of life construction machinery remanufacturing, *Journal of Cleaner Production*, Sayı: 124, Sayfa: 191-203.
- Zamli, K. Z., Alkazemi, B. Y., Kendall, G.** (2016). A Tabu Search hyper-heuristic strategy for t-way test suite generation, *Applied Soft Computing*, 44 (2016) 57–74.

- Zapfel, G., Wasner, M.** (2002). Planning and optimization of hub-and-spoke transportation networks of cooperative third-party logistics providers, *International Journal of Production Economics*, Volume 78, Issue 2, 21 Temmuz 2002, Sayfa 207-220.
- Zhang, D., Zhan, Q., Chen, Y., Li, S.** (2016). Joint optimization of logistics infrastructure investments and subsidies in a regional logistics network with CO2 emission reduction targets, *Transportation Research*, Bölüm: D, Article in Press
- Zhang, Q., Sun, J., Xiao, G., Tsang, E.** (2007). Evolutionary algorithms refining a heuristic: Hyper-heuristic for shared-path protections in WDM networks under SRLG constraints, *IEEE Transactions on Systems, Man and Cybernetics*.
- Zhang, W., Xu, D.** (2014). Integrating the logistics network design with order quantity determination under uncertain customer demands, *Expert Systems with Applications*, Sayı: 41, Sayfa: 168–175.
- Zhao, Y.** (2009). Optimization Design for Multi-echelon Logistics Network in Remanufacturing System, *2009 Second International Conference on Intelligent Computation Technology and Automation*.
- Zhou, G., Cao, Z., Meng, Z., Xu, Q.** (2008). A GA-based Approach on A Repair Logistics Network Design with M/M/s model, *International Conference on Computational Intelligence and Security*.
- Zohal, M., Soleiman, H.** (2016). Developing an ant colony approach for green closed-loop supply chain network design: a case study in gold industry, *Journal of Cleaner Production*, Sayı: 133, Sayfa: 314-337.

EKLER

EK A : $2 \times 2 \times 2 \times 2_1$ Problemi GAMS Modeli

EK B : $2 \times 2 \times 2 \times 2_2$ Problemi GAMS Modeli

EK C : $2 \times 2 \times 2 \times 2_3$ Problemi GAMS Modeli

EK D : $6 \times 1 \times 4 \times 6_1$ Problemi GAMS Modeli

EK E : $6 \times 1 \times 4 \times 6_2$ Problemi GAMS Modeli

EK F : $6 \times 1 \times 4 \times 6_3$ Problemi GAMS Modeli

EK G : $10 \times 2 \times 5 \times 10_1$ Problemi GAMS Modeli

EK H : $10 \times 2 \times 5 \times 10_2$ Problemi GAMS Modeli

EK I : $10 \times 2 \times 5 \times 10_3$ Problemi GAMS Modeli

EK J : $12 \times 2 \times 8 \times 12_1$ Problemi GAMS Modeli

EK K : $12 \times 2 \times 8 \times 12_2$ Problemi GAMS Modeli

EK L : $12 \times 2 \times 8 \times 12_3$ Problemi GAMS Modeli

EK M : $2 \times 2 \times 2 \times 2_1$ Problemi İçin Bulunan Sonuçlar

EK N : $2 \times 2 \times 2 \times 2_2$ Problemi İçin Bulunan Sonuçlar

EK O : $2 \times 2 \times 2 \times 2_3$ Problemi İçin Bulunan Sonuçlar

EK P : $6 \times 1 \times 4 \times 6_1$ Problemi İçin Bulunan Sonuçlar

EK Q : $6 \times 1 \times 4 \times 6_2$ Problemi İçin Bulunan Sonuçlar

EK R : $6 \times 1 \times 4 \times 6_3$ Problemi İçin Bulunan Sonuçlar

EK S : $10 \times 2 \times 5 \times 10_1$ Problemi İçin Bulunan Sonuçlar

EK T : $10 \times 2 \times 5 \times 10_2$ Problemi İçin Bulunan Sonuçlar

EK U : $10 \times 2 \times 5 \times 10_3$ Problemi İçin Bulunan Sonuçlar

EK V : $12 \times 2 \times 8 \times 12_1$ Problemi İçin Bulunan Sonuçlar

EK W : $12 \times 2 \times 8 \times 12_2$ Problemi İçin Bulunan Sonuçlar

EK X : $12 \times 2 \times 8 \times 12_3$ Problemi İçin Bulunan Sonuçlar

EK Y : $100 \times 2 \times 20 \times 300_1$ Problemi İçin Bulunan Sonuçlar

EK Z : $100 \times 2 \times 20 \times 300_2$ Problemi İçin Bulunan Sonuçlar

EK AA : $100 \times 2 \times 20 \times 300_3$ Problemi İçin Bulunan Sonuçlar

EK BB : $1000 \times 60 \times 50 \times 300_1$ Problemi İçin Bulunan Sonuçlar

EK CC : $1000 \times 60 \times 50 \times 300_2$ Problemi İçin Bulunan Sonuçlar

EK DD : $1000 \times 60 \times 50 \times 300_3$ Problemi İçin Bulunan Sonuçlar

EK EE : $6 \times 4 \times 4 \times 720$ Problemi İçin Bulunan Sonuçlar

EK FF : $4 \times 2 \times 2 \times 80$ Problemi İçin Bulunan Sonuçlar



EK A : 2x2x2x2₁ Problemi GAMS Modeli

SETS

S tedârikçiler / S1, S2 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2 /

I müşteriler / C1, C2 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedârikçi S'nin kapasitesi

/ S1 300

S2 400 /

D(K) fabrika K'nin kapasitesi

/ F1 500

F2 250 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 700

DC2 700 /

DM(I) müşteri I'nın talebi

/ C1 350

C2 300 /

V(J) dağıtım merkezi J'nin sabit işletme mâliyeti

/ DC1 25000

DC2 25000 /

G(K) fabrika K'nın sabit işletme mâliyeti

/ F1 150000

F2 75000 /

E(K) fabrika K'nın değişken mâliyeti

/ F1 5

F2 7 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma mâliyeti

	C1	C2
DC1	200	140
DC2	120	260 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma mâliyeti

	C1	C2
C1	999999	200
C2	200	999999;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma mâliyeti

	DC1	DC2
F1	15	30
F2	20	25 ;

TABLE

T(S,K) tedârikçi S'den fabrika K'ya birim satınalma ve taşıma mâliyeti

	F1	F2
S1	60	100
S2	80	90 ;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

$B(S,K)$ tedârikçi S'den fabrika K'ya taşınan hammadde miktarı

$F(K,J)$ fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

$Q(J,I)$ dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

$M(J)$ eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

$P(K)$ eğer fabrika K aktifse 1'dir aksi halde 0'dır

$Y(J,I)$ eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

$N(J,I)$ eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

$R(J,I)$ eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam mâliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE $M, P, Y, N, R, X, U(I)$;

EQUATIONS

$COST$ amaç fonksiyon tanımı

$FACTORY_EQUALITY(K)$ fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

$DISCENTER_CAPACITY(J)$ dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

$DEMAND(I)$ müşteri I'nın talebi karşılanmalıdır

$DISCENTER_EQUALITY(J)$ dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

$FACTORY_CAPACITY(K)$ fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedârikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$\text{COST} \dots Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O))) ;$$

$$\text{DEMAND}(I) \dots \text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I) ;$$

$$\text{DISCENTER_CAPACITY}(J) \dots \text{SUM}(I, Q(J,I)) = L = W(J)*M(J) ;$$

$$\text{DISCENTER_EQUALITY}(J) \dots \text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J) ;$$

$$\text{FACTORY_CAPACITY}(K) \dots \text{SUM}(J, F(K,J)) = L = D(K)*P(K) ;$$

$$\text{FACTORY_EQUALITY}(K) \dots \text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K);$$

$$\text{SUPPLIER_CAPACITY}(S) \dots \text{SUM}(K, B(S,K)) = L = H(S) ;$$

$$\text{ONLY_SERVICED_BY_ONE_DC}(I) \dots \text{SUM}(J, Y(J,I)) = E = 1;$$

$$\text{OPEN_DC_START_ROUTE}(J) \dots \text{SUM}(I, N(J,I)) = E = M(J);$$

$$\text{OPEN_DC_END_ROUTE}(J) \dots \text{SUM}(I, R(J,I)) = E = M(J);$$

$$\text{ONLY_ONE_START_ROUTE}(I) \dots \text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1;$$

```

ONLY_ONE_END_ROUTE(I)..          SUM(J, R(J,I)*Y(J,I))  +  SUM((J,O),
X(J,I,O)*Y(J,I)*Y(J,O)) =E= 1;

```

```

SUBTOUR_ELIMINATION(I,O).. U(I) - U(O) + CARD(I)* SUM(J, X(J,I,O))
=L= CARD(I) - 1;

```

```

MODEL LAT_MODEL /ALL/ ;

```

```

OPTION OPTCR=0;

```

```

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

```

```

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

```

EK B : 2×2×2×2₂ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2 /

I müşteriler / C1, C2 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 450

S2 300 /

D(K) fabrika K'nin kapasitesi

/ F1 300

F2 400 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 800

DC2 850 /

DM(I) müşteri I'nın talebi

/ C1 250

C2 400 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 120000

F2 90000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 8

F2 6 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 30000

DC2 33000 / ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	75	80
S2	60	100 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2
F1	20	15
F2	10	30 ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2
DC1	150	200
DC2	110	180 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2
C1	999999	140
C2	140	999999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

X(J,I,O) eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

U(I) müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$\text{COST} \dots Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O))) ;$$

$$\text{DEMAND}(I) \dots \text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I) ;$$

$$\text{DISCENTER_CAPACITY}(J) \dots \text{SUM}(I, Q(J,I)) = L = W(J)*M(J) ;$$

$$\text{DISCENTER_EQUALITY}(J) \dots \text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J) ;$$

$$\text{FACTORY_CAPACITY}(K) \dots \text{SUM}(J, F(K,J)) = L = D(K)*P(K) ;$$

$$\text{FACTORY_EQUALITY}(K) \dots \text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K) ;$$

$$\text{SUPPLIER_CAPACITY}(S) \dots \text{SUM}(K, B(S,K)) = L = H(S) ;$$

$$\text{ONLY_SERVICED_BY_ONE_DC}(I) \dots \text{SUM}(J, Y(J,I)) = E = 1 ;$$

```

OPEN_DC_START_ROUTE(J).. SUM(I, N(J,I)) =E= M(J);

OPEN_DC_END_ROUTE(J).. SUM(I, R(J,I)) =E= M(J);

ONLY_ONE_START_ROUTE(I).. SUM(J, N(J,I)*Y(J,I)) + SUM((J,O),
X(J,O,I)*Y(J,I)*Y(J,O)) =E= 1;

ONLY_ONE_END_ROUTE(I).. SUM(J, R(J,I)*Y(J,I)) + SUM((J,O),
X(J,I,O)*Y(J,I)*Y(J,O)) =E= 1;

SUBTOUR_ELIMINATION(I,O).. U(I) - U(O) + CARD(I)* SUM(J, X(J,I,O))
=L= CARD(I) - 1;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

```

EK C : 2×2×2×2₃ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2 /

I müşteriler / C1, C2 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 600

S2 500 /

D(K) fabrika K'nin kapasitesi

/ F1 350

F2 550 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 900

DC2 850 /

DM(I) müşteri I'nın talebi

/ C1 400

C2 450 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 150000

F2 110000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 6

F2 9 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 40000

DC2 35000 / ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	55	75
S2	60	65 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2
F1	30	25
F2	35	20 ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2
DC1	160	200
DC2	180	150 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2
C1	999999	250
C2	250	999999;

SCALAR UR üründeki birim hammadde kullanım oranı $/1/$;

VARIABLES

$B(S,K)$ tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

$F(K,J)$ fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

$Q(J,I)$ dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

$M(J)$ eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

$P(K)$ eğer fabrika K aktifse 1'dir aksi halde 0'dır

$Y(J,I)$ eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

$N(J,I)$ eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

$R(J,I)$ eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE $M, P, Y, N, R, X, U(I)$;

EQUATIONS

$COST$ amaç fonksiyon tanımı

$FACTORY_EQUALITY(K)$ fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

$DISCENTER_CAPACITY(J)$ dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

$DEMAND(I)$ müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$\text{COST} \dots Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O))) ;$$

$$\text{DEMAND}(I) \dots \text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I) ;$$

$$\text{DISCENTER_CAPACITY}(J) \dots \text{SUM}(I, Q(J,I)) = L = W(J)*M(J) ;$$

$$\text{DISCENTER_EQUALITY}(J) \dots \text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J) ;$$

$$\text{FACTORY_CAPACITY}(K) \dots \text{SUM}(J, F(K,J)) = L = D(K)*P(K) ;$$

$$\text{FACTORY_EQUALITY}(K) \dots \text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K) ;$$

$$\text{SUPPLIER_CAPACITY}(S) \dots \text{SUM}(K, B(S,K)) = L = H(S) ;$$

$$\text{ONLY_SERVICED_BY_ONE_DC}(I) \dots \text{SUM}(J, Y(J,I)) = E = 1 ;$$

```

OPEN_DC_START_ROUTE(J)..  SUM(I, N(J,I)) =E= M(J);

OPEN_DC_END_ROUTE(J)..   SUM(I, R(J,I)) =E= M(J);

ONLY_ONE_START_ROUTE(I)..    SUM(J,  N(J,I)*Y(J,I))  +  SUM((J,O),
X(J,O,I)*Y(J,I)*Y(J,O)) =E= 1;

ONLY_ONE_END_ROUTE(I)..      SUM(J,  R(J,I)*Y(J,I))  +  SUM((J,O),
X(J,I,O)*Y(J,I)*Y(J,O)) =E= 1;

SUBTOUR_ELIMINATION(I,O)..  U(I) - U(O) + CARD(I)* SUM(J, X(J,I,O))
=L= CARD(I) - 1;

```

```

MODEL LAT_MODEL /ALL/ ;

```

```

OPTION OPTCR=0;

```

```

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

```

```

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

```

EK D : 6×1×4×6₁ Problemi GAMS Modeli

SETS

S tedârikçiler / S1, S2, S3, S4, S5, S6 /

K fabrikalar / F1 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4 /

I müşteriler / C1, C2, C3, C4, C5, C6 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedârikçi S'nin kapasitesi

/ S1 50

S2 40

S3 30

S4 45

S5 50

S6 40 /

D(K) fabrika K'nin kapasitesi

/ F1 200 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 70

DC2 50

DC3 80

DC4 100 /

DM(I) müşteri I'nın talebi

/ C1 30

C2 25

C3 35

C4 20

C5 35

C6 40 /

V(J) dağıtım merkezi J'nin sabit işletme mâliyeti

/ DC1 2000

DC2 2000

DC3 2000

DC4 2000 /

G(K) fabrika K'nın sabit işletme mâliyeti

/ F1 10000 /

E(K) fabrika K'nın değişken mâliyeti

/ F1 20 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6
DC1	240	280	240	360	370	260
DC2	200	300	240	280	240	200
DC3	300	200	240	280	280	290
DC4	260	160	250	200	350	240 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma mâliyeti

	DC1	DC2	DC3	DC4
F1	30	50	75	80 ;

TABLE

T(S,K) tedârikçi S'den fabrika K'ya birim satınalma ve taşıma mâliyeti

F1

S1 25
S2 20
S3 30
S4 20
S5 25
S6 35 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6
C1	999999	105	60	220	200	85
C2	105	999999	100	320	285	160
C3	60	100	999999	270	185	60
C4	220	320	270	999999	220	240
C5	200	285	185	220	999999	125
C6	85	160	60	240	125	999999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedârikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam mâliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedârikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = DM(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)*Y(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)*Y(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)*\text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK E : 6×1×4×6₂ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6 /

K fabrikalar / F1 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4 /

I müşteriler / C1, C2, C3, C4, C5, C6 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 25

S2 30

S3 15

S4 20

S5 30

S6 35 /

D(K) fabrika K'nin kapasitesi

/ F1 150 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 30

DC2 40

DC3 50

DC4 40 /

DM(I) müşteri I'nın talebi

/ C1 15

C2 20

C3 25

C4 30

C5 20

C6 25 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 1600

DC2 1800

DC3 1900

DC4 1800 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 8000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 15 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6
DC1	240	200	260	280	230	240
DC2	220	210	275	200	240	265
DC3	195	245	260	220	280	205
DC4	260	290	280	220	260	250 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4
F1	20	30	45	35 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

F1

S1	15
S2	25
S3	20
S4	30
S5	20
S6	15 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6
C1	999999	100	105	180	120	150
C2	100	999999	140	145	155	200
C3	105	140	999999	200	160	140
C4	180	145	200	999999	145	125
C5	120	155	160	145	999999	165
C6	150	200	140	125	165	999999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE $M, P, Y, N, R, X, U(I)$;

EQUATIONS

$COST$ amaç fonksiyon tanımı

$FACTORY_EQUALITY(K)$ fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

$DISCENTER_CAPACITY(J)$ dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

$DEMAND(I)$ müşteri I'nın talebi karşılanmalıdır

$DISCENTER_EQUALITY(J)$ dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

$FACTORY_CAPACITY(K)$ fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

$SUPPLIER_CAPACITY(S)$ tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

$ONLY_SERVICED_BY_ONE_DC(I)$ her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

$OPEN_DC_START_ROUTE(J)$ eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

$OPEN_DC_END_ROUTE(J)$ eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

$ONLY_ONE_START_ROUTE(I)$ her müşteri I'ya bir araç ulaşmalıdır

$ONLY_ONE_END_ROUTE(I)$ her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = DM(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)*Y(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)*Y(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)*\text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK F : 6×1×4×6₃ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6 /

K fabrikalar / F1 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4 /

I müşteriler / C1, C2, C3, C4, C5, C6 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 45

S2 50

S3 20

S4 35

S5 40

S6 35 /

D(K) fabrika K'nin kapasitesi

/ F1 230 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 40

DC2 60

DC3 45

DC4 70 /

DM(I) müşteri I'nın talebi

/ C1 40

C2 20

C3 45

C4 50

C5 20

C6 30 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 1700

DC2 2500

DC3 2000

DC4 2700 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 9000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 25 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6
DC1	265	230	280	300	210	260
DC2	220	260	250	280	295	200
DC3	295	200	210	275	200	240
DC4	275	230	240	200	260	280 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4
F1	25	30	40	35 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

F1

S1 20
S2 35
S3 30
S4 45
S5 50
S6 35 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6
C1	999999	105	130	195	180	110
C2	105	999999	145	110	105	145
C3	130	145	999999	165	185	125
C4	195	110	165	999999	145	185
C5	180	105	185	145	999999	115
C6	110	145	125	185	115	999999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)*Y(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)*Y(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)* \text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK G : 10×2×5×10₁ Problemi GAMS Modeli

SETS

S tedârikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedârikçi S'nin kapasitesi

/ S1 100

S2 80

S3 60

S4 90

S5 100

S6 80

S7 110

S8 90

S9 60

S10 100 /

D(K) fabrika K'nin kapasitesi

/ F1 250

F2 200 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 110

DC2 90

DC3 120

DC4 110

DC5 130 /

DM(I) müşteri I'nın talebi

/ C1 35

C2 35

C3 50

C4 45

C5 40

C6 30

C7 30

C8 30

C9 40

C10 45 /

V(J) dağıtım merkezi J'nin sabit işletme mâliyeti

/ DC1 40000

DC2 40000

DC3 40000

DC4 40000

DC5 40000 /

G(K) fabrika K'nın sabit işletme mâliyeti

/ F1 150000

F2 75000 /

E(K) fabrika K'nın değişken mâliyeti

/ F1 25

F2 30 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
DC1	230	260	260	265	340	270	220	180	260	270
DC2	220	240	240	260	220	210	275	200	240	265
DC3	260	200	220	230	240	200	260	280	230	240
DC4	250	180	270	200	180	245	260	220	280	205
DC5	280	300	260	245	260	290	300	220	260	250 ;

TABLE

A(K,J) abrika K'dan dağıtım merkezi J'ye birim taşıma mâliyeti

	DC1	DC2	DC3	DC4	DC5
F1	45	80	30	50	75
F2	65	55	45	60	30 ;

TABLE

T(S,K) tedârikçi S'den fabrika K'ya birim satınalma ve taşıma mâliyeti

	F1	F2
S1	20	15
S2	25	45
S3	35	20
S4	35	30
S5	20	40
S6	30	15
S7	35	20
S8	40	30
S9	20	35
S10	25	40 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	99999	200	140	70	260	120	140	65	150	60
C2	200	99999	140	185	145	170	100	145	140	190
C3	140	140	99999	80	270	35	40	140	10	90
C4	70	185	80	99999	280	50	90	100	90	10
C5	260	145	270	280	99999	290	230	190	270	280
C6	120	170	35	50	290	99999	65	135	40	60
C7	140	100	40	90	230	65	99999	120	40	100
C8	65	145	140	100	190	135	120	99999	150	100
C9	150	140	10	90	270	40	40	150	99999	100
C10	60	190	90	10	280	60	100	100	100	99999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedârikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam mâliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedârikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = DM(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)* \text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK H : 10×2×5×10₂ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 90
S2 100
S3 80
S4 70
S5 60
S6 70
S7 100
S8 60
S9 50
S10 80 /

D(K) fabrika K'nin kapasitesi

/ F1 300
F2 400 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 200
DC2 150
DC3 140

DC4 100

DC5 130 /

DM(I) müşteri I'nın talebi

/ C1 40

C2 35

C3 25

C4 70

C5 25

C6 50

C7 80

C8 80

C9 65

C10 55 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 30000

DC2 27000

DC3 26000

DC4 20000

DC5 24000 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 255000

F2 300000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 20

F2 15 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
DC1	180	190	200	180	245	210	240	200	240	200
DC2	220	210	245	200	220	200	180	245	195	220
DC3	220	230	240	180	220	180	230	185	220	240
DC4	200	245	180	200	245	190	220	245	200	230
DC5	230	180	220	220	220	210	180	200	180	210 ;

TABLE

A(K,J) abrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4	DC5
F1	45	45	55	30	65
F2	70	80	45	75	30 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	30	35
S2	25	45
S3	35	20
S4	40	25
S5	30	50
S6	40	25
S7	35	20
S8	45	25
S9	25	35
S10	15	40 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	999999	140	100	150	160	140	100	100	100	90
C2	140	999999	170	70	100	195	100	70	75	120
C3	100	170	999999	80	170	85	90	140	80	90
C4	150	70	80	999999	185	95	90	100	90	100
C5	160	100	170	185	999999	100	95	90	120	150
C6	140	195	85	95	100	999999	105	135	110	95
C7	100	100	90	90	95	105	999999	110	70	100
C8	100	70	140	100	90	135	110	999999	135	90
C9	100	75	80	90	120	110	70	135	999999	120
C10	90	120	90	100	150	95	100	90	120	999999 ;

SCALAR UR ürünlerdeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)* \text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK İ : 10×2×5×10₃ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 80
S2 110
S3 60
S4 70
S5 100
S6 50
S7 60
S8 100
S9 60
S10 90 /

D(K) fabrika K'nin kapasitesi

/ F1 250
F2 350 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 100
DC2 350
DC3 300

DC4 100

DC5 150 /

DM(I) müşteri I'nın talebi

/ C1 80

C2 50

C3 20

C4 35

C5 40

C6 90

C7 40

C8 30

C9 40

C10 55 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 14000

DC2 30000

DC3 27000

DC4 15000

DC5 19000 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 200000

F2 310000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 30

F2 25 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
DC1	230	185	220	220	220	210	205	220	210	245
DC2	220	200	245	245	220	210	240	200	220	195
DC3	220	230	240	215	220	225	230	185	220	240
DC4	260	200	240	200	245	220	210	190	220	230
DC5	265	210	270	220	220	210	230	220	210	240 ;

TABLE

A(K,J) abrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4	DC5
F1	15	40	65	90	40
F2	40	25	35	80	55 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	30	50
S2	40	30
S3	45	75
S4	25	35
S5	45	55
S6	80	45
S7	35	20
S8	25	35
S9	15	40
S10	40	25 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	999999	125	95	95	90	100	105	95	125	105
C2	125	999999	90	120	150	90	90	95	150	130
C3	95	90	999999	120	110	125	90	135	105	100
C4	95	120	120	999999	150	115	155	110	95	120
C5	90	150	110	150	999999	90	120	115	140	145
C6	100	90	125	115	90	999999	100	170	185	95
C7	105	90	90	155	120	100	999999	150	90	100
C8	95	95	135	110	115	170	150	999999	95	100
C9	125	150	105	95	140	185	90	95	999999	135
C10	105	130	100	120	145	95	100	100	135	999999 ;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

$X(J,I,O)$ eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

$U(I)$ müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

COST .. $Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O)))$;

DEMAND(I) .. $\text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = DM(I)$;

DISCENTER_CAPACITY(J) .. $\text{SUM}(I, Q(J,I)) = L = W(J)*M(J)$;

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J)$;

FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K)*P(K)$;

FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K))*UR = G = \text{SUM}(J, F(K,J))*P(K)$;

SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;

ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;

OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)) = E = M(J)$;

OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)) = E = M(J)$;

ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,O,I)*Y(J,I)*Y(J,O)) = E = 1$;

ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I)*Y(J,I)) + \text{SUM}((J,O), X(J,I,O)*Y(J,I)*Y(J,O)) = E = 1$;

SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I)* \text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK J : 12×2×8×12₁ Problemi GAMS Modeli

SETS

S tedârikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5, DC6, DC7, DC8 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedârikçi S'nin kapasitesi

/ S1 70

S2 110

S3 80

S4 60

S5 80

S6 80

S7 90

S8 80

S9 90

S10 90

S11 70

S12 80

/

D(K) fabrika K'nin kapasitesi

/ F1 360

F2 400 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 80

DC2 80

DC3 180

DC4 80

DC5 100

DC6 140

DC7 100

DC8 140 /

DM(I) müşteri I'nın talebi

/ C1 60

C2 80

C3 30

C4 50

C5 70

C6 50

C7 50

C8 80

C9 70

C10 50

C11 50

C12 40 /

V(J) dağıtım merkezi J'nin sabit işletme mâliyeti

/ DC1 2000

DC2 2000

DC3 2000

DC4 2000

DC5 2000

DC6 2000

DC7 2000

DC8 2000 /

G(K) fabrika K'nın sabit işletme mâliyeti

/ F1 9000

F2 1000 /

E(K) fabrika K'nın değişken mâliyeti

/ F1 15

F2 12 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
DC1	310	300	310	305	320	300	360	325	320	315	300	320
DC2	330	320	340	335	300	300	300	300	350	320	340	330
DC3	335	340	325	330	380	300	320	350	355	340	360	340
DC4	340	380	350	345	380	320	340	360	320	340	320	340
DC5	350	310	360	355	380	300	360	320	360	345	365	340
DC6	330	350	320	325	360	340	380	335	320	325	355	325
DC7	350	370	340	355	340	300	360	360	350	345	315	345
DC8	340	320	320	340	340	300	360	345	340	340	340	340 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma mâliyeti

	DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8
F1	15	20	30	25	10	30	15	10

F2 20 10 15 10 15 10 30 25 ;

TABLE

T(S,K) tedârikçi S'den fabrika K'ya birim satınalma ve taşıma mâliyeti

	F1	F2
S1	15	25
S2	20	20
S3	20	20
S4	20	20
S5	25	10
S6	10	15
S7	25	10
S8	30	30
S9	15	20
S10	20	10
S11	30	15
S12	10	30 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma mâliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1	99999	60	40	10	295	80	125	50	50	10	70	80
C2	60	99999	100	75	235	80	90	30	90	50	80	60
C3	40	100	99999	25	335	110	165	80	35	50	75	100
C4	10	75	25	99999	310	90	135	60	40	20	70	85
C5	295	235	335	310	99999	290	230	250	310	285	280	245
C6	80	80	110	90	290	99999	65	100	130	80	145	135
C7	125	90	165	135	230	65	99999	120	170	120	170	150

C8 50 30 80 60 250 100 120 99999 60 40 50 35
 C9 50 90 35 40 310 130 170 60 99999 50 40 70
 C10 10 50 50 20 285 80 120 40 50 99999 65 70
 C11 70 80 75 70 280 145 170 50 40 65 99999 35
 C12 80 60 100 85 245 135 150 35 70 70 35 99999;
 SCALAR UR ürünlerdeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedârikçi S'den fabrika K'ya taşınan hammadde miktarı
 F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı
 Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı
 M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır
 P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır
 Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır
 N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır
 R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır
 X(J,I,O) eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır
 U(I) müşteri rotalarında alt turları engellemek için kullanılır
 Z toplam mâliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedârikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$\text{COST} \dots Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O))) ;$$

$$\text{DEMAND}(I) \dots \text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I) ;$$

$$\text{DISCENTER_CAPACITY}(J) \dots \text{SUM}(I, Q(J,I)) = L = W(J)*M(J) ;$$

$$\text{DISCENTER_EQUALITY}(J) \dots \text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J) ;$$

```

FACTORY_CAPACITY(K) .. SUM(J, F(K,J)) =L= D(K)*P(K) ;

FACTORY_EQUALITY(K) .. SUM(S, B(S,K))*UR =G= SUM(J, F(K,J))*P(K);

SUPPLIER_CAPACITY(S) .. SUM(K, B(S,K)) =L= H(S) ;

ONLY_SERVICED_BY_ONE_DC(I).. SUM(J, Y(J,I)) =E= 1;

OPEN_DC_START_ROUTE(J).. SUM(I, N(J,I)) =E= M(J);

OPEN_DC_END_ROUTE(J).. SUM(I, R(J,I)) =E= M(J);

ONLY_ONE_START_ROUTE(I).. SUM(J, N(J,I)*Y(J,I)) + SUM((J,O),
X(J,O,I)*Y(J,I)*Y(J,O)) =E= 1;

ONLY_ONE_END_ROUTE(I).. SUM(J, R(J,I)*Y(J,I)) + SUM((J,O),
X(J,I,O)*Y(J,I)*Y(J,O)) =E= 1;

SUBTOUR_ELIMINATION(I,O).. U(I) - U(O) + CARD(I)* SUM(J, X(J,I,O))
=L= CARD(I) - 1;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

```

EK K : 12×2×8×12₂ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5, DC6, DC7, DC8 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 50

S2 50

S3 80

S4 70

S5 80

S6 80

S7 80

S8 100

S9 140

S10 50

S11 50

S12 70

/

D(K) fabrika K'nin kapasitesi

/ F1 500

F2 350 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 90
DC2 75
DC3 140
DC4 100
DC5 95
DC6 100
DC7 135
DC8 160 /

DM(I) müşteri I'nın talebi

/ C1 70
C2 50
C3 50
C4 50
C5 60
C6 70
C7 50
C8 80
C9 70
C10 70
C11 80
C12 40 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 1000
DC2 1000
DC3 2000
DC4 1000

DC5 1100

DC6 1500

DC7 1100

DC8 1600 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 14000

F2 9000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 16

F2 25 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
DC1	320	340	360	320	340	330	360	370	340	355	340	300
DC2	300	360	320	360	360	340	300	340	330	360	320	360
DC3	340	355	360	325	320	340	320	360	340	300	360	325
DC4	330	360	300	300	365	340	340	320	340	320	300	300
DC5	340	300	320	350	355	325	360	365	340	340	320	350
DC6	340	320	320	340	330	340	390	335	320	325	340	300
DC7	340	340	360	360	340	300	360	360	350	345	340	320
DC8	320	325	325	320	340	350	355	325	360	365	340	340 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8
F1	30	35	15	30	15	10	15	25

F2 30 15 15 35 15 30 10 20 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	30	15
S2	25	10
S3	20	15
S4	20	10
S5	30	15
S6	10	15
S7	30	10
S8	15	30
S9	10	25
S10	20	10
S11	30	15
S12	20	30 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1	99999	100	105	95	125	100	150	120	100	180	95	110
C2	100	99999	105	130	170	180	110	100	115	95	130	125
C3	105	105	99999	160	150	110	165	120	110	130	95	105
C4	95	130	160	99999	170	100	135	115	140	160	165	110
C5	125	170	150	170	99999	185	100	135	140	155	140	175
C6	100	180	110	100	185	99999	135	140	165	105	160	135
C7	150	110	165	135	100	135	99999	120	170	120	170	150

C8	120	100	120	115	135	140	120	99999	160	95	140	120
C9	100	115	110	140	140	165	170	160	99999	100	105	125
C10	180	95	130	160	155	105	120	95	100	99999	165	125
C11	95	130	95	165	140	160	170	140	105	165	99999	115
C12	110	125	105	110	175	135	150	120	130	125	115	99999;

SCALAR UR üründeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

X(J,I,O) eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

U(I) müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$COST \dots Z = E = \sum(K, G(K) * P(K)) + \sum(J, V(J) * M(J)) + \sum((K, J), E(K) * F(K, J)) + \sum((S, K), T(S, K) * B(S, K)) + \sum((K, J), A(K, J) * F(K, J)) + \sum((J, I), N(J, I) * C(J, I)) + \sum((J, I), R(J, I) * C(J, I)) + \sum(O, \sum((J, I), X(J, I, O) * L(I, O))) ;$$

$$DEMAND(I) \dots \sum(J, Q(J, I) * Y(J, I) * M(J)) = G = DM(I) ;$$

$$DISCENTER_CAPACITY(J) \dots \sum(I, Q(J, I)) = L = W(J) * M(J) ;$$

DISCENTER_EQUALITY(J) .. $\text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I)) * M(J)$;
FACTORY_CAPACITY(K) .. $\text{SUM}(J, F(K,J)) = L = D(K) * P(K)$;
FACTORY_EQUALITY(K) .. $\text{SUM}(S, B(S,K)) * UR = G = \text{SUM}(J, F(K,J)) * P(K)$;
SUPPLIER_CAPACITY(S) .. $\text{SUM}(K, B(S,K)) = L = H(S)$;
ONLY_SERVICED_BY_ONE_DC(I) .. $\text{SUM}(J, Y(J,I)) = E = 1$;
OPEN_DC_START_ROUTE(J) .. $\text{SUM}(I, N(J,I)) = E = M(J)$;
OPEN_DC_END_ROUTE(J) .. $\text{SUM}(I, R(J,I)) = E = M(J)$;
ONLY_ONE_START_ROUTE(I) .. $\text{SUM}(J, N(J,I) * Y(J,I)) + \text{SUM}((J,O), X(J,O,I) * Y(J,I) * Y(J,O)) = E = 1$;
ONLY_ONE_END_ROUTE(I) .. $\text{SUM}(J, R(J,I) * Y(J,I)) + \text{SUM}((J,O), X(J,I,O) * Y(J,I) * Y(J,O)) = E = 1$;
SUBTOUR_ELIMINATION(I,O) .. $U(I) - U(O) + \text{CARD}(I) * \text{SUM}(J, X(J,I,O)) = L = \text{CARD}(I) - 1$;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

EK L : 12×2×8×12₃ Problemi GAMS Modeli

SETS

S tedarikçiler / S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12 /

K fabrikalar / F1, F2 /

J dağıtım merkezleri / DC1, DC2, DC3, DC4, DC5, DC6, DC7, DC8 /

I müşteriler / C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12 / ;

ALIAS (I,O);

PARAMETERS

H(S) tedarikçi S'nin kapasitesi

/ S1 90
S2 70
S3 80
S4 45
S5 100
S6 40
S7 80
S8 60
S9 55
S10 120
S11 50
S12 70 /

D(K) fabrika K'nin kapasitesi

/ F1 420
F2 360 /

W(J) dağıtım merkezi J'nin kapasitesi

/ DC1 100

DC2 90

DC3 70

DC4 90

DC5 120

DC6 80

DC7 70

DC8 160 /

DM(I) müşteri I'nın talebi

/ C1 50

C2 75

C3 65

C4 40

C5 35

C6 50

C7 100

C8 50

C9 60

C10 70

C11 100

C12 50 /

V(J) dağıtım merkezi J'nin sabit işletme maliyeti

/ DC1 1500

DC2 2000

DC3 1800

DC4 1400

DC5 1600

DC6 1300

DC7 1100

DC8 2000 /

G(K) fabrika K'nın sabit işletme maliyeti

/ F1 12000

F2 10000 /

E(K) fabrika K'nın değişken maliyeti

/ F1 16

F2 10 / ;

TABLE

C(J,I) dağıtım merkezi J'den müşteri I'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
DC1	340	300	360	350	345	340	340	360	320	365	300	315
DC2	330	315	370	300	330	300	310	365	295	325	365	335
DC3	320	345	360	310	330	325	345	310	330	310	345	330
DC4	300	365	305	300	360	300	330	355	300	330	315	320
DC5	345	340	330	335	330	315	310	365	295	325	350	340
DC6	360	345	340	340	360	340	380	335	320	340	375	380
DC7	345	320	365	300	325	360	340	380	335	310	330	325
DC8	340	325	360	340	300	300	320	350	355	340	320	315 ;

TABLE

A(K,J) fabrika K'dan dağıtım merkezi J'ye birim taşıma maliyeti

	DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8
F1	80	30	50	55	30	65	55	35
F2	55	45	60	45	75	30	60	45 ;

TABLE

T(S,K) tedarikçi S'den fabrika K'ya birim satınalma ve taşıma maliyeti

	F1	F2
S1	20	15
S2	25	35
S3	40	55
S4	80	45
S5	35	25
S6	25	30
S7	30	40
S8	30	15
S9	35	20
S10	40	30
S11	20	35
S12	20	35 ;

TABLE

L(I,O) müşteri I'dan müşteri O'ya toplam taşıma maliyeti

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1	99999	210	140	95	260	130	140	100	150	110	155	130
C2	210	99999	110	100	135	115	170	125	180	165	145	95
C3	140	110	99999	135	140	215	130	115	245	135	150	125
C4	95	100	135	99999	150	185	135	150	245	235	135	185
C5	260	135	140	150	99999	110	140	140	165	170	185	150
C6	130	115	215	185	110	99999	170	125	190	165	145	120
C7	140	170	130	135	140	170	99999	120	170	120	170	150
C8	100	125	115	150	140	125	120	99999	165	135	120	100

C9 150 180 245 245 165 190 180 165 99999 120 135 175
C10 110 165 135 235 170 165 195 135 120 99999 150 170
C11 155 145 150 135 185 145 110 120 135 150 99999 160
C12 130 95 125 185 150 120 155 100 175 170 160 99999 ;

SCALAR UR ürünlerdeki birim hammadde kullanım oranı /1/ ;

VARIABLES

B(S,K) tedarikçi S'den fabrika K'ya taşınan hammadde miktarı

F(K,J) fabrika K'dan dağıtım merkezi J'ye taşınan ürün miktarı

Q(J,I) dağıtım merkezi J'den müşteri I'ya taşınan ürün miktarı

M(J) eğer dağıtım merkezi J aktifse 1'dir aksi halde 0'dır

P(K) eğer fabrika K aktifse 1'dir aksi halde 0'dır

Y(J,I) eğer dağıtım merkezi J müşteri I'ya ürün sağlıyorsa 1'dir aksi halde 0'dır

N(J,I) eğer dağıtım merkezi J'den çıkan araç ilk olarak müşteri I'ya uğruyorsa 1'dir aksi halde 0'dır

R(J,I) eğer müşteri I'dan çıkan araç dağıtım merkezi J'ya geri dönüyorsa 1'dir aksi halde 0'dır

X(J,I,O) eğer dağıtım merkezi J'den çıkan araç müşteri I'dan sonra müşteri O'ya uğruyorsa 1'dir aksi halde 0'dır

U(I) müşteri rotalarında alt turları engellemek için kullanılır

Z toplam maliyet;

POSITIVE VARIABLE B, F, Q ;

BINARY VARIABLE M, P, Y, N, R, X, U(I);

EQUATIONS

COST amaç fonksiyon tanımı

FACTORY_EQUALITY(K) fabrika K'ya ulaşan hammadde ve çıkan ürün denk olmalıdır

DISCENTER_CAPACITY(J) dağıtım merkezi J'ye ulaşan ürün kapasiteyi geçmemelidir

DEMAND(I) müşteri I'nın talebi karşılanmalıdır

DISCENTER_EQUALITY(J) dağıtım merkezi J'ye gelen ve çıkan ürün denk olmalıdır

FACTORY_CAPACITY(K) fabrika K'ya ulaşan hammadde kapasiteyi geçmemelidir

SUPPLIER_CAPACITY(S) tedarikçi S'den çıkan hammadde kapasiteyi geçmemelidir

ONLY_SERVICED_BY_ONE_DC(I) her müşteri I'ya sadece bir dağıtım merkezinden ürün sağlanmalıdır

OPEN_DC_START_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezinden çıkan bir araç olmalıdır

OPEN_DC_END_ROUTE(J) eğer dağıtım merkezi J aktifse dağıtım merkezine dönen bir araç olmalıdır

ONLY_ONE_START_ROUTE(I) her müşteri I'ya bir araç ulaşmalıdır

ONLY_ONE_END_ROUTE(I) her müşteri I'dan bir araç dönmelidir

SUBTOUR_ELIMINATION(I,O) müşteri rotalarında alt turlar engellenmelidir;

$$\text{COST} \dots Z = E = \text{SUM}(K, G(K)*P(K)) + \text{SUM}(J, V(J)*M(J)) + \text{SUM}((K,J), E(K)*F(K,J)) + \text{SUM}((S,K), T(S,K)*B(S,K)) + \text{SUM}((K,J), A(K,J)*F(K,J)) + \text{SUM}((J,I), N(J,I)*C(J,I)) + \text{SUM}((J,I), R(J,I)*C(J,I)) + \text{SUM}(O, \text{SUM}((J,I), X(J,I,O)*L(I,O))) ;$$

$$\text{DEMAND}(I) \dots \text{SUM}(J, Q(J,I)*Y(J,I)*M(J)) = G = \text{DM}(I) ;$$

$$\text{DISCENTER_CAPACITY}(J) \dots \text{SUM}(I, Q(J,I)) = L = W(J)*M(J) ;$$

$$\text{DISCENTER_EQUALITY}(J) \dots \text{SUM}(K, F(K,J)) = G = \text{SUM}(I, Q(J,I))*M(J) ;$$

```

FACTORY_CAPACITY(K) .. SUM(J, F(K,J)) =L= D(K)*P(K) ;

FACTORY_EQUALITY(K) .. SUM(S, B(S,K))*UR =G= SUM(J, F(K,J))*P(K);

SUPPLIER_CAPACITY(S) .. SUM(K, B(S,K)) =L= H(S) ;

ONLY_SERVICED_BY_ONE_DC(I).. SUM(J, Y(J,I)) =E= 1;

OPEN_DC_START_ROUTE(J).. SUM(I, N(J,I)) =E= M(J);

OPEN_DC_END_ROUTE(J).. SUM(I, R(J,I)) =E= M(J);

ONLY_ONE_START_ROUTE(I).. SUM(J, N(J,I)*Y(J,I)) + SUM((J,O),
X(J,O,I)*Y(J,I)*Y(J,O)) =E= 1;

ONLY_ONE_END_ROUTE(I).. SUM(J, R(J,I)*Y(J,I)) + SUM((J,O),
X(J,I,O)*Y(J,I)*Y(J,O)) =E= 1;

SUBTOUR_ELIMINATION(I,O).. U(I) - U(O) + CARD(I)* SUM(J, X(J,I,O))
=L= CARD(I) - 1;

MODEL LAT_MODEL /ALL/ ;

OPTION OPTCR=0;

SOLVE LAT_MODEL USING MINLP MINIMIZING Z ;

DISPLAY B.L, F.L, Q.L, M.L, P.L, Y.L, N.L, R.L, X.L, Z.L ;

```

EK M : $2 \times 2 \times 2 \times 2_1$ Problemi İçin Bulunan Sonuçlar

$2 \times 2 \times 2 \times 2_1$ problemi için bulunan optimum değer 312090 olup, Çizelge E.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (saniye cinsinden) göstermektedir.

Çizelge M.1 : $2 \times 2 \times 2 \times 2_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3

Çizelge M.1 (Devam) : $2 \times 2 \times 2 \times 2_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2

Çizelge M.1 (Devam) : $2 \times 2 \times 2 \times 2_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	1	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	3
312090	0	0,00%	2	312090	0	0,00%	5	312090	0	0,00%	2
312090	0	0,00%	2	312090	0	0,00%	4	312090	0	0,00%	2
312090	0	0,00%	1	312090	0	0,00%	5	312090	0	0,00%	3

EK N : 2×2×2×2₂ Problemi İçin Bulunan Sonuçlar

2×2×2×2₂ problemi için bulunan optimum değer 300490 olup, Çizelge N.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge N.1 : 2×2×2×2₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3

Çizelge N.1 (Devam) : 2×2×2×2₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4

Çizelge N.1 (Devam) : $2 \times 2 \times 2 \times 2_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre	Sonuç	Fark	Yüzde	Süre	Sonuç	Fark	Yüzde	Süre
			(sn)				(sn)				(sn)
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	2	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	3	300490	0	0,00%	6	300490	0	0,00%	4
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	6	300490	0	0,00%	3
300490	0	0,00%	3	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	3
300490	0	0,00%	1	300490	0	0,00%	5	300490	0	0,00%	4
300490	0	0,00%	2	300490	0	0,00%	5	300490	0	0,00%	4

EK O : $2 \times 2 \times 2 \times 2_3$ Problemi İçin Bulunan Sonuçlar

$2 \times 2 \times 2 \times 2_3$ problemi için bulunan optimum değer 372680 olup, Çizelge O.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge O.1 : $2 \times 2 \times 2 \times 2_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3

Çizelge O.1 (Devam) : $2 \times 2 \times 2 \times 2_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	3

Çizelge O.1 (Devam) : $2 \times 2 \times 2 \times 2_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	3	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	3
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	2	372680	0	0,00%	6	372680	0	0,00%	3
372680	0	0,00%	1	372680	0	0,00%	5	372680	0	0,00%	4
372680	0	0,00%	3	372680	0	0,00%	6	372680	0	0,00%	3

EK P : 6×1×4×6₁ Problemi İçin Bulunan Sonuçlar

6×1×4×6₁ problemi için bulunan optimum değer 35520 olup, Çizelge P.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge P.1 : 6×1×4×6₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
35520	0	0,00%	8	35525	5	0,01%	19	35525	5	0,01%	12
35520	0	0,00%	8	35525	5	0,01%	17	35525	5	0,01%	11
35525	5	0,01%	9	35530	10	0,03%	18	35530	10	0,03%	12
35520	0	0,00%	9	35530	10	0,03%	17	35525	5	0,01%	11
35520	0	0,00%	9	35530	10	0,03%	18	35520	0	0,00%	11
35520	0	0,00%	9	35520	0	0,00%	19	35530	10	0,03%	12
35520	0	0,00%	9	35525	5	0,01%	17	35525	5	0,01%	11
35520	0	0,00%	8	35520	0	0,00%	17	35525	5	0,01%	11
35520	0	0,00%	9	35530	10	0,03%	17	35525	5	0,01%	11
35525	5	0,01%	8	35530	10	0,03%	17	35525	5	0,01%	11
35520	0	0,00%	8	35530	10	0,03%	19	35525	5	0,01%	11
35525	5	0,01%	9	35530	10	0,03%	17	35520	0	0,00%	11
35525	5	0,01%	8	35525	5	0,01%	18	35525	5	0,01%	12
35520	0	0,00%	9	35530	10	0,03%	19	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	18	35530	10	0,03%	11
35525	5	0,01%	9	35530	10	0,03%	17	35525	5	0,01%	12
35525	5	0,01%	8	35525	5	0,01%	19	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	18	35525	5	0,01%	11
35520	0	0,00%	9	35525	5	0,01%	19	35525	5	0,01%	12
35520	0	0,00%	8	35520	0	0,00%	17	35525	5	0,01%	11
35520	0	0,00%	8	35530	10	0,03%	18	35530	10	0,03%	11
35525	5	0,01%	9	35520	0	0,00%	18	35525	5	0,01%	11
35520	0	0,00%	9	35525	5	0,01%	19	35520	0	0,00%	12
35520	0	0,00%	9	35525	5	0,01%	17	35520	0	0,00%	12
35525	5	0,01%	9	35525	5	0,01%	17	35520	0	0,00%	12
35520	0	0,00%	8	35525	5	0,01%	16	35520	0	0,00%	11
35520	0	0,00%	8	35525	5	0,01%	17	35525	5	0,01%	12

Çizelge P.1 (Devam) : $6 \times 1 \times 4 \times 6_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
35520	0	0,00%	9	35530	10	0,03%	17	35525	5	0,01%	12
35525	5	0,01%	9	35520	0	0,00%	18	35520	0	0,00%	12
35520	0	0,00%	9	35530	10	0,03%	19	35530	10	0,03%	11
35520	0	0,00%	8	35525	5	0,01%	18	35520	0	0,00%	12
35525	5	0,01%	9	35520	0	0,00%	16	35525	5	0,01%	11
35525	5	0,01%	9	35530	10	0,03%	18	35520	0	0,00%	12
35525	5	0,01%	8	35525	5	0,01%	19	35525	5	0,01%	12
35525	5	0,01%	8	35530	10	0,03%	16	35525	5	0,01%	11
35520	0	0,00%	8	35520	0	0,00%	19	35525	5	0,01%	12
35525	5	0,01%	8	35525	5	0,01%	18	35525	5	0,01%	12
35520	0	0,00%	9	35520	0	0,00%	17	35525	5	0,01%	12
35520	0	0,00%	8	35530	10	0,03%	17	35530	10	0,03%	12
35520	0	0,00%	9	35525	5	0,01%	19	35520	0	0,00%	13
35525	5	0,01%	9	35525	5	0,01%	17	35520	0	0,00%	11
35520	0	0,00%	8	35520	0	0,00%	16	35525	5	0,01%	11
35520	0	0,00%	9	35525	5	0,01%	16	35530	10	0,03%	11
35525	5	0,01%	9	35530	10	0,03%	18	35520	0	0,00%	12
35520	0	0,00%	8	35525	5	0,01%	16	35520	0	0,00%	12
35520	0	0,00%	8	35520	0	0,00%	18	35520	0	0,00%	12
35520	0	0,00%	8	35530	10	0,03%	17	35520	0	0,00%	11
35520	0	0,00%	9	35530	10	0,03%	17	35530	10	0,03%	11
35520	0	0,00%	9	35525	5	0,01%	16	35525	5	0,01%	10
35525	5	0,01%	8	35530	10	0,03%	16	35520	0	0,00%	12
35525	5	0,01%	8	35520	0	0,00%	19	35520	0	0,00%	12
35520	0	0,00%	9	35525	5	0,01%	17	35520	0	0,00%	12
35520	0	0,00%	8	35530	10	0,03%	19	35520	0	0,00%	11
35525	5	0,01%	8	35525	5	0,01%	16	35525	5	0,01%	12
35520	0	0,00%	9	35520	0	0,00%	17	35525	5	0,01%	12
35520	0	0,00%	8	35525	5	0,01%	19	35525	5	0,01%	12
35520	0	0,00%	9	35525	5	0,01%	16	35530	10	0,03%	11
35520	0	0,00%	8	35525	5	0,01%	19	35525	5	0,01%	11
35520	0	0,00%	9	35530	10	0,03%	16	35525	5	0,01%	12
35525	5	0,01%	8	35525	5	0,01%	17	35520	0	0,00%	12
35520	0	0,00%	8	35525	5	0,01%	17	35520	0	0,00%	13
35520	0	0,00%	8	35525	5	0,01%	17	35525	5	0,01%	12
35520	0	0,00%	9	35525	5	0,01%	17	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	18	35520	0	0,00%	11
35525	5	0,01%	9	35525	5	0,01%	19	35530	10	0,03%	12
35525	5	0,01%	9	35525	5	0,01%	17	35530	10	0,03%	11
35520	0	0,00%	8	35525	5	0,01%	16	35525	5	0,01%	12
35520	0	0,00%	9	35525	5	0,01%	18	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	17	35520	0	0,00%	12

Çizelge P.1 (Devam) : $6 \times 1 \times 4 \times 6_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
35520	0	0,00%	8	35525	5	0,01%	19	35530	10	0,03%	10
35520	0	0,00%	9	35530	10	0,03%	18	35525	5	0,01%	11
35520	0	0,00%	8	35530	10	0,03%	19	35525	5	0,01%	11
35520	0	0,00%	8	35530	10	0,03%	16	35520	0	0,00%	11
35525	5	0,01%	8	35530	10	0,03%	19	35525	5	0,01%	11
35520	0	0,00%	8	35520	0	0,00%	18	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	18	35525	5	0,01%	12
35520	0	0,00%	8	35525	5	0,01%	17	35525	5	0,01%	12
35525	5	0,01%	9	35525	5	0,01%	19	35530	10	0,03%	11
35520	0	0,00%	9	35525	5	0,01%	18	35525	5	0,01%	12
35520	0	0,00%	9	35525	5	0,01%	18	35525	5	0,01%	12
35525	5	0,01%	8	35525	5	0,01%	18	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	18	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	18	35520	0	0,00%	12
35520	0	0,00%	8	35520	0	0,00%	17	35530	10	0,03%	12
35520	0	0,00%	9	35530	10	0,03%	19	35530	10	0,03%	11
35525	5	0,01%	8	35525	5	0,01%	16	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	17	35520	0	0,00%	12
35520	0	0,00%	9	35525	5	0,01%	16	35525	5	0,01%	11
35525	5	0,01%	8	35520	0	0,00%	17	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	16	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	18	35530	10	0,03%	12
35520	0	0,00%	8	35520	0	0,00%	18	35525	5	0,01%	11
35520	0	0,00%	8	35525	5	0,01%	16	35525	5	0,01%	12
35520	0	0,00%	9	35530	10	0,03%	17	35525	5	0,01%	11
35525	5	0,01%	9	35525	5	0,01%	18	35525	5	0,01%	10
35520	0	0,00%	8	35525	5	0,01%	19	35525	5	0,01%	11
35520	0	0,00%	8	35520	0	0,00%	16	35525	5	0,01%	11
35520	0	0,00%	9	35520	0	0,00%	18	35525	5	0,01%	10
35525	5	0,01%	8	35520	0	0,00%	18	35525	5	0,01%	12
35520	0	0,00%	8	35525	5	0,01%	16	35525	5	0,01%	12

EK Q : 6×1×4×6₂ Problemi İçin Bulunan Sonuçlar

6×1×4×6₂ problemi için bulunan optimum değer 26220 olup, Çizelge Q.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge Q.1 : 6×1×4×6₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
26220	0	0,00%	10	26220	0	0,00%	17	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	18	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	16	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	11
26220	0	0,00%	9	26220	0	0,00%	19	26225	5	0,02%	13
26225	5	0,02%	10	26225	5	0,02%	19	26225	5	0,02%	11
26220	0	0,00%	9	26220	0	0,00%	19	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	18	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	16	26220	0	0,00%	11
26220	0	0,00%	9	26230	10	0,04%	17	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	18	26225	5	0,02%	12
26225	5	0,02%	9	26225	5	0,02%	19	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	11
26220	0	0,00%	10	26220	0	0,00%	17	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	13
26220	0	0,00%	10	26220	0	0,00%	18	26225	5	0,02%	11
26220	0	0,00%	9	26220	0	0,00%	19	26220	0	0,00%	11
26220	0	0,00%	10	26220	0	0,00%	17	26220	0	0,00%	11
26220	0	0,00%	10	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	18	26225	5	0,02%	11

Çizelge Q.1 (Devam) : $6 \times 1 \times 4 \times 6_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	18	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	18	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	18	26220	0	0,00%	11
26220	0	0,00%	9	26225	5	0,02%	16	26220	0	0,00%	11
26220	0	0,00%	10	26220	0	0,00%	16	26220	0	0,00%	11
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	11
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	12
26225	5	0,02%	9	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	10	26220	0	0,00%	16	26220	0	0,00%	11
26220	0	0,00%	10	26220	0	0,00%	18	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	17	26225	5	0,02%	11
26220	0	0,00%	9	26225	5	0,02%	17	26225	5	0,02%	11
26220	0	0,00%	9	26220	0	0,00%	16	26225	5	0,02%	13
26220	0	0,00%	10	26225	5	0,02%	16	26220	0	0,00%	11
26220	0	0,00%	10	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	9	26220	0	0,00%	17	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	11
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	10	26220	0	0,00%	17	26225	5	0,02%	11
26220	0	0,00%	10	26220	0	0,00%	17	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	18	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	11
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	18	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	11
26220	0	0,00%	10	26225	5	0,02%	18	26220	0	0,00%	12
26220	0	0,00%	10	26220	0	0,00%	17	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	19	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	17	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	16	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	18	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	16	26225	5	0,02%	12
26225	5	0,02%	10	26220	0	0,00%	19	26225	5	0,02%	13
26220	0	0,00%	9	26220	0	0,00%	19	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	19	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	10
26220	0	0,00%	9	26220	0	0,00%	19	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	10
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	12

Çizelge Q.1 (Devam) : $6 \times 1 \times 4 \times 6_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
26220	0	0,00%	10	26225	5	0,02%	16	26220	0	0,00%	10
26220	0	0,00%	9	26230	10	0,04%	19	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	16	26225	5	0,02%	12
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	18	26220	0	0,00%	11
26220	0	0,00%	9	26225	5	0,02%	16	26225	5	0,02%	13
26220	0	0,00%	10	26225	5	0,02%	18	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	19	26225	5	0,02%	12
26220	0	0,00%	9	26220	0	0,00%	19	26220	0	0,00%	11
26220	0	0,00%	9	26220	0	0,00%	18	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	19	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	10
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	12
26225	5	0,02%	10	26225	5	0,02%	19	26220	0	0,00%	11
26220	0	0,00%	9	26225	5	0,02%	19	26225	5	0,02%	11
26220	0	0,00%	10	26220	0	0,00%	19	26225	5	0,02%	11
26220	0	0,00%	10	26220	0	0,00%	18	26220	0	0,00%	10
26220	0	0,00%	10	26220	0	0,00%	19	26220	0	0,00%	12
26220	0	0,00%	9	26225	5	0,02%	17	26220	0	0,00%	12
26220	0	0,00%	10	26225	5	0,02%	18	26225	5	0,02%	11
26220	0	0,00%	10	26225	5	0,02%	16	26220	0	0,00%	11
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	11
26220	0	0,00%	9	26220	0	0,00%	17	26225	5	0,02%	11
26220	0	0,00%	9	26225	5	0,02%	17	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	17	26225	5	0,02%	12
26225	5	0,02%	9	26225	5	0,02%	17	26225	5	0,02%	12
26220	0	0,00%	10	26220	0	0,00%	19	26225	5	0,02%	12
26220	0	0,00%	10	26225	5	0,02%	16	26225	5	0,02%	10

EK R : $6 \times 1 \times 4 \times 6_3$ Problemi İçin Bulunan Sonuçlar

$6 \times 1 \times 4 \times 6_3$ problemi için bulunan optimum değer 38925 olup, Çizelge R.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge R.1 : $6 \times 1 \times 4 \times 6_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
38930	5	0,01%	10	38925	0	0,00%	17	38935	10	0,03%	11
38925	0	0,00%	10	38925	0	0,00%	18	38930	5	0,01%	12
38925	0	0,00%	9	38925	0	0,00%	17	38925	0	0,00%	11
38925	0	0,00%	10	38930	5	0,01%	16	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	17	38935	10	0,03%	11
38925	0	0,00%	10	38930	5	0,01%	17	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	16	38930	5	0,01%	12
38930	5	0,01%	10	38925	0	0,00%	17	38935	10	0,03%	11
38925	0	0,00%	9	38930	5	0,01%	18	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	18	38925	0	0,00%	11
38925	0	0,00%	9	38925	0	0,00%	18	38930	5	0,01%	13
38930	5	0,01%	10	38935	10	0,03%	18	38930	5	0,01%	11
38930	5	0,01%	10	38930	5	0,01%	18	38925	0	0,00%	11
38930	5	0,01%	9	38935	10	0,03%	18	38930	5	0,01%	12
38925	0	0,00%	10	38930	5	0,01%	19	38925	0	0,00%	12
38925	0	0,00%	9	38930	5	0,01%	17	38930	5	0,01%	11
38925	0	0,00%	9	38925	0	0,00%	19	38925	0	0,00%	12
38925	0	0,00%	10	38935	10	0,03%	17	38935	10	0,03%	11
38930	5	0,01%	10	38925	0	0,00%	16	38935	10	0,03%	12
38930	5	0,01%	10	38925	0	0,00%	17	38935	10	0,03%	12
38930	5	0,01%	9	38930	5	0,01%	18	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	17	38935	10	0,03%	11
38930	5	0,01%	9	38935	10	0,03%	16	38935	10	0,03%	13
38930	5	0,01%	10	38930	5	0,01%	16	38925	0	0,00%	11
38925	0	0,00%	9	38925	0	0,00%	18	38930	5	0,01%	11

Çizelge R.1 (Devam) : $6 \times 1 \times 4 \times 6_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
38930	5	0,01%	9	38930	5	0,01%	18	38930	5	0,01%	10
38925	0	0,00%	10	38925	0	0,00%	19	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	18	38935	10	0,03%	10
38930	5	0,01%	9	38925	0	0,00%	17	38935	10	0,03%	11
38930	5	0,01%	10	38935	10	0,03%	18	38930	5	0,01%	12
38925	0	0,00%	10	38925	0	0,00%	17	38935	10	0,03%	12
38930	5	0,01%	10	38925	0	0,00%	18	38935	10	0,03%	12
38925	0	0,00%	10	38925	0	0,00%	17	38935	10	0,03%	12
38930	5	0,01%	9	38935	10	0,03%	18	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	17	38925	0	0,00%	11
38930	5	0,01%	10	38935	10	0,03%	17	38935	10	0,03%	11
38925	0	0,00%	9	38930	5	0,01%	17	38925	0	0,00%	12
38930	5	0,01%	10	38930	5	0,01%	18	38930	5	0,01%	11
38930	5	0,01%	10	38930	5	0,01%	18	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	19	38935	10	0,03%	11
38930	5	0,01%	10	38930	5	0,01%	19	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	16	38930	5	0,01%	12
38930	5	0,01%	9	38930	5	0,01%	18	38930	5	0,01%	11
38930	5	0,01%	9	38925	0	0,00%	17	38930	5	0,01%	12
38930	5	0,01%	10	38930	5	0,01%	17	38935	10	0,03%	12
38925	0	0,00%	9	38925	0	0,00%	17	38925	0	0,00%	12
38930	5	0,01%	10	38935	10	0,03%	19	38925	0	0,00%	12
38925	0	0,00%	9	38935	10	0,03%	18	38930	5	0,01%	11
38925	0	0,00%	10	38925	0	0,00%	17	38935	10	0,03%	13
38925	0	0,00%	10	38930	5	0,01%	18	38930	5	0,01%	10
38925	0	0,00%	9	38935	10	0,03%	18	38935	10	0,03%	11
38925	0	0,00%	10	38930	5	0,01%	16	38925	0	0,00%	10
38925	0	0,00%	10	38935	10	0,03%	19	38930	5	0,01%	12
38925	0	0,00%	10	38930	5	0,01%	18	38925	0	0,00%	11
38930	5	0,01%	10	38925	0	0,00%	18	38925	0	0,00%	11
38925	0	0,00%	10	38935	10	0,03%	18	38935	10	0,03%	12
38925	0	0,00%	9	38930	5	0,01%	18	38925	0	0,00%	11
38925	0	0,00%	9	38925	0	0,00%	17	38930	5	0,01%	12
38925	0	0,00%	9	38930	5	0,01%	18	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	17	38925	0	0,00%	11
38925	0	0,00%	10	38935	10	0,03%	17	38930	5	0,01%	12
38925	0	0,00%	9	38930	5	0,01%	19	38935	10	0,03%	12
38925	0	0,00%	10	38930	5	0,01%	18	38930	5	0,01%	11
38930	5	0,01%	9	38930	5	0,01%	17	38935	10	0,03%	12
38925	0	0,00%	9	38935	10	0,03%	19	38930	5	0,01%	11
38925	0	0,00%	9	38935	10	0,03%	16	38925	0	0,00%	11
38930	5	0,01%	9	38930	5	0,01%	18	38925	0	0,00%	12

Çizelge R.1 (Devam) : $6 \times 1 \times 4 \times 6_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
38925	0	0,00%	10	38930	5	0,01%	18	38935	10	0,03%	12
38925	0	0,00%	10	38930	5	0,01%	16	38935	10	0,03%	12
38925	0	0,00%	9	38935	10	0,03%	17	38930	5	0,01%	11
38930	5	0,01%	9	38930	5	0,01%	18	38925	0	0,00%	12
38930	5	0,01%	10	38930	5	0,01%	19	38925	0	0,00%	12
38925	0	0,00%	9	38930	5	0,01%	19	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	19	38925	0	0,00%	10
38930	5	0,01%	10	38925	0	0,00%	17	38930	5	0,01%	12
38925	0	0,00%	9	38925	0	0,00%	18	38925	0	0,00%	11
38925	0	0,00%	10	38935	10	0,03%	18	38930	5	0,01%	12
38930	5	0,01%	10	38930	5	0,01%	19	38930	5	0,01%	11
38930	5	0,01%	9	38930	5	0,01%	19	38935	10	0,03%	11
38925	0	0,00%	10	38925	0	0,00%	19	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	17	38930	5	0,01%	11
38930	5	0,01%	9	38930	5	0,01%	19	38930	5	0,01%	11
38930	5	0,01%	10	38935	10	0,03%	18	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	17	38930	5	0,01%	12
38925	0	0,00%	10	38935	10	0,03%	17	38925	0	0,00%	12
38930	5	0,01%	10	38935	10	0,03%	18	38930	5	0,01%	11
38930	5	0,01%	9	38925	0	0,00%	19	38925	0	0,00%	12
38925	0	0,00%	9	38935	10	0,03%	17	38930	5	0,01%	12
38930	5	0,01%	9	38930	5	0,01%	17	38925	0	0,00%	11
38930	5	0,01%	9	38925	0	0,00%	16	38935	10	0,03%	10
38930	5	0,01%	10	38935	10	0,03%	17	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	17	38925	0	0,00%	13
38925	0	0,00%	9	38935	10	0,03%	18	38930	5	0,01%	11
38925	0	0,00%	10	38930	5	0,01%	18	38930	5	0,01%	12
38930	5	0,01%	10	38935	10	0,03%	18	38925	0	0,00%	11
38930	5	0,01%	9	38925	0	0,00%	17	38925	0	0,00%	12
38925	0	0,00%	10	38930	5	0,01%	17	38925	0	0,00%	11
38930	5	0,01%	9	38935	10	0,03%	18	38925	0	0,00%	11
38925	0	0,00%	9	38935	10	0,03%	18	38935	10	0,03%	12
38925	0	0,00%	10	38935	10	0,03%	16	38935	10	0,03%	12

EK S : 10×2×5×10₁ Problemi İçin Bulunan Sonuçlar

10×2×5×10₁ problemi için bulunan optimum değer 460330 olup, Çizelge S.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge S.1 : 10×2×5×10₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
460375	45	0,01%	30	460535	205	0,04%	43	460505	175	0,04%	33
460380	50	0,01%	30	460445	115	0,02%	38	460505	175	0,04%	33
460510	180	0,04%	26	460645	315	0,07%	42	460540	210	0,05%	31
460415	85	0,02%	27	460505	175	0,04%	44	460340	10	0,00%	31
460330	0	0,00%	29	460535	205	0,04%	42	460520	190	0,04%	34
460370	40	0,01%	26	460605	275	0,06%	37	460395	65	0,01%	35
460470	140	0,03%	31	460425	95	0,02%	39	460470	140	0,03%	32
460410	80	0,02%	27	460330	0	0,00%	41	460330	0	0,00%	33
460460	130	0,03%	31	460480	150	0,03%	39	460465	135	0,03%	36
460445	115	0,02%	30	460625	295	0,06%	44	460390	60	0,01%	34
460455	125	0,03%	28	460630	300	0,07%	38	460490	160	0,03%	31
460450	120	0,03%	30	460780	450	0,10%	39	460555	225	0,05%	31
460420	90	0,02%	29	460370	40	0,01%	43	460395	65	0,01%	36
460375	45	0,01%	30	460600	270	0,06%	42	460445	115	0,02%	31
460335	5	0,00%	26	460510	180	0,04%	44	460420	90	0,02%	33
460445	115	0,02%	26	460535	205	0,04%	38	460425	95	0,02%	36
460470	140	0,03%	30	460720	390	0,08%	37	460425	95	0,02%	32
460335	5	0,00%	26	460620	290	0,06%	42	460525	195	0,04%	34
460455	125	0,03%	31	460790	460	0,10%	44	460490	160	0,03%	33
460465	135	0,03%	27	460600	270	0,06%	41	460490	160	0,03%	31
460480	150	0,03%	31	460410	80	0,02%	41	460485	155	0,03%	36
460490	160	0,03%	27	460610	280	0,06%	41	460335	5	0,00%	32
460460	130	0,03%	28	460725	395	0,09%	40	460365	35	0,01%	34
460495	165	0,04%	30	460730	400	0,09%	38	460455	125	0,03%	34
460515	185	0,04%	27	460465	135	0,03%	41	460495	165	0,04%	31
460445	115	0,02%	26	460555	225	0,05%	43	460550	220	0,05%	31
460455	125	0,03%	29	460710	380	0,08%	39	460395	65	0,01%	35

Çizelge S.1 (Devam) : $10 \times 2 \times 5 \times 10_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
460365	35	0,01%	28	460780	450	0,10%	39	460505	175	0,04%	36
460495	165	0,04%	31	460355	25	0,01%	41	460545	215	0,05%	30
460385	55	0,01%	27	460455	125	0,03%	42	460485	155	0,03%	35
460485	155	0,03%	28	460385	55	0,01%	38	460425	95	0,02%	35
460440	110	0,02%	31	460545	215	0,05%	39	460455	125	0,03%	32
460510	180	0,04%	31	460580	250	0,05%	37	460475	145	0,03%	37
460395	65	0,01%	29	460490	160	0,03%	43	460355	25	0,01%	32
460430	100	0,02%	30	460365	35	0,01%	40	460415	85	0,02%	31
460500	170	0,04%	27	460570	240	0,05%	39	460475	145	0,03%	31
460430	100	0,02%	29	460540	210	0,05%	44	460475	145	0,03%	33
460405	75	0,02%	26	460765	435	0,09%	44	460380	50	0,01%	34
460500	170	0,04%	29	460390	60	0,01%	39	460390	60	0,01%	32
460415	85	0,02%	31	460470	140	0,03%	43	460410	80	0,02%	36
460470	140	0,03%	28	460695	365	0,08%	43	460450	120	0,03%	33
460405	75	0,02%	31	460430	100	0,02%	40	460510	180	0,04%	31
460445	115	0,02%	27	460475	145	0,03%	44	460550	220	0,05%	34
460375	45	0,01%	26	460455	125	0,03%	44	460335	5	0,00%	32
460435	105	0,02%	31	460410	80	0,02%	39	460440	110	0,02%	33
460425	95	0,02%	30	460410	80	0,02%	40	460390	60	0,01%	34
460460	130	0,03%	30	460445	115	0,02%	39	460390	60	0,01%	32
460345	15	0,00%	31	460435	105	0,02%	38	460365	35	0,01%	34
460430	100	0,02%	28	460465	135	0,03%	40	460425	95	0,02%	35
460330	0	0,04%	26	460455	125	0,03%	39	460480	150	0,03%	35
460355	25	0,01%	26	460390	60	0,01%	39	460370	40	0,01%	31
460475	145	0,03%	28	460435	105	0,02%	43	460375	45	0,01%	31
460490	160	0,03%	31	460725	395	0,09%	42	460550	220	0,05%	31
460330	0	0,00%	29	460700	370	0,08%	38	460415	85	0,02%	31
460345	15	0,00%	29	460765	435	0,09%	40	460475	145	0,03%	31
460385	55	0,01%	26	460715	385	0,08%	40	460360	30	0,01%	35
460370	40	0,01%	28	460725	395	0,09%	41	460355	25	0,01%	35
460470	140	0,03%	29	460405	75	0,02%	38	460450	120	0,03%	32
460415	85	0,02%	30	460520	190	0,04%	40	460525	195	0,04%	31
460470	140	0,03%	27	460715	385	0,08%	38	460365	35	0,01%	32
460450	120	0,03%	31	460560	230	0,05%	43	460515	185	0,04%	34
460370	40	0,01%	26	460395	65	0,01%	41	460485	155	0,03%	36
460440	110	0,02%	28	460650	320	0,07%	42	460375	45	0,01%	32
460475	145	0,03%	27	460330	0	0,00%	42	460395	65	0,01%	31
460365	35	0,01%	30	460700	370	0,08%	37	460380	50	0,01%	34
460465	135	0,03%	27	460450	120	0,03%	44	460485	155	0,03%	33
460330	0	0,00%	27	460475	145	0,03%	39	460380	50	0,01%	35
460465	135	0,03%	31	460585	255	0,06%	44	460465	135	0,03%	34
460495	165	0,04%	29	460520	190	0,04%	37	460385	55	0,01%	32

Çizelge S.1 (Devam) : $10 \times 2 \times 5 \times 10_1$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
460475	145	0,03%	27	460530	200	0,04%	39	460330	0	0,00%	32
460460	130	0,03%	26	460415	85	0,02%	43	460555	225	0,05%	32
460370	40	0,01%	28	460495	165	0,04%	44	460400	70	0,02%	33
460505	175	0,04%	27	460475	145	0,03%	37	460515	185	0,04%	35
460500	170	0,04%	26	460440	110	0,02%	41	460460	130	0,03%	31
460385	55	0,01%	28	460350	20	0,00%	37	460525	195	0,04%	34
460495	165	0,04%	30	460480	150	0,03%	39	460395	65	0,01%	36
460400	70	0,02%	27	460385	55	0,01%	40	460500	170	0,04%	36
460490	160	0,03%	31	460575	245	0,05%	37	460525	195	0,04%	33
460335	5	0,00%	26	460580	250	0,05%	39	460550	220	0,05%	33
460390	60	0,01%	29	460630	300	0,07%	40	460335	5	0,00%	34
460450	120	0,03%	27	460730	400	0,09%	44	460475	145	0,03%	33
460360	30	0,01%	27	460665	335	0,07%	42	460365	35	0,01%	32
460365	35	0,01%	30	460445	115	0,02%	37	460390	60	0,01%	37
460465	135	0,03%	30	460685	355	0,08%	43	460415	85	0,02%	37
460480	150	0,03%	31	460385	55	0,01%	43	460520	190	0,04%	31
460480	150	0,03%	27	460440	110	0,02%	37	460410	80	0,02%	35
460400	70	0,02%	30	460685	355	0,08%	39	460405	75	0,02%	31
460415	85	0,02%	27	460445	115	0,02%	43	460460	130	0,03%	32
460470	140	0,03%	30	460645	315	0,07%	39	460425	95	0,02%	31
460465	135	0,03%	26	460355	25	0,01%	41	460530	200	0,04%	31
460480	150	0,03%	29	460460	130	0,03%	38	460455	125	0,03%	31
460350	20	0,00%	26	460430	100	0,02%	39	460395	65	0,01%	31
460380	50	0,01%	27	460615	285	0,06%	44	460435	105	0,02%	33
460445	115	0,02%	30	460615	285	0,06%	41	460560	230	0,05%	34
460465	135	0,03%	29	460630	300	0,07%	41	460465	135	0,03%	30
460485	155	0,03%	30	460410	80	0,02%	38	460430	100	0,02%	35
460370	40	0,01%	28	460770	440	0,10%	40	460330	0	0,00%	33
460390	60	0,01%	29	460645	315	0,07%	42	460450	120	0,03%	35
460395	65	0,01%	30	460750	420	0,09%	43	460405	75	0,02%	31
460370	40	0,01%	31	460525	195	0,04%	40	460345	15	0,00%	37

EK T : $10 \times 2 \times 5 \times 10_2$ Problemi İçin Bulunan Sonuçlar

$10 \times 2 \times 5 \times 10_2$ problemi için bulunan optimum değer 725515 olup, Çizelge T.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge T.1 : $10 \times 2 \times 5 \times 10_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
725690	175	0,02%	26	725845	330	0,05%	38	725725	210	0,03%	36
725680	165	0,02%	27	726155	640	0,09%	40	725520	5	0,00%	31
725635	120	0,02%	27	725915	400	0,06%	42	725580	65	0,01%	33
725695	180	0,02%	30	726190	675	0,09%	40	725655	140	0,02%	33
725775	260	0,04%	30	725800	285	0,04%	42	725810	295	0,04%	33
725765	250	0,03%	31	726035	520	0,07%	40	725610	95	0,01%	34
725755	240	0,03%	30	725560	45	0,01%	40	725805	290	0,04%	31
725630	115	0,02%	29	726155	640	0,09%	42	725595	80	0,01%	34
725555	40	0,01%	29	725880	365	0,05%	38	725515	0	0,00%	35
725520	5	0,00%	30	725680	165	0,02%	40	725620	105	0,01%	31
725690	175	0,02%	30	725875	360	0,05%	37	725755	240	0,03%	36
725590	75	0,01%	31	726030	515	0,07%	39	725695	180	0,02%	36
725620	105	0,01%	30	725945	430	0,06%	44	725570	55	0,01%	35
725765	250	0,03%	28	726065	550	0,08%	42	725515	0	0,00%	34
725730	215	0,03%	31	725810	295	0,04%	44	725560	45	0,01%	30
725705	190	0,03%	27	726025	510	0,07%	40	725740	225	0,03%	34
725620	105	0,01%	30	725610	95	0,01%	43	725715	200	0,03%	34
725720	205	0,03%	28	725555	40	0,01%	41	725745	230	0,03%	31
725590	75	0,01%	26	725820	305	0,04%	42	725615	100	0,01%	36
725655	140	0,02%	27	725670	155	0,02%	38	725635	120	0,02%	34
725755	240	0,03%	28	725860	345	0,05%	40	725525	10	0,00%	35
725785	270	0,04%	27	725760	245	0,03%	41	725670	155	0,02%	31
725590	75	0,01%	27	725725	210	0,03%	38	725585	70	0,01%	35
725565	50	0,01%	28	726205	690	0,10%	44	725850	335	0,05%	36
725750	235	0,03%	27	725670	155	0,02%	44	725525	10	0,00%	31
725640	125	0,02%	27	726175	660	0,09%	40	725845	330	0,05%	32
725795	280	0,04%	31	725865	350	0,05%	40	725605	90	0,01%	35
725785	270	0,04%	29	726100	585	0,08%	42	725770	255	0,04%	35
725650	135	0,02%	29	726220	705	0,10%	39	725790	275	0,04%	31

Çizelge T.1 (Devam) : $10 \times 2 \times 5 \times 10_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
725605	90	0,01%	30	725710	195	0,03%	39	725785	270	0,04%	32
725740	225	0,03%	28	725960	445	0,06%	41	725745	230	0,03%	34
725745	230	0,03%	30	725805	290	0,04%	37	725625	110	0,02%	35
725705	190	0,03%	27	725820	305	0,04%	41	725600	85	0,01%	36
725540	25	0,00%	28	725775	260	0,04%	41	725865	350	0,05%	34
725755	240	0,03%	29	726155	640	0,09%	44	725655	140	0,02%	37
725765	250	0,03%	29	726135	620	0,09%	44	725515	0	0,00%	36
725590	75	0,01%	30	725790	275	0,04%	43	725725	210	0,03%	31
725555	40	0,01%	28	725515	0	0,00%	38	725740	225	0,03%	34
725675	160	0,02%	30	725715	200	0,03%	42	725630	115	0,02%	34
725555	40	0,01%	27	726080	565	0,08%	43	725540	25	0,00%	30
725615	100	0,01%	29	725965	450	0,06%	41	725635	120	0,02%	33
725600	85	0,01%	30	725715	200	0,03%	39	725775	260	0,04%	31
725615	100	0,01%	30	726160	645	0,09%	37	725780	265	0,04%	31
725805	290	0,04%	30	725615	100	0,01%	39	725575	60	0,01%	32
725800	285	0,04%	30	726075	560	0,08%	41	725560	45	0,01%	34
725790	275	0,04%	26	726030	515	0,07%	40	725605	90	0,01%	31
725515	0	0,00%	29	725575	60	0,01%	38	725810	295	0,04%	36
725695	180	0,02%	26	725560	45	0,01%	41	725575	60	0,01%	34
725560	45	0,01%	27	725595	80	0,01%	42	725835	320	0,04%	34
725750	235	0,03%	30	725800	285	0,04%	41	725785	270	0,04%	35
725785	270	0,04%	30	725810	295	0,04%	37	725615	100	0,01%	33
725560	45	0,01%	28	726190	675	0,09%	42	725835	320	0,04%	34
725650	135	0,02%	30	725540	25	0,00%	43	725800	285	0,04%	35
725520	5	0,00%	26	725975	460	0,06%	44	725855	340	0,05%	32
725625	110	0,02%	31	726145	630	0,09%	40	725875	360	0,05%	35
725635	120	0,02%	30	725920	405	0,06%	39	725760	245	0,03%	36
725770	255	0,04%	29	725545	30	0,00%	43	725655	140	0,02%	35
725670	155	0,02%	30	725730	215	0,03%	41	725575	60	0,01%	33
725745	230	0,03%	30	725905	390	0,05%	40	725700	185	0,03%	35
725775	260	0,04%	26	725905	390	0,05%	41	725565	50	0,01%	33
725670	155	0,02%	31	725540	25	0,00%	38	725620	105	0,01%	34
725560	45	0,01%	26	726200	685	0,09%	39	725735	220	0,03%	36
725630	115	0,02%	26	725965	450	0,06%	41	725700	185	0,03%	30
725650	135	0,02%	29	726080	565	0,08%	43	725660	145	0,02%	33
725630	115	0,02%	28	726075	560	0,08%	44	725655	140	0,02%	34
725790	275	0,04%	29	725880	365	0,05%	43	725625	110	0,02%	32
725545	30	0,00%	30	725830	315	0,04%	37	725685	170	0,02%	30
725610	95	0,01%	27	725645	130	0,02%	43	725770	255	0,04%	34
725750	235	0,03%	29	726200	685	0,09%	37	725540	25	0,00%	34
725705	190	0,03%	26	726185	670	0,09%	43	725835	320	0,04%	35
725745	230	0,03%	31	726115	600	0,08%	40	725655	140	0,02%	32

Çizelge T.1 (Devam) : $10 \times 2 \times 5 \times 10_2$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
725800	285	0,04%	29	725515	0	0,00%	38	725855	340	0,05%	31
725710	195	0,03%	27	725655	140	0,02%	42	725550	35	0,00%	35
725725	210	0,03%	28	726195	680	0,09%	37	725550	35	0,00%	30
725550	35	0,00%	29	725735	220	0,03%	43	725820	305	0,04%	33
725580	65	0,01%	30	725915	400	0,06%	37	725515	0	0,00%	36
725795	280	0,04%	27	725615	100	0,01%	41	725665	150	0,02%	36
725575	60	0,01%	27	725930	415	0,06%	38	725740	225	0,03%	36
725675	160	0,02%	26	726195	680	0,09%	44	725625	110	0,02%	31
725530	15	0,00%	31	725720	205	0,03%	43	725780	265	0,04%	35
725615	100	0,01%	26	725965	450	0,06%	42	725750	235	0,03%	34
725660	145	0,02%	31	725650	135	0,02%	38	725655	140	0,02%	37
725645	130	0,02%	31	725695	180	0,02%	44	725630	115	0,02%	30
725705	190	0,03%	29	725710	195	0,03%	43	725625	110	0,02%	32
725590	75	0,01%	31	725720	205	0,03%	41	725625	110	0,02%	34
725550	35	0,00%	27	725990	475	0,07%	42	725700	185	0,03%	37
725750	235	0,03%	29	726215	700	0,10%	37	725825	310	0,04%	32
725630	115	0,02%	31	725605	90	0,01%	37	725710	195	0,03%	36
725700	185	0,03%	31	725985	470	0,06%	44	725785	270	0,04%	32
725590	75	0,01%	30	725640	125	0,02%	37	725860	345	0,05%	36
725660	145	0,02%	28	725770	255	0,04%	43	725700	185	0,03%	33
725665	150	0,02%	31	725525	10	0,00%	37	725780	265	0,04%	35
725580	65	0,01%	30	725650	135	0,02%	44	725580	65	0,01%	36
725710	195	0,03%	27	726130	615	0,08%	39	725775	260	0,04%	33
725595	80	0,01%	29	725525	10	0,00%	44	725845	330	0,05%	32
725620	105	0,01%	30	726160	645	0,09%	41	725660	145	0,02%	32
725565	50	0,01%	28	726000	485	0,07%	42	725810	295	0,04%	32
725750	235	0,03%	29	726160	645	0,09%	40	725530	15	0,00%	36
725560	45	0,01%	28	726120	605	0,08%	43	725615	100	0,01%	36
725570	55	0,01%	27	726220	705	0,10%	44	725795	280	0,04%	34

EK U : $10 \times 2 \times 5 \times 10_3$ Problemi İçin Bulunan Sonuçlar

$10 \times 2 \times 5 \times 10_3$ problemi için bulunan optimum değer 658580 olup, Çizelge U.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge U.1 : $10 \times 2 \times 5 \times 10_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
658670	90	0,01%	29	658965	385	0,06%	40	658690	110	0,02%	31
658660	80	0,01%	30	658980	400	0,06%	43	658675	95	0,01%	33
658740	160	0,02%	29	658805	225	0,03%	37	658850	270	0,04%	34
658755	175	0,03%	30	658720	140	0,02%	43	658825	245	0,04%	31
658770	190	0,03%	31	658945	365	0,06%	37	658820	240	0,04%	34
658665	85	0,01%	28	659070	490	0,07%	38	658905	325	0,05%	36
658615	35	0,01%	30	658830	250	0,04%	38	658650	70	0,01%	34
658620	40	0,01%	26	658660	80	0,01%	42	658850	270	0,04%	32
658590	10	0,00%	27	659065	485	0,07%	37	658675	95	0,01%	31
658605	25	0,00%	28	659155	575	0,09%	39	658780	200	0,03%	32
658690	110	0,02%	28	658930	350	0,05%	37	658650	70	0,01%	32
658590	10	0,00%	28	659000	420	0,06%	43	658685	105	0,02%	35
658580	0	0,00%	29	659010	430	0,07%	42	658825	245	0,04%	36
658605	25	0,00%	27	659210	630	0,10%	37	658785	205	0,03%	32
658780	200	0,03%	28	658635	55	0,01%	42	658655	75	0,01%	35
658790	210	0,03%	27	658810	230	0,03%	38	658835	255	0,04%	33
658585	5	0,00%	27	658795	215	0,03%	37	658790	210	0,03%	34
658770	190	0,03%	29	658730	150	0,02%	42	658700	120	0,02%	31
658805	225	0,03%	29	659105	525	0,08%	44	658890	310	0,05%	34
658720	140	0,02%	30	659000	420	0,06%	37	658880	300	0,05%	35
658800	220	0,03%	26	659180	600	0,09%	40	658615	35	0,01%	33
658630	50	0,01%	27	658655	75	0,01%	38	658610	30	0,00%	34
658830	250	0,04%	30	658710	130	0,02%	39	658600	20	0,00%	33
658780	200	0,03%	29	658700	120	0,02%	37	658725	145	0,02%	34
658680	100	0,02%	26	658775	195	0,03%	44	658595	15	0,00%	33
658615	35	0,01%	29	659220	640	0,10%	41	658680	100	0,02%	34
658655	75	0,01%	27	659100	520	0,08%	43	658715	135	0,02%	32
658670	90	0,01%	29	659000	420	0,06%	42	658755	175	0,03%	35
658630	50	0,01%	31	658585	5	0,00%	38	658835	255	0,04%	33

Çizelge U.1 (Devam) : $10 \times 2 \times 5 \times 10_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
658740	160	0,02%	31	658785	205	0,03%	39	658895	315	0,05%	34
658830	250	0,04%	29	658960	380	0,06%	39	658730	150	0,02%	35
658655	75	0,01%	28	658960	380	0,06%	39	658590	10	0,00%	33
658790	210	0,03%	30	659220	640	0,10%	38	658600	20	0,00%	35
658820	240	0,04%	31	658705	125	0,02%	38	658720	140	0,02%	33
658605	25	0,00%	31	658625	45	0,01%	37	658750	170	0,03%	31
658635	55	0,01%	28	658655	75	0,01%	42	658765	185	0,03%	34
658620	40	0,01%	27	658925	345	0,05%	44	658635	55	0,01%	33
658615	35	0,01%	29	658840	260	0,04%	42	658580	0	0,00%	33
658725	145	0,02%	30	659035	455	0,07%	44	658710	130	0,02%	35
658580	0	0,00%	26	658580	0	0,00%	44	658805	225	0,03%	35
658705	125	0,02%	28	659105	525	0,08%	38	658710	130	0,02%	35
658685	105	0,02%	31	659070	490	0,07%	44	658685	105	0,02%	32
658715	135	0,02%	29	659085	505	0,08%	37	658665	85	0,01%	35
658665	85	0,01%	26	659070	490	0,07%	43	658900	320	0,05%	32
658580	0	0,00%	28	658780	200	0,03%	40	658835	255	0,04%	37
658795	215	0,03%	27	658580	0	0,00%	38	658870	290	0,04%	33
658765	185	0,03%	29	658860	280	0,04%	37	658830	250	0,04%	34
658640	60	0,01%	27	658790	210	0,03%	39	658590	10	0,00%	36
658580	0	0,00%	29	658900	320	0,05%	41	658725	145	0,02%	32
658810	230	0,03%	27	659065	485	0,07%	44	658610	30	0,00%	34
658600	20	0,00%	28	659065	485	0,07%	37	658590	10	0,00%	35
658580	0	0,00%	28	658755	175	0,03%	44	658635	55	0,01%	35
658615	35	0,01%	28	659185	605	0,09%	38	658720	140	0,02%	35
658730	150	0,02%	31	658910	330	0,05%	42	658865	285	0,04%	31
658670	90	0,01%	28	659225	645	0,10%	42	658905	325	0,05%	35
658660	80	0,01%	27	658880	300	0,05%	44	658625	45	0,01%	35
658580	0	0,00%	27	658715	135	0,02%	38	658785	205	0,03%	31
658625	45	0,01%	26	658925	345	0,05%	37	658770	190	0,03%	33
658640	60	0,01%	28	658715	135	0,02%	44	658905	325	0,05%	31
658665	85	0,01%	29	658930	350	0,05%	44	658750	170	0,03%	31
658705	125	0,02%	31	658725	145	0,02%	38	658585	5	0,00%	36
658580	0	0,00%	29	659050	470	0,07%	42	658645	65	0,01%	35
658790	210	0,03%	31	658720	140	0,02%	37	658610	30	0,00%	36
658775	195	0,03%	30	658985	405	0,06%	40	658885	305	0,05%	35
658830	250	0,04%	27	659135	555	0,08%	40	658740	160	0,02%	30
658725	145	0,02%	30	658890	310	0,05%	44	658770	190	0,03%	36
658770	190	0,03%	28	658810	230	0,03%	37	658580	0	0,00%	34
658670	90	0,01%	28	658895	315	0,05%	37	658790	210	0,03%	34
658805	225	0,03%	31	658785	205	0,03%	39	658735	155	0,02%	32
658730	150	0,02%	28	658865	285	0,04%	43	658580	0	0,00%	32
658665	85	0,01%	29	659130	550	0,08%	42	658730	150	0,02%	34

Çizelge U.1 (Devam) : $10 \times 2 \times 5 \times 10_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
658660	80	0,01%	28	658665	85	0,01%	38	658665	85	0,01%	37
658620	40	0,01%	29	659000	420	0,06%	37	658610	30	0,00%	31
658805	225	0,03%	29	658755	175	0,03%	39	658730	150	0,02%	35
658625	45	0,01%	26	658920	340	0,05%	37	658790	210	0,03%	32
658795	215	0,03%	26	658915	335	0,05%	44	658645	65	0,01%	33
658650	70	0,01%	29	658810	230	0,03%	40	658590	10	0,00%	35
658600	20	0,00%	30	659100	520	0,08%	43	658610	30	0,00%	31
658810	230	0,03%	27	658860	280	0,04%	41	658670	90	0,01%	35
658840	260	0,04%	31	658840	260	0,04%	43	658875	295	0,04%	32
658690	110	0,02%	27	658625	45	0,01%	43	658655	75	0,01%	33
658595	15	0,00%	29	658680	100	0,02%	44	658865	285	0,04%	37
658800	220	0,03%	31	659195	615	0,09%	40	658620	40	0,01%	35
658745	165	0,03%	31	658720	140	0,02%	41	658670	90	0,01%	34
658720	140	0,02%	27	658825	245	0,04%	38	658905	325	0,05%	31
658705	125	0,02%	27	658955	375	0,06%	40	658615	35	0,01%	32
658620	40	0,01%	30	659185	605	0,09%	43	658635	55	0,01%	35
658780	200	0,03%	29	658655	75	0,01%	44	658580	0	0,00%	30
658610	30	0,00%	29	658580	0	0,00%	39	658890	310	0,05%	32
658675	95	0,01%	26	658975	395	0,06%	42	658790	210	0,03%	32
658685	105	0,02%	30	658940	360	0,05%	38	658685	105	0,02%	31
658620	40	0,01%	28	658595	15	0,00%	43	658680	100	0,02%	35
658635	55	0,01%	29	658800	220	0,03%	39	658605	25	0,00%	34
658710	130	0,02%	27	658830	250	0,04%	44	658630	50	0,01%	35
658715	135	0,02%	31	658940	360	0,05%	39	658740	160	0,02%	31
658635	55	0,01%	30	658945	365	0,06%	37	658865	285	0,04%	33
658790	210	0,03%	26	658635	55	0,01%	39	658610	30	0,00%	35
658660	80	0,01%	26	658775	195	0,03%	38	658675	95	0,01%	33
658800	220	0,03%	28	659080	500	0,08%	38	658585	5	0,00%	36
658820	240	0,04%	31	658775	195	0,03%	43	658725	145	0,02%	34

EK V : 12×2×8×12₁ Problemi İçin Bulunan Sonuçlar

12×2×8×12₁ problemi için bulunan optimum değer 51930 olup, Çizelge V.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge V.1 : 12×2×8×12₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
51930	0	0,00%	61	51965	35	0,07%	1	51950	20	0,04%	59
51965	35	0,07%	62	51975	45	0,09%	2	52020	90	0,17%	63
51940	10	0,02%	62	51960	30	0,06%	3	52010	80	0,15%	62
51940	10	0,02%	56	52080	150	0,29%	4	51975	45	0,09%	67
51975	45	0,09%	61	52080	150	0,29%	5	51930	0	0,00%	63
51955	25	0,05%	52	51990	60	0,12%	6	52015	85	0,16%	70
51930	0	0,00%	56	52015	85	0,16%	7	52020	90	0,17%	68
51935	5	0,01%	62	52025	95	0,18%	8	51930	0	0,00%	71
51945	15	0,03%	57	51930	0	0,00%	9	51970	40	0,08%	66
51970	40	0,08%	62	51995	65	0,13%	10	51945	15	0,03%	60
51970	40	0,08%	52	51955	25	0,05%	11	51960	30	0,06%	59
51955	25	0,05%	63	52050	120	0,23%	12	51945	15	0,03%	60
51945	15	0,03%	57	52075	145	0,28%	13	52000	70	0,13%	70
51950	20	0,04%	62	52015	85	0,16%	14	52025	95	0,18%	71
51950	20	0,04%	53	52060	130	0,25%	15	51955	25	0,05%	69
51940	10	0,02%	53	52060	130	0,25%	16	51935	5	0,01%	64
51960	30	0,06%	55	52075	145	0,28%	17	51960	30	0,06%	60
51930	0	0,00%	62	51970	40	0,08%	18	51990	60	0,12%	59
51960	30	0,06%	55	52060	130	0,25%	19	51975	45	0,09%	59
51965	35	0,07%	59	51935	5	0,01%	20	51970	40	0,08%	66
51945	15	0,03%	57	52040	110	0,21%	21	51930	0	0,00%	67
51960	30	0,06%	58	51945	15	0,03%	22	52020	90	0,17%	61
51930	0	0,00%	58	52045	115	0,22%	23	52010	80	0,15%	67
51975	45	0,09%	54	52055	125	0,24%	24	51950	20	0,04%	61
51940	10	0,02%	58	52025	95	0,18%	25	51950	20	0,04%	66

Çizelge V.1 (Devam) : 12×2×8×12₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
51980	50	0,10%	55	51960	30	0,06%	26	51935	5	0,01%	60
51980	50	0,10%	57	51930	0	0,00%	27	51950	20	0,04%	58
51960	30	0,06%	61	52025	95	0,18%	28	51950	20	0,04%	69
51970	40	0,08%	57	51955	25	0,05%	29	51940	10	0,02%	65
51940	10	0,02%	61	52070	140	0,27%	30	52015	85	0,16%	70
51975	45	0,09%	60	52010	80	0,15%	31	52000	70	0,13%	63
51945	15	0,03%	52	52055	125	0,24%	32	51960	30	0,06%	61
51940	10	0,02%	57	52010	80	0,15%	33	51955	25	0,05%	62
51935	5	0,01%	55	52020	90	0,17%	34	51970	40	0,08%	62
51965	35	0,07%	56	51970	40	0,08%	35	51975	45	0,09%	67
51940	10	0,02%	58	51990	60	0,12%	36	51965	35	0,07%	63
51980	50	0,10%	59	52040	110	0,21%	37	52005	75	0,14%	68
51955	25	0,05%	55	52050	120	0,23%	38	51930	0	0,00%	59
51955	25	0,05%	52	51945	15	0,03%	39	51950	20	0,04%	69
51945	15	0,03%	62	51940	10	0,02%	40	52010	80	0,15%	65
51965	35	0,07%	61	52020	90	0,17%	41	51965	35	0,07%	64
51945	15	0,03%	58	52015	85	0,16%	42	52005	75	0,14%	59
51960	30	0,06%	62	51965	35	0,07%	43	52010	80	0,15%	69
51975	45	0,09%	53	52015	85	0,16%	44	51945	15	0,03%	66
51955	25	0,05%	53	52030	100	0,19%	45	52020	90	0,17%	61
51965	35	0,07%	62	52025	95	0,18%	46	52010	80	0,15%	68
51965	35	0,07%	58	51955	25	0,05%	47	51970	40	0,08%	68
51945	15	0,03%	61	52030	100	0,19%	48	52025	95	0,18%	70
51965	35	0,07%	53	51970	40	0,08%	49	52030	100	0,19%	66
51940	10	0,02%	52	51995	65	0,13%	50	51985	55	0,11%	62
51945	15	0,03%	61	52030	100	0,19%	51	51970	40	0,08%	59
51975	45	0,09%	63	51935	5	0,01%	52	51935	5	0,01%	70
51930	0	0,00%	58	52080	150	0,29%	53	52005	75	0,14%	59
51975	45	0,09%	62	51985	55	0,11%	54	51940	10	0,02%	70
51950	20	0,04%	62	52060	130	0,25%	55	51930	0	0,00%	70
51970	40	0,08%	53	52080	150	0,29%	56	51965	35	0,07%	61
51945	15	0,03%	53	51975	45	0,09%	57	51960	30	0,06%	70
51965	35	0,07%	58	51985	55	0,11%	58	52010	80	0,15%	64
51945	15	0,03%	62	51930	0	0,00%	59	51950	20	0,04%	61
51955	25	0,05%	63	51975	45	0,09%	60	51940	10	0,02%	64
51970	40	0,08%	56	51970	40	0,08%	61	51970	40	0,08%	69
51970	40	0,08%	53	52015	85	0,16%	62	51935	5	0,01%	65
51975	45	0,09%	59	52060	130	0,25%	63	51975	45	0,09%	63
51970	40	0,08%	58	51950	20	0,04%	64	51950	20	0,04%	63
51960	30	0,06%	61	51980	50	0,10%	65	51985	55	0,11%	63
51980	50	0,10%	57	52010	80	0,15%	66	51980	50	0,10%	71
51935	5	0,01%	63	51930	0	0,00%	67	52005	75	0,14%	61

Çizelge V.1 (Devam) : 12×2×8×12₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
51965	35	0,07%	61	51985	55	0,11%	68	52025	95	0,18%	65
51975	45	0,09%	62	52055	125	0,24%	69	52005	75	0,14%	68
51965	35	0,07%	60	52050	120	0,23%	70	52010	80	0,15%	65
51965	35	0,07%	57	52005	75	0,14%	71	51935	5	0,01%	70
51945	15	0,03%	53	51970	40	0,08%	72	52000	70	0,13%	64
51930	0	0,00%	61	51980	50	0,10%	73	52015	85	0,16%	59
51960	30	0,06%	52	52040	110	0,21%	74	51955	25	0,05%	65
51970	40	0,08%	52	52050	120	0,23%	75	51965	35	0,07%	63
51965	35	0,07%	60	51965	35	0,07%	76	51975	45	0,09%	58
51955	25	0,05%	56	52075	145	0,28%	77	52015	85	0,16%	59
51935	5	0,01%	53	51955	25	0,05%	78	51940	10	0,02%	65
51945	15	0,03%	63	51955	25	0,05%	79	52025	95	0,18%	67
51955	25	0,05%	54	52040	110	0,21%	80	52010	80	0,15%	60
51965	35	0,07%	59	52000	70	0,13%	81	51975	45	0,09%	68
51965	35	0,07%	60	51935	5	0,01%	82	51955	25	0,05%	64
51950	20	0,04%	57	51970	40	0,08%	83	52000	70	0,13%	60
51975	45	0,09%	61	52075	145	0,28%	84	51940	10	0,02%	61
51955	25	0,05%	63	51940	10	0,02%	85	51965	35	0,07%	59
51960	30	0,06%	53	51955	25	0,05%	86	51955	25	0,05%	63
51930	0	0,00%	52	51970	40	0,08%	87	51940	10	0,02%	71
51940	10	0,02%	52	52080	150	0,29%	88	51980	50	0,10%	60
51965	35	0,07%	58	52050	120	0,23%	89	52025	95	0,18%	68
51975	45	0,09%	57	52005	75	0,14%	90	52025	95	0,18%	67
51975	45	0,09%	55	52030	100	0,19%	91	51950	20	0,04%	65
51955	25	0,05%	57	52010	80	0,15%	92	52035	105	0,20%	60
51955	25	0,05%	63	52000	70	0,13%	93	51935	5	0,01%	60
51935	5	0,01%	56	51935	5	0,01%	94	52000	70	0,13%	61
51970	40	0,08%	54	51960	30	0,06%	95	51945	15	0,03%	61
51980	50	0,10%	54	52055	125	0,24%	96	52015	85	0,16%	59
51965	35	0,07%	57	52035	105	0,20%	97	52010	80	0,15%	62
51975	45	0,09%	53	51965	35	0,07%	98	51935	5	0,01%	64
51960	30	0,06%	62	52010	80	0,15%	99	51930	0	0,00%	67
51935	5	0,01%	55	52070	140	0,27%	100	52005	75	0,14%	62

EK W : 12×2×8×12₂ Problemi İçin Bulunan Sonuçlar

12×2×8×12₂ problemi için bulunan optimum değer 73105 olup, Çizelge W.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge W.1 : 12×2×8×12₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
73125	20	0,03%	62	73275	170	0,23%	92	73245	140	0,19%	66
73125	20	0,03%	62	73175	70	0,10%	86	73250	145	0,20%	61
73175	70	0,10%	61	73155	50	0,07%	85	73200	95	0,13%	67
73130	25	0,03%	55	73230	125	0,17%	91	73145	40	0,05%	60
73110	5	0,01%	58	73300	195	0,27%	78	73155	50	0,07%	61
73170	65	0,09%	57	73290	185	0,25%	83	73125	20	0,03%	69
73175	70	0,10%	55	73210	105	0,14%	78	73185	80	0,11%	61
73120	15	0,02%	53	73110	5	0,01%	87	73125	20	0,03%	71
73145	40	0,05%	59	73200	95	0,13%	83	73235	130	0,18%	59
73135	30	0,04%	59	73130	25	0,03%	88	73115	10	0,01%	59
73140	35	0,05%	57	73155	50	0,07%	82	73140	35	0,05%	63
73125	20	0,03%	56	73195	90	0,12%	90	73235	130	0,18%	65
73165	60	0,08%	62	73230	125	0,17%	93	73150	45	0,06%	61
73170	65	0,09%	54	73135	30	0,04%	80	73120	15	0,02%	69
73130	25	0,03%	55	73300	195	0,27%	93	73235	130	0,18%	69
73110	5	0,01%	63	73320	215	0,29%	84	73230	125	0,17%	67
73150	45	0,06%	57	73210	105	0,14%	84	73250	145	0,20%	60
73165	60	0,08%	60	73195	90	0,12%	92	73215	110	0,15%	62
73130	25	0,03%	63	73145	40	0,05%	86	73200	95	0,13%	59
73170	65	0,09%	61	73180	75	0,10%	81	73130	25	0,03%	61
73130	25	0,03%	52	73120	15	0,02%	82	73205	100	0,14%	65
73160	55	0,08%	56	73125	20	0,03%	82	73165	60	0,08%	62
73155	50	0,07%	56	73125	20	0,03%	82	73120	15	0,02%	67
73105	0	0,00%	60	73255	150	0,21%	85	73230	125	0,17%	65
73125	20	0,03%	54	73135	30	0,04%	88	73235	130	0,18%	60
73120	15	0,02%	53	73160	55	0,08%	92	73195	90	0,12%	63
73115	10	0,01%	62	73210	105	0,14%	85	73185	80	0,11%	65
73120	15	0,02%	58	73185	80	0,11%	91	73215	110	0,15%	67
73145	40	0,05%	56	73310	205	0,28%	85	73170	65	0,09%	68

Çizelge W.1 (Devam) : 12×2×8×12₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
73105	0	0,00%	58	73145	40	0,05%	92	73185	80	0,11%	62
73155	50	0,07%	57	73175	70	0,10%	78	73235	130	0,18%	65
73130	25	0,03%	62	73165	60	0,08%	89	73215	110	0,15%	62
73145	40	0,05%	55	73235	130	0,18%	90	73140	35	0,05%	69
73140	35	0,05%	58	73160	55	0,08%	85	73190	85	0,12%	69
73105	0	0,00%	60	73270	165	0,23%	79	73140	35	0,05%	64
73115	10	0,01%	62	73110	5	0,01%	94	73190	85	0,12%	69
73165	60	0,08%	60	73150	45	0,06%	93	73105	0	0,00%	71
73150	45	0,06%	55	73260	155	0,21%	85	73215	110	0,15%	65
73165	60	0,08%	58	73235	130	0,18%	79	73175	70	0,10%	64
73120	15	0,02%	55	73275	170	0,23%	86	73215	110	0,15%	65
73115	10	0,01%	56	73265	160	0,22%	81	73220	115	0,16%	62
73165	60	0,08%	53	73105	0	0,00%	85	73225	120	0,16%	70
73160	55	0,08%	52	73185	80	0,11%	81	73210	105	0,14%	67
73115	10	0,01%	55	73105	0	0,00%	91	73245	140	0,19%	60
73135	30	0,04%	52	73240	135	0,18%	80	73125	20	0,03%	66
73165	60	0,08%	58	73160	55	0,08%	92	73155	50	0,07%	60
73115	10	0,01%	55	73245	140	0,19%	79	73150	45	0,06%	65
73145	40	0,05%	57	73180	75	0,10%	83	73215	110	0,15%	68
73165	60	0,08%	57	73110	5	0,01%	79	73120	15	0,02%	69
73110	5	0,01%	61	73200	95	0,13%	93	73215	110	0,15%	58
73130	25	0,03%	60	73205	100	0,14%	80	73105	0	0,00%	64
73125	20	0,03%	62	73280	175	0,24%	91	73170	65	0,09%	65
73110	5	0,01%	55	73185	80	0,11%	85	73185	80	0,11%	63
73105	0	0,00%	62	73305	200	0,27%	80	73180	75	0,10%	61
73105	0	0,00%	59	73215	110	0,15%	77	73180	75	0,10%	66
73150	45	0,06%	62	73105	0	0,00%	85	73230	125	0,17%	61
73105	0	0,00%	55	73180	75	0,10%	83	73235	130	0,18%	59
73110	5	0,01%	63	73275	170	0,23%	86	73160	55	0,08%	68
73160	55	0,08%	57	73260	155	0,21%	78	73140	35	0,05%	68
73120	15	0,02%	62	73130	25	0,03%	82	73240	135	0,18%	71
73130	25	0,03%	62	73245	140	0,19%	90	73105	0	0,00%	63
73170	65	0,09%	54	73160	55	0,08%	78	73200	95	0,13%	66
73105	0	0,00%	52	73220	115	0,16%	93	73200	95	0,13%	69
73165	60	0,08%	60	73190	85	0,12%	83	73170	65	0,09%	69
73170	65	0,09%	58	73295	190	0,26%	94	73230	125	0,17%	70
73155	50	0,07%	63	73190	85	0,12%	87	73140	35	0,05%	62
73105	0	0,00%	53	73165	60	0,08%	80	73245	140	0,19%	66
73180	75	0,10%	53	73255	150	0,21%	81	73180	75	0,10%	65
73165	60	0,08%	56	73245	140	0,19%	78	73210	105	0,14%	71
73155	50	0,07%	53	73200	95	0,13%	82	73195	90	0,12%	65
73105	0	0,00%	53	73265	160	0,22%	88	73170	65	0,09%	65

Çizelge W.1 (Devam) : 12×2×8×12₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
73140	35	0,05%	54	73320	215	0,29%	79	73230	125	0,17%	66
73170	65	0,09%	56	73210	105	0,14%	88	73215	110	0,15%	67
73130	25	0,03%	52	73295	190	0,26%	78	73190	85	0,12%	64
73135	30	0,04%	52	73275	170	0,23%	83	73105	0	0,00%	64
73160	55	0,08%	61	73260	155	0,21%	83	73195	90	0,12%	64
73160	55	0,08%	52	73325	220	0,30%	94	73225	120	0,16%	70
73125	20	0,03%	60	73205	100	0,14%	90	73110	5	0,01%	64
73135	30	0,04%	57	73260	155	0,21%	89	73195	90	0,12%	66
73140	35	0,05%	60	73120	15	0,02%	88	73200	95	0,13%	62
73125	20	0,03%	52	73150	45	0,06%	83	73145	40	0,05%	70
73145	40	0,05%	56	73195	90	0,12%	84	73160	55	0,08%	60
73135	30	0,04%	62	73275	170	0,23%	83	73240	135	0,18%	71
73120	15	0,02%	62	73210	105	0,14%	92	73150	45	0,06%	59
73135	30	0,04%	54	73290	185	0,25%	93	73245	140	0,19%	64
73170	65	0,09%	57	73115	10	0,01%	84	73105	0	0,00%	59
73150	45	0,06%	52	73135	30	0,04%	80	73210	105	0,14%	63
73170	65	0,09%	60	73280	175	0,24%	90	73140	35	0,05%	66
73175	70	0,10%	63	73250	145	0,20%	85	73150	45	0,06%	65
73175	70	0,10%	52	73150	45	0,06%	88	73235	130	0,18%	64
73110	5	0,01%	58	73230	125	0,17%	92	73250	145	0,20%	69
73180	75	0,10%	53	73155	50	0,07%	84	73145	40	0,05%	64
73130	25	0,03%	52	73170	65	0,09%	85	73225	120	0,16%	68
73170	65	0,09%	53	73155	50	0,07%	91	73110	5	0,01%	71
73125	20	0,03%	52	73165	60	0,08%	80	73105	0	0,00%	58
73130	25	0,03%	57	73260	155	0,21%	86	73250	145	0,20%	69
73175	70	0,10%	60	73140	35	0,05%	77	73215	110	0,15%	69
73160	55	0,08%	52	73125	20	0,03%	84	73205	100	0,14%	64
73160	55	0,08%	61	73175	70	0,10%	88	73250	145	0,20%	58
73160	55	0,08%	62	73105	0	0,00%	83	73230	125	0,17%	62

EK X : 12×2×8×12₃ Problemi İçin Bulunan Sonuçlar

12×2×8×12₃ problemi için bulunan optimum değer 98140 olup, Çizelge X.1 algoritmalarla yapılan 100 denemede bulunan en iyi değerleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge X.1 : 12×2×8×12₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
98150	10	0,01%	58	98395	255	0,26%	92	98180	40	0,04%	68
98195	55	0,06%	57	98190	50	0,05%	81	98175	35	0,04%	70
98210	70	0,07%	62	98165	25	0,03%	79	98145	5	0,01%	61
98155	15	0,02%	57	98180	40	0,04%	88	98165	25	0,03%	62
98180	40	0,04%	61	98405	265	0,27%	88	98185	45	0,05%	66
98165	25	0,03%	58	98325	185	0,19%	85	98295	155	0,16%	70
98165	25	0,03%	60	98365	225	0,23%	85	98330	190	0,19%	64
98155	15	0,02%	63	98245	105	0,11%	77	98230	90	0,09%	67
98155	15	0,02%	61	98140	0	0,00%	78	98170	30	0,03%	61
98160	20	0,02%	53	98350	210	0,21%	84	98315	175	0,18%	64
98140	0	0,00%	53	98140	0	0,00%	85	98200	60	0,06%	67
98185	45	0,05%	60	98235	95	0,10%	92	98230	90	0,09%	63
98180	40	0,04%	53	98200	60	0,06%	86	98170	30	0,03%	62
98170	30	0,03%	54	98310	170	0,17%	85	98235	95	0,10%	71
98210	70	0,07%	57	98300	160	0,16%	87	98280	140	0,14%	65
98210	70	0,07%	61	98350	210	0,21%	77	98325	185	0,19%	70
98140	0	0,00%	53	98315	175	0,18%	83	98240	100	0,10%	71
98195	55	0,06%	57	98180	40	0,04%	90	98335	195	0,20%	63
98185	45	0,05%	56	98215	75	0,08%	92	98255	115	0,12%	58
98185	45	0,05%	60	98405	265	0,27%	83	98255	115	0,12%	61
98160	20	0,02%	54	98150	10	0,01%	84	98245	105	0,11%	66
98220	80	0,08%	60	98360	220	0,22%	92	98235	95	0,10%	59
98140	0	0,00%	62	98415	275	0,28%	89	98155	15	0,02%	68
98235	95	0,10%	62	98415	275	0,28%	87	98240	100	0,10%	67
98140	0	0,00%	59	98410	270	0,28%	89	98170	30	0,03%	67
98150	10	0,01%	58	98170	30	0,03%	94	98270	130	0,13%	64
98165	25	0,03%	62	98330	190	0,19%	83	98240	100	0,10%	64
98190	50	0,05%	61	98325	185	0,19%	88	98150	10	0,01%	61
98190	50	0,05%	55	98240	100	0,10%	90	98205	65	0,07%	60

Çizelge X.1 (Devam) : $12 \times 2 \times 8 \times 12_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
98215	75	0,08%	62	98400	260	0,26%	86	98140	0	0,00%	69
98140	0	0,00%	57	98220	80	0,08%	93	98305	165	0,17%	70
98170	30	0,03%	56	98140	0	0,00%	84	98165	25	0,03%	59
98145	5	0,01%	59	98190	50	0,05%	78	98245	105	0,11%	69
98220	80	0,08%	63	98230	90	0,09%	92	98210	70	0,07%	60
98185	45	0,05%	62	98200	60	0,06%	89	98300	160	0,16%	69
98145	5	0,01%	61	98295	155	0,16%	83	98140	0	0,00%	63
98185	45	0,05%	54	98355	215	0,22%	88	98320	180	0,18%	67
98145	5	0,01%	59	98340	200	0,20%	87	98190	50	0,05%	59
98185	45	0,05%	58	98210	70	0,07%	83	98155	15	0,02%	69
98200	60	0,06%	56	98250	110	0,11%	87	98255	115	0,12%	61
98210	70	0,07%	53	98175	35	0,04%	79	98200	60	0,06%	63
98210	70	0,07%	58	98425	285	0,29%	92	98275	135	0,14%	66
98150	10	0,01%	57	98355	215	0,22%	78	98140	0	0,00%	67
98160	20	0,02%	58	98355	215	0,22%	77	98270	130	0,13%	65
98225	85	0,09%	62	98420	280	0,29%	88	98265	125	0,13%	61
98170	30	0,03%	58	98375	235	0,24%	86	98265	125	0,13%	63
98140	0	0,00%	54	98235	95	0,10%	81	98220	80	0,08%	69
98205	65	0,07%	55	98350	210	0,21%	79	98250	110	0,11%	59
98140	0	0,00%	62	98140	0	0,00%	89	98260	120	0,12%	63
98190	50	0,05%	57	98165	25	0,03%	91	98235	95	0,10%	71
98210	70	0,07%	59	98320	180	0,18%	92	98310	170	0,17%	62
98230	90	0,09%	58	98275	135	0,14%	80	98270	130	0,13%	68
98185	45	0,05%	58	98170	30	0,03%	83	98160	20	0,02%	61
98235	95	0,10%	56	98320	180	0,18%	77	98190	50	0,05%	69
98170	30	0,03%	56	98285	145	0,15%	84	98305	165	0,17%	60
98230	90	0,09%	62	98210	70	0,07%	80	98195	55	0,06%	66
98175	35	0,04%	53	98375	235	0,24%	92	98150	10	0,01%	69
98190	50	0,05%	58	98425	285	0,29%	91	98220	80	0,08%	69
98220	80	0,08%	55	98175	35	0,04%	89	98170	30	0,03%	59
98185	45	0,05%	53	98270	130	0,13%	84	98330	190	0,19%	68
98195	55	0,06%	53	98390	250	0,25%	85	98145	5	0,01%	62
98195	55	0,06%	62	98330	190	0,19%	91	98270	130	0,13%	70
98165	25	0,03%	58	98405	265	0,27%	83	98170	30	0,03%	70
98195	55	0,06%	59	98175	35	0,04%	86	98265	125	0,13%	70
98175	35	0,04%	57	98370	230	0,23%	81	98275	135	0,14%	61
98160	20	0,02%	52	98235	95	0,10%	89	98200	60	0,06%	70
98155	15	0,02%	60	98280	140	0,14%	87	98275	135	0,14%	70
98145	5	0,01%	56	98355	215	0,22%	91	98330	190	0,19%	59
98210	70	0,07%	59	98265	125	0,13%	79	98220	80	0,08%	69
98210	70	0,07%	58	98400	260	0,26%	82	98270	130	0,13%	69
98155	15	0,02%	62	98300	160	0,16%	84	98310	170	0,17%	67

Çizelge X.1 (Devam) : $12 \times 2 \times 8 \times 12_3$ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
98220	80	0,08%	53	98190	50	0,05%	92	98170	30	0,03%	70
98155	15	0,02%	57	98400	260	0,26%	93	98245	105	0,11%	67
98220	80	0,08%	57	98245	105	0,11%	78	98260	120	0,12%	70
98195	55	0,06%	54	98140	0	0,00%	92	98250	110	0,11%	69
98160	20	0,02%	62	98155	15	0,02%	87	98315	175	0,18%	60
98145	5	0,01%	55	98320	180	0,18%	88	98255	115	0,12%	66
98185	45	0,05%	55	98400	260	0,26%	84	98155	15	0,02%	64
98195	55	0,06%	62	98305	165	0,17%	87	98155	15	0,02%	67
98145	5	0,01%	60	98365	225	0,23%	84	98150	10	0,01%	60
98180	40	0,04%	52	98220	80	0,08%	90	98165	25	0,03%	70
98200	60	0,06%	63	98215	75	0,08%	81	98170	30	0,03%	61
98160	20	0,02%	56	98190	50	0,05%	87	98220	80	0,08%	59
98155	15	0,02%	58	98390	250	0,25%	88	98270	130	0,13%	63
98190	50	0,05%	54	98245	105	0,11%	87	98240	100	0,10%	70
98190	50	0,05%	60	98295	155	0,16%	93	98280	140	0,14%	69
98160	20	0,02%	56	98405	265	0,27%	90	98170	30	0,03%	65
98235	95	0,10%	59	98330	190	0,19%	88	98180	40	0,04%	60
98235	95	0,10%	61	98145	5	0,01%	79	98285	145	0,15%	60
98220	80	0,08%	56	98200	60	0,06%	88	98170	30	0,03%	69
98155	15	0,02%	57	98430	290	0,30%	82	98205	65	0,07%	69
98195	55	0,06%	62	98300	160	0,16%	83	98265	125	0,13%	69
98215	75	0,08%	53	98280	140	0,14%	79	98230	90	0,09%	69
98215	75	0,08%	59	98195	55	0,06%	80	98300	160	0,16%	68
98190	50	0,05%	54	98230	90	0,09%	94	98155	15	0,02%	70
98175	35	0,04%	56	98250	110	0,11%	84	98275	135	0,14%	66
98160	20	0,02%	59	98215	75	0,08%	85	98300	160	0,16%	64
98145	5	0,01%	54	98300	160	0,16%	85	98140	0	0,00%	63
98180	40	0,04%	57	98320	180	0,18%	79	98190	50	0,05%	60
98160	20	0,02%	55	98260	120	0,12%	94	98290	150	0,15%	69

EK Y : 100×2×20×300₁ Problemi İçin Bulunan Sonuçlar

100×2×20×300₁ problemi için bulunan en iyi değer 132538 olup, Çizelge Y.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge Y.1 : 100×2×20×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
132988	450	0,34%	195	134362	1824	1,38%	367	132924	386	0,29%	235
133486	948	0,72%	209	133430	892	0,67%	384	133220	682	0,51%	223
133210	672	0,51%	211	132654	116	0,09%	393	133375	837	0,63%	228
133317	779	0,59%	210	137147	4609	3,48%	350	133891	1353	1,02%	214
133226	688	0,52%	213	134956	2418	1,82%	325	133292	754	0,57%	229
132852	314	0,24%	188	136733	4195	3,17%	365	135262	2724	2,06%	252
133258	720	0,54%	213	134107	1569	1,18%	350	133183	645	0,49%	248
132967	429	0,32%	190	133201	663	0,50%	340	135071	2533	1,91%	229
132719	181	0,14%	219	135143	2605	1,97%	379	135417	2879	2,17%	218
133071	533	0,40%	201	134436	1898	1,43%	341	132703	165	0,12%	252
132820	282	0,21%	216	134374	1836	1,39%	322	135174	2636	1,99%	232
133089	551	0,42%	213	134513	1975	1,49%	335	134031	1493	1,13%	252
132564	26	0,02%	187	133853	1315	0,99%	362	135422	2884	2,18%	239
133125	587	0,44%	190	137101	4563	3,44%	358	135604	3066	2,31%	243
133241	703	0,53%	218	136976	4438	3,35%	325	134147	1609	1,21%	228
132538	0	0,00%	204	132839	301	0,23%	362	135165	2627	1,98%	216
133819	1281	0,97%	207	133780	1242	0,94%	365	132794	256	0,19%	232
133318	780	0,59%	217	136642	4104	3,10%	345	135198	2660	2,01%	252
133560	1022	0,77%	186	135195	2657	2,00%	333	135592	3054	2,30%	215
132607	69	0,05%	199	133839	1301	0,98%	331	132754	216	0,16%	257
132734	196	0,15%	197	136672	4134	3,12%	323	134341	1803	1,36%	219
132943	405	0,31%	191	136447	3909	2,95%	350	133190	652	0,49%	255
133609	1071	0,81%	224	136760	4222	3,19%	378	135174	2636	1,99%	248
133833	1295	0,98%	210	135582	3044	2,30%	374	135278	2740	2,07%	216
133664	1126	0,85%	185	132610	72	0,05%	347	135839	3301	2,49%	227
132570	32	0,02%	185	137812	5274	3,98%	373	134688	2150	1,62%	223
133145	607	0,46%	207	136190	3652	2,76%	383	133903	1365	1,03%	218
133470	932	0,70%	203	136355	3817	2,88%	344	134967	2429	1,83%	229
133324	786	0,59%	198	134751	2213	1,67%	374	133491	953	0,72%	238

Çizelge Y.1 (Devam) : 100×2×20×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
133311	773	0,58%	196	137260	4722	3,56%	376	133963	1425	1,08%	235
133543	1005	0,76%	190	136019	3481	2,63%	381	134293	1755	1,32%	235
133624	1086	0,82%	202	134937	2399	1,81%	385	133332	794	0,60%	224
132879	341	0,26%	209	132660	122	0,09%	323	134616	2078	1,57%	234
133850	1312	0,99%	198	137494	4956	3,74%	326	134028	1490	1,12%	255
133529	991	0,75%	193	137486	4948	3,73%	330	134997	2459	1,86%	237
133754	1216	0,92%	220	134547	2009	1,52%	372	132739	201	0,15%	256
132730	192	0,14%	224	134497	1959	1,48%	374	135003	2465	1,86%	230
132593	55	0,04%	187	133163	625	0,47%	373	133776	1238	0,93%	252
133123	585	0,44%	218	134887	2349	1,77%	352	135400	2862	2,16%	228
132542	4	0,00%	199	133777	1239	0,93%	383	133121	583	0,44%	231
133021	483	0,36%	219	137597	5059	3,82%	341	133326	788	0,59%	251
133529	991	0,75%	205	134124	1586	1,20%	364	134466	1928	1,45%	219
133703	1165	0,88%	201	134254	1716	1,29%	374	135155	2617	1,97%	237
132586	48	0,04%	196	136306	3768	2,84%	386	134504	1966	1,48%	216
133254	716	0,54%	200	133703	1165	0,88%	335	133477	939	0,71%	218
132548	10	0,01%	209	132697	159	0,12%	366	134555	2017	1,52%	219
132845	307	0,23%	193	135202	2664	2,01%	377	134432	1894	1,43%	231
132688	150	0,11%	221	132987	449	0,34%	357	134297	1759	1,33%	257
132994	456	0,34%	190	133901	1363	1,03%	387	133789	1251	0,94%	260
133600	1062	0,80%	192	136187	3649	2,75%	366	132835	297	0,22%	254
133692	1154	0,87%	200	135392	2854	2,15%	387	134317	1779	1,34%	249
132823	285	0,22%	189	135138	2600	1,96%	354	134907	2369	1,79%	243
133653	1115	0,84%	211	133027	489	0,37%	336	134532	1994	1,50%	249
133087	549	0,41%	214	135388	2850	2,15%	363	132988	450	0,34%	255
133203	665	0,50%	211	136490	3952	2,98%	384	132975	437	0,33%	222
133311	773	0,58%	208	135243	2705	2,04%	375	135526	2988	2,25%	225
132780	242	0,18%	184	135854	3316	2,50%	339	133688	1150	0,87%	217
133206	668	0,50%	195	136985	4447	3,36%	341	134404	1866	1,41%	216
132956	418	0,32%	206	137092	4554	3,44%	362	135122	2584	1,95%	238
132783	245	0,18%	208	137178	4640	3,50%	388	133413	875	0,66%	239
133780	1242	0,94%	194	134606	2068	1,56%	357	135181	2643	1,99%	219
132616	78	0,06%	201	135567	3029	2,29%	372	132765	227	0,17%	230
133832	1294	0,98%	206	134632	2094	1,58%	338	133483	945	0,71%	226
132960	422	0,32%	185	135728	3190	2,41%	331	135492	2954	2,23%	215
133621	1083	0,82%	214	134430	1892	1,43%	322	132829	291	0,22%	261
133453	915	0,69%	215	133148	610	0,46%	329	135388	2850	2,15%	249
133410	872	0,66%	194	132843	305	0,23%	347	134700	2162	1,63%	255
133347	809	0,61%	223	134030	1492	1,13%	329	134695	2157	1,63%	224
132732	194	0,15%	219	132597	59	0,04%	382	132566	28	0,02%	230
133111	573	0,43%	214	133843	1305	0,98%	343	134968	2430	1,83%	248
132709	171	0,13%	220	134971	2433	1,84%	336	133036	498	0,38%	220

Çizelge Y.1 (Devam) : 100×2x20×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
133069	531	0,40%	200	136731	4193	3,16%	327	132589	51	0,04%	247
132962	424	0,32%	199	135024	2486	1,88%	393	133217	679	0,51%	255
132638	100	0,08%	198	135865	3327	2,51%	346	133063	525	0,40%	246
132974	436	0,33%	200	137657	5119	3,86%	384	133200	662	0,50%	234
132864	326	0,25%	193	135753	3215	2,43%	385	133585	1047	0,79%	239
133780	1242	0,94%	186	134021	1483	1,12%	363	134372	1834	1,38%	232
133098	560	0,42%	198	136049	3511	2,65%	374	133899	1361	1,03%	219
133784	1246	0,94%	207	133751	1213	0,92%	323	134517	1979	1,49%	232
133241	703	0,53%	206	136038	3500	2,64%	338	132545	7	0,01%	247
132656	118	0,09%	187	134299	1761	1,33%	338	134913	2375	1,79%	248
133580	1042	0,79%	191	132621	83	0,06%	387	134044	1506	1,14%	237
133319	781	0,59%	220	134460	1922	1,45%	393	133234	696	0,53%	241
132893	355	0,27%	192	136970	4432	3,34%	366	133884	1346	1,02%	245
132707	169	0,13%	184	135148	2610	1,97%	379	134194	1656	1,25%	231
133747	1209	0,91%	224	135779	3241	2,45%	382	134180	1642	1,24%	221
133147	609	0,46%	204	132815	277	0,21%	327	133891	1353	1,02%	243
133698	1160	0,88%	205	132847	309	0,23%	390	132579	41	0,03%	256
133125	587	0,44%	195	134460	1922	1,45%	370	133868	1330	1,00%	258
133850	1312	0,99%	222	135663	3125	2,36%	353	132785	247	0,19%	246
133385	847	0,64%	221	135040	2502	1,89%	357	132654	116	0,09%	251
133381	843	0,64%	200	133905	1367	1,03%	377	134827	2289	1,73%	246
133793	1255	0,95%	193	132824	286	0,22%	330	135199	2661	2,01%	252
133097	559	0,42%	218	137273	4735	3,57%	369	133071	533	0,40%	236
132561	23	0,02%	189	135333	2795	2,11%	350	134318	1780	1,34%	244
132996	458	0,35%	221	136912	4374	3,30%	380	134524	1986	1,50%	248
132656	118	0,09%	210	134865	2327	1,76%	349	132773	235	0,18%	222
133839	1301	0,98%	220	134071	1533	1,16%	348	133740	1202	0,91%	236
133385	847	0,64%	220	135119	2581	1,95%	357	133032	494	0,37%	243
133853	1315	0,99%	221	133031	493	0,37%	357	132903	365	0,28%	241

EK Z : 100×2×20×300₂ Problemi İçin Bulunan Sonuçlar

100×2×20×300₂ problemi için bulunan en iyi değer 164718 olup, Çizelge Z.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge Z.1 : 100×2×20×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
165223	505	0,31%	223	167281	2563	1,56%	335	167205	2487	1,51%	238
164772	54	0,03%	201	170567	5849	3,55%	382	168012	3294	2,00%	241
165076	358	0,22%	185	169870	5152	3,13%	344	168405	3687	2,24%	251
165066	348	0,21%	216	170137	5419	3,29%	391	165826	1108	0,67%	232
165233	515	0,31%	190	169833	5115	3,11%	393	168157	3439	2,09%	248
166246	1528	0,93%	201	169543	4825	2,93%	343	167702	2984	1,81%	215
166083	1365	0,83%	217	171185	6467	3,93%	378	164880	162	0,10%	247
166260	1542	0,94%	205	165860	1142	0,69%	340	168756	4038	2,45%	240
164810	92	0,06%	215	164767	49	0,03%	359	165184	466	0,28%	228
165835	1117	0,68%	217	168794	4076	2,47%	326	168326	3608	2,19%	233
165577	859	0,52%	216	167624	2906	1,76%	341	165833	1115	0,68%	230
165301	583	0,35%	216	167112	2394	1,45%	367	166765	2047	1,24%	230
165060	342	0,21%	206	171010	6292	3,82%	326	166562	1844	1,12%	252
165064	346	0,21%	194	165346	628	0,38%	380	168412	3694	2,24%	220
165055	337	0,20%	190	169397	4679	2,84%	323	164929	211	0,13%	249
165215	497	0,30%	195	164992	274	0,17%	356	167552	2834	1,72%	227
166289	1571	0,95%	208	167045	2327	1,41%	323	165091	373	0,23%	214
166143	1425	0,87%	192	168641	3923	2,38%	328	167135	2417	1,47%	221
165729	1011	0,61%	186	167898	3180	1,93%	361	168269	3551	2,16%	260
165571	853	0,52%	224	169334	4616	2,80%	389	165722	1004	0,61%	248
164718	0	0,00%	201	168136	3418	2,08%	364	166947	2229	1,35%	253
166347	1629	0,99%	206	164756	38	0,02%	351	165972	1254	0,76%	216
165697	979	0,59%	197	169873	5155	3,13%	381	167742	3024	1,84%	224
165633	915	0,56%	206	168031	3313	2,01%	386	167024	2306	1,40%	253
164985	267	0,16%	203	168733	4015	2,44%	324	168591	3873	2,35%	260
164852	134	0,08%	204	169790	5072	3,08%	379	166959	2241	1,36%	252
166228	1510	0,92%	199	167588	2870	1,74%	331	166499	1781	1,08%	222
165053	335	0,20%	213	169676	4958	3,01%	368	167031	2313	1,40%	228
165371	653	0,40%	223	165569	851	0,52%	352	167755	3037	1,84%	247

Çizelge Z.1 (Devam) : 100×2×20×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
165298	580	0,35%	220	167819	3101	1,88%	358	167442	2724	1,65%	241
165221	503	0,31%	190	169490	4772	2,90%	389	168517	3799	2,31%	234
165592	874	0,53%	205	166472	1754	1,06%	370	165201	483	0,29%	245
166301	1583	0,96%	208	169225	4507	2,74%	373	166609	1891	1,15%	260
165758	1040	0,63%	214	167637	2919	1,77%	325	168012	3294	2,00%	227
165699	981	0,60%	204	166640	1922	1,17%	372	167552	2834	1,72%	237
165464	746	0,45%	204	165902	1184	0,72%	372	167818	3100	1,88%	227
166307	1589	0,96%	197	170936	6218	3,77%	340	166818	2100	1,27%	229
165613	895	0,54%	220	169445	4727	2,87%	385	167284	2566	1,56%	240
165550	832	0,51%	211	167476	2758	1,67%	381	168079	3361	2,04%	241
165734	1016	0,62%	208	170958	6240	3,79%	367	166752	2034	1,23%	258
165579	861	0,52%	194	167950	3232	1,96%	347	166921	2203	1,34%	228
165447	729	0,44%	225	171126	6408	3,89%	384	167994	3276	1,99%	252
165350	632	0,38%	220	168225	3507	2,13%	375	165653	935	0,57%	247
165351	633	0,38%	220	166648	1930	1,17%	384	166079	1361	0,83%	254
165926	1208	0,73%	198	166729	2011	1,22%	359	164886	168	0,10%	239
165030	312	0,19%	196	167475	2757	1,67%	358	168527	3809	2,31%	242
165878	1160	0,70%	199	166603	1885	1,14%	354	166330	1612	0,98%	218
164993	275	0,17%	210	166928	2210	1,34%	379	168117	3399	2,06%	231
165582	864	0,52%	188	168143	3425	2,08%	388	166427	1709	1,04%	247
165391	673	0,41%	191	168814	4096	2,49%	369	164945	227	0,14%	218
165595	877	0,53%	198	170986	6268	3,81%	367	167553	2835	1,72%	234
164875	157	0,10%	195	164862	144	0,09%	392	166674	1956	1,19%	226
165297	579	0,35%	214	169048	4330	2,63%	337	168229	3511	2,13%	217
165984	1266	0,77%	185	171033	6315	3,83%	360	165120	402	0,24%	251
165718	1000	0,61%	203	170889	6171	3,75%	386	165352	634	0,38%	246
165053	335	0,20%	217	171176	6458	3,92%	351	167953	3235	1,96%	255
165361	643	0,39%	193	165224	506	0,31%	359	164944	226	0,14%	217
165322	604	0,37%	193	165463	745	0,45%	368	167561	2843	1,73%	236
164906	188	0,11%	216	171221	6503	3,95%	330	165771	1053	0,64%	250
165302	584	0,35%	223	165603	885	0,54%	335	165662	944	0,57%	257
166176	1458	0,89%	197	170496	5778	3,51%	367	166674	1956	1,19%	258
166111	1393	0,85%	191	170148	5430	3,30%	376	166808	2090	1,27%	255
165319	601	0,36%	217	169084	4366	2,65%	369	165753	1035	0,63%	231
166332	1614	0,98%	197	166004	1286	0,78%	368	167039	2321	1,41%	217
165676	958	0,58%	191	170609	5891	3,58%	380	167375	2657	1,61%	219
166338	1620	0,98%	224	169078	4360	2,65%	367	167183	2465	1,50%	218
166226	1508	0,92%	219	170293	5575	3,38%	355	164736	18	0,01%	218
165584	866	0,53%	200	166414	1696	1,03%	349	167378	2660	1,61%	214
164955	237	0,14%	203	170519	5801	3,52%	334	168370	3652	2,22%	224
165615	897	0,54%	224	170307	5589	3,39%	351	166772	2054	1,25%	246
165029	311	0,19%	190	167142	2424	1,47%	381	164944	226	0,14%	246

Çizelge Z.1 (Devam) : 100×2×20×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
165021	303	0,18%	201	168470	3752	2,28%	385	168710	3992	2,42%	220
166060	1342	0,81%	192	168140	3422	2,08%	367	166204	1486	0,90%	219
165872	1154	0,70%	185	169536	4818	2,92%	390	166593	1875	1,14%	252
166113	1395	0,85%	212	166579	1861	1,13%	354	165000	282	0,17%	257
166120	1402	0,85%	184	170117	5399	3,28%	341	168797	4079	2,48%	257
166267	1549	0,94%	205	166312	1594	0,97%	325	166383	1665	1,01%	254
166085	1367	0,83%	204	168295	3577	2,17%	390	165022	304	0,18%	236
165493	775	0,47%	204	167676	2958	1,80%	367	168340	3622	2,20%	242
164916	198	0,12%	198	170296	5578	3,39%	338	166968	2250	1,37%	230
165877	1159	0,70%	191	169999	5281	3,21%	379	168082	3364	2,04%	228
166003	1285	0,78%	210	171294	6576	3,99%	339	168119	3401	2,06%	226
165203	485	0,29%	222	167321	2603	1,58%	393	165596	878	0,53%	261
165174	456	0,28%	188	165935	1217	0,74%	329	168306	3588	2,18%	260
164810	92	0,06%	211	167734	3016	1,83%	383	168013	3295	2,00%	226
165899	1181	0,72%	189	166600	1882	1,14%	372	165285	567	0,34%	233
165930	1212	0,74%	207	168531	3813	2,31%	366	168670	3952	2,40%	225
164756	38	0,02%	197	166834	2116	1,28%	392	166438	1720	1,04%	224
165675	957	0,58%	215	170436	5718	3,47%	328	165936	1218	0,74%	239
165410	692	0,42%	199	171168	6450	3,92%	388	166847	2129	1,29%	260
165207	489	0,30%	201	170207	5489	3,33%	329	165103	385	0,23%	253
165430	712	0,43%	196	169737	5019	3,05%	388	166978	2260	1,37%	221
164907	189	0,11%	223	167436	2718	1,65%	325	167837	3119	1,89%	253
165810	1092	0,66%	201	167317	2599	1,58%	373	166504	1786	1,08%	214
164958	240	0,15%	193	168351	3633	2,21%	330	167911	3193	1,94%	216
165221	503	0,31%	206	168504	3786	2,30%	378	168693	3975	2,41%	248
165963	1245	0,76%	209	167363	2645	1,61%	338	167247	2529	1,54%	257
165914	1196	0,73%	190	165701	983	0,60%	349	166041	1323	0,80%	234
165697	979	0,59%	218	168050	3332	2,02%	358	166422	1704	1,03%	258
165270	552	0,34%	216	170151	5433	3,30%	358	165608	890	0,54%	239

EK AA : 100×2×20×300₃ Problemi İçin Bulunan Sonuçlar

100×2×20×300₃ problemi için bulunan en iyi değer 183052 olup, Çizelge AA.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge AA.1 : 100×2×20×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
183589	537	0,29%	190	188583	5531	3,02%	360	186309	3257	1,78%	240
184120	1068	0,58%	222	187432	4380	2,39%	376	186247	3195	1,75%	233
184008	956	0,52%	212	184860	1808	0,99%	374	183698	646	0,35%	248
183434	382	0,21%	214	189631	6579	3,59%	335	183609	557	0,30%	234
183823	771	0,42%	186	185544	2492	1,36%	324	185308	2256	1,23%	261
183838	786	0,43%	205	189118	6066	3,31%	373	183423	371	0,20%	246
184594	1542	0,84%	218	186657	3605	1,97%	356	183260	208	0,11%	249
184708	1656	0,90%	199	184830	1778	0,97%	336	183844	792	0,43%	238
184355	1303	0,71%	207	189528	6476	3,54%	353	183579	527	0,29%	244
183467	415	0,23%	223	185539	2487	1,36%	325	187193	4141	2,26%	223
184721	1669	0,91%	225	187785	4733	2,59%	353	185687	2635	1,44%	236
183864	812	0,44%	206	187167	4115	2,25%	365	184780	1728	0,94%	246
184717	1665	0,91%	213	188523	5471	2,99%	390	184408	1356	0,74%	243
184224	1172	0,64%	193	184134	1082	0,59%	344	184360	1308	0,71%	240
183379	327	0,18%	185	184893	1841	1,01%	343	185689	2637	1,44%	227
183052	0	0,00%	222	183429	377	0,21%	329	187401	4349	2,38%	222
183686	634	0,35%	188	190293	7241	3,96%	353	183507	455	0,25%	249
183575	523	0,29%	186	187973	4921	2,69%	388	184044	992	0,54%	254
183801	749	0,41%	193	183667	615	0,34%	341	184930	1878	1,03%	219
184128	1076	0,59%	218	188956	5904	3,23%	364	184937	1885	1,03%	248
183322	270	0,15%	207	187540	4488	2,45%	344	183763	711	0,39%	215
184479	1427	0,78%	214	189816	6764	3,70%	325	184537	1485	0,81%	243
184529	1477	0,81%	185	189451	6399	3,50%	338	186153	3101	1,69%	258
183349	297	0,16%	191	189959	6907	3,77%	343	185573	2521	1,38%	253
183590	538	0,29%	219	184401	1349	0,74%	393	186745	3693	2,02%	218
184740	1688	0,92%	198	188824	5772	3,15%	348	184248	1196	0,65%	214
184293	1241	0,68%	219	190125	7073	3,86%	372	186452	3400	1,86%	255
184784	1732	0,95%	217	185309	2257	1,23%	353	184685	1633	0,89%	219
183662	610	0,33%	209	184782	1730	0,95%	335	185072	2020	1,10%	219

Çizelge AA.1 (Devam) : 100×2×20×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
183808	756	0,41%	207	190080	7028	3,84%	393	184859	1807	0,99%	235
184040	988	0,54%	200	185784	2732	1,49%	342	184504	1452	0,79%	238
184293	1241	0,68%	197	186142	3090	1,69%	364	185583	2531	1,38%	250
183658	606	0,33%	207	188627	5575	3,05%	342	185798	2746	1,50%	254
184281	1229	0,67%	199	188456	5404	2,95%	366	184742	1690	0,92%	231
184738	1686	0,92%	191	188096	5044	2,76%	342	186924	3872	2,12%	260
183835	783	0,43%	217	190167	7115	3,89%	324	187242	4190	2,29%	254
184736	1684	0,92%	218	183815	763	0,42%	376	184375	1323	0,72%	220
184212	1160	0,63%	197	189307	6255	3,42%	350	186257	3205	1,75%	258
183676	624	0,34%	202	186420	3368	1,84%	361	185415	2363	1,29%	243
183452	400	0,22%	212	189181	6129	3,35%	345	183597	545	0,30%	235
183722	670	0,37%	185	190079	7027	3,84%	388	184040	988	0,54%	259
184696	1644	0,90%	192	183088	36	0,02%	369	187485	4433	2,42%	219
184333	1281	0,70%	186	189891	6839	3,74%	328	183868	816	0,45%	222
184810	1758	0,96%	220	188334	5282	2,89%	336	184872	1820	0,99%	244
183807	755	0,41%	197	188308	5256	2,87%	330	184068	1016	0,56%	255
183461	409	0,22%	205	187997	4945	2,70%	384	184564	1512	0,83%	251
184311	1259	0,69%	200	187433	4381	2,39%	365	183427	375	0,20%	219
183670	618	0,34%	219	183242	190	0,10%	346	186713	3661	2,00%	223
184611	1559	0,85%	205	188567	5515	3,01%	368	185478	2426	1,33%	225
183373	321	0,18%	194	185135	2083	1,14%	332	183202	150	0,08%	241
184235	1183	0,65%	217	188029	4977	2,72%	355	187265	4213	2,30%	236
184608	1556	0,85%	224	183275	223	0,12%	329	187542	4490	2,45%	226
183585	533	0,29%	192	186579	3527	1,93%	356	187262	4210	2,30%	244
183865	813	0,44%	203	188572	5520	3,02%	358	187435	4383	2,39%	225
184263	1211	0,66%	216	189704	6652	3,63%	347	184448	1396	0,76%	215
183852	800	0,44%	218	183249	197	0,11%	376	187270	4218	2,30%	238
183058	6	0,00%	216	185393	2341	1,28%	330	187182	4130	2,26%	224
184598	1546	0,84%	209	189163	6111	3,34%	368	183601	549	0,30%	232
184274	1222	0,67%	188	184091	1039	0,57%	324	187574	4522	2,47%	227
184102	1050	0,57%	219	189622	6570	3,59%	327	183640	588	0,32%	228
184071	1019	0,56%	204	190237	7185	3,93%	382	184288	1236	0,68%	255
184880	1828	1,00%	223	183470	418	0,23%	327	187484	4432	2,42%	243
184162	1110	0,61%	200	187874	4822	2,63%	363	183516	464	0,25%	241
183124	72	0,04%	218	185823	2771	1,51%	380	184431	1379	0,75%	217
184554	1502	0,82%	219	188908	5856	3,20%	348	186347	3295	1,80%	226
183737	685	0,37%	212	183249	197	0,11%	390	184136	1084	0,59%	229
184731	1679	0,92%	200	185958	2906	1,59%	327	186326	3274	1,79%	228
183989	937	0,51%	201	186446	3394	1,85%	344	184361	1309	0,72%	245
184710	1658	0,91%	201	186964	3912	2,14%	358	187450	4398	2,40%	259
183172	120	0,07%	201	187176	4124	2,25%	355	185768	2716	1,48%	261
184723	1671	0,91%	185	188062	5010	2,74%	354	186946	3894	2,13%	245

Çizelge AA.1 (Devam) : 100×2×20×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
183726	674	0,37%	204	190331	7279	3,98%	388	183770	718	0,39%	238
183187	135	0,07%	224	190046	6994	3,82%	352	187522	4470	2,44%	215
184208	1156	0,63%	199	187008	3956	2,16%	368	186775	3723	2,03%	251
184429	1377	0,75%	207	187575	4523	2,47%	358	187199	4147	2,27%	246
183285	233	0,13%	211	190097	7045	3,85%	364	184000	948	0,52%	224
184273	1221	0,67%	202	185106	2054	1,12%	332	183955	903	0,49%	228
183829	777	0,42%	185	185885	2833	1,55%	361	183978	926	0,51%	214
183295	243	0,13%	191	187960	4908	2,68%	342	184627	1575	0,86%	216
184759	1707	0,93%	202	190291	7239	3,95%	344	183849	797	0,44%	248
184760	1708	0,93%	209	184755	1703	0,93%	382	185681	2629	1,44%	256
184782	1730	0,95%	188	184928	1876	1,02%	373	183839	787	0,43%	243
184325	1273	0,70%	206	189866	6814	3,72%	381	187382	4330	2,37%	240
184538	1486	0,81%	220	188517	5465	2,99%	376	183129	77	0,04%	239
184162	1110	0,61%	211	183866	814	0,44%	371	183689	637	0,35%	261
184007	955	0,52%	210	185956	2904	1,59%	371	186538	3486	1,90%	237
183243	191	0,10%	199	187736	4684	2,56%	375	184608	1556	0,85%	216
184368	1316	0,72%	204	184606	1554	0,85%	383	186575	3523	1,92%	224
183598	546	0,30%	218	188502	5450	2,98%	392	183435	383	0,21%	249
183701	649	0,35%	184	187989	4937	2,70%	335	186106	3054	1,67%	223
184418	1366	0,75%	220	188958	5906	3,23%	326	183615	563	0,31%	215
184039	987	0,54%	221	188738	5686	3,11%	362	183350	298	0,16%	227
184748	1696	0,93%	186	186280	3228	1,76%	350	184749	1697	0,93%	245
184702	1650	0,90%	214	188441	5389	2,94%	326	187402	4350	2,38%	215
184844	1792	0,98%	210	189144	6092	3,33%	389	183867	815	0,45%	224
183786	734	0,40%	214	187045	3993	2,18%	334	183216	164	0,09%	260
183665	613	0,33%	192	190227	7175	3,92%	365	183473	421	0,23%	236
183154	102	0,06%	221	187410	4358	2,38%	349	183486	434	0,24%	227
184335	1283	0,70%	193	188254	5202	2,84%	349	185304	2252	1,23%	250
184481	1429	0,78%	216	186642	3590	1,96%	392	186949	3897	2,13%	224

EK BB : 1000×60×50×300₁ Problemi İçin Bulunan Sonuçlar

1000×60×50×300₁ problemi için bulunan en iyi değer 195496 olup, Çizelge BB.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge BB.1 : 1000×60×50×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
198769	3273	1,67%	323	208823	13327	6,82%	622	199339	3843	1,97%	393
196229	733	0,37%	315	200255	4759	2,43%	608	198482	2986	1,53%	363
199271	3775	1,93%	316	195760	264	0,14%	631	197519	2023	1,03%	403
196729	1233	0,63%	337	200695	5199	2,66%	635	202744	7248	3,71%	371
196317	821	0,42%	317	197393	1897	0,97%	638	202581	7085	3,62%	413
198264	2768	1,42%	322	202174	6678	3,42%	617	198955	3459	1,77%	373
196756	1260	0,64%	313	203916	8420	4,31%	577	199650	4154	2,12%	415
195920	424	0,22%	322	206900	11404	5,83%	584	196630	1134	0,58%	371
198670	3174	1,62%	361	195603	107	0,05%	618	202401	6905	3,53%	407
197097	1601	0,82%	343	197621	2125	1,09%	550	199390	3894	1,99%	366
195536	40	0,02%	345	205102	9606	4,91%	588	197981	2485	1,27%	391
195815	319	0,16%	353	206433	10937	5,59%	543	198545	3049	1,56%	394
199011	3515	1,80%	348	195808	312	0,16%	551	195955	459	0,23%	418
196063	567	0,29%	315	199638	4142	2,12%	576	203134	7638	3,91%	369
197840	2344	1,20%	363	207440	11944	6,11%	545	200121	4625	2,37%	388
198243	2747	1,41%	315	199870	4374	2,24%	639	195733	237	0,12%	357
196682	1186	0,61%	329	205926	10430	5,34%	633	198282	2786	1,43%	414
195496	0	0,00%	351	195938	442	0,23%	570	202800	7304	3,74%	423
197926	2430	1,24%	343	197619	2123	1,09%	630	198911	3415	1,75%	372
197481	1985	1,02%	311	196989	1493	0,76%	612	199935	4439	2,27%	418
199093	3597	1,84%	367	206320	10824	5,54%	602	196699	1203	0,62%	382
197961	2465	1,26%	330	207413	11917	6,10%	553	202755	7259	3,71%	424
196788	1292	0,66%	363	205367	9871	5,05%	575	196759	1263	0,65%	414
197690	2194	1,12%	345	206376	10880	5,57%	561	202936	7440	3,81%	373
197908	2412	1,23%	339	196339	843	0,43%	534	195592	96	0,05%	395
196087	591	0,30%	325	199612	4116	2,11%	644	199045	3549	1,82%	425
196633	1137	0,58%	358	197339	1843	0,94%	558	199049	3553	1,82%	399
196410	914	0,47%	310	202293	6797	3,48%	632	198465	2969	1,52%	369
197692	2196	1,12%	306	200276	4780	2,45%	557	201976	6480	3,31%	392

Çizelge BB.1 (Devam) : 1000×60×50×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
199386	3890	1,99%	356	204368	8872	4,54%	605	196514	1018	0,52%	425
195514	18	0,01%	341	208804	13308	6,81%	567	197633	2137	1,09%	389
195596	100	0,05%	349	197468	1972	1,01%	578	202998	7502	3,84%	399
198109	2613	1,34%	347	197601	2105	1,08%	538	202883	7387	3,78%	426
197900	2404	1,23%	323	204360	8864	4,53%	594	200677	5181	2,65%	434
195501	5	0,00%	348	198112	2616	1,34%	593	201083	5587	2,86%	424
198666	3170	1,62%	365	207791	12295	6,29%	597	199309	3813	1,95%	421
196855	1359	0,70%	307	202323	6827	3,49%	592	202896	7400	3,79%	411
199004	3508	1,79%	347	196666	1170	0,60%	586	197623	2127	1,09%	361
197598	2102	1,08%	359	203579	8083	4,13%	643	198670	3174	1,62%	367
198091	2595	1,33%	326	199970	4474	2,29%	582	197266	1770	0,91%	434
196749	1253	0,64%	342	196109	613	0,31%	569	199613	4117	2,11%	397
197355	1859	0,95%	345	197394	1898	0,97%	553	197870	2374	1,21%	380
198339	2843	1,45%	347	198598	3102	1,59%	603	196160	664	0,34%	411
198861	3365	1,72%	315	197561	2065	1,06%	538	199200	3704	1,89%	432
198653	3157	1,61%	359	198175	2679	1,37%	612	198310	2814	1,44%	426
198723	3227	1,65%	347	197522	2026	1,04%	617	200664	5168	2,64%	394
197379	1883	0,96%	355	197230	1734	0,89%	552	200080	4584	2,34%	393
199113	3617	1,85%	316	203559	8063	4,12%	538	198026	2530	1,29%	424
198917	3421	1,75%	337	201931	6435	3,29%	635	198553	3057	1,56%	405
197411	1915	0,98%	321	203823	8327	4,26%	539	200055	4559	2,33%	391
197784	2288	1,17%	350	200200	4704	2,41%	582	198291	2795	1,43%	407
195984	488	0,25%	342	197317	1821	0,93%	585	199261	3765	1,93%	360
197179	1683	0,86%	368	206687	11191	5,72%	536	203193	7697	3,94%	359
196632	1136	0,58%	317	207249	11753	6,01%	548	197901	2405	1,23%	356
199329	3833	1,96%	328	201643	6147	3,14%	529	199419	3923	2,01%	394
197970	2474	1,27%	327	198519	3023	1,55%	644	201266	5770	2,95%	369
198552	3056	1,56%	352	199678	4182	2,14%	631	198918	3422	1,75%	373
199201	3705	1,90%	368	203752	8256	4,22%	568	199863	4367	2,23%	409
199392	3896	1,99%	306	197101	1605	0,82%	547	201094	5598	2,86%	403
195959	463	0,24%	370	197731	2235	1,14%	565	200256	4760	2,43%	362
196794	1298	0,66%	347	206645	11149	5,70%	635	199868	4372	2,24%	426
198173	2677	1,37%	350	196797	1301	0,67%	548	199660	4164	2,13%	409
198904	3408	1,74%	335	200156	4660	2,38%	616	197987	2491	1,27%	414
197487	1991	1,02%	370	203961	8465	4,33%	577	199077	3581	1,83%	383
195575	79	0,04%	369	207836	12340	6,31%	560	202172	6676	3,41%	413
196531	1035	0,53%	338	198206	2710	1,39%	602	201107	5611	2,87%	368
198316	2820	1,44%	337	197959	2463	1,26%	579	196102	606	0,31%	366
197158	1662	0,85%	371	207071	11575	5,92%	627	198260	2764	1,41%	364
198322	2826	1,45%	319	207497	12001	6,14%	550	202274	6778	3,47%	405
197259	1763	0,90%	362	202038	6542	3,35%	644	200944	5448	2,79%	359
197184	1688	0,86%	340	208398	12902	6,60%	572	200425	4929	2,52%	371

Çizelge BB.1 (Devam) : 1000×60×50×300₁ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
197004	1508	0,77%	350	209155	13659	6,99%	566	202291	6795	3,48%	417
198309	2813	1,44%	364	207289	11793	6,03%	640	201715	6219	3,18%	405
198071	2575	1,32%	362	203413	7917	4,05%	596	200354	4858	2,48%	430
198920	3424	1,75%	364	206141	10645	5,45%	591	199755	4259	2,18%	419
196199	703	0,36%	347	199219	3723	1,90%	587	199298	3802	1,94%	411
196634	1138	0,58%	316	205315	9819	5,02%	614	195659	163	0,08%	400
198066	2570	1,31%	352	200258	4762	2,44%	538	196969	1473	0,75%	383
198084	2588	1,32%	350	199335	3839	1,96%	588	200501	5005	2,56%	358
197712	2216	1,13%	354	207258	11762	6,02%	557	198665	3169	1,62%	403
197354	1858	0,95%	320	196897	1401	0,72%	554	202627	7131	3,65%	361
199087	3591	1,84%	361	203566	8070	4,13%	552	199722	4226	2,16%	406
197192	1696	0,87%	338	207168	11672	5,97%	632	198179	2683	1,37%	428
198958	3462	1,77%	316	197526	2030	1,04%	546	197937	2441	1,25%	384
196460	964	0,49%	320	203187	7691	3,93%	559	196466	970	0,50%	393
195508	12	0,01%	347	207773	12277	6,28%	570	197971	2475	1,27%	378
198587	3091	1,58%	371	207535	12039	6,16%	578	198228	2732	1,40%	377
196961	1465	0,75%	345	201328	5832	2,98%	528	201368	5872	3,00%	427
198412	2916	1,49%	311	203638	8142	4,16%	577	197836	2340	1,20%	383
198885	3389	1,73%	329	197449	1953	1,00%	630	195660	164	0,08%	363
199150	3654	1,87%	313	199036	3540	1,81%	543	202878	7382	3,78%	403
198452	2956	1,51%	315	204201	8705	4,45%	579	202920	7424	3,80%	379
199351	3855	1,97%	366	202116	6620	3,39%	637	198579	3083	1,58%	399
196546	1050	0,54%	371	208647	13151	6,73%	612	195950	454	0,23%	387
196782	1286	0,66%	337	198162	2666	1,36%	582	198714	3218	1,65%	363
198180	2684	1,37%	332	202654	7158	3,66%	548	199105	3609	1,85%	386
198577	3081	1,58%	369	202550	7054	3,61%	540	195579	83	0,04%	426
195924	428	0,22%	360	202946	7450	3,81%	588	199938	4442	2,27%	395
198422	2926	1,50%	356	206045	10549	5,40%	577	197572	2076	1,06%	367
199386	3890	1,99%	306	208090	12594	6,44%	550	196041	545	0,28%	377

EK CC : 1000×60×50×300₂ Problemi İçin Bulunan Sonuçlar

1000×60×50×300₂ problemi için bulunan en iyi değer 170814 olup, Çizelge CC.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge CC.1 : 1000×60×50×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
174096	3282	1,92%	312	177116	6302	3,69%	565	176989	6175	3,62%	357
172916	2102	1,23%	329	173298	2484	1,45%	569	175950	5136	3,01%	364
172870	2056	1,20%	289	172948	2134	1,25%	519	171492	678	0,40%	349
173676	2862	1,68%	327	178899	8085	4,73%	542	175135	4321	2,53%	373
173364	2550	1,49%	316	175192	4378	2,56%	618	175096	4282	2,51%	372
173907	3093	1,81%	316	171760	946	0,55%	589	174772	3958	2,32%	360
171669	855	0,50%	318	182370	11556	6,77%	520	176777	5963	3,49%	361
172055	1241	0,73%	330	177378	6564	3,84%	582	172042	1228	0,72%	352
173404	2590	1,52%	317	177764	6950	4,07%	571	177476	6662	3,90%	391
173346	2532	1,48%	307	173128	2314	1,35%	584	174764	3950	2,31%	365
170814	0	0,00%	314	180592	9778	5,72%	627	172257	1443	0,84%	384
172315	1501	0,88%	335	179289	8475	4,96%	549	174696	3882	2,27%	395
172357	1543	0,90%	306	179357	8543	5,00%	525	173907	3093	1,81%	365
170947	133	0,08%	323	176234	5420	3,17%	601	174807	3993	2,34%	370
172781	1967	1,15%	290	177871	7057	4,13%	614	175047	4233	2,48%	396
172854	2040	1,19%	309	176550	5736	3,36%	581	172431	1617	0,95%	399
172778	1964	1,15%	306	173331	2517	1,47%	549	174666	3852	2,26%	386
172241	1427	0,84%	297	179108	8294	4,86%	609	171579	765	0,45%	335
172445	1631	0,95%	289	176425	5611	3,28%	588	171232	418	0,24%	391
171207	393	0,23%	299	178501	7687	4,50%	611	172144	1330	0,78%	349
174057	3243	1,90%	291	180479	9665	5,66%	544	171373	559	0,33%	398
173795	2981	1,75%	296	173453	2639	1,54%	581	175909	5095	2,98%	389
170908	94	0,06%	341	181926	11112	6,51%	590	174123	3309	1,94%	348
173172	2358	1,38%	320	182655	11841	6,93%	573	175714	4900	2,87%	368
171234	420	0,25%	311	176607	5793	3,39%	574	176543	5729	3,35%	344
170963	149	0,09%	341	174286	3472	2,03%	555	172551	1737	1,02%	366
171805	991	0,58%	311	179464	8650	5,06%	523	173403	2589	1,52%	348
174000	3186	1,87%	319	171746	932	0,55%	571	176922	6108	3,58%	347
171573	759	0,44%	308	178177	7363	4,31%	570	176603	5789	3,39%	386

Çizelge CC.1 (Devam) : 1000×60×50×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
170814	0	0,00%	290	175908	5094	2,98%	531	173221	2407	1,41%	341
171305	491	0,29%	345	178106	7292	4,27%	517	173599	2785	1,63%	336
172629	1815	1,06%	293	182324	11510	6,74%	552	173787	2973	1,74%	371
173872	3058	1,79%	290	182299	11485	6,72%	601	173959	3145	1,84%	395
173694	2880	1,69%	300	178427	7613	4,46%	529	177012	6198	3,63%	336
173480	2666	1,56%	300	179492	8678	5,08%	552	173570	2756	1,61%	330
171687	873	0,51%	331	171477	663	0,39%	584	171981	1167	0,68%	397
171524	710	0,42%	300	176265	5451	3,19%	518	176181	5367	3,14%	397
172692	1878	1,10%	291	178565	7751	4,54%	610	172841	2027	1,19%	385
173524	2710	1,59%	290	176324	5510	3,23%	625	173291	2477	1,45%	390
173603	2789	1,63%	308	171275	461	0,27%	535	176808	5994	3,51%	364
173678	2864	1,68%	318	173900	3086	1,81%	564	172969	2155	1,26%	338
174100	3286	1,92%	322	174911	4097	2,40%	618	176102	5288	3,10%	379
173948	3134	1,83%	347	179377	8563	5,01%	607	172036	1222	0,72%	388
172092	1278	0,75%	287	173244	2430	1,42%	534	176481	5667	3,32%	349
172595	1781	1,04%	312	172762	1948	1,14%	547	175330	4516	2,64%	338
174088	3274	1,92%	337	179895	9081	5,32%	625	172771	1957	1,15%	366
174167	3353	1,96%	295	171708	894	0,52%	592	176693	5879	3,44%	380
171594	780	0,46%	333	172183	1369	0,80%	574	174666	3852	2,26%	377
173894	3080	1,80%	291	172825	2011	1,18%	595	172766	1952	1,14%	336
173548	2734	1,60%	302	178986	8172	4,78%	619	174723	3909	2,29%	364
171189	375	0,22%	301	171471	657	0,38%	596	171648	834	0,49%	359
174180	3366	1,97%	348	181315	10501	6,15%	619	174998	4184	2,45%	332
172354	1540	0,90%	285	175622	4808	2,81%	590	172366	1552	0,91%	341
172954	2140	1,25%	335	171047	233	0,14%	515	171195	381	0,22%	339
173449	2635	1,54%	330	180770	9956	5,83%	532	171867	1053	0,62%	396
171302	488	0,29%	335	179482	8668	5,07%	525	171894	1080	0,63%	327
172938	2124	1,24%	322	177515	6701	3,92%	620	172097	1283	0,75%	381
174015	3201	1,87%	288	176562	5748	3,37%	601	177171	6357	3,72%	377
171072	258	0,15%	331	171178	364	0,21%	568	173898	3084	1,81%	381
171843	1029	0,60%	301	182049	11235	6,58%	563	174259	3445	2,02%	372
174101	3287	1,92%	297	177549	6735	3,94%	551	176783	5969	3,49%	384
173589	2775	1,62%	292	172361	1547	0,91%	600	174289	3475	2,03%	338
173066	2252	1,32%	341	177489	6675	3,91%	588	171176	362	0,21%	339
173206	2392	1,40%	324	182343	11529	6,75%	617	174449	3635	2,13%	381
172460	1646	0,96%	337	182379	11565	6,77%	519	171399	585	0,34%	328
171847	1033	0,60%	324	181408	10594	6,20%	584	176551	5737	3,36%	338
172786	1972	1,15%	293	178920	8106	4,75%	608	174255	3441	2,01%	334
172645	1831	1,07%	308	177365	6551	3,84%	544	172283	1469	0,86%	333
173487	2673	1,56%	348	176526	5712	3,34%	560	172492	1678	0,98%	389
173612	2798	1,64%	337	177893	7079	4,14%	604	173439	2625	1,54%	390
172585	1771	1,04%	310	172990	2176	1,27%	560	173044	2230	1,31%	342

Çizelge CC.1 (Devam) : 1000×60×50×300₂ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
172557	1743	1,02%	303	182254	11440	6,70%	549	172653	1839	1,08%	371
174027	3213	1,88%	324	172367	1553	0,91%	528	175430	4616	2,70%	390
173052	2238	1,31%	334	173973	3159	1,85%	587	173803	2989	1,75%	369
173644	2830	1,66%	298	175428	4614	2,70%	530	170938	124	0,07%	344
173519	2705	1,58%	339	172482	1668	0,98%	528	171544	730	0,43%	334
173310	2496	1,46%	285	178992	8178	4,79%	574	174535	3721	2,18%	383
171160	346	0,20%	285	175214	4400	2,58%	549	171224	410	0,24%	358
174068	3254	1,90%	295	177831	7017	4,11%	563	172045	1231	0,72%	330
173489	2675	1,57%	295	171806	992	0,58%	515	176105	5291	3,10%	366
171341	527	0,31%	328	182045	11231	6,57%	621	177237	6423	3,76%	377
174106	3292	1,93%	339	181977	11163	6,54%	571	175318	4504	2,64%	357
174100	3286	1,92%	317	177991	7177	4,20%	526	172231	1417	0,83%	386
171760	946	0,55%	313	177611	6797	3,98%	517	173770	2956	1,73%	359
172069	1255	0,73%	330	171673	859	0,50%	553	172994	2180	1,28%	355
171543	729	0,43%	333	182579	11765	6,89%	584	172589	1775	1,04%	373
171574	760	0,44%	337	178704	7890	4,62%	520	174968	4154	2,43%	359
173749	2935	1,72%	301	177171	6357	3,72%	535	176556	5742	3,36%	346
173200	2386	1,40%	307	173358	2544	1,49%	578	177151	6337	3,71%	332
173131	2317	1,36%	309	180999	10185	5,96%	600	171965	1151	0,67%	394
172463	1649	0,97%	337	181512	10698	6,26%	577	175425	4611	2,70%	359
172611	1797	1,05%	321	176486	5672	3,32%	532	174699	3885	2,27%	329
173893	3079	1,80%	344	175139	4325	2,53%	614	175390	4576	2,68%	336
173590	2776	1,63%	331	182498	11684	6,84%	622	171306	492	0,29%	357
172097	1283	0,75%	344	171169	355	0,21%	536	174830	4016	2,35%	358
172625	1811	1,06%	304	171064	250	0,15%	587	177218	6404	3,75%	346
171268	454	0,27%	348	171262	448	0,26%	561	172819	2005	1,17%	333
171969	1155	0,68%	291	176899	6085	3,56%	594	177199	6385	3,74%	343
173397	2583	1,51%	287	178823	8009	4,69%	549	174436	3622	2,12%	369
172068	1254	0,73%	287	172334	1520	0,89%	603	174886	4072	2,38%	339

EK DD : 1000×60×50×300₃ Problemi İçin Bulunan Sonuçlar

1000×60×50×300₃ problemi için bulunan en iyi değer 196330 olup, Çizelge DD.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge DD.1 : 1000×60×50×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
196524	194	0,10%	356	208233	11903	6,06%	615	202107	5777	2,94%	371
199307	2977	1,52%	328	202196	5866	2,99%	623	199414	3084	1,57%	413
199289	2959	1,51%	348	205223	8893	4,53%	525	203019	6689	3,41%	406
199239	2909	1,48%	334	208657	12327	6,28%	616	203015	6685	3,40%	418
198119	1789	0,91%	304	198014	1684	0,86%	626	198972	2642	1,35%	361
199415	3085	1,57%	318	198268	1938	0,99%	539	202042	5712	2,91%	357
200182	3852	1,96%	300	201997	5667	2,89%	622	201903	5573	2,84%	402
197263	933	0,48%	345	208847	12517	6,38%	564	203619	7289	3,71%	413
199697	3367	1,71%	336	202857	6527	3,32%	592	198737	2407	1,23%	375
200105	3775	1,92%	327	207124	10794	5,50%	599	202381	6051	3,08%	393
196693	363	0,18%	345	202205	5875	2,99%	589	202422	6092	3,10%	360
198815	2485	1,27%	327	202679	6349	3,23%	636	200506	4176	2,13%	397
198745	2415	1,23%	311	202381	6051	3,08%	534	202274	5944	3,03%	399
199474	3144	1,60%	298	200183	3853	1,96%	629	201699	5369	2,73%	415
199282	2952	1,50%	304	199843	3513	1,79%	610	199723	3393	1,73%	394
198062	1732	0,88%	329	202425	6095	3,10%	522	199754	3424	1,74%	368
199474	3144	1,60%	306	208863	12533	6,38%	617	200773	4443	2,26%	398
197508	1178	0,60%	351	198086	1756	0,89%	531	202946	6616	3,37%	401
198957	2627	1,34%	321	199669	3339	1,70%	584	197029	699	0,36%	409
199619	3289	1,68%	350	204920	8590	4,38%	624	196715	385	0,20%	368
197447	1117	0,57%	352	205854	9524	4,85%	606	198112	1782	0,91%	356
199502	3172	1,62%	293	207368	11038	5,62%	532	197038	708	0,36%	366
197688	1358	0,69%	321	202641	6311	3,21%	582	199300	2970	1,51%	375
197728	1398	0,71%	331	198377	2047	1,04%	543	199897	3567	1,82%	403
196427	97	0,05%	353	207450	11120	5,66%	636	203313	6983	3,56%	368
196922	592	0,30%	308	199475	3145	1,60%	540	197886	1556	0,79%	426
196434	104	0,05%	300	207930	11600	5,91%	630	197834	1504	0,77%	392
198497	2167	1,10%	311	207323	10993	5,60%	627	204049	7719	3,93%	392
197613	1283	0,65%	321	203592	7262	3,70%	604	203864	7534	3,84%	416

Çizelge DD.1 (Devam) : 1000×60×50×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
196434	104	0,05%	314	209181	12851	6,55%	574	198743	2413	1,23%	354
198711	2381	1,21%	326	203580	7250	3,69%	567	203659	7329	3,73%	401
199139	2809	1,43%	318	201105	4775	2,43%	566	199397	3067	1,56%	397
197807	1477	0,75%	306	203004	6674	3,40%	632	203831	7501	3,82%	403
199134	2804	1,43%	346	205711	9381	4,78%	572	196508	178	0,09%	382
200218	3888	1,98%	332	205424	9094	4,63%	560	199765	3435	1,75%	426
197400	1070	0,55%	323	203737	7407	3,77%	550	201862	5532	2,82%	399
199844	3514	1,79%	308	197369	1039	0,53%	531	203424	7094	3,61%	426
199505	3175	1,62%	325	208251	11921	6,07%	603	196712	382	0,19%	408
196915	585	0,30%	341	204458	8128	4,14%	596	202262	5932	3,02%	408
196435	105	0,05%	343	201862	5532	2,82%	565	197205	875	0,45%	426
196330	0	0,00%	312	205721	9391	4,78%	570	203047	6717	3,42%	413
197457	1127	0,57%	331	209093	12763	6,50%	627	199864	3534	1,80%	370
197525	1195	0,61%	312	206303	9973	5,08%	604	202890	6560	3,34%	363
197835	1505	0,77%	343	208261	11931	6,08%	633	197081	751	0,38%	364
196713	383	0,20%	324	206022	9692	4,94%	592	201329	4999	2,55%	415
200067	3737	1,90%	296	201181	4851	2,47%	576	202575	6245	3,18%	410
197112	782	0,40%	313	205024	8694	4,43%	586	197575	1245	0,63%	373
196469	139	0,07%	310	198188	1858	0,95%	617	198650	2320	1,18%	391
198665	2335	1,19%	337	197313	983	0,50%	571	197430	1100	0,56%	387
200051	3721	1,90%	301	205935	9605	4,89%	559	203297	6967	3,55%	401
197700	1370	0,70%	334	204036	7706	3,93%	586	200307	3977	2,03%	419
198965	2635	1,34%	355	210058	13728	6,99%	566	198229	1899	0,97%	383
199166	2836	1,44%	333	206404	10074	5,13%	631	200280	3950	2,01%	400
197262	932	0,47%	338	202183	5853	2,98%	539	197377	1047	0,53%	426
197929	1599	0,81%	297	197310	980	0,50%	599	202335	6005	3,06%	384
200132	3802	1,94%	324	198923	2593	1,32%	631	204122	7792	3,97%	364
196721	391	0,20%	298	200364	4034	2,05%	561	196740	410	0,21%	412
196809	479	0,24%	327	205943	9613	4,90%	629	198131	1801	0,92%	383
196713	383	0,20%	355	199626	3296	1,68%	621	202719	6389	3,25%	426
199969	3639	1,85%	340	208927	12597	6,42%	544	201713	5383	2,74%	389
199212	2882	1,47%	309	200683	4353	2,22%	605	201104	4774	2,43%	426
199762	3432	1,75%	354	199662	3332	1,70%	627	199243	2913	1,48%	366
197557	1227	0,62%	353	198615	2285	1,16%	568	196714	384	0,20%	360
197433	1103	0,56%	346	203617	7287	3,71%	561	197849	1519	0,77%	363
198175	1845	0,94%	351	207209	10879	5,54%	538	200294	3964	2,02%	414
199473	3143	1,60%	336	205551	9221	4,70%	584	200545	4215	2,15%	415
198458	2128	1,08%	305	206582	10252	5,22%	579	199107	2777	1,41%	407
198437	2107	1,07%	321	209155	12825	6,53%	569	196851	521	0,27%	383
197147	817	0,42%	306	209754	13424	6,84%	610	202797	6467	3,29%	395
196622	292	0,15%	298	200566	4236	2,16%	560	198006	1676	0,85%	398
199004	2674	1,36%	314	207503	11173	5,69%	569	198747	2417	1,23%	356

Çizelge DD.1 (Devam) : 1000×60×50×300₃ problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
196407	77	0,04%	302	205625	9295	4,73%	626	203133	6803	3,47%	378
198728	2398	1,22%	346	197665	1335	0,68%	602	200872	4542	2,31%	384
198715	2385	1,21%	320	207822	11492	5,85%	627	201045	4715	2,40%	425
197187	857	0,44%	342	203671	7341	3,74%	563	198021	1691	0,86%	414
197468	1138	0,58%	329	200315	3985	2,03%	559	200825	4495	2,29%	378
198525	2195	1,12%	355	201158	4828	2,46%	637	203477	7147	3,64%	369
199675	3345	1,70%	316	202267	5937	3,02%	594	199748	3418	1,74%	421
197249	919	0,47%	293	208107	11777	6,00%	617	203630	7300	3,72%	360
196565	235	0,12%	311	201227	4897	2,49%	534	198271	1941	0,99%	370
197352	1022	0,52%	324	198280	1950	0,99%	524	201851	5521	2,81%	416
200071	3741	1,91%	301	209316	12986	6,61%	590	202228	5898	3,00%	418
197327	997	0,51%	317	205747	9417	4,80%	573	198388	2058	1,05%	397
197163	833	0,42%	316	197346	1016	0,52%	531	196792	462	0,24%	424
198361	2031	1,03%	340	201352	5022	2,56%	629	199487	3157	1,61%	422
198312	1982	1,01%	336	200811	4481	2,28%	575	202413	6083	3,10%	418
198700	2370	1,21%	307	198824	2494	1,27%	554	197557	1227	0,62%	368
199260	2930	1,49%	341	208310	11980	6,10%	564	201539	5209	2,65%	356
200086	3756	1,91%	348	198915	2585	1,32%	571	203603	7273	3,70%	387
200105	3775	1,92%	333	199211	2881	1,47%	598	197551	1221	0,62%	362
197446	1116	0,57%	327	204233	7903	4,03%	597	197370	1040	0,53%	417
199918	3588	1,83%	343	208547	12217	6,22%	535	197148	818	0,42%	404
198620	2290	1,17%	313	203995	7665	3,90%	611	200283	3953	2,01%	410
198728	2398	1,22%	320	204826	8496	4,33%	594	196457	127	0,06%	388
196970	640	0,33%	310	205338	9008	4,59%	548	198034	1704	0,87%	367
199928	3598	1,83%	305	210068	13738	7,00%	538	198691	2361	1,20%	392
197474	1144	0,58%	305	202886	6556	3,34%	569	198923	2593	1,32%	425
200104	3774	1,92%	300	205805	9475	4,83%	559	198746	2416	1,23%	420
199191	2861	1,46%	302	202711	6381	3,25%	525	197733	1403	0,71%	422
198143	1813	0,92%	334	203640	7310	3,72%	582	203712	7382	3,76%	360

EK EE : 6×4×4×720 Problemi İçin Bulunan Sonuçlar

6×4×4×720 problemi için bulunan en iyi değer 247258 olup, Çizelge EE.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge EE.1 : 6×4×4×720 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
249570	2312	0,94%	263	249339	2081	0,84%	457	248646	1388	0,56%	261
248681	1423	0,58%	284	261150	13892	5,62%	488	248369	1111	0,45%	288
247530	272	0,11%	254	256455	9197	3,72%	439	257047	9789	3,96%	277
248908	1650	0,67%	283	247392	134	0,05%	509	249373	2115	0,86%	295
249264	2006	0,81%	267	253732	6474	2,62%	490	253328	6070	2,45%	278
250361	3103	1,25%	285	258159	10901	4,41%	454	251770	4512	1,82%	296
249165	1907	0,77%	257	250506	3248	1,31%	445	248753	1495	0,60%	273
248101	843	0,34%	261	251817	4559	1,84%	508	247826	568	0,23%	281
250938	3680	1,49%	256	262632	15374	6,22%	454	252026	4768	1,93%	312
249793	2535	1,03%	268	249460	2202	0,89%	519	250589	3331	1,35%	293
249728	2470	1,00%	278	258732	11474	4,64%	487	248620	1362	0,55%	299
249395	2137	0,86%	294	254886	7628	3,09%	523	249123	1865	0,75%	269
251290	4032	1,63%	285	252111	4853	1,96%	464	247875	617	0,25%	259
250121	2863	1,16%	264	253297	6039	2,44%	462	248412	1154	0,47%	267
248472	1214	0,49%	283	250856	3598	1,46%	491	250118	2860	1,16%	274
248819	1561	0,63%	277	259452	12194	4,93%	459	254903	7645	3,09%	311
247545	287	0,12%	264	256952	9694	3,92%	457	255257	7999	3,24%	312
250568	3310	1,34%	257	254546	7288	2,95%	464	251303	4045	1,64%	276
249877	2619	1,06%	301	262911	15653	6,33%	465	251794	4536	1,83%	263
251062	3804	1,54%	251	258095	10837	4,38%	518	247965	707	0,29%	308
248366	1108	0,45%	300	257015	9757	3,95%	478	251467	4209	1,70%	265
249841	2583	1,04%	294	256283	9025	3,65%	473	252306	5048	2,04%	258
250889	3631	1,47%	295	262527	15269	6,18%	464	249819	2561	1,04%	298
249821	2563	1,04%	299	256052	8794	3,56%	480	249244	1986	0,80%	288
252133	4875	1,97%	297	248591	1333	0,54%	440	250195	2937	1,19%	310
252195	4937	2,00%	256	251365	4107	1,66%	523	255738	8480	3,43%	270
249177	1919	0,78%	276	263026	15768	6,38%	512	252386	5128	2,07%	259

Çizelge EE.1 (Devam) : 6×4×4×720 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
251679	4421	1,79%	294	254167	6909	2,79%	506	252625	5367	2,17%	282
250501	3243	1,31%	286	253812	6554	2,65%	433	249770	2512	1,02%	307
247676	418	0,17%	252	249526	2268	0,92%	441	250635	3377	1,37%	305
248960	1702	0,69%	275	260159	12901	5,22%	502	248612	1354	0,55%	313
249278	2020	0,82%	276	250152	2894	1,17%	467	255962	8704	3,52%	297
249620	2362	0,96%	281	249688	2430	0,98%	484	255743	8485	3,43%	260
251297	4039	1,63%	272	248576	1318	0,53%	521	250680	3422	1,38%	288
249484	2226	0,90%	295	258138	10880	4,40%	526	254186	6928	2,80%	281
250987	3729	1,51%	270	255850	8592	3,47%	528	248434	1176	0,48%	272
248639	1381	0,56%	249	253607	6349	2,57%	457	255691	8433	3,41%	309
250592	3334	1,35%	280	256826	9568	3,87%	522	248880	1622	0,66%	271
251903	4645	1,88%	274	255305	8047	3,25%	471	248748	1490	0,60%	279
251853	4595	1,86%	299	248905	1647	0,67%	484	249363	2105	0,85%	265
247258	0	0,00%	297	250719	3461	1,40%	464	250928	3670	1,48%	310
251060	3802	1,54%	279	249365	2107	0,85%	517	249104	1846	0,75%	312
249308	2050	0,83%	255	260551	13293	5,38%	455	254984	7726	3,12%	272
252190	4932	1,99%	272	258341	11083	4,48%	453	252319	5061	2,05%	280
251171	3913	1,58%	297	252261	5003	2,02%	500	253672	6414	2,59%	274
251241	3983	1,61%	299	249428	2170	0,88%	438	252715	5457	2,21%	272
251689	4431	1,79%	251	262680	15422	6,24%	502	254515	7257	2,93%	305
250035	2777	1,12%	269	249250	1992	0,81%	442	247616	358	0,14%	273
249419	2161	0,87%	260	254498	7240	2,93%	528	256621	9363	3,79%	286
248198	940	0,38%	262	254491	7233	2,93%	465	247499	241	0,10%	301
249541	2283	0,92%	298	248737	1479	0,60%	483	251842	4584	1,85%	286
248785	1527	0,62%	269	253823	6565	2,66%	509	247260	2	0,00%	270
249485	2227	0,90%	270	253832	6574	2,66%	479	256047	8789	3,55%	314
249887	2629	1,06%	281	261795	14537	5,88%	449	250399	3141	1,27%	305
248911	1653	0,67%	265	259549	12291	4,97%	521	249735	2477	1,00%	261
249933	2675	1,08%	259	247867	609	0,25%	491	254463	7205	2,91%	286
249188	1930	0,78%	249	260997	13739	5,56%	515	248427	1169	0,47%	288
250906	3648	1,48%	267	249974	2716	1,10%	495	256586	9328	3,77%	268
249377	2119	0,86%	275	262929	15671	6,34%	443	249930	2672	1,08%	278
247784	526	0,21%	270	251407	4149	1,68%	437	256508	9250	3,74%	275
250572	3314	1,34%	277	257749	10491	4,24%	457	251391	4133	1,67%	262
250270	3012	1,22%	293	249171	1913	0,77%	457	250485	3227	1,31%	270
251212	3954	1,60%	264	260278	13020	5,27%	499	252889	5631	2,28%	258
251644	4386	1,77%	265	253206	5948	2,41%	453	253831	6573	2,66%	288
248424	1166	0,47%	272	253465	6207	2,51%	521	256954	9696	3,92%	277
251170	3912	1,58%	258	247448	190	0,08%	528	250978	3720	1,50%	262
248377	1119	0,45%	280	248098	840	0,34%	488	248057	799	0,32%	260
247751	493	0,20%	284	261380	14122	5,71%	508	253796	6538	2,64%	276
250840	3582	1,45%	249	259327	12069	4,88%	437	248869	1611	0,65%	294

Çizelge EE.1 (Devam) : 6×4×4×720 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
248605	1347	0,54%	274	250232	2974	1,20%	526	253450	6192	2,50%	266
251070	3812	1,54%	283	251249	3991	1,61%	434	256739	9481	3,83%	305
247386	128	0,05%	281	262428	15170	6,14%	489	251396	4138	1,67%	297
252056	4798	1,94%	303	257790	10532	4,26%	505	253433	6175	2,50%	283
249849	2591	1,05%	275	263157	15899	6,43%	496	251498	4240	1,71%	301
247692	434	0,18%	283	252747	5489	2,22%	507	255449	8191	3,31%	304
251756	4498	1,82%	272	257855	10597	4,29%	491	256297	9039	3,66%	285
252201	4943	2,00%	297	249286	2028	0,82%	500	247262	4	0,00%	288
248467	1209	0,49%	252	250527	3269	1,32%	487	256747	9489	3,84%	313
247637	379	0,15%	286	254730	7472	3,02%	506	256549	9291	3,76%	299
250194	2936	1,19%	296	249793	2535	1,03%	522	251744	4486	1,81%	314
250440	3182	1,29%	262	250219	2961	1,20%	508	248215	957	0,39%	273
247626	368	0,15%	297	254392	7134	2,89%	476	253938	6680	2,70%	305
248119	861	0,35%	276	256670	9412	3,81%	492	247362	104	0,04%	277
248774	1516	0,61%	264	256842	9584	3,88%	481	255410	8152	3,30%	310
249356	2098	0,85%	252	250163	2905	1,17%	433	250900	3642	1,47%	311
248612	1354	0,55%	284	247835	577	0,23%	496	254625	7367	2,98%	301
251976	4718	1,91%	258	247439	181	0,07%	476	256322	9064	3,67%	263
247601	343	0,14%	282	250724	3466	1,40%	499	251443	4185	1,69%	287
251951	4693	1,90%	261	247707	449	0,18%	508	254577	7319	2,96%	279
249860	2602	1,05%	292	262197	14939	6,04%	490	254078	6820	2,76%	275
247893	635	0,26%	295	251931	4673	1,89%	479	248398	1140	0,46%	262
250148	2890	1,17%	280	262726	15468	6,26%	495	256609	9351	3,78%	309
250248	2990	1,21%	261	259313	12055	4,88%	456	249520	2262	0,91%	270
251876	4618	1,87%	271	252230	4972	2,01%	433	253479	6221	2,52%	311
251605	4347	1,76%	263	250768	3510	1,42%	508	255924	8666	3,50%	284
249173	1915	0,77%	256	247868	610	0,25%	500	247650	392	0,16%	285
249058	1800	0,73%	284	261747	14489	5,86%	524	247555	297	0,12%	300
250857	3599	1,46%	299	259972	12714	5,14%	499	248729	1471	0,59%	302
249012	1754	0,71%	258	261822	14564	5,89%	469	257023	9765	3,95%	302
250318	3060	1,24%	249	261906	14648	5,92%	488	254712	7454	3,01%	309

EK FF : 4×2×2×80 Problemi İçin Bulunan Sonuçlar

4×2×2×80 problemi için bulunan en iyi değer 446736 olup, Çizelge FF.1 algoritmalarla yapılan 100 denemede bulunan çözümleri ve çözüm sürelerini göstermektedir. Sonuçlarda kullanılan yöntemler 2006 yılında Altıparmak ve diğerleri tarafından geliştirilen Genetik Algoritma ve Jayaraman ve Ross tarafından 2003 yılında geliştirilen Tavlama Benzetimi Algoritmasıdır. Çizelgede Fark kolonu yöntemin o denemede bulduğu sonuç ile bulunan en iyi sonuç arasındaki farkı, Yüzde kolonu bu fark ile bulunan en iyi sonuç değeri oranını ve Süre kolonu ise yöntemin o denemede bulduğu en iyi çözümü üretme süresini (sn.) göstermektedir.

Çizelge FF.1 : 4×2×2×80 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
446840	104	0,02%	95	449042	2306	0,52%	151	446955	219	0,05%	110
447026	290	0,06%	93	449768	3032	0,68%	158	447099	363	0,08%	97
447004	268	0,06%	82	448039	1303	0,29%	151	447166	430	0,10%	94
446976	240	0,05%	96	446770	34	0,01%	131	447352	616	0,14%	100
446762	26	0,01%	83	449370	2634	0,59%	132	446968	232	0,05%	112
447109	373	0,08%	94	449046	2310	0,52%	140	447069	333	0,07%	113
447311	575	0,13%	93	448868	2132	0,48%	142	446771	35	0,01%	103
446771	35	0,01%	91	448241	1505	0,34%	158	447646	910	0,20%	110
447257	521	0,12%	95	448153	1417	0,32%	138	446825	89	0,02%	98
447281	545	0,12%	96	448173	1437	0,32%	155	446790	54	0,01%	96
447112	376	0,08%	83	449242	2506	0,56%	156	447491	755	0,17%	108
447238	502	0,11%	89	448025	1289	0,29%	148	447176	440	0,10%	109
447262	526	0,12%	86	449578	2842	0,64%	148	447289	553	0,12%	109
447129	393	0,09%	99	448942	2206	0,49%	143	446772	36	0,01%	95
447181	445	0,10%	90	448739	2003	0,45%	141	446752	16	0,00%	94
446736	0	0,00%	83	447872	1136	0,25%	146	447399	663	0,15%	112
447035	299	0,07%	87	448339	1603	0,36%	133	447237	501	0,11%	94
447242	506	0,11%	93	449247	2511	0,56%	152	447012	276	0,06%	102
447215	479	0,11%	97	447077	341	0,08%	158	446805	69	0,02%	109
447137	401	0,09%	86	447246	510	0,11%	145	447395	659	0,15%	106
447045	309	0,07%	85	447979	1243	0,28%	148	447482	746	0,17%	96
446751	15	0,00%	95	448731	1995	0,45%	143	447434	698	0,16%	101
446827	91	0,02%	93	447146	410	0,09%	156	446920	184	0,04%	110
446742	6	0,00%	83	448523	1787	0,40%	155	447068	332	0,07%	93
447169	433	0,10%	93	447161	425	0,10%	144	447703	967	0,22%	99
447225	489	0,11%	82	447127	391	0,09%	132	447229	493	0,11%	102
447338	602	0,13%	90	449575	2839	0,64%	141	447395	659	0,15%	109

Çizelge FF.1 (Devam) : 4×2×2×80 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
447275	539	0,12%	91	448589	1853	0,41%	134	447174	438	0,10%	98
447214	478	0,11%	92	447496	760	0,17%	139	447420	684	0,15%	96
447125	389	0,09%	91	448835	2099	0,47%	134	447455	719	0,16%	101
447268	532	0,12%	87	449253	2517	0,56%	152	446741	5	0,00%	112
447100	364	0,08%	98	448038	1302	0,29%	140	446996	260	0,06%	109
447044	308	0,07%	88	446964	228	0,05%	132	447386	650	0,15%	101
446945	209	0,05%	82	447925	1189	0,27%	140	447159	423	0,09%	106
447397	661	0,15%	85	449200	2464	0,55%	150	446862	126	0,03%	94
447112	376	0,08%	98	447595	859	0,19%	138	447574	838	0,19%	101
446829	93	0,02%	95	449620	2884	0,65%	134	447489	753	0,17%	95
447064	328	0,07%	86	447531	795	0,18%	157	447400	664	0,15%	101
447291	555	0,12%	88	448324	1588	0,36%	138	447149	413	0,09%	101
447074	338	0,08%	82	447719	983	0,22%	132	447529	793	0,18%	102
447400	664	0,15%	99	448890	2154	0,48%	160	447550	814	0,18%	96
447184	448	0,10%	84	447058	322	0,07%	151	446810	74	0,02%	96
447223	487	0,11%	95	449471	2735	0,61%	140	447355	619	0,14%	112
447228	492	0,11%	85	448971	2235	0,50%	137	446926	190	0,04%	106
446920	184	0,04%	92	447789	1053	0,24%	158	446879	143	0,03%	101
447272	536	0,12%	93	448593	1857	0,42%	153	446961	225	0,05%	105
447329	593	0,13%	88	447240	504	0,11%	139	447581	845	0,19%	107
446999	263	0,06%	92	449856	3120	0,70%	133	447068	332	0,07%	105
447271	535	0,12%	87	447633	897	0,20%	131	447667	931	0,21%	100
447300	564	0,13%	92	448041	1305	0,29%	137	446977	241	0,05%	96
447213	477	0,11%	91	449269	2533	0,57%	146	447010	274	0,06%	106
446942	206	0,05%	84	448898	2162	0,48%	148	447348	612	0,14%	99
447207	471	0,11%	88	449055	2319	0,52%	143	447337	601	0,13%	97
446758	22	0,00%	82	448786	2050	0,46%	146	447077	341	0,08%	112
447174	438	0,10%	98	447377	641	0,14%	144	447501	765	0,17%	93
447318	582	0,13%	98	448022	1286	0,29%	136	447303	567	0,13%	102
447162	426	0,10%	85	448870	2134	0,48%	148	447291	555	0,12%	109
446981	245	0,05%	87	446787	51	0,01%	153	447024	288	0,06%	107
447235	499	0,11%	94	447647	911	0,20%	137	446803	67	0,01%	100
447107	371	0,08%	90	447915	1179	0,26%	148	447126	390	0,09%	110
446751	15	0,00%	82	449624	2888	0,65%	155	446802	66	0,01%	99
447315	579	0,13%	97	448646	1910	0,43%	158	446841	105	0,02%	101
446765	29	0,01%	93	447780	1044	0,23%	154	447163	427	0,10%	112
446800	64	0,01%	93	448930	2194	0,49%	140	446867	131	0,03%	109
447246	510	0,11%	99	449475	2739	0,61%	134	446841	105	0,02%	107
447281	545	0,12%	97	446868	132	0,03%	143	447574	838	0,19%	95
446741	5	0,00%	90	449107	2371	0,53%	155	446889	153	0,03%	109
446749	13	0,00%	88	449708	2972	0,67%	136	447217	481	0,11%	108
446994	258	0,06%	94	448665	1929	0,43%	154	447103	367	0,08%	105

Çizelge FF.1 (Devam) : 4×2×2×80 problemi için bulunan sonuçlar.

Hipersezgisel Algoritma				Genetik Algoritma				Tavlama Benzetimi			
Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)	Sonuç	Fark	Yüzde	Süre (sn)
447401	665	0,15%	88	446836	100	0,02%	132	447016	280	0,06%	106
447176	440	0,10%	90	448075	1339	0,30%	148	447125	389	0,09%	100
447098	362	0,08%	82	449544	2808	0,63%	153	447106	370	0,08%	109
447159	423	0,09%	92	448234	1498	0,34%	152	447562	826	0,18%	97
447294	558	0,12%	90	447134	398	0,09%	158	447594	858	0,19%	99
446804	68	0,02%	98	446797	61	0,01%	155	447065	329	0,07%	100
447046	310	0,07%	87	448856	2120	0,47%	140	447466	730	0,16%	108
447389	653	0,15%	90	448617	1881	0,42%	133	447385	649	0,15%	97
446923	187	0,04%	89	446929	193	0,04%	140	447706	970	0,22%	107
447401	665	0,15%	95	446744	8	0,00%	139	447191	455	0,10%	107
447252	516	0,12%	97	449299	2563	0,57%	141	447206	470	0,11%	101
446959	223	0,05%	83	449596	2860	0,64%	154	447573	837	0,19%	102
447332	596	0,13%	99	449364	2628	0,59%	146	446951	215	0,05%	101
447169	433	0,10%	93	447970	1234	0,28%	153	447251	515	0,12%	95
447137	401	0,09%	85	446833	97	0,02%	151	446743	7	0,00%	98
447092	356	0,08%	94	447843	1107	0,25%	141	447388	652	0,15%	98
447228	492	0,11%	89	449840	3104	0,69%	139	447119	383	0,09%	109
447236	500	0,11%	98	447703	967	0,22%	158	447613	877	0,20%	101
447162	426	0,10%	92	446786	50	0,01%	149	446774	38	0,01%	96
447073	337	0,08%	92	447286	550	0,12%	132	447155	419	0,09%	95
446860	124	0,03%	97	447717	981	0,22%	134	447463	727	0,16%	99
447311	575	0,13%	82	449300	2564	0,57%	158	447515	779	0,17%	111
447354	618	0,14%	98	447648	912	0,20%	144	447701	965	0,22%	109
446816	80	0,02%	85	448803	2067	0,46%	142	446825	89	0,02%	111
446863	127	0,03%	96	447811	1075	0,24%	148	447339	603	0,13%	95
446863	127	0,03%	90	448546	1810	0,41%	132	447388	652	0,15%	102
447369	633	0,14%	90	449220	2484	0,56%	150	446738	2	0,00%	105
447324	588	0,13%	87	448349	1613	0,36%	152	447188	452	0,10%	100
446973	237	0,05%	93	449313	2577	0,58%	151	447013	277	0,06%	111
446914	178	0,04%	97	449816	3080	0,69%	134	446769	33	0,01%	111
447387	651	0,15%	85	447365	629	0,14%	138	447362	626	0,14%	105



ÖZGEÇMİŞ

Ad-Soyad : Vural EROL
Doğum Tarihi ve Yeri : 25 Ağustos 1981
E-posta : vuralerol@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2003, İstanbul Teknik Üniversitesi, İşletme Fakültesi, Endüstri Mühendisliği Bölümü
- **Yüksek Lisans** : 2006, Yıldız Teknik Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Sistem Mühendisliği Programı

MESLEKİ DENEYİM:

- 2003-2016 yılları arasında Telekomunikasyon, İlaç, Danışmanlık, Sigorta ve Lojistik sektörlerinde çeşitli projelerde farklı pozisyonlarda görev yapmıştır.

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR VE SUNUMLAR:

- **Erol V.**, Baskak M., Kayakutlu, G. 2010. Methods Used In Logistic Network Design, *International Logistics and Supply Chain Congress' 2010*, November 4-5, 2010 İstanbul, Turkey.
- **Erol V.**, Baskak M., Kayakutlu, G. 2016. Lojistik Ağ Tasarımında Kullanılan Yöntemler ve Karmaşık Tamsayılı Doğrusal Programlama Modeli, *Lojistik Dergisi*, Sayı: 38, İstanbul, Turkey.

