

T.C.
KARA HARP OKULU
SAVUNMA BİLİMLERİ ENSTİTÜSÜ
HAREKÂT ARAŞTIRMASI ANA BİLİM DALI

**BÜTÜNLEŞİK TEDARİK ZİNCİRİ AĞINDA TESİS YERİ
SEÇİMİ PROBLEMİ İÇİN BULANIK ÇOK AMAÇLI
PROGRAMLAMA MODELİNE SEZGİSEL BİR YAKLAŞIM:
TAVLAMA BENZETİMİ ALGORİTMASI**

DOKTORA TEZİ

Hazırlayan
Hüseyin Ali SARIKAYA

Tez Danışmanı
Prof.Dr. Orhan TÜRKBEY

ANKARA - 2014

TEZ TANITIM FORMU

TEZİN TARİHİ: 27.03.2014

TEZİN TİPİ : Doktora Tezi

TEZİN BAŞLIĞI: Bütünleşik Tedarik Zinciri Ağında Tesis Yeri Seçimi Problemi için Bulanık Çok Amaçlı Programlama Modeline Sezgisel Bir Yaklaşım: Tavlama Benzetimi Algoritması.

TEZİN YAPILDIĞI BİRİM: Kara Harp Okulu Savunma Bilimleri Enstitüsü Harekât Araştırması Ana Bilim Dalı

SPONSOR KURULUŞ: -

DAĞITIM LİSTESİ: Kara Harp Okulu Savunma Bilimleri Enstitüsü Tez Hazırlama, Onay, Dağıtım ve Muhafaza Esasları Kılavuzunda Belirtilen Yerlere

TEZİN ÖZETİ:

Geleneksel tedarik zinciri ağı tasarım problemleri genellikle tek amaçlı olarak ele alınmıştır. Ancak, tedarik zincirleri gerçek hayatta birbirleri ile çelişen amaçları olan organizasyonların meydana getirdiği karmaşık ağlardır. Bu çalışmada, piyasa taleplerinin belirsiz olduğu bütünleşik bir tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için çok ürünlü, çok aşamalı ve çok dönemli planlama modeli önerilmiştir. Tedarik zinciri planlama modeli, birbiriyle çelişen birkaç amacı doyurmak için karışık tamsayılı doğrusal olmayan programlama problemi olarak bina edilmiştir. Önerilen model iki amaç fonksiyonundan oluşmaktadır. Birincisi, tedarik zincirindeki sabit tesis açma ve işletme maliyetleri ile mesafelere bağlı olarak belirlenen taşıma maliyetlerinin en azlanmasıdır. İkincisi, Ekonomik Üretim Miktarı (EÜM) modeline göre satın alma, sipariş verme, stokta tutma ve yok satma maliyetlerinin en azlanmasıdır.

Karar vericilerin kesin olmayan hedef deęerlerini önerilen modele dahil edebilmek için bulanık hedef programlama yaklaşımı kullanılmış, model, GAMS/DICOPT optimizasyon programı kullanılarak çözülmüştür. Bütünleşik tedarik zinciri ağında tesis yeri seçimi problemlerinin karmaşıklığı NP–Zor problemler sınıfına girmektedir. Modelin karmaşık yapısından dolayı, problem boyutu büyüdükçe çözüm zorlaşmakta ve hatta imkansız hale gelmektedir. Bu nedenle, büyük boyutlu problemlere de kabul edilebilir çözümler üretebilmek için Tavlama Benzetimi Algoritması tabanlı bir sezgisel yöntem geliştirilmiş ve Borland C++ Builder programlama dili ile kodlanmıştır. Bu yazılımdan elde edilen sonuçlar ile GAMS/DICOPT optimizasyon programından elde edilen sonuçlar karşılaştırılmış, önerilen Tavlama Benzetimi Algoritması tabanlı sezgisel yöntemin etkinliği test edilmiştir. Çalışmada sunulan uygulama sonuçları, bulanık modelleme ve çözüm yaklaşımlarının daha gerçekçi tedarik zinciri modelleri oluşturulmasında kullanılabileceğini göstermiştir.

ANAHTAR KELİMELEER: Tedarik Zinciri Ağı, Tesis Yeri Seçimi, Bulanık Hedef Programlama, Tavlama Benzetimi Algoritması.

SAYFA SAYISI : 118

GİZLİLİK DERECEİ : Tasnif Dışı

T.C.
KARA HARP OKULU
SAVUNMA BİLİMLERİ ENSTİTÜSÜ
HAREKÂT ARAŞTIRMASI ANA BİLİM DALI

**BÜTÜNLEŞİK TEDARİK ZİNCİRİ AĞINDA TESİS YERİ
SEÇİMİ PROBLEMİ İÇİN BULANIK ÇOK AMAÇLI
PROGRAMLAMA MODELİNE SEZGİSEL BİR YAKLAŞIM:
TAVLAMA BENZETİMİ ALGORİTMASI**

DOKTORA TEZİ

Hazırlayan
Hüseyin Ali SARIKAYA

Tez Danışmanı
Prof.Dr. Orhan TÜRKBEY

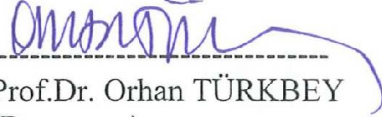
ANKARA - 2014

KARA HARP OKULU

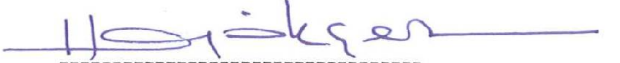
SAVUNMA BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Hüseyin Ali SARIKAYA'nın, Bütünleşik Tedarik Zinciri Ağında Tesis Yeri Seçimi Problemi İçin Bulanık Çok Amaçlı Programlama Modeline Sezgisel Bir Yaklaşım: Tavlama Benzetimi Algoritması konulu tez çalışması, jürimiz tarafından HAREKÂT ARAŞTIRMASI Ana Bilim Dalında DOKTORA tezi olarak kabul edilmiştir.

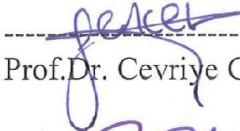
Başkan


Prof.Dr. Orhan TÜRKBEY
(Danışman)

Üye


Prof.Dr. Hadi GÖKÇEN

Üye


Prof.Dr. Cevriye GENCER

Üye


Yrd.Doç.Dr. Ramazan ŞAHİN

Üye


Yrd.Doç.Dr. Özkan BALI

ONAY

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

27/ 03 / 2014



Önder Haluk TEKBAŞ
Prof.Dr.
Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren danışmanım Prof.Dr. Orhan TÜRKBEY'e, Tez İzleme Komitesi toplantılarındaki öneri ve yönlendirmeleri ile katkı sağlayan Prof.Dr. Hadi GÖKÇEN ve Prof.Dr. Cevriye GENCER'e, bana zaman ayıran, bilgi ve tecrübesi ile destek olan, C++ kodu yazımında yardımcı olan Arş.Gör. Emre ÇALIŞKAN'a teşekkürlerimi sunarım.

Bu zorlu süreçte manevi destekleri ve sabırlarıyla beni hiçbir zaman yalnız bırakmayan annem, eşim Firdevs ve canım çocuklarım Kağan Efe ve Anday Ege'ye çok teşekkür ederim.

Bu tezi, bütün eğitim ve öğrenim hayatım boyunca beni her zaman destekleyen ve hiçbir yardımı esirgemeyen güzel babama ithaf ediyorum.

T.C.
KARA HARP OKULU
SAVUNMA BİLİMLERİ ENSTİTÜSÜ
HAREKÂT ARAŞTIRMASI ANA BİLİM DALI
ANKARA 2014

BÜTÜNLEŞİK TEDARİK ZİNCİRİ AĞINDA TESİS YERİ
SEÇİMİ PROBLEMİ İÇİN BULANIK ÇOK AMAÇLI
PROGRAMLAMA MODELİNE SEZGİSEL BİR YAKLAŞIM:
TAVLAMA BENZETİMİ ALGORİTMASI

DOKTORA TEZİ

Hüseyin Ali SARIKAYA

ÖZET

Geleneksel tedarik zinciri ağı tasarım problemleri genellikle tek amaçlı olarak ele alınmıştır. Ancak, tedarik zincirleri gerçek hayatta birbirleri ile çelişen amaçları olan organizasyonların meydana getirdiği karmaşık ağlardır. Bu çalışmada, piyasa taleplerinin belirsiz olduğu bütünleşik bir tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için çok ürünlü, çok aşamalı ve çok dönemli planlama modeli önerilmiştir. Tedarik zinciri planlama modeli, birbiriyle çelişen birkaç amacı doyurmak için karışık tamsayılı doğrusal olmayan programlama problemi olarak bina edilmiştir. Önerilen model iki amaç fonksiyonundan oluşmaktadır. Birincisi, tedarik zincirindeki sabit tesis açma ve işletme maliyetleri ile mesafelere bağlı olarak belirlenen taşıma maliyetlerinin en azlanmasıdır. İkincisi, Ekonomik Üretim Miktarı (EÜM) modeline göre satın alma, sipariş verme, stok bulundurma ve yok satma maliyetlerinin en azlanmasıdır.

Karar vericilerin kesin olmayan hedef değerlerini önerilen modele dahil edebilmek için bulanık hedef programlama yaklaşımı kullanılmış, model, GAMS/DICOPT optimizasyon programı kullanılarak çözülmüştür. Bütünleşik

tedarik zinciri ağında tesis yeri seçimi problemlerinin karmaşıklığı NP–Zor problemler sınıfına girmektedir. Modelin karmaşık yapısından dolayı, problem boyutu büyüdükçe çözüm zorlaşmakta ve hatta imkansız hale gelmektedir. Bu nedenle, büyük boyutlu problemlere de kabul edilebilir çözümler elde edebilmek için Tavlama Benzetimi Algoritması tabanlı bir sezgisel yöntem geliştirilmiş ve Borland C++ Builder programlama dili ile kodlanmıştır. Bu yazılımdan elde edilen sonuçlar ile GAMS/DICOPT optimizasyon programından elde edilen sonuçlar karşılaştırılmış, önerilen Tavlama Benzetimi Algoritması tabanlı sezgisel yöntemin etkinliği test edilmiştir. Çalışmada sunulan uygulama sonuçları, bulanık modelleme ve çözüm yaklaşımlarının daha gerçekçi tedarik zinciri modelleri oluşturulmasında kullanılabileceğini göstermiştir.

Anahtar Kelimeler : Tedarik Zinciri Ağı, Tesis Yeri Seçimi, Bulanık Hedef Programlama, Tavlama Benzetimi Algoritması.

Tez Yöneticisi : Prof.Dr. Orhan TÜRKBEY

Sayfa Sayısı : 118

T.C.
TURKISH MILITARY ACADEMY
DEFENSE SCIENCES INSTITUTE
DEPARTMENT OF OPERATIONS RESEARCH
ANKARA 2014

**A HEURISTIC APPROACH TO FUZZY MULTI-OBJECTIVE
PROGRAMMING MODEL FOR FACILITY LOCATION PROBLEM IN AN
INTEGRATED SUPPLY CHAIN NETWORK: SIMULATED ANNEALING
ALGORITHM**

Ph.D. THESIS

Hüseyin Ali SARIKAYA

ABSTRACT

Traditional supply chain network design problems are often taken as a single objective. However, supply chains are complex networks formed by organizations having conflicting objectives with each other in real life. In this study, a multi-product, multi-stage and multi-period planning model is proposed to achieve multiple incommensurable goals in an integrated supply chain network with uncertain market demands. The supply chain planning model is constructed as a mixed-integer nonlinear programming problem to satisfy several conflicting objectives with each other. The proposed model consists of two objective functions. The first one is minimizing the fixed opening and operating costs with transportation costs determined depending on distances. Second one is minimizing the purchasing, ordering, inventory and backlogging costs according to Economic Production Quantity (EPQ) model.

Fuzzy goal programming approach is used in order to include decision maker's imprecise goal values in proposed model. The model is solved using GAMS/DICOPT optimization program. Complexity of facility location problems in an integrated supply chain network falls into NP-Hard class problems. Due to the

complexity of the model, the problem size grows, the solution is becoming difficult and even impossible. Therefore, in order to achieve acceptable solutions to large-scale problems, Simulated Annealing Algorithm based heuristic methods have been developed and Borland C ++ Builder programming language is coded. The results obtained from this approach and GAMS/DICOPT obtained from optimization software problems have been compared and the effectiveness of the Simulated Annealing algorithm based heuristic has been tested. The application results presented in this study, demonstrates that fuzzy modeling and solution approaches could be used in the creation of more realistic models of the supply chain.

Keywords : Supply Chain Network, Facility Location, Fuzzy Goal Programming, Simulated Annealing.

Advisor : Prof.Dr. Orhan TÜRKBEY

Number of Pages : 118

İÇİNDEKİLER

TEŞEKKÜR	i
ÖZET	ii
ABSTRACT	iv
İÇİNDEKİLER	vi
TABLolar	ix
ŞEKİLLER	x
KISALTMALAR	xii
ÖNSÖZ	xiii
GİRİŞ	1

BİRİNCİ BÖLÜM

TEDARİK ZİNCİRİ YÖNETİMİ

1. TEDARİK ZİNCİRİ KAVRAMI	5
a. Tanımlar	5
b. Tedarik Zinciri Ağ Yapısı	8
c. Tedarik Zincirinin Amacı	10
2. TEDARİK ZİNCİRİ YÖNETİMİNDE KARAR SÜREÇLERİ	10
a. Tedarik Zincirinde Karar Seviyeleri	10
b. Tedarik Zincirinde Karar Alanları	11
(1) Yer Seçimi Kararları	11
(2) Üretim Kararları	12
(3) Stok Kararları	12
(4) Taşıma Kararları	12
(5) Dağıtım Ağ Kararları	13
2. TEDARİK ZİNCİRİNDE STOK KONTROLÜ	13
a. Stok Kontrolü ile İlgili Tanımlar	14
b. Stok Kontrolünün Amacı	15
c. Stok Maliyetleri	15
ç. Stok Kontrol Politikaları	16

d. Stok Kontrol Modelleri.....	18
(1) Deterministik Stok Kontrol Modelleri.....	19
(2) Olasılıklı Stok Kontrol Modelleri.....	28

İKİNCİ BÖLÜM

TEDARİK ZİNCİRİ AĞINDA TESİS YERİ SEÇİMİ PROBLEMİ

1. GENEL.....	29
2. TESİS YERİ SEÇİMİ PROBLEMLERİNİN SINIFLANDIRILMASI	30
3. TESİS YERİ SEÇİMİ PROBLEM TİPLERİ.....	33
4. ÇÖZÜM TEKNİKLERİ.....	33
5. TEDARİK ZİNCİRİ AĞINDA TESİS YERİ SEÇİMİ PROBLEMLERİ İLE İLGİLİ LİTERATÜR.....	34

ÜÇÜNCÜ BÖLÜM

TEDARİK ZİNCİRİ PLANLAMASINDA KARŞILAŞILAN BELİRSİZLİKLER

1. TEDARİK ZİNCİRİNDE BELİRSİZLİKLERİN İNCELENMESİ	43
2. BULANIKLIK KAVRAMI	48
a. Bulanık Mantık.....	48
b. Bulanık Küme Teorisi.....	48
c. Bulanık Kümelerde Temel Kavramlar.....	49
ç. Üyelik Fonksiyonları	52
d. Bulanık Kümelerde Temel İşlemler	53
3. BULANIK ORTAMDA KARAR VERME.....	55
4. BULANIK HEDEF PROGRAMLAMA.....	56
5. TEDARİK ZİNCİRİ ÜRETİM/DAĞITIM PLANLAMASINDA BULANIK KÜME TEORİSİNİN KULLANILDIĞI ÇALIŞMALAR	58

DÖRDÜNCÜ BÖLÜM

ÖNERİLEN BÜTÜNLEŞİK TEDARİK ZİNCİRİ PLANLAMA MODELİ

1. PROBLEM TANIMI	62
2. ÖNERİLEN MATEMATİKSEL MODELİN BULANIK HEDEF PROGRAMLAMA FORMÜLASYONU	69
a. Bulanık Talep Kısıtının Çözüm Yöntemi	71
b. Bulanık Çok Amaçlı Programlama Modelinin Çözüm Yöntemi	73
c. Önerilen Modelin Çözüm Algoritması	75
2. MODELİN TEST EDİLMESİ	76
a. Örnek Problem Uygulaması	76
b. Sayısal Analiz	80
c. Test Problemlerinin Üretilmesi	82
3. ÖNERİLEN SEZGİSEL YÖNTEM	92
a. Tavlama Benzetimi Algoritması	92
b. Önerilen Sezgisel Yöntemin Adımları	94
c. Deneysel Çalışma	99
(1) Parametrelerin Belirlenmesi	99
(2) Sayısal Sonuçlar	104

BEŞİNCİ BÖLÜM

SONUÇ VE ÖNERİLER

1. SONUÇ	109
2. GELECEKTE YAPILACAK ÇALIŞMALAR İÇİN ÖNERİLER	110
KAYNAKÇA	111
EKLER	119

TABLÖLAR

	<u>Sayfa</u>
Tablo-1: Modellerin Karakteristikleri	41
Tablo-2: Modellerin Çözüm Teknikleri	42
Tablo-3: Belirsizlik Özelliklerinin Sınıflandırılması	44
Tablo-4: Üretim Sistemlerindeki Belirsizlik Modellerinin Sınıflandırılması	45
Tablo-5: Belirsizlik Altında Üretim Planlama Modelleri	46
Tablo-6: Tesislerin Ürün Kapasiteleri (Birim)	77
Tablo-7: Maliyet Parametreleri (TL.)	77
Tablo-8: Amaç Fonksiyonlarının Çözümleri ve Aralık Değerleri	79
Tablo-9: Bulanık Amaçların Alt ve Üst Sınır Değerleri	79
Tablo-10: Örnek Problem için Önerilen Matematiksel Modelin Çıktı Sonuçları ..	82
Tablo-11: Test Problemi Sınıflandırma Aralıkları	84
Tablo-12: Test Problemi Boyutları	85
Tablo-13: I. Tip Örnek Test Problemlerinin Hesaplama Sonuçları	87
Tablo-14: II. Tip Örnek Test Problemlerinin Hesaplama Sonuçları	88
Tablo-15: III. Tip Örnek Test Problemlerinin Hesaplama Sonuçları	89
Tablo-16: IV. Tip Örnek Test Problemlerinin Hesaplama Sonuçları	90
Tablo-17: Tavlama Sürecinin Optimizasyon Problemlerinde Karşılığı	93
Tablo-18: Sayısal Sonuçlar	105

ŞEKİLLER

	Sayfa
Şekil-1: Tedarik Zinciri Süreci.....	6
Şekil-2: Tedarik Zinciri Yapısı	8
Şekil-3: Tedarik Zinciri Ağ Yapısı.....	9
Şekil-4: Tedarik Zinciri Yönetiminde Kara Seviyeleri	11
Şekil-5: Ekonomik Sipariş Modeli (Yok Satmasız)	19
Şekil-6: Siparişlerin Gecikmeli Olarak Karşılandığı ESM Modeli	21
Şekil-7: Ekonomik Üretim Miktarı Modeli (Yok Satmasız)	23
Şekil-8: Ekonomik Üretim Miktarı Modeli (Yok Satmalı)	25
Şekil-9: Tedarik Zinciri Planlamasında Karşılaşılan Belirsizlikler.....	44
Şekil-10: X Evrensel Kümesinde $\tilde{A}$ Bulanık Altkümesi.....	49
Şekil-11: Çan Şeklinde $\tilde{A}$ Bulanık Sayısının Grafiği	51
Şekil-12: Üçgensel $\tilde{A}$ Bulanık Sayısının Grafiği.....	52
Şekil-13: Trapezoid $\tilde{A}$ Bulanık Sayısının Grafiği	52
Şekil-14: S Şeklinde $\tilde{A}$ Bulanık Sayısının Grafiği	53
Şekil-15: Önerilen Matematiksel Modelin Tedarik Zinciri Ağ Yapısı	64
Şekil-16: Üçgensel Bulanık Sayı $\mu_{(d_{it})}$ Grafiksel Gösterimi.....	72
Şekil-17: Doğrusal Monoton Azalan $\mu_{f_g(Z_g)}$ Grafik Gösterimi.....	74
Şekil-18: Önerilen Matematiksel Modelin Akış Diyagramı.....	76
Şekil-19: Test Problemi Üretme Kaba (Pseudo) Kodu	83
Şekil-20: TB Sezgiseli için Probleme Özgü Örnek Çözüm Gösterimi	94
Şekil-21: Komşu Çözümlerin Üretilmesi	96
Şekil-22: TB Algoritmasının Sezgiseli Adımları.....	99
Şekil-23: 1. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (15-15-10-20-10)	100
Şekil-24: 2. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (15-15-10-20-10)	101
Şekil-25: 1. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (20-20-20-30-15).....	101

Şekil-26: 2. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15).....	102
Şekil-27: 1. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10).....	102
Şekil-28: 2. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10).....	103
Şekil-29: 1. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15).....	103
Şekil-30: 2. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15).....	104
Şekil-31: GAMS/DICOPT Çözücüsü ile Önerilen Sezgisel Yöntemin Elde Ettiği Çözümlerin Karşılaştırılması.....	106
Şekil-32: Önerilen TB Tabanlı Sezgisel Yöntemin, GAMS/DICOPT Çözücüsü ile Elde Edilen Yaklaşık Değerlerden Sapma Oranları.....	107
Şekil-33: Test Problemlerinin Önerilen Sezgisel Yaklaşım ile Elde Edilen Çözüm Süreleri.....	107

KISALTMALAR

AS	: Alt Sınır
BEÜM	: Bulanık Ekonomik Üretim Miktarı
BHP	: Bulanık Hedef programlama
DOP	: Doğrusal Olmayan Programlama
ESM	: Ekonomik Sipariş Miktarı
EÜM	: Ekonomik Üretim Miktarı
GA	: Genetik Algoritma
HM	: Hazırlık Maliyeti
GIS	: Coğrafi Bilgi Sistemi
SAM	: Satın Alma Maliyeti
SVM	: Sipariş Verme Maliyeti
TB	: Tavlama Benzetimi
TYSP	: Tesis Yeri Seçimi Problemi
TZ	: Tedarik Zinciri
TZY	: Tedarik Zinciri Yönetimi
ÜKM	: Üretim Kaybı Maliyeti
ÜM	: Üretim Maliyeti
ÜS	: Üst Sınır
YM	: Yok Satma Maliyeti

ÖNSÖZ

Günümüz koşulları, küreselleşme ve rekabetin artması nedeniyle işletmelerin ürün fiyatlarını, dolayısıyla maliyetlerini ve verimliliklerini daha iyi kontrol etmelerini zorunlu kılmaktadır. Birden fazla işletmeyi kapsayan tedarik zinciri yapısı, tek bir işletme gibi davranarak kaynakların ortak kullanımı sonucu bir sinerji yaratmayı hedeflemektedir. Sonuç ise yüksek kalitede, düşük maliyetle, piyasaya hızlı bir şekilde sunulan ve müşteri memnuniyeti sağlayan hizmet ya da ürünler olmaktadır.

Tedarik zincirlerinde ağ yapısı sadece birçok firmanın, tesisin, organizasyonun ve bunlar arasındaki ilişkilerin bir arada gösterildiği bir yapı değildir. Günümüz küresel sistemlerinde firmaların rekabet avantajı elde etmek için işbirliği yapmalarını sağlayan bir yönetim şeklidir. Artık rekabet firmalar arasında değil tedarik ağları arasında yaşanmaktadır. Firmaların rekabet avantajı elde etmeleri, içinde bulundukları ağların rekabet gücüne bağlıdır.

Geleneksel tedarik zinciri ağı tasarım problemleri genellikle tek amaçlı olarak ele alınmıştır. Ancak, gerçek hayatta tedarik zincirleri birbirleri ile çelişen amaçlara sahip organizasyonların meydana getirdiği karmaşık ağlardır. Bu durum, nicel ve nitel amaçlar arasındaki etkileşimi açıkça dikkate alan, bütünsel tedarik, üretim, dağıtım planlama kararlarında çok amaçlı bir yapı kullanılmasını gerekli kılmaktadır.

Tedarik zincirleri aynı zamanda gerçek hayatta belirsiz bir ortamda faaliyet göstermektedirler. Bu belirsizlik amaçlara ait hedef değerlerinde, ürün fiyatlarında ve müşteri taleplerinde olabilmektedir. Ancak bugüne kadar geliştirilen modellerde ya belirsizlik dikkate alınmamış veya olasılık metotları kullanılarak yaklaşık çözümler bulunmaya çalışılmıştır. Olasılık dağılımlarının oluşturulduğu verilerin çoğunlukla yetersiz olması veya kesin olmaması karar vericileri subjektif değerlendirmeler yapmak zorunda bırakmaktadır. Bulanık matematiksel programlamada belirsiz parametrelerin, bilinen eğilim özelliklerine sahip herhangi bir olasılık dağılımına

uymak zorunda olmaması nedeniyle, bulanık küme teorisi bu anlamda belirsizliklerin tanımlanmasında uygun bir yapı oluşturmaktadır.

Bu tez çalışmasında, belirsizlik altında, malzeme ihtiyaç kısıtlı, çok ürünlü, çok aşamalı ve çok dönemli bir tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için bütünleşik tedarik zinciri planlama modeli önerilmiştir. Önerilen modelde, karar vericilerin kesin olmayan hedef değerleri ile belirsiz müşteri taleplerini temsil etmek üzere bulanık hedef programlama yaklaşımı kullanılmıştır. Ayrıca, tedarik zinciri literatüründeki benzer çalışmalardan farklı olarak Ekonomik Üretim Miktarı (EÜM) modeline göre sipariş verme, stokta tutma ve yok satma maliyetlerine yer verilmiş, model, GAMS/DICOPT optimizasyon programı kullanılarak çözülmüştür. NP-Zor problemler sınıfına giren modelin karmaşık yapısından dolayı, problem boyutu büyüdükçe çözüm zorlaşmakta ve hatta imkansız hale gelmektedir. Bu nedenle, büyük boyutlu problemlere de kabul edilebilir çözümler üretebilmek için Tavlama Benzetimi Algoritması tabanlı bir sezgisel yöntem geliştirilmiş ve Borland C++ Builder programlama dili ile kodlanmıştır. Bu yazılımdan elde edilen sonuçlar, GAMS/DICOPT optimizasyon yazılımından elde edilen problemler ile karşılaştırılarak geliştirilen sezgisel yöntemin etkinliği test edilmiştir. Çalışmada sunulan uygulama sonuçları, bulanık modelleme ve çözüm yaklaşımlarının daha gerçekçi tedarik zinciri modelleri oluşturulmasında kullanılabileceğini göstermiştir.

Bu tez çalışmasında geliştirilen matematiksel model, herhangi bir işletmede olduğu gibi, aynı zamanda TSK'nın tedarik, üretim, ulaştırma (taşıma), ikmal (dağıtım), bakım vb. faaliyetlerinin de daha etkin bir tedarik zinciri ağ yapısının oluşturulmasında kullanılabilir. Böylece, TSK'da da yüksek kalitede, düşük maliyette, muharebe sahasına hızlı bir şekilde sunulan ve her seviyedeki asker için müşteri memnuniyeti sağlayan hizmet ya da ürünler elde edilebilir.

GİRİŞ

Günümüz koşullarında hızlı gelişen teknoloji ile ortaya çıkan yoğun rekabet, işletmeleri düşük maliyet ve yüksek müşteri memnuniyeti doğrultusunda çalışmaya zorlamaktadır. Şirketlerin günümüzün rekabetçi iş ortamında örgütsel performanslarını artırmak için uygulayacakları stratejilerinin başarısı; yüksek müşteri memnuniyetini sağlamaya yönelik olarak, tedarikçiler ve müşteriler ile karşılıklı güven ve işbirliğine dayalı çalışmaların geliştirilmesine bağlıdır. Özellikle yüksek müşteri memnuniyetini sağlamak “Tedarik Zinciri Yönetimi (TZY)”ni ve buna paralel olarak “Lojistik” kavramını ön plana çıkarmıştır. Burada tedarik zinciri ve lojistik konuları; hangi tesislerin hangi ürünleri üreteceği, bu tesislerin nerelerde kurulacağı, üretilen ürünlerin en az maliyet ve buna paralel olarak en yüksek müşteri memnuniyetini sağlayacak şekilde hangi kanallardan ulaştırılacağı alanında yoğunlaşmıştır. TZY içinde de bu bahsedilen nedenlerden dolayı tesis yeri seçimi problemleri ayrı bir önem taşımaktadır. Çünkü tesislerin nereye kurulacağı bir işletme için stratejik bir karardır.

En genel anlamıyla düşünüldüğünde bir işletmenin amacı, ürettiği bir ürün veya hizmeti tüketicisine ulaştırıp hizmet verdiği alanda gelir sağlamaktır. Bu alanda sayıları sürekli artan rakipler ile artan rekabet koşullarında ayakta kalma ve rekabet üstünlüğünün sağlanabilmesi için işletmelerin ellerindeki kaynakları en yüksek verimlilikte, en yüksek kalitede, en düşük maliyetle sağlaması gerekmektedir. İşletmeler rekabet şartlarında değişen müşteri taleplerine uygun esnek bir üretimi gerçekleştirmek ve üretimden son müşteriye kadar uzanan tedarik zinciri içindeki aksaklıkları gidermek zorundadır.

Tedarik zinciri, hammaddenin kabul edilmesinden, ara ve bitmiş ürünlere dönüştürülmesine, bitmiş ürünlerin müşterilere dağıtılmasına kadar işlemler kümesinin yürütüldüğü bir tesisler (üretim tesisleri, dağıtım merkezleri, depolar vb.) ağıdır (Bramel ve Simchi-Levi, 1997).

Tedarik zinciri ağının tasarlanması, bu ağların ana alt sistemlerinin karmaşıklığı, bu alt sistemler arasındaki etkileşimler ve çok sayıda amaç

fonksiyonunun varlığı gibi dış faktörlerin etkileri nedeniyle güç bir görevdir (Gümüş ve diğ., 2009). Bu karmaşıklık, geçmişte bu alandaki araştırmaların çoğunu tedarik zinciri ağının sadece bir parçasına odaklanmayı zorunlu kılmıştır. Bununla birlikte, son zamanlarda dikkatler artan bir şekilde tedarik zinciri ağının performans, tasarım ve analizinin bir bütün olarak ele alınmasına yoğunlaştırılmaktadır.

Yatırımcılar, ulaştırma, hammadde ve dağıtım maliyetlerinin yüksek olduğu günümüzde, yüksek hizmet seviyesi ile müşterileri tatmin etmek zorundadırlar. Günümüzde tedarik zincirinin farklı amaçları olmasına karşın, geleneksel tedarik zincirinde satın alma, üretim, dağıtım, planlama ve diğer fonksiyonlar karar vericiler tarafından birbirinden bağımsız olarak ele alınmıştır. Pazarlardaki küresel risklerin üstesinden gelebilmek için karar vericiler, farklı amaçların (üretim/dağıtım, tesis yeri seçimi, satın alma, sipariş verme, stokta tutma ve yok satma maliyetlerinin en azlanması vb.) birbirleri ile entegre edildiği bir mekanizma geliştirmek zorundadırlar.

Gerçek dünya ortamındaki kararlar, çoklu, birbiriyle çatışan ve yetersiz kriterlerin bulunduğu durumda verilir. Tedarik zincirleri birbirleri ile çatışan amaçlara sahip organizasyonların meydana getirdiği karmaşık ağlardır. Bu durum, nicel ve nitel amaçlar arasındaki etkileşimi göz önüne alan, bütünsel tedarik, üretim ve dağıtım planlama kararlarında çok amaçlı bir yapı kullanılmasını gerekli kılmaktadır (Selim ve diğ., 2004). Özellikle, taktik ve stratejik seviyedeki birçok karar problemi, örneğin stratejik planlama problemi birçok çatışan amaç ve ölçütü içeren modelleri açıkça göz önüne almak zorundadır.

TZY, hammadde tedariki aşamasından başlayarak kendi içerisinde diğer başka işlemler ve depolama faaliyetleri yoluyla çok sayıda bitmiş ürünlerin müşterilere doğru hareketinin yönetildiği işlevsel bir yaklaşımı gerektirir. Bir tedarik zinciri yöneticisi, böylece, tedarik zincirindeki bir grup şirket tarafından üstlenilen temel iş süreçlerinin etkili bir entegrasyonu ve senkronizasyonunu başarmayı amaçlar. Gelişmiş altyapı ve bilgi teknolojisindeki devrim sayesinde daha duyarlı tedarik zinciri ağına sahip olan şirketler müşteri memnuniyetini daha iyi seviyeye getirmeye başladılar. Bu nedenle, planlama ve tedarik zincirinden en iyi şekilde yararlanmak birçok şirket için öncelikli hedef haline gelmiştir (Shapiro, 2007).

Bütün tedarik zincirinin optimizasyonu etkili planlama kararları ile başarılabilir. Stratejik seviyede verilen tipik kararlar, üretim ve/veya depolama tesislerinin yerleştirilmesi ile ilgilidir. Üretim/dağıtım ağlarının yerleştirilmesi ve yapılandırılması uzun yıllardır çalışılıyor olsa da, çok sayıda önemli gerçek dünya problemleri henüz yeterince ele alınmamıştır. Ayrıca, son birkaç yıl öncesine kadar, tedarik zinciri ağı genellikle en fazla iki kademeye kadar olan tesisler, bu kademelerle talep noktaları arasında bir dağıtım kanalları sistemi ve göreceli olarak basit maliyet yapısı olan tipik bir ağaç yapısından oluşmaktadır. Her ne kadar bu konular literatürde ayrı ayrı ele alınmışsa da, tedarik zinciri planlamasındaki çeşitli endüstriyel projelerden elde edilen tecrübeler, şirketlerin bütün yukarıda belirtilen konuların tedarik zinciri ağına hep birlikte incelenmesi gerektiğini göstermektedir.

Tedarik zinciri ile ilgili yapılan birçok çalışmada oluşturulan deterministik modeller belirsiz parametrelerin kesin olarak bilindiğini varsayar ve belirsizliğin bulunduğu ortamda birçok maliyet ve talep tatmini arasındaki ödünleşmeyi gerçekçi olarak yakalayamaz. Belirsizlik altında tedarik zinciri ile ilgili yapılan çalışmalar sınırlı ve hala gerçek hayat problemlerini yeterince yansıtmaktan uzaktır. Bununla birlikte; birden fazla amacın çatışan doğası ile çevresel katsayılar ve parametrelerdeki bilginin belirsizliği nedeniyle, tedarik zincirindeki birçok üretim/dağıtım planlama karar problemleri için geleneksel deterministik yöntemler, etken çözümler elde edebilmek için uygun değildir.

Bu tez çalışmasında, belirsizlik altında, malzeme ihtiyaç kısıtlı, çok ürünlü, çok aşamalı ve çok dönemli bir tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için tedarik zinciri planlama modeli önerilmiştir. Ayrıca, tedarik zinciri literatüründeki benzer çalışmalardan farklı olarak Ekonomik Üretim Miktarı (EÜM) modeline göre sipariş verme, stokta tutma ve yok satma maliyetlerine yer verilmiştir.

Karar vericilerin kesin olmayan hedef değerleri ile belirsiz müşteri taleplerini modele dahil edebilmek için bulanık hedef programlama yaklaşımı kullanılmış, model, GAMS/DICOPT optimizasyon programı kullanılarak çözülmüştür. Bütünleşik tedarik zinciri ağına tesis yeri seçimi problemlerinin karmaşıklığı NP-Zor problemler sınıfına girmektedir. Modelin karmaşık yapısından

dolayı, problem boyutu büyüdükçe çözüm zorlaşmakta ve hatta imkansız hale gelmektedir. Bu nedenle, büyük boyutlu problemlere de kabul edilebilir çözümler üretebilmek için Tavlama Benzetimi Algoritması tabanlı bir sezgisel yöntem geliştirilmiş ve Borland C++ Builder programlama dili ile kodlanmıştır. Bu yazılımdan elde edilen sonuçlar, GAMS/DICOPT optimizasyon programından elde edilen sonuçlar ile karşılaştırılmış, geliştirilen sezgisel yöntemin etkinliği test edilmiştir. Çalışmada sunulan uygulama sonuçları, bulanık modelleme ve çözüm yaklaşımlarının daha gerçekçi tedarik zinciri modelleri oluşturulmasında kullanılabileceğini göstermiştir.

Çalışmanın giriş bölümünde, çalışmanın bütününe temel teşkil edecek ve okuyucuya TZY'ne ilişkin terminolojiyi açıklayıcı kısa tanımlara yer verilmiş, ikinci bölümde tedarik zinciri ağında tesis yeri seçimi problemi incelenmiştir. Üçüncü bölümde tedarik zinciri üretim/dağıtım planlamasında karşılaşılan belirsizlikler ve modelleme yaklaşımları incelenmiştir. Dördüncü bölümde önerilen bütünleşik tedarik zinciri ağında tesis yeri seçimi problemi için bulanık çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeli sunulmuş, modelin bulanık hedef programlama formülasyonu, çözüm yöntemi ve aşamaları açıklanmış, örnek bir problem üzerinde uygulama ve sayısal analiz yapılmıştır. Bu bölümde ayrıca, önerilen sezgisel yaklaşım ve deneysel çalışma sunulmuştur. Beşinci bölümde, sonuç ve ileride yapılabilecek çalışmalar anlatılmıştır.

BİRİNCİ BÖLÜM

TEDARİK ZİNCİRİ YÖNETİMİ

1. TEDARİK ZİNCİRİ KAVRAMI

a. Tanımlar

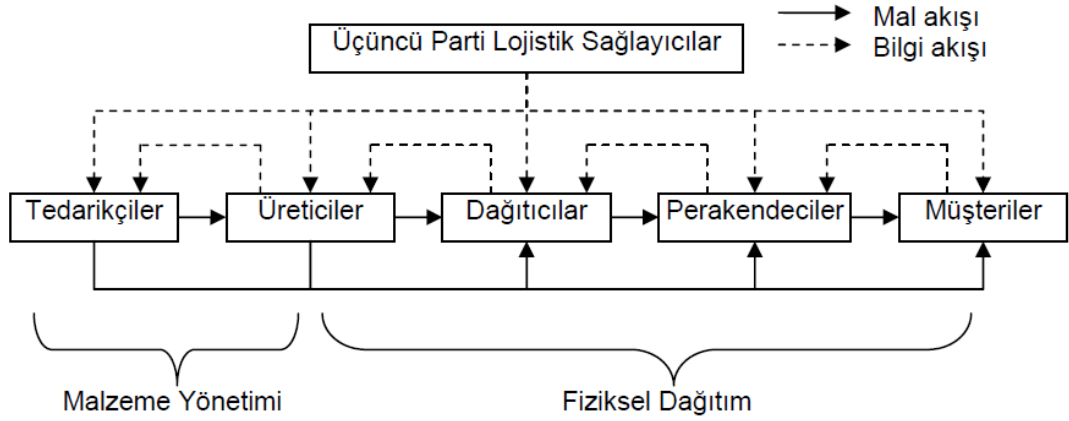
Bir üretim işletmesinde tedarik zinciri, mal ve hizmetlerin tedarik aşamasından, üretimine ve tüketiciye ulaşıncaya kadar birbirini izleyen tüm faaliyetlerin ve fonksiyonların bütünüdür. Basit bir tedarik zinciri, ara ürün tedarikçi, üretici, dağıtıcı ve müşteri olmak üzere dört bileşenden meydana gelir.

Tedarik zinciri kavramı son ürünün üretilmesi ve dağıtımı (tedarikçinin tedarikçisinden müşterinin müşterisine kadar) ile ilgili bütün çabaları kapsar. Bu çabalar planlama, (tedarik ve talebin yönetimi), kaynak (hammadde ve yarı ürünler) temini, üretim (imalat ve montaj), teslimat (depolama ve stok takibi, sipariş alımı ve yönetimi, bütün kanal boyunca dağıtım ve müşteriye teslim) olmak üzere dört temel süreçten oluşur. Bir başka tanıma göre tedarik zinciri;

- Hammadde ve malzemeler ile yarı ürünleri tedarik etmek,
- Hammadde ve malzemeler ile yarı ürünleri bitmiş ürünlere dönüştürmek,
- Bu ürünlere değer kazandırmak,
- Bu ürünleri perakendecilere veya müşterilere dağıtmak ve tutundurmak,
- Farklı süreçlerde faaliyet gösteren birimler (tedarikçiler, üreticiler, toptancılar, lojistik hizmet sağlayıcılar ve perakendeciler) arasındaki bilgi alışverişini kolaylaştırmak için faaliyet gösteren birbiriyle ilişkili bir dizi iş sürecini senkronize eden bütünlük bir sistemdir (Min ve Zhou, 2002).

Tedarik Zinciri Yönetimi (TZY) kavramı 1990'lerden günümüze kadar giderek artan bir ilgi görmektedir. Bunun en büyük nedenlerinden biri TZY'nin sunduğu bütünlük yönetim sisteminin tüm zincirdeki ortakların ihtiyaçlarını karşılamada ve memnuniyetlerini artırmada etkin bir yaklaşım sunmasıdır. Ayrıca

geniş kapsamından dolayı, TZY'nin amaçlarına ulaşmak için tek bir dönemlik veya tek bir projelik çalışmalar yeterli değildir (Kabak, 2008). Bir tedarik zinciri, Şekil-1'de gösterildiği gibi malzemenin ileriye doğru ve bilginin geriye doğru akışı olarak karakterize edilebilir.



Şekil-1: Tedarik Zinciri Süreci (Min ve Zhou, 2002)

Tipik bir tedarik zincirinde yarı ürün ve/veya hammaddeler üretim tesislerinde bitmiş ürünlere dönüştürülür, ürünler depolanır ve/veya dağıtıcılar veya perakendeciler aracılığı ile veya doğrudan son kullanıcıya (müşteri) ulaştırılır. Tedarik zinciri genel olarak; tedarikçiler, üretim tesisleri, depolar, dağıtım merkezleri, satış noktaları (müşteri merkezleri) ile bu birimler arasında akan hammadde, iç stoklar ve bitmiş ürünlerden oluşan bir ağ yapısı oluşturur. Bu yapıda organizasyonların odaklandıkları iki önemli performans ölçütü mevcuttur; maliyetler ve hizmet düzeyi. Firmalar maliyetlerini azaltarak ve hizmet düzeylerini geliştirerek kalıcı rekabet üstünlüğü elde etmek için tedarik zincirinin etkin yönetilmesine ihtiyaç duyarlar. Burada devreye TZY girer (Simchi-Levi ve diğ., 2003).

Literatür incelendiğinde, TZY ile ilgili birbirinden farklı çok sayıda tanımın yapılmış olduğu görülmektedir. Burada tedarik zinciri konusu ile ilgilenen hemen herkesin üzerinde hemfikir olduğu iki tanım verilerek TZY denildiğinde temel olarak ne anlaşılması gerektiği ortaya konacaktır. TZY, Simchi-Levi ve diğ. (2003) tarafından şu şekilde tanımlanmıştır: TZY; hedeflenen müşteri hizmet düzeyini en düşük maliyetle karşılamak, doğru zamanda doğru miktarda ve doğru yerde mal üretimi ve dağıtımını sağlamak üzere tedarikçiler, üreticiler, dağıtıcılar ve

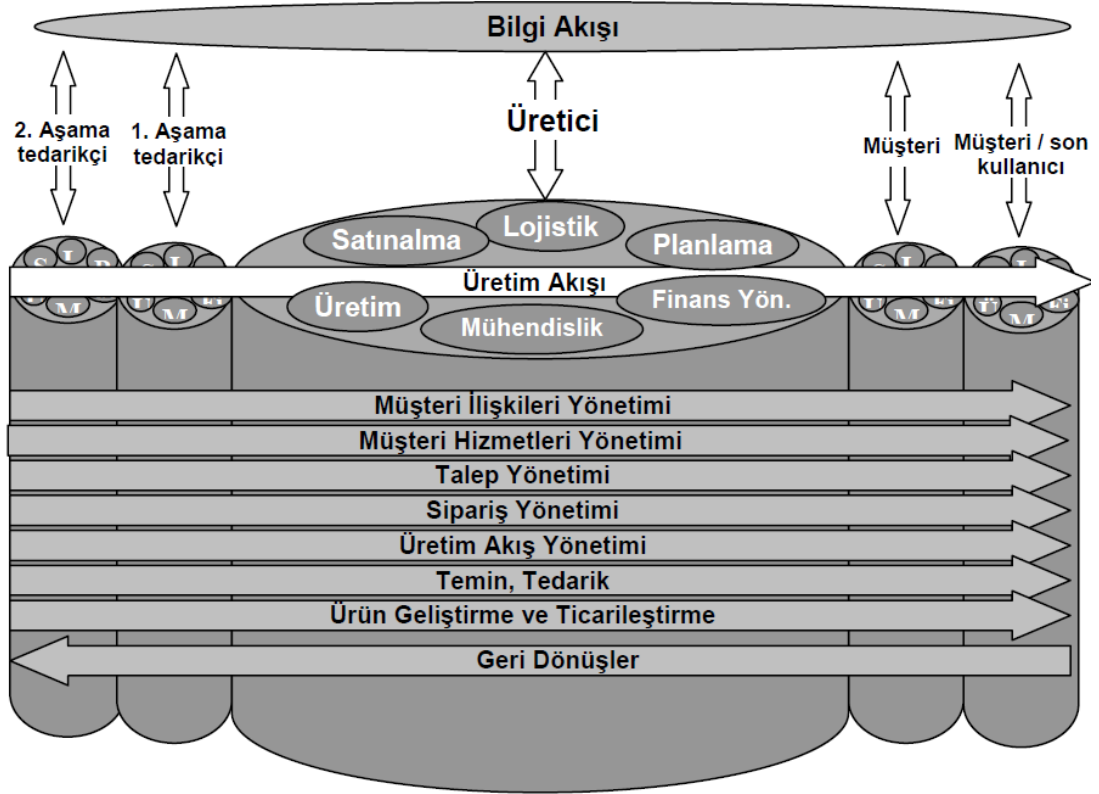
perakendecileri en verimli şekilde entegre eden bir yaklaşımdır. Tedarik zinciri ağı etkin bir şekilde bütünleştirilirse sistem maliyetini en küçüklemek ve hizmet düzeyi gereklerini sağlamak için mallar doğru miktarlarda, doğru yerlerde ve doğru zamanda üretilebilir ve dağıtılabılır.

Yukarıdaki tanım TZY ile ilgili temel özellikleri ortaya koymaktadır. İlk olarak TZY’nde maliyete ve müşteri hizmet düzeyine etki eden her tesis dikkate alınır. İkincisi TZY’nin amacı tüm sistem açısından etkinliği ve en düşük maliyeti sağlamaktır. Son olarak da TZY’de zincirdeki tüm noktaların en iyi şekilde bütünleştirilmesi amaçlandığından stratejik, taktik ve operasyonel tüm faaliyetler TZY kapsamına girer (Melo ve diğ., 2005).

Stock ve Lambert (2001), TZY’ni ilk tedarikçiden son tüketiciye kadar kilit iş süreçlerinin bütünleşmesi olarak tanımlamış ve bu bütünleşmenin özellikle müşteri ve paydaşlar için değer yaratan ürünleri, hizmetleri ve bilgileri sağlaması üzerinde durmuştur. Tanımda sözü edilen kilit iş süreçleri şunlardır:

- (1) Müşteri ilişkileri yönetimi
- (2) Müşteri hizmetleri yönetimi
- (3) Talep yönetimi
- (4) Sipariş yönetimi
- (5) Üretim akış yönetimi
- (6) Temin, tedarik
- (7) Ürün geliştirme ve ticarileştirme
- (8) Geri dönüşler

Şekil-2’de sekiz kilit iş süreciyle birlikte ürün akışları ve bilgi bağlantıları gösterilmiştir. Bilgi akışı bir süreç olmamasına rağmen tedarik zinciri için kilit kavramlardan biridir. Unutulmamalıdır ki ürün akışları ancak bilgi akışları ile başlatılabilir.

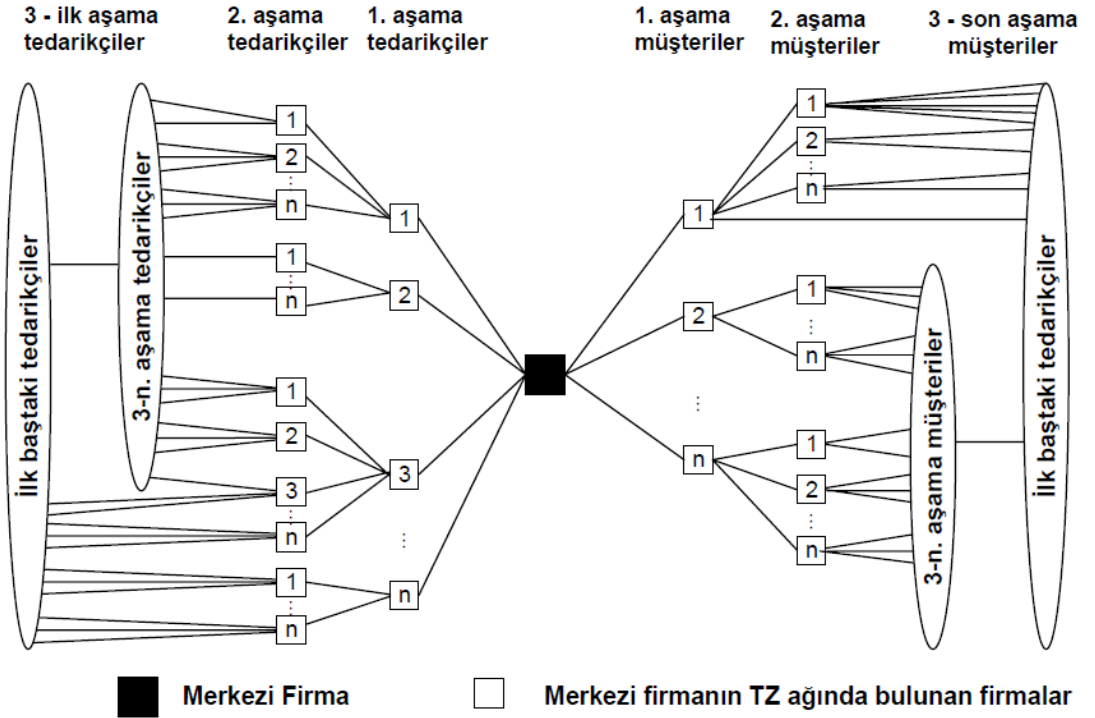


Şekil-2: Tedarik Zinciri Yapısı (Stock ve Lambert, 2001)

Yukarıda yapılan tanımlardan; tedarikçi, üretici, dağıtım merkezi ve müşteriden oluşan tedarik zincirinde birçok firma, tesis ve organizasyonun yer aldığı anlaşılmaktadır. Bunlar arasındaki ilişkiler ve akışlar (ürün ve bilgi) ise TZY konusuna girmektedir. Bu şekilde düşünüldüğünde tüm tedarik zincirinin bir ağ yapısı oluşturduğu sonucuna varılabilir.

b. Tedarik Zinciri Ağ Yapısı

Tedarik zinciri, ilk tedarikçiden hammadde alımından ürünün son müşteriye ulaştırılmasına kadar tüm ilişkileri ve akışları dikkate aldığı için oldukça karmaşık bir yapıya sahiptir. Eğer tedarik zincirindeki her birim bir düğüm, bu birimlerin birbirleriyle ilişkileri yay olarak düşünülür ve birimler arası akışlar yaylar ile ifade edilirse; tedarik zinciri, bir ağ yapısı olarak görülebilir. Şekil-3'te genel bir tedarik zinciri ağ yapısı verilmiştir. Bu ağ yapısı incelendiğinde ilk tedarikçiden son kullanıcıya kadar olan tüm sürecin yönetiminin ne kadar karmaşık olduğu görülmektedir.



Şekil-3: Tedarik Zinciri Ağ Yapısı (Stock ve Lambert, 2001)

İşletmelerin ağ yapıları lojistik işlemlerin etkinliğinin ve verimliliğinin artmasına ilave olarak firmaya piyasada ayırt edici üstünlükler sağlamaktadır. Tedarik zincirinin iyi yönetilebilmesi için ağ yapısının nasıl oluştuğunu anlamak ve bununla ilgili açık bilgiye sahip olmak gerekir. Bir firmanın ağ oluşturma aşamasında üç yapısal yönde inceleme yapılmalıdır. Bunlar; tedarik zinciri üyelerinin belirlenmesi, tedarik zinciri üyeleri veya faaliyetler arasındaki iş süreçleri bağlantılarının ortaya konulması ve ağın yapısal boyutunun (Dikey yapısı, yatay yapısı ve merkezi firmanın zincirin uç noktalarına göre yeri.) analiz edilmesi gerekir (Kabak, 2008).

Tedarik zincirlerinde ağ yapısı sadece birçok firmanın, tesisin, organizasyonun ve bunlar arasındaki ilişkilerin bir arada gösterildiği bir yapı değildir. Günümüz küresel sistemlerinde firmaların rekabet avantajı elde etmek için işbirliği yapmalarını sağlayan bir yönetim şeklidir. Artık rekabet firmalar arasında değil tedarik ağları arasında yaşanmaktadır. Firmaların rekabet avantajı elde etmeleri, içinde bulundukları ağların rekabet gücüne bağlıdır (Kulmala ve diğ., 2004; Lakhal ve diğ., 1999).

c. Tedarik Zincirinin Amacı

Tedarik zincirindeki her bir oyuncunun amacı, en yeni bilgiyi zincirdeki diğer işletmelere sunarak daha mükemmel bir arz ve talep dengesinin sağlanmasına ve ürünün üretim noktasından tüketim noktasına en kısa zamanda ve en düşük maliyetle ulaşmasına katkıda bulunmaktır. Tedarik zincirinin kısa vadeli amacı, gereksiz stokları ortadan kaldırmak ve üretim ile müşteriye yanıt verebilme hızını artırmak iken uzun vadeli amacı, müşteri beklentilerini doğru yerde teslim edilmiş doğru ürünle karşılamak, bu şekilde pazar payını ve karları artırmaktır. Başka bir ifadeyle tedarik zincirinde, zincirden geçen ürünlere değer katmak ve bu ürünlerin istenilen miktarlarda, uygun nitelikte ve istenilen zamanda rekabetçi bir maliyet anlayışıyla dağıtılmasını sağlamak amaçlanmaktadır.

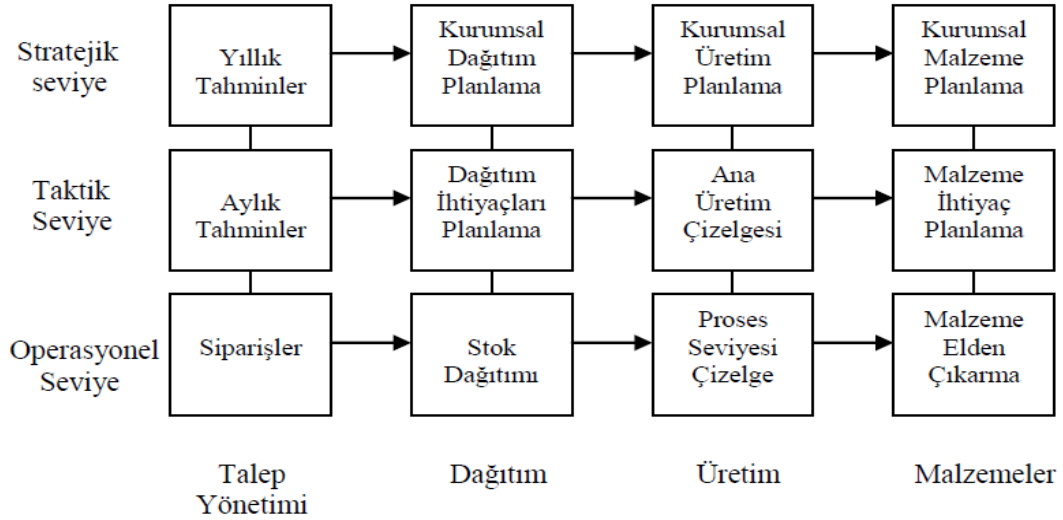
2. TEDARİK ZİNCİRİ YÖNETİMİNDE KARAR SÜREÇLERİ

a. Tedarik Zincirinde Karar Seviyeleri

TZY stratejik, taktik ve operasyonel olmak üzere üç hiyerarşik seviyeden oluşur (Chopra ve Meindl, 2001). Genel bir yaklaşımla, dağıtım sisteminin ve üretim planlama süreçlerinin tasarımına ilişkin konuların stratejik düzeyde ele alındığı, mevcut bir üretim-dağıtım sistemine yönelik optimizasyon problemlerinin ise taktik seviyede ilgilenilen sorunlar olduğu söylenebilir.

Stratejik düzeyde karşılaşılan tedarik zinciri problemleri arasında, yer seçimi kararları, talep planlama, dağıtım kanalı planlaması, stratejik anlaşmalar, yeni ürün geliştirme, dış kaynak kullanımı, tedarikçi seçimi, bilgi teknolojisi seçimi, fiyatlandırma ve ağı yeniden yapılandırılması gibi kararlar sayılabilir. Her ne kadar tedarik zinciri problemlerinin çoğu, yapısı gereği stratejik olsa da bir tedarik zincirinde birçok taktik düzeyde problemle de karşılaşmaktadır. Stok kontrolü, üretim-dağıtım süreçlerinin koordinasyonu, siparişlerle taşıma süreçlerinin bütünleştirilmesi, malzeme işleme tarzı, makine/teçhizat seçimi ve yerleşim düzeninin belirlenmesi gibi problemler, bir tedarik zincirinde taktik düzeyde incelenen karar alanlarıdır. Operasyonel düzeyde ise taktik planlar temel alınarak, araç rotalama/çizelgeleme, işgücü çizelgeleme, malzeme hareketleri, kayıt tutma şekli ve paketleme gibi daha alt düzeydeki kararlar ele alınmaktadır. Bazı tedarik

zinciri problemleri, farklı karar seviyelerini kapsayan hiyerarşik, çok kademeli bir planlama gerektirdiğinden, karşılaşılan sorunların, yukarıda adı geçen karar seviyeleri bazında ayrıştırılması her zaman mümkün değildir (Min ve Zhou, 2002).



Şekil-4: Tedarik Zinciri Yönetiminde Karar Seviyeleri

Stratejik seviyede yapılan bir hatanın, operatif seviyedeki gösterilen çabalar ile düzeltilmeyeceği bir gerçektir. Lojistikten beklenen faydanın sağlanabilmesi için alt fonksiyon sahalarının kesin olarak anlaşılması, stratejik, taktik ve operatif seviyede gerçekleştirilecek olan planlamalarda dikkat edilmesi gereken kriterlerin bilinmesi gerekmektedir.

b. Tedarik Zincirinde Karar Alanları

TZY’nde birçok karar verme problemi söz konusudur. Yukarıda karar seviyelerine göre değinilen bu alanlardan en önemlileri, aşağıda açıklanmıştır.

(1) Yer Seçimi Kararları

Üretim tesislerinin ve depoların sayılarının ve coğrafi konumlarının belirlenmesi, tedarik zinciri oluşturmada ilk karar adımıdır. Tesislerin konumlarının belirlenmesi, uzun dönem planlarında kaynaklara bağlıdır. Bu kararlar firmaların rekabet güçlerini önemli ölçüde etkilemektedir. Yer seçimi kararları, üretim ve taşıma maliyetleri, vergiler, bölgesel bileşenler gibi etkenlere bağlı olarak

tanımlanır. Ayrıca seçilen konumun, firma, tedarikçiler ve müşteriler arasındaki optimum ağı oluşturmaya amaçlanır.

(2) Üretim Kararları

Hangi üründen hangi üretim tesisinde ne kadar üretileceğinin, hangi tedarikçilerden ne kadar malzeme/ara ürün temin edileceğinin ve üretim tesisleri ile depo/dağıtım merkezlerinin kapasitelerinin belirlenmesi gibi kararları içerir. Bu kararların gelir, maliyet, finansman ve müşteri hizmetleri üzerinde önemli etkileri vardır. Bu kararlar verilirken, tesislerin konumları veri olarak alınır. Operasyonel seviyedeki üretim kararları ise detaylı üretim çizelgelemesine dayanır. Taktik düzeyde belirlenen ana üretim çizelgesine uygun bir şekilde, makineler bazında üretim çizelgelerinin oluşturulması ve araç gerecin bakımı gibi süreçlere ilişkin kararları içerir.

(3) Stok Kararları

İşletmelerin stok ve taşıma maliyetlerinin dağıtım ağlarının genişlemesiyle artış göstermesi nedeniyle başta çok kademeli stok yönetimi olmak üzere yeni nesil stok yönetim usulleri geliştirilmektedir. Çok kademeli stok teorisi, sayısal analiz ve optimizasyon yöntemleri ile çeşitli kademelerden oluşan entegre lojistik sistemlerin modellenmesine olanak sağlar. Çok kademeli stok yönetimi stok politikalarını bütünsel olarak optimize eder, talep ve ikmal değişkenliğine stokastik ve deterministik yöneylem tekniklerinden istifade ile açıklık getirir. Çok kademeli stok sistemi tasarımında en önemli konu, her kademede bulundurulmasına ihtiyaç duyulan ürün miktarının belirlenmesidir. Dinamik programlama, çok kademeli stok optimizasyonunda yararlanılabilecek en uygun yöneylem tekniklerindendir.

(4) Taşıma Kararları

Tedarik zinciri boyunca ürünlerin nasıl taşınacağına ilişkin verilmesi gereken kararlar, stok kararlarıyla çok yakından bağlantılıdır. Ürünlerin taşınmasında hangi taşıma tipinin kullanılacağına belirlenmesi, bu alanda ele

alınabilecek kararlardan birisidir. Hava yollarını kullanmak, hızlı, güvenilir ve daha az güvenlik stoğu tutulmasını gerektiren bir yol olmakla birlikte pahalı bir yoldur. Deniz yoluyla veya demiryoluyla taşıma daha ucuz olabilir, ancak bu durumda nakliye sırasında karşılaşılabilecek belirsizliklere karşı tampon görevi görecektir. Yüksek miktarda stok tutulması gerekir. Ayrıca hedeflenen müşteri memnuniyeti düzeyi ile tesislerin, depoların ve müşterilerin coğrafi yerleşimleri, bu kararların alınmasında önemli rol oynar. Ürünlerin taşınmasında kullanılacak araç filosunun büyüklüğünün tespiti, bu araçların rotalarının belirlenmesi ve çizelgelenmesi, araçlarda taşınacak kargo büyüklüğünün ve ürün bileşiminin belirlenmesi, kilit önem taşıyan taşıma kararlarındandır.

(5) Dağıtım Ağı Kararları

Dağıtım ağı tasarımı, stratejik planlamanın uygulandığı ana alanlardan biridir. Bir stratejik dağıtım ağı planı, verilen bir planlama ufku boyunca belirli bir ihtiyaçlar setini karşılamak için geliştirilir. İyi bir plan; doğru malları, doğru miktarda, doğru yerde, doğru zamanda müşteriye sağlamalı ve toplam dağıtım maliyetini en az yapan uygun bir dağıtım ağını tanımlamalıdır. Dağıtım ağı planı, karı ve müşteriye sunulan hizmeti de optimum yaparken açılacak dağıtım merkezlerinin sayısı, yerleri ve hangi müşterilere hizmet vereceklerinin belirlenmesinin yanı sıra kullanılacak taşıma yöntemlerinin seçimi gibi teknik detayları da içermelidir. Depo sayısı arttıkça, teslim maliyeti azalır ve depo maliyeti artar. Bu ifadenin tersi de doğrudur; yani depo sayısı azaldıkça, teslim maliyeti artar. Dolayısıyla, toplam dağıtım maliyetini en az yapmak için depo ve taşıma maliyeti arasında en iyi dengenin bulunması önemlidir.

3. TEDARİK ZİNCİRİNDE STOK KONTROLÜ

Son yıllarda TZY, yöneylem araştırması uygulayıcıları arasında önemli bir araştırma alanı olarak ortaya çıkmıştır. Yöneylem araştırmacıları tarafından, tedarik zincirinin daha verimli ve ekonomik olmasını sağlamak için birçok araştırma gerçekleştirilmiştir. İşlerin düzgün ve verimli yürütülmesi, tedarik zincirinin ana alanlarının düzgün ve verimli çalıştırılmasını gerektirir. Bu ana alanlardan birisi de stok kontrolüdür. İster hammadde, ister proses içi, kısmen veya tamamen bitmiş

ürünler olsun, tedarik zincirinin her aşamasında stoklar vardır. Tedarik zinciri yöneticisi, elde stok bulundurmanın maliyetleri artırması ile karşı karşıya kaldığından, tedarik zincirinde stok kontrolü etkin yönetim için kritik öneme sahiptir.

İşletmelerde stok temel olarak arz talep dengesizliğini karşılamak üzere bulundurulur. Stokların, işletmenin müşteri isteklerine tepki verme hızına büyük etkisi vardır. Aynı zamanda stok, üretim sistemindeki malzeme akış hızını da etkiler. İstenilen stok türünü istenilen zamanda ve istenilen miktarda hazır bulundurma ve bunu en ekonomik biçimde gerçekleştirme faaliyeti stok kontrolü olarak adlandırılır (Behret, 2011).

a. Stok Kontrolü ile İlgili Tanımlar

Stok, olabilecek talebi karşılamak üzere beklenmeyen gecikmeler ve aksamalar durumunda talebin karşılanmasını garanti altına almak amacıyla elde bulundurulan atıl kaynaklardır. Bir üretim sisteminde, üretilen ürüne doğrudan veya dolaylı olarak katılan bütün fiziksel varlıklar ve ürünün kendisi stoku oluşturur. Stoklar; tedârikçiler, üreticiler, çeşitli depolar, toptancı ve perakende mağazaları ve hatta tüketicilerde oluşur. Stoklar genel olarak dört sınıfa ayrılırlar.

(1) Hammadde Stokları: Hammaddeler dışarıdan alınıp üretimde doğrudan kullanılarak son ürünü elde etmede yararlanılan malzemelerdir. Hammadde ile ifade edilen genelde metaller, boya, özel çelikler, kimyasal maddeler vb.dir. Ayrıca motorlar, özel devreler, silindirler gibi organizasyon dışından alınan satın alma parçaları da organizasyonun hammadde stokları içinde yer alır.

(2) Yarı Ürün Stokları: Üretimdeki stok, fabrikanın sistemi içerisinde işlem gören veya işlem görmeyi bekleyen her çeşit malzeme, parça veya montaj parçasından oluşmaktadır.

(3) Ürün Stokları: Bütün üretim aşamalarından geçerek tamamlanmış ürünlerdir. Bu ürünler; toptancı, perakendeci veya diğer satış noktalarına gönderilmek üzere stokta bekletilirler.

(4) İşletme Malzemeleri: Bakım-onarım malzemeleri, yedek parçalar, temizlik malzemeleri ve kırtasiye malzemelerinden oluşur.

b. Stok Kontrolünün Amacı

Stok kontrolünün amacı; işletme bünyesinde, üretim sürecinde kullanılan hammadde ve malzeme girdi ve çıktılarının kayıtlarını, ayrıntılı bir şekilde tutmak, sipariş verilen malzemelerin kayıtlarını tutarak, istenen anda, malzeme sayımlarını gerçekleştirmek ve sonuçları kontrol ederek, aksaklıkları ortaya çıkarmak, uygun bir stok politikası belirleyerek birbiriyle ilişkili olan masrafları en az düzeyde tutmaktır. Üretim sistemi büyüdükçe ve ürün çeşidi arttıkça, tedarik, talep ve üretime ilişkin faktörlerdeki belirsizlik ve aralarındaki ilişkilerin karmaşıklığı, stokta tutmayı zorunlu kılar. Stok kontrolü, malzemelerin ve ürünlerin gerektiği zaman yeterli miktarda sağlanmasını güvence altına almalı ve aşırı stok maliyetlerine engel olmalıdır.

c. Stok Maliyetleri

Stok maliyetleri, stok yönetimi sistemlerinin seçimi veya sipariş verme politikalarının belirlenmesi amacıyla yapılacak değerlendirmelerde ve modellerde, sistemin çalışmasını belirleyecek değişkenlerin hesabı için kullanılan parametrelerdir. Bu amaçla başlıca üç çeşit stok maliyeti tanımlanır.

(1) Sipariş Verme veya Üretim Hazırlık Maliyeti

(a) Sipariş Verme Maliyeti (Ordering Cost): Satın alınan her çeşit malzemenin siparişinin gerektirdiği tedarik işlemleri giderleridir. Örneğin; telefon edilmesi, piyasa araştırması yapılması vb.

(b) Üretim Hazırlık Maliyeti (Setup Cost): Malzeme akış süreci içindeki akışı başlatmak için verilen “iş emri” şeklindeki atölye siparişlerinin gerçekleştirilmesi sırasında ortaya çıkan maliyetlerdir. Örneğin; üretim hattının ve makinanın işe hazırlanması sırasında geçen zamanın fırsat maliyeti, bu amaçla kullanılan personelin maliyeti, takım/kalıp değiştirme maliyeti vb.

(2) **Stokta Tutma Maliyeti:** Ürün veya hammadde olarak stoklarda bekleyen her çeşit malzemenin getireceği malî yükü ölçmek amacıyla kullanılır. Beş çeşit bileşeni vardır: Sermaye maliyeti (finansman getiri oranı), depolama ve elleçleme maliyeti, stok riski (yıpranma, raf ömrü dolma, kayıp ve demode olma) maliyeti, stok hizmet (sigorta, bakım, özel hizmet-soğuk ortam, malzeme karıştırma vd.) maliyeti ve vergi.

(3) **Elde Bulundurmama Maliyeti:** Talep olduğu halde ürün bulunmamasının getirdiği maliyet bir fırsat maliyeti olarak değerlendirilebilir. Elde bulundurmama maliyeti üç türlü oluşabilir.

(a) **Bekleyen Sipariş (Backorder) Maliyeti:** Müşterinin talebinin öngörülen zamandan daha geç karşılanma durumudur. Bu durumda müşteriden yapılacak tahsilat da gecikebilecektir.

(b) **Kayıp Satış (Lost Sales, Backlogging Cost) Maliyeti:** Müşteri talebinin, öngörülen zamanda karşılanamaması nedeniyle müşterinin siparişini iptal etmesi veya bir daha hiç sipariş vermemesi durumuna karşılık gelen maliyettir. Bu durumda söz konusu müşteri kaybedilecektir.

(c) **Üretim Kaybı Maliyeti (ÜKM):** Ürün ve yarı ürün stoklarının bulunmaması nedeniyle ürün talebinin karşılanamaması durumunda ortaya çıkan maliyettir.

ç. **Stok Kontrol Politikaları**

Her işletmenin kendine özgü stok kontrol sistemine ve stok politikalarına ihtiyacı vardır. İşletmelerin büyüklüğü, üst yöneticilerinin politikaları, üretim tipi, mali olanakları vb. etkenlere göre farklı stok kontrol sistemlerine sahip olan işletmeler için temel stok politikaları incelenmiştir.

(1) **Çevrim Stoku Bulundurma Politikası:** Ürünler ve malzemeler, partiler halinde sipariş edilirler. Dolayısıyla tüketim sürerken bir kısım stokta bekler. Bir satın alma veya üretim partisine karşılık gelen ve her parti için

bütünlenen stok miktarı, çevrim stoku olarak düşünülür. Çevrim stoku, işletmelerde ölçek ekonomilerinden ve maliyet indiriminden faydalanmak amacıyla tutulur.

Çevrim stoku bulundurma politikasının temel amacı birim satın alma, sipariş ve stokta tutma maliyetlerini en azlayarak ölçek ekonomisinden faydalanacak kadar stokta tutmaktır. Çevrim stoku ile parti büyüklüğü arasında doğrudan bir ilişki vardır. Çevrim stoku bulundurma politikasının karar problemi en ekonomik sipariş miktarının belirlenmesi problemidir.

(2) Güvenlik Stoku Bulundurma Politikası: Güvenlik stoku, talepteki belirsizlik ve talebin tahmin edileni aşması durumunda ürün yokluğu ile karşılaşılması için elde tutulan bir stoktur. Eğer, bir ürünün ortalama kullanım miktarı ve standart sapması hesaplanabiliyorsa, güvenlik stoku bulunabilir. Güvenlik stoku, yok satma durumuna karşı ekstra bir emniyet valfi gibidir. Güvenlik stokları, yeniden sipariş zamanındaki talebin, beklenen talebi veya gerçekleşen teslim zamanının, beklenen teslim zamanını aşması durumunda talebi karşılamak için ihtiyaç duyulan miktardır. Güvenlik stokunun, firmanın maliyetleri üzerine iki etkisi vardır; yok satma maliyetini düşürür ve stokta tutma maliyetini artırır.

(3) Mevsimlik Stok Bulundurma Politikası: Bir mevsim başlamadan önce mevsim boyunca oluşacak tüketimi karşılamak üzere tutulan stoktur. Örneğin, kırtasiye sektöründe okul öncesi dönemde fazla stok bulundurulur.

(4) En Uygun Müşteri Hizmet Düzeyini Yakalama Politikası: Müşteri hizmet düzeyi, elde bulundurmama maliyetinin tam olarak belirlenemediği durumlarda, müşteri talebini karşılama oranı olarak da kullanılabilir. Örneğin, 100 müşteri talebinin 95'i karşılanmışsa, müşteri hizmet düzeyi %95'dir. Yüksek güvenlik stoğu, yüksek hizmet düzeyi sağlar. Güvenlik stoku özellikle belirsiz tedarik sürelerine yanıt verebilmek amacıyla tutulur. Tedarik sürelerinin yüksek olduğu durumlarda % 100 hizmet düzeyi için sonsuz stok tutmak uygun bir yaklaşım değildir. Bir işletme, farklı malzeme stokları için farklı hizmet düzeylerini hedefleyebilir. Bitmiş ürünler için % 90 hizmet düzeyinin yeterli olup olmadığı veya hammadde yetersizliği nedeniyle boşa geçen üretim sürelerinin hangi düzeyde kabul edilebilir olduğu gibi yanıtlanması zor sorular, hedefleri ortaya koyar. Hizmet

düzeyinin artırılması için stok yatırımlarının artırılması gereklidir. Değişik tiplerde stokları olan büyük bir şirkette gerçekleşen toplam müşteri hizmet düzeyi ile toplam stok yatırımı arasında üssel bir ilişki olduğu söylenebilir.

d. Stok Kontrol Modelleri

Stok kontrol modelleri, stoka olan talebin bağımlı ya da bağımsız olmasına göre tamamen farklı şekilde ele alınır. Bağımlı talep yapısına sahip stok modelleri için malzeme ihtiyaç planlaması sistemleri geliştirilmiş ve günümüze kadar birçok yazılım bu sistemler için tasarlanmıştır (Behret, 2011). Bağımlı talep yapısına sahip sistemler bu tez çalışmasının konusu olmadığı için burada incelenmemiştir.

Bağımsız talep içeren stok modellerinin incelenmesinde talep tahmini önemlidir, çünkü stok yenileme siparişinin miktarı ve zamanı talep tahminine bağlı olarak hesaplanmaktadır. Son ürün talebinin birçok dış etkene göre değişkenlik göstermesi ve belirsizlik içermesi nedeniyle doğru tahminin yapılarak modele girdi oluşturacak talep değerinin belirlenmesi stok kontrolü için hayati önem taşır.

Bağımsız talep içeren stok modelleri stok yenileme politikalarına göre iki gruba ayrılabilir. Bunlardan ilki stokların sürekli olarak izlendiği ve stok yeniden sipariş noktasına geldiğinde yeni bir sipariş verildiği sürekli gözden geçirmeli modeldir. Bu modelde eldeki stok sürekli olarak izlenir ve ne zaman stok belli seviyenin altına düşerse sipariş o zaman verilir. Sürekli gözden geçirmeli modellerde sipariş miktarı sabittir ancak sipariş aralığı talep değişkenliğine bağlı olarak değişir. Bu modeller literatürde sabit sipariş miktarı modeli, ESM modeli ya da (Q,r) modeli olarak da bilinmektedir. Diğer model ise stok durumunun belirli zaman aralıkları ile kontrol edildiği ve stok seviyesini belirli bir eşiğe yükseltmek için sipariş verildiği dönemsel gözden geçirmeli stok kontrol modelidir. Bu modelde, sipariş aralığı sabittir ancak sipariş miktarı talebin değişkenliğine bağlı olarak değişir. Bu modeller ise literatürde dönemsel gözden geçirmeli model, sabit sipariş aralığı sistemi ya da (R,r) modeli olarak bilinmektedir (Behret, 2011).

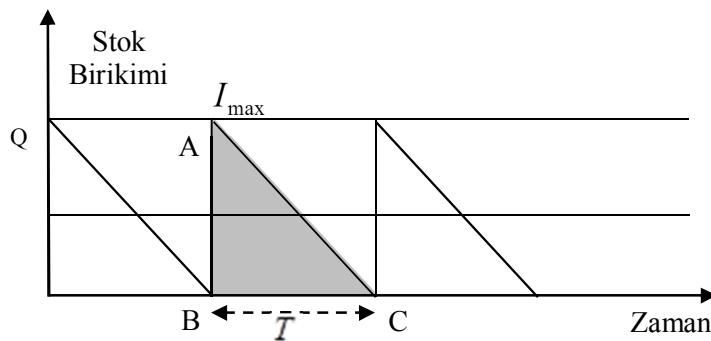
Stok kontrol modelleri ayrıca, dönem sayısına göre statik-dinamik, ürün çeşidine göre tek ürünlü-çok ürünlü, hiyerarşik olarak tek aşamalı-çok aşamalı olmak üzere değişik şekillerde sınıflandırmalara ayrılabilir.

İlk olarak Harris tarafından 1913 yılında tanıtılan ESM modeli ile başlayan matematiksel stok modelleri, talep ve/veya tedarik (temin) süresi bilgilerinin kesin olarak bilindiği ve belirsiz olduğu duruma göre deterministik modeller ve olasılıklı modeller olarak iki grupta incelenebilir.

(1) Deterministik Stok Kontrol Modelleri: Stok kontrolü literatüründe bulunan ilk modellerdir. Deterministik stok kontrol modelleri, genellikle çok az parametre içerir ve bunların kesin olarak bilindikleri varsayılır.

(a) Ekonomik Sipariş Miktarı (ESM) Modeli: Maliyetler arası ilişkileri incelemek amacıyla oluşturulan en temel modeldir. Amaç; ekonomik sipariş miktarının bulunması, buna bağlı olarak siparişler arası sürenin hesaplanması, dönem içinde verilecek sipariş sayısının ve dönem boyu toplam maliyetin bulunmasıdır (Şekil-5). Bu model aşağıdaki varsayımları içerir:

- Talep belirli ve hızı sabittir.
- Siparişler eşit aralıklarla verilir.
- Fiyat sabittir.
- Sipariş edilen mallar bir anda karşılanmaktadır (Tedarik süresi sıfırdır.)
- Yok satmaya izin verilmemektedir.



Şekil-5: Ekonomik Sipariş Modeli (Yok Satmasız)

T : Siparişler arası süredir ve $T = \frac{Q}{D}$, N : Sipariş sayısı ise, $N = \frac{D}{Q}$

formülleri kullanılarak hesaplanır.

Stokta tutma maliyeti, stokta tutulan miktar ile birim stokta tutma maliyetinin çarpılması sonucu elde edilir. Stokta tutulan miktar, Şekil 5'te taralı ABC üçgeninin alanına $\left(\frac{QT}{2}\right)$ eşittir. Stokta Tutma Maliyeti = $C_h \frac{QT}{2}$ olarak bulunur.

Tek dönem için oluşan toplam maliyet; dönem içinde sadece bir kere sipariş verileceğinden bir sipariş maliyeti, sipariş verilen parti için satın alma maliyeti ve dönem boyunca elde tutulan stoklar için stokta tutma maliyetinin toplamıdır.

$$TM(Q) = C_0 + CQ + C_h \frac{QT}{2} \quad (1.1)$$

Yıllık toplam maliyet ise, tek dönemlik toplam maliyetin yıl içerisinde verilecek sipariş sayısı (N) ile çarpılması sonucu bulunur. N yerine D/Q ve T yerine Q/D yazılarak sadeleştirme yapıldığında aşağıdaki maliyet fonksiyonu elde edilir.

$$TM(Q) = \frac{C_0 D}{Q} + CD + C_h \frac{Q}{2} \quad (1.2)$$

Yukarıda verilen toplam maliyet fonksiyonunun sipariş miktarına göre türevi alınır sıfıra eşitlenirse toplam maliyeti en düşük yapan ekonomik sipariş miktarı (Q^*) bulunmuş olur.

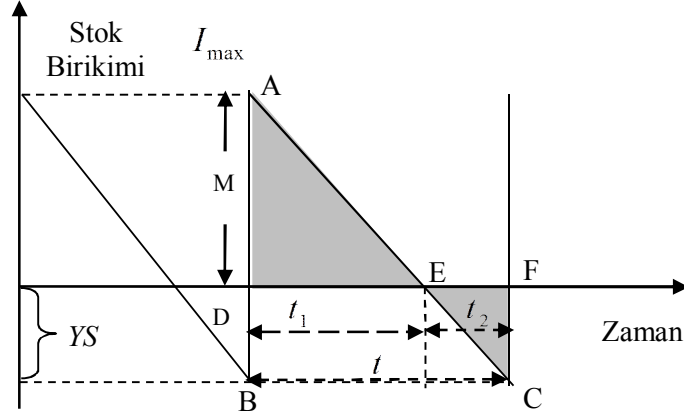
$$\frac{TM(Q)}{\partial(Q)} = \frac{C_0 D}{Q^2} + 0 + \frac{C_h}{2} = 0 \Rightarrow Q^* = \sqrt{\frac{2C_0 D}{C_h}} \quad (1.3)$$

(b) Siparişlerin Gecikmeli Olarak Karşılandığı (Yok Satmalı) Model: Bazı durumlarda, üreticiler ellerinde hiç ürün kalmadığında, malzeme tedarik edilinceye kadar üretimlerini geciktirir ya da müşteri taleplerini geri çevirirler. Burada siparişlerin gecikmeli olarak karşılandığı model (Şekil-6) için

Ekonomik Sipariş Miktarı (Q^*) ve maksimum yok satma düzeyi (b^*) bulunacaktır.

Bu model aşağıdaki varsayımları içerir:

- Talep belirli ve her periyotta sabittir.
- Tedarik süresi kesin olarak bilinmekte ve sıfırdır.
- Fiyat sabittir.
- Müşteri talepleri geciktirilebilir.



Şekil-6: Siparişlerin Gecikmeli Olarak Karşılandığı ESM Modeli

$$\text{Ort. Stokta Tutulan Miktar} = \frac{1}{2}(Q-b)t_1 \quad \text{Ort. Yok Satılan Miktar} = \frac{b}{2}t_2 \quad \text{BEC}$$

üçgeni ile AFC üçgeni benzerdir. Buna göre;

$$\frac{|BC|}{|AC|} = \frac{|BE|}{|AF|} \Rightarrow \frac{Q-b}{Q} = \frac{t_1}{T} \Rightarrow t_1 = \frac{(Q-b)}{Q}T \text{ elde edilir.}$$

$$T \text{ yerine } \frac{Q}{D} \text{ yazılırsa, } t_1 = \frac{Q-b}{D} \text{ olarak bulunur.}$$

EGF üçgeni ile CDF üçgeni benzerdir. Buna göre;

$$\frac{|EG|}{|CD|} = \frac{|GF|}{|DF|} \Rightarrow \frac{t_2}{T} = \frac{b}{Q} \Rightarrow t_2 = \frac{b}{Q}T \text{ elde edilir.}$$

$$T \text{ yerine } \frac{Q}{D} \text{ yazılırsa, } t_2 = \frac{b}{D} \text{ olarak bulunur. Bir dönemlik toplam maliyet;}$$

$$TM(Q, b) = C_0 + CQ + C_h \frac{t_1(Q-b)}{2} + \pi \frac{bt_2}{2} \quad (1.4)$$

Yıllık Toplam Maliyet; bir dönemlik toplam maliyetin sipariş sayısı (N) ile çarpılması sonucu aşağıdaki denklem elde edilir.

$$TM(Q, b) = \frac{C_0 D}{Q} + CD + C_h \frac{(Q-b)^2}{2Q} + \pi \frac{b^2}{2Q} \quad (1.5)$$

elde edilir. Yukarıdaki fonksiyonun Q ve b 'ye göre türevi alınır ve sıfıra eşitlenirse bu fonksiyonu minimum yapan Q ve b değerleri bulunur.

$$\frac{\partial TM(Q, b)}{\partial Q} = 0 \Rightarrow -\frac{C_0 D}{Q^2} + \frac{C_h}{2} - \frac{b^2(C_h + \pi)}{4Q^2} = 0 \quad (1.6)$$

$$\frac{\partial TM(Q, b)}{\partial b} = 0 \Rightarrow b = \frac{C_h Q}{C_h + \pi} \quad (1.7)$$

olarak bulunur. Eşitlik (1.6), (1.7)'de yerine konulursa eşitlik (1.8) elde edilir. Bulunan Q^* değeri, eşitlik (1.7)'de Q yerine yazılıp sadeleştirme yapıldığında eşitlik (1.9) bulunur.

$$Q^* = \sqrt{\frac{2C_0 D}{C_h}} \sqrt{\frac{C_h + \pi}{\pi}} \quad (1.8)$$

$$b^* = \sqrt{\frac{2C_0 D}{\pi}} \sqrt{\frac{C_h}{C_h + \pi}} \quad (1.9)$$

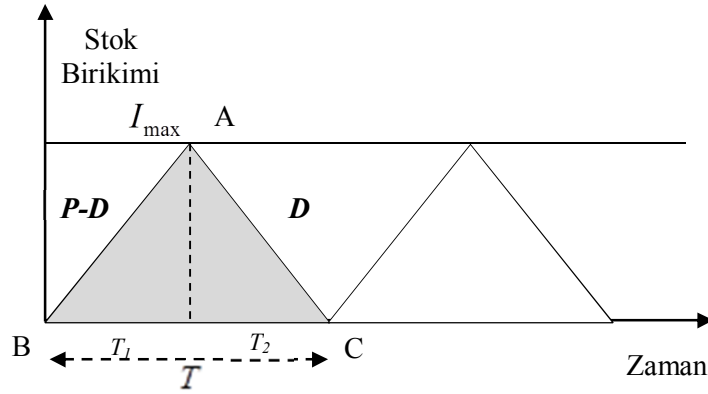
(c) Ekonomik Üretim Miktarı (Yoksatmasız)

Modeli: Ekonomik Üretim Miktarı (EÜM) modeli, ESM modelinin genişletilmiş bir versiyonudur. Bu modelin varsayımları ESM modelinin varsayımlarıyla aynıdır. Sadece dönem başına sipariş edilen ürünlerin ulaşımı bir anda olmamakta ve

siparişler günlük üretim hızı (P) ile ulaşmaktadır. Bu günlük üretim hızını karşılayacak talep hızı (D) ise, yıllık talep miktarının, yıl içerisindeki toplam çalışma günü sayısına bölünmesi ile elde edilir. Burada önemli nokta, stok tükenmesi durumunun olmaması için günlük üretim hızının her zaman için günlük talep hızından büyük olması gerekliliğidir ($P > D$).

Şekil-7 incelendiğinde; siparişlerin tam parti halinde karşılanamadığı, T_1 süresince ihtiyaçların üretilerek temin edildiği, üretim hızının (P), talep hızından (D) büyük olması sebebiyle stoklarda üretim miktarı ile talep arasındaki fark kadar ($P-D$) artış olduğu, T_1 süresinin sonunda sipariş miktarının tamamlanmasıyla üretimin durduğu, stok seviyesinin en yüksek düzeye ulaştığı ve bu zamandan sonra gelen taleplerin T_2 süresince stoklardan karşılandığı görülmektedir. Bu model aşağıdaki varsayımları içerir:

- Talep belirli ve her periyotta sabittir.
- Üretim hızı belirli ve her periyotta sabittir.
- Verilen siparişler periyot içinde alınmaktadır.
- Yok satmaya müsaade edilmemektedir.
- Üretim hızı talepten büyüktür.



Şekil-7: Ekonomik Üretim Miktarı Modeli (Yok Satmasız)

Bu modelde toplam maliyetin üç unsuru bulunmaktadır. Bunlar; hazırlık (sipariş) maliyeti, üretim (satınalma) maliyeti, stokta tutma maliyeti. EÜM (Yoksatmasız) modeline göre toplam maliyet aşağıdaki gibi hesaplanabilir.

Hiç tüketim olmadığında; $Q = T_1 P$, $T_1 = \frac{Q}{P}$ ve $I_{\max} = T_1 (P - D)$ olacaktır.

$$I_{\max} = \frac{Q}{P}(P-D) = Q\left(1 - \frac{D}{P}\right) \quad (1.10)$$

Tek dönem için oluşan toplam maliyet; dönem içinde tezgahların sadece bir defa hazırlanması nedeniyle bir hazırlık maliyeti, üretilen parti için üretim maliyeti ve dönem boyunca elde tutulan stoklar için stokta tutma maliyetinin toplamıdır.

$$TM(Q) = C_0 + CQ + C_h \frac{TI_{\max}}{2} \quad (1.11)$$

$$TM(Q) = C_0 + CQ + C_h \frac{Q}{2D} Q \left(1 - \frac{D}{P}\right) \quad (1.12)$$

Yıllık Toplam Maliyet ise, tek dönemlik toplam maliyetin yıl içindeki dönem sayısı ile çarpılması sonucu bulunur ve aşağıdaki denklem elde edilir:

$$TM(Q) = \frac{C_0 D}{Q} + CD + \frac{C_h}{2} Q \left(1 - \frac{D}{P}\right) \quad (1.13)$$

Yukarıda verilen toplam maliyet fonksiyonunun üretim miktarına göre türevi alınır, sıfıra eşitlenirse toplam maliyeti en az yapan nokta (Ekonomik Üretim Miktarı) bulunmuş olur.

$$\frac{\partial TM(Q)}{\partial Q} = 0 \Rightarrow -\frac{C_0 D}{Q^2} + 0 + \frac{C_h}{2} \left(1 - \frac{D}{P}\right) = 0 \quad (1.14)$$

$$Q^* = \sqrt{\frac{2C_0 D}{C_h \left(1 - \frac{D}{P}\right)}} \quad (1.15)$$

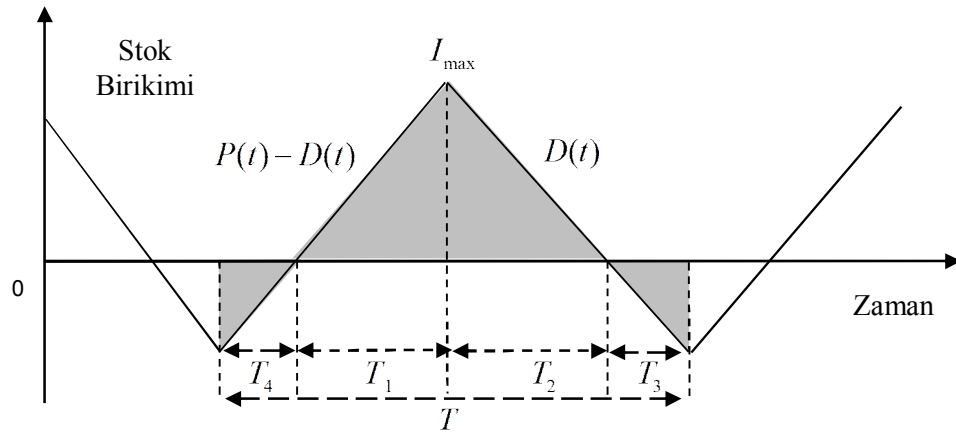
(ç) Ekonomik Üretim Miktarı (Yoksatmalı)

Modeli: Siparişlerin belirli bir üretim kapasitesine bağlı olarak sonradan karşılandığı, aynı zamanda yoksatmanın da bulunduğu bir modeldir. Doğal olarak modelin çalışabilmesi için üretim hızının talep hızından büyük olması gerekmektedir. Yoksatmadan kasıt, stok bulunmaması durumunda müşteri taleplerinin

ertelenmesidir. Stok bulunmaması nedeniyle müşteri taleplerinde oluşacak kayıp söz konusu değildir.

Aşağıdaki şekil incelendiğinde; T_1 süresince stoklarda $P-D$ hızında artışın devam ettiği, T_1 süresinin sonunda verilen sipariş miktarının karşılanmasından dolayı üretimin durduğu, bundan sonra gelen taleplerin mevcut stoklardan karşılandığı, T_2 süresi sonunda stok seviyesinin sıfıra düştüğü, T_3 süresinin sonuna kadar gelen taleplerin karşılanamadığı, gelen müşteri taleplerinin borca alındığı, T_4 süresi başında üretime geçilerek borca alınan ihtiyaçların karşılanmaya çalışıldığı görülür. Modelde kullanılan varsayımlar aşağıdaki gibidir:

- Talep hızı belirli ve her periyotta sabittir.
- Üretim hızı belirli ve her periyotta sabittir.
- Verilen siparişler periyot içinde alınmaktadır.
- Yok satmaya müsaade edilmektedir.
- Üretim hızı talep hızından büyüktür.



Şekil-8: Ekonomik Üretim Miktarı Modeli (Yok Satmalı)

Bu modelde toplam maliyetin dört unsuru bulunmaktadır. Bunlar; hazırlık (sipariş) maliyeti, üretim (satınalma) maliyeti, stokta tutma maliyeti ve yoksatma maliyeti. EÜM modeline göre toplam maliyet aşağıdaki gibi hesaplanabilir.

Şekil-8 incelendiğinde maksimum stok seviyesi $I_{\max} = T_1(P - D)$ ve $I_{\max} = T_2D$ olur. Dolayısıyla aşağıdaki denklem elde edilir.

$$T_1(P - D) = T_2 D \quad (1.16)$$

Yoksatma seviyesi ise, $b = T_4(P - D)$ ve $b = T_3 D$ olur. Dolayısıyla aşağıdaki denklem elde edilir.

$$T_3 D = T_4 (P - D) \quad (1.17)$$

T_4 ve T_1 dönemlerinde üretim ve tüketim, T_2 ve T_3 dönemlerinde sadece tüketim vardır. Tüketilen miktar ancak üretilen miktar kadar olacağından aşağıdaki denklem elde edilir.

$$(T_1 + T_4)(P - D) = (T_2 + T_3) D \quad (1.18)$$

Ayrıca sipariş miktarı, üretimin olduğu dönem ve üretimin olmadığı dönem süreleri şöyledir;

$$Q = (T_1 + T_4) P \Rightarrow (T_1 + T_4) = \frac{Q}{P} \quad (1.19)$$

$$I_{\max} + b = (T_2 + T_3) D \Rightarrow (T_2 + T_3) = \frac{I_{\max} + b}{D} \quad (1.20)$$

Eşitlikler yerine konulduğunda;

$$\frac{Q}{P}(P - D) = \frac{I_{\max} + b}{D} D \quad (1.21)$$

Bir önceki denklemden maksimum stok seviyesi aşağıdaki gibi elde edilir.

$$I_{\max} = Q \left(1 - \frac{D}{P} \right) - b \quad (1.22)$$

T_1 ve T_2 dönemlerinde talep stoklardan karşılanmaktadır. Dolayısıyla;

$$T_1 + T_2 = \frac{I_{\max}}{P-D} + \frac{I_{\max}}{D} = \frac{Q\left(1-\frac{D}{P}\right)-b}{P-D} + \frac{Q\left(1-\frac{D}{P}\right)-b}{D} \quad (1.23)$$

$$T_1 + T_2 = \left[Q\left(1-\frac{D}{P}-b\right) \right] \left(\frac{1}{P-D} + \frac{1}{D} \right) \quad (1.24)$$

T_3 ve T_4 dönemlerinde talep stoklardan karşılanamamaktadır. Dolayısıyla;

$$T_3 + T_4 = \frac{b}{D} + \frac{b}{P-D} \quad (1.25)$$

elde edilir. Bir dönem için toplam maliyet aşağıda ifade edildiği gibi olacaktır.

$$TM(Q, b) = C_0 + CQ + Ch(T_1 + T_2) \frac{I_{\max}}{2} + \pi(T_3 + T_4) \frac{b}{2} \quad (1.26)$$

Bulunan $T_1 + T_2$ ve $T_3 + T_4$ değerleri eşitlikte yerine konur ve eşitlik düzenlenirse bir dönemlik maliyet aşağıdaki gibi olacaktır.

$$TM(Q, b) = C_0 + CQ + \frac{Ch}{2} \left[Q\left(1-\frac{D}{P}\right)-b \right]^2 \left(\frac{1}{P-D} + \frac{1}{D} \right) + \frac{\pi b^2}{2} \left(\frac{1}{P-D} + \frac{1}{D} \right) \quad (1.27)$$

Bir dönemlik maliyet yıl içerisindeki dönem sayısı ile çarpılarak yıllık toplam maliyet aşağıdaki gibi bulunur.

$$TM(Q, b) = CD + C_0 \frac{D}{Q} + \frac{ChD}{2Q} \left[Q\left(1-\frac{D}{P}\right)-b \right]^2 \left(\frac{1}{P-D} + \frac{1}{D} \right) + \frac{\pi b^2}{2} \frac{D}{Q} \left(\frac{1}{P-D} + \frac{1}{D} \right) \quad (1.28)$$

Yukarıdaki düzenlenmiş eşitlikte verilen yıllık toplam maliyet fonksiyonunu en düşük yapan üretim ve yoksatma miktarlarını bulmak amacıyla denklem sırasıyla sipariş ve yok satma miktarına göre türevi alınır ve sıfıra eşitlenir. Buna göre;

$$\frac{\partial TM}{\partial Q} = 0 \Rightarrow Q^* = \sqrt{\frac{2C_0D(Ch + \pi)}{Ch \cdot \pi \left(1 - \frac{D}{P}\right)}} \quad (1.29)$$

$$\frac{\partial TM}{\partial b} = 0 \Rightarrow b^* = \sqrt{\frac{2C_0D \left(1 - \frac{D}{P}\right)}{Ch(Ch + \pi)}} \quad (1.30)$$

Yukarıda tanıtılan deterministik ESM modellerine ilave olarak modelin genişletilmiş versiyonları literatürde yerini almıştır. Bunlardan bazıları aşağıdadır:

- Miktar esaslı indirimleri göz önünde bulunduran modeller,
- Geriye siparişe izin veren modeller,
- Çok ürün için ekonomik sipariş miktarı modelleri,
- Çok aşamalı ekonomik sipariş miktarı modeli.

(2) Olasılıklı Stok Kontrol Modelleri: Bu modeller, deterministik modellerden farklı olarak, gerçek hayatta karşılaşılan stok kontrol problemlerinde mevcut olan belirsizlik ve değişkenlik faktörlerini modelleyebilmek üzere formüle edilmişlerdir. Talep ve temin süresindeki belirsizliklerin model ile ilişkilendirilmesi nedeniyle olasılıklı stok modelleri, deterministik modellerden daha fazla model parametresine ihtiyaç duyarlar ve bazı girdi parametrelerinin olasılıklı ifadeler ile tanımlanmasını gerektirirler. Olasılıklı stok kontrol modelinin optimum sonucu, beklenen toplam maliyet fonksiyonunu en küçükleyen bir dizi değişken kümesini oluşturur. Olasılıklı stok modelleri, talebin, tedarik süresinin veya her iki faktörün de rassal sayılar ile tanımlandığını varsayarlar. Olasılıklı bir stok kontrol modelinin altında yatan matematiksel yapı, deterministik modelinkinden çok daha karmaşıktır ve hizmet düzeyi veya dolum oranı gibi performans ölçümleri için yönetimin bir karar politikası benimsemesini gerektirir.

Olasılıklı stok kontrol modelleri bu tez çalışmasının konusu olmadığı için burada incelenmemiştir. Ayrıntılı bilgi için Behret (2011)'in çalışması incelenebilir.

İKİNCİ BÖLÜM

TEDARİK ZİNCİRİ AĞINDA TESİS YERİ SEÇİMİ PROBLEMİ

1. GENEL

Stratejik planlama bakımından tedarik zinciri yönetimi için tesis yeri seçimi kararları önemlidir. Bu kararlar dağıtım ağının yapılandırılmasında birer araçtır ve tesis yeri kümeleri (Ör.; üretim tesisleri, dağıtım merkezleri, depolar vb.), tesislerin kapasiteleri ve tiplerinin belirlenmesini içerir.

Tesislerin yerleştirilmesi, tedarik zincirinin düzenlenmesi, tedarikçilerin seçilmesi, tedarik zinciri boyunca fiziksel ve bilgi akışının tanımlanması ortak stratejik planlamanın kritik bölümleridir. Bu uzun dönem kararlar şirketin faaliyetlerini etkiler, şirket ve çevresi hakkında derin bilgi sahibi olmayı gerektirir. Şirketler farklı durumlarda tesis yeri seçimi ve tedarik zinciri planlaması problemleri ile karşılaşabilirler. Birkaç örnekten bahsedelim:

- Yeni coğrafi alanlara faaliyetlerini genişleten bir şirket yeni tesisler yerleştirmek zorunda kalabilir ve tedarik zincirinin bir parçası olarak dizayn edebilir.
- Faaliyetler üretim kapasitesinin sınırlarına yaklaştığında yeni tesislerin açılması veya mevcut tesislerin genişletilmesi gerçek bir stratejik problemdir.
- Bir eski bina olması durumunda onu yenileştirmek mi, kapatmak mı yoksa yenisi ile değiştirmenin mi iyi olacağının kararı verilmelidir.
- Talep azalırken tedarik zincirini yeniden yapılandırmak, neredeki faaliyetlerin azaltılması gerektiğinin seçimini içerir.
- Şirketlerin birleştirilmesi tedarik zincirlerinin de birleştirilmesini gerektirir.

Bütün bu örnekler uzun süreli tedarik zinciri tasarımında tesis yerleşimini planlamanın önemini altını çizmektedir.

Tesis Yeri Seçimi Problemi (TYSP), üzerinde çok çalışılan kombinatoryal en iyileme problemlerindendir (Mirchandani ve Francis, 1990). TYSP, tesislerin kurulacağı alternatif yerlerin bir kümesi ve kurulacak tesislerden ihtiyaçları karşılanacak müşterilerin bir kümesi verildiğinde, en az maliyet ile müşteri ihtiyaçlarını karşılayacak şekilde tesislerin kurulacağı yerlerin seçimi problemidir.

TYSP ile ilgili literatürdeki ilk çalışma 1909 yılında Alfred Weber (1909) tarafından yapılmıştır. Weber, birçok müşteriye hizmet verecek bir tesis yerinin talep noktaları ile arasındaki toplam mesafeyi en azlayacak şekilde belirlenmesi için bir model önermiştir. Tesis yeri seçimi teorisi, Hakimi'nin 1964'teki yayınıyla büyük ivme kazanmıştır. Hakimi bu çalışmasını haberleşme ağının ve otoyol sistemindeki polis noktalarının yerlerinin değiştirilmesi üzerine yapmış, olay yerleri ve en yakın hizmet merkezleri arasındaki toplam mesafeyi en az yapmayı amaçlamıştır. Yer seçimi analizleri Densham ve Rushton (1992) tarafından Coğrafi Bilgi Sistemleri (GIS) için uygulanmıştır. Drezner ve Wesolowsky (1995) kentsel alanda tek yönlü yollardan veya caddelerden oluşan düzgün bir grid ağı üzerinde iki temel yer seçimi problemi olan “Weber” ve “Minimaks” problem tekniklerini kullanarak hizmet merkezlerinin yerleştirilmesi üzerine çalışmalar yapmışlardır. ReVelle ve diğ. (2002), kesikli yer seçimi problemleri için 5 çeşit (0/1) tamsayılı programlama modelini ele almışlar ve çalışma yaptıkları bölgede bu modelleri birbirleri ile karşılaştırmışlardır. Modellerde tam ve sezgisel çözüm teknikleri tartışılmış, çok amaçlılık ve ekonomik faktörler de dikkate alınmıştır.

2. TESİS YERİ SEÇİMİ PROBLEMLERİNİN SINIFLANDIRILMASI

TYSP için amaç fonksiyonları, kısıtlar, müşteri sayıları ve tipleri, tesis sayıları ve tipleri, çözüm metotları ve problemin sahip olduğu diğer niteliklere göre çeşitli sınıflandırmalar yapılabilir. Literatürde, Klose ve Drexel (2005)'in sınıflandırılması dikkate alınarak tesis yeri seçimi modelleri aşağıdaki gibi sınıflandırılabilir:

Problemin yer aldığı uzaya göre yapılan sınıflandırmalarda TYSP, sürekli yer seçimi (continuous location) modelleri, şebeke yer seçimi (network location) modelleri ve kesikli yer seçimi (discrete location) ya da karma tamsayılı

programlama modelleri olmak üzere 3 sınıf altında toplanmaktadır. Sürekli uzayda yer alan problemlerde tesisler, herhangi bir yere yerleştirilebilir. Şebeke uzayında tesisler, sadece düğümler ve bu düğümleri birbirine bağlayan yollardan oluşan şebeke üzerine yerleştirilebilir. Kesikli uzaydaki problemlerde tesisler sadece aday noktalar üzerine yerleştirilebilir.

TYSP amaç fonksiyonuna göre toplamın en azlanması (minsum) ya da en büyüğün en azlanması (minmax) olmak üzere ikiye ayrılmaktadır. Toplamın en azlanmasına çalışan modeller, ortalama uzaklığın ya da maliyetin en azlanmasını sağlamak amacıyla tasarlanmışlardır. En büyüğün en azlanmasına çalışan modeller ise en büyük uzaklığın en azlanması için sunulan modellerdir. Birinci tip modeller genellikle ticari amaçlı işletmelerde karşılaşılan yer seçimi problemini çözmek için kullanılırken, ikinci tip modeller kar amacı gütmeyen işletmelerdeki (hastane, okul, ambulans vb.) yer seçimi problemini çözmek amacıyla kullanılmaktadır.

Tesislerde kapasite kısıtının olup olmadığına göre modeller kapasitesiz tesis yer seçimi modelleri ve kapasite kısıtlı tesis yer seçimi modelleri olarak ikiye ayrılmaktadır. Kapasiteli tesis yer seçimi modelleri ise kendi arasında müşteri talebinin bir tesisten ya da birden fazla tesisten karşılanmasına dayalı olarak tek-kaynaklı (single-source) ve çok-kaynaklı (multi-source) modeller olmak üzere iki sınıfa ayrılmaktadır.

TYSP’ni, tedarik zinciri yapısına göre tek aşamalı ve çok aşamalı olmak üzere iki sınıfta incelemek mümkündür. Tek aşamalı tedarik zincirine sahip TYSP hizmet dağıtımını için sadece müşteriler ve talep edilen ürünü sağlayan tedarikçiler arası ürün dağıtımını incelemektedir. Çok aşamalı tedarik zincirine sahip TYSP ise ürün akışında fabrikalar, satıcılar, depolar, müşteriler gibi değişik hiyerarşik kademeler arası ürün akışını ele almaktadır.

Talep türüne göre yapılan TYSP’nin sınıflandırması, taleplerin tek ürün veya çok ürünlü olması şeklinde yapılmaktadır. Yine talep türüne göre yapılan diğer bir sınıflandırma da ise taleplerin bölünebilir olup olmaması problemin sınıfını belirlemektedir. Bölünebilir talepli problemlerde müşterinin ürün talebi farklı

tesislerden karşılanabilirken, bölünemez talepli problemlerde müşteri talebinin tamamı sadece tek bir tesisten karşılanmaktadır.

Talep esnekliğine göre: Literatürdeki çoğu tesis yer seçimi modelinde talep esnekliği olmadığı, yani talebin uzaklığa bağlı olarak değişmediği kabul edilir. Talep uzaklıktan etkileniyorsa, model içerisinde bu durum dikkate alınarak amaç fonksiyonu maliyetin en azlanması yerine karın en çoklanması olarak değiştirilmektedir.

Dönem sayısına göre modeller statik ve dinamik modeller olarak ikiye ayrılmaktadır. Statik modeller sistemin performansını en iyilemeye çalışırken tüm değişkenleri aynı anda ele almaktadır. Dinamik modeller ise bunun tersine aralarında veri değişiklikleri olan değişik dönemleri aralıklarını ele alıp, her dönem için değişen koşullara uygun çözümler önerilmektedir (Erlenkotter, 1981).

TYSP için diğer bir sınıflandırma da problemin girdi parametrelerinin özellikleri açısından yapılmaktadır. Deterministik modellerde parametre değerleri kesin olup, bu problemlerin çözüm yaklaşımları ve çözüm zamanları nispeten daha kolay ve hızlıdır. Bunun yanında çoğu gerçek yaşam problemlerinde olduğu gibi, stokastik tesis yeri seçimi modellerinde girdi parametreleri kesin olmayan, olasılıklı değerlerdir. Bu tür modeller gerçek-yaşam problemlerinin değişkenliğini, rastgele sayıların olasılık dağılımları ya da kesin olmayan parametreler ile olası gelecek senaryoları kullanarak gidermeye çalışmaktadır (Owen ve Daskin, 1998).

Müşterilere hizmet şekline göre sınıflandırmada; klasik tesis yer seçimi modellerinde her bir müşterinin talebinin bir ya da birden fazla tesisten doğrudan karşılandığı kabul edilmektedir. Ancak, müşteri talebinin aracın kapasitesinden küçük olduğu ve bir araçla birden fazla hizmet vermenin söz konusu olduğu durumlarda müşteri talepleri bir rota ile karşılanmaktadır. Bu durumda model, yerleşim-rotalama modeli olarak adlandırılır.

TYSP’nde ilk çalışmalar genellikle tek tesisi sisteme yerleştirme konusu ile ilgilenmiştir. Sonraki çalışmalar p adet ($p > 1$) tesisin sisteme yerleştirilmesi veya

kısıtlar ve amaç fonksiyonu sayesinde modelin tesis sayısını kendi belirlemesini sağlayan çalışmalar şeklinde yapılmıştır.

Sektör tipini esas alarak TYSP iki sınıfa ayrılmaktadır. Özel sektörde yapılan çalışmalarda amaç fonksiyonu genellikle karın en çoklanması ya da maliyetin en azlanması olurken, kamu sektörü modellerinin amacı tüm vatandaşların idare etkinliğinden en iyi şekilde faydalanabilmesini sağlamaktır.

3. TESİS YERİ SEÇİMİ PROBLEM TİPLERİ

Literatürde geçen TYSP'nin çoğunluğu şebeke uzayında yer almaktadır. Tesislerin aday noktalar içerisinde hangilerine yerleştirileceği ve müşterilerin bu tesislerin hangisinden hizmet alacağı problemin temelini oluşturmaktadır. Eğer bir TYSP noktalar ve bu noktaları birleştiren bağlantılar şeklinde yapılandırılmışsa, müşteri talepleri noktalarda olabileceği gibi bu noktalar arasındaki bağlantılarda da oluşabilir. Şebeke tesis yeri problemleri beş şekilde sınıflandırılabilir. Bu sınıflandırma aynı zamanda TYSP'nin amaç fonksiyonlarına göre yapılan sınıflandırmasına da benzemektedir. Bunlar p -medyan problemleri, kaplama problemleri, merkez problemleri, ana dağıtım üssü tesis yeri seçimi problemleri ve hiyerarşik tesis yeri seçimi problemleridir (Arabani ve Farahani, 2012).

4. ÇÖZÜM TEKNİKLERİ

TYSP tipleri NP-Zor (karar problemleri için NP-Tam) problemler olduğundan problemin boyutu büyüdükçe optimal çözümü bulmakta zorluk çekilebilir. NP-Zor, optimizasyon problemlerini sınıflandırmak için kullanılan bir terimdir. Bugüne kadar bu tip problemler için polinom zamanlı bir algoritma geliştirilememiştir. NP-Zor problemlerin optimal çözümünü bulmak, üssel zamanlı (dolayısı ile çok uzun sürebilecek) bir algoritma yoluyla mümkündür (Garey ve Johnson, 1979).

TYSP'ni çözmek için tam (optimal) çözüm teknikleri veya sezgisel teknikler kullanılabilir. Tam çözüm teknikleri, optimizasyon tekniklerinden en önemlisi olan karışık tamsayılı programlamayı içerir. Burada amaç, hizmet kısıtlarına

bağlı olarak doğrusal maliyet fonksiyonunu optimum yapmaktır. Bugün tamsayı programlama çözümü için kullanılan temel algoritma, gevşetilmiş tamsayı programlamaya uygulanan Dal-Sınır tekniği ile Simpleks algoritmasıdır. Bu tür tamsayı problemleri çözebilen çok sayıda ticari yazılım mevcuttur. Diğer taraftan, optimal çözüme yakın olur çözümleri daha çabuk elde etmek için sezgisel teknikler geliştirilmiştir. Sezgisel teknikler, optimal çözümü bulmak için izlenecek metotların çok zaman alacağı durumlarda kullanılmaktadır. Sezgisel tekniklerden en önemlisi ise Lagrange optimizasyonudur. Bu metodun sonuçları tamsayı programlamadan daha az matematik formülasyon gerektirmesine rağmen çözüm çok daha zor olabilmektedir. Yer seçimi problemlerini çözmek için kullanılabilecek sezgisel tekniklerden bazıları ise, değişim sezgiselleri, düşme sezgiselleri, sıralı yer seçim problemleri, tavlama benzetimi, tabu arama ve genetik algoritmalarıdır.

5. TEDARİK ZİNCİRİ AĞINDA TESİS YERİ SEÇİMİ PROBLEMLERİ İLE İLGİLİ LİTERATÜR

Tesis yeri seçimi ve stratejik planlama, yönetim bilimi ve yöneylem araştırması alanında çok sayıda yayının konusu olmaktadır. İlk araştırmalar tekli tesislere ve sonra çoklu tesislere odaklanmıştır. Daha gerçekçi problemler için mevcut modellere yeni kısıtlar eklenmiştir.

Geçmiş yıllarda, birçok çalışma dinamik TYSP'ne yönelmiştir. Bu problem sınıfında, kararlar uzun dönemli planlama ufkuna yayılmış ve karar değişkenleri zamana bağlı olmuştur. Amaç, bu planlama ufku boyunca amaç fonksiyonunun en azlanması ve en çoklanması olmuştur. Örneğin, talep ve lojistik kısıtların en az maliyetle doyurulmasını sağlamak için büyüyen bir şirket önümüzdeki beş yıl içinde lojistik sistemini nasıl geliştirebilir.

O kadar çok sayıda şirketin lojistik ağları var ki, herhangi bir organizasyona uyarlanabilecek tamamen genel bir model kurmak gerçekçi görünmemektedir. En son modeller, bazı gerçek durumları yansıtmaya veya TYSP'nin bazı belirgin yönlerine odaklanma düşüncesi ile birçok ilave özelliği içinde barındırmaktadır. Son modellerin temel karakteristikleri arasında aşağıdakiler gösterilebilir:

- Çok aşamalı tedarik zinciri: Tedarikçiler, üretim tesisleri, depolar, müşteriler vb.
- Çoklu ürünler ve ürün aileleri,
- Karmaşık ürün akışı: Üretim tesisleri ve depoların birkaç katmanı, üretim tesisi ve depolar arasında ürün değişimi, bazı müşterilere doğrudan dağıtım, tersine lojistik, yeniden üretim vb.
- Birkaç ulaştırma modu, birkaç ürün teknolojisi arasında seçim,
- Çeşitli kısıtlar: Kapasite kısıtları, bütçe kısıtları vb.
- Karmaşık maliyetlerin yapısı: Sabit veya değişken maliyetler, doğrusal veya doğrusal olmayan maliyetler,
- Hibrit stratejik/taktik stok modelleri: Ortalama stok, güvenlik stoku, çevrim stoku.

Literatürde tedarik zinciri ağı tasarım problemleri ile ilgili değişik çalışmalar vardır. Bunlardan çok aşamalı, çok dönemli, stok kontrollü, lojistik/tedarik zinciri ağı ve tesis yeri seçimi problemleri ile ilgili olanları aşağıdadır. Ballou (1968) tarafından yapılan öncü çalışmadan beri, dinamik tesis yeri seçimi problemlerine değinen birçok çalışma yapılmıştır. Problemin kapasiteli versiyonu ile ilgili olarak, işletim kapasitesinin artırılması/azaltılması olasılığı sıkça ele alınmıştır. Kapasite kısıtlarının içerildiği en erken çalışmalardan birisi Erlenkotter (1981) tarafından yapılan çalışmadır. Yazar, artan talepleri karşılamak için hangi dönemlerde tesis kümelerinin kapasitelerinin artırılması gerektiği kararının verilmesini analiz etmişlerdir. Melachrinoudis ve diğ. (1995) kapasiteli çok dönemli TYSP'nin çok amaçlı versiyonunu çalışmışlardır. Ara tesis kümelerinin (dağıtım merkezleri vb.) yerleşim kararı problemi yakın zamanda Hormozi ve Khumawala (1996) tarafından çalışılmıştır. Bu modeller daha sonra çok kademeli tesislerin yerleştirilmesi problemine genişletilmişlerdir. Pirkul ve Jayaraman (1998) problemin statik versiyonu ile ilgilenmişlerdir. Saldanha da Gama (2002) modelinde bütçe kısıtını ele alırken, uluslararası tesislerin dinamik yerleştirilmesini incelemiştir. Wang ve diğ. (2003) statik tesis yerleşim problemlerinin içine bütçe kısıtını eklemiştir.

Shulman (1991) sonlu kapasiteli çoklu tesislerin ele alındığı karışık tamsayılı programlama modelini önermiştir. Kapasiteli ara tesisler kümesinin en iyi

yerleşim kararı probleminde dinamik ve çok ürünlü model Canel ve Khumawala (2001) tarafından ele alınmıştır. Melachrinoudis ve Min (2000) çok amaçlı ortamda tek bir ara tesisin yeniden yerleştirilmesine odaklanmışlardır. Özdamar ve Yazgaç (1997) üretim merkezi ve depolarının yer aldığı üretim/dağıtım sistemini geliştirmişler, üretim kısıtı ve stok denge kısıtları altında stok ve ulaştırma maliyetlerini en az yapmayı amaçlamışlardır. Hinojosa ve diğ. (2000) çok ürünlü dağıtım ağında çok aşamalı tesis yeri seçimi ile dinamikliği birleştiren bir çalışma yapmışlardır. Bu problem daha sonra Velten (2002) tarafından stok konularına genişletilmiştir. Stoklar aynı zamanda Gue (2003) tarafından muharebe lojistiği için önerilen dinamik modelde ele alınmıştır.

Petrovic ve diğ. (1999) bulanık kısıtlar altında tedarik zinciri davranışlarını modellemişlerdir. Geliştirdikleri modelde, belirsiz talep ve belirsiz dağıtımın tedarik zinciri davranışları üzerinde büyük rol oynadığını göstermişlerdir.

Melkote ve Daskin (2001) tesis yeri seçimi ve ulaştırma ağı planını birlikte optimize eden model geliştirmişlerdir. Bu model klasik basit sınırsız kapasiteli sabit maliyetli TYSP'ni genelleştirmektedir. Syarif ve diğ. (2002) belirli bir kapasiteye sahip potansiyel tesislerin (fabrikalar, dağıtım merkezleri vb.) kurulmasının sabit maliyeti ile müşteri talebinin tesislerden taşınması maliyeti toplamını en azlayacak tesislerin ve dağıtım ağının en uygun tasarımını elde etmeye çalışmışlardır. NP-Zor karışık tamsayılı doğrusal programlama yapısındaki bu problemi çözmek için genetik algoritma tabanlı bir yöntem geliştirmişlerdir.

Chen ve Lee (2004) belirsiz piyasa talepleri ve ürün fiyatları ile tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için, çok ürünlü, çok kademeli ve çok dönemli planlama modeli önermişlerdir. Talepteki belirsizlik kesikli olasılık senaryoları ile modellenmiş ve satıcı ve alıcıların ürün fiyatlarında birbiriyle uyuşmayan tercihlerini tanımlamak için bulanık kümeler kullanılmıştır.

Cordeau ve diğ. (2006) çok ürünlü, çok dönemli, çok tesisli tek ülke ağında statik bir model sunmuştur. Karar değişkenleri, üretim tesisleri ve depoların sayısı, kapasiteleri ve teknolojileri, tedarikçi seçimi, dağıtım kanallarının seçimi, ulaştırma modları ve malzeme akışıdır. Malzeme cetveli de ayrıca dikkate alınmıştır. Problemi

çözmek için iki metod sunmuşlardır: Simpleks tabanlı dal sınır yaklaşımı ve Benders ayrıştırması yaklaşımı. Hamer-Lavoie ve Cordeau (2006) çalışmalarında tedarikçiler ve malzeme cetvelini modelden çıkararak modeli basitleştirmişlerdir. Stokastik talepli dinamik olan model, güvenlik stoklarını da dikkate almaktadır. Yazarlar güvenlik stoku ile ilgili kısıtlar için doğrusal yaklaştırma (linear approximation) ve geçerli eşitsizliklerle güçlendirilmiş dal sınır metodunu önermişlerdir.

Dias ve diğ. (2006) iki kademeli ağı (tesis yerleri ve müşteriler) yeniden yapılandırılması üzerinde çalışmışlardır. Yazarlar, planlama ufku boyunca tesislerin birden fazla açılabilceğini, kapanabilceğini ve tekrar açılabilceğini varsaymışlar ve bu durumları üç senaryo ile çalışmışlardır: Problemlerin hepsi primal-dual sezgisellerle çözülmüştür. Geliştirdikleri modelde planlama ufku boyunca tesislerin açılma/kapanması olasılıklıdır ve kapanan bir tesisin tekrar açılması yasaklanmıştır.

Melo ve diğ. (2006) kapasite artırılması/azaltılması senaryoları ile ağların yeniden yerleştirilmesini amaçlamışlardır. Kapasite mevcut tesis ile yenisi arasında veya iki tesis arasında belli koşullar altında değişkenlik gösterebilir. Kapasitedeki her değişiklik maliyetle cezalandırılmaktadır. Bu modelde planlama ufkunun sonuna kadar kapanan tesisler yeniden açılmaz ve yeni kurulan tesisler daima açık kalır. Literatürde nadir olan bazı bütçe kısıtları da modele eklenmiştir. Bu her bir tesis için modüler bir kapasite girişini önermektedir.

Vila ve diğ. (2006) dinamik bir model önermişler, kereste endüstrisinde bir uygulamayı ele almışlar, deterministik talepli uluslararası bir ağı düşünmüşlerdir. Çalışmalarında, dış tedarikçiler, kapasiteli üretim tesisleri ve depolar, mevcut teknoloji kümeleri arasından seçim, tesislere kapasite opsiyonu ekleme olasılığı ve standart olanların yerine geçmek üzere değiştirilmiş ürünleri de ele almışlar ve üç seviye depolamayı tanımlamışlardır. Amaç fonksiyonları, maksimize edilmesi gereken vergi sonrası kârdır.

Altıparmak ve diğ., (2006) plastik ürünler üreten bir firmanın tedarik zinciri ağını tasarlamak için çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeli önermişler ve bu modeli genetik algoritmaya dayanan bir yaklaşım ile çözmüştür. Modelin amaçları şunlardır: (1) tedarik zincirinin toplam maliyetini en

küçükmek, (2) hizmet düzeyini en büyükmek (hizmet düzeyi kabul edilebilir teslim süresi ile ölçülmüştür), (3) dağıtım merkezlerinin dengeli kapasite kullanımını en büyükmek. Yöntem tüm tedarik zincirindeki tedarikçileri, üretim tesislerini, dağıtım merkezlerini ve müşterileri dikkate alan bütünsel bir yaklaşım sunmaktadır.

Lu ve Bostel (2007), yeniden üretim opsiyonu kullanan tersine lojistik sistemler için tesis yeri seçimi modeli kurmuşlar, ileri ve tersine akışı birlikte optimize etmeye çalışmışlardır. İki aşama ve üç tip tesis içeren problem için karma tamsayılı programlama modeli geliştirilmiş, çözümünde Lagrange sezgisellerine dayalı bir algoritma kullanılmıştır.

Ko ve Evans (2007) problemi üçüncü parti lojistik (3PL) olarak formüle etmek ve çözebilmek için, aynı anda ileri ve tersine akışı dikkate alan, çok dönemli, iki kademeli, çok ürünlü, kapasiteli ağ tasarımı problemi ve karışık tamsayılı doğrusal olmayan programlama modeli sunmuşlardır. Bu tür ağ tasarım problemleri NP-Zor problem sınıfına girdiğinden, gerçek boyutlu problemi çözmek için bu çalışmada genetik operatörler ve simpleks aktarma algoritmalarından oluşan bir genetik algoritma tabanlı sezgisel önerilmiştir.

Thanh ve diğ. (2008) deterministik talepli çok aşamalı, çok ürünlü tedarik zinciri ağında üretim/dağıtım sisteminin tasarlanması ve planlanması için karışık tamsayılı programlamayı önermişlerdir. Literatürde var olan modelleri gerçek hayat problemlerine genişletmişlerdir.

Xu ve diğ. (2008) tedarik zinciri ağı problemi için rassal bulanık talepli, çok amaçlı programlama modeli önermişlerdir. Beklenen değer ve şans kısıtı operatörü kullanılarak model deterministik çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeline dönüştürülmüştür. Çözüm için Prüfer sayıları gösterimiyle yayılan ağaç tabanlı genetik algoritma kullanılmıştır.

Gebennini ve diğ. (2009) çok kademeli üretim/dağıtım sisteminde tesislerin sayısı (depolama sistemleri, dağıtım merkezleri vb.), yerlerinin seçimi, müşteri taleplerinin o yerlere atanmasını ve aynı zamanda stokastik ortamda stok kontrolü, üretim hızı ve hizmet seviyesinin belirlenmesi ile üretim ve/veya depolama tesisleri

arasındaki ürün akışı ve tersine lojistik faaliyetlerin optimize edildiği bir model ortaya koymuşlardır. Model tedarik zinciri ağında stratejik, taktik ve operasyonel kararları bütünleştirmesi açısından literatürde bir ilktir.

Paksoy ve diğ. (2009) deterministik, çok aşamalı tedarik zinciri ağının tasarlanmasında çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeli geliştirmişlerdir. Çalışmada tedarik zinciri ağı tasarımı için; aşamalar arasındaki toplam taşıma maliyetlerinin en azlanması, ESM modeline göre stok bulundurma ve sipariş verme maliyetinin en azlanması, aşamalar arasındaki taşınan malzemelerin azalan varyansı yoluyla gereksiz ve kullanılmayan tesis ve dağıtım merkezi kapasitelerinin en azlanmasından oluşan üç farklı amaç belirlenmiştir.

Qin ve diğ. (2009) satıcıların stokastik taleplerinin normal dağıldığı varsayımı altında belirli hizmet seviyesinde tesis yeri, stok ve taşıma maliyetlerinin toplamını en azlamak için karışık tamsayılı doğrusal olmayan model önermişler, problemi çözmek için Tavlama Benzetimi Algoritması geliştirmişlerdir.

Zheng ve Liang (2009) tedarik zinciri ağındaki dinamik tesis yeri seçimi problemlerini iki safhada incelemişlerdir. Birinci safhada paydaşların seçimi, ikinci safhada mevcut tesislerin kapasite transferi yoluyla yeniden yerleştirilmesi, envanterin bütünleştirilmesi, ulaştırma ve tedarik kararları, yatırım için verilen bütçenin yeterliliği ve tedarik zinciri ağının genel yapısı gibi mevcut literatürde şu ana kadar çok az önem verilen konulara odaklanmıştır.

Sourirajan ve diğ. (2009), perakendecilerdeki tek bir ürünü takviye eden üretim tesisi olan iki kademeli bir tedarik zincirini ele almışlardır. Amaç, tesis yeri, boru hattı stok ve güvenlik stoku maliyetleri toplamını en aza indirecek ağ dağıtım merkezlerini tespit etmektir. Modeli çözmek ve daha önceki geliştirilen bir Lagrange sezgiselinin performansını karşılaştırmak için genetik algoritma kullanmışlardır.

Bashiri ve Badri (2010), çok aşamalı çok ürünli tedarik zinciri ağı tasarımı için dinamik karışık tamsayılı doğrusal programlama modeli önermişlerdir. Model, tedarikçilerin seçimini, tedarikçiden tedarikçiye gönderilecek hammadde miktarını, üretim tesislerinde üretilerek depo ve pazarlara gönderilecek ürün miktarını ve

tedarik zincirinde net karı ençoklamak için bazı diğer stratejik ve taktik kararları belirlemeyi amaç edinmiştir.

Benyoucef ve diğ. (2013), çalışmalarında tedarik zinciri ağı tasarımı için entegre tesis yeri seçimi ve tedarikçi seçimi kararlarının birlikte verildiği karışık tamsayılı doğrusal olmayan stokastik programlama modelini ele almışlardır. Amaç, tedarik zinciri ağı üzerindeki taşıma maliyetleri ile dağıtım merkezlerinin sabit açma ve işletme maliyetleri, stok, güvenlik stocku, sipariş verme maliyetlerini en azlamaktır. Problemi çözmek için Lagrange Gevşetimi tabanlı bir yaklaşım önerilmiştir.

Nezhad ve diğ. (2013), çalışmalarında, stratejik tedarik zinciri planlamasında tesis yeri seçimi problemi için kapasitesiz, tek kaynaklı, çok ürünlü üretim/dağıtım planlama modelini ele almışlardır. NP-Zor olan optimizasyon problemini çözmek için, yerel arama operatörlerinin de kullanıldığı iki adet Lagrange Gevşetimi tabanlı sezgisel yaklaşım önermişlerdir.

Shahabi ve diğ. (2013), çoklu tedarikçiler, üretim tesisleri, aktarma noktaları ve perakendecilerden oluşan dört aşamalı tedarik zinciri ağı için tesis yeri seçimi ve stok kontrolünün koordine edildiği matematiksel bir model geliştirmişlerdir.

Sabzevari ve diğ. (2014), bir çelik endüstrisinde tedarik zinciri ağının stratejik ve taktik planlamasına odaklanmışlardır. Her geçen gün artan çelik ürünleri talebinin üretim ve depolama kapasitelerini zorlaması nedeniyle, ülke çapındaki tedarik zinciri ağındaki tesislerin üretim, dağıtım ve stok kapasitelerinin planlamasını ele almışlardır. Müşterilerin stokastik taleplerinin normal dağıldığı varsayımı altında hizmet seviyesini gerçekleştirmek için güvenlik stokları ile birlikte üretim kapasitelerini belirlemeyi amaçlamışlardır. Dinamik, çok ürünlü tedarik zinciri ağı tasarımı oluşturdukları değişik boyutlardaki problemleri çözmek için karışık tamsayılı doğrusal ve doğrusal olmayan programlama modelleri geliştirmişlerdir.

Literatürde konu ile ilgili günümüze kadar yapılan çalışmalarda geliştirilen modellerden dikkat çeken bazılarının özellikleri Tablo-1’de özetlenmiştir. Modellerin çözüm metotları ise Tablo-2’dedir.

Tablo-1: Modellerin Karakteristikleri

	Planlama Ufku (S/D) ¹	Amaç Fonk. (C/P/M) ²	Belirlilik- Belirsizlik (D/S/B) ³	Çok ürünlü	Tesis Kad.	Tesis Tipi	Açma/ Kapama	Dış Tedarik	Malz. Cetveli	Stok	Kapasite kısıtları (C/E/R/M) ⁴	Bütçe kısıtları
Ballou (1968)	D	C	D	—	İki	Tek	—	—	—	—	—	—
Erlenkotter (1981)	D	C	D	—	İki	Tek	—	—	—	—	—	—
Shulman (1991)	D	C	D	—	İki	Tek	—	—	—	—	E,M	—
Hormozi, Khumawala (1996)	D	C	D	—	İki	Tek	—	—	—	—	C	—
Pirkul, Jayaraman (1998)	S	C	D	√	İki	Tek	—	—	—	—	C	—
Melachrinoudis ve Min (2000)	D	M	D	—	İki	Tek	√	—	—	—	E,R	√
Hinojosa ve diğ. (2000)	D	C	D	√	İki	Tek	√	—	—	—	C	√
Canel ve Khumawala (2001)	D	C	D	√	Çok	Tek	√	—	—	—	C	—
Velten (2002)	D	C	D	√	Çok	Tek	—	—	—	√	√	—
Syarif ve diğ. (2002)	D	C	D	√	Çok	Tek	—	—	—	—	√	—
Saldanha da Gama (2002)	D	C	D	√	İki	Tek	—	—	—	—	C	√
Gue (2003)	D	C	D	√	İki	Tek	—	—	—	√	√	—
Wang ve diğ. (2003)	S	C	D	—	İki	Tek	—	—	—	—	—	√
Chen ve Lee (2004)	D	M	B	√	Çok	Çok	—	—	—	√	—	—
Melo ve diğ. (2005)	D	C	D	√	Çok	Çok	—	—	—	√	C,E,R,M	√
Hamer-Lavoie ve Cordeau (2006)	D	C	S	—	Çok	Tek	—	—	—	√	C	—
Dias ve diğ. (2006)	D	M	D	—	İki	Tek	√	—	—	—	C	—
Vila ve diğ. (2006)	D	P	D	√	Çok	Tek	—	√	—	√	C,E,M	—
Altıparmak ve diğ. (2006)	S	M	D	√	Çok	Üç	√	—	—	√	C,E,M	√
Lu ve Bostel (2007)	S	C	D	—	Çok	Üç	—	—	—	—	C	—
Ko ve Evans (2007)	D	C	D	√	Çok	Tek	—	3PL	—	—	C	—
Thanh ve diğ. (2008)	D	C	D	√	Çok	İki	√	√	—	—	C,E,R,M	—
Xu ve diğ. (2008)	D	M	B	√	Çok	Tek	—	—	—	—	C	—
Qin ve diğ. (2009)	D	C	S	√	Çok	Tek	—	—	—	√	—	—
Gebennini ve diğ. (2009)	D	C	S	√	Çok	Tek	—	—	√	√	C	—
Paksoy ve diğ. (2009)	D	M	D	—	Çok	Tek	—	—	—	√	C	—
Zheng ve Liang (2009)	D	C	D	—	İki	Çok	√	√	—	√	C,E,R,M	√
Sourirajan ve diğ. (2009)	S	C	D	—	İki	Tek	—	—	—	√	C	—
Bashiri ve Badri (2010)	D	P	D	√	Çok	İki	—	—	—	—	C	—
Nezhad ve diğ. (2013)	S	C	D	√	Çok	Tek	√	—	—	—	—	—
Shahabi ve diğ. (2013)	S	C	D	—	Çok	Çok	√	—	—	√	C	—
Benyoucef ve diğ. (2013)	D	M	S	—	Çok	İki	√	—	—	√	C	√
Sabzevari ve diğ. (2014)	D	C	S	√	Çok	Çok	√	—	—	√	C,E,R,M	—

¹ S: Statik, D: Dinamik² C: Maliyet, P: Kar, M: Çok amaçlı prog.³ D: Deterministik, S: Stokastik, B: Bulanık⁴ C: Kapasiteli, E: Kapasite artışı, R: Kapasite azaltımı, M: Modüler kapasite

Tablo-2: Modellerin Çözüm Teknikleri

	Tam Çözüm Metotları	Ticari Yazılımlar	Sezgisel Metotlar	Meta-Sezgisel Metotlar
Ballou (1968)	X			
Erlenkotter (1981)	X		X	
Shulman (1991)		X		
Hormozi ve Khumawala (1996)			X	
Pirkul ve Jayaraman (1998)	X		X	
Melachrinoudis ve Min (2000)		X		
Hinojosa ve diğ. (2000)	X		X	
Canel ve Khumawala (2001)	X			
Velten (2002)			X	
Syarif ve diğ. (2002)				GA
Saldanha da Gama (2002)			X	
Gue (2003)			X	
Wang ve diğ. (2003)				TA
Chen, Lee (2004)				GA
Melo ve diğ. (2005)		X		
Dias ve diğ. (2006)			X	
Hamer-Lavoie ve Cordeau (2006)	X			GA
Dias ve diğ. (2006)	X			
Vila ve diğ. (2006)	X	X		
Altıparmak ve diğ. (2006)				GA
Lu ve Bostel (2007)	X		X	
Ko ve Evans (2007)				GA
Thanh ve diğ. (2008)	X			
Xu ve diğ. (2008)				GA
Qin ve diğ. (2009)				TB
Gebennini ve diğ. (2009)	X			
Paksoy ve diğ. (2009)		X		
Zheng ve Liang (2009)	X			
Sourirajan ve diğ. (2009)	X			GA
Bashiri ve Badri (2010)	X			
Nezhad ve diğ. (2013)			X	
Shahabi ve diğ. (2013)		X		
Benyoucef ve diğ. (2013)			X	
Sabzevari ve diğ. (2014)	X	X		

ÜÇÜNCÜ BÖLÜM

TEDARİK ZİNCİRİ PLANLAMASINDA KARŞILAŞILAN BELİRSİZLİKLER

1. TEDARİK ZİNCİRİNDE BELİRSİZLİKLERİN İNCELENMESİ

Gerçek hayatta karşılaşılan tedarik zincirleri belirsiz bir ortamda faaliyet gösterirler. Tedarik zincirinin bütünlük yapısı, sistemin karmaşıklığını da artırmaktadır. Birçok alt sistemin birleşmesinden oluşan tedarik zinciri, gerek bütünlük yapısı gerekse içinde barındırdığı insan faktörü nedeniyle çok sayıda belirsizlik içermektedir. Belirsizlik, etkili bir tedarik zinciri üretim/dağıtım planlama modeli yaklaşımının ortaya koyulabilmesi için ele alınması gereken en önemli husustur.

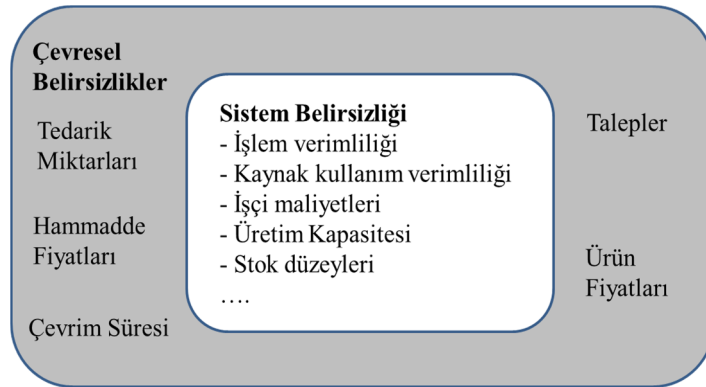
Belirsizlik bir işi yapmak için gerekli olan bilgi ile sahip olunan bilgi arasındaki fark olarak tanımlanabilir (Mula ve diğ., 2006). Zimmermann (2000) ise uygulamaya yönelik olarak belirsizliği şu şekilde tanımlamıştır: “Belirli bir durum için bir kişiyi, bir sistemi, sistemin davranışını veya diğer özelliklerini deterministik ve sayısal olarak tanımlamaya, belirlemeye ve tahmin etmeye nitel ve nicel olarak uygun bilgiyi hazırlayamıyorsa belirsizlik var demektir.”

Belirsizliğin nedenlerini bilmek, belirsizliği anlayabilmek ve modelleyebilmek için oldukça önemlidir. Petrovic (2001), belirsizlik kaynaklarının rassal olaylar, delil (veri) yetersizliği, veri kesinliğinin eksikliği ya da yargılardaki belirsizlikler olabileceğini belirtmiştir. Zimmermann (2000) çalışmasında belirsizliğin nedenlerini altı başlık altında incelemiştir: Bunlar; bilgi eksikliği, bilgi çokluğu (karmaşıklık), çelişen bulgular, birden çok anlama gelme, muğlaklık, ölçüm ve inançlardır. Eğer bir sistemdeki belirsizlik modellenmek isteniyorsa öncelikle belirsizliğin türü burada tanımlanan türlere göre belirlenmelidir. Daha sonra sisteme girdilerin ne olacağı ve sayısal bilgilerin ölçekleri belirlenmeli ve oluşturulacak model ile elde edilmesi beklenen çıktı saptanmalıdır. Tablo-3’te verilen sınıflandırmaya göre belirtilen hususlar göz önünde bulundurularak, kullanılacak modele karar verilebilir.

Tablo-3: Belirsizlik Özelliklerinin Sınıflandırılması (Zimmermann, 2000)

1. Belirsizliğin sebepleri
(a) Bilgi eksikliği
(b) Bilgi bolluğu (karmaşıklık)
(c) Çelişen bulgular
(d) Birden çok anlama gelme
(e) Ölçüm
(f) İnanç
2. Elde edilebilir bilgi (girdi)
(a) Sayısal
(b) Küme veya aralık değerli
(c) Sözel
(d) Sembolik
3. Sayısal bilginin ölçeği
(a) Nominal
(b) Sıralı
(c) Kardinal
4. Elde edilmesi planlanan bilgi (çıktı)
(a) Sayısal
(b) Küme veya aralık değerli
(c) Sözel
(a) Sembolik

Bir tedarik zinciri üretim/dağıtım sistemini etkileyen birçok belirsizlikler genel olarak; çevresel belirsizlik ve sistem belirsizliği başlıkları altında toplanabilir. Çevresel belirsizlik, incelenen sistemin dışından kaynaklanan belirsizlik türüdür. Talepte ve tedarik miktarında meydana gelebilecek belirsizlikler bu gruba girer. Sistem belirsizliği ise incelenen tedarik zincirinin içsel yapısından kaynaklanan belirsizlik türüdür. İşlem verimlilikleri, kaynak kullanım verimliliği, üretim sistemindeki hatalar sistem belirsizliği olarak adlandırılabilir (Mula ve diğ., 2006). Aşağıdaki şekilde, bahsedilen belirsizlik türleri örnekleriyle birlikte verilmiştir.



Şekil-9: Tedarik Zinciri Planlamasında Belirsizlikler

Tedarik zincirini gerçek hayata uygun olarak planlayabilmek için özellikle uzun vadeli stratejik planlamada belirsizliklerin kesinlikle modellenmesi gerekir. Bu noktada hangi yöntem ile belirsizliğin modelleneyeceği cevaplanması gereken en önemli sorudur (Zimmerman, 2006). Literatürde çok sayıda araştırmacı tedarik zinciri üretim/dağıtım sistemlerindeki belirsizliği modelleyebilmek için çeşitli yöntemler sunmuştur. Mula ve diğ. (2006), belirsizlik altında tedarik zinciri üretim/dağıtım planlama ile ilgili modelleri kapsamlı bir şekilde incelemişlerdir. Belirsizlik altında üretim planlama modellerinin, Tablo-4'te verilen, kavramsal modeller, analitik modeller, yapay zeka modelleri ve benzetim modellerinden oluşan dört ana başlıkta sınıflandırmasını yaparak mevcut çalışmaları özetlemiştir. Daha sonra literatürdeki belirsizlik altında üretim planlama modellerini Tablo-5'teki gibi gruplamışlardır.

Tablo-4: Üretim Sistemlerindeki Belirsizlik Modellerinin Sınıflandırılması (Mula ve diğ., 2006)

Kavramsal modeller	Analitik modeller
Kazanç faktörleri	Hiyerarşik süreçler
Emniyet stoku	Matematiksel programlama
Tedarik zamanı	Stokastik programlama
Stoklama	Deterministik yaklaşımlar
İhtiyaç fazlası planlama	Laplace dönüşümleri
Malzeme ihtiyaç planlaması	Markov karar süreçleri
Esneklik	
Yapay zeka modelleri	Benzetim modelleri
Uzman sistemler	Monte Carlo teknikleri
Öğrenme eğrileri	Olasılık dağılımları
Bulanık küme teorisi	Sezgisel yöntemler
Bulanık mantık	Ağ modelleme
Sinir ağları	Kuyruk teorisi
Genetik algoritmalar	Dinamik sistemler
Çoklu ajan sistemleri	

Tablo-5: Belirsizlik Altında Üretim Planlama Modelleri (Mula ve diğ., 2006)

Araştırma konuları	Modelleme türü	Çalışma sayısı
1. Toplu Planlama	Yapay zeka modelleri	8
	Benzetim modelleri	2
2. Hiyerarşik Üretim Planlaması	Analitik modeller	3
3. Malzeme İhtiyaç Planlaması	Kavramsal modeller	9
	Analitik modeller	6
	Yapay zeka modelleri	4
	Benzetim modelleri	10
4. Kapasite Planlaması	Analitik modeller	4
	Benzetim modelleri	1
5. Üretim Kaynakları Planlaması	Analitik modeller	7
	Yapay zeka modelleri	5
	Benzetim modelleri	2
6. Envanter Yönetimi	Analitik modeller	10
	Yapay zeka modelleri	5
7. Tedarik Zinciri Planlaması	Kavramsal modeller	1
	Analitik modeller	5
	Yapay zeka modelleri	5

Kavramsal modeller, klasik modellere belirsizliği içeren bazı kavramların eklenmesi ile oluşturulan yöntemlerdir. Örneğin klasik malzeme ihtiyaçları planlamasında “fire oranları”nın ilave edilmesiyle kavramsal bir model oluşturulabilir. Analitik modeller içerisine matematik programlama ve stokastik modelleme girmektedir. Yapay zeka modelleri ile bulanık modelleme, yapay sinir ağları ve genetik algoritma gibi teknikler kastedilmektedir. Yazarların yaptıkları çalışmaya göre; analitik modelleme yaklaşımı, özelde de stokastik programlama en çok karşılaşılan yaklaşımdır. Birçok analitik model sadece tek yönlü bir belirsizliği dikkate almış ve basit yapılı üretim süreçlerinde denenmiştir. Birden çok belirsizliği dikkate alan ve çok sayıda ürünü içeren daha karmaşık süreçlerin modellenmesinde ise yapay zeka ve benzetim daha çok kullanılmıştır. Yapay zeka modellerinden bulanık kümeler teorisi üretim yönetimi için oldukça çekici bir araç olarak tespit edilmiştir. Ayrıca çalışmada kavramsal yaklaşımlara da değinilmektedir.

Literatür incelendiğinde, tedarik zinciri üretim/dağıtım planlama problemlerinde dört çeşitli belirsizlik kaynağının olduğu görülmektedir. Buna göre; talepte, ürün fiyatlarında ve ürünleri üretmek için gerekli kaynak miktarlarında (ara ürün ve/veya hammaddeler) ve stok düzeyinde belirsizlikler söz konusudur. En çok

üzerinde durulan ise talep belirsizliği olmuştur. Talep belirsizliği üretim planlama ve stok kontrol modellerinin temel problemidir. Üretim planlarının doğru bir şekilde oluşturulabilmesi, kapasite planlaması, hammadde tedarigi, en uygun stok ve sipariş miktarlarının belirlenebilmesi için müşteri taleplerinin önceden tahmin edilmesi gerekir. Talep tahminleri geleceğe dair öngörülerini barındırdığı için belirsizlikler içerir. İşletmeler yüksek tepki hızına sahip olmak için yüksek hizmet düzeyi belirlemelidirler. Ancak yüksek hizmet düzeyi büyük miktarlarda stok tutmayı gerektirir ki bu da yüksek maliyet demektir. İşletmeler müşteri hizmet düzeyi ve stok maliyetleri arasında bir denge noktası bulmalıdırlar. Müşteri hizmet düzeyi işletme kârlılığını en çoklayan düzeydir. Optimum hizmet düzeyi fazla stok bulundurma ve yetersiz stok bulundurma maliyetlerini dengeleyen düzeydir. Optimum müşteri hizmet düzeyinin belirlenebilmesi talebin doğru tahminine bağlıdır (Behret, 2011).

Tedarik zincirindeki belirsizlikleri modelleyebilmek için genellikle stokastik teknikler kullanılmıştır. Bu durumda, belirsizlik kaynakları geçmiş verilerin analiziyle elde edilen olasılık dağılımlarıyla modellenmektedir. Ancak, geçmişe ait veri her zaman bulunamayabileceği gibi güvenilirliği ile ilgili sorunlar da olabilir. Üstelik bu parametreler stokastik yöntemlerle modellenenirse dahi uygulamada genellikle güçlükler yaşanır.

Giderek kısalan ürün yaşam çevrimleri talepteki değişkenliği ve istatistiksel analizlerin güvenilirliğini sürekli olarak azaltmaktadır. Bu nedenle olasılık teorisi pazar talebini ve stok parametrelerini değerlendirmek için uygun bir yaklaşım olmaktan çıkmıştır. Bunun yanı sıra eldeki verinin yeterliliği ve güvenilirliği ile ölçüm ve veri toplama yöntemlerinden kaynaklanabilecek belirsizlikler de düşünüldüğünde modelleme yaklaşımı olarak olasılık teorisinin yerine bulanık küme teorisi kullanılmalıdır.

Yapılan literatür araştırması sonucunda; bulanık küme teorisi belirsizlik kavramını ele alışı ve sözlü ifadeleri modelleyebilmesi nedeniyle belirsizlik ve değişkenliğin yönetimi için bu çalışmada seçilmiş modelleme yaklaşımıdır. Sonraki kısımlarda bulanıklık kavramı, bulanık karar verme ve bulanık hedef programlama ile ilgili bilgiler ve konu ile ilgili literatürde daha önce yapılmış olan çalışmalar incelenmiştir.

2. BULANIKLIK KAVRAMI

a. Bulanık Mantık

Lotfi A. Zadeh tarafından 1965'te ilk defa önerilen bulanık mantık, belirsizlikleri modellemek için yaygın olarak kullanılan bir yaklaşımdır. Özellikle 2000'li yılların başından itibaren uygulaması artan bulanık mantık, insan düşünme sistemi gibi yaklaşık olarak değerlendirmeyi ve modellemeyi olanaklı hale getirmektedir. Bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için oluşturulmuş bir matematik düzen olarak tanımlanabilir. Bilindiği gibi istatistikte ve olasılık teorisinde, belirsizliklerle değil kesinliklerle çalışılır, ama insanların yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir. Bulanık mantık ile klasik mantık arasındaki temel fark burada ortaya çıkmaktadır. Klasik matematiksel yöntemlerde verilerin kesin (crisp) olmasından dolayı, karmaşık sistemleri modellemek ve kontrol etmek zordur. Bulanık mantık, bu tip zorlukların üstesinden gelmek için kullanılan etkin bir düşünce yapısı üretir.

Gerçek hayatta karşılaşılan bazı nesne toplulukları, net olarak tanımlanmış üyelik kriterlerine sahip değildir. Örneğin, elemanları kartal, şahin, serçe olan kuşlar kümesine fasulye, fil ve yılanın girmeyeceği açıktır. Bununla birlikte yarası ve penguen için bunu söylemek o kadar kolay değildir. Elbette, 1'den çok daha büyük reel sayıların topluluğu, güzel kadınların topluluğu veya yakışıklı erkekler topluluğu bilinen matematiksel anlamda kümeler veya sınıflar oluşturmaz. Ancak, gerçek yaşamda, kesin olmayan şekilde tanımlanan topluluklar veya kümeler, insan düşüncesinde önemli bir rol oynar. Bulanık Küme Teorisi, yukarıdaki gibi kesin sınırları olmayan problemleri tanımlamak ve çözmek için geliştirilmiştir. Klasik Küme Teorisinden daha geniş bir çerçeve yaratan Bulanık Küme Teorisi, potansiyel olarak da daha geniş bir uygulama alanına sahiptir.

b. Bulanık Küme Teorisi

$\tilde{A}$, x ile gösterilen elemanların oluşturduğu X evrensel kümesinin bir alt kümesi olmak üzere, ilki bu alt kümenin bir elemanı, ikincisi ise bu elemanın

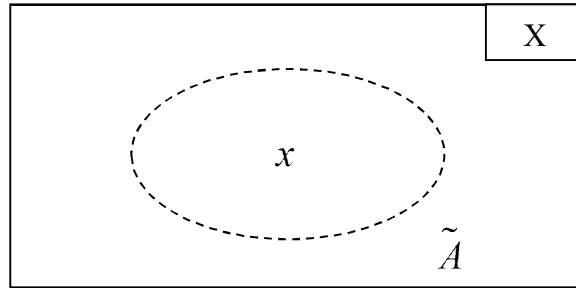
kümeyle yönelik derecesini gösteren sayısal değer olmak üzere, elemanları sıralı ikililer olan küme olarak tanımlanır. Bulanık kümelerin gösterimi aşağıdaki gibidir:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x), x \in X\} \quad (3.1)$$

Eşitlik (3.1) yerine Zadeh (1972), aynı zamanda aşağıdaki gösterim şeklini de önermiştir. X sonlu bir küme olduğunda $\{x_1, x_2, x_3, \dots, x_n\}$ $\tilde{A}$ bulanık kümesi şu şekilde belirtilir:

$$\tilde{A} = \mu_{\tilde{A}}(x_1)/x_1 + \dots + \mu_{\tilde{A}}(x_n)/x_n = \sum \mu_i(x_i)/x_i \quad (3.2)$$

Bulanık küme tanımında yer alan $\mu_{\tilde{A}}(x)$ terimine X 'in üyelik fonksiyonu denir. $\mu_{\tilde{A}}(x)$ $[0,1]$ kapalı aralığında değerler alabilir ve bu değer x elemanının $\tilde{A}$ kümesine ait olma derecesini gösterir (Şekil-10). X evrensel kümesinde tanımlanan, bulanık küme $\tilde{A}$ için üyelik fonksiyonu şöyle ifade edilir: $\mu : x \rightarrow [0,1]$



Şekil-10: X Evrensel Kümesinde $\tilde{A}$ Bulanık Altkümesi

c. Bulanık Kümelerde Temel Kavramlar

Bulanık kümelerde kullanılan semboller ve ifadeler ile klasik kümelerde kullanılan ifadelerin büyük bölümü birbirine benzemektedir.

Eşitlik: Klasik kümeler için eğer, her iki küme de aynı elemana sahipse eşittir denir. Bulanık kümelerde ise durum biraz değişik olmakla beraber, bulanık

üyelik dereceleri eşit olmasına bakılır. İki bulanık küme için, ancak ve ancak bütün $x \in X$ için $\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x)$ ise, $A=B$ denir.

Kapsama: Klasik kümelerde, eğer A 'nın tüm elemanları B 'de görülüyorsa $A \subset B$ idi. Bulanık kümeler için ise elemanların o kümelere olan üyelik dereceleri de dikkate alınmalıdır. Bütün $x \in X$ için ancak ve ancak $\mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$ ise $\tilde{A}$ bulanık kümesi $\tilde{B}$ bulanık kümesinin alt kümesidir denilir ve $A \subset B$ ile gösterilir.

Destek: Bir bulanık $\tilde{A}$ kümesinin desteği, X 'in alt kümesidir ve aşağıdaki gibi ifade edilir:

$$Des A = \{x \mid \mu_{\tilde{A}}(x) > 0 \text{ ve } x \in X\} \quad (3.3)$$

α Kesimi: Bulanık $\tilde{A}$ kümesini α seviye kümesi, X 'in kesin alt kümesidir ve şu şekilde gösterilir:

$$A_{\alpha} = \{x \mid \mu_{\tilde{A}}(x) \geq \alpha \text{ ve } x \in X\} \quad (3.4)$$

Geçiş Noktası: $\tilde{A}$ bulanık kümesinin geçiş noktaları, üyelik derecesi 0.5 olan X evrensel kümesinin elemanlarıdır.

Boş Küme: Bir $\tilde{A}$ bulanık kümesi, ancak ve ancak, $\forall x \in X$ için $\mu_{\tilde{A}}(x) = 0$ ise boş küme olarak tanımlanabilir.

Eşitlik: $\tilde{A}$ ve $\tilde{B}$ bulanık kümelerinin eşit olması için gerek ve yeter şart;

$$\forall x \in X \text{ için } \mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x) \quad (3.5)$$

Normallik: Bir bulanık $\tilde{A}$ kümesi, ancak ve ancak, $\sup_x \mu_{\tilde{A}}(x) = 1$ olması durumunda normaldir. “ $\sup A$ ”, $\tilde{A}$ bulanık kümesinde üyelik derecelerinden “en yüksek” olanı tanımlar.

Konvekslik: Bir bulanık $\tilde{A}$ kümesi, ancak ve ancak, aşağıdaki şartlar altında konvektir:

$$\mu_{\tilde{A}}(x) = (\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)) \quad \forall x_1, x_2 \in X, \quad \forall \lambda \in [0,1] \quad (3.6)$$

Kardinalite: $\tilde{A}$ bulanık kümesinin kardinalitesi;

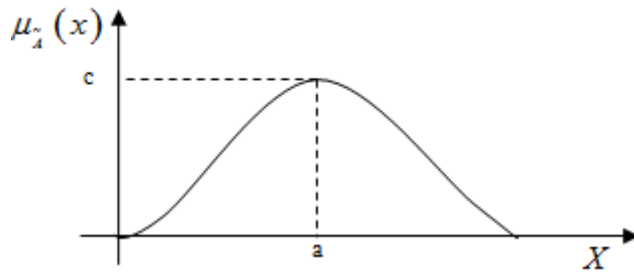
$$Kardinalite(A) = \sum_{x \in X} \mu_{\tilde{A}}(x) \quad (3.7)$$

ç. Üyelik Fonksiyonları

Birçok uygulama probleminde, bir takım bulanık sayıların üyelik fonksiyonlarının bilinmesine ihtiyaç vardır. Fakat bu üyelik fonksiyonu tahmini veya belirlenmesi metotları oldukça kapsamlıdır. Burada, bulanık sayılar için literatürde en çok ele alınan dört tip üyelik fonksiyonu incelenmiştir. Bunlar:

(1) Çan Şeklinde Üyelik Fonksiyonu

Üyelik fonksiyonunun grafiği Şekil-11'deki gibi olan $\tilde{A} = \{a; c\}$ sayısına çan şeklinde bulanık sayı denir, c, $\tilde{A}$ bulanık kümesinin yüksekliğini gösterir.



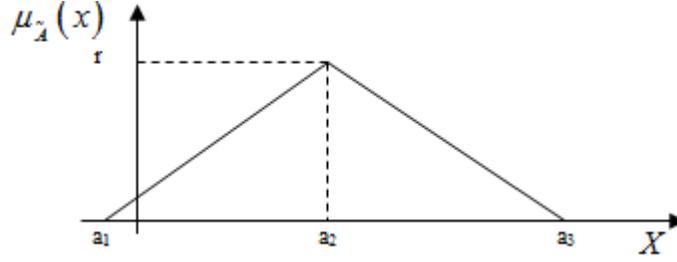
Şekil-11: Çan Şeklinde $\tilde{A}$ Bulanık Sayısının Grafiği

Çan şeklinde bulanık sayılar için üyelik fonksiyonu;

$$\mu_{\tilde{A}}(x) = c \cdot \exp\left[\frac{-(x-a)}{b}\right] \quad (3.8)$$

(2) Üçgensel Üyelik Fonksiyonu

Üyelik fonksiyonunun grafiği Şekil-12'deki gibi olan $\tilde{A} = \{a_1, a_2, a_3; r\}$ sayısına üçgensel bulanık sayı denir, r , $\tilde{A}$ bulanık kümesinin yüksekliğini gösterir.



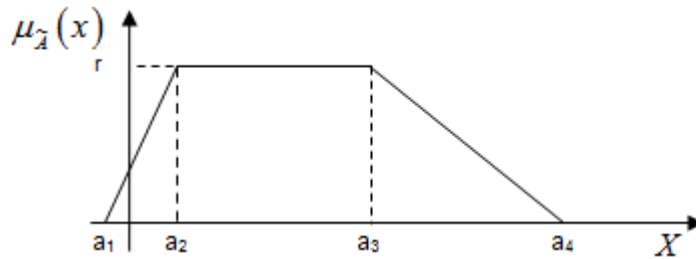
Şekil-12: Üçgensel $\tilde{A}$ Bulanık Sayısının Grafiği

Üçgensel bulanık sayılar için üyelik fonksiyonu;

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - a_1}{a_2 - a_1} \cdot r & a_1 \leq x \leq a_2 \text{ ise,} \\ \frac{a_3 - x}{a_3 - a_2} \cdot r & a_2 \leq x \leq a_3 \text{ ise,} \\ 0 & d.d. \end{cases} \quad (3.9)$$

(3) Trapezoid (Yamuksal) Üyelik Fonksiyonu

Üyelik fonksiyonunun grafiği Şekil-13'teki gibi olan $\tilde{A} = \{a_1, a_2, a_3, a_4; r\}$ sayısına trapezoid bulanık sayı denir r , $\tilde{A}$ bulanık kümesinin yüksekliğini gösterir.



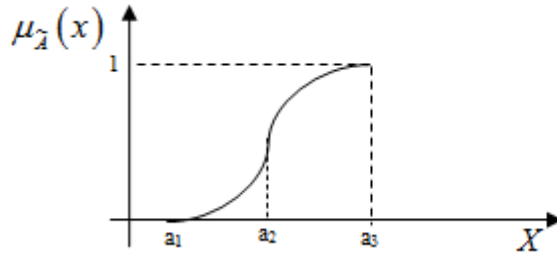
Şekil-13: Trapezoid $\tilde{A}$ Bulanık Sayısının Grafiği

Trapezoid bulanık sayılar için üyelik fonksiyonu;

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-a_1}{a_2-a_1} \cdot r & a_1 \leq x \leq a_2 \\ r & a_2 \leq x \leq a_3 \\ \frac{a_3-x}{a_3-a_2} \cdot r & a_3 \leq x \leq a_4 \\ 1 & d.d. \end{cases} \quad (3.10)$$

(4) S Şeklinde Üyelik Fonksiyonu

Üyelik fonksiyonunun grafiği Şekil-14'teki gibi olan $\tilde{A} = \{a_1, a_2, a_3; 1\}$ sayısına S şeklinde bulanık sayı denir.



Şekil 14: S Şeklinde $\tilde{A}$ Bulanık Sayısının Grafiği

S şeklinde bulanık sayılar için üyelik fonksiyonu;

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & x \leq a_1 \\ 2 \left(\frac{x-a_1}{a_2-a_1} \right)^2 & a_1 \leq x \leq a_2 \\ 1 - 2 \left(\frac{x-a_1}{a_3-a_1} \right)^2 & a_2 \leq x \leq a_3 \\ 1 & x \geq a_3 \end{cases} \quad (3.11)$$

d. Bulanık Kümelerde Temel İşlemler

Birleşim: $\tilde{A}$ ve $\tilde{B}$ bulanık kümelerine ait üyelik fonksiyonları sırasıyla $\mu_{\tilde{A}}(x)$ ve $\mu_{\tilde{B}}(x)$ olsun. İki bulanık kümenin birleşiminin üyelik fonksiyonu, bireysel üyelik fonksiyonlarının maksimumunu olarak tanımlanır.

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \quad \forall x \in X \quad (3.12)$$

Kesişim: Birleşim işleminde tanımlanan A ve B bulanık sayılarının kesişimi bireysel üyelik fonksiyonlarının minimumu olarak tanımlanır.

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \quad \forall x \in X \quad (3.13)$$

Tümleyen: $\tilde{A}$ ve B bulanık kümeleri için aşağıdaki şart sağlanıyor ise, A ve B için tümleyendir denir ve $B = A^C$ ve $A = B^C$ ile gösterilir.

$$\mu_{\tilde{A}}(x) = 1 - \mu_{\tilde{B}}(x), \quad \forall x \in X \quad (3.14)$$

Cebirsel Toplam: $\tilde{A}$ ve B bulanık kümelerinin cebirsel toplamı, $\tilde{C} = \tilde{A} + \tilde{B}$ ise, $\tilde{C} = \{(x, \mu_{\tilde{A} + \tilde{B}}(x)), x \in X\}$

$$\mu_{\tilde{A} + \tilde{B}}(x) = \mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x) - \mu_{\tilde{A}}(x) \cdot \mu_{\tilde{B}}(x) \quad (3.15)$$

Sınırlı Toplam: $\tilde{A}$ ve B bulanık kümelerinin sınırlı toplamı, $\tilde{C} = \tilde{A} \oplus \tilde{B}$ ise,

$$\mu_{\tilde{A} \oplus \tilde{B}}(x) = \min\{1, \mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x)\} \quad (3.16)$$

Cebirsel Fark: $\tilde{A}$ ve B bulanık kümelerinin cebirsel farkı, $\tilde{C} = \tilde{A} - \tilde{B}$ ise,

$$\mu_{\tilde{A} \ominus \tilde{B}}(x) = \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, \quad x \in X \quad (3.17)$$

Sınırlı Fark: $\tilde{A}$ ve B bulanık kümelerinin sınırlı farkı, $\tilde{C} = \tilde{A} \ominus \tilde{B}$ ise

$$\tilde{C} = \{(x, \mu_{\tilde{A} \ominus \tilde{B}}(x)), x \in X\}$$

$$\mu_{\tilde{A} \ominus \tilde{B}}(x) = \max\{0, \mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x) - 1\}, \quad x \in X \quad (3.18)$$

3. BULANIK ORTAMDA KARAR VERME

Geleneksel karar verme problemi sırasıyla; karar verici, amaç, karar ölçütü, seçenekler, olaylar ve sonuç olmak üzere altı bileşenden oluşur. Burada, amaç bileşeni bir maksimizasyon veya minimizasyon işlemi olarak yorumlanabilir. Fayda, kar, gelir ve maliyet fonksiyonları ise karar ölçütlerini oluşturur. Evrensel bir küme, seçenekler kümesi olarak kabul edilebilir. Evrensel kümenin hangi elemanlarının karar probleminin çözümü olarak kabul edilip edilemeyeceğini ifade eden kısıtlayıcı koşulları ise olayları belirler. Bu bakış açısından, mevcut durumu veya kısıtlayıcı koşullarını dikkate alarak, karar vericinin belirlediği amaç veya hedef doğrultusunda ilerleme çabası, karar problemlerinin özünü oluşturur (Özkan, 2003).

Bulanık bir ortamda karar verme problemi de yukarıda ele alınan bileşenlerle açıklanabilir. Burada, söz konusu bileşenlerden karar verici ve seçenekler kümesinde (evrensel küme) herhangi bir bulanıklık olmadığı kabul edilmiştir. Amaç ve karar ölçütü bileşenleri ise aşağıda açıklanan anlamda bulanıklık içerebilir. Karar verici amaç fonksiyonu için ulaşmak istediği erişim düzeyini bulanık olarak belirleyebilir. Ayrıca, karar ölçütünü gösteren fonksiyonun (kar, maliyet vb.) parametre değerleri bulanık sayılarla tanımlanabilir. Birbirini tamamlayan amaç ve karar ölçütü bileşenleri, *bulanık bir hedef* olarak ele alınabilir. Bulanık bir hedef, evrensel kümenin bir alt kümesi olan $\tilde{G}$ bulanık kümesi veya $\mu_{\tilde{G}}(x)$ üyelik fonksiyonu ile ifade edilebilir. $\mu_{\tilde{G}}(x)$ üyelik fonksiyonu $\mu_{\tilde{G}}(x) \in [0,1]$ koşulu ile belirli bir x vektörünün bulanık hedefe olan üyelik derecesini gösterir. $\mu_{\tilde{G}}(x) = 1$ iken ilgili hedefe tamamen ulaşıldığı, $\mu_{\tilde{G}}(x) = 0$ iken ilgili hedefe tamamen ulaşılmadığı ve $0 < \mu_{\tilde{G}}(x) < 1$ iken ilgili hedefe kısmen ulaşıldığı düşünülür. Diğer taraftan, olayları niteleyen kısıtlayıcıların parametre değerleri ve/veya sağ taraf sabitleri bulanık olabilir. Ayrıca, kısıtlayıcılarda yer alan $\leq$, $=$ ve $\geq$ ilişkilerinde bazı toleranslara izin verilebilir. Dolayısıyla, bulanık ortamdaki olaylar bileşeni *bulanık kısıtlayıcılar* olarak ele alınabilir. Bulanık bir kısıtlayıcı, evrensel kümede yer alan $\tilde{C}$ bulanık kümesi veya $\mu_{\tilde{C}}(x)$ üyelik fonksiyonu ile ifade edilebilir. Bulanık kısıtlayıcı kümesinin üyelik fonksiyonu, $\mu_{\tilde{C}}(x) \in [0,1]$ koşulu ile belirli bir x vektörünün bulanık kısıtlayıcıya üyelik derecesini gösterir. Burada, ilgili kısıtlayıcının tamamen

doğurulduğu durum $\mu_{\tilde{C}}(x)=1$ ile, ilgili kısıtlayıcının tamamen doğurulmadığı durum $\mu_{\tilde{C}}(x)=0$ ile ve ilgili kısıtlayıcının kısmen doğurulduğu durum ise $0 < \mu_{\tilde{C}}(x) < 1$ ile ifade edilir.

Bulanık hedefler ve/veya bulanık kısıtlayıcılarla verilen bir kararın bulanık olması kaçınılmazdır. Bellman ve Zadeh'e göre bulanık bir karar, verilen hedefler ve kısıtlayıcıların uzlaştırılmasından belirlenen bulanık bir küme olarak tanımlanır (Yager ve diğ., 1987). Bulanık hedef ve bulanık kısıtlayıcıların bir alt kümesi olan bulanık karar kümesi, $\tilde{D}$ kümesi veya $\mu_{\tilde{D}}(x)$ üyelik fonksiyonu ile ifade edilebilir. Bulanık karar kümesi, bulanık kısıtlayıcı doyumunun ve bulanık hedef başarımının eşanlı olarak karşılanma derecesini gösterir. Bulanık karar kümesi, genellikle “ $\tilde{G}$ hedefine ulaşmak ve $\tilde{C}$ kısıtlayıcısını doyumak” şeklinde ifade edilen bir kurala göre belirlenir (Kacprzyk ve Orlovski, 1987). Bu kural, bulanık karar kümesinin, hedef ve kısıtlayıcıların bir kesişim kümesi olarak tanımlanmasını gerektirir. Dolayısıyla, bulanık karar kümesi matematiksel olarak $\tilde{D} = \tilde{G} \cap \tilde{C}$ şeklinde ifade edilebilir. Burada, kesişim kümesi genellikle minimum işlemcisi ile belirlenir. n adet bulanık hedef ve m adet bulanık kısıtlayıcının bulanık karar kümesi aşağıdaki gibi tanımlanır (Zimmermann, 2001).

$$\mu_{\tilde{D}}(x) = \min \left[\mu_{\tilde{G}_i}(x), \mu_{\tilde{C}_j}(x) \right]; \quad \forall x \in X; \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (3.19)$$

Karar vericiler, bulanık karar kümesinin bulanıklıktan arındırılmasını veya $\mu_{\tilde{D}}(x)$ kümesinden geleneksel bir kararın verilmesini isteyebilirler. Böyle bir durum, bulanık karar kümesinin en yüksek üyelik dereceli elemanının belirlenmesi anlamına gelir. Bu ise, Bellman ve Zadeh'e göre matematiksel olarak aşağıdaki gibi ifade edilir (Yager ve diğ., 1987). Burada, x^M en iyileme yönündeki bir kararı ifade eder (Özkan, 2003).

$$\mu_{\tilde{D}}(x^M) = \max_{x \in U} \mu_{\tilde{D}}(x) = \max_{x \in U} \left\{ \min \left[\mu_{\tilde{G}_i}(x), \mu_{\tilde{C}_j}(x) \right] \right\}; \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (3.20)$$

4. BULANIK HEDEF PROGRAMLAMA

Hedef programlama modeli çok amaçlı programlama modellerinin bir türüdür. Optimizasyon düşüncesine dayanan çok amaçlı programlama modellerinde, birbiriyle çatışan amaçları kısıtlayıcı kümesine göre eşanlı olarak doyuran bir çözüm vektörünün belirlenmesi amaçlanır. Hedef programlama modelinde ise, karar vericinin doyurucu bulduğu bir çözüm belirlenmeye çalışılır. Bu nedenle, hedef programlama modelinin optimizasyon düşüncesinden daha çok bir doyum düşüncesine dayandığı söylenir (Özkan, 2003).

Bulanık ortamda karar verme kavramı ise ilk kez Bellman ve Zadeh (1970)'in bulanık amaç ve kısıtlara dayalı bir karar teorisi ile literatüre girmiştir. Daha sonra birçok araştırmacı gerçek yaşamdaki karar verme problemleri için bulanık mantık yöntemini kullanmışlardır. Zimmermann (1978), birden fazla amaç fonksiyonuna sahip olan problemi çözmek için bulanık doğrusal programlama modelini geliştirmiştir. Hedef programlamada bulanık küme teorisi ilk olarak Narasimhan (1980) tarafından kullanılmıştır. Narasimhan, bulanık hedefleri bulanık eşitlikler olarak kabul ederek, onları üçgensel üyelik fonksiyonları ile nitelemiştir. Zimmermann'ın bulanık doğrusal programlama modeli için geliştirdiği çözüm yaklaşımından esinlenen Narasimhan, bulanık hedef programlama modelinin çözümünü Bellman ve Zadeh'in bulanık karar kümesi kavramına dayanarak belirlemeye çalışmıştır. Bir doğrusal programlama dizisinin çözümünden oluşan Narasimhan yaklaşımında, oluşturulan alt problemlerden en yüksek λ değerini (üyelik derecesini) veren problemin çözümü, bulanık hedef programlama modelinin çözümü olarak kabul edilir. Bulanık hedeflerinin simetrik üçgensel üyelik fonksiyonlarıyla nitelenmesi halinde, Narasimhan yaklaşımına özdeş sonuçlar veren bir çözüm yöntemi Hannan (1981) tarafından geliştirilmiş ve bulanık hedef programlama modeli tek bir problem olarak formüle edilmiştir. Daha sonra çeşitli araştırmacılar bulanık hedef programlama için farklı çözüm yöntemleri önermişlerdir. Ignizio (1982), Wang ve Wang (1997) ve Cadenas ve Verdegay (2000) çok amaçlı problemlerde bulanık küme teorisinin kullanımını incelemişlerdir.

Bulanık matematiksel programlama Zimmerman (1978, 1991) tarafından, karar vericinin ulaşmak istediği amaç fonksiyonunun değeri için bir istek seviyesi

(hedef aralığı-aspirasyon seviyesi) kurulabileceğini ve kısıtların her birinin bir bulanık küme olarak modellenebileceğini ortaya atarak yaygınlaştırılmıştır. Bu yaklaşımda, belirsiz parametreler bulanık sayılar ve kısıtlar bulanık kümeler olarak kabul edilir. Bir kısıtın doyum derecesi, o kısıtın normalize edilmiş üyelik fonksiyonu açısından tanımlanır ve amaç fonksiyonu bir bulanık hedef veya açık bir fonksiyon iken küçük ölçekte kısıt ihlaline izin verilir.

Bulanıklık yaklaşımının belirsizlikle başedebilmenin diğer bir yolu olan stokastik programlama yaklaşımı üzerindeki avantajı, belirsiz parametrelerin herhangi bir olasılık dağılımını takip etmek zorunda olmaması ve belirsiz parametrelerin eşdeğer formülasyonunun belirsiz parametrelerin sayısındaki artışla birlikte büyük patlama yapmasına izin vermemesidir. Bulanıklık yaklaşımının dezavantajı ise, belirsizliğin kesin yapısını temsil etmekteki yetersizlik ve sonuçların bulanıklaştırma yaklaşımına bağlı olmasıdır.

Bu tez çalışmasında mevcut literatür dikkate alınarak gerçek hayat problemlerinde uygulanmak üzere hem belirsizlikleri modelleyebilen hem de çözümü zor olmayan bir bulanık hedef programlama modeli önerilmiştir.

5. TEDARİK ZİNCİRİ ÜRETİM/DAĞITIM PLANLAMASINDA BULANIK KÜME TEORİSİNİN KULLANILDIĞI ÇALIŞMALAR

Tedarik zinciri literatürünün kısa geçmişi incelendiğinde bulanık küme teorisinin kullanıldığı çalışmaların son zamanlarda artış gösterdiği anlaşılmaktadır. Petrovic ve diğ. (1999), bulanık ortamda seri bağlı bir tedarik zincirinin davranışını modellemişlerdir. Geliştirdikleri model, tedarik zincirindeki her bir stok için belirsizlik altında, kabul edilebilir maliyetler ile tedarik zincirinin kabul edilebilir bir hizmet düzeyini veren sipariş miktarlarını belirlemektedir. Model, özel amaçlı benzetim programı ile işletilerek sınırlı bir planlama dönemi boyunca seri bağlı tedarik zincirinin performans ölçüleri ve dinamikleri analiz edilmiştir.

Stok kontrol sistemleri tedarik zinciri sisteminin alt sistemlerinden biri olan malzeme akış sistemlerinin bir parçasıdır. Türkşen ve diğ. (2002), tedarik zincirinde bütün faaliyetlerin ve ilişkilerin belirsizlik içermeleri nedeniyle model geliştirmede

bulanık küme teorisinin kullanılmasının uygun olduğunu belirtmişler ve çalışmalarında tedarik zinciri modelini bulanık bakış açısıyla incelemişlerdir.

Roy ve Maitti (1997, 1998) çalışmalarında Klasik ESM modelini bulanık hedefler tanımlayarak modellemişler ve bulanık doğrusal olmayan programlama metodunu kullanarak bu model için bir çözüm üretmişlerdir. Daha sonra Klasik ESM modelini hem bulanık geometrik programlama hem de bulanık doğrusal olmayan programlama metodlarını kullanarak çözmüşlerdir. En son çalışmalarında ise çok ürünlü çok amaçlı stok modelini bulanık doğrusal programlama ve bulanık hedef programlama metodlarını kullanarak çözüm geliştirmişlerdir.

Chen ve Lee (2004) belirsiz piyasa talepleri ve ürün fiyatları ile tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için, çok ürünlü, çok kademeli ve çok dönemli planlama modeli önermişlerdir. Talepteki belirsizlik kesikli olasılık senaryoları ile modellenmiş ve satıcı ve alıcıların ürün fiyatlarında birbirleriyle uyuşmayan tercihlerini tanımlamak için bulanık kümeler kullanılmıştır. Wang ve Shu (2005), tedarik zincirindeki belirsizlikleri ele alan ve stok stratejilerini belirleyen alternatif bir çerçeve oluşturabilmek için olabilirlik teorisi ve genetik algoritma yaklaşımını birleştiren bulanık tedarik zinciri modeli sunmuşlardır.

Paksoy (2005), malzeme ihtiyaç kısıtı altında çok aşamalı bir tedarik zinciri ağı tasarımı problemi için karışık tamsayılı bir doğrusal programlama modeli geliştirmiştir. Xie ve diğ. (2006), tedarik zincirinin belirsiz müşteri talepleri ile işletildiği ve bulanık küme ile modellendiği sıralı bir tedarik zincirinde stok yönetimi ve kontrolünü sağlamak için iki seviyeli hiyerarşik bir metod tasarlamıştır. Chen ve diğ. (2007), dağıtım merkezlerinin yer seçiminin yapıldığı çok amaçlı tedarik zinciri ağı tasarım probleminde iki aşamalı bulanıklık yaklaşımını kullanmışlardır.

Xu ve diğ. (2008) Çin’de yer alan bir likör üreticisi için tedarikçiler, üretim tesisleri, dağıtım merkezleri ve müşterilerden oluşan tedarik zinciri ağ yapısında, açılacak üretim tesisi ve dağıtım merkezlerine karar vermek ve dağıtım stratejisini belirlemek üzere rassal bulanık talepli, çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeli önermişlerdir.

Liang (2006), üretim/dağıtım planlama karar problemini çözmek için taşıma maliyetlerini ve toplam teslim zamanını en azlayan bir bulanık çok amaçlı doğrusal programlama modeli önermiştir. Liang (2007), belirsizlik altında tedarik zincirinde bulanık hedefler ve kesin kısıtları olan bütünleşik üretim/dağıtım planlama karar problemlerini çözmek için bulanık doğrusal programlama yaklaşımını sunmuştur. Bu yaklaşımda karar vericinin bulanık hedeflerini temsil eden parçalı doğrusal üyelik fonksiyonları kullanılmış ve etkileşimli karar verme süreci yoluyla daha esnek bir doktrin başarılmıştır. Liang ve Cheng (2009), her bir işletme maliyeti kategorisi için paranın zaman değerini dikkate alarak, çok ürünlü, çok dönemli bir tedarik zincirindeki bütünleşik üretim/dağıtım planlama karar problemini çözmek için yeni bir bulanık çok amaçlı programlama yaklaşımı önermişlerdir.

Selim ve diğ. (2008), çok amaçlı üretim-dağıtım planlama probleminde hem merkezi hem de ademi merkezi tedarik zinciri tasarım yapısı arasında karşılaştırmalı analiz yapabilmek için bulanık hedef programlama tekniğini kullanmışlardır. Torabi ve Hassini (2008), tedarikçiden dağıtıma kadar olan tedarik zincirinde planlama kararlarını vermek üzere kapsamlı bir çalışma sunmuştur. Modelde planlamada karşılaşılan belirsizlikleri çözmek için amaçlarda ve kısıtlarda bulanık parametreler kullanılmıştır.

Işık ve Özdemir (2010) çalışmalarında, çok amaçlı, çok ürünlü ve çok dönemli bulanık bir bütünleşik üretim planlama problemini ele almışlar, problemin çözümü için etkileşimli olabilsel doğrusal programlama modeli önermişlerdir.

Mula ve diğ. (2010), geliştirdikleri çok aşamalı, çok ürünlü tedarik zinciri üretim/dağıtım planlama modelinde bulanık matematiksel programlamanın etkinliğini ortaya koymuşlardır. Problemden stokta tutma ve yoksatma kısıtlarda yer almaktadır. Talepteki belirsizlik bulanık ele alınmıştır. Model, üretim ve dağıtım safhasından oluşmaktadır. Üretim safhasında operatif amaçları gerçekleştirmek için, her bir üretim tesisindeki üretim kapasitesinin, üretim zamanı da dikkate alınarak, etkin bir şekilde tahsis edilmesi sağlanmaya çalışılmaktadır. Dağıtım safhasında ise talebin karşılanması ve stok yönetimi gibi üretim sonrası faaliyetler ele alınmıştır. Güvenlik stoku talepteki belirsizliğe karşı tampon vazifesi görmektedir.

Paksoy ve diğ. (2012), verilerin üçgensel bulanık sayılarla modellendiği çok amaçlı doğrusal programlama metodu ile yenilebilir bir sebze yağı üreticisinin bütünleşik tedarik zinciri ağına bulanık küme teorisini uygulamıştır. Paksoy ve Pehlivan (2012), çok aşamalı tedarik zinciri optimizasyonu için üçgensel ve trapezoidal üyelik fonksiyonlarına sahip bir bulanık doğrusal programlama modeli önermişlerdir. Model, tedarik zincirindeki bulanık kapasitelere sahip tesislerin en az maliyetle tedarik zinciri ağındaki konfigürasyonunu belirlemektedir. Paksoy ve diğ. (2013), çalışmalarında birden fazla amacın ödünleşmesinin sağlandığı, çoklu tedarikçi, üretici, dağıtım merkezi ve perakendecilerden oluşan, bütünleşik bulanık tedarik zinciri ağ sistemi ele almışlar, problemi çözmek için yeni bir karışık tamsayılı doğrusal programlama modeli önerilmişlerdir. Bu problem, bulanık parametreleri, hammadde kalitesine göre tedarikçi seçimini ve tedarikçi sözleşmelerini içermektedir.

Kabak ve Ülengin (2012), uzun dönemli kaynak atama, ürün tedariki ve üretim kararlarının verilebilmesini sağlayan tedarik zinciri ağ yapılandırma kararları için olabirsel doğrusal programlama modeli (ODP) geliştirmişlerdir. Önerilen ODP'nin amacı firmanın tedarik zinciri faaliyetleri sonucunda oluşan kârı en büyüklemeektir.

Pishvae ve diğ. (2012), belirsiz koşullar altında sosyal sorumlu tedarik zinciri ağ tasarımı için güçlü olabirsel programlama yaklaşımını önermişlerdir. Bu amacı gerçekleştirmek için önce, maliyeti en azlayan ve tedarik zinciri sosyal sorumluluğunu en çoklayan iki amaçlı matematiksel programlama modeli geliştirmişlerdir. Daha sonra, belirsiz parametreler ile etkili bir şekilde başa çıkmak için, yeni bir güçlü olabirsel programlama yaklaşımı önermişlerdir.

Fahimnia ve diğ. (2013), bütünleşik üretim/dağıtım planlaması modelleri ve teknikleri üzerine bir inceleme ve eleştiri yapmışlardır.

DÖRDÜNCÜ BÖLÜM

ÖNERİLEN BÜTÜNLEŞİK TEDARİK ZİNCİRİ PLANLAMA MODELİ

1. PROBLEM TANIMI

Matematiksel modelin gerçek hayattaki problemleri daha iyi yansıtabilmesi için çok dönemli, çok aşamalı, malzeme ihtiyaç kısıtları ile stokta tutma ve yok satmayı göz önünde bulunduran, bütünleşik tedarik zinciri ağındaki tesis yeri seçimi problemi için bulanık çok amaçlı doğrusal olmayan programlama modeli tasarlanmıştır.

Buna göre; planlama dönemi başlangıcından sonuna kadar bir firma üretim tesisleri için hammadde/yarı ürün gereksinimlerini belirli sayıdaki tedarikçiler aracılığı ile karşılamaktadır. Her tedarikçinin temin edebileceği hammadde/yarı ürün türü belirli ve kapasiteleri sınırlıdır. Model genel olarak tedarikçiyi seçmek ve sonra hangi tedarikçinin hangi hammaddeyi dağıtacağını ayrıntılı olarak belirlemek zorundadır.

Üretim tesislerine farklı bileşenler şeklinde gelen hammadde/yarı ürünler birtakım işlemlerden geçerek üretim için kullanılmaktadır. Bu ürünler belli oranlarda bir araya getirilerek müşteri bölgelerine ulaşacak bitmiş ürünleri oluşturmaktadır. Bu nedenle müşteri taleplerinin karşılanması için tedarik zincirindeki malzeme ihtiyaç planının bitmiş ürün içindeki hammadde/yarı ürün oranları dikkate alınarak gözden geçirilmesi gerekmektedir.

Ürünler üretim tesislerinden ve dağıtım merkezlerinden geçerek müşterilere ulaşmaktadır. Üretim tesislerinden gönderilen bitmiş ürünlerden belirli bir oranı ise çeşitli nedenlerle (defolu üretim, taşıma sırasında meydana gelen hasarlar vb.) dağıtım merkezlerine teslim edilememektedir. Dönem başlarında müşteri taleplerinde değişimler meydana geldiğinden, üretim tesislerinden dağıtım merkezlerine gönderilen ürünlerden bazıları elde kalmakta, bazıları ise müşteri taleplerini karşılayamamaktadır. Eğer üretim tesislerinden dağıtım merkezlerine gönderilen ürünler o dönem için müşteri taleplerinden fazla ise, firma bir sonraki dönem için

elde stokta tutmakta, bu durumda firma stokta tutma maliyetine katlanmaktadır. Eğer üretim tesislerinden dağıtım merkezlerine gönderilen ürünler o dönem için müşteri taleplerinden az ise firma karşılayamadığı ürün için yok satma maliyetine katlanmak zorunda kalmakta ve yok satılan miktar bir sonraki döneme devredilmektedir.

Modelin Varsayımları

Çalışma kapsamında EÜM (Yoksatmalı) modeli ve TSYP'nin birlikte ele alındığı matematiksel model ile ilgili varsayımlar aşağıdaki şekildedir:

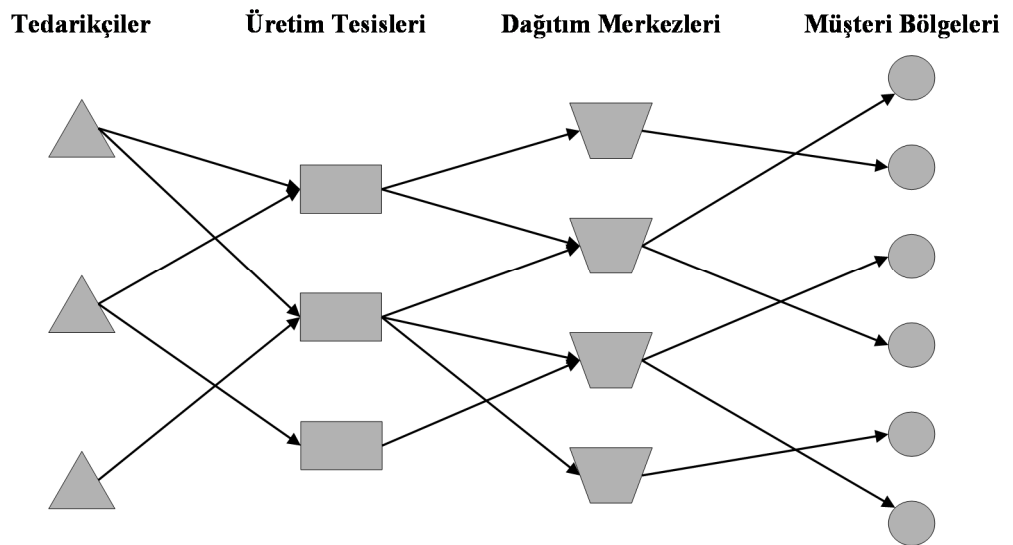
- Tedarikçiler, üretim tesisleri ve dağıtım merkezlerinin sayıları ve kapasiteleri belirlidir.
- Müşteriler bulundukları konum ve talebe bağlı olarak kendilerine en yakın dağıtım merkezinden hizmet almaktadır.
- Her müşteri bölgesi planlama dönemi boyunca yalnız bir dağıtım merkezinden hizmet almaktadır.
- Verilen siparişler dönem içinde bir anda alınmaktadır (Temin süresi sıfır).
- Üretim hızı talep hızından büyüktür.
- Yok satmaya izin verilmektedir.
- Amaç fonksiyonları da belirsiz aspirasyon seviyelerinin olması nedeniyle bulanık varsayılmıştır.

Klasik EÜM modelinden farklı olarak bu çalışmada ürünler için müşteri taleplerinin bulanık olduğu varsayılmıştır. Tedarik zinciri ağındaki belirsizliklerden en önemlisi de müşteri taleplerindeki belirsizliktir. Burada Klasik EÜM modelinin “talep hızı belirli ve her dönemde sabittir” varsayımı gevşetilmiş ve gerçek dünya durumlarına daha uygun bir EÜM modeli sunulmaya çalışılmıştır.

EÜM modeli, ESM modelinin genişletilmiş bir türüdür. Bu modelin varsayımları ESM modelinin varsayımları ile aynıdır. EÜM modelinin çözümü üretim, hazırlık ve stokta tutma maliyetlerinin toplamını en azlayan üretim miktarının bulunmasını sağlar. Türev alma işlemleri sonucu EÜM eşitlik (1.28)'deki gibi bulunur.

Literatürde EÜM modelindeki parametrelerin tamamının veya bir kısmının bulanık ele alındığı çok sayıda çalışma vardır. Chang (1999), üretim miktarının üçgensel bulanık sayılarla ifade edildiği bir Bulanık Ekonomik Üretim Miktarı (BEÜM) probleminin çözümü için genelleme ilkesine bağlı bulanık aritmetik işlemler kullanarak bulanık toplam maliyet fonksiyonunu tanımlamış ve elde edilen bulanık toplam maliyeti merkezi durulaştırma yöntemiyle durulaştırmıştır. Daha sonraki bir çalışmada Hsieh (2002), üretim miktarı haricindeki bütün parametrelerin trapezoid bulanık sayılarla ifade edildiği bir model ile üretim miktarı da dahil bütün parametrelerin bulanık sayılarla ifade edildiği iki ayrı model önermiştir. Chang ve Chang (2006) modellerinde, maliyet fonksiyonuna üretim maliyetini de dahil etmiş ve parti büyüklüğüne bağlı birim maliyet yapısını incelemişlerdir. Modellerinde üretim parti büyüklüğü, günlük talep ve günlük üretim hızı parametreleri ve birim ürün maliyet değeri bulanık sayılarla ifade edilmiştir. Chen ve diğ. (2007) çalışmalarında Hsieh (2002)'in modelini genişleterek indirimli bir fiyattan satılabilen kusurlu ürünler için BEÜM modeli önermişlerdir. Chen ve Chang (2008) BEÜM modelini genişleterek yeniden onarılamayan hasarlı ürünler için çözüm önermişlerdir. BEÜM modelleri için ayrıntılı literatür taraması Behret (2011)'in çalışmasında mevcuttur.

Tedarikçiler, üretim tesisleri, dağıtım merkezleri ve müşterilerden oluşan çok aşamalı tedarik zinciri ağ yapısı aşağıdaki şekilde görülmektedir.



Şekil-15: Önerilen Matematiksel Modelin Tedarik Zinciri Ağ Yapısı

Notasyon

İndisler:

i : Tedarikçiler, $i = 1, 2, 3, \dots, I$

j : Üretim tesisi sayısı, $j = 1, 2, 3, \dots, J$

k : Dağıtım merkezi sayısı, $k = 1, 2, 3, \dots, K$

l : Müşteri sayısı, $l = 1, 2, 3, \dots, L$

r : Hammadde çeşitleri, $r = 1, 2, 3, \dots, R$

t : Dönemler, $t = 1, 2, 3, \dots, T$

Karar Değişkenleri:

$X_{ijrt} = t$. dönemde i tedarikçisinden j üretim tesisine gönderilen r . hammadde miktarı.

$Y_{jkt} = t$. dönemde j üretim tesisinden k dağıtım merkezine taşınan ürün miktarı.

$Z_{klt} = t$. dönemde k dağıtım merkezinden l müşterisine gönderilen ürün miktarı.

$$w_{jt} = \begin{cases} 1, & t. \text{ dönemde } j \text{ üretim tesisi açılırsa} \\ 0, & \text{dd.} \end{cases}$$

$$z_{kt} = \begin{cases} 1, & t. \text{ dönemde } k \text{ dağıtım merkezi açılırsa} \\ 0, & \text{dd.} \end{cases}$$

$$v_{klt} = \begin{cases} 1, & t. \text{ dönemde } k \text{ d/m'nden } l \text{ müşteri bölgesine ürün gönderilirse.} \\ 0, & \text{dd.} \end{cases}$$

Amaç Fonksiyonu Maliyet Katsayıları:

$S_{ij} = i$ tedarikçisinden j üretim tesisine hammadde göndermenin birim taşıma maliyeti.

$T_{jk} = j$ üretim tesisinden k dağıtım merkezine gönderilen ürünlerin birim taşıma maliyeti.

$U_{kt} = k$ dağıtım merkezinden l müşterisine gönderilen ürünlerin birim taşıma maliyeti.

$f_{jt} = t$. dönemde üretim tesislerinin sabit maliyeti (açma ve işletme).

$g_{kt} = t$. dönemde dağıtım merkezlerinin sabit maliyeti (açma ve işletme).

$Cc_{kt} = t.$ dönemde k dağıtım merkezindeki satın alma maliyeti.

$Co_{kt} = t.$ dönemde k dağıtım merkezindeki sipariş verme maliyeti.

$Ch_{kt} = t.$ dönemde k dağıtım merkezindeki stokta tutma maliyeti.

$\pi_{kt} = t.$ dönemde k dağıtım merkezindeki yok satma maliyeti.

Teknolojik Katsayılar:

$a_{irt} = t.$ dönemde i tedarik merkezinin $r.$ hammadde kapasitesi.

$b_{jt} = t.$ dönemde j üretim tesisinin ürün kapasitesi.

$c_{kt} = t.$ dönemde k dağıtım merkezinin ürün kapasitesi.

$\delta_r =$ Birim bitmiş ürün içindeki $r.$ hammaddenin kullanım miktarı.

$\gamma_{jkt} = t.$ dönemde j üretim tesisinden k dağıtım merkezine gönderilemeyen ürünlerin oranı.

Genel Veriler:

$d_{lt} = t.$ dönemde l müşteri bölgesinin ürün talebi.

$P_t = t.$ dönemde açılacak toplam üretim tesisi sayısının üst sınırı.

$W_t = t.$ dönemde açılacak toplam dağıtım merkezi sayısının üst sınırı.

$$\text{Min } f_1 = \sum_{ijt} S_{ij} X_{ijrt} + \sum_{jkt} T_{jk} Y_{jkt} + \sum_{klt} U_{kl} Z_{klt} + \sum_{jt} f_{jt} w_{jt} + \sum_{kt} g_{kt} z_{kt} \quad (4.1)$$

$$\text{Min } f_2 = \text{Hazırlık (Sipariş)} + \text{Üretim (Satın Alma)} + \text{Stokta Tutma} + \text{Yok Satma Maliyeti}$$

$$\begin{aligned} &= \sum_{klt} Cc_{kt} \cdot Z_{klt} + \sum_t Co_{kt} \cdot \frac{\sum_{klt} Z_{klt}}{\sqrt{\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} (Ch_{kt} + \pi_{kt})}{\sum_{kt} Ch_{kt} \cdot \pi_{kt} \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)}}} \\ &+ \frac{\sum_{klt} Ch_{kt} \cdot Z_{klt}}{2 \cdot \sqrt{\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} (Ch_{kt} + \pi_{kt})}{\sum_{kt} Ch_{kt} \cdot \pi_{kt} \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)}}} \left[\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} (Ch_{kt} + \pi_{kt})}{\sum_{kt} Ch_{kt} \cdot \pi_{kt} \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)} \cdot \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right) - \sqrt{\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} Ch_{kt}}{\sum_{kt} (Ch_{kt} + \pi_{kt}) \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)}} \right] \cdot \left(\frac{1}{\sum_{jkt} Y_{jkt} - \sum_{klt} Z_{klt}} + \frac{1}{\sum_{klt} Z_{klt}} \right) \\ &+ \frac{\sum_{kt} \pi_{kt}}{2} \cdot \left(\sqrt{\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} Ch_{kt}}{\sum_{kt} (Ch_{kt} + \pi_{kt}) \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)}}} \right)^2 \cdot \frac{\sum_{klt} Z_{klt}}{\sqrt{\frac{2 \sum_{kt} Co_{kt} \sum_{klt} Z_{klt} (Ch_{kt} + \pi_{kt})}{\sum_{kt} Ch_{kt} \cdot \pi_{kt} \left(1 - \frac{\sum_{klt} Z_{klt}}{\sum_{jkt} Y_{jkt}}\right)}}} \left(\frac{1}{\sum_{jkt} Y_{jkt} - \sum_{klt} Z_{klt}} + \frac{1}{\sum_{klt} Z_{klt}} \right) \end{aligned} \quad (4.2)$$

Kısıtlar:

$$\sum_j X_{ijrt} \leq a_{irt} \quad \forall_i \in I, \quad \forall_r \in R, \quad \forall_t \in T \quad (4.3)$$

$$\sum_i X_{ijrt} - \delta_r \sum_k Y_{jkt} = 0 \quad \forall_j \in J, \quad \forall_r \in R, \quad \forall_t \in T \quad (4.4)$$

$$\sum_k Y_{jkt} \leq b_{jt} w_{jt} \quad \forall_j \in J, \quad \forall_t \in T \quad (4.5)$$

$$\sum_j w_{jt} \leq P_t \quad \forall_t \in T \quad (4.6)$$

$$\sum_j Y_{jkt} - \gamma_{jkt} \sum_j Y_{jkt} \geq \sum_l Z_{klt} \quad \forall_k \in K, \quad \forall_t \in T \quad (4.7)$$

$$Z_{klt} \geq d_{lt} v_{klt} \quad \forall_k \in K, \quad \forall_l \in L, \quad \forall_t \in T \quad (4.8)$$

$$\sum_l Z_{klt} \leq c_{kt} z_{kt} \quad \forall_k \in K, \quad \forall_t \in T \quad (4.9)$$

$$\sum_k z_{kt} \leq W_t \quad \forall_t \in T \quad (4.10)$$

$$\sum_l v_{klt} = 1 \quad \forall_k \in K, \quad \forall_t \in T \quad (4.11)$$

$$w_{jt}, z_{kt}, v_{klt} \in \{0,1\} \quad (4.12)$$

$$X_{ijrt}, Y_{jkt}, Z_{klt} \geq 0 \quad \forall_{i,j,k,l,r,t} \quad (4.13)$$

Geliştirilen bulanık çok amaçlı programlama modelinde birbiriyle çatışan iki amaç vardır. Birinci amaç, tedarik zincirindeki sabit tesis açma ve işletme maliyetleri ile uzaklıklara bağlı olarak belirlenen taşıma maliyetlerinin en azlanmasıdır. İkinci amaç; EÜM modeline göre hazırlık, üretim, stokta tutma ve yok satma maliyetlerinin en azlanmasıdır.

Kısıt (4.3), gönderilen hammadde/yarı ürünlerin tedarik merkezinin kapasitesini aşamayacağını garanti etmektedir. Kısıt (4.4), tedarik merkezlerinden üretim tesislerine gönderilen hammadde/yarı ürünlerin bitmiş ürün üretmek için gerekli olan hammadde/yarı ürün kadar olmasını garanti etmektedir (Birinci aşama

denge kısıtı). Kısıt (4.5), üretim tesislerinden gönderilen ürünlerin açık olan tesislerin kapasitesinden fazla olmamasını garanti eder. Kısıt (4.6), her bir dönem için açılabilir toplam üretim tesisi sayısının üst sınırını belirlemektedir. Kısıt (4.7), üretim tesislerinde üretilen ürünler ile dağıtım merkezlerine gönderilemeyen defolu ürün miktarının toplamının en az dağıtım merkezine gönderilen miktar kadar olmasını sağlar (İkinci aşama denge kısıtı). Kısıt (4.8), dağıtım merkezlerinden müşteri bölgelerine gönderilen ürün miktarının en az müşteri talebi kadar olmasını garanti etmektedir. (Talep kısıtı). Kısıt (4.9), dağıtım merkezi kapasite kısıtıdır. Kısıt (4.10), her bir dönem için açılabilir toplam dağıtım merkezi sayısının üst sınırını belirlemektedir. Kısıt (4.11), bir müşteri noktasının planlama dönemi boyunca açık olan en uygun dağıtım merkezinden yalnızca bir kez hizmet almasını sağlamaktadır. Kısıt (4.12) ve (4.13), işaret kısıtlarıdır.

2. ÖNERİLEN MATEMATİKSEL MODELİN BULANIK HEDEF PROGRAMLAMA FORMÜLASYONU

Belirsizliği ele alış tarzı bulanıklık yaklaşımını diğer yaklaşımlardan farklı kılmaktadır. Diğer belirsizlik metotlarının aksine bulanık matematiksel programlama belirsiz parametreleri bulanık sayılar ve bu belirsiz parametrelerle ilişkili olan kısıtları bulanık kümeler olarak ele alır. Bir kısıtın doyum derecesi, üyelik fonksiyonu değerinin bir ölçüde kısıt ihlali anlamına geldiği, kısıtların üyelik fonksiyonu cinsinden tanımlanır. Bu şekilde bir miktar kısıt ihlaline izin verilmektedir. Bulanık hedef programlama için geliştirilen çözüm yaklaşımlarının birçoğunda bulanık hedefler, işlemsel kolaylık sağlaması nedeniyle Zimmermann tipi üyelik fonksiyonları ile nitelenmiştir. Bulanık hedef programlama matematiksel olarak aşağıdaki gibi ifade edilmiştir:

$$\begin{aligned} \text{Max } f(x) &= \{f_1(x), f_2(x), \dots, f_k(x)\}^T \\ \text{Öyle ki :} & \\ g_j(x) &\leq \tilde{b}_j \quad j = 1, 2, \dots, m \end{aligned} \tag{4.14}$$

Burada x karar değişkenlerini, $\tilde{b}$ bulanık hedef ve mevcut kaynak değerlerini göstermektedir. Eğer j 'inci kısıt, x karar değişkeni cinsinden $g_j(x) \leq \tilde{b}$

ise ve belirsiz parametre b , b ile $b+\Delta b$ aralığında değişiyorsa, kısıtın üyelik fonksiyonu aşağıdaki gibi olur.

$$\mu_{g_j}(x)(b) = \begin{cases} 1 & \text{eğer } g_j(x) < b_j \\ 1 - \frac{g_j(x) - b_j}{\Delta b_j}, & \text{eğer } b_j \leq g_j(x) \leq b_j + \Delta b_j \\ 0 & \text{eğer } g_j(x) > b_j + \Delta b_j \end{cases} \quad (4.15)$$

Yukarıdaki kısıtta, b 'nin yüksek değerleri için çoğunlukla kısıt doyumunun başarılması arzu edilir. Bu durum için pratik bir benzetme, piyasa koşullarının kesin olarak bilindiği talebin değeri b ve belirsiz koşullar altında aynı talep b 'nin $b+\Delta b$ aralığında olmasına izin verilmektedir. Δb 'nin değeri uzman görüşüne dayanarak veya geçmişteki elde mevcut verilerden elde edilebilir. Burada belirtilmesi gereken önemli bir husus, bulanık matematiksel programlamada belirsiz parametrelerin, bilinen eğilim özelliklerine sahip herhangi bir olasılık dağılımına uymak zorunda olmamasıdır. Bulanık matematiksel programlama Δb aralık ölçüsünün belirsiz parametreler aralığında olmasına izin verir. Bulanık hedefler literatürde üçgensel, ikiz kenar, parçalı doğrusal, iç bükey biçimli parçalı doğrusal, yarı iç bükey biçimli parçalı doğrusal, s biçimli parçalı doğrusal ve dış bükey biçimli parçalı doğrusal şeklinde farklı özellikteki üyelik fonksiyonları ile nitelenmiştir. Söz konusu üyelik fonksiyonları genellikle karar verici ile görüşülerek oluşturulur. Üyelik fonksiyonları, doğası gereği her ne kadar birçok değişik tipte olabilse de, burada doğrusal monoton artan ve doğrusal monoton azalan farz edilmiştir. Yukarıdaki matematiksel model için başka değişik tipte üyelik fonksiyonları seçilebilir. Ancak, bu durum problemin karmaşıklığını artıracaktır.

Kısıtlar için üyelik fonksiyonlarının tanımlanmasına ilave olarak, benzer üyelik fonksiyonları karar vericilerin beklentilerini yansıtan alt ve üst limitleri ile amaç fonksiyonları için de formüle edilebilir. Yukarıdaki örnek problem için amaç fonksiyonu sınırları aynı optimizasyon probleminin iki farklı durumu, sağ taraf değeri b_j ve $b_j+\Delta b_j$, çözülerek hesaplanabilir. Amaç fonksiyonunun bu alt ve üst sınırları sırasıyla $f^{\min}$ ve $f^{\max}$ olarak tanımlanabilir. Bu durumda amaç fonksiyonları için üyelik fonksiyonları aşağıdaki gibi olur.

$$\mu_{f_k}(x) = \begin{cases} 1 & \text{eğer } f_k(x) > f_k^{\max} \\ 1 - \frac{f_k^{\max} - f_k(x)}{f_k^{\max} - f_k^{\min}} & \text{eğer } f_k^{\min} \leq f_k(x) \leq f_k^{\max} \\ 0 & \text{eğer } f_k(x) < f_k^{\min} \end{cases} \quad (4.16)$$

Bulanık matematiksel programlamanın son aşaması, kısıt doyumunu en fazla yaparken amaç fonksiyonlarının çatışan hedeflerini dengeleyen bir optimizasyon modeli kurmaktır. Bellman ve Zadeh (1970), bütün kısıtların ve amaç fonksiyonlarının en küçük üyelik fonksiyonlarının en çoklanması ile başarılabileceğini aşağıdaki şekilde göstermişlerdir.

$$\max_{x \geq 0} \min_{j,k} \{ \mu_{f_k}(x), \mu_{g_j}(x) \} = \max_{x \geq 0} \mu_{\bar{D}}(x) \quad (4.17)$$

Yukarıdaki üyelik fonksiyonları ile bu kriteri kullanarak, optimizasyon problemi böylece aşağıdaki formülasyonda λ karar değişkeninde gösterilen optimizasyon problemine dönüştürülür.

$$\begin{aligned} & \text{Max } \lambda \\ & \text{s.t. :} \\ & \lambda \leq \mu_{f_k}(x), \quad k = 1, 2, \dots, n \\ & \lambda \leq \mu_{g_j}(x), \quad j = 1, 2, \dots, m \end{aligned} \quad (4.18)$$

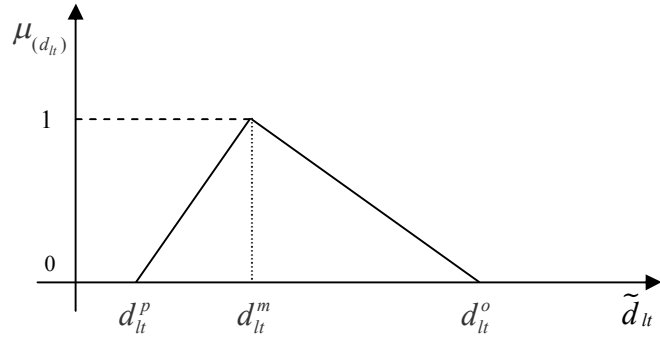
Son olarak, yukarıdaki optimizasyon problemi λ değeri ve optimal karar değişkeni değerini ortaya koyan karar değişkeni vektörü x için çözülür.

a. Bulanık Talep Kısıtının Çözüm Yöntemi

Önerilen modelde belirsiz talep parametreleri üçgensel bulanık sayılar ile gösterilmiştir. Üçgensel bulanık sayılar, bu parametrelerin kötümser, beklenen ve iyimser değerlerini gösterecek şekilde tanımlanmıştır. Böylece bulanık bir talep kısıtı oluşturulmuştur. Lai ve Hwang (1992, 1994) matematiksel işlemleri daha etkin yapabilmek için üyelik fonksiyonlarını simetrik üçgensel fonksiyonlar ile

göstermiştir. Buna göre talepteki belirsizliği temsil etmek üzere $\tilde{d}_{lt}^m$ üçgenin modal noktası olup hedefin en olası değerini, $\tilde{d}_{lt}^p$ ve $\tilde{d}_{lt}^o$ ise sağ ve sol ayakların ortaya olan mesafeleri olup, hedefin $\tilde{d}_{lt}^m$ 'den sapma miktarını (en kötümser ve en iyimser değerlerini) gösterir. Talep kısıtı için üçgensel üyelik fonksiyonu $\mu_{(d_{lt})}$ ile gösterilir.

Üçgensel bulanık sayılar, verilerin kolay elde edilebilmesi, bulanık aritmetik işlemlerin basitliği ve esnekliğinden kaynaklanan hesaplama kolaylığı nedeniyle literatürde yaygın olarak kullanılmaktadır (Ör. Lai ve Hwang, 1992, 1994; Liang 2006, 2007, 2008, 2009, 2011). Üçgensel üyelik fonksiyonunun grafik gösterimi aşağıdaki gibidir:



Şekil-16: Üçgensel Bulanık Sayı $\mu_{(d_{lt})}$ Grafıksel Gösterimi

Yukarıda formüle edilen orijinal bulanık çok amaçlı programlama modelindeki Kısıt (4.8)'den anımsanacağı üzere, müşteri talebi d_{lt} en olası ve en az olası değerlere sahip üçgensel bulanık sayıdır. Bu çalışmada d_{lt} değerini durulaştırmak için Lai ve Hwang (1992)'ın önerdiği ağırlıklı ortalama metodu kullanılmıştır. Eğer kabul edilebilir en küçük üyelik derecesi, α , verilirse, buna karşılık gelen denk talep kısıtı aşağıdaki şekilde olur:

$$Z_{klt} \geq \left(w_1 \cdot d_{lt,\alpha}^p + w_2 \cdot d_{lt,\alpha}^m + w_3 \cdot d_{lt,\alpha}^o \right) \cdot v_{klt} \quad \forall_{k,l,t} \quad (4.19)$$

Burada en kötümser, en olası ve en iyimser değerleriyle $w_1+w_2+w_3=1$ 'dir. Uygulamada ağırlıklar karar vericinin tecrübe ve bilgisine göre belirlenir. Bu

çalışmada bulanık talep kısıtı için $w_1=4/6$, $w_2=w_3=1/6$ olan, Lai ve Hwang (1992)'in önerdikleri en olası değer kavramı benimsenmiştir. Bunun temel nedeni, en olası değer genellikle en önemli olması ve böylece daha fazla ağırlık almasıdır.

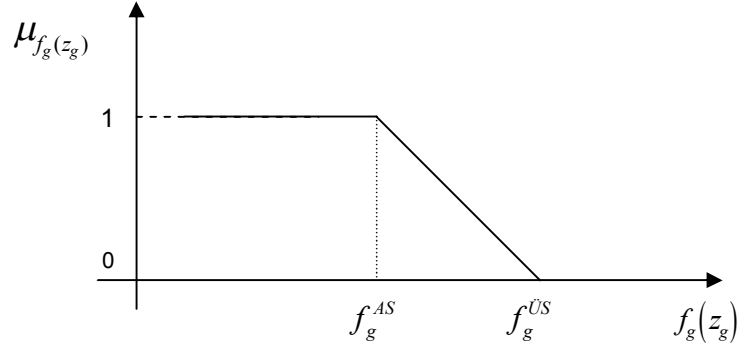
Gerçek hayatta bir karar verici, önerilen yaklaşımı kullanırken uygun bir α -kesim seviyesini belirlemek zorundadır. α -kesim seviyesindeki bir değişiklik, hedef değeri ve dolayısıyla çıktı sonuçlarını etkileyecektir. α -kesim değeri ve üç kritik noktanın göreceli ağırlıkları, karar vericinin tecrübe ve bilgisine dayalı olarak sübjektif bir şekilde ayarlanabilir.

b. Bulanık Çok Amaçlı Programlama Modelinin Çözüm Yöntemi

Dağıtım ağlarının doğası gereği, bu konuda geliştirilen modellerin büyük bir bölümü karışık tamsayılı doğrusal programlama şeklindedir. Ancak, bütünleşik tedarik zinciri ağı tasarımı problemlerinde koşullar doğrusal olmayan bir yapı arz ettiğinden, doğrusal olmayan fonksiyonların kullanıldığı bir problem çözüm mantığı uygulanmalıdır. Gerçek hayatta karşılaşılan problemler için geliştirilen karar modellerinin kısıtlarından en az biri veya amaç fonksiyonunun doğrusal olmadığı durumlar için geliştirilen tüm kavram ve teknikler “Doğrusal Olmayan Programlama” adı altında incelenmektedir. Geliştirilen model, gerçek durumları yansıtma ve/veya tesis yeri seçimi problemlerinin belirgin yönlerine odaklanma düşüncesi ile birçok ilave özelliği içinde barındırmaktadır. EÜM modelindeki üretim hızı ve talep hızı arasındaki ilişkinin doğrusal olmayan yapısı önerilen modelde doğrusal olmayan programlama yaklaşımını zorunlu kılmaktadır.

Bu çalışmada geliştirilen bulanık çok amaçlı doğrusal olmayan programlama modeli, bütün amaçları temsil etmek üzere doğrusal üyelik fonksiyonlarını kullanarak, bulanık aritmetik işlemlerdeki hesaplama kolaylığı ve esnekliğini ortaya koymaktadır. Amaç fonksiyonlarının doğrusal monoton azalan üyelik fonksiyonları ile grafik gösterimi aşağıdaki gibi tanımlanır:

$$f_g(z_g) = \begin{cases} 1, & z_g \leq z_g^{AS}, \\ \frac{z_g^{\bar{US}} - z_g}{z_g^{\bar{US}} - z_g^{AS}}, & z_g^{AS} < z_g < z_g^{\bar{US}}, \quad g = 1, 2, \dots, K \\ 0, & z_g \geq z_g^{\bar{US}} \end{cases} \quad (4.20)$$



Şekil-17: Doğrusal Monoton Azalan $\mu_{f_g(Z_g)}$ Grafik Gösterimi

Burada g 'inci amaç fonksiyonu için Z_g^{AS} ve $Z_g^{\bar{US}}$ sırasıyla pozitif ideal çözüm (AS, alt sınır) ve negatif ideal çözüm (ÜS, üst sınır)'ü göstermektedir. Doğrusal üyelik fonksiyonları, karar vericiden amaç fonksiyon değeri aralığını $[Z_g^{AS}, Z_g^{\bar{US}}]$ seçmesi istenerek belirlenebilir. Bir amaç fonksiyonu için olası değer aralığı, karar verici ve/veya uzmanın bilgi ve tecrübesine dayanarak tahmin edilebilir ve karar vericinin denk üyelik fonksiyon derecesi normalde $[0,1]$ aralığındadır.

Bütün bulanık kümeleri birleştirmek için Bellman ve Zadeh (1970)'in bulanık karar verme minimum işlemcisi kullanılır. Diğer bir deyişle, belirlenen üyelik fonksiyonlarını kullanarak, bütünleşik tedarik zinciri ağında tesis yeri seçimi problemi için bulanık çok amaçlı karışık tamsayılı doğrusal olmayan programlama modeli, λ karar değişkeninde gösterilen aşağıdaki optimizasyon problemine dönüştürülür.

$$\begin{aligned}
& \text{Max } \lambda \\
& \text{s.t.} \\
& \lambda \leq f_g(z_g) \quad \forall_g \\
& \text{Kısıt } (4.3-4.7), (4.9-4.11), (4.19) \\
& w_{jt}, z_{kt}, v_{klt} \in \{0,1\} \\
& X_{ijrt}, Y_{jkt}, Z_{klt} \geq 0 \quad \forall_{i,j,k,l,r,t} \\
& 0 \leq \lambda \leq 1
\end{aligned} \tag{4.21}$$

c. Önerilen Modelin Çözüm Algoritması

Adım 1: Bütünleşik tedarik zinciri ağında tesis yeri seçimi problemini çözmek için orijinal bulanık çok amaçlı doğrusal olmayan programlama (DOP) modelini formüle et.

Adım 2: En küçük üyelik derecesi α 'yı belirle, daha sonra ağırlıklı ortalama metodunu kullanarak bulanık eşitsizlik kısıtlarını duru hale getir.

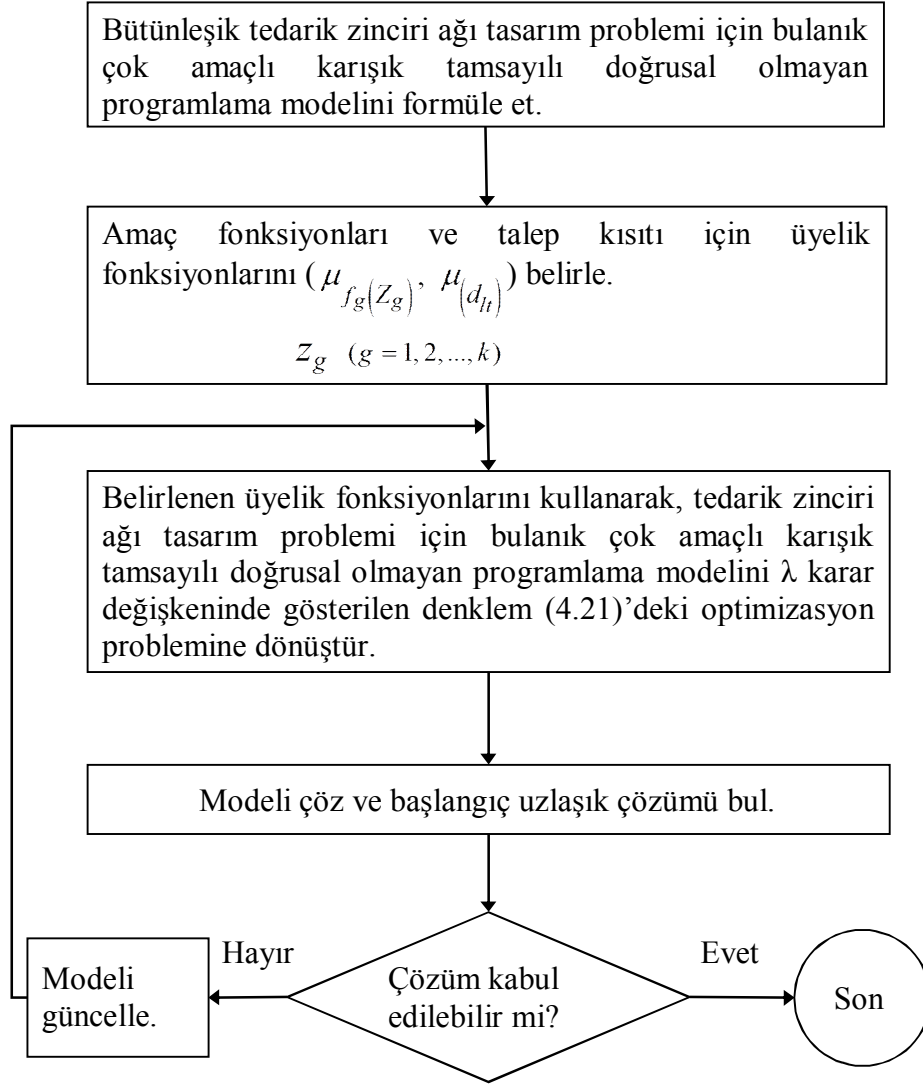
Adım 3: Her bulanık amaç için alt sınır (AS, Pozitif İdeal Çözüm) ve üst sınır (ÜS, Negatif İdeal Çözümü) değerlerini belirle ve bunlara karşılık gelen doğrusal monoton azalan/artan üyelik fonksiyonlarını tanımla.

Adım 4: Bütün bulanık kümeleri birleştirmek için minimum işlemcisini kullanarak yardımcı değişken λ 'yı ata ve orijinal problemi denk sıradan DOP problemine dönüştür.

Adım 5: Sıradan tek amaçlı DOP modelini çöz ve başlangıç uzlaşık (trade-off) çözümü elde et.

Adım 6: Bulanık karar verme sürecini uygula ve güncelleştir. Eğer karar verici başlangıç çözümünü yeterli bulmaz ise, model, tercih edilen bir doyurucu çözüm elde edilinceye kadar güncelleştirilir.

Geliştirilen modelin akış diyagramı aşağıdadır:



Şekil-18: Önerilen Matematiksel Modelin Akış Diyagramı

3. MODELİN TEST EDİLMESİ

a. Örnek Problem Uygulaması

Bu bölümde örnek bir problem üzerinden anlatılan model için gerçek hayattaki endüstriyel durumu yansıtan küçük parametreler kümesi oluşturulmuştur. Model ile ilgili tesislerin ürün kapasite aralıkları Tablo-6'da verilmiştir. Ürün taşıma maliyetleri, 100 km.de harcanan yakıt miktarının, yakıtın litre fiyatı ve tesisler arasındaki uzaklıkların çarpılması ile belirlenmiştir. Tesisler arasındaki uzaklıklar sırasıyla tedarik merkezleri ile üretim tesisleri arası 100-380 km., üretim tesisleri ile dağıtım merkezleri arası 70-220 km., dağıtım merkezleri ile

müşteri bölgeleri arası 10-60 km. aralığında rassal uniform (düzgün) dağılımdan üretilmiştir. Ürün taşıma maliyetleri, 100 km.'de harcanan yakıt miktarının, yakıtın litre fiyatı ve tesisler arasındaki uzaklıkların çarpılması ile belirlenmiştir. Modelde bir birim bitmiş ürün üretmek için kullanılacak hammadde/yarı ürün miktarı 3-5 birim, üretim tesislerinden dağıtım merkezlerine gönderilen bitmiş ürünlerden çeşitli nedenlerle (defolu üretim, taşıma sırasında meydana gelen hasarlar vb.) teslim edilemeyen belirli bir oranı temsil etmek üzere modele dahil edilen fire oranı 0.1-0.4 birim, müşteri bölgelerindeki talepler ise her dönem için 100-150 birim ürün olacak şekilde rassal uniform dağılımdan üretilmiştir. Diğer maliyet parametreleri ve aralıkları Tablo-7'de verilmiştir.

Tablo-6: Tesislerin Ürün Kapasiteleri (Birim)

Tesisler	Aralık
Tedarik Merkezleri	500-1000
Üretim Tesisleri	300-500
Dağıtım Merkezleri	200-400

Tablo-7: Maliyet Parametreleri (TL.)

Parametreler	Aralık
Üretim Tesisi Açma ve İşletme Maliyeti	700-950
Dağıtım Merkezi Açma ve İşletme Maliyeti	190-350
Satın Alma Maliyeti	50-70
Sipariş Verme Maliyeti	200-300
Stokta Tutma Maliyeti	5-35
Yok Satma Maliyeti	20-40

Önerilen modelin çözüm yordamının kolay anlaşılmasını sağlamak için, 3 dönemi kapsayan, her dönem için 3 aday tedarik merkezi, 3 aday üretim tesisi, 3 aday dağıtım merkezinin olduğu, planlama dönemi boyunca toplam 5 müşteri bölgesine hizmet veren küçük boyutlu bir örnek problem oluşturulmuştur.

Yukarıda parametre aralıkları verilen ve boyutları tanımlanan bütünleşik tedarik zinciri ağında tesis yeri seçimi problemini çözmek için önerilen bulanık çok

amaçlı doğrusal olmayan programlama modelinin etkileşimli çözüm aşamaları aşağıda sunulmuştur.

İlk önce; çok aşamalı, çok dönemli ve çok amaçlı bütünleşik tedarik zinciri ağında tesis yeri seçimi problemini çözmek için orijinal bulanık çok amaçlı doğrusal olmayan programlama modeli denklem (4.21)'deki gibi formüle edilerek kısıt (4.19) kullanılır ve $\alpha=0.5$ kesim katsayısında bulanık eşitsizlik kısıtı duru hale getirilir. Daha sonra; karar vericinin her bulanık sayının üçgensel dağılımının en olası değerini kesin değer olarak belirlediği varsayımı ile örnek problem, amaç fonksiyonlarının alt ve üst sınırlarını belirlemek için, Lai ve Hwang (1992) tarafından önerilen yöntem kullanılarak durulaştırılan talep kısıtı ile birlikte diğer kısıtlar altında her amaç tek başına çözülür. Bulanık amaç fonksiyonlarının monoton azalan doğrusal üyelik fonksiyonları denklem (4.20) yoluyla gösterimi aşağıdadır:

$$f_1(z_1) = \begin{cases} 1, & z_1 \leq 51771.768 \\ \frac{84938.424 - z_1}{33166.656} & 51771.768 < z_1 < 84938.424 \\ 0, & z_1 \geq 84938.424 \end{cases} \quad (4.22)$$

$$f_2(z_2) = \begin{cases} 1, & z_2 \leq 32949.862 \\ \frac{36707.484 - z_2}{3757.622} & 36707.484 < z_2 < 32949.862 \\ 0, & z_2 \geq 36707.484 \end{cases} \quad (4.23)$$

Son olarak; bütün bulanık kümeleri birleştirmek için minimum işlemcisi kullanılarak yardımcı değişken λ atanır ve orijinal problem denk sıradan doğrusal olmayan programlama problemine dönüştürülür. Tablo-8, örnek test probleminin tek amaçlı doğrusal olmayan programlama modelinden elde edilen en iyi çözümleri, her bulanık amaç için karşılık gelen aralık değerleri ve elde edilen başlangıç uzlaşık çözümlerini göstermektedir.

Tablo-8: Amaç Fonksiyonlarının Çözümleri ve Aralık Değerleri

Amaç Fonk.	DOP-1 (Min Z_1)	DOP-2 (Min Z_2)	(AS, ÜS)	Başlangıç Çözümü (Max λ)
λ	100%	100%	-	0.8286
Z_1 (TL)	51771.768*	84938.424	(51771.768, 84938.424)	56133.798
Z_2 (TL)	36707.484	32949.862*	(32949.862, 36707.484)	33465.219

* Sıradan tek amaçlı doğrusal olmayan programlama modeli ile elde edilen yaklaşık çözümleri ifade etmektedir.

Ayrıca, eğer karar verici başlangıç çözümünü yeterli bulmaz ise model, tercih edilen bir doyurucu çözüm elde edilinceye kadar güncelleştirilebilir. Önerilen yaklaşımda karar vericinin, elde edilen çözümlerin amaç fonksiyon değerlerine dayanarak, bu çözümlerini iyileştirmek istediği ve bulanık amaç fonksiyonlarının alt ve üst sınırlarını güncelleştirdiği varsayılın. Bu durumda, iyileştirilmiş çözümlerin amaç fonksiyon değerleri $Z_1 = 56107.198$ TL. ve $Z_2 = 33465.219$ TL. elde edilir. Önerilen modelin başlangıç çözümleri ile iyileştirilmiş çözümlerinin amaç fonksiyon değerlerinin karşılaştırılması Tablo-9’da gösterilmiştir.

Tablo-9: Bulanık Amaçların Alt ve Üst Sınır Değerleri

Karar Değişkeni	Başlangıç Çözümü	İyileştirilmiş Çözüm
$(Z_g^{AS}, Z_g^{ÜS})$	$(Z_1^{AS}, Z_1^{ÜS}) = (51771.77, 84938.42)$ $(Z_2^{AS}, Z_2^{ÜS}) = (32949.86, 36707.49)$	$(Z_1^{AS}, Z_1^{ÜS}) = (51771.77, 56133.80)$ $(Z_2^{AS}, Z_2^{ÜS}) = (32949.86, 33465.22)$
Z_1 (TL)	56133.798	56107.198
Z_2 (TL)	33465.219	33465.219
λ	0.8286	0.00000012163

Önerilen çok amaçlı doğrusal olmayan programlama modeli, GAMS/DICOPT çözücüsü (MINOS ve BARON çözücüleri de denemiş, ancak birincisinde daha kötü çözümler elde edilmiş, ikincisinde ise herhangi bir çözüme ulaşamamıştır.) ile Windows XP Pro Intel i3 işlemci 3,20 GHz, 3 GB RAM özelliklerine sahip bilgisayar ile çözülmüştür. Önerilen matematiksel modelin karışık tamsayılı programlama modeli olmasından dolayı GAMS/DICOPT çözücüsü arka planda dal-sınır (branch-and-bound) algoritması kullanmaktadır. Önerilen model için oluşturulan küçük boyutlu problemin GAMS/DICOPT program kodları EK-A’da, program çıktıları EK-B’dedir.

b. Sayısal Analiz

Önerilen modeldeki çözüm yaklaşımını kullanmanın birkaç nedeni şöyle açıklanabilir: İlk olarak, önerilen model etken çözüm vermektedir. Bir uç nokta (uygun çözüm) vektörü $\bar{x} \in S$ (S uygun çözüm bölgesidir.) ve buna karşılık gelen amaç fonksiyonu $Z_g(\bar{x})$ ancak ve ancak onu baskılayan başka hiçbir uygun çözüm $x \in S$ öyle ki $Z_g(x) \leq Z_g(\bar{x}) \quad \forall_g$ ve $Z_g(x) < Z_g(\bar{x})$ en az bir g için, $g=1,2,\dots,k$ mevcut değilse bir etken çözümdür (Zimmermann 1978, Hannan 1981). Bu çalışmada, doğrusal üyelik fonksiyonları ve minimum işlemcisinin kullanıldığı, bulanık matematiksel programlama için etken çözümler veren Bellman ve Zadeh (1970)'in bulanık karar verme kavramı ile Zimmermann (1978)'in bulanık programlama metodu benimsenmiştir. Ayrıca, Zimmermann (1976, 1978) en çoklayan çözümün bulanık kümeleri birleştiren minimum işlemcisi için neden her zaman etken çözüm verdiğini açıklamıştır. Tablo-8'de gösterildiği gibi, durulaştırılmış talep kısıtı ile birlikte diğer kısıtlar altında birinci amaç fonksiyonu tek başına çözüldüğünde $Z_1 = 51771.768$, $Z_2 = 36707.484$ elde edilirken, aynı kısıtlar altında ikinci amaç fonksiyonu tek başına çözüldüğünde $Z_1=84938.424$, $Z_2=32949.862$ elde edilmiştir. Önerilen bulanık çok amaçlı matematiksel programlama modeli kullanılarak elde edilen başlangıç çözümleri ($Z_1 = 56133.798$ ve $Z_2 = 33465.219$), tek amaçlı doğrusal olmayan programlama modelleri DOP-1 ve DOP-2 yoluyla bulunan yaklaşık çözümler ile karşılaştırıldığında, yine bir etken çözüm olduğunu göstermektedir.

İkincisi, geleneksel deterministik modeller, birden fazla amacın çatışan doğası ve gerçek hayat bütünleşik tedarik zinciri planlama problemlerinin parametreleri ile ilgili belirsizlik nedeniyle etkili çözümler elde etmenin uygun yöntemleri değildir. Önerilen modelden elde edilen çözümler ise, Tablo-8 ve Tablo-9'da görüldüğü gibi, birbirine bağımlı amaç fonksiyonları arasındaki çatışma ve etkileşimli ödünleşmeyi sağlayarak gerçek hayat problemlerine daha yakın çözümler vermektedir. Böylece önerilen model; bir tarafta uzaklıklara bağlı toplam maliyet ve sabit açma/işletme maliyetlerini, diğer tarafta satın alma, sipariş verme, stokta tutma ve yok satma maliyetlerini eşzamanlı olarak en azlamaktadır.

Önerilen modelin diğer bir avantajı, bütünleşik tedarik zinciri ağındaki tesis yeri seçimi problemini çözmek için bulanık hedeflerde doğrusal üyelik fonksiyonları

ile bulanık talep kısıtında üçgensel dağılımı ($\alpha=0.5$ kesim seviyesinde, ağırlıklı ortalama yöntemi) benimseyerek, model formülasyonu ve aritmetik işlemler açısından büyük hesaplama kolaylığı ve esneklik sağlamaktadır. Burada formüle edilen orijinal bulanık çok amaçlı doğrusal olmayan programlama problemi, bulanık kümeleri birleştirmek için minimum işlemcisini kullanan Zimmermann (1976, 1978)'ın bulanık hedef programlama yöntemine dayanarak denk doğrusal olmayan programlama problemine dönüştürülmüştür.

Ayrıca; başlangıç ve iyileştirilmiş çözümlerin karşılaştırılması, bulanık amaç fonksiyonlarının alt ve üst sınırlarındaki değişimlerin hesaplama sonuçlarını etkilediğini göstermektedir. Tablo-8 ve Tablo-9'da bulanık amaçların alt ve üst sınırları için başlangıç değerleri $(Z_1^{AS}, Z_1^{ÜS}) = (51771.768, 84938.424)$ TL. ve $(Z_2^{AS}, Z_2^{ÜS}) = (32949.862, 36707.484)$ TL. Başlangıç çözümlerinin nihai amaç fonksiyon değerleri $Z_1 = 56133.798$ TL ve $Z_2 = 33465.219$ TL'dir. Bu bulgu göstermektedir ki, önerilen modelin uygulanmasında her bulanık amaç fonksiyonu için karşılık gelen doğrusal üyelik fonksiyonlarının etkili şekilde belirlenebilmesi için karar vericinin bulanık amaç fonksiyonlarının uygun alt ve üst sınırlarını belirlemesi gerekmektedir. Uygulamada, her bulanık amaç fonksiyonu için sıradan tek amaçlı doğrusal olmayan programlama çözümleri çoğunlukla amaç fonksiyonlarının alt ve üst sınırları için başlangıç noktaları olarak kullanılmıştır. Başlangıç çözümlerinden elde edilen değerlerin alt ve üst sınır aralığı içinde kaldığı görülmektedir.

Önerilen yaklaşımda karar vericinin, başlangıç çözümünün amaç fonksiyon değerlerine dayanarak, daha iyi bir çözüm elde etmek istediği ve bulanık amaç fonksiyonlarının alt ve üst sınırlarını yenilediği varsayılmıştır. Tablo-10, önerilen modelin çözümünden elde edilen sonuçları göstermektedir. Bulanık amaç fonksiyonlarının alt ve üst sınırları başlangıç çözümünün amaç fonksiyon değerlerine bağlı olarak $(Z_1^{AS}, Z_1^{ÜS}) = (51771.768, 56133.798)$ TL. ve $(Z_2^{AS}, Z_2^{ÜS}) = (32949.862, 33465.219)$ TL. olarak güncelleştirilmiştir. Son olarak iyileştirilmiş çözümler $Z_1 = 56133.798$ TL ve $Z_2 = 33465.219$ TL elde edilmiştir. Buna bağlı olarak $Q^*=374.590$ ve $b^*=1.642$ birim elde edilmiştir. İyileştirilmiş çözümünden elde edilen sonuçlara göre; her dönemde açılacak tesisler ve sayıları ile hangi müşterinin hangi dağıtım merkezinden hangi dönemde hizmet alacağı Tablo-10'da gösterilmiştir.

Tablo-10: Örnek Problem için Önerilen Matematiksel Modelin Çıktı Sonuçları

Karar değişkeni	Çıktı sonuçları
Amaç fonksiyonu ve λ değeri	$Z_1= 56107.198$ TL, $Z_2= 33465.219$ TL, $\lambda= 0.00000012163$
X_{ijrt} (birim)	$X_{1212}=178.4$, $X_{1213}= 944$, $X_{1232}= 786$, $X_{1223}= 556$, $X_{1211}=178.44$, $X_{1232}= 786$, $X_{1233}= 457.019$, $X_{3321}= 299.7$, $X_{3322}= 275.70$, $X_{3331}= 512$, $X_{3332}= 528$
Y_{jkt} (birim)	$Y_{213}= 119.41$, $Y_{232}= 137.78$, $Y_{233}= 160.17$, $Y_{322}= 114.8$, $Y_{331}= 115.96$
Z_{klt} (birim)	$Z_{153}= 107.47$, $Z_{222}= 103.33$, $Z_{311}= 104.37$, $Z_{333}= 128.13$, $Z_{342}= 12$
$v_{klt} (0,1)$	$v_{153}= v_{222}= v_{311}= v_{333}= v_{342}= 1$, diğerleri 0
$w_{jt} (0,1)$	$w_{22}= w_{23}= w_{31}= w_{32}= 1$, diğerleri 0
$z_{kt} (0,1)$	$z_{13}= z_{21}= z_{22}= z_{31}= z_{32}= z_{33}=1$, diğerleri 0
Q^* (EÜM)	374.59 birim.
b^* (Yok satılan miktar)	1.642 birim.

Bu durumda daha iyi bir etken çözüm elde edilmiştir. Bütünleşik tedarik zinciri ağında çok amaçlı üretim/dağıtım planlama problemi için, karar verici normalde tercih edilen etken çözümün mümkün olduğunca amaç fonksiyonunun en iyi alt sınırına (Pozitif İdeal Çözüm, tek amaçlı doğrusal olmayan programlama çözümü) yakın olanını seçecektir.

Sonuç olarak, önerilen model karar vericiye doyurucu tercih edilen bir çözüm elde edilinceye kadar bulanık/belirsiz veri ile ilgili parametreleri güncelleştirme imkânı sağlayan, bulanık karar verme sürecini kolaylaştıran sistematik bir çerçeve sağlamaktadır. Daha da önemlisi bu çalışmada ele alınan çözüm yaklaşımı, çözüm süreci boyunca tercih edilen doyurucu etken bir çözüme ulaşmaya kadar karar vericinin arama yönünün belirlenmesini sağlar.

c. Test Problemlerinin Üretilmesi

Oluşturulan modelin etkinliği ve geçerliliği, gerçek durumların modele mümkün olduğu kadar doğru bir şekilde yansıtılmasına bağlıdır. Önerilen

modelin performansı ile ilgili kabul edilebilir bir güven seviyesini başarabilmek ve çözüm yordamının tutarlı sonuçlar verdiğini test edebilmek için yukarıda tanımlanan parametre aralıkları ile Borland C++ Builder programlama dili kullanılarak değişik boyutlarda test problemleri üretilmiştir. Literatürde benzer çalışma olmadığından test problemleri tarafımızdan üretilmiştir. Test problemi üretme C++ kodları EK-C’de sunulmuştur. Test problemi üretme kaba (pseudo) kodu aşağıdaki şekilde verilmiştir.

```
Başla
  Problem kombinasyonlarını belirle (Küçük boyutlu, Orta boyutlu, Büyük boyutlu)
  Her problem kombinasyonu için;
    Dönem sayısı, T
    Müşteri sayısı, L
    Dağıtım merkezi sayısı, K
    Üretim Tesisi sayısı, J
    Tedarik merkezi sayısı, I
    Bilgilerini belirle.
    Kombinasyondaki her bir problem için;
      Ürün için gerekli hammadde sayısını belirle. U(2,4)

      Her dönemdeki her tedarik merkezindeki hammadde kapasitelerini üret.
      U(4500,6000) (Kesikli düzgün dağılım)

      Her dönemdeki her üretim tesisinin kapasitesini üret. U(2500,3000)
      (Kesikli düzgün dağılım)

      Her dönemdeki her dağıtım merkezinin kapasitesini üret. U(200,400)
      (Kesikli düzgün dağılım)

      Her dönemdeki üretim tesisi sabit açma maliyetlerini üret. U(700,950)
      (Sürekli düzgün dağılım)

      Her dönemdeki dağıtım merkezi sabit açma maliyetlerini üret. U(190,320)
      (Sürekli düzgün dağılım)

      Her dönemdeki her müşterinin talebini üret. U(100,150)
      (Kesikli düzgün dağılım)

      Ürün içindeki her hammadde kullanım miktarını üret. U(2,4)
      (Kesikli düzgün dağılım)

      Her tedarik merkezi ile her üretim tesisi arasındaki uzaklıkları üret. U(100,380)
      (Sürekli düzgün dağılım)

      Her üretim tesisi ile her dağıtım merkezi arasındaki uzaklıkları üret. U(70,220)
      (Sürekli düzgün dağılım)

      Her dağıtım merkezi ile her müşteri arasındaki uzaklıkları üret. U(10,60)
      (Sürekli düzgün dağılım)

      Her dönemdeki her dağıtım merkezinin;
        Satın alma maliyetini üret. U(50,70) (Sürekli düzgün dağılım)
        Sipariş verme maliyetini üret. U(200,300) (Sürekli düzgün dağılım)
        Stokta tutma maliyetini üret. U(5,35) (Sürekli düzgün dağılım)
        Yok satma maliyetini üret. U(20,40) (Sürekli düzgün dağılım)

      Kombinasyondaki problem sayısı bitince sonlandır
    Problem kombinasyon sayısı bitince sonlandır
Son
```

Şekil-19: Test Problemi Üretme Kaba (Pseudo) Kodu

Problem setleri, problemin içerdği tedarik merkezi-üretim tesisi-dağıtım merkezi-müşteri bölgesi-dönem (i-j-k-l-t) sayıları dikkate alınarak oluşturulmuştur.

Üretilen test problemleri, GAMS/DICOPT çözücüsü ile elde edilen çözüm süreleri ve amaç fonksiyonu değerlerinin büyüklüğü (maliyetler) dikkate alınarak dört farklı tipe ayrılmıştır. Problem tipleri sınıflandırılırken dönem sayısı ve müşteri sayısının problemin çözüm süresi ile amaç fonksiyonu değerlerindeki artışa neden olan en önemli iki faktör olduğu tespit edilmiştir. Müşteri sayısındaki artışa bağlı olarak ise, sırasıyla aday dağıtım merkezleri, aday üretim tesisleri ve aday tedarik merkezleri arasından açılacak tesis sayıları da artmaktadır. Buna göre; dönem sayısı üst sınırı 7, müşteri sayısı üst sınırı 10 ve aday tesis sayıları üst sınırı 5 olan test problemleri I. Tip (küçük boyutlu); dönem ve müşteri sayısı 10-20 aralığında, aday tesis sayıları 5-7 aralığında olan test problemleri II. Tip (orta boyutlu); dönem sayısı 10, müşteri sayısı 20-40 aralığında ve aday tesis sayıları 10-20 aralığında olan test problemleri III. Tip (büyük boyutlu); dönem sayısı 15-20 aralığında, müşteri sayısı 20-40 aralığında ve aday tesis sayıları 10-20 olan test problemleri IV. Tip (çok büyük boyutlu) problemler olarak sınıflandırılmıştır. Test problemi sınıflandırma aralıkları aşağıdaki tabloda verilmiştir.

Tablo-11: Test Problemi Sınıflandırma Aralıkları

Parametreler	Problem Tipleri			
	I. Tip	II. Tip	III. Tip	IV. Tip
Aday Tedarik Merkezi Sayısı	3–5	5–7	10–20	10–20
Aday Üretim Tesisi Sayısı	3–5	5–7	10–20	10–20
Aday Dağıtım Merkezi Sayısı	3–5	5–7	10–20	10–20
Müşteri Sayısı	5–10	10–20	20–40	20–40
Dönem Sayısı	3–7	10–20	10	15–20

Üretilen test problemlerinden seçilen farklı boyutta 40 adet probleme ilişkin bilgiler Tablo-11’de verilmiştir.

Tablo-11: Test Problemi Boyutları

Tip	Sıra Nu.	Boyut	Aday tedarik merkezi sayısı (i)	Aday üretim tesisi sayısı (j)	Aday dağıtım merkezi sayısı (k)	Müşteri sayısı (l)	Dönem sayısı (t)
I	1	4-3-4-5-3	4	3	4	5	3
	2	4-3-3-7-3	4	3	3	7	3
	3	3-3-5-7-3	3	3	5	7	3
	4	5-3-5-10-3	5	3	5	10	3
	5	4-4-3-5-5	4	4	3	5	5
	6	3-4-4-10-5	3	4	4	10	5
	7	4-4-4-5-7	4	4	4	5	7
	8	3-4-3-7-7	3	4	3	7	7
	9	5-3-4-10-7	5	3	4	10	7
	10	5-5-5-10-7	5	5	5	10	7
II	11	7-5-5-10-10	7	5	5	10	10
	12	5-6-7-10-10	5	6	7	10	10
	13	6-7-7-20-10	6	7	7	20	10
	14	7-5-6-10-15	7	5	6	10	15
	15	7-5-7-15-15	7	5	7	15	15
	16	5-6-7-15-15	5	6	7	15	15
	17	6-5-5-20-15	6	5	5	20	15
	18	6-7-5-20-15	6	7	5	20	15
	19	6-5-5-20-20	6	5	5	20	20
	20	7-7-5-20-20	7	7	5	20	20
III	21	15-15-10-20-10	15	15	10	20	10
	22	10-20-10-20-10	10	20	10	20	10
	23	15-20-20-20-10	15	20	20	20	10
	24	15-10-10-30-10	15	10	10	30	10
	25	10-15-10-30-10	10	15	10	30	10
	26	10-10-15-30-10	10	10	15	30	10
	27	20-20-15-30-10	20	20	15	30	10
	28	10-10-10-40-10	10	10	10	40	10
	29	20-15-10-40-10	20	15	10	40	10
	30	15-15-15-40-10	15	15	15	40	10
IV	31	20-10-15-30-15	20	10	15	30	15
	32	20-20-20-30-15	20	20	20	30	15
	33	20-20-10-40-15	20	20	10	40	15
	34	20-10-15-30-20	20	10	15	30	20
	35	20-15-20-30-20	20	15	20	30	20
	36	15-10-10-40-20	15	10	10	40	20
	37	15-10-20-40-20	15	10	20	40	20
	38	20-10-20-40-20	20	10	20	40	20
	39	20-15-20-40-20	20	15	20	40	20
	40	20-20-20-40-20	20	20	20	40	20

Her bir problem boyutu için üretilen örnek problemlerin 72000 saniye (20 saat) çözülmesiyle elde edilen değerler Tablo-12-15'te verilmiştir. 30 ve daha büyük sayıda dönem (t) ile 50 ve daha büyük sayıda müşteri (l) içeren büyük boyutlu problemlerin çözüm süresi 20 saati aştığından, üretilen test problemlerinden seçilen 40 adet örnek problem için kaynak kullanımı (resource usage) 72000 saniye ile sınırlandırılmıştır. Bu süre içerisinde bulanık amaç fonksiyonlarının uygun alt ve üst sınırlarını belirlenmiş, önce başlangıç çözümleri, daha sonra başlangıç çözümünün amaç fonksiyon değerlerine dayanarak, iyileştirilmiş çözümlere ulaşılmıştır. Tablolar incelendiğinde; iyileştirilmiş çözümden elde edilen değerlerin, başlangıç çözümlerinden elde edilen değerlere oranla amaç fonksiyonlarının alt sınır değerlerine daha çok yakınsadığı görülmektedir. Bu durum da karar vericinin (tedarik zinciri yöneticisinin) daha doyurucu etken çözümlere ulaşmasını sağlamaktadır.

Bununla birlikte, GAMS/DICOPT optimizasyon programının doğrusal olmayan programlama modellerinde optimal çözümleri garanti edememesi nedeniyle, üretilen test problemlerinde yaklaşık çözümler elde edilmiştir.

Test problemleri üretiminde ve GAMS/DICOPT çözücüsü ile çözümlerin elde edilmesinde, Windows XP Pro Intel i3 işlemci 3,20 GHz, 3 GB RAM özelliklerine sahip bilgisayar kullanılmıştır.

Tablo-12: I. Tip Örnek Test Problemlerinin Hesaplama Sonuçları

Sıra Nu.	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>t</i>	Amaç Fonksiyonu/CPU Süresi	DOP-1 (Min Z_1)	DOP-2 (Min Z_2)	(AS/ÜS)	Başlangıç Çözümü	İyileştirilmiş Çözüm
1	4	3	4	5	3	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 46669.21405 36874.43677 0.084	100% 122177.6619 35595.80226 0.092	- (46669.21405,122177.6619) (36874.43677,35595.80226) -	0.894681 54498.59695 35730.46626 0.36	0.267e-8 54300.19695 35730.46626 0.25
2	4	3	3	7	3	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 74648.49 47920.71 0.05	100% 139568.05 46151.88 0.033	- (74648.49, 139568.05) (47920.71, 46151.88) -	0.953145 77690.32 46234.76 0.12	0.000193 77689.73 46234.74 0.14
3	3	3	5	7	3	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 128841.3603 57308.46337 0.266	100% 174471.6473 48772.67737 0.105	- (128841.3603,174471.6473) (57308.46337,48772.67737) -	0.724717 140475.6707 51122.43319 0.34	0.000201 140168.0708 51122.43319 0.28
4	5	3	5	10	3	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 123351.3491 74538.53382 0.259	100% 247188.8224 70126.05313 0.512	- (123351.3491,247188.8224) (74538.53382,70126.05313) -	0.946482 129978.8863 70362.20039 0.49	0.752e-8 129956.7699 70362.20039 0.46
5	4	4	3	5	5	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 45163.0359 32397.56032 0.139	100% 85781.64243 29810.6017 0.089	- (45163.0359,85781.64243) (32397.56032,29810.6017) -	0.891477 49329.01496 30091.34621 0.37	0.3028 49122.21496 30091.34621 0.28
6	3	4	4	10	5	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 100714.3984 68103.84701 0.476	100% 188194.8964 61237.82914 0.226	- (100714.3984,188194.8964) (68103.84701,61237.82914) -	0.87951 111236.5387 62065.11491 1.97	0.012275 110931.9501 62065.11491 1.06
7	4	4	4	5	7	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 44245.62777 36332.31283 0.243	100% 81930.82057 31636.29907 0.144	- (44245.62777,81930.82057) (36332.31283,31636.29907) -	0.848975 49771.98046 32345.51370 0.56	0.440e-8 49482.58046 32345.51370 0.42
8	3	4	3	7	7	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 80313.06242 46316.98758 0.204	100% 132513.6619 43209.00218 0.136	- (80313.06242,132513.6619) (46316.98758,43209.00218) -	0.879676 86247.52021 43582.96860 0.50	0.492e-8 85500.02021 43582.96860 0.37
9	5	3	4	10	7	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 126752.3668 69178.3225 0.469	100% 268444.2668 62669.21959 0.286	- (126752.3668,268444.2668) (69178.3225,62669.21959) -	0.847867 148307.98115 63639.65581 0.67	0.222e-8 148307.98115 63639.65581 0.62
10	5	5	5	10	7	λ Z_1 (TL) Z_2 (TL) CPU Süresi (sn.)	100% 95762.1041 68826.38428 1.282	100% 188172.7514 57369.55355 0.334	- (95762.1041,188172.7514) (68826.38428,57369.55355) -	0.8512 109512.85003 58972.59157 2.57	0.201e-8 109512.85003 58972.59157 1.45

Tablo-13: II. Tip Örnek Test Problemlerinin Hesaplama Sonuçları

Sıra Nu.	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>t</i>	Amaç Fonksiyonu/CPU Süresi	DOP-1 (Min Z_1)	DOP-2 (Min Z_2)	(AS/ÜS)	Başlangıç Çözümü	İyileştirilmiş Çözüm
11	7	5	5	10	10	λ	100%	100%	-	0.947501	0.378697
						Z_1 (TL)	86307.97115	251293.7295	(86307.97115,251293.7295)	94782.14143	94095.36994
						Z_2 (TL)	65859.90455	58663.94687	(65859.90455,58663.94687)	59041.72551	59029.63647
						CPU Süresi (sn.)	2.933	0.562	-	2.00	1.98
12	5	6	7	10	10	λ	100%	100%	-	0.915524	0.504e-8
						Z_1 (TL)	76152.6282	182389.5086	(76152.6282,182389.5086)	85023.28060	84987.54463
						Z_2 (TL)	64938.27954	58321.61375	(64938.27954,58321.61375)	58880.56398	58880.56398
						CPU Süresi (sn.)	2.617	0.843	-	2.95	2.56
13	6	7	7	20	10	λ	100%	100%	-	0.871119	0.000157
						Z_1 (TL)	175181.7413	422969.6741	(175181.7413,422969.6741)	207116.81980	207111.80221
						Z_2 (TL)	136740.2079	116231.5608	(136740.2079,116231.5608)	118874.72929	118874.31401
						CPU Süresi (sn.)	384.357	3.107	-	15.15	14.54
14	7	5	6	10	15	λ	100%	100%	-	0.881887	0.238283
						Z_1 (TL)	60167.42955	138286.6606	(60167.42955,138286.6606)	69394.30600	69394.30600
						Z_2 (TL)	71834.18882	57834.83648	(71834.18882,57834.83648)	59448.06820	59203.46520
						CPU Süresi (sn.)	4.136	1.343	-	7.47	6.91
15	7	5	7	15	15	λ	100%	100%	-	0.845587	0
						Z_1 (TL)	97351.19842	187656.7356	(97351.19842,187656.7356)	111209.5727	111209.5727
						Z_2 (TL)	97462.71392	85859.54491	(97462.71392,85859.54491)	87651.21976	87651.21976
						CPU Süresi (sn.)	86.794	2.741	-	12.31	12.03
16	5	6	7	15	15	λ	100%	100%	-	0.847507	0.185e-8
						Z_1 (TL)	118935.5033	198040.7259	(118935.5033,198040.7259)	130998.52105	130960.75907
						Z_2 (TL)	99356.30021	85276.9127	(99356.30021,85276.9127)	87423.92520	87423.92520
						CPU Süresi (sn.)	138.858	2.741	-	13.13	14.73
17	6	5	5	20	15	λ	100%	100%	-	0.841104	6.3e-05
						Z_1 (TL)	138148.4908	365079.9954	(138148.4908,365079.9954)	174207.0536	174204.7694
						Z_2 (TL)	137921.7007	120337.1413	(137921.7007,120337.1413)	123045.8485	123027.5644
						CPU Süresi (sn.)	167.485	2.593	-	33.43	16.29
18	6	7	5	20	15	λ	100%	100%	-	0.896392	0.001494
						Z_1 (TL)	204296.1785	468759.092	(204296.1785,468759.092)	231696.69712	231650.60182
						Z_2 (TL)	131416.0633	113990.0611	(131416.0633,113990.0611)	115795.53731	115792.84042
						CPU Süresi (sn.)	124.312	2.982	-	20.89	13.57
19	6	5	5	20	20	λ	100%	100%	-	0.868906	0.000221
						Z_1 (TL)	158085.3097	486679.5618	(158085.3097,486679.5618)	201161.92385	201161.92385
						Z_2 (TL)	130916.1046	109057.6369	(130916.1046,109057.6369)	111873.62317	111847.2611
						CPU Süresi (sn.)	1001.46	4.163	-	48.05	41.92
20	7	7	5	20	20	λ	100%	100%	-	0.9272343	0.256e-7
						Z_1 (TL)	154933.9299	492625.631	(154933.9299,492625.631)	179448.57991	179428.72053
						Z_2 (TL)	134645.3722	114300.5662	(134645.3722,114300.5662)	115780.96136	115780.96136
						CPU Süresi (sn.)	20.732	4.961	-	24.15	20.61

Tablo-14: III. Tip Örnek Test Problemlerinin Hesaplama Sonuçları

Sıra Nu.	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>t</i>	Amaç Fonksiyonu/CPU Süresi	DOP-1 (Min Z_1)	DOP-2 (Min Z_2)	(AS/ÜS)	Başlangıç Çözümü	İyileştirilmiş Çözüm
21	15	15	10	20	10	λ	100%	100%	-	0.961742	0.000105
						Z_1 (TL)	166139.232	457706.1961	(166139.232,457706.1961)	177294.1285	177292.9603
						Z_2 (TL)	138111.2881	115361.0372	(138111.2881,115361.0372)	116231.4263	116231.3351
						CPU Süresi (sn.)	292.407	72.494	-	787.52	906.69
22	10	20	10	20	10	λ	100%	100%	-	0.916535	5.6e-05
						Z_1 (TL)	147965.503	393234.1807	(147965.503,393234.1807)	168436.7932	168435.6401
						Z_2 (TL)	142523.9899	113555.5087	(142523.9899,113555.5087)	115973.3559	115973.2197
						CPU Süresi (sn.)	1000.038	10.807	-	635.34	752.61
33	15	20	20	20	10	λ	100%	100%	-	0.901804	0.000512
						Z_1 (TL)	124171.4868	362142.4921	(124171.4868,362142.4921)	147539.2564	147527.293
						Z_2 (TL)	134687.0823	111813.5772	(134687.0823,111813.5772)	114059.6609	114058.511
						CPU Süresi (sn.)	1001.489	27.174	-	1000.88	1000.9
24	15	10	10	30	10	λ	100%	100%	-	0.928158	4.5e-05
						Z_1 (TL)	209841.8162	585619.3122	(209841.8162,585619.3122)	236838.3423	236837.1245
						Z_2 (TL)	207301.2786	175900.5588	(207301.2786,175900.5588)	178156.4426	178156.3409
						CPU Süresi (sn.)	1000.039	1000.179	-	1000.59	616.5
25	10	15	10	30	10	λ	100%	100%	-	0.896063	0.000192
						Z_1 (TL)	219788.7982	608353.2645	(219788.7982,608353.2645)	260175.07770	260167.317
						Z_2 (TL)	207744.7132	171725.7502	(207744.7132,171725.7502)	175469.45822	175468.7388
						CPU Süresi (sn.)	1000.039	1000.039	-	72000.94	7922.33
26	10	10	15	30	10	λ	100%	100%	-	0.864036	0.000235
						Z_1 (TL)	217919.2613	596010.608	(217919.2613,596010.608)	269325.99389	256874.00762
						Z_2 (TL)	214085.1132	169627.0932	(214085.1132,169627.0932)	175666.47589	174207.60238
						CPU Süresi (sn.)	1003.295	1000.04	-	30323.85	5479.54
27	20	20	15	30	10	λ	100%	100%	-	0.951097	0.434956
						Z_1 (TL)	215146.6157	634555.7285	(215146.6157,634555.7285)	235656.816	226735.78650
						Z_2 (TL)	205023.0414	171728.7683	(205023.0414,171728.7683)	173356.9452	172648.76030
						CPU Süresi (sn.)	1001.423	84.85	-	1000.99	13883.17
28	10	10	10	40	10	λ	100%	100%	-	0.930952	3.3e-05
						Z_1 (TL)	286191.8095	614978.7613	(286191.8095,614978.7613)	308893.8564	308893.1184
						Z_2 (TL)	264850.3009	232381.7948	(264850.3009,232381.7948)	234623.6768	234623.6039
						CPU Süresi (sn.)	1000.039	1000.055	-	30518.41	5758.09
29	20	15	10	40	10	λ	100%	100%	-	0.675599	0.970853
						Z_1 (TL)	282248.5938	764035.0004	(282248.5938,764035.0004)	758386.47078	296126.7316
						Z_2 (TL)	270150.4023	229271.3266	(270150.4023,229271.3266)	814234.38050	246321.4255
						CPU Süresi (sn.)	1000.397	1000.085	-	23.57	1000.8
30	15	15	15	40	10	λ	100%	100%	-	0.939778	0.0022
						Z_1 (TL)	312819.8969	745585.1177	(312819.8969,745585.1177)	338882.0452	338824.7038
						Z_2 (TL)	267255.7642	226369.574	(267255.7642,226369.574)	228831.8374	228826.42
						CPU Süresi (sn.)	1004.124	1000.07	-	14672.25	50013.41

Tablo-15: IV. Tip Örnek Test Problemlerinin Hesaplama Sonuçları

Sıra Nu.	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>t</i>	Amaç Fonksiyonu/CPU Süresi	DOP-1 (Min Z_1)	DOP-2 (Min Z_2)	(AS/ÜS)	Başlangıç Çözümü	İyileştirilmiş Çözüm
31	20	10	15	30	15	λ	100%	100%	-	0.913616	1.5e-05
						Z_1 (TL)	193158.2209	607533.9895	(193158.2209,607533.9895)	228953.61395	228953.063
						Z_2 (TL)	212905.4066	168141.1593	(212905.4066,168141.1593)	172008.06936	172008.0099
						CPU Süresi (sn.)	1000,117	48,484	-	31904,27	4033,84
32	20	20	20	30	15	λ	100%	100%	-	0.957743	0.343781
						Z_1 (TL)	208933.8891	645850.3704	(208933.8891,645850.3704)	227396.72655	227390.37433
						Z_2 (TL)	217742.5369	167688.1908	(217742.5369,167688.1908)	169803.34391	169802.61619
						CPU Süresi (sn.)	1000.133	1000.117	-	48418.53	4483.74
33	20	20	10	40	15	λ	100%	100%	-	0.941239	0.000701
						Z_1 (TL)	244325.7121	766863.6794	(244325.7121,766863.6794)	275030.5384	275009.02727
						Z_2 (TL)	267730.8422	229969.0748	(267730.8422,229969.0748)	232187.9921	232186.43758
						CPU Süresi (sn.)	1000.117	172.175	-	1001.41	14740.00
34	20	10	15	30	20	λ	100%	100%	-	0.922944	0.000507
						Z_1 (TL)	217018.6908	623580.639	(217018.6908,623580.639)	248346.8698	248331.0008
						Z_2 (TL)	204898.0187	165638.6065	(204898.0187,165638.6065)	168663.7934	168662.261
						CPU Süresi (sn.)	1000,134	70,292	-	14699,22	2107,25
35	20	15	20	30	20	λ	100%	100%	-	0.935937	0.146e-8
						Z_1 (TL)	157187.65	465598.2956	(157187.65,465598.2956)	176945.26663	176945.2667
						Z_2 (TL)	202216.2718	163399.8156	(202216.2718,163399.8156)	165886.50233	165886.5023
						CPU Süresi (sn.)	1000,209	123,924	-	23253,06	1002,11
36	15	10	10	40	20	λ	100%	100%	-	-	-
						Z_1 (TL)	267645.0953	770714.8242	(267645.0953,770714.8242)	0.00000	0.00000
						Z_2 (TL)	261695.732	221004.6177	(261695.732,221004.6177)	0.00000	0.00000
						CPU Süresi (sn.)	1001,634	73,26	-	13,46	116,63
37	15	10	20	40	20	λ	100%	100%	-	-	-
						Z_1 (TL)	259701.7938	824116.5424	(259701.7938,824116.5424)	0.00000	0.00000
						Z_2 (TL)	270181.1764	217181.9171	(270181.1764,217181.9171)	0.00000	0.00000
						CPU Süresi (sn.)	1002,27	236,105	-	118,64	387,88
38	20	10	20	40	20	λ	100%	100%	-	-	-
						Z_1 (TL)	274412.3851	851947.8866	(274412.3851,851947.8866)	0.00000	0.00000
						Z_2 (TL)	272273.9587	220525.091	(272273.9587,220525.091)	0.00000	0.00000
						CPU Süresi (sn.)	1000,756	226,9	-	39,25	400,39
39	20	15	20	40	20	λ	100%	100%	-	-	-
						Z_1 (TL)	284631.0016	833023.2916	(284631.0016,833023.2916)	0.00000	0.00000
						Z_2 (TL)	269085.3714	218459.4992	(269085.3714,218459.4992)	0.00000	0.00000
						CPU Süresi (sn.)	1000,244	298,705	-	44,90	490,71
40	20	20	20	40	20	λ	100%	100%	-	-	-
						Z_1 (TL)	234732.6981	803868.7426	(234732.6981,803868.7426)	0.00000	0.00000
						Z_2 (TL)	268456.9688	219714.3439	(268456.9688,219714.3439)	0.00000	0.00000
						CPU Süresi (sn.)	1000,21	342,445	-	19971,96	523,13

Test problemlerinin çözümlerinden de anlaşılacağı üzere, problem boyutunun büyüklüğü arttıkça problemi çözmek için gerekli zaman ve toplam sistem maliyeti artmaktadır. Ayrıca, amaç fonksiyonlarının alt ve üst sınır değerleri arasındaki aralığın artış gösterdiği ve GAMS/DICOPT çözücüsü ile elde edilen yaklaşık çözümlerin, küçük boyutlu problemlere göre alt sınır değerlerinden daha fazla sapmaya maruz kaldığı (çözümün kötüleştiği) görülmektedir. Tablo-15'te görüldüğü gibi, 36, 37, 38, 39 ve 40 numaralı test problemlerinde GAMS/DICOPT optimizasyon programı, amaç fonksiyonlarının alt ve üst sınır değerlerini elde etmesine rağmen herhangi bir çözüme ulaşamamaktadır. Bütünleşik tedarik zinciri ağında tesis yeri seçimi problemlerinin karmaşıklığı NP-Zor problemler sınıfına girer ve lojistik tesislerin sayısı, yerleri ve bunlara müşteri taleplerinin atanmasının eşzamanlı belirlenmesi genel bir bakış açısını gerektirir. Tesis yeri seçimi problemlerini optimal şekilde çözmek için geliştirilen algoritmalar kombinatoriyal olarak patlamaktadır ve bu tür problemleri pratik uygulamalarda istendiği gibi defalarca çözmek için gereken kaynaklar kısıtlayıcıdır. NP-Zor, optimizasyon problemlerini sınıflandırmak için kullanılan bir terimdir. Bugüne kadar bu tip problemler için polinom zamanlı bir algoritma geliştirilememiştir. NP-Zor problemlerin optimal çözümünü bulmak, üssel zamanlı (dolayısı ile çok uzun sürebilecek) bir algoritma yoluyla mümkündür (Garey ve Johnson, 1979). Bu nedenle problemin çözümünde genellikle sezgisel algoritmalar kullanılmaktadır.

Ayrıca, bu çalışmada oluşturulan matematiksel modelin doğrusal olmayan yapısı problemin karmaşıklığını daha da artırmakta ve çözümü zorlaştırmaktadır. Çalışmada yararlanılan GAMS/DICOPT çözücüsü, doğrusal olmayan programlama modellerinin çok küçük problemlerde dahi optimal çözümlerini garanti edememektedir. GAMS/DICOPT çözücüsünden elde edilen çözümler, yukarıda da belirtildiği gibi yaklaşık çözümlerdir. Bu nedenle, önerilen bulanık çok amaçlı doğrusal olmayan programlama modeli için, GAMS/DICOPT çözücüsünün elde ettiği yaklaşık çözümlerden daha iyi çözümler elde edebilmek ve çözüm elde edilemeyen büyük boyutlu problemleri kabul edilebilir sürede çözebilmek amacıyla sezgisel yöntem geliştirilmelidir.

En iyi bilinen genel sezgisel metotlar; Tabu Arama, Tavlama Benzetimi (TB) ve Genetik Algoritmalar. Son yıllarda bu sezgisellerin popülaritesi artmıştır

ve literatürde benzerlerinden üstün performans gösteren birçok yayınlanmış çalışma bulunmaktadır. Bu çalışmada, Bütünleşik Tedarik Zinciri Ağında Tesis Yeri Seçimi Problemleri üzerinde TB tabanlı sezgisel algoritma uygulaması yapılmış ve bu algoritmanın performansı değişik parametreler ile karşılaştırılarak karar vericiye en uygun alternatifler sunulmaya çalışılmıştır.

4. ÖNERİLEN SEZGİSEL YÖNTEM

a. Tavlama Benzetimi Algoritması

TB, kombinatoriyal eniyileme problemleri için iyi çözümler veren stokastik arama yöntemidir. Bu yaklaşım, metallerin fiziksel olarak tavlama işleminden esinlenilerek kombinatoriyal problemlerin çözümleri için geliştirilmiştir. TB Algoritması, yapısal bozukluklar içeren metal kristallerinin yüksek bir sıcaklıktan başlayarak kademeli olarak soğutulması ve böylece yapısal bozukluklarından arındırılarak mükemmelleştirilmesi (minimum enerjili kristal yapısına dönüşmesi) mantığına dayanır. Bu yaklaşım Metropolis ve diğ. (1953) çalışmasını temel almaktadır. Söz konusu çalışma ilk halinde belirli bir ısı seviyesinde atomların dengeli dağılımlarını bulma amacıyla geliştirilmiş ve enerji değişimlerini taklit etmiştir. Yaklaşım ile matematiksel minimizasyon arasındaki bağlantı Pincus (1970) tarafından kurulmuştur. Ancak, optimizasyon problemlerinin çözümünde kullanılabilmesi fikri Kirkpatrick ve diğ. (1983) tarafından ortaya atılmıştır.

Bu yöntem ile TB Algoritması çözüm kümesini dolaşmaya önceden belirlenmiş sıcaklıkla ve rassal olarak seçtiği bir çözümle başlar. Daha sonra komşuluk fonksiyonu yardımıyla bu çözüme komşu bir çözüm üretir. Üretilen komşu çözüm ile mevcut çözümün amaç fonksiyonu değerleri arasındaki farkı, ortalaması o anki sıcaklık olan üssel dağılımdan seçilen rassal bir sayı ile karşılaştırır. Amaç fonksiyonu değerleri arasındaki fark bu rassal sayıdan daha küçük ise komşu çözümü kabul ederek bu çözüme ilerler, aksi halde mevcut çözümü terk etmeksizin farklı bir komşu çözüm üretir. Her seferinde farklı bir rassal sayı ile aynı karşılaştırmayı yapar. Belli sayıda iterasyon boyunca aynı sıcaklık korunur, iterasyon sayısına ulaşıncaya kadar sıcaklık kademeli olarak azaltılır. Sıcaklık, önceden belirlenmiş bir

seviyenin altına indiğinde ise algoritma arama esnasında bulduğu en iyi çözümü rapor ederek sonlanır.

TB’nde kötü çözümü seçme olasılığı sistemli bir şekilde sıcaklıkla azaltılır. TB, sıcaklık yüksek iken (ilk iterasyonlarda) hemen hemen tüm komşu çözümleri kabul ettiği için Monte Carlo Aramaya, sıcaklık düşük iken (son iterasyonlarda) hemen hemen sadece mevcut çözümden daha iyi komşu çözümleri kabul ettiği için Yerel Aramaya benzer. Ara iterasyonlarda ise kısmen Eşik Kabulünü andırır. TB Algoritmasının temel amacı, çözüm uzayında aranmadık bölge bırakmamaktır. TB’nin kombinatorial optimizasyon problemleri için optimuma yakın çözümler veren kullanışlı bir yöntem olarak kullanıldığı söylenebilir. TB, gezgin satıcı problemi, çizelgeleme, karesel atama problemi, şebeke tasarımı gibi birçok kombinatorial eniyileme probleminin çözümünde kullanılmıştır.

Fiziksel tavlama süreci ile kombinatorial optimizasyon problemleri arasındaki benzerlik şöyle açıklanabilir:

Tablo-15: Tavlama Sürecinin Optimizasyon Problemlerinde Karşılığı (Reeves, 1993)

Tavlama Süreci	Kombinatorial Eniyileme
Sıcaklık	Kontrol parametresi
Enerji	Amaç fonksiyon değeri (maliyet)
Durum değişikliği	Komşu çözüm
Minimum enerji (kristalleşme)	Sezgisel çözüm
Sistemin durumları	Uygun çözüm

Herhangi bir problemin çözümünde TB Algoritmasının kullanılması için bazı parametrelerin belirlenmesi gerekir. Bu parametreler: başlangıç sıcaklığı (T_0), her sıcaklıktaki iterasyon sayısı, soğutma fonksiyonu, algoritmayı durdurma kriteri. Başlangıç sıcaklığı bir girdi parametresidir. Sıcaklık, kötü çözümlerin kabul edilme olasılığını kontrol etmek için kullanılır. İterasyon sayısı, her sıcaklıkta üretilen çözümlerin sayısıdır. Soğutma fonksiyonu, bir önceki iterasyon sıcaklığına bağlı olarak mevcut iterasyondaki sıcaklığı belirler. Başlangıç sıcaklığı ile birlikte iterasyon sayısı ve soğutma fonksiyonu, soğutma çizelgesi olarak adlandırılır. Bu

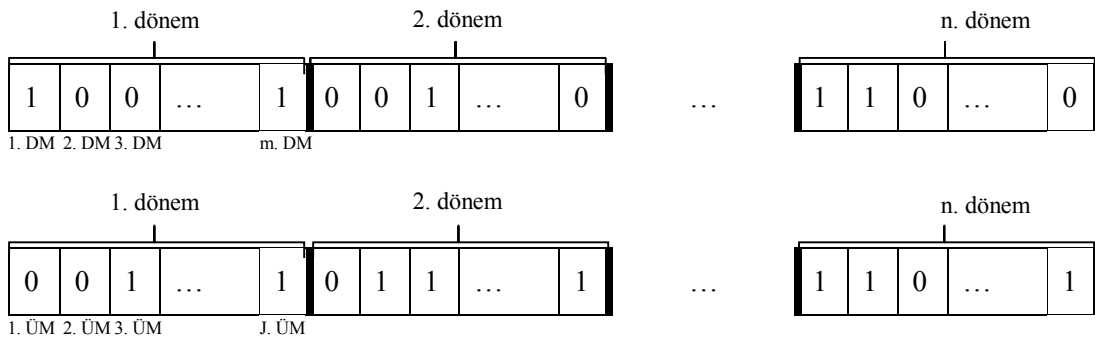
çizelge, çözüm kalitesinde veya yakınsama oranında büyük etkiye sahiptir. Her sıcaklık değişiminde elde edilen çözüm, çok sayıda ardışık sıcaklık değişimlerinde değişmiyor ise TB Algoritması durdurulur (Güner ve Altıparmak, 2003).

Bu tez çalışmasında TB Algoritmasının seçilmesinin nedeni şöyle açıklanabilir: Yukarıda adı geçen diğer sezgisel algoritmalar global en iyi yerine genellikle yerel en iyi noktaya yönelme eğilimindedir. Kontrollü bir yaklaşımla daha kötüye gidişlere izin vermek veya “yokuş yukarı hareket” etmek TB Algoritmasının bu soruna getirdiği bir çözümdür. Yani TB Algoritması, global en iyi değerden daha kötü bir değere geçişe kontrollü bir şekilde izin vererek yerel en iyiden kurtulmayı sağlamaktadır. Böylece yüksek sıcaklıklarda (çözümleri kabul etme olasılığı yüksek) diğer algoritmalarla göre daha geniş bir alanı taramaya imkan vermektedir. Literatürde bütünleşik tedarik zinciri ağında üretim/dağıtım planlamasında TB Algoritmasının kullanıldığı çok sayıda çalışma bulunmaktadır (bkz. Altıparmak ve diğ., 2006).

b. Önerilen Sezgisel Yöntemin Adımları

(1) Başlangıç Çözümünün ve Parametrelerin Belirlenmesi

Önerilen TB Algoritması tabanlı sezgisel için ele alınan probleme özgü örnek çözüm gösterimi aşağıdaki şekilde verilmiştir.



Şekil-20: TB Sezgiseli için Probleme Özgü Örnek Çözüm Gösterimi

Çözüm vektörü problemin ana kararlarından birisi olan dağıtım merkezi ve üretim merkezi açma–kapama kararını göstermektedir. Problemin çok dönemli yapısı

itibariyle çözüm vektörü dağıtım merkezleri için $(m+n)$, üretim merkezleri için ise $(j+n)$ uzunluğundadır.

Çözüm vektöründe her bir hücredeki “0” ve “1” değerleri ilgili dönemde ilgili dağıtım/üretim merkezinin sırasıyla “kapalı” veya “açık” olduğunu belirtmektedir.

Başlangıçta, çözümler rassal olarak belirlenmiştir. Açma ve kapama kararları her bir dönemdeki dağıtım/üretim merkezi için eşit olasılıkla verilmektedir.

Problem için önerilen bütünleşik tedarik zinciri ağındaki her bir müşteri için sırasıyla, rassal olarak bir dönem seçilmektedir. Seçilen dönemdeki açık olan dağıtım merkezlerinden birisi yine rassal olarak (talep–kapasite kısıtını sağlayacak şekilde) seçilip müşteri bu dağıtım merkezine atanmaktadır. Yine aynı dönem içerisindeki açık olan üretim merkezlerinden birisi rassal olarak seçilip ilgili dağıtım merkezi bu üretim merkezine atanmaktadır (DM–ÜM kapasite kısıtını sağlayacak şekilde). Bu atamalar yapıldıktan sonra ilgili dönemdeki kapasiteler güncellenmektedir.

Burada tüm hareketlerin rassal olarak seçilmesi çözüm uzayında olabildiğince fazla sayıda çözüm aranmasını sağlamak içindir. Aynı şekilde ilgili dönemdeki tedarikçilerden birine üretim merkezi atanmalıdır. Bu işlemler her bir müşterinin herhangi bir dönemdeki talepleri karşılanıncaya kadar devam ettirilir.

Tüm atamalar yapıldıktan sonra, problem için önerilen matematiksel modeldeki tam sayılı X , Y ve Z karar değişkenleri belirlenir. Bu değişkenlerin belirlenmesinden sonra yine modeldeki amaç fonksiyon değerleri hesaplanır. Bu amaç fonksiyonu değerlerine göre problemin bulanık yapısından kaynaklı üyelik fonksiyonu derecesi de hesaplanır.

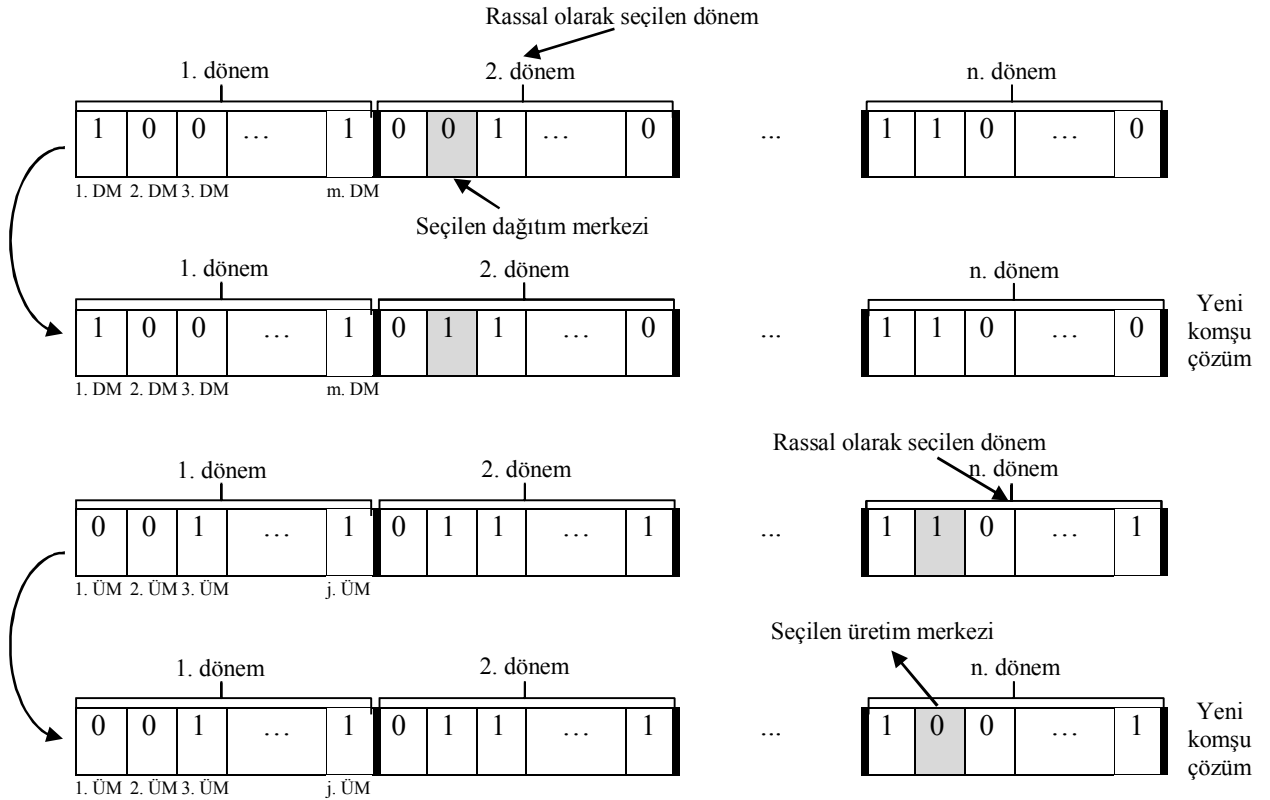
Algoritmanın Her İterasyonunda Yapılan İşlemler

Algoritmanın her iterasyonunda ise, iterasyon sayısı (n) ve sıcaklık değerleri güncellenir $[n=0, T=T_i]$.

(2) Komşu Çözüm Üretme

Elde edilen çözümün komşusunu üretmek için, literatürde 0–1 (binary) kodlama şeklinde verilen çözümler için komşuluk üretme mekanizmalarından olan “ADD” / “DROP” hareket mekanizmaları kullanılmıştır. “ADD” hareketinde çözüm vektörünün seçilen “0” değerli bir elemanı “1” değeri alırken, “DROP” hareketinde çözüm vektörünün seçilen “1” değerli elemanı “0” değeri alır.

Ele alınan problemde ise bir komşu çözüm üretmek için rassal olarak seçilen bir dönemdeki seçilen üretim/dağıtım merkezleri açık (1) ise DROP hareketiyle kapalı (0) hale getirilmekte, kapalı (0) ise ADD hareketiyle açık (1) hale getirilmektedir. Böylece oluşturulan yeni komşu çözüm için yukarıda belirtildiği gibi yeniden her müşteri için atamalar belirlenmekte, problemin X, Y ve Z karar değişkenleri ile amaç fonksiyon değerleri hesaplanmaktadır. Aşağıda şekilsel komşuluk üretimi gösterilmiştir.



Şekil-21: Komşu Çözümlerin Üretilmesi

Başlangıç parametreleri belirlendikten sonra her bir sıcaklık değerinde belirlenen iterasyon sayısı kadar komşu çözümler üretilir ve amaç fonksiyon değerleri hesaplanır.

(3) Yeni Çözümlerin Kabulü

TB Algoritmasının, Yerel Arama Algoritmalarından farkı, oluşacak kötü çözümleri de belli bir olasılık değerine (kabul olasılığı) göre kabul etmesidir. Böylelikle Yerel Arama Algoritmalarının yerel optimumlara takılması dezavantajından kurtularak çözüm uzayının farklı noktalarının aranması mümkün olmaktadır.

Eğer her bir iterasyondaki komşu çözümün amaç fonksiyonu değeri eldeki mevcut çözümün amaç fonksiyonu değerinden daha iyiye bu komşu çözüm direkt olarak mevcut çözümle yer değiştirir ve ilerleyen iterasyonlarda yeni komşu çözümler bu çözümden üretilir. Diğer türlü kötü bir amaç fonksiyonu değerine sahipse kabul olasılığına bakılır.

Çalışmada ele alınan problemin iki farklı minimizasyon amaç fonksiyonu olmasından dolayı bir çözüme karşılık gelen iki amaç fonksiyonu değeri bulunmaktadır. Belirtildiği gibi eğer yeni komşu çözümün her iki amaç fonksiyonu değeri eldeki mevcut çözümün amaç fonksiyonu değerlerinden daha iyiye (daha küçükse) bu komşu çözüm yeni çözüm olarak kabul edilmektedir. Ancak en azından bir çözüm dahi kötü ise kabul olasılığına bakılmaktadır.

Kabul olasılığında iki parametre söz konusudur. İlki algoritmanın mevcut sıcaklık (T) değeri, ikinci ise mevcut ve komşu çözümlerin amaç fonksiyonları arasındaki fark (Δ)'tır. Kabul olasılığı aşağıdaki şekilde hesaplanır.

$$K.O. = e^{\left(\frac{-\Delta}{T}\right)}$$

Her iki amaç fonksiyonu değerlerindeki farka ve o anki sıcaklık değerine göre iki ayrı kabul olasılığı değeri hesaplanır. (0,1) arasında düzgün dağılıma göre

bir olasılık değeri (p) üretilir. Eğer bu olasılık değeri iki kabul olasılığı değerinin bir tanesinden daha küçükse ($p \leq K.O._1$ veya $p \leq K.O._2$) üretilen yeni komşu çözüm kabul edilerek iterasyonlar devam ettirilir.

Eğer belirlenen maksimum iterasyon sayısına ulaşırsa soğutma çizelgesi uygulanarak mevcut sıcaklık düşürülür.

(4) Soğutma Çizelgesinin Çalıştırılması

Soğutma çizelgesi, başlangıçta çözüm uzayında aramayı çok iyi gerçekleştirirken (kötü çözümlerin kabul edilme olasılığı yüksek), algoritmanın sonlarına doğru mevcut çözümün etrafında derinlemesine aramayı çok iyi gerçekleştirecek şekilde belirlenmelidir.

Soğutma oranı, algoritmanın performansında kullanılan soğutma tipine (geometrik, logaritmik, vb.) göre çok daha önemlidir. Önerilen algorithmada literatürde “Geometrik Soğutma Çizelgesi” olarak verilen yaklaşım kullanılmıştır. Bu çizelgede önceden belirlenmiş bir “soğutma oranı” (α) na göre mevcut sıcaklık düşürülür. “ i ” mevcut sıcaklık seviyesini göstermek üzere;

$$T_{i+1} = \alpha \cdot T_i$$

şeklinde bir sonraki sıcaklık değeri hesaplanmaktadır. Sıcaklığın yavaş soğutulması önerilir. Bu nedenle, α değeri genellikle 0.80–0.99 arasında seçilir.

(5) Durdurma Kriteri

Önerilen Algoritma için durdurma kriteri olarak önceden belirlenmiş bir minimum sıcaklık değeri dikkate alınmıştır. Mevcut sıcaklık değeri soğutma çizelgesi ile bu minimum değere ulaştığında algoritma sonlanmaktadır. Önerilen sezgisel algoritmanın adımları prosedür olarak aşağıda verilmiştir.

Başlangıç çözümü (s) rassal olarak belirle ve amaç fonksiyonları $f(s_1)$ ve $f(s_2)$ hesapla Başlangıç sıcaklığı (T_0) belirle Soğutma oranı (α) ve her sıcaklık değerinde komşuluk aranacak iterasyon sayısını (N) belirle Başlangıç çözümünü ve amaç fonksiyon değerlerini en iyi çözüm olarak işaretle $s \rightarrow s^{iyi}, f(s_1) \rightarrow f(s_1^{iyi}), f(s_2) \rightarrow f(s_2^{iyi})$ İlk iterasyondan başla (n=0) s 'in komşuluk üretme hareketlerini kullanarak (ADD / DROP) bir komşusunu üret (s') ve amaç fonksiyonu değerlerini hesapla $f(s'_1)$ ve $f(s'_2)$ $\Delta_1 = f(s'_1) - f(s_1) \quad \Delta_2 = f(s'_2) - f(s_2)$ $\Delta_1 \leq 0$ ve $\Delta_2 \leq 0$ ise yeni komşuluk çözümünü mevcut çözüm olarak al $s' \rightarrow s$ Değilse $[0,1]$ aralığında u rassal sayısı üret $u \leq e^{-\Delta_1/T}$ veya $u \leq e^{-\Delta_2/T}$ ise yine yeni komşuluk çözümünü mevcut çözüm olarak al $s' \rightarrow s$ Elde edilen çözüm o zamana kadarki en iyi çözümünden iyiyse en iyi çözümü güncelle $f(s') \leq f(s_{iyi})$ ise $s' \rightarrow s_{iyi}$ İterasyon sayısını artır (n=n+1) Eğer n=N ise sıcaklığı azalt: $T_{i+1} = \alpha.T_i$ Son sıcaklık değerine kadar tekrar et. ($T_{i+1} \leq T_{son}$) s_{iyi} çözümünü problemin çözümü olarak al.

Şekil-22: TB Algoritması Tabanlı Sezgiselin Adımları

Yukarıda TB Sezgiseline dayalı önerilen algoritma Borland C++ Builder programlama dili ile kodlanmıştır. Bilgisayar koşumları 1.80 Ghz işlemci hızına ve 2.00 GB RAM belleğe sahip bir bilgisayarda gerçekleştirilmiştir. TB Algoritması tabanlı sezgiselin C++ kodları EK-Ç'de sunulmuştur.

c. Deneysel Çalışma

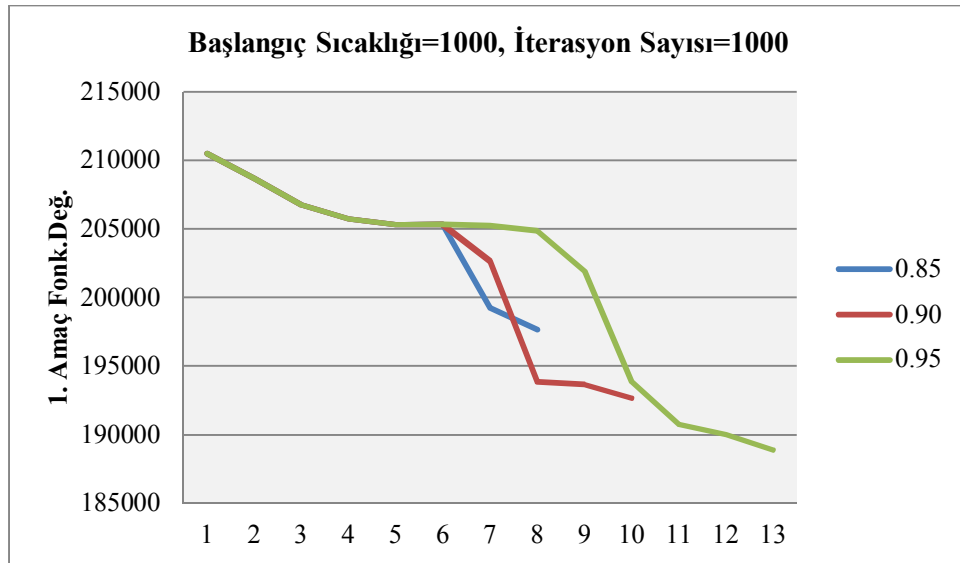
(1) Parametrelerin Belirlenmesi

TB Algoritmasının performansını etkileyen üç parametre mevcuttur: Başlangıç sıcaklığı, her sıcaklıktaki iterasyon sayısı ve soğutma oranı. Bu parametrelerin her biri için üç farklı değer dikkate alınarak belirlenen test

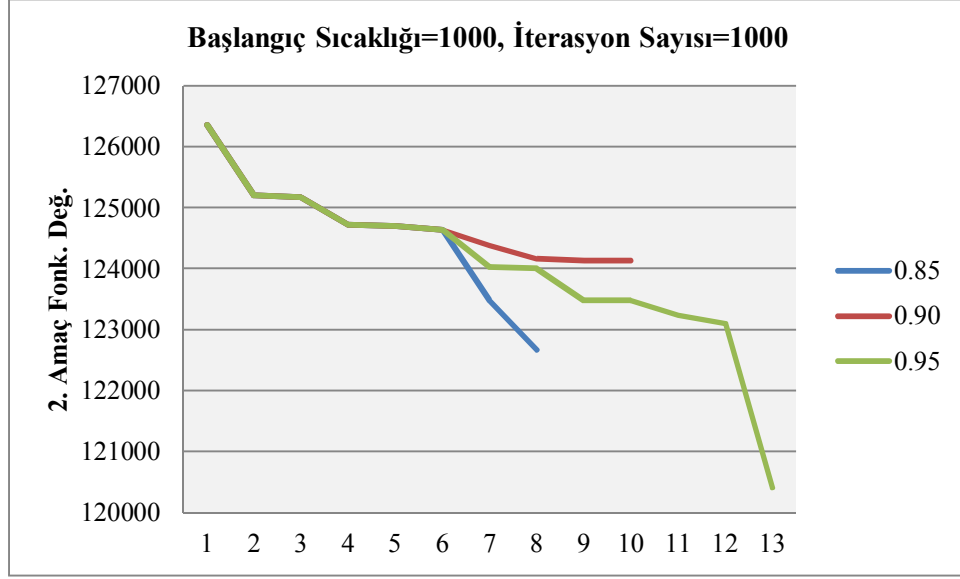
problemlerinde alıřtırılmıř ve her bir parametrenin performansı analiz edilmiřtir. Parametre analizinin yapılması iin farklı boyutlu iki adet (III. Tip, 21 numaralı ve IV. Tip, 32 numaralı) test problemi seilmiřtir.

İyi bir sezgisel yöntem, bařlangı özümüne baėlı olmamalıdır. Bařlangı özümü; bařlangıta kötü özümleri kabul edecek ve elde edilen son özümün bařlangı özümünden baėımsız olmasını saėlayacak kadar yüksek sıcaklıkta olmalıdır. Bařlangı özümüne baėımlılıėı azaltmak ve durdurma kriterinin özüm üzerindeki etkisini en aza indirmek iin bařlangı sıcaklıėı {1000, 5000, 10000} olarak alınmıřtır. Her sıcaklıktaki iterasyon sayısı deėeri birden fazla problemin denenmesi sonucu hem özüm zamanını fazla uzatmaması hem de özüm uzayını küçük deėerlere nazaran daha geniř bir řekilde taramasından dolayı {100, 500, 1000} olarak alınmıřtır. Son olarak soėutma oranı deėerleri de literatürde verilen en düşük deėer olan 0,80 deėerinden yüksek {0.85, 0.90, 0.95} deėerleri seilmiřtir.

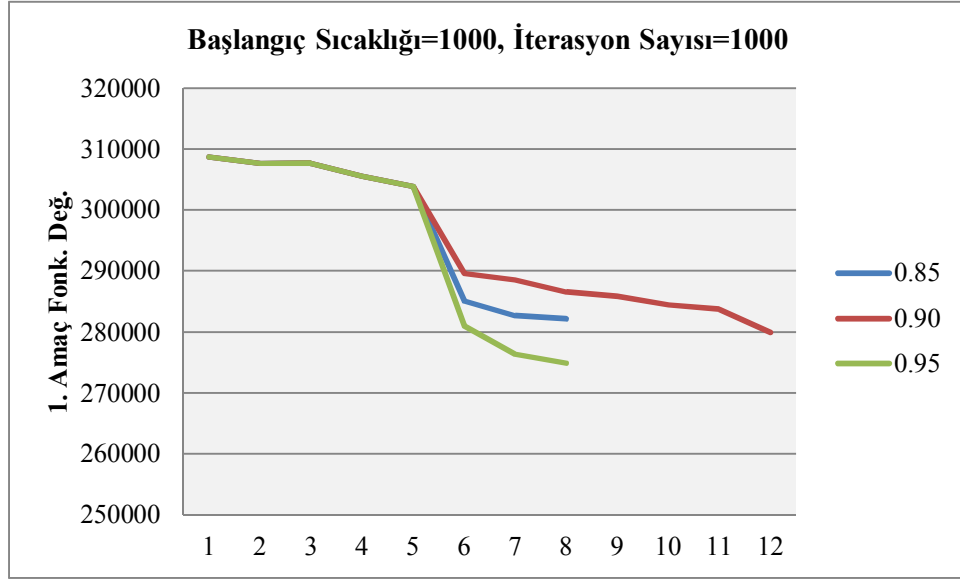
Ele alınan problemlerde öncelikle iterasyon sayısı ve bařlangı sıcaklıėı deėerleri sabit tutularak soėutma oranının performansı analiz edilmiřtir. Her iki test probleminde de bařlangı sıcaklıėının 1000 ve iterasyon sayısının 1000 olduėu durumlarda ama fonksiyon deėerlerinin en düşük olduėu görölmüřtür. Analiz sonucu elde edilen grafikler ařaėıda verilmiřtir.



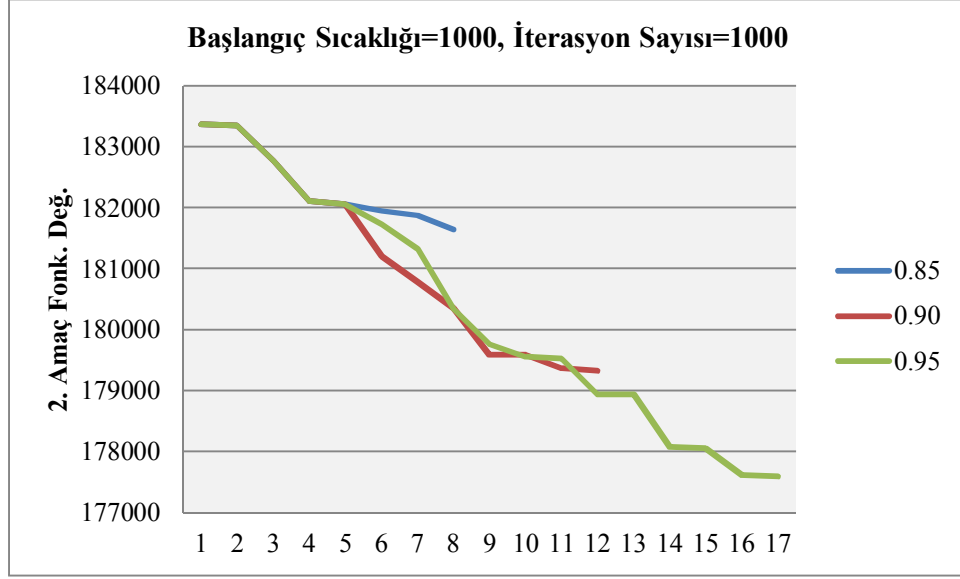
řekil-23: 1. Ama Fonksiyonu iin Soėutma Oranının özüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10)



Şekil-24: 2. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10)

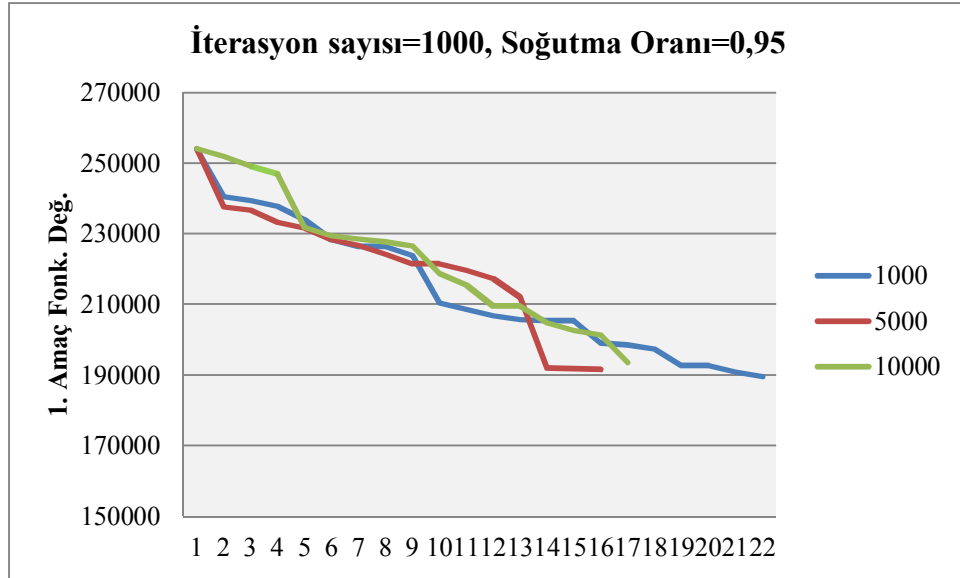


Şekil-25: 1. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15)

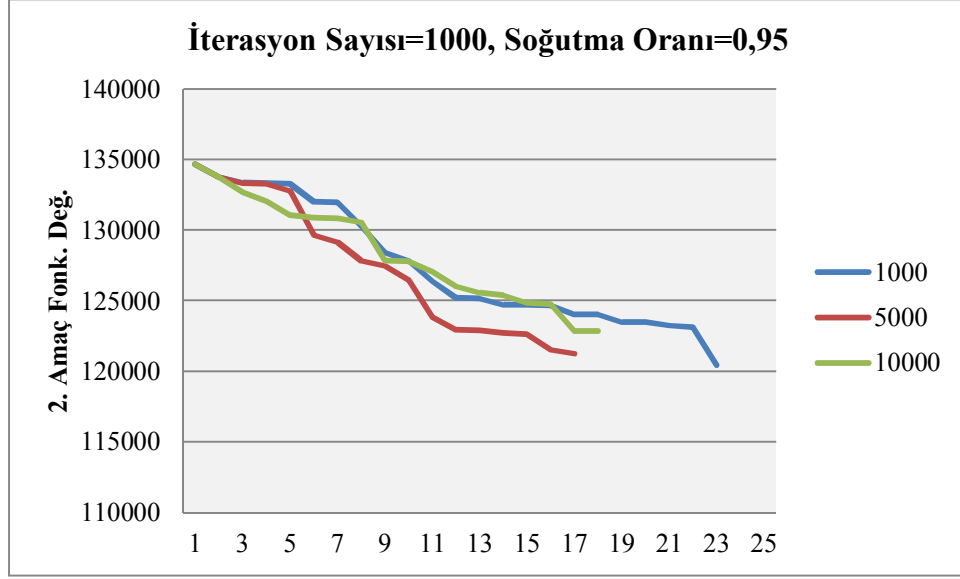


Şekil-26: 2. Amaç Fonksiyonu için Soğutma Oranının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15)

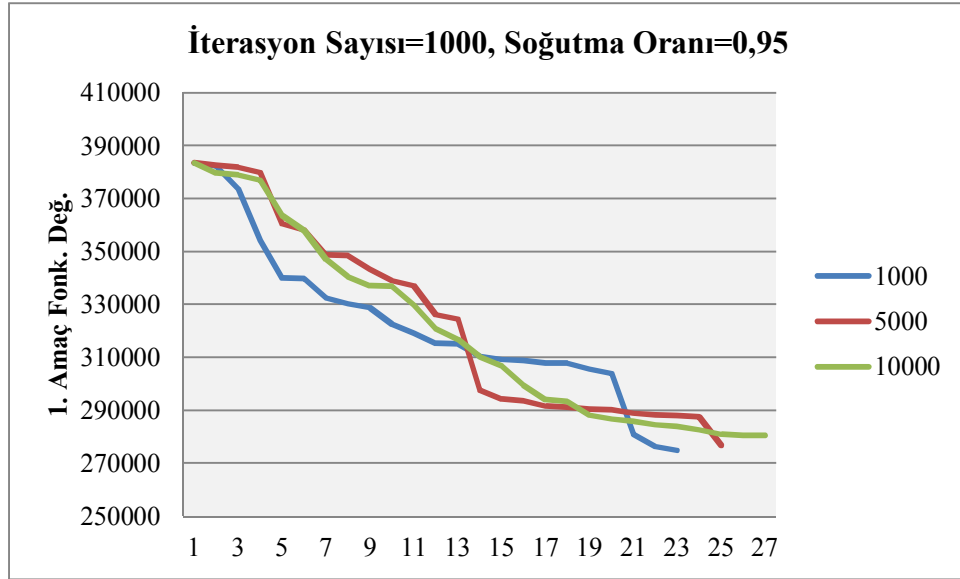
Yukarıdaki çalışma sonucu elde edilen 0,95 değeri ve iterasyon sayısı 1000 sabit tutulmuş, başlangıç sıcaklığının, TB Algoritmasının performansı üzerindeki etkisi incelenmiş ve her iki test problemi için aşağıdaki grafikler elde edilmiştir.



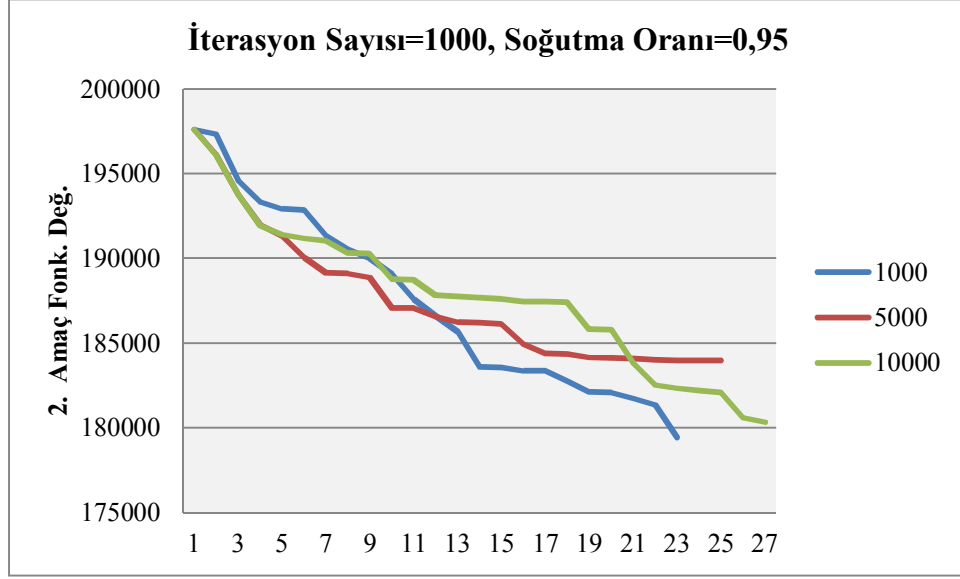
Şekil-27: 1. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10)



Şekil-28: 2. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (15–15–10–20–10)



Şekil-29: 1. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15)



Şekil-30: 2. Amaç Fonksiyonu için Başlangıç Sıcaklığının Çözüm Kalitesi Üzerindeki Etkisi (20–20–20–30–15)

Yapılan deneysel çalışma sonucunda önerilen TB Algoritması tabanlı sezgisel yaklaşım için aşağıdaki parametrelerin en iyi amaç fonksiyon değerlerini verdiği belirlenmiştir:

- Başlangıç sıcaklığı: 1000
- Her sıcaklık değerindeki iterasyon sayısı: 1000
- Soğutma oranı: 0,95

(2) Sayısal Sonuçlar

Önerilen TB Algoritması tabanlı sezgisel yaklaşımın parametrelerinin belirlenmesinden sonra, üretilen test problemlerinden seçilen 27 adedi bu algoritma ile koşturulmuştur. Her bir problem 10 kez çalıştırılmış ve elde edilen amaç fonksiyon değerleri ile çözüm sürelerinin ortalaması alınmıştır. Aşağıdaki sonuç tablosunda GAMS/DICOPT optimizasyon programı ve TB Algoritması tabanlı sezgisel ile çalıştırılan örnek test problemlerinin karşılaştırmalı performans analizi gösterilmiştir. Amaç fonksiyonu açısından performansı matematiksel modelden elde edilen alt sınır değerinden % sapma olarak ölçülmüştür. GAMS/DICOPT optimizasyon programı ve TB Algoritması tabanlı sezgisel yaklaşım ile elde edilen çözümler, CPU süreleri ve bulunan en iyi değerden (yaklaşık değer) sapma oranları Tablo-16’da verilmiştir.

Tablo-16: Sayısal Sonuçlar

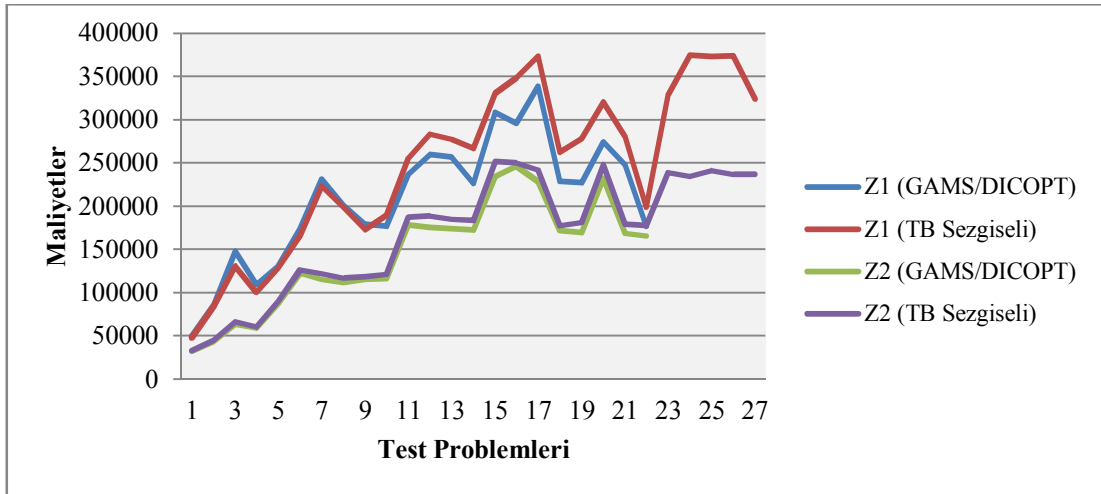
Tipi	Test Prob.	Problem Boyutu						GAMS/DICOPT			TB Alg. Tabanlı Yaklaşım				1. Amaç Fonk. Sapma	2. Amaç Fonk. Sapma
		i	j	k	l	t	λ	Toplam Maliyet		CPU Süresi (sn.)	λ	Toplam Maliyet**		CPU Süresi (sn.)**		
								Z ₁ (a)	Z ₂ (aa)			Z ₁ (b)	Z ₂ (bb)			
I	7	4	4	4	5	7	0,440e-8	49482,58	32345,51	1,37	0,652511	48365,54	33268,11	24,43	-2,26	2,85
	8	3	4	3	7	7	0,492e-8	85500,02	43582,97	1,21	0,353087	84042,40	45219,60	22,39	-1,70	3,76
	9	5	3	4	10	7	0,222e-8	148307,98	63639,66	2,05	0,432612	130989,96	66362,41	33,85	-11,68	4,28
	10	5	5	5	10	7	0,201e-8	109512,85	58972,59	5,64	0,723922	100454,65	60532,54	42,96	-8,27	2,65
II	16	5	6	7	15	15	0,185e-8	130960,76	87423,93	169,46	0,657836	128722,50	90094,38	87,61	-1,71	3,05
	17	6	5	5	20	15	0,000063	174204,77	123027,56	219,80	0,622204	166096,45	126980,52	81,26	-4,65	3,21
	18	6	7	5	20	15	0,001494	231650,60	115792,84	161,75	0,531372	223440,66	122156,38	94,40	-3,54	5,50
	19	6	5	5	20	20	0,000221	201161,92	111847,26	1095,59	0,646652	199201,34	116781,27	105,00	-0,97	4,39
	20	7	7	5	20	20	0,256e-7	179428,72	115780,96	70,45	0,778814	173595,87	118800,55	127,27	-3,25	2,61
III	21	15	15	10	20	10	0,000105	177292,96	116231,34	2059,11	0,737498	189432,45	121333,02	220,64	6,85	4,39
	24	15	10	10	30	10	0,000045	236837,12	178156,34	3617,31	0,629997	255481,66	187518,93	202,65	7,87	5,26
	25	10	15	10	30	10	0,000192	260167,32	175468,74	81923,35	0,521111	283435,55	188974,83	211,26	8,94	7,70
	26	10	10	15	30	10	0,000235	256874,01	174207,60	37806,73	0,657936	277556,56	184834,56	229,37	8,05	6,10
	27	20	20	15	30	10	0,434956	226735,79	172648,76	15970,43	0,636894	267093,84	183818,11	434,31	17,80	6,47
	28	10	10	10	40	10	0,000033	308893,12	234623,60	38276,59	0,407033	331095,45	251634,54	206,94	7,19	7,25
	29	20	15	10	40	10	0,970853	296126,73	246321,43	3024,85	0,482416	348861,18	250429,70	329,24	17,81	1,67
	30	15	15	15	40	10	0,002200	338824,70	228826,42	66689,85	0,622868	373806,31	241789,07	363,49	10,32	5,66
IV	31	20	10	15	30	15	0,000015	228953,06	172008,01	36986,71	0,782479	263188,97	177784,21	401,75	14,95	3,36
	32	20	20	20	30	15	0,343781	227390,37	169802,62	11302,52	0,729025	278180,21	181251,65	705,50	22,34	6,74
	33	20	20	10	40	15	0,000701	275009,03	232186,44	16913,70	0,522670	320416,29	247993,90	546,55	16,51	6,81
	34	20	10	15	30	20	0,000507	248331,00	168662,26	17876,90	0,645576	280616,74	179553,08	520,30	13,00	6,46
	35	20	15	20	30	20	0,146e-8	176945,27	165886,50	25379,30	0,629190	199199,93	177793,33	771,40	12,58	7,18
	36	15	10	10	40	20	*	*	*	1204,98	0,558017	328952,86	238989,40	420,11	-	-
	37	15	10	20	40	20	*	*	*	1745,16	0,674865	374483,07	234413,82	681,13	-	-
	38	20	10	20	40	20	*	*	*	1667,00	0,599703	373149,71	241240,01	735,70	-	-
	39	20	15	20	40	20	*	*	*	1831,56	0,640230	374013,89	236673,15	879,42	-	-
	40	20	20	20	40	20	*	*	*	18803,95	0,645302	324611,75	237003,24	1020,56	-	-

* Herhangi bir çözüm elde edilememiştir.

** Her bir problem 10 kez çalıştırılmış ve elde edilen amaç fonksiyon değerleri ile çözüm sürelerinin ortalaması alınmıştır.

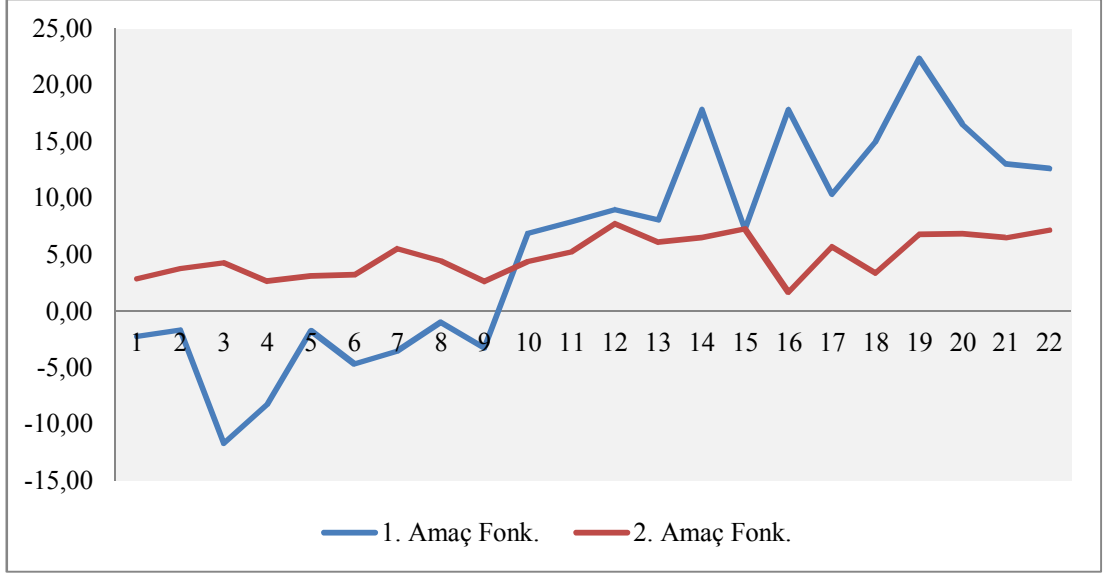
Geliştirilen sezgisel yöntem kullanılarak birinci amaç fonksiyon değerinde %-11,68-%22,34 aralığında, ikinci amaç fonksiyon değerinde %1,67-%7,18 aralığında sapmalar meydana gelmiştir. I. ve II. Tip test problemlerinin TB Algoritması tabanlı sezgisel yaklaşım ile elde edilen çözümleri incelendiğinde; birinci amaç fonksiyon değerinin, GAMS/DICOPT optimizasyon programı ile elde edilen çözümlerden daha iyi olduğu görülmektedir. İkinci amaç fonksiyon değerinde ise düşük sapma oranları ile daha kötü çözümler elde edilmiştir. III. ve IV. Tip test problemlerinde GAMS/DICOPT çözücüsü her iki amaç fonksiyon değeri için daha iyi çözümler vermektedir. Ancak, 36. test probleminden itibaren GAMS/DICOPT optimizasyon programı çözüm elde edememektedir.

III. ve IV. Tip test problemlerinde ise, her iki amaç fonksiyonunda da bir iyileşme sağlanamamış, problemin boyutu büyüdükçe birinci amaç fonksiyon değerinde kötüleşmenin arttığı gözlenmiştir. Aşağıdaki şekilde, farklı boyutlardaki test problemleri için GAMS/DICOPT çözücüsünün elde ettiği çözümler ile önerilen TB Algoritması tabanlı sezgisel yöntemin elde ettiği çözümlerin karşılaştırılması grafiksel olarak gösterilmiştir.



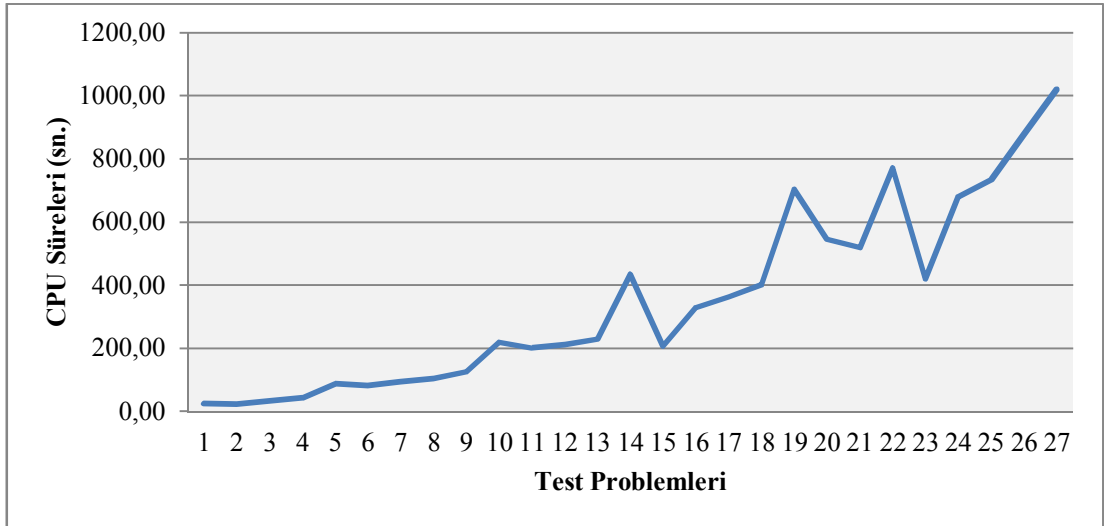
Şekil-31: GAMS/DICOPT Çözücüsü ile Önerilen Sezgisel Yöntemin Elde Ettiği Çözümlerin Karşılaştırılması

Farklı boyutlardaki test problemleri için önerilen sezgisel yöntemin, GAMS/DICOPT çözücüsü ile elde edilen yaklaşık değerlerden sapma oranları ise, grafiksel olarak aşağıdaki şekilde gösterilmiştir.



Şekil-32: Önerilen TB Tabanlı Sezgisel Yöntemin, GAMS/DICOPT Çözücüsü ile Elde Edilen Yaklaşık Değerlerden Sapma Oranları

Geliştirilen sezgisel yöntem, GAMS/DICOPT paket programı ile elde edilen en iyi sonuçların çözüm süreleri ile karşılaştırılmış ve problem boyutu büyüdükçe çok daha kısa sürede tüm test problemleri için çözüme ulaştığı görülmüştür. Aşağıda, TB Algoritması tabanlı sezgisel yaklaşım ile çözülen farklı boyutlardaki test problemlerinin çözüm sürelerinin grafiği verilmiştir.



Şekil-33: Test Problemlerinin Önerilen Sezgisel Yaklaşım ile Elde Edilen Çözüm Süreleri

Her iki yöntemle elde edilen çözümler incelendiğinde; küçük boyutlu problemlerde, önerilen sezgisel yöntemin daha iyi çözümler elde ettiği, problemin boyutu büyüdükçe GAMS/DICOPT optimizasyon programının önerilen sezgisel yöntemden daha iyi çözümler elde ettiği görülmektedir. Ancak, problem boyutu belirli bir büyüklüğe ulaştıktan sonra GAMS/DICOPT optimizasyon programı, problemlerin parametrelerini oluşturmaya ve amaç fonksiyonlarının alt ve üst sınır değerlerini bulmasına rağmen yaklaşık da olsa bir çözüm elde edememektedir. Bütünleşik tedarik zinciri ağında tesis yeri seçimi problemleri NP-Zor sınıfına girdiğinden, bu çalışmada oluşturulan büyük boyutlu problem setlerini GAMS/DICOPT optimizasyon programı ile çözmenin mümkün olmadığı Tablo-16'dan anlaşılmaktadır. Bu nedenle bu tür problemlere çözümler elde edebilmek için TB Algoritması tabanlı sezgisel yaklaşım önerilmiştir. Önerilen sezgisel yaklaşım sayesinde her boyuttaki problem çözümler hale gelmiştir. Aynı zamanda, her iki yöntemin çözüm süreleri karşılaştırıldığında; önerilen sezgisel yöntemin GAMS/DICOPT paket programından çok daha kısa sürede çözümler elde ettiği görülmektedir.

Sonuç olarak; bu çalışmada önerilen sezgisel yaklaşım, GAMS/DICOPT paket programının kısıtlılığı dolayısıyla çözüm elde edilemeyen büyük boyutlu problemler için, karar vericiye (tedarik zinciri yöneticisine), çok kısa bir zamanda kabul edilebilir çözümler sunmaktadır.

BEŞİNCİ BÖLÜM

SONUÇ VE ÖNERİLER

1. SONUÇ

Tedarik zinciri ağ tasarımı modellemesine olan ilgi ve bu konulardaki araştırma alanları son yıllarda küresel pazarların ortaya çıkması, müşteri taleplerinin hızla artması, giderek artan sistem maliyetleri vb. nedenlerle büyük artış göstermiştir. Küresel pazarlarda, sürekli rekabet halindeki işletmeler dikkatlerini iş süreçlerinin etkinliği ve etkililiği üzerinde yoğunlaştırmışlardır. Planlama, kontrol ve tasarım gibi birçok stratejik fonksiyonu bünyesinde barındıran TZY'nin önemi daha iyi kavranmaya başlanmıştır.

Bu çalışmada, belirsizlik altında, malzeme ihtiyaç kısıtlı, çok ürünlü, çok aşamalı ve çok dönemli bir tedarik zinciri ağındaki birden fazla ölçülemeyen amacı gerçekleştirmek için tedarik zinciri planlama modeli önerilmiştir. Önerilen modelde, karar vericilerin kesin olmayan hedef değerleri ile belirsiz müşteri taleplerini temsil etmek üzere bulanık hedef programlama yaklaşımı kullanılmıştır. Ayrıca, tedarik zinciri literatüründeki benzer çalışmalardan farklı olarak siparişlerin gecikmeli olarak karşılandığı EÜM modeline göre sipariş verme, stokta tutma ve yok satma maliyetlerine yer verilmiştir.

Bulanık hedef programlama modeli, sistemin çıktılarının en iyilenmesinin yanında en iyi çıktıyı veren girdi bileşiminin belirlenmesine ve optimal bir sistemin tasarlanmasına yardımcı olabilmektedir. Problemdaki bulanık hedeflerin istek seviyelerinin belirlenmesi için, model her bir amaç için ayrı ayrı çözülerek başlangıç çözümler elde edilmiş ve üyelik fonksiyonlarının alt ve üst sınırları belirlenmiştir. Eğer karar verici başlangıç çözümünü yeterli bulmaz ise modelin, tercih edilen bir doyurucu çözüm elde edilinceye kadar bulanık amaç fonksiyonlarının alt ve üst sınırlarını güncelleştirilmesi imkanı sağlanmıştır. Önerilen modelin temel avantajı, karar vericinin tercih edilen etken çözümü elde etmek için çözüm süreci esnasında arama yönünü ayarlayan sistematik bir çerçeve sunmasıdır.

Bu çalışmada önerilen çok amaçlı doğrusal olmayan programlama modeli için oluşturulan test problemlerinin çözümünde kullanılan GAMS/DICOPT optimizasyon programı, problemin NP-Zor problemler sınıfına girmesi ve doğrusal olmayan yapısı nedeniyle küçük boyutlu problemde dahi optimal çözümü garanti edememektedir. Bu nedenle, üretilen test problemlerinin GAMS/DICOPT optimizasyon programı ile çözümünde yaklaşık sonuçlar elde edilmiştir. Üstelik, problem boyutu büyüdükçe çözüm daha da zorlaşmakta ve belli bir boyuttan sonra (dönem, müşteriler ve aday tesis sayısına bağlı olarak) imkansız hale gelmektedir. Bu tür problemleri çözebilmek için, önerilen matematiksel modele özgü TB Algoritması tabanlı sezgisel bir yöntem geliştirilmiştir. Geliştirilen yöntemin etkinliğinin gösterilmesi ve karşılaştırma yapmak için seçilen test problemleri üzerinde sayısal analizler yapılmıştır. Yapılan analizler sonucunda test problemleri için önerilen sezgisel yöntemin, GAMS/DICOPT paket programı ile çözüm elde edilemeyen büyük boyutlu problemlerde de kabul edilebilir çözümlere ulaştığı görülmüştür. Böylece, büyük boyutlu problemler için GAMS/DICOPT optimizasyon programı ile yaklaşık olarak dahi elde edilemeyen çözümler, geliştirilen TB Algoritması tabanlı sezgisel yöntem ile elde edilir hale gelmiştir. Önerilen matematiksel modelin GAMS/DICOPT optimizasyon programı ile elde edilen çözümler ile TB Algoritması tabanlı sezgisel ile elde edilen çözümler, çözüm süreleri bakımından karşılaştırıldığında, TB Algoritmasının, GAMS/DICOPT paket programından çok daha kısa sürede kabul edilebilir çözümlere ulaştığı görülmüştür.

2. GELECEKTE YAPILACAK ÇALIŞMALAR İÇİN ÖNERİLER

Gelecekte bu konu ile ilgili yapılacak çalışmalarda, hedeflere ilişkin erişim düzeyleri (istek seviyeleri) ve toleransların (üyelik fonksiyonlarının alt ve üst sınırları) karar verici tarafından belirlenmesi ile daha farklı sonuçların elde edilmesi mümkün olacaktır. Yapılan çalışmada sadece hedeflerin erişim düzeyleri ile talep kısıtı bulanık olarak incelenmiştir, hedeflerin katsayılarının veya diğer kısıtların da bulanık olduğu durumların incelenmesi literatüre farklı uygulamalar kazandırılması açısından önem taşımaktadır. Ayrıca, çalışmada geliştirilen TB Algoritması tabanlı sezgisel yöntem, orta ve büyük boyutlu test problemlerinde GAMS/DICOPT paket programının elde ettiğinden daha iyi çözümler elde edilmesi için iyileştirilebilir, diğer meta-sezgisel yöntemlerle karşılaştırılabilir.

KAYNAKÇA

- ALTIPARMAK, Fulya, M. GEN, L. LIN, T. PAKSOY. "A Genetic Algorithm Approach for Multi-Objective Optimization of Supply Chain Networks", **Computers and Industrial Engineering**, 51, 2006, 197-216.
- ARABANI A.B. ve R.Z. FARAHANI. "Facility Location Dynamics: An Overview of Classifications and Applications", **Computers and Industrial Engineering**, 62 (1), 2012, 408-420.
- BALLOU, R.H. "Dynamic Warehouse Location Analysis", **Journal of Marketing Research**, 5, 1968, 271-276.
- BEHRET, Hülya. **Üretim Sistemlerinde Bulanık Tek Dönemli Stok Kontrol Modelleri** (Doktora Tezi), İTÜ Fen Bilimleri Enstitüsü, 2011.
- BELLMAN, R.E., L.A. ZADEH. "Decision-Making in a Fuzzy Environment", **Management Science**, 17 (4), 1970, 141-164.
- BENYOUCEF, Lyes, Xiaolan XIE, Guy Aime TANONKOU. "Supply Chain Network Design with Unreliable Suppliers: A Lagrangian Relaxation-Based Approach", **International Journal of Production Research**, 51 (21), 2013, 6435-6454.
- BRAMEL J., D. SIMCHI-LEVI. (Ed.) **The Logic Of Logistics: Theory, Algorithms, and Applications for Logistics Management**, Springer Series in Operations Research, New York, 1997.
- CADENAS, J.M. ve J.L. VERDEGAY. "Using Ranking Functions in Multiobjective Fuzzy Linear Programming", **Fuzzy Sets and Systems**, 111, 2000, 47-53.
- CANEL, C., KHUMAWALA, B.M. "Multi-Period International Facilities Location: An Algorithm and Application", **International Journal of Production Economics**, 35, 1997, 1891-1910.
- CHANG, S.C. "Fuzzy Production Inventory for Fuzzy Product Quantity with Triangular Fuzzy Number", **Fuzzy Sets and Systems**, 107 (1), 1999, 37-57.
- CHANG, P.T., C.H. CHANG. "An Elaborative Unit Cost Structure-Based Fuzzy Economic Production Quantity Model", **Mathematical and Computer Modeling**, 43, 11-12, 2006, 1337-1356.
- CHEN, S.H., S.M. CHANG. "Optimization of Fuzzy Production Inventory with Unrepairable Defective Products Model for Items with Imperfect Quality", **International Journal of Production Economics**, 113, 2008, 887-894.
- CHEN, L. ve W. LEE. "Multi Objective Optimization of Multi Echelon Supply Chain Networks with Uncertain Product Demands and Prices", **Computers and Chemical Engineering**, 28, 2004, 1131-1144.

CHEN, S.H., C.C.WANG, S.M.CHANG. “Fuzzy Economic Production Quantity Model for Items with Imperfect Quality”, **International Journal of Innovative Computing, Information and Control**, 3 (1), 2007, 85-95.

CHOPRA, S., P. MEINDL (Ed.). **Supply Chain Management: Strategy, Planning, and Operation**, Pearson Prentice Hall, 2nd Edition, New Jersey, 2001.

CHURCH, R. ve C. REVELLE. “The Maximal Covering Location Problem”, **Papers of the Regional Science Association**, 32, 1974.

COHEN, M.A., H.L. LEE. “Resource Deployment Analysis of Global Manufacturing and Distribution Networks”, **Journal of Manufacturing and Operations Management**, 2, 1989, 81-104.

DASKIN, Mark S. **Network and Discrete Location: Models, Algorithms, and Applications**, New York, NY: Wiley, 1995.

DENSHAM, P. J. ve G. RUSHTON, “A More Efficient Heuristic for Solving Large P-Median Problems”, **Papers in Regional Science: The Journal of The RSAI**, 72, 1992, 307-329.

DREZNER, Z. ve G.O. WESOLOWSKY. “Location on a One-way Rectilinear Grid”, **Journal of the Operational Research Society**, 46, 6, 1995, 735-746.

DREZNER, Zvi. **Facility Location: A Survey of Applications and Methods**, Springer-Verlag, New York, 1995.

DREZNER, Zvi ve H.W. HAMACHER. **Facility Location: Applications and Theory**, Springer-Verlag, New York, 2002.

ERLENKOTTER, D.A., “Comparative Study of Approaches to Dynamic Location Problems”, **European Journal of Operational Research**, 6, 1981, 133-143.

FAHIMNIA, B., R.Z. FARAHANI, R. MARIAN, L. LUONG. “A Review and Critique on Integrated Production–Distribution Planning Models and Techniques”, **Journal of Manufacturing Systems**, 32, 2013, 1-19.

GAREY, M.R. ve D.S. JOHNSON. **Computers and Intractability: A Guide to The Theory of NP-Completeness**, New York, Freeman and Company, 1979.

GUE, K.R. “A Dynamic Distribution Model for Combat Logistics”, **Computers and Operations Research**, 30, 2003, 367–81.

GÜNER, Ertan ve Fulya ALTIPARMAK, “İki Ölçütlü Tek Makinalı Çizelgeleme Problemi için Sezgisel Bir Yaklaşım”, **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, 18 (3), 2003, 27-42.

GÜMÜŞ, A.T., A.F. GÜNERİ, S. KELEŞ. “Supply Chain Network Design Using An Integrated Neuro-Fuzzy and MILP Approach: A Comparative Design Study”, **Expert Systems with Applications**, 36 (10), 2009, 12570-12577.

HAKIMI, S. L. "Optimum Locations of Switching Centers and The Absolute Centers and Medians of a Graph, **Operations Research**, 12, 1964, 450 - 459.

HANNAN, E.L. "On Fuzzy Goal Programming", **Decision Sciences**, 12 (3), 1981, 522-531.

HARRIS, F. "How Many Parts to Make At Once", **Factory Mag Manag**, 10 (2), 1913, 136–152.

HINOJOSA, Y., J. PUERTO, F. FERNANDEZ, "A Multi-Period Two-Echelon Multi-Commodity Capacitated Plant Location Problem." **European Journal of Operational Research**, 123 (2), 2000, 271–291.

HSIEH, C.H. "Optimization Of Fuzzy Production Inventory Models", **Information Sciences**, 146, 2002, 29-40.

IGNIZO, James. **Linear Programming in Single & Multiple Objective Systems**, Englewood Cliffs, Prentice Hall, 1982.

İŞİK, A.T., M. ÖZDEMİR. "Bütünleşik Üretim Planlamasında Etkileşimli Olabilirlikçi Doğrusal Programlama Modeli ve Bir Uygulama", **DEÜ İşletme Fakültesi Dergisi**, 11 (2), 2010, 81-117.

INUIGUCHI, M., M. SAKAWA, Y. KUME. "The Usefulness of Possibilistic Programming in Production Planning Problems", **International Journal of Production Economics**, 33, 1994, 45–52.

KABAK, Özgür. **Olabilirsel Doğrusal Programlama ile Tedarik Zinciri Ağ Yapısının Modellenmesi ve Bir Uygulama (Doktora Tezi)**, İTÜ Fen Bilimleri Enstitüsü, İstanbul, 2008.

KABAK, Özgür ve Füsun ÜLENGİN. "Possibilistic Linear-Programming Approach for Supply Chain Networking Decisions", **European Journal of Operational Research**, 209, 2011, 253–264.

KACPRZYK, Janusz ve Sergei A. ORLOVSKI. **Optimization Models Using Fuzzy Sets and Possibility Theory**, Dordrecht, Kluwer Academic Publishers, 1987.

KIRKPATRICK S., C.D. GELATT, M.P. VECCHI. "Optimization by Simulated Annealing", **Science**, New Series, 220 (4598), 1983, 671-680.

KO, Hyun Jeung, Gerald W. EVANS. "A Genetic Algorithm-Based Heuristic for The Dynamic Integrated Forward/Reverse Logistics Network For 3PLs", **Computers & Operations Research**, 34, 2007, 346–366.

KLOSE, Andreas ve Andreas DREXL. "Facility Location Models for Distribution System Design", **European Journal of Operational Research**, 162, 2005, 4-29.

KULMALA, Harri Ilmari. "Developing Cost Management in Customer–Supplier Relationships: Three Case Studies", **Journal of Purchasing and Supply Management**, 10, 2004, 65–77.

LAI, Young-Jou, Ching-Lai HWANG. "A New Approach to Some Possibilistic Linear Programming Problem", **Fuzzy Sets and Systems**, 49, 1992, 121-133.

LAI, Young-Jou, Ching-Lai HWANG. "Fuzzy Multiple Objective Decision Making, Methods and Applications", **Lecture Notes in Economics and Mathematical Systems**, Springer, Berlin, 1994.

LAKHAL, S., A. MARTEL, O. KETTANI, ve M. ORAL. "On The Optimization of Supply Chain Networking Decisions", **European Journal of Operational Research**, 129, 2001, 259-270.

LIANG, Tien-Fu. "Distribution Planning Decisions Using Interactive Fuzzy Multi-Objective Linear Programming", **Fuzzy Sets and Systems**, 157, 2006, 1303-1316.

LIANG, Tien-Fu. "Applying Fuzzy Goal Programming to Production/Transportation Planning Decisions in a Supply Chain", **International Journal of Systems Science**, 38, 2007, 293-304.

LIANG, Tien-Fu. "Integrating Production-Transportation Planning Decision With Fuzzy Multiple Goals in Supply Chains", **International Journal of Production Research**, 46 (15), 2008, 1477-1497.

LIANG, Tien-Fu., Hung-Wen CHENG. "Application of Fuzzy Sets to Manufacturing/Distribution Planning Decisions with Multi-Product and Multi-Time Period in Supply Chains", **Expert Systems with Applications**, 36, 2009, 3367-3377.

LIANG, Tien-Fu. "Application of Fuzzy Sets to Manufacturing/Distribution Planning Decisions in Supply Chains", **Information Sciences**, 181, 2011, 842-854.

LU, Zhiqiang, Nathalie BOSTEL. "A Facility Location Model for Logistics Systems Including Reverse Flows: The Case of Remanufacturing Activities", **Computers and Operations Research**, 34, 2007, 299-323.

MARIANOV, V. ve C. REVELLE. "Siting Emergency Services", Z. DREZNER (Ed.), **Facility Location: A Survey of Articles, Applications And Methods**, Springer Series in Operations Research, New York, 10, 1995, 199-222.

MARIANOV, V. ve C. REVELLE. "The Queueing Maximal Availability Location Problem: A Model for The Siting of Emergency Vehicles", **European Journal of Operational Research**, 93, 1996, 110-120.

MELACHRINOUDIS, E. ve H. MIN, "The Dynamic Relocation and Phase-Out of a Hybrid, Two-Echelon Plant/Warehousing Facility: A Multiple Objective Approach", **European Journal of Operational Research**, 123, 2000, 1-15.

MELKOTE, Sanjay ve Mark S. DASKIN. "Capacitated Facility Location/Network Design Problems", **European Journal of Operational Research**, 129, 2001, 481-495.

MELO, M.T., S. NICKEL, F. SALDANHA-DA-GAMA. "Facility Location and Supply Chain Management – A Review", **European Journal of Operational Research**, 196, 2009, 401-412.

METROPOLIS N., A. ROSENBLUTH, M. ROSENBLUTH, A. TELLER, E. TELLER. "Equation of State Calculations by Fast Computing Machines", **Journal of Chemical Physics**, 21(6), 1953, 1087-1092.

MIN H. ve G. ZHOU, "Supply Chain Modeling: Past, Present and Future", **Computers & Industrial Engineering**, 43, 2002, 231-249.

MIRCHANDANI, P. B. ve R.L. FRANCIS. **Discrete Location Theory**, Wiley, New York, 1990.

MULA, J., R. POLER, J.P. GARCIA-SABATER, F.C. LARIO, "Models for Production Planning Under Uncertainty: A Review", **International Journal of Production Economics**, 103, 2006, 271-285.

MULA, J., D. PEIDRO, R. POLER. "The Effectiveness of a Fuzzy Mathematical Programming Approach for Supply Chain Production Planning with Fuzzy Demand", **International Journal of Production Economics**, 128, 2010a, 136-143.

MULA J., D. PEIDRO, M.D. MADRANOREO, E. VICENS. "Mathematical Programming for Supply Chain Production and Transport Planning", **European Journal of Operational Research**, 204, 2010b, 377-390.

NARASIMHAN, Ram. "Goal Programming in a Fuzzy Environment", **Decision Sciences**, 11, 1980, 325-336.

OWEN, S.H. ve M.S. DASKIN. "Strategic Facility Location: A Review", **European Journal of Operational Research**, 111, 1998, 423-447.

ÖZDAMAR L. ve T. YAZGAÇ. "Capacity Driven Due Date Settings In Make-to-Order Production Systems", **International Journal of Production Economics**, 49 (1), 29-44, 1997.

ÖZKAN, Mustafa M, **Bulanık Hedef Programlama**, Bursa, Ekin Kitabevi, 2003.

ÖZTÜRK, Özer ve Orhan GERDAN, **Stok Planlama**, Kara Harp Okulu Basımevi, 2005.

PAKSOY, Turan, Eren ÖZCEYLAN, G.W.WEBER. "A Multi-Objective Mixed Integer Programming Model for Multi Echelon Supply Chain Network Design and Optimization", **System Research and Information Technologies**, METU, 2009.

PAKSOY, Turan, Nimet YAPICI PEHLİVAN, Eren ÖZCEYLAN. "Application of Fuzzy Optimization to a Supply Chain Network Design: A Case Study of an Edible Vegetable Oils Manufacturer", **Applied Mathematical Modelling**, 36, 2012, 2762-2776.

PAKSOY, Turan, Y.P.PEHLİVAN. "A Fuzzy Linear Programming Model for The Optimization of Multi-Stage Supply Chain Networks with Triangular and Trapezoidal Membership Functions", **Journal of the Franklin Institute**, 349, 2012, 93-109.

PAKSOY, Turan, Nimet YAPICI PEHLİVAN, Eren ÖZCEYLAN. “A New Tradeoff Model for Fuzzy Supply Chain Network Design and Optimization”, **Human and Ecological Risk Assessment: An International Journal**, 19 (2), 2013, 492-594.

PETROVIC, D.R. ROY, R. PETROVIC. “Supply Chain Modeling Using Fuzzy Sets”, **International Journal of Production Economics**, 59, 1999, 443.

PINCUS, M. “A Monte-Carlo Method for The Approximate Solution of Certain Types of Constrained Optimization Problems”, **Operations Research**, 18, 1970, 1225–1228.

PISHVAE, M.S., J. RAZMI, S.A. TORABI, “Robust possibilistic programming for socially responsible supply chain network design: A new approach”, **Fuzzy Sets and Systems**, 206, 1-20, 2012.

PİRKUL, Hasan ve V. JAYARAMAN, “A Multi-Commodity, Multiplant, Capacitated Facility Location Problem: Formulation and Efficient Heuristic Solution”, **Computers Operations Research**, 25, 869–878, 1998.

PYKE D.F. ve M.A. COHEN. “Performance Characteristics of Stochastic Integrated Production Distribution Systems”, **European Journal of Operational Research**, 68 (1), 23-48, 1993.

REEVES, C.R. **Modern Heuristic Techniques for Combinatorial Problems**, Wiley and Sons, 1993.

REVELLE, Charles S., Justin C. WILLIAMS S. ve John J. BOLAND. “Counterpart Models in Facility Location Science and Reserve Selection Science”, **Environmental Modeling and Assessment**, 7, 2002, 71-80.

ROY, T. K. ve M. MAITI. “A Fuzzy EOQ Model with Demand-Dependent Unit Cost Under Limited Storage Capacity”, **European Journal of Operational Research**, 99, 1997, 425-432.

ROY, T. K. ve M. MAITI. “Multi-Objective Inventory Models of Deteriorating Items with Some Constraints in a Fuzzy Environment”, **Computers and Operations Research**, 25 (12), 1998, 085-1095.

QIN, J.F., L. SHI, MIAO, ve G. TAN, “Optimal Model and Algorithm for Multi-Commodity Logistics Network Design Considering Stochastic Demand and Inventory Control”, **Systems Engineering — Theory & Practice**, 29 (4), 2009.

SELİM, Hasan, Ceyhun ARAZ, ve İrem ÖZKARAHAN. “An Integrated Multi-Objective Supply Chain Model In A Fuzzy Environment”, **Makine Mühendisleri Odası Endüstri Mühendisliği Dergisi**, 15 (3), 2004, 2-16.

SELİM, Hasan, Ceyhun ARAZ, ve İrem ÖZKARAHAN. “Collaborative Production-Distribution Planning in Supply Chain: A Fuzzy Goal Programming Approach”, **Transp. Res E**, 44, 2008, 396-419.

SHAHABI, Mehrdad, S. AKBARINASAJI, A.UNNIKRISHNAN ve R. JAMES. “Integrated Inventory Control and Facility Location Decisions in a Multi-Echelon Supply Chain Network with Hubs”, **Networks & Spatial Economics**, 13 (4), 2013, 497-514.

SHAPIRO, Jeremy F. **Modeling the Supply Chain (2nd Edition)**, Duxbury Thomson Learning, 2007.

SIMCHI-LEVI, D., P. KAMINSKY, E. SIMCHI-LEVI. **Designing and Managing the Supply Chain: Concepts, Strategies, and Cases**, 2nd Edition, McGraw-Hill, New York, 2003.

SHULMAN, A. “An Algorithm for Solving Dynamic Capacitated Plant Location Problems with Discrete Expansion Sizes”, **Operations Research**, 39, 1991, 423-436.

SOURIRAJAN K., L. OZSEN, R. UZSOY. “A Genetic Algorithm for A Single Product Network Design Model with Lead Time and Safety Stock Considerations”, **European Journal of Operational Research**, 197 (2), 2009, 599-608.

STOCK James ve Douglas LAMBERT. **Strategic Logistics Management**, McGraw-Hill, New York, 2001.

SYARIF, A., Y.S.YUN, M.GEN. "Study on Multi-Stage Logistic Chain Network: A Spanning Tree-Based Genetic Algorithm Approach", **Computer and Industrial Engineering**, 43, 2002, 299–314.

THANH, P.N., N. BOSTEL, O. PETON. “A Dynamic Model for Facility Location in The Design of Complex Supply Chains”, **International Journal of Production Economics**, 113, 2008, 678–693.

TORABI S. Ali, Elkafi HASSINI. “An Interactive Possibilistic Programming Approach for Multiple Objective Supply Chain Master Planning”, **Fuzzy Sets and Systems**, 159, 2008, 193–214.

TUZKAYA, Umut ve Semih ÖNÜT. “A Holonic Approach Based İntegration Methodology for Transportation and Warehousing Functions of The Supply Network”, **Computers and Industrial Engineering**, 56, 2009, 708-723.

TÜRKŞEN İsmail Burhan, M. Hossein Fazel ZARANDI ve S.SAGHIRI. “Supply Chain: Crisp and Fuzzy Aspects”, **Int. J. of. Applied Mathematics and Computer Science on computing with words and perceptions**, 12 (3), 2002, 423-436.

XIE, Y., D. PETROVIC, K. BURNHAM. “A Heuristic Procedure for The Two-Level Control of Serial Supply Chains Under Fuzzy Customer Demand”, **International Journal of Production Economics**, 102 (1), 2006, 37-50.

XU, J., Q. LIU, ve R. WANG. “A Class of Multi-Objective Supply Chain Networks Optimal Model Under Random Fuzzy Environment and Its Application to The Industry of Chinese Liquor, **Information Sciences**, 178 (8), 2008, 2022-2043.

VILA, D., A. MARTEL, R. BEAUREGARD. "Designing Logistics Networks In Divergent Process Industries: A Methodology and Its Application to The Lumber Industry", **International Journal of Production Economics**, 102, 2006, 358–378.

WANG, J. ve Y.-F. SHU. "Fuzzy Decision Modeling for Supply Chain Management", **Fuzzy Sets and Systems**, 150, 2005, 107-127.

WANG, H.-F. ve M.-L. WANG. "A Fuzzy Multiobjective Linear Programming", **Fuzzy Sets and Systems**, 86, 1997, 61-72.

WEBER, Alfred. **Über den Standort der Industrien**, Erster Teil: Reine Theorie des Standorte, 1909.

YAGER, R. "A Procedure for Ordering Fuzzy Subsets of The Unit Interval", **Information Sciences**, 24, 1981, 143–161.

ZADEH, L.A. "Fuzzy sets", **Information and Control**, 8 (3), 1965, 338.

ZADEH, L.A. "Fuzzy Sets As a Basis for a Theory of Possibility", **Fuzzy Sets and Systems**, 1, 1978, 3–28.

ZIMMERMANN, H.J. "Description and Optimization of Fuzzy Systems", **International Journal of General Systems**, 2, 1976, 209-215.

ZIMMERMANN, H.J. "Fuzzy Programming and Linear Programming with Several Objective Functions", **Fuzzy Sets and Systems**, 1, 1978, 45-55.

ZIMMERMAN, H.J. **Fuzzy Sets Theory and Its Applications**, Kluwer Academics Publishers, 4th Edition, Boston, 2001.

ZIMMERMANN, H.J. "An Application Oriented View of Modelling Uncertainty", **European Journal of Operational Research**, 122, 2000, 190-198.

EKLER

- EK-A** : MODELİN GAMS/DICOPT PROGRAM KODLARI
- EK-B** : MODELİN GAMS/DICOPT PROGRAM ÇIKTILARI
- EK-C** : TEST PROBLEMİ ÜRETME C++ KODLARI
- EK-Ç** : TB ALGORİTMASI TABANLI SEZGİSELİN C++ KODLARI

MODELİN GAMS/DICOPT PROGRAM KODLARI

```

$title Model (EPQ)
option iterlim = 1000000 ;
option minlp = dicopt, nlp = conopt, mip = cplex, reslim = 72000 ;
file opt cplex option file /cplex.opt/;
put opt;
put 'preind 0';
putclose opt;
option SYSOUT = ON ;
sets
i tedarik merkezleri /sup1,sup2,sup3/
j üretim tesisleri /plant1,plant2,plant3/
k dağıtım merkezleri /warehouse1,warehouse2,warehouse3/
l müşteriler /1*5/
r hammaddeler /raw1,raw2,raw3/
t dönem sayısı /1*3/;
Parameter a(i,r,t) i tedarik merkezinin r hammadde kapasitesi (1000 ton cinsinden)
/sup1.raw1.1 5604
sup1.raw1.2 5486
sup1.raw1.3 5944
sup1.raw2.1 5712
sup1.raw2.2 5283
sup1.raw2.3 4556
sup1.raw3.1 4734
sup1.raw3.2 5286
sup1.raw3.3 5361
sup2.raw1.1 5761
sup2.raw1.2 4544
sup2.raw1.3 5800
sup2.raw2.1 5132
sup2.raw2.2 5843
sup2.raw2.3 5079
sup2.raw3.1 4813
sup2.raw3.2 5098
sup2.raw3.3 5206
sup3.raw1.1 4845
sup3.raw1.2 4961
sup3.raw1.3 5491
sup3.raw2.1 5092
sup3.raw2.2 5237
sup3.raw2.3 5743
sup3.raw3.1 5512
sup3.raw3.2 5028
sup3.raw3.3 5859/;
Parameter b(j,t) j üretim tesisinin ürün kapasitesi (1000 ton cinsinden)
/plant1.1 659
plant1.2 525
plant1.3 692

```

plant2.1	981
plant2.2	521
plant2.3	967
plant3.1	717
plant3.2	647
plant3.3	844/;

Parameter $c(k,t)$ k dagitim merkezinin urun kapasitesi (1000 ton cinsinden)

/warehouse1.1	210
warehouse1.2	219
warehouse1.3	208
warehouse2.1	345
warehouse2.2	307
warehouse2.3	389
warehouse3.1	275
warehouse3.2	244
warehouse3.3	205/;

Parameter $f_u(j,t)$ j uretim tesisinin sabit maliyetleri (acma ve isletme)

/plant1.1	827.5
plant1.2	842.5
plant1.3	935
plant2.1	830
plant2.2	880
plant2.3	777.5
plant3.1	820
plant3.2	795
plant3.3	862.5/;

Parameter $f_w(k,t)$ k dagitim merkezinin sabit maliyetleri (acma ve isletme)

/warehouse1.1	202.6
warehouse1.2	279.6
warehouse1.3	241.8
warehouse2.1	229.2
warehouse2.2	268.4
warehouse2.3	309
warehouse3.1	244.6
warehouse3.2	213.8
warehouse3.3	317.4/;

Parameter $d(l,t)$ t. donemde l musterisinin urun talebi

/1.1	120
1.2	116
1.3	108
2.1	106
2.2	125
2.3	105
3.1	114
3.2	133
3.3	130
4.1	100
4.2	136
4.3	134
5.1	100

5.2 106

5.3 124/;

Parameter AS(l,t) t. donemde l musterisinin urun talebi alt limit;

$AS(l,t) = d(l,t) * 0.8$;

Parameter US(l,t) t. donemde l musterisinin urun talebi ust limit;

$US(l,t) = d(l,t) * 1.4$;

Parameter ro(r) Bir birim bitmis urun uretmek icin gerekli olan r hammaddesi miktarı

/raw1 3

raw2 4

raw3 3/;

Parameter alfa(j,t) Fire oranı

/plant1.1 0.3

plant1.2 0.2

plant1.3 0.1

plant2.1 0.2

plant2.2 0.1

plant2.3 0.1

plant3.1 0.1

plant3.2 0.3

plant3.3 0.3/;

Parameter d1(i,j) i tedarik merkezi ile j uretim tesisi arasindaki mesafe (km.)

/sup1.plant1 307.2

sup1.plant2 209.2

sup1.plant3 268

sup2.plant1 254

sup2.plant2 156

sup2.plant3 335.2

sup3.plant1 268

sup3.plant2 237.2

sup3.plant3 329.6/;

Parameter d2(j,k) j uretim tesisi ile k dagitim merkezi arasindaki mesafe (km.)

/plant1.warehouse1 146.5

plant1.warehouse2 101.5

plant1.warehouse3 134.5

plant2.warehouse1 125.5

plant2.warehouse2 130

plant2.warehouse3 181

plant3.warehouse1 197.5

plant3.warehouse2 130

plant3.warehouse3 76/;

Parameter d3(k,l) k dagitim merkezi ile l musterileri arasindaki mesafe (km.)

/warehouse1.1 35.5

warehouse1.2 11.5

warehouse1.3 56.5

warehouse1.4 43.5

warehouse1.5 51.5

warehouse2.1 21.5

warehouse2.2 18

warehouse2.3 40

warehouse2.4 14

warehouse2.5 18.5

warehouse3.1 30

warehouse3.2 13.5

warehouse3.3 49.5

warehouse3.4 16

warehouse3.5 18.5/;

Parameter Cc(k,t) Birim satinalma maliyeti (TL)

/warehouse1.1 54

warehouse1.2 67.2

warehouse1.3 59.2

warehouse2.1 65.8

warehouse2.2 64.4

warehouse2.3 53

warehouse3.1 58.8

warehouse3.2 65.8

warehouse3.3 62.6/;

Parameter Co(k,t) Siparis verme maliyeti (TL)

/warehouse1.1 216

warehouse1.2 227

warehouse2.1 295

warehouse2.2 276

warehouse3.1 260

warehouse3.2 216/;

Parameter Ch(k,t) k dagitim merkezindeki yillik stokta bulundurma maliyeti

/warehouse1.1 9.5

warehouse1.2 26

warehouse2.1 13.4

warehouse2.2 11.6

warehouse3.1 29.9

warehouse3.2 34.1/;

Parameter Cb(k,t) k dagitim merkezindeki yillik yoksatma maliyeti

/warehouse1.1 27.2

warehouse1.2 31.6

warehouse2.1 28.8

warehouse2.2 27.8

warehouse3.1 32.8

warehouse3.2 31.2/;

Scalar f 100 km. basina harcanan yakit (litre cinsinden) /5.1/ ;

Parameter C1(i,j) i ted. mrk.den j ur. tesisine gond. urunun br. tasima maliyeti ;

$C1(i,j) = f * d1(i,j) / 100$;

Parameter C2(j,k) j ur. tesisinden k dagitim mrk.ne gond. urunun br. tasima maliyeti ;

$C2(j,k) = f * d2(j,k) / 100$;

Parameter C3(k,l) k dagitim mrk.den l musterisine gond. urunun br. tasima maliyeti ;

$C3(k,l) = f * d3(k,l) / 100$;

Parameter price(t) Harcanan 1 lt. akaryakitinin donem basina fiyati

/1 1.14

2 1.19

3 1.47/;

Parameter PU(t) acilabilecek top. uretim tesisi sayisi ust siniri /1 1, 2 2, 3 2/;

Parameter PW(t) acilabilecek top. dagitim mrk. sayisi ust siniri /1 2, 2 1, 3 2/;

Variables

$X(i,j,r,t)$ j üretim tesisinde i tedarik merkezinden r hammaddesi kull. üretilen miktar

$Y(j,k,t)$ j üretim tesisinden k dağıtım merkezine gönderilen p ürün miktarı

$Z(k,l,t)$ k dağıtım merkezinden l müşterisine gönderilen miktar

$v(k,l,t)$ k dağıtım merkezinden l müşterisine p ürün gönderilirse

$w(j,t)$ j üretim tesisi açık ise

$zz(k,t)$ k dağıtım merkezi açık ise

quantity Ekonomik Üretim Miktarı

backlog Yoksatma miktarı ;

Free Variables lambda ;

Free Variable TM, SM;

Positive Variables X, Y, Z, quantity, backlog ;

Binary Variables v, w, zz ;

Equations

objective1 amaç fonksiyonu 1

objective2 amaç fonksiyonu 2

kisitobj1 amaç kısıtı 1

kisitobj2 amaç kısıtı 2

kisit2(i,r,t) Tedarik merkezlerinden üretim tesislerine gönderilen hammaddelerin tedarik merkezinin kapasitesini asmamasını sağlar

kisit3(j,r,t) Tedarik merkezlerinden üretim tesislerine gönderilen hammaddelerin en az ürünü üretmek için gerekli olan hammadde kadar olmasını sağlar

kisit4(j,t) Üretim tesislerinden dağıtım merkezlerine gönderilen ürünlerin açık olan tesislerin kapasitesinden fazla olmamasını sağlar

kisit5(t) Acilabilecek dağıtım merkezi sayısının üst sınırı

kisit6(k,t) Dağıtım merkezi akis denge kısıtı

kisit7(k,l,t) Ürünler için müşteri taleplerinin dağıtım merkezlerinden gönderilen toplam ürün miktarından fazla olmamasını sağlar

kisit8(k,t) Müşteri noktasındaki toplam ürün miktarının dağıtım merkezlerinin toplam ürün kapasitesini aşmayacağını garanti eder

kisit9(t) Acilabilecek müşteri noktası sayısının üst sınırı

kisit10(l) Her müşterinin sadece bir dağıtım merkezinden hizmet almasını sağlar

epq Ekonomik Üretim Miktarı

yoksatma Yoksatılan miktar ;

objective1 .. lambda = 1 - (sum((i,j,r,t), C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t), C2(j,k)*price(t)*Y(j,k,t)) + sum((k,l,t), C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t), fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t))) / (114359 - 68082.1) ;

objective2 .. lambda = 1 - (36421.6 - (sum((k,l,t), Cc(k,t)*Z(k,l,t)) + sum((k,l,t), Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))) + ((sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqrt(sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))) - sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t))))+1/sum((k,l,t), Z(k,l,t)))+(sum((k,l,t), Cb(k,t)*Z(k,l,t))*sqrt(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/

```

(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t),
Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*((1/(sum((j,k,t),
Y(j,k,t))-sum((k,l,t), Z(k,l,t))))+1/sum((k,l,t), Z(k,l,t)))) / (36421.6 - 29165.7) ;
kisitobj1 .. TM =e= sum((i,j,r,t), C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t),
C2(j,k)*price(t)*Y(j,k,t)) +sum((k,l,t), C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t),
fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t)) ;
kisitobj2 .. SM =e= sum((k,l,t), Cc(k,t)*Z(k,l,t)) + sum((k,l,t),
Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))) + ((sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqrt(sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-
sum((k,l,t), Z(k,l,t))))+1/sum((k,l,t), Z(k,l,t)))+ ((sum((k,l,t),
Cb(k,t)*Z(k,l,t))*sqrt(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t))))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/
(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t),
Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*((1/(sum((j,k,t),
Y(j,k,t))-sum((k,l,t), Z(k,l,t))))+1/sum((k,l,t), Z(k,l,t)))) ;
kisit2(i,r,t) .. sum((j), X(i,j,r,t)) =l= a(i,r,t) ;
kisit3(j,r,t) .. sum((i), X(i,j,r,t)) =e= sum((k), ro(r)*Y(j,k,t)) ;
kisit4(j,t) .. sum((k), Y(j,k,t)) =l= b(j,t)*w(j,t) ;
kisit5(t) .. sum((j), w(j,t)) =l= PU(t) ;
kisit6(k,t) .. sum((j), Y(j,k,t)) - sum((j), alfa(j,t)*Y(j,k,t)) =g= sum((l), Z(k,l,t)) ;
kisit7(k,l,t) .. Z(k,l,t) =g= (1/6)*(4*d(l,t) + (AS(l,t) + (d(l,t)-AS(l,t))/2) + (d(l,t) +
(US(l,t) - d(l,t))/2))*v(k,l,t) ;
kisit8(k,t) .. sum((l), Z(k,l,t)) =l= c(k,t)*zz(k,t) ;
kisit9(t) .. sum((k), zz(k,t)) =l= PW(t) ;
kisit10(l) .. sum((k,t),v(k,l,t)) =e= 1 ;
epq .. quantity =e= sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))) ;
yoksatma .. backlog =e= sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))) ;
Y.l(j,k,t) = 100 ;
quantity.l = 1 ;
backlog.l = 1 ;
v.l(k,l,t) = 1 ;
Z.l(k,l,t) = 1 ;
Model TZY /all/ ;
Solve TZY using minlp maximizing lambda ;
Display X.l, X.m, Y.l, Y.m, Z.l, Z.m, quantity.l, quantity.m, backlog.l, backlog.m ;
file outfile / C:\test_problemleri\sonuc3_3_3_5_3.txt/ ;
put outfile;
put lambda.l:0:6 ;
put TZY.modelstat;

```

MODELİN GAMS/DICOPT PROGRAM ÇIKTILARI

```

--- Job prob3_3_3_5_3_3.gms Start 10/07/12 17:15:59
GAMS Rev 146 Copyright (C) 1987-2006 GAMS Development. All rights reserved
*** License file too old for this version of GAMS.
*** Maintenance expired 44 days ago.
Licensee: Huseyin SARIKAYA G051019:1151AP-WIN
        Bilkent Univeristy, Dept. of Industrial Engineering DC5538
        License for teaching and research at degree granting institutions
--- Starting compilation
--- prob3_3_3_5_3_3.gms(292) 3 Mb
--- Starting execution
--- prob3_3_3_5_3_3.gms(283) 4 Mb
--- Generating MINLP model TZY
--- prob3_3_3_5_3_3.gms(285) 6 Mb
--- 125 rows 221 columns 1,140 non-zeroes
--- 16,267 nl-code 288 nl-non-zeroes
--- 63 discrete-columns
--- prob3_3_3_5_3_3.gms(285) 4 Mb
--- Executing DICOPT

```

Dicopt2x-C Nov 27, 2006 WIN.DI.DI 22.3 042.035.041.VIS

```

--- DICOPT: Starting major iteration 1
--- DICOPT: Setting up first (relaxed) NLP.
C O N O P T 3 Nov 27, 2006 WIN.CO.CO 22.3 017.058.041.VIS Library 314Q

```

C O N O P T 3 x86/MS Windows version 3.14Q-017-058
 Copyright (C) ARKI Consulting and Development A/S
 Bagsvaerdvej 246 A
 DK-2880 Bagsvaerd, Denmark

Using default options.

Reading Data

Iter	Phase	Ninf	Infeasibility	RGmax	NSB	Step	InItr	MX	OK
0	0		6.2097738558E+04 (Input point)						
			Pre-triangular equations:		0				
			Post-triangular equations:		4				
1	0		3.5098850000E+04 (After pre-processing)						
2	0		2.5034414062E+02 (After scaling)						
10	0	26	2.4988517873E+02		0.0E+00		T	T	
20	0	26	1.9044111055E+02		0.0E+00		T	T	
30	0	26	1.6977572260E+02		1.0E-02		T	T	
40	0	26	1.3688920753E+02		3.6E-02		T	T	
50	0	26	9.4290368317E+01		9.6E-02		T	T	

```

Iter Phase Ninf Infeasibility RGmax NSB Step InItr MX OK
56 1 25 4.9512929847E+01 1.5E+01 24 1.0E+00 3 T T

```

** Feasible solution. Value of objective = -4.62949060738

```

Iter Phase Ninf Objective RGmax NSB Step InItr MX OK
61 3 9.1578874363E-01 2.7E-02 6 1.0E+00 12 T T
63 3 9.1578874346E-01 0.0E+00 6

```

** Optimal solution. Reduced gradient less than tolerance.

--- DICOPT: Setting up first MIP

GAMS/Cplex Nov 27, 2006 WIN.CP.CP 22.3 032.035.041.VIS For Cplex 10.1
Cplex 10.1.0, GAMS Link 32
Cplex licensed for 1 use of lp, qp, mip and barrier, with 2 parallel threads.

Reading data...
Starting Cplex...
Unable to load names.
Tried aggregator 1 time.
MIP Presolve eliminated 5 rows and 11 columns.
MIP Presolve modified 279 coefficients.
Aggregator did 45 substitutions.
Reduced MIP has 76 rows, 173 columns, and 619 nonzeros.
Presolve time = 0.00 sec.
Clique table members: 35.
MIP emphasis: balance optimality and feasibility.
Tried aggregator 1 time.
LP Presolve eliminated 0 rows and 1 columns.
Reduced LP has 76 rows, 172 columns, and 618 nonzeros.
Presolve time = 0.00 sec.
Initializing dual steep norms . . .

Iteration log . . .
Iteration: 1 Dual objective = 2.347423
Root relaxation solution time = 0.00 sec.

Nodes				Cuts/			
Node	Left	Objective	IInf	Best Integer	Best Node	ItCnt	Gap
0	0	0.9129	8		0.9129	22	
*	0+	0	0	0.8800	0.9129	22	3.74%
		0.8855	7	0.8800	Cuts: 28	32	0.63%

GUB cover cuts applied: 1
Clique cuts applied: 3
Cover cuts applied: 2
Implied bound cuts applied: 1
Flow cuts applied: 1

```

Fixing integer variables, and solving final LP...
Tried aggregator 1 time.
LP Presolve eliminated 120 rows and 216 columns.
Reduced LP has 6 rows, 13 columns, and 38 nonzeros.
Presolve time = 0.00 sec.
Initializing dual steep norms . . .

```

Iteration log . . .

Iteration:	1	Scaled dual infeas =	0.000176
Iteration:	4	Dual objective =	1.536975

Proven optimal solution.

MIP Solution: 0.879972 (37 iterations, 0 nodes)
Final Solve: 0.879972 (5 iterations)

Best possible:	0.879972
Absolute gap:	0.000000
Relative gap:	0.000000

--- DICOPT: Starting major iteration 2

--- DICOPT: Setting up next NLP

C O N O P T 3 Nov 27, 2006 WIN.CO.CO 22.3 017.058.041.VIS Library 314Q

C O N O P T 3 x86/MS Windows version 3.14Q-017-058
Copyright (C) ARKI Consulting and Development A/S
Bagsvaerdvej 246 A
DK-2880 Bagsvaerd, Denmark

Using default options.

Reading Data

Iter	Phase	Ninf	Infeasibility	RGmax	NSB	Step	InItr	MX	OK
0	0		5.1655082661E-02	(Input point)					
			Pre-triangular equations:			11			
			Post-triangular equations:			44			
1	0		3.3306690739E-16	(After pre-processing)					
2	0		1.6653345369E-16	(After scaling)					

```
** Feasible solution. Value of objective = 0.879971693685
```

Iter	Phase	Ninf	Objective	RGmax	NSB	Step	InItr	MX	OK
4	3		8.7997169369E-01	0.0E+00	0				

**** Optimal solution.** There are no superbasic variables.

--- DICOPT: Checking convergence

--- DICOPT: Setting up next MIP

--- DICOPT: Log File:

Major Step	Major Iter	Objective Function	CPU time (Sec)	Iterations	Evaluation Errors	Solver
NLP	1	0.91579	0.03	63	0	conopt
MIP	1	0.87997	0.06	42	0	cplex
NLP	2	0.87997<	0.04	4	0	conopt

GAMS/Cplex Nov 27, 2006 WIN.CP.CP 22.3 032.035.041.VIS For Cplex 10.1
 Cplex 10.1.0, GAMS Link 32
 Cplex licensed for 1 use of lp, qp, mip and barrier, with 2 parallel threads.

Reading data...
 Starting Cplex...
 Unable to load names.
 Tried aggregator 1 time.
 MIP Presolve eliminated 9 rows and 18 columns.
 MIP Presolve modified 303 coefficients.
 Aggregator did 45 substitutions.
 Reduced MIP has 77 rows, 173 columns, and 643 nonzeros.
 Presolve time = 0.01 sec.
 Clique table members: 35.
 MIP emphasis: balance optimality and feasibility.
 Tried aggregator 1 time.
 LP Presolve eliminated 0 rows and 1 columns.
 Reduced LP has 77 rows, 172 columns, and 642 nonzeros.
 Presolve time = 0.00 sec.
 Initializing dual steep norms . . .

Iteration log . . .
 Iteration: 1 Dual objective = 2.095410
 Root relaxation solution time = 0.00 sec.

Nodes		Objective		Cuts/		ItCnt	Gap
Node	Left	IInf	Best Integer	Best Node			
0	0	0.9129	8	0.9129	27		
* 0+	0		0	0.8731	0.9129	27	4.56%
		0.8855	9	0.8731	Cuts: 28	40	1.41%
		0.8776	7	0.8731	Cuts: 12	56	0.51%
* 0+	0		0	0.8754	0.8776	56	0.26%

Clique cuts applied: 4
 Cover cuts applied: 3
 Implied bound cuts applied: 3
 Flow cuts applied: 1
 Gomory fractional cuts applied: 2
 Fixing integer variables, and solving final LP...
 Tried aggregator 1 time.
 LP Presolve eliminated 125 rows and 223 columns.
 Reduced LP has 6 rows, 13 columns, and 38 nonzeros.

Presolve time = 0.00 sec.
 Initializing dual steep norms . . .

Iteration log . . .

Iteration: 1 Scaled dual infeas = 0.000176
 Iteration: 4 Dual objective = 1.532355

Proven optimal solution.

MIP Solution: 0.875352 (56 iterations, 0 nodes)
 Final Solve: 0.875352 (5 iterations)

Best possible: 0.875352
 Absolute gap: 0.000000
 Relative gap: 0.000000

--- DICOPT: Starting major iteration 3

--- DICOPT: Setting up next NLP

C O N O P T 3 Nov 27, 2006 WIN.CO.CO 22.3 017.058.041.VIS Library 314Q

C O N O P T 3 x86/MS Windows version 3.14Q-017-058

Copyright (C) ARKI Consulting and Development A/S

Bagsvaerdvej 246 A

DK-2880 Bagsvaerd, Denmark

Using default options.

Reading Data

Iter	Phase	Ninf	Infeasibility	RGmax	NSB	Step	InItr	MX	OK
0	0		1.9015300087E-02 (Input point)						
			Pre-triangular equations:			11			
			Post-triangular equations:			44			
1	0		4.4408920985E-16 (After pre-processing)						
2	0		2.2204460493E-16 (After scaling)						

** Feasible solution. Value of objective = 0.875351678083

Iter	Phase	Ninf	Objective	RGmax	NSB	Step	InItr	MX	OK
4	3		8.7535167808E-01	0.0E+00	0				

** Optimal solution. There are no superbasic variables.

--- DICOPT: Checking convergence

--- DICOPT: Search stopped on worsening of NLP subproblems

--- DICOPT: Log File:

Major Step	Major Iter	Objective Function	CPU time (Sec)	Iterations	Evaluation Errors	Solver
NLP	1	0.91579	0.03	63	0	conopt

MIP	1	0.87997	0.06	42	0	cplex
NLP	2	0.87997<	0.04	4	0	conopt
MIP	2	0.87535	0.08	61	0	cplex
NLP	3	0.87535	0.04	4	0	conopt

--- DICOPT: Terminating...

--- DICOPT: Stopped on NLP worsening

The search was stopped because the objective function
of the NLP subproblems started to deteriorate.

--- DICOPT: Best integer solution found: 0.879972

--- Restarting execution

--- prob3_3_3_5_3_3.gms(285) 0 Mb

--- Reading solution for model TZY

--- Executing after solve

--- prob3_3_3_5_3_3.gms(292) 3 Mb

--- Putfile opt C:\Users\HAS\Documents\gamssdir\projdir\cplex.opt

--- Putfile outfile C:\test_problemleri\sonuc3_3_3_5_3.txt

*** Status: Normal completion

--- Job prob3_3_3_5_3_3.gms Stop 10/07/12 17:16:00 elapsed 0:00:00.759

TEST PROBLEMİ ÜRETME C++ KODLARI

```
//-----
#include <vcl.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <vector.h>
#include <fstream.h>
#include <string.h>
#pragma hdrstop
//-----

#pragma argsused
int main(int argc, char* argv[])
{
    int i,j,k,amac;
    int ted,ure,dag,mus,ham,don;
    double Z01, Z02, Z03, Z04, u1, u2;
    vector<int> ro, lambda, PU, PW;
    vector<vector<vector<int>>> a;
    vector<vector<int>> b, c, d;
    vector<vector<double>> fu, fw, d1, d2, d3, Cc, Co, Ch, Cb, alfa;
    vector<double> price;
    ofstream in;
    ifstream out;
    out.precision(4);
    int don1[3]={3,5,7}, mus1[3]={5,7,10}, dag1[3]={3,4,5}, ure1[3]={3,4,5},
    ted1[3]={3,4,5};
    ///1.temel parametrelerin üretilmesi
    ham=random(2)+2;
    ///1.bitti
    for(don=0;don<3;don++)
    {
        for(mus=0;mus<3;mus++)
        {
            for(dag=0;dag<3;dag++)
            {
                for(ure=0;ure<3;ure++)
                {
                    for(ted=0;ted<3;ted++)
                    {
                        a.resize(ted1[ted]);
                        for(i=0;i<ted1[ted];i++)
                        {
                            a[i].resize(ham);
                            for(j=0;j<ham;j++)
                            {
                                a[i][j].resize(don1[don]);
                                for(k=0;k<don1[don];k++)
```

```

        {
            a[i][j][k]=random(1500)+4500;
        }
    }
}
b.resize(ure1[ure]);
for(i=0;i<ure1[ure];i++)
{
    b[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        b[i][j]=random(500)+2500;
    }
}
c.resize(dag1[dag]);
for(i=0;i<dag1[dag];i++)
{
    c[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        c[i][j]=random(200)+200;
    }
}
fu.resize(ure1[ure]);
for(i=0;i<ure1[ure];i++)
{
    fu[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        fu[i][j]=(random(100)/100.00)*250+700;
    }
}
fw.resize(dag1[dag]);
for(i=0;i<dag1[dag];i++)
{
    fw[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        fw[i][j]=(random(100)/100.00)*140+190;
    }
}
}
d.resize(mus1[mus]);
for(i=0;i<mus1[mus];i++)
{
    d[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        d[i][j]=random(50)+100;
    }
}
}

```

```

ro.resize(ham);
for(i=0;i<ham;i++) ro[i]=random(2)+2;
d1.resize(ted1[tet]);
for(i=0;i<ted1[tet];i++)
{
    d1[i].resize(ure1[ure]);
    for(j=0;j<ure1[ure];j++)
    {
        d1[i][j]=(random(100)/100.00)*280+100;
    }
}
d2.resize(ure1[ure]);
alfa.resize(ure1[ure]);
for(i=0;i<ure1[ure];i++)
{
    d2[i].resize(dag1[dag]);
    for(j=0;j<dag1[dag];j++)
    {
        d2[i][j]=(random(100)/100.00)*150+70;
    }
    alfa[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {
        alfa[i][j]=((random(4)+1)/10.0);
    }
}
d3.resize(dag1[dag]);
for(i=0;i<dag1[dag];i++)
{
    d3[i].resize(mus1[mus]);
    for(j=0;j<mus1[mus];j++)
    {
        d3[i][j]=(random(100)/100.00)*50+10;
    }
}
Cc.resize(dag1[dag]);
Co.resize(dag1[dag]);
Ch.resize(dag1[dag]);
Cb.resize(dag1[dag]);
price.resize(don1[don]);
PU.resize(don1[don]);
PW.resize(don1[don]);
for(i=0;i<dag1[dag];i++)
{
    Cc[i].resize(don1[don]);
    Co[i].resize(don1[don]);
    Ch[i].resize(don1[don]);
    Cb[i].resize(don1[don]);
    for(j=0;j<don1[don];j++)
    {

```

```

        Cc[i][j]=(random(100)/100.00)*20+50;
        Co[i][j]=(random(100)/100.00)*100+200;
        Ch[i][j]=(random(100)/100.00)*30+5;
        Cb[i][j]=(random(100)/100.00)*20+20;
    }
}
for(i=0;i<don1[don];i++)
{
    price[i]=(random(100)/100.00)+1;
    PU[i]=random(ure1[ure]-1)+1;
    PW[i]=random(dag1[dag]-1)+1;
}

    u1=(random(1)+8)/10.00;
    u2=(random(1)+11)/10.00;
for(amac=0;amac<3;amac++)
{
    if(amac==2)
    {
        out.open("C:\\test_problemleri\\amac1.txt");
        out>>Z01>>Z03;
        out.close();
        out.open("C:\\test_problemleri\\amac2.txt");
        out>>Z02>>Z04;
        out.close();
        if(Z01>Z02) swap(Z01,Z02);
        if(Z03>Z04) swap(Z03,Z04);
        /*out.open("C:\\test_problemleri\\amac3.txt");
        out>>Z03;
        out.close();
        out.open("C:\\test_problemleri\\amac4.txt");
        out>>Z04;
        out.close();*/
        //out.open("C:\\test_problemleri\\amac5.txt");
        //out>>Z12;
        //out.close();
        //out.open("C:\\test_problemleri\\amac6.txt");
        //out>>Z13;
        //out.close();
    }
}
///2.yazdırılacak dosyanın tanımlanması
AnsiString str;
str="C:\\test_problemleri\\prob"+IntToStr(ted1[ted])+"_ ";
str=str+IntToStr(ure1[ure])+"_ ";
str=str+IntToStr(dag1[dag])+"_ ";
str=str+IntToStr(mus1[mus])+"_ ";
str=str+IntToStr(don1[don])+"_ ";
str=str+IntToStr(amac+1);
str=str+".gms";
in.open(str.c_str());

///2.bitti

```

////4.GAMS dosyasının yazdırılması

```

in<<"$Title Model (EPQ)"<<"\n";
in<<"option iterlim = 100000 ;"<<"\n";
in<<"option minlp = dicopt, nlp = conopt, mip = cplex;"<<"\n";
in<<"file opt cplex option file /cplex.opt/;"<<"\n";
in<<"put opt;"<<"\n";
in<<"put 'preind 0';"<<"\n";
in<<"putclose opt;"<<"\n";
in<<"option SYSOUT = ON ;"<<"\n"<<"\n";

in<<"sets"<<"\n";
in<<"i tedarik merkezleri /";
for(i=0;i<ted1[ted]-1;i++) in<<"sup"<<i+1<<" ";
in<<"sup"<<ted1[ted]<<"/"<<"\n";
in<<"j üretim tesisleri /";
for(i=0;i<ure1[ure]-1;i++) in<<"plant"<<i+1<<" ";
in<<"plant"<<ure1[ure]<<"/"<<"\n";
in<<"k dağıtım merkezleri /";
for(i=0;i<dag1[dag]-1;i++) in<<"warehouse"<<i+1<<" ";
in<<"warehouse"<<dag1[dag]<<"/"<<"\n";
in<<"l müşteriler /1*"<<mus1[mus]<<"/"<<"\n";
in<<"r hammaddeler /";
for(i=0;i<ham-1;i++) in<<"raw"<<i+1<<" ";
in<<"raw"<<ham<<"/"<<"\n";
in<<"t dönem sayısı /1*"<<don1[don]<<"/;"<<"\n";
in<<"Parameter a(i,r,t) i tedarik merkezinin r hammadde
kapasitesi (1000 ton)"<<"\n";
in<<"/"<<"\n";
for(i=0;i<ted1[ted];i++)
    for(j=0;j<ham;j++)
        for(k=0;k<don1[don];k++)
in<<"sup"<<i+1<<". "<<"raw"<<j+1<<". "<<k+1<<"t"<<a[i][j][k]<<"\n";
in<<"/;"<<"\n";
in<<"Parameter b(j,t) j üretim tesisinin ürün kapasitesi (1000 ton
cinsinden)"<<"\n";
in<<"/"<<"\n";
for(i=0;i<ure1[ure];i++)
    for(j=0;j<don1[don];j++)
        in<<"plant"<<i+1<<". "<<j+1<<"t"<<b[i][j]<<"\n";
in<<"/;"<<"\n";
in<<"Parameter c(k,t) k dağıtım merkezinin ürün kapasitesi
(1000 ton)"<<"\n";
in<<"/"<<"\n";
for(i=0;i<dag1[dag];i++)
    for(j=0;j<don1[don];j++)
        in<<"warehouse"<<i+1<<". "<<j+1<<"t"<<c[i][j]<<"\n";
in<<"/;"<<"\n";
in<<"Parameter fu(j,t) j üretim tesisinin sabit maliyetleri (acma
ve işletme)"<<"\n";
in<<"/"<<"\n";

```

```

for(i=0;i<ure1[ure];i++)
    for(j=0;j<don1[don];j++)
        in<<"plant"<<i+1<<"."<<j+1<<"t"<<fu[i][j]<<"n';
in<<"/;"<<"n';
in<<"Parameter fw(k,t) k dagitim merkezinin sabit maliyetleri
(acma ve isletme)"<<"n';
in<<"/"<<"n';
for(i=0;i<dag1[dag];i++)
    for(j=0;j<don1[don];j++)
in<<"warehouse"<<i+1<<"."<<j+1<<"t"<<fw[i][j]<<"n';
in<<"/;"<<"n';
in<<"Parameter d(l,t) t. donemde l musterisinin urun
talebi"<<"n';
in<<"/"<<"n';
for(i=0;i<mus1[mus];i++)
    for(j=0;j<don1[don];j++)
        in<<i+1<<"."<<j+1<<"t"<<d[i][j]<<"n';
in<<"/;"<<"n';
in<<"Parameter AS(l,t) t. donemde l musterisinin urun talebi alt
limit;"<<"n';
in<<"AS(l,t) = d(l,t)*"<<u1<<"n';
in<<","<<"n';
in<<"Parameter US(l,t) t. donemde l musterisinin
urun talebi ust limit;"<<"n';
in<<"US(l,t) = d(l,t)*"<<u2<<"n';
in<<","<<"n';
in<<"Parameter ro(r) Bir birim bitmis urun uretmek icin gerekli
olan r hammaddesi miktarı"<<"n';
in<<"/"<<"n';
for(i=0;i<ham;i++)
    in<<"raw"<<i+1<<"t"<<ro[i]<<"n';
in<<"/;"<<"n';
in<<"Parameter alfa(j,t) Fire oranı"<<"n';
in<<"/"<<"n';
for(j=0;j<ure1[ure];j++)
    for(i=0;i<don1[don];i++)
        in<<"plant"<<j+1<<"."<<i+1<<"t"<<alfa[j][i]<<"n';
in<<"/;"<<"n';
in<<"Parameter d1(i,j) i tedarik merkezi ile j uretim tesisi
arasındaki mesafe (km.)"<<"n';
in<<"/"<<"n';
for(i=0;i<ted1[ted];i++)
    for(j=0;j<ure1[ure];j++)
in<<"sup"<<i+1<<"."<<"plant"<<j+1<<"t"<<d1[i][j]<<"n';
in<<"/;"<<"n';
in<<"Parameter d2(j,k) j uretim tesisi ile k dagitim merkezi
arasındaki mesafe (km.)"<<"n';
in<<"/"<<"n';
for(i=0;i<ure1[ure];i++)

```

```

        for(j=0;j<dag1[dag];j++)
in<<"plant"<<i+1<<". "<<"warehouse"<<j+1<<"t"<<d2[i][j]<<"\n";
        in<<"/;"<<"\n";
        in<<"Parameter d3(k,l) k dagitim merkezi ile l musterileri
arasindaki mesafe (km.)"<<"\n";
        in<<"/;"<<"\n";
        for(i=0;i<dag1[dag];i++)
            for(j=0;j<mus1[mus];j++)
in<<"warehouse"<<i+1<<". "<<j+1<<"t"<<d3[i][j]<<"\n";
        in<<"/;"<<"\n";

        in<<"Parameter Cc(k,t) Birim satinalma maliyeti (TL) /";
        for(j=0;j<dag1[dag];j++)
            for(i=0;i<don1[don];i++)
in<<"warehouse"<<j+1<<". "<<i+1<<"t"<<Cc[j][i]<<"\n";
        in<<"/;"<<"\n";
        in<<"Parameter Co(k,t) Siparis verme maliyeti (TL cinsinden) /";
        for(j=0;j<dag1[dag];j++)
            for(i=0;i<don1[don]-1;i++)
in<<"warehouse"<<j+1<<". "<<i+1<<"t"<<Co[j][i]<<"\n";
        in<<"/;"<<"\n";
        in<<"Parameter Ch(k,t) k dagitim merkezindeki yillik stokta
tutma maliyeti /";
        for(j=0;j<dag1[dag];j++)
            for(i=0;i<don1[don]-1;i++)

in<<"warehouse"<<j+1<<". "<<i+1<<"t"<<Ch[j][i]<<"\n";
        in<<"/;"<<"\n";

        in<<"Parameter Cb(k,t) k dagitim merkezindeki yillik yoksatma
maliyeti /";
        for(j=0;j<dag1[dag];j++)
            for(i=0;i<don1[don]-1;i++)

in<<"warehouse"<<j+1<<". "<<i+1<<"t"<<Cb[j][i]<<"\n";
        in<<"/;"<<"\n";

        in<<"Scalar f 100 km. basina harcanan yakit (litre cinsinden)
/5.1/ ;"<<"\n";

        in<<"Parameter C1(i,j) i tedarik merkezinden j uretim tesisine
gonderilen urunun birim tasima maliyeti ;"<<"\n";
        in<<"C1(i,j) = f * d1(i,j) / 100 ;"<<"\n";

        in<<"Parameter C2(j,k) j uretim tesisinden k dagitim merkezine
gonderilen urunun birim tasima maliyeti ;"<<"\n";
        in<<"C2(j,k) = f * d2(j,k) / 100 ;"<<"\n";

        in<<"Parameter C3(k,l) k dagitim merkezinden l musterisine
gonderilen urunun birim tasima maliyeti ;"<<"\n";

```

```

in<<"C3(k,l) = f * d3(k,l) / 100 ;"<<"\n";

fiyati"<<"\n";
in<<"Parameter price(t) Harcanan 1 lt. akaryakitın donem basına
in<<"/"<<"\n";
for(i=0;i<don1[don];i++)
    in<<i+1<<"t"<<price[i]<<"\n";
in<<"/;"<<"\n";

in<<"Variables"<<"\n";
in<<"X(i,j,r,t) j üretim tesisinde i tedarik merkezinden r
hammaddesi kullanılarak üretilen miktar"<<"\n";
in<<"Y(j,k,t) j üretim tesisinden k dağıtım merkezine gönderilen
p ürün miktarı"<<"\n";
in<<"Z(k,l,t) k dağıtım merkezinden l müşterisine gönderilen
miktar"<<"\n";
in<<"v(k,l,t) k dağıtım merkezinden l müşterisine p ürün
gönderilirse"<<"\n";
in<<"w(j,t) j üretim tesisi açık ise"<<"\n";
in<<"zz(k,t) k dağıtım merkezi açık ise"<<"\n";
in<<"quantity Ekonomik Üretim Miktarı"<<"\n";
in<<"backlog Yoksatma miktarı ;"<<"\n";

if(amac==2) in<<"Free Variables lambda ;"<<"\n";
in<<"Free Variable TM, SM;"<<"\n";
in<<"Positive Variables X, Y, Z, quantity, backlog ;"<<"\n";
in<<"Binary Variables v, w, zz ;"<<"\n";

in<<"Equations"<<"\n";
if(amac==0)
{
    in<<"objective1 amac fonksiyonu 1"<<"\n";
    in<<"kisitobj2"<<"\n";
}
else if(amac==1)
{
    in<<"objective2 amac fonksiyonu 2"<<"\n";
    in<<"kisitobj1"<<"\n";
}
else
{
    in<<"objective1 amac fonksiyonu 1"<<"\n";
    in<<"objective2 amac fonksiyonu 2"<<"\n";
    in<<"kisitobj1 amac kısıtı 1"<<"\n";
    in<<"kisitobj2 amac kısıtı 2"<<"\n";
}
in<<"kisit2(i,r,t) Tedarik merkezlerinden üretim tesislerine
gönderilen hammaddelerin tedarik merkezinin kapasitesini asmamasını sağlar"<<"\n";

```

in<<"kisit3(j,t) Tedarik merkezlerinden üretim tesislerine gönderilen hammadelerin en az ürünü üretmek için gerekli olan hammadde kadar olmasını sağlar"<<"n';

in<<"kisit4(j,t) Üretim tesislerinden dağıtım merkezlerine gönderilen ürünlerin açık olan tesislerin kapasitesinden fazla olmamasını sağlar"<<"n';

in<<"kisit5(t) Acilabilecek dağıtım merkezi sayısının üst sınırı"<<"n';

in<<"kisit6(k,t) Dağıtım merkezi akis denge kisiti"<<"n';

in<<"kisit7(k,l,t) Ürünler için müşteri taleplerinin dağıtım merkezlerinden gönderilen toplam ürün miktarından fazla olmamasını sağlar"<<"n';

in<<"kisit8(k,t) Müşteri noktasındaki toplam ürün miktarının dağıtım merkezlerinin toplam ürün kapasitesini aşmayacağını garanti eder"<<"n';

in<<"kisit9(t) Acilabilecek müşteri noktası sayısının üst sınırı"<<"n';

in<<"kisit10(l) Her müşterinin sadece bir dağıtım merkezinden hizmet almasını sağlar "<<"n';

in<<"epq Ekonomik Üretim Miktarı"<<"n';

in<<"yoksatma Yoksatılan miktar ;"<<"n'<<"n';

if(amac==0)

{

in<<"objective1 .. TM =e= sum((i,j,r,t), C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t), C2(j,k)*price(t)*Y(j,k,t)) +sum((k,l,t), C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t), fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t)) ;"<<"n'<<"n';

in<<"kisitobj2 .. SM =e= sum((k,l,t), Cc(k,t)*Z(k,l,t)) + sum((k,l,t), Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))+(sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqr[sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))*sqr(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/"<<"n';

in<<"sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))*sqr(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/"<<"n';

in<<"(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))/"<<"n';

}

else if(amac==1)

{

in<<"objective2 .. SM =e= sum((k,l,t), Cc(k,t)*Z(k,l,t)) + sum((k,l,t), Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),

```

Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))) + "<<"n';
in<<"((sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqr[sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t))) - "<<"n';
in<<"sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))*((1/(sum((j,k,t),
Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+ ((sum((k,l,t),
Cb(k,t)*Z(k,l,t))*sqr(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/"<<"n';
in<<"(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))
;"<<"n'<<"n';
in<<"kisitobj1 .. TM =e= sum((i,j,r,t),
C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t), C2(j,k)*price(t)*Y(j,k,t)) +sum((k,l,t),
C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t), fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t))
;"<<"n';
}
else
{
in<<"objective1 .. lambda =l= ("<<Z02<<" - (sum((i,j,r,t),
C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t), C2(j,k)*price(t)*Y(j,k,t)) +sum((k,l,t),
C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t), fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t))) /
("<<Z02<<" - "<<Z01<<") ;"<<"n'<<"n';
in<<"objective2 .. lambda =l= ("<<Z04<<" - (sum((k,l,t),
Cc(k,t)*Z(k,l,t)) + sum((k,l,t), Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))) + "<<"n';
in<<"((sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqr[sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t))) - "<<"n';
in<<"sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))*((1/(sum((j,k,t),
Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+ ((sum((k,l,t),
Cb(k,t)*Z(k,l,t))*sqr(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/"<<"n';
in<<"(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t)))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t),
Z(k,l,t)))) / ("<<Z04<<" - "<<Z03<<") ;"<<"n'<<"n';

```

in<<"kisitobj1 .. TM =e= sum((i,j,r,t),
C1(i,j)*price(t)*X(i,j,r,t)) + sum((j,k,t), C2(j,k)*price(t)*Y(j,k,t)) +sum((k,l,t),
C3(k,l)*price(t)*Z(k,l,t)) + sum((j,t), fu(j,t)*w(j,t)) + sum((k,t), fw(k,t)*zz(k,t))
;"<<"n';

in<<"kisitobj2 .. SM =e= sum((k,l,t), Cc(k,t)*Z(k,l,t)) +
sum((k,l,t), Co(k,t)*Z(k,l,t))/sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))) +"<<"n';

in<<"((sum((k,l,t), Ch(k,t)*Z(k,l,t)))/(2*sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*sqrt(sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t))) -"<<"n';

in<<"sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))*((1/(sum((j,k,t),
Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))+(sum((k,l,t),
Cb(k,t)*Z(k,l,t))*sqrt(sqrt(2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(1-sum((k,l,t),
Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t), Cb(k,t)*(Ch(k,t)+Cb(k,t)))))))/"<<"n';

in<<"(2*sqrt((2*sum((k,l,t), Co(k,t)*Z(k,l,t))*(sum((k,t),
Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t),
Y(j,k,t)))))))*((1/(sum((j,k,t), Y(j,k,t))-sum((k,l,t), Z(k,l,t)))+1/sum((k,l,t), Z(k,l,t)))
;"<<"n';

}

in<<"kisit2(i,r,t) .. sum((j), X(i,j,r,t)) =l= a(i,r,t) ;"<<"n';

in<<"kisit3(j,t) .. sum((i,r), X(i,j,r,t)) =e= sum((k,r),
ro(r)*Y(j,k,t)) ;"<<"n';

in<<"kisit4(j,t) .. sum((k), Y(j,k,t)) =l= b(j,t)*w(j,t) ;"<<"n';

in<<"kisit5(t) .. sum((j), w(j,t)) =l="<<ure1[ure]-1<<" ;"<<"n';

in<<"kisit6(k,t) .. sum((j), Y(j,k,t)) - sum((j), alfa(j,t)*Y(j,k,t))
=g= sum((l), Z(k,l,t)) ;"<<"n';

in<<"kisit7(k,l,t).. Z(k,l,t) =g= (1/6)*(4*d(l,t) + (AS(l,t) +
(d(l,t)-AS(l,t))/2) + (d(l,t) + (US(l,t) - d(l,t))/2))*v(k,l,t) ;"<<"n';

in<<"kisit8(k,t).. sum((l), Z(k,l,t))=l= c(k,t)*zz(k,t) ;"<<"n';

in<<"kisit9(t) .. sum((k), zz(k,t)) =l="<<dag1[dag]-1<<"
;"<<"n';

in<<"kisit10(l) .. sum((k,t),v(k,l,t)) =e= 1 ;"<<"n';

in<<"epq .. quantity =e= sqrt((2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(sum((k,t), Ch(k,t)+Cb(k,t))))/(sum((k,t), Ch(k,t)*Cb(k,t))*(1-
sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))))) ;"<<"n';

in<<"yoksatma .. backlog =e= sqrt(2*sum((k,l,t),
Co(k,t)*Z(k,l,t))*(1-sum((k,l,t), Z(k,l,t))/sum((j,k,t), Y(j,k,t)))/(sum((k,t),
Cb(k,t)*(Ch(k,t)+Cb(k,t)))))) ;"<<"n'<<"n';;

in<<"Y.l(j,k,t) = 10 ;"<<"n';

in<<"quantity.l = 1 ;"<<"n';

in<<"backlog.l = 1 ;"<<"n';

```

in<<"v.l(k,l,t) = 1 ;"<<"\n";
in<<"Z.l(k,l,t) = 1 ;"<<"\n";

in<<"Model TZY /all/ ;"<<"\n";
if(amac==0) in<<"Solve TZY using minlp minimizing TM
;"<<"\n";

else if(amac==1) in<<"Solve TZY using minlp minimizing SM
;"<<"\n";

else in<<"Solve TZY using minlp maximizing lambda ;"<<"\n";
in<<"Display X.l, X.m, Y.l, Y.m, Z.l, Z.m, quantity.l,
quantity.m, backlog.l, backlog.m ;"<<"\n";
if(amac==0) in<<"file outfile / C:\\test_problemleri\\amac1.txt/
;"<<"\n";

else if(amac==1) in<<"file outfile /
C:\\test_problemleri\\amac2.txt/ ;"<<"\n";
else in<<"file outfile /
C:\\test_problemleri\\sonuc"<<ted1[ted]<<"_"<<ure1[ure]<<"_"<<dag1[dag]<<"_"<<
<mus1[mus]<<"_"<<don1[don]<<".txt/ ;"<<"\n";
in<<"put outfile;"<<"\n";
if(amac!=2)
{
in<<"put TM.l:0:5 /"<<"\n";
in<<"put SM.l:0:5 ;"<<"\n";
}
else
{
in<<"put lambda.l:0:6 ;"<<"\n";
in<<"put TM.l:0:5 /"<<"\n";
in<<"put SM.l:0:5 ;"<<"\n";
}

in.close();
/////4.bitti

/////5.GAMS çalıştırılması

str="C:\\Users\\HAS\\Desktop\\TEZ_CALISMASI\\Programlar\\gams22.3\\gams.exe
"+str;
//str=str+;
system(str.c_str());
/////5.bitti
}
}
}
}
}
}
return 0;
}
//-----

```

TAVLAMA BENZETİMİ ALGORİTMASI TABANLI SEZGİSELİN C++ KODLARI

```
//-----

#include <vc1.h>
#pragma hdrstop
#pragma warning( disable: W8012)
#include "odev2.h"

//-----

#pragma package(smart_init)
#pragma resource "*.dfm"

TForm1 *Form1;
int i,j,k,k1,tcd,ut,dm,m,h,per,q,iter;
double
alt,ust,f,objus1,objas1,objus2,objas2,stat,quantity,backlog,lambda,TM,SM,kap;
vector<int> ra,PU,PW;
vector<double> price,kopy;
vector<vector<int> > d,acik_dm,acik_ut;
vector<vector<double> > b,c,fu,fw,d1,d2,d3,Cc,Co,Ch,Cb,kapasiteut,kapasitedm;
vector<vector<vector<double> > > a, alfa,Z,Y,kapasiteted;
vector<vector<vector<vector<double> > > > X;
SimulatedAnnealing SA;
ofstream out1;

//-----

__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}

//-----

void SimulatedAnnealing::neighbor()
{
    int toplam;
    vector<int> fakedm, fakeut;
    acik_dm.clear();
    acik_ut.clear();
    acik_dm.resize(per);
    acik_ut.resize(per);
    for(k1=0;k1<tcd;k1++){
        for(k=0;k<ut;k++){
            for(i=0;i<dm;i++) fill(Y[k][i].begin(),Y[k][i].end(),0);
            for(i=0;i<h;i++) fill(X[k1][k][i].begin(),X[k1][k][i].end(),0);}}
    for(i=0;i<dm;i++)
    {
        for(j=0;j<m;j++) fill(Z[i][j].begin(),Z[i][j].end(),0);
    }
}
```

```

i=random(per);
k=random(ut);
SA.vector01ut[i][k]=1-SA.vector01ut[i][k];
i=random(per);
k=random(dm);
SA.vector01dm[i][k]=1-SA.vector01dm[i][k];
for(i=0;i<per;i++)
{
    for(j=0;j<dm;j++)
    {
        if(SA.vector01dm[i][j]==1)
        {
            acik_dm[i].push_back(j);
        }
    }
    for(j=0;j<ut;j++){
        if(SA.vector01ut[i][j]==1) acik_ut[i].push_back(j);
    }
}
kapasitedm=c;
kapasiteut=b;
kapasiteted=a;
int stop=0;
durak6: for(j=stop;j<m;j++)
{
    vector<int> kopy1,kopydon,kopydm,kopyut;
    vector<double> kopy2,kopy3;
    int k2,it1;
    kopy1.clear(); kopy2.clear(); kopy3.clear();
    kopy1.resize(per);
    copy(d[j].begin(),d[j].end(),kopy1.begin());
durak4:    it1=distance(kopy1.begin(),min_element(kopy1.begin(),kopy1.end()));
    fakedm.clear(); fakeut.clear();
    fakedm.resize(acik_dm[it1].size());
    fakeut.resize(acik_ut[it1].size());
    double tal, tal1, tal2;
    if(*min_element(kopy1.begin(),kopy1.end())==INT_MAX)
    {
        kopydon.resize(per); for(i=0;i<per;i++) kopydon[i]=i;
        kopydm.resize(dm); for(i=0;i<dm;i++) kopydm[i]=i;
        kopyut.resize(ut); for(i=0;i<ut;i++) kopyut[i]=i;
dur14:    it1=random(kopydon.size());
        int it2=random(kopydm.size());
        if(acik_dm[kopydon[it1]].size()!=dm) goto dur0;
        else goto dur11;
dur0:    if(SA.vector01dm[kopydon[it1]][kopydm[it2]]==1)
        {
            //kopy[it2]=INT_MAX;
            kopydm.erase(kopydm.begin()+it2);
            if(kopydm.size()==0)

```

```

        {
            kopydon.erase(kopydon.begin()+it1);
            kopydm.resize(dm);
            for(i=0;i<dm;i++) kopydm[i]=i;
            goto dur14;
        }
        it2=random(kopydm.size());
        goto dur0;
    }
    else
    {
dur11:        tal=(4*d[j][kopydon[it1]] + (d[j][kopydon[it1]]*0.9 +
(d[j][kopydon[it1]]-d[j][kopydon[it1]]*0.9)/2) + (d[j][kopydon[it1]] +
(d[j][kopydon[it1]]*1.4 - d[j][kopydon[it1]]/2))/6.000;
            if(tal<=kapasitedm[kopydm[it2]][kopydon[it1]])
            {
dur2:                vector<int> sayi;
                        k2=random(kopyut.size());
                        if(tal<kapasiteut[kopyut[k2]][kopydon[it1]])
                        {
                            SA.vector01dm[kopydon[it1]][kopydm[it2]]=1;
                            acik_dm[kopydon[it1]].push_back(kopydm[it2]);
                            SA.vector02[0][j]=kopydon[it1];
                            SA.vector02[1][j]=kopydm[it2];
                            SA.vector02[2][j]=kopyut[k2];
                            if(SA.vector01ut[kopydon[it1]][kopyut[k2]]==0)
                            {
                                SA.vector01ut[kopydon[it1]][kopyut[k2]]=1;
                                acik_ut[kopydon[it1]].push_back(kopyut[k2]);
                            }
                        }

Z[kopydm[it2]][j][kopydon[it1]]=Z[kopydm[it2]][j][kopydon[it1]]+tal;

kapasitedm[kopydm[it2]][kopydon[it1]]=kapasitedm[kopydm[it2]][kopydon[it1]]-
tal;

                        tal1=tal/(1-
alfa[kopyut[k2]][kopydm[it2]][kopydon[it1]]);

Y[kopyut[k2]][kopydm[it2]][kopydon[it1]]=Y[kopyut[k2]][kopydm[it2]][kopydon[it
1]]+tal1;

kapasiteut[kopyut[k2]][kopydon[it1]]=kapasiteut[kopyut[k2]][kopydon[it1]]-tal1;
                        int k3=random(ted);
                        double toph=0;
                        for(i=0;i<h;i++)
toph=toph+kapasiteted[k3][i][kopydon[it1]];
                        if(tal<toph){
                            SA.vector02[3][j]=k3;
                            for(i=0;i<h;i++)
                            {

```

```

        tal2=ra[i]*tal1;

X[k3][kopyut[k2]][i][kopydon[it1]]=X[k3][kopyut[k2]][i][kopydon[it1]]+tal2;

kapasiteded[k3][i][kopydon[it1]]=kapasiteded[k3][i][kopydon[it1]]-tal2;
    }
    }
    stop=j+1;
    goto durak6;
}
else
{
    kopyut.erase(kopyut.begin()+k2);
    goto dur2;
}
}
else
{
    kopydm.erase(kopydm.begin()+it2);
    if(kopydm.size()==0)
    {
        kopydon.erase(kopydon.begin()+it1);
        kopydm.resize(dm);
        for(i=0;i<dm;i++) kopydm[i]=i;
        goto dur14;
    }
    it2=random(kopydm.size());
    goto dur0;
}
}
}
if(acik_dm[it1].size()==0)
{
    kopy1[it1]=INT_MAX;
    goto durak4;
}
else
{
    fakedm=acik_dm[it1];
    kopy2.clear();
    kopy2.resize(fakedm.size());
    for(i=0;i<int(kopy2.size());i++) kopy2[i]=d3[fakedm[i]][j];
durak3:
k1=distance(kopy2.begin(),min_element(kopy2.begin(),kopy2.end()));
int kont1=fakedm[k1];
int kont2=it1;
double kontrol=kapasitedm[fakedm[k1]][it1];
tal= (4*d[j][it1] + (d[j][it1]*0.9 + (d[j][it1]-d[j][it1]*0.9)/2) +
(d[j][it1] + (d[j][it1]*1.4 - d[j][it1])/2))/6.000;
if(tal<=kapasitedm[fakedm[k1]][it1])

```

```

{
    if(acik_ut[it1].size()==0)
    {
        kopy.clear();
        kopy.resize(ut);
        for(i=0;i<ut;i++) kopy[i]=d2[i][k1];
        int
it2=distance(kopy.begin(),min_element(kopy.begin(),kopy.end()));
        acik_ut[it1].push_back(it2);
        if(tal<kapasiteut[it2][it1])
        {
            SA.vector02[0][j]=it1;
            SA.vector02[1][j]=acik_dm[it1][k1];
            SA.vector02[2][j]=it2;

Z[acik_dm[it1][k1]][j][it1]=Z[acik_dm[it1][k1]][j][it1]+tal;

kapasitedm[acik_dm[it1][k1]][it1]=kapasitedm[acik_dm[it1][k1]][it1]-tal;
            tal=tal/(1-alfa[it2][acik_dm[it1][k1]][it1]);

Y[it2][acik_dm[it1][k1]][it1]=Y[it2][acik_dm[it1][k1]][it1]+tal1;
            kapasiteut[it2][it1]=kapasiteut[it2][it1]-tal1;
        }
        k2=acik_ut[it1].size()-1;
    }
    else
    {
        fakeut=acik_ut[it1];
        kopy3.clear();
        kopy3.resize(fakeut.size());
        for(i=0;i<int(kopy3.size());i++) kopy3[i]=d2[fakeut[i]][k1];
dur:
        k2=distance(kopy3.begin(),min_element(kopy3.begin(),kopy3.end()));
        int kont3=fakeut[k2];
        double kontrol1=kapasiteut[fakeut[k2]][it1];
        if(tal<kapasiteut[fakeut[k2]][it1])
        {
            SA.vector02[0][j]=it1;
            SA.vector02[1][j]=fakedm[k1];
            SA.vector02[2][j]=fakeut[k2];
            Z[fakedm[k1]][j][it1]=Z[fakedm[k1]][j][it1]+tal;
            double kont4=Z[fakedm[k1]][j][it1];

kapasitedm[fakedm[k1]][it1]=kapasitedm[fakedm[k1]][it1]-tal;
            double kont5=kapasitedm[fakedm[k1]][it1];
            tal=tal/(1-alfa[fakeut[k2]][fakedm[k1]][it1]);

Y[fakeut[k2]][fakedm[k1]][it1]=Y[fakeut[k2]][fakedm[k1]][it1]+tal1;
            kont4=Y[fakeut[k2]][fakedm[k1]][it1];

```

```

        kapasiteut[ fakeut[k2]][it1]=kapasiteut[ fakeut[k2]][it1]-
tal1;
        kont5=kapasiteut[ fakeut[k2]][it1];
    }
    else
    {
        kopy3[k2]=INT_MAX;
        if(*min_element(kopy3.begin(),kopy3.end())==INT_MAX)
        {
            kopy1[it1]=INT_MAX;
            goto durak4;
        }
        goto dur;
    }
}
}
else
{
    kopy2[k1]=INT_MAX;
    if(*min_element(kopy2.begin(),kopy2.end())==INT_MAX)
    {
        kopy1[it1]=INT_MAX;
        goto durak4;
    }
    goto durak3;
}
}
kopy.clear();
kopy.resize(ted);
for(i=0;i<ted;i++) kopy[i]=d1[i][acik_ut[it1][k2]];
dur1:    int k3=distance(kopy.begin(),min_element(kopy.begin(),kopy.end()));
double toph=0;
for(i=0;i<h;i++) toph=toph+kapasiteted[k3][i][it1];
if(tal<toph){
    SA.vector02[3][j]=k3;
    for(i=0;i<h;i++)
    {
        tal2=ra[i]*tal1;
        X[k3][acik_ut[it1][k2]][i][it1]=X[k3][acik_ut[it1][k2]][i][it1]+tal2;
        kapasiteted[k3][i][it1]=kapasiteted[k3][i][it1]-tal2;
    }
}
else
{
    kopy[k3]=INT_MAX;
    goto dur1;
}
}
}

```

```

}
//-----
void amac()
{
    quantity=0;
    double top1=0,top2=0,top3=0,top4=0,top5=0,top6=0,top7=0,top8=0,top9=0;
    for(i=0;i<dm;i++){
        for(j=0;j<m;j++){
            for(k=0;k<per;k++){
                top1=top1+(Co[i][k]*Z[i][j][k]);
                top4=top4+Z[i][j][k];
                top7=top7+(Cc[i][k]*Z[i][j][k]);
                top8=top8+(Ch[i][k]*Z[i][j][k]);
                top9=top9+(Cb[i][k]*Z[i][j][k]);
            }
        }
    }
    for(i=0;i<dm;i++){
        for(j=0;j<per;j++){
            top2=top2+Ch[i][j]+Cb[i][j];
            top3=top3+(Ch[i][j]*Cb[i][j]);
            top6=top6+(Cb[i][j]*(Ch[i][j]*Cb[i][j]));
        }
    }
    for(i=0;i<ut;i++){
        for(j=0;j<dm;j++){
            for(k=0;k<per;k++){
                top5=top5+Y[i][j][k];
            }
        }
    }
    quantity=(2*top1*top2)/(top3*(1-(top4/top5)));
    quantity=sqrt(quantity);
    backlog=(2*top1*(1-(top4/top5))/top6;
    backlog=sqrt(backlog);
    SM=top7+(top1/quantity);
    SM=SM+((top8/(2*quantity))*pow(((quantity*(1-(top4/top5)))-
backlog),2)*((1/(top5-top4)))+(1/top4)));
    SM=SM+(((top9*pow(backlog,2))/(2*quantity))*((1/(top5-top4)))+(1/top4)));
    top1=0;
    for(i=0;i<ted;i++){
        for(j=0;j<ut;j++){
            for(k=0;k<h;k++){
                for(k1=0;k1<per;k1++){
                    top1=top1+((f*d1[i][j]/100)*price[k1]*X[i][j][k][k1]);
                }
            }
        }
    }
    top2=0;
    for(i=0;i<ut;i++){
        for(j=0;j<dm;j++){
            for(k=0;k<per;k++){
                top2=top2+((f*d2[i][j]/100)*price[k]*Y[i][j][k]);
            }
        }
    }
    top3=0;
    for(i=0;i<dm;i++){
        for(j=0;j<m;j++){
            for(k=0;k<per;k++){
                top3=top3+((f*d3[i][j]/100)*price[k]*Z[i][j][k]);
            }
        }
    }

```

```

        }}}
    top4=0;
    double kont1;
    for(i=0;i<per;i++){
        kont1=acik_ut[i].size();
        for(k=0;k<int(acik_ut[i].size());k++){
            kont1=0;
            for(j=0;j<dm;j++){
                kont1=kont1+Y[acik_ut[i][k]][j][i];
            }
            if(kont1!=0) top4=top4+(fu[acik_ut[i][k]][i]);}
    top5=0;
    for(i=0;i<per;i++){
        for(k=0;k<int(acik_dm[i].size());k++){
            kont1=0;
            for(j=0;j<m;j++){
                kont1=kont1+Z[acik_dm[i][k]][j][i];
            }
            if(kont1!=0) top5=top5+(fw[acik_dm[i][k]][i]);}
    TM=top1+top2+top3+top4+top5;
    lambda=(objus1-TM)/(objus1-objas1);
    stat=(objus2-SM)/(objus2-objas2);
}

//-----

void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    CheckListBox1->Clear();
    OpenFileDialog1->InitialDir="C:\\";
    if(OpenFileDialog1->Execute())
    {
        CheckListBox1->Items->AddStrings(OpenFileDialog1->Files);
    }
}

//-----

void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    int count=0;
    double bestTM,bestSM,Temp,time,yer;
    vector<vector<int>> yedek01ut, yedek01dm, yedek02;
    string str;
    ifstream in,in1;
    ofstream out,out1,out2;
    clock_t start,finish;
    in.precision(10);
    in1.precision(10);
    out.precision(10);
    out1.precision(10);

```

```

out2.precision(10);
out.open("C:\\Work\\HAS2v2-1\\Sasonuclar.txt");
while(count!=CheckListBox1->Items->Count)
{
    if(CheckListBox1->Checked[count])
    {
        iter=-1;
        in.clear();
        in.open(CheckListBox1->Items->Strings[count].c_str());
        in>>str>>ted;
        in>>str>>ut;
        in>>str>>dm;
        in>>str>>m;
        in>>str>>h;
        in>>str>>per;
        in>>str;
        a.resize(ted);
        for(i=0;i<ted;i++){
            a[i].resize(h);
            for(j=0;j<h;j++){
                a[i][j].resize(per);
                for(k=0;k<per;k++)
                    in>>a[i][j][k];}}
        in>>str;
        b.resize(ut);
        for(i=0;i<ut;i++){
            b[i].resize(per);
            for(j=0;j<per;j++)
                in>>b[i][j];}
        in>>str;
        c.resize(dm);
        for(i=0;i<dm;i++){
            c[i].resize(per);
            for(j=0;j<per;j++)
                in>>c[i][j];}
        in>>str;
        fu.resize(ut);
        for(i=0;i<ut;i++){
            fu[i].resize(per);
            for(j=0;j<per;j++)
                in>>fu[i][j];}
        in>>str;
        fw.resize(dm);
        for(i=0;i<dm;i++){
            fw[i].resize(per);
            for(j=0;j<per;j++)
                in>>fw[i][j];}
        in>>str;
        d.resize(m);
        for(i=0;i<m;i++){

```

```

        d[i].resize(per);
        for(j=0;j<per;j++)
            in>>d[i][j];}
in>>str>>alt;
in>>str>>ust;
in>>str;
ra.resize(h);
for(i=0;i<h;i++)
    in>>ra[i];
in>>str;
alfa.resize(ut);
for(i=0;i<ut;i++){
    alfa[i].resize(dm);
    for(j=0;j<dm;j++){
        alfa[i][j].resize(per);
        for(k=0;k<per;k++)
            in>>alfa[i][j][k];} }
in>>str;
d1.resize(ted);
for(i=0;i<ted;i++){
    d1[i].resize(ut);
    for(j=0;j<ut;j++)
        in>>d1[i][j];}
in>>str;
d2.resize(ut);
for(i=0;i<ut;i++){
    d2[i].resize(dm);
    for(j=0;j<dm;j++)
        in>>d2[i][j];}
in>>str;
d3.resize(dm);
for(i=0;i<dm;i++){
    d3[i].resize(m);
    for(j=0;j<m;j++)
        in>>d3[i][j];}
in>>str;
Cc.resize(dm);
for(i=0;i<dm;i++){
    Cc[i].resize(per);
    for(j=0;j<per;j++)
        in>>Cc[i][j];}
in>>str;
Co.resize(dm);
for(i=0;i<dm;i++){
    Co[i].resize(per);
    for(j=0;j<per-1;j++)
        in>>Co[i][j];}
in>>str;
Ch.resize(dm);
for(i=0;i<dm;i++){

```

```

        Ch[i].resize(per);
        for(j=0;j<per-1;j++)
            in>>Ch[i][j];}
    in>>str;
    Cb.resize(dm);
    for(i=0;i<dm;i++){
        Cb[i].resize(per);
        for(j=0;j<per-1;j++)
            in>>Cb[i][j];}
    in>>str>>f;
    price.resize(per);
    in>>str;
    for(i=0;i<per;i++)
        in>>price[i];
    in>>str>>objus1;
    in>>str>>objas1;
    in>>str>>objus2;
    in>>str>>objas2;
    in.close();
    SA.vector01ut.resize(per, vector<int>(ut,0)); yedek01ut.resize(per,
vector<int>(ut,0));
    SA.vector01dm.resize(per, vector<int>(dm,0));
    yedek01dm.resize(per, vector<int>(dm,0));
    SA.vector02.resize(4, vector<int>(m)); yedek02.resize(4,
vector<int>(m));
    kapasitedm.resize(dm, vector<double>(per,0));
    kapasiteut.resize(ut, vector<double>(per,0));
    acik_dm.resize(per);
    acik_ut.resize(per);
    X.resize(ted);
    Y.resize(ut);
    Z.resize(dm);
    kapasiteted.resize(ted);
    for(k1=0;k1<ted;k1++){ X[k1].resize(ut); kapasiteted[k1].resize(h);
        for(k=0;k<ut;k++){ X[k1][k].resize(h); Y[k].resize(dm);
            for(i=0;i<dm;i++){ Z[i].resize(m); Y[k][i].resize(per);
fill(Y[k][i].begin(),Y[k][i].end(),0);
                for(j=0;j<m;j++){ Z[i][j].resize(per);
fill(Z[i][j].begin(),Z[i][j].end(),0);}}
                    for(i=0;i<h;i++){ X[k1][k][i].resize(per);
fill(X[k1][k][i].begin(),X[k1][k][i].end(),0);
                        kapasiteted[k1][i].resize(per);}}}
    SA.Tfirst=LabeledEdit1->Text.ToDouble();
    SA.Tlast=LabeledEdit4->Text.ToDouble();
    SA.iternum=LabeledEdit2->Text.ToInt();
    SA.ro=LabeledEdit3->Text.ToDouble();
    start=clock();
    kapasitedm=c;
    kapasiteut=b;
    kapasiteted=a;

```

```

for(i=0;i<m;i++)
{
    bas:        int yeter1=0, yeter2=0;
                int it1=random(per);
                if(d[i][it1]==INT_MAX) d[i][it1]=yer;
                int it2=random(dm);
                double tal, tal1, tal2;
                tal=(4*d[i][it1] + (d[i][it1]*0.9 + (d[i][it1] -d[i][it1] *0.9)/2) +
(d[i][it1] + (d[i][it1] *1.4 - d[i][it1] )/2))/6.000;
durak1:        if(tal<=kapasitedm[it2][it1])
                {
                    if(int(acik_dm[it1].size())<=dm-1)
                    {
                        SA.vector01dm[it1][it2]=1;
                        SA.vector02[0][i]=it1;
                        SA.vector02[1][i]=it2;
                        Z[it2][i][it1]=Z[it2][i][it1]+tal;
                        kap=Z[it2][i][it1];
                        kapasitedm[it2][it1]=kapasitedm[it2][it1]-tal;
                    }
                    else
                    {
                        yer=d[i][it1];
                        k=it1;
                        d[i][it1]=INT_MAX;
                        goto bas;
                    }
                }
            }
        else
        {
            it2=random(dm);
            yeter1=yeter1+1;
            if(yeter1==dm) goto bas;
            else goto durak1;
        }
    }
    int it3=random(ut);
durak2:        if(tal<kapasiteut[it3][it1])
                {
                    if(int(acik_ut[it1].size())<=ut-1)
                    {
                        SA.vector01ut[it1][it3]=1;
                        SA.vector02[2][i]=it3;
                        tal1=tal/(1-alfa[it3][it2][it1]);
                        Y[it3][it2][it1]=Y[it3][it2][it1]+tal1;
                        kapasiteut[it3][it1]=kapasiteut[it3][it1]-tal1;
                    }
                }
            }
        else
        {
            it3=random(ut);

```

```

        yeter2=yeter2+1;
        if(yeter2==dm) goto bas;
        else goto durak2;
    }
    kopy.clear();
    kopy.resize(ted);
    for(j=0;j<ted;j++) kopy[j]=d1[j][it3];
durak0:    int
it0=distance(kopy.begin(),min_element(kopy.begin(),kopy.end()));
    double toph=0;
    for(j=0;j<h;j++) toph=toph+kapasiteted[it0][j][it1];
    if(tal<toph){
        SA.vector02[3][i]=it0;
        for(j=0;j<h;j++)
        {
            tal2=tal1*ra[j];
            X[it0][it3][j][it1]=X[it0][it3][j][it1]+tal2;
            kapasiteted[it0][j][it1]=kapasiteted[it0][j][it1]-tal2;
        }
    }
    else
    {
        kopy[it0]=INT_MAX;
    }
it0=distance(kopy.begin(),min_element(kopy.begin(),kopy.end()));
    goto durak0;
}
}
for(i=0;i<per;i++)
{
    for(j=0;j<dm;j++)
    {
        if(SA.vector01dm[i][j]==1) acik_dm[i].push_back(j);
    }
    for(j=0;j<ut;j++){
        if(SA.vector01ut[i][j]==1) acik_ut[i].push_back(j);
    }
}
SA.bestvectorut.resize(per, vector<int>(ut,0));
SA.bestvectordm.resize(per, vector<int>(dm,0));
SA.bestvector02.resize(4, vector<int>(m));
SA.bestvectorut=SA.vector01ut;
SA.bestvectordm=SA.vector01dm;
SA.bestvector02=SA.vector02;
amac();
if(stat<lambda) SA.obj=stat;
else SA.obj=lambda;
if(SA.obj<0)
{
    TM=3*TM;
}

```

```

        SM=3*SM;
    }
    out<<CheckListBox1->Items->Strings[count].c_str()<<'\n';
    SA.best=SA.obj;
    bestTM=TM;
    bestSM=SM;
    Temp=SA.Tfirst;
    double L1,L2,L3,nL1,nL2,nL3;
    L1=SA.obj; L2=TM; L3=SM;
    while(Temp>SA.Tlast)
    {
        for(iter=0;iter<SA.itemnum;iter++)
        {
            yedek01ut=SA.vector01ut;
            yedek01dm=SA.vector01dm;
            yedek02=SA.vector02;
            if(Temp<88){
                int klm;
                klm=++klm;
            }
            SA.neighbor();
            amac();
            if(stat<lambda) nL1=stat;
            else nL1=lambda;
            if(nL1<0)
            {
                nL2=3*TM;
                nL3=3*SM;
            }
            else
            {
                nL2=TM;
                nL3=SM;
            }
            if(nL2<L2 && nL3<L3){
                L1=nL1;
                L2=nL2;
                L3=nL3;
                if(nL2<bestTM && nL3<bestSM)
                {
                    SA.best=nL1;
                    bestTM=nL2;
                    bestSM=nL3;
                    SA.bestvectorut=SA.vector01ut;
                    SA.bestvectordm=SA.vector01dm;
                    SA.bestvector02=SA.vector02;}}
            else{
                if(((nL2-L2)/(Temp+0.000))<(-381))
                {
                    SA.accept1=exp(-(-381));

```

```

    }
    else SA.accept1=exp(-(nL2-L2)/(Temp+0.000));
    if(((nL3-L3)/(Temp+0.000))<(-381))
    {
        SA.accept2=exp(-(-381));
    }
    else SA.accept2=exp(-(nL3-L3)/(Temp+0.000));
    double u=random(1000)/1000.000;
    if(u<SA.accept1 && u<SA.accept2){
        L1=nL1;
        L2=nL2;
        L3=nL3;
    }
    else
    {
        SA.vector01ut=yedek01ut;
        SA.vector01dm=yedek01dm;
        SA.vector02=yedek02;
    }
}
Temp=Temp*SA.ro;
}
finish=clock();
time=(finish-start)/CLK_TCK;
out<<SA.best<<"t"<<bestTM<<"t"<<bestSM<<"t"<<time<<"n"<<"n";
for(i=0;i<per;i++)
{
    for(j=0;j<ut;j++)
    {
        out<<SA.bestvectorut[i][j]<<" ";
    }
    out<<"\n";
}
out<<"n"<<"n";
for(i=0;i<per;i++)
{
    for(j=0;j<dm;j++)
    {
        out<<SA.bestvectordm[i][j]<<" ";
    }
    out<<"\n";
}
out<<"n"<<"n";
for(i=0;i<4;i++)
{
    for(j=0;j<m;j++)
    {
        out<<SA.bestvector02[i][j]<<" ";
    }
}

```

```

        out<<"\n';
    }
}
ra.clear(); PU.clear(); PW.clear();
price.clear();
b.clear(); c.clear(); d.clear();
fu.clear(); fw.clear(); alfa.clear(); d1.clear(); d2.clear(); d3.clear();
Cc.clear(); Co.clear(); Ch.clear(); Cb.clear();
a.clear();
SA.vector01ut.clear();
SA.vector01dm.clear();
SA.vector02.clear();
Z.clear();
acik_dm.clear();
acik_ut.clear();
kapasitedm.clear();
kapasiteut.clear();
kapasiteted.clear();
count=count+1;
}
out.close();
ShowMessage("BÝTTYÝ!!!!");

}
//-----

void __fastcall TForm1::CheckBox1Click(TObject *Sender)
{
    if(CheckBox1->Checked)
    {
        for(i=0;i<CheckListBox1->Items->Count;i++)
            CheckListBox1->Checked[i]=true;
    }
    else
    {
        for(i=0;i<CheckListBox1->Items->Count;i++)
            CheckListBox1->Checked[i]=false;
    }
}
//-----

void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    string str;
    str="C:\\Work\\HAS2v2-1\\SAsonuclar.txt";
    str="C:\\Windows\\notepad.exe "+str;
    system(str.c_str());
}
//-----

```