

**TEDARİK ZİNCİRİ AĞ TASARIMI
PROBLEMLERİ İÇİN ÇOK AMAÇLI VE
ÇOK AŞAMALI ÇÖZÜM YAKLAŞIMLARI**

Mehmet ALEGÖZ

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

Mayıs-2015

**Bu tez çalışması Anadolu Üniversitesi Bilimsel Araştırma Projeleri
Komisyonu Başkanlığı tarafından desteklenmiştir. Proje No: 1501F024**

JÜRİ VE ENSTİTÜ ONAYI

Mehmet ALEGÖZ'ün, **Tedarik Zinciri Ağ Tasarımı Problemleri İçin Çok Amaçlı ve Çok Aşamalı Çözüm Yaklaşımları** başlıklı **Endüstri Mühendisliği** anabilim dalındaki Yüksek Lisans Tezi, 26.05.2015 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı Soyadı	İmza
Üye (Tez Danışmanı)	: Yrd. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK	
Üye	: Prof. Dr. Refail KASIMBEYLİ	
Üye	: Doç. Dr. Zeynep Pelin BAYINDIR	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

TEDARİK ZİNCİRİ AĞ TASARIMI PROBLEMLERİ İÇİN ÇOK AMAÇLI VE ÇOK AŞAMALI ÇÖZÜM YAKLAŞIMLARI

Mehmet ALEGÖZ

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK

2015, 69 sayfa

Bu tezde, tedarikçi seçimi ve tedarik zinciri ağ tasarımı problemleri için bir çok ölçütlü karar verme yaklaşımı, bir çok amaçlı matematiksel model ve bir de metasezgisel yaklaşım geliştirilmiştir. Öne sürülen yaklaşımların etkinliğinin gösterilmesi için bir üretim firmasının tedarik zinciri ağı ele alınmıştır. Firmanın üretim yeri sabit ve değiştirilemez olduğundan tüm tedarik zinciri ağının tasarımı problemi 2 ayrı probleme ayrıştırılmıştır. İlk problem tedarikçilerle üretim alanı arasındaki ağın oluşturulmasına odaklanırken ikinci problem üretim tesisi, depolar ve müşteriler arasındaki ağın oluşturulmasına odaklanmıştır. İlk problem bir tedarikçi seçimi problemi olarak ele alınmıştır. Bu sebeple öncelikle tedarikçi seçimine etki eden ölçütler ve bu ölçütlerin ağırlıkları belirlenmiş daha sonra da bu ölçütlere göre tedarikçiler değerlendirilmiştir. Seçime etki eden hem nitel hem de nicel ölçütler olduğu için nitel ve nicel ölçütlerin bir arada ele alınmasına olanak tanıyan bir algoritmanın kullanımı gerekliliği ortaya çıkmıştır. Bu sebeple ilk problemin çözümü için Buckley'in bulanık AHP yaklaşımı kullanılmıştır. İkinci problem için çok amaçlı bir matematiksel model geliştirilmiş ve firmadan elde edilen verilere göre çözdürülmüştür. Hesaplama sonuçları modelin kabul edilebilir bir zamanda optimum çözümü verdiğini göstermiştir. Diğer taraftan, problem NP-Hard sınıfından bir problem olduğu için bu model büyük boyutlu problemlerde optimum çözümü veremeyecektir, bu sebeple bir metasezgisel yaklaşım geliştirilmiştir. Metasezgisel yaklaşımla yapılan hesaplama sonuçları önerilen yaklaşımın bir saniyeden daha az bir süre içerisinde yüksek kalitede sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: Tedarikçi seçimi, tedarik zinciri ağı tasarımı, metasezgisel yaklaşımlar

ABSTRACT

Master of Science Thesis

MULTI OBJECTIVE AND MULTI ECHELON SOLUTION APPROACHES FOR SUPPLY CHAIN NETWORK DESIGN PROBLEM

Mehmet ALEGÖZ

**Anadolu University
Graduate School of Science
Industrial Engineering Program**

Supervisor: Asst. Prof. Dr. Zehra KAMIŞLI ÖZTÜRK

2015, 69 pages

In this thesis, a multi criteria decision making approach, a multi objective mathematical model and a metaheuristic approach are developed for supplier selection and supply chain network design problems. The supply chain network of a production company is considered in order to show the efficiency of proposed approaches. Since the location of the production plant is fixed the supply chain network design problem divided into two separate problem. The first problem focuses on creating the network between the suppliers and production plant while the second problem focuses on creating the network between production plant, warehouses and customers. The first problem is thought as a supplier selection problem. Therefore, first the criteria and criteria weights which affect the supplier selection are determined and then the suppliers are evaluated according to those criteria. Since, there are both qualitative and quantitative criteria, it became necessary to use an algorithm which gives the opportunity of analyzing both qualitative and quantitative criteria together. For this reason, Buckley's Fuzzy AHP Algorithm is used for the first problem. For the second problem, a multi objective mathematical model is created and it is solved according to the data obtained from the company. The computational results showed that the model gives optimum solution within reasonable time. However, since the problem is an NP-Hard class problem, this model cannot find the optimum solution in big dimensional problems. For this reason, a metaheuristic approach is developed. The computational results of the metaheuristic approach shows that the proposed approach gives high quality solutions within less than a second.

Keywords: Supplier selection, supply chain network design, metaheuristic approaches

TEŞEKKÜR

Tezimi 1501F024 proje numarasıyla Bilimsel Araştırma Projesi olarak destekleyen Anadolu Üniversitesi Proje Destek Birimi'ne, Yüksek lisans eğitimim boyunca bana 2228-A programıyla burs desteği sağlayan TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı (BİDEB)'na ve başta tez danışmanım Yrd. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK olmak üzere yüksek lisans eğitimim boyunca bana bilgi ve tecrübelerini aktaran tüm hocalarıma teşekkür ederim.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ.....	vii
1. GİRİŞ	1
1.1. Tezin Amacı ve Kapsamı.....	1
1.2. Tedarik Zinciri ve Tedarik Zinciri Yönetimi Kavramları.....	1
1.3. Tedarik Zinciri Ağı Tasarımı.....	2
2. TEDARİKÇİ SEÇİMİ PROBLEMİ	
2.1. Literatür Taraması.....	4
2.2. Ölçütlerin Belirlenmesi.....	5
2.2.1 Ürün.....	6
2.2.2 Firma.....	6
2.2.3 Hizmet.....	7
2.3. Tip-2 Bulanık Kümeler.....	9
2.4. Kullanılan Çözüm Algoritması.....	10
2.5. Algoritmanın Problemden Uygulanması.....	12
2.6. Tedarikçilerin Değerlendirilmesi.....	16
3. AĞ TASARIMI İÇİN KESİN ÇÖZÜM YAKLAŞIMLARI	20
3.1. Literatür Taraması.....	20
3.2. Tez Kapsamında İncelenen Firmaya Dair Bilgiler.....	21
3.3. Amaçların Belirlenmesi.....	21
3.4. Matematiksel Modelin Geliştirilmesi.....	26
3.5. Hesaplama Sonuçları.....	30
4. AĞ TASARIMI İÇİN METASEZGİSEL YAKLAŞIMLAR	36
4.1. Metasezgisel Yaklaşımların Özellikleri.....	36
4.2. Literatür Taraması.....	37
4.3. Uygulama Verilerinin Oluşturulması.....	38
4.3.1 Küçük Boyutlu Problem.....	39
4.3.2 Orta Boyutlu Problem.....	39
4.3.3 Büyük Boyutlu Problem.....	39
4.4. Önerilen Gösterim Şekli.....	41
4.5. Kullanılan Metasezgisel Yaklaşım.....	42
4.6. Yasaklı Arama ve Rassal Atama.....	43
4.7. Yasaklı Arama ve Yerel Arama.....	44

4.8. Atama İçin Kullanılan Diğer Sezgisel Yaklaşımlar.....	46
4.8.1 En Düşük Maliyetli Depoya Atama.....	46
4.8.2 En Yakın Depoya Atama.....	47
4.8.3 En Düşük Amaç Fonksiyonu Değeri Veren Depoya Atama.....	47
5. SONUÇLAR	50
KAYNAKLAR.....	52
Ek-1 Şehirlerin Talep ve Depo Açma Gideri Bilgileri.....	54
Ek-2 Problem İçin Öneilen Metasezgisel Yaklaşımın C# Kodu.....	57

ŞEKİLLER DİZİNİ

2.1. Aralıklı tip-2 bulanık kümenin üst ve alt üyelik fonksiyonu	10
2.2. Ana ölçütlerin ağırlıklarını gösteren pasta grafiği	14
2.3. Alt ölçütlerin ağırlıklarını gösteren pasta grafiği	16
3.1. Birim maliyetin uzaklığa bağlı değişimi	23
3.2. Bir üretim alanı iki aday depo ve bir müşteriden oluşan tedarik zinciri	25
4.1. Problem boyutuna bağlı olarak çözüm süresinin değişimi	40

ÇİZELGELER DİZİNİ

2.1. Ölçütler ve alt ölçütler.....	8
2.2. Sözlü ifadelerin tip-2 bulanık küme karşılıkları.....	11
2.2. Bir uzmanın görüşünü yansıtan örnek matris.....	12
2.4. Ana ölçütler için bulanık geometrik ortalamalar	12
2.5. Ürün ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar	13
2.6. Firma ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar	13
2.7. Hizmet ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar.....	13
2.8. Ana ölçütlerin bulanık ağırlıkları	13
2.9. Ürün ana ölçütünün alt ölçütlerinin bulanık ağırlıkları.....	13
2.10. Firma ana ölçütünün alt ölçütlerinin bulanık ağırlıkları	14
2.11. Hizmet ana ölçütünün alt ölçütlerinin bulanık ağırlıkları	14
2.12. Ana ölçütlerin ağırlıkları	14
2.13. Alt ölçütlerin ağırlıkları.....	15
2.14. Maliyet alt ölçütü açısından tedarikçilerin değerlendirilmesi.....	16
2.15. Kalite alt ölçütü açısından tedarikçilerin değerlendirilmesi.....	17
2.16. Uzmanların ortak görüşüne göre tedarikçilerin ölçütler için ağırlıkları.....	17
2.17. Tedarikçilerin genel ağırlıkları.....	18
3.1. Şehir talepleri ve depo açma giderleri.....	22
3.2. Uzaklığa bağlı birim taşıma maliyeti	22
3.3. Birim maliyet fonksiyonuyla gerçek değerlerin karşılaştırması	24
3.4. Çözüm sonucu elde edilen veriler	31
3.5. Adana deposu hakkında bilgiler	32
3.6. Bilecik deposu hakkında bilgiler.....	33
3.7. Samsun deposu hakkında bilgiler.....	34
3.8. Farklı ağırlıklara göre elde edilen çözümler	35
4.1. Farklı boyutlardaki problemlerin kesin çözüm sonuçları.....	40
4.2. Problem için önerilen gösterim şekli	41
4.3. Örnek bir gösterim	41
4.4. Kesin çözümle rassal atama yaklaşımının karşılaştırılması	44
4.5. Kesin çözümle yerel aramayla atama yaklaşımının karşılaştırılması	45
4.6. Kesin çözümle en düşük maliyetli depoya atama karşılaştırılması.....	46
4.7. Kesin çözümle en yakın depoya atama yaklaşımının karşılaştırılması.....	47
4.8. Kesin çözümle önerilen yaklaşımın karşılaştırılması.....	48
4.9. Önerilen yaklaşımın farklı boyuttaki problemlerde verdiği sonuçlar	49

1. GİRİŞ

Bu bölümde öncelikle tezin amacından ve kapsamından bahsedilmiş sonrasında ise tedarik zinciri yönetimi ve tedarik zinciri ağ tasarımı kavramlarına yönelik temel bilgiler verilmiştir.

1.1. Tezin Amacı ve Kapsamı

Tez kapsamında Bilecik merkezli bir firmanın tedarik zinciri ağının tasarımına odaklanılmıştır. Firmanın üretim tesisi sabit ve değiştirilemez olmak üzere Bilecik'te yer almaktadır. Bu sebeple, firmanın tüm tedarik zincirinin tasarımını iki ayrı probleme ayrıştırma imkanı doğmuştur. Bu problemlerden birincisi zincirin ilk halkasını oluşturan tedarikçilerin belirlenmesi ve tedarikçilerle üretim tesisi arasındaki ağın oluşturulması; ikinci kısmı da üretim alanı, depolar ve müşterilerden oluşan ağın oluşturulması olarak belirlenmiştir. İlk problem yapısı itibariyle bir tedarikçi seçimi problemi olarak ele alınmış ve çözümü için bulanık çok ölçütlü karar verme tekniklerinden yararlanılmıştır. İkinci problemin çözümü içinse çok amaçlı karma tamsayılı matematiksel model ve metasezgisel yaklaşımlar geliştirilmiştir. Çalışma ile firma için uzun yıllar kullanılabilecek, rekabetçi, maliyet ve müşteri memnuniyeti odaklı yeni bir tedarik zinciri ağı tasarlamak amaçlanmıştır.

1.2. Tedarik Zinciri ve Tedarik Zinciri Yönetimi Kavramları

Bir tedarik zinciri içerisinde tedarikçilerin, üreticilerin, dağıtımçıların ve perakendecilerin bulunduğu bütünleşik bir yapı olarak tanımlanabilir [1]. Bu yapı ürün için gerekli hammaddenin temini aşamasından, son ürün olarak müşteriye ulaşması aşamasına kadar gerçekleştirilen tüm süreçleri içerir. Bu süreçler içerisinde ürün için gerekli hammaddenin tedarikçiden tedarik edilmesi, hammaddelerin ürüne dönüştürülmesi, bitmiş ürünlerin depolara taşınması ve oradan da müşterilere gönderilmesi süreçleri sayılabilir. Bunlara ek olarak zincir boyunca sayılamayacak kadar çok sayıda alt süreç vardır. Bu süreçlerin her biri kendine özgü amaçlara sahiptir ve bu amaçlar diğer süreçlerin amaçları ile çoğu

zaman çelişmektedir. Bu yüzden bu süreçlerin her birinin tek başına optimize edilmesi bütün bir sistemin optimize edilmesini sağlamaz. Bu da firmanın başarısını engelleyici bir durumdur. Bu durumu aşmak için tüm süreçlerin oluşturduğu sistemi bütünüyle ele alma ve sistemi optimize ederek farklı süreçler arasında bir ödüleşme noktası bulma zorunluluğu ortaya çıkmıştır. Tedarik zinciri yönetimi kavramının ortaya çıkmasındaki temel neden de budur. Tedarik zinciri yönetimi yaklaşımıyla ilk tedarikçiden son müşteriye kadar olan tüm süreçler sistem bakış açısıyla ele alınmakta ve bütün bir sistem optimize edilmeye çalışılmaktadır.

Tedarik zincirlerinde ileri yönlü ve geri yönlü akışlar söz konusudur. Geleneksel tedarik zincirlerinde ileri yönlü olarak ürün ve bilgi akışının geri yönlü olarak ise bilgi ve para akışının olduğu söylenebilir. Özellikle 20. yüzyılın sonlarına doğru küreselleşmenin ve geniş kapsamlı şirket yapılarının oluşmasıyla iş hayatından ve akademik dünyadan pek çok kişi tedarik zinciri yönetimi kavramına odaklanmıştır. Bu konuya olan ilgi o günlerden sonra artarak devam etmiştir. Günümüzde de pek çok araştırmacı tedarik zincirlerinin tasarımı, yönetimi, tedarik zinciri ağlarının optimizasyonu ve tedarik zinciri ağlarındaki performans ölçütleri üzerine çalışmalar yapmakta ve bu kapsamlı yapının veriminin artırılmasının yollarını aramaktadır. Geleneksel tedarik zinciri yapısına ek olarak bazı araştırmacılar tersine lojistik kavramı üzerinde durmakta ve bir ürünün geri dönüşümünün sağlanması, tekrar işlenmesi ve tekrar kullanılması konularında çalışmalar yapmaktadır.

1.3. Tedarik Zinciri Ağı Tasarımı

Tedarik zinciri ağı tasarımı tedarik zincirinin yapısının oluşturulması sürecidir. Oluşturulan bu yapı uzun süre kullanılacağı için süreç boyunca verilen kararlar şirketin uzun vadeli performansını etkileyecek stratejik kararlardır [2]. Tedarik zinciri ağının tasarlanmasıyla tedarik zinciriyle ilgili izleyen soruların cevaplarının verilmesi amaçlanır.

- Her bir tesis nereye açılmalıdır?
- Her bir tesisin kapasitesi ne kadar olmalıdır?
- Hangi tesislerden kaçar tane açılmalıdır?

- Tesisler arasında nasıl bir akış olmalıdır, hangi tesisler hangi tesislere bağılı olarak çalışmalıdır?
- Şirket hangi tedarikçilerle çalışmalıdır, tedarikçilerini hangi ölçütlere göre seçmelidir?

Tedarik zinciri ağının tasarımı sürecini etkileyen en önemli faktör firmanın amacıdır. Zira amacı toplam tedarik zinciri maliyetini en aza indirmek olan bir firmanın kuracağı tedarik zinciri ağı ile amacı müşterilere en hızlı hizmeti vermek olan bir firmanın kuracağı tedarik zinciri ağı aynı olmayacaktır. Bu da tedarik zinciri ağının tasarımı sürecinde yapılması gereken ilk şeyin amaçların doğru belirlenmesi olduğunu göstermektedir. Tedarik zinciri ağının amacı belirlenirken firmanın stratejik amaçları dikkate alınmalı ve tüm ağın tasarımı dolaylı olarak firmanın stratejik amaçlarına göre yapılmalıdır.

Tedarik zinciri yönetimi sadece ulusal boyutta bir kavram değildir [3]. Özellikle şirketlerin çok uluslu bir yapıya dönüşmesiyle, küresel tedarik zinciri ağlarının tasarımı ve yönetimi konusu gündeme gelmiş, araştırmacılar tarafından küresel tedarik zinciri ağlarının tasarımına yönelik yaklaşımlar geliştirilmiştir. Ulusal tedarik zinciri ağlarının tasarımında karşılaşılan güçlükler ve problemlere ek olarak uluslararası tedarik zinciri ağlarının tasarımında kur farkları, gümrük kuralları, ülkelere arası siyasi ve ekonomik belirsizlikler, ülkeler arası taşıma ilkeleri farklılıkları gibi ek belirsizlikler ve güçlükler vardır.

2. TEDARİKÇİ SEÇİMİ PROBLEMİ

Tedarik zinciri ağı tasarımının ilk kısmını tedarikçilerin belirlenmesi oluşturmaktadır. Tedarikçi seçimi problemi her firmanın üzerinde ciddiyle durması gereken önemli bir problemdir. Çünkü bir firmanın üretim süreçlerinin sorunsuz bir şekilde yürüyebilmesinin ilk şartlarından biri de doğru tedarikçi ile çalışmaktır. Tedarikçinin ürünü tedarik etmedeki performansı firmanın üretim performansını doğrudan etkilemektedir. Bu da doğru tedarikçi ile çalışmanın önemini ortaya koymaktadır.

Bu bölümde sırasıyla literatür taraması, tedarikçi seçimi için ölçütlerin ve ölçüt ağırlıklarının belirlenmesi ve son olarak ölçütlere göre tedarikçi seçimi konularına yer verilmiştir.

2.1. Literatür Taraması

Tedarikçi seçimi problemi, içerisinde nitel ve nicel pek çok faktörü barındıran ve bu faktörler arasında ödünleşme noktasının arandığı çok ölçütlü bir karar verme problemi olarak ele alınabilir. Literatürde tedarikçi seçimine yönelik farklı yaklaşımların kullanıldığı pek çok çalışma mevcuttur.

Ávila ve ark. [4] tedarikçi seçimi sürecini çeşitli verilerin toplandığı ve analiz edildiği zaman ve kaynak harcanması gereken bir süreç olarak tanımlamıştır. Yaptıkları çalışmada tedarikçi seçiminde etkili olan ana ölçütleri kalite, finans, sinerji, maliyet ve üretim sistemi olarak tanımlamışlar ve bu ana ölçütlerin her birinin altında belli alt ölçütler tanımlamışlardır. Daha sonra bu ölçütlerin ağırlıklarını belirlemek için çeşitli anketler yapmışlar ve çok ölçütlü karar verme yöntemiyle ağırlıkları belirlemişlerdir. Rajesh ve Malliga [5] yaptıkları çalışmada kalitenin iyileştirilmesi, esnekliğin sağlanması ve temin süresinin azaltılması için iyi tedarikçilerle uzun vadeli bir ortaklığın kurulması gerektiğini belirtmişlerdir. Dolayısıyla tedarikçi seçiminin stratejik bir problem olduğunu ifade etmişlerdir. Bu motivasyonla stratejik açıdan tedarikçilerin seçimi için Analitik Hiyerarşi Prosesi (AHP) ve Kalite Fonksiyonu Yayılımı (QFD) yöntemlerini birleştiren bir yaklaşım öne sürmüşlerdir. Xia ve Wu [6] tedarikçi seçimi problemini nitel ve nicel faktörlerin birlikte ele alındığı çok ölçütlü bir karar verme problemi olarak

tanımlamış, en iyi tedarikçileri seçmek için aralarında çelişki bulunan bu nicel ve nitel faktörlerin birlikte ele alınması gerektiğini ifade etmişlerdir. Bu sebeple çalışmalarında AHP, bulanık mantık ve çok amaçlı karma tam sayılı programlamayı içeren bir yaklaşım kullanmışlardır. Ayrıca kurulan matematiksel modeli çözmesi için de etkin bir yöntem önermişlerdir. Rezaei ve Ortt, [7] firmaların vermesi gereken stratejik kararlardan birinin de tedarikçilerin gruplandırılması olduğunu ifade etmiştir. Bu motivasyonla, bulanık AHP yöntemiyle tedarikçileri gruplandıran bir yaklaşım öne sürmüşler ve bu yaklaşımı tava üretimi yapan bir firmada uygulamışlardır. Gruplandırma yapıldıktan sonra ayrıca her bir grup için uygulanabilecek çeşitli stratejilerden bahsetmişlerdir. Masi ve ark. [8] literatürde benzerine çok nadir rastlanan bir çalışma yapmışlardır. Literatürde çok farklı özelliklere sahip sayısız tedarikçi seçimi yaklaşımı olduğunu ve bunlardan her birinin bazı koşullarda iyi sonuç vereceğini ifade etmişler ve bu motivasyonla farklı koşullarda hangi yaklaşımın daha iyi çalışabileceğini belirleyen bir model geliştirmişlerdir. Zouggari ve Benyoucef [9] çoğunlukla nitel ölçütlerin olduğu bulanık mantık temelli bir tedarikçi seçimi yaklaşımı geliştirmişlerdir. Çalışmada, performans stratejisi, hizmet kalitesi, inovasyon ve risk gibi nitel ölçütler kullanılmıştır. Bu nitel ölçütler de bulanık mantık yaklaşımlarıyla değerlendirilmiştir.

2.2. Ölçütlerin Belirlenmesi

Tedarikçi seçimi probleminin çözümü için öncelikle tedarikçi seçimine etki eden ölçütlerin ve bu ölçütlerin süreçteki ağırlıklarının belirlenmesi gerekmektedir. Ölçütlerin belirlenmesi sürecinde temel olarak iki kaynaktan yararlanılmıştır. Bu kaynaklardan biri konuyla ilgili çalışmaları içeren bilimsel çalışmalar, bir diğeri ise satın alma ve tedarik zinciri yönetimi alanında uzun süre çalışmış, sektör tecrübesine sahip uzmanlardır. Bu iki kaynaktan edinilen bilgiler derlenmiş ve tedarikçi seçimi için 3 ana ölçütün altında 14 ölçüt belirlenmiştir. Bu ölçütler ve alt ölçütler başlıklar halinde ilerleyen alt bölümlerde verilmiştir.

2.2.1. Ürün

Ürün ana ölçütü tedarikçi seçiminde doğrudan ürünle ilgili olan alt ölçütleri içermektedir. Bu alt ölçütler maliyet, kalite, tedarik kapasitesi ve ürün özelleştirme yeteneği olarak belirlenmiştir.

- Maliyet: Tedarik edilen ürünün maliyetini ele alan alt ölçüttür.
- Kalite: Tedarik edilen ürünün kalitesini ele alan alt ölçüttür.
- Tedarik Kapasitesi: Tedarikçinin belli bir temin süresinde söz konusu ürün için tedarik kapasitesini ele alan alt ölçüttür.
- Ürün Özelleştirme Yeteneği: Tedarikçinin firmanın beklentilerini ya da özel taleplerini göz önünde bulundurarak üründe kısmi özelleştirmeler yapabilme gücünü ele alan alt ölçüttür.

2.2.2. Firma

Firma ana ölçütü tedarikçi firmanın kendine has özellikleriyle ilgili alt ölçütleri barındıran ana ölçüttür. Bu ana ölçütün altında coğrafi konum, saygınlık ve referanslar, finansal istikrar, özel teknolojiler ve patentler ve son olarak yönetim ve organizasyon alt ölçütleri yer alır.

- Coğrafi Konum: Tedarikçinin coğrafi konumunu ve firmaya yakınlığını ele alan alt ölçüttür.
- Saygınlık ve Referanslar: Tedarikçinin sektördeki deneyimini, daha önce çalıştığı ve kendisine referans olabilecek firmaları ve saygınlığını ele alan alt ölçüttür.
- Finansal İstikrar: Tedarikçinin genel anlamda finansal durumunu ve finansal risklerini ele alan alt ölçüttür.
- Özel Teknolojiler ve Patentler: Tedarikçi firmanın bünyesinde bulundurduğu, rekabet avantajı sağlayabilecek özel teknolojileri ve varsa patentleri ele alan alt ölçüttür.
- Yönetim ve Organizasyon: Tedarikçinin genel anlamda yönetim sürecini ve organizasyonel yapısını ele alan alt ölçüttür.

2.2.3. Hizmet

Hizmet ana ölçütü altında tedarikçinin süreç boyunca verdiği hizmeti değerlendiren alt ölçütler bulunmaktadır. Bu alt ölçütler temin süresi, garanti ve satış sonrası destek, iletişim ve bilgi sistemleri, eğitim ve sürekli iyileştirme, çevresel ve sosyal sorumluluk olarak sıralanabilir.

- Temin Süresi: Tedarikçinin ürünü temin etme süresini ele alan alt ölçüttür.
- Garanti ve Satış Sonrası Destek: Tedarikçinin tedarik ettiği ürün için verdiği garantiyi ve ürün tedarikinden sonra sağladığı satış sonrası desteği ve hizmetleri ele alan alt ölçüttür.
- İletişim ve Bilgi Sistemleri: Tedarikçinin sahip olduğu iletişim ve bilgi sistemleri altyapısını, kullandığı yazılımları ve donanımları ele alan alt ölçüttür.
- Eğitim ve Sürekli İyileştirme: Tedarikçinin şirket içi eğitim ve sürekli iyileştirme kültürünü, değişime ve gelişime olan ilgisini ve şirket çalışanlarının sürekli iyileştirme yönündeki motivasyonunu ele alan alt ölçüttür.
- Çevresel ve Sosyal Sorumluluk: Tedarikçinin sahip olduğu çevresel ve sosyal sorumluluk bilincini ve bu konuda gerçekleştirdiği/planladığı projeleri ele alan alt ölçüttür.

Tüm bu ölçütler ve alt ölçütler Çizelge 2.1’de özetlenmiştir. Her bir alt ölçüt için literatürden bazı kaynaklara yer verilmiştir. Çizelge 2.1’deki kaynaklara ek olarak pek çok alt ölçütün önemi uzmanlar tarafından da teyit edilmiştir

Çizelge 2.1’de de görüldüğü gibi tedarikçi seçiminde ele alınması gereken alt ölçütlerin bazıları niceldir ve herhangi iki tedarikçiyi bu nicel ölçütler açısından kıyaslamak istendiğinde rakamlar aracılığıyla çok kolay biçimde kıyaslama yapılabilir. Örnek olarak, maliyet ölçütü açısından tedarikçileri kıyaslamak çok kolaydır. Diğer taraftan ele alınması gereken ölçütler arasında yönetim ve organizasyon, eğitim ve sürekli iyileştirme, çevresel ve sosyal sorumluluk gibi nicel ifadelerle çok kolay ifade edilemeyecek alt ölçütler mevcuttur ve tedarikçinin seçimi sürecinde güvenilir bir sonuca varabilmek için bu alt ölçütlerin de ele alınması gerekmektedir. Bu da tedarikçi seçimi probleminin çözümü için nicel ve

nitel ölçütleri bir arada değerlendirebilen bir yaklaşımın kullanılmasını zorunlu kılmaktadır.

Çizelge 2.1. Ölçütler ve alt ölçütler

Ölçüt	Alt Ölçüt	Kaynak
Ürün	C_{11} Maliyet	[4]
		[5]
C_1	C_{12} Kalite	[6]
		[10]
	C_{13} Tedarik Kapasitesi	[7]
		[6]
	C_{14} Ürün Özelleştirme Yeteneği	[8]
Firma	C_{21} Coğrafi Konum	[9]
C_2	C_{22} Yönetim ve Organizasyon	[10]
	C_{23} Finansal İstikrar	[4]
		[9]
	C_{24} Saygınlık ve Referanslar	[8]
		[10]
	C_{25} Özel Teknolojiler ve Patentler	[7]
Hizmet	C_{31} Temin Süresi	[7]
C_3	C_{32} Garanti ve Satış Sonrası Destek	[8]
		[6]
	C_{33} İletişim ve Bilgi Sistemleri	[7]
	C_{34} Eğitim ve Sürekli İyileştirme	[7]
		[7]
	C_{35} Çevresel ve Sosyal Sorumluluk	[9]
		[8]

Tez kapsamında tedarikçi seçimi probleminin çözümü için aralıklı Tip-2 Bulanık AHP yönteminden yararlanılmıştır. Böylece kesin yargılarla karşılaştırılması mümkün olmayan iki alt ölçütün bulanık ifadelerle analizi mümkün olmuştur.

2.3. Tip-2 Bulanık Kümeler

Tip-2 bulanık kümelerle ilgili başlıca tanımlar ve özellikler aşağıdaki gibidir. [11]

Tanım 1: $\tilde{\tilde{A}}$ kümesi X evrensel kümesinde bir tip-2 bulanık küme olsun. Bu durumda $\tilde{\tilde{A}}$ kümesi aşağıdaki gibi bir $\mu_{\tilde{\tilde{A}}}$ tip-2 üyelik fonksiyonu ile ifade edilir.

$$\tilde{\tilde{A}} = \{((x, u), \mu_{\tilde{\tilde{A}}}(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1], 0 \leq \mu_{\tilde{\tilde{A}}}(x, u) \leq 1\} \quad (2.1)$$

burada J_x , $[0, 1]$ aralığını ifade eder.

Ayrıca $\tilde{\tilde{A}}$ tip-2 bulanık kümesi aşağıdaki gibi de ifade edilebilir.

$$\tilde{\tilde{A}} = \{((x, u), \mu_{\tilde{\tilde{A}}}(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1], 0 \leq \mu_{\tilde{\tilde{A}}}(x, u) \leq 1\} \quad (2.2)$$

Burada da benzer şekilde $J_x \subseteq [0, 1]$ 'dir ve $\iint$ ifadesi kabul edilebilir tüm x ve u değerlerinin birleşimini ifade eder.

Tanım 2: $\tilde{\tilde{A}}$, X evreninde $\mu_{\tilde{\tilde{A}}}$ üyelik fonksiyonuyla temsil edilen bir tip-2 bulanık küme olsun. Eğer $\mu_{\tilde{\tilde{A}}}(x, u) = 1$ şartı sağlanırsa bu durumda $\tilde{\tilde{A}}$ kümesine aralıklı tip-2 bulanık küme denir. Aralıklı tip-2 bulanık küme, tip-2 bulanık kümenin özel bir halidir ve aşağıdaki gibi gösterilir.

$$\tilde{\tilde{A}} = \int_{x \in X} \int_{u \in J_x} 1/(x, u) \quad (2.3)$$

burada da $J_x \subseteq [0, 1]$ 'dir.

Tip-2 bulanık kümenin üst ve alt üyelik fonksiyonlarının her biri birer tip-1 üyelik fonksiyonudur. Aralıklı tip-2 bulanık kümlere yapıları gereği daha fazla esneklik sağlarlar. Şekil 2.1'de örnek bir yamuksal aralıklı tip-2 bulanık küme gösterilmiştir, bu küme aşağıdaki gibi ifade edilir.

$$\tilde{\tilde{A}} = (\tilde{\tilde{A}}^U, \tilde{\tilde{A}}^L) = ((a_{i1}^U, a_{i2}^U, a_{i3}^U, a_{i4}^U; H_1(\tilde{\tilde{A}}^U), H_2(\tilde{\tilde{A}}^U)), (a_{i1}^L, a_{i2}^L, a_{i3}^L, a_{i4}^L; H_1(\tilde{\tilde{A}}^L), H_2(\tilde{\tilde{A}}^L))) \quad (2.4)$$

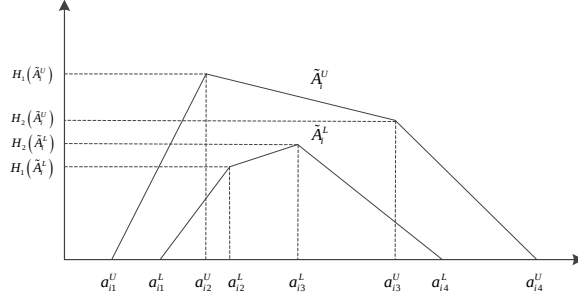
Bu ifadede $\tilde{\tilde{A}}_i^U$ ve $\tilde{\tilde{A}}_i^L$ tip-1 bulanık kümelerdir. $a_{i1}^U, a_{i2}^U, a_{i3}^U, a_{i4}^U, a_{i1}^L, a_{i2}^L, a_{i3}^L$ and a_{i4}^L değerleri ise $\tilde{\tilde{A}}_i$ aralıklı tip-2 bulanık kümesinin referans noktalarıdır. $H_j(\tilde{\tilde{A}}_i^U)$, $\tilde{\tilde{A}}_i^U$ () üyelik fonksiyonundaki $a_{i(j+1)}^U$

elemanının üyelik değerini göstermektedir. Benzer şekilde $H_j(\tilde{A}_i^L)$ ise $\tilde{A}_i^L$ (

 üyelik fonksiyonunda $a_{i(j+1)}^L$ elemanının üyelik değerini göstermektedir.

$$H_1(\tilde{A}_i^U) \in [0,1], H_2(\tilde{A}_i^U) \in [0,1], H_1(\tilde{A}_i^L) \in [0,1]$$

$$H_2(\tilde{A}_i^L) \in [0,1] \text{ ve } 1 \leq i \leq n$$



Şekil 2.1.  Aralıklı tip-2 kümesinin üst üyelik fonksiyonu $\tilde{A}_i^U$ ve alt üyelik fonksiyonu $\tilde{A}_i^L$

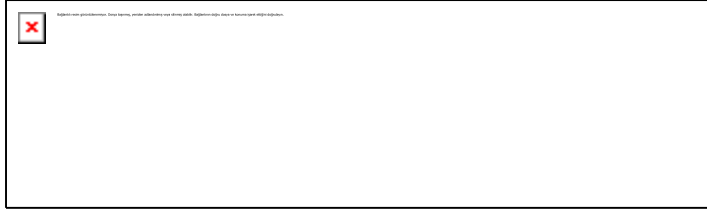
2.4. Kullanılan Çözüm Algoritması

Ağırlıkların belirlenmesi ve tedarikçilerin seçimi için aralıklı tip-2 bulanık kümeyi kullanmaya olanak sağlayan Buckley'in bulanık AHP algoritması [12] kullanılmıştır. Bu algoritma temel olarak 6 adımdan oluşmaktadır. Bu adımlar izleyen şekilde sıralanabilir.

Adım 1: Uzmanlara danışarak problemin hiyerarşik yapısındaki tüm ölçütler arasında ikili karşılaştırma matrislerini oluştur. Uzmanlar karşılaştırılan iki ölçütün birbirine göre önem derecelerini ifade etmek için Çizelge 2.2'deki sözlü ifadeleri kullanmalıdır.

Adım 2: Uzmanların sözlü ifadeler ile oluşturduğu ikili karşılaştırma matrislerini Saaty'nin skalasını [13] kullanarak sayısal değerlere dönüştür ve her bir matris için tutarlılığı hesapla, tutarsız olan görüşleri analizden çıkar.

Adım 3: Uzmanların sözlü ifadeleri ile oluşturulan ikili karşılaştırma matrislerini Çizelge 2.2 yardımıyla aşağıdaki gibi aralıklı tip-2 bulanık ikili karşılaştırma matrislerine dönüştür.



(2.5)



(2.6)

Adım 4: Aşağıdaki formülü kullanarak her bir ölçüt için bulanık geometrik ortalamaları hesapla.



(2.7)

Burada

$$\sqrt[n]{\tilde{a}_{ij}} = \left(\left(\sqrt[n]{\tilde{a}_{ij4}^U}, \sqrt[n]{\tilde{a}_{ij3}^U}, \sqrt[n]{\tilde{a}_{ij2}^U}, \sqrt[n]{\tilde{a}_{ij1}^U}; H_1(\tilde{a}_{ij}^U), H_2(\tilde{a}_{ij}^U) \right), \left(\sqrt[n]{\tilde{a}_{ij4}^L}, \sqrt[n]{\tilde{a}_{ij3}^L}, \sqrt[n]{\tilde{a}_{ij2}^L}, \sqrt[n]{\tilde{a}_{ij1}^L}; H_1(\tilde{a}_{ij}^L), H_2(\tilde{a}_{ij}^L) \right) \right) \quad (2.8)$$

Adım 5: Her bir ölçüt için bulanık ağırlıkları aşağıdaki gibi hesapla.

$$\tilde{w}_i = \tilde{r}_i \otimes (\tilde{r}_1 \oplus \tilde{r}_2 \oplus \dots \oplus \tilde{r}_n)^{-1} \quad (2.9)$$

Adım 6: Her bir ölçütün ağırlığını hesaplamak için bulanık ağırlıkları aşağıdaki formül aracılığıyla durulaştır.

$$BNP\tilde{w}_i = \frac{1}{2} \left(\frac{1}{2} \sum_{i=1}^4 (a_i^L + a_i^U) \right) \otimes \frac{1}{4} \left(\sum_{i=1}^2 (H_i(A^L) + H_i(A^U)) \right) \quad (2.10)$$

Adım 7: Ağırlıkları aşağıdaki formülü kullanarak normalize et.

$$w_i = w_i / \sum_{i=1}^n w_i, i=1,2,3..n \quad (2.11)$$

Çizelge 2.2. Sözlü ifadelerin tip-2 bulanık küme karşılıkları

Absolutely strong (AS)	((8;9;9;10;1;1),(8,5;9;9;9,5;0,9;0,9))
Very strong (VS)	((6;7;7;8;1;1),(6,5;7;7;7,5;0,9;0,9))
Fairly strong (FS)	((4;5;5;6;1;1),(4,5;5;5;5,5;0,9;0,9))
Slightly strong (SS)	((2;3;3;4;1;1),(2,5;3;3;3,5;0,9;0,9))
Equal (E)	((1;1;1;1;1;1),(1,1;1;1;1,0,9;0,9))
Slightly weak (SW)	((0,25;0,33;0,33;0,5;1;1),(0,22;0,33;0,33;0,4;0,9;0,9))
Fairly weak (FW)	((0,17;0,2;0,2;0,25;1;1),(0,18;0,2;0,2;0,22;0,9;0,9))
Very weak (VW)	((0,13;0,14;0,14;0,17;1;1),(0,13;0,14;0,14;0,15;0,9;0,9))
Absolutely weak (AW)	((0,1;0,11;0,11;0,13;1;1),(0,11;0,11;0,11;0,12;0,9;0,9))

2.5. Algoritmanın Problemdede Uygulanması

Adım 1: Uzman görüşlerini almak için öncelikle Google Docs üzerinde bir anket hazırlanmıştır. Daha sonra bu anket LinkedIn’de pek çok tedarik zinciri yönetimi ve satın alma grubunda yayınlanmış ve konunun uzmanlarından anketi doldurmaları rica edilmiştir. Bu kapsamda toplamda 17 uzmanın görüşü alınmıştır. Çizelge 2.3’te bir uzmanın görüşü örnek olarak verilmiştir.

Çizelge 2.3. Bir uzmanın görüşünü yansıtan örnek matris

	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}
C_{31}	E	SS	FS	SS	VS
C_{32}	SW	E	SS	SS	FS
C_{33}	FW	SW	E	SW	FS
C_{34}	SW	SW	SS	E	SS
C_{35}	VW	FW	FW	SW	E

Adım 2: Her bir uzmanın görüşü Saaty’nin skalasına göre [13] sayısal ifadelere dönüştürülmüş ve tutarlılıkları hesaplanmıştır. Örnek olarak yukarıdaki matrisin tutarlılığı 0,032 olarak hesaplanmıştır ve bu değer 0,1 değerinden küçük olduğu için uzmanın görüşünün tutarlı olduğu anlaşılmıştır. Bu işlem bütün uzman görüşleri için tekrarlanmıştır ve uzman görüşlerinin tamamının tutarlı olduğu anlaşılmıştır. Daha sonra tüm uzmanların görüşlerinin geometrik ortalamaları alınarak tüm uzmanların ortak görüşlerini yansıtan ikili karşılaştırma matrisleri oluşturulmuştur.

Adım 3: Uzmanların ortak görüşleriyle oluşturulan ikili karşılaştırma matrisleri tip-2 bulanık ikili karşılaştırma matrislerine dönüştürülmüştür.

Adım 4: Bu adımda tip-2 bulanık ikili karşılaştırma matrislerinden formül yardımıyla bulanık geometrik ortalamalar hesaplanmıştır. Sonraki dört çizelgede ölçütler ve alt ölçütler için bulanık geometrik ortalamalar verilmiştir.

Çizelge 2.4. Ana ölçütler için bulanık geometrik ortalamalar

C_1	1,26	1,44	1,44	1,59	1,00	1,00	1,36	1,44	1,44	1,65	0,90	0,90
C_2	0,63	0,69	0,69	0,79	1,00	1,00	0,60	0,69	0,69	0,74	0,90	0,90
C_3	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,90	0,90

Çizelge 2.5. Ürün ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar

C_{11}	1,86	2,14	2,14	2,38	1,00	1,00	2,01	2,14	2,14	2,41	0,90	0,90
C_{12}	1,41	1,73	1,73	2,00	1,00	1,00	1,58	1,73	1,73	2,12	0,90	0,90
C_{13}	0,59	0,76	0,76	1,00	1,00	1,00	0,59	0,76	0,76	0,92	0,90	0,90
C_{14}	0,32	0,38	0,38	0,50	1,00	1,00	0,31	0,38	0,38	0,43	0,90	0,90

Çizelge 2.6. Firma ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar

C_{31}	2,70	3,38	3,38	4,00	1,00	1,00	3,05	3,38	3,38	4,09	0,90	0,90
C_{32}	0,28	0,34	0,34	0,44	1,00	1,00	0,27	0,34	0,34	0,38	0,90	0,90
C_{33}	1,00	1,24	1,24	1,52	1,00	1,00	1,07	1,24	1,24	1,52	0,90	0,90
C_{34}	1,15	1,33	1,33	1,52	1,00	1,00	1,24	1,33	1,33	1,49	0,90	0,90
C_{35}	0,50	0,64	0,64	0,87	1,00	1,00	0,48	0,64	0,64	0,78	0,90	0,90

Çizelge 2.7. Hizmet ana ölçütünün alt ölçütleri için bulanık geometrik ortalamalar

C_{31}	3,37	4,00	4,00	4,59	1,00	1,00	3,69	4,00	4,00	4,53	0,90	0,90
C_{32}	1,78	2,17	2,17	2,64	1,00	1,00	1,88	2,17	2,17	2,52	0,90	0,90
C_{33}	0,70	0,90	0,90	1,15	1,00	1,00	0,76	0,90	0,90	1,12	0,90	0,90
C_{34}	0,43	0,52	0,52	0,66	1,00	1,00	0,45	0,52	0,52	0,61	0,90	0,90
C_{35}	0,28	0,34	0,34	0,44	1,00	1,00	0,27	0,34	0,34	0,38	0,90	0,90

Adım 5: Bu adımda formül yardımıyla her bir ölçüt ve alt ölçüt için bulanık ağırlıklar hesaplanmıştır. Sonraki dört çizelgede ana ölçütler ve alt ölçütler için bulanık ağırlıklar verilmiştir.

Çizelge 2.8. Ana ölçütlerin bulanık ağırlıkları

C_1	0,37	0,46	0,46	0,55	1,00	1,00	0,40	0,46	0,46	0,56	0,90	0,90
C_2	0,19	0,22	0,22	0,27	1,00	1,00	0,18	0,22	0,22	0,25	0,90	0,90
C_3	0,30	0,32	0,32	0,35	1,00	1,00	0,30	0,32	0,32	0,34	0,90	0,90

Çizelge 2.9. Ürün ana ölçütünün alt ölçütlerinin bulanık ağırlıkları

C_{11}	0,32	0,43	0,43	0,57	1,00	1,00	0,34	0,43	0,43	0,54	0,25	0,25
C_{12}	0,24	0,35	0,35	0,48	1,00	1,00	0,27	0,35	0,35	0,47	0,25	0,25
C_{13}	0,10	0,15	0,15	0,24	1,00	1,00	0,10	0,15	0,15	0,21	0,25	0,25
C_{14}	0,05	0,08	0,08	0,12	1,00	1,00	0,05	0,08	0,08	0,10	0,25	0,25

Çizelge 2.10. Firma ana ölçütünün alt ölçütlerinin bulanık ağırlıkları

C_{31}	0,32	0,49	0,49	0,71	1,00	1,00	0,37	0,49	0,49	0,67	0,90	0,90
C_{32}	0,03	0,05	0,05	0,08	1,00	1,00	0,03	0,05	0,05	0,06	0,90	0,90
C_{33}	0,12	0,18	0,18	0,27	1,00	1,00	0,13	0,18	0,18	0,25	0,90	0,90
C_{34}	0,14	0,19	0,19	0,27	1,00	1,00	0,15	0,19	0,19	0,24	0,90	0,90
C_{35}	0,06	0,09	0,09	0,15	1,00	1,00	0,06	0,09	0,09	0,13	0,90	0,90

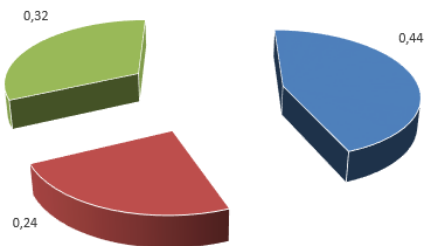
Çizelge 2.11. Hizmet ana ölçütünün alt ölçütlerinin bulanık ağırlıkları

C_{31}	0,36	0,50	0,50	0,70	1,00	1,00	0,40	0,50	0,50	0,64	0,90	0,90
C_{32}	0,19	0,27	0,27	0,40	1,00	1,00	0,20	0,27	0,27	0,36	0,90	0,90
C_{33}	0,07	0,11	0,11	0,18	1,00	1,00	0,08	0,11	0,11	0,16	0,90	0,90
C_{34}	0,05	0,07	0,07	0,10	1,00	1,00	0,05	0,07	0,07	0,09	0,90	0,90
C_{35}	0,03	0,04	0,04	0,07	1,00	1,00	0,03	0,04	0,04	0,05	0,90	0,90

Son olarak Adım 6 ve Adım 7’de verilen formüller yardımıyla ölçütlerin ve alt ölçütlerin ağırlıkları hesaplanmıştır. Ana ölçütlerin ağırlıkları Çizelge 2.12’de, bu ağırlıkları gösteren pasta grafiği ise Şekil 2.2’de verilmiştir.

Çizelge 2.12. Ana ölçütlerin ağırlıkları

	Global Ağırlık (%)	Global Ağırlık (%)
Ürün	0,443	%44
Firma	0,236	%24
Hizmet	0,321	%32

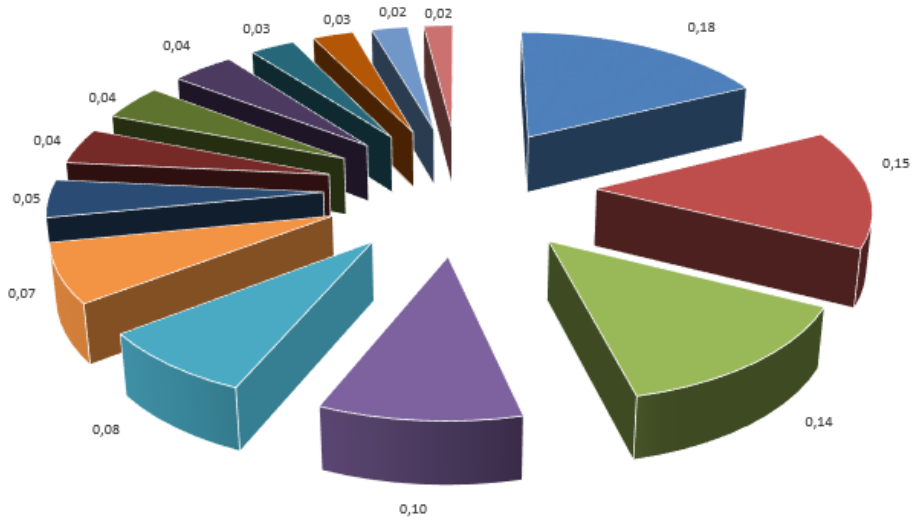


Şekil 2.2. Ana Ölçütlerin Ağırlıklarını Gösteren Pasta Grafiği

Benzer şekilde her bir alt ölçütün ağırlığı hesaplanmış ve alt ölçüt ağırlığı bağlı bulunduğu ana ölçütün ağırlığı ile çarpılarak alt ölçütün global ağırlığı elde edilmiştir. Alt ölçütlerin global ağırlıkları Çizelge 2.13’te, bu ağırlıkları gösteren pasta grafiği ise Şekil 2.3’te verilmiştir.

Çizelge 2.13. Alt ölçütlerin ağırlıkları

Ölçüt	Alt Ölçüt	Global Ağırlık	Global Ağırlık (%)
Ürün	Maliyet	0,176	%18
Ürün	Kalite	0,147	%15
Hizmet	Temin Süresi	0,139	%14
Firma	Coğrafi Konum	0,104	%10
Hizmet	Garanti ve Satış Sonrası Destek	0,082	%8
Ürün	Tedarik Kapasitesi	0,073	%7
Firma	Saygınlık ve Referanslar	0,046	%5
Firma	Finansal İstikrar	0,044	%4
Ürün	Ürün Özelleştirme Yeteneği	0,043	%4
Hizmet	İletişim ve Bilgi Sistemleri	0,041	%4
Hizmet	Eğitim ve Sürekli İyileştirme	0,032	%3
Firma	Özel Teknolojiler ve Patentler	0,029	%3
Hizmet	Çevresel ve Sosyal Sorumluluk	0,024	%2
Firma	Yönetim ve Organizasyon	0,020	%2



Şekil 2.3. Alt ölçütlerin ağırlıklarını gösteren pasta grafiği

2.6. Tedarikçilerin Değerlendirilmesi

Ölçüt ve ölçüt ağırlıkları belirlendikten sonra tedarikçi seçimi sürecine geçilmiştir. Firmanın çalışabileceği 4 adet tedarikçi mevcuttur. Bunlar Tedarikçi A, Tedarikçi B, Tedarikçi C ve Tedarikçi D olarak isimlendirilmiştir. Daha sonra firma yetkililerinden bu dört tedarikçiyi 14 alt ölçütün her biri açısından değerlendirmeleri istenmiştir. Değerlendirme sürecinde yine Çizelge 2.2’de verilen sözlü ifadelerin kullanılması istenmiştir. Bir satın alma yetkilisinin yaptığı örnek değerlendirmeler Çizelge 2.14 ve Çizelge 2.15’te verilmiştir. Satın alma yetkilisi tedarikçileri Çizelge 2.14’te maliyet alt ölçütüne göre Çizelge 2.15’te ise kalite alt ölçütüne göre değerlendirmiştir.

Çizelge 2.14. Maliyet alt ölçütü açısından tedarikçilerin değerlendirilmesi

	A	B	C	D
A	E	FS	SW	SS
B	FW	E	SW	FW
C	SS	SS	E	SS
D	SW	FS	SW	E

Çizelge 2.15. Kalite alt ölçütü açısından tedarikçilerin değerlendirilmesi

	A	B	C	D
A	E	SW	SS	SW
B	SS	E	FS	SS
C	SW	FW	E	SW
D	SS	SW	SS	E

Bu değerlendirme işlemi her bir uzman tarafından 14 alt ölçüt açısından yapılmıştır. Daha sonra ilk aşamada olduğu gibi tutarlılıklar hesaplanmış tutarlılıklarda bir problem görülmeince de uzmanların görüşlerinin geometrik ortalaması alınarak uzmanların 14 farklı alt ölçütün her biri için yaptığı ortak değerlendirme elde edilmiştir. Sonrasında yine önceki aşamada olduğu gibi Tip-2 bulanık ikili karşılaştırma matrisleri oluşturulmuş ve ağırlıklar hesaplanmıştır. 14 alt ölçütün her biri için uzmanların ortak görüşlerine göre belirlenen tedarikçi ağırlıkları Çizelge 2.16’da verilmiştir.

Çizelge 2.16. Uzmanların ortak görüşüne göre tedarikçilerin her bir ölçüt için ağırlıkları

Alt Ölçüt	Alt Ölçüt Ağırlığı	Tedarikçi A	Tedarikçi B	Tedarikçi C	Tedarikçi D
Maliyet	0,18	0,3	0,09	0,42	0,19
Kalite	0,15	0,15	0,45	0,17	0,23
Temin Süresi	0,14	0,29	0,51	0,11	0,09
Coğrafi Konum	0,10	0,32	0,27	0,23	0,18
Garanti ve Satış Sonrası Destek	0,08	0,19	0,41	0,23	0,17
Tedarik Kapasitesi	0,07	0,08	0,32	0,44	0,16
Saygınlık ve Referanslar	0,05	0,21	0,17	0,36	0,26

Finansal İstikrar	0,04	0,28	0,21	0,26	0,25
Ürün Özelleştirme Yeteneği	0,04	0,12	0,37	0,35	0,16
İletişim ve Bilgi Sistemleri	0,04	0,27	0,23	0,29	0,21
Eğitim ve Sürekli İyileştirme	0,03	0,17	0,46	0,23	0,14
Özel Teknolojiler ve Patentler	0,03	0,29	0,18	0,21	0,32
Çevresel ve Sosyal Sorumluluk	0,02	0,57	0,12	0,16	0,15
Yönetim ve Organizasyon	0,02	0,23	0,24	0,32	0,21

Son olarak, Çizelge 2.16'daki tedarikçi ağırlıkları her bir alt ölçütün önem derecesiyle çarpılmış ve her bir tedarikçinin toplam ağırlığı elde edilmiştir. Bu işlem de Çizelge 2.17'de özetlenmiştir.

Çizelge 2.17. Tedarikçilerin genel ağırlıkları

Alt Ölçüt	Tedarikçi A	Tedarikçi B	Tedarikçi C	Tedarikçi D
Maliyet	0,0529	0,0159	0,0741	0,0335
Kalite	0,0221	0,0662	0,0250	0,0338
Temin Süresi	0,0403	0,0708	0,0153	0,0125
Coğrafi Konum	0,0325	0,0274	0,0234	0,0183
Garanti ve Satış Sonrası Destek	0,0156	0,0338	0,0189	0,0140
Tedarik Kapasitesi	0,0058	0,0233	0,0320	0,0116
Saygınlık ve Referanslar	0,0097	0,0079	0,0166	0,0120

Finansal İstikrar	0,0124	0,0093	0,0115	0,0111
Ürün Özelleştirme Yeteneği	0,0053	0,0162	0,0154	0,0070
İletişim ve Bilgi Sistemleri	0,0116	0,0099	0,0125	0,0090
Eğitim ve Sürekli İyileştirme	0,0052	0,0142	0,0071	0,0043
Özel Teknolojiler ve Patentler	0,0082	0,0051	0,0059	0,0090
Çevresel ve Sosyal Sorumluluk	0,0142	0,0030	0,0040	0,0037
Yönetim ve Organizasyon	0,0046	0,0047	0,0063	0,0042
Toplam Ağırlık	0,240	0,308	0,268	0,184
Toplam Ağırlık (%)	%24	%31	%27	%18

Nitel ve nicel tüm ölçütlerin değerlendirilmesi ve uzmanların ortak görüşleriyle B tedarikçisinin en iyi tedarikçi olduğu sonucuna ulaşılmıştır ve firmaya B tedarikçisi ile çalışması önerilmiştir.

3. AĞ TASARIMI İÇİN KESİN ÇÖZÜM YAKLAŞIMLARI

Bu bölümde tedarik zinciri ağı tasarımı için geliştirilmiş çözüm yaklaşımlarına ve sonrasında da tez kapsamında geliştirilen matematiksel modele ve hesaplama sonuçlarına yer verilmiştir.

3.1. Literatür Taraması

Tedarik zinciri ağı tasarımı problemi için literatürde pek çok çalışma mevcuttur fakat bu çalışmalar kapsamında kurulan modellerin geneli tek amaçlıdır [14, 15, 16] ve bu amaç çoğu çalışmada maliyetin enküçüklenmesidir.

Tek amaçlı modellere örnek olarak Tsiakis ve ark. [14] tarafından geliştirilen model verilebilir. Bu model tedarikçileri, üretim alanlarını, depoları, dağıtım merkezlerini ve müşterileri içeren çok aşamalı ve tek amaçlı bir tedarik zinciri ağı modelidir. Yaptıkları çalışmayla araştırmacılar depoların ve dağıtım merkezlerinin sayı kapasite ve yerini belirleyerek toplam tedarik zinciri maliyetini enküçüklemeyi amaçlamışlardır.

Guillén ve ark. [17] bir tedarik zincirinin ekonomik hedefleri gerçekleştirirken aynı zamanda müşteri memnuniyetini de sağlaması gerektiğini belirterek sistemin doğasındaki çok amaçlılığı ifade etmiştir. Yani bir tedarik zinciri ağının tek amacı maliyet olmamalı ayrıca müşterilerin memnuniyetine yönelik bazı amaçları da barındırmalıdır. Literatürde az sayıda da olsa iki veya daha fazla amacın bulunduğu tedarik zinciri ağı tasarımı modelleri de mevcuttur. Pazhami ve ark. [18] iki amaçlı bir tedarik zinciri ağı modeli kurmuşlardır. Kurdukları modelin bir amacı toplam tedarik zinciri maliyetini enküçüklemekken bir diğer amacı da hizmet seviyesini enbüyüklemektir. Hizmet seviyesini ölçmek içinse AHP, TOPSIS gibi çok ölçütlü karar verme tekniklerinden yararlanmışlardır. Her bir depo ve melez tesis için hizmet seviyesini ifade eden bir etkinlik puanı belirlemişler ikinci bir amaç olarak da kullanılan tesisler için etkinlik puanlarının toplamını enbüyüklemeye çalışmışlardır. Böylece aslında modeli etkinlik değeri yüksek melez tesisler ve depoları kullanmaya zorlamışlardır. Bu da hizmet seviyesinin artmasını sağlayıcı bir amaç olmuştur. Ayrıca araştırmacılar çalışmalarında hedef programlamanın farklı varyasyonlarını kullanmışlar ve farklı varyasyonlardan elde edilen çözümleri

kıyaslamışlardır. Wang ve ark. [19] yaptıkları çalışmada önemli fakat çok az odaklanılan bir konuyu incelemişler ve bir yeşil tedarik zinciri ağı modeli geliştirmişlerdir. Modellerinin bir amacı toplam tedarik zinciri ağı maliyetinin enküçüklenmesiyken bir diğer amacı da çevresel etkidir. Bu iki amaç birbiri ile çeliştiği için geliştirdikleri model ile bu iki amaç arasındaki ödünleşmeyi sağlamışlardır. Hiremath ve ark. [20] üç amaçlı bir tedarik zinciri ağı modeli geliştirmiştir. Modelin amaçları toplam maliyeti enküçükmek, doluluk oranını enbüyüklemek ve kaynak kullanım oranını enbüyüklemektir. Üretim alanları, merkezi dağıtım merkezleri, bölgesel dağıtım merkezleri ve müşteriler çalışmalarının kapsamını oluşturmaktadır.

3.2. Tez Kapsamında İncelenen Firmaya Dair Bilgiler

Tez kapsamında bir firma için çok amaçlı bir tedarik zinciri ağı modeli geliştirilmiştir. İlk bölümde kısaca bahsedildiği üzere ele alınan firma, Bilecik merkezli bir üretim firmasıdır. Türkiye'nin neredeyse bütün illerinde müşterilere sahiptir. Ürünler Bilecik'teki fabrikada üretilmekte ve depolara gönderilmektedir. Daha sonra depolardan talepler doğrultusunda müşterilere iletilmektedir. Fabrikadan müşteriye doğrudan taşıma şirket politikası gereği yapılmamaktadır. Fabrikadan depoya gönderilecek olan miktar belirlenirken söz konusu depoya atanan müşterilerin aylık toplam talepleri dikkate alınmaktadır. Yine şirket politikası gereği, toplamda en fazla üç şehirde depo açılması uygun görülmektedir. Firmanın şehirler bazında aylık ortalama talepleri ve söz konusu şehirde kiralanacak olan bir deponun aylık kira ve işletme giderleri (TL) ilk 10 şehir için Çizelge 3.1'de verilmiştir. Çizelgenin tamamı ise Ek 1'de verilmiştir. İller arası mesafeler Karayolları Genel Müdürlüğü'nün web sitesinden [21] elde edilmiştir.

3.3. Amaçların Belirlenmesi

Firma tedarik zinciri ağını tasarlarlarken temel olarak iki amacı göz önünde bulundurmak istemektedir. Bu iki amaçtan ilki toplam tedarik zinciri maliyetinin enküçüklenmesidir. Toplam tedarik zinciri maliyeti fabrikadan depolara olan taşımaların maliyeti, depolardan müşterilere olan taşımaların maliyeti ve açılan

depoların aylık kira ve işletme giderlerinden oluşmaktadır. Burada taşıma maliyetlerinin maliyet fonksiyonuna ciddi bir etkisinin olduğu söylenebilir. Bu yüzden elde edilen sonucun güvenilirliği ancak taşıma maliyetlerinin doğru hesaplanmasıyla sağlanabilir. Bu da taşıma maliyetlerinin hesaplanmasına odaklanmayı zorunlu kılmaktadır.

Çizelge 3.1. İlk 10 şehir için şehir talepleri ve depo açma giderleri

Şehir Plaka Kodu	Şehir Adı	Aylık Talep (Palet)	Aylık Depo Kiralama ve İşletme Giderleri (TL)
01	ADANA	551	28000
02	ADİYAMAN	2	24000
03	AFYON	37	27000
04	AĞRI	4	19000
05	AMASYA	29	26000
06	ANKARA	863	45000
07	ANTALYA	1190	33000
08	ARTVİN	137	23000
09	AYDIN	11	26000
10	BALIKESİR	239	22000

Literatürdeki pek çok çalışmada taşıma maliyetleri uzaklık ve ürün miktarının çarpımı olarak kabul edilmektedir. Bu bağlamda, miktar sabit tutulduğunda taşıma maliyetinin uzaklıkla doğru orantılı olarak değiştiği söylenebilir. Yani bir ürünü x birim taşımanın maliyeti K ise aynı ürünü $2x$ birim taşımanın maliyetinin $2K$ olduğu varsayılır. Fakat firmadan aldığımız veriler bu varsayımın doğru olmadığını ortaya koymuştur. Çizelge 3.2’de firmadan alınmış, çeşitli uzaklıklara bağlı birim taşıma maliyetleri yer almaktadır.

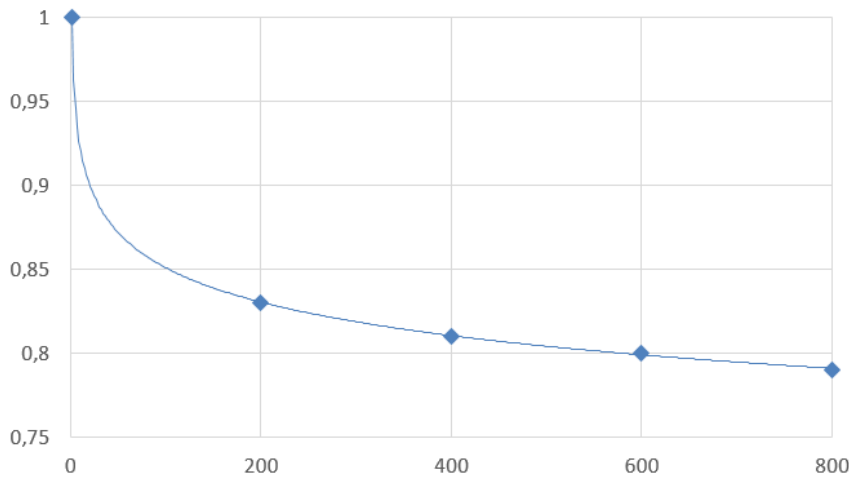
Çizelge 3.2. Uzaklığa bağlı birim taşıma maliyeti

Uzaklık (km)	1	200	400	600	800
Birim Maliyet (palet/km)	1,00	0,83	0,81	0,80	0,79

Görüldüğü gibi birim maliyet aslında uzaklıkla doğrusal olarak değişmemektedir. Örnek vermek gerekirse 1 palet ürünü 200 km taşımanın maliyeti 166 TL iken $(1 \times 200 \times 0,83)$ aynı ürünü 400 km taşımanın maliyeti 324 TL $(1 \times 400 \times 0,81)$ olarak bulunur. Doğrusal olarak ele alınsaydı bu değer 324 değil 332 TL olarak bulunurdu. Bu da taşıma maliyetinin yanlış hesaplanmasına yol açardı. Aslında burada firma verileriyle teyit ettiğimiz bu gerçek, literatürde *mesafe ekonomisi* olarak bilinen kavramdır. Mesafe ekonomisi kavramına göre bir ürünün taşıma mesafesi arttıkça ürünün birim maliyeti belli miktarda azalmaktadır. Bu sebeple, taşıma maliyetlerini doğru hesaplamak amacıyla modelde mesafe ekonomisi kavramı da dikkate alınmıştır.

Burada çözüme kavuşturulması gereken bir diğer sorun da farklı değerler için birim maliyetin değerinin ne olacağıdır. Çünkü Çizelge 3.2’de görüldüğü gibi firma verileri sadece beş mesafe değeri için birim maliyeti vermektedir. Fakat iki şehrin arasındaki mesafe çok farklı değerler alabilir. Örnek olarak, iki şehir arasındaki uzaklık 173 km veya 981 km olabilir. Bu durumda birim maliyet ne olacaktır? Bu sorunun cevabını verebilmek için hata karelerini enküçükleyen basit bir model kurulmuştur.

Kurulan model sonucunda yukarıdaki verilerin $c(x) = x^{-0,035}$ eğrisiyle çok büyük oranda uyumlu olduğu saptanmıştır. Aşağıda, Şekil 3.1’de bu fonksiyonun grafiği verilmiştir. Çizelge 3.3’te de fonksiyonun aldığı değerlerle firma verileri karşılaştırılmıştır.



Şekil 3.1. Birim maliyetin uzaklığa bağlı değişimi

Çizelge 3.3. Birim maliyet fonksiyonuyla gerçek değerlerin karşılaştırması

Uzaklık	Birim Maliyet (palet/km)	$c(x) = x^{-0,035}$ değeri
1	1	1
200	0,83	0,831
400	0,81	0,811
600	0,80	0,799
800	0,79	0,791

Çizelge 3.3'teki veriler incelendiğinde x mesafe olmak üzere $c(x) = x^{-0,035}$ fonksiyonunun firmanın sağladığı verilere çok büyük oranda uyum sağladığı görülmektedir. Bu bağlamda $c(x) = x^{-0,035}$ fonksiyonu uzaklığa bağlı birim maliyet fonksiyonu olarak kullanılacaktır.

Firmanın ikinci amacı ise depoların müşterilere olan uzaklığının enküçüklenmesi, yani depoların mümkün olduğunca müşterilere yakın olmasının sağlanmasıdır. Depoların müşterilere yakınlığı pek çok avantaj sağlamaktadır. Burada bu avantajlardan bazılarını yer vereceğiz.

- Müşterilere kısa temin süreleriyle ürün sağlama: Ürünler depolarda toplanıp talepler doğrultusunda müşterilere iletiğinden depoların müşteriye yakınlığı ürünlerin kısa temin süreleri içinde müşterilere iletebilmesine olanak tanıyacaktır. Böylece müşteri memnuniyeti oluşturulacak ve zamanla firmanın saygınlığı/tercih edilirliliği artacaktır.
- Müşterilerle kapsamlı biçimde iletişim kurabilme: Depoların müşterilere yakın olması aslında firmanın müşterilere yakın olması anlamına gelmektedir. Bu sayede firma personeli müşterilerle telefon e-posta gibi kanallara ek olarak, kolayca toplantılar ve yüz yüze ziyaretler yapabilecektir. Benzer şekilde müşteriler de ihtiyaç duyduğunda firma personeline kolayca ulaşabilecektir. Bu karşılıklı kolay ulaşılabilirlik de yine müşterilerin memnuniyetini artırıcı bir unsurdur ve firmayı tercih etmelerinin bir sebebi olabilir.
- Müşterinin içinde bulunduğu coğrafi ve ekonomik koşulları kolayca analiz edebilme: Firma personeli müşteriye yakın olduğu için müşterinin coğrafi ve ekonomik koşullara bağlı durumunu kolayca analiz edebilecektir.

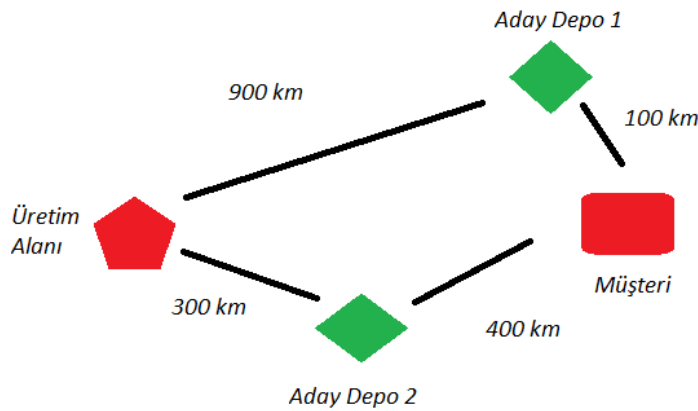
Böylece söz konusu şehirdeki ve yakın şehirlerdeki fırsatları, tehditleri önceden görüp önlem alabilmesi, şirket bünyesinde yeni promosyonlar, özel fırsatlar vs. hazırlaması mümkün olacaktır.

Yukarıdaki sebeplerden dolayı firma tedarik zincirini tasarlariken ikinci amaç olarak da açılan depoların müşterilere yakın olması amacını belirlemiştir.

Bir matematiksel modelde çok amaçlılıktan bahsedebilmek için amaçların mutlaka birbiri ile çelişmesi gerekmektedir. Birbiri ile çelişmeyen (ikisi de aynı yöne giden) amaçlardan oluşan bir modelde çok amaçlılıktan ve ödünleşme noktası bulmaktan bahsedilemez. Çünkü çelişmeyen amaçlar zaten aynı yönlüdür ve birisini optimize etmek diğerini optimize etmekle aynı şeydir. Örnek olarak, bir birimi 10 TL'den satılan bir ürün için bir amaç kazanılan paranın enbüyüklenmesi, bir diğer amaç da satılan ürün miktarının enbüyüklenmesi olsun. Bu iki amaç birbiri ile çelişmediği için (aynı yönlü olduğu için) birinin optimum çözümü diğerinin de optimum çözümüdür. Dolayısıyla bu örnek problem için çok amaçlılıktan ve ödünleşme noktasından bahsedilemez.

Model kapsamında ele aldığımız amaçlar da ilk başta aynı yönlü gibi görülebilir. Fakat aslında bu iki amaç birbiri ile çelişmektedir. Aşağıda basit bir örnek ile amaçların çeliştiği gösterilmiştir.

Şekil 3.2'de görüldüğü gibi bir palet ürünün tek bir müşteriye aday depolardan biri ile iletileceğini varsayalım. Aday depoların maliyetleri aynı olsun. Bu durumda ürün farklı amaçlara göre Aday Depo 1 veya Aday Depo 2 aracılığıyla müşteriye iletir.



Şekil 3.2. Bir üretim alanı iki aday depo ve bir müşteriden oluşan tedarik zinciri

Amaç olarak toplam tedarik zinciri maliyetinin minimizasyonu seçilirse, aday Depo 1 aracılığıyla iletilmesi durumunda maliyet $1.900.900^{-0,035} + 1.100.100^{-0,035} = 794,4$ TL olacaktır. Aday depo 2 seçilirse bu maliyet benzer şekilde 570,04 TL olacaktır. Yani toplam tedarik zinciri maliyetinin enküçüklenmesi amacına göre en iyi çözüm Aday Depo 2'dir. Diğer taraftan amaç olarak depoların müşteriye yakınlığı amacı seçilirse bu durumda $100 \text{ km} < 400 \text{ km}$ olduğu için Aday Depo 1'in seçilmesi uygun olacaktır.

Bu basit örnekte görüldüğü üzere bu iki amaç birbiri ile çelişmektedir. Dolayısıyla kurulacak olan modelde çok amaçlılıktan söz edilebilir ve modelden beklenen de bu iki amaç için bir ödünleşme noktasının bulunmasıdır.

Bu iki amacın farklı ağırlıklarda ödünleşme noktasının bulunması için ağırlıklı toplam yöntemine göre skalerleştirme ve hedef programlama yönteminden yararlanılmıştır. Diğer bir ifadeyle, bu iki amaç modelde 2 ayrı hedef olarak ele alınmış ve hedeflerden pozitif yönlü sapmaların toplamı enküçüklenmiştir.

3.4. Matematiksel Modelin Geliştirilmesi

Kümeler

H : Üretim alanı ($h = 1$)

I : Depolar ($i = 1 \dots 81$)

J : Müşteriler ($j = 1 \dots 81$)

Parametreler

m_i : Üretim alanı ile i şehrine açılan depo arasındaki uzaklık

n_{ij} : i şehrine açılan depo ile j şehrindeki müşteri arasındaki uzaklık

d_j : j şehrindeki müşterinin aylık ortalama talebi

c_i : i şehrine açılan deponun aylık kira ve işletme giderleri

K : Depo kapasitesi

T_1, T_2 : Hedef değerleri

Karar Değişkenleri

x_i : Üretim alanından i şehrine açılan depoya gönderilen ürün miktarı

s_{ij} : i şehrine açılan depodan j şehrindeki müşteriye gönderilen ürün miktarı

y_i : (ikili değişken) Eğer i şehrine depo açılırsa 1 açılmazsa 0

$g_1^-, g_1^+, g_2^-, g_2^+$: Hedeflerden pozitif ve negatif yönlü sapmalar

Bu ifadelere bağlı olarak öncelikle iki amaçlı bir matematiksel model kurulmuştur. Daha sonra iki amaçlı modelin amaçları birer hedef olarak ifade edilmiş ve hedef programlama modeli kurulmuştur. İki amaçlı model izleyen şekildedir.

$$\min z_1 = \sum_{i=1}^{81} m_i^{0,965} \cdot x_i + \sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij}^{0,965} \cdot s_{ij} + \sum_{i=1}^{81} y_i \cdot c_i \quad (3.1)$$

$$\min z_2 = \sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij} \cdot s_{ij} \quad (3.2)$$

s. t.

$$\sum_{j=1}^{81} s_{ij} \leq K \cdot y_i \quad \forall i \quad (3.3)$$

$$x_i \leq K \cdot y_i \quad \forall i \quad (3.4)$$

$$\sum_{i=1}^{81} s_{ij} \geq d_j \quad \forall j \quad (3.5)$$

$$x_i \geq \sum_{j=1}^{81} s_{ij} \quad \forall i \quad (3.6)$$

$$\sum_{i=1}^{81} y_i \leq 3 \quad (3.7)$$

$$s_{ij}, x_i \geq 0 \text{ ve tamsayı}, \quad y_i \in \{0,1\} \quad (3.8)$$

Yukarıdaki modelin hedef programlama modeline dönüştürülmesi sürecinde kısıtlarda hiçbir değişiklik yapılmamış sadece amaçlar hedefler olarak ifade edilmiş ve hedeflerden pozitif yönlü sapmaların toplamının enküçüklendiği bir amaç fonksiyonu kullanılmıştır. Hedef programlama modelindeki her bir eşitliğin açıklaması izleyen şekildedir.

Hedefler

Maliyet Hedefi: Önceki bölümde taşıma maliyetleri için birim maliyetin (x değeri uzaklık olmak üzere) uzaklığa bağlı olarak $y = x^{-0,035}$ şeklinde olduğu gösterilmişti. Bu bağlamda i şehrine açılan depo ile j şehrindeki müşteri arasındaki taşımanın maliyetini $n_{ij}^{-0,035} \cdot n_{ij} \cdot s_{ij}$ şeklinde yazmak mümkündür. $n_{ij}^{-0,035} \cdot n_{ij}$

ifadesi $n_{ij}^{0,965}$ olarak yazılırsa bu durumda taşıma maliyeti $n_{ij}^{0,965} \cdot s_{ij}$ olarak yazılabilir. Aynı durum üretim alanından depolara yapılan taşıma için de geçerlidir. Üretim alanından depolara olan taşıma da benzer şekilde $m_i^{0,965} \cdot x_i$ şeklinde yazılabilir. Bu bağlamda maliyet hedefi aşağıdaki gibi yazılabilir.

$$\sum_{i=1}^{81} m_i^{0,965} \cdot x_i + \sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij}^* \cdot s_{ij} + \sum_{i=1}^{81} y_i \cdot c_i + g_1^- - g_1^+ = T_1 \quad (3.9)$$

Burada T_1 değeri problem sadece bu hedefe göre çözdürüldüğünde (ikinci hedef dikkate alınmadığında) elde edilen çözümdür. Eşitliğin ilk kısmı üretim alanından depolara yapılan taşımaların maliyetidir, ikinci kısmı depolardan müşterilere yapılan taşımaların maliyetidir ve son kısmı da açılan depoların aylık kira ve işletme giderleridir.

Depoların Müşterilere Yakın Olması Hedefi: Depoların müşterilere yakın olması hedefi aşağıdaki gibi yazılabilir.

$$\sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij} \cdot s_{ij} + g_2^- - g_2^+ = T_2 \quad (3.10)$$

Burada T_2 değeri problem sadece ikinci hedefe göre çözdürüldüğünde (birinci hedef dikkate alınmadığında) elde edilen çözümdür. $n_{ij} \cdot s_{ij}$ çarpımının küçük çıkması depodan müşteriye yapılan s_{ij} taşımalarının yakın şehirler arasında yapılması ile (n_{ij} uzaklığının küçük olması) ile mümkündür. Bu da modeli müşterilere yakın yerlere depo açıp taşımaları yakın yerlerden yapmaya zorlayacaktır.

Kısıtlar

$$\sum_{j=1}^{81} s_{ij} \leq K \cdot y_i \quad \forall i \quad (3.3)$$

3.3 kısıtı ile herhangi bir depodan bir şehre ürün göndermek için öncelikle o deponun açılmış olması sağlanmaktadır. Eğer i . şehre depo açılmazsa oradan herhangi bir ürün gönderimi olmayacaktır fakat depo açılırsa müşterilere oradan toplamda en fazla depo kapasitesi kadar ürün gönderimi mümkün olabilecektir.

$$x_i \leq K \cdot y_i \quad \forall i \quad (3.4)$$

3.4 kısıtı ile de üretim alanından herhangi bir depoya ürün göndermek için o deponun açılmış olması sağlanmaktadır. Eğer i şehrine depo açılmışsa o depoya fabrikadan toplamda en fazla depo kapasitesi kadar ürün gönderilebilir, açılmamışsa o depoya fabrikadan ürün gönderilmez.

$$\sum_{i=1}^{81} s_{ij} \geq d_j \quad \forall j \quad (3.5)$$

3.5 kısıtı ile depolardan herhangi bir müşteriye gönderilen toplam ürün miktarının en az o müşterinin talebi kadar olması sağlanmaktadır.

$$x_i \geq \sum_{j=1}^{81} s_{ij} \quad \forall i \quad (3.6)$$

3.6 kısıtı ile herhangi bir depoya üretim alanından gönderilen ürün miktarının en az o depoya atanan müşterilerin toplam talepleri kadar olması sağlanmaktadır.

$$\sum_{i=1}^{81} y_i \leq 3 \quad (3.7)$$

3.7 kısıtı ile toplamda en fazla 3 tane depo açılması sağlanmaktadır.

$$s_{ij}, x_i \geq 0 \text{ ve tamsayı, } y_i \in \{0,1\} \quad (3.8)$$

3.8’de yer alan kısıtlar işaret ve tamsayı olma kısıtlarıdır.

Amaç Fonksiyonu

Önceki bölümde bahsedildiği gibi amaçlar modelde hedef olarak ifade edilmiştir. Her iki hedef de minimizasyon olduğu için amaç fonksiyonu hedeflerden pozitif yönlü sapmaların enküçüklenmesi olacaktır.

$$\min z = w_1 \cdot g_1^+ + w_2 \cdot g_2^+ \quad (3.11)$$

Burada w_1 değeri ve w_2 değeri hedeflerin ağırlıklarıdır. Farklı ağırlık değerlerine göre farklı Pareto çözümler elde edebilmek mümkün olacaktır.

Özetle, kurulan model izleyen şekildedir.

$$\begin{aligned} \min z &= w_1 \cdot g_1^+ + w_2 \cdot g_2^+ \\ \text{s. t} \end{aligned} \quad (3.11)$$

$$\sum_{i=1}^{81} m_i^{0,965} \cdot x_i + \sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij}^{0,965} \cdot s_{ij} + \sum_{i=1}^{81} y_i \cdot c_i + g_1^- - g_1^+ = T_1 \quad (3.9)$$

$$\sum_{i=1}^{81} \sum_{j=1}^{81} n_{ij} \cdot s_{ij} + g_2^- - g_2^+ = T_2 \quad (3.10)$$

$$\sum_{j=1}^{81} s_{ij} \leq K \cdot y_i \quad \forall i \quad (3.3)$$

$$x_i \leq K \cdot y_i \quad \forall i \quad (3.4)$$

$$\sum_{i=1}^{81} s_{ij} \geq d_j \quad \forall j \quad (3.5)$$

$$x_i \geq \sum_{j=1}^{81} s_{ij} \quad \forall i \quad (3.6)$$

$$\sum_{i=1}^{81} y_i \leq 3 \quad (3.7)$$

$$s_{ij}, x_i \geq 0 \text{ ve tamsayı}, \quad y_i \in \{0,1\} \quad (3.8)$$

3.5. Hesaplama Sonuçları

Modelin firma verileri ile çözdürülmesi için GAMS optimizasyon programından yararlanılmıştır. Modelin çözdürüldüğü bilgisayar Intel i7 işlemci, 12 GB RAM, 4 GB harici ekran kartı ve Windows 8.1 Pro işletim sistemine sahip bir bilgisayardır. Modelin amaç fonksiyonu $\min z = g_1^+$ şeklinde düzenlenip çözdürüldüğünde z değeri 4428661 olarak bulunmuş ve bu değer T_1 parametresi olarak belirlenmiştir. Benzer şekilde $\min z = g_2^+$ amaç fonksiyonuna sahip modelin çözümü sonucu elde edilen $z = 2750558$ değeri de T_2 parametresi olarak belirlenmiştir. Depo kapasitesi firma görüşleri doğrultusunda tek bir depo tüm talebi karşılayabilecek şekilde belirlenmiştir. Böylece aslında minimum depo sayısının 1 olabilmesi ve modelin 1 ile 3 arasında bir sayıda depo açabilmesi olanağı sağlanmıştır. Modelin her iki hedefine de eşit (0,5) ağırlık verilmiş ve çözdürülmüş, çözüm sonucu elde edilen veriler Çizelge 3.4'te özetlenmiştir.

Çizelge 3.4 incelendiğinde birinci hedeften pozitif yönlü sapmanın (g_1^+) 177936 olduğu görülmektedir. Bu da aslında daha hızlı bir hizmet vermek ve müşterilere daha yakın olmak için katlanılan maliyettir. Benzer şekilde ikinci hedeften pozitif yönlü sapma da (g_2^+) 691079 olarak bulunmuştur. Bu da daha düşük maliyetle hizmet verebilmek için temin süresinden ve müşterilere yakınlıktan bir miktar ödün verildiğini göstermektedir.

Modelin çözümü yukarıda özellikleri verilen bilgisayar ile 583,91 saniye sürmüştür. Bu da yaklaşık 10 dakikalık bir çözüm süresine tekabül etmektedir. Problem NP-Hard (çözümü polinom zamanlı olmayan-zor problemler) sınıfından

bir problem olduğu için problem boyutundaki çok küçük bir artışın bile problemin çözüm süresini ciddi anlamda artırabileceği bilinen bir gerçektir.

Çizelge 3.4. Çözüm sonucu elde edilen veriler

Çözüm Süresi (sn)	z değeri	g_1^+ değeri	g_2^+ değeri	Açılan Depolar	Depo Kapasiteleri
583,91	434507,5	177936	691079	Adana	1490
				Bilecik	9460
				Samsun	1517

Çizelge 3.4 incelendiğinde modelin izin verilen maksimum sayıda depoyu açtığı görülmektedir. Bunun nedeni aslında depo sayısı arttıkça 2. hedef olan depoların müşteriye yakınlığının daha iyi sağlanabilmesidir. Depo sayısı ne kadar fazla olursa müşterilere daha yakın olabilmek o kadar mümkün olabilecektir. Bu yüzden model izin verilen maksimum sayı olan 3 depoyu açmayı uygun bulmuştur.

Model Türkiye'nin kuzeyinde bulunan Samsun ile güneyinde bulunan Adana'ya depo açarak aslında kuzey ve güneydeki illere hızlı ve kısa temin süreleriyle hizmet verebilme olanağını doğurmuştur. Ayrıca, modelin diğer bir depo yeri olarak Bilecik'i tercih etmesi de gayet mantıklıdır. Zira birincisi üretim alanı Bilecik'tedir, ikincisi de Bilecik şehri İstanbul, Kocaeli, Bursa İzmir, Antalya gibi pek çok büyük şehre yakın sayılabilecek, depo açma maliyeti diğer şehirlere göre nispeten daha düşük olan ve devlet teşvikinin sağlandığı bir şehirdir. Geçmişten günümüze pek çok firmanın bu olumlu yönlerinden dolayı Bilecik ilini üretim alanı ya da depo alanı olarak tercih ettiği bilinmektedir.

Açılan depolar, depolardan yapılan aylık toplam akışlar, depoya atanan müşteriler ve müşterilerin depoya uzaklıkları sırasıyla Çizelge 3.5, Çizelge 3.6 ve Çizelge 3.7'de verilmiştir.

Çizelge 3.5 incelendiğinde Adana şehrine açılan depoya toplamda 21 şehrin atandığı görülmektedir. Bu şehirler içerisinde en yüksek talebe sahip olan şehir deponun açıldığı şehir olan Adana şehridir. Adana şehrinin talebi depoya atanan şehirlerin toplam talebinin 1/3'ünden fazlasını oluşturmaktadır. Bu da deponun, talebin 1/3'ünden fazlasına çok hızlı bir hizmet verildiğini göstermektedir. Depoya

atanan en yüksek talepli ikinci müşteri Hatay'dır ve Hatay şehrinin talebi toplam depo talebinin yaklaşık 1/5'ini oluşturmaktadır. Deponun Hatay şehrine uzaklığı 191 km'dir ve bu da Hatay şehrine hızlı hizmet verilebilme olanağını sağlamaktadır. Benzer durum üçüncü en yüksek talebe sahip olan ve deponun aylık akışının yaklaşık 1/7'sini oluşturan İçel şehri için de geçerlidir. Deponun İçel şehrine uzaklığı sadece 69 km'dir bu da depodan İçel şehrine çok hızlı hizmet verilmesini sağlayacaktır. Genel olarak deponun müşterilerine uzaklığı incelendiğinde deponun müşterilere ortalama uzaklığının 431,48 km olduğu fakat talep bazlı ağırlıklı ortalama uzaklığının 186,74 km olduğu görülmektedir. Buradan, deponun özellikle yüksek talepli şehirlere yakın olacak şekilde kurulduğu anlaşılmaktadır.

Çizelge 3.5. Adana deposu hakkında bilgiler

Depo Bilgileri	Bu Depoya Atanan Şehirler	Depoya Uzaklıklar (km)	Depodan Gönderilen Aylık Ürün Miktarı
Deponun Açıldığı Şehir	Adana	0	551
Adana	Adıyaman	329	2
	Bitlis	727	2
	Diyarbakır	518	41
Aylık Toplam Akış 1490 palet	Elazığ	486	77
	Gaziantep	205	89
	Hakkari	900	1
Bu depoya Atanan Toplam Müşteri Sayısı 21	Hatay	191	269
	İçel	69	197
	Malatya	388	121
	Kahramanmaraş	185	34
	Mardin	530	40
Deponun Müşterilere Ortalama Uzaklığı 431,48 km	Muş	738	2
	Niğde	205	27
	Siirt	701	2
	Şanlıurfa	342	6
	Van	895	22
Deponun Müşterilere Ağırlıklı Ortalama Uzaklığı 186,74 km	Batman	614	21
	Şırnak	711	19
	Kilis	242	8
	Osmaniye	85	5

Açılan ikinci depo olan Bilecik deposuna ilişkin bilgiler Çizelge 3.6'da yer almaktadır. Çizelge 3.6 incelendiğinde Bilecik şehrinde açılan deponun müşterilere olan ortalama uzaklığının 343,18 km olduğu görülmektedir. Deponun müşterilere olan talep bazlı ağırlıklı ortalama uzaklığının ise 315,26 km olduğu görülmektedir.

Bu iki değerin birbirine yakın olması da Bilecik şehrine açılan deponun yüksek talebe sahip şehirlerin arasında, hepsine mümkün olduğu kadar yakın olabilecek bir konumda bulunduğunu göstermektedir.

Çizelge 3.6 Bilecik deposu hakkında bilgiler

Depo Bilgileri	Bu Depoya Atanan Şehirler	Depoya Uzaklıklar (km)	Depodan Gönderilen Aylık Ürün Miktarı
Deponun Açıldığı Şehir	Afyon	210	37
Bilecik	Ankara	313	863
	Antalya	474	1190
	Aydın	523	11
Aylık Toplam Akış	Balıkesir	246	239
9460 palet	Bilecik	0	45
	Bolu	213	19
Bu depoya Atanan	Burdur	352	2
Toplam Müşteri	Bursa	95	642
Sayısı	Çanakkale	366	18
39	Çankırı	444	5
	Denizli	397	97
Deponun	Edirne	477	116
Müşterilere	Eskişehir	80	96
Ortalama Uzaklığı	Isparta	351	75
343,18 km	İstanbul	247	2870
	İzmir	417	756
Deponun	Kastamonu	459	30
Müşterilere Ağırlıklı	Kayseri	620	276
Ortalama Uzaklığı	Kırklareli	458	13
315,26 km	Kırşehir	486	5
	Kocaeli	136	648
	Konya	418	433
	Kütahya	110	43
	Manisa	381	66
	Muğla	542	439
	Nevşehir	577	37
	Sakarya	99	106
	Tekirdağ	379	71
	Uşak	249	33
	Yozgat	529	13
	Zonguldak	282	5
	Aksaray	523	18
	Karaman	531	32
	Kırıkkale	388	15
	Bartın	371	17
	Yalova	126	14
	Karabük	347	2
	Düzce	168	17

Bilecik şehrine kurulan depo Adana şehrine kurulan depo gibi yüksek talepli bir şehirde kurulmamış aksine oldukça düşük talepli bir şehirde kurulmuştur. Bunun sebebi Bilecik şehrindeki depoya atanan şehirler arasında birbirleriyle zıt konumlu pek çok yüksek talepli şehir olmasıdır. İstanbul, İzmir, Antalya ve Ankara bahsedilen yüksek talepli ve zıt konumlu şehirlerden bazılarıdır. Dolayısıyla Bilecik deposu, Adana deposu gibi bölgedeki tek yüksek talepli şehre kurulmuş bir depo değil, Batı Anadolu, Marmara ve Ege gibi bölgelerde bulunan pek çok yüksek talepli şehre hizmet vermeye çalışan yüksek kapasiteli bir depodur. Bu yüzden depolara olan ortalama uzaklıkla depolara olan ağırlıklı ortalama uzaklık Adana deposunun aksine bu depo için birbirine yakın değerlere sahiptir.

Çizelge 3.7. Samsun deposu hakkında bilgiler

Depo Bilgileri	Bu Depoya Atanan Şehirler	Depoya Uzaklıklar (km)	Depodan Gönderilen Aylık Ürün Miktarı
Deponun Açıldığı Şehir	Ağrı	742	4
Samsun	Amasya	131	29
	Artvin	566	137
	Çorum	172	17
Aylık Toplam Akış	Erzincan	444	7
1517 palet	Erzurum	561	40
	Giresun	196	25
Bu depoya Atanan	Gümüşhane	358	1
Toplam Müşteri	Kars	761	1
Sayısı	Ordu	152	135
20	Rize	407	36
	Samsun	0	614
Deponun	Sinop	163	22
Müşterilere	Sivas	338	165
Ortalama Uzaklığı	Tokat	230	48
404,55 km	Trabzon	333	215
	Tunceli	574	2
Deponun	Bayburt	436	9
Müşterilere Ağırlıklı	Ardahan	675	7
Ortalama Uzaklığı	Iğdır	852	3
203,24 km			

Son olarak, Samsun şehrine açılan depoya ilişkin veriler incelendiğinde (Çizelge 3.7) bu depoya atanan şehirler içerisinde en yüksek talebe sahip olan şehrin Samsun olduğu ve deponun bölgedeki en yüksek talebe sahip şehirde kurulduğu görülmektedir. Samsun şehrinin talebi depoya atanan şehirlerin toplam talebinin 1/3'ünden fazlasını oluşturmaktadır. Bölgede ikinci en yüksek talebe sahip olan

şehir Trabzon'dur ve Trabzon şehrinin depoya uzaklığı 333 km'dir. Bu da Trabzon şehrine hızlı bir hizmet verebilmek için uygun bir mesafedir.

Deponun müşterilere ortalama uzaklığı 404,55 km, talep bazlı ağırlıklı ortalama uzaklığı ise 203,24 km'dir. Bu da Samsun şehrinde kurulan deponun tıpkı Adana şehrinde kurulan depo gibi özellikle büyük talepli şehirlere yakın olacak şekilde kurulduğunu göstermektedir.

Burada değinilmesi gereken bir diğer konu da hedeflerden pozitif yönlü sapmaların ağırlıklarıdır. Çünkü ağırlıkların değişmesi ile elde edilen çözümler de değişebilecek, farklı Pareto çözümler bulunabilecektir. Bir döngü kurularak ağırlıklar 0,1 ile 0,9 arasında değiştirildiğinde Çizelge 3.8'deki sonuçlara varılmıştır.

Çizelge 3.8. Farklı ağırlıklara göre elde edilen çözümler

w_1	w_2	z	g_1^+	g_2^+
0,9	0,1	169896,5	78872	989117
0,8	0,2	252468,4	91883	894810
0,7	0,3	324208,6	129727	777999
0,6	0,4	383193,2	177936	691079
0,5	0,5	434507,5	177936	691079
0,4	0,6	479955,6	303117	597848
0,3	0,7	446633	716133	331133
0,2	0,8	361472,8	1084132	180808
0,1	0,9	212429,1	2124291	0

4. AĞ TASARIMI PROBLEMİ İÇİN METASEZGİSEL YAKLAŞIMLAR

4.1 Metasezgisel Yaklaşımların Özellikleri

Günümüzde sahip olduğumuz teknoloji büyük boyutlu ve özellikle NP-Hard sınıfı optimizasyon problemleri için kesin çözümü bulmakta çok sınırlı kalmaktadır. Karesel Atama Problemi, Gezgin Satıcı Problemi gibi araştırmacıların ilgi odağında olan pek çok NP-hard problemin büyük boyutlu versiyonlarının çözümü için metasezgisel yaklaşımlardan yararlanılmaktadır.

Metasezgisel yaklaşımlar sahip oldukları bazı özellikler sayesinde yerel optimumlara takılmaktan kurtulabilecek yetenekli algoritmalar. Bu algoritmalar çok kısa bir süre içerisinde oldukça kaliteli çözümler verebilir, kesin yöntemler ile çözümü bulunamayan problemleri kısa bir süre içerisinde çözüme kavuşturabilir fakat hiç bir metasezgisel algoritma optimum çözümü garanti etmez. Ayrıca hiç bir metasezgisel algoritma optimum çözüme ne kadar yaklaşıldığı konusunda bir bilgi sağlamaz.

Genel olarak bir metasezgisel algoritmanın diğerine göre daha iyi çalıştığı ve üstün olduğu da söylenemez. Bazı problem türleri için bir metasezgisel iyi sonuçlar verirken bazı başka problem türleri için başka bir metasezgisel iyi sonuçlar verebilir. Her bir metasezgisel algoritmanın kendine has bazı olumlu ve olumsuz yönleri vardır. Günümüzde araştırmacılar bir metasezgisel algoritmanın sahip olduğu olumsuz özellikleri gidermek için onu başka bir metasezgisel algoritma ile melez biçimde kullanmakta böylece daha kaliteli çözümler elde etmeyi ummaktadır.

Bir metasezgisel algoritma elde edilen çözümün optimum çözüme olan uzaklığını vermediği için metasezgisel yaklaşımlarla elde edilen çözümlerin kalitesini incelemek de ayrıca üzerinde durulması gereken bir konudur. Bu konuda da pek çok yöntem vardır. Bazı araştırmacılar küçük boyutlu ve kesin çözümü bilinen problemlerde metasezgiseli test etmekte ve metasezgiselin ürettiği çözümün optimum çözüme olan uzaklığına bakmaktadır. Bunun da mümkün olmadığı durumlarda ise araştırmacılar problem için bir alt sınır belirlemekte ve metasezgiselin ürettiği çözümün bu alt sınıra olan uzaklığına bakıp optimum çözümden olabilecek en fazla uzaklığı hesaplamaktadırlar. Örnek olarak, gezgin

satıcı problemi aslında atama probleminin özel bir halidir ve atama probleminden tek farkı alt tur engelleme kısıtının olmasıdır. Bu alt tur engelleme kısıtı sebebiyle aynı problem atama problemini olarak ele alındığında elde edilen çözüm değeri gezgin satıcı problemi olarak ele alındığında elde edilen çözüm değerinden daha küçüktür ya da bu değerler eşittir. Bu açıdan n şehirli bir gezgin satıcı problemi için metasezgisel algoritma ile bir çözüm bulunduğunda bu çözüm aynı problemin atama problemi olarak çözülmesi sonucu elde edilen çözümle karşılaştırılarak optimum çözüme olabilecek en fazla uzaklık hesaplanabilir. Bu değer olabilecek en fazla uzaklığı verdiği için bu değer üzerinden doğru bir yorum yapabilmek de çok kolay değildir. Zira metasezgisel algoritma ile bulunan çözüm optimum çözüm de olabilir. Örnek olarak, teorik alt sınır ile metasezgiselin ürettiği çözüm arasındaki fark %10 olarak bulunmuş olsun. Bu durumda tek bilinen şey optimum çözüme uzaklığın en fazla %10 olduğudur, diğer taraftan bulunan çözümün optimum çözüm olabileceği de unutulmamalıdır.

4.2. Literatür Taraması

Tedarik zinciri ağ tasarımı problemi de daha önce belirtildiği üzere NP-Hard (çözüm zamanı polinom zamanlı olmayan-zor) problemler sınıfındandır. Bu da problem boyutundaki artışların belli bir noktadan sonra problemin çözümünü imkansız kılabileceğini göstermektedir. Bu sebeple problemin çözümü için metasezgisellere dayanan çok sayıda yaklaşım geliştirilmiştir.

Costa ve ark. [22] tedarik zinciri ağlarının tasarımında kullanılmak üzere yeni ve etkin bir gösterim öne sürmüşlerdir. Bu gösterimi genetik algoritma ile birlikte probleme uygulamışlar ve yüksek kalitede çözümler elde etmeyi başarmışlardır. Yao ve Hsu [23] da çok aşamalı tedarik zincirlerinin tasarımı için genetik algorithmadan yararlanmışlardır. Easwaran ve Üster [24] kapalı çevrim tedarik zincirlerinin tasarımı için yasaklı arama ve Benders Ayırıştırması'na dayanan bir yaklaşım kullanmışlardır. Öncelikle toplam maliyeti enküçükleyecek karma tamsayılı doğrusal bir model geliştirmişler daha sonra bu modelin çözümü için farklı komşuluk yaklaşımlarını kullanan iki yasaklı arama algoritmasını melez biçimde kullanmışlardır. Araştırmacılar ayrıca özellikle büyük boyutlu problemlerde çözüm kalitesinin artırılması ve hesaplama süresinin kısaltılması için

kullandıkları yaklaşıma Benders Ayırıştırması'nı da dahil etmişlerdir. Govindan [25] ve arkadaşları çok aşamalı bir tedarik zinciri için iki amaçlı bir matematiksel model geliştirmişlerdir ve bu modeli çözebilmek için etkin bir metasezgisel önermişlerdir. İnceledikleri tedarik zinciri tedarikçiler, üretim alanları, dağıtım merkezleri, çapraz sevkiyat alanları ve perakendecilerden oluşan çok geniş kapsamlı bir tedarik zinciridir. Bu tedarik zinciri için geliştirdikleri matematiksel model toplam maliyetin ve çevreye olan olumsuz etkinin enküçüklenmesini amaçlamaktadır. Castillo-Villar ve Herbert-Acero [26] tedarik zinciri ağlarının tasarımında çok az odaklanılan bir konuya odaklanmışlar, toplam maliyeti enküçükleyen bir tedarik zinciri ağında kalite maliyetlerine de dikkat edilmesi gerektiğine değinmişlerdir. Bu amaçla kurdukları tedarik zinciri ağı modelinde çeşitli kalite maliyetlerini de ele almışlar, modelin çözdürülmesi için de tavlama benzetimi ve genetik aloritmadan yararlanmışlardır. Çözüm sonucu elde edilen bulguları incelediklerinde genetik algoritmanın tavlama benzetimine göre daha iyi çalıştığını ve kalite maliyetlerinin süreç üzerinde önemli etkisinin olduğunu göstermişlerdir.

4.3. Uygulama Verilerinin Oluşturulması

Tez kapsamında yapılan çalışmaların bir parçası olarak, kurulan model için büyük boyutlu problemleri çözebilecek bir metasezgisel geliştirilmiştir. Bu metasezgisel geliştirilirken temel olarak iki performans ölçütü üzerinde durulmuştur. Bu performans ölçütleri çözüm zamanı ve elde edilen çözümün optimum çözüme olan uzaklığıdır. Literatürde firma verileriyle yüksek oranda uyuyan bir test problemi olmadığı için firma verilerinden GAMS optimizasyon programının çözüme varabileceği küçük (29 şehir) orta (65 şehir) ve büyük (81 şehir) boyutlu üç adet test problemi oluşturulmuştur. Bu test problemleri ile amaçlanan, geliştirilen metasezgisel algoritmanın küçük orta ve büyük boyutlu problemlerde nasıl çalıştığının gözlemlenmesidir. Oluşturulan küçük orta ve büyük boyutlu test problemleri aşağıdaki gibidir.

4.3.1. Küçük Boyutlu Problem

Türkiye'nin 81 şehrinin içerisinde rassal olarak seçilmiş 29 şehirden oluşmaktadır. Şehirlere ilişkin talep, depo maliyetleri vs. veriler önceki bölümde verilen Çizelge 3.1'den elde edilmiştir.

Seçilen 29 şehir şu şekildedir: Adana, Ankara, Antalya, Balıkesir, Bursa, Edirne, Eskişehir, Gaziantep, Hatay, İçel, İstanbul, İzmir, Kayseri, Kocaeli, Konya, Malatya, Muğla, Ordu, Sakarya, Samsun, Sivas, Tekirdağ, Trabzon, Şanlıurfa, Van, Kırıkkale, Bartın, Karabük, Düzce.

4.3.2. Orta Boyutlu Problem

Plaka kodu 1 ile 65 arasında olan 65 şehirden oluşmaktadır. Bu problem için de yine önceki bölümde verilen Çizelge 3.1'deki veriler kullanılmıştır.

4.3.3. Büyük Boyutlu Problem

Önceki bölümde çözülmüş ve yorumlanmış olan Türkiye'nin 81 ilinin tamamını içeren problemidir.

Bu problemlerin çözümleri sonucunda elde edilen bulgular her bir problem için Çizelge 4.1'de özetlenmiştir.

Çizelge 4.1'de de görüldüğü gibi şehir sayısının artması çözüm zamanını çok daha fazla oranda artırmaktadır. Şehir sayısı ile problemin çözüm süresi arasındaki ilişkiyi gösteren grafik Şekil 4.1'de verilmiştir.

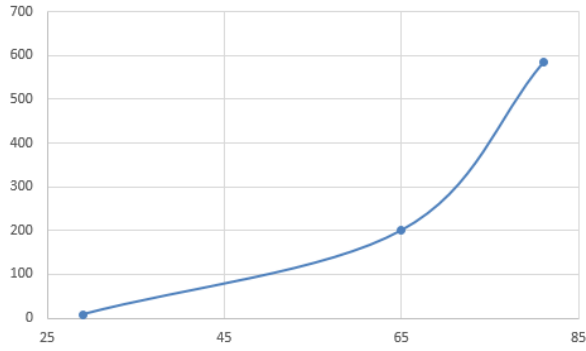
Çizelge 4.1'de ve Şekil 4.1'de görüldüğü gibi 29 şehirli bir problem yaklaşık 9 saniyede çözülürken 65 şehirli bir problemin çözüm süresi 401,95 saniyedir. Şehir sayısındaki 2,24 katlık bu artış çözüm zamanını 47,39 kat artırmıştır. Benzer bir artış 65 şehir ile 81 şehirli problemlerin çözüm zamanları için de geçerlidir. Buradaki veriler problemin NP-Hard sınıfından bir problem olduğunu teyit etmektedir.

Tedarik zinciri ağ tasarımı problemleri şirketten şirkete değişebilecek çok farklı ölçekte problemlerdir. Bazı şirketlerin bu modeli 200-300 şehir için çözmesi gerekebilmektedir. Bu durumda GAMS de dahil olmak üzere hiçbir

optimizasyon yazılımının sonuç verememesi ihtimali yüksektir. Bu yüzden problemi kısa zamanda çözebilecek ve kaliteli çözümler verebilecek bir metasezgiselin geliştirilmesi zorunluluğu ortaya çıkmıştır. Bu çalışma kapsamında böyle bir metasezgiselin geliştirilmesi hedeflenmiştir.

Çizelge 4.1. Farklı boyutlardaki problemlerin kesin çözüm sonuçları

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	g_1^+ değeri	g_2^+ değeri	Açılan Depolar	Depo Kapasiteleri
29 Şehir	8,48	328031,5	411350	244713	Ankara Antalya Kocaeli	3494 1629 5580
65 Şehir	200,64	428911,5	174693	683130	Adana Bilecik Samsun	1437 9327 1498
81 Şehir	583,91	434507,5	177936	691079	Adana Bilecik Samsun	1490 9460 1517



Şekil 4.1. Problem boyutuna bağlı olarak çözüm süresinin değişimi

Geliştirilen metasezgisel temelde iki karar vermelidir. Birincisi hangi şehirlere depo açılacağı, ikincisi ise hangi müşterinin hangi depoya atanacağı kararıdır. Bu iki karardan herhangi biri yanlış verildiğinde optimum çözüme ciddi anlamda uzak bir çözüm elde edilecektir. Örnek olarak, geliştirilen metasezgiselin açılması gereken depoları doğru bir şekilde belirlediğini varsayalım. Bu durumda eğer bu metasezgisel her bir depoya doğru müşterileri atamazsa tedarik zinciri ağı

olması gerekenden çok farklı bir yapıda olacak ve bulunan çözüm optimum çözümden oldukça uzak bir çözüm olacaktır. Kısacası, kullanılan metasezgiselin depoları doğru seçme ve depolara müşterilere uygun biçimde atama yeteneğine sahip olması gerekmektedir.

4.4. Önerilen Gösterim Şekli

Bir metasezgisel algoritmanın verdiği çözümün kalitesini etkileyen en önemli faktörlerden biri de problem için kullanılan gösterim şeklidir.

Problem kapsamında özel bir permütasyon gösterimi kullanılmıştır. Permütasyonun toplam boyutu şehirlerin sayısına eşittir. Permütasyonda ilk n hücredeki değerler depoların açılacağı şehirleri ifade etmektedir. Tüm müşteriler belli bir kurala göre bu depolardan birine atanacaktır. Örnek olarak 29 şehirli bir problemde eğer toplamda 2 depo açılması isteniyorsa Çizelge 4.2’deki permütasyon kullanılır.

Çizelge 4.2. Problem için önerilen gösterim şekli

1	2	3	4	5	6	7	8	9	10	...	27	28	29
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	...	P27	P28	P29
Açılan Depolar													
Depolara Atanacak Müşteriler													

Çizelge 4.2’deki permütasyon üzerindeki P1 ve P2 değerleri deponun açılacağı şehirleri ifade etmektedir. Permütasyondaki hücrelerin sayısı şehirlerin toplam sayısına eşittir ve 29’dur, 1 ile 29 arasındaki her bir değerden permütasyon üzerinde tam olarak bir tane yer alır ve ilgili şehri ifade eder.

Aşağıdaki örnekte de benzer şekilde k tane şehir rassal olarak hücrelere yerleştirilmiştir.

Çizelge 4.3. Örnek bir gösterim

9	14	2	4	5	8	11	17	6	28	...	24	1	13
k tane													

Eğer problem kapsamında 1 adet depo açılması planlanıyorsa ilk hücredeki şehir (9. şehir) depo yeri olarak seçilir ve k tane şehrin tamamı bu depoya atanır. Eğer 2 tane depo açılması planlanıyorsa ilk 2 hücredeki şehirler (9. ve 14. şehir) depo yerleri olarak seçilir ve belli bir kurala göre k tane şehir bu depolara atanır. Benzer şekilde toplamda 3 depo açılması planlandığında ise ilk 3 hücredeki şehirler (9. şehir, 14. şehir ve 2. şehir) depo olarak seçilir, diğer şehirler belli bir kurala göre bu depolara atanır.

Kısacası $n \leq k$ olmak üzere toplamda n adet depo açılmak istendiğinde ilk n hücredeki şehirler depo yeri olarak seçilir diğer şehirler ise belli bir kurala göre bu depolara atanır.

4.5. Kullanılan Metasezgisel Yaklaşım

Açılacak depoların belirlenmesi sürecinde yasaklı arama algoritmasından yararlanılmıştır. Yasaklı arama algoritması her bir iterasyonda ilk n hücredeki değerlerden amaç fonksiyonu açısından en uygun olanını kendi kurallarına göre değiştirecek böylece her bir iterasyonda depo yerlerinden bir tanesi değişmiş olacaktır. Bu algoritma ile amaçlanan, depo kombinasyonunun her iterasyonda değiştirilmesi ile amaç fonksiyonu değerini en aza indirecek depo kombinasyonunun belirlenmesi ve bu yolla açılacak depolara karar verilmesidir.

Yasaklı arama algoritması Glover tarafından 1986 yılında öne sürülen tek çözüm temelli bir metasezgisel yaklaşımdır. Yerel optimumlardan kurtulmak için özel bir yapıya sahiptir. Günümüzde en çok kullanılan tek çözüm temelli metasezgisellerden biridir. Yerel optimumlardan kurtulmayı sağlayan özel yapısının yanında bir diğer önemli özelliği de arama sürecinde çeşitli verileri tutan hafızalara sahip olmasıdır.

Yasaklı arama algoritması yerel arama algoritmasına benzer fakat yerel arama algoritmasından farklı olarak yasaklı arama algoritması hiçbir hareketin amaç fonksiyonunu iyileştirmediği durumlarda amaç fonksiyonunu kötüleştiren bir hareketi de kabul eder. Genellikle her adımda tüm komşuluklar tek tek araştırılır. Komşuluklar içerisinde amaç fonksiyonunu en çok iyileştiren hareket yapılır. Eğer hiçbir hareket amaç fonksiyonunda iyileşmeye yol açmıyorsa bu durum yerel optimum noktaya varıldığını gösterir ve algoritma daha kötü bir çözümü kabul

ederek yerel optimum noktadan kurtulur. Bu süreçte turların oluşması ve önceki adımlarda yapılan hareketlerin tekrar tekrar yapılması mümkündür. Bu olumsuz durumdan kurtulmak için yasak listesi adı verilen bir liste kullanılır ve o listedeki hareketler aspirasyon koşulu sağlanmadıkça yapılmaz. Yasak listesinde daha önce yapılmış olan hareketler önceden belli olan veya çalışma anında rassal olarak belirlenen bir süre boyunca tutulur. Her bir iterasyonda yasak listesi güncellenir. Yasak listesine yasaklı arama algoritmasının kısa dönemli hafızası denir. Bunun dışında, en iyi çözümlerin tutulduğu hafızaya orta dönemli hafıza, arama uzayına dair bilgilerin tutulduğu hafızaya uzun dönemli hafıza denir. Yasak listesinde bulunan bir hareket o zamana kadar bulunmuş olan en iyi çözümden daha iyi bir çözüm veriyorsa bu hareket yapılabilir. Buna aspirasyon koşulunun sağlanması denir.

Yasaklı arama algoritmasının sözde kodu aşağıdaki gibidir.

$s = s_0$ (Başlangıç çözümü oluştur)

Kısa orta ve uzun dönemli hafızaları başlat

Tekrarla

- En kabul edilebilir s' komşuluğunun bulunması (Bu komşuluk yasak olmamalı ya da aspirasyon koşulu sağlanmalı)
- $s = s'$
- Yasak listesini ve bulunan en iyi çözümü güncelle
- Durdurma koşulu sağlanınca dur

Daha önce de belirtildiği gibi açılacak depoların belirlenmesi sürecinde yasaklı arama algoritmasından yararlanılmıştır. Bu noktada verilmesi gereken bir diğer karar da depolara müşterilerin nasıl atanacağını belirlenmesidir. Çünkü belli bir depo kombinasyonu için müşterilerin depolara farklı şekilde atanması farklı amaç fonksiyonu değerleri verecektir. Atama işleminin yapılması için de farklı yaklaşımlar denenmiştir. Denenen tüm yaklaşımlar öncelikle 65 şehirli probleme uygulanmıştır. Sonraki alt bölümlerde bu yaklaşımlara yer verilmiştir.

4.6. Yasaklı Arama ve Rassal Atama

Bu çözüm yaklaşımında depoların belirlenmesi sürecinde yasaklı arama algoritmasından yararlanılmıştır. Depolar belirlendikten sonra müşterilerin bu depolara atanması ise rassal olarak gerçekleştirilmiştir. Bu yaklaşıma göre

problemin çözdürülmesiyle elde edilen sonuçlar ve bu sonuçların kesin çözüm ile karşılaştırması Çizelge 4.4'te yapılmıştır.

Çizelge 4.4. Kesin çözümle rassal atama yaklaşımının karşılaştırılması

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	Açılan Depolar
Kesin Çözüm	200,64	428911,5	Adana Bilecik Samsun
Metasezgisel Yaklaşım	0,19	1120028	Bilecik Ankara Sakarya
Optimum Çözüme Uzaklık	%161,13		

Çizelge 4.4'teki veriler, uygulanan çözüm yaklaşımının kesin çözüm yöntemiyle yapılan çözümün süresine göre çok daha kısa sürede sonuç verdiğini göstermektedir fakat kullanılan çözüm yaklaşımının verdiği sonucun kesin çözüm ile bulunan sonuca %161,13 uzak olduğu, bu sebeple kabul edilemez bir sonuç olduğu da görülmektedir. Buradan ayrıca bir metasezgiselin tek başına verimli sonuçlar üretemeyebileceği bu yüzden melez yaklaşımların denenmesi gerektiği sonucuna varılabilir.

4.7 Yasaklı Arama ve Yerel Arama

Bu yaklaşımda depo yerlerinin belirlenmesi için yine yasaklı arama yönteminden yararlanmıştır fakat depolara müşterilerin atanması rassal olarak yapıldıktan sonra yerel arama algoritması ile iyileştirilmektedir. Böylece her bir iterasyonda öncelikle yasaklı arama algoritmasıyla depo yerleri belirlenmiş daha sonra bu depolara müşteriler yerel arama algoritması ile amaç fonksiyonunu mümkün olduğu kadar iyileştirecek şekilde atanmıştır. Bu yaklaşıma göre problemin çözdürülmesiyle elde edilen sonuçlar ve bu sonuçların kesin çözüm ile karşılaştırması Çizelge 4,5'te yapılmıştır.

Çizelge 4.5. Kesin çözümle yerel aramayla atama yaklaşımının karşılaştırılması

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	Açılan Depolar
Kesin Çözüm	200,64	428911,5	Adana Bilecik Samsun
Metasezgisel Yaklaşım	34,41	470582,5	Adana Bilecik Samsun
Optimum Çözüm Uzaklık	%9,72		

Çizelge 4.5'teki veriler incelendiğinde müşterilerin depolara yerel arama algoritmasıyla atanmasının çözüm kalitesini ciddi anlamda iyileştirdiği ve optimum çözüme olan uzaklığı %161,13'ten %9,72'ye düşürdüğü görülmektedir. Bu da atama işleminin rassal olarak değil de belli bir kurala göre yapılmasının oldukça yararlı olduğunu göstermektedir. Bu yaklaşımla çözümde sağlanan iyileşme oldukça radikal bir iyileşme olsa da optimum çözüme olan %10'a yakın bu uzaklık kabul edilebilir bir uzaklık değildir. Ayrıca yerel arama algoritmasının her iterasyonda tekrarlanması çözüm süresini belirgin biçimde artırmış önceki yaklaşımda 1 saniyenin altında olan çözüm süresi bu yaklaşımla yarım dakika dolaylarında olmuştur. Bu da özellikle büyük boyutlu problemler için algoritmanın etkinliği konusunda bir risk teşkil etmektedir. Çünkü büyük boyutlu problemlerde her bir iterasyonda yerel aramanın yapılması çözüm süresini ciddi düzeyde artıracaktır.

Bu yaklaşım ile yapılan çözüm sonucu açılan depolar incelendiğinde müşteriler belli bir kurala göre depolara atandığında yasaklı arama algoritmasının çok iyi çalıştığı ve kesin çözümde bulunan depolarla aynı depoları açabildiği görülmektedir. Dolayısıyla, optimum çözüme olan %10'a yakın uzaklığın müşterilerin depolara farklı şekilde atanmasından kaynaklandığı sonucuna varılabilir. Yerel arama algoritmasının yerel optimum noktalara takılabildiği ve global optimum noktayı bulamayabildiği bilinen bir gerçektir. Bu sebeple depolara müşterilerin atanması sürecinde çalışacak bir sezgiselin geliştirilmesi gerektiği

anlaşılmaktadır. Bu sezgisel yerel aramadan daha iyi çalışabilecek ve daha kaliteli çözümler bulabilecek bir sezgisel olmalıdır. Ayrıca bu sezgisel çözüm zamanı açısından da büyük boyutlu problemlerde risk teşkil etmeyecek bir sezgisel olmalıdır. Bu bağlamda, kullanılacak bir sezgiselin kabul edilebilir ve uygulanabilir bir sezgisel olması için optimum çözüme olan uzaklığının %1 dolaylarında olması ve çözüm süresinin de 5 saniyenin altında olması hedeflenmiştir.

4.8. Atama İçin Kullanılan Diğer Sezgisel Yaklaşımlar

4.8.1. En Düşük Maliyetli Depoya Atama

Bu yaklaşıma göre depolar yasaklı arama algoritmasına göre belirlendikten sonra müşteriler minimum maliyetli olacak şekilde depolara atanmaktadır. Bu yaklaşıma göre problemin çözdürülmesiyle elde edilen sonuçlar ve bu sonuçların kesin çözüm ile karşılaştırması Çizelge 4.6’da yapılmıştır.

Bu yaklaşım sonucu çözüm zamanı yerel aramaya göre belirgin biçimde düşmüş ve yarım dakikadan 1 saniyenin altına inmiştir. Fakat optimum çözüme olan uzaklık %33,85’tir. Bu uzaklık değeri rassal atamaya göre daha iyi olsa da yerel aramanın bulduğu yaklaşık %10 değerindeki uzaklıktan çok daha kötüdür ve kabul edilemezdir.

Çizelge 4.6. Kesin çözümle en düşük maliyetli depoya atama yaklaşımının karşılaştırılması

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	Açılan Depolar
Kesin Çözüm	200,64	428911,5	Adana Bilecik Samsun
Metasezgisel Yaklaşım	0,21	574108,5	Ankara Uşak Kocaeli
Optimum Çözüme Uzaklık	%33,85		

4.8.2. En Yakın Depoya Atama

Bu yaklaşımda depo yerleri yasaklı arama ile belirlendikten sonra her bir müşteri kendisine en yakın depoya atanmıştır. Bu yaklaşıma göre yapılan atama yerel aramaya göre yapılan atamadan yaklaşık %3 daha iyi çözüm vermiştir ve çözüm süresi de yerel aramaya göre çok daha iyidir. Bu yaklaşıma göre problemin çözdürülmesiyle elde edilen sonuçlar ve bu sonuçların kesin çözüm ile karşılaştırması Çizelge 4.7’de yapılmıştır.

Çözüm süresi açısından oldukça etkin bir algoritma olsa da optimum çözüme olan %7 uzaklık çoğu zaman kabul edilebilir bir uzaklık değildir. Dolayısıyla bu çözüm yaklaşımından daha etkin bir yaklaşım bulunması gerektiği ortaya çıkmıştır.

Çizelge 4.7. Kesin çözümle en yakın depoya atama yaklaşımının karşılaştırılması

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	Açılan Depolar
Kesin Çözüm	200,64	428911,5	Adana Bilecik Samsun
Metasezgisel Yaklaşım	0,18	458503,5	Adana Bilecik Samsun
Optimum Çözüme Uzaklık	%6,9		

4.8.3. En Düşük Amaç Fonksiyonu Değerini Veren Depoya Atama

Bu yaklaşım yukarıda kullanılan iki yaklaşımın birleşimi şeklinde düşünülebilir. Bu yaklaşıma göre depo yerleri yasaklı arama algoritmasına göre seçildikten sonra müşteriler en düşük amaç fonksiyonu değerine sahip depoya atanır. Atama sürecinde her bir müşteri için farklı depolara atanması durumunda oluşacak amaç fonksiyonu değerleri hesaplanır, en küçük amaç fonksiyonu değerini veren depoya atanır. Bu işlem yapılması zor ve zaman alıcı bir işlem gibi görünse de aslında yasaklı arama algoritmasıyla benzer düzeyde bir karmaşıklığa sahiptir.

Zira n deponun açılacağı k şehirli bir problem için her iterasyonda yasaklı arama algoritması $k.(n-k)$ tane yer değiştirmeyi amaç fonksiyonu açısından kontrol etmektedir. Bu sezgisel de $(n-k)$ tane müşterinin her birisi için k tane depoyu denediği için $(n-k).k$ karmaşıklığa sahiptir. Yani karmaşıklık açısından yasaklı arama ile aynıdır. Oysa bir yerel arama algoritması $[(n - k)^2 - (n - k)]/2$ karmaşıklığa sahiptir.

Kısacası bu yaklaşım yerel arama algoritmasına göre daha az karmaşıklığa sahip bir yaklaşımdır. Bu yaklaşıma göre problemin çözülmesiyle elde edilen sonuçlar ve bu sonuçların kesin çözüm ile karşılaştırması Çizelge 4.8’de yapılmıştır. Çizelge 4.8’deki veriler de bahsettiğimiz bilgileri doğrulamaktadır. Bu yaklaşım ile yapılan çözüm için çözüm süresi 0,29 saniye bulunmuştur. Bu değer hedeflenen değer olan 5 saniyenin altında olduğu için kabul edilebilirdir. Ayrıca bu yaklaşım ile problemin optimum çözümünün elde edildiği de görülmektedir. Atamaların her birisi en küçük amaç fonksiyonu değerine göre yapıldığı için problemin optimum çözümünün bulunması muhtemel olmaktadır.

Çizelge 4.8. Kesin çözümle önerilen yaklaşımın karşılaştırılması

Problem Boyutu	Çözüm Süresi (sn.)	z değeri	Açılan Depolar
Kesin Çözüm	200,64	428911,5	Adana Bilecik Samsun
Metasezgisel Yaklaşım	0,29	428911,5	Adana Bilecik Samsun
Optimum Çözüme Uzaklık	%0		

Bu yaklaşım ayrıca küçük (29 şehir) ve büyük (81 şehir) boyutlu problemler için de denenmiştir. Tüm bu problemlerde elde edilen sonuçlar Çizelge 4.9’da özetlenmiştir.

Çizelge 4.9’da da görüldüğü gibi kullanılan melez yaklaşım farklı boyuttaki problemlerde kısa sürede optimum çözümü bulabilmektedir.

Çizelge 4.9. Önerilen yaklaşımın farklı boyuttaki problemlerde verdiği sonuçlar

Problem Boyutu	GAMS Çözüm Süresi (sn.)	Metasezgisel Yaklaşım Çözüm Süresi (sn.)	Optimum Çözümünden Uzaklık
29 Şehir	8,48	0,27	%0
65 Şehir	200,64	0,29	%0
81 Şehir	583,91	0,34	%0

4. SONUÇLAR

Tedarikçilerin üretim tesislerinin depoların dağıtım merkezlerinin perakendecilerin ve müşterilerin oluşturduğu bir ürünün ham maddesinin temini aşamasından müşteriye ulaşması aşamasına kadar geçirdiği tüm süreçleri kapsayan ağa tedarik zinciri ağı denir. Bu ağ üzerindeki her bir elemanın konumunun kapasitesinin sayısının ve bu elemanlar arasındaki akışın belirlenmesi sürecine tedarik zinciri ağ tasarımı denir. Tez kapsamında Bilecik merkezli bir üretim firmasının tedarik zinciri ağının tasarımına odaklanılmıştır. Üretim tesisinin yeri sabit ve değiştirilemez olduğu için ağ tasarımı problemi iki ayrı probleme ayrıştırılarak incelenmiştir.

Bu problemlerden ilki firmanın çalışacağı tedarikçileri belirlemeyi amaçlayan tedarikçi seçimi problemidir. Bu problemin çözümüyle tedarik zincirinin ilk halkası olan tedarikçiler belirlenmiş ve tedarikçilerle müşteriler birbirine bağlanmıştır. Tedarikçi seçimi için maliyet, kalite, temin süresi gibi nicel ölçütlerin yanında çevresel ve sosyal sorumluluk, eğitim ve sürekli iyileştirme gibi nitel ölçütlere de değerlendirme kapsamına dahil edilmiştir. Bu bağlamda, değerlendirme işleminin yapılabilmesi için nicel ve nitel faktörleri birlikte ele alabilen ve Tip-2 bulanık kümelerin kullanımına olanak tanıyan Buckley'in Bulanık AHP algoritmasından yararlanılmıştır. Algoritmayla öncelikle uzman görüşlerinden ve literatürden yararlanılarak ölçütlerin önem dereceleri belirlenmiş sonrasında 4 tedarikçi nicel ve nitel tüm yönleriyle değerlendirilmiştir. Seçim üzerine etkin eden tüm ölçütlerin analize dahil edilmesi analiz sonuçlarının güvenilirliğini de artırmıştır. Tedarikçilerin genel ağırlıklarına bakıldığında aslında birbirine bariz biçimde üstünlük kurabilen bir tedarikçinin olmadığı görülmektedir. Bu da her bir tedarikçinin belli ölçütlerde diğerlerine üstünlük sağladığını göstermiştir. Burada önemli olan genel anlamda diğerlerinde üstün olan tedarikçinin seçimidir, o da Tedarikçi B olmuştur.

Tedarik zinciri ağının ikinci kısmı olarak da üretim tesisi, depolar ve müşterilerden oluşan ağın tasarımı problemi ele alınmıştır. Bu problemde firma kısıtları ve hedefleri dahilinde depoların açılacağı şehirlerin belirlenmesi, depoların toplam sayılarının belirlenmesi, her bir depoya müşterilerin atanması ve her bir

deponun aylık toplam akış miktarına göre kapasitelerinin belirlenmesi karar değişkenleri olarak ele alınmış ve problem GAMS optimizasyon yazılımında çözdürülerek iki amaç için ödünleşmenin sağlandığı nokta belirlenmiştir. Sonuçlar hem birinci hedeften hem ikinci hedeften pozitif yönlü sapmaların olduğunu göstermektedir. Buna göre ilk hedef olan maliyet hedefinden pozitif yönlü sapma müşterilere yakın olmak için maliyetten bir miktar ödün verildiğini, ikinci hedef olan müşterilere yakınlık hedefinden pozitif yönlü sapma da maliyetin düşürülmesi için müşterilere yakınlıktan bir miktar ödün verildiğini göstermektedir. Analizler sonucu dikkat çeken bir diğer önemli konu da problem boyutundaki büyümenin problemin çözümünü zorlaştırdığını ve çözüm süresini ciddi anlamda artırdığını göstermiştir. GAMS optimizasyon yazılımı 65 boyutlu bir problemi 201 saniyede çözerken problem boyutu yaklaşık %25 artırıldığında çözüm süresi neredeyse 3 kat artmakta ve 584 saniyeye çıkmaktadır. Bu radikal artışlar belli bir noktaya kadar sürecek, belli bir şehir sayısından sonra ise kesin yaklaşımlar çözüm vermeyecektir. Bu da özellikle büyük boyutlu problemlerin çözümü için bir metasezgisel yaklaşımın geliştirilmesini zorunlu kılmış, bu motivasyonla kısa sürede çok kaliteli sonuçlar veren bir metasezgisel yaklaşım geliştirilmiştir. Geliştirilen metasezgisel farklı boyutlarda 3 problemde denenmiş ve 1 saniyeden daha az sürede çok kaliteli sonuçlar verdiği görülmüştür. Bu metasezgisel yaklaşım, ilerde şirketin müşterilerinin artması durumunda tedarik zinciri ağını yeniden tasarlamak için güvenle kullanılabilecektir.

Özetle, firma için tedarikçi, üretim alanı, depolar ve müşterilerden oluşan bir tedarik zinciri ağı oluşturulmuştur. Bu ağda tedarikçilerin yeri belirlenirken 14 farklı ölçüt göz önünde bulundurulmuş, depoların yerinin seçiminde ve müşterilerin depolara atanmasında ise iki hedefe odaklanılmıştır. Böylece sadece maliyet odaklı bir tedarik zinciri ağı tasarımıyla kaçınılmış, zincirin her halkasında müşteri memnuniyeti de hesaba katılmıştır.

KAYNAKLAR

- [1] Beamon, B., "Supply chain design and analysis: Models and methods", *Int. J. Production Economics*, 55, 281-294, 1998.
- [2] Ballou, R., "Unresolved issues in supply chain network design", *Information Systems Frontiers*, 3, 417-426, 2001.
- [3] Meixell, M. ve Gargeya, B., "Global supply chain design: A literature review and critique", *Transportation Research Part E*, 41, 531-550, 2005.
- [4] Ávila, P., Mota, A., Pires, A., Bastos, J. ve Teixeira, J., "Supplier's selection model based on an empirical study", *Procedia Technology*, 5, 625-634, 2012.
- [5] Rajesh, G. ve Malliga, P., "Supplier selection based on AHP QFD methodology", *Procedia Engineering*, 64, 1283-1292, 2013.
- [6] Xia, W. ve Wu, Z., "Supplier selection with multiple criteria in volume discount environments", *Omega*, 35, 494 – 504, 2007.
- [7] Rezaei, J. ve Ortt, R., "Multi-criteria supplier segmentation using a fuzzy preference relations based AHP", *European Journal of Operational Research*, 225, 75-84, 2013.
- [8] Masi, D., Micheli, G. ve Cagno, E., "A meta-model for choosing a supplier selection technique within an EPC company", *Journal of Purchasing & Supply Management*, 19, 5-15, 2013.
- [9] Zouggar, A. ve Benyoucef, L., "Simulation based fuzzy TOPSIS approach for group multi-criteria supplier selection problem", *Engineering Applications of Artificial Intelligence* 25, 507-519, 2012.
- [10] Bruno, G., Esposito, E., Genovese, A., ve Passaro, R., "AHP-based approaches for supplier evaluation: problems and perspectives," *Journal of Purchasing and Supply Management*, 18, 159-172, 2012.
- [11] Chen, S., ve Lee, L., "Fuzzy multiple attributes group decision-making based on the interval type-2 TOPSIS method", *Expert Systems with Applications*, 37, 2790-2798, 2010
- [12] Buckley J., "Fuzzy hierarchical analysis", *Fuzzy Sets and Systems*, 17, 233-247, 1985.
- [13] Saaty, T., "How to make a decision: The analytic hierarchy process", *European Journal of Operational Research*, 48, 9-26, 1990.
- [14] Tsiakis, P., Shah, N. ve Pantelides, C., "Design of multi-echelon supply chain networks under demand uncertainty", *Ind. Eng. Chem. Res.*, 40, 3585-3604, 2001.
- [15] Pirkul, H. ve Jayaraman, V., "Production, transportation, and distribution planning in a multi commodity tri-echelon system", *Transportation Science*, 30, 291-302, 1996.
- [16] Pirkul, H. ve Jayaraman, V., "A multi-commodity, multi-plant, capacitated facility location problem: Formulation and efficient heuristic solution", *Computers Ops Res*, 25, 869-878, 1998.
- [17] Guillén, G., Mele, F., Bagajewicz, M.J., Espuña, A. ve Puigjaner, L., "Multiobjective supply chain design under uncertainty", *Chemical Engineering Science*, 60, 1535 – 1553, 2005.

- [18]Pazhani, S., Ramkumar, N., Narendran, T. ve Ganesh, K., “A bi-objective network design model for multi-period, multi-product closed-loop supply chain”, *Journal of Industrial and Production Engineering*, 30, 264-280, 2013.
- [19]Wang, F., Lai, X. ve Shi, N., “A multi-objective optimization for green supply chain network design”, *Decision Support Systems*, 51, 262–269, 2011.
- [20]Hiremath, N.C., Sahu, S. ve Tiwari, M., “Multi objective outbound logistics network design for a manufacturing supply chain”, *J Intell Manuf*, 24, 1071–1084, 2013.
- [21]Anonim, Karayolları Genel Müdürlüğü, <http://www.kgm.gov.tr/>
- [22]Costa, A., Celano, G., Fichera, S. ve Trovato, E., “A new efficient encoding/decoding procedure for the design of a supply chain network with genetic algorithms”, *Computers & Industrial Engineering*, 59 986–999, 2010.
- [23]Yao, M. ve Hsu, H., “A new spanning tree-based genetic algorithm for the design of multi-stage supply chain networks with nonlinear transportation costs”, *Optim Eng*, 10, 219–237, 2009.
- [24]Easwaran, G ve Üster, H., “Tabu search and benders decomposition approaches for a capacitated closed-loop supply chain network design problem”, *Transportation Science*, 43, 301-320, 2009.
- [25]Govindan, K., Jafarian, A. ve Nourbakhsh, V., “Biobjective integrating sustainable order allocation and sustainable supply chain network strategic design with stochastic demand using a novel robust hybrid multi-objective metaheuristic”, *Computers & Operations Research*, Accepted Manuscript In Press
- [26]Castillo-Villar, K. ve Herbert-Acero, J., “A metaheuristic-based approach for the capacitated supply chain network design problem Including imperfect quality and rework”, *IEEE Computational Intelligence Magazine*, 9, 31-45, 2014.

Ek 1: Şehirlerin talep ve depo açma gideri bilgileri

Şehir Plaka Kodu	Şehir Adı	Aylık Talep (Palet)	Aylık Depo Kiralama ve İşletme Giderleri (TL)
01	ADANA	551	28000
02	ADIYAMAN	2	24000
03	AFYON	37	27000
04	AĞRI	4	19000
05	AMASYA	29	26000
06	ANKARA	863	45000
07	ANTALYA	1190	33000
08	ARTVİN	137	23000
09	AYDIN	11	26000
10	BALIKESİR	239	22000
11	BİLECİK	45	21000
12	BİNGÖL	0	23000
13	BITLİS	2	24000
14	BOLU	19	17000
15	BURDUR	2	22000
16	BURSA	642	21000
17	ÇANAKKALE	18	22000
18	ÇANKIRI	5	23000
19	ÇORUM	17	21000
20	DENİZLİ	97	19000
21	DİYARBAKIR	41	21000
22	EDİRNE	116	21000
23	ELAZIĞ	77	27000
24	ERZİNCAN	7	25000
25	ERZURUM	40	29000
26	ESKİŞEHİR	96	36000
27	GAZİANTEP	89	24000
28	GİRESUN	25	25000
29	GÜMÜŞHANE	1	27000
30	HAKKARİ	1	28000
31	HATAY	269	24000
32	ISPARTA	75	27000
33	İÇEL	197	21000

34	İSTANBUL	2870	56000
35	İZMİR	756	36000
36	KARS	1	32000
37	KASTAMONU	30	24000
38	KAYSERİ	276	23000
39	KIRKLARELİ	13	22000
40	KIRŞEHİR	5	21000
41	KOCAELİ	648	38000
42	KONYA	34	22000
43	KÜTAHYA	433	24000
44	MALATYA	43	22000
45	MANİSA	121	26000
46	KAHRAMANMARAŞ	66	29000
47	MARDİN	40	23000
48	MUĞLA	439	27000
49	MUŞ	2	23000
50	NEVŞEHİR	37	21000
51	NİĞDE	27	22000
52	ORDU	135	27000
53	RİZE	36	19000
54	SAKARYA	106	23000
55	SAMSUN	614	17000
56	SİİRT	2	19000
57	SİNOP	22	21000
58	SİVAS	165	19000
59	TEKİRDAĞ	71	28000
60	TOKAT	48	21000
61	TRABZON	215	34000
62	TUNCELİ	2	19000
63	ŞANLIURFA	6	26000
64	UŞAK	33	21000
65	VAN	22	27000
66	YOZGAT	13	16000
67	ZONGULDAK	5	19000
68	AKSARAY	18	22000
69	BAYBURT	9	18000
70	KARAMAN	32	22000

71	KIRIKKALE	15	23000
72	BATMAN	21	14000
73	ŞIRNAK	19	19000
74	BARTIN	17	21000
75	ARDAHAN	7	23000
76	IĞDIR	3	12000
77	YALOVA	14	28000
78	KARABÜK	2	21000
79	KİLİS	8	19000
80	OSMANİYE	5	24000
81	DÜZCE	17	26000

Ek 2 Problem için önerilen metasezgisel yaklaşımın C# kodu

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Diagnostics;
namespace ConsoleApplication12
{
    class Program
    {
        static void data(ref int[,] costs, ref int[,] distances, ref int[] demands, ref int[] firststeps,
ref int[] rentingscs)
        {
            String input = File.ReadAllText(@"C:\costs3.txt");
            int i = 0, j = 0;
            int[,] costmatrisi = new int[81, 81];
            foreach (var row in input.Split('\n'))
            {
                j = 0;
                foreach (var col in row.Trim().Split('\t'))
                {
                    costmatrisi[i, j] = int.Parse(col.Trim());
                    j++;
                }
                i++;
            }
            String input2 = File.ReadAllText(@"C:\distances3.txt");
            int a = 0, b = 0;
            int[,] distancematrisi = new int[81, 81];
            foreach (var row in input2.Split('\n'))
            {
                b = 0;
                foreach (var col in row.Trim().Split('\t'))
                {
                    distancematrisi[a, b] = int.Parse(col.Trim());
                    b++;
                }
                a++;
            }
            int[] demand = { 551, 2, 37, 4, 29, 863, 1190, 137, 11, 239, 45, 0, 2, 19, 2, 642, 18, 5, 17,
97, 41, 116, 77, 7, 40, 96, 89, 25, 1, 1, 269, 75, 197, 2870, 756, 1, 30, 276, 13, 5, 648, 34, 433,
43, 121, 66, 40, 439, 2, 37, 27, 135, 36, 106, 614, 2, 22, 165, 71, 48, 215, 2, 6, 33, 22, 13, 5, 18,
9, 32, 15, 21, 19, 17, 7, 3, 14, 2, 8, 5, 17 };
            int[] firststep = { 609, 828, 174, 1054, 497, 256, 382, 975, 420, 203, 0, 937, 1082, 177,
287, 81, 298, 359, 445, 322, 945, 384, 830, 775, 918, 69, 765, 695, 818, 1287, 754, 286, 603,
204, 338, 1068, 370, 495, 370, 391, 115, 338, 93, 756, 309, 704, 1010, 435, 1020, 462, 516,
661, 855, 84, 545, 1086, 512, 597, 308, 550, 799, 874, 868, 205, 1187, 425, 231, 420, 846, 426,
315, 1020, 1146, 302, 1056, 1137, 106, 283, 793, 674, 140 };
            int[] rentingc = { 28000, 24000, 27000, 19000, 26000, 45000, 33000, 23000, 26000,
22000, 21000, 23000, 24000, 17000, 22000, 21000, 22000, 23000, 21000, 19000, 21000,
```

```

21000, 27000, 25000, 29000, 26000, 24000, 25000, 27000, 28000, 24000, 27000, 21000,
56000, 36000, 32000, 24000, 23000, 22000, 21000, 38000, 22000, 24000, 22000, 26000,
29000, 23000, 27000, 23000, 21000, 22000, 27000, 19000, 23000, 17000, 19000, 21000,
19000, 28000, 21000, 34000, 19000, 26000, 21000, 27000, 16000, 19000, 22000, 18000,
22000, 23000, 14000, 19000, 21000, 23000, 12000, 28000, 21000, 19000, 24000, 26000 };
    demands = demand;
    costs = costmatrisi;
    distances = distancematrisi;
    firststeps = firststep;
    rentingcs = rentingc;
}
static void random(ref int[] dizi)
{
    for (int j = dizi.Length - 1; j >= 0; j--)
    {
        Random rnd = new Random();
        int k = rnd.Next(0, j);
        int tutucu = dizi[j];
        dizi[j] = dizi[k];
        dizi[k] = tutucu;
    }
}
static void assign(int[] cities, int number, int[,] costs, int[] firststeps, int[] order, int[]
demands, int[,] distances, ref int[] dizi1, ref int[] dizi2, ref int[] dizi3)
{
    int[] atama = new int[81];
    int[] kontrol = new int[81];
    int[] capacity = { 15000, 15000, 15000 };
    int[] closeness = new int[81];
    for (int i = 0; i < 81; i++)
    {
        int a = firststeps[order[0]] + distances[order[0], cities[i]] + costs[order[0], cities[i]];
        int b = firststeps[order[1]] + distances[order[1], cities[i]] + costs[order[1], cities[i]];
        int c = firststeps[order[2]] + distances[order[2], cities[i]] + costs[order[2], cities[i]];
        if (a < b && a < c)
            closeness[i] = order[0];
        else if (b < c && b < a)
            closeness[i] = order[1];
        else if (c < b && c < a)
            closeness[i] = order[2];
    }
    for (int i = 0; i <= number; i++)
    {
        for (int j = 0; j < 81; j++)
        {
            if (closeness[j] == order[i] && demands[cities[j]] <= capacity[i] && kontrol[j] != 1)
            {
                atama[j] = closeness[j];
                capacity[i] = capacity[i] - demands[cities[j]];
                kontrol[j] = 1;
            }
        }
    }
    for (int i = 0; i <= number; i++)
    {

```

```

        for (int j = 0; j < 81; j++)
        {
            if (demands[cities[j]] <= capacity[i] && kontrol[j] != 1)
            {
                atama[j] = order[i];
                capacity[i] = capacity[i] - demands[j];
                kontrol[j] = 1;
            }
        }
    }
    for (int i = 0; i < 81; i++)
        dizi1[i] = cities[i];
    for (int i = 0; i < 81; i++)
        dizi2[i] = atama[i];
    for (int i = 0; i < 3; i++)
        dizi3[i] = capacity[i];
}

static double calculate(int number, int[] assignorder, int[] assignedwh, int[,] costs, int[,]
distances, int[] demands, int[] firststeps, int[] order, int[] rentingscs)
{
    double cost = 0;
    for (int i = 0; i <= number; i++)
    {
        cost = cost + rentingscs[order[i]];
    }
    double service = 0;
    for (int i = 0; i < 81; i++)
    {
        cost = cost + costs[assignedwh[i], assignorder[i]] * demands[assignorder[i]] +
firststeps[assignedwh[i]] * demands[assignorder[i]];
    }
    double deviation1 = cost - 4428661;
    for (int i = 0; i < 81; i++)
    {
        service = service + distances[assignedwh[i], assignorder[i]] *
demands[assignorder[i]];
    }
    double deviation2 = service - 2750558;
    double objective = 0.5 * deviation1 + 0.5 * deviation2;
    return objective;
}

static void Main(string[] args)
{
    Stopwatch watch = new Stopwatch();
    watch.Start();
    int[,] maliyetler = new int[81, 81];
    int[,] uzakliklar = new int[81, 81];
    int[,] tabulist = new int[81, 81];
    int[] sehirler = new int[81];
    int[] depomaliyetleri = new int[81];
    int[] ilkasama = new int[81];
    int[] atamasirasi = new int[81];
    int[] atanandepo = new int[81];
    int[] talepler = new int[81];
    int[] sıra = new int[3];

```

```

int[] kapasite = new int[3];
int[] optkapasite = new int[3];
int[] optsira = new int[81];
int[] optatanand = new int[81];
double optimum = 50000000;
double w = 0;
double amac;
double cozum;
int dep1 = 0, dep2 = 0, dep3 = 0;
int a = 0, b = 0, c = 0;
int deg1 = -1, deg2 = -1;
int tutucu;
for (int deposayisi = 0; deposayisi <= 2; deposayisi++)
{
    for (int i = 0; i < 81; i++)
        sehirler[i] = i;
    random(ref sehirler);
    data(ref maliyetler, ref uzakliklar, ref talepler, ref ilkasama, ref depomaliyetleri);
    for (int i = 0; i < 3; i++)
        sıra[i] = sehirler[i];
    assign(sehirler, deposayisi, maliyetler, ilkasama, sıra, talepler, uzakliklar, ref
atamasirasi, ref atanandepo, ref kapasite);
    amac = calculate(deposayisi, atamasirasi, atanandepo, maliyetler, uzakliklar, talepler,
ilkasama, sıra, depomaliyetleri);
    cozum = amac;
    w = amac;
    Console.WriteLine("başlangıç çözüm = {0}", cozum);
    for (int tabusearch = 0; tabusearch < 50; tabusearch++)
    {
        for (int i = 0; i < 81; i++)
        {
            for (int j = 0; j < 81; j++)
            {
                tabulist[i, j] = tabulist[i, j] - 1;
            }
        }
        cozum = w;
        for (int i = 0; i <= deposayisi; i++)
        {
            for (int j = deposayisi + 1; j < 81; j++)
            {
                tutucu = sehirler[i];
                sehirler[i] = sehirler[j];
                sehirler[j] = tutucu;
                for (int m = 0; m < 3; m++)
                    sıra[m] = sehirler[m];
                assign(sehirler, deposayisi, maliyetler, ilkasama, sıra, talepler, uzakliklar, ref
atamasirasi, ref atanandepo, ref kapasite);
                amac = calculate(deposayisi, atamasirasi, atanandepo, maliyetler, uzakliklar,
talepler, ilkasama, sıra, depomaliyetleri);
                if (amac - cozum <= 0 && tabulist[i, j] <= 0 && tabulist[j, i] <= 0)
                {
                    cozum = amac;
                    deg1 = i;
                    deg2 = j;
                }
            }
        }
    }
}

```

```

        a = sıra[0];
        b = sıra[1];
        c = sıra[2];
    }
    if (tabulist[i, j] > 0 && tabulist[j, i] > 0 && amac < optimum)
    {
        cozum = amac;
        deg1 = i;
        deg2 = j;
        a = sıra[0];
        b = sıra[1];
        c = sıra[2];
    }
    tutucu = sehirler[i];
    sehirler[i] = sehirler[j];
    sehirler[j] = tutucu;
}
}
tutucu = sehirler[deg1];
sehirler[deg1] = sehirler[deg2];
sehirler[deg2] = tutucu;
for (int i = 0; i < 3; i++)
    sıra[i] = sehirler[i];
assign(sehirler, deposayisi, maliyetler, ilkasama, sıra, talepler, uzakliklar, ref
atamasirasi, ref atanandepo, ref kapasite);
tabulist[deg1, deg2] = 10;
tabulist[deg2, deg1] = 10;
Console.WriteLine("degisim {0} ile {1} çözüm {2}. depolar {3}-{4}-{5}", deg1, deg2,
cozum, a, b, c);
if (cozum <= optimum)
{
    optimum = cozum;
    dep1 = a;
    dep2 = b;
    dep3 = c;
    for (int i = 0; i < 81; i++)
    {
        optsıra[i] = atamasirasi[i];
        optatanand[i] = atanandepo[i];
    }
    for (int i = 0; i < 3; i++)
    {
        optkapasite[i] = kapasite[i];
    }
}
}
}
double gap = 100 * (optimum - 434507.5) / 434507.5;
string[] decoder = { "ADANA", "ADIYAMAN", "AFYON", "AĞRI", "AMASYA", "ANKARA",
"ANTALYA", "ARTVİN", "AYDIN", "BALIKESİR", "BİLECİK", "BİNGÖL", "BİTLİS", "BOLU",
"BURDUR", "BURSA", "ÇANAKKALE", "ÇANKIRI", "ÇORUM", "DENİZLİ", "DİYARBAKIR",
"EDİRNE", "ELAZIĞ", "ERZİNCAN", "ERZURUM", "ESKİŞEHİR", "GAZİANTEP", "GİRESUN",
"GÜMÜŞHANE", "HAKKARİ", "HATAY", "ISPARTA", "İÇEL", "İSTANBUL", "İZMİR", "KARS",
"KASTAMONU", "KAYSERİ", "KIRKLARELİ", "KIRŞEHİR", "KOCAELİ", "KONYA", "KÜTAHYA",
"MALATYA", "MANİSA", "KAHRAMANMARAŞ", "MARDİN", "MUĞLA", "MUŞ", "NEVŞEHİR",

```

```

"NiĞDE", "ORDU", "RİZE", "SAKARYA", "SAMSUN", "SİİRT", "SİNOP", "SİVAS", "TEKİRDAĞ",
"TOKAT", "TRABZON", "TUNCELİ", "ŞANLIURFA", "UŞAK", "VAN", "YOZGAT", "ZONGULDAK",
"AKSARAY", "BAYBURT", "KARAMAN", "KIRIKKALE", "BATMAN", "ŞIRNAK", "BARTIN",
"ARDAHAN", "İĞDIR", "YALOVA", "KARABÜK", "KİLİS", "OSMANİYE", "DÜZCE" };
    Console.WriteLine("depolar {0}, {1}, {2}", dep1, dep2, dep3);
    Console.WriteLine("\nBulunan en iyi çözüm {0}\nOptimum çözümünden uzaklık (yüzde)
{1:0.##}\nAçılan depolar : {2},{3},{4}", optimum, gap, decoder[dep1], decoder[dep2],
decoder[dep3]);
    for (int i = 0; i < 81; i++)
    {
        Console.WriteLine("{0} - {1} depo", decoder[optsira[i]], decoder[optatanand[i]]);
    }
    for (int i = 0; i < 3; i++)
    {
        Console.WriteLine("{0}. depo kalan kapasite {1}", i + 1, 15000 - optkapasite[i]);
    }
    watch.Stop();
    Console.WriteLine("Program Süresi: {0}", watch.Elapsed.Milliseconds);
}
}
}

```