

**BAŐKENT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**GÜNLÜK GAZETE DAĞITIM PLANLAMASI**

**TUSAN DERYA**

**YÜKSEK LİSANS TEZİ**

**2008**

# **GÜNLÜK GAZETE DAĞITIM PLANLAMASI**

## **DAILY NEWSPAPER DISTRIBUTION PLANNING**

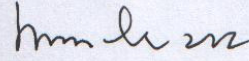
**TUSAN DERYA**

Başkent Üniversitesi  
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin  
ENDÜSTRİ Mühendisliği Anabilim Dalı İçin Öngördüğü  
YÜKSEK LİSANS TEZİ  
olarak hazırlanmıştır.

2008

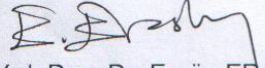
Fen Bilimleri Enstitüsü Müdürlüğü'ne,  
Bu çalışma, jürimiz tarafından **ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI'nda**  
**YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan



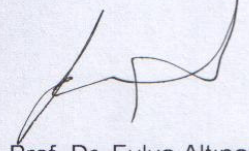
Prof. Dr. İmdat KARA

Danışman



Yrd. Doç. Dr. Ergün ERASLAN

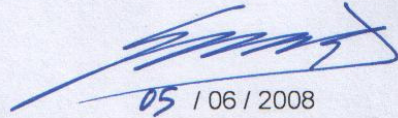
Üye



Prof. Dr. Fulya Altıparmak

ONAY

Bu tez 15 / 05 / 2008 tarihinde Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri  
üyeleri tarafından kabul edilmiştir.



05 / 06 / 2008

Prof. Dr. Emin AKATA

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜR

## **TEŐEKKÜR**

Bugüne kadar maddi ve manevi desteklerini benden esirgemeyen ailem başta olmak üzere, bu çalışmanın hazırlanması sırasında karşılaşılan güçlüklerin aşılmasında değerli fikir ve önerileriyle bana yol gösteren danışman hocam Sayın Yrd. Doç. Dr. Ergün ERASLAN'a ve hocam Sayın Prof. Dr. İmdat KARA'ya, ayrıca manevi desteęiyle beni hiçbir zaman yalnız bırakmayan çok değerli arkadaşım Utkan BAL'a teşekkürü bir borç bilirim.

## ÖZ

### GÜNLÜK GAZETE DAĞITIM PLANLAMASI

Tusan DERYA

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Gazete Dağıtım Problemi (GDP), her gün basım merkezlerinde basılan gazetelerin, dağıtım araçları kullanılarak, bazı özel kısıtlar altında bayilere taşınması problemidir. Bu çalışmada, Türk basın sektöründe gazete dağıtım hizmeti veren bir firmanın dağıtım araçlarının rotalarının belirlenmesi problemi ele alınmıştır. Problem, talepleri bilinen coğrafi olarak dağınık başbayilere hizmet vermek için merkezi bir basım merkezinden hareket ederek, depoya geri dönen homojen bir araç filosu tarafından katedilen toplam mesafeyi enküçükleyecek, kapasite ve zaman kısıtlarını aşmayan optimum dağıtım rotalarının belirlenmesi olarak tanımlanıp, çözümü için bir tamsayılı doğrusal karar modeli önerilmiştir. Fakat, problem boyutunun artmasıyla birlikte çözüm süresinin üstel olarak artması sonucu, çok bayili bölgelerin yaklaşık çözümlerini makul sürelerde bulmak amacıyla melez bir sezgisel algoritma da önerilmiştir. Deney tasarımı ile belirlenen en iyi parametre seti, dağıtım kanalı farklı büyüklükte olan bölgeler üzerinde test edilmiş ve elde edilen sonuçlar matematiksel model çözümleriyle, çözüm kalitesi ve çözüm süresi açısından karşılaştırılmıştır.

**ANAHTAR KELİMELEER:** Gazete Dağıtım Problemi, Zaman ve Kapasite kısıtlı Araç Rotalama Problemi, Tamsayılı Programlama, Melez Algoritma

**Danışman:** Yrd. Doç. Dr. Ergün ERASLAN, Başkent Üniversitesi, Endüstri Mühendisliği Bölümü.

## **ABSTRACT**

### **DAILY NEWSPAPER DISTRIBUTION PLANNING**

Tusan DERYA

Baskent University, Institute of Science

The Department of Industrial Engineering

Newspaper Distribution Problem (NDP), is the problem of transporting the newspaper from the printing centers to the news agents with distribution vehicles under special constraints. In this study, we determine the routes of vehicles of a leading newspaper distributor company in Turkish press sector. The problem is defined as determining optimal distribution routes for a fleet of homogeneous vehicles, starting and ending at the printing center that is required to serve a number of geographically dispersed news agents with known demands under capacity and time constraints, while minimizing the total distribution cost. For the solution of the problem we propose an integer linear programming model. The solution time of the problem grows exponentially as the number of distribution point increase; Hence, a hybrid algorithm is proposed to obtain a close-to-optimal solution in reasonable time for the regions which have more than 40 news agents. The performance of the hybrid algorithm, for which the best parameter set is determined by design analysis, is tested on the problems with different sizes. The results are compared with those of the mathematical model in terms of solution quality and computation time.

**KEYWORDS:** Newspaper Distribution Problem, Capacitated Vehicle Routing Problem with Time Deadlines, Integer Programming, Hybrid Algorithm

**Thesis Supervisor:** Yrd. Doç. Dr. Ergün ERASLAN, Baskent University, Department of Industrial Engineering.

# İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ .....	i
ABSTRACT .....	ii
İÇİNDEKİLER LİSTESİ .....	iii
ŞEKİLLER LİSTESİ .....	iv
ÇİZELGELER LİSTESİ .....	v
SİMGELER VE KISALTMALAR LİSTESİ .....	vi
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. GÜNLÜK GAZETE DAĞITIMI .....</b>	<b>4</b>
2.1. Gazete İşletmelerinde Dağıtım Fonksiyonu .....	4
2.2. Türkiye’de Gazete Dağıtım Yapısındaki Gelişmeler .....	5
2.3. Gazete Dağıtım Aşamaları .....	6
2.4. Gazete Dağıtımını İle İlgili Araştırmalar.....	7
<b>3. BİR DAĞITIM ŞİRKETİNDE GAZETE DAĞITIM PLANLAMASI .....</b>	<b>9</b>
3.1. Mevcut Durum ve Problem Tanıtımı .....	9
3.2. Veri Derlenmesi .....	12
3.2.1. Araç sayısı .....	12
3.2.2. Başbayilerin yeri ve talep miktarı .....	12
3.2.3. Uzaklık ve zaman değerleri .....	14
3.3. Matematiksel Model .....	15
3.3.1. Karar modelinin geliştirilmesi .....	15
3.3.2. Modelin bölge temelinde çözümü .....	17
3.3.3. Modelin uygulamada sınırları .....	22
3.4. Gazete Dağıtım Planlaması İçin Melez Bir Sezgisel Algoritma .....	23
3.4.1. Komşu üretme stratejisi .....	25
3.4.2. Yeniliğe dayanan hafıza fonksiyonları .....	30
3.4.3. Tavlama planı ve kabul fonksiyonu .....	33
3.4.4. Uzun dönemli tabu arama .....	34
3.4.5. Deneysel çalışma .....	37
3.4.5.1. Kurucu sezgisel için uygun parametre setinin belirlenmesi	37
3.4.5.2. TB algoritması için uygun parametre setinin belirlenmesi	39
3.4.6. Karar modeli ve melez algoritma sonuçları .....	42
<b>4. SONUÇ VE ÖNERİLER .....</b>	<b>43</b>
KAYNAKLAR LİSTESİ .....	45
EK AÇIKLAMALAR LİSTESİ .....	48
EKLER LİSTESİ .....	66

## ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil 3.1 Ele alınan firmanın dağıtım ağı .....	10
Şekil 3.2 Sony Route Planner yazılımı .....	14
Şekil 3.3 Paralel turlu kurucu sezgisel algoritması .....	24
Şekil 3.4 Gazete dağıtım problemi için çözüm örneği .....	25
Şekil 3.5 Tekli değişim hareketi ile elde edilen bir komşu çözüme örnek ...	26
Şekil 3.6 İkili değişim hareketi ile elde edilen bir komşu çözüme örnek .....	26
Şekil 3.7 Tekli değişim hareketi ile elde edilen komşu çözümler .....	27
Şekil 3.8 Tur boşaltma mekanizmasının adımları .....	29
Şekil 3.9 Melez algoritmanın adımları .....	35
Şekil 5.1 İniş algoritması .....	52
Şekil 5.2 Tavlama benzetimi algoritması .....	53
Şekil 5.3 Komşu arama metodu adımları .....	55
Şekil 5.4 Kısa dönem hafızalı tabu arama algoritması .....	57
Şekil 5.5 Tabu aramada kuvvetlendirme yaklaşımı algoritması .....	63
Şekil 5.6 Değişken komşuluk arama algoritması .....	65



## ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 Basım merkezlerinin bilgileri .....	9
Çizelge 3.2 Bölgelere göre araç sayının alt ve üst sınırları .....	12
Çizelge 3.3 Firmanın başbayileri ve talep miktarları .....	13
Çizelge 3.4 Geliştirilen karar modeli sonuçları .....	18
Çizelge 3.5 Adana bölgesine dağıtım yapan araçların zaman çizelgesi .....	20
Çizelge 3.6 Ankara bölgesine dağıtım yapan araçların zaman çizelgesi .....	20
Çizelge 3.7 Antalya bölgesine dağıtım yapan araçların zaman çizelgesi .....	20
Çizelge 3.8 İstanbul bölgesine dağıtım yapan araçların zaman çizelgesi ...	21
Çizelge 3.9 İzmir bölgesine dağıtım yapan araçların zaman çizelgesi .....	21
Çizelge 3.10 Trabzon bölgesine dağıtım yapan araçların zaman çizelgesi ..	21
Çizelge 3.11 Mevcut sistem ile karar modeli sonuçlarının karşılaştırılması ..	22
Çizelge 3.12 Paralel turlu kurucu sezgisel parametre değerleri .....	38
Çizelge 3.13 Paralel turlu kurucu sezgisel deney sonuçları .....	38
Çizelge 3.14 Tavlama Benzetimi parametre değerleri .....	39
Çizelge 3.15 Tavlama Benzetimi deney sonuçları .....	40
Çizelge 3.16 Paralel turlu kurucu sezgisel için en iyi parametre seti .....	41
Çizelge 3.17 Tavlama Benzetimi için en iyi parametre seti .....	41
Çizelge 3.18 Tabu Arama için parametre seti .....	41
Çizelge 3.19 Melez algoritma sonuçları .....	41
Çizelge 3.20 Karar modeli ve melez algoritma sonuçlarının karşılaştırılması	42

## SİMGELER VE KISALTMALAR

$G(V, A)$	$V$ düğüm ve $A$ ayrıt kümesine sahip bir şebekenin graf gösterimi
$m$	Araç sayısı
$V$	Düğüm kümesi
$A$	Ayrıt kümesi
$n$	Serimdeki toplam düğüm sayısı
$Q$	Her bir aracın (ortak) kapasitesi (gazete)
$TC$	Bir aracın gidebileceği en fazla süre (dakika)
$q_i$	$i$ . başbayinin günlük ortalama talebi
$c_{ij}$	Başbaya $i$ ile başbaya $j$ arasındaki uzaklık (km)
$t_{ij}$	Başbaya $i$ ile başbaya $j$ arasındaki seyahat süresi (dakika)
$x_{ij}$	1, araç $i$ . başbayiden $j$ . başbaya giderse; 0, diğer durumda
$u_i$	$i$ . başbayiden çıkışta araçta bulunan gazete miktarı
$v_i$	$i$ . başbaya gelen aracın, başlangıç noktasından itibaren geçirdiği süre
$T$	Sıcaklık
$E_i$	$i$ . durumundaki enerji
$k_B$	Boltzmann sabiti
$S$	Çözüm uzayı
$\Delta$	Enerji farkı
$f(x)$	Mevcut araç rotalamanın maliyeti
$e$	Herhangi bir hareket niteliği
$M$	Markov zinciri uzunluğu
$\bar{e}$	Herhangi bir hareket niteliğinin tersi
$P_{ij}(t)$	$t$ . denemede $i$ durumundan $j$ durumuna geçme olasılığı
$P_T\{X=i\}$	$T$ sıcaklığında $i$ durumunda olma olasılığı
$N(H, x)$	Mevcut araç rotalamanın $H$ hafızasında tutulan komşuları
$N(x)$	Mevcut araç rotalamanın komşuları
$T(m)$	$m$ . markov zincirindeki sıcaklık
$T_0$	Sıcaklık parametresinin başlangıç değeri
$T_S$	Sıcaklık parametresinin son değeri
$T_m$	Sıcaklık parametresinin mevcut değeri
$\alpha$	Soğutma oranı
$k$	Her sıcaklıkta kabul edilecek hareket sayısı
$x$	Mevcut araç rotalama
$x'$	Mevcut araç rotalamanın bir komşusu
$x_0$	Başlangıç araç rotalama
$x_{eç}$	Bulunan en düşük maliyetli araç rotalama
$h$	Herhangi bir hareket
$N_k$	Komşuluk yapılarının kümesi
$N_k(s)$	$s$ 'nin $k$ . komşuluğuna ait çözüm kümesi
$D_{ij}$	$i, j$ başbayiler arasındaki uzaklık (km)
$b_k$	$k$ . başbaya varış zamanı
$b_{jk}$	$j$ . başbayinin tura eklenmesiyle $k$ . başbaya varış zamanı
$\beta_1$	uzaklık için ağırlık değeri
$\beta_2$	zaman için ağırlık değeri
$td$	Tekli değişim hareketi
$id$	İkili değişim hareketi
$tdts$	Tekli değişim hareketi için tabu süresinin uzunluğu
$idts$	İkili değişim hareketi için tabu süresinin uzunluğu

$tdtb[i]$	$i$ . düğümün $td$ hareketi ile yeri değiştirilmesinin tabu başlangıcı
$idtb[i, j]$	$i, j$ düğümlerinin $id$ hareketi ile yerlerinin değiştirilmesinin tabu başlangıcı
$K_{tdts}$	tekli değişim hareketi için kısa tabu süresi
$O_{tdts}$	tekli değişim hareketi için orta tabu süresi
$U_{tdts}$	tekli değişim hareketi için uzun tabu süresi
$K_{idts}$	ikili değişim hareketi için kısa tabu süresi
$O_{idts}$	ikili değişim hareketi için orta tabu süresi
$U_{idts}$	ikili değişim hareketi için uzun tabu süresi
$k_r$	Kısa tabu sürelerini elde etmek için kullanılan katsayı
$b_r$	Uzun tabu sürelerini elde etmek için kullanılan katsayı
$elu$	Elit liste uzunluğu
$d_{ij}$	$i$ ile $j$ arasındaki uzaklık (km)
SÇ	Sezgisel algoritma ile elde edilen çözüm
OPT	Problemin optimum çözümü
OS	Optimumdan sapma oranı (%)
GDP	Gazete Dağıtım Problemi
KARP	Kapasiteli Araç Rotalama Problemi
ÇDZKARP	Çok Depolu Zaman ve Kapasite kısıtlı Araç Rotalama Problemi
TDZKARP	Tek Depolu Zaman ve Kapasite kısıtlı Araç Rotalama Problemi
MKARP	Mesafe ve Kapasite kısıtlı Araç Rotalama Problemi
YAYSAT	Yayın Satış Pazarlama ve Dağıtım A.Ş.
GAMEDA	Gazete Mecmua Dağıtım A.Ş.
BBD	Birleşik Basın Dağıtım A.Ş.
TA	Tabu Arama
TB	Tavlama Benzetimi
DKA	Değişken Komşuluk Arama
GSP	Gezgin Satıcı Problemi
SGSP	Simetrik Gezgin Satıcı Problemi

## 1. GİRİŞ

Gazete, çağdaş toplumdaki kitle iletişim araçlarının en önemlilerinden biridir. Gazete, belirli boyutlu ve sınırlı sayfalı olup birbirini izleyen numaralarla yayınlanan, günlük olaylara ilişkin çeşitli yazı, resim ve ilânları içeren belirli bir eder karşılığında satılan, genellikle günlük süreli bir basım ürünüdür.

İşletmenin iktisadi mal ve hizmet üretmek ve/veya pazarlamak için faaliyette bulunan kuruluş olduğundan hareketle; gazete işletmesini, haber ve fikir üreten iktisadi kuruluş şeklinde tanımlamak mümkündür. Gazete işletmeleri ilke olarak hizmet üretmektedirler. Gazete işletmelerini diğer işletmelerden ayırdeden nitelik ise, görünürde tek bir mal olmasına karşılık, çok farklı iki pazarlama teşkilatı gerektirmesidir. Gazete işletmelerinde iki farklı hizmet ortaya konularak, iki farklı müşteri grubu (okuyucular ve reklam verenler) elde tutulmaya çalışılır. Bu işletmeler reklam sahiplerine çekici görünüp, daha çok gelir elde edebilmek için iyi bir ürün ortaya koyarak, çok sayıda okuyucuya ulaşabilmelidir.

Zamanla yarışmak ve bu yarışı kazanmak zorunda olan gazete işletmeleri iki aşamalı bir engeli aşmak zorundadırlar. Bu aşamalardan ilki, baskıya hazırlık ve baskı aşamasıdır. Bu konuda, Türkiye Avrupa'nın önde gelen makine parkı ve son teknolojiye sahip ülkelerinden birisi olup, teknolojiye ayak uydurmuş durumdadır. Ancak ikinci aşama olan dağıtım konusunda gelişmiş ülkelerle karşılaştırıldığında geri kalmış durum gözükmektedir. Artan rekabet şartları içerisinde karlılıklarını korumak ve devamlılıklarını sağlamak isteyen gazete işletmeleri için maliyetlerin en aza indirilmesi kaçınılmaz bir zorunluluktur. Gazete işletmelerinin toplam maliyetleri içerisinde yer alan önemli kalemlerden olan dağıtım maliyetlerinin enküçüklenmesi bu açıdan özel önem arz etmektedir. Ayrıca, gazete satışını dolayısı ile tirajını etkileyen faktörlerin başında dağıtım kanalı gelmektedir. Basın sektöründeki yoğun ve titiz çalışmaların ardında yetersiz bir dağıtım kanalı olması tüm çalışmaların değerini geçersiz kılmaktadır. Basın sektöründe üretilen gazete çok kısa ömürlüdür. Bir gazetenin ömrü en fazla 24 saattir. İlk 12 saatte gazete satılmamışsa, satış şansı hemen hemen hiç yoktur. Gazete bu sürede

okuyucusuna ulařtırılmadıđı veya satılmadıđı takdirde deęeri hurda kađıt fiyatına dūřmektedir.

En genel tanımıyla Gazete Dađıtım Problemi (GDP), her gūn basım merkezlerinde basılan gazetelerin, dađıtım araları kullanılarak, bazı Őzel kısıtlar altında bayilere tařınması problemidir. Buradaki ama, genellikle aralar tarafından katedilen toplam mesafeyi ve/veya kullanılan ara sayısını enkūklemektir. Bu probleme iliřkin karřılařılabilir kısıtlar, her aracın mevcut kapasitesinin ařılmaması, araların yolda geirecekleri toplam sūrenin ŐngŐrūlen belli bir deęeri ařmaması ve gazetelerin belli zaman aralıklarında belli noktalara dađıtılması řeklinde sıralanabilir.

GDP, bazı yŐnleriyle yŐneylem arařtırması literatūrūnde iyi bilinen Kapasiteli Ara Rotalama Problemine (KARP) benzemektedir. Biimsel olarak KARP; merkezi bir depoda teslimat iřini gerekleřtiren  $m$  tane  $Q$  kapasiteli  $V$  hızına sahip araların, o cođrafî alanda dūzgūn bir řekilde farklı noktalara dađılmış  $n$  tane mūřterinin taleplerini en az maliyetle karřılaması iin kullanması gereken en uygun rotalarının bulunması olarak tanımlanır. Talepleri karřılamak iin tūm araların kullanılması durumunda  $m$  tane rota bulunur.

KARP'nin en Őnemli uzantılarından birisi zaman kısıtlamalı olanıdır. Bu tūr kısıtlamalı problemlerde her mūřteriye, verilen zaman aralıđında servis yapılması řartı vardır. Dolayısıyla servise bařlama zamanının, zaman kısıtının bařladıđı zamana eřit ya da ondan būyūk olması gerekir. Ayrıca varıř zamanı da zaman kısıtının sona erdiđi zamandan kūuk olmalıdır. Talep noktasına verilen zamandan erken ulařılması durumunda ara beklemek zorundadır. GDP'de ise zaman kısıtı aralık olarak ele alınmamaktadır. Araların sadece Őnceden belirlenen zaman kısıtının sona erdiđi zamana kadar talep noktalarına ulařma zorunlulukları vardır.

Bu alıřmada, 6 řehirde basım merkezi bulunan, Tūr basım sektörūnde gazete dađıtım hizmeti veren bir firmanın gazete dađıtım problemi incelenerek, tamsayılı programlama yardımıyla dađıtım araları iin, en iyi rotalama planının bulunması

hedeflenmiştir. Daha sonra, bulunan araç rotaları üzerinden hangi aracın ne zaman yüklenip, yola çıkması gerektiği sorusuna cevap aranmıştır. Fakat problemin NP-zor oluşu ve problem boyutunun artmasıyla birlikte çözüm süresinin üstel olarak artması sonucu, çözüm için yapılan çalışmaları daha çok sezgisel yöntemler üzerinde yoğunlaşmasına neden olmuştur. Bunun için tamsayılı doğrusal karar modelin mantıksal zaman dilimi içerisinde cevap veremediği çok boyutlu bölgelerin yaklaşık çözümlerini bulmak amacıyla, yerel aramaya dayalı algoritmalar arasında yer alan Tabu Arama (TA), Tavlama Benzetimi (TB) ve Değişken Komşuluk Arama (DKA) algoritmaların karması olan bir melez algoritma da önerilmiştir. Ayrıca önerilen melez algoritma içerisinde, literatürde genellikle kullanılan her aracın oluşturduğu turun iyileşmesini sağlayan 2-opt sezgiseli yerine, Concorde adında Simetrik Gezgin Satıcı Probleminin (SGSP) optimal çözümünü kısa sürede veren bilgisayar kodu kullanılmıştır.

İkinci bölümde, gazete işletmelerinde dağıtım fonksiyonu ile ilgili bilgiler, gazete dağıtımının tarihçesi, Türkiye'deki gazete dağıtım aşamaları ve gazete dağıtım ile ilgili araştırmalar ele alınmaktadır.

Üçüncü bölümde, ele alınan gazete dağıtım probleminin çözümü için önerilen tamsayılı doğrusal karar modeli ve melez algoritma anlatılmıştır. Melez algoritmanın uygun parametre setlerinin belirlenmesi amacıyla deneysel çalışma yapılmıştır. Daha sonra, melez algoritmanın buldukları sonuçlar, çözüm kalitesi ve çözüm zamanı açısından tamsayılı doğrusal karar modelinin bulduğu sonuçlara göre karşılaştırılmaktadır.

Son olarak, sonuç ve öneriler bölümünde, çalışmadan elde edilen sonuçlar değerlendirilmektedir.

## 2. GÜNLÜK GAZETE DAĞITIMI

### 2.1. Gazete İşletmelerinde Dağıtım Fonksiyonu

Dağıtım, belli merkezlerde toplanıp sınıflandırılan malların farklı tüketim ünitelerinin farklı tüketim modellerine göre, talep edildikleri yerde, talep şartlarına uygun olarak hazır bulundurulması olarak tanımlanmaktadır. Gazete dağıtımı, okuyucunun satın almayı istediği gazeteyi, istediği yerde ve istediği zaman bulabileceği şekilde düzenlenmelidir. Dağıtım faktörü, gazetenin stok edilememe özelliğinden dolayı ayrı bir önem arz etmektedir. Dağıtımda temel ilke, gazetenin asgari maliyetle okuyucuya zamanında ulaşmasıdır. Vakit ve nakit gazete dağıtımında iki önemli unsurdur. Bu dengeyi sağlamada, dağıtıma ilişkin hedeflerin belirlenmesi gerekir.

Gazete işletmeleri, basılan gazetelerin okuyucuyla buluşturulmasında önemli ölçüde araçlardan faydalanmaktadır. Dağıtım kanalı, gazete işletmeleri için en kritik unsurdur. Dağıtım kanalında yer alan araçlar, gazete işletmelerinin ürünlerini okuyucuya ulaştırarak kazanç elde ederken, aynı zamanda okuyucuların ihtiyaç duyduğu ürünleri kolay, çabuk ve arzu edilen şekilde okuyucuya ulaştırılmasını sağlamaktadırlar. Dağıtım organizasyonu, dağıtım şirketi → başbayı → bayi şeklinde gerçekleştirilir. Dağıtım şirketleri tek bir gazete işletmesine bağlı olarak çalışmamaktadırlar. Hedef kitlesi, pazarlama amaçları, satış uygulamaları birbirinden farklı olan yayınların, aynı işleyiş içinde farklı dağıtım amaç ve politikaları uygulaması, bu farklılıkları yansıtmak açısından zor olabilmektedir.

Gazete işletmeleri için dağıtım önemli bir maliyet ögesidir. Türkiye'de günlük gazetelerin çoğu maliyetlerinin altında fiyatlarla dağıtım şirketlerine teslim edilmektedir. Başbayiler, yayınların bir aylık satış bedellerini dağıtım şirketlerine teminat olarak ödemektedirler. Dağıtım şirketleri ise, bu paranın bir kısmını gazete işletmelerine başbayilik teminatı olarak vermektedir.

## 2.2. Türkiye’de Gazete Dağıtım Yapısındaki Gelişmeler

Cumhuriyet’in ilanından 1959 yılına kadar İstanbul gazeteleri, gazete şirketlerinin kendi olanakları içerisinde kiralık araçlar, otobüs, tren veya vapurlarla taşınmaktaydı. 1950’li yıllarda Anadolu’ya gazete, en erken 3 gün geçikme ile gelmekteydi. Gazeteler, ayrı bir dağıtım kuruluşu olmadığı için belirli bayilikler ve koltukçu olarak isimlendirilen seyyar gazete bayileri tarafından satılmaktaydı.

Okuyucuya daha çabuk varmak, daha çok yerde satışı sağlamak amacıyla 4 Eylül 1959’da İstanbul’da ilk dağıtım şirketi olan Gazete Meczua Dağıtım Şirketi (GAMEDA), Milliyet, Hürriyet, Cumhuriyet, Dünya, Tercüman ve Yeni Sabah gazetelerinin ortaklığıyla kurulmuştur. 1959 yılında GAMEDA’nın kuruluşundan kısa bir süre sonra Trakya ve Marmara bölgelerinde çalışmalarında aksaklık görülen bayilerin bir kısmı değiştirilerek yerine yenileri kurulmuştur.

1963 yılında Hürriyet Gazetesi, GAMEDA’dan ayrılarak Web Ofset, bayilerin büyük bir kısmı ve bazı Hürriyet mensupları ile Hür Dağıtım’ı kurmuştur. Türkiye genelinde her dağıtım şirketinin ürünlerini satan bayiler birbirinden ayrılmış, karşı şirketin ürününü dağıtan bayilerin sözleşmeleri iptal edilmiştir. GAMEDA yönetim aksaklıkları nedeni ile 1991 yılında iflas etmiştir. GAMEDA’nın iflas etmesi ile birlikte 24 Eylül 1992’de Milliyet, Türkiye ve Cumhuriyet yayın gruplarınınca Yaysat Yayın Satış Pazarlama ve Dağıtım A.Ş. (YAYSAT) kurulmuştur. Ekim 1993’de Hürriyet ve Sabah gruplarının %50’şer hisseleri ile Birleşik Basın Dağıtım (BBD) Şirketi kurulmuştur. 1994 yılında Hürriyet Gazetesi, YAYSAT şirketine katılmıştır. YAYSAT; Milliyet, Hürriyet, Türkiye, Cumhuriyet, Son Havadis, Posta, Evrensel, Global, Orta Doğu, Milli Gazete, Gazete Ege, Bursa 2000, Bursa Haber v.s. olmak üzere 150 yayınevine ait 35 günlük gazetenin dağıtım ve pazarlamasını yapmaktadır.

Biryay adı altında, BBD ve YAYSAT’ın %50’şer ortaklığı ile Haziran 1996 yılında yeni bir şirket kurulmuştur. Böylece bu iki dağıtım şirketi şekil olarak Biryay’a bağlanmıştır. Bu dağıtım şirketleri kendi basın grupları dışındaki yayınların dağıtımında Biryay tarafından taşeron olarak kullanılmaktadır.



Dost Dağıtım, 1997 yılında Akşam Gazetesi'nin BBD tarafından dağıtımı durdurulunca kurulmuştur. Aslı Yayıncılık grubuna ait tüm gazete ve dergiler için Dost Dağıtım A.Ş. oluşturularak, başta Akşam gazetesi olmak üzere bu gruba ait Güneş, Önce Vatan, Dost, Küp, Alem Gazete ve dergilerinin dağıtımı bu şirket tarafından sağlanmaktadır.

1997 yılından sonra Star gazetesinin yayın hayatına atılması ve dağıtımı konusunda YAYSAT ve BBD ile sorun çıkması sonucu yeni bir dağıtım anlayışı ile gazete dağıtımını kendisi yapmaya ve spot noktalarda satış yaptırmaya başlamıştır. Ağustos 2002 tarihinde birlikte yıllardır işbirliği içerisinde olan BBD ve YAYSAT dağıtım yollarını ayırarak iki rakip kuruluş haline gelmiştir.

### **2.3. Gazete Dağıtım Aşamaları**

Bayilere kurulan bilgisayarlı iletişim ve hesap sistemi kullanılarak, bayi talepleri akşamdan alınmaktadır. Gazetelerin haber akışı durmaksızın devam ederken, ertesi günün gazetesi genelde saat 00:00'de baskıya girmektedir. Eğer önemli haber ve acil gelişmeler varsa, baskı saat 08:00'e kadar ertelenmektedir. Normal koşullar altında ve talep miktarına bağlı olmak üzere, gazetelerin baskısı genellikle en geç saat 03:00 civarında tamamlanmaktadır. Baskısı tamamlanan gazeteler şirketin dağıtım araçlarına yüklendikten sonra, ilgili bölgelerdeki başbayilere gönderilmektedir.

Dağıtım aracı kendi bölgesinde bulunan başbayilere gazetelerini teslim etmektedir. Başbayide çalışan işçiler tarafından, 50'lik 100'lük paketler teslim alındıktan sonra, akşamdan hazırlanmış satış kotalarına göre her bayiye, taşeron dağıtım araçlarıyla veya başbayinin kendi araçlarıyla gazete dağıtımı yapılmaktadır. Son satıcı tarafından teslim alınan gazeteler tezgaha yerleştirilip, okura sunulmaktadır.

## 2.4. Gazete Dağıtımı İle İlgili Araştırmalar

Bu bölümde, yöneylem araştırması literatüründe incelenmekte olan GDP ile ilgili yapılan çalışmalardan bazı örnekler verilecektir.

Holt ve Watts [1] tarafından gazete dağıtımı konusunda yapılan bu çalışma, daha sonraki araştırmalara öncü niteliğinde bir çalışmadır. Bu çalışmada, GDP'nin tanımı ve kısıtlarından bahsetmişlerdir.

Mantel ve Fontein [2], Dutch gazetesinde bir çalışma yapmışlardır. Bu çalışmada, 60 talep noktasına 1100 gazete kapasiteli araçlarla gazete dağıtımı için, zaman kısıtı ve araç kapasitesini gözönünde bulundurarak, Clarke ve Wright tasarruf sezgiseli ile dağıtım planı geliştirmişlerdir. İki nokta arasındaki uzaklığı Floyd yöntemine göre hesaplamışlardır. Ayrıca gazete dağıtımı için matematiksel model önermişler, ancak bu modeli çözümde kullanmamışlardır.

Hunter ve Buer [3], günlük 37.000 adet gazeteyi 14 tane 1 tonluk araçlar kullanarak, 513 başbahiye saat 04:30'a kadar dağıtmak için önce rotala, sonra grupla çözüm yaklaşımını kullanmışlardır. Bu çalışmada, 2 nokta arasındaki uzaklığı Brimberg ve Love ağırlıklandırma yöntemi ile hesaplamışlardır. Problemi zaman kısıtlı araç rotalama problemi olarak ele alıp, Greedy ve Or-opt sezgiselleriyle çözmeye çalışmışlardır.

Ree ve Yoon [4], birden fazla basım merkezinden bayilere gazete dağıtım problemini, çok araçlı zaman kısıtlı dağıtım güzargahı bulma problemi olarak ele alıp, önce grupla sonra rotala yaklaşımıyla 2 aşamada çözmeye çalışmışlardır. İlk aşamada, Genelleştirilmiş Atama Problemini çözmek için 1989'da Guignard ve Rosenwein tarafından önerilen dal-sınır tekniğine dayalı bir sezgisel yöntemi kullanarak, başbayileri basım merkezlerine atamışlardır. İkinci aşamada ise, TB kullanılarak her salkımda oluşan turları kendi içinde çözmüşlerdir.

Dillmann vd. [5], gazete dağıtım problemini ve kısıtlarını tanıttıktan sonra, Almanya'da 200 yayınevi, 96 başbayı ve 110.000 son satıcıdan oluşan gazete dağıtım problemini, zaman kısıtlı araç rotalama problemi olarak ele alıp, çözmek için önce grupta sonra rotala çözüm yöntemini önermişlerdir. Grupların oluşturulması için Sweeping algoritmasından yararlanmışlardır. Gruplar içindeki turları belirlemek için, Clarke ve Wright tasarruf yöntemini kullanmışlardır.

Buer vd. [6], TB ve TA modern sezgisellerini kullanarak, gazete dağıtım probleminin çözümünü aramışlardır. İki modern algoritma için parametre setleri ve değerleri değiştirilerek, altı farklı algoritma geliştirmişlerdir. Elde edilen altı farklı sezgiselin sonuçlarına bakıp, hangi sezgiselin daha iyi sonuç verdiğini tartışmışlardır. TB'nin parametre seti doğru seçildiğinde, TA'dan daha iyi çözümler elde ettiği sonucuna varmışlardır.

Son zamanlarda yapılan çalışmaların birinde, Song vd. [7], 3 basım merkezi, 400 başbayı bulunan Chosun gazetesinde, başbayileri basım merkezlerine atama problemi, optimum dağıtım rotalarını bulma problemini ve araç çizelgeleme problemini ele almıştır. Başbayileri basım merkezlerine atama problemini çözmek için Pişmanlık Matrisi Hesaplama (Regret Distance Calculation) algoritması, araç rotalarını belirlemek için Ağırlıklandırılmış Tasarruf (Weighted Savings) algoritmasını ve araç çizelgeleme problemini çözmek için Değiştirilmiş Öncelikli Rota (Modified Urgent Route First) algoritması kullanmışlardır.

Başka bir çalışmada, Russell vd. [8], 47 araçla yaklaşık 150.000 adet gazeteyi 200 başbaya dağıtım yapan Tulsa World gazetesinde bir çalışma yapmışlardır. Problemi zaman kısıtlı Açık Araç Rotalama Problemi olarak ele almışlardır. Problemin çözümü için dinamik tabu liste yapısını kullanan TA modern sezgiselini kullanmışlardır.

### 3. BİR DAĞITIM ŞİRKETİNDE GAZETE DAĞITIM PLANLAMASI

#### 3.1. Mevcut Durum ve Problemin Tanımı

Ele alınan firma, 2001 yılından bu yana basın sektöründe gazete dağıtım işinde faaliyet göstermektedir. Türkiye genelinde 28 ulusal gazetenin satış, satış planlaması ve dağıtımından sorumlu olarak çalışmaktadır. Firmanın 60 araçlık filosu her gün 27.696 km yol katederek, Türkiye genelinde 188 başbayaie ulaşmaktadır. Gazeteler, 16 tonluk (50.000 gazete) kapasitelere sahip araçlarla Türkiye'nin en uç noktalarına dağıtılmaktadır. Ayrıca İstanbul, Ankara, İzmir, Adana, Antalya ve Trabzon'da konumlandırılmış basım merkezleriyle dağıtımını yaptığı gazetelerin baskı işlerini de yapmaktadır. Çizelge 3.1'de basım merkezlerinin bilgileri verilmiştir.

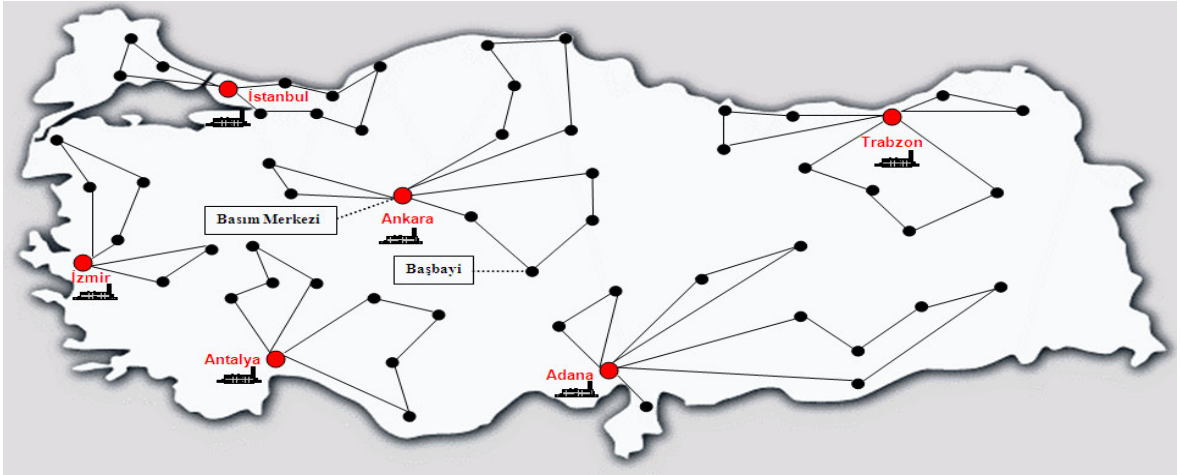
Çizelge 3.1: Basım merkezlerinin bilgileri

No	Bölge	İlçe	Kapasite (gazete/saat)	Dolum liman sayısı
1	Adana	Yüreğir	105.000	3
2	Ankara	Keçiören	180.000	5
3	Antalya	Aksu	60.000	2
4	İstanbul	Esenyurt	550.000	10
5	İzmir	Alsancak	180.000	3
6	Trabzon	Arsin	60.000	1

Aşağıda basın sektöründe karşılaşılan bazı sorunlar sıralanmıştır.

- Sektördeki yoğun rekabet nedeniyle ve tüketici açısından ürünlerin birbirlerinin yerine geçebilir nitelikte olmasından dolayı dağıtımdaki aksaklıklar kayıp satışlara yol açabilmektedir.
- Basın sektöründe üretilen gazeteler çok kısa ömürlüdür. Yoğun ve titiz çalışmaların ardında yetersiz bir dağıtım kanalı olması tüm çalışmaların değerini geçersiz kılar.
- Gazetelerin kısa ömürlü oluşlarından dolayı gazeteler depolanamaz.
- En son haberleri yayınlatabilmek için gazeteler mümkün olduğunca basım işini ertelerken, okuyucular ise gazetelerini erken saatlerde almak istiyorlar. Bundan dolayı üretilen gazetelerin dağıtımını için çok dar zaman aralığı kalmaktadır.

Bu çalışmada, yukarıda belirtilen sorunlar nedeniyle ve mevcut dağıtım maliyetinin yüksek olmasından dolayı, 6 basım merkezinden 188 başbayaie olan dağıtım ağı üzerinde çalışılmıştır. Şekil 3.1’de firmaya ait örnek dağıtım ağı gösterilmektedir. Ele alınan bu problem, “Çok Depolu Zaman ve Kapasite Kısıtlı Araç Rotalama Problemi (ÇDZKARP)”’dir. Fakat, bir başbayinin talebi sadece bağlı olduğu basım merkezi tarafından karşılandığı için ÇDZKARP’i, birbirinden bağımsız 6 tane Tek Depolu Zaman ve Kapasite Kısıtlı Araç Rotalama Problemi (TDZKARP) olarak ele alınabilir. Dolayısıyla problem, talepleri bilinen ve coğrafi olarak dağınık başbayilere hizmet vermek için merkezi bir basım merkezinden hareket ederek, depoya geri dönen homojen bir araç filosu tarafından katedilen toplam mesafeyi enküçükleyecek, kapasite ve zaman kısıtlarını aşmayan optimum araç rotalarının belirlenmesi olarak tanımlanmıştır.



Şekil 3.1: Ele alınan firmanın dağıtım ağı

Ele alınan çalışma aşağıda sıralanan kısıtları içermektedir.

- Zaman kısıtı: Tüm başbayilere gazeteler en geç saat 05:30’da ulaşmalıdır.
- Kapasite kısıtı: Bir rotadaki başbayilerin toplam talep miktarı araç kapasitesini geçemez.
- Bölge kısıtı: Her bölgenin gazetelerinin ekleri ve içerikleri farklı olduğu için her basım merkezi kendi bölgesine hizmet verebilir.
- Araçlar trafik kurallarında belirtilen hız limitlerini geçemezler.
- Her rota basım merkezinden başlar, basım merkezinde biter.
- Her başbayaie, bir araç tarafından yalnızca bir kere ziyaret edilir.

GDP'nin deęişmeyen üç tane elemanı (araçlar, basım merkezleri, başbayiler) vardır. Bu elemanların şu temel özelliklere sahip olduğu varsayılmaktadır.

Araçlar;

- Hepsi aynı özelliklere (kapasite ve hız) sahiptir.
- Sınırlı kapasiteleri (50.000 gazete) vardır.
- Hepsi basım merkezlerindedir.
- Arızalanmazlar.

Basım merkezleri;

- Her basım merkezinin, kendi bölgesindeki başbayi taleplerini karşılayacak miktarda gazete basabilme kapasitesi vardır.
- Saat 23:45'e kadar tüm başbayi talepleri basılmaktadır.
- Bir aracın doldurma süresi ortalama 15 dakikadır.

Başbayiler;

- Basım merkezlerinin bulunduğu düğüm dışındaki her düğüm bir başbayiyi ifade etmektedir.
- Her başbayinin önceden bilinen sabit miktarlarda gazete talepleri vardır.
- Her başbayinin talebi bulunduğu bölgedeki basım merkezinde basılmaktadır
- Servis süresi başbayilerin talep miktarına bağlıdır.
- Başbayiler dükkanlarını zamanında açarlar.

## 3.2. Veri Derlenmesi

### 3.2.1. Araç sayısı

GDP çözümünde toplam gidilen yolun enküçüklemesi istenirken, aynı zamanda dağıtım yaparken kullanılan araç sayısının enküçüklenmesi istenir. Zaman kısıtı dikkate alınmadan dağıtım yapılmak istenirse, kaç araç kullanılması gerektiği (3.1) eşitliği ile hesap edilir.

$$\text{Araç sayısının alt sınırı} = \frac{\text{Bir bölgedeki tüm baş bayi talepleri toplamı}}{\text{Araç kapasitesi}} \quad (3.1)$$

Üst sınır olarak kurucu sezgisel sonucu kullanılmıştır. Kurucu sezgisel Bölüm 3.4'de ayrıntılı olarak anlatılmıştır. Çizelge 3.2'de kullanılacak araç sayısı için alt ve üst sınır değerleri verilmiştir.

Çizelge 3.2: Bölgelere göre araç sayısının alt ve üst sınırları

Bölgeler	Toplam talep	Araç kapasitesi	Alt Sınır	Üst Sınır
Adana	450.900	50.000	10	12
Ankara	395.419	50.000	8	12
Antalya	193.515	50.000	4	6
İstanbul	619.220	50.000	13	14
İzmir	256.957	50.000	6	7
Trabzon	78.566	50.000	2	3

### 3.2.2. Başbayilerin yeri ve talep miktarı

Firmaya ait 188 başbayinin Türkiye'deki konumları, hangi üretim merkezlerine bağlı oldukları ve talep miktarları Çizelge 3.3'de verilmiştir. Gazete talep miktarları günlük ortalama net satışlardır. Bu bilgiler ele alınan dağıtım firmasından alınmıştır. Ayrıca [www.dorduncukuvvetmedya.com](http://www.dorduncukuvvetmedya.com) web adresinde firmanın dağıttığı gazete tiraj raporları yayınlanmaktadır.

Çizelge 3.3 : Firmanın başbayileri ve talep miktarları

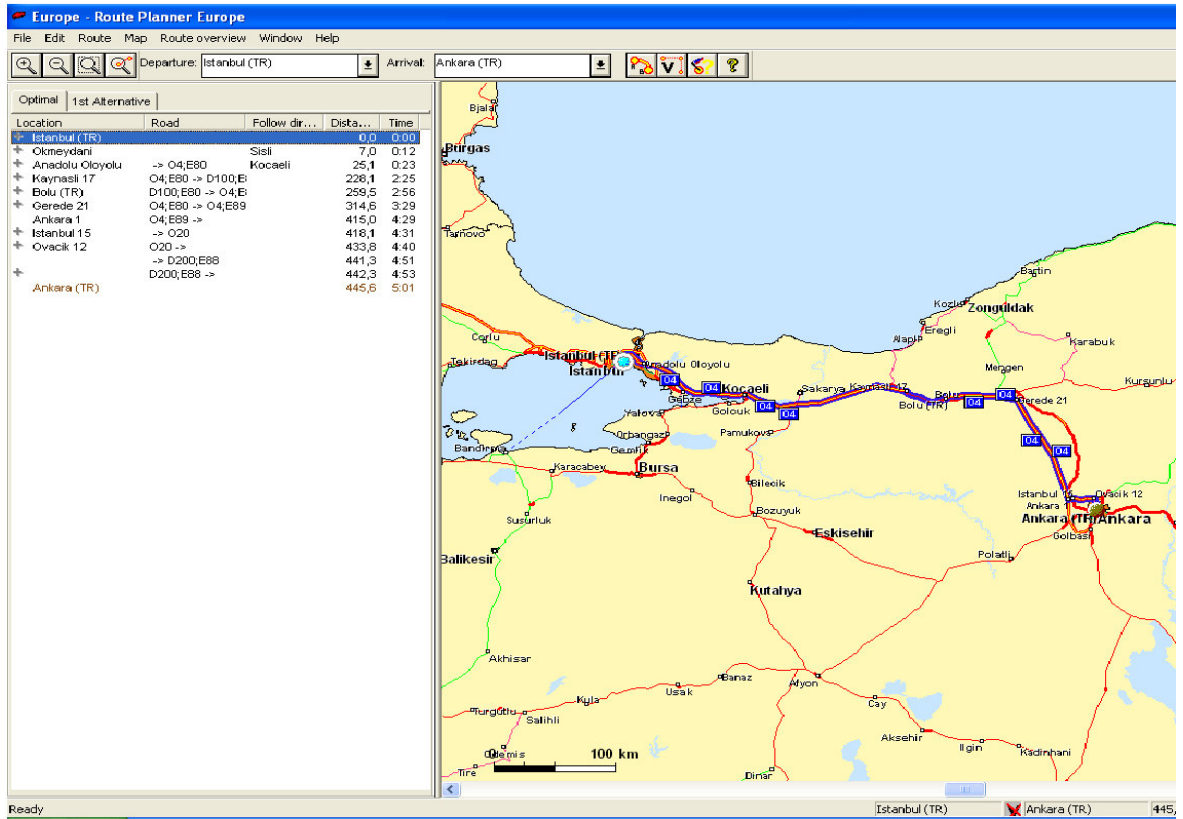
İSTANBUL BAYİLERİ			İZMİR BAYİLERİ			ANKARA BAYİLERİ		
NO	BAŞBAYİLER	TALEP	NO	BAŞBAYİLER	TALEP	NO	BAŞBAYİLER	TALEP
1	<b>ESENYURT</b>	0	1	<b>İZMİR</b>	0	1	<b>ESENBÖĞA</b>	0
2	BANDIRMA	6483	2	AYDIN	11233	2	AFYON	10385
3	BOZUYUK	3203	3	DİDİM	1862	3	MERZİFON	3659
4	BURSA 1	33364	4	İNCİRLİOVA	2184	4	ASTI	5286
5	BURSA 2	33364	5	KUSADASI	3134	5	BAHCELİEVLER 06	5286
6	GEMLİK	4986	6	NAZILLI	7916	6	BALGAT	5286
7	M. KEMAL PAŞA	5679	7	AKÇAY	5382	7	CANKAYA	5286
8	ORHANGAZI	4427	8	AYVALIK	2981	8	ERYAMAN	12727
9	CANAĞKALE	5525	9	BALIKESİR	15554	9	ETLİK	35366
10	DÜZCE	9711	10	EZİNE	1988	10	GAZİOSMANPAŞA	5286
11	EDİRNE	7729	11	BERGAMA	5407	11	HAVAALANI	2814
12	KEŞAN	4295	12	BORNOVA	20755	12	İSTASYON	21864
13	AKSARAY 34	12219	13	BUCA	16376	13	KIZILAY	5286
14	AVÇILAR	12328	14	CEŞME	1881	14	KOLEJ	5286
15	BAHCELİEVLER 34	12713	15	DIKILI	1570	15	POLATLI	6037
16	BAKIRKOY	12330	16	GOZTEPE 35	21275	16	SEREFLİKÖÇİSAR	3193
17	BESİKTAS	3732	17	KARSIYAKA	24556	17	ULUS	5286
18	BEYKOZ	10843	18	KONAK	21275	18	UMİTKOY	14360
19	BOSTANCI	6663	19	OTOĞAR	2749	19	YENİMAHALLE	14360
20	CERAHPAŞA	12219	20	ODEMiŞ	7129	20	BARTIN	7547
21	ETİLER	3732	21	TORBALI	4585	21	BOLU	7454
22	FENERBAHÇE	6663	22	URLA	2478	22	GEREDE	2536
23	GOZTEPE 34	6663	23	YENİ BOSTANLI	5712	23	CANKIRI	14295
24	HAZNEGAR	15633	24	KULA	2807	24	CORUM	11915
25	İÇERENKOY	6663	25	MANİSA	14929	25	SUNGURLU	4304
26	KARTAL	10362	26	SALİHLİ	8224	26	ESKİŞEHİR	38001
27	KIZILTOPRAK	6663	27	TURGUTLU	6600	27	KARABÜK	13081
28	KUCUKYALI	19003	28	BODRUM	4370	28	KASTAMONU	20914
29	MECİDİYEKOY	7221	29	MARMARİS	3388	29	KIRIKKALE	20560
30	SARIYER	12878	30	MİLAS	6159	30	KİRSEHİR	13888
31	SEFAKOY	13124	31	MUĞLA	4572	31	AKŞEHİR	5509
32	SUADIYE	8871	32	UŞAK	17927	32	KONYA	40416
33	SİSLİ	7221				33	SORGUN	6795
34	TAKSİM	13029		<b>ADANA BAŞBAYİLERİ</b>		34	YERKOY	2551
35	TUZLA	5185	<b>NO</b>	<b>BAŞBAYİLER</b>	<b>TALEP</b>	35	CAYCUMA	5878
36	UMRANIYE	28700	1	<b>İNCİRLİK</b>	0	36	DEVREK	3922
37	USKÜDAR	8871	2	ADANA 1. BÖLGE	33811	37	ZONGULDAK	8804
38	ÜSTGÖZTEPE	6663	3	ADANA 2. BÖLGE	33811			
39	YAKACIK	10362	4	CEYHAN	9224		<b>ANTALYA BAŞBAYİLERİ</b>	
40	YENİOTOĞAR	13574	5	KOZAN	6846	<b>NO</b>	<b>BAŞBAYİLER</b>	<b>TALEP</b>
41	YENİBOSNA	12713	6	POZANTI	1286	1	<b>AKSU</b>	0
42	YESİLKÖY	13124	7	ADİYAMAN	39700	2	DINAR	6034
43	ZEYNEPKAMİL	8871	8	AKSARAY 68	20229	3	SANDIKLI	4278
44	4. LEVENT	3732	9	GAZİANTEP	45905	4	ALANYA	13564
45	500 EVLER	37138	10	KİLİS	6320	5	ANTALYA 1	17450
46	LULEBURGAZ	6405	11	NİZİP	6321	6	ANTALYA 2	17450
47	DERİNCE	19039	12	ANTAKYA	18020	7	ELMALI	2051
48	GEBZE	23170	13	DÖRTYOL	6964	8	KAS	2693
49	İZMİT	25495	14	İSKENDERUN	15885	9	KEMER	1936
50	ADAPAZARI	20479	15	ERDEMLİ	6816	10	KUMLUCA	3237
51	PAMUKOVA	1285	16	MERSİN	37588	11	MANAVGAT	10026
52	CORLU	8893	17	SİLİFKE	9681	12	SERİK	5007
53	TEKİRDAĞ	7754	18	TARSUS	17621	13	BUCAK	3133
54	YALOVA	9426	19	ELBİSTAN	7136	14	BURDUR	5036
55	KDZ. EREĞLİ	8804	20	KAHRAMANMARAS	25313	15	DENİZLİ	46938
			21	KARAMAN	12898	16	İSPARTA	26542
			22	KAYSERİ	6011	17	ANAMUR	5135
			23	EREĞLİ	6652	18	SEYDİŞEHİR	4946
<b>NO</b>	<b>BAŞBAYİLER</b>	<b>TALEP</b>	24	NEVŞEHİR	16553	19	FETHİYE	8116
1	<b>ARSIN</b>	0	25	BOR	3698	20	GÖCEK	8116
2	ARTVİN	10585	26	NİĞDE	9019	21	ORTACA	1827
3	BAYBURT	4365	27	KADIRLI	5616			
4	ERZURUM	218	28	OSMANIYE	11188			
5	GİRESUN	6003	29	SANLIURFA	30788			
6	GÜMÜŞHANE	8855						
7	ORDU	8966						
8	ÜNYE	6899						
9	FINDIKLI	875						
10	RİZE	6794						
11	AKÇAABAŞ	6574						
12	BESİKDÜZÜ	2308						
13	SÜRMENE	2101						
14	TRABZON	14023						



### 3.2.3. Uzaklık ve zaman değerleri

İki başbayı arasındaki uzaklık ve yolculuk zamanı, coğrafi bilgi sistemi olan Sony Route Planner yazılımı kullanılarak hesaplanmıştır. Bu hesaplamalarda, herhangi iki başbayı arasındaki uzaklık, mevcut trafik kuralları ve araçların kullanabilecekleri yollar göz önünde bulundurularak, bu başbayiler arasındaki en kısa mesafe olarak belirlenmiştir. Oluşturulan 29x29'lik (Adana), 37x37'lik (Ankara), 21x21'lik (Antalya), 55x55'lik (İstanbul), 32x32'lik (İzmir), 14x14'lik (Trabzon) uzaklık ve zaman matrisleri simetrik ve üçgen eşitliğini sağlayan matrislerdir.

Servis süreleri başbayilerin talep miktarına bağlıdır. Bir çalışan, 100'lük paketler halindeki 10.000 adet gazeteyi, araçtan ortalama 8 dakikada boşaltmaktadır. Başbayı taleplerine göre servis süreleri hesap edilip, zaman matrisine eklenmiştir. Şekil 3.2'de Sony Route Planner yazılımı gösterilmektedir.



Şekil 3.2: Sony Router Planner yazılımı

### 3.3. Matematiksel Model

#### 3.3.1. Karar modelinin geliştirilmesi

GDP,  $V = \{1, 2, \dots, n\}$  düğüm seti ve  $A = \{(i, j) : i \neq j \in V\}$  ayrıt setini gösteren bir  $G = (V, A)$  serimi üzerinde, merkezi bir basım merkezinde bulunan aynı tip ve kapasiteye sahip olan homojen araçlar için, basım merkezini içeren bir coğrafi alanda değişik noktalara dağılmış tüm başbayilere en az maliyetle gazete dağıtım yapılacak rotaların bulunması problemi olarak modellenmiştir. Bu problemde, her başbaya sadece bir araçtan hizmet almakta ve her araç sadece bir rota izlemektedir. Araçlar için kapasite ve zaman kısıtının yanı sıra, araçların basım merkezinden hareket edip rotanın sonunda basım merkezine geri dönmesi zorunluluğu vardır. Ele alınan probleme, Kara'nın [9] polinom boyutta düğüm tabanlı Mesafe ve Kapasite Kısıtlı Araç Rotalama Problemi (MKARP) modeli uyarlanarak, aşağıdaki karar modeli önerilmiştir. Karar modelinde kullanılan parametre ve karar değişkenleri aşağıda verilmiştir:

#### **Parametreler:**

$Q$  : Her bir aracın kapasitesi (gazete)

$TC$ : Bir aracın gidebileceği en fazla süre (dakika)

$q_i$  :  $i$ . başbayinin günlük ortalama talebi (gazete)

$c_{ij}$  : başbaya  $i$  ile başbaya  $j$  arasındaki uzaklık (km)

$t_{ij}$  : başbaya  $i$  ile başbaya  $j$  arasındaki seyahat süresi (dakika)

#### **Karar Değişkenleri:**

$x_{ij}$ : 1, araç  $i$ . başbayiden  $j$ . başbaya giderse; 0, diğer durumda ( $i \neq j, i, j \in V$ )

$u_i$  :  $i$ . başbayiden çıkışta araçta bulunan gazete miktarı

$v_i$  :  $i$ . başbaya kadar gelen aracın, basım merkezinden itibaren geçirdiği süre

Verilen tanımlamalara göre gazete dağıtım problemi için geliştirilen tamsayılı doğrusal karar modeli aşağıdaki gibidir.

$$\text{Enk } w = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (1)$$

Kısıtları altında

$$\sum_{i=1}^n x_{ij} = 1 \quad , \quad j = 2, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad , \quad i = 2, \dots, n \quad (3)$$

$$\sum_{i=2}^n x_{1i} = m \quad (4)$$

$$\sum_{i=2}^n x_{i1} = m \quad (5)$$

$$u_i + (Q - q_i)x_{1i} \leq Q \quad , \quad i = 2, \dots, n \quad (6)$$

$$u_i - \sum_{\substack{j=2 \\ j \neq i}}^n q_j x_{ji} \geq q_i \quad , \quad i = 2, \dots, n \quad (7)$$

$$u_i - u_j + Qx_{ij} + (Q - q_i - q_j)x_{ji} \leq Q - q_j \quad , \quad i \neq j, \quad i, j = 2, \dots, n \quad (8)$$

$$v_i \geq t_{1i} + \sum_{\substack{j=1 \\ j \neq i}}^n (t_{1j} + t_{ji} - t_{1i})x_{ji} \quad , \quad i = 2, \dots, n \quad (9)$$

$$v_i \leq TC - (TC - t_{1i})x_{1i} \quad , \quad i = 2, \dots, n \quad (10)$$

$$v_i - v_j + (TC - t_{1j} + t_{ij})x_{ij} + (TC - t_{1j} - t_{ji})x_{ji} \leq TC - t_{1j} \quad , \quad i \neq j \geq 2 \quad (11)$$

$$x_{ij} \in \{0, 1\} \quad , \quad \forall (i, j) \in A \quad (12)$$

$$t_{1i} + t_{ij} > TC \text{ veya } q_i + q_j > Q \text{ ise, } i \neq j, \quad x_{ij} = 0 \text{ ve } t_{ii} = 0$$

Karar modelinin amaç fonksiyonu,  $w$ ; tüm araçlar tarafında katedilecek toplam mesafeyi göstermektedir.

(2) ve (3) numaralı kısıtlar atama kısıtlarıdır ve her ara düğümün sadece bir defa ziyaret edilmesini sağlamaktadır. (4) ve (5) numaralı kısıtlar, basım merkezinden  $m$  aracın çıkmasını ve aynı sayıda aracın basım merkezine girmesini sağlamaktadır.

(6), (7) ve (8) numaralı kısıtlar, kapasiteye ilişkin kısıtlardır. Burada (8) numaralı kısıt, Kara vd. [10] tarafından Kapasiteli Araç Rotalama Problemi için önerilen kısıtlardır ve (6) ve (7) numaralı kısıtlarla birlikte herhangi bir rotada bir aracın en fazla kapasitesi kadar gazete almasını sağlar. Bu kısıtlar aynı zamanda, problem için uygun olmayan turların (alt turların) oluşumunu da engelleme görevi görürler.

Benzer şekilde, (9), (10) ve (11) numaralı kısıtlar, bir araç rotasının süresinin en fazla  $TC$  olmasını öngörür.

(12) numaralı kısıt, karar değişkenlerinin 0-1 tamsayı değişken olduğunu göstermektedir.

Önerilen modelde,  $n^2 - n$  kadar 0-1 tamsayı değişken olup, kısıt sayısı  $2n^2$  kadardır. Yani, önerilen model, tamsayı değişken ve kısıt sayısı yönüyle  $O(n^2)$  olup, herhangi bir kodla doğrudan kullanılabilir özelliktedir.

### **3.3.2. Modelin bölge temelinde çözümü**

Yukarıda önerilen karar modeli, birbirinden bağımsız olan 6 bölge için toplanan verilerle ayrı ayrı oluşturulmuş; eniyi çözümü ise Intel Pentium 4 3.0 GHz, 3.04 Ghz iki işlemcili ve 2 GB Ram bulunan bir bilgisayar sisteminde 54.000 saniye (15 saat) süre sınırı verilip, CPLEX 10.0 paket programı kullanılarak aranmıştır. Her bölge için çözüm sonuçları ve süreleri Çizelge 3.4'de verilmiştir. Adana, Ankara, Antalya, İstanbul, İzmir, Trabzon bölgeleri için bulunan araç turları, araçların üzerindeki yük miktarı ve yolculuk zamanları sırasıyla Ek 1, Ek 2, Ek 3, Ek 4, Ek 5,

Ek 6'da verilmiştir. Eklerde ayrıtlar üzerindeki rakamlar, başbayiden çıkışta araçta bulunan gazete miktarını ( $u_i$ ), z'nin değerleri ise basım merkezinden itibaren geçirdiği süreyi ( $v_i$ ) göstermektedir.

Çizelge 3.4: Geliştirilen karar modeli sonuçları

<b>Bölgeler</b>	<b>n</b>	<b>m</b>	<b>w (km)</b>	<b>süre (sn)</b>
Adana	29	12	5735	44
		11	5755	205
Ankara	37	12	5769	14
		11	5734	3523
Antalya	21	6	2698	0,20
İstanbul	55	14	4314*	54000
İzmir	32	7	2897	17
Trabzon	14	3	1459	0,02

Basılan gazeteler tüm araçlara aynı anda yüklenemediği için, araçların basım merkezinden ayrılma zamanları birbirinden farklıdır. GDP'de gazetelerin başbayilere en geç teslim zamanı tüm araçlar için aynı olduğu düşünülürse, dağıtım için ilk yola çıkan aracın, sonra yola çıkan araçlara göre daha fazla zamanı vardır. Bundan dolayı gazetelerin zamanında başbayilere teslim edilebilmesi için, araç rotalama ve çizelgeleme problemi eş zamanlı olarak ele alınmak zorundadır.

Ele alınan problemde, tüm gazetelerin üretimi saat 23:45'de bitmektedir ve gazeteler saat 05:30'a kadar başbayilere teslim edilmelidir. Araçlara gazete yükleme süresi ortalama 15 dakika olduğu için, her 15 dakikada bir araç belirlenen turu takip etmek üzere basım merkezinden ayrılmaktadır. Eğer aynı anda birden fazla araca gazeteler yüklenebiliyor ise, her 15 dakikada dolum liman sayısı kadar araç, kendi turlarını takip etmek üzere yola çıkmaktadır. İlk yola çıkan aracın dağıtım için 330 dakika süresi varken, sonra çıkacak araçlar için bu zaman aralığı daralmaktadır. Bu nedenle, araçların tur zamanları büyükten küçüğe doğru sıralanıp, gazetelerin yüklemesi tur zamanı en fazla olan araçtan, en az olan araca doğru sırasıyla yapılmalıdır. Tur zamanı en fazla olan araç, saat 00:00'de basım

\* 54000 sn zaman sınırı içerisinde bulunan en iyi değer

merkezinden ilk olarak ayrılan araçtır. Bu kural İstanbul bölgesi haricindeki diğer 5 bölgeye uygulanmıştır.

İstanbul bölgesinde bulunan bazı tur süreleri az çıkmıştır. Bu durum tur içerisindeki başbayi taleplerinin fazla olmasından ve araç kapasitesinin diğer başbayilere hizmet vermek için yeterli olmadığından kaynaklanmaktadır. GDP'nin çözümünde önemli olan her turda uğranılacak son başbayi taleplerinin son teslim saatine kadar karşılanması olduğu için, araçların depoya geri dönüş süreleri dikkate alınmamıştır. Fakat depoya erken dönen araçların yeni bir turu tamamlayacak kadar süresi olursa, araç sayısını azaltmak için aynı aracı arka arkaya farklı turlara gönderilebilir.

Araçların tekrar kullanma yaklaşımı sonucunda araçlar tarafından katedilen yol azalmamakta, ancak kullanılan araç sayısı azalmaktadır. Çoğu dağıtım şirketi araç filosunu genelde kiraladığı için, kullanılan araç sayısı dağıtım maliyetlerinde önemli bir kalemdir. Bu yüzden, İstanbul bölgesinin dağıtımında kullanılan araç sayısını azaltabilmek için, tur süresi fazla olan araçların teslimatı gecikmemesi dikkate alınarak, tur süresi az olup basım merkezine hemen dönebilecek olan araçların yüklenmesi ilk yapılmıştır.

Basım merkezine geri dönen araçlara yeniden gazete yüklemesi yapılacağından, 15 dakika sonra yola çıkabilirler. Depoya ilk geri dönen araca ilk gazete yüklemesi yapıldıktan sonra, saat 05:30'u geçmeyecek en uzun dağıtım süresi olan turdaki başbayilerin taleplerini karşılamak üzere araç tekrar yola çıkmaktadır. Bu çalışma diğer bölgelerdeki tur sürelerinin fazla olmasından dolayı yapılmamıştır. Hangi aracın ne zaman yola çıktığı bölgelere göre sırasıyla Çizelge 3.5 – Çizelge 3.10'de verilmektedir. Yapılan araç çizelgeleme sonucunda, 6 bölgede dağıtım yapan tüm araçların başbayi taleplerini saat 05:30'a kadar karşıladığı ve gazetelerin tesliminde hiç bir gecikme olmadığı görülmektedir. Ayrıca İstanbul bölgesinde kullanılan araç sayısı 14'ten 10'a düşürülmüştür.

Çizelge 3.5: Adana bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Teslimat için en geç zaman
00:00	9	322	05:22	05:30
00:00	11	321	05:21	05:30
00:00	4	319	05:19	05:30
00:15	2	308	05:23	05:30
00:15	10	300	05:15	05:30
00:15	8	296	05:11	05:30
00:30	7	274	05:04	05:30
00:30	3	260	04:50	05:30
00:30	1	250	04:40	05:30
00:45	6	225	04:30	05:30
00:45	5	194	04:59	05:30

Çizelge 3.6: Ankara bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Teslimat için en geç zaman
00:00	11	329	05:29	05:30
00:00	7	322	05:22	05:30
00:00	6	319	05:19	05:30
00:00	1	315	05:15	05:30
00:00	3	313	05:13	05:30
00:15	10	305	05:20	05:30
00:15	4	304	05:19	05:30
00:15	5	301	05:16	05:30
00:15	8	301	05:16	05:30
00:15	9	247	04:22	05:30
00:30	2	155	03:05	05:30

Çizelge 3.7: Antalya bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Teslimat için en geç zaman
00:00	2	319	05:19	05:30
00:00	5	314	05:14	05:30
00:15	1	301	05:16	05:30
00:15	6	251	04:26	05:30
00:30	4	237	04:27	05:30
00:30	3	206	03:56	05:30

Çizelge 3.8: İstanbul bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Araç no	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Dönüş süresi (dk)	Basım merkezine dönüş saati
00:00	3	3	84	01:24	19	01:43
00:00	6	6	85	01:25	25	01:50
00:00	5	5	90	01:30	47	02:17
00:00	10	10	113	01:53	23	02:16
00:00	2	2	326	05:26	-	-
00:00	4	4	323	05:23	-	-
00:00	12	12	304	05:04	-	-
00:00	13	13	270	04:30	-	-
00:00	14	14	254	04:14	-	-
00:00	1	1	243	04:03	-	-
01:58	3	8	178	04:56	-	-
02:05	6	11	148	04:33	-	-
02:31	10	7	145	04:56	-	-
02:32	5	9	131	04:43	-	-

Çizelge 3.9: İzmir bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Teslimat için en geç zaman
00:00	3	325	05:25	05:30
00:00	5	325	05:25	05:30
00:00	1	315	05:15	05:30
00:15	2	292	05:07	05:30
00:15	7	287	05:02	05:30
00:15	6	210	03:45	05:30
00:30	4	197	03:47	05:30

Çizelge 3.10: Trabzon bölgesine dağıtım yapan araçların zaman çizelgesi

Basım merkezinden ayrılma saati	Tur no	Tur süresi (dk)	Son başbayiden ayrılma saati	Teslimat için en geç zaman
00:00	2	301	05:01	05:30
00:15	1	275	04:50	05:30
00:30	3	229	04:19	05:30

Firmanın mevcut dağıtım ağı ile tamsayılı doğrusal karar modeli sonuçlarının karşılaştırılması Çizelge 3.11'de verilmiştir.



Çizelge 3.11: Mevcut sistem ile karar modeli sonuçlarının karşılaştırılması

Bölgeler	Mevcut dağıtım ağı		Karar modeli sonuçları		(1) – (2) arasındaki % fark
	Katedilen toplam mesafe (km) (1)	m	Katedilen toplam mesafe (km) (2)	m	
Adana	6603	13	5755	11	14,7
Ankara	6505	12	5734	11	13,4
Antalya	3160	7	2698	6	17,1
İstanbul	5909	16	4314*	10**	37
İzmir	3986	9	2897	7	37,6
Trabzon	1533	3	1459	3	5,1
<b>Toplam</b>	<b>27696</b>	<b>60</b>	<b>22957</b>	<b>48</b>	<b>20,6</b>

Bu çalışmada ele alınan problem için, tamsayı doğrusal karar modeli yardımıyla İstanbul bölgesi hariç diğer bölgelerde en iyi çözüm bulunarak, bütün sistemin dağıtım maliyetlerinde mevcut uygulamaya göre %20 azalma sağlanmıştır. Sonuçlar bölgesel olarak incelendiğinde, İstanbul ve İzmir bölgelerinde %37, Antalya bölgesinde %17, Adana bölgesinde % 15, Ankara bölgesinde %14 ve Trabzon bölgesinde %5 oranında maliyetlerde iyileştirme olduğu görülmektedir. Türkiye geneline yayılmış olan başbayilerinin taleplerini zamanında karşılayabilmek için mevcut uygulamada kullanılan 60 araç sayısı, önerilen karar modeli yardımıyla 48'ye düşürülmüştür. Adana, Ankara, Antalya, İstanbul, İzmir, Trabzon bölgeleri için bulunan araç turları ve mevcut uygulamadaki araç turları sırasıyla Ek 7, Ek 8, Ek 9, Ek 10, Ek 11, Ek 12'da verilmiştir.

### 3.3.3. Modelin uygulamada sınırları

Gazete Dağıtım Planlaması kısa dönemli planlama problemidir. Geliştirilen tamsayı doğrusal karar modeli ile 54.000 saniye süre sınırı içerisinde İstanbul bölgesinin optimal sonucu bulunamamıştır. Bundan dolayı, GDP'de çok bayili bölgelerin yaklaşık çözümünü makul sürelerde bulmak amacıyla bir melez algoritma geliştirilmiştir. Melez algoritma ilerki bölümde anlatılmaktadır.

\* 54000 sn zaman sınırı içerisinde bulunan en iyi değer

\*\* Tamsayı doğrusal karar modeli sonuçlarına çizelgeleme uygulanarak araç sayısı 10'a düşürülmüştür.

### 3.4. Gazete Dağıtım Planlaması İçin Melez Bir Sezgisel Algoritma

Önerilen melez algoritma, yerel aramaya dayalı algoritmalar arasında yer alan Tabu Arama, Tavlama Benzetimi ve Değişken Komşuluk Arama sezgisellerinin karmasıdır. Kısa dönemli tabu algoritması yerine kötü çözümleri belirli olasılıkla kabul eden TB kullanılmıştır. Ayrıca tabu arama algoritmasının rotalama problemi çözmedeki performansını arttırmak amacıyla, DKA ile zengin yöre yapısı önerilmiştir.

Kombinatoriyel optimizasyon alanının en tanınmış ve en övülen problemi, şüphesiz Gezgin Satıcı Problemi'dir (GSP). GSP, temel olarak bir şehirden başlayıp tekrar aynı şehire dönmek zorunda olan bir gezgin için, her şehri bir kere ziyaret etmesini sağlayacak en iyi turun bulunması olarak tanımlanabilir. Concorde [11], Simetrik Gezgin Satıcı Problemini (SGSP) çözmek için David Applegate, Robert E. Bixby, Vašek Chvátal ve William J. Cook tarafından ANSI C++ programlama diliyle yazılmış bir bilgisayar kodudur. TSPLIB'deki [12] 110 SGSP'den 107 tanesinin optimal sonuçlarını bulmuştur. Çözdüğü en büyük boyutlu problemde 15.112 şehir vardır. GDP'nin çözümü için literatürdeki bazı çalışmalarda genellikle tur geliştirici algoritma olarak 2-opt sezgiseli kullanılmıştır. Bu sezgiselde, her tur içinde iki farklı düğümün yerleri birbiriyle değiştirilerek daha iyi sonuç aranmaktadır. Bu çalışmada, tur geliştirici algoritma olarak kullanılan 2-opt sezgiseli yerine kısa sürede optimal sonuç veren Concorde bilgisayar kodu kullanılmıştır.

Başlangıç çözümü elde etmek için kullanılan kurucu sezgisel, Solomon'un [13] zaman kısıtlı araç rotalama ve çizelgeleme problemi için önerdiği sıralı ekleme sezgiselinin geliştirilmiş şeklidir. Şekil 3.3'de paralel turlu kurucu sezgiselinin algoritması verilmiştir. Solomon'un önerdiği sezgiselden farklı olarak ekleme yapılırken, tüm turlar göz önünde bulundurulmaktadır. Zaman kısıtını sağlamak amacıyla, tüm başbayilerin basım merkezine olan uzaklıklarına göre büyükten küçüğe doğru sıralanmaktadır. Algoritmanın başlangıcında basım merkezine en uzak olan başbayinin bir tura eklenmesinden başlanmaktadır. Daha sonra sırasıyla basım merkezine sonraki en uzak başbayinin, tüm turlar arasından en uygun yere eklenmesiyle devam etmektedir.

- Adım 1:**  $\beta_1$  : uzaklık için ağırlık değeri ,  $\beta_2$  : zaman için ağırlık değeri
- Adım 2:** Tüm başbayilerin üretim merkezlerine uzaklıklarına göre büyükten küçüğe doğru sırala
- Adım 3:** Yerleştirilmemiş başbayiler arasından üretim merkezine en uzak olan başbayiyi seç:  $j$
- Adım 4:**  $m$  tur için  $f_3(r)$  değerini hesapla ( $r = 1, 2, \dots, m$ )  
 $f_3(r) = \beta_1 f_1(i, j, k) + \beta_2 f_2(i, j, k)$  ,  $\beta_1 + \beta_2 = 1$  ,  $\beta_1 \geq 0$  ,  $\beta_2 \geq 0$
- Adım 5:** Eğer herhangi bir tur için uygun olan çözüm yok ise,  
 $m = m + 1$  ve  $j$  düğümünü yeni tura ata.  
 Değilse (en az bir tur için uygun çözüm varsa)  
 $r = \arg \min \{ f_3(u); u=1, 2, \dots, m \}$   
 $j$  başbayisini,  $f_3(u)$  değeri en küçük olan turdaki  $i, k$  başbayiler arasına yerleştir.
- Adım 6:** Eğer atanmamış başbayi varsa Adım 3'e git, Değilse DUR

Şekil 3.3: Paralel turlu kurucu sezgisel algoritması

Her adımda seçilen başbayi uygun olan tura eklenmektedir. Seçilen başbayi ( $j$ ) belirli bir turda ( $r$ ) en uygun yere eklemek ( $i$  ile  $k$  arası) için (3.2) ve (3.3) eşitlikleriyle  $f_1(i, j, k)$  ve  $f_2(i, j, k)$  değerleri hesap edilmektedir.

$$f_1(i, j, k) = D_{ij} + D_{jk} - D_{ik} \quad (3.2)$$

$D_{ij}$  :  $i, j$  başbayiler arasındaki uzaklık (km)

$$f_2(i, j, k) = b_{jk} - b_k \quad (3.3)$$

$b_k$  :  $k$ . başbayiye varış zamanı

$b_{jk}$  :  $j$ . başbayinin tura eklenmesiyle  $k$ . başbayiye varış zamanı

$$f_3(r) = \beta_1 f_1(i, j, k) + \beta_2 f_2(i, j, k) \quad , \quad \beta_1 + \beta_2 = 1 \quad , \quad \beta_1 \geq 0 \quad , \quad \beta_2 \geq 0 \quad (3.4)$$

$\beta_1$  ,  $\beta_2$  sırasıyla uzaklık ve zaman için ağırlık değerleridir.

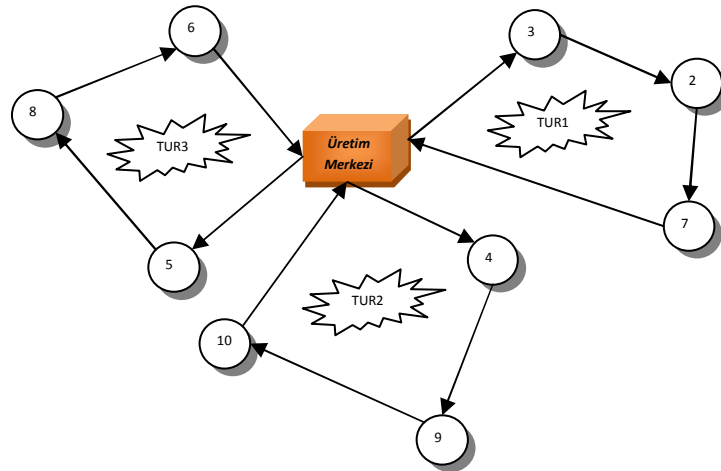
(3.4) eşitliğindeki  $f_3(r)$  değeri seçilen başbayinin ( $j$ ),  $r$  turu üzerindeki en uygun yerini tespit etmede kullanılmaktadır. Dolayısıyla, seçilen başbayi (3.5) eşitliğindeki  $f_3(u)$  değeri en küçük olan tura eklenmektedir.

$$r = \arg \min \{ f_3(u); u=1, 2, \dots, m \} \quad (3.5)$$

Başta belirlenen  $m$  araç sayısı yerleri belirlenmemiş bazı başbayiler için uygun olmayan çözümler oluşturabilmektedir. Eğer seçilen başbayi eklenirken, tüm turlar için uygun olmayan çözümler varsa, yeni bir tur açılmaktadır. Algoritmanın sonucunda  $s$  tane yeni tur açılmışsa paralel turlu kurucu sezgisel sonucunda  $m+s$  tane tur oluşturulmuştur.

### 3.4.1. Komşu üretme stratejisi

GDP için bir çözüm noktası,  $V_i$  ( $i = 1, \dots, n$ ) düğümlerinin bir sıralaması olarak gösterilmiştir. Dizilerden her biri problemin olası çözümler uzayındaki rassal bir noktayı simgelemektedir. Bu sıralamada basım merkezi düğümü ( $i = 1$ ),  $K$  araç (veya rota) sayısı olmak üzere,  $K+1$  sayıda kullanılmaktadır. Şekil 3.4'de GDP için çözüm örneği verilmiştir.



tur 1: 

1	3	2	7	1
---	---	---	---	---

tur 2: 

1	4	9	10	1
---	---	---	----	---

tur 3: 

1	5	8	6	1
---	---	---	---	---

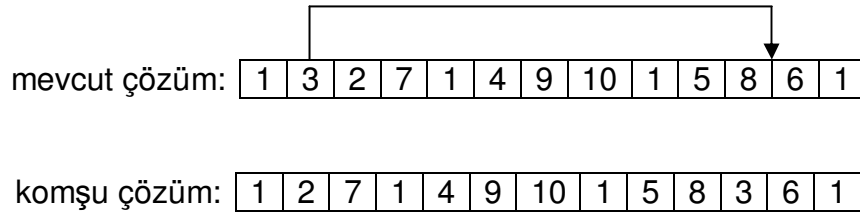
çözüm: 

1	3	2	7	1	4	9	10	1	5	8	6	1
---	---	---	---	---	---	---	----	---	---	---	---	---

Şekil 3.4: Gazete dağıtım problemi için çözüm örneği

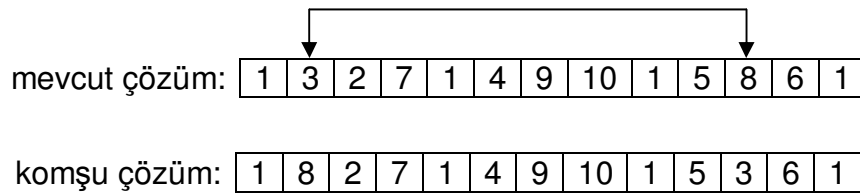
Komşu aramaya dayalı metotlarda,  $x$  mevcut çözümünden, onun bir komşusu olan  $x'$  çözümüne geçişin nasıl yapılacağı belirlenmesi gerekmektedir. Bu işlem hareket mekanizması olarak bilinir ve uygun hareket mekanizmasının seçimi, kullanılan metodun performansını hem çözüm kalitesi hem de hesaplama maliyeti açısından etkilemektedir (Taillard, [14]; Woodruff, [15]). Bu çalışmada,  $x$  mevcut çözümünün komşuları, bu çözümden tekli değişim ve ikili değişim hareketleriyle elde edilebilecek komşu çözümlerden oluşturulmuştur.

Tekli değişim (td): Çözüm noktasında bir turda bulunan bir eleman farklı turdaki iki elemanın arasına yerleştirilir. Şekil 3.5'de tekli değişim hareketine bir örnek verilmiştir.



Şekil 3.5: Tekli değişim hareketi ile elde edilen bir komşu çözüme örnek

İkili değişim (id): Çözüm noktasında bir turda bulunan bir eleman farklı turda bulunan diğer bir elemanla yerleri birbirleriyle değiştirilir. Şekil 3.6'de ikili değişim hareketine bir örnek verilmiştir.

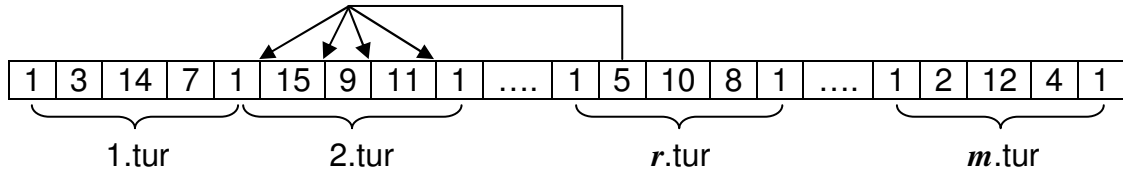


Şekil 3.6: İkili değişim hareketi ile elde edilen bir komşu çözüme örnek

Bu çalışmada geliştirilen melez algoritmada,  $N(x)$  komşu kümesinin örneklenmesi stratejisi kullanılmaktadır. Bu stratejiye göre, mevcut çözümden yapılabilen ve tabu olmayan tüm hareketler incelenir ve bunlar arasından en iyi olan seçilmektedir. Ancak, mevcut çözümün komşu sayısı çok fazla ve/veya komşuların

değerlendirilmesinin hesaplama maliyeti yüksek ise, incelenecek hareketlerin sayısının belli bir stratejiye göre sınırlandırılması kaçınılmaz olmaktadır.

Tekli değişim hareket mekanizmasıyla  $r$ . turda yer alan  $i$ . düğüm, bulunduğu tur haricindeki tüm turlara sırasıyla eklenmektedir. Şekil 3.7'de  $m$  turdan oluşan bir mevcut çözümden tekli değişim hareketi ile elde edilecek komşu çözümler gösterilmektedir.  $r$ . turdaki 5 nolu düğümün yeri tekli değişim hareket mekanizmasıyla değiştirilecektir. 5 nolu düğüm 2. turda 1-15, 15-9, 9-11, 11-1 düğümleri arasına yerleştirilebilir. 1, 15, 9, 11, 5 düğümlerinden oluşan Gezgin Satıcı Problemi (GSP) Concorde bilgisayar kodu ile çözülür ve optimal sıralama 2. tur olarak belirlenmektedir. Bu çözüm tekli değişim hareket mekanizmasıyla bulunan komşulardan bir tanesidir. 5 nolu düğüm aynı yöntemle diğer turlara eklenerek diğer komşular bulunmaktadır.



Şekil 3.7: Tekli değişim hareketi ile elde edilen komşu çözümler

İkili değişim hareket mekanizmasıyla  $r$ . turda yer alan  $i$ . düğüm, diğer tüm düğümlerle ( $r$ . turdaki düğümler hariç) yer değiştirmektedir. Herhangi bir iterasyonda  $i$ . düğüm  $r$ . turda yer aldığı düşünülürse, komşu sayıları tekli değişim hareketi için (3.6) eşitliğiyle, ikili değişim hareketi için (3.7) eşitliğiyle hesap edilir.

Düğüm sayısı( $r$ ):  $r$ . turda bulunan düğüm (başbayi) sayısı

$n$  : serimde bulunan toplam düğüm (başbayi) sayısı

No\_idts( $i$ ): ikili değişim hareketinde  $i$ . düğüm ile tabu olarak sınırlandırılan toplam hareket sayısı

No\_tdks( $i$ ):  $i$ . düğüm için tekli değişim hareket mekanizması kullanılarak elde edilebilen komşuların toplam sayısı

No\_idks( $i$ ):  $i$ . düğüm için ikili değişim hareket mekanizması kullanılarak elde edilebilen komşuların toplam sayısı

$$\text{No\_tdks}(i): \text{Araç sayısı} - 1 \quad (3.6)$$

$$\text{No\_idks}(i): n - \text{Düğüm sayısı}(r) - \text{No\_idts}(i) \quad (3.7)$$

Örneğin 50 düğümlü bir serimde, kullanılan araç sayısı 5 ise,  $\text{No\_tdks}(i)=4$ ,  $\text{No\_idks}(i) < 50$  komşusu vardır. 100 düğümlü bir serimde, kullanılan araç sayısı 15 ise,  $\text{No\_tdks}(i)=14$ ,  $\text{No\_idks}(i) < 100$  komşusu vardır. Komşu sayıları, problem boyutunun büyümesiyle üstel olarak artmamaktadır. Dolayısıyla herhangi bir aday stratejisi kullanmaya gerek duyulmayıp, mevcut çözümün tüm komşuları incelenerek, en iyisinin seçilmesine karar verilmiştir.

Geliştirilen TB algoritmasında, DKA stratejisi kullanılarak elde edilen komşu çözümlerin iyileştirilmesi amaçlanmıştır. Bu stratejiye göre öncelikle mevcut çözümden bir düğüm( $i$ ) seçilir. Mevcut çözüme başlangıçta hangi hareket tipinin (tekli değişim veya ikili değişim) uygulanacağı eşit olasılıkla belirlenmektedir. Hareket tipi seçildikten sonra, mevcut çözümden o hareket tipiyle elde edilebilen  $\text{No\_tdks}(i)$  veya  $\text{No\_idks}(i)$  tane komşu çözüm üretilmektedir. Daha sonra,  $\text{No\_tdks}(i)$  veya  $\text{No\_idks}(i)$  komşu çözüm içerisinde en düşük maliyetli olanı kabul fonksiyonuna göre yeni mevcut çözüm olarak kabul veya reddedilmektedir. Eğer aday komşu, mevcut çözüm olarak kabul edilirse aynı hareket mekanizmasıyla komşu üretmeye devam edilir, eğer kabul edilmezse diğer hareket mekanizmasına geçilir.

GDP çözümünde kullanılan amaç fonksiyonu tüm araçlar tarafından katedilecek toplam mesafeyi ve kullanılan araç sayısını enküçükmektir. Tur boşaltma mekanizmasıyla kullanılan araç sayısını azaltmaya çalışılmıştır. Bu mekanizma tekli değişim hareket mekanizmasıyla birlikte çalışmaktadır. Tekli hareket mekanizması ilk çalıştığında tüm turlardaki düğüm sayılarını hesaplamaktadır. Eğer herhangi bir turdaki düğüm sayısı 1 ise, o turda bulunan tek düğümü diğer turlara eklemeye çalışmaktadır. Eğer tek olan düğümü herhangi bir tura ekleyip uygun olan çözüm bulunursa, kullanılan araç sayısında azalma olmaktadır. Eğer uygun olan çözüm bulunamazsa, rassal bir düğüm seçilir ve bu seçilen düğüm için tekli hareket mekanizması çalıştırılmaktadır. Şekil 3.8'de tur boşaltma mekanizmasının adımları verilmiştir.

**Adım 1:** Tüm turlardaki düğüm sayılarını bul. ( $r = 1, \dots, m$ )

**Adım 2:** Eğer herhangi bir turdaki düğüm sayısı 1 ise

$x$  ( tek kalmış düğüm) düğümü için tekli hareket mekanizması çalıştır.

Eğer  $x$  düğümü için uygun olan bir çözüm varsa

$$m = m - 1$$

Adım 1'e dön.

değilse

Adım 3'e git.

değilse

Adım 3'e git.

**Adım 3:** rassal olarak bir düğüm seç:  $x$  ( $x = 2, \dots, n$ )

**Adım 4:**  $x$  düğümü için tekli değişim hareket mekanizması çalıştır.

Eğer  $x$  düğümü için uygun olan çözüm varsa

Adım 1'e dön.

Değilse

Hareket mekanizması değiştir.

Şekil 3.8: Tur boşaltma mekanizmasının adımları

Geçmişte yapılan çalışmalar incelendiğinde, uygun olmayan çözümlerinde kabul edildiği görülmüştür. Bu çalışmada, uygun olmayan çözümleri kabul etmek yerine uygun hale getirilmeye çalışılmıştır. GDP çözümünde daha önceden anlatılan iki farklı hareket mekanizması (tekli değişim ve ikili değişim) kullanılmıştır. Çeşitlendirmeyi sağlamak amacıyla DKA modern sezgiseli ile hareket mekanizmaları değiştirilerek kullanılmıştır. Kullanılan hareket mekanizmasıyla tüm sonuçlar uygun olmayan çözüm çıkarsa, diğer hareket mekanizmasıyla aramaya devam edilmektedir. Tekli değişim ve ikili değişim hareket mekanizmasıyla tüm sonuçlar uygun olmayan çözüm çıkarsa, tamir fonksiyonu devreye girmektedir. Tamir fonksiyonunun amacı eldeki uygun olmayan çözümleri uygun çözüm haline getirmektir. Tamir fonksiyonunu nasıl çalıştığını anlatmak için aşağıda bir örnek verilmiştir.



Mevcut çözümdeki 1. turda yer alan 3 nolu düğümün yeri önce tekli değişim sonra ikili değişim hareket mekanizmasıyla değiştirilmesi denenmiştir ve tüm komşular uygun olmayan çözüm çıktığı varsayılmaktadır. Tamir fonksiyonu, tekli değişim hareket mekanizmasıyla elde edilen uygun olmayan komşular arasından en küçük amaç fonksiyonu değerine sahip olan komşu çözüm üzerinden çalışmaktadır. Şekil 6.3'de gösterildiği gibi en küçük amaç fonksiyonu değerine sahip komşu çözüm, 3 nolu düğümün 3. turdaki 8 ve 6 nolu düğümler arasına yerleştirilmesiyle elde edilmektedir. Mevcut çözümün uygun çözüm olduğu bilinmektedir. 3 nolu düğümün 3. tura eklenmesiyle birlikte çözüm uygun olmayan hale gelmektedir. Çözümü uygun olmayan hale getiren 3. turdur. Uygun çözüm elde etmek için 3. turdaki 3 nolu düğüm haricindeki diğer tüm düğümlerin başka turlara eklenmesi gerekmektedir. Bunun için 3. turda yer alan 5, 8 ve 6 nolu düğümler için tekli değişim hareket mekanizmasıyla komşu çözümler araştırılmalıdır. Bulunan uygun çözümler arasından amaç fonksiyonu değerini en küçük yapan komşu çözüm tavlama benzetim algoritmasının metropolis fonksiyonuyla mevcut çözüm olarak kabul edilmektedir.

### 3.4.2. Yeniliğe dayanan hafıza fonksiyonları

Yapılan bir hareketin tabu olarak tanımlanması ve yapılacak bir hareketin tabu olup olmadığının kontrol edilmesi için iki ayrı tabu listesi kullanılmıştır. Tekli değişim ve ikili değişim hareketleri için tutulan bu listeler sırasıyla  $(n-1)$  ve  $(n-1) \times (n-1)$  boyutlu matrisler ile gösterilmektedir.

$tdtb[i]$ :  $i$  düğümünün mevcut çözümdeki yerini değiştiren bir hareket için tabu süresinin başlangıcı,

$idtb[i, j]$ :  $i, j$  düğümlerinin mevcut çözümdeki yerlerini birbirleriyle değiştiren bir hareket için tabu süresinin başlangıcı.

Tekli değişim hareketiyle  $i$ . düğüm taşınmışsa, tabu süresi boyunca  $i$ . düğüm yerinde kalır. İkili değişim hareketiyle  $i$ . ve  $j$ . düğümlerin yerleri birbirleriyle değiştirilmişse, tabu süresi boyunca  $i$ . ve  $j$ . düğümleri birbirleriyle yer değiştiremez. Yapılan bir hareketin kaç iterasyon boyunca tabu olacağı tabu süresi ile

belirlenmektedir. Hareketin tekli deęişim ve ikili deęişim nitelięi iermesine gre farklı iki sabit tabu sresi kullanılmıřtır.

tdts: Mevcut zmdeki tek dęmn yerini deęişiren bir hareket iin tabu sresinin uzunluęu,

idts: Mevcut zmdeki iki dęmn yerini birbirleriyle deęiřtiren bir hareket iin tabu sresinin uzunluęu.

Yukarıda aıklanan tabu listelerinden de anlařılacaęı zere, seilen bir hareket tabu listesine kaydedilerek tabu sresi boyunca yasaklanmaktadır. Buna gre, arama boyunca gerekleřtirilen iterasyon sayısı "iter" ile gsterilecek olursa, herhangi bir iterasyonda mevcut zme tekli deęişim hareketi ile  $i$  dęmnn yeri deęiřtirildięinde,  $tdtb[i]$  matrisi (3.8) eřitlięi ile gncelleřtirilmektedir. Eęer mevcut zme ikili deęişim hareketi ile  $i$  ve  $j$  dęmlerinin yerleri birbirleriyle deęiřtirildięinde,  $idtb[i, j]$  matrisi (3.9) eřitlięi ile gncelleřtirilmektedir. Burada amalanan,  $i$  dęmnn yerini tdts sresi boyunca ve  $i, j$  dęmlerinin yerlerini birbirleriyle idts sresi boyunca bir daha deęiřtirilmemesidir.

$$tdtb[i]=iter \quad (3.8)$$

$$idtb[i, j]=iter \quad (3.9)$$

Aıklanan zelliklere gre, mevcut iterasyonda denenen hareketlerin tabu olup olmadıkları (3.10) ve (3.11) eřitsizlikleri ile belirlenmektedir.  $tdtb[i]$  hareketi iin (6.11) eřitsizlięi saęlanıyorsa veya  $idtb[i, j]$  hareketi iin (6.12) eřitsizlięi saęlanıyorsa bu hareket tabudur.

$$iter \leq tdtb[i] + tdts \quad (3.10)$$

$$iter \leq idtb[i, j] + idts \quad (3.11)$$

oęu TA uygulamalarında kısa dnemli hafıza fonksiyonu olarak sabit tabu sreli tek bir tabu listesini kullanılmaktadır. Seilen hareketlerin nitelikleri bu listeye kaydedilmekte ve belli sayıda iterasyon getikten sonra listeden silinmektedir. Ancak, Glover [18], tabu listesinin yapısının veya tabu sresinin uzunluęunun

arama boyunca değiştirilmesinin TA'nın performansında etkili olacağını belirtmiştir. Bu amaçla, son zamanlardaki TA uygulamalarında dinamik tabu süreleri kullanılmıştır (Laguna ve Glover, [16]; Chiang ve Kouvelis, [17]; Costamagna ve arkadaşları, [18]).

Bu çalışmada yapılan denemelerde kısa tabu süreleri kullanıldığı zaman, daha önce bulunan yerel optimumların yeniden bulunmasının önlenemediği görülmüştür. Sürekli aynı çözümler bulunmaktadır. Buna karşın uzun tabu süreleri kullanıldığında, hareketlerin çoğu tabu olduğu için iyi hareketlerin yapılması engellenmektedir veya uygun çözüm bulamamaktadır. Kısa tabu süreleri kuvvetlendirme rolünü üstlenirken, uzun tabu süreleri çeşitlendirme etkisi oluşturmuştur. Bu nedenle, aramanın hem kuvvetlendirilmesini hem de çeşitlendirilmesini sağlamak amacıyla dinamik tabu süreleri kullanılmıştır.

Tabu sürelerinin değiştirilmesi için çözüm haritalama yöntemi kullanılmıştır. Bu yöntemin temelinde tüm çözümleri kayıt altında tutmak vardır. Çözümleri kaydetmek için 3 değer belirlenmiştir: araç sayısı, toplam uzaklık ve toplam süre. Bu 3 değer aynı olduğu çözümler aynı çözümlerdir. Tüm bulunan çözümler ve tekrar sayıları bir dosyaya kaydedilmektedir. Dinamik tabu süreleri tekli değişim ve ikili değişim hareketleri için farklı olmak üzere, kısa (K), orta (O) ve uzun (U) tabu sürelerinden oluşmaktadır.

$K_{tdts}$ ,  $O_{tdts}$ ,  $U_{tdts}$ : tekli değişim hareketleri için kısa, orta ve uzun tabu süreleri.

$K_{idts}$ ,  $O_{idts}$ ,  $U_{idts}$ : ikili değişim hareketleri için kısa, orta ve uzun tabu süreleri.

Dinamik tabu sürelerinin hesaplanmasında  $0 < k_r \leq 1$  ve  $0 < b_r \leq 1$  olmak üzere iki farklı değer kullanılmıştır. Eğer belirli iterasyon boyunca uygun çözüm bulunamıyor veya amaç fonksiyonunda iyileşme yoksa tüm iyi hareketler tabudur. Bu durumda önceden belirlenmiş oran kadar tabu süresi azaltılmaktadır ( $K_{tdts}=k_r \cdot tdts$  veya  $K_{idts}=k_r \cdot idts$ ). Eğer çözümler önceden belirlenmiş olan "aynı çözüm tekrar edebilme" sınırını aşmışsa, tabu süresi belirlenen oran kadar artırılmaktadır ( $U_{tdts}=b_r \cdot tdts$  veya  $U_{idts}=b_r \cdot idts$ ).

Bu çalışmada, genel amaca göre tabu yıkma kriteri kullanılmıştır. Genel amaca göre tabu yıkma, arama boyunca bulunan en iyi çözümden daha düşük maliyet veren bir hareketin tabu olmasına rağmen yapılabilmesidir. Bu kriterin kullanımıyla, yüksek kaliteli yerel optimumların, belki de global optimumun, tabu sürelerinden dolayı arama alanı dışında bırakılması önlenmektedir.

### 3.4.3. Tavlama planı ve kabul fonksiyonu

TB algoritmasında aşağıdaki parametreler kullanılmıştır:

$T_0$  : Sıcaklık parametresinin başlangıç değeri

$T_s$  : Sıcaklık parametresinin son değeri

$T_m$  : Sıcaklık parametresinin mevcut değeri

$\alpha$  : Sıcaklık azaltma oranı (  $0 < \alpha < 1$  )

$k$  : Her sıcaklık değerinde kabul edilecek hareket sayısını gösteren sabit

Tavlama planına göre, sıcaklık parametresinin değeri ( $T_m=T_0$ ), her sıcaklıkta kabul edilen hareket sayısı  $k$  sınırına ulaşıncaya kadar sabit tutulmakta ve daha sonra  $\alpha$  oranında azaltılmaktadır ( $T_m=\alpha.T_m$ ).  $T_m < T_s$  koşulu sağlandığı zaman algoritma durdurulmaktadır.

Geliştirilen TB algoritmasında, basit TB’de kullanılan standart kabul fonksiyonu kullanılmıştır. Buna göre mevcut çözümü gösteren  $x$  çözümünden, komşu üretme stratejisine göre herhangi bir hareketle elde edilen bir  $x'$  komşu çözümünün, yeni mevcut çözüm olarak kabul edilip edilmemesi (3.12) ve (3.13) eşitliği ile belirlenmektedir. Eğer (3.12) eşitliğinin sonucu negatif bir sayı çıkarsa, (3.13) eşitliğine göre pozitif bir sayının üstel değeri 1 veya daha büyük olacağı için amaç fonksiyonunda iyileşme sağlayan tüm hareketler kabul edilmektedir.

$$\Delta = f(x') - f(x) \quad (3.12)$$

$$\begin{cases} x = x' & U(0,1) \leq \exp(-\Delta / T_m) \\ x = x & \text{d.d.} \end{cases} \quad (3.13)$$

#### 3.4.4. Uzun dönemli tabu arama

Bu çalışmada aramanın çeşitlendirilmesini yani çözüm uzayındaki farklı bölgelerin araştırılmasını sağlamak amacıyla dinamik tabu sürelerinin kullanımının yanı sıra hareket mekanizmasının değiştirilmesi esasına dayanan değişken komşuluk arama yöntemi kullanılmıştır. Eğer yapılan hareket sonucunda amaç fonksiyonu daha iyileşmişse aynı hareket mekanizmasıyla çözüm aramaya devam edilmektedir. Amaç fonksiyonunda iyileşme olmuyor ise, hareket mekanizması değiştirilmektedir.

Kullanılan kuvvetlendirme stratejisiyle, arama boyunca karşılaşılan ve iyi çözümler içerdiği görülen arama alanlarının yeniden ve daha detaylı olarak araştırılması amaçlanmaktadır. Bu stratejiye göre, öncelikle dinamik tabu sürelerini ve değiştirilmiş hareket mekanizmasını kullanan TB algoritması  $T_s$  sayısı ile ifade edilen son sıcaklığa düşene kadar çalıştırılmaktadır. Elit listesi adı verilen ve  $elu$  ile gösterilen sabit sayıda çözümün kaydedilebileceği bir listeye, TB algoritmasının çalıştırılması sonucunda elde edilen en iyi çözümler kaydedilmektedir. Eğer elit listede kayıtlı çözümlerin sayısı  $elu$  değerine ulaşmışsa, kaydedilmek istenen bir çözümün listedeki herhangi bir çözümden daha iyi olması gerekmektedir. Bu durumda kaydedilen yeni çözüme karşın, kayıtlı en kötü çözüm listeden silinmekte ve elit listesinin uzunluğu sabit kalmaktadır. Ayrıca, bu liste artan maliyet sırasına göre düzenlenmektedir. TB algoritmasının  $T_s$  değeri son sıcaklığa düştükten sonra, kısa dönemli hafızadaki bütün bilgiler silinmekte ve elit listedeki en iyi çözüm, başlangıç çözümü olarak alınıp  $T_s$  değeri %50 oranında artırılarak TB algoritması yeniden çalıştırılmaktadır. Yüksek sıcaklıklarda kötü çözümlerin kabul edilme olasılığı daha yüksek olduğu için elit listedeki çözümler düşük sıcaklıklarda aranma yoluna gidilmiştir. Böylece kötü çözümler daha az kabul edilecek ve farklı çözüm noktalarına kaymadan olduğu noktanın komşuları daha iyi araştırılacaktır. Bu işlemler elit listesindeki bütün çözümler incelenene kadar devam etmektedir. Şekil 3.9'de melez algoritmasının tüm adımları verilmiştir.

**Adım 0:** Bir başlangıç sıcaklığı seç:  $T_0 > 0$  , Son sıcaklık değeri seç:  $T_s > 0$   
Soğutma oranı seç:  $0 < \alpha < 1$  , Her sıcaklıktaki iterasyon sayısı seç: *ito*  
 $h_1(x, r)$  :  $x$  düğümünün  $r$ . tura eklenmesi (tekli değişim)  
 $h_2(x, j)$  :  $x$  düğümü ile  $j$  düğümünün ile birbiriyle yer değiştirmesi (ikili değişim)  
td\_uç = 1 (tekli değişim hareketi ile daha iyi çözüm varsa veya metropolis fonksiyonu tarafından kötü çözüm kabul edilmişse 1, yoksa 0 değerini alır)  
id\_uç = 0 (ikili değişim hareketi ile daha iyi çözüm varsa veya metropolis fonksiyonu tarafından kötü çözüm kabul edilmişse 1, yoksa 0 değerini alır)

**Adım 1:** Paralel turlu kurucu sezgisel ile başlangıç çözümü bul:  $X_0$

sayac = 0 , iterasyon = 0 ,  $l = 1$

$X_{best} = X_0$  ,  $f(X_{best}) = f(X_0)$  ,  $X_{mevcut} = X_0$  ,  $f(X_{mevcut}) = f(X_0)$

elit\_list [1] =  $X_0$  ,  $f(\text{elit\_list}[1]) = f(X_0)$

$s = 2, \dots, elu$  için elit\_list [s] = 0 ve  $f(\text{elit\_list}[s]) = \infty$

**Adım 2:** Tüm turlardaki düğüm sayılarını bul.

Eğer bir turda tek kalmış  $x$  düğümü varsa, tur boşaltma mekanizmasını çalıştır.

Mevcut çözümün komşularını bul: ( $x$  düğümü  $r_l$ . turda yer almaktadır)

$X_{komşu}(k) = h_1(x, r)$  ;  $k = 1, \dots, \text{No\_tdks}(x)$  ;  $r = 1, \dots, m$  ;  $r \neq r_l$

En düşük amaç fonksiyonu değerine sahip komşu çözümü seç:  $X_{komşu}(k)^*$

Eğer  $X_{komşu}(k)^*$  komşu çözümü uygun bir çözümse,

$X_{mevcut} = X_{komşu}(k)^*$

$m = m - 1$

Eğer  $f(X_{mevcut}) < f(X_{best})$  ise

$f(X_{best}) = f(X_{mevcut})$  ,  $X_{best} = X_{mevcut}$

elit listesini güncelle (uzun dönemde güncelleme yapma)

herhangi bir  $s = 2, \dots, elu$  için  $f(X_{best}) < f(\text{elit\_list}[s])$  ise

$f(\text{elit\_list}[s]) = f(X_{best})$  ,  $\text{elit\_list}[s] = X_{best}$

elit listeyi sırala

Adım 2'e dön.

Değilse Adım 3'e git

Değilse Adım 3 git.

**Adım 3:** Eğer sayac < *ito* ise

rassal olarak bir düğüm seç:  $x$  ( $x = 2, \dots, n$ )

Değilse  $T_0 = \alpha T_0$

Eğer  $T_0 < T_s$  ise Adım 5'e git

Değilse

Her sıcaklıktaki iterasyon sayacını sıfırla: sayac = 0

rassal olarak bir düğüm seç:  $x$  ( $x = 2, \dots, n$ )

**Adım 4:** Eğer td\_uç = 1 Adım 4.1'e git , Eğer id\_uç = 1 Adım 4.2'e git

**Adım 4.1:** Mevcut çözümün komşularını bul (tekli değişim hareket mekanizması):

$X_{komşu}(k) = h_1(x, r)$ ;  $k = 1, \dots, \text{No\_tdks}(x)$ ;  $r = 1, \dots, m$ ;  $r \neq r_l$  ( $x$  düğümü  $r_l$ . turdadır)

Adım 4.3'e git.

**Adım 4.2:** Mevcut çözümün komşularını bul (ikili değişim hareket mekanizması):

$X_{komşu}(k) = h_2(x, j)$  ;  $k = 1, \dots, \text{No\_idks}(x)$  ;  $j = 1, \dots, n$  ;  $x \neq j$  , Adım 4.3'e git.

Şekil 3.9: Melez algoritmanın adımları devam ediyor

**Adım 4.3:** En düşük amaç fonksiyonu değerine sahip komşu çözümü seç:  $X_{\text{komşu}}(k)^*$   
Eğer  $X_{\text{komşu}}(k)$  komşu çözümü uygun bir çözümse,  
 $\Delta = f(X_{\text{komşu}}(k)) - f(X_{\text{mevcut}})$   
Eğer  $\Delta < 0$  ise (tabu yıkma kriteri)  
 $f(X_{\text{mevcut}}) = f(X_{\text{komşu}}(k)^*)$  ,  $X_{\text{mevcut}} = X_{\text{komşu}}(k)^*$   
Eğer  $f(X_{\text{mevcut}}) < f(X_{\text{best}})$  ise  
 $f(X_{\text{best}}) = f(X_{\text{mevcut}})$  ,  $X_{\text{best}} = X_{\text{mevcut}}$   
elit listesini güncelle (uzun dönemde güncelleme yapma)  
herhangi bir  $s = 2, \dots, \text{elu}$  için  $f(X_{\text{best}}) < f(\text{elit\_list}[s])$  ise  
 $f(\text{elit\_list}[s]) = f(X_{\text{best}})$  ,  $\text{elit\_list}[s] = X_{\text{best}}$   
elit listeyi sırala  
sayac = sayac + 1 , iterasyon = iterasyon + 1  
Eğer tekli değişim hareketi yapılmışsa,  $\text{td\_uç} = 1$ ,  $\text{id\_uç} = 0$  ve Adım 2'e dön.  
Değilse  $\text{id\_uç} = 1$ ,  $\text{td\_uç} = 0$  ve Adım 4'e dön.

Eğer  $\Delta > 0$  ve  $h_1(x, r)$  veya  $h_2(x, j)$  hareketi tabu değilse  
 $y = U(0, 1)$  düzgün dağılıma göre sayı üret.  
 $Z = \exp(-\Delta / T_0)$   
Eğer  $y < z$  ise  
 $f(X_{\text{mevcut}}) = f(X_{\text{komşu}}(k)^*)$  ,  $X_{\text{mevcut}} = X_{\text{komşu}}(k)^*$   
sayac = sayac + 1 , iterasyon = iterasyon + 1  
Eğer tekli değişim hareketi yapılmışsa,  $\text{td\_uç} = 1$ ,  $\text{id\_uç} = 0$   
 $\text{tdtb}[x] = \text{iterasyon}$  (tabu güncelleştirme)  
Adım 2'e dön.  
Değilse  $\text{id\_uç} = 1$ ,  $\text{td\_uç} = 0$   
 $\text{idtb}[x, j] = \text{iterasyon}$  (tabu güncelleştirme)  
Adım 4'e dön.

Değilse  
sayac = sayac + 1 , iterasyon = iterasyon + 1  
Eğer tekli değişim hareketi yapılmışsa  
 $\text{td\_uç} = 0$ ,  $\text{id\_uç} = 1$ ,  $\text{tdtb}[x] = \text{iterasyon}$  (tabu güncelleştirme)  
Değilse  
 $\text{id\_uç} = 0$ ,  $\text{td\_uç} = 1$ ,  $\text{idtb}[x, j] = \text{iterasyon}$  (tabu güncelleştirme)  
Adım 4'e dön.

Değilse  
sayac = sayac + 1 , iterasyon = iterasyon + 1  
Eğer tekli değişim hareketi yapılmışsa,  
 $\text{td\_uç} = 0$  ,  $\text{tdtb}[x] = \text{iterasyon}$  (tabu güncelleştirme)  
Değilse,  
 $\text{id\_uç} = 0$  ,  $\text{idtb}[x, j] = \text{iterasyon}$  (tabu güncelleştirme)  
Eğer  $\text{td\_uç} = 0$  ve  $\text{id\_uç} = 0$  ise tamir fonksiyonu çalıştır.  
 $\text{td\_uç} = 1$   
Adım 4'e dön.

**Adım 5:** Eğer  $l > \text{elu}$  ise  
DUR  
Değilse  
 $X_{\text{mevcut}} = \text{elit\_list}[l]$  ,  $f(X_{\text{mevcut}}) = f(\text{elit\_list}[l])$   
 $T_0 = (1.50)^* T_0$  , sayac = 0 , iterasyon = 0  
 $\text{tdtb}[i] = -\infty$  ;  $i = 1, \dots, n$   
 $\text{idtb}[i, j] = -\infty$  ;  $i, j = 1, \dots, n$   
 $l = l + 1$   
Adım 2'e dön.

Şekil 3.9: Melez algoritmanın adımları

### 3.4.5. Deneysel çalışma

Algoritmanın göstereceği performans, algoritma parametrelerine verilecek değerlerle yakından ilgilidir. Bu nedenle, GDP için geliştirilen TA, TB ve DKA modern sezgisellerinden oluşmuş melez algoritmanın parametre setlerini ve etkinliklerini belirlemek amacıyla deneysel çalışma yapılmıştır. Geliştirilen melez algoritma, optimum çözümü bilinen test problemi üzerinde buldukları çözümlerin, problemin optimum çözümünden sapma oranına göre karşılaştırılmıştır. Optimum çözümünden sapma oranı (3.14) eşitliği ile hesaplanmaktadır.

$$OS = \frac{SÇ - OPT}{OPT} * 100 \quad (3.14)$$

SÇ: Sezgisel algoritma ile elde edilen çözüm

OPT: Problemin optimum çözümü

OS: Optimumdan sapma oranı (%)

Test problemi olarak İzmir bölgesi seçilmiştir ve optimum çözüm olarak bölüm 3.3'de anlatılan karar modelinin sonucu alınmıştır. Geliştirilen melez algoritma BORLAND C++ dilinde kodlanmış ve denemeler Intel Pentium 4 3.0 GHz, 3.04 Ghz iki işlemcili ve 2 GB Ram bulunan bir bilgisayar sisteminde yapılmıştır.

#### 3.4.5.1. Kurucu sezgisel için uygun parametre setinin belirlenmesi

(3.4) eşitliğinde  $f_3(r)$  değeri seçilen başbayinin ( $j$ ),  $r$  turu üzerindeki en uygun yerini tespit etmede kullanılmaktadır. Yapılan ön çalışmalarda,  $\beta_1$  ve  $\beta_2$  ağırlık değerlerinin değişmesiyle sonuçların değiştiği gözlenmiştir. Her problem için en iyi parametre setinin belirlenmesi için denemeler yapılmıştır. Kullanılan parametre setleri Çizelge 3.12'deki gibidir.



Çizelge 3.12: Paralel turlu kurucu sezgisel parametre değerleri

Parametreler	Faktörler	Düzeyler				
	$\beta_1$	0	0.25	0.5	0.75	1
	$\beta_2$	1	0.75	0.5	0.25	0

Kurucu sezgisel algoritmasında olasılıklı hiçbir yapı olmadığı için her çalıştırıldığında aynı sonucu vermektedir. Bu yüzden seçilen faktörlerin oluşturduğu her kombinasyon düzeyinde 1 deneme yapılmıştır. Sonuçlar Çizelge 3.13'da verilmiştir.

Çizelge 3.13: Paralel turlu kurucu sezgisel deney sonuçları

Bölgeler	OPT (km)	$\beta_1: 0$ $\beta_2: 1$		$\beta_1: 0.25$ $\beta_2: 0.75$		$\beta_1: 0.5$ $\beta_2: 0.5$		$\beta_1: 0.75$ $\beta_2: 0.25$		$\beta_1: 1$ $\beta_2: 0$	
		SÇ (km)	OS (%)	SÇ (km)	OS (%)	SÇ (km)	OS (%)	SÇ (km)	OS (%)	SÇ (km)	OS (%)
Adana	5755	5850	1,65	5817	1,08	5817	1,08	5819	1,11	5819	1,11
Ankara	5734	5806	1,26	5806	1,26	5813	1,38	5813	1,38	5813	1,38
Antalya	2698	2705	0,26	2705	0,26	2705	0,26	2705	0,26	2705	0,26
İstanbul	4314	4757	10,3	4526	4,91	4582	6,21	4705	9,06	4709	9,16
İzmir	2897	3005	3,73	3000	3,56	3046	5,14	3046	5,14	3052	5,35
Trabzon	1459	1459	0	1459	0	1459	0	1459	0	1459	0

Çizelge 3.13 incelendiğinde, her bölge için parametre setlerinin farklı çıktığı görülmektedir. Bunun nedeni uzaklık ve zaman matrislerinin yapılarının farklı olmasından kaynaklanmaktadır. Her bölge için en iyi parametre setleri aşağıda verilmiştir.

- Adana :  $\beta_1 = 0.25$  ,  $\beta_2 = 0.75$  veya  $\beta_1 = 0.5$  ,  $\beta_2 = 0.5$   
 Ankara :  $\beta_1 = 0$  ,  $\beta_2 = 1$  veya  $\beta_1 = 0.25$  ,  $\beta_2 = 0.75$   
 Antalya : Hepsi aynı sonuç vermektedir.  
 İstanbul :  $\beta_1 = 0.25$  ,  $\beta_2 = 0.75$   
 İzmir :  $\beta_1 = 0.25$  ,  $\beta_2 = 0.75$   
 Trabzon : Hepsi aynı sonuç vermektedir.

### 3.4.5.2. TB algoritması için uygun parametre setinin belirlenmesi

Geliştirilen melez algorithmda kısa dönemli tabu arama yerine TB kullanılmaktadır. TB algoritması, seçilen tavlama planına karşı oldukça duyarlıdır. Eğer  $T$  sıcaklık parametresinin değeri çok yüksek seçilirse algoritma iyi çözümlere yakınsayamamakta, aksi halde de bulunduğu yerel optimumların kalitesi düşmektedir. Bu yüzden, TB algoritmasının en iyi parametre setinin belirlenmesi amacıyla deneysel çalışma yapılmıştır.

TB algoritmasının, başlangıç sıcaklığı ( $T_0$ ), algoritmanın ne zaman sona ereceğini belirleyen bitiş sıcaklığı ( $T_s$ ), her sıcaklıkta kabul edilmesi gereken hareket sayısı ( $k$ ),  $k$  sayısı tamamlandıktan sonra sıcaklığın ne kadar azalacağını belirleyen sıcaklık azaltma oranı ( $\alpha$ ) olmak üzere 4 farklı parametre mevcuttur. Deneysel çalışmaya dahil edilen faktörler ve bunların düzeyleri Çizelge 3.14.'de verilmektedir.

Çizelge 3.14: Tavlama Benzetimi parametre değerleri

	<b>Faktörler</b>	<b>Düzyer</b>		
<b>Parametreler</b>	$T_0$	50	100	150
	$T_s$	20	10	5
	$\alpha$	0.99	0.95	0.90
	$k$	5	10	15

Başlangıç sıcaklığı için seçilen 3 farklı düzey 50, 100 ve 150 değerleri iken, bitiş sıcaklığı için 5, 10 ve 20 değerleri göz önünde tutulmuştur. Sıcaklığın hangi oranda azaltılacağını belirleyen  $\alpha$  değeri 0.90, 0.95 ve 0.99 düzeylerinde alınmış ve algoritma boyunca her sıcaklık değerinde kabul edilecek hareket sayısı 5, 10 ve 15 iterasyon sayısı olarak seçilmiştir. TB algoritması için seçilen faktörlerin oluşturduğu her kombinasyon düzeyinde paralel kurucu sezgisel başlangıç çözümü kullanılarak 5 deneme yapılmıştır. Buna göre elde edilen toplam deneme sayısı 405 (= 81 x 5) olmuştur. Bu denemelerde, performans ölçütü olarak optimumdan sapma oranı kullanılmıştır. Tüm kombinasyonlar ayrı ayrı, İzmir bölgesi üzerinde denenerek en iyi set seçilmiştir. İzmir bölgesinin optimal sonucu 2897 km'dir. Çizelge 3.15.'de görüldüğü gibi en küçük optimumdan sapma değeri,

21 nolu parametre setinde oluşmuştur. Yani  $T_0 = 50$  ;  $T_s = 5$  ve  $\alpha = 0.99$  ve  $k=15$  olarak alındığında, algoritmamızın performansı en iyi konumundadır. Bundan sonra çözülen tüm problemlerde, bu parametre seti kullanılmıştır.  $T_0 = 100$ ;  $T_s = 5$ ,  $\alpha = 0.99$  ve  $k=15$  parametre setinde de yaklaşık olarak aynı sonucu elde etmemize rağmen, daha yüksek değerler seçilmeyerek, algoritmanın çalışma zamanının uzaması engellenmiştir.

Çizelge 3.15: Tavlama Benzetimi deney sonuçları

Set	$T_0$	$T_s$	$\alpha$	$k$	Ort w	OS %	Set	$T_0$	$T_s$	$\alpha$	$k$	Ort w	OS %
1	50	20	0.99	5	3034	4,73	41	100	10	0.95	10	3006,5	3,78
2	50	20	0.99	10	3028,5	4,54	42	100	10	0.95	15	3005,4	3,74
3	50	20	0.99	15	3025,5	4,44	43	100	10	0.90	5	3025	4,42
4	50	20	0.95	5	3050	5,28	44	100	10	0.90	10	3024	4,38
5	50	20	0.95	10	3049	5,25	45	100	10	0.90	15	3016	4,11
6	50	20	0.95	15	3048	5,21	46	100	5	0.99	5	2942,4	1,57
7	50	20	0.90	5	3052	5,35	47	100	5	0.99	10	2934	1,28
8	50	20	0.90	10	3052	5,35	<b>48</b>	<b>100</b>	<b>5</b>	<b>0.99</b>	<b>15</b>	<b>2922,7</b>	<b>0,89</b>
9	50	20	0.90	15	3049,5	5,26	49	100	5	0.95	5	3005,2	3,73
10	50	10	0.99	5	3018,5	4,19	50	100	5	0.95	10	3002	3,62
11	50	10	0.99	10	3005	3,73	51	100	5	0.95	15	3000	3,56
12	50	10	0.99	15	3000,6	3,58	52	100	5	0.90	5	3006,2	3,77
13	50	10	0.95	5	3027,2	4,49	53	100	5	0.90	10	3006,4	3,78
14	50	10	0.95	10	3028	4,52	54	100	5	0.90	15	2996,2	3,42
15	50	10	0.95	15	3026,5	4,47	55	150	20	0.99	5	2976	2,73
16	50	10	0.90	5	3032	4,66	56	150	20	0.99	10	2974,8	2,69
17	50	10	0.90	10	3029,5	4,57	57	150	20	0.99	15	2968	2,45
18	50	10	0.90	15	3029,5	4,57	58	150	20	0.95	5	2998,2	3,49
19	50	5	0.99	5	2946,4	1,71	59	150	20	0.95	10	2996,5	3,43
20	50	5	0.99	10	2944	1,62	60	150	20	0.95	15	2984	3,00
<b>21</b>	<b>50</b>	<b>5</b>	<b>0.99</b>	<b>15</b>	<b>2921,6</b>	<b>0,85</b>	61	150	20	0.90	5	3000	3,56
22	50	5	0.95	5	3007,2	3,80	62	150	20	0.90	10	3002,4	3,64
23	50	5	0.95	10	3005	3,73	63	150	20	0.90	15	3000	3,56
24	50	5	0.95	15	3005	3,73	64	150	10	0.99	5	2966	2,38
25	50	5	0.90	5	3032,2	4,67	65	150	10	0.99	10	2964,5	2,33
26	50	5	0.90	10	3030	4,59	66	150	10	0.99	15	2968	2,45
27	50	5	0.90	15	3028	4,52	67	150	10	0.95	5	2998,2	3,49
28	100	20	0.99	5	3005	3,73	68	150	10	0.95	10	2996,5	3,43
29	100	20	0.99	10	3002,4	3,64	69	150	10	0.95	15	2984	3,00
30	100	20	0.99	15	3000	3,56	70	150	10	0.90	5	3007,2	3,80
31	100	20	0.95	5	3007,2	3,80	71	150	10	0.90	10	3005	3,73
32	100	20	0.95	10	3006,5	3,78	72	150	10	0.90	15	3005	3,73
33	100	20	0.95	15	3005	3,73	73	150	5	0.99	5	2962,4	2,26
34	100	20	0.90	5	3018	4,18	74	150	5	0.99	10	2954	1,97
35	100	20	0.90	10	3018	4,18	75	150	5	0.99	15	2942,7	1,58
36	100	20	0.90	15	3016,2	4,11	76	150	5	0.95	5	3005,4	3,74
37	100	10	0.99	5	3002	3,62	77	150	5	0.95	10	3004	3,69
38	100	10	0.99	10	3000,8	3,58	78	150	5	0.95	15	3000	3,56
39	100	10	0.99	15	3000,2	3,56	79	150	5	0.90	5	3004,2	3,70
40	100	10	0.95	5	3007,4	3,81	80	150	5	0.90	10	3004,4	3,71
							81	150	5	0.90	15	3008,2	3,84

Geliştirilen melez algoritma, Çizelge 3.16, Çizelge 3.17 ve Çizelge 3.18'deki parametre değerleri ile her bölge için 5'er kez koşturulmuş ve elde edilen çözüm değerleri Çizelge 3.19'de verilmiştir.

Çizelge 3.16: Paralel turlu kurucu sezgisel için en iyi parametre seti

Paralel turlu kurucu sezgisel parametreleri	Bölgeler	$\beta_1$	$\beta_2$
	Adana	0.5	0.5
	Ankara	0.25	0.75
	Antalya	0.5	0.5
	İstanbul	0.25	0.75
	İzmir	0.25	0.75
	Trabzon	0.5	0.5

Çizelge 3.17: Tavlama Benzetimi için en iyi parametre seti

TB Parametreleri	$T_0$	$T_s$	$\alpha$	$k$
	50	5	0.99	15

Çizelge 3.18: Tabu Arama için parametre seti

TA parametreleri	Dinamik tabu listesi kullanılmıştır	
	Tabu liste uzunluk (tekli değişim ve ikili değişim)	$\sqrt{n}$
	Değişim oranı	%10
	İzin verilen uygun olmayan çözüm sayısı	10
	İzin verilen tekrarlanan çözüm sayısı	5

Çizelge 3.19: Melez algoritma sonuçları

Bölgeler		1	2	3	4	5	Ort
Adana $n=29$ $m=11$	w (km)	5817	5763	5817	5763	5817	<b>5795,4</b>
	süre (sn)	16	15	15	16	15	<b>15,4</b>
Ankara $n=37$ $m=11$	w (km)	5751	5753	5751	5743	5753	<b>5750,2</b>
	süre (sn)	225	235	227	232	224	<b>228,6</b>
Antalya $n=21$ $m=6$	w (km)	2702	2702	2702	2702	2702	<b>2702</b>
	süre (sn)	4	4	4	4	4	<b>4</b>
İstanbul $n=55$ $m=14$	w (km)	4397	4438	4397	4418	4418	<b>4413,6</b>
	süre (sn)	722	738	718	733	712	<b>725,2</b>
İzmir $n=32$ $m=6$	w (km)	2919	2897	2941	2919	2897	<b>2914,6</b>
	süre (sn)	21	22	22	22	21	<b>21,6</b>
Trabzon $n=14$ $m=3$	w (km)	1459	1459	1459	1459	1459	<b>1459</b>
	süre (sn)	0,01	0,01	0,01	0,01	0,01	<b>0,01</b>

### 3.4.6. Karar modeli ve melez algoritma sonuçları

Çizelge 3.20'de her bölgenin Cplex 10.0 paket programı ve geliştirilen melez algoritmasıyla çözüm sonuçları birbirleriyle karşılaştırılarak verilmiştir.

Çizelge 3.20: Karar modeli ve melez algoritma sonuçlarının karşılaştırılması

Bölgeler	Geliştirilen Melez Algoritma		Geliştirilen Karar Modeli		(1)-(2) arasındaki % fark
	$\bar{w}$ (km) (1)	süre (sn)	$w$ (km) (2)	süre (sn)	
Adana $n=29$ $m=11$	5795,4	15,4	5755	205	<b>0,70</b>
Ankara $n=37$ $m=11$	5750,2	228,6	5734	3523	<b>0,28</b>
Antalya $n=21$ $m=6$	2702	4	2698	0,20	<b>0,15</b>
İstanbul $n=55$ $m=14$	4413,6	725,2	4314*	54000	<b>2,31</b>
İzmir $n=32$ $m=7$	2914,6	21,6	2897	17	<b>0,61</b>
Trabzon $n=14$ $m=3$	1459	0,01	1459	0,02	<b>0</b>

Melez algoritma ile en iyi çözümü bilinen problemlerde (Adana, Ankara, Antalya, İzmir, Trabzon) en iyi değerden % 0 ile % 0,70 arasında bir ortalama sapma elde edilmiştir. Bunun yanısıra Trabzon bölgesinin çözümünde en iyiye ulaşılmıştır. En iyi değer bilinmediği İstanbul bölgesinde ise Cplex ile 54.000 sn içerisinde bulunan çözümden %2,31 daha kötü sonuç bulunmuştur.

GDP kısa dönemli planlama problemleri arasındadır ve çok sık aralıklarla çözümüne ihtiyaç duyulabilir. Bu nedenle, algoritma için kritik performans ölçütlerinden biri de koşum zamanıdır. Geliştirilen tamsayılı doğrusal karar modeli İstanbul bölgesi hariç, 5 bölgenin optimal sonuçlarını kısa sürede bulmuştur.

#### 4. SONUÇ VE ÖNERİLER

Gazete dağıtımı, günlük hayatta genellikle kişilerin sezgi veya tecrübelerine dayanarak yapılmakta, bu nedenle ortaya çıkan planlar, maliyet veya süre açısından istenilen düzeyde olmamaktadır. GDP yöneylem araştırmasında zengin bir literatüre sahip değildir. Literatürde incelenen çalışmalarda ise, problemin NP-zor oluşu ve problem boyutunun artmasıyla birlikte çözüm süresinin üstel olarak artması sonucu, çözüm için yapılan çalışmaları daha çok sezgisel yöntemler üzerinde yoğunlaşmasına neden olmuştur. Bu durumda bile mevcut uygulamalara kıyasla çok daha iyi çözümler elde edilmiştir.

Bu çalışmada ele alınan problem için, tamsayılı doğrusal karar modeli yardımıyla İstanbul bölgesi haricinde diğer bölgelerin en iyi çözümleri bulunarak, mevcut uygulamaya göre %21 oranında maliyet azaltımı sağlanmıştır. Ayrıca, firmanın dağıtım yaparken kullandığı araç sayısında da %20 azalma sağlanmıştır. Makul boyutlarda olan bölgeler için modelin en iyi çözümü bir paket program yardımıyla rahatlıkla bulunmuştur. Tamsayılı doğrusal karar modeli sonuçlarına göre ele alınan firma, 48 araçla ve günde 22.957 km yol katederek, Türkiye genelinde 188 başbayisine saat 05:30'a kadar gazeteleri ulaştırabilmektedir.

Çalışma sonucuna göre, tamsayılı karar modeli kurarak, ekonomik yönüyle günümüz işletmelerinde dağıtım maliyetinin enküçüklenmesiyle tasarrufların yapılabileceği ve mevcut dağıtım maliyetlerinin daha alt seviyelerde gerçekleştirilebileceği gösterilmiştir. Ayrıca en iyi çözümü hedeflemesi bakımından, önceki çalışmalardan ayrılmaktadır. Geliştirilen karar modeli, probleme özgü başka kısıtların da doğrudan eklenmesine izin verecek kadar esnek olduğu için, benzer yapıda fakat değişik kısıtlar da içeren (en fazla gidilebilir yol kısıtı gibi) problemlerin çözümünde kullanılabilir.

Çok bayili bölgelerin yaklaşık çözümünü makul sürelerde bulmak amacıyla TA, TB ve DKA modern sezgisellerinden oluşan melez bir algoritma geliştirilmiş ve algoritmanın en iyi parametre setinin belirlenmesi için deneysel çalışmalar yapılmıştır. Yapılan deneysel çalışmalar sonucunda, başlangıç çözümü paralel

turlu kurucu sezgiselle belirlendiğinde, başlangıç sıcaklığı ( $T_0$ ) 50, algoritmanın ne zaman sona ereceğini belirleyen bitiş sıcaklığı ( $T_s$ ) 5, her sıcaklıkta kabul edilmesi gereken hareket sayısı ( $k$ ) 15,  $k$  sayısı tamamlandıktan sonra sıcaklığın ne kadar azalacağını belirleyen sıcaklık azaltma oranı ( $\alpha$ ) 0.99, tekli değişim ve ikili değişim tabu süreleri  $\sqrt{n}$  olarak alındığında, algoritmanın iyi sonuçlara oldukça kısa sürelerde ulaştığı görülmüştür.

Bu çalışmada problem, tek ürünlü, sabit talep değerleri ve homojen araç filosuyla çözülmüştür. Çalışmanın devamında, çok ürünlü, stokastik talep değerleri olan ve heterojen araç filosuyla dağıtım problemi olarak düşünülebilir. Geliştirilen melez algoritmanın kullandığı hareket tipinin sayısı artırılarak, daha iyi sonuçlar elde edilebilir.

## KAYNAKLAR LİSTESİ

- [1] Holt, J.N., Watts, A.M., Vehicle routing and scheduling in the newspaper industry, In: B. Golden, A. Assad (Eds.), Vehicle Routing: Methods and Studies, North-Holland, New York, 1988.
- [2] Mantel, R.J., Fontein, M., A Practical solution to a newspaper distribution problem, International Journal of Production Economics, 30-31, 591-599, 1993
- [3] Hurter, A.P., Buer, M. V., The newspaper production/distribution: Medium sized newspapers, The Journal of Business Logistics 17 (1), 85-108, 1996.
- [4] Ree, S., Yoon, B., A two-stage heuristic approach for the newspaper delivery problem, Computers and Industrial Engineering, 30(3), 501-509, 1996.
- [5] Dillmann, R., Becker, B., Beckefeld, V., Practical aspects of route planning for magazine and newspaper wholesalers, European Journal of Operational Research 90, 1-12, 1996
- [6] Buer, M.G.V., Woodruff, D.L., Olson, R.T., Solving the medium newspaper production/distribution problem. European Journal of Operational Research, 115, 237-253, 1999
- [7] Song, S.H., Lee, K. S., Kim G.S., A practical approach to solving a newspaper logistics problem using a digital map, Computers and Industrial Engineering, 43, 315-330, 2002.
- [8] Russell, R., Chiang, W.C., Zepeda, D., Integrating multi-product production and distribution in newspaper logistics, European Journal of Operational Research, 35 (5), 1576-1588, 2008
- [9] Kara, İ., Integer programming formulations for distance and capacity constrained vehicle routing problem, Teknik rapor, Ankara, Türkiye, 2007.
- [10] Kara, İ., Laporte, G., Bektaş, T., A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem, European Journal of Operational Research, 158, 793-795, 2004.
- [11] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Concorde: a code for solving traveling salesman problem, [www.tsp.gatech.edu/concorde.html](http://www.tsp.gatech.edu/concorde.html), 1997.
- [12] <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>
- [13] Solomon M., Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research, 21:498–516, 1987.
- [14] Taillard, E., 1990, Some efficient heuristic methods for the flow shop sequencing problem, European Journal of Operational Research, 47, 65-74.



- [15] Woodruff, D.L., Simulated annealing and tabu search: lessons from a line search, *Computers & Operations Research*, 21(8), 823-839, 1994.
- [16] Laguna, M., Glover, F., Bandwidth packing: a tabu search approach, *Management Science*, 39(4), 492-500, 1993.
- [17] Chiang, W.C., Kouvelis, P., An improved tabu search heuristic for solving facility layout design problem, *International Journal of Production Research*, 34(9), 2565-2585, 1996.
- [18] Costamagna, E., Fanni, A., Giacinto, G., A tabu search algorithm for the optimization of telecommunication networks, *European Journal of Operational Research*, 106, 357-372, 1998.
- [19] Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., Optimization by simulated annealing, *Science*, 220(4598), 671-680, 1983.
- [20] Cerny, V., A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, *Journal of optimization theory and applications*, 45, 41–51, 1985.
- [21] Reeves, Colin, *Modern heuristic techniques for combinatorial problems*, John Wiley & Sons, New York, A.B.D., 1993.
- [22] Alabaş, Ç., Tabu arama ve tavlama benzetimi algoritmalarıyla bilgisayar şebekelerinin topolojik optimizasyonu, *Gazi Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi*, 29-56, 1999.
- [23] Eglese, R.W., Simulated annealing: a tool for operational research, *European Journal of Operational Research*, 46, 271-281, 1990.
- [24] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, 21, 1087-1092, 1953.
- [25] Aarst, J., Korst, J., *Simulated annealing and Boltzmann machines*, Wiley, 1989.
- [26] Hansen, P., The steepest ascent mildest descent heuristic for combinatorial programming, *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy, 1986.
- [27] Glover, F., Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, 1(3), 533-549, 1986.
- [28] Glover, F., Tabu search – part I, *ORSA Journal on Computing*, 1(3), 190-206, 1989.
- [29] Glover, F., Tabu search – part II, *ORSA Journal on Computing*, 2(1), 4-32, 1990.

- [30] Glover, F., Laguna, M., Modern heuristic techniques for combinatorial problems, Blackwell Syntefic Publications, 1993.
- [31] Marett, R., Wright, M., A comparison of neighborhood search techniques for multi-objective combinatorial problems, Computers & Operations Research, 23(5), 465-483, 1996.
- [32] Lokketangen, A., Tabu search – using the search experience to guide the search process. An introduction with examples, AICOM, 8(2), 78-85, 1995.
- [33] Glover, F., Laguna, M., Tabu Search, Kluwer Academic Publishers, 1997.
- [34] Kolahan, F., Liang, M., A tabu search approach to optimization of drilling operations, Computers & Industrial Engineering, 31(1/2), 371-374, 1996.
- [35] İslam, A., Ekşioğlu, M., A tabu search approach for the single machine mean tardiness problem, Journal of Operational Research Society, 48, 751-755, 1997.
- [36] Burkard, R.E., Çela, E., Heuristics for biquadratic assignment problems and their computational comparison, European Journal of Operational Research, 83, 283-300, 1995.
- [37] Nowicki, E., Smutnicki, C., A fast tabu search algorithm for the permutation flow-shop problem, European Journal of Operational Research, 91, 160-175, 1996.
- [38] Chiang, W.C., Kouvelis, P., An improved tabu search heuristic for solving facility layout design problem, International Journal of Production Research, 34(9), 2565-2585, 1996.
- [39] Glover, F., Taillard, E., Werra, D., A user's guide to tabu search, Annals of Operations Research, 41, 3-28, 1993.
- [40] Kim, Y.D., Lim, H.G., Park, M.W., Some heuristics for a flowshop scheduling problem in a printed circuit board assembly process, European Journal of Operational Research, 91, 124-143, 1996.

## **EK AÇIKLAMALAR LİSTESİ**

	Sayfa
<b>5.1.</b> Tavlama Benzetimi .....	49
<b>5.2.</b> Tabu arama .....	54
<b>5.3.</b> Değişken Komşuluk Arama .....	65

## 5. EK AÇIKLAMALAR

### 5.1. Tavlama Benzetimi

TB algoritması, katıların belirli bir başlangıç sıcaklığından başlayarak yavaş yavaş soğutulduğu tavlama sürecinin benzetimi olan olasılıklı bir arama algoritmasıdır. “Tavlama Benzetimi” ismi, katıların fiziksel tavlama süreci ile olan benzerliklerinden ileri gelmektedir. TB algoritması, birbirinden bağımsız olarak, Kirkpatrick vd. [19] ve Cerny [20] tarafından önerilmiştir. Son yıllarda birçok araştırmacı TB algoritmasını kombinatoriyal eniyileme problemlerinde kullanmaktadır. Günümüze kadar bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme gibi farklı alanlardaki bir çok optimizasyon problemlerine uygulanmıştır [21].

Tavlama terimi fiziksel olarak, ısı banyosundaki bir katının yüksek enerji durumlarından başlayarak daha düşük enerji durumlarının elde edilme sürecini temsil etmektedir. Bu süreç çok genel olarak iki işlemden oluşmaktadır (Kirkpatrick vd., [19]):

1. Isı banyosunun başlangıç sıcaklığının katının eriyebileceği bir değere kadar yükseltilmesi.
2. Katının parçacıkları kendini düzenleyene kadar ısı banyosunun sıcaklığının giderek azaltılması.

Katının sıvı durumunda tüm parçacıkları gelişi güzel hareket ederler. Katı durumda ise bir kafes şeklinde düzenlenirler. Bu durumda sistemin enerjisi en azdır ve bu duruma yer durumu denmektedir. Bir katının yer durumu, sıcaklık yeteri kadar yükseltilmiş ve soğutma da yeteri kadar yavaş yapılmışsa elde edilir. Aksi halde katının bulunduğu durum yarı kararlı bir durumdur (Alabaş, [22]). Tavlama süreci ile kombinatoriyal optimizasyon problemleri arasındaki benzerlik şöyle açıklanabilir: Katının farklı durumları, problemdeki mümkün farklı çözümlere ve sistem enerjisi

problemin amaç fonksiyonuna karşılık gelmektedir. Yer durumu global optimumu, yarı kararlı durum ise yerel optimumu göstermektedir (Eglese, [23]).

Metropolis vd. [24], bir katının ısı banyosunda, ısı dengeye kadar olan gelişimini benzetmek için basit bir algoritma sunmuşlardır. Açıklanan bu algoritma Monte Carlo tekniklerine dayanmaktadır ve aşağıda belirtilen yöntemle katının durumlarının yani kaydettiği aşamaların bir sırasını üretmektedir (Aarst ve Korst, [25]).

Mevcut durumu  $i$  ve enerjisi  $E_i$  olan katının, mevcut durumu bir hareket mekanizması kullanılarak bir sonraki  $j$  durumuna küçük bir değişiklik ile dönüştürülür. Böylece son durumun enerjisi  $E_j$  olmaktadır. Eğer  $(E_j - E_i)$  enerji farkı sıfıra eşit veya daha küçükse,  $j$  durumu yeni mevcut çözüm olarak kabul edilir. Eğer enerji farkı sıfırdan büyükse,  $j$  durumu (5.1) eşitliğine göre belirli bir olasılıkla kabul edilir. Bu eşitlikteki  $T$  ısı banyosunun sıcaklığını ve  $k_B$  "Boltzman sabiti" olarak bilinen fiziksel bir sabiti ifade etmektedir.

$$\exp\left(\frac{E_i - E_j}{k_B * T}\right) \quad (5.1)$$

Yukarıda verilen kabul kuralı "Metropolis kriteri" ve tanımlanan algoritma "Metropolis algoritması" olarak bilinmektedir. Eğer sıcaklık yeteri kadar yavaş azaltılırsa, katı her sıcaklıkta ısı dengeye ulaşabilecektir. Metropolis algoritmasında ısı dengeye ulaşabilmek için, verilen bir sıcaklık değerinde oldukça fazla sayıda geçişler (sonraki durumda hareketler) üretilir. Isıl denge, Boltzmann dağılımı ile karakterize edilmektedir. (5.2) eşitliğinde ifade edilen bu dağılım, katının  $T$  sıcaklığında  $E_i$  enerjili  $i$  durumunda olma olasılığını vermektedir.

$$P_T\{X = i\} = \frac{1}{Z(t)} \exp\left(\frac{-E_i}{k_B * T}\right) \quad (5.2)$$

(5.2) eşitliğindeki  $X$ , katının mevcut durumunu gösteren olasılıklı bir değişkendir.  $Z(t)$  ise bölme fonksiyonudur ve (5.3) eşitliğinde verilmektedir.

$$Z(t) = \sum_j \exp\left(\frac{-E_j}{k_B * T}\right) \quad (5.3)$$

(5.3) eşitliğinde mümkün tüm durumların toplamı dikkate alınmakta ve Boltzman dağılımı TB algoritmasının analizinde önemli bir rol oynamaktadır.

TB ile üretilen çözümlerin amaç fonksiyon değerleri genel bir azalma eğilimindedir. Fakat bazı durumlarda amaç fonksiyon değerleri yüksek olan çözümlerde kabul edilebilmektedir. Bu yolla, yerel minimum etrafında yapılan aramadan çıkılıp, daha iyi bir yerel belki de global bir minimum için aramaya devam etmek amaçlanmaktadır. TB'nin kombinatorial optimizasyon problemleri için optimuma yakın çözümler veren kullanışlı bir yöntem olarak kullanıldığı söylenebilir (Aarst ve Korst, [25]).

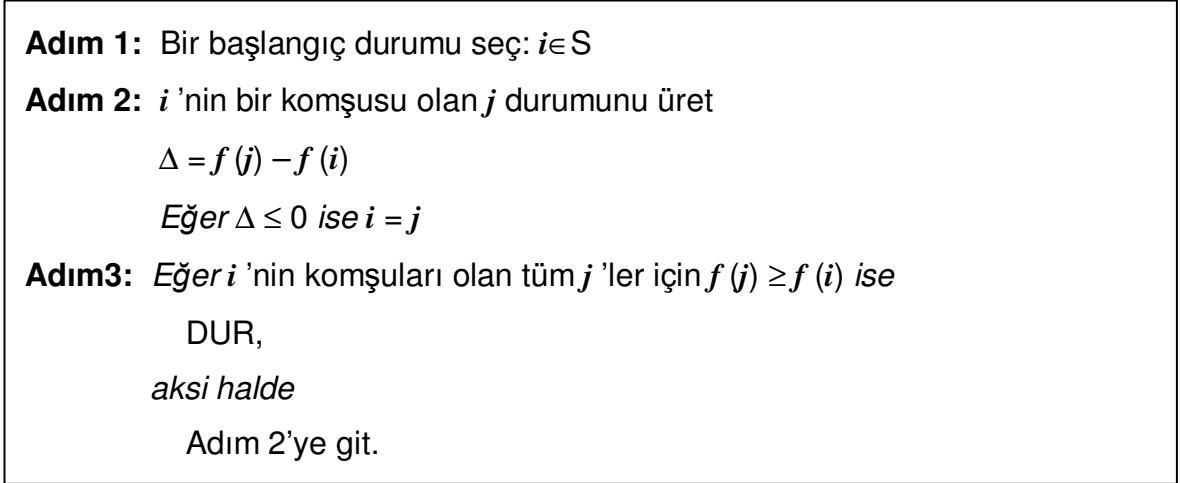
TB algoritmasında, metropolis algoritması kombinatorial optimizasyon problemlerinin çözümlerinin bir sırasını üretmek için kullanılmaktadır. Daha öncede bahsedildiği gibi, çok parçalı fiziksel bir sistem ile bir kombinatorial optimizasyon problemi arasındaki benzerliğin aşağıdaki eşitliklere dayandığı kabul edilmektedir.

- Problemdaki çözümler, fiziksel sistemdeki durumlara eşittir.
- Bir çözümün maliyeti, bir durumun enerjisine eşittir.

Tüm çözümlerin sonlu kümesi S çözüm uzayı ile ve S'in üyeleri için tanımlanan maliyet değerleri  $f$  maliyet fonksiyonu ile tanımlanacak olursa, söz konusu problem, tüm S üzerinde  $f$ 'i en küçükleyen ve  $i \in S$  olan bir  $i$  çözümünün veya durumunun bulunmasıdır.

TB komşu arama metoduna dayalı algoritmalarından birisidir. Komşu aramanın basit bir şekli olan "iniş algoritması", keyfi olarak seçilen bir başlangıç çözümü ile aramaya başlar. Daha sonra uygun bir hareket mekanizması ile bu çözümün bir komşusu üretilir ve maliyetteki değişim hesaplanır. Eğer maliyette bir azalma söz

konusuysa, komşu çözüm yeni mevcut çözüm olarak kabul edilir, aksi halde mevcut çözüm değişmez. Bu süreç mevcut çözümün hiçbir komşusu maliyette iyileşme sağlamayana kadar devam eder ve böylece iniş algoritması yerel bir minimumda durmuş olur. İniş algoritmasının program taslağı Şekil 5.1'de verilmektedir (Eglese, [23]).



Şekil 5.1. İniş algoritması

İniş algoritması basit ve hızlı olsa da, global minimumdan oldukça uzak olabilen yerel minimum çözümleri bulma dezavantajına sahiptir. Bulunan bu çözümün iyileştirilmesi için kullanılan bir yöntem, iniş algoritmasının farklı başlangıç çözümleriyle birkaç kez çalıştırılması ve bulunan en iyi yerel optimumun seçilmesidir. TB algoritması ise, bu stratejinin yerine, maliyette yükselmeye yol açan komşu hareketleri bazen kabul ederek, yerel optimum tuzaklarından kurtulmaya çalışmaktadır. Maliyette yükselmeye yol açan bir hareketin kabul edilip edilmemesi, kontrollü bir olasılığa göre rassal olarak belirlenmektedir. Maliyet fonksiyonu  $f$  'de  $\Delta$  kadar bir yükselmeye yol açan hareketin kabul edilme olasılığı "kabul fonksiyonu" olarak adlandırılır ve genellikle (5.4) eşitliği ile ifade edilir. Bu eşitlikte  $T$ , fiziksel tavlamadaki sıcaklığa karşılık gelen bir kontrol parametresidir.

$$\exp\left(\frac{-\Delta}{T}\right) \quad (5.4)$$

Kabul fonksiyonuna göre,  $f$  'de meydana gelen küçük artışların kabul edilme olasılığı, büyük artışların kabul edilme olasılığından daha fazladır. Ayrıca,  $T$  yüksek olduğunda hareketlerin çoğu kabul edilecektir.  $T$  sifıra yaklaştıkça ise,  $f$  'de artışa yol açan hareketlerin çoğu reddedilecektir. Bu nedenle TB algoritmasında, yerel optimum tuzaklarına düşülmesini engellemek için göreceli olarak yüksek bir  $T$  değeri ile aramaya başlanır. TB algoritması, bir taraftan sıcaklık yavaş yavaş azaltılırken, her sıcaklık değerinde belli sayıda hareket deneyerek arama işlemini sürdürür. Basit bir TB algoritmasının adımları Şekil 5.2.'de görülmektedir.

**Adım 1:** Bir başlangıç durumu seç:  $i \in S$   
 Bir başlangıç sıcaklığı seç:  $T > 0$   
 Sıcaklık değişim sayacını sıfırla :  $t = 0$

**Adım 2:** Durdurma koşulu sağlanmış ise DUR,  
 Değilse tekrar sayacını sıfırla:  $n=0$

**Adım 3:**  $i$  'nin bir komşusu olan  $j$  durumunu rassal olarak üret.  
 $\Delta = f(j) - f(i)$   
 Eğer  $\Delta < 0$  ise  $i = j$ , değilse ve  $U(0,1) < \exp(-\Delta/T)$  ise  $i = j$ .

**Adım 4:**  $n=n+1$  Eğer  $n < M$  ise Adım 3'e git, değilse  $t = t + 1$ ,  
 $T = T(t)$  ve Adım 2'ye git.

Şekil 5.2. Tavlama benzetimi algoritması

Şekil 3.2'de  $M$  her sıcaklık değerinde denenecek hareket sayısına ve  $T(t)$  sıcaklığın aldığı  $t$ . değere karşılık gelmektedir. TB algoritmasının global optimum çözümlere yakınsama hızı,  $M$  ve  $T(t)$ ,  $t = 0, 1, 2, \dots$  parametreleri tarafından belirlenmektedir. Ancak pratikte, algoritmanın parametre değerlerinin uygulamaya yönelik seçimi "tavlama" veya "soğutma" planı ile belirlenmektedir. TB algoritmasında, başlangıç sıcaklığının, sıcaklık azaltma oranının, her sıcaklıktaki tekrar sayısının ve durdurma koşulunun belirlenmesi tavlama veya soğutma planı olarak tanımlanmaktadır. Soğutma planının seçimi, algoritmanın performansı üzerinde çok önemli bir etkiye sahiptir.



## 5.2. Tabu Arama

Tabu Arama (TA), kombinatorial optimizasyon problemlerini çözmek için geliştirilmiş bir sezgisel tekniktir. Benzer bir yaklaşım Hansen [26] tarafından önerilmiş, ancak ilk olarak Glover [27] tarafından ortaya konan TA'nın bugünkü modern şekli, yine Glover [28], [29] tarafından verilmiştir. TA, günümüze kadar, çizelgeleme, ulaştırma, tesis yerleşimi, telekomünikasyon, uzman sistemler, sınır ağları gibi bir çok alana başarıyla uygulanmıştır (Glover ve Laguna, [30]).

TA, başlangıç çözümü, hareket mekanizması, aday liste stratejileri, hafıza, tabu yıkma kriterleri, durdurma koşulları olarak adlandırılan temel elemanlara sahiptir. TA'da başlangıç çözümü rassal olarak seçilebileceği gibi, başlangıç çözümünün belirlenmesinde herhangi başka bir algoritma kullanılabilir. Hareket mekanizması ise, mevcut çözümde yapılan bir değişiklikle elde edilebilecek yeni çözümleri belirler ve mümkün hareketler, mevcut çözümün tüm komşularını oluştururlar. Hareket mekanizmasının, problem yapısına bağlı olmasının yanı sıra uygun bir şekilde belirlenmesi bu metodun performansı açısından oldukça önemlidir. Zeki aramanın, daha sistematik yol gösterme yöntemlerine dayalı olması gerektiği varsayımından hareketle, TA'da rassallık ya hiç kullanılmaz ya da oldukça sınırlı bir yolla uygulanır. Buna göre, çoğu TA uygulamaları deterministiktir. Ancak TA'nın özel bir durumu, temel TA prensipleri tarafından yapılan hareket değerlendirmelerine dayanarak elde edilen olasılıklara göre hareketleri seçer (Glover ve Laguna, [30]).

Kombinatorial problemler için geliştirilen sezgisel algoritmaların çoğu, "komşu arama" metodunu kullanır. Komşu aramada, her mümkün çözümün, diğer mümkün çözümlerden oluşan komşuları vardır (Marett ve Wright, [31]). TA'nın temel özelliklerini bir optimizasyon problemi üzerinde ve komşu aramaya dayalı olarak göstermek daha açıklayıcı ve anlaşılması kolay olacaktır. Bu amaçla, (5.5) eşitliğinde tanımlanan minimizasyon problemi kullanılmıştır.

$$\begin{array}{ll} \text{Minimum} & f(x) \\ \text{Kısıt} & x \in X \end{array} \quad (5.5)$$

Burada  $f(x)$ , doğrusal veya doğrusal olmayan bir amaç fonksiyonudur.  $x \in X$  şartı,  $x$  vektörü üzerindeki tüm kısıtları içermektedir. Komşu aramada, her  $x \in X$  çözümü,  $N(x) \subset X$  olmak üzere, bir  $N(x)$  komşular kümesine sahiptir. “Hareket” olarak adlandırılan bir işlem ile  $x$  çözümünden direkt olarak bir  $x' \in N(x)$  çözümüne ulaşılabilir. Komşu aramada, komşuların simetrik olduğu kabul edilmektedir, yani  $x', x$  için bir komşusuysa,  $x$  de  $x'$  için bir komşudur. Komşu arama metodunun adımları Şekil 5.3’de tanımlanmaktadır. Burada, hareketlerin seçim kriterlerinin ve aramanın durdurulması kriterlerinin önceden verildiği kabul edilmiştir (Glover ve Laguna, [30]).

Komşu aramayı kullanan TA’daki en önemli temel eleman, arama işlemini kontrol etmek için kullanılan esnek hafızadır. Bunun için, arama boyunca karşılaşılan durumlar hakkında seçici bilgileri tutan bir  $H$  hafızası görevlendirilir.  $H$  hafızası, “kısa dönemli hafıza” olarak adlandırılmaktadır. TA, esnek hafıza fonksiyonlarını kullanarak, arama boyunca elde edilen geçmiş çözümlerin belli özelliklerini eski haline getiren hareketleri (çözümler arasındaki geçişler) yasaklar. Yapılmasına izin verilmeyen hareketler “tabu” olarak sınıflandırılırlar ve “tabu listesi” adı verilen bir liste üzerinde kısa dönemli hafızada kaydedilirler. Bu hareketler, belli bir süre sonra tabu listesinden çıkarılırlar ve tekrar yapılmalarına izin verilir (Glover ve Laguna, [30]).

**Adım 1:** (Başlangıç değerlerini verme)

**Adım 1.1:** Bir başlangıç çözümü seç:  $x \in X$ .

**Adım 1.2:** En iyi çözümü ve maliyetini kaydet:  $x_{eç} = x$  ve  $f_{eç} = f(x_{eç})$ .

**Adım 2:** (Seçim ve durdurma)

Bir çözüm seç:  $x' \in N(x)$ .

Eğer  $N(x)$ ’in hiçbir üyesi seçim kriterlerini sağlamıyorsa veya toplam tekrar sayısının belli bir sınırı geçmişse DUR.

**Adım 3:** (Güncelleştirme)

$x = x'$  güncelleştirmesini yap

Eğer  $f(x) < f_{eç}$  ise Adım 1.2’yi gerçekleştir. Daha sonra Adım 2’ye git.

Şekil 5.3. Komşu arama metodu adımları

TA'da, komşu arama metodundaki  $N(x)$  komşu kümesi yerine  $N(H,x)$  komşu kümesi kullanılır ve böylece hafıza her iterasyonda, mevcut çözümden bir hareketle hangi çözümlere ulaşılabileceğini belirlemiş olur. Kısa dönem hafızalı TA'da,  $N(H,x)$ ,  $N(x)$  komşu kümesinin bir altkümesidir ve  $N(x)$  komşu kümesindeki hangi çözümlerin  $N(H,x)$  komşu kümesinde yer almayacağını belirlemek için tabu sınıflandırmalarından yararlanır. Tabu sınıflandırmasında yer alan komşular,  $N(H,x)$  komşu kümesine dahil edilmeyecektir. TA,  $N(H,x)$  komşu kümesindeki en iyi çözümü, yeni mevcut çözüm olarak seçer. Ancak,  $N(H,x)$ 'de çok fazla komşu varsa veya bu komşuların değerlendirilmesi pahalıya mal oluyorsa, tüm  $N(H,x)$  komşu kümesinin araştırılması yerine bunun bir altkümesi araştırılır. TA'da söz konusu altküme, "aday listesi" olarak adlandırılmaktadır. Buna göre, kısa dönemli hafızanın kullanıldığı TA algoritması Şekil 5.4'de görülmektedir. Burada TA, herhangi bir başlangıç çözümü ile aramaya başlamaktadır. Her iterasyonda mevcut çözümün araştırılan komşuları içerisinde tabu olmayan bir hareketle elde edilebilen ve en iyi amaç fonksiyonu değerine sahip olanı seçilmekte ve bu hareketle mevcut çözüm, seçilen komşu çözüme dönüştürülmektedir.

İlerideki bölümlerde daha detaylı olarak açıklanacağı gibi, tabu olan bir hareket, tabu yıkma kriterlerinden birini veya birkaçını sağlıyorsa, tabu olmasına rağmen, mevcut çözümü oluşturmak için seçilebilir. Daha sonra çevrimi önlemek yani arama boyunca bulunan geçmiş çözümlerin belli özelliklerini eski haline getirmemek, yerel optimum tuzağına düşmemek ve aramayı güçlendirmek amacıyla, seçilen hareket veya hareketler yasaklanmaktadır. Bunun için, ilerki bölümlerde anlatılan yeniliğe dayalı hafıza fonksiyonları kullanılarak tabu listesi (veya listeleri) güncelleştirilmektedir. Arama işlemi, yeni mevcut çözüm üzerinden devam etmekte ve bir durdurma koşulu sağlanınca sona ermektedir. Uygulamalarda genellikle aşağıdaki durdurma koşullarından biri veya birkaçı dikkate alınmaktadır (Glover ve Laguna., [30]).

- Optimum çözümü bilinen problemlerde, optimum çözümün bulunması,
- Mevcut çözümün komşusunun olmaması,
- İzin verilen maksimum iterasyon sayısına ulaşılması,
- Mevcut çözümde daha fazla iyileştirme sağlanamaması.

**Adım 1:** (Başlangıç değerlerini verme)

**Adım 1.1:** Bir başlangıç çözümü seç:  $x \in X$ .

**Adım 1.2:** Hafıza kaydını sıfırla:  $H = \emptyset$ .

**Adım 1.3:** En iyi çözümü ve maliyetini kaydet:  $x_{e\check{c}} = x$  ve  $f_{e\check{c}} = f(x_{e\check{c}})$ .

**Adım 2:** (Seçim ve durdurma)

Aday listesini oluştur.

Bu liste içinden minimum  $f(x)$  değerini veren  $x'$  çözümünü seç.

Durdurma koşulu gerçekleşmişse DUR.

**Adım 3:** (Güncelleştirme)

$x = x'$  ve  $H$  hafızasını güncelleştir.

Eğer  $f(x) < f_{e\check{c}}$  ise Adım 1.3'ü gerçekleştir. Daha sonra Adım 2'ye git.

Şekil 5.4. Kısa dönem hafızalı tabu arama algoritması

$x$  mevcut çözümünden  $x'$  çözümüne yapılan bir hareketin niteliği, bu hareketin sonucu olarak değişen herhangi bir kavramı ifade etmektedir. Niteliklerin genel tipleri aşağıdaki gibidir.

- (H1): Seçilen bir  $x_j$  değişkeninin 0 değerinden 1 değerine değişmesi,
- (H2): Seçilen bir  $x_k$  değişkeninin 1 değerinden 0 değerine değişmesi,
- (H3): (H1) ve (H2) değişkenlerinin birlikte değişmesi,
- (H4):  $f(x)$  amaç değerinden  $f(x')$  amaç değerine olan değişim,
- (H5): Tanımlanan bir  $g$  fonksiyonuna göre  $g(x)$ 'den  $g(x')$ 'e olan değişim,
- (H6):  $g(x') - g(x)$  fark değeri tarafından gösterilen değişim,
- (H7): Eş zamanlı olarak ilgilenilen birden fazla  $g$  fonksiyonu için (H5) ve (H6)'nın birlikte değişimleri.

Tek bir hareket, birden fazla niteliğin ortaya çıkmasına yol açabilir. Örneğin, iki değişkenin değerini eşzamanlı olarak değiştiren bir hareket (H1), (H2), (H3) niteliklerinin üçünü birden gerçekleştirebilir. (H5), (H6) ve (H7)'de kullanılan  $g$  fonksiyonu, stratejik olarak seçilen ve  $f$  amaç fonksiyonundan tamamen farklı bir fonksiyon olabilir. Örneğin  $g$  fonksiyonu, verilen herhangi bir çözüm ile o ana kadar

bulunan en iyi çözüm veya bulunan son yerel optimum arasındaki farkın bir ölçütü olabilir (Glover ve Laguna, [30]).

TA'da, kaydedilen hareket nitelikleri, tabu kısıtlamaları olarak adlandırılan yasakları uygulamak için kullanılır. Böylece, bu nitelikler tarafından gösterilen değişimleri tersine çevirecek hareketlerin seçilmesi engellenmiş olur. Herhangi bir iterasyonda,  $x$  çözümünden  $x'$  çözümüne yapılan ve  $e$  niteliği içeren bir hareket gerçekleştiğinde, bu niteliğin tersini ifade eden  $\bar{e}$  niteliği için bir kayıt tutulur. Böylece  $\bar{e}$  niteliğini ortaya çıkaracak başka bir hareketin yapılması yasaklanır ve bu hareket tabu olarak sınıflandırılır. Bir hareketin tabu olmasına yol açabilen kısıtlamalardan bazıları aşağıdaki gibidir (Glover ve Laguna, [30]).

- (K1):  $x_j$  değişkeninin 1 değerinden 0 değerine değişmesi (daha önce 0 değerinden 1 değerine değişmişse),
- (K2):  $x_k$  değişkeninin 0 değerinden 1 değerine değişmesi (daha önce 1 değerinden 0 değerine değişmişse),
- (K3): (K1) ve (K2) kısıtlarından en az birinin gerçekleşmesi,
- (K4): (K1) ve (K2) kısıtlarının birlikte gerçekleşmesi,
- (K5):  $g(x)$  fonksiyonunun  $v''$  değerinden  $v'$  değerine değişmesi (daha önce  $v'$  değerinden  $v''$  değerine değişmişse).

Tabu kısıtlamaları, bir hareketin tersine dönmesini, yani eski haline gelmesini engellemelerinin yanı sıra, verilen çözümden uzaklaşmayı sağlayan bir arama yönünün tekrarlanmasını da önlerler. Örneğin, verilen bir iterasyonda, mevcut  $x$  çözümünden  $Ma$  hareketi ile en iyi  $x'$  komşusu elde edilsin:  $x' = Ma(x)$ . Bir sonraki iterasyonda yeni mevcut çözüm olarak kabul edilen  $x'$  çözümünden  $Mb$  hareketi ile elde edilen en iyi çözümün  $Mb(Ma(x)) = x$  olma olasılığı çok yüksektir. Kısacası, yeni mevcut çözüm tersine dönmüş ve 2 iterasyon uzunluğunda bir çevrim oluşmuştur. TA ise,  $Mb$  hareketini (veya  $Mb$  hareketine yol açan niteliği) tabu yaparak yasaklar. Böylece, çevrimleri önleyerek aramanın farklı bir yönde devam etmesini sağlar. Tabu hareketlerin kısa dönemli hafızada kaydedildikleri tabu

listesi üzerinde kaç iterasyon boyunca kalacakları “tabu süresi” ile belirlenir. Bu süre, tabu listesinin uzunluğu olarak bilinmektedir.

Tabu listeleri, tabu durumlarının kısa dönemli hafızada kaydını tutmak için kullanılan ortak bir yöntemdir. Bunlar, yeniliğe dayanan hafıza yapılarıdır. Bu yapıda en son harekete göre hafıza güncelleştirilmekte ve hafızadaki en eski bilgi hafızadan çıkarılmaktadır. Ana amaç, yeni yapılmış hareket niteliklerine tabu durumu vermek ve kısa bir süre için bu nitelikleri, dolayısıyla bu nitelikleri içeren hareketleri tabu olarak tutmaktır (Lokketangen, [32]).

Yeniliğe dayanan hafıza fonksiyonları, tabu kısıtlamalarını sağlayan hareketlerin, tabu listesinde hangi yöntem ile tutulacağını belirlemek için kullanılır. Böylece herhangi bir iterasyonda, bir hareketin tabu olup olmadığı tespit edilir. Belli bir uygulama için kullanılan nitelikler  $e$  ile ifade edilecek olursa, yeniliğe dayanan hafıza fonksiyonlarının en genel iki örneği,  $t_{başla}(e)$  ve  $t_{bitir}(e)$  dizileridir. Bu diziler kullanılarak, eğer  $e$ ,  $i$ . iterasyonda seçilen hareketin bir niteliği ise, (5.6) eşitliği kullanılarak  $(i+1)$ . iterasyondan itibaren  $\bar{e}$  niteliğini içeren bir hareketin seçilmesi yasaklanmış olur. Bu yasak, tabu süresi  $t$  için (5.7) eşitliği gerçekleşene kadar devam eder.

$$t_{başla}(\bar{e}) = i + 1 \quad (5.6)$$

$$t_{bitir}(\bar{e}) = i + t \quad (5.7)$$

Mevcut iterasyon sayısı,  $t_{bitir}$  değerinden büyük duruma geldiğinde söz konusu niteliği içeren hareket veya hareketler için tabu durumu kalkmış olacaktır. Veri yapısı ne olursa olsun, yeniliğe dayanan hafıza fonksiyonları kullanılarak tabu durumlarının oluşturulmasında, tabu süresinin (tabu listesi uzunluğunun) iyi belirlenmesi çok önemli bir konudur. Eğer tabu süresi çok kısaysa, arama sürekli olarak yerel optimumlar etrafında gerçekleşecektir. Tabu süresi çok uzun olduğunda ise, hareketlerin çoğu tabu olacağından bulunan yerel optimumların kalitesi düşecektir (Lokketangen, [32]). Tabu süresinin belirlenmesi için kullanılan kurallar, statik ve dinamik kurallar olmak üzere sınıflandırılmaktadır. Statik kurallar,

arama boyunca tabu süresini sabit tutarken, dinamik kurallar deęişmesine izin verirler.

Tabu yıkma kriterleri, tabu olan bir hareketin yasak olmasına rağmen yapılabileceęi durumları belirtmek için kullanılır. Uygun tabu yıkma kriterlerinin kullanılması, TA performansı üzerinde oldukça etkilidir. En basit tabu yıkma kriteri, aramada mevcut iterasyona kadar bulunan en iyi çözümden daha iyi bir çözüm veren hareketin tabu olup olmamasına bakılmaksızın yapılmasıdır. Bu kriter, TA'nın ortaya konulmasından bu yana en sık kullanılan tabu yıkma kriteridir. Tabu yıkma kriterleri genel olarak, yokluk, amaç, arama yönü ve etki (bir hareketin mevcut çözümü deęiştirme derecesi) açısından sınıflandırılmaktadır (Glover ve Laguna, [33]).

Amaca göre tabu yıkma:

1. Global: Eğer bir hareket, mevcut iterasyona kadar bulunan en iyi çözümden daha iyi bir çözüm veriyorsa tabu olmasına rağmen seçilir.
2. Bölgesel: Eğer bir hareket, bulunduğu arama alanındaki en iyi çözümden daha iyi bir çözüm veriyorsa tabu olmasına rağmen seçilir.

Yokluęa göre tabu yıkma: Eğer tüm mümkün hareketler tabu ise, içlerinden tabu süresinin bitmesi en yakın olan seçilir.

Arama yönüne göre tabu yıkma: Eğer bir hareket, arama yönünü iyi veya kötü yönde deęiştirmiyorsa tabu olmasına rağmen seçilir.

Etkiye göre tabu yıkma: Eğer düşük etkili bir hareket, yüksek etkili bir hareketin gerçekleşmesine yol açacaksa tabu olmasına rağmen seçilir.

TA, mümkün ve tabu olmayan en iyi hareketin, makul zaman ve çabayla seçilmesi kuralını kullanır. En iyi hareket, amaç fonksiyonu açısından düşünöldüğünde, en iyi amaç fonksiyonu deęerini veren harekettir. Mevcut  $x$  çözümünün  $N(H,x)$  komşu kümesi geniş ve/veya bu kümedeki elemanların deęerlendirilmesinin hesaplama maliyeti yüksekse, aday liste stratejilerinin kullanımı kaçınılmaz olmaktadır. Aday listeleri, ilgilenilen problem yapısına göre veya problem yapısından bağımsız genel

stratejilere göre oluşturulurlar. Bir aday liste stratejisinin etkinliđi, hesaplama zorluđu azaltması aısından düşünmemelidir. Bunun yerine, belli bir bilgisayar zamanı içinde bulunan çözümlerin kalitesi, kullanılan aday liste stratejisi için bir etkinlik ölçütü olmalıdır. Artı hedef, elit liste, ardıl filtre genel aday liste stratejileridir. Bu stratejiler ařađıda kısaca açıklanmaktadır (Glover ve Laguna, [33]).

**Artı Hedef Stratejisi:** Bu strateji, arama boyunca elde edilen geçmiş bilgilere dayanarak, yapılacak hareketin kalitesi için bir başlangı değeri oluşturur. Bu başlangı değerini sađlayan bir hareket bulunana kadar, mevcut çözümlerden yapılabilecek hareketler araştırılır. Böyle bir hareket bulunduktan sonra, seçilen bir “artı” değerine eşit sayıda ilave hareketler incelenir ve tüm bu hareketler içinden en iyi olanı seçilir. Bu stratejide ne çok az ne de çok fazla çözümlerin incelenmesini önlemek için, incelenecek toplam hareket sayısı seçilen minimum ve maksimum değerleri arasında tutulur.

**Elit Strateji:** Elit aday liste yaklaşımı, öncelikle tüm (veya göreceli olarak fazla sayıda) hareketleri araştırarak, bulunan en iyi  $k$  tane hareketten oluşan bir ana liste hazırlar. Burada  $k$ , bir parametre değeri olmaktadır. Daha sonra, takip eden her iterasyonda, ana listedeki mevcut en iyi hareket seçilir. Listedeki hareketlerin kalitesi belli bir değerin altına düřtüğünde veya belli sayıda iterasyon tamamlandığında yeni bir ana liste oluşturularak aynı işlemler tekrarlanır. Bu strateji, herhangi bir iterasyonda seçilmemiş fakat iyi olan bir hareketin, daha sonraki bazı iterasyonlarda da iyi hareket olacağı düşüncesine dayalıdır. Ancak bu strateji için, ilgilenilen komşu yapısına bađlı olarak, aday listeden seçilen bir hareketin mevcut çözüme uygulanabilirliğine dikkat edilmeli ve bu amaçla ana listede gerekli güncelleřtirmeler yapılmalıdır.

**Ardıl Filtre Stratejisi:** Bu strateji, para operasyonlara ayrılabilen hareketler için uygulanmaktadır. Yüksek kaliteli sonuçlar verecek hareket paralarıyla ayrı ayrı ilgilenilerek, toplamda incelenmesi gereken hareket sayısı oldukça azaltılmaktadır. Örneđin, “ekleme” ve “ıkarma” paralarından oluşan ikili deđişim hareketleri için eđer 100 mümkün ekleme ve 100 mümkün ıkarma varsa, 10.000 tane ikili



değişim hareketi söz konusudur. Ancak, sadece 8 en iyi ekleme ve 8 en iyi çıkarma hareket parçasıyla ilgilenilirse, 64 tane değişim hareketinin incelenmesi yetecektir.

Bazı uygulamalarda, kısa dönemli TA hafızası oldukça yüksek kaliteli çözümlerin üretilmesi için yeterli olmaktadır (Kolahan ve Liang, [34]; İslam ve Ekşioğlu, [35]). Bununla beraber, TA genellikle, kısa dönemli hafızanın yanısıra uzun dönemli hafızanın kullanılması ve ilgili stratejiler eklenmesiyle çok daha güçlü olmaktadır (Bukard ve Çela, [36]; Nowicki ve Smutnicki, [37]; Chiang ve Kouvelis, [38]). Uzun dönemli hafıza fonksiyonları, aramanın bölgesel kuvvetlendirilmesini ve global çeşitlendirilmesini sağlamak amacıyla kullanılmaktadır. Böylece uzun dönemli hafıza fonksiyonları, tabu listeleri tarafından oluşturulan kısa dönemli hafıza fonksiyonları ile birlikte “öğrenme” ve “öğrenmeme” arasında karşılıklı bir etki oluştururlar. Uzun dönemli TA hafızasının oluşturulmasında kullanılan sıklığa dayalı yaklaşım, kuvvetlendirme ve çeşitlendirme stratejileri aşağıda ayrıntılı olarak açıklanmaktadır.

Sıklığa dayanan yaklaşım, kısa dönemli hafızadaki hareket seçiminde kullanılan prensipleri genişleterek, yeniliğe dayanan yaklaşım tarafından üretilen bilgiyi tamamlayıcı nitelikte bir bilgi türü sağlar. Sıklığa dayanan yaklaşımda, sıklıkların oranlardan oluştuğu kabul edilmektedir. Bu oranların pay değerleri, “geçicilik” ve “kalıcılık” olmak üzere iki farklı ölçüt ile ifade edilmektedir. Geçicilik ölçütü, bir niteliğin mevcut çözümü değiştirdiği (çözüme girdiği veya çözümden çıktığı) iterasyonların toplam sayısını göstermektedir. Kalıcılık ölçütü ise, bir niteliğin mevcut çözüme ait olduğu iterasyonların toplam sayısını göstermektedir. Sıklıkları oluşturan oranların paydaları ise, genellikle aşağıdaki üç değerden biri ile elde edilmektedir (Glover ve Laguna, [33]):

1. Pay tarafından gösterilen tüm olayların toplam meydana gelme sayıları
2. Payların toplamı veya ortalaması,
3. Maksimum pay değeri.

Pay değerlerinin negatif olabildiği durumlarda, (2) ve (3) için mutlak pay değerleri kullanılmaktadır. Geçicilik sıklıkları, niteliklerin hangi sıklıkta değiştiklerini ve kalıcılık sıklıkları, niteliklerin hangi sıklıkta mevcut çözümün üyesi olduklarını belirlemek için kullanılmaktadırlar. Eğer aranan çözüm alanı, yüksek kaliteli çözümlerden oluşuyorsa, büyük bir kalıcılık sıklığı, oldukça çekici bir niteliğin varlığını gösterebilir veya düşük kaliteli çözümler söz konusuysa bunun aksini gösterebilir. Büyük bir geçicilik sıklığı ise, amaç fonksiyonunun daha iyi değerler alabilmesi için, çözümlere giren ve çözümlerden çıkan bir niteliğin varlığını gösterebilir. Geçicilik ve kalıcılık sıklıkları, uzun dönemli hafızada, aramanın yönünü belirlemek için kullanılır. Bu amaçla, arama boyunca elde edilen sıklık ölçütleri, hangi hareketlerin seçileceğini belirlemek için ödül ve ceza bilgilerine dönüştürülürler.

Kuvvetlendirme stratejileri, aramanın geçmişinde bulunmuş iyi çözüm niteliklerini ve hareket kombinasyonlarını teşvik etmek amacıyla değiştirilmiş seçim kurallarını kullanırlar (Glover vd., [39]). Ayrıca, arama sırasında karşılaşılmış çekici çözüm bölgelerine yeniden dönüş yaparak, bu bölgelerin daha detaylı araştırılmasını sağlarlar. Bu son söylenen türdeki kuvvetlendirme stratejilerinin basit bir algoritması Şekil 5.5'de görülmektedir.

- Adım 1:** Kısa dönem hafızalı TA'yı çalıştır.
- Adım 2:** Elit seçim stratejisini uygula.
- Adım 3:** Repeat
- Adım 3.1:** Elit çözümlerden birisini seç.
- Adım 3.2:** Seçilen çözümü başlangıç çözümü olarak kısa dönem hafızalı TA'yı yeniden çalıştır.
- Adım 3.3:** Bulunan yeni ve uygun çözümleri elit listesine ekle.
- Adım 3.4:** *Until* (iterasyon < limit ve elit liste =  $\emptyset$ )

Şekil 5.5. Tabu aramada kuvvetlendirme yaklaşımı algoritması

Bu kuvvetlendirme stratejisinde, elit (oldukça yüksek kaliteli yerel optimum) seçim stratejisi önemli bir konudur. Elit seçim stratejisi, kuvvetlendirme stratejisi,

uygulamalarda farklı şekillerde ele alınmaktadır. Bunlardan en çok kullanılan üç tanesi aşağıda kısaca açıklanmıştır (Glover ve Laguna, [33]).

Birincisi, elit listeye kaydedilen çözümlerin istenilen bir dereceye göre birbirinden farklı olmasını sağlamak amacıyla çeşitlendirme ölçütü kullanır. Daha sonra, mevcut tüm kısa dönemli hafıza bilgileri silinir ve kaydedilen en iyi çözüm, başlangıç çözümü olarak dikkate alınıp kısa dönem hafızalı TA yeniden çalıştırılır.

İkincisi, sınırlı uzunluktaki bir elit listesi oluşturur ve yeni bir çözüm, eğer kaydedilen diğer çözümlerden daha iyiye bu listenin sonuna eklenir. Listenin mevcut son üyesi, aramayı yeniden başlatmak için kullanılır. Bununla beraber, bu başlangıç çözümü ile ilgili kaydedilmiş kısa dönemli hafıza bilgileri silinmez ve böylece yeni bir çözüm yolu oluşturulur.

Üçüncüsü ise, kısa dönem hafızalı TA'yı yeniden başlatırken, daha önce üretilen çözümlerin arama esnasında bulunmamış komşularını başlangıç çözümü olarak kullanır. Elit listeyi oluşturmak için, bu komşuların kalitesi dikkate alınır ve yerel optimumların komşuları gibi sadece belli tipteki çözümler göz önünde tutulur.

Yerel optimizasyona dayanan arama metotları, kombinatoriyal optimizasyon problemi tarafından tanımlanan çözüm uzayını araştırmadaki etkinliklerini artırmak için genellikle çeşitlendirme stratejileri kullanırlar. TA'daki çeşitlendirme, kısa dönemli hafıza fonksiyonları ile bir miktar sağlanmaktadır, fakat çeşitlendirme, özellikle uzun dönemli hafızanın belli özellikleri tarafından güçlendirilmektedir. TA çeşitlendirme stratejileri, aramayı yeni bölgelere yönlendirmek üzere tasarlanır. Bu amaçla, sık kullanılmayan nitelikleri çözüme eklemek için, genellikle değiştirilmiş seçim kuralları kullanılır. Ayrıca, sık kullanılmayan nitelikleri desteklemek için bu niteliklerin altkümelerini aday çözümlere ekleyen metotlara periyodik olarak başvurulmakta veya çözüm süreci kısmen yada tamamen yeniden başlatılmaktadır. Değiştirilmiş seçim kuralları stratejisinde, arama boyunca elde edilen geçmiş bilgilerden yararlanarak bir hareketin değeri genellikle (5.8)

eşitliğindeki ceza fonksiyonu kullanılarak değiştirilmektedir (Kim vd., [40]; Glover ve Laguna, [33]).

$$\text{Hareket değeri} = \text{Hareket değeri} + d \cdot \text{Ceza} \quad (5.8)$$

Buradaki ceza değeri genellikle sıklık ölçütlerinin bir fonksiyonudur ve  $d$  bir çeşitlendirme parametresidir. Bu parametrenin büyük değerler alması çeşitlendirmenin daha fazla olmasına yol açar. Ayrıca, negatif cezalar yani ödüller sık kullanılmayan niteliklerin desteklenmesini sağlamaktadır.

### 5.3. Değişken Komşuluk Arama

Değişken komşuluk arama, global optimizasyon ve kombinatoriyal problemlerin çözümünde kullanılan bir modern sezgiseldir. Arama yaparken yerel minimuma rastlandığında komşuluk yapısının değiştirilmesi şeklinde basit bir prensibe dayanmaktadır. Buna göre, değişken komşuluk arama algoritması Şekil 5.6'de verilmiştir.

Bu modern sezgisel 2 basit bileşene dayanır:

1. Bir komşuluk yapısında yer alan yerel minimum noktası diğer komşuluk için yerel minimum değildir.
2. Bir global minimum, bütün mümkün komşuluk yapıları açısından bakıldığında bir yerel minimumdur.

#### Notasyon

$N_k$  : komşuluk yapılarının kümesini ifade etmektedir. (  $k=1, 2, \dots, k_{max}$  )

$N_k(s)$ :  $s$ 'nin  $k$ . komşuluğuna ait çözüm kümesidir.

**Adım 1:** Bir başlangıç çözüm üret:  $X_0$

**Adım 2:** Komşuluk yapısı seç:  $\{N_k\}, k=1, 2, \dots, k_{max}$

$k=1$

**Adım 3:** Repeat

**Adım 3.1:** Seçilen komşuluk yapısıyla komşu üret:  $X_{komsu}$

**Adım 3.2:** Eğer  $f(X_{komsu}) < f(X_{mevcut})$  ise

$X_{mevcut} = X_{komsu}, k=1$

değilse  $k=k+1$

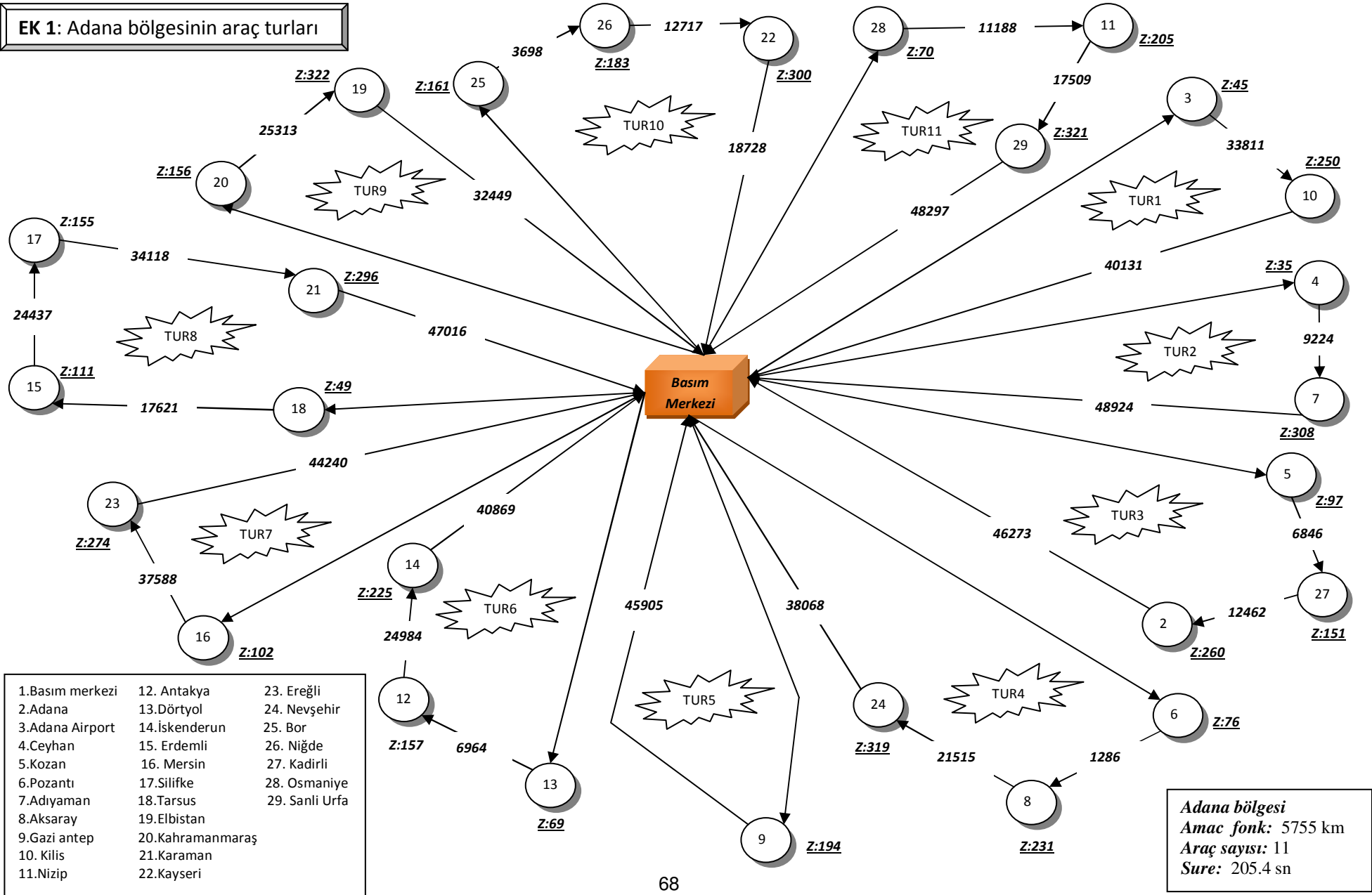
**Adım 3.3:** Until ( $k > k_{max}$ )

Şekil 5.6. Değişken komşuluk arama algoritması

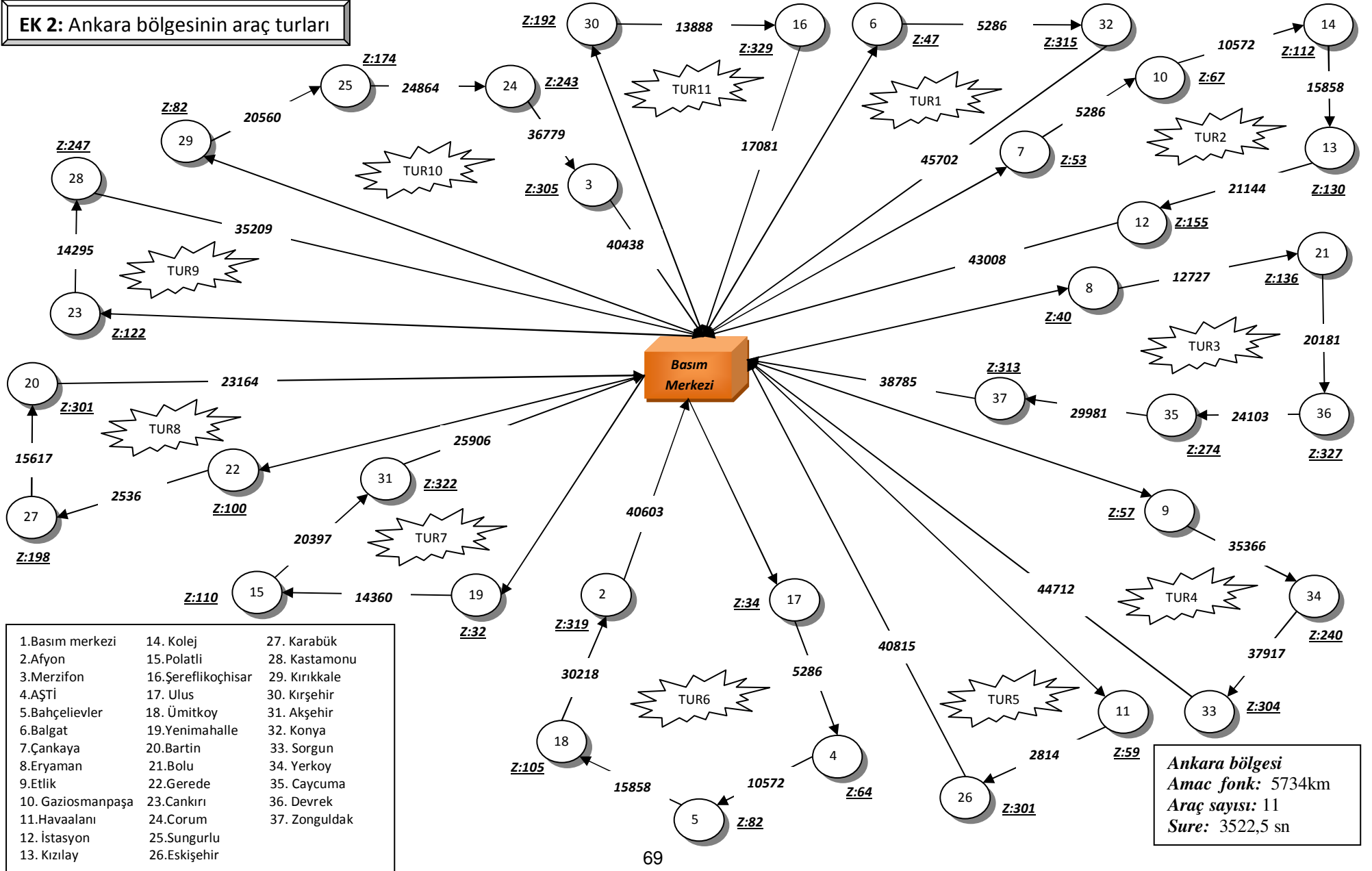
## EKLER LİSTESİ

	<u>Sayfa</u>
Ek 1 Adana bölgesinin araç turları .....	67
Ek 2 Ankara bölgesinin araç turları .....	68
Ek 3 Antalya bölgesinin araç turları .....	69
Ek 4 İstanbul bölgesinin araç turları .....	70
Ek 5 İzmir bölgesinin araç turları .....	71
Ek 6 Trabzon bölgesinin araç turları .....	72
Ek 7 Adana bölgesinin mevcut ve karar modeli ile bulunan araç turları .....	73
Ek 8 Ankara bölgesinin mevcut ve karar modeli ile bulunan araç turları .....	74
Ek 9 Antalya bölgesinin mevcut ve karar modeli ile bulunan araç turları .....	75
Ek 10 İstanbul bölgesinin mevcut ve karar modeli ile bulunan araç turları ...	76
Ek 11 İzmir bölgesinin mevcut ve karar modeli ile bulunan araç turları .....	77
Ek 12 Trabzon bölgesinin mevcut ve karar modeli ile bulunan araç turları ...	78

## EK 1: Adana bölgesinin araç turları

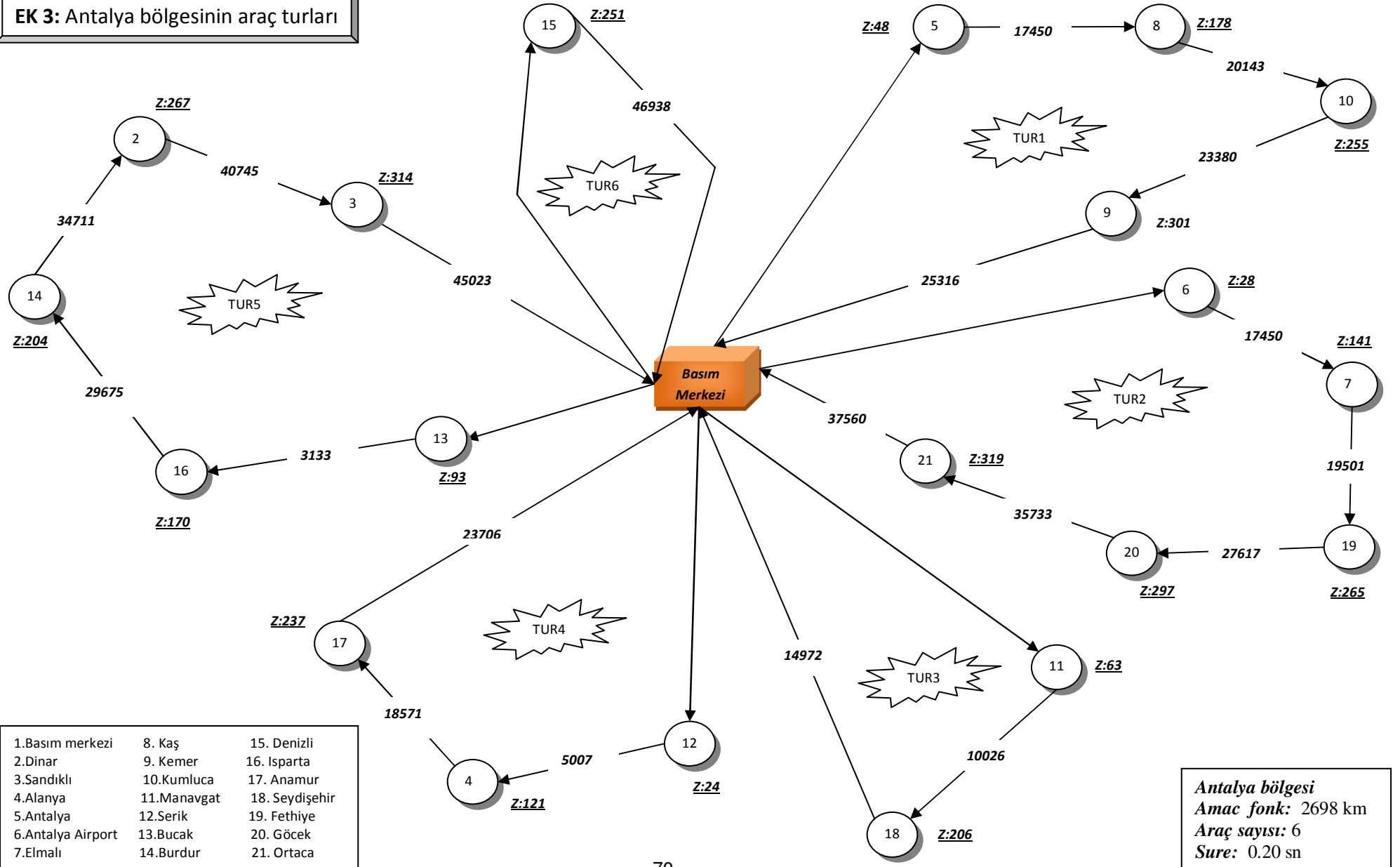


## EK 2: Ankara bölgesinin araç turları

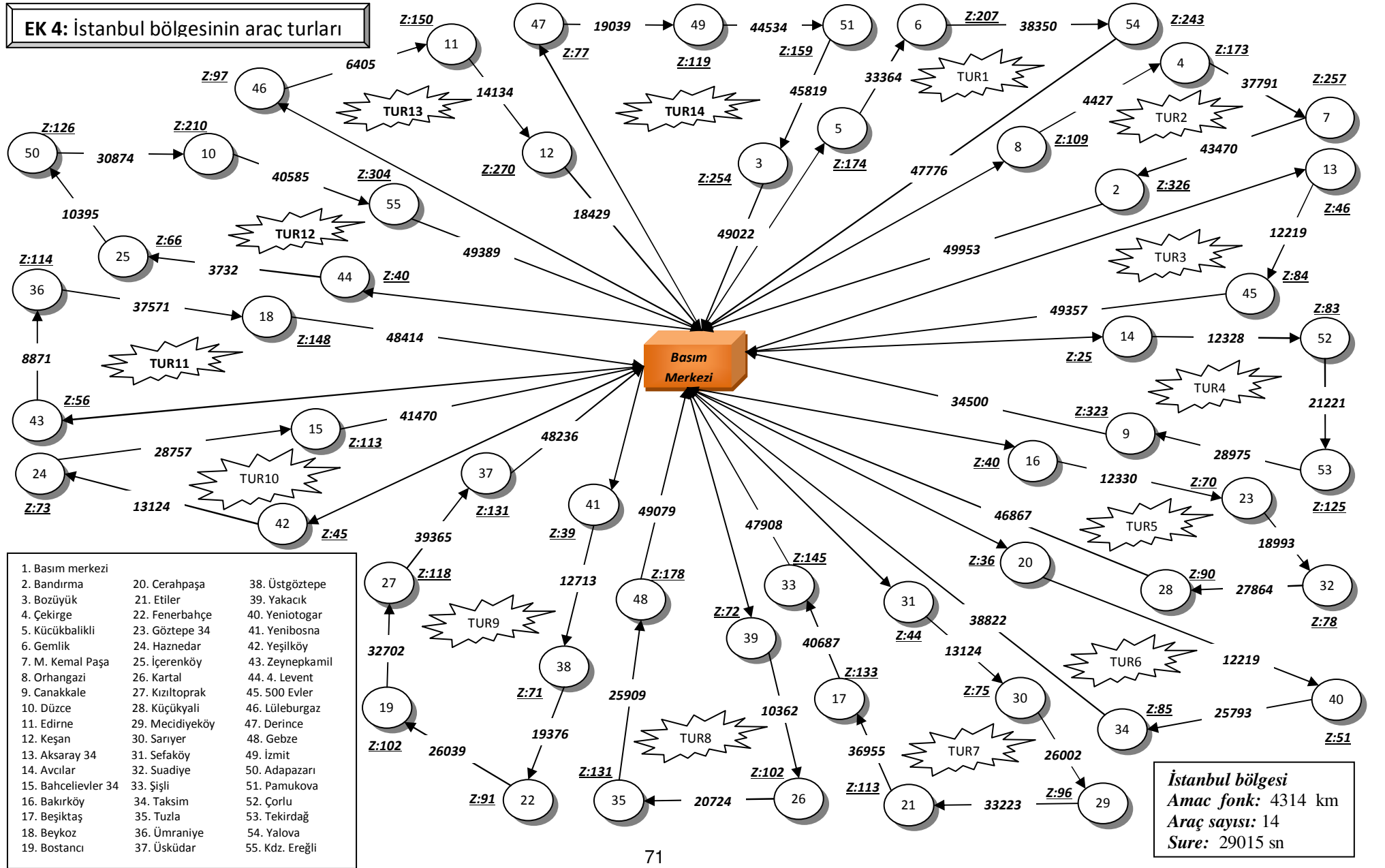




### EK 3: Antalya bölgesinin araç turları



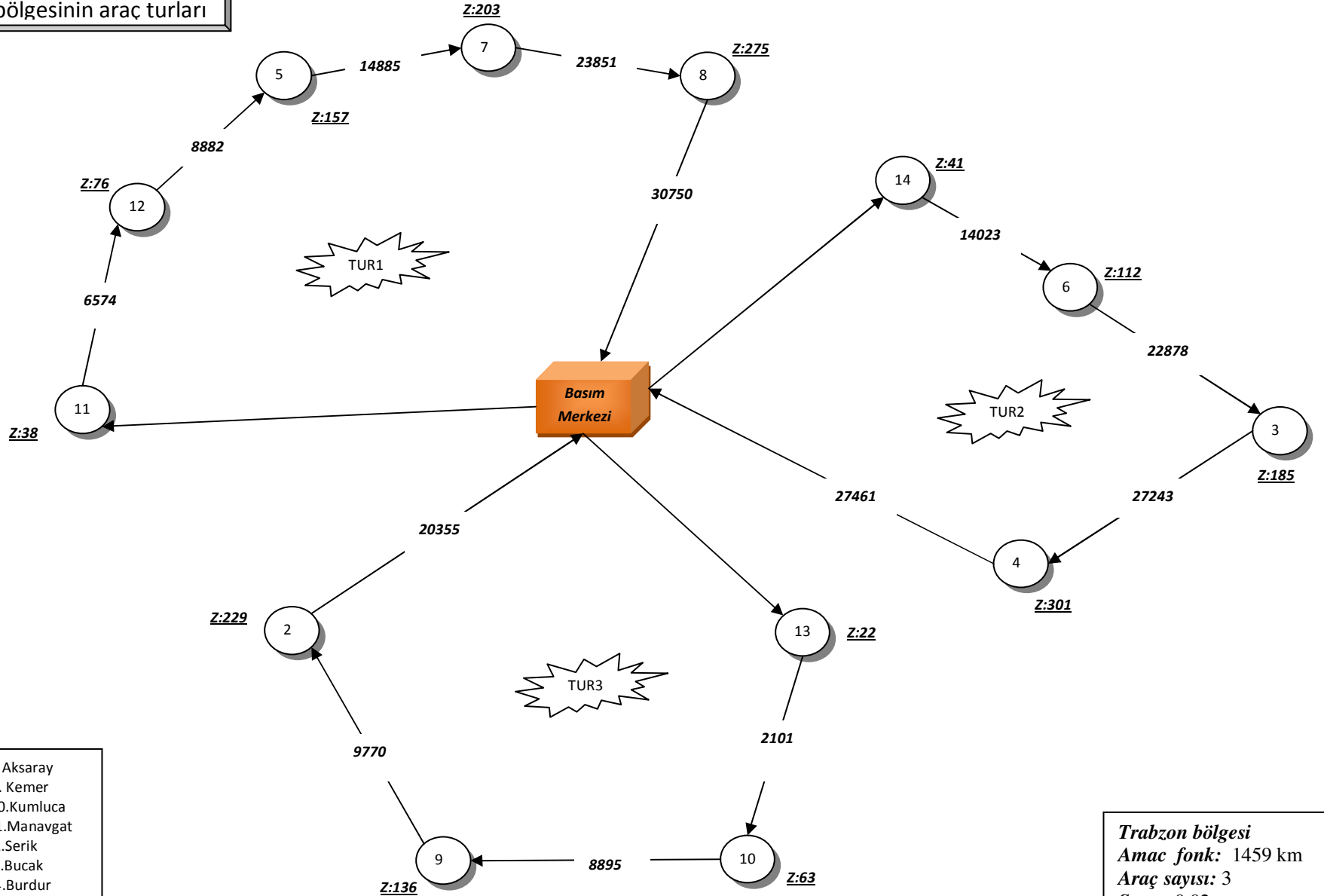
## EK 4: İstanbul bölgesinin araç turları



- |                     |                 |                 |
|---------------------|-----------------|-----------------|
| 1. Basım merkezi    | 20. Cerahpaşa   | 38. Üstgöztepe  |
| 2. Bandırma         | 21. Etiler      | 39. Yakacık     |
| 3. Bozüyük          | 22. Fenerbahçe  | 40. Yeniotogar  |
| 4. Çekirge          | 23. Göztepe 34  | 41. Yenibosna   |
| 5. Küçükbalıklı     | 24. Haznedar    | 42. Yeşilköy    |
| 6. Gemlik           | 25. İçerenköy   | 43. Zeynepkamil |
| 7. M. Kemal Paşa    | 26. Kartal      | 44. 4. Levent   |
| 8. Orhangazi        | 27. Kızıltoprak | 45. 500 Evler   |
| 9. Canakkale        | 28. Küçükyalı   | 46. Lüleburgaz  |
| 10. Düzce           | 29. Mecidiyeköy | 47. Derince     |
| 11. Edirne          | 30. Sarıyer     | 48. Gebze       |
| 12. Keşan           | 31. Sefaköy     | 49. İzmit       |
| 13. Aksaray 34      | 32. Suadiye     | 50. Adapazarı   |
| 14. Avcılar         | 33. Şişli       | 51. Pamukova    |
| 15. Bahçelievler 34 | 34. Taksim      | 52. Çorlu       |
| 16. Bakırköy        | 35. Tuzla       | 53. Tekirdağ    |
| 17. Beşiktaş        | 36. Ümraniye    | 54. Yalova      |
| 18. Beykoz          | 37. Üsküdar     | 55. Kdz. Ereğli |
| 19. Bostancı        |                 |                 |



## EK 6: Trabzon bölgesinin araç turları



- |                    |              |
|--------------------|--------------|
| 1. Basım merkezi   | 8. Aksaray   |
| 2. Dinar           | 9. Kemer     |
| 3. Sandıklı        | 10. Kumluca  |
| 4. Alanya          | 11. Manavgat |
| 5. Antalya         | 12. Serik    |
| 6. Antalya Airport | 13. Bucak    |
| 7. Elmalı          | 14. Burdur   |

**Trabzon bölgesi**  
**Amac fonk:** 1459 km  
**Araç sayısı:** 3  
**Süre:** 0.02 sn



## EK 8

### Ankara bölgesinin optimal turları



### Ankara bölgesinin mevcut turları



## EK 9

### Antalya bölgesinin optimal turları



### Antalya bölgesinin mevcut turları





## EK 10

### Istanbul bölgesinin optimal turları



### Istanbul bölgesinin mevcut turları





## EK 11

### Izmir bölgesinin optimal turları



### Izmir bölgesinin mevcut turları





