KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

GEZGİN SATICI PROBLEMİ İÇİN VERİ MADENCİLİĞİ TABANLI BİR MODEL ÖNERİSİ

ATAKAN ALKAN

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

GEZGİN SATICI PROBLEMİ İÇİN VERİ MADENCİLİĞİ TABANLI BİR MODEL ÖNERİSİ

ATAKAN ALKAN

Prof.Dr. Alpaslan FIĞLALI **Danışman, Kocaeli Üniv.**	....
Doç.Dr. Orhan ENGİN **Jüri Üyesi, Selçuk Üniv.**	....
Yrd.Doç.Dr. Gülşen AKMAN **Jüri Üyesi, Kocaeli Üniv.**	....
Prof.Dr. Zerrin ALADAĞ **Jüri Üyesi, Kocaeli Üniv.**	....
Doç.Dr. Ayhan DEMİRİZ **Jüri Üyesi, Sakarya Üniv.**	....

Tezin Savunulduğu Tarih: 27.05.2014

ÖNSÖZ ve TEŞEKKÜR

Kombinasyonel optimizasyon problemleri Endüstri Mühendisliği çalışma alanının önemli konularından biridir. Bu problem tiplerine gezgin satıcı problemi, kesme problemi, araç rotalama problemi, araç yükleme problemi, çizelgeleme problemleri, tesis yerleşimi problemi, kuadratik atama problemi gibi örnekler verilebilir. Bu problemlerden gezgin satıcı problemi temel bir problem türü olarak tanımlanmış ve üzerinde oldukça fazla çalışılan popüler bir problemdir. NP-zor sınıfında yer alan bu problemin küçük boyutlu türleri için kesin çözüm bulunabilirken, problem boyutu büyüdükçe optimal çözüme ulaşabilme zor bir hal almaktadır. Bu çalışmada, gezgin satıcı problemine kısa sürede iyi ve etkin bir çözüm bulunabilmesi amacıyla veri madenciliği yöntemine dayanan bir yaklaşım geliştirilmiştir.

Çalışmamın ortaya çıkması sürecinden başlayarak her aşamasında gösterdiği destek ve ilgiyle çalışmanın ilerlemesine büyük destek sağlayan danışman hocam Sayın Prof. Dr. Alpaslan FIĞLALI'ya,

Bu çalışmanın öncesinde ve esnasında, her ne zaman neye ihtiyacım olsa bana maddi manevi hiçbir desteğini esirgemeyen kıymetli eşim Sevgi ALKAN'a ve hayatımıza dahil olduğu andan itibaren bize kattığı büyük manevi desteği ile kıymetli kızım Yağmur ALKAN'a,

Sonsuz minnet ve teşekkürlerimi sunuyorum.

Mayıs - 2014 Atakan ALKAN

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER DİZİNİ ve KISALTMALAR

α	: Feromon izi
β	: Sezgisel bilgi geçiş olasılığı
C_{ij}	: i şehri ile j şehri arasındaki mesafe/bağ uzunluğu, (km)
k	: Veri madenciliği ile belirlenen şehir çifti sayısı
N	: Şehir sayısı
M	: En iyi tur sayısı
P	: Pozisyon matrisi
P_{ij}	: i şehrinin j pozisyonuna üyeliği
r	: Rassal sayı
S	: Başlangıç çözümü
S*	: En iyi çözüm
$\tau_{ij}(t)$	: T anında feromon iz miktarı
X_p	: Parçacık pozisyonu
V_p	: Parçacık hızı, (m/s)

Kısaltmalar

AKA	: Aşağıdaki Kısıtlar Altında
ASIPATH	: A Simple Path Mining (Basit Patika Madenciliği Algoritması)
DFJ	: Dantzig-Fulkerson-Johnson Modeli
GA	: Genetik Algoritma
GSP	: Gezgin Satıcı Problemi
KKO	: Karınca Kolonisi Optimizasyonu
KS	: Karınca Sistemi
MTZ	: Miller-Tucker-Zemlin Modeli
MYA	: Minimum Yayılan Ağaç
NP	: Non Polynomial Time (Polinomsal Zamanda Çözülemeyen)
ODBC	: Open Database Connectivity (Açık Veritabanı Bağlantısı)
OLAP	: Online Analytical Processing (Çevrimiçi Analitik İşleme)
SQL	: Structured Query Language (Yapılandırılmış Sorgu Dili)
Z	: Amaç Fonksiyonu

GEZGİN SATICI PROBLEMİ İÇİN VERİ MADENCİLİĞİ TABANLI BİR MODEL ÖNERİSİ

ÖZET

Gezgin Satıcı Problemi, belirli sayıda şehirden oluşan ve şehirler arasındaki uzaklıkların tanımlı olduğu bir şebekede, bütün şehirlere sadece bir kere uğramak ve başlanılan şehre geri dönmek şartıyla oluşturulacak en kısa turun bulunmasını amaçlayan bir problemdir. Temel bir problem türü olarak rotalama, sırlama, çizelgeleme gibi problemlerin modellenmesine ve çözümüne de temel oluşturur. Gezgin Satıcı Problemi, NP-zor grubunda yer alan bir problem olduğundan kesin çözümünün elde edilmesi problem boyutu büyüdükçe çok uzun süreler almaktadır.

Veri madenciliği büyük veri yığınları içindeki anlamlı ve kullanılabilir bilginin ortaya çıkarılmasında kullanılan bir yöntemdir. Birçok alanda oldukça fazla miktardaki verinin analizi yoluyla daha önce keşfedilmemiş bilgilerin belirlenmesine olanak sağlamaktadır.

Bu çalışmada, gezgin satıcı probleminin çözümüne yönelik veri madenciliği temelli bir yöntem önerilmiştir. Veri madenciliği tekniklerinden ardışık zamanlı örüntüler tekniği benzeri bir algortima ile rassal olarak üretilen gezgin satıcı turlarında en çok tekrar eden şehir çiftleri tespit edilmiştir. Sonrasında bu şehir çiftlerinin matamatiksel modele bir kısıt olarak eklenmesiyle çözüm kalitesini mümkün olduğunca koruyarak çözüm süresinin azaltılmasına çalışılmıştır. Önerilen yöntem literatürde yer alan veri kümelerine uygulanmış ve elde edilen sonuçlar tartışılmıştır.

Anahtar Kelimeler: Gezgin Satıcı Problemi, Optimizasyon, Veri Madenciliği.

DATA MINING BASED MODEL FOR THE TRAVELING SALESMAN PROBLEM

ABSTRACT

Traveling Salesman Problem is to find a shortest possible tour that visits each city exactly once for a given list of cities and back to the starting city. As a major problem, it provides modeling and solution of many other problems such as routing, sequencing, scheduling etc. Since the traveling salesman problem is a well-known NP-hard problem, it takes too long times solving the problem optimally when the problem size grows.

Data mining is a method that is used detecting of meaningful and usable knowledge in large datas. By analyzing considerable amounts of data, data mining enables determination of knowledges that is not previously discovered.

In this study, a data mining based algorithm is proposed for the solution of the traveling salesman problem. An algorithm based on sequence mining is used to identify the most repetitive pairs of cities in the travelling salesman tours that is generated randomly. Then these pairs of cities is added to mathematical model as constraints to decrease the solution time of the model. The proposed method is applied on literature problems and the results are discussed.

Keywords: Traveling Salesman Problem, Optimization, Data Mining.

GİRİŞ

Optimizasyon; bir karar probleminin en iyi çözümünü elde edebilmek için mevcut çözüm alternatifleri içinden seçim yapma ve karar verme olarak tanımlanabilir. Diğer bir ifade ile problemin sonucunun, ulaşılmak istenen amaca göre en uygun olduğu değeri bulma işlemidir. Mümkün çözümler kümesinin kesikli olduğu ve amacın en iyi çözümün bulunması olduğu optimizasyon problemleri ise kombinasyonel optimizasyon problemleri olarak ifade edilir. Kombinasyonel problemler denildiğinde, belirli bir kümeden bir veya daha çok amaç fonksiyonunu eniyilemek amacıyla yapılan seçimler ifade edilmektedir. Kombinasyonel optimizasyon problemleri birçok alanda karşımıza çıkan ve genelde çözümü zor olan problemlerdir.

Endüstri Mühendisliği alanında da kombinasyonel optimizasyon problemlerine sıklıkla rastlanmaktadır. Bunlar içerisinde en çok bilinenler gezgin satıcı problemi, sıralama problemleri, çizelgeleme problemleri, tesis yerleşimi problemi, kesme problemleri, yükleme problemleri, araç rotalama problemleri, kuadratik atama problemleri vb. olarak sıralanabilir.

Gezgin Satıcı Problemi (GSP), kombinasyonel optimizasyon alanında günümüze kadar üzerinde oldukça fazla çalışılmış olan problemlerden biridir. Gezgin Satıcı Problemi basitçe bir gezgin satıcının tanımlanmış bir noktalar kümesi içinde bütün noktaları dolaşarak başladığı noktaya dönmesini sağlayacak en kısa turun bulunmasının hedeflendiği bir problemdir. Problemin basit tanımına karşın, çözümünün zorluğu ve bu zorluğun problem boyutuna bağlı olarak daha da artması, birçok problemin gezgin satıcı problemi yapısına benzetilerek çözüme kavuşturulmaya çalışılması gezgin satıcı probleminin popülerliğini sürdürmesine olanak sağlamaktadır. Bu tez çalışmasında Gezgin Satıcı Probleminin kesin çözümü üzerinde çalışılmış, ancak diğer kombinasyonel problemler üzerinde de uygulama imkanı bulunan bir veri madenciliği tabanlı yaklaşım geliştirilmeye çalışılmıştır.

Veri tabanı teknolojisindeki hızlı gelişmenin doğurduğu veri sayısındaki artışa rağmen, elde edilen anlamlı bilgi sayısında azalma olmaktadır. Veri Madenciliği bu soruna cevap veren, büyük ölçekli veriler içinde saklı kalmış anlamlı bilgiye ulaşmak için, sahip olunan verileri analiz ederek, yorumlama sürecidir. Bu çalışmada, veri madenciliği tekniklerinden ardışık zamanlı örüntüler tekniğine benzer bir yaklaşım ile gezgin satıcı probleminin kesin çözümüne daha etkin bir şekilde elde etmek amacıyla bir yaklaşım önerilmiştir.

Bu tez çalışması, 3 ana bölümden oluşmaktadır. Birinci bölümde Gezgin Satıcı Probleminin tanımı, çeşitleri ve gerçek hayatta karşılaşıldığı alanlara değinilmektedir. Bölümün devamında gezgin satıcı probleminin çözümü için kullanılan sezgisel, meta-sezgisel ve kesin çözüm yöntemleri ve bu alanda yapılan çalışmalar özetlenmektedir.

İkinci bölümde, veri madenciliğinin tanımına, tarihsel gelişimine, tercih edilme nedenlerine, veri madenciliği uygulama alanlarına, sürecine ve tekniklerine değinilmektedir.

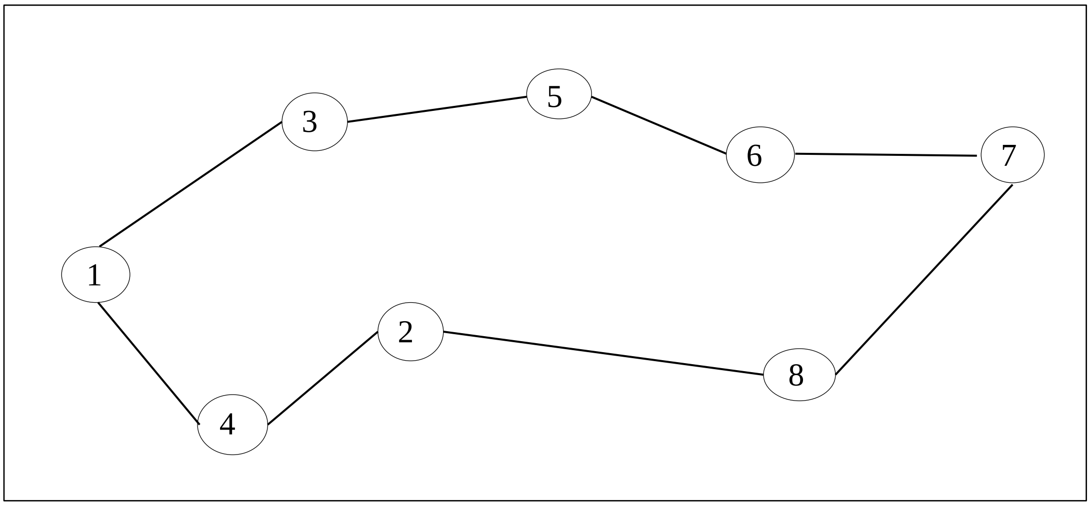
Üçüncü bölümde, gezgin satıcı probleminin kesin çözümüne yönelik olarak önerilen veri madenciliği tekniklerinden ardışık zamanlı örüntüler tekniğine benzer bir yaklaşımın adımları detaylı olarak anlatılmıştır. Yine bu bölümde önerilen yaklaşımın adımlarında kullanılacak olan gerekli parametre değerlerinin nasıl belirleneceğini ve kullanılacağını belirlemek amacıyla tam faktöriyel deney tasarımı gerçekleştirilmiştir. Deney tasarımında elde edilen sonuçlara göre de önerilen yaklaşımın literatür problemlerine uygulanması sonucu elde edilen sonuçlar ortaya konulmuştur.

Son bölümde ise, elde edilen sonuçlar yorumlanmış, sonraki çalışmalar için olası gelişme alanlarına değinilmiştir.

1. GEZGİN SATICI PROBLEMİ VE ÇÖZÜM YÖNTEMLERİ

Gezgin Satıcı Problemi, belirli sayıda şehirden oluşan ve şehirler arasındaki uzaklıkların tanımlı olduğu bir şebekede, bütün şehirlere sadece bir kere uğramak ve başlanılan şehre geri dönmek şartıyla oluşturulacak en kısa turun bulunmasının hedeflendiği bir problem olarak tanımlanabilir. Örnek bir gezgin satıcı turu Şekil 1.1'de görüldüğü gibidir.

Kombinasyonel optimizasyon problemleri arasında en çok bilinen problemlerden biri olan Gezgin Satıcı Problemi, yapısı nedeniyle ağ teorisinin de kapsamına girmektedir. Ağ teorisi kısaca, belirli sayıda düğümün ve bu düğümleri birbirine bağlayan bağların olduğu bir yapıyı inceleyen yaklaşımdır. Gezgin satıcı problemi ise ağ teorisinde; düğümler şehirleri, bağlar şehirler arasındaki yolları ve bağların ağırlıkları ise şehirler arasındaki mesafeyi göstermek üzere ağdaki en kısa Hamilton döngüsünün bulunması olarak tanımlanabilir. Hamilton döngüsü, bir düğümden başlayarak belirli sayıdaki düğüme sadece bir defa uğrayarak, başlangıç düğümüne dönen bağlar tarafından oluşturulan bir yoldur.



Şekil 1.1. Örnek gezgin satıcı turu

Gezgin Satıcı Problemi gerçek yaşamda, malzeme akış sistemi tasarımı, devre kartlarında delikler açılması, robotların depolar ve üretim istasyonları arası yönlendirilmesi, uçaklar için havaalanı rotalaması, büyük ölçekli devrelerin tasarlanması, tek makinede ürün değişiminde ayar süreleri en küçüklenmesi, lojistikte çeşitli dağıtım problemlerinin tamamı ya da bir bölümü olarak karşımıza çıkabilmektedir. Sıralanan alanların dışında literatürde [1] birçok örnek yer almaktadır.

Gezgin Satıcı Problemi, NP-zor sınıfında yer alan bir problemidir. Çözülmesi zor olan kombinasyonel optimizasyon problemlerinde, çözüm uzayının tamamının taranmasını gerektiren geleneksel çözüm yöntemlerinde problem çözümü değişken sayısının artması ile imkansız hale gelebilmektedir. Bu nedenle problemlerin kısa zamanda değişik yaklaşımlarla çözümü düşünülmüş, çözüm uzayının yalnızca belirli bir kısmının tarandığı sezgisel yöntemler denenmiştir. Çözüm uzayının tümü gözden geçirilmediği için sezgisel yöntemlerle optimum sonuca ulaşma garantisi yoktur.

Gezgin Satıcı Problemi için geliştirilmiş ya da GSP üzerinde uygulanan birçok çözüm yöntemi bulunmaktadır. Bu yöntemlerin bazıları analitik süreçler yardımıyla, en kısa tura ulaşmayı garantilemeye çalışmaktadır. Tam sayma, dinamik programlama, dal ve sınır, dal ve kes gibi yöntemler bu grupta yer almaktadır. Bu tür yöntemler en kısa turun kesinliği karşılığında uzun süreler alabilmektedir. Süre gereksinimini en aza indirebilmek için bazı yaklaşımlar geliştirilmiştir.

Çözüm yöntemlerinin bazıları özellikle GSP için geliştirilmiştir. Oldukça kısa sürede çözüm elde edilmesine imkan veren bu tür çalışmalardan elde edilen bilgilerin diğer kombinasyonel eniyileme problemleri için kullanılması oldukça zordur. Bunun için mevcut problemin, Gezgin Satıcı Problemi formuna dönüştürülmesi gerekmektedir. Sezgisel en iyi tur oluşturma ve geliştirme yöntemleri bu gruba girmektedir.

GSP problemleri için öne çıkan diğer bir grup çözüm yöntemi meta sezgisellerdir. Genetik Algoritmalar, Karınca Kolonileri, Tabu Arama, Tavlama Benzetimi, Sinir Ağları gibi meta sezgisel yöntemler GSP için kullanıldığı gibi diğer kombinasyonel problemler için de kullanılmakta ve başarılı sonuçlar elde edilmektedir. Bu yöntemler küresel en iyiyi garanti edememekle birlikte kısa sürede tatmin edici derecede iyi çözümler vermeleri nedeniyle sıklıkla kullanılmaktadırlar [2].

1.1. Gezgin Satıcı Problemi Çeşitleri

Gezgin Satıcı Problemi için birçok sınıflandırma bulunmaktadır. En çok bilineni, bir noktadan diğer bir noktaya ulaşım mesafesinin, gidiş ve dönüş yönünde eşit olduğu “Simetrik Gezgin Satıcı” problemidir. Gidiş dönüş ulaşım mesafesinin farklı olması durumunda ise “Asimetrik Gezgin Satıcı” problemi olarak tanımlanır. Asimetrik Gezgin Satıcı problemlerinde satıcının hareket yönü önem kazanır. Blaser ve diğ. ile Kwon ve diğ. yaptığı çalışmalar bu problem türünün çözümü için değişik çözüm örnekleridir [3, 4].

Gezgin Alıcı Probleminde, ziyaret edilmesi gereken şehirlerin yanında, satın alınması gereken bir malzeme listesi de söz konusudur. Malzemeler farklı şehirlerde değişik fiyatlarla satılmaktadır. Amaç toplam yol ve satın alma maliyetini minimize etmektir. Pearn ve diğ. yaptıkları çalışmada mevcut turun geliştirilmesi içi bazı öneriler sunulmaktadır [5].

Çoklu GSP’de, bir ana depo ve bu depodan belli güzergahlara dağıtılması gereken ürünler söz konusudur. Dağıtım belli sayıda, problemin yapısına göre sınırlı ya da sınırsız kapasiteye sahip, araçlar tarafından gerçekleştirilmektedir. Her bir aracın gerçekleştireceği tur bir gezgin satıcı problemidir. Ancak araç turlarının birbirini etkilemesi nedeniyle farklı bir yapıya sahiptir. Bektaş, bu tip problemin çözümü için kullanılan yaklaşımları özetlemektedir [6].

Çinli Postacı probleminde, postacı mevcut ağdaki bütün yolları kullanarak ve bütün yollara bir kez uğrayarak en kısa uzunluktaki turu tamamlamak istemektedir [7]. Ağ teorisinde bu tur Euler turu olarak adlandırılır. Bu problem türü problemin şebekesinde tanımlı tüm yollara sadece bir kez uğramayı gerektirerek, gezgin satıcı probleminden farklılaşır.

Zamana bağlı gezgin satıcı problemi, Ödül Toplayan gezgin satıcı problemi, Gezgin Vaiz Problemi, En Büyük gezgin satıcı problemi, On-line gezgin satıcı problemi, Öncelik kısıtlı gezgin satıcı problemi, Minimum gecikmeli gezgin satıcı problemi, Darboğaz gezgin satıcı problemi, Genelleştirilmiş gezgin satıcı problemi [8], v.b birçok gezgin satıcı problemi çeşidi bulunmaktadır.

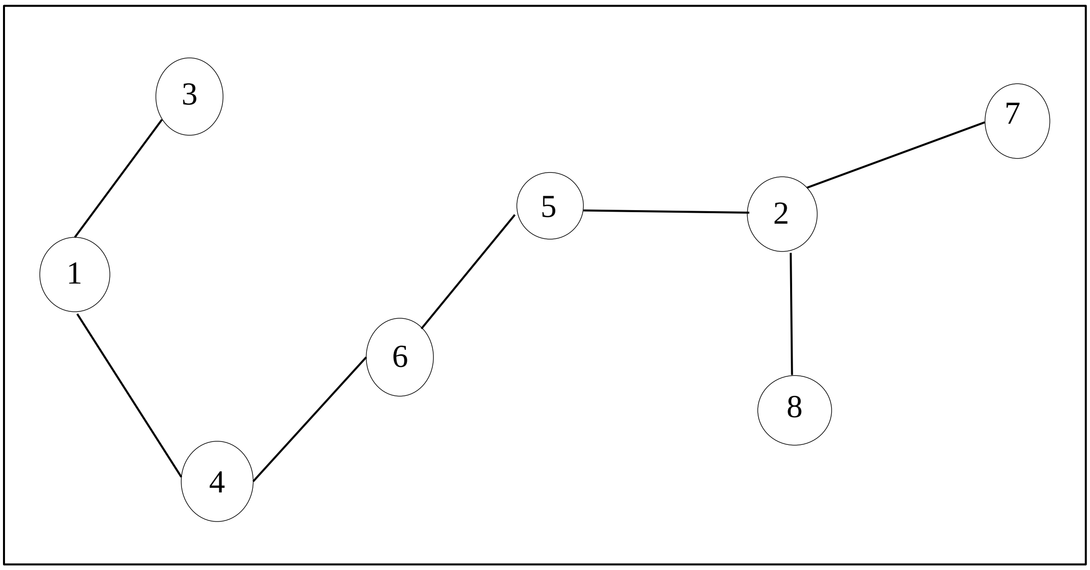
Yukarıda sayılan problemlerin yanında ufak tefek değişikliklerle tanımlanabilecek birçok çeşit mevcuttur. Şehirlerin bir defa ya da daha çok sayıda ziyaretine izin verilmesi; gidiş dönüş uzaklıklarının farklı ya da aynı olması; aynı yolun birden çok kez kullanılması ya da kullanılmaması; bütün yolların kullanılması ya da kullanılmaması; satıcının uğradığı şehirlerde bazı görevleri yerine getirmek zorunda olup olmaması; satıcı için taşıma kapasitesinin olup olmadığı; birden çok satıcı olup olmadığı, başlangıç ve varış noktalarının aynı olup olamayacağı; bağlar ya da noktalarda olasılıklı yapının bulunup bulunmadığı; şehirlere ya da uzaklıklara ait bilgilerin zamanla değişip değişmediği; birden çok amacın bulunup bulunmadığı, gibi değişikliklerle onlarca değişik tipte gezgin satıcı problemi tanımlanabilir [2].

Bu çalışma kapsamında, gezgin satıcı probleminin kesin çözümüne yönelik olarak önerilen yaklaşımın etkinliğinin gösteriminde yukarıda tanımlanmış olan simetrik gezgin satıcı problemi üzerinde çalışılmıştır.

GSP için çözümde farklılık yaratan bir başka nokta ise şehirler arasındaki uzaklıkları gösteren matrisin yapısıdır. Uzaklık matrisi bir düzlem üzerinde yer alan noktalara ait koordinatlar arasındaki doğrusal uzaklıklarının hesaplanmasıyla oluşturulmuş ise problemin düzlemsel olduğu söylenir. Bu tür problemlerde önemli olan nokta, şehirler arası yolların doğrusal olmasıdır. Bu tip problemler için grafik çözümler önerilebilmektedir. Bir plaka üzerinde matkapla delme problemi düzlemsel problemler için iyi bir örnek olarak gösterilebilir. Ancak gerçek yaşam problemlerinin birçoğu düzlemsel değildir. Düzlemsel yapıdaki iki boyutun yanına, şehirler arası rakım farklarında olduğu gibi, üçüncü bir boyutun katılması da mümkündür. Bu iki tip problemde de doğrusallığın kontrolü için üçgen eşitsizliği kullanılır. Üçgen eşitsizliği basit olarak, bir üçgendeki iki kenarın uzunlukları toplamının, üçüncü kenar uzunluğundan büyük olması anlamına gelmektedir. Bu eşitsizliğin sağlanmaması durumunda üçüncü kenarın doğrusal olmayan bir yapı gösterdiği anlaşılır. Gerçek yaşam problemlerine daha yakın olan bu tür problemlerin çözümü nispeten daha zordur. Bu nedenle geliştirilmiş olan yöntemlerin bazılarında, düzlemsellik ya da üçgen eşitsizliğini sağlama varsayımı yapılmaktadır [2].

1.2. Sezgisel Çözüm Yöntemleri

Literatüre bakıldığında birçok gezgin satıcı problemi için geliştirilmiş birçok sezgisel çözüm yöntemi yer almaktadır. Bunun nedeni sezgisellerin, analitik yöntemlerin aksine polinomiyal olmayan bu tür problemler için istendiği gibi polinomiyal zamanda çözüm önermeleridir. Gezgin satıcı problemine benzer bir özellik gösteren "Minimum Yayılan Ağaç (MYA) Problemi" sezgisel bir yöntem kullanılarak küresel en iyi çözüme ulaştırılabilmektedir. Bir ağ sistemindeki bütün noktalara ulaşan en kısa yolun bulunması olarak tanımlanabilecek ve Şekil 1.2'de görülen bir problem Kruskal ya da Prim tarafından önerilen basit sezgisellerden herhangi biri ile küresel en iyi çözüme ulaştırılmaktadır [9]. Bu problemin, çözüm uzayında gezgin satıcı probleminden çok daha fazla sayıda alternatif çözüm bulundurmasına karşın bu denli basit şekilde çözüme ulaştırılması GSP ve benzeri problemler için, polinomiyal zamanda çözüm üreten bir çözümün olabileceği umudunun oluşmasına neden olmaktadır. Bunun yanında sezgisel çözümler küresel en iyi çözümü garanti etmeseler de, polinomiyal zamanda kabul edilebilir derecede en iyi çözüm değerine yaklaşabildikleri durumda, zaman kısıtının bulunduğu uygulamalar için vazgeçilmez hale gelmektedirler.



Şekil 1.2. Minimum yayılan ağaç

Gezgin satıcı problemi için literatürde birçok sezgisel yöntem önerilmiştir. Bunlar arasından en çok bilinen birkaç yönteme burada yer verilmiştir. GSP için geliştirilen sezgisel yöntemler tur oluşturma yöntemleri ve tur geliştirme yöntemleri olarak iki grupta toplanabilir. Adından da anlaşılacağı gibi ilk grup, bazı kriterleri göz önüne alarak sırayla noktaları birbirine bağlamakta ve en iyi çözümü oluşturmaya çalışmaktadır. Aşağıda bahsedilen en yakın komşu, en yakın ekleme, greedy ve Christofides gibi yöntemler bu gruba girmektedir. İkinci grup ise mevcut bir çözümün çeşitli hareketler yoluyla geliştirilmesinin hedeflendiği yöntemlerdir. Aşağıda anlatılan 2-opt, 3-opt gibi r-opt denemeleri ve Lin-Kerninghan yöntemleri bu gruba girmektedir. Bu iki grubun özelliklerini bir arada bulunduran birleşik yöntemler de mevcuttur.

1.2.1. En yakın komşu yöntemi

En yakın komşu yönteminde, var olan şehirler içerisinden rasgele bir şehir gezgin satıcı turunun başlangıcı olarak seçilir ve bu şehre en yakın şehir turdaki sonraki adım olarak belirlenir. Ekleme işlemine bütün şehirler tura katılıncaya kadar devam edilir ve tüm şehirler tura eklenmeden önce turun kapanmasına izin verilmez. Bu yöntemde, hangi şehrin başlangıç şehri olarak seçildiği ve mevcut yolun hangi ucundan devam edileceği kararı farklı uzunlukta tur sonuçları elde edilebilmesine neden olabilmektedir. Ancak genellikle en yakındaki şehrin seçimi sırasında geride bırakılan diğer şehirlerin tura katılması için uzun geri dönüşler gerektiğinden küresel en iyiden uzak çözümler elde edildiği görülmektedir.

1.2.2. En yakın ekleme yöntemi

İki şehir arasındaki bir tur ile çözüme başlanır. Daha sonra bu tura en yakındaki şehirlerin, sırayla mevcut turdaki şehirlerin arasına girerek tura eklenme durumları denenerek, turdaki mümkün olan en küçük büyüme bulunur ve bu şehir tura eklenir. Bu işlem bütün şehirler tura ekleninceye kadar devam ettirilir. Bu yöntemde de başlangıç şehrinin seçimi ve ekleme sırası kuralı çözümü değiştirebilmektedir.

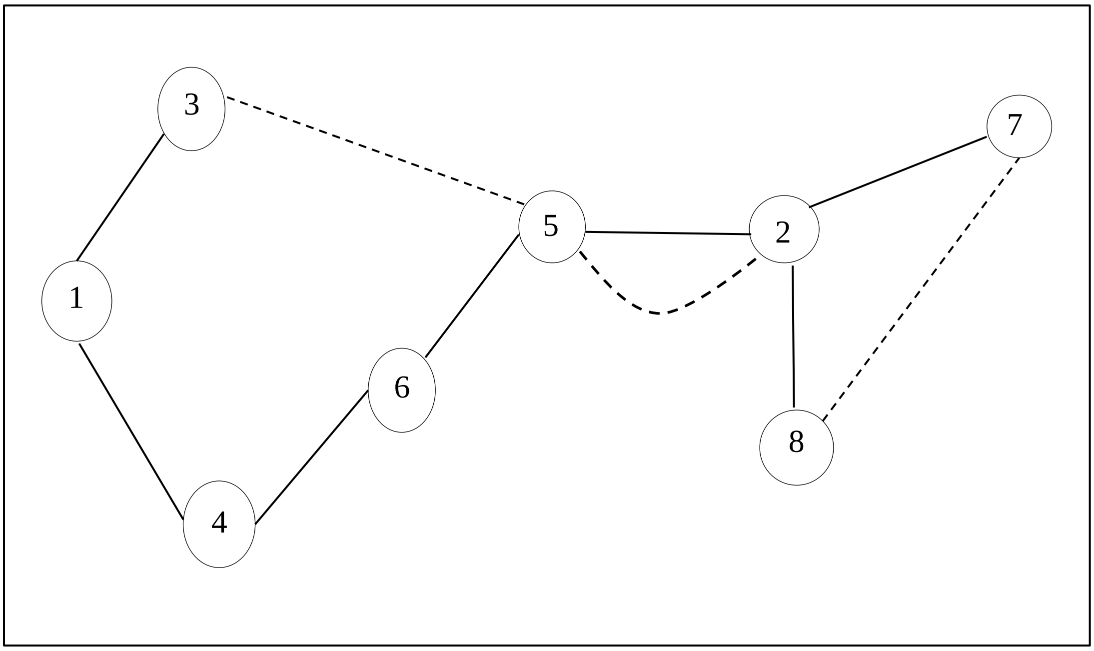
En yakın ekleme ve en yakın komşu yöntemleri genellikle başlangıç çözüm elde edilmesi amacıyla kullanılır. Ancak DePuy ve diğ. bu iki yöntemden yararlanarak GSP ve benzeri kombinasyonel problemlerin çözümü için kullanılabilecek bir meta sezgisel önermişlerdir [10].

1.2.3. Greedy yöntemi

Olası bütün bağların küçükten büyüğe doğru sıralanması ile yönteme başlanır. Daha sonra derecesi zaten iki olan noktalara tekrar bağlanmaksızın ve bağ sayısının şehir sayısına eşit olmasından önce turun kapatılması önlenerek, en küçükten başlayarak, sırayla bağların eklenmesi yoluyla bir tur oluşturulur [8].

1.2.4. Christofides algoritması

Algoritma üçgen eşitsizliğini sağlayan GSP için Minimum Yayılan Ağaç çözümünü temel almaktadır. Minimum yayılan ağaç çözümünde derecesi tek olan noktalar arasında, her noktaya sadece tek bağın yapıldığı bir mükemmel eşleşme ağı oluşturulur. Bu ağ ile başlangıçtaki minimum yayılan ağaç çözümü birleştirilerek M ağı Şekil 1.3'deki gibi oluşturulur. M ağı içerisinden Euler turu oluşturulur ve tur ziyaret edilen noktanın tekrar ziyaretini engelleyecek biçimde Şekil 1.1.'de görüldüğü gibi Hamilton turuna dönüştürülerek çözüm sonlandırılır. Algoritmanın en kötü performans durumunda bile en kısa turdan 1,5 kat uzaklaşabileceği ispatlanmıştır [11, 12].



Şekil 1.3. Christofides algoritmasında M ağı

1.2.5. R-opt denemeleri

R-opt denemeleri mevcut bir turun geliştirilmesi için kullanılabilecek bir yöntemdir. Yöntemde “r” ile belirtilen sayıda bağ çıkarılıp, yerine tur yapısını bozmayacak ve daha kısa tur oluşturacak şekilde yeni “r” adet bağ eklenir. Denemelere gelişme oluşturacak hareket bulunamadığı noktaya kadar devam edilir. Kısaltma yapan değişimi elde etmek için birçok deneme yapılması gerekebilmektedir. R sayısı büyüdükçe yapılması olası deneme sayısı da üssel olarak artmaktadır, ancak en iyiyi bulma olasılığı da artmaktadır. En iyinin bulunması ile süre arasında bir ödünleşim söz konusu olduğundan genellikle 2-opt ve 3-opt kullanılmaktadır.

1.2.6. Lin-kerninghan yöntemi

Lin-Kerninghan yöntemi simetrik gezgin satıcı problemi için küresel en iyi ya da ona yakın çözümler üretmek için kullanılan en iyi yöntemlerden biri olarak bilinir [13, 14]. Algoritma, r-opt yönteminin kullanımına dayanmaktadır. Yöntem, çalışması sırasında gereksinime göre “r” değerini değiştirmektedir. Her iterasyonda “r” değerinin arttırılmasının çözümün daha da kısalmasına neden olup olmayacağı test edilir. Bu işlemlere bir durdurma koşulu sağlanıncaya kadar devam edilir.

Algoritma temel olarak bazı değişimlerle bir turdan daha kısa yeni bir tur elde edilmesini sağlamak üzere, geliştirme yapamadığı noktaya kadar devam eder. Bu denemeler rasgele oluşturulmuş birçok tur ile tekrarlanabilir. Yöntem başlangıç turuna bakmaksızın kendi başına oldukça başarılıdır.

1.3. Meta Sezgisel Yöntemler

Meta sezgisel yöntemler, karıncalar, genetik seçilim, sinir ağları gibi doğadaki bazı başarılı yapıların çalışma prensiplerinden esinlenerek geliştirilmiş çözüm yaklaşımlarıdır. Sezgisel yöntemlerle karşılaştırıldığında, daha geniş bir çözüm uzayını tarayabilen, farklı problemler üzerine uygulanabilen meta sezgiseller, özellikle konveks olmayan, türevlenemeyen, klasik çözüm yöntemleriyle çok uzun sürede çözülebilen ya da modelleme zorluklarının karşılaşıldığı problemlerde sıklıkla kullanılmaktadırlar.

Meta sezgisel yöntemler başlığı altında literatürde birçok yöntem mevcuttur ve zamanla yenileri eklenmektedir. Bu bölümde, meta sezgisel yöntemler denildiğinde akla ilk gelen, üzerinde en çok çalışma yapılmış olan Karınca Kolonileri, Genetik Algoritmalar, Tabu Arama, Tavlama Benzetimi, Sinir Ağları ve Parçacık Sürü Optimizasyonu yöntemleri kısaca anlatılmış ve gezgin satıcı problemi uygulamalarına değinilmiştir.

1.3.1. Karınca algoritması

Karınca Kolonisi Optimizasyonu (KKO), gerçek karıncaların yuvaları ile yiyecek noktaları arasındaki en kısa yolu bulma yeteneklerinden esinlenilerek geliştirilmiş, matematiksel olarak karınca koloni davranışlarının modellenmesine dayalı populasyon temelli bir yaklaşımdır. Karınca kolonilerinde karıncalar yiyeceklerini ararken, öncelikle yuvalarının etrafında rasgele dolaşarak keşfe başlarlar. Yiyecek kaynaklarını bulduklarında, kalite ve miktarını değerlendirdikten sonra bir kısmını yuvaya taşırlar. Karıncaların alternatif yolların söz konusu olduğu durumlarda, öncelikle bu yollara eşit olasılıkla dağılırlarken belli bir süre sonra en kısa olan yolda yoğunlaştıkları, en sonunda tümünün en kısa olan yolu kullandıkları gözlemlenmiştir. Kısa yolun seçilebilmesi, aynı kaynağın tekrar bulunabilmesi için karıncaların yiyecek kaynağından yuvaya dönüş sırasında yiyeceğin kalitesine ve miktarına bağlı olarak kimyasal feromon maddesini geçtikleri yolun üzerine bırakmaları sayesinde mümkün olmaktadır. Yiyecek arama esnasında karıncalar gidecekleri yolun seçiminde, feromon miktarını göz önüne almaktadırlar. Feromon bulunması durumunda rasgele seçim yapılırken, feromon yoğunluğuyla orantılı olarak feromonun yer aldığı yolun seçim olasılığı artmaktadır [2].

Karınca kolonileri ile ilgili ilk olarak M. Dorigo tarafından sunulan Karınca Sistemi (KS) Algoritması, gerçek karıncaların en kısa yolu bulmasından hareketle gezgin satıcı problemi üzerinde uygulanmıştır [15, 16]. Stützle ve Dorigo karınca kolonileriyle GSP'nin çözümünde geliştirilen karınca sistemi, karınca kolonisi sistemi ve minimum-maksimum karınca sistemi algoritmalarını özetlemişlerdir [17]. Buna göre GSP'nin çözümünde kullanılan temel bir karınca sistemi algoritmasının adımları Şekil 1.4'de görüldüğü gibidir.

```
BAŞLA
  Parametreleri belirle
  Feromon izlerini başlat
      WHILE (Durdurma kriteri==0)
          Karınca Turları Üret
          Yerel Arama Uygula (İsteğe Bağlı)
          İzleri Güncelle
      END
  BİTİR
```

Şekil 1.4. GSP için temel karınca sistemi algoritmasının adımları

Karınca sistemi ile GSP'nin çözümünde, ilk olarak m adet karınca rasgele seçilmiş başlangıç şehirlerine yerleştirilmektedir. Daha sonra her şehir için iteratif olarak bir durum değişim kuralı uygulanarak bir şehirden diğerine hareket edilmekte ve gezgin satıcı turları oluşturmaktadırlar. Bir i şehrinde yer alan k karıncasının j şehrine gitme olasılığı, bu iki şehir arasında t anında bulunan feromon iz miktarı $\tau_{ij}(t)$ ile iki şehir arasındaki uzaklığın fonksiyonu olan sezgisel bilgi kullanılarak (1.1) denkleminde görüldüğü hesaplanır. Denklemde α ve β sırasıyla feromon izi ve sezgisel bilginin geçiş olasılığındaki etkilerini göstermektedir. Karıncalar daha kısa mesafede bulunan ve feromon izi daha güçlü olan şehirleri sonraki şehir olarak seçme eğiliminde olacaklardır. Bir karınca o ana kadar geçmiş olduğu şehirleri tabu listesi adı verilen hafızasında tutmaktadır. Böylece daha önce gidilen şehirlerin seçilmesi önlenerek olursuz turların önüne geçilmekte ve tur tamamlandığında feromon güncellemesi yapılacak bağların belirlenmesi için bu bilgi kullanılabilmektedir.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} . [\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} . [\eta_{il}(t)]^{\beta}} \quad , \qquad j \in N_i^k \tag{1.1}$$

Karıncaların turlarının tamamlamasının ardından, feromon bilgisi güncellenmektedir. Bu işlem (1.2) denkleminde görüldüğü gibi öncelikle (i, j) bağı üzerinde mevcut feromon miktarının buharlaşma sabiti ρ yardımıyla azaltılması ve sonrasında bu bağ üzerinden geçen turların uzunluklarının ($L^{k}(t)$) tersiyle orantılı olarak oluşturulan feromon miktarının (1.3) eklenmesiyle gerçekleştirilir. Böylece üzerinden daha kısa turlar geçen ve daha çok karıncanın geçtiği bağlara daha fazla feromon bırakılarak, sonraki jenerasyonlarda seçilme olasılığı artmaktadır. Ayrıca feromon buharlaşması yardımıyla eski kötü sonuçların feromon hafızasından silinmesi mümkün olmakta böylece yerel en iyi çözümlere sıkışmanın önüne geçilebilmektedir [2].

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{i=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{1.2}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & \text{eğer bağ (i, j) karınca } k \text{ tarafından kullanılmışsa} \\ 0 & \text{aksi takdirde} \end{cases} \tag{1.3}$$

Dorigo bu algoritmayı değişik boyutlardaki birçok GSP probleminde denemiş, küçük ölçekli GSP'lerde (75 şehirden az) başarılı olurken daha büyük ölçeklilerde başarılı olmadığı görülmüştür [18]. Karınca sistemi algoritmasının geliştirilmesi için elitist strateji adı verilen bir yaklaşım geliştirilmiş ve çözümde o ana kadar bulunmuş en iyi tura ait bağlara daha fazla feromon bırakılması sağlanarak çözümler hız ve kalite açısından geliştirilmiştir.

GSP üzerinde çalışılan diğer bir karınca yöntemi, Karınca Koloni Sistemi, karınca sistem yaklaşımını temel alarak eksikliklerini gidermek için geliştirilmiştir [19-21]. Karınca Koloni Sisteminin, karınca sisteminden üç temel farkı bulunmaktadır. Karınca Koloni Sisteminde daha hızlı ilerlemenin sağlanması için sadece o ana kadar bulunmuş en iyi çözüme ait bağlarda feromon güncellenmesine izin verilmektedir. Bir karınca i şehrinden j şehrine ilerlediğinde, diğer karıncaların da aynı yolu takip ederek erken yakınsamanın önlenmesi için (i, j) bağı üzerinde yer alan feromon bir miktar azaltılmakta, yerel feromon güncellemesi yapılmaktadır. Ayrıca bir karıncaya ait tur oluştururken daha agresif bir yol izlemektedir; i şehrinde yer alan k karıncası q_0 olasılıkla $[\tau_{il}(t)][\eta_{il}(t)]^{\beta}$ değeri en büyük olan l şehrine geçecektir; (1-q_0) olasılıkla ise karınca sisteminde olduğu gibi (1.1) denklemine göre (α=1) geçiş gerçekleşmektedir.

Karınca Sisteminin doğrudan geliştirildiği başka bir yaklaşım ise maksimum-minimum karınca sistemidir [22-24]. Sistemde feromon miktarının çözüm üzerindeki etkisini sınırlayabilmek için, alt sınır ve üst sınır değerleri ($0<\tau_{min}\leq\tau_{ij}\leq\tau_{max}\leq 1$), belirlenmiştir. İterasyonların başlangıcında rasgele hareketi güçlendirmek amacıyla feromon değerleri üst sınıra eşitlenmektedir. Geçiş olasılıkları karınca sisteminde olduğu gibi (1.1) denklemine göre hesaplanmaktadır. Ancak feromon güncellenmesi karınca koloni sisteminde olduğu gibi o ana kadar bulunmuş en iyi çözüme ait tur bağları üzerinde ya da iterasyondaki en iyi çözüm göz önüne alınarak gerçekleştirilmektedir. Bai ve diğ. melezleme ve teorik analiz arasındaki farkı gidermek amacıyla maksimum-minimum karınca sistemine dayalı bir modeli asimetrik gezgin satıcı probleminin çözümünde kullanmışlardır. Önerdikleri algoritma klasik yapıdaki yönteme göre çok daha güçlü arama yeteneğine sahiptir [25].

GSP'nin çözümü üzerinde karınca koloni sisteminin performansını geliştirmek için paralel işlem yaklaşımı da uygulanmıştır. Randall ve Lewis aynı anda çok işlemcinin kullanılabildiği ve karınca görevlerinin işlemcilere dağıtıldığı donanımsal paralel yaklaşım kullanırken; Chu ve diğ. ile Tsai ve diğ. karıncaların yaratıldıktan sonra gruplara ayrıldığı ve aynı işlemci üzerinde çalıştığı algoritmik yaklaşımı kullanmışlardır [26-28]. Bu yaklaşımda gruplar kendi içlerinde karınca koloni sistemi işlemlerini ayrık olarak gerçekleştirmekte, belirli çevrim sayısında grupların en iyi bireyleri ya da tüm grupların en iyileri feromon güncellemesi yoluyla gruplar arası haberleşmeyi gerçekleştirmektedirler [2].

1.3.2. Genetik algoritmalar

Genetik algoritmalar klasik yöntemlerle çözümü zor olan problemlere çözüm bulmak amacıyla biyolojik evrim sürecinin bilgisayar ortamında kodlandığı ve bu evrim süreci kullanılarak optimum ya da optimuma yakın sonuçların araştırıldığı bir arama yöntemi olarak tanımlanabilir. Genetik algoritmalar ilk olarak 1975 yılında Holland tarafından önerilmiştir. Genetik algoritmaların kısa sürede çözüme ulaşması ve global çözüm bulmada etkin olması gibi avantajlarının yanı sıra yerel çözüm bulmada çok başarılı olmaması ve çözüm kalitesindeki artışın nispeten yavaş olması şeklinde dezavantajları da bulunmaktadır. Genetik algoritmalar birçok optimizasyon probleminin çözümünde kullanılmış ve iyi sonuçlar elde edilmiştir [29].

Genetik algoritmaların ilk aşamasında incelenen problemin parametreleri uygun şekilde kodlanarak kromozomlar oluşturulur. Daha sonra bu kromozomlardan oluşan bir başlangıç popülasyonu meydana getirilir. Popülasyonda yer alan her bir kromozom problemin çözümü için bir adaydır. Ardından popülasyondan bazı kromozomlar seçilir ve doğadaki genetik süreçlere dayanan çaprazlama ve mutasyon genetik operatörleri uygulanarak daha iyi kromozomlar elde edilmeye çalışılır. Ardı ardına gerçekleştirilen iterasyonlar ile yeni nesiller oluşturulur ve bu yeni nesillerin verdiği uygunluk değerleri popülasyondakilerle karşılaştırılır. Yeni nesillerde yer alan kromozomlar içinde popülasyondakilere göre daha iyi uygunluk değerine sahip olan kromozomlar popülasyona dahil edilirler. Bu sayede popülasyonda yer alan kromozomlar bir evrim süreci içinde kendilerini daha iyi kromozomlarla yenilerler. Bu süreç yeni nesil sayısı belirli bir sayıya ulaşıncaya kadar veya popülasyondaki uygunluk değerlerinde belirli bir miktarda iyileşme olmayıncaya kadar devam eder. Temel bir genetik algoritma adımları Şekil 1.5'de görüldüğü gibidir:

```
BAŞLA
  İlk  populasyonu rasgele oluştur
        WHILE (Durdurma kriteri= =0)
                Populasyondan ebeveynleri seç
                Seçilen  ebeveynlerden çocukları üret
                Bireyleri mutasyona uğrat
                Çocukları ekleyerek populasyonu genişlet
                Kötü bireyleri silerek populasyonu eski büyüklüğüne döndür
        END
  BİTİR
```

Şekil 1.5. Temel genetik algoritma adımları

Çaprazlama işleminde ebeveyn olarak seçilen bireylere ait kalıtımsal bilgilerin, onları daha iyi yapan kısımlarının çocuklarda bir araya gelerek daha iyi bireylerin oluşması hedeflenmektedir. Bu amaçla bireyleri ifade eden dizilerin belirli kısımları rasgelelik yardımıyla belirli noktalardan kesilerek bir araya getirilmektedir. İşlem sonucunda iyi bireylerin de kötü bireylerin de ortaya çıkması mümkündür. Bu durumun kontrol altında tutulması için populasyondaki bireyler belirli bir olasılıkla çaprazlama işlemine tabi tutulmaktadır. Yüksek çaprazlama oranı yeni jenerasyonda fazla sayıda yeni birey oluşturulmasına neden olurken, mevcut populasyondaki iyi bireylerin kaybedilmesine neden olabilmektedir. Düşük çaprazlama oranında ise mevcut populasyondaki iyi bireylerin uzun süreler boyunca yaşaması nedeniyle, yeni iyi çözümlerin bulunması zorlaşmaktadır. İyi bireylerin korunmasını sağlayan elitist olarak adlandırılan diğer bir strateji ise populasyondaki en iyi birey ya da bireylerin korunmasına dayanmaktadır. Mutasyon işlemi, bireyin diğer bireyler ile etkileşime girmeden kendi içinde gerçekleşen bir işlemdir. Çözümü ifade eden dizinin rasgele bir kısmı bozunmaya uğratılarak, daha iyi çözümlerin elde edilmesi, yerel en iyiye sıkışma probleminin önüne geçilmesi hedeflenir.

Temel olarak probleme özel tanımlanan uygunluk fonksiyonuna dayanarak en uygun çözümü bulmak için çeşitli yolları deneyen genetik algoritmalar, olası çözümlerin dizi olarak formüle edilebildiği kombinasyonel problemlerin çözümünde sıklıkla kullanılırlar. GSP'nin genetik algoritmalar ile çözümünde problemin yapısına daha uygun olduğu için satıcının dolaştığı turdaki şehirlere ait numaraların sırasının kromozom olarak alındığı permütasyonel kodlama kullanılır. Bu kodlama yöntemi ile çözümü ifade eden dizinin daha küçük olmasının sağlanmasına karşın; GA'da karşılaşılan temel sorun, çaprazlama aşamasında uygunsuz bireylerin ortaya çıkabilmesidir. Şekil 1.6'da, 9 şehirden oluşan gezgin satıcının turlarını gösteren ve çaprazlama işlemi için seçilen ebeveyn kromozomlarının, 4. sıradaki şehir sonrasından kesilerek çaprazlanması işlemi görülmektedir. İşlem sonucun oluşturulan yeni bireylerden ilkinde 2. ve 5. şehirler, ikincisinde ise 4. ve 8. şehirler tekrarlı olarak yer almaktadırlar. Yeni çözümler mevcut halleriyle gezgin satıcı turu ifade etmemektedirler [2].

5	2	3	7	1	4	6	9	8	Ebeveyn1
3	4	8	7	2	9	5	1	6	Ebeveyn2
5	2	3	7	**2**	9	**5**	1	6	Çocuk1
3	4	8	7	1	**4**	6	9	**8**	Çocuk2
5	2	3	7	4	9	8	1	6	Düzeltilmiş Çocuk1
3	4	8	7	1	2	6	9	5	Düzeltilmiş Çocuk2

Şekil 1.6. Çaprazlama işleminde oluşan uygunsuz çözümler ve düzeltilişi

Permütasyonel kodlamada çaprazlama sonucunda oluşan olursuz turların önüne geçebilmek amacıyla, çocuklarda tekrar eden şehirlerin silinerek yerine turda eksik olanların yazılması yoluyla düzeltilmesi uygun olabilir, ancak bu durum ebeveyn çözümlerdeki sıralamanın doğrudan aktarılmamasından dolayı veri kaybına neden olabilmektedir.

Whitley ve diğ. tarafından önerilen bağ kombinasyonu çaprazlama operatörü temel olarak ebeveynlerde her şehir için bağlantılı olduğu diğer şehirlerin, bir bağlantı haritasında tutulması ve yeni neslin bu bağlantı haritasına göre mümkün olduğunca ebeveynlerde bulunan bağlar kullanılarak oluşturulması mantığına dayanmaktadır. Böylece tekrarlı bireylerin önlenmesi durumunda ebeveynlerde bulunmayan ve turu uzatabilecek bağların yeni nesillerde bulunmasının önüne geçilebilmektedir [30].

Bean permütasyonel problemlerin çözümünde çaprazlama etkinliğini arttırabilmek amacıyla Rasgele Sayı Kodlama çaprazlama yöntemini öne sürmüştür. Yöntemde çaprazlama işlemi GSP turundaki şehirlerin pozisyonlarını tutan gerçel sayı vektörleri üzerinden gerçekleştirilmektedir. Elde edilen pozisyon vektörleri sıralama işlemine tabi tutularak tur yeniden oluşturulmaktadır. Böylece şehirler mümkün olduğunca ebeveynlerde bulundukları pozisyonlara yerleşmekte ve olursuz turlar önlenebilmektedir [31].

Gezgin Satıcı Problemi gibi zor problemlerin çözümü için genetik algoritma kullanımında karşılaşılan önemli bir sorun da erken yakınsama problemidir. Erken yakınsama, populasyondaki tüm kromozomların aynı yerel en iyi çözüm noktasında sıkışması anlamına gelmektedir. Kromozomların birbiriyle aynı olması nedeniyle populasyon tabanlı çözüm algoritmalarının önerdiği dağıtık arama avantajı ortadan kalkmaktadır. Böyle durumlarda algoritmanın küresel en iyiye ulaşma olasılığı oldukça düşüktür.

Populasyonun birbirinden ayrı ilerleyen ancak zaman zaman haberleşen parçalara bölündüğü paralel algoritma uygulamaları, erken yakınsama problemin çözümü için önerilen bir yaklaşımdır. Paralel algoritmalar ayrık yapıları nedeniyle baskın bireylerin populasyonu etkileyerek, aynı çözüme yönelmesinin önüne geçebilmektedirler. Paralel yapı, populasyondaki bireylerin başlangıçta rasgele olarak gruplara ayrılması ve grupların zaman zaman birey nakli yoluyla haberleşmesi şeklinde oluşturulabildiği gibi; uygunluk fonksiyonu değerlerine göre ayrılarak daha iyi ve kötü bireylere farklı mutasyon ve çaprazlama oranları uygulanması yoluyla da oluşturulabilir [32-34]. Paralel algoritmaların diğer bir avantajı da farklı işlemcilere dağıtılabilmeleri nedeniyle toplam işlemci gücünün artması böylece, çözülebilen problem büyüklüğü ve çözümleme hızının arttırılmasına imkan tanımasıdır. Katayama ve diğ. ile Sena ve diğ. paralel genetik algoritma kullanımı ile çaprazlama ve seçim yöntemlerinin etkileşimini GSP çözüm performansını göz önüne alarak değerlendirmiştir. Erken yakınsamanın önlenmesi için algoritma işlem süresince değişen adaptif mutasyon oranı kullanımı ya da populasyondaki aynı bireylerin bir kısmının silinerek yerlerine rasgele bireylerin eklenmesi gibi yöntemler de kullanılmıştır [35, 36]. Chen ve Chien paralel algoritma yapısı ile genetik algoritma ve karınca kolonisi yöntemlerini birlikte kullanarak gezgin satıcı problemine yönelik bir yaklaşım önermişlerdir. Karınca kolonisi ile oluşturdukları başlangıç çözümlerini genetik algoritma ile iyileştirerek karınca kolonisi çözümlerini iyileştirmişler ve belli bir döngü sayısından sonra iletişim stratejilerini kullanarak yakınsama hızını arttırmak için gruplar arasındaki öğrenme deneyimini takas etmişlerdir [37].

Genetik algoritma kullanılarak GSP çözümünde performansın geliştirilmesi için çaprazlama, mutasyon oranları ve faktörleri etkileşimlerinin deney tasarımıyla belirlenmesi [38], yerel arama operatörlerinin kullanılması ve başlangıç populasyonun GSP'ye özel oluşturulması [39, 40] gibi yaklaşımlar da kullanılmıştır.

Genetik algoritma simetrik gezgin satıcı problemleri dışındaki genelleştirilmiş GSP, kübik GSP, çoklu GSP gibi GSP türevlerinin yanında, GSP benzeri robot iş çizelgeleme, araç rotalama problemlerinin çözümünde de simetrik GSP mantığına benzer şekilde kullanılmıştır.

Çolak çalışmasında, genetik algoritmalar yardımı ile gezgin satıcı problemine çözüm aramış ve geliştirdiği algoritmanın uygulamasını Adana ilinde gıda sektöründe faaliyet gösteren bir firma üzerinde gerçekleştirmiştir. Firmanın güncel olarak kullandığı rotalar ile algoritma ile elde edilen rotalar karşılaştırılarak algoritmanın etkinliği gösterilmiştir [29]. Nagata ve Soler yeni bir çaprazlama operatörü kullanarak geliştirdikleri genetik algoritma ile asimetrik gezgin satıcı problemlerinin çözümü üzerine çalışmışlardır. Önerdikleri yeni yapının rekabetçi bir özellik gösterdiği ifade edilmiştir [41].

1.3.3. Tavlama benzetimi

Tavlama Benzetimi, Kirkpatrick ve diğ. tarafından 1983 yılında geliştirilmiş, metallerin ısıl işlemi ile bir optimizasyon probleminin çözümü arasındaki benzerlikleri temel alan bir yöntemdir [42]. Isıl işlem fiziksel sürecinde amaç, bir metalin kristal yapısı en küçük enerji durumuna ulaşacak şekilde, metali dikkatlice soğutarak moleküller arası kuvvetli bağlara sahip katı yoğunluk elde etmektir. Aşamalı ve dikkatli soğutma daha düşük enerjiyle, dayanıklı malzeme vermektedir. Eğer soğutma ani bir biçimde yapılırsa optimal sertlik özelliklerine sahip olmayan malzeme ortaya çıkmaktadır. Isı, metalin katılaştığı noktaya ulaştığında işlemler tamamlanmaktadır [43].

Optimizasyon problemlerinde bir çözümden başka bir çözüme doğru ilerlerken mümkün olan en iyi gelişmenin seçildiği hırslı algoritmaların kolaylıkla yerel en iyiye sıkıştıkları bilinmektedir. Fiziksel metal tavlama işleminde olduğu gibi, en iyileme aşamasında da mevcut çözümlerden daha iyi çözümlere biraz daha yavaş ilerlemek hatta kötü çözümlerin de bazen seçilebilmesine imkan vermek küresel en iyiye ulaşılmasını mümkün kılabilmektedir.

Tavlama Benzetimi Algoritması adımları temel olarak Şekil 1.7'de görüldüğü gibidir. Algoritmada görüldüğü gibi mevcut çözümden daha kötü bir çözümün kabul edilmesi olasılığı eğer $r < e^{-\Delta/T}$ olursa mevcuttur, ancak bu olasılık yüksek sıcaklık değerlerinde fazla iken, sıcaklık düştükçe azalmaktadır ve sıcaklık 0'a düştüğünde sadece daha iyi çözümleri kabul eden hırslı bir yapıya dönüşmektedir. Aynı sıcaklık seviyesinde malzeme tanecik yapısının dengeye ulaşması için yapılana benzer şekilde bir miktar beklenmekte, yani aynı olasılık seviyesinde belirli sayıda deneme yapılmaktadır. Buna göre algoritma performansı başlangıç sıcaklığı, azaltma miktarı, her sıcaklık seviyesinde yapılan deneme sayısı ve komşuluğu bulmak için kullanılan harekete göre değişmektedir [2].

```
BAŞLA
Başlangıç çözümü S'i oluştur ve en iyi çözüm olarak ata S* = S
Başlangıç sıcaklığını belirle
        WHILE (Donma gerçekleşmemişse)
                WHILE (Bu sıcaklıkta dengeye ulaşılmamışsa)
                Mevcut çözüme ait rasgele bir komşu çözüm S' belirle
                Δ = Tur uzunluğu(S') – Tur uzunluğu (S) hesapla
                        IF Δ ≤ 0
                        S = S' olarak ata.
                                IF Tur uzunluğu(S) < Tur uzunluğu (S*), S* = S.
                        ELSE
                        [0,1] aralığında uniform değişen rasgele sayı r belirle
                                IF r < e^(-Δ/T), S = S'.
                        END
                Sıcaklığı düşür
                END
        END
Çözüm S*'ı döndür
BİTİR
```

Şekil 1.7. Tavlama benzetimi algoritmasının adımları

Johnson ve McGeoch çalışmalarında tavlama benzetiminin gezgin satıcı problemi çözümünde kullanımının temellerini anlatmış ve bu konuda yapılan çalışmaları özetlemişlerdir. Belirli bir sıcaklık seviyesinde yapılması gereken deneme sayısının en az olası komşuluk sayısıyla orantılı olması gerekmektedir [44].

Tavlama benzetiminde karşılaşılan temel sorun, başarımın yüksek seviyede elde edilmesi için algoritmanın çalışma süresinin arttırılması gerekliliğidir. Bu sorunun önüne geçilmesi için taranacak komşuluklardan uzaktaki şehirleri içerenlerin budanması, başlangıç çözümü için sezgisel yöntemlerin kullanılması yoluyla daha düşük başlangıç sıcaklıklarıyla çalışılması gibi yollar tercih edilebilmektedir. Klasik tavlama benzetiminde daha kötüye yönelen çözümler için bir sınır bulunmamaktadır. Bu durumda çözümü oldukça kötüleştirip iyi çözüm bölgesinden uzaklaştıracak hareketlere izin verilmesi mümkündür. Bu durumun önüne geçebilmek için; aday komşuluk eğer mevcut çözümden daha kötü bir çözüm öneriyorsa ancak amaç değeri mevcut çözümün amaç değeri ya da bulunan en iyi amaç değerinden belirli bir eşiğe kadar kötüyse kabul edilebildiği, eşik temelli yaklaşımların kullanılması yoluna gidilmektedir [2].

Geng ve diğ. çalışmalarında gezgin satıcı probleminin çözümü için tavlama benzetimi ve greedy arama tekniğine dayanan bir yerel arama algoritması geliştirmişlerdir. Daha doğru çözümler elde edebilmek amacıyla standart tavlama benzetimi yapısına farklı olasılıklarda üç çeşit mutasyon kombinasyonu ilave etmişlerdir. Önerdikleri algoritmanın bilinen algoritmalara göre zaman ve sonuç avantajı sağladığı görülmektedir [45]. Chen ve Chien genetik algoritma, tavlama benzetimi, karınca kolonisi ve parçacık sürü optimizasyonu yöntemlerinin birlikte kullanıldığı bir algoritma ile gezgin satıcı probleminin çözümüne yönelik bir çalışma gerçekleştirmişlerdir. Karınca kolonisi ile bir başlangıç çözümleri oluşturulmuş, tavlama benzetimi yöntemini içeren bir genetik algoritma yaklaşımı ile başlangıç çözümleri daha iyi bir çözüme götürülmeye çalışılmıştır. Son adımda ise parçacık sürü optimizasyonu ile feromon güncellemesi gerçekleştirilmiştir [46].

1.3.4. Tabu arama

Tabu Arama yöntemi, hafıza esaslı arama strateji temeline dayanan Glover tarafından geliştirilmiş bir yöntemdir [47-48]. Yöntemde, önceki aşamalarda elde edilen bilgi, gelecek aşamalardaki yönelimleri belirlemek için kullanılmaktadır. Zor kombinasyonel eniyileme problemlerindeki yerel optimallik sorununu çözebilecek bir eniyileme yöntemi olarak öne sürülmüştür [49].

Bu yöntemde bütün olasılıkları denemek çok fazla zaman alacağından dolayı, en az hesaplama ile en iyi çözüme ulaşmak için arama uzayındaki denenmiş çözümlerin listesinin bellekte tutulmasından dolayı arama sınırlandırılmış ve yerel optimumdan kurtulma bu sayede gerçekleştirilmiştir. Diğer bir ifadeyle, tabu arama yerel optimuma sıkışma problemini aşmak için bir bellek fonksiyonu kullanıp, küresel optimumu hızlı bir şekilde aramada bir veya daha çok yerel arama prosedürünü hiyerarşik olarak yönlendiren zeki bir yaklaşımdır [50].

Tabu aramada arama komşuluk mekanizmasına dayanmaktadır. Şekil 1.8’de görüldüğü gibi her iterasyonda mevcut çözümde gelişme ortaya çıkarabilecek olası tüm hareketler (örneğin GSP için 2-opt) denenerek mevcut çözümün komşuluğunda amaç fonksiyonunda en iyi ilerlemeyi verecek hareket belirlenir ve tabu listesinde yer almaması koşuluyla gerçekleştirilir. Önceki çözümlere dönülmesini önlemek için eski çözümün kendisi ya da eski çözüme dönüşü sağlayacak hareket tabu listesine kaydedilerek, belirli iterasyon sayısı için yasaklanır. Tabu listesinde kalma süresi, tabu listesinin uzunluğuna bağlıdır. Her iterasyonda yeni gelen eleman listedeki en eski elemanın yerine yazılır. İstek kriterinin sağlanması durumunda, bir hareket tabu listesinde olsa bile gerçekleştirilebilir. Örneğin, bir hareket bilinen en iyi çözümden daha iyi bir çözüm elde edilmesine imkan tanıyorsa kabul edilebilir. İterasyon limitine ulaşıldığında, belirli bir süre boyunca gelişme sağlanamamışsa ya da tabu listesinde olmayan komşuluk kalmamışsa çözüm sonlandırılır. Standart tabu arama algoritmasına ait akış şeması Şekil 1.8'de görüldüğü gibidir [2].

```
BAŞLA
İlk  çözümü oluştur
        WHILE (Durdurma kriteri==0)
                Tüm komşu çözümler kümesi üret
                Komşu çözümleri değerlendir
                En uygun sonucu seç
                Tabu listesini güncelle
        END
BİTİR
```

Şekil 1.8. Standart tabu arama algoritmasının adımları

Tabu arama, yapı olarak her iterasyonda en iyi gelişmeyi sağlayan komşuluğa hareket edilen Tepe Tırmanma algoritmasına benzemektedir. Ancak Tepe Tırmanma algoritmasında mevcut komşuluklar içinde gelişme sağlayan bir yön yoksa arama durdurulurken, tabu aramada en az gerilemeye neden olan hareket seçilerek ilerlemeye devam edilir. Bu aşamada önceki çözüm ya da ona geri dönülmesini sağlayacak hareket tabu listesine eklenerek, yerel çözüme sıkışmasının önüne geçilmeye çalışılmaktadır. Tabu liste uzunluğu algoritmanın başarımında önemli bir faktördür. Liste uzunluğunun az olması durumunda sıkışılan yerel eniyiye geri dönülmesi mümkün olurken, uzun olması durumunda küresel en iyi çözümün bulunduğu bölgenin aranmaması sorunuyla karşılaşılabilir.

Tsubakitani ve Evans gezgin satıcı problemi için en uygun tabu listesi uzunluğunun ne olması gerektiğini belirlemeye çalışmışlardır. Tabu listesi büyüklüğünün şehir sayısının yarısından fazla olması durumunda sonuçların kötüleştiğini, 2-opt hareketi kullanıldığında şehir sayısının 1/4'ü, 3-opt hareketi kullanılması durumunda ise 1/8 ya da 1/16'sı büyüklüğünde tabu listesinin kullanılmasının uygun olduğunu öne sürmüşlerdir. Değindikleri diğer bir önemli bir nokta ise tabu listesinde yer alan bağların sadece birinin bile komşu çözümde bulunmasının yasaklanması durumunda tabu listesinin daha kısa olacağıdır [51].

Tabu liste uzunluğu problem büyüklüğüyle doğrudan ya da logaritmik olarak da sabit belirlenebilmekle birlikte, yine problem büyüklüğüyle ilintili olarak belirli aralıkta rasgele belirlenmesinin de mümkün olduğu ancak bu konuda önerilebilecek bir yönelimin görülmediği ortaya koyulmuştur. Tabu listesiyle yakın iterasyonlardaki çözümlere dönülmesinin önlenmesi kısa dönem hafıza olarak değerlendirilmektedir. Orta dönem hafıza stratejisiyle ise daha önce bulunan başarılı çözümlerin ya da bu turlarda sıklıkla bulunan bağların sonraki çözümlerde seçilmesi yoluyla iyi çözümler etrafındaki daha iyi çözümlerin bulunması hedeflenmektedir. Uzun dönem hafızada ise algoritmanın aynı çözümler etrafında sıkışmasını önlemek için bağlara tekrar sayısına göre ceza katsayısı uygulanması, komşuluk arama hareketinin değiştirilmesi, daha fazla sayıda kötü yönde harekete izin verilmesi gibi yollarla, çözümün farklılaştırılması sağlanmaya çalışılmıştır.

Gezgin satıcı problemi çözümünde başarılı sonuçlar elde edilen Tekrarlı Tabu Arama Algoritması hafıza stratejilerinin bir kombinasyonuyla çalışmaktadır [52-54]. Klasik tabu arama birçok defa çalıştırılarak elde edilen çözümler arasından en iyisi ya da iyiler arasından rasgele seçilen bazıları orta dönemli hafızada olduğu gibi tekrar tabu aramaya tabi tutulmaktadır. Ayrıca yerel çözüme sıkışmış çözümleri yerelden kurtarmak için uzun dönemli hafızada olduğu gibi farklı komşuluk hareketi kullanılarak eldeki çözüm bir miktar bozulmakta daha sonra klasik aramaya devam edilmektedir.

1.3.5. Parçacık sürü optimizasyonu

Parçacık Sürü Optimizasyonu, kuş sürülerinin arama davranışını temel alan populasyon tabanlı bir en iyileme algoritmasıdır. Yöntem ilk olarak Kennedy ve

Eberhart tarafından önerilmiştir [55]. Parçacık sürü optimizasyonunda her çözüm sürüyü oluşturan sosyal bir parçacık olarak kabul edilir. Parçacıklar çözüm uzayı üzerinde uçarak kendi başlarına arama yaparken bir yandan da sürü içersindeki diğer parçacıkların arama davranışından etkilenirler. Algoritmanın temel adımları Şekil 1.9'da görüldüğü gibi gerçekleşmektedir.

```
BAŞLA
İlk  parçacık populasyonunu oluştur
        WHILE (Durdurma kriteri= = 0)
                FOR (x_p pozisyonundaki her p parçacığı için )
                        IF (x_p,  peniyi_p'den daha iyiyse)
                                peniyi_p = x_p
                        END
                END
                Şu ana kadar bulunan en iyi çözümü geniyi_p olarak belirle.
                FOR (her p parçacığı için )
                        v_p= Hız_hesapla (x_p, peniyi_p, geniyi_p)
                        x_p = Pozisyon_belirle(x_p, v_p)
                END
        END
BİTİ
```

Şekil 1.9. Parçacık sürü optimizasyonu algoritması adımları

Parçacığın pozisyonu (x_p) o anda bulunan çözümü ifade eder. Parçacık sürekli hareket halindedir ve bu hareket denklem 1.5'de görüldüğü gibi pozisyon vektörüne her iterasyonda eklenen bir hız vektörü (v_p) ile sağlanır. Hız vektörü ise parçacığın o ana kadar bulduğu en iyi çözüm ($peniyi_p$) ile sürünün o andaki en iyi çözümünden ($geniyi_p$) etkilenerek denklem 1.4'deki gibi her iterasyonda güncellenir [2].

$$v_p(t) = v_p(t-1) + c_1.rand_1(peniyi_p(t-1) - x_p(t-1)) + c_2.rand_2(geniyi_p(t-1) - x_p(t-1)) \quad (1.4)$$

$$x_p(t) = x_p(t-1) + v_p(t) \quad (1.5)$$

Bireye ait iyi ile sürü en iyi çözümleri c_1 ve c_2 katsayılarıyla ölçeklendirilerek hız vektörüne etkirler. Bu katsayılar [0,1] aralığında rasgele değişen $rand_1$ ve $rand_2$

değerleriyle olasılıklı hale getirilir. Yerel en iyi çözüme sıkışmanın önüne geçmek için eski hız vektörünün bir *w* atalet katsayısı değeriyle çarpılması yoluna gidilebilir. Böylece mevcut hız bireyin ya da sürünün en iyi çözümünde meydana gelebilecek değişikliklere daha az duyarlı bir yapıya kavuşarak dağıtık yapının korunması mümkün olmaktadır.

Klasik algoritmada çözümler problemin her boyutu gerçel sayı ile ifade edilen çok boyutlu gerçel sayı dizileri ile ifade edilirler. Bu yapısıyla gezgin satıcı problemi gibi permütasyonel problemlerin çözümünde kullanılabilmesi için gerçel sayıların sıralama vektörüne dönüştürülmesi ya da algoritmanın yapısının sıralama vektörüyle çalışacak şekilde değiştirilmesi gerekmektedir. Bu iki yaklaşım da gezgin satıcı problemi çözümü için uygulanmıştır.

Pang ve diğ. sürekli ve kesikli çözüm uzayları arasında dönüşümün temel alındığı bir parçacık sürü optimizasyonu algoritması ile gezgin satıcı problemi çözümü üzerinde çalışmışlardır. Pozisyon güncelleme ve hız hesaplama işlemleri sürekli uzayda gerçekleştirilirken, yerel arama işlemi kesikli uzayda gerçekleştirilmiştir. Yerel en iyi çözüme sıkışmanın önüne geçebilmek için kaotik bir operatörden yararlanmışlardır. Operatör sürekli uzayda pozisyon ve hız vektörlerini rasgele sayılarla çarparak rasgele bir şekilde değiştirmektedir. Çalışmada kaotik operatörün kullanımının etkin olup olmadığın belirlenmesi ve yerel arama operatörleri ile etkileşimli bir şekilde seçim yapmak için algoritmanın dört farklı versiyonu çeşitli karşılaştırma problemleri üzerinde denenmiştir [56].

Pang ve diğ. gezgin satıcı probleminin çözümü için bulanık mantık temelli bir parçacık sürü optimizasyonu yaklaşımı önermişlerdir. Çalışmada [0,1] arasında değişen gerçel sayı değerlerinden oluşan pozisyon matrisleri kullanılmıştır. P pozisyon matrisindeki p_{ij} değeri verilen turdaki i. şehrin j. pozisyona üyeliğini göstermektedir [57].

Wang ve diğ. gezgin satıcı probleminin parçacık sürü optimizasyonu ile çözümünde gerçel ve sıralama vektörleri arasında dönüşüm yapmak yerine, pozisyon vektörünü doğrudan sıralama vektörüyle tutmuş ve farklı hız vektörü yapısı kullanmıştır [58].

Wang ve Yang benzer şekilde pozisyon vektörü olarak sıralama vektörü tutmaktadır. Hız vektörü ise şehir kaydırma operatörü kullanılarak belirlenmektedir [59]. Zhong ve diğ. gezgin satıcı probleminin parçacık sürü optimizasyonu ile çözümünde sıralama ya da gerçel sayı vektörleri yerine, bağları ifade eden ikili vektörler kullanmışlardır. Pozisyon vektörleri arası fark belirlenirken bir şehirde olup diğerinde olmayan bağlara 1 olasılığı verilirken, diğer bağlara c_1 ve c_2 ile sınırlandırılan rasgele olasılıklar verilmektedir [60].

Goldbarg ve diğ., çalışmalarında sıralama vektörünü kullanarak parçacık sürü optimizasyonu hız formülünün yapısını değiştirerek farklı hız operatörleri yaklaşımını önermişlerdir. Böylece parçacık sürü optimizasyonunun yapısı gereği bir parçacığa iki çeşit hız operatörü uygulanabilir. İlk operatör parçacığın mevcut çözümünü diğer parçacıklarla etkileşmeden kendi başına geliştirmeye çalışacağı tekli hız operatörüdür. Bu operatör diğer meta sezgisellerdeki yerel arama ile aynı işlevi görmektedir. Diğer operatör ise parçacığın diğer bir parçacık ile etkileşime girerek daha iyi bir çözüme ulaşmaya çalıştığı operatördür. Kaydırma operatörünü kullandıkları bu yaklaşımda daha iyi çözümün elde edildiği noktada durulması hedeflenmiştir. İkili hız operatöründe bireyin en iyi çözümü ya da sürünün en iyi çözümüyle, mevcut pozisyonun ilişkisi kullanılmıştır. Çözümün başlangıcında yerel hız operatörünün uygulanmasına daha büyük olasılık verilirken iterasyonlar ilerledikçe ikili operatörlerin gerçekleşme olasılığı arttırılmaktadır. Yaklaşımın büyük problemlerde de başarılı sonuçlar verdiği öne sürülse de başarı, temelde tekli hız operatöründe kullanılan ve tek başına kullanıldığı takdirde de oldukça başarı sonuçlar elde edilebilen Lin-Kerninghan sezgiselinin kullanılmasına dayanmaktadır [61]. Bunun yerine basit ters çevirme operatörünün kullanıldığı durumda, Zhong ve diğ. tarafında sunulan yaklaşıma göre daha zayıf sonuçlar ortaya çıkmaktadır.

1.3.6. Yapay sinir ağları

Yapay Sinir Ağları, insan beyninin işleyişini taklit ederek, zekice davranışların yapay olarak geçekleştirmeye çalışıldığı yaklaşımlardır. Yapay sinir ağları biyolojik sinir ağlarının işleyiş yönteminden esinlenilerek geliştirilmiştir. Yapay sinir ağları modellerinin geliştirilmesinde beynin işleyiş kuralları kullanılmıştır. Yapay sinir ağlarının öğrenme özelliği bu yaklaşımın cazibesini arttırmıştır. Yapay sinir ağları bir sisteme ilişkin tek veya çoklu parametrelere bağlı olarak tanımlanan giriş verileri ile sistemin yine tek veya çoklu parametrelere bağlı olarak tanımlanabilen çıkışları arasında ilişki kurabilme yeteneğine sahiptir. Bu ilişkinin doğrusal bir formda olması zorunlu değildir. Yapay sinir ağları çıkış değerleri bilinmeyen tanımlanmış sistem girişlerine uygun çıkışlar da üretebilirler.

Yapay sinir ağları öğrenme yöntemlerine göre birkaç gruba ayrılır. En çok bilineni; eğiticili öğrenme olarak adlandırılan, giriş ve çıkış değerlerinin verildiği, sinir ağının girişlere göre istenen çıkışları oluşturduğu yapıdır. Bu yöntem giriş ve çıkış arası ilişkinin belirlenmesinin önemli olduğu problem yapılarında kullanılmaktadır. Optimizasyon problemleri için ise çıkış değerinin ağ yapısının amaç fonksiyonunda yarattığı sonuca göre belirlendiği, öğreticisiz öğrenmenin yapıldığı sinir ağları kullanılır [2].

Gezgin satıcı problemi için en çok bilinen sinir ağı uygulaması Hopfield ve Tank tarafından gerçekleştirilen uygulamadır [62]. Önerilen ağda şehir sayısının karesi (n^2) kadar nöron bulunmaktadır. Bu yerleşimde nöronlar “0” ya da “1” değerini alabilirler. Bu şekliyle bağ matrisine benzemekle birlikte satırlar şehirleri, sütunlar ise o şehrin turda kaçıncı sırada olduğunu göstermektedir. Bir nöronun “1” değerini alması durumunda bulunduğu satırdaki şehrin, sütundaki sırada yer aldığını göstermektedir. Her satır ve sütunda sadece tek “1” değerinin bulunmasına izin verildiği modelde çıkış değeri olarak tur uzunluğunu gösteren bir enerji fonksiyonu alır. Enerji miktarının hedef değer olan 0’dan sapmasını en küçüklemek üzere nöron değerleri değiştirilerek çözüme ulaşılmaya çalışılır.

Gezgin satıcı problemi için önerilen diğer bir sinir ağı yaklaşımı ise Kohonen tarafından önerilen kendi kendine öğrenen ağlardır. Bu ağlarda giriş değeri o şehre ait koordinatları gösterirken, çıkış nöronlarına ait ağırlıklar ise şehrin turdaki yerini göstermektedir. Burada da öğreticisiz öğrenme yöntemiyle nöron ağırlıkları değiştirilerek en kısa tur bulunmaya çalışılır [63]. Yanping ve diğ. İse Kohonen ağ yerleşimini değiştirerek ve öğrenme yöntemini geliştirerek başarılı sayılabilecek sonuçlar elde etmişlerdir [64]. Tamura ve diğ. ise Hopfield ağını yerel optimuma sıkışmaması için geliştirerek başarılı sonuçlar elde etmişlerdir [65].

1.4. Kesin Çözüm Yöntemleri

Bu bölümde gezgin satıcı problemine ait küresel en iyi çözümü elde etmek amacıyla kullanılabilecek kesin çözüm yöntemleri anlatılmaktadır. Yöneylem araştırması alanındaki gezgin satıcı problemine benzer ya da farklı özellikteki birçok problemin çözümde de kullanılabilen yöntemlerden bir kısmının, gezgin satıcı probleminin çözümünde daha etkin kullanılabilmesi amacıyla özel yaklaşımlar geliştirilmiştir. Bu özel yaklaşımlar diğer yöneylem araştırması problemleri için doğrudan katkı sağlamamalarına rağmen, problemlerin çözümü için farklı bakış açıları önermektedirler. Ayrıca belirli bir kısmı oldukça benzer yapıdaki araç rotalama problemleri gibi ağ problemlerinin çözümüne doğrudan katkı sağlayabilmektedirler.

Dal ve sınır, kesme düzlemi gibi yöntemlerde en iyi çözüme ne kadar yaklaşıldığının belirlenmesi, çözümün en iyi olduğunun ispatlanması, çözüm sürecinin etkinliğinin arttırılması gibi amaçlar için çözüme ait amaç değerinin belirli değerin altında olmayacağı ve üstüne çıkamayacağını gösteren alt sınır ve üst sınırın belirlenmesine ihtiyaç duyulmaktadır. Bu iki değerin kesişmesi durumunda elde edilen çözümün en iyi olduğuna ait kanıt elde edilmiş olur. Gezgin satıcı problemi gibi en küçükleme problemleri için üst sınır değeri iyi bir sezgisel yardımıyla elde edilirken, alt sınır değeri problemin gevşetmesine ait amaç değeri ile belirlenir. Çözüm süreci bu iki değeri yaklaştırmaya yönelik olarak sürdürülür [2].

1.4.1. Sayma yöntemi

Gezgin satıcı problemi için küresel en iyi çözümü bulabilmede kullanılabilecek ilk yöntem bütün olası permütasyonların denenmesidir. Ancak n adet şehrin bulunduğu bir problemde, değerlendirilecek tur sayısının n! ile orantılı olarak büyümesi nedeniyle, yöntemin zaman karmaşıklığı oldukça yüksek olmaktadır. Bu nedenle, az sayıda şehrin bulunduğu problemler bile, yıllarca süren bilgisayar hesaplamaları gerektirmektedir.

Örneğin 20 şehrin bulunduğu bir ağda $(n-1)!/2$ = 60.822.550.204.416.000 adet farklı gezgin satıcı turu mevcuttur. Her bir turun uzunluğunun belirlenmesi ve mevcut en iyi çözümden daha iyi olup olmadığının belirlenmesi için sadece tek bir işlem yapıldığı ve 3 Ghz işlemci hızına sahip bir bilgisayarda çalışıldığı düşünülürse, denemelerin yaklaşık olarak 234 gün süreceği hesaplanır. Ancak, 21'inci şehirde bu sürenin 20 ile; 22. şehirde 20 ve 21 ile çarpılması ve sayı arttıkça şehir sayısının bir eksiğiyle çarpılarak gerekli sürenin hesaplanması gerekmektedir. Ayrıca gerçek işlem süresinin, tek bir sıralamayı denemek için 'n' ile orantılı sayıda işlem yapılması gerektiğinden, bu sürenin en az 'n' katı olduğu düşünülürse, bu yöntemin makul sürede çözüm önermediği kolayca anlaşılabilir [2].

1.4.2. Dinamik programlama

Dinamik programlama yöntemi tam saymaya benzer bir yapıda, bir tür sayma mantığı ile çözüme ulaşmaktadır. Ancak sayma sırasında gerçekleştirilen tekrarlı hesaplamaların, işlemsel yükün bir kısmını bellek yüküyle değiştirmek suretiyle, önüne geçerek çok daha kısa sürelerde çözüm elde edilmesini sağlamaktadır.

Gezgin satıcı probleminin dinamik programlama yöntemi kullanılarak çözümü için Held ve Karp tarafından önerilen yaklaşım bu alanda, bütün gezgin satıcı problemlerini çözebilecek bir yaklaşım olarak bilenen en iyi zaman karmaşıklığına sahiptir [66].

S'in 1 şehrini ve farklı en az bir adet şehri içeren bir alt küme olduğu ve j'nin S içinde 1'den farklı bir şehir olduğu kabul edilsin. C(S;j), 1'den başlayan S içindeki bütün şehirleri dolaşan ve j'de biten en kısa yolu ifade etsin. Bu durumda i ile j şehirlerini birbirine bağlayan yolun uzunluğu d_{ij} olmak üzere, en kısa tur Şekil 1.10'da görülen algoritma kullanılarak belirlenebilir.

for: Bütün j ler için $C(\{1,j\}, j) = d_{1j}$.
for: s = 3 den n'e kadar (alt kümlerin büyüklüğü burada belirlenmektedir)
    for: S'in n büyüklüğünde ve 1'i içeren bütün alt kümeleri için,
        for: Bütün $j \in S, j \neq 1$ için,
            $\{ C(S,j) = \min_{i \neq j,\, i \in s} [C(S - \{j\}, i) + d_{ij}] \}$
en iyi $= \min_{j \neq 1} [C(\{1,2, \ldots, n\}, j) + d_{j1}$.

Şekil 1.10. Dinamik programlama ile çözüm algoritması

Algoritmada her noktaya ulaşmak için geçerli en kısa yol ve uzaklık hafızada tutularak bu bilgiye ait yol her adımda bir şehir geliştirilir. İlk adımda bütün j noktalarına 1 noktasından en kısa ulaşım uzaklığı olarak (1,j) uzunluğu atanır. Daha sonraki adımda ise öncelikle yoldaki şehir sayısı bir arttırılarak üçe çıkarılır ve üçüncü j noktasına en kısa ulaşım uzaklığı ve yolu belirlenmeye çalışılır. Bunun için daha önce her j için hesaplanmış olan bütün diğer noktaların yol uzunluklarına üçüncü şehre uzunlukları eklenerek en kısa olanı belirlenir ve o şehir için en kısa yol ve uzunluğu olarak kaydedilir. Bu şekilde bütün şehirler yollara ekleninceye kadar devam edilir. En son aşamada mevcut en son şehirden başlangıç şehri olan 1'e uzunlukları da yol uzunluğuna eklenerek en kısa tur ve uzunluğu belirlenir [2].

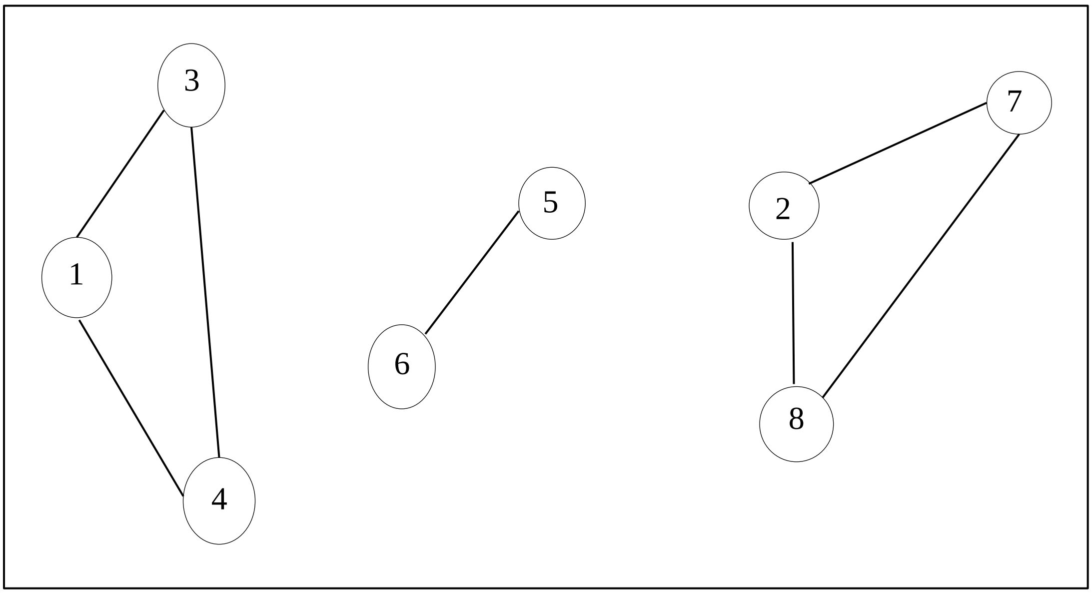
1.4.3. Dal ve sınır yöntemi

Dal ve sınır yöntemi, problemin gevşetilmiş çözümünden yola çıkarak, dallandırmalar yoluyla çözüm uzayının bölümlendirilmesi ve çözüm uzayının sıkılaştırılarak araştırılması yoluyla istenen çözümün bulunmasını sağlayan bir kesin çözüm yöntemidir. Bir eniyileme probleminin gevşetilmiş hali, sahip olduğu kısıtlar nedeniyle asıl problemin tüm olurlu bölgesini kapsayan ancak amaç fonksiyonu asıl problemin değerinin altında veya ona eşit çözümler elde eden modellemesi olarak ifade edilmektedir. Örneğin tamsayılı çözüm elde edilmesi istenen bir doğrusal modelin, tamsayı kısıtları göz önüne alınmadığında gevşetilmiş çözümü elde edilir. Kısıtlama olmaksızın yapılan arama, tamsayı çözümler uzayını da içeren ancak daha geniş, gevşetilmiş, bir uzayda gerçekleştirilir. Bu durumda elde edilen sonucun tamsayı değerlerden oluşması mümkündür. Ancak bazı değişkenlerin bu kısıtı sağlamaması durumunda bu değişkenlere ait dallandırmalar yoluyla çözüm uzayının daraltılması ve istenen tamsayılı, sıkı çözüme ulaşılması mümkün olur.

Gezgin satıcı probleminde de gevşetilmiş çözüm, gezgin satıcı turu oluşturmayan çözümler içerebilmektedir, ancak elde edilen amaç değeri GSP çözümü için bir alt sınır ifade etmektedir. Gevşetilmiş çözüm, en iyi gezgin satıcı turunu engellemeyecek şekilde adım adım dallandırma ve her dalda yeni kısıt ekleme yoluyla daraltılarak en iyi çözüm bulunur. Bu işlem sırasında en kısa tur uzunluğu için alt sınır ifade eden mevcut çözümün amaç değeri, adım adım yükselerek en son adımda en kısa tur uzunluğunun kendisine eşit olur [2].

Gezgin Satıcı Problemi için dal ve sınır yöntemi kullanılarak yapılmış birçok çalışma mevcuttur. Çalışmalar, genel olarak Atama Problemi ya da En Küçük Yayılan Ağaç problemlerini gevşetilmiş problem olarak alarak çözüm elde etmektedirler. Eastman [67], Little, Murty, Sweeney ve Karel [68], Shapiro[69], Bellmore ve Malone [70], Smith, Srinivasan ve Thompson [71], Carpaneto ve Toth[72] atama problemini; Held ve Karp [73, 74], Christofides [75], Helbig, Hansen ve Krarup [76], Smith ve Thompson [77], Volgenant ve Jonker [78] en küçük yayılan ağaç problemini gevşetme olarak kullanarak GSP üzerinde çalışmışlardır.

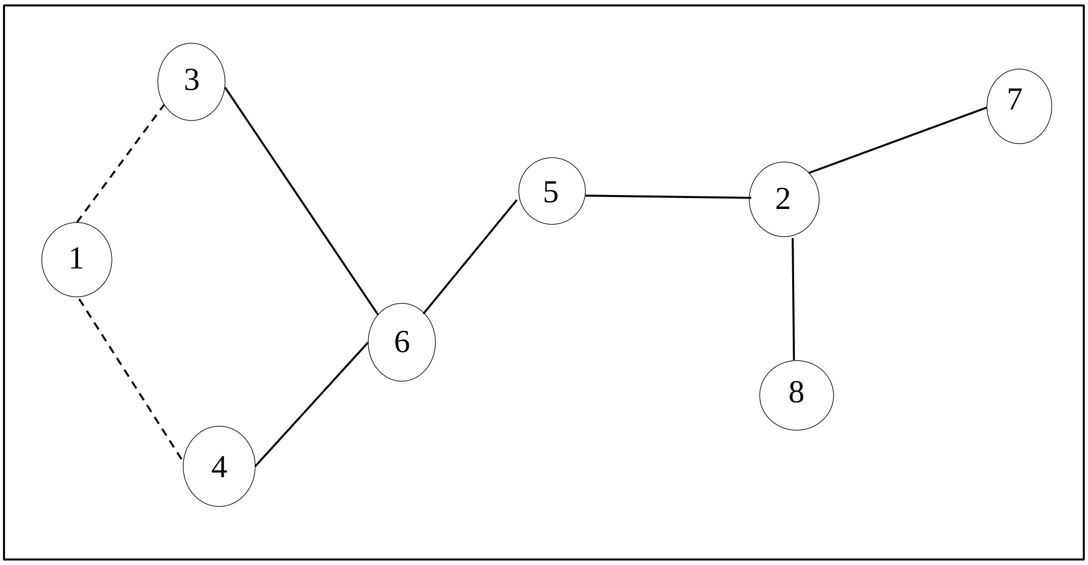
Atama Problemi, Gezgin Satıcı Probleminde olduğu gibi her noktanın bir başka noktaya bağlanmasının gerektiği ve bunun en küçük maliyetle gerçekleştirilmesinin gerekli olduğu bir problemdir. Macar algoritması kullanılarak çözüme ulaştırılması mümkündür. Ancak, atama problemi tek bir tur bulunmasını öngörmediğinden elde edilen çözümde Şekil 1.11'de görüldüğü gibi alt turlar oluşabilmektedir. Bu durumda alt turda bulunan bağlardan birinin '1' ya da '0' değerini alması, yani atama çözümünde bulunması ya da bulunmaması durumu değerlendirilerek dallandırma yapılabilir. Bir bağın çözümde bulunmasını önlemek için o bağa ait ulaşım maliyeti, seçilmemesini sağlayacak şekilde arttırılır. Bu şekilde alt tur oluşumu engellenerek tek bir tur elde edilinceye kadar çözüm ilerletilerek en küçük maliyetli gezgin satıcı turu belirlenir [2].



Şekil 1.11. Alt turlar

En Küçük Yayılan Ağaç Probleminde de amaç Gezgin Satıcı Probleminde olduğu gibi bütün noktalara en az taşıma maliyetiyle yayılmaktır. Şekil 1.12'de görülen 1-ağacı olarak isimlendirilen yayılan ağaç türünde öncelikle 1 şehri dışında kalan şehirler için en küçük yayılan ağaç belirlenir daha sonra 1 şehri ile mevcut yayılan ağacı birleştiren en kısa iki bağ belirlenerek ağaca eklenir. Yayılan ağaç çözümünün yapılabilmesi için Kruskal ya da Prim tarafından önerilen yöntemlerin kullanılması mümkündür [79]. Bu şekilde elde edilen çözümler Gezgin Satıcı Turlarını da içermekle birlikte yayılan ağaç mantığıyla bir noktadan ikiden çok bağın çıkması da mümkündür. Bu durumun engellenmesi amacıyla atama probleminde olduğu gibi dallanmaların gerçekleştiği noktalara bağlanan yollar için dal-sınır algoritması kullanılarak çözüm elde edilebilmektedir.

Küresel en iyi çözümü bilinen problemler için, atama problemi ile elde edilen alt sınır değeri, en kısa tur uzunluğuna 1- ağacı ile elde edilen alt sınır değerinden daha yakın olduğundan atama problemi ile elde edilen gevşetilmiş çözüm, 1-ağacı ile elde edilen çözümden daha sıkı olarak kabul edilmektedir [2].



Şekil 1.12. 1-ağacı

1.4.4. Doğrusal çözüm yaklaşımları

Gezgin Satıcı Probleminin çözümüne yönelik geliştirilen doğrusal yaklaşımları iki ayrı model kapsamında incelemek mümkündür. Bu modellerden ilki Dantzig-Fulkerson-Johnson (DFJ) tarafından önerilmiştir ve problemi alt tur engelleyici kesmeler yardımıyla iteratif olarak çözüme ulaştırmaktadır [80]. Diğer model ise Miller-Tucker-Zemlin (MTZ) tarafından ortaya koyulan, problemin çözümü için gerekli kesme eşitsizlikleri başlangıçta belirli olan yaklaşımdır [81].

1.3.4.1. Dantzig-Fulkerson-Johnson (DFJ) modeli

Dantzig ve diğ. tarafından gezgin satıcı probleminin çözümü için önerilen model, atama probleminin gevşetilmesi durumunda ortaya çıkan alt turların engellenebilmesi için, bu alt turları engelleyecek kısıt kullanımına dayanmaktadır. Alt turları engelleyecek bu kısıtlar daha sonra kesme düzlemi olarak isimlendirilmiştir. Buna göre; n adet şehirden oluşan gezgin satıcı problemi n(n-1)/2 adet bileşenden oluşan ve c ile gösterilebilecek bir maliyet vektörü ve yine n(n-1)/2 adet bağı ile ifade eden x vektörü ile ifade edilebilir. Eğer S bütün turların bağ vektörünü gösteriyorsa, çözüm (1.6) denklemi probleminin çözümü ile bulunabilir.

$$\min c^T x \text{ , öyleki } x \in S \tag{1.6}$$

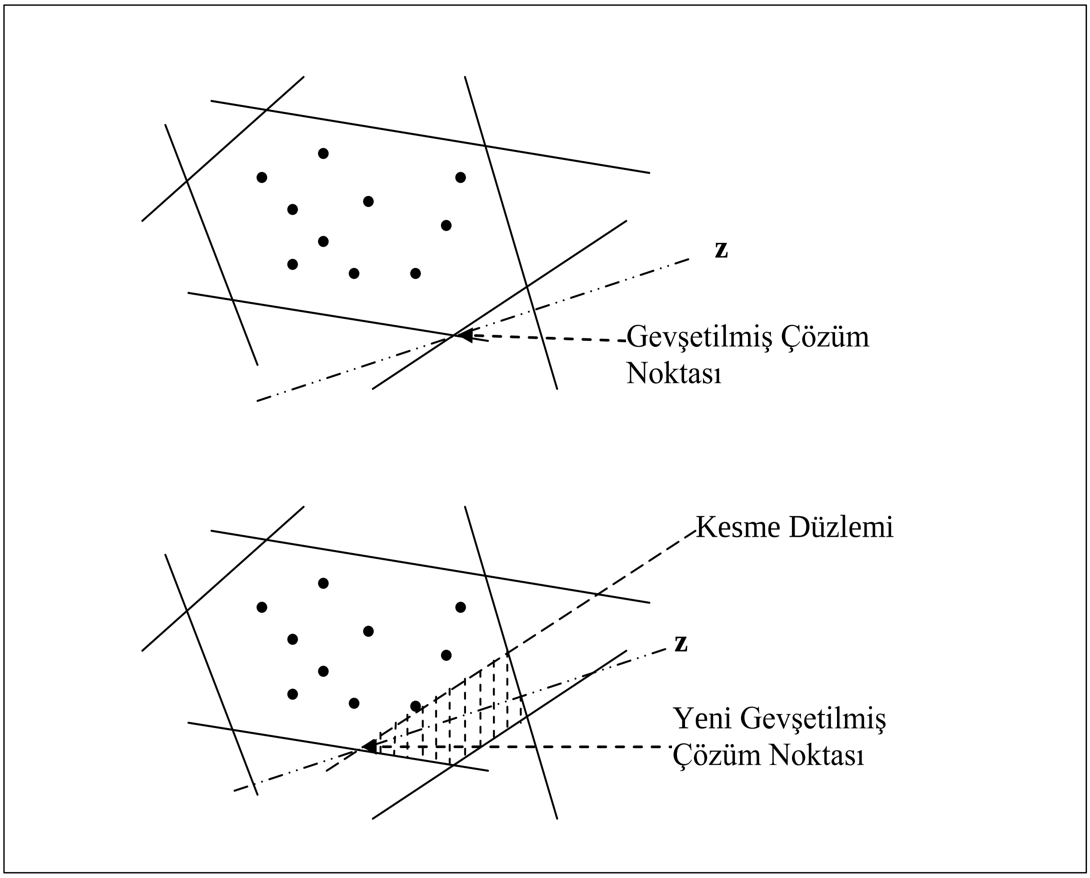
Bu durumda da S kümesinin matematiksel olarak ifade edilmesinde, yani çözümün tur oluşturmasını sağlayacak kısıtların belirlenebilmesi bir sorun olarak ortaya çıkmaktadır. Bu sorunu giderebilmek amacıyla, çözülmesi istenen probleme yakın bir problem tanımlanarak onun çözümü gerçekleştirilebilir. ‘A’ değişken katsayılarını, ‘b’ ise sağ taraf değerini ifadeden vektör olmak üzere (1.7) denklemi gibi bir doğrusal model kullanılarak çözüme ulaşılması mümkün olabilmektedir.

$$\min c^T x \text{ , öyleki } Ax \leq b \tag{1.7}$$

Burada, $\{ x: Ax \leq b \}$ şeklinde tanımlanan P çokyüzlüsü S’i içermektedir ve sınırlıdır. (1.7) eşitsizliğinin çözülmesi durumunda S’in elemanı olan bir x* çözümü bulunmuşsa, bu çözüm optimaldir. Aksi takdirde x* çözümünü dışarıda bırakan ancak S’teki tüm noktaları içeren yeni eşitsizliklerin çözüme eklenmesi gerekmektedir. Dantzig tarafından ilk defa ortaya atılan bu eşitsizlikler S’in elemanı olan yeni bir x* çözümü bulunabilmesi için çözüm uzayını daraltmaktadırlar. Bu çeşit eşitsizliklere kesme, kesme düzlemi ya da alt tur engelleme kısıtı adı verilmektedir [2].

Kesme düzlemi yöntemi, Gomory tarafından önerilen tam sayı çözümler elde edilmesi amacıyla önerilmiş olan bir yöntemdir [82]. Kesme düzlemi, gevşetilmiş çözümde istenen değişkenlerin tam sayı değerini almaması durumunda final simpleks çözümden türetilerek ve bu çözüme eklenerek, tam sayılı çözümlerin elde edilmesini sağlamaktadır. Gomory kesmeleri gevşetilmiş çözüm uzayını tam sayılı çözümler elde etmek amacıyla daraltmakta, tam sayılı olmayan çözüm uzayının bir kısmını keserek çözüm uzayının dışında bırakmaktadır. Elde edilen çözüm istenen tamsayı kısıtlarını sağlayana kadar Gomory kesmeleri eklenmeye devam etmektedir.

Gezgin Satıcı Problemi için önerilen kesmeler Şekil 1.13'de görüldüğü gibi gevşetilmiş problem çözümünden gezgin satıcı turu yapısında olmayan çözümleri (taralı alan) keserek dışarıda bırakmaktadır [2].



Şekil 1.13. Kesme düzlemi yöntemi

Karar değişkenleri x_{ij}, i noktasından j noktasına yapılan bağı; c_{ij} bağ uzunluklarını göstermek üzere, gezgin satıcı tur uzunluğunu gösteren amaç değeri aşağıdaki gibi basit doğrusal bir formda ifade edilir:

$$x_{ij} = \begin{cases} 1, & \text{eğer i noktasından j bağ var ise} \\ 0, & \text{aksi durumda} \end{cases}$$

$$Z_{\min} = \sum_{i} \sum_{j>i} c_{ij} x_{ij} \tag{1.8}$$

Tamsayılı gezgin satıcı problemlerini karmaşıklaştıran şey modelin kısıtlarıdır. Biraz dikkatli düşünmek sistemi açık ve anlaşılır hale getirecektir. Simetrik bir problem için, uygun bir çözümde, herhangi bir i noktasına ilişkin iki x değişkeni 1 değerini alabilir. Bu bağlantılardan birisi i'nin güzergahtaki önceki şehirle, diğeri ise i'nin sonraki şehirle bağlantısını kurar. Yani bir noktaya gelen bir bağ olması gerektiği gibi, noktadan çıkan bir bağ olması gerekmektedir. Bu gerekliliği matematiksel modele eklediğimizde modelin tamamı (1.9) denkleminde görüldüğü gibi ifade edilir:

$$Z_{\min} = \sum_{i} \sum_{j>i} c_{ij} x_{ij}$$

A.K.A.

$$\sum_{i} x_{ij} = 1 \quad (i = 1...n) \tag{1.9}$$

$$\sum_{j} x_{ij} = 1 \quad (j = 1...n)$$

$$x_{ij} = 0 \text{ veya } 1$$

Atama problemiyle aynı olan bu gevşetilmiş çözüm, Şekil 1.11'de görüldüğü gibi her durumda istenen çözümü vermemekte, alt turların oluşmasını önleyememektedir. Dantzig tarafından öne sürülen alt tur engelleme kısıtları aşağıdaki gibi ifade edilebilir. S, ziyaret edilecek noktaların/şehirlerin uygun alt kümesi olmak üzere;

$$\begin{aligned} &\sum_{i,j\in S} x_{ij} \le |S| - 1 \\ &2 \le |S| \le n - 1 \end{aligned} \tag{1.10}$$

Bu kısıt, oluşan alt turların her birinin kendi içinde yaptığı bağların sayısının, o alt turda yer alan nokta sayısının toplamından en az bir eksik ya da ona eşit olması gerektiğini ifade eder, bu duruma sadece S alt kümesinin tüm noktaları kapsaması yani gezgin satıcı turunun oluşması durumunda izin verilir. Bu şekilde elde edilen sıkılaştırılmış x^* çözümü S kümesine ait ise çözüm sonlandırılır. Ancak aksi takdirde yeni çözüme göre gerekli incelemeler yapılarak, yeni kesmeler eklenir ve çözüme devam edilir. Daha sonraki çalışmalarda Dantzig tarafından sunulan alt tur engelleme kısıtlarına benzer şekilde gezgin satıcı turu oluşturmayan çözümlerin, çözüm uzayından çıkarılması amacıyla kullanılan birçok kesme düzlemi eklenmiştir. Tarak, klik ağacı, yıldız, yol, merdiven ve domino eşitsizlikleri bu alanda en çok bilinen eşitsizliklerdendir [2].

Yukarıdaki modelde, x değişkenlerinin 0 ya da 1 değerini alması gerekmektedir. Şehir sayısının giderek arttığı problemlerde tamsayılı çözümleri etkin bir şekilde yapacak bir yönteme ihtiyaç duyulmaktadır. Final simpleks tablo üzerine yeni kesmeler ekleyerek tam sayılı çözümler elde edilmesini sağlayan Gomory kesmeleri küçük problemlerde başarı sağlamakla birlikte, artan problem büyüklükleri karşısında yetersiz kalmaktadır. Bu durumda tamsayı kısıtlarının yerine $0 \leq x_{ij} \leq 1$ kısıtı koyularak, kesme ve dal-sınır yöntemlerinin karışımından oluşan dallandır ve kes yöntemi ortaya konulmuştur.

Kesme düzlemleri eklenerek, tam sayı kısıtları olmaksızın yapılan çözümlerden elde edilen amaç değeri giderek artmakta ancak belli bir süre sonra, meydana gelen artış oldukça küçük düzeylere ulaşabilmektedir. Böyle bir durumda, çözümde henüz tamsayı olmayan değişkenler (x^e) mevcut ise bu değişkenler için, $x^e \leq 0$ ve $x^e \geq 1$ kısıtları eklenerek dallanmalar yapılabilir, böylece tamsayılı çözüm elde imkanı doğmaktadır. Bu tür dallandırmaları istenilen tamsayılı çözüm elde edilinceye dek devam ettirmek mümkündür. Bir adım ötesinde ise bir dallanma sonrasında elde edilen çözüme yeniden kesme değişkenleri eklenmesi yer almaktadır. Yöntemde, dallanma ve kesmenin ne zaman yapılacağı, ne tür kesme kullanılacağı, kesme üretim yöntemleri gibi faktörler başarımı etkilemektedir [2].

Padberg ve Rinaldi [83-85], Grotschel [86], Crowder ve Padberg [87], Grötschel ve Holland [88], Chvatal [89], Grötschel ve Pulleybank [90], Applegate ve diğ. [91, 92] dalandır ve kes yöntemi kullanılarak gezgin satıcı problemi için çözüm aramışlardır.

1.3.4.2. Miller-Tucker-Zemlin (MTZ) modeli

Gezgin satıcı probleminin tamsayılı çözümüne yönelik, Miller ve diğ. tarafından önerilen model denklem (1.11)'de görüldüğü gibidir. Modelde problem sınırlarını belirleyen kesme düzlemi denklemlerinin hepsi ilk aşamada yer almakta ve tam sayı kısıtları ile çözüm gerçekleştirilmektedir. Burada şehirlerin sıralarını belirten u_i yapay değişkenleri dışarıdan eklenmektedir [81].

$$Z_{min} = \sum_{i} \sum_{j>i} c_{ij} x_{ij}$$

A.K.A.

$$\sum_{i} x_{ij} = 1 \quad (i = 1...n)$$

$$\sum_{j} x_{ij} = 1 \quad (j = 1...n) \tag{1.11}$$

$$u_i - u_j + n\, x_{ij} \leq (n-1), \quad (i = 1...n)\ (j = 1...n)$$

$$1 \leq u_i \leq n-1$$

$$x_{ij} = 0 \text{ veya } 1$$

Asimetrik ve simetrik problemlerin her ikisi için de kullanılabilen bu modelde n(n-1) adet ikili değişken ve n adet sürekli değişken bulunmaktadır. Sadece n^2 sabit değişkenin bulunduğu Miller-Tucker-Zemlin modelinin, Dantzig-Fulkerson-Johnson modeline göre avantajı kısıt sayısının baştan belli olması ve modelin tek çözümde sonuca ulaştırmasıdır. Ancak n(n-1) adet ikili değişken problem büyüdükçe, Dantzig-Fulkerson-Johnson modelinde olduğu gibi Gomory kesme yöntemi yetersiz kalarak çözüm süresinin uzamasına neden olmaktadır. Desrochers ve Laporte [93] Miller-Tucker-Zemlin modelinin biraz daha sıkılaştırılmış (1.12) denkleminde görülen halini sunmuşlardır.

$$Z_{min} = \sum_{i} \sum_{j>i} c_{ij} x_{ij}$$

A.K.A.

$$\sum_{i} \sum_{j} x_{ij} = 1 \quad (i = 1...n)\ (j = 1...n)$$

$$u_i - u_j + (n-1)\, x_{ij} + (n-3)\, x_{ij} \leq (n-2), \quad (i = 2...n)\ (j = 2...n)(i \neq j) \qquad (1.12)$$

$$u_i \geq 1 + (n-3) x_{i1} + \sum_{\substack{j=2 \\ j \neq i}} x_{ji}\ , (i = 2...n)$$

$$u_i \geq n - 1 - (n-3) x_{1i} - \sum_{\substack{j=2 \\ j \neq i}} x_{ij}\ , (i = 2...n)$$

$$x_{ij} = 0 \text{ veya } 1$$

2.VERİ MADENCİLİĞİ

2.1. Veri Madenciliği Tanımı

Veri madenciliğinin amacı veri yığınından bilgi elde etmektir. Elde edilen bu bilginin bazı ideal özelliklere sahip olması gerekir. Bu ideal özellikler bilginin doğru, anlaşılır ve ilginç olmasıdır. İlginçlikten kastedilen, keşfedilen bilginin kullanıcı için yeni, şaşırtıcı ve kullanışlı olmasıdır [94].

Veri madenciliği adını ve popülaritesini, saklanan verilerin bir dağa benzetilmesinden ve bu dağın içinde çok değerli taşların olmasından alır. Sorun ise, bu yığın içerisinde değersiz kayaların da bulunmasıdır ve değerli olanlara ulaşabilmek için değersizlerin ayıklanması gerektiğidir [95]. Ayrıca veri madenciliğinin etkili kullanımı ile projelerde maliyetler azaltılıp, gelirler artırılabilir [96]. Veri madenciliği büyük hacimli dallardaki örüntüleri araştıran matematiksel algoritmaları kullanır. Veri madenciliği hipotezleri keşfeder, sonuçları birleştirmek için insan yeteneğini kullanır.

Veri madenciliği; çok çeşitli verilere dayanarak daha önce keşfedilmemiş bilgileri ortaya çıkarmak, karar vermek ve eylem planını gerçekleştirmek için kullanma sürecidir [97]. Bu noktada kendi başına bir çözüm değil çözüme ulaşmak için verilecek karar sürecini destekleyen, problemi çözmek için gerekli olan bilgileri sağlamaya yarayan bir araçtır.

Sonuç olarak veri madenciliği, önceden bilinmeyen ilişkilerin bulunması için bugünün endüstrisinde yaratılan büyük miktarlardaki veriyi analiz eden bir yoldur. Yüksek güçlü bilgisayarlara ve gereken yazılımlara kolay ve düşük fiyatlarla ulaşılabilmesi bu teknolojinin işlemesini olanaklı kılmıştır. İnternet ise birçok noktadaki verinin toplanmasını sağlamaktadır [98].

Veri madenciliği tanımlarda öne çıkan noktalar şunlardır: Veri Madenciliği; büyük ve karmaşık verilerle çalışır ve her türlü veriyi kullanarak çözümler üretebilir. İstatistik, yapay zekâ, makine öğrenmesi, veri tabanlarında bilgi keşfi, bilgisayar bilimi vb. gibi disiplinlerden faydalanır. Daha önceden bilinmeyen, doğrulanabilir, etkinleştirilebilir enformasyon arar. Otomatik veya yarı otomatik olarak çalışan çözüm araçları kullanır. Birçok endüstride farklı tipte problemlere çözümler ortaya koyabilmek amacıyla kullanılmaktadır. Sorunlara göre değişen çözüm araçları vardır.

Veri madenciliğinin temelleri 1960'lı yıllara dayanmaktadır. Bu yıllarda veri toplama sistemleri, enformasyon teknolojisini kullanmıştır. 1960'lardan sonra gözle görülür gelişmeler meydana gelmiştir. 1970'ler de ise ilişkisel veri tabanları kullanılmaya başlanmış, 1980'lerde ise ilişkisel veri tabanları popüler olmaya başlamıştır. 1990 ve 2000'lerde çoklu ortam veri tabanları ve web veri tabanları kullanılmıştır. Günümüzde veri ambarlama ile veri madenciliğinin oluşum süreci devam etmektedir.

1960'lı yıllar ve öncesinde başlayan veri toplama ve veritabanı yaratma çalışmaları ilkel dosyalama işlemlerinden ibaretti. Cevaplanabilen karar problemleri ayrıntıya girmeden belirli bir döneme ilişkin problemler iken, kullanabilen teknolojiler bilgisayarlar, teypler ve diskler olmuştur. Bu yıllarda ürün sağlayıcı firmalar IBM ve CDC'dir. Bu dönemde geriye dönük, statik veri dağıtımı yapılmıştır. Bu yıllarda sadece geriye dönük aranılan veriye ulaşılırken bu verilerden enformasyon elde edilmediği görülmektedir [99].

1980'li yıllara geçildiğinde statik veri dağıtımından dinamik veri dağıtımına geçildiği görülmektedir. İlişkisel veri tabanlarının oluşumu, SQL ve ODBC ile veri kaynaklarına ulaşım bu yıllarda gerçekleşmiştir. Ürün sağlayıcılardaki artışta dikkat çekicidir [99].

1990'lı yıllarda veri toplama ve saklamadaki gelişimin sonucu olarak veri ambarları oluşturulmaya başlanmış ve karar destek sistemleri devreye girmiştir. Dinamik veri dağıtımı çoklu düzeylerde yapılmaya başlanmıştır. OLAP ve çok boyutlu veri tabanları göze çarpan değişimlerdir [99].

Bugün ise veri madenciliği tam anlamı ile yapılmaya başlanmış olup geriye dönük yapılabilen veri dağıtımına ek olarak ileriye dönük tahminlere imkan veren proaktif enformasyon dağıtımı da yapılmaya başlanmıştır. Burada da ön plana çıkan değişim enformasyon dağıtımı olmuştur. 1960'lı yıllarda yalnızca istenilen verinin elde edilmesiyle sonuçlanan işlemler, artık geleceğe dönük tahminler ve bu tahminlerin nedenlerinin açıklanmasına dönüşen işlemler halini almıştır [99].

2.2. Veri Madenciliğinin Tercih Edilme Nedenleri

Son yıllarda veri madenciliği tekniklerine olan ilgi hızlı bir şekilde artmaktadır. 1980'ler de şirketler, müşterileri, rakipleri, ürünleri ile ilgili verilerden oluşan veri tabanları oluşturmuşlardır. Bu veri tabanları potansiyel altın madeni gibidir. Sayısı milyonları geçen bu veriler gizli bilgiler içerirler ve bunlara kolaylıkla SQL veri tabanı sorgulama dili ya da başka yüzeysel sorgulama dilleri kullanılarak ulaşılabilir. SQL sadece bir sorgulama dilidir ve önceden bilinen sınırlamalar altında bilgileri bulmaya yardım eder [100]. Veri madenciliği algoritmaları tipik olarak, veri tabanının alt gruplarında ya da uygun kümelerde belirginleşir. Birçok durumda, tekrarlanabilen SQL sorguları kullanılır ve ortalama sonuçlar elde edilir. Bunu elle yapmak mümkündür fakat oldukça yorucu ve uzun süren bir iştir.

Bilgisayarlarda ağ kullanımında artan gelişmeye paralel olarak veri tabanları ile bağlantı kurmakta kolaylaşmaktadır. Böylelikle demografik verili dosyalar ile müşteri dosyaları arasında bağlantı kurulabilmekte ve belirli popülasyon gruplarının kimliklerinin belirlenmesi sağlanabilmektedir. Son birkaç yılda makine öğrenimi teknikleri oldukça gelişmiştir. Sinir ağları, genetik algoritmalar ve diğer basit uygulanabilir öğrenme teknikleri ile veri tabanlarıyla ileri düzey bağlantılar kurulması mümkün olmaktadır. Bu sayede müşteri ile hizmet veren arasındaki ilişki, kişisel bilgileri hizmet verenin masasındaki bilgisayardan merkezi bilgi sistemlerine kolaylıkla gönderebilmektedir [101].

2.3. Veri Madenciliği Uygulama Alanları

Günümüzde veri madenciliği teknikleri farklı alanlarda uygulanmaya başlanmıştır. Pazarlama, bankacılık, sigorta, perakende, telekomünikasyon, sağlık, endüstri vb. birçok alanda farklı uygulama ve analizler veri madenciliği teknik ve algoritmaları ile gerçekleştirilmektedir [102].

Pazarlama alanında yapılan çalışmalara bakıldığında; müşteri segmentasyonunun gerçekleştirilmesi, müşterilerin demografik özellikleri arasında bağlantıların kurulması, çeşitli pazarlama kampanyalarının gerçekleştirilmesi, çapraz satış analizlerinin gerçekleştirilmesi, müşteri değerlendirmesi ve müşteri ilişkileri yönetimi, kaybedilen müşterilerin geri kazanılması, sepet analizi yardımıyla ürünlerin raflara dağılımı vb. birçok konuda veri madenciliği çalışma ve analizlerinin yapıldığı görülmektedir.

Bankacılık alanında yapılan çalışmalar ise farklı finansal göstergeler arasındaki gizli korelasyonlarının bulunması, kredi kartı dolandırıcılıklarının tespiti, müşteri segmentasyonunun belirlenmesi, kredi taleplerinin değerlendirilmesi, usulsüzlüklerin tespiti, risk analizleri ve yönetimi, kar analizi ve portföy yönetimi gibi konular olarak görülmektedir.

Sigortacılık alanında ise yeni poliçe talep edecek müşterilerin tahmin edilmesinde, sigorta dolandırıcılıklarının tespitinde, riskli müşteri tipinin belirlenmesinde veri madenciliği kullanılmaktadır.

Perakendecilik alanında; satış noktası veri analizlerinde, alış-veriş sepeti analizlerinde, tedarik ve mağaza yerleşim optimizasyonunda veri madenciliği yöntemlerinden faydalanılmaktadır.

Telekomünikasyon alanında; kalite ve iyileştirme analizlerinde, hatların yoğunluk tahminlerinde, iletişim desenlerinin belirlenmesinde, kaynakların daha iyi kullanılmasında, servis kalitesinin arttırılmasında veri madenciliği uygulamaları görülmektedir.

Sağlık ve ilaç alanında yapılan veri madenciliği çalışmaları test sonuçlarının tahmin edilmesi, ürün geliştirme, tıbbi teşhis ve tedavi sürecinin belirlenmesi, DNA içerisinde genlerin sıralarının belirlenmesi, protein analizlerinin yapılması, hastalık haritalarının hazırlanması, hastalık tanılarının gerçekleştirilmesi, sağlık politikalarına yön verilmesi vb. olarak sıralanabilir.

Endüstri alanında ise kalite kontrol analizlerinde, lojistik faaliyetlerde, üretim süreçlerinin optimizasyonunda veri madenciliği teknikleri kullanılmaktadır.

2.4. Veri Madenciliği Teknikleri

Veri madenciliği teknikleri işlevlerine göre 3 temel gruba ayrılır. Bunlar sınıflama, kümeleme ve birliktelik kuralları-sıralı örüntüler şeklinde sıralanabilir.

2.4.1. Sınıflama

Sınıflama, veri kümesinin önceden belirlenmiş olan çıktılara uygun olarak ayrıştırılmasını sağlayan bir tekniktir. Sınıflama tekniğinde çıktılar, önceden belirlendiği için veri kümesi denetimli olarak öğrenilir.

Sınıflama tekniğinde, girdiler çeşitli niteliklerine göre bir sınıflayıcı bir model tarafından sınıflara atanmaktadır. Eldeki nesnelerin bir sınıfa atanıp atanmayacağının ya da sınıflardan hangisine atanacağının belirlenmesi sürecidir. Başka bir ifade ile nesneler veya durumlar için uygun sınıf tahmin edilmesidir. Sınıflama girdileri, her biri bir sınıf etiketi ile etiketlenecek gözlem veya örneklerden oluşan bir eğitim kümesidir. Çıktı ise modelin her bir gözleme, niteliklere dayalı olarak atadığı sınıf etiketidir. Sınıflama, makine öğrenmesinin de önemli araçlarından biridir. Tümevarımsal öğrenmenin temel çerçevesi eğitim örneklerinden oluşan bir eğitim kümesi ile test örneklerinden oluşan bir test kümesini içerir. Sınıflama iki adımda gerçekleşir. Bunlar verilerin eğitimi ve modelin testidir. Eğitim, eğitim kümesinden çıkarımla modelin oluşturulması, test ise test kümesini kullanarak modelin kesinliğinin kontrol edilmesidir. Modellerin kesinliğinin belirlenmesi için test örneklerinin iyi bilinen sınıfı, model tarafından tahmin edilen sınıf ile karşılaştırılır. Test örneklerinin model tarafından doğru olarak sınıflanma oranı kesinlik oranını verir. Girdilerden bu çıktıları üreten model, daha sonra sınıf etiketi bilinmeyen veya kayıp olan yeni gözlem veya örneklerin sınıf etiketini tahmin etmek için kullanılabilir. Ana sınıflama teknikleri olarak lojistik regresyon, diskriminant analizi, karar ağaçları, bayesgil sınıflayıcıları, eğer-sonra kural çıkarımları, diğer mantıksal formüller yapay sinir ağları, bulanık kümeler, kaba kümeler sayılabilir [103].

Sınıflama tekniği tam sınıflama ve kısmi sınıflama olmak üzere ikiye ayrılmaktadır. Buradaki tam sınıflama kavramı veri içindeki tüm sınıflar ve örnekleri kapsayan modeller ile ilgilidir. Kısmi sınıflamada, tam sınıflamada olduğu gibi veri sınıflarının özellikleri gösterilmektedir. Ancak kısmi sınıflandırma modellerinde tüm sınıflar veya verilen sınıfın tüm örnekleri kapsanmayabilir.

2.4.2. Kümeleme

Kümeleme, veri kümesindeki benzer kayıtların gruplandırılmasını sağlayan bir tekniktir. Kümeleme işlemi çoğunlukla başka bir veri madenciliği uygulaması için bir ilk işlem olarak kullanılır [104]. Kümeleme tekniğinde genellikle K-ortalamalar algoritması ya da Kohonen şebekesi gibi istatistiksel yöntemler kullanılmaktadır. Kullanılan yöntemden bağımsız olarak işlem süreci aynıdır. Başlangıçta her kayıt oluşturulan kümelerle karşılaştırılır. Kayıt kendisine en yakın olan kümeye atanır ve bu kümeyi tanımlayan değeri değiştirir. Sonrasında optimum çözüm bulununcaya kadar kayıtlar yeniden atanır ve küme merkezleri ayarlanır [105]. En yaygın kullanılan kümeleme algoritması K-ortalamalar algoritmasıdır. K-ortalamalar algoritması diğer kümeleme teknikleri ile karşılaştırıldığında büyük veritabanlarının kümelenmesinde oldukça etkin bir algoritmadır.

K-ortalamalar algoritması oldukça etkin bir algoritma olmakla birlikte, sadece nümerik veri ile çalışır. Fakat veri madenciliği uygulamaları sıklıkla kategorik verileri de içermektedir. K ortalamalar algoritmasının geliştirilmesi ile elde edilen k modlar algoritması ise kategorik veriler üzerinde çalışabilen bir algoritmadır. K-ortalamalar algoritmasında küme merkezleri, küme ortalaması alınarak hesaplanırken, k modlar algoritmasında küme merkezlerinin belirlenmesinde kümede en sık tekrarlanan değerler dikkate alınır [106].

2.4.3. Birliktelik kuralları ve ardışık zamanlı örüntüler

Birliktelik kuralları ve ardışık zamanlı örüntüleri birbirinden ayıran özellik zaman kavramının uygulamada dikkate alınmasından kaynaklanmaktadır. Nesneler arasındaki birlikteliklerin belli bir zaman dönemi boyunca incelenmesi "ardışık zamanlı örüntü çözümlemesi" olarak da isimlendirilir [107].

Birliktelik kuralları ticaret, mühendislik, fen ve sağlık sektörlerinin içinde bulunduğu birçok alanda uygulanmaktadır. Veri madenciliğinin özel bir uygulama alanı olan birliktelik kuralları için veri madenciliği araştırmalarında çok büyük yatırımlar yapılmaktadır. Birliktelik kuralları aynı işlem içinde çoğunlukla beraber görülen nesneleri tanımlayan kuralları içermektedir. Birliktelik kurallarının bulunması ile pazar sepeti analizi yapılmaktadır. Pazar sepeti analizinde, müşteriler tarafından satın alınan ürünler nesneleri tanımlamaktadır. İşlem ise birçok nesneyi içinde bulunduran tek bir satın almayı temsil etmektedir. Pazar sepeti analizinde sıklıkla beraber alınan nesneler üzerine çalışılmaktadır. Bulunan birliktelik kuralları ile nesnelerin birbiri ile nasıl ilişkili olduğu bilgisine ulaşılmaktadır.

Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın alma eğiliminde olduğunun belirlenmesi, müşteriye daha fazla ürün satma yollarından birisidir [108].

Birliktelik analizi, bir veri kümesindeki kayıtlar arasındaki bağlantıları arayan denetimsiz veri madenciliği şeklidir. Birliktelik analizi çoğu zaman perakende sektöründe süpermarket müşterilerinin satın alma davranışlarını ortaya koymak için kullanıldığından "pazar sepeti analizi" olarak da adlandırılır [97].

Ardışık zamanlı örüntüler analizinden ise birbiriyle ilişkisi olan ancak birbirini izleyen zamansal dönemlerde gerçekleşen ilişkilerin tanımlanmasında faydalanılmaktadır.

Demiriz [109] tarafından önerilen ASIPATH (A SImple PATH) algoritması, veri yığınları içerisinde ardışık zamanlı örüntülerin bulunabilmesi amacıyla kullanılan bir sıra madenciliği algoritmasıdır. Veri yığınını içeren kümenin tümüyle sadece tek bir kez taranması ve bununla birlikte birkaç kısmi tarama ile ardışıklık ilişki içinde olan verilerin tespitini kolaylıkla sağlayabilen bir algoritmadır. Bu algoritma paralel algoritma tasarımında genellikle uygulanan seri olarak çalışan algoritmaların paralel halde çalıştırılması yaklaşımının aksine, çoklu işlemci ortamı için tasarlanmıştır. Bu sayede algoritma kısa sürede etkin sonuçlar üretebilmektedir.

Bu tez çalışması kapsamında gerçekleştirilen veri madenciliği uygulamasında, oluşturulan gezgin satıcı turlarındaki sık tekrar eden ardışık şehir çiftlerinin tespitinde ASIPATH algoritması kullanılmıştır.

2.5. Optimizasyonda Veri Madenciliği Teknikleri Kullanımı

Literatür incelendiğinde, veri madenciliği tekniklerinin sınırlı sayıda ve genellikle optimizasyonda kullanılan algoritmaların performansını arttırmak amacıyla kullanıldığı görülmektedir. Bu tez çalışmasında ise amaçlanan, optimizasyonun veri madenciliği tekniği ile daha da etkin bir şekilde gerçekleştirilmesidir. Bu bakış açısı ile bakıldığında literatürde oldukça az sayıda çalışma görülmekte, tez konusu kapsamındaki gezgin satıcı probleminin kesin çözümüne yönelik olarak ise veri madenciliği tekniklerini kullanan hiçbir çalışmaya rastlanmamaktadır.

Koonce ve Tsai çalışmalarında, atölye tipi iş sıralama problemlerinin çözümünde veri madenciliği algoritmalarının yeni bir kullanımını ortaya koymuşlarıdır. Genetik algoritma ile oluşturulan bir çizelge veri kümesinden, ilişkiler ortaya çıkarmak ve kural setleri geliştirmek amacıyla veri madenciliğinden faydalanmışlardır. Sonuçta veri madenciliği kullanımı ile algoritmaların daha etkin bir şekilde kullanılabileceğine değinilmiştir [110].

Santos ve diğ. çalışmalarında, araç rotalama probleminin genetik algoritma ile çözümünde algoritmanın performansını artırmak için alternatifler ortaya koymuşlardır. Genetik algoritma, lokal arama ile genetik algoritma, veri madenciliği modülünü içeren genetik algoritma, lokal arama ve veri madenciliğini içeren genetik algoritma olmak üzere dört algoritmanın çözüm sonuçlarını karşılaştırmışlardır. Çalışmada, veri madenciliği tekniklerinden apriori algoritması ve lokal arama çözüm kalitesini iyileştirmek amacıyla kullanılmıştır. Sonuçta, apriori algoritması ve lokal aramanın birlikte kullanıldığı genetik algoritmanın, diğer algoritmalara göre daha iyi sonuçlar verdiğini ortaya koymuşlardır [111].

Ceran, esnek akış tipi çizelgeleme problemlerinde maksimum tamamlanma zamanının minimizasyonuna yönelik çözüm için genetik algoritma kullanmıştır. Bu çözümler arasındaki ilişkileri ortaya çıkarmak için ise veri madenciliğinden faydalanılmıştır. Veri madenciliği analizi sonucunda problem türlerine uygun, genetik algoritmadan elde edilen çözümlere göre çeşitli kurallar elde edilmiştir. Önceden bilinmeyen ancak potansiyel olarak kullanışlı olan bu kurallardan elde edilecek sonuçlar ile farklı tekniklerin birlikte kullanılmasıyla iyi çözümler elde edilebilmesinin mümkün olduğu ifade edilmiştir [112].

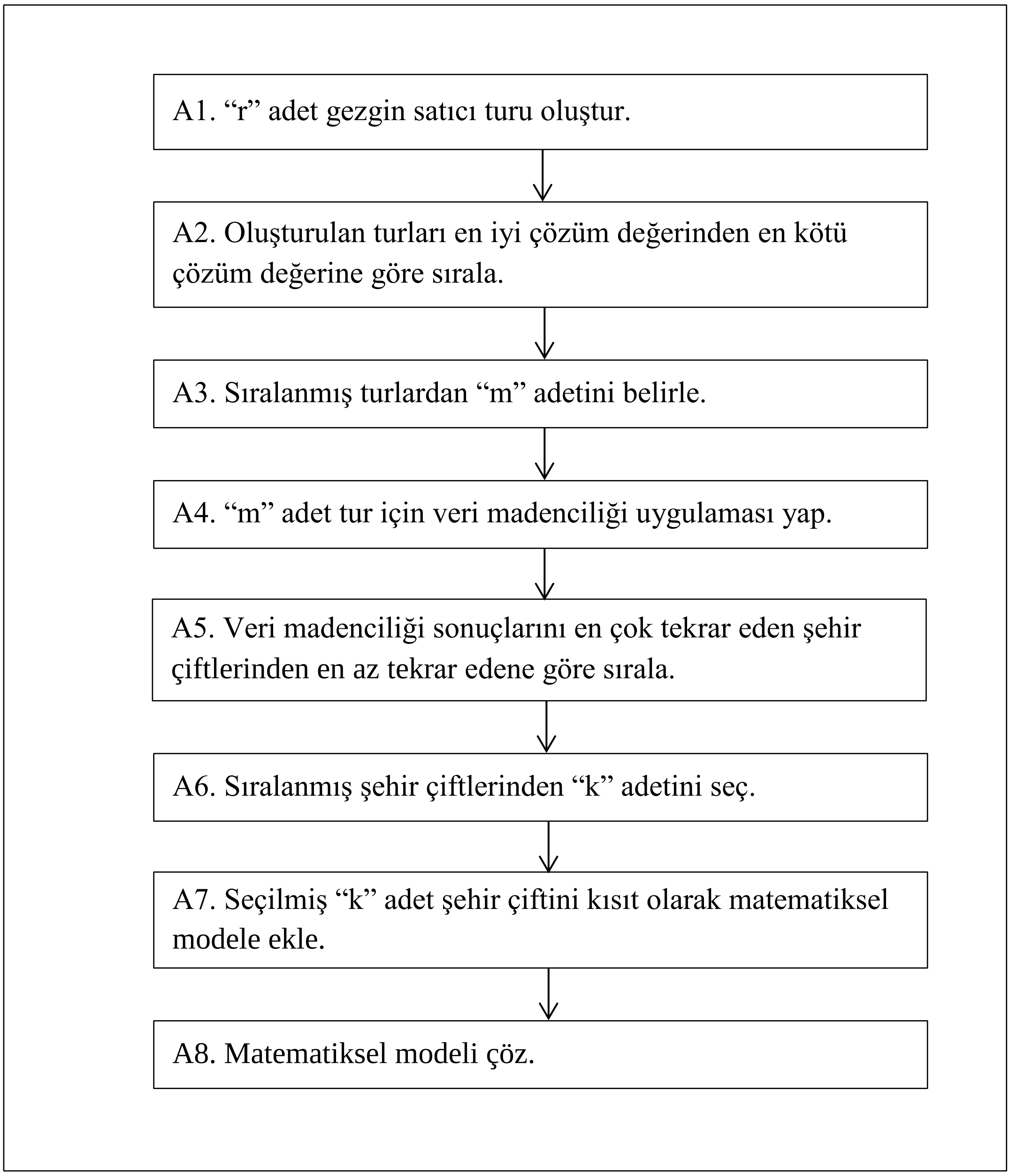
Olafsoon ve Li, tek makine problemini; geliş zamanı, işlem süresi, işlerin ağırlığı, teslim zamanı kriterlerinden yararlanarak hem veri madenciliğinde kullanılan klasik karar ağacı ile hem de geliştirilmiş karar ağacı yapısı ile çözmüşlerdir. Sonuç olarak, klasik karar ağacı yaklaşımından daha iyi sonuçlar elde etmişlerdir [113].

3. VERİ MADENCİLİĞİ TABANLI MODEL ÖNERİSİ

Gezgin satıcı probleminin çözümü sonucunda, bir başlangıç şehirden başlayıp mevcut tüm şehirlere sadece bir kez uğrayan ve başlangıç şehrine dönen en kısa uzunluktaki bir tur elde edilmektedir. Elde edilen bu turun tam sayılı kesin çözümünün bulunması, problemi oluşturan şehir sayısı arttıkça daha da zor bir hale gelmektedir. Bu zorluk hem çözümün her iterasyonunda yeni alt turların ortaya çıkması ve bu alt turları engellemek için yeni kısıtların çözüm modeline eklenmesinden hem de çözüm süresinin artışından kaynaklanmaktadır. Tez çalışması kapsamında, kesin çözüme ulaşmada karşılaşılan bu zorlukların üstesinden gelebilmek için veri madenciliği temeline dayanan bir çözüm önerisi geliştirilmiştir.

Önerilen çözüm önerisi ile veri madenciliğinin büyük veri setleri içinde saklı kalmış ilişkileri ortaya çıkarabilme yeteneğinden faydalanmak amaçlanmıştır. Bölüm 2.4.3'de anlatılan veri madenciliği tekniklerinden ardışık zamanlı örüntüler tekniğine benzer bir yaklaşım olan ASIPATH algoritması ile rassal olarak oluşturulmuş gezgin satıcı turları içinde en çok tekrar eden şehir çiftleri belirlenmiştir. Bu şehir çiftlerinin başlangıç matematiksel modelde kısıt olarak kullanılması ile çözüm kalitesini mümkün olduğunca koruyarak çözüm süresinin azaltılmasına çalışılmıştır.

Gezgin satıcı probleminin kesin çözümüne yönelik olarak önerilen çözüm yöntemi 3 aşamadan oluşmaktadır. İlk aşamada gezgin satıcı turları rassal olarak üretilmekte, ikinci aşamada üretilen rassal turlardan ASIPATH algoritması ile çok tekrar eden şehir çiftleri belirlenmektedir. Son aşamada ise belirlenen bu şehir çiftleri matematiksel modele kısıt olarak eklenerek gezgin satıcı probleminin çözümü gerçekleştirilmektedir. Önerilen çözüm yaklaşımının akış şeması Şekil 3.1'de görülmektedir.



Şekil 3.1. Önerilen çözüm yaklaşımının akış şeması

Önerilen çözüm yaklaşımı Şekil 3.1.’de görüldüğü gibi 8 adımdan oluşmaktadır. Önerilen yaklaşım için; şekilde A1 adımı olarak görülen 1. adımda çözümü gerçekleştirilecek problemdeki şehir sayısına bağlı olarak “r” adet rassal gezgin satıcı turu oluşturulmakta ve bu turların toplam mesafesi hesaplanmaktadır. Oluşturulacak “r” adetlik rassal gezgin satıcı turu sayısının belirlenmesinde çözümü gerçekleştirilecek tüm problemler için ortak bir anlam taşıması amacıyla, “N” şehir sayısını göstermek üzere, “r = X N” eşitliği kullanılmıştır. Bu eşitlikteki “X” katsayısının ne olacağı kararı için tam faktöriyel deney tasarımı yapılmıştır.

Önerilen yaklaşımın A2 adımında, “r” adet olarak rassal olarak üretilen ve tur uzunluğu hesaplanan gezgin satıcı turları en iyi (en kısa) tur uzunluğundan en kötü (en uzun) tur uzunluğuna doğru sıralanmıştır.

Önerilen yaklaşımın A3 adımında, tur uzunluğu değerine göre en iyiden en kötüye sıralanan gezgin satıcı turları içinden en iyiden başlamak üzere “m” adeti alınmıştır. Buradaki “m” adet alınacak iyi tur sayısı değerini belirlemek için, yine “N” şehir sayısını göstermek üzere, “m = Y N” eşitliği kullanılmıştır. Yine bu eşitlikteki “Y” katsayısının ne olacağı kararı için tam faktöriyel deney tasarımı yapılmıştır.

Önerilen yaklaşımın A4 adımı için, A3 adımında “m” adet olarak belirlenmiş en iyi gezgin satıcı turları bir veri kümesi oluşturmuştur. Bu veri kümesi içinden, bir veri madenciliği yaklaşımı olan ASIPATH algoritması ile “m” adet gezgin satıcı turu içinde en çok tekrar eden şehir çiftleri belirlenmiştir.

Önerilen yaklaşımın A5 adımında, gerçekleştirilen veri madenciliği işlemi sonucunda elde edilen en çok tekrar eden şehir çiftlerinin en çok tekrar edenden en az tekrar edene göre sıralanması yapılmıştır. Böylelikle rassal olarak oluşturulan gezgin satıcı turları içinde en çok tekrar eden şehir çiftleri en çok tekrar edenden en az tekrar edene göre ortaya çıkarılmıştır.

Önerilen yaklaşımın A6 adımında, veri madenciliği yöntemi ile elde edilerek en çok tekrar edenden en az tekrar edene göre sıralanmış olan şehir çiftleri arasından “k” adedi seçilmiştir. “k” adet şehir çifti seçim işlemi gerçekleştirilirken sıralamada ilk olarak görülen, yani en çok tekrar ettiği tespit edilen şehir çifti ilk olarak alınmış sonrasında bu şehir çiftindeki şehirlerden herhangi birini içermeyen diğer şehir çifti aranmıştır. Örneğin en çok tekrar eden şehir çiftinin (1, 9) olarak belirlendiği durumda, ikinci olarak tercih edilecek şehir çifti “1” ve “9” şehirlerini içermeyen bir diğer şehir çifti olacaktır. Bu değerlendirme doğrultusunda “k” değerinin ne olacağı kararı için tam faktöriyel deney tasarımı yapılmıştır.

Önerilen yaklaşımın A7 adımında, A6 adımında belirlenmiş olan “k” adetlik şehir çifti gezgin satıcı probleminin kesin çözümünü elde etmek amacıyla sırasıyla matematiksel modele kısıt olarak eklenmiştir. Böylelikle matematiksel modelin daha etkin ve daha kısa zamanda bir çözüme ulaşması amaçlanmıştır.

Önerilen yaklaşımın son adımı A8'de ise, veri madenciliği tekniği ile belirlenmiş olan kısıtların eklendiği matematiksel model gezgin satıcı probleminin kesin çözümünü bulmak için çözülmüştür. Çözüm sonuçları; hesaplanan tur uzunluğu değeri ve çözüm süresi olarak elde edilmiştir.

3.1. Deneysel Çalışma

Gezgin satıcı probleminin kesin çözümüne yönelik olarak önerilen çözüm yönteminin adımlarında kullanılacak olan gerekli parametre değerlerinin nasıl belirleneceğini ve kullanılacağını belirlemek amacıyla tam faktöriyel deney tasarımı gerçekleştirilmiştir.

A1 adımındaki oluşturulacak olan rassal gezgin satıcı turu sayısı "r"nin "N" şehir sayısını göstermek üzere, "r = X N" eşitliği ile belirlenmesinin uygun olacağı düşünülmüştür. Buradaki X katsayısının deney tasarımı ile belirlenmesi aşamasında 2500, 5000, 10000 ve 20000 olmak üzere 4 farklı seviyeden incelenmesine karar verilmiştir. Dolayısıyla üretilecek rassal tur sayısı 4 seviye için sırasıyla 250000, 500000, 1000000 ve 2000000 olacaktır.

A3 adımındaki seçilecek olan "m" adetlik en iyi tur sayısının yine "N" şehir sayısını göstermek üzere, "m = Y N" eşitliği ile belirlenmesine karar verilmiştir. Buradaki Y katsayısının deney tasarımı ile belirlenmesi aşamasında 500 ve 1000 olarak 2 farklı seviyede incelenmesi uygun olarak değerlendirilmiştir. Daha fazla sayıda seçilecek iyi tur sayısı, çözüm kalitesini kötüleştirebilecek şehir çiftlerini de veri madenciliği sürecine dahil edeceği için bu aşamada incelenecek seviyenin 2 olması yeterli görülmüştür. Burada da "Y"ye bağlı olarak 50000 ve 100000 adetlik iyi tur sayısı deney tasarımı sürecinde incelenmiştir.

A6 adımında ise, veri madenciliğinden elde edilen sonuçlardan "k" adetlik şehir çiftlerinin belirlenmesi gerekmektedir. "k" adetlik şehir çiftleri sayısının deney tasarımında belirlenmesinde 3 seviye göz önünde bulundurulmuştur. Matematiksel modele kısıt olarak eklenecek bu 3 seviye şehir çifti sayısı 5, 10 ve 15 adet olarak incelenmiştir.

Yukarıda sayılan 3 faktör ve bu faktörlerin farklı seviyeleri için 5 tekrarlı tam faktöriyel deney tasarımı gerçekleştirilmiştir. Bu durumda toplamda 120 adet deneme yapılmış ve sonuçları incelenmiştir. Deneyler Tablo 3.1'de özetlenmiştir.

Tablo 3.1. Önerilen çözüm yöntemi için deneyler

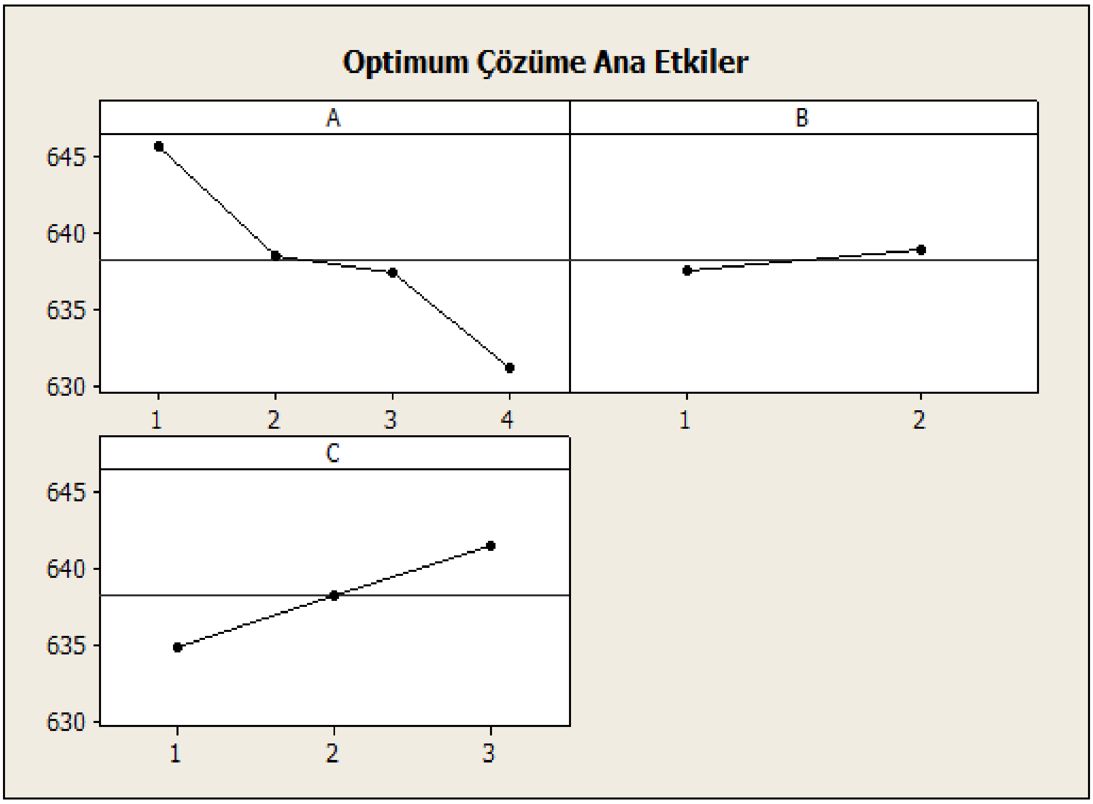
Faktör Adı	Seviyeler	Sayı
Üretilen Rassal Tur Sayısı (r = X N)	250000 500000 1000000 2000000	4
Alınan İyi Çözüm Sayısı (m = Y N)	50000 100000	2
Belirlenen Şehir Çifti Sayısı	5 10 15	3
Tekrar Sayısı		5
Toplam	(4) (2) (3) (5)	120

Deneyler literatürde tanımlanmış orta büyüklükteki bir gezgin satıcı problemi olan ve 101 şehirden oluşan eil101 problemi üzerinde gerçekleştirilmiştir. Tablo 3.1'de sıralanmış olan faktörlerin her bir seviyesinin farklı kombinasyonu için 5 farklı rassal sayı kökü üzerinden elde edilen değerlere göre eil101 problemi çözülerek toplamda 120 adet problemin çözüm sonucuna ve çözüm süresine ulaşılmıştır.

Örnek gezgin satıcı problemi için deney tasarımı; matematiksel modelin optimum çözümü bulması ya da optimum çözümden en az sapmayı sağlaması ve matematiksel modelin çözümü için geçen süre kriterleri için ayrı ayrı gerçekleştirilmiştir. Bu kriterler için deney tasarımı sonuçları sırasıyla Tablo 3.2 ve Şekil 3.2 ile Tablo 3.3 ve Şekil 3.3'de görülmektedir.

Tablo 3.2. eil101 problemi optimum çözüm kriteri deney tasarımı sonuçları

Kaynak	Kareler Toplamı	Serbestlik Derecesi	Kareler Ortalaması	F	p
Üretilen Rassal Tur Sayısı (A)	3121,36	3	1040,45	27,09	0,0
Alınan İyi Çözüm Sayısı (B)	46,87	1	46,87	1,22	0,272
Belirlenen Şehir Çifti Sayısı (C)	891,45	2	445,72	11,60	0,0
A B	42,29	3	14,10	0,37	0,77
A C	157,02	6	26,17	0,68	0,665
B C	17,15	2	8,58	0,22	0,80
A B C	59,58	6	9,93	0,26	0,955
Hata	3687,20	96	38,41		
Toplam	8022,92	119			



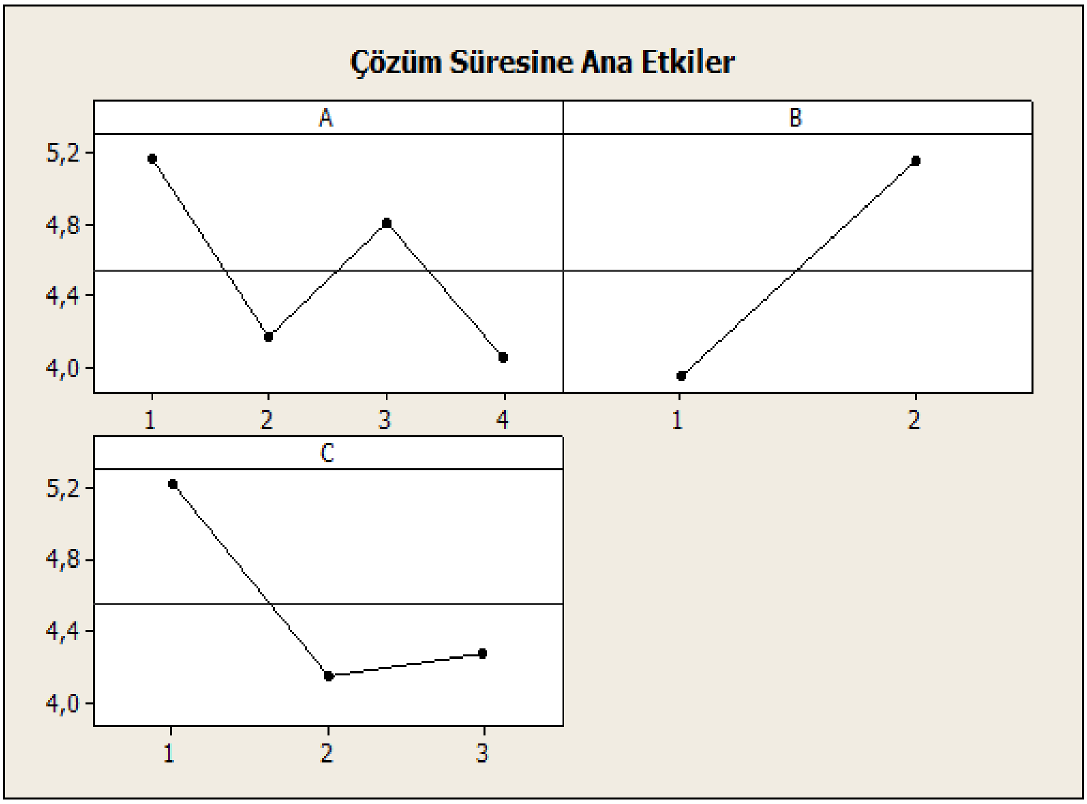
Şekil 3.2. Optimum çözüm kriteri faktör seviyeleri

Matematiksel modelin optimum çözümü bulması yada optimum çözümden en az sapmayı sağlaması kriterine göre, Tablo 3.2’de görülen p değerleri, faktör etkilerinin rasgele elde edilme olasılığını göstermektedir. %95 güvenilirlikle değerlendirildiğinde, p değerleri α=0.05 değerinden küçük olan “üretilen rassal tur sayısı” faktörünün ve “belirlenen şehir çifti sayısı” faktörünün birincil etkilerinin önemli olduğu görülmektedir. Tekil etkilerde yine Tablo 3.2’de görülen Kareler Ortalaması ya da F değerleri göz önüne alındığında, sonuç üzerinde en yüksek etkiye, en büyük değere sahip “üretilen rassal tur sayısı” faktörünün sahip olduğu görülmektedir. İkinci sırada ise “belirlenen şehir çifti sayısı” faktörü gelmektedir. İkili etkiler açısından bakıldığında ise hiçbir faktörün p değeri α=0.05 değerinden küçük olmadığı için faktörlerin etkileşimlerinin bir etkisi olmadığı anlaşılmaktadır.

Şekil 3.2’de ise faktörlerin hangi seviyelerinin çözüm üzerinde etkili olduğu anlaşılmaktadır. Gerçekleştirilen deney tasarımı sonuçlarına göre; matematiksel modelin optimum çözümü bulması yada optimum çözümden en az sapmayı sağlaması kriterine göre “üretilen rassal tur sayısı” faktörünün seviyesi 2000000 adet olarak yani “20000 N” olarak belirlenmiştir. “Alınan iyi çözüm sayısı” faktörünün seviyesi 50000 adet yani “500 N” olarak ortaya çıkmıştır. “Belirlenen şehir çifti sayısı” faktörünün seviyesi ise 5 olarak görülmüştür. Bu sonuçlardan hareketle yapılacak problem denemeleri için optimum çözüme ulaşabilmek amacıyla üretilecek rassal tur sayısı “20000 N”, alınacak iyi çözüm sayısı “500 N” ve belirlenen şehir çifti sayısı “5” olarak genelleştirilmiştir.

Tablo 3.3. eil101 problemi çözüm süresi kriteri deney tasarımı sonuçları

Kaynak	Kareler Toplamı	Serbestlik Derecesi	Kareler Ortalaması	F	p
Üretilen Rassal Tur Sayısı (A)	25,185	3	8,395	13,31	0,0
Alınan İyi Çözüm Sayısı (B)	43,681	1	43,681	6,82	0,010
Belirlenen Şehir Çifti Sayısı (C)	27,837	2	13,919	21,7	0,0
A B	89,646	3	29,882	0,87	0,32
A C	21,024	6	3,504	0,55	0,77
B C	7,429	2	3,715	0,58	0,56
A B C	50,703	6	8,451	1,32	0,25
Hata	614,796	96	6,404		
Toplam	880,302	119			



Şekil 3.3. Çözüm süresi kriteri faktör seviyeleri

Matematiksel modelin çözümü için geçen süre kriterine göre, Tablo 3.3 %95 güvenilirlikle değerlendirildiğinde, p değerleri α=0.05 değerinden küçük olan "üretilen rassal tur sayısı" faktörünün ve "belirlenen şehir çifti sayısı" faktörünün birincil etkilerinin önemli olduğu görülmektedir. Tekil etkilerde yine Tablo 3.3'de görülen Kareler Ortalaması ya da F değerleri göz önüne alındığında, sonuç üzerinde en yüksek etkiye, en büyük değere sahip "belirlenen şehir çifti sayısı" faktörünün sahip olduğu görülmektedir. İkinci sırada ise "üretilen rassal tur sayısı" faktörü gelmektedir. İkili etkiler açısından bakıldığında ise hiçbir faktörün p değeri α=0.05 değerinden küçük olmadığı için faktörlerin etkileşimlerinin bir etkisi olmadığı anlaşılmaktadır.

Şekil 3.3'de ise faktörlerin hangi seviyelerinin çözüm üzerinde etkili olduğu anlaşılmaktadır. Gerçekleştirilen deney tasarımı sonuçlarına göre; matematiksel modelin çözümü için geçen süre kriterinin göz önünde bulundurulduğu durumda ise, "üretilen rassal tur sayısı" faktörünün seviyesi 2000000 adet olarak yani "20000 N" olarak görülmüştür. "Alınan iyi çözüm sayısı" faktörünün seviyesi 50000 adet yani "500 N" olarak ortaya çıkmıştır. "Belirlenen şehir çifti sayısı" faktörünün seviyesi ise burada farklılık göstererek 10 olarak ortaya çıkmıştır. Bu sonuçlardan hareketle yapılacak problem denemeleri için çözüme ulaşabilmek için geçen süre dikkate alınmak istendiğinde, üretilecek rassal tur sayısı “20000 N”, alınacak iyi çözüm sayısı “500 N” ve belirlenen şehir çifti sayısı “10” olarak genelleştirilmiştir.

3.2. Literatür Problemleri Denemeleri

Gezgin satıcı probleminin kesin çözümüne yönelik olarak veri madenciliği temeline dayanan yeni model önerisinin başarımını ortaya koyabilmek adına literatür problemleri üzerinde önerilen çözüm yöntemi denenmiştir. Denemesi yapılacak problemler, gezgin satıcı problemi için geniş bir kaynak imkanı sağlayan TSPLIB’den seçilmiştir. Çözümü gerçekleştirilen problemler için “üretilen rassal tur sayısı”, “alınan iyi çözüm sayısı” ve “belirlenen şehir çifti sayısı” değerleri Tablo 3.4’de görülmektedir.

Bölüm 3.1’de açıklanan ve en iyi seviyeleri belirlenen faktörler göz önünde bulundurularak problemlerin çözümleri gerçekleştirilmiştir. Matematiksel modelin optimum çözümü bulması ya da optimum çözümden en az sapmayı sağlaması kriterine göre elde edilen çözüm değerleri Tablo 3.5’de, matematiksel modelin çözümü için geçen süre kriterine göre elde edilen çözüm değerleri Tablo 3.6’da görülmektedir.

Tablo 3.4. Literatür problemleri için parametre değerleri

Problem Adı	Üretilen Rassal Tur Sayısı (20000 N)	Alınan İyi Çözüm Sayısı (500 N)	Belirlenen Şehir Çifti Sayısı
ulysses16	320000	8000	5 ve 10
gr17	340000	8500	5 ve 10
gr21	420000	10500	5 ve 10
ulysses22	440000	11000	5 ve 10
gr24	480000	12000	5 ve 10
fri26	520000	13000	5 ve 10
bays29	580000	14500	5 ve 10
bayg29	580000	14500	5 ve 10
dantzig42	840000	21000	5 ve 10
swiss42	840000	21000	5 ve 10
gr48	960000	24000	5 ve 10
eil51	1020000	25500	5 ve 10
berlin52	1040000	26000	5 ve 10
brazil58	1160000	29000	5 ve 10
st70	1400000	35000	5 ve 10
eil76	1520000	38000	5 ve 10
pr76	1520000	38000	5 ve 10
kroA100	2000000	50000	5 ve 10
kroB100	2000000	50000	5 ve 10
kroC100	2000000	50000	5 ve 10
kroD100	2000000	50000	5 ve 10
rd100	2000000	50000	5 ve 10
eil101	2020000	50500	5 ve 10
lin105	2100000	52500	5 ve 10
gr120	2400000	60000	5 ve 10
kroA150	3000000	75000	5 ve 10
kroB150	3000000	75000	5 ve 10
brg180	3600000	90000	5 ve 10
kroA200	4000000	100000	5 ve 10
kroB200	4000000	100000	5 ve 10
a280	5600000	140000	5 ve 10

Tablo 3.5. Literatür problemlerinin optimum çözüme göre sonuç değerleri

Problem Adı	Optimum Çözüm		Önerilen Çözüm ile		Sapma Sonuç (%)	Sapma Süre (%)
	Sonuç	Süre	Sonuç	Süre		
ulysses16	6859	4,18	6859	1,86	0	-55,5
gr17	2085	0,91	2146	0,34	2,93	-62,64
gr21	2707	0,1	2758	0,06	1,88	-40
ulysses22	7013	185,11	7013	19,84	0	-89,28
gr24	1272	0,05	1272	0,03	0	-57,14
fri26	937	0,2	937	0,15	0	-25
bays29	2020	0,24	2022	0,19	0,1	-20,83
bayg29	1610	0,23	1610	0,17	0	-26,09
dantzig42	699	2,37	699	0,81	0	-65,82
swiss42	1273	0,78	1273	0,51	0	-34,62
gr48	5046	8,58	5055	3,57	0,18	-58,39
eil51	426	0,56	426	0,29	0	-48,21
berlin52	7542	0,49	7618	0,42	1,01	-14,29
brazil58	25395	37,73	25395	17,36	0	-53,99
st70	675	12,43	675	8,19	0	-34,11
eil76	538	2,32	543	1,77	0,93	-23,71
pr76	108159	66,9	108159	43,07	0	-35,62
kroA100	21282	45,25	21466	24,88	0,86	-45,02
kroB100	22141	4840,1	22141	599,45	0	-87,61
kroC100	20749	38,5	20749	36,42	0	-5,4
kroD100	21294	1901	21294	302,9	0	-84,07
rd100	7910	29,54	7935	21,43	0,32	-27,45
eil101	629	4,63	629	2,79	0	-39,74
lin105	14379	58,87	14379	53,78	0	-8,65
gr120	6942	64,3	6945	38,98	0,04	-39,38
kroA150	26524	1385	26893	65,39	1,39	-95,28
kroB150	26130	40000	26287	8971,3	0,6	-77,57
brg180	1950	77,92	1970	10,53	1,03	-86,49
kroA200	29368	53094	29375	37672	0,02	-29,05
kroB200	29437	31535	30025	3489,9	2	-88,93
a280	2579	5870,8	2660	4418,7	3,14	-24,73

Tablo 3.6. Literatür problemlerinin çözüm süresine göre sonuç değerleri

Problem Adı	Optimum Çözüm		Önerilen Çözüm ile		Sapma Süre (%)	Sapma Sonuç (%)
	Sonuç	Süre	Sonuç	Süre		
ulysses16	6859	4,18	7122	0,88	-78,95	3,83
gr17	2085	0,91	2159	0,18	-80,22	3,55
gr21	2707	0,1	2803	0,03	-70	3,55
ulysses22	7013	185,11	7085	19,44	-89,5	1,03
gr24	1272	0,05	1320	0,06	-14,29	3,77
fri26	937	0,2	937	0,09	-55	0
bays29	2020	0,24	2054	0,22	-8,33	1,68
bayg29	1610	0,23	1630	0,15	-34,78	1,24
dantzig42	699	2,37	699	0,56	-76,37	0
swiss42	1273	0,78	1273	0,46	-41,03	0
gr48	5046	8,58	5055	2,21	-74,24	0,18
eil51	426	0,56	433	0,15	-73,21	1,64
berlin52	7542	0,49	7618	0,28	-42,86	1,01
brazil58	25395	37,73	25793	15,27	-59,53	1,57
st70	675	12,43	675	10,4	-16,33	0
eil76	538	2,32	544	1,24	-46,55	1,12
pr76	108159	66,9	110337	25,55	-61,81	2,01
kroA100	21282	45,25	21466	23,65	-47,73	0,86
kroB100	22141	4840,1	22141	485,67	-89,97	0
kroC100	20749	38,5	20749	29,04	-24,57	0
kroD100	21294	1901	21309	401,26	-78,89	0,07
rd100	7910	29,54	7975	13,35	-54,81	0,82
eil101	629	4,63	630	1,98	-57,24	0,16
lin105	14379	58,87	14429	45,98	-21,9	0,35
gr120	6942	64,3	6945	27,95	-56,53	0,04
kroA150	26524	1385	26929	46,69	-96,63	1,53
kroB150	26130	40000	26482	8649,7	-78,38	1,35
brg180	1950	77,92	2010	4,18	-94,64	3,08
kroA200	29368	53094	29428	27280	-48,62	0,2
kroB200	29437	31535	30077	2531,1	-91,97	2,17
a280	2579	5870,8	2673	1451,8	-75,27	3,64

SONUÇLAR ve ÖNERİLER

Gezgin satıcı problemi, kombinasyonel optimizasyon problemleri arasında oldukça popüler olan NP zor sınıfında bir problemdir. Problemin basit olarak tanımlanması ve modellenebilmesi, gerçek yaşamda karşılaşılan birçok problemin küçük eklemeler ile bu problem yapısı üzerinden ifade edilebilmesi bu problemi tanımlanmasını üzerinden uzun zaman geçmesine rağmen popüler kılmaktadır. Bunların yanında, gezgin satıcı probleminin ve/veya bu probleme benzetilerek geliştirilen problemlerin kesin çözümünü elde etmek gelişen teknoloji ile daha kısa zamanlarda yapılabiliyor olmasına rağmen, kesin çözüme ulaşabilme süresi problemin NP zor olmasından ötürü zaman almaktadır.

Veri madenciliği yöntemi, büyük veri yığınları arasından anlamlı ilişkiler bulmaya ve bu ilişkiler yardımıyla da çıkarımlar yapmaya imkan sağlayan bir yöntemdir. Sağladığı bu avantajlı durum ile gerçek yaşamda büyük veriler içinde saklı olan bilgilere ışık tutarak hem rekabetçi bir gelişim sağlanmasına hem de zamandan büyük oranda tasarruf edilebilmesine fırsat sağlamaktadır.

Bu tez çalışması kapsamında, veri madenciliğinin ortaya koyduğu bu avantajlı durum gezgin satıcı probleminin kesin çözümünün elde edilmesinde kullanılmıştır. Literatüre bakıldığında veri madenciliği yöntemlerinin optimizasyon alanında çok sınırlı olarak kullanıldığı görülmektedir. Üstelik gezgin satıcı probleminin kesin çözümüne yönelik olarak veri madenciliği yaklaşımına dayanan hiçbir çalışmaya rastlanmamıştır. Bu yönüyle tez çalışması yenilikçi bir bakış açısı getirmektedir.

Çalışma kapsamında, önerilen yöntemin etkinliği gezgin satıcı problemleri ile ilgili literatürdeki test problemleri ile sınanmıştır. Problemlerin çözümünde kullanılacak yöntemin içerdiği üretilecek rassal tur sayısı, alınacak iyi çözüm sayısı ve belirlenecek şehir çifti sayısı faktörlerinin seviyelerinin ne olacağı kararı tam faktöriyel deney tasarımı ile belirlenmiştir. Faktörlerin seviyelerinin belirlenmesinden sonra belirlenen literatür problemleri çözülmüş ve sonuçları elde edilmiştir.

Matematiksel modelin optimum çözümü bulması ya da optimum çözümden en az sapmayı sağlaması kriterine göre elde edilen çözüm değerleri incelendiğinde; problemlerin %52'sinde optimum çözüm elde edilirken çözüm zamanında %5 ile %90 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %26'sında optimum çözümden %1'in altında bir sapma ile kesin çözüm bulunurken çözüm zamanında %20 ile %78 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %13'ünde optimum çözümden %1 ile %2 aralığında bir sapma ile kesin çözüm bulunurken çözüm zamanında %14 ile %96 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %6'sında optimum çözümden %2 ile %3 aralığında bir sapma ile kesin çözüm bulunurken çözüm zamanında %62 ile %89 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin sadece bir tanesinde, %3'ünde, optimum çözümden %3,14'lük bir sapma ile kesin çözüm bulunurken çözüm zamanında %24 oranında bir zaman tasarrufu sağlanmıştır.

Matematiksel modelin çözümü için geçen süre kriterine göre elde edilen çözüm değerleri incelendiğinde; problemlerin %19'unda optimum çözüm elde edilirken çözüm zamanında %16 ile %90 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %26'sında optimum çözümden %1'in altında bir sapma ile kesin çözüm bulunurken çözüm zamanında %21 ile %79 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %29'unda optimum çözümden %1 ile %2 aralığında bir sapma ile kesin çözüm bulunurken çözüm zamanında %8 ile %97 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %6'sında optimum çözümden %2 ile %3 aralığında bir sapma ile kesin çözüm bulunurken çözüm zamanında %62 ile %92 arasında değişen bir zaman tasarrufu sağlanmıştır. Problemlerin %19'unda optimum çözümden %3 ile %3,83 aralığında bir sapma ile kesin çözüm bulunurken çözüm zamanında %14 ile %95 arasında değişen bir zaman tasarrufu sağlanmıştır.

Tüm çözümler açısından bakıldığında, önerilen çözüm yaklaşımının optimum çözümden en fazla %3,83 kadar saptığı ve bu sapmada çözüm süresinden %79'a yakın tasarruf sağladığı görülmektedir. Bunun yanında çözüm zamanından optimum çözüm süresine göre en yüksek %96,63'lük bir tasarruf sağlanırken bu durumda optimum çözümden %3,08'lik bir sapma meydana geldiği görülmektedir. Bu tasarruf oranları optimum çözümden %3'ler civarında sapmalarla meydana geliyor olmasına rağmen, çözüm süresi dikkate alındığında oldukça iyi bir iyileştirme olarak görülmektedir. Ayrıca, önerilen çözüm yaklaşımının ağır matematiksel işlemlere gerek duymaksızın, oldukça basit sayılabilecek bir yöntem ile bu kadar etkili sonuçlar üretmesi de bir başarı ölçütü olarak değerlendirilmektedir.

Bu çalışma kapsamında veri madenciliği teknik ve algoritmalarından optimizasyon alanında etkili bir şekilde faydalanılabileceği gösterilmiştir. Optimizasyon çalışmalarında oldukça geniş bir uygulama alanı bulan sezgisel ve metasezgisel yöntemlerin veri madenciliği yöntemleri ile desteklenmesi, entegre çözüm yöntemi olarak birlikte kullanılması yoluyla daha da etkin sonuçlar alınabileceği düşünülmekte ve bu alan üzerinde çalışmalar yapılması önerilmektedir.

Ayrıca, önerilen çözüm yaklaşımının, kombinasyonel optimizasyon alanındaki diğer problem türlerine göre dizayn edilip, bu problemlerin çözümünde de kullanılması gelecekte yapılacak çalışmalara bırakılmıştır.

KAYNAKLAR

[1] Applegate D. L., Bixby R. E., Chvatal V., Cook W. J., *The Traveling Salesman Problem: A Computational Study,* 1st ed., Princeton University Press, New Jersey, 2007.

[2] Terzi Ü., Gezgin satıcı problemi için diferansiyel gelişim algoritması tabanlı bir metasezgisel önerisi, Doktora Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2009, 262170.

[3] Blaser M., Manthey B., Sgall J., An improved approximation algorithm for the asymmetric TSP with strengthened triangle inequality, *Journal of Discrete Algorithms*, 2006, **4**, 623-632.

[4] Kwon S. H., Kim H. T., Kang M. K., Determination of the candidate arc set for the asymmetric traveling salesman problem, *Computers & Operations Research*, 2005, **32**, 1045-1057.

[5] Pearn W. L., Chien R. C., Improved solutions for the traveling purchaser problem, *Computers & Operations Research*, 1998, **25**, 879-885.

[6] Bektaş T., The multiple traveling salesman problem: an overview of formulations and solution procedures, *The International Journal of Management Science*, 2006, **34**, 209-219.

[7] http://www.nist.gov/dads/HTML/chinesePostman.html, (Ziyaret Tarihi: 22 Şubat 2014).

[8] Gutin G., Punnen A. P., *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers, Boston, 2002.

[9] Foulds L. R., *Combinatorial Optimization for Undergraduates,* Springer Verlag, Berlin, 1984.

[10] DePuy G. W., Moraga R. J., Whitehouse G. E., Meta - RaPS: a simple and effective approach for solving the traveling salesman problem, *Transportation Research Part E: Logistics and Transportation Review*, 2005, **41**, 115-130.

[11] http://en.wikipedia.org/wiki/Christofides_heuristic, (Ziyaret Tarihi: 21 Nisan 2013).

[12] Engebretsen L., Karpinski M., TSP with bounded metrics, *Journal of Computer and System Sciences*, 2006, **72**, 509-546.

[13] Lin S., Kerninghan B., An effective heuristic algorithm for the traveling salesman problem, *Operations Research*, 1973, **21**, 498-516.

[14] Helsgaun K., An effective implementation of the lin-kernighan traveling salesman heuristic, *European Journal of Operational Research,* 2000, **126**, 106-130.

[15] Colorni A., Dorigo M., Maniezzo V., Distributed optimization by ant colonies, *First European Conference on Artificial Life,* Paris, France, 11-13 December 1991.

[16] Dorigo M., Maniezzo V., Colorni A., The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics – Part B,* 1996, **26**, 29-42.

[17] Stützle T., Dorigo M, ACO algorithms for the traveling salesman problem, *UniversitÃ Libre de Bruxelles,* Tech.Rep.IRIDIA/99-2, 1-18, 1999.

[18] Uğur A., Aydın D., Ant sistem algoritmasının java ile görselleştirilmesi, *Yöneylem Araştırması/Endüstri Mühendisliği XXIV. Kongresi,* Adana, Türkiye, 15-18 Haziran 2004.

[19] Dorigo M., Gambardella L. M., Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation,* 1997, **1**, 53-66.

[20] Gambardella L. M., Dorigo M, Solving symmetric and asymmetric TSPs by ant colonies, *IEEE International Conference on Evolutionary Computation,* Japan, 20-22 May 1996.

[21] Dorigo M., Gambardella L. M., Ant colonies for the travelling salesman problem, *BioSystems,* 1997, **43**, 73-81.

[22] Stützle T., Hoos H. H., The MAX–MIN ant system and local Search for the traveling salesman problem, *IEEE International Conference on Evolutionary Computation,* Indianapolis, USA, 13-16 April 1997.

[23] Stützle T., Hoos H. H., Improvements on the ant system: introducing the MAX–MIN, Editors: Albrecht, R.F., Smith, G. D., Steele N. C., *Ant system artificial neural networks and genetic algorithms,* 1st ed., Springer Verlag, Vienna, 245–249, 1998.

[24] Stützle T., Hoos H. H., The max–min ant system, *Future Generation Computer Systems,* 2000, **16**, 889-914.

[25] Bai J., Yang G., Chen Y., Hu L., Pan C., A model induced max-min ant colony optimization for asymmetric traveling salesman problem, *Applied Soft Computing,* 2013, **13**, 1365-1375.

[26] Randall M., Lewis A., A parallel implementation of ant colony optimization, *Journal of Parallel and Distributed Computing,* 2002, **62**, 1421–1432.

[27] Chu S. C., Roddick J. F., Pan J. S., Ant colony system with communication strategies, *Information Sciences,* 2004, **167**, 63–76.

[28] Tsai C. F., Tsai C. W., Tseng C. C., A new hybrid heuristic approach for solving large traveling salesman problem, *Information Sciences,* 2004, **166**, 67-81.

[29] Çolak S., Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama, *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi,* 2010, **19**, 423-438.

[30] Whitley D., Starkweather T., Shaner D., Scheduling problems and traveling salesmen: the genetic edge recombination operator, *Third Int. Conf. On Genetic Algorithms*, Virginia, USA, June 1989.

[31] Bean J., Genetics and random keys for sequencing and optimization, *ORSA Journal on Computing,* 1994, **6**, 154-160.

[32] Lin Z., Li Y., Huang Z. C., Genetic algorithm based on classification for the traveling salesman problem, *Third International Conference on Natural Computation*, Hainan, China, 24-27 August 2007.

[33] Martikainen J., Ovaska S. J., Hierarchical two-population genetic algorithm, *Int. J. of Computational Intelligence Research*, 2006, **2**, 367-380.

[34] Wang L., Maciejewski A. A., Siegel H. J., Roychowdhury V. P., A comparative study of five parallel genetic algorithms using the traveling salesman problem, *12th. International Parallel Processing Symposium on International Parallel Processing Symposium,* Florida, USA, 30 March-3 April 1998.

[35] Katayama K., Hirabayashi H., Narihisa H., Analysis of crossovers and selections in a coarse-grained paralel genetic algorithm, *Mathematical and Computer Modelling*, 2003, **38**, 1275-1282.

[36] Sena G. A., Megherbi D., Isern G., Implementation of a parallel genetic algorithm on a cluster of workstations: traveling salesman problem, a case study, *Future Generation Computer Systems*, 2001, **17**, 477-488.

[37] Chen S., Chien C., Parallelized genetic ant colony systems for solving the traveling salesman problem, *Expert System with Applications,* 2011, **38**, 3873-3883.

[38] Pongcharoen P., Chainate W., Thapatsuwan P., Exploration of genetic parameters and operators through travelling salesman problem, *ScienceAsia*, 2007, **33**, 215-222.

[39] Jayalakshmi G. A., Sathiamoorthy S., Rajaram R., A Hybrid genetic algorithm-a new approach to solve traveling salesman problem, *Int. J. of Computational Engineering Science,* 2001, **2**, 339-355.

[40] Pullan W., Adapting the genetic algorithm to the travelling salesman problem, *Evolutionary Computation,* 2003, **2**, 1029-1035.

[41] Nagata Y., Soler D., A new genetic algorithm for the asymmetric traveling salesman problem, *Expert System with Applications*, 2012, **39**, 8947-8953.

[42] Kirkpatrick S., Gelatt C. D., Vecchi M. P., Optimization by simulated annealing, *Science*, 1983, **220**, 671-680.

[43] Bakır M. A., Altunkaynak B., *Tamsayılı Programlama,* 1. Baskı, Nobel Yayın Dağıtım, Ankara, 2003.

[44] Johnson D. S., McGeoch L. A., The traveling salesman problem: a case study in local optimization, Editors: Aarts E. H., Lenstra J. K., *Local Search in Combinatorial Optimization*, Wiley and Sons, London, 215-310, 1997.

[45] Geng X., Chen Z., Yang W., Shi D., Zhao K., Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search, *Applied Soft Computing*, 2011, **11**, 3680-3689.

[46] Chen S., Chien C., Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques, *Expert System with Applications*, 2011, **38**, 14439-14450.

[47] Glover F., Tabu search - part I, *ORSA J. on Computing*, 1989, **1**, 190-206.

[48] Glover F., Tabu search - part II, *ORSA J. on Computing*, 1990, **2**, 4-32.

[49] Bağış A., Determining fuzzy membership functions with tabu search-an application to control, *Fuzzy Sets and Systems*, 2003, **139**, 209-225.

[50] Cowling P. I., Keuthen R., Embedded local search approaches for routing optimization, *Computers & Operations Research*, 2005, **32**, 465-490.

[51] Tsubakitani S., Evans J.R., Optimizing tabu list size for the traveling salesman problem, *Computers & Operations Research*, 1998, **25**, 91-97.

[52] Misevicius A., Using iterated tabu search for the traveling salesman problem, *Informacines Technologijos ir Valdymas*, 2004, **32**, 29-40.

[53] Misevicius A., Smolinskas J., Tomkevicius A., Iterated tabu search for the traveling salesman problem: new results, *Information Technology and Conrol*, 2005, **34**, 327-337.

[54] Baykoç Ö. F., İşleyen S. K, Simetrik gezgin satıcı problemi için bir etkin tekrarlı yerel arama algoritması, *Teknoloji*, 2007, **10,** 99-106.

[55] Kennedy J., Eberhart R. C, Particle swarm optimization, *IEEE International Conference on Neural Networks*, 1995, **4**, 1942-1948.

[56] Pang W., Wang K., Zhou C., Dong L., Liu M., Zhang H., Wang J., Modified particle swarm optimization based on space transformation for solving traveling salesman problem, *Third International Conference on Machine Learning and Cybernetics*, Shanghai, China, 26-29 August 2004.

[57] Pang, W., Wang, K., Zhou, C., Dong, L., Fuzzy discrete particle swarm optimization for solving traveling salesman problem, *Fourth International Conference on Computer and Information Technology,* Wuhan, China, 14-16 Eylül 2004.

[58] Wang K., Huang L., Zhou C., Pang W., P particle swarm optimization for solving traveling salesman problem, *Second International Conference on Machine Learning and Cybernetics,* Xian, China, 24-27 Ağustos 2003.

[59] Wang, C. Z., Yang Q., An improved discrete particle swarm optimization algorithm for TSP, *International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops,* California, USA, 2-5 November 2007.

[60] Zhong W., Zhang J., Chen W., A novel discrete particle swarm optimization to solve traveling salesman problem, *IEEE Congres on Evolutionary Computation,* Singapoer, 25-28 Eylül 2007.

[61] Goldbarg E. F. G., Goldbarg M. C., Souza G. R. D., Particle swarm optimization algorithm for the traveling salesman problem, Editor: Federico Greco, *Travelling Salesman Problem,* In-Teh, Vienna, 75-95, 2008.

[62] Hopfield J. J., Tank D. W., Neural computation of decisions in optimization problems, *Biol Cyber,* 1985, **52**, 141-152.

[63] Kohonen T., The self organizing map, *Proc IEEE,* 1990, **78**, 74-90.

[64] Yanping B., Zhang W., Jin Z., A new self-organising maps strategy for solving the travelling salesman problem, *Chaos, Solitons & Fractals,* 2006, **28**, 1082-1089.

[65] Tamura H., Zhang Z., Xu X., Ishii M., Tang Z., Lagrangian object relaxation neural network for combinatorial optimization problems, *Neurocomputing,* 2005, **68**, 297-305.

[66] Held M., Karp R. M., A dynamic programming approach to sequencing problems, *Journal of SIAM,* 1962, **10**, 196–210.

[67] Eastman W. L., Linear programming with pattern constraints, Doktora Tezi, Harvard Üniversitesi, Cambridge, 1958.

[68] Little J. D. C., Murty K. G., Sweeney D. W., Karel C., An algorithm for the traveling salesman problem, *Operations Research,* 1963, **11**, 972-989.

[69] Shapiro D. M., Algorithms for the solution of the optimal cost and bottleneck traveling salesman problem, Yüksek Lisans Tezi, Washington Üniversitesi, St. Lois, 1966.

[70] Bellmore M., Malone J. C., Pathology of traveling-salesman subtour elimination algorithms, *Operations Research,* 1971, **19**, 278-307.

[71] Smith T. H. C. , Srinivasan V., Thompson G. L., Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems, *Ann. Dicrete Mathematics,* 1977, **1**, 495-506.

[72] Carpaneto G., Toth P., Some new branching and bounding criteria for the asymmetric traveling salesman problem, *Management Science,* 1980, **26**, 736-743.

[73] Held M., Karp R. M., Traveling-salesman problem and minimum spanning trees, *Operations Research,* 1970, **18**, 1138-1162.

[74] Held M., Karp R. M., Traveling-salesman problem and minimum spanning trees, *Math. Programming,* 1971, **1**, 6-25.

[75] Christofides N., The shortest Hamiltonian chain of a graph, *SIAM Journal of Applied Math.,* 1970, **19**, 689-696.

[76] Helbig K., Hansen, Krarup J., Improvements of Held-Karp algorithm for the symmetric traveling-salesman problem, *Math. Programming,* 1974, **7**, 87-96.

[77] Smith T. H. C. , Thompson G. L., A LIFO implicit enumaration search algorithm for the symmetric traveling salesman problem using Held and Karp's 1-tree relaxation, *Ann. Dicrete Mathematics,* 1977, **1**, 479-493.

[78] Volgenant T., Jonker R., A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation, *European Journal Of Operations Research,* 1982, **9**, 83-89.

[79] Foulds L. R., *Combinatorial Optimization for Undergraduates,* 1st ed., Springer, New York, 1984.

[80] Dantzig G. B., Fulkerson D. R., Johnson S. M., Solution of a large-scale traveling salesman problem, *Operations Research*, 1954, **2**, 393-410.

[81] Miller C., Tucker A., Zemlin R., Integer programming formulations and travelling salesman problems, *Journal of ACM,* 1960, **7**, 326-329.

[82] Gomory R. E., An algorithm for integer solutions to linear programs. Editors: Graves, R. L., Wolfe, P., *Recent Advances in Mathematical Programming,* 1st ed., McGraw-Hill, New York, 269-302, 1963.

[83] Padberg M., Rinaldi G., A branch and cut algorithm for the resolution of large scale travelling salesman problems, *SIAM Review*, 1991, **33**, 60-100.

[84] Padberg M., Rinaldi G., An efficient algorithm for the minimum capacity cut problem, *Mathematical Programming,* 1990, **7**, 19-36.

[85] Padberg M., Rinaldi G., Facet identification for the symmetric travelling salesman polytope, *Mathematical Programming,* 1990, **47**, 219-257.

[86] Grotschel M., On the symmetric travelling salesman problem: Solution of a 120 city problem, *Mathematical Programming Studies,* 1980, **12**, 61-77.

[87] Crowder H., Padberg M. W., Solving large-scale symmetric travelling salesman problems to optimality, *Management Science*, 1980, **26**, 495-509.

[88] Grötschel M., Holland O., Solution of large-scale symmetric travelling salesman problems, *Mathematical Programming,* 1991, **61**, 141-202.

[89] Chvatal V., Edmonds polytopes and weakly Hamiltonian graphs, *Mathematical Programming*, 1973, **5**, 29-40.

[90] Grötschel M., Pulleybank W. R., Clique tree inequalities and the symmetric travelling salesman problem, *Mathematics of Operations Research,* 1986, **11**, 537-569.

[91] Applegate D. L., Bixby R. E., Chvatal V., Cook W., Espinoza D. G., Goycoolea M, Helsgaun K., Certification of an optimal TSP tour through 85,900 cities, *Operations Research Letters,* 2009, **37**, 11-15.

[92] Applegate D. L., Bixby R. E., Chavatal V., Cook W., On the solution of traveling salesman problems, *Documenta Mathematica,* Extra Volume ICM 1998, 645-656.

[93] Desrochers M., Laporte G., Improvements and extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints, *Operational Research Letters,* 1991, **10**, 27-36.

[94] Hui S., Jha G., Application data mining for customer service support, *Information and Management*, 2000, **38,** 1-13.

[95] Shah S., Kursak A., Data mining and genetic algorithms based gene/SNP selection, *Artificial Intelligence in Medicine*, 2004, **31,** 183-196.

[96] Yalçıntaş G., Veri Madenciliği, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2003.

[97] Bland C., The discovery of multiple level profile association rules, Doctoral Thesis, Graduate School of Computer and Information Science of University of Missisipi, Missisipi, 2002.

[98] Hudairy H., Data mining and decision making support in the governmental sector, Master Thesis, Faculty of Graduate School of The University of Louisville, Kentucky, 2004.

[99] Huberty C., *Applied Discriminant Analysis,* 1st ed., John Wiley & Sons Inc., Canada, 1994.

[100] Wei C., Chiu T., Turning telecommunications call details to churn prediction: a data mining approach, *Expert Systems with Applications*, 2002, **23**, 103-102.

[101] Aryeetey K., Data analysis and predictive modelling using the variable precision rough set approach, Master Thesis, Faculty of Graduate Study and Research of University of Regina, Canada, 2003.

[102] Gönülol S., Gezgin satıcı problemi için veri madenciliği tabanlı sezgisel bir yaklaşım, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2009, 232763.

[103] Emel G. G., Taşkın Ç., Veri madenciliğinde karar ağaçları ve bir satış analizi uygulaması, *Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi*, 2005, **6**, 221-239.

[104] Tantuğ A. C., Veri Madenciliği ve Demetleme, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2002, 126982.

[105] Hui S., Jha G., Application data mining for customer service support, *Information and Management*, 2000, **38**, 1-13.

[106] San O., Huynh V., Nakamori Y., An alternative extention of the k means algorithm for clustering categorical data, *Applied Math. and Computer Science*, 2004, **14**, 241-247.

[107] Goebel M., Gruenwald L., A Survey of data mining and knowledge discovery software tools, *ACM SIGKDD Explorations Newsletter*, 1999, **1**, 20-33.

[108] Han J., Kamber, M., *Data mining-concept, techniques, academic*, 1st ed., PRESS, USA, 2001.

[109] Demiriz A., ASIPATH: a simple path mining algorithm, *16th International Conference on Parallel And Distributed Computing And Systems*, Massachusetts, USA, 9-11 November 2004.

[110] Koonce D. A, Tsai S. C., Using data mining to find patterns in genetic algorithm solutions to a job shop Schedule, *Computers & Industrial Engineering*, 2000, **38**, 361-374.

[111] Santos H. G., Ochi L. S., Marinho E. H., Drummond, L. M. A., Combining an evolutionary algorithm with data mining to solve a single-vehicle routing problem, *Neurocomputing*, 2006, **70**, 70-77.

[112] Ceran G., Esnek akış tipi çizelgeleme problemlerinin veri madenciligi ve genetik algoritma kullanılarak çözülmesi, Yüksek Lisans Tezi, Konya Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2006.

[113] Olafsson S., Li X., Learning effective new single machine dispatching rules from optimal scheduling data, *Int. J. Production Economics*, 2010, **128**, 118-126.

EKLER

Ek-A

Tablo Ek-A.1. ulysses16 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	6859	3,66	0	-12,44
2 kısıt	6859	1,92	0	-54,07
3 kısıt	6859	2,44	0	-41,63
4 kısıt	6859	2,15	0	-48,56
5 kısıt	6859	1,86	0	-55,50
6 kısıt	6859	2,61	0	-37,56
7 kısıt	7035	0,67	2,57	-83,97
8 kısıt	7122	0,88	3,83	-78,95
Optimal Sonuç	6859	Optimal Süre		4,18

Tablo Ek-A.2. gr17 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	2085	0,77	0	-15,38
2 kısıt	2085	0,76	0	-16,48
3 kısıt	2088	0,36	0,14	-60,44
4 kısıt	2090	0,29	0,24	-68,13
5 kısıt	2146	0,34	2,93	-62,64
6 kısıt	2146	0,36	2,93	-60,44
7 kısıt	2152	0,42	3,21	-53,85
8 kısıt	2159	0,18	3,55	-80,22
Optimal Sonuç	2085	Optimal Süre		0,91

Tablo Ek-A.3. gr21 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	2707	0,01	0	-90
2 kısıt	2707	0,01	0	-90
3 kısıt	2707	0,01	0	-90
4 kısıt	2758	0,06	1,88	-40
5 kısıt	2758	0,06	1,88	-40
6 kısıt	2758	0,04	1,88	-60
7 kısıt	2758	0,04	1,88	-60
8 kısıt	2758	0,03	1,88	-70
9 kısıt	2758	0,03	1,88	-70
10 kısıt	2803	0,03	3,55	-70
Optimal Sonuç	2707	Optimal Süre		0,1

Tablo Ek-A.4. ulysses22 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	7013	81,07	0	-56,20
2 kısıt	7013	81,09	0	-56,19
3 kısıt	7013	81,03	0	-56,23
4 kısıt	7013	51,23	0	-72,32
5 kısıt	7013	19,84	0	-89,28
6 kısıt	7013	20,66	0	-88,84
7 kısıt	7013	19,47	0	-89,48
8 kısıt	7085	23,65	1,03	-87,22
9 kısıt	7085	23,6	1,03	-87,25
10 kısıt	7085	19,44	1,03	-89,50
Optimal Sonuç	7085	Optimal Süre		185,11

Tablo Ek-A.5. gr24 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	1272	0,05	0	-28,57
2 kısıt	1272	0,04	0	-42,86
3 kısıt	1272	0,03	0	-57,14
4 kısıt	1272	0,03	0	-57,14
5 kısıt	1272	0,03	0	-57,14
6 kısıt	1272	0,03	0	-57,14
7 kısıt	1272	0,04	0	-42,86
8 kısıt	1310	0,05	2,99	-28,57
9 kısıt	1317	0,06	3,54	-14,29
10 kısıt	1320	0,06	3,77	-14,29
Optimal Sonuç	1272	Optimal Süre		0,07

Tablo Ek-A.6. fri26 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	937	0,2	0	0
2 kısıt	937	0,18	0	-10
3 kısıt	937	0,18	0	-10
4 kısıt	937	0,15	0	-25
5 kısıt	937	0,15	0	-25
6 kısıt	937	0,15	0	-25
7 kısıt	937	0,15	0	-25
8 kısıt	937	0,14	0	-30
9 kısıt	937	0,14	0	-30
10 kısıt	937	0,09	0	-55
Optimal Sonuç	937	Optimal Süre		0,2

Tablo Ek-A.7. bays29 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	2020	0,23	0	-4,17
2 kısıt	2022	0,21	0,10	-12,50
3 kısıt	2022	0,20	0,10	-16,67
4 kısıt	2022	0,20	0,10	-16,67
5 kısıt	2022	0,19	0,10	-20,83
6 kısıt	2022	0,18	0,10	-25
7 kısıt	2022	0,17	0,10	-29,17
8 kısıt	2022	0,17	0,10	-29,17
9 kısıt	2054	0,22	1,68	-8,33
10 kısıt	2054	0,22	1,68	-8,33
Optimal Sonuç	2020	Optimal Süre		0,24

Tablo Ek-A.8. bayg29 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	1610	0,21	0	-8,70
2 kısıt	1610	0,21	0	-8,70
3 kısıt	1610	0,20	0	-13,04
4 kısıt	1610	0,20	0	-13,04
5 kısıt	1610	0,17	0	-26,09
6 kısıt	1610	0,18	0	-21,74
7 kısıt	1610	0,09	0	-60,87
8 kısıt	1615	0,17	0,31	-26,09
9 kısıt	1630	0,20	1,24	-13,04
10 kısıt	1630	0,15	1,24	-34,78
Optimal Sonuç	1610	Optimal Süre		0,23

Tablo Ek-A.9. dantzig42 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	699	1,84	0	-22,36
2 kısıt	699	1,66	0	-29,96
3 kısıt	699	1,31	0	-44,73
4 kısıt	699	1,27	0	-46,41
5 kısıt	699	0,81	0	-65,82
6 kısıt	699	0,68	0	-71,31
7 kısıt	699	0,76	0	-67,93
8 kısıt	699	0,60	0	-74,68
9 kısıt	699	0,57	0	-75,95
10 kısıt	699	0,56	0	-76,37
Optimal Sonuç	699	Optimal Süre		2,37

Tablo Ek-A.10. swiss42 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	1273	0,63	0	-19,23
2 kısıt	1273	0,53	0	-32,05
3 kısıt	1273	0,54	0	-30,77
4 kısıt	1273	0,51	0	-34,62
5 kısıt	1273	0,51	0	-34,62
6 kısıt	1273	0,46	0	-41,03
7 kısıt	1273	0,56	0	-28,21
8 kısıt	1273	0,45	0	-42,31
9 kısıt	1273	0,42	0	-46,15
10 kısıt	1273	0,46	0	-41,03
Optimal Sonuç	1273	Optimal Süre		0,78

Tablo Ek-A.11. gr48 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	5046	7,94	0	-7,46
2 kısıt	5046	7,25	0	-15,50
3 kısıt	5046	7,34	0	-14,45
4 kısıt	5055	3,4	0,18	-60,37
5 kısıt	5055	3,57	0,18	-58,39
6 kısıt	5055	2,65	0,18	-69,11
7 kısıt	5055	2,35	0,18	-72,61
8 kısıt	5055	2,23	0,18	-74,01
9 kısıt	5055	2,21	0,18	-74,24
10 kısıt	5055	2,21	0,18	-74,24
Optimal Sonuç	5046	Optimal Süre		8,58

Tablo Ek-A.12. eil51 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	426	0,56	0	0
2 kısıt	426	0,61	0	8,93
3 kısıt	426	0,43	0	-23,21
4 kısıt	426	0,26	0	-53,57
5 kısıt	426	0,29	0	-48,21
6 kısıt	426	0,26	0	-53,57
7 kısıt	427	0,26	0,23	-53,57
8 kısıt	433	0,18	1,64	-67,86
9 kısıt	433	0,15	1,64	-73,21
10 kısıt	433	0,15	1,64	-73,21
Optimal Sonuç	426	Optimal Süre		0,56

Tablo Ek-A.13. berlin52 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	7542	0,34	0	-30,61
2 kısıt	7542	0,32	0	-34,69
3 kısıt	7542	0,32	0	-34,69
4 kısıt	7618	0,42	1,01	-14,29
5 kısıt	7618	0,42	1,01	-14,29
6 kısıt	7618	0,42	1,01	-14,29
7 kısıt	7618	0,49	1,01	0
8 kısıt	7618	0,40	1,01	-18,37
9 kısıt	7618	0,29	1,01	-40,82
10 kısıt	7618	0,28	1,01	-42,86
Optimal Sonuç	7542	Optimal Süre		0,49

Tablo Ek-A.14. brazil58 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	25395	34,44	0	-8,72
2 kısıt	25395	34,63	0	-8,22
3 kısıt	25395	17,44	0	-53,78
4 kısıt	25395	17,19	0	-54,44
5 kısıt	25395	17,36	0	-53,99
6 kısıt	25395	16,87	0	-55,29
7 kısıt	25395	16,83	0	-55,39
8 kısıt	25395	17,28	0	-54,20
9 kısıt	25793	17,25	1,57	-54,28
10 kısıt	25793	15,27	1,57	-59,53
Optimal Sonuç	25395	Optimal Süre		37,73

Tablo Ek-A.15. st70 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	675	13,63	0	9,65
2 kısıt	675	11,24	0	-9,57
3 kısıt	675	7,84	0	-36,93
4 kısıt	675	8,68	0	30,17
5 kısıt	675	8,19	0	-34,11
6 kısıt	675	8,75	0	-29,61
7 kısıt	675	9,22	0	-25,82
8 kısıt	675	6,30	0	-49,32
9 kısıt	675	7,02	0	-43,52
10 kısıt	675	10,4	0	-16,33
Optimal Sonuç	675	Optimal Süre		12,43

Tablo Ek-A.16. eil76 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	543	1,91	0,93	-17,67
2 kısıt	543	2,39	0,93	3,02
3 kısıt	543	1,95	0,93	-15,95
4 kısıt	543	1,24	0,93	-46,55
5 kısıt	543	1,77	0,93	-23,71
6 kısıt	543	1,31	0,93	-43,53
7 kısıt	544	1,2	1,12	-48,28
8 kısıt	544	1,42	1,12	-38,79
9 kısıt	544	1,18	1,12	-49,14
10 kısıt	544	1,24	1,12	-46,55
Optimal Sonuç	538	Optimal Süre		2,32

Tablo Ek-A.17. pr76 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	108159	55,88	0	-16,47
2 kısıt	108159	43,52	0	-34,95
3 kısıt	108159	58,09	0	-13,17
4 kısıt	108159	46,25	0	-30,87
5 kısıt	108159	43,07	0	-35,62
6 kısıt	108159	47,34	0	-29,24
7 kısıt	108651	49,48	0,45	-26,04
8 kısıt	110337	51,29	2,01	-23,33
9 kısıt	110337	47,54	2,01	-28,94
10 kısıt	110337	25,55	2,01	-61,81
Optimal Sonuç	108159	Optimal Süre		66,9

Tablo Ek-A.18. kroA100 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	21454	39,22	0,81	-13,33
2 kısıt	21454	34,74	0,81	-23,23
3 kısıt	21454	34,21	0,81	-24,40
4 kısıt	21466	28,8	0,86	-36,35
5 kısıt	21466	24,88	0,86	-45,02
6 kısıt	21466	26,58	0,86	-41,26
7 kısıt	21466	25,88	0,86	-42,81
8 kısıt	21466	24,25	0,86	-46,41
9 kısıt	21466	26,96	0,86	-40,42
10 kısıt	21466	23,65	0,86	-47,73
Optimal Sonuç	21282	Optimal Süre		45,25

Tablo Ek-A.19. kroB100 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	22141	4730,8	0	-2,26
2 kısıt	22141	2942	0	-39,22
3 kısıt	22141	1047,1	0	-78,37
4 kısıt	22141	779,12	0	-83,90
5 kısıt	22141	599,45	0	-87,61
6 kısıt	22141	553,38	0	-88,57
7 kısıt	22141	644,14	0	-86,69
8 kısıt	22141	532,71	0	-88,99
9 kısıt	22141	514,19	0	-89,38
10 kısıt	22141	485,67	0	-89,97
Optimal Sonuç	22141	Optimal Süre		4840,1

Tablo Ek-A.20. kroC100 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	20749	43,61	0	13,27
2 kısıt	20749	37,72	0	-2,03
3 kısıt	20749	38,41	0	-0,23
4 kısıt	20749	42,54	0	10,49
5 kısıt	20749	36,42	0	-5,40
6 kısıt	20749	35,79	0	-7,04
7 kısıt	20749	31,45	0	-18,31
8 kısıt	20749	36,09	0	-6,26
9 kısıt	20749	35,71	0	-7,25
10 kısıt	20749	29,04	0	-24,57
Optimal Sonuç	20749	Optimal Süre		38,5

Tablo Ek-A.21. kroD100 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	21294	1349	0	-29,04
2 kısıt	21294	1652,2	0	-13,09
3 kısıt	21294	778,66	0	-59,04
4 kısıt	21294	245,65	0	-87,08
5 kısıt	21294	302,9	0	-84,07
6 kısıt	21294	272,36	0	-85,67
7 kısıt	21294	331,23	0	-82,58
8 kısıt	21309	303,34	0	-84,04
9 kısıt	21309	318,13	0	-83,27
10 kısıt	21309	401,26	0	-78,89
Optimal Sonuç	21294	Optimal Süre		1901

Tablo Ek-A.22. rd100 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	7910	29,5	0	-0,14
2 kısıt	7910	26,59	0	-9,99
3 kısıt	7910	31,41	0	6,33
4 kısıt	7910	30,48	0	3,18
5 kısıt	7935	21,43	0,32	-27,45
6 kısıt	7935	21,40	0,32	-27,56
7 kısıt	7935	23,90	0,32	-19,09
8 kısıt	7935	20,03	0,32	-32,19
9 kısıt	7935	18,73	0,32	-36,59
10 kısıt	7975	13,35	0,82	-54,81
Optimal Sonuç	7910	Optimal Süre		29,54

Tablo Ek-A.23. eil101 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	629	3,15	0	-31,97
2 kısıt	629	3,86	0	-16,63
3 kısıt	629	6,01	0	29,81
4 kısıt	629	7,9	0	70,63
5 kısıt	629	2,79	0	-39,74
6 kısıt	629	4,66	0	0,65
7 kısıt	630	3,29	0,16	-28,94
8 kısıt	630	3,31	0,16	-28,51
9 kısıt	630	5,46	0,16	17,93
10 kısıt	630	1,98	0,16	-57,24
Optimal Sonuç	630	Optimal Süre		3,27

Tablo Ek-A.24. lin105 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	14379	53,92	0	-8,41
2 kısıt	14379	53,02	0	-9,94
3 kısıt	14379	54,78	0	-6,95
4 kısıt	14379	58,43	0	-0,75
5 kısıt	14379	53,78	0	-8,65
6 kısıt	14379	50,06	0	-14,97
7 kısıt	14379	45,91	0	-22,01
8 kısıt	14379	45,53	0	-22,66
9 kısıt	14379	45,58	0	-22,58
10 kısıt	14379	45,98	0	-21,90
Optimal Sonuç	14379	Optimal Süre		58,87

Tablo Ek-A.25. gr120 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	6942	65,77	0	2,29
2 kısıt	6945	48,96	0,04	-23,86
3 kısıt	6945	51,59	0,04	-19,77
4 kısıt	6945	45,89	0,04	-28,63
5 kısıt	6945	39,98	0,04	-39,38
6 kısıt	6945	35,67	0,04	-44,53
7 kısıt	6945	36,50	0,04	-43,23
8 kısıt	6945	30,45	0,04	-52,64
9 kısıt	6945	31,45	0,04	-51,09
10 kısıt	6945	27,95	0,04	-56,53
Optimal Sonuç	6945	Optimal Süre		64,3

Tablo Ek-A.26. kroB150 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	26185	9715,2	0,21	-75,71
2 kısıt	26243	9436,9	0,43	-76,41
3 kısıt	26271	9526,8	0,54	-76,18
4 kısıt	26279	9125,6	0,57	-77,19
5 kısıt	26287	8971,3	0,60	-77,57
6 kısıt	26445	8918,4	1,21	-77,70
7 kısıt	26460	8792,5	1,26	-78,02
8 kısıt	26463	8578,1	1,27	-78,55
9 kısıt	26475	8645,3	1,32	-78,39
10 kısıt	26482	8649,7	1,35	-78,38
Optimal Sonuç	26130	Optimal Süre		40000

Tablo Ek-A.27. kroA150 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	26580	86,50	0,21	-93,76
2 kısıt	26765	74,36	0,91	-94,63
3 kısıt	26765	58,98	0,91	-95,74
4 kısıt	26765	68,31	0,91	-95,07
5 kısıt	26893	65,39	1,39	-95,28
6 kısıt	26896	52,13	1,40	-96,24
7 kısıt	26896	53,85	1,40	-96,11
8 kısıt	26896	55,77	1,40	-95,97
9 kısıt	26896	58,89	1,40	-95,75
10 kısıt	26929	46,69	1,53	-96,63
Optimal Sonuç	26524	Optimal Süre		1385

Tablo Ek-A.28. brg180 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	1950	12,19	0	-84,36
2 kısıt	1950	10,67	0	-86,31
3 kısıt	1960	25,31	0,51	-67,52
4 kısıt	1960	8,12	0,51	-89,58
5 kısıt	1970	10,53	1,03	-86,49
6 kısıt	1970	5,94	1,03	-92,38
7 kısıt	1980	10,32	1,54	-86,76
8 kısıt	1990	3,99	2,05	-94,88
9 kısıt	1990	5,42	2,05	-93,04
10 kısıt	2010	4,18	3,08	-94,64
Optimal Sonuç	1950	Optimal Süre		77,92

Tablo Ek-A.29. kroA200 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	29370	46855	0,01	-11,75
2 kısıt	29370	43749	0,01	-17,60
3 kısıt	29372	41963	0,01	-20,96
4 kısıt	29375	38917	0,02	-26,70
5 kısıt	29375	37672	0,02	-29,05
6 kısıt	29375	32680	0,02	-38,45
7 kısıt	29388	32176	0,07	-39,40
8 kısıt	29388	30548	0,07	-42,46
9 kısıt	29408	28155	0,14	-46,97
10 kısıt	29428	27280	0,20	-48,62
Optimal Sonuç	29368	Optimal Süre		53094

Tablo Ek-A.30. kroB200 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	30002	34917	1,92	10,73
2 kısıt	30014	55423	1,96	75,75
3 kısıt	30015	3838,6	1,96	-87,83
4 kısıt	30025	4280,3	2	-86,43
5 kısıt	30025	3489,9	2	-88,93
6 kısıt	30077	3783,8	2,17	-88,00
7 kısıt	30077	3360	2,17	-89,35
8 kısıt	30077	3000	2,17	-90,49
9 kısıt	30077	2605,6	2,17	-91,74
10 kısıt	30077	2531,1	2,17	-91,97
Optimal Sonuç	29437	Optimal Süre		31535

Tablo Ek-A.31. a280 problemi tüm çözüm sonuçları

Eklenen Kısıt Sayısı	Çözüm Sonucu	Çözüm Süresi	Sapma Sonuç (%)	Sapma Süre (%)
1 kısıt	2579	5846,5	0	-0,41
2 kısıt	2592	5319,2	0,50	-9,40
3 kısıt	2592	4917,6	0,50	-16,24
4 kısıt	2660	4794,1	3,14	-18,34
5 kısıt	2660	4418,7	3,14	-24,73
6 kısıt	2660	4312	3,14	-26,55
7 kısıt	2660	4896,9	3,14	-16,59
8 kısıt	2670	4412,9	3,53	-24,83
9 kısıt	2670	1302,1	3,53	-77,82
10 kısıt	2673	1451,8	3,64	-75,27
Optimal Sonuç	2579	Optimal Süre		5870,8

KİŞİSEL YAYINLAR VE ESERLER

[1] Doğruparmak Ş., Keskin G. A., Yaman S., **Alkan A.**, Using component analysis and fuzzy c-means clustering fort he assessment of air quality monitoring, *Atmospheric Pollution Research,* 2014 (Basılmak üzere kabul edildi).

[2] Bozyer Z., **Alkan A.**, Fığlalı A., Kapasite kısıtlı araç rotalama probleminin çözümü için önce grupla sonra rotala merkezli sezgisel zlgoritma önerisi, *Gazi Üniversitesi Bilişim Teknolojileri Dergisi,* 2014, **7** (Basılmak üzere kabul edildi).

[3] Akman G., **Alkan A.**, Tedarik zinciri yönetiminde bulanık AHP yöntemi kullanılarak tedarikçilerin performansının ölçülmesi: otomotiv yan sanayinde bir uygulama, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi,* 2006, **9**, 23-46.

[4] Özçift B., Çiçek E., **Alkan A.**, Aladağ Z., Self-Deciding Demand Forecast System for A Spare Parts Company, *EURO-INFORMS 26th EUROPEAN Conference on Operational Research,* Rome, Italy, 1-4 July 2013.

[5] Hatipoğlu T., Kocabaş M., **Alkan A.**, Cihan A., Measuring Job Satisfaction of Shift Workers Based On Fuzzy Systems Approach, *International Conference Flexible Automation and Intelligent Manufacturing,* Porto, Portugal, 26-28 June 2013.

[6] **Alkan A.**, Serdar A. U., Aladağ Z., Taşıma maliyetinin belirlenmesine karar analizi yaklaşımı, *First International Symposium on Innovative Technologies for Engineering and Science,* Sakarya, Türkiye, 7-9 Haziran 2013.

[7] **Alkan A.**, Erdoğan L. B., Aladağ Z., Rekabetçi ortamda karar ağacı ve oyun teorisi temeline dayanan bir karar verme uygulaması, *13th International Conference on Econometrics Operations Research and Statistics,* Gazimagusa, Kuzey Kıbrıs Türk Cumhuriyeti, 24-26 Mayıs 2012.

[8] **Alkan A.**, Korkmaz Ü., Aladağ Z., Kocaeli-İzmit belediyesi sınırları içerisinde afet yönetim depolarının yer seçimi için bir uygulama, *13th International Conference on Econometrics Operations Research and Statistics,* Gazimagusa, Kuzey Kıbrıs Türk Cumhuriyeti, 24-26 Mayıs 2012.

[9] **Alkan A.**, Çakır S., Aladağ Z., VIKOR yöntemi ile tedarikçi performansının değerlendirilmesi, *12th International Symposium on Econometrics Statistics and Operations Research,* Denizli, Türkiye, 26-29 Mayıs 2011.

[10] Alkan S., Yuluğkural Y., **Alkan A.**, Aladağ Z., A supplier selection problem with axiomatic design method, *International Symposium on Engineering and Architectural Sciences of Balkan, Caucasus and Turkic Republics,* Isparta, Turkey, 22-24 October 2009.

[11] Akman G., **Alkan A.**, An evaluation model of supplier performance: a fuzzy logic approach, *International Conference on Modeling and Simulation,* Konya, Turkey, 28-30 August 2006.

[12] **Alkan A.**, Karaca E., Aladağ Z., Otomotiv sektöründe emisyon tarihlerine bağlı aşamalı karar analizi, *XI. Ulusal Üretim Araştırmaları Sempozyumu,* İstanbul, Türkiye, 23-24 Haziran 2011.

[13] Korucu M. K., **Alkan A.**, Karademir A., Aladağ Z., Uludağ O., Kocaeli'nde Evsel Katı Atık Yönetiminde En Uygun Bertaraf Seçeneğinin Belirlenmesi için AHP ve ANP Yöntemlerinin Bir Karşılaştırılması, *9. Ulusal Çevre Mühendisliği Kongresi,* Samsun, Türkiye, 5-8 Ekim 2011.

[14] Aksu B., Cihan A., **Alkan A.**, Aladağ Z., Haftalık Ders Çizelgesi Hazırlamada Bulanık Hedef Programlama ile Ders Ataması, *YA/EM'2008-Yöneylem Araştırması/Endüstri Mühendisliği-XXVIII Ulusal Kongresi,* İstanbul, Türkiye, 30 Haziran-2 Temmuz 2008.

[15] Taş N., Tezcan E., **Alkan A.**, Aladağ Z., Tedarik Zinciri Yönetiminde Tedarikçi Seçimine Bulanık Yaklaşım, *ÜAS'2008-VIII. Ulusal Üretim Araştırmaları Sempozyumu,* İstanbul, Türkiye, 24-25 Ekim 2008.

[16] Baynal K., **Alkan A.**, İbik Ö. A., Rekabet Ortamında Hizmet Kalitesinin Önemi ve Bir Havayolu İşletmesinde Hizmet Kalitesinin Gerçekleştirilmesine Yönelik Bir Uygulama, *YA/EM'07 - Yöneylem Araştırması/Endüstri Mühendisliği-XXVII Ulusal Kongresi,* İzmir, Türkiye, 2-4 Temmuz 2007.

[17] **Alkan A.**, Alkan S., Göktürk İ. F., Aladağ Z., Sağlık Sektöründe Bir Satınalma Kararının Analitik Serim Süreci Yöntemi ile Çözülmesi, *VI. Endüstri-İşletme Mühendisliği Kurultayı,* Bursa, Türkiye, 9-10 Kasım 2007.

[18] Çiftçi C., **Alkan A.**, Aladağ Z., Türk Deniz Kuvvetleri Sahil Stoklarının Belierlenmesinde Analitik Hiyerarşi Süreci Kullanılması, *VI. Endüstri-İşletme Mühendisliği Kurultayı,* Bursa, Türkiye, 9-10 Kasım 2007.

[19] Çiftçi C., **Alkan A.**, Aladağ Z., Türk Deniz Kuvvetleri ve Kurumsal Kaynak Planlama, *YA/EM'2006 - Yöneylem Araştırması/Endüstri Mühendisliği - XXVI Ulusal Kongresi,* Kocaeli, Türkiye, 3-5 Temmuz 2006.

[20] Akman G., Güneyli O., **Alkan A.**, Kalite Yönetim Sisteminin Belgelendirilmesi Sürecinin Performansını İyileştirmede Kısıtlar Teorisi Yaklaşımının Kullanımı, *YA/EM'2006 - Yöneylem Araştırması/Endüstri Mühendisliği - XXVI Ulusal Kongresi,* Kocaeli, Türkiye, 3-5 Temmuz 2006.

[21] Eranıl B., Varış Ö., **Alkan A.**, Aladağ Z., İşe Alım Sürecinde Etkileşimli Beklenti Düzeyi ve Analitik Hiyerarşi Prosesi Yaklaşımı, *YA/EM'2006 - Yöneylem Araştırması/Endüstri Mühendisliği - XXVI Ulusal Kongresi,* Kocaeli, Türkiye, 3-5 Temmuz 2006.

ÖZGEÇMİŞ

1981'de Kocaeli'de doğdu. İlk, orta ve lise öğrenimini doğduğu şehirde tamamladı. 1999 yılında girdiği Kocaeli Üniversitesi Endüstri Mühendisliği Bölümü'nden 2003 yılında mezun oldu. 2003 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında yüksek lisansa başladı. 2006 yılında yüksek lisans programını tamamlayarak aynı yıl Kocaeli Üniversitesi Endüstri Mühendisliği Anabilim Dalında doktora programına başladı. Halen Kocaeli Üniversitesi Mühendislik Fakültesinde Araştırma Görevlisi olarak görev yapmaktadır. Evli ve bir çocuk babasıdır.