



**YETERİNCE YAKIN KOMŞU ARAMA SEZGİSELİNİN SAYAÇ OKUMA
SÜRECİ İÇİN KULLANILMASI**

Can TOPAÇLIOĞLU

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MAYIS 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Can TOPAÇLIOĞLU

24/05/2021

YETERİNCE YAKIN KOMŞU ARAMA SEZGİSELİNİN SAYAÇ OKUMA SÜRECİ İÇİN KULLANILMASI

(Yüksek Lisans Tezi)

Can TOPAÇLIOĞLU

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2021

ÖZET

Yeterince Yakın Gezgin Satıcı Problemi (YY-GSP), pandemi döneminde temasın azaltılıp sosyal mesafenin artması, teknolojinin gelişmesi ve yeni iletişim protokollerinin yaygın hale gelmesi ile popülerlik kazanmıştır. Bu problemde, klasik Gezgin Satıcı Probleminin (GSP) aksine, müşterileri ziyaret etmek için müşterilerin bulundukları noktalara “yeterince yakın” olmak yeterlidir. Bütün müşterilerin disk şeklindeki komşuluk alanlarına sahip olduğu ve satıcının bu alanları kullanarak ziyaret turunu oluşturduğu varsayılır. İnsansız hava araçları ile hedeflerin belli bir mesafeden görüntülenmesinin yeterli olduğu keşif uçuşları, müşterilerin kargolarını almak için belli bir noktaya gelmeyi kabul ettikleri durumda oluşan teslimat turu, elektrik, su, doğalgaz sayaçlarının gerekli teknolojiler ile uzaktan okunabildiği sayaç okuma süreci YY-GSP’ye örnek olarak gösterilebilir. Bu çalışmada sayaç okuma problemi, YY-GSP olarak ele alınmıştır. Klasik GSP’ye göre YY-GSP’nin tur maliyetinde sağlayabileceği iyileştirme potansiyeli ölçülmüş, problem en yakın komşu arama sezgiseli ile çözüme ulaştırılmıştır. Çalışmada 36 ile 250 arasında değişen düğüm sayılarına sahip 22 veri seti kullanılmıştır. Ayrıca YY-GSP’nin gerçek hayat sayaç okuma süreçlerine etkisini görmek için, Türkiye-Ankara’da örnek bir yerleşim bölgesi seçilmiş ve bu yerleşim bölgesi içerisinde yapılacak sayaç okuması için YY-GSP çözümü elde edilmiştir. Çalışma kapsamında geliştirilen ve “Yeterince Yakın Komşu Arama” (YYKA) adı verilen, bir sezgisel ile elde edilen sonuçlar, klasik GSP çözümü ile karşılaştırılmıştır. Tur maliyetinde YY-GSP’nin klasik GSP’ye göre örnek veri setlerinde %49’a varan yüksek oranda tasarruflar sağladığı görülmüştür. Gerçek hayat uygulamasında ise, %26 oranında iyileştirme sağlanmıştır. Bu çalışma, YY-GSP’de tur maliyetleri için, klasik bir çözüm kurucu sezgisel kullanılması ile herhangi kesin çözüm metodu ya da metasezgisel dahi kullanılmadan YY-GSP’nin ne kadar etkili bir GSP varyasyonu olduğunu göstermiştir.

Bilim Kodu : 90620

Anahtar Kelimeler : Yeterince yakın gezgin satıcı problemi, En yakın komşu arama, Sezgisel yöntemler, Çember geometrisi

Sayfa Adedi : 47

Danışman : Prof. Dr. Selçuk Kürşat İŞLEYEN

USING CLOSE-ENOUGH NEIGHBORHOOD SEARCH HEURISTIC FOR
AUTOMATED METER READING

(M. Sc. Thesis)

Can TOPAÇLIOĞLU

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

May 2021

ABSTRACT

Close Enough Travelling Salesman Problem (CETSP) gained popularity with coronavirus pandemic, where direct contact between each other is being reduced, social distance practices are being applied and new communication protocols becoming more accessible with the recent advances in technology. In this problem, unlike classical Travelling Salesman Problem (TSP), a seller is only required to be “close enough” to customers in order for customers to be considered visited by the salesman. All customers have a disk shaped neighborhood area and salesman will use these areas to sequence the visits. Cases in which a route is to be constructed for maintenance and reconnaissance flights by an UAV, where imaging from a distance is enough, and a shipment delivery model where recipients can travel to certain points to receive their packages, a utility meter reading problem in which necessary technologies are integrated to both meters and readers can be considered CETSP. In this study, utility meter reading process is defined as CETSP. Improvement potential of CETSP versus classical TSP is measured. Problem is solved by a heuristic based on nearest neighborhood search heuristic. To measure the performance, 22 data sets are used, varying in nodes between 36 and 250. Also, to measure the performance of CETSP in real life applications, a residential area in Ankara-Turkey is selected as a case to apply CETSP solution. Close Enough Neighborhood Search (CENS) heuristic is developed for this study. Results obtained from this algorithm is compared with classical TSP solution. Improvements as high as 49% are obtained using CETSP over TSP. In real life application, improvement rate is 26%. This study showed the effectiveness of using CETSP, even with using a constructive heuristic instead of exact solutions methods or metaheuristics.

Science Code : 90620

Key Words : Close enough travelling salesman problem, Nearest neighborhood search, Heuristic methods, Circle geometry

Page Number : 47

Supervisor : Prof. Dr. Selçuk Kürşat İŞLEYEN

TEŞEKKÜR

Bu çalışmanın gerçekleşmesinde katkıları bulunan, aşağıda adı geçen kişilere sonsuz şükran ve teşekkürlerimi sunarım.

Her zaman yol gösterici olan değerli danışmanım, Sayın Prof. Dr. Selçuk Kürşat İŞLEYEN'e tez sürecim boyunca gösterdiği büyük sabır ve paha biçilemez yardımları için.

Ayrıca değerli arkadaşlarım Başak ZÜLFİKAR ŞEN'e, Özgür Ozan ŞEN'e, Ufuk AYDAN'a ve Yiğit Levent ÇEÇEN'e, çalışmam boyunca benden hiçbir zaman yardımlarını esirgemedikleri, zaman ayırıp beni dinledikleri için.

Çalışmam boyunca desteklerini esirgememiş değerli aileme, bana gösterdikleri saygı ve sabır için.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ.....	1
2. YETERİNCE YAKIN GEZGİN SATICI PROBLEMİ	5
2.1. Alt-Üst Sınır Çalışmaları.....	5
2.2. Kesin Çözüm Çalışmaları	5
2.3. Sezgisel Çözüm Çalışmaları	6
2.4. YY-GSP İçin Matematiksel Model.....	11
2.5. Çalışmaların Karşılaştırılması.....	12
3. YETERİNCE YAKIN KOMŞU ARAMA (YYKA)	13
3.1. Kavramlar.....	13
3.2. Algoritma	14
3.2.1. Varsayımlar	15
3.3. Örnek Çözüm	16
4. HESAPLAMA SONUÇLARI	27
4.1. Kullanılan Veri Setleri	27
4.1.1. Uygulama.....	28
4.2. Bulgular ve Yorum.....	28

Sayfa

5. SONUÇ VE ÖNERİLER	39
KAYNAKLAR	41
EKLER	43
EK-1. YYKA algoritması akış şeması	44
ÖZGEÇMİŞ	47

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Çalışmaların karşılaştırılması	12
Çizelge 3.1. Örnek çözüm veri seti düğüm koordinatları	16
Çizelge 4.1. Düzenli veri setleri için YYKA1 tur maliyetleri	29
Çizelge 4.2. Düzenli veri setleri için YYKA2 tur maliyetleri	30
Çizelge 4.3. Düzenli veri setleri için YYKA1 süperdüğüm sayıları.....	31
Çizelge 4.4. Düzenli veri setleri için YYKA2 süperdüğüm sayıları.....	32
Çizelge 4.5. Düzensiz veri setleri için YYKA1 tur maliyetleri	32
Çizelge 4.6. Düzensiz veri setleri için YYKA1 tur maliyetleri	33
Çizelge 4.7. Düzensiz veri setleri için YYKA1 düğüm sayıları	34
Çizelge 4.8. Düzensiz veri setleri için YYKA2 düğüm sayıları	35
Çizelge 4.9. YYKA için sonuçların karşılaştırılması.....	36

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. (a) GSP, (b) YY-GSP	2
Şekil 3.1. i düğümü için $SZ(3)$ bölgesi.....	14
Şekil 3.2. Örnek çözüm veri seti ağ yapısı.....	17
Şekil 3.3. $SZ(2)$ bölgeleri	18
Şekil 3.4. $SZ(3)$ bölgeleri	19
Şekil 3.5. Süperdüğümler.....	20
Şekil 3.6. Klasik GSP turu	21
Şekil 3.7. $SZ(3)$ seviyesindeki süperdüğümler.....	22
Şekil 3.8. $SZ(2)$ seviyesindeki süperdüğümler.....	23
Şekil 3.9. $SZ(1)$ seviyesindeki süperdüğümler.....	24
Şekil 3.10. YYKA1 turu	25
Şekil 3.11. YYKA2 turu	26
Şekil 4.1. (a) $b2-76$ veri seti için ağ yapısı, (b) $bn7-50$ veri seti için ağ yapısı	27
Şekil 4.2. $bati67$ veri seti için ağ yapısı.....	28

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

%

Yüzde

s

Saniye

Kısaltmalar

Açıklamalar

CENS

Close-Enough Neighborhood Search

CETSP

Close-Enough Travelling Salesman Problem

CPLEX

IBM ILOG CPLEX Optimization Studio

EYK

En Yakın Komşu Arama

GSP

Gezgin Satıcı Problemi

NFC

Near Field Communication

NP

Non-Deterministic Polynomial-Time

RFID

Radio Frequency Identification

SZ

Steiner Zone

SZH

Steiner Zone Heuristic

SZVNS

Steiner Zone Variable Neighborhood Search

TSP

Travelling Salesman Problem

TSPLIB

Traveling Salesman Problem Library

Wi-Fi

Wireless Fidelity

YY-GSP

Yeterince Yakın Gezgin Satıcı Problemi

YYKA

Yeterince Yakın Komşu Arama

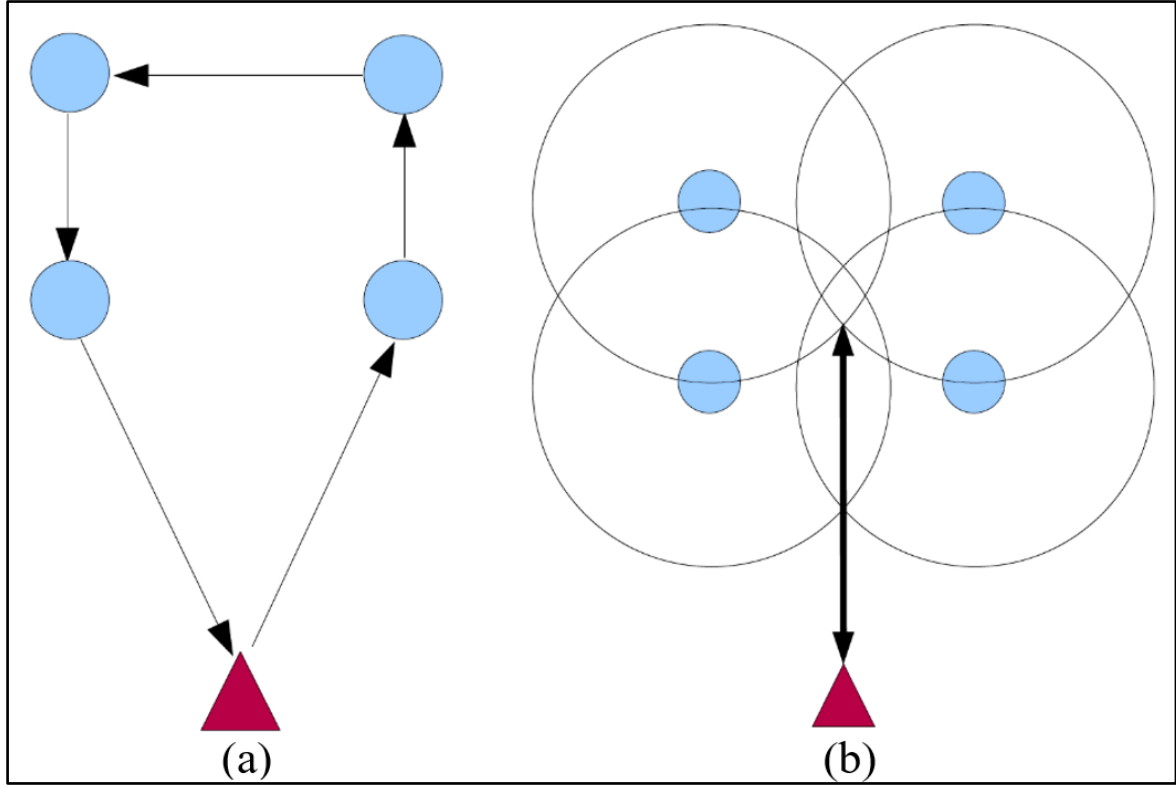
1. GİRİŞ

Kombinatorial optimizasyon problemlerinin en bilinenlerinden biri Gezgin Satıcı Problemi (GSP)'dir. Bu problemde bir satıcı, belli bir şehirden başlayarak, müşterilerinin bulunduğu bütün şehirleri yalnızca bir kere ziyaret ettikten sonra başladığı şehre dönmektedir. GSP'de amaç, açıklandığı şekilde oluşturulan Hamilton turu için en küçük maliyeti bulmaktır. Bahsedilen maliyet; zaman, mesafe, para vb. kaynaklar olabilir.

Günümüz koronavirüs pandemisinin sonucu olarak, sosyal mesafe kurallarının uygulanması ve temasın azaltılması, doğrudan teması gerektiren bu problem çeşidinde farklı arayışlara neden olmuştur. Özellikle teknolojinin son yıllardaki gelişimi, yeni iletişim protokollerini daha kolay ve ucuz şekilde uygulanabilir hale getirmiştir. Bu iletişim protokollerinden yararlanılarak satıcının, müşteri ile arasında belli bir mesafe olsa dahi ziyaretini tamamlayabileceği varsayılan GSP modeline Yeterince Yakın Gezgin Satıcı Problemi (YY-GSP) denmiştir.

YY-GSP'de satıcının, müşterileri tek tek ziyaret etmek zorunluluğu yoktur. Bu zorunluluğu kaldırmak için, satıcının müşteriye “yeterince yakın” olmasının yeterli olduğu varsayılmıştır. Bunun için, çeşitli teknolojiler kullanılarak, müşterilerin bulunduğu düğümleri merkez kabul eden komşuluk alan diskleri oluşturulmuştur. Bu komşuluk alan diskleri, önceden belirlenmiş yarıçaplara sahip, çember şeklindeki alanlardan oluşmaktadır. Eğer satıcı, bu komşuluk alanı içerisinde herhangi bir noktayı ziyaret ederse, komşuluk alanı sahibi müşteriyi de ziyaret etmiş sayılacaktır. Kısaca açıklamak gerekirse, YY-GSP'de müşteriler satıcıdan hizmet almak için, kendi komşuluk alanında bulunan herhangi bir noktaya gitmeyi kabul etmektedirler [1]. Bu alanlar, diğer düğümlerin komşuluk alanları ile ayrık, teğet ya da iç içe/kesişimli olabilir. Komşuluk alanları kullanılarak, klasik GSP'ye göre maliyetten tasarruf edilebilir. Problem, klasik GSP'nin genelleştirilmiş halidir [2].

Şekil 1.1(a)'daki klasik GSP'de, satıcının her bir düğümü ayrı ayrı ziyaret etmesi ve depoya geri dönmesi ile oluşan tur görülmektedir. Şekil 1.1(b)'de ise, aynı ağ üzerinde, düğümlerin belli yarıçapa sahip komşuluk alan disklerinin kullanılması ile oluşan YY-GSP geometrik yapısı gösterilmiştir. Bu yapı üzerinde, satıcının komşuluk alan disklerinin kesişimlerinde bulunan tek bir noktayı ziyaret etmesi, düğümlerin ziyaret edilmesi ve turun tamamlanması için yeterlidir.



Şekil 1.1. (a) GSP, (b) YY-GSP [3]

YY-GSP birçok gerçek hayat uygulaması için uyarlanabilir. Bir insansız hava aracının keşif amacı ile hedeflerin belli bir uzaklığındaki noktaları ziyaret etmesinin yeterli olduğu durumlarda, kullanacağı rotanın belirlenmesi problemi YY-GSP olarak değerlendirebilir [3, 4]. RFID, Wi-Fi, Bluetooth, NFC gibi iletişim protokollerinin herkes tarafından rahatlıkla kullanılabildiği günümüz teknolojisi, YY-GSP'nin, doğalgaz, su, elektrik gibi faturalandırma yapılan hizmetlerin sayaç okumasında kullanılan rotaların oluşturulması problemlerine uygulanabilmesini sağlamaktadır [5, 6]. Ayrıca, müşterilerin kargolarını almak için belli bir noktaya kadar gelmeyi kabul ettikleri durumda, kargo araçlarının teslimat rotasını oluşturmak için YY-GSP kullanılabilir.

YY-GSP, literatürde incelendiği kadarıyla, [7]'de komşuluk içeren GSP olarak tanımlanmış, fakat YY-GSP(*CETSP*) adını alması [5]'te olmuştur. Çalışmalar kapsamında, problem ilk tanımlandığından beri, çözüm geliştirme için kesin çözüm yöntemleri [8, 6, 4, 9] ve sezgisel çözümler önerilmiş [5, 3, 10, 11, 12, 13, 14, 15], optimal çözüm için alt ve üst sınırların bulunması için yaklaşımlar geliştirilmiştir [9, 10, 13, 7]. Ayrıca problem, birçok çalışmada doğrusal olmayan bir matematiksel model olarak ifade edilmiş [6, 4, 8, 12, 3], farklı optimizasyon yöntemleriyle sonuçlar alınmıştır. Problem için, zaman penceresi [12],

düzlemde üçüncü boyut [8, 15] gibi kısıt ve özellikler eklenerek, farklı YY-GSP çalışmaları yapılmıştır.

GSP kombinatorial optimizasyon problemi olarak NP-zor sınıfında yer almaktadır [14]. Bu nedenle çözüm zamanı, problem boyutunun artışı ile üstel olarak artmaktadır. Problemin NP-zor sınıfında yer almasından dolayı, sezgisel ve metasezgisel yöntemler, problemin çözümünde sıklıkla kullanılmaktadır. YY-GSP de, GSP'nin bir varyasyonu olarak aynı sınıfta yer almaktadır.

Bu çalışmada, YY-GSP, sayaç okuma süreçlerine uyarlanmıştır. Çeşitli teknolojilerin entegre edildiği sayaçlar, problem içerisinde düğüm olarak tanımlanmış, satıcı ise sayaç okuma personeli olarak düşünülmüştür. Çalışma kapsamında sezgisel bir yöntem olan en yakın komşu arama, problem için uyarlanmıştır. Çalışmanın amacı, YY-GSP'nin, klasik GSP'ye göre sağladığı iyileştirmenin ölçülmesi, aynı zamanda YY-GSP'nin sezgisel bir algoritma ile çözüme ulaştırılmasıdır. Çalışmada 36 ile 250 arasında değişen düğüm sayılarına sahip ağlar örnek problemler olarak kullanılmıştır. Ayrıca YY-GSP'nin gerçek hayat sayaç okuma süreçlerine etkisini görmek için, Türkiye-Ankara'da örnek bir yerleşim bölgesi seçilmiş ve bu yerleşim bölgesi içerisinde yapılacak sayaç okuması için YY-GSP çözümü ile klasik GSP çözümü karşılaştırılmıştır. Çalışma sonucunda, örnek veri setleri için hesaplanan tur maliyetlerinde, YY-GSP'nin klasik GSP'ye göre %49'a varan yüksek oranlarda tasarruflar sağladığı görülmüştür. Gerçek hayat uygulamasında ise, %26 oranında tur maliyetinde iyileştirme sağlanmıştır. Bu çalışma, YY-GSP'nin tur maliyetindeki iyileştirme potansiyelini göstermek adına var olan çalışmalar bütününe katkıda bulunmuştur.

Çalışmanın devamı şu şekilde yapılandırılmıştır; ikinci bölümünde bu alanda şimdiye kadar yapılan çalışmalar ayrıntılı olarak özetlenmiştir. Bir sonraki bölümde, problem çözüm yöntemi ayrıntılı olarak açıklanmış, varsayımlar belirtilmiş ve geliştirilen sezgisel, örnek bir çözüm ile açıklanmıştır. Dördüncü bölümde farklı örnek problemler için sezgisel yöntem uygulanmış, sonuçlar tablo halinde yansıtılmıştır. Gerçek bir sayaç okuma süreci için seçilmiş bir mahalle ağına ilişkin detaylar ve geliştirilen algoritmanın çalıştırma sonuçları da bu bölümde aktarılmıştır. Ayrıca bu bölümde sonuçların yorumlarına yer verilmiştir. Son bölümde ise çalışmanın sonuçları açıklanmış, YY-GSP çalışmalarının geleceği tartışılmıştır.

2. YETERİNCE YAKIN GEZGİN SATICI PROBLEMİ

Bu bölümde, YY-GSP hakkında ayrıntılı bilgi verilmiştir. Literatürdeki alt-üst sınır çalışmaları, kesin ve sezgisel çözüm yöntemleri anlatılmıştır. Aynı zamanda Mennell (2009) tarafından geliştirilmiş, YY-GSP modelini yansıtan matematiksel bir model verilmiş ve açıklanmıştır. Şimdiye kadar olan çalışmaların karşılaştırılması yapılmıştır.

2.1. Alt-Üst Sınır Çalışmaları

Dumitrescu ve Mitchell (2003) çalışmalarında, müşterilerin, satıcı ile buluşmayı kabul edebileceği eşit diskler şeklinde bir komşuluğa sahip olduğu bir yapı tanımlamıştır [7]. Ayrık diskler ve kesişimli diskler şeklinde iki farklı durumdan bahsedilmiştir. Sonuç olarak sabit faktörlü ve polinom zamanlı yaklaşım algoritmaları geliştirilmiştir.

Yeni bir ayrıştırma planı geliştirmek amacıyla yola çıkan bir çalışmada, bu sayede ziyaret edilecek noktaların daha iyi belirlenerek alt-üst sınırların iyileştirilmesi amaçlanmıştır [9]. Aynı yazarlar, bir sonraki çalışmalarında ayrıştırma planını ikinci dereceden koni programlama ile geliştirerek hazırladıkları sezgisel yöntemde, önceki alt-üst sınır çalışmalarına göre daha iyi sonuçlar elde edildiğini belirtmişlerdir [10].

2020 yılındaki çalışmada [13] yazarlar, alt ve üst sınırların bulunması için yeni ve etkin bir algoritma geliştirmişlerdir.

Ayrıca Mennell (2009)'da çalışması kapsamında, YY-GSP için kullanılmak üzere alt-üst sınır yaklaşımları yapmıştır. Çalışma sezgisel yöntem ağırlıklı olduğu için, ayrıntılı özet ilgili bölümde verilmiştir.

2.2. Kesin Çözüm Çalışmaları

YY-GSP modelini kenar kaplama kısıtlarıyla değerlendiren bir çalışmada [6], tamsayı doğrusal bir model ile kesme düzlemi algoritması kullanılarak problem kesin çözüme ulaştırılmaya çalışılmıştır. Problem, sayaç okuma süreçleri üzerinden örneklendirilmiştir. Çözümler için iki farklı işlem ile rastgele üretilen 960 problem kullanılmıştır. Bu problemlerde, boyut belirlerken düğüm sayıları ve kenarlar arasında orantısal bir sabit kümesi kullanılmış, çözüm elde edilirken iki farklı komşuluk alanı diski yarıçap sayısı

kullanılmıştır. Çözüm sürelerinin, problem boyutuna ve ağların geometrik yapısına göre değiştiği görülmüştür. Küçük yarıçap kullanılarak çözülebilen bazı problemlerin, daha büyük yarıçap kullanıldığı durumda çözüme ulaştırılamadığı görülmüştür. Ayrıca, yarıçap artışının, tur maliyetini azaltmakta etkili olduğu belirtilmiştir.

2015 yılındaki çalışmada [4] yazarlar, bir insansız hava aracının, tarla bakımı keşif uçuşları için belli noktaların fotoğrafları çekmesi gerektiğini, bu noktalar arası rotalama problemi için, GSP ve YY-GSP çözümlerinin uygulanabileceğini belirtmişlerdir. Çalışma kapsamında hem GSP, hem YY-GSP için çözümler üretilip bu çözümleri karşılaştırmışlardır. GSP için sezgisel bir çözüm kullanılırken, YY-GSP çözümü için, dal-sınır algoritması tabanlı bir kesin çözüm yöntemi kullanmışlardır. Ağda bulunan komşuluk alanlarını, yerdeki noktaların merkezlerini oluşturduğu yarımküreler şeklinde belirlemişlerdir. Yarımkürelerin yarıçapları üç farklı büyüklükte olacak şekilde çözümler yapılmıştır. Her çözüm için, hesaplama süresini, fotoğraf kalitesini ve tur maliyetini karşılaştırmışlardır. YY-GSP çözümünün tur maliyetini düşürdüğü, fotoğraf kalitesini arttırdığı fakat kesin çözüm yöntemi kullanıldığı için hesaplama süresinin arttığı gözlemlenmiştir.

Başka bir çalışmada, dal-sınır algoritması, ikinci dereceden koni programlama ile birlikte kullanılıp kesin çözümler elde edilmeye çalışılmıştır [8].

2.3. Sezgisel Çözüm Çalışmaları

Gulczynski, Heath ve Price (2006), ilk defa Yeterince Yakın (Close-Enough) terimini kullanmıştır [5]. Çalışma, sayaç okuma süreçlerini örnek göstererek yapılmıştır. RFID etiketlerinin sayaçlarda kullanılarak sayaçların bu etiketlerin oluşturduğu alanlar içerisinde okunabildiğini belirtilmiş, bu durumda oluşan GSP yapısı YY-GSP olarak adlandırılmıştır. Sabit bir depoyu başlangıç ve bitiş düğümü olarak kullanan ağlarda, düğümlerin alan olarak eşit disk şeklinde komşuluklara sahip oldukları belirtilmiştir. Ayrıca YY-GSP'nin uzaysal pencereye sahip bir GSP olduğunu belirtip, zaman penceresi içeren GSP ile benzer olduğunu söylemişlerdir.

Çalışma kapsamında tanımlanan süperdüğüm (*supernode*) kümesi, her bir müşteri düğümünün, kümedeki en az bir noktaya, en fazla komşuluk alanı diski yarıçapı r uzaklığında olan noktalar kümesidir. Genel olarak, süperdüğüm kümesinin büyüklüğünün

tur maliyeti ile direk ilişkisi olmasa da, küme büyüklüğünün azaltılması, tur maliyetlerinde tasarruf sağlamıştır. Bu nedenle, süperdüğüm kümesini oluşturmak ve büyüklüğünü azaltmak için altı farklı sezgisel yöntem geliştirmişlerdir. Bu yöntemlerin ilk üçünü döşeme yöntemleri (*tiling methods*) başlığı altında toplamışlardır. Bu yöntemlerde ortak olarak, bütün ağı, r yarıçapına sahip çemberlere dıştan teğet olan çokgenlerle kaplamışlardır. Bu şekilde orijinal ağda bulunan herhangi bir düğümün, içinde bulunduğu çokgenin dış teğet çemberine en fazla r uzaklıkta olması sağlanmıştır.

Kaydırma (*Shifting*) yönteminde çokgenleri koordinat düzleminde küçük miktarlarda kaydırarak, süperdüğüm kümesinin büyüklüğünü azaltmayı amaçlamışlardır. Kaydırma yapılarak elde edilen süperdüğüm kümelerinden en az büyüklüğe sahip olan küme çözüm için seçilmektedir.

Birleştirme (*merging*) yönteminde ise seçilen iki çokgen alanının kapsadığı düğümler değerlendirilerek, bu süperdüğümlerin aynı çokgen içerisinde ifade edilip edilemeyeceğine göre süperdüğüm setinin büyüklüğü yeniden belirlenmiştir. Seçilmek için aday olan çokgen kalmayana kadar bu işlem tekrarlanmaktadır.

Son döşeme yöntemi, dairesel uzantı (*circular extension*) olarak adlandırılmıştır. Bu yöntemde, süperdüğümler için, bu süperdüğümlerden en fazla r uzaklığında olan düğüm sayısı süperdüğümün derecesi olarak tanımlanmıştır. İki ya da daha fazla dairenin kesişim noktasında bulunan süperdüğümler belirlenerek gereksiz olan çemberlerin kümeden çıkartılması ile süperdüğüm sayısı azaltılmaya çalışılmıştır.

Steiner bölgesi (*Steiner zone*) yönteminde, iki farklı düğüm için, birbirlerinden en fazla $2r$ uzaklıkta olmaları durumunda bir kesişim bölgesi oluşturacakları belirtilmiştir. Bahsedilen kesişim bölgelerine, Steiner bölgesi adı verilmiştir. Bu durumda, süperdüğüm kümesinin büyüklüğü, Steiner bölgelerinden en fazla kesişime sahip olan bölgeler içerisindeki noktalardan süperdüğüm seçimi yapılması ile azaltılabilir. Bu bölgelerdeki herhangi bir nokta, süperdüğüm olabilir. Süperdüğüm sayısının azaltılması için, en büyük dereceye sahip Steiner bölgeleri seçilmiştir. Bu bölgelerden rastgele seçilmiş noktalar ile süperdüğüm kümesi oluşturulmuştur.

Bir sonraki sezgisel yöntem olan daire süpürme (*sweeping circle*) yönteminde, döşeme yöntemlerine benzer şekilde bütün ağ, birbirlerinden belli uzaklıktaki daireler ile kaplanmaktadır. Bu dairelerden en büyük dereceye sahip olanların merkezleri, süperdüğüm kümesine eklenmektedir. Bu işlem, bütün düğümler kapsanıncaya kadar etmektedir.

Son yöntem olan yarıçap yakınlığı (*radial adjacency*) yönteminde, düğümlerden birbirine uzaklığı en fazla r olanların 1, daha uzak olanları 0 ile ifade edildiği bir düğüm-yakınlık matrisi oluşturulmaktadır. Bu matris içerisinde, satırlardaki 1 değerlerinin sayısına göre en büyük dereceye sahip olan düğüm seçilmiştir. Ayrıca bu düğüm ile yakınlığındaki diğer düğümlerin geometrik ortalaması alınarak seçilen düğüm ile geometrik ortalamaya sahip nokta karşılaştırılmıştır. Bu karşılaştırmada daha büyük dereceye sahip olan nokta, süperdüğüm kümesine eklenmiştir.

Bütün sezgisel yöntemlerden elde edilen süperdüğüm kümelerini, belirtilmeyen GSP sezgisel çözüm yöntemleri ile çözüp, optimale yakın çözümler elde etmişlerdir. Ekonomizasyon bölümünde ise, herhangi bir süperdüğüm i noktasının çözümünden çıkartıp, $i-1$ ile $i+1$ süperdüğümleri arasındaki en kısa yolu gösteren yolu bulmuşlardır. Daha sonra i noktasını olabildiğince bu çizgiye yaklaştırmışlardır. Bu ekonomizasyon yöntemini her süperdüğüm için uygulamışlardır. Sonuç olarak, döşeme yöntemlerinin ve Steiner bölgesi yönteminin iyi sonuçlar verdiği gözlemlenmiştir.

Diğer bir çalışmada, Steiner bölgesi yöntemi kullanılarak YY-GSP, yeterince yakın araç rotalama problemi ve sıralama tabanlı takım oryantiring problemi için çözümler üretilmiştir [3]. YY-GSP öncelikle konveks olmayan, doğrusal olmayan karışık tamsayılı programlama modeli olarak ifade edilmiştir.

Daha sonra çözüm için Steiner bölgesi sezgiseli (Steiner-zone Heuristic-SZH) adı verilen bir sezgisel algoritma geliştirilmiştir. Yazar, Gulczynski ve diğerleri (2006) kapsamında geliştirilen Steiner bölgesi yönteminden yola çıkan SZH algoritması ile komşuluk alanı disklerinin kesişimlerini belirleyerek, bu alanlara Steiner bölgesi (*Steiner Zone – SZ*) adını vermiştir. Bu bölgeler üzerinden seçilen noktalar ile tur oluşturan çalışma, bir sonraki aşamada, bulunan tur sürelerini, geliştirilen iyileştirme yöntemi ile kısaltmaya çalışmıştır. Daha sonra kendi sonuçlarını, belirlenen süperdüğüm kümesi üzerinde 15 farklı çözüm yöntemi ile alınan çözümleri ile karşılaştırmıştır. Çalışma, daha önce tanımlanmış bir örnek

veri seti bulunmadığı için, problemde kullanılmak üzere kendi veri setini üretmiş, ayrıca TSPLIB üzerinden edinilen veri setleri de çözüm elde etmek için kullanılmıştır. Düğüm sayıları, depo dahil 17 ile 1001 arasında değişen, ayrıca yarıçaplarına göre kesişim oranları farklı olan, toplamda 48 veri seti SZH algoritmasında kullanılmıştır.

SZH algoritmasının birinci aşamasında kullanılan ilk yaklaşımda, bütün SZ'ler belirlenmiş, bulunan SZ'ler içerisinde, bütün düğümler için en yüksek dereceli olan SZ kaydedilmiş, bu SZ'nin kapsadığı düğümler değerlendirmeden çıkarıldıktan sonra geriye kalan düğümler üzerinden, kapsanmamış düğüm kalmayana kadar algoritma çalışmaya devam etmiştir. Bu yöntemin yavaş ve fazla işlem gücü gerektiği belirtilmiştir. Diğer versiyonda ise, en yüksek dereceli SZ'nin kaydedilmesinin yanı sıra, bu SZ'yi oluşturan, en az 3. dereceden olan SZ'ler de kaydedilmiş, sonrasında algoritma bir önceki versiyonda olduğu gibi devam etmiştir. Bu versiyonun daha hızlı olduğu ve daha iyi sonuçlar verdiği belirtilmiştir.

İkinci aşamada, bu SZ'ler içerisinde GSP çözümünde kullanılacak noktaların belirlenmesi yer almıştır. Bunun için i düğümünün k derecesindeki SZ'sinin, kesişim açılarının gösterdiği noktalar arasında düz bir çizgi çekilmiş, bu düz çizginin orta noktası çözüm için kullanılmıştır. Tek dereceye sahip SZ'ler için ise, SZ sahibi i düğümünün komşuluk alanı diski üzerinde, depo noktasına en yakın nokta seçilerek çözüme katılmıştır. Bu süreç ile oluşturulan süperdüğüm kümesinden, Concorde TSP Solver kullanılarak, en az maliyetli tur üretilmiştir.

Algoritmanın son aşaması tur iyileştirme olarak adlandırılmıştır. Süperdüğümlerin elde edildiği SZ sıralamasının sabit tutulduğu bu aşamada, gezici çokgen problemi (*touring polygons problem*) çalışmaya uyarlanarak gezici Steiner bölgeleri problemi (*touring Steiner zones problem*) olarak ağ üzerinde kullanılmıştır. Bu problem, hem yazarın kendi geliştirdiği, SZ'lerin sınırları üzerinde artan büyüklükte sayılarla ayırıştırma yapan yinelemeli bir iyileştirme sezgiseli ile hem de ikinci dereceden koni programlama modeli olarak CPLEX üzerinde çözülmüştür. Bu iki farklı yaklaşım, sırasıyla SZ1 ve SZ2 olarak adlandırılmıştır.

Çalışma, farklı yazarlardan alınmış 15 farklı çözüm yöntemi ile kendi SZ1 ve SZ2 çözümlerini karşılaştırmıştır. Bu karşılaştırmalar, tur maliyeti, çözüm yöntemi ile bulunan en iyi çözümden sapmaları ve çözüm süreleri üzerinden yapılmıştır. Karşılaştırmalar

üzerinde yapılan yorumlarda, genel olarak Steiner bölgesi yönteminin çözümlerde kullanılmasının çözümü ve çözüm süresini iyileştirdiği yorumlanmıştır. Ayrıca, çalışma kapsamında geliştirilen SZH yönteminin iki versiyonu SZ1 ve SZ2'nin sapma ve çözüm süresi açısından değerlendirildiğine diğer çözümlerden üstün olduğu belirtilmiştir. Sonuç olarak Steiner bölgesi içeren çözüm yöntemlerin kısa zamanlarda yüksek kaliteli sonuçlar verdiği söylenmiştir.

2018 yılındaki çalışmada [11] yazarlar, problem çözümü için Steiner bölgelerini kullanmışlardır. Çözüm sezgiseli olarak Steiner bölgesi değişken komşu arama (*Steiner-zone variable neighborhood search-SZVNS*) adını verdikleri bir sezgisel geliştirilmiştir. Üç farklı aşamadan oluşan bu sezgiselde, ilk aşamada bazı düğümler, kendi komşuluk alan diskleri, başka komşuluk alanı diskleri tarafından kapsandığı için problemden çıkartılmıştır. Bu aşamaya veri temizleme adı verilmiştir. Kurulum adı verilen ikinci aşamada, çizgi süpürme adı verilen bir yöntem ile sadece bulununca problem çözümünü iyileştirebilecek Steiner bölgelerinin hesaplanmış, bu sayede çözüm zamanı ve kalitesi iyileştirilmeye çalışılmıştır. Daha sonra, en az sayıda Steiner bölgesi kullanmak adına, bulunan bölgeler için küme kaplama problemi çözülmüştür. Küme kaplama probleminde bulunan Steiner bölgesi kümesi için, son aşamada üç farklı yöntem ile çözüme gidilmiştir. İlk yöntemde, çözüm üretmek adına Steiner bölgelerinin merkez noktaları seçilirken, ikinci yöntemde bölgelere ait noktalardan, depo noktasına en yakın olan noktalar çözüm için seçilmiştir. Son yöntemde ise, değişken komşu arama yöntemi kullanılarak problem çözülmüştür. Son aşamada, bulunan sonuçları iyileştirmek için, altı farklı yöntem geliştirilmiş, bu yöntemler, yapılabilecek herhangi bir iyileştirme kalmayana kadar yinelemeli olarak sırayla uygulanmıştır.

Sonuç olarak 842 farklı veri setiyle geliştirilen algoritmayı çalıştırmışlardır. Çözümü bilinen küçük veri setlerinde, bu yöntem ile %93.2 oranında optimal çözümün bulunduğunu belirtmişlerdir. Çözümü bilinmeyen veri setlerinde ise, geliştirdikleri sezgisel yöntemin, daha önceki sezgiseller ile karşılaştırılabilir performans gösterdiğini belirtmişlerdir.

2019 yılındaki bir çalışmada [12], zaman aralıkları, YY-GSP'ye entegre edilmiştir. En yakın komşu arama sezgiselini temel alan, çevresel komşu arama zaman aralığı (*perimeter neighborhood search time window*) adı verilen bir algoritma kullanmışlardır. Üç farklı aşamadan oluşan bu algoritmada, ilk aşama düğümlerin ziyaret edilmesi gereken zamanlara

göre sıralanmasıdır. İkinci aşamada, vurma noktaları kavramı, iki komşu düğümün, klasik bir GSP çözümünde oluşturdukları kenar ile bu kenarın, komşu düğümlerin komşuluk alan disklerinin çevrelerini kestiği noktalar olarak tanımlanmıştır. Son aşamada, bu noktalar, klasik zaman aralıklı GSP çözümleriyle iyileştirilerek çözüm elde edilmiştir. 21 farklı veri setinde uygulanan algoritma bu veri setlerinin 20 tanesinde optimal çözüme ulaşmıştır. Çözüm süresinin ise veri seti büyüklüğüne bağlı olabileceğini belirtmişlerdir.

2.4. YY-GSP İçin Matematiksel Model

Mennell (2009), YY-GSP'yi karışık tam sayılı doğrusal olmayan matematiksel programlama modeli olarak inceleyip, aşağıdaki formülasyonu geliştirmiştir.

$$\text{Minimize } \sum_{i=0}^n \sum_{j=0}^n e_{ij} \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \quad (2.1)$$

Subject to

$$(x_i - a_i)^2 + (y_i - b_i)^2 \leq r^2 \quad i = 1, \dots, n \quad (2.2)$$

$$\sum_{i=0}^n e_{ij} = 1 \quad i, j = 0, \dots, n \quad (2.3)$$

$$\sum_{j=0}^n e_{ij} = 1 \quad i, j = 0, \dots, n \quad (2.4)$$

$$\sum_{j=0}^n e_{ij} \leq |S| - 1 \quad |S| \subseteq \{1, \dots, n\}, 2 \leq S \leq n \text{ için} \quad (2.5)$$

$$\sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \geq \epsilon \quad i, j = 0, \dots, n \quad (2.6)$$

$$e_{ij} \in \{0, 1\}, (x_i, y_i) \in R \times R \quad i, j = 0, \dots, n \quad (2.7)$$

e_{ij} , i ile j düğümlerinin rota sıralamasında arka arkaya gelmesi halinde 1, aksi halde 0 değerini alan ikili bir karar değişkenidir. x_i ve y_i ise i düğümünü ziyaret etmek için kullanılacak, düğümün komşuluk diskinde yer alan noktanın koordinatlarını ifade etmektedir. Amaç fonksiyonu (Eş. 2.1) tur uzunluğunu minimize etmektedir. Ancak uzunluk katsayıları sabit birer uzunluk değil, kullanılan düğümler için seçilen noktaların birbirine olan uzaklığının fonksiyonlarıdır. İlk kısıt (Eş. 2.2), seçilecek her noktanın, kullanılan düğümün disk komşuluk alanı içerisinde olmasını gerektirmektedir. $i=0$ noktasında disk olmadığı için bu nokta kısıt için kullanılmamaktadır. Eş. 2.3 ve Eş. 2.4 her noktanın yalnızca bir defa ziyaret edilmesini sağlamaktadır. Eş. 2.5 çözüm içerisinde alt turlar oluşmasını engellemek için kullanılmaktadır. Son olarak Eş. 2.6 seçilen iki nokta arasındaki uzaklığın pozitif olmasını sağlamaktadır.

2.5. Çalışmaların Karşılaştırılması

Çizelge 2.1’de, incelenen çalışmaların karşılaştırılması görülebilir. Bu çizelgede çalışmalar; YY-GSP’nin GSP ile karşılaştırılıp karşılaştırılmadığı, içerikte uygulanan yöntemin kesin ya da sezgisel yöntem olup olmadığı, alt-üst sınır çalışması içerip içermediği, YY-GSP için herhangi bir çözüm bulunup bulunmadığı ve YY-GSP için geliştirilmiş herhangi bir matematiksel model bulunup bulunmadığı yönünden incelenmiştir.

Çizelge 2.1. Çalışmaların karşılaştırılması

ÇALIŞMA	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	Bu Çalışma
YY-GSP, GSP KASILASTIRMASI	-	✓	-	-	-	-	-	-	-	-	-	✓
KESİN YÖNTEM	-	✓	-	✓	-	✓	-	-	-	-	-	-
SEZGİSEL YÖNTEM	✓	-	✓	-	✓	-	✓	✓	✓	✓	✓	✓
ALT-ÜST SINIR BULMA	✓	-	-	-	✓	-	✓	✓	-	-	✓	-
YY-GSP ÇÖZÜMÜ BULMA	✓	✓	✓	✓	-	✓	-	-	✓	✓	-	✓
YY-GSP İÇİN MATEMATİKSEL MODEL GELİŞTİRME	✓	✓	-	✓	-	✓	-	-	-	✓	-	-

İncelenen literatürde, YY-GSP için yapılan çalışmalar, GSP ile karşılaştırmada eksik kalmıştır. GSP’ye kıyasla, elde edilebilecek tasarrufların net bir şekilde ortaya konmasının, YY-GSP’nin önemini vurgulamakta önemli olduğu anlaşılmaktadır. Ayrıca, YY-GSP’nin tüm yönleriyle anlaşılabilmesi için yapılacak çalışmaların, daha sonraki çalışmalar için kaynak olacağı da görülmektedir. Bu çalışmada, açıklanan eksikliklerin giderilerek literatüre katkıda bulunulması amaçlanmıştır.

Bu tez çalışmasında YY-GSP, sayaç okuma süreçleri için uyarlanmıştır. Çalışma kapsamında, sezgisel bir yöntem geliştirilmiş, bu yöntem kullanılarak örnek problemler çözüme ulaştırılmıştır. Ayrıca YY-GSP’nin gerçek hayattaki iyileştirme potansiyeli ölçülmüştür. Sonuçlar GSP sonuçları ile karşılaştırılmıştır.

3. YETERİNCE YAKIN KOMŞU ARAMA (YYKA)

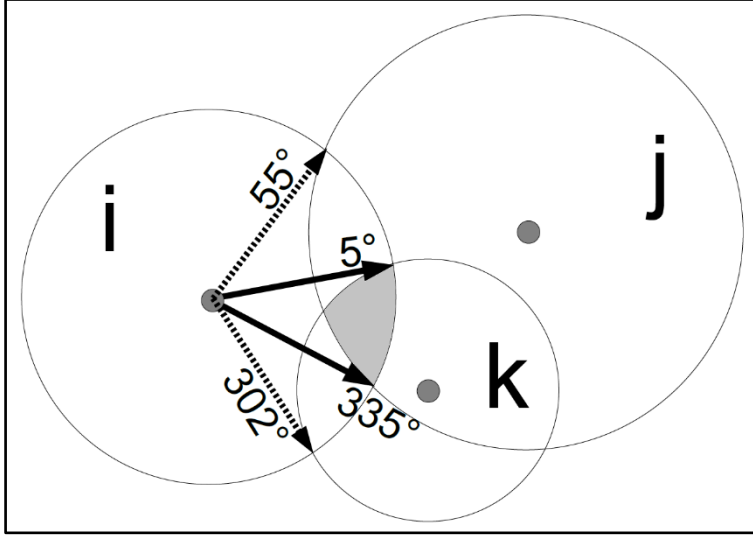
Bu bölümde, çalışma kapsamında geliştirilmiş Yeterince Yakın Komşu Arama (YYKA) sezgiseline ait kavramlar açıklanmış, algoritmanın genel işleyişi ve varsayımlar verilmiştir. Ayrıca örnek bir çözüm görselleştirilmiştir.

Literatürde oluşturulan matematiksel modeller, incelendiği kadarıyla çoğunlukla doğrusal olmayan modellerdir. YY-GSP çözümünde daha önce kullanılan dal-sınır ve ikinci derece koni programlama algoritmaları kullanılabilir. Bu çalışmada YY-GSP'nin çözüm üzerindeki iyileştirme potansiyelini gösterebilmek adına, sezgisel bir yöntem geliştirilmiştir. Daha önce geliştirilmiş olan Steiner bölgesi yöntemi [3, 5] süperdüğüm kümesini oluşturmak ve büyüklüğünü azaltmak için kullanılmıştır. Daha sonra problem için uyarlanmış en yakın komşu arama sezgiseli ile ziyaret edilmesi gereken noktalar belirlenip, klasik GSP çözüm yöntemleri uygulanmıştır.

3.1. Kavramlar

Bu bölümde sezgisel yöntemin anlaşılması adına, Steiner bölgesi ve süperdüğüm kavramları ayrıntılı olarak açıklanacaktır.

Steiner bölgesi: Daha önce [5]'de tanımlandığı üzere, Steiner bölgeleri (*Steiner Zones*), komşuluk alan disklerinin kesiştiği alanları ifade etmektedir. $SZ(k)$, k derecesinden Steiner bölgesi ifadesi, k kadar düğümün komşuluk alanı disklerinin kesişmesi ile oluşmaktadır. i düğümünün $SZ(k)$ bölgesi, k sayıdaki düğüm setinde bulunan diğer düğümlerin komşuluk alanlarıyla kesişimlerinin noktalarının, i düğümüne göre başlangıç ve bitiş açılarıyla tanımlanmaktadır. Şekil 3.1 bu duruma bir örnektir. Boyanmış bölge Steiner bölgesini, çizgili oklar ise j ve k düğümleri ile olan ayrı ayrı kesişim alanları için başlangıç ve bitiş açılarını göstermektedir.



Şekil 3.1. i düğümü için $SZ(3)$ bölgesi [3]

Süperdüğüm: $SZ(k)$ Steiner bölgesinde bulunan herhangi bir nokta, k kadar olan düğüm sayılarının hepsine yeterince yakın denmiştir. Bu noktalara süperdüğüm (*supernode*) adı verilmiştir. Bütün $SZ(k)$ bölgeleri için, uygun birer süperdüğüm seçilerek süperdüğüm kümesi oluşturulmaktadır. Problemin karmaşıklığını olabildiğince azaltmak için, süperdüğümlerin sayısının azaltılması istenir [15]. Bu kümedeki düğümlerin ziyaret edilmesi, $SZ(k)$ bölgesini oluşturan bütün düğümlerin ziyaret edilmesi anlamına gelmektedir.

3.2. Algoritma

Problem, daha önce belirtildiği üzere sayaç okuma süreçleri için uyarlanmıştır. Bu durumda, gerekli teknolojilerin sayaçlara ve sayaç okuma aletlerine entegre edildiği düşünülerek, sayaç okuma personelinin, sayaçlara belli bir uzaklıkta sayaçları okuyabildiği varsayılmıştır. Sayaç okuma personeli satıcı, sayaçlar ise düğümler haline gelmiştir.

Yeterince Yakın Komşu Arama (YYKA) ismi verilen sezgisel yöntemde ilk olarak komşuluk alanı disklerinin birbiriyle kesişme açıları hesaplanarak bütün $SZ(2)$ bölgeleri bulunmuştur. Daha sonra bu bölgelerin birbiriyle kesişimlerine bakılarak, daha üst derecedeki bölgeler belirlenmiştir. Her bir bölge için süperdüğümler belirlenip, bu noktalar arasından GSP çözümünde yer alacak olanlar, en yakın komşu arama sezgiseli ile seçilmiştir. Bu seçim, noktaların kümesinin büyüklüğünü azaltarak GSP çözümünde kullanılacak kümeyi oluşturmaktadır.

3.2.1. Varsayımlar

Çözüm yöntemi için kullanılan varsayımlar aşağıdaki şekilde sıralanmaktadır:

- Her bir veri setinde, komşuluk alan diskleri için kullanılan yarıçap r sabittir ve her bir düğüm için aynıdır.
- $r=0$ durumunda, problem GSP'ye indirgenmektedir.
- Bulunan SZ(2) bölgeleri, konveks bölgelerdir. Dolayısıyla 2. dereceden başlayarak, derecesi küçük bölgelerin kesişimi ile oluşturulan daha büyük dereceli bölgelerin tümü konveks olacaktır.
- Steiner bölgelerinden seçilen süperdüğüm kümesi, her bir bölgenin kesişim başlangıç ve bitiş noktalarını birbirine bağlayan düz çizginin orta noktası seçilerek oluşturulmuştur.
- Derecesi 1 olan Steiner bölgeleri için, süperdüğüm olarak komşuluk alanı diskinin merkez noktası, yani alan sahibi düğümün kendisi kullanılmıştır.
- Bir SZ(k)'yı temsil eden süperdüğüm ziyaret edildiğinde, o bölgeyi oluşturan komşuluk alanlarının sahibi bütün düğümler ziyaret edilmiş sayılacaktır.
- XY koordinat düzleminde (100, 100) noktasında bulunan bir depo, bütün örnek veri setleri için başlangıç ve bitiş noktası olarak belirlenmiştir.
- Depo noktası hiçbir Steiner bölgesine dahil değildir,
- Depo noktasında bulunan satıcı, herhangi bir düğümü ziyaret etmiş sayılmaz.
- Bütün uzaklıklar Öklid uzaklıkları olarak XY koordinat sisteminde hesaplanmıştır.

YYKA sezgiseli için iki farklı varyasyon kullanılmıştır. Bu varyasyonlardan birincisi, problem sırasında bulunulan süperdüğümünden bir sonraki süperdüğüme giderken, daha önce ziyaret edilmiş olan bir düğümü içeren aday noktaya gidilmesini engellemektedir. İkinci varyasyon ise bu duruma izin vermektedir. Bu iki varyasyonlardan bahsederken sırasıyla YYKA1 ve YYKA2 ifadeleri kullanılacaktır. Ayrıca YYKA sezgiseli, kullanılabilecek en yüksek SZ derecesini kullanarak çalışmaktadır. Bu durumun nedeni, önceki çalışmalarda gösterildiği üzere, daha yüksek dereceli SZ'lerin ziyaret edilmesinin, ziyaret edilecek süperdüğüm sayısını azaltması dolayısıyla çözüm uzayını daraltıp daha kısa sürede daha iyi sonuçlar elde edilmesine olanak sağlamasıdır. Daha sonra ziyaret edilmesi öngörülen bu noktaların oluşturduğu ağ için Concorde TSP kullanılarak çözüm elde edilmiştir. YYKA algoritmasına ait bir akış şeması, EK-1'de görülebilir.

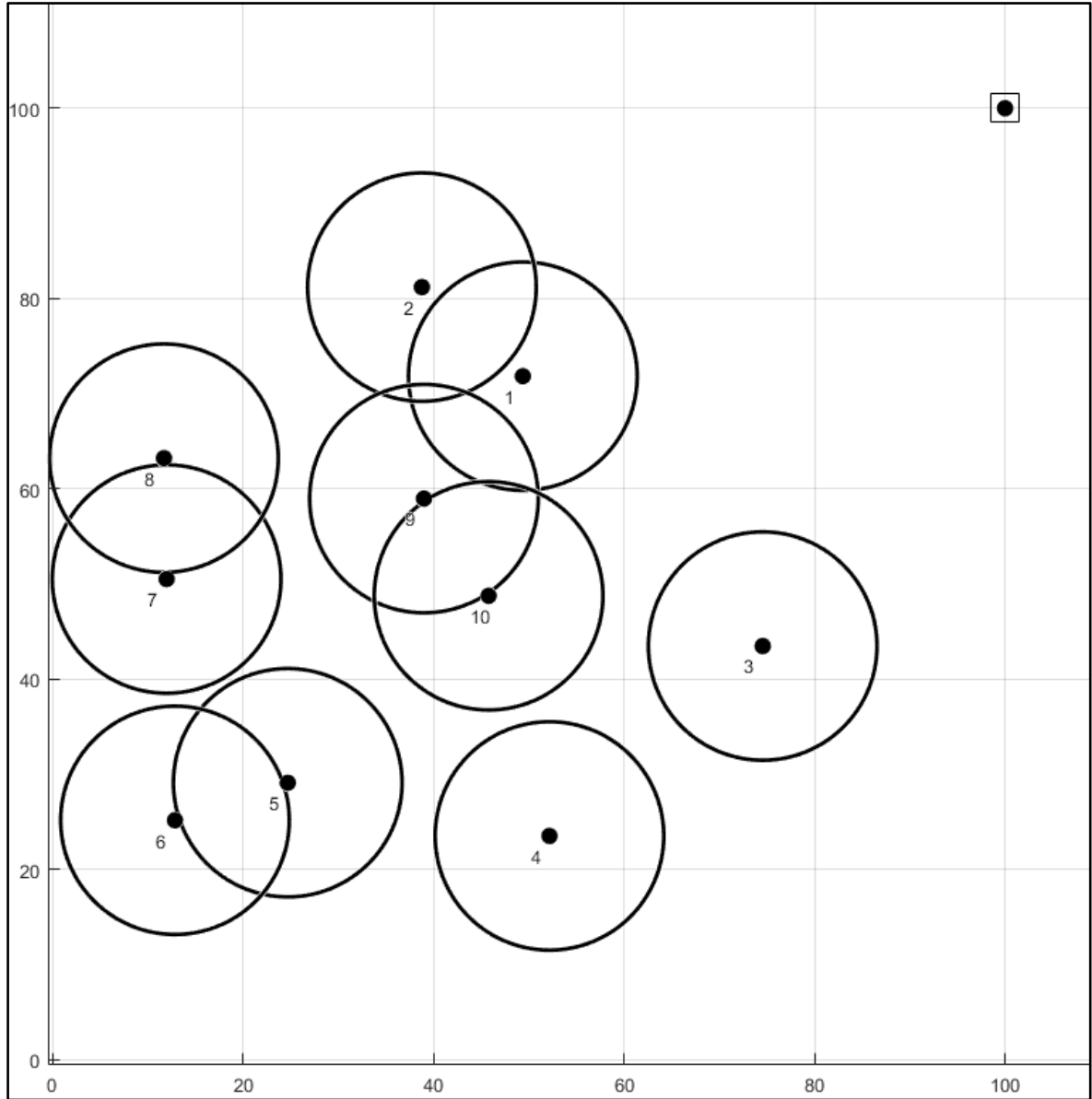
3.3. Örnek Çözüm

Örnek olarak, 10 düğümden oluşan bir veri seti için çözüm yapılmıştır. Bu çözüm aşağıda adım adım açıklanmıştır. Çizelge 3.1’de, kullanılan veri setindeki düğümlerin 2 boyutlu koordinat düzleminde bulundukları konumların X ve Y değerleri verilmiştir. Ayrıca, çalışmada kullanılan diğer veri setlerinde olduğu gibi, (100, 100) noktası, başlangıç ve bitiş olarak kullanılmak üzere depo konumu olarak belirlenmiştir.

Çizelge 3.1. Örnek çözüm veri seti düğüm koordinatları

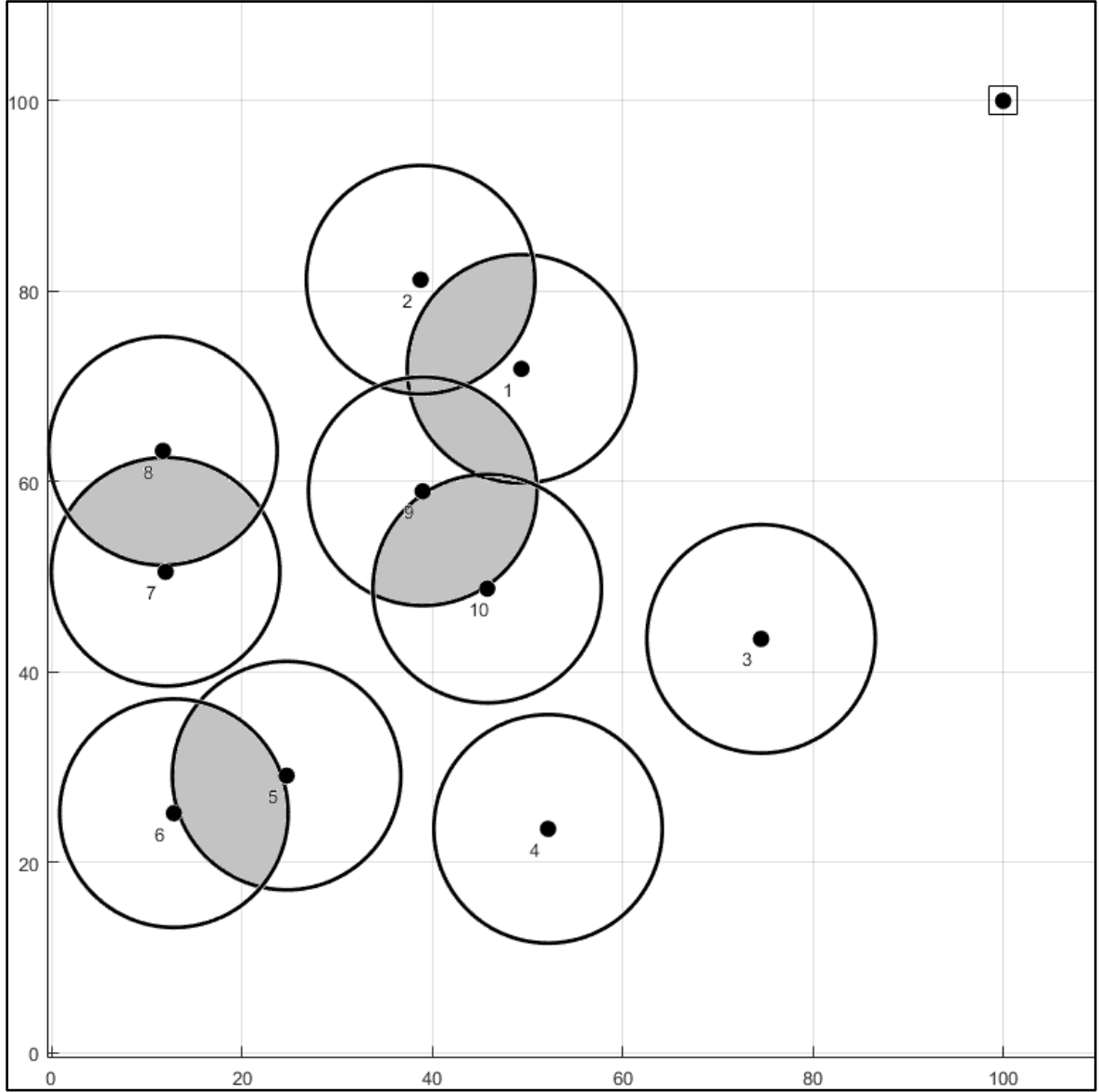
Düğüm	X	Y
1	49,38151	71,84628
2	11,71157	63,23152
3	74,5573	43,4874
4	45,78512	48,76667
5	11,99216	50,52823
6	24,70444	29,12366
7	39,00311	58,98115
8	12,86813	25,17705
9	38,78698	81,20736
10	52,1759	23,53475

Şekil 3.2’de örnek çözüm veri setinin ağ yapısı görülebilir. Kare içerisinde yer alan sağ üst köşedeki nokta, depodur. Satıcı, depo noktasından yola çıkacak, bütün düğümleri ziyaret ettikten sonra buraya geri dönecektir. Siyah noktalar, disklerin merkezlerini yani müşterilerin bulunduğu düğümleri göstermektedir. Bu düğüm noktaları, görünürlük açısından büyütülmüştür. Müşteri düğümlerinin sol altında yer alan sayılar, düğümlerin numarasıdır.



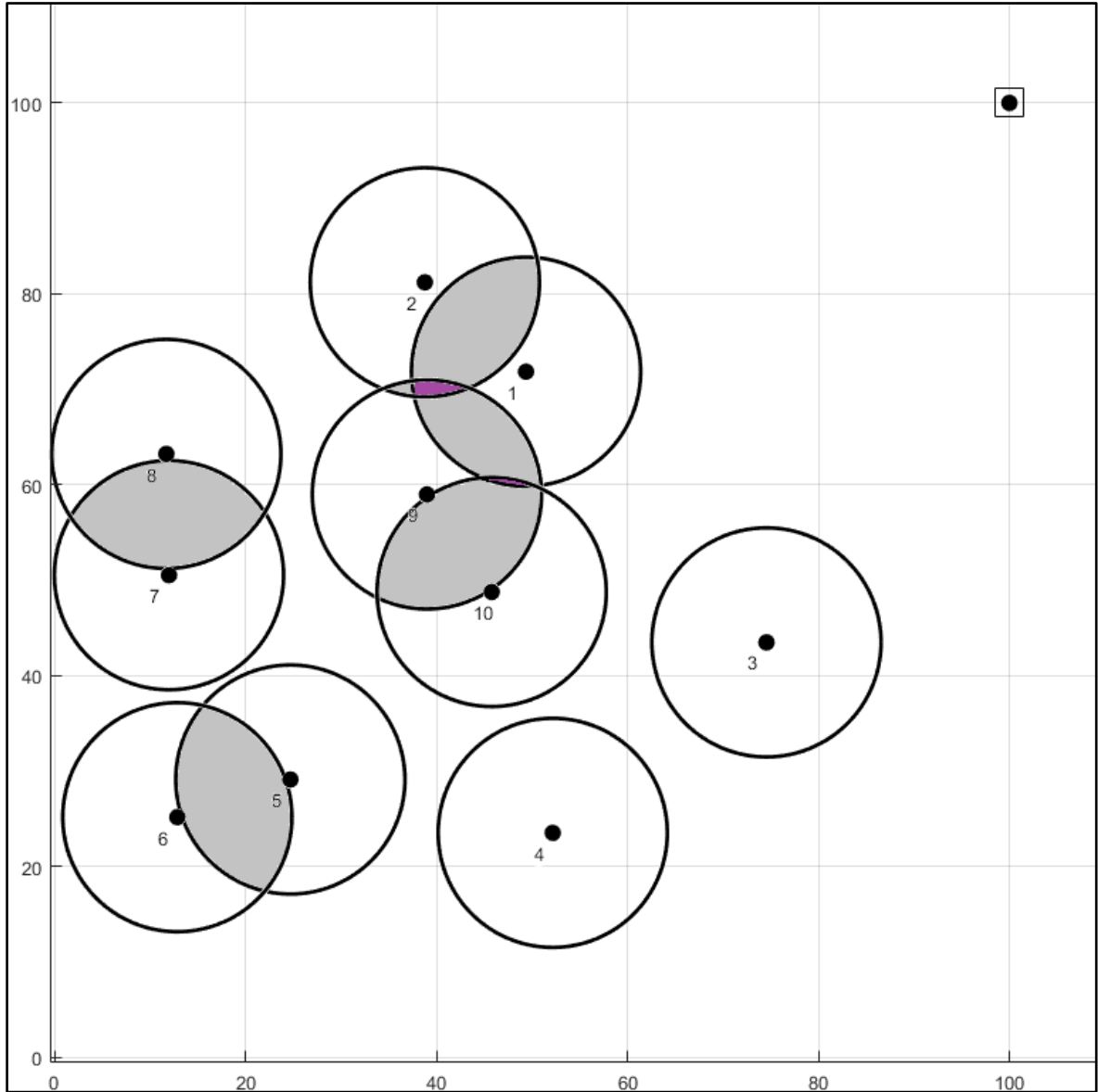
Şekil 3.2. Örnek çözüm veri seti ağ yapısı

Disklerin arasındaki kesişimlerin bulunması ile oluşturulan $SZ(2)$ bölgeleri Şekil 3.3'te gri renkle gösterilmiştir. Bu bölgelerde bulunan her bir nokta, iki adet düğümün komşuluğunda bulunmaktadır. Yedi adet $SZ(2)$ bulunmaktadır. Disklerin alanı içinde bulunan ancak gri olmayan bölgeler ise 1. dereceden Steiner bölgeleridir. Bu bölgelerden de 10 adet bulunmaktadır.



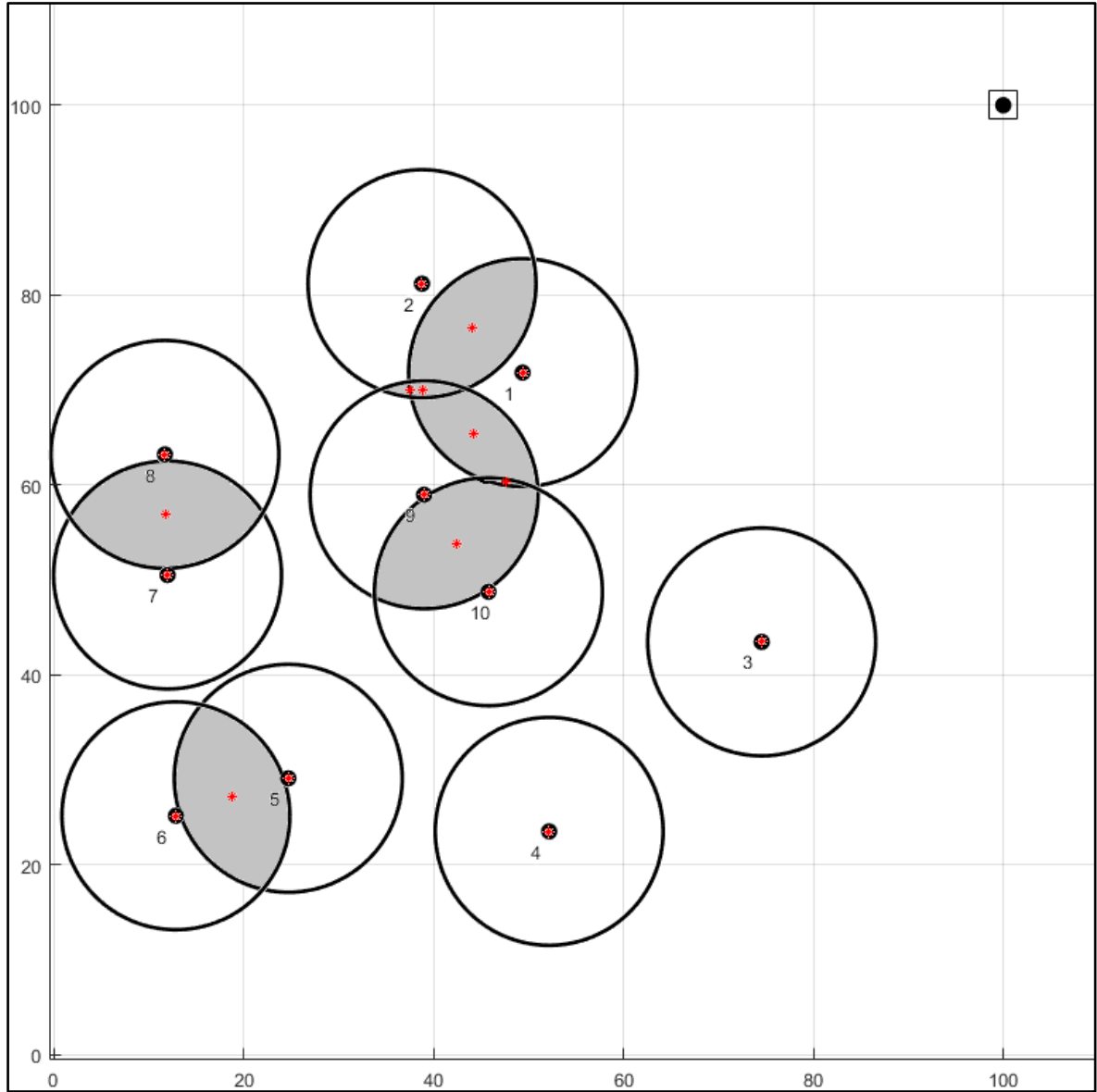
Şekil 3.3. $SZ(2)$ bölgeleri

$SZ(3)$ bölgeleri ise Şekil 3.4'te bulunmaktadır. Mor renkle gösterilen bölgelerde her bir nokta, 3 farklı düğümün komşuluğunda birden bulunmaktadır. İki adet bulunan $SZ(3)$ 'lerin biri $\{1, 2, 9\}$, diğeri ise $\{1, 9, 10\}$ düğümlerinin komşuluk alanlarından oluşmaktadır.



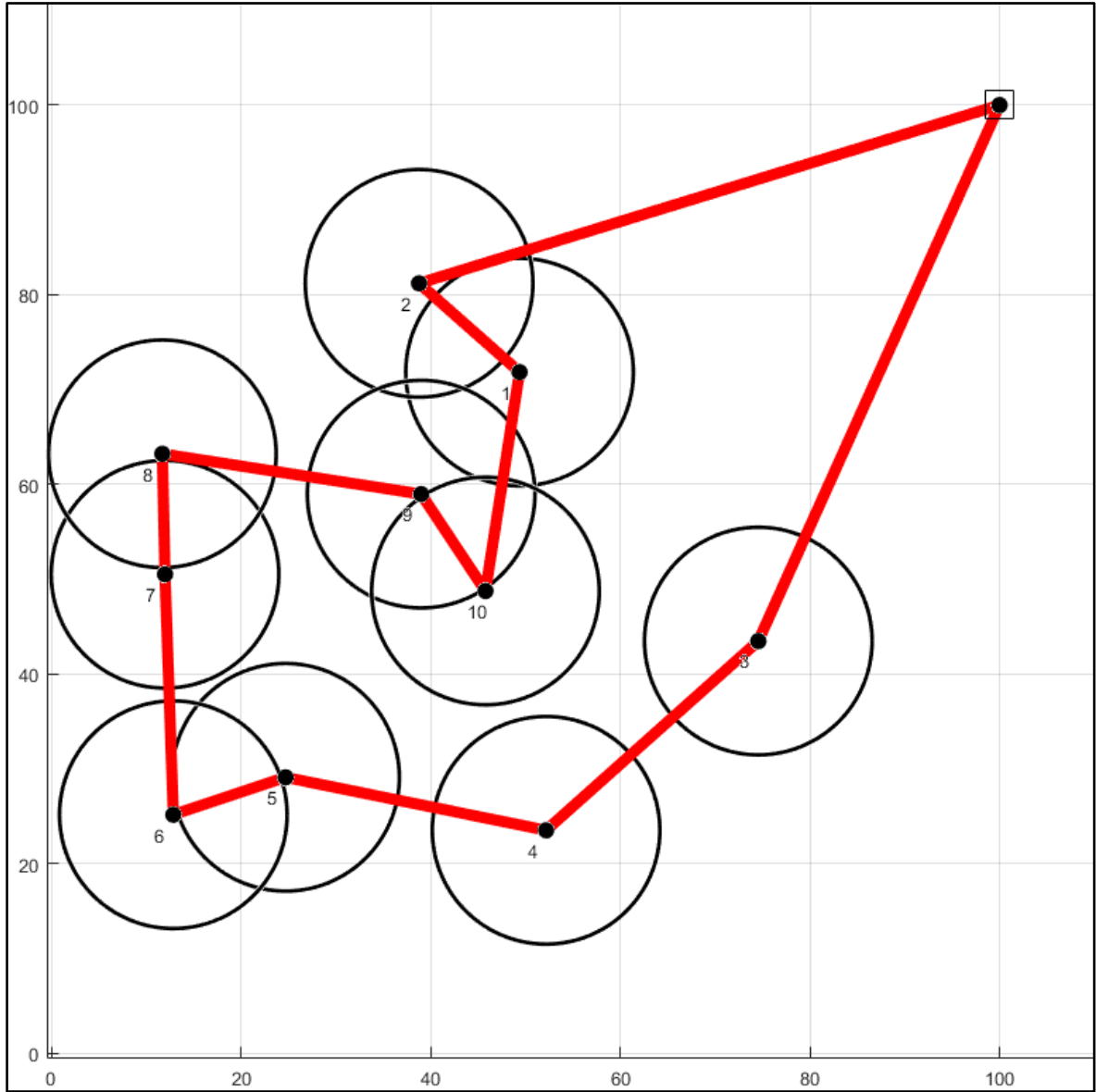
Şekil 3.4. $SZ(3)$ bölgeleri

Bu bölgelerden elde edilen süperdüğümler, kırmızı yıldızlar şeklinde Şekil 3.5'te görülebilir. Süperdüğüm kümesi, her bir SZ'nin başlangıç ve bitiş noktalarının birleştiren çizginin orta noktalarından oluşturulmaktadır. 1. dereceden SZ'ler için ise SZ sahibi düğümün kendisi süperdüğüm kümesine eklenmektedir.



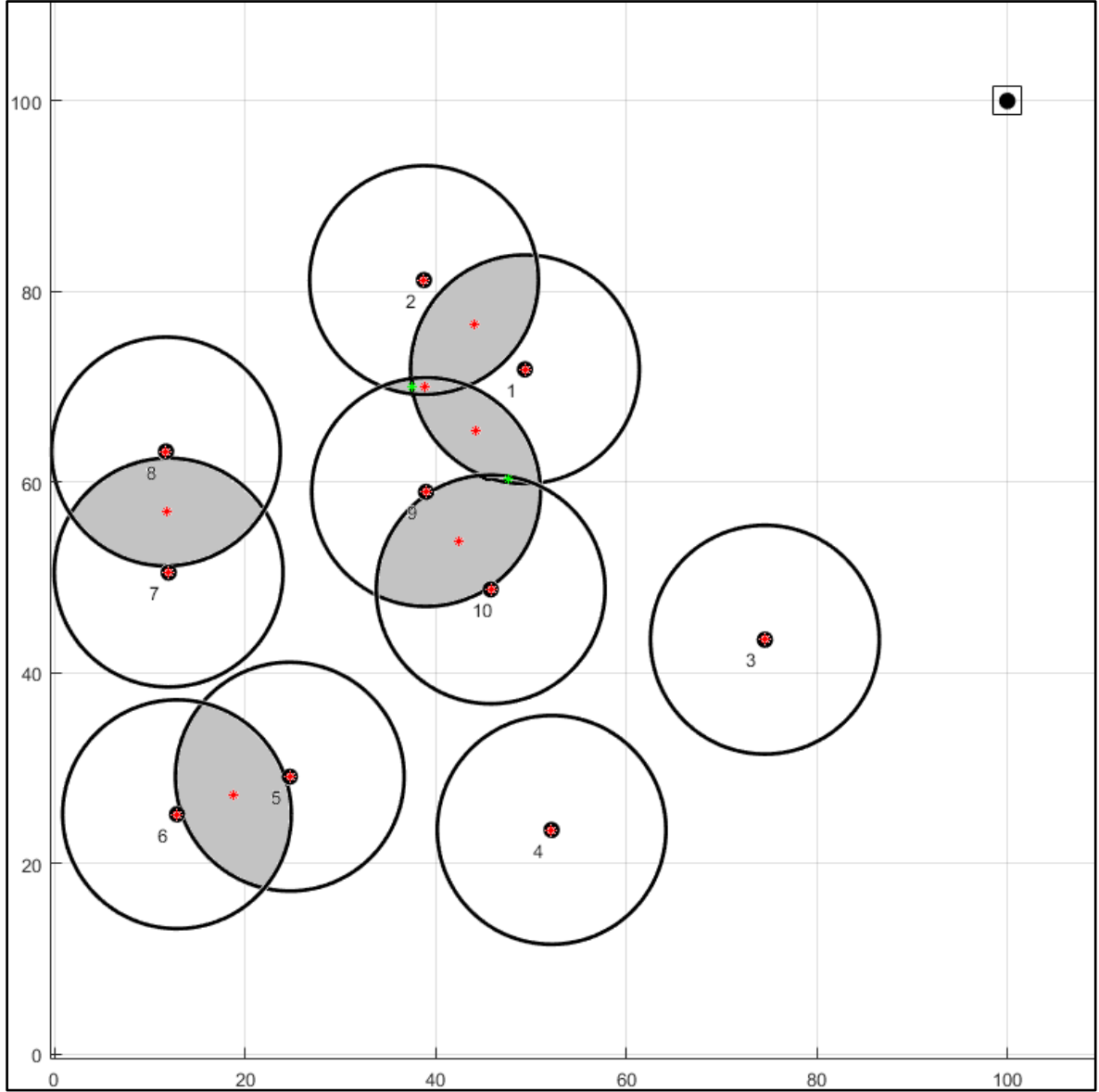
Şekil 3.5. Süperdüğümler

Müşteri düğümleri üzerindeki klasik GSP çözümü Şekil 3.6'da incelenebilir. Kırmızı çizgilerle gösterilen rota ile 311 birimlik bir tur maliyeti elde edilmiştir. Depodan başlanarak her düğüm, klasik GSP'de olduğu gibi tek tek ziyaret edilmiş ve depoya geri dönmüştür.



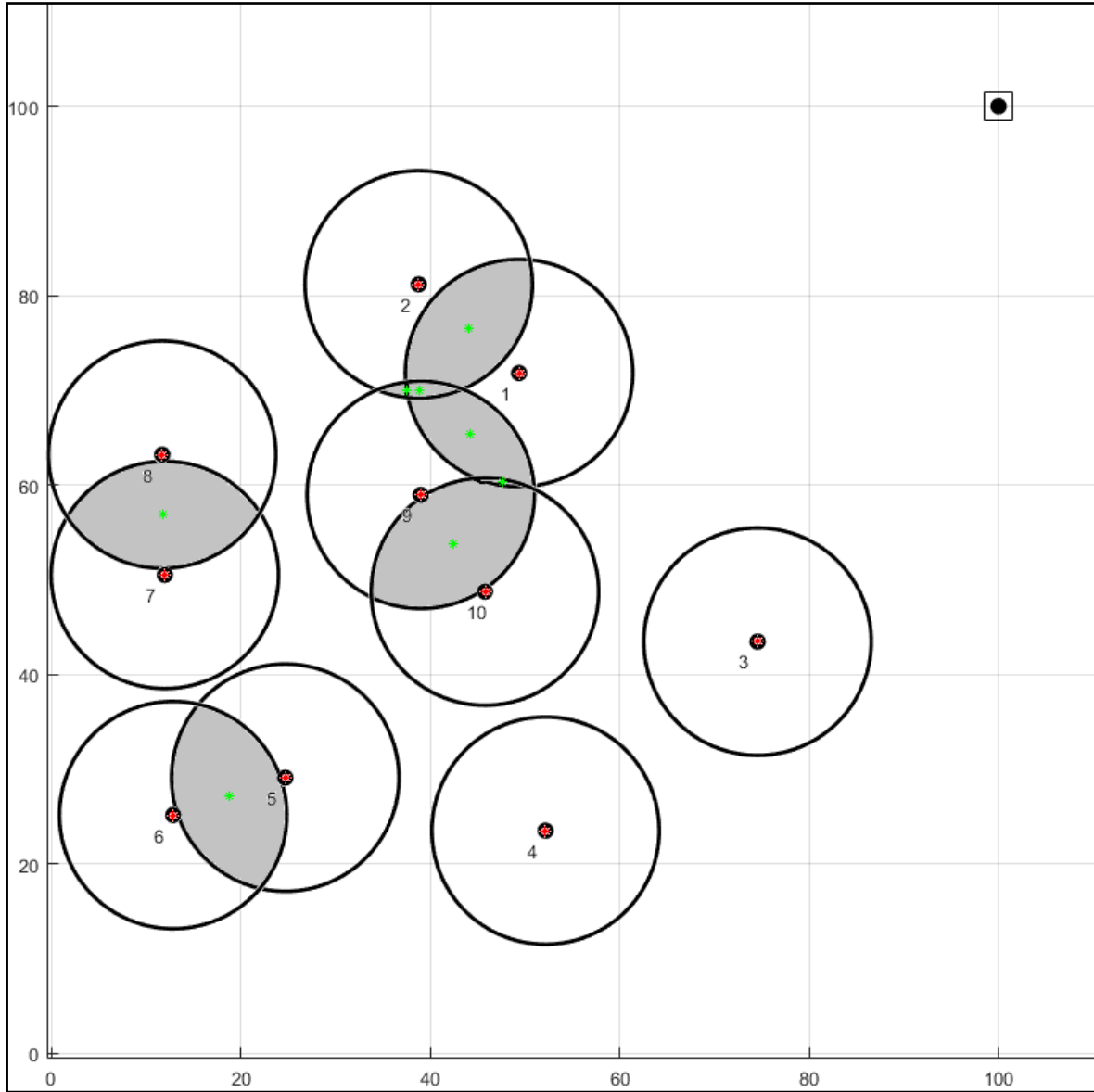
Şekil 3.6. Klasik GSP turu

YYKA algoritması, en büyük SZ derecesinin kullanarak aramasına başlamakta, sonuç bulunana kadar her adımda bir derece daha düşük dereceli SZ'leri kontrol etmektedir. Şekil 3.7'de, 3. Derece, yani bu problemdeki en yüksek SZ derecesinden başlayan YYKA algoritmasının kullanabileceği süperdüğümler bulunmaktadır. Yeşil renkli yıldızlar, bu seviyede tur oluşturmak için kullanılabilir süperdüğümleri gösterirken, daha düşük dereceli SZ'lerde bulundukları için kullanılamayan süperdüğümler kırmızı renkle gösterilmiştir. Kullanılabilir düğümlere gidilmesi ile sadece $\{1, 2, 9, 10\}$ düğümleri ziyaret edilebilir. Geriye kalan $\{3, 4, 5, 6, 7, 8\}$ düğümleri ziyaret edilemediği için 3. dereceden SZ'ler ile çözüm bulunamamaktadır.



Şekil 3.7. $SZ(3)$ seviyesindeki süperdüğümler

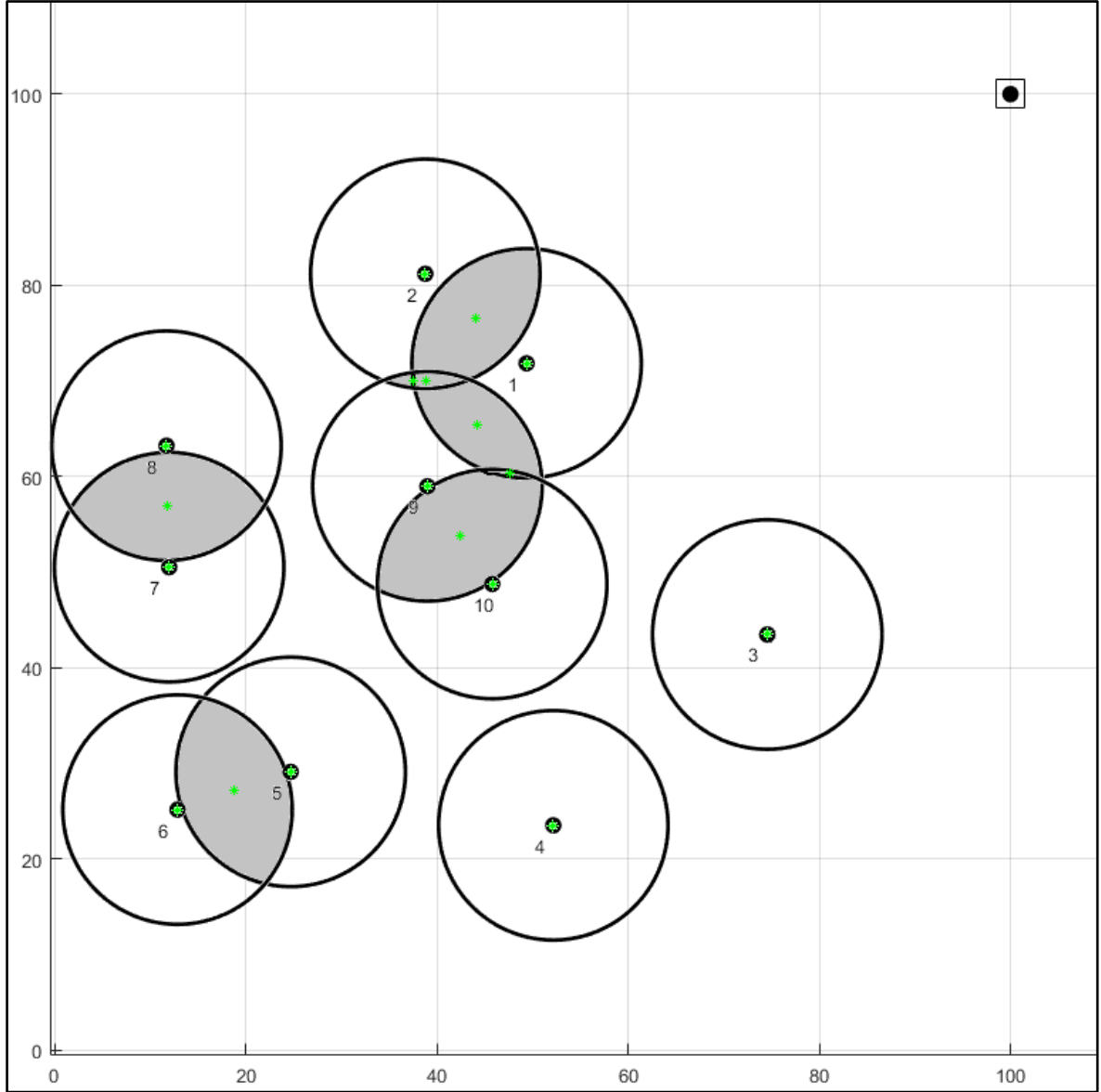
Bir sonraki seviyede, $SZ(3)$ ve $SZ(2)$ bölgelerinde yer alan süperdüğümler kullanılabilir haldedir. Şekil 3.8’de yeşil renkle gösterilen süperdüğümler $\{1, 2, 5, 6, 7, 8, 9, 10\}$ düğümlerinin komşuluğundadır. Ancak $\{3, 4\}$ düğümleri halen herhangi bir şekilde ziyaret edilemediği için $SZ(2)$ seviyesinde de çözüm yoktur.



Şekil 3.8. SZ(2) seviyesindeki süperdüğümler

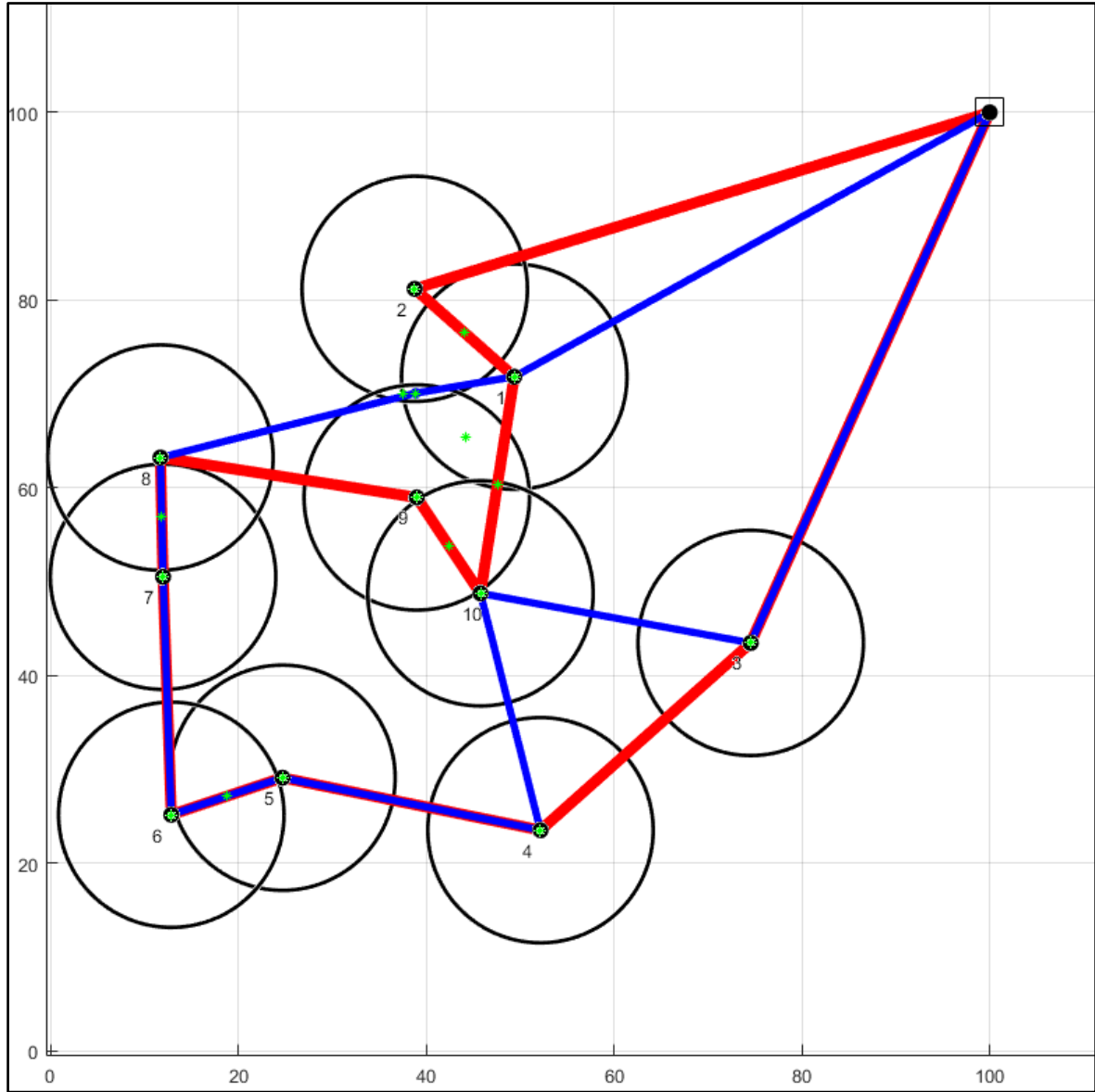
SZ(1) seviyesinde ise artık her düğüm ziyaret edilebilir hale gelmiştir. Bütün süperdüğümler yeşil renk ile gösterilip, gidilebilir durumdadır. Şekil 3.9’da 1. derece SZ’ler kullanılarak oluşturulmuş süperdüğüm kümesi görülebilir.

Süperdüğüm setleri ve kullanılacak en yüksek SZ dereceleri, YYKA1 ve YYKA2 için, aday süperdüğüm seçimi kurallarının farklı olması dolayısıyla farklı şekilde bulunabilir. Bu örnekte her iki versiyon aynı dereceyi kullandığı için süperdüğüm setleri aynı olmuştur.



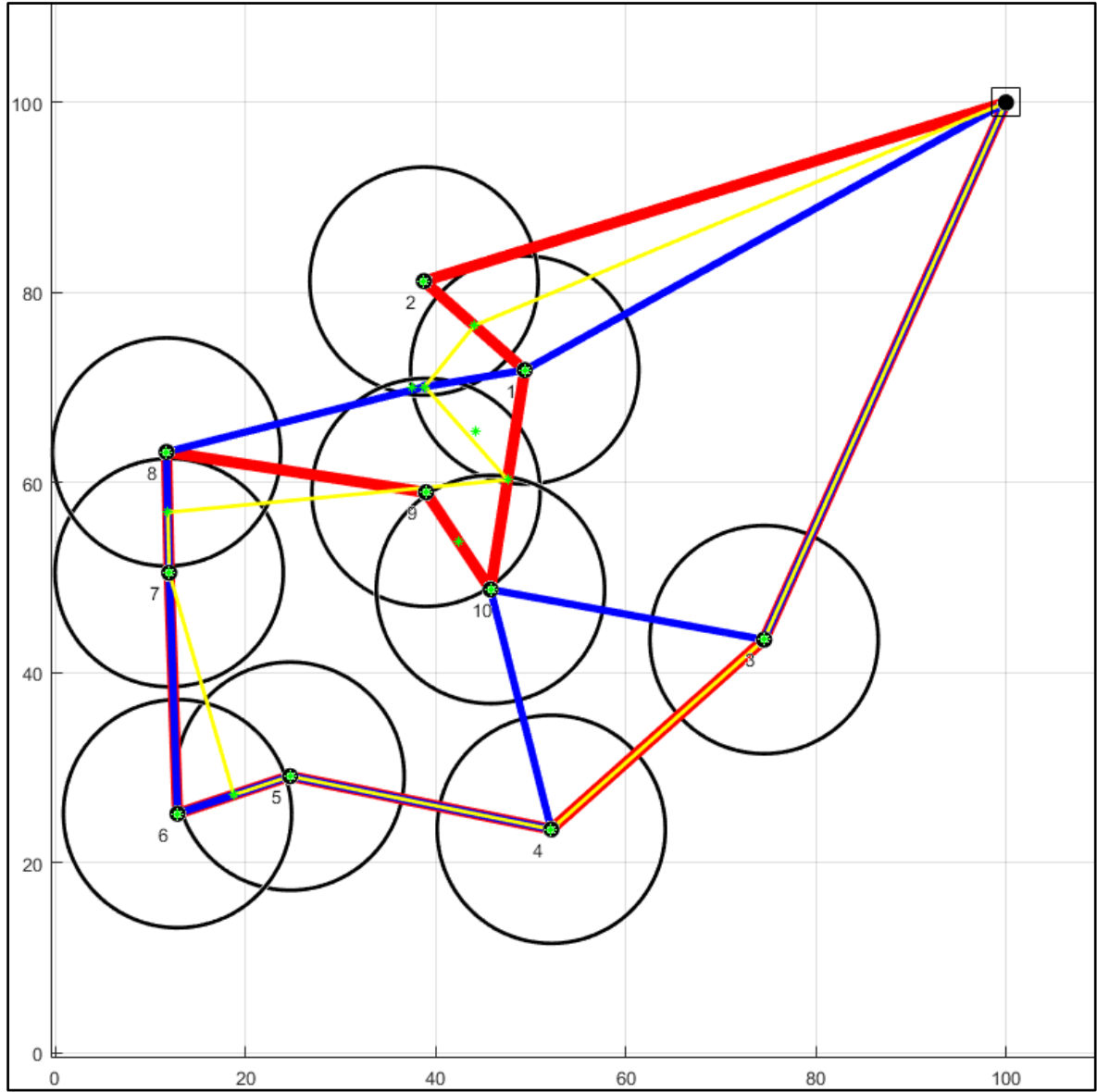
Şekil 3.9. SZ(1) seviyesindeki süperdüğümler

İlk olarak YYKA1 çözümü uygulanmıştır. Bu algorithmada, bulunulan süperdüğümden bir sonraki süperdüğüme geçişte, aday süperdüğümün daha önce ziyaret edilmiş bir düğümün komşuluğunda bulunmaması gereklidir. Depodan başlanılarak, en yakındaki süperdüğüme gidilmiş, daha sonra yukarıda belirtilen kurala uygun olarak bir sonraki ziyaret edilecek süperdüğümler seçilerek çözüm kümesi oluşturulmuştur. Bu çözüm kümesini kullanarak Concorde TSP Solver ile klasik GSP çözümü elde edilmiştir. Şekil 3.10'da, mavi çizgilerle belirtilen rota YYKA1 ile elde edilen süperdüğümlerin oluşturduğu rotayı göstermektedir. Tur maliyeti 292 olarak bulunmuştur. Tur maliyetinde iyileştirme oranı %6,11 olmuştur.



Şekil 3.10. YYKA1 turu

YYKA2 çözümünde, gidilecek bir sonraki süperdüğümde aranan tek şart, daha önce ziyaret edilmemiş en az bir düğümün komşuluğunda olmasıdır. Depo başlangıç ve bitiş noktası olarak belirlenmiş, bütün düğümler ziyaret edilene kadar çözüm kümesi genişletilmiştir. Daha sonra yine Concorde TSP Solver kullanılarak 278 birim tur maliyeti elde edilmiştir. Hesaplanan tur maliyeti iyileştirme oranı %10,61 olmuştur. Şekil 3.11, YYKA2 turunu sarı renk ile göstermektedir.



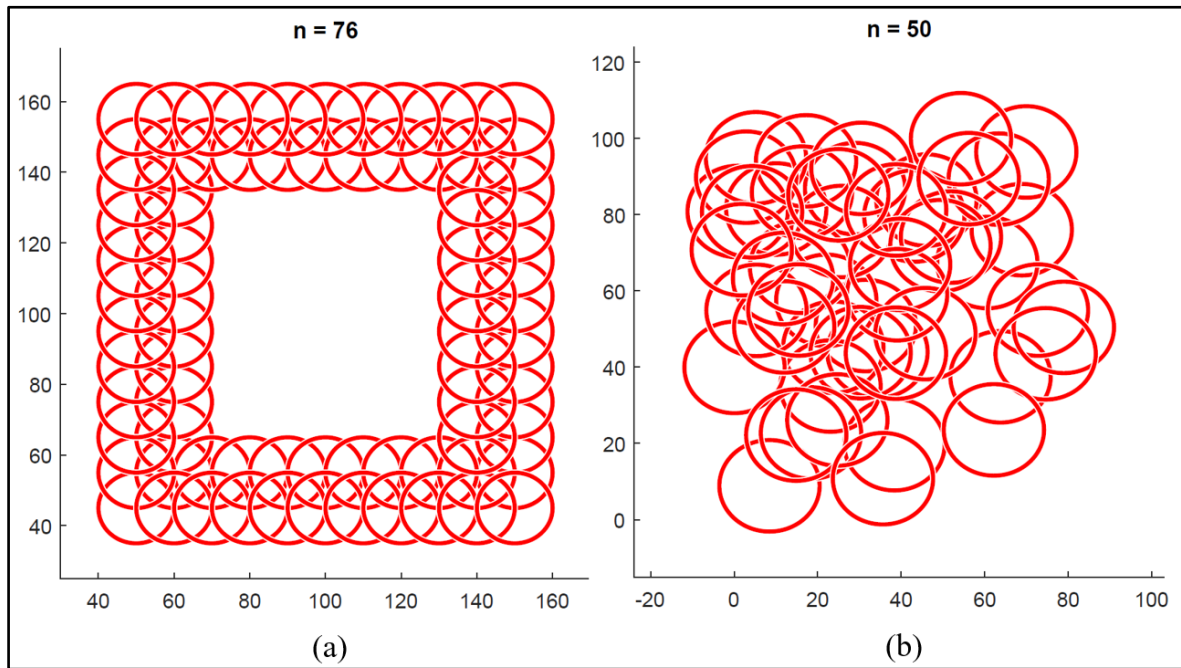
Şekil 3.11. YYKA2 turu

4. HESAPLAMA SONUÇLARI

Bu bölümde kullanılan veri setleri açıklanıp, algoritmanın çalıştırma sonuçları verilmiştir. Çalışmada elde edilen sonuçlar çizelge halinde özetlenmiş ve yorumlanmıştır.

4.1. Kullanılan Veri Setleri

YYKA algoritması, düzenli ve düzensiz olarak adlandırılan iki grup halinde, toplam 22 ayrı örnek veri seti için koşturulmuştur. Düzenli ve düzensiz örnek veri setleri için ağ yapısı grafik gösterimi Şekil 4.1’de incelenebilir.

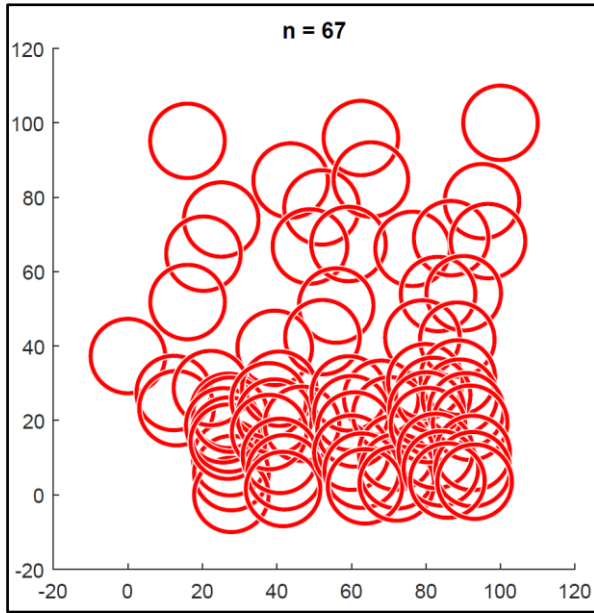


Şekil 4.1. (a) *b2-76* veri seti için ağ yapısı, (b) *bn7-50* veri seti için ağ yapısı

Düzenli olarak adlandırılan veri setleri, belli bir düzende dizilmiş düğümlerden ve bu düğümlerin komşuluk disk alanlarından oluşmaktadır. Bu veri setleri [3]’de oluşturulmuş veri setlerinden seçilmiştir. Düzensiz veri setlerinde ise belirli bir düzen yoktur, düğümler koordinat sisteminde, diğer düğümlerin konumundan bağımsız olarak herhangi bir yerde bulunabilirler. Bu veri setleri, 1000 farklı düğümün bulunduğu bir ağ içerisinde 50 adet düğümün rastgele seçilmesi ile oluşturulmuş ağlardır.

4.1.1. Uygulama

Gerçek hayat uygulaması olarak, Türkiye'nin Ankara ilinde, bir bölgedeki yerleşim alanı kullanılmıştır. 67 düğümden oluşan ağda, gerçek hayatı doğru yansıtmak adına apartmanlar, kümelenmiş müstakil evler ve işyerlerinin bulunduğu bir alan seçilmiştir. Şekil 4.2'de gerçek hayat çözümünde kullanılan veri setinin ağ yapısı incelenebilir. Gerçek hayat uygulaması, belli bir düzen takip etmediği için düzensiz veri setleri altında verilmiştir.



Şekil 4.2. *bati67* veri seti için ağ yapısı

4.2. Bulgular ve Yorum

Visual basic dilinde kodlanmış YYKA1 ve YYKA2 sezgiseli, çözüm uzayından uygun noktaların seçilmesi için çalıştırılmıştır. Bütün koşmalar 4 çekirdek ve 3100 MHz saat hızına sahip işlemci, 16 GB RAM ve Windows 10 işletim sistemi kullanılarak yapılmıştır. Koşmaların sonuçları, süreler ve karşılaştırmalar, düzenli ve düzensiz veri setleri için aşağıdaki çizelgelerde incelenebilir.

Çizelge 4.1'de düzenli veri setleri için YYKA1 kullanılarak elde edilen tur maliyetleri, hesaplama süreleri ve iyileştirme oranları verilmiştir. YYKA1 için ortalama hesaplama süresi 2,391 saniye olmuştur. Ortalama iyileştirme oranı ise %18,460 olarak hesaplanmıştır. En yüksek oranlı iyileştirme, b3-126 probleminde 1,616 saniye hesaplama süresi ile

%49,699 olarak bulunmuştur. Bu oran ayrıca çalışmada bulunan en yüksek değerli iyileştirme oranıdır.

Çizelge 4.1. Düzenli veri setleri için YYKA1 tur maliyetleri

Problem	Düğüm Sayısı	Klasik GSP Çözüm	YYKA1 Çözüm	YYKA1 Süre (s)	YYKA1 İyileştirme Miktarı	YYKA1 İyileştirme Oranı(%)
b1-36	36	430	385	0,060	45	10,465
b2-76	76	830	487	0,329	343	41,325
b3-126	126	1330	669	1,613	661	49,699
b4-184	184	1910	1216	7,957	694	36,335
b5-250	250	2571	1537	16,822	1034	40,218
cc2-36	36	207	204	0,059	3	1,449
cc3-60	60	389	384	0,082	5	1,285
cc4-104	104	635	630	0,183	5	0,787
cc5-148	148	922	915	0,428	7	0,759
rd2-60	60	179	162	0,071	17	9,497
rd3-180	180	436	414	0,578	22	5,046
ch200	200	1440	1085	0,507	355	24,652778
Ortalama	121,7	939,9	674	2,391	265,91667	18,460

Çizelge 4.2’de ise YYKA2 için aynı değerler görülebilir. Ortalama çözüm süresi 0,879 ile YYKA1’e göre daha kısa olurken, ortalama iyileştirme %16,798 olmuştur. En yüksek iyileştirme oranı b5-250 probleminde %35.239 olarak gerçekleşmiştir.

Çizelge 4.2. Düzenli veri setleri için YYKA2 tur maliyetleri

Problem	Düğüm Sayısı	Klasik GSP Çözüm	YYKA2 Çözüm	YYKA2 Süre (s)	YYKA2 İyileştirme Miktarı	YYKA2 İyileştirme Oranı(%)
b1-36	36	430	399	0,065	31	7,209
b2-76	76	830	532	0,142	298	35,904
b3-126	126	1330	954	0,658	376	28,271
b4-184	184	1910	1287	2,228	623	32,618
b5-250	250	2571	1665	5,071	906	35,239
cc2-36	36	207	206	0,061	1	0,483
cc3-60	60	389	371	0,080	18	4,627
cc4-104	104	635	589	0,180	46	7,244
cc5-148	148	922	835	0,414	87	9,436
rd2-60	60	179	160	0,086	19	10,615
rd3-180	180	436	413	1,049	23	5,275
ch200	200	1440	1085	0,518	355	24,652778
Ortalama	121,7	939,9	708	0,879	231,91667	16,798

Çizelge 4.3, düzenli veri setleri için YYKA1 algoritmasından elde edilen ve çözüm için kullanılan süperdüğümler hakkında bilgi vermektedir. Kullanılan düğüm sayıları, beklendiği üzere azalmıştır. GSP'ye göre kullanılan düğüm sayısında ortalama %37.283 oranında bir azalma tespit edilmiştir.

Çizelge 4.3. Düzenli veri setleri için YYKA1 süperdüğüm sayıları

Problem	Düğüm Sayısı	YYKA1 Süperdüğüm Sayısı	YYKA1 Düğüm Sayısı Farkı	YYKA1 Düğüm Sayısı Farkı Oranı(%)	YYKA1 Çözüm Uzaı	YYKA1 En Küçük SZ Derecesi
b1-36	36	12	24	66,667	36	3
b2-76	76	38	38	50,000	716	2
b3-126	126	58	68	53,968	1754	2
b4-184	184	114	70	38,043	3160	1
b5-250	250	137	113	45,200	4632	1
cc2-36	36	34	2	5,556	60	1
cc3-60	60	58	2	3,333	96	1
cc4-104	104	102	2	1,923	176	1
cc5-148	148	144	4	2,703	252	1
rd2-60	60	30	30	50,000	108	2
rd3-180	180	90	90	50,000	468	2
ch200	200	40	160	80,000	40	5
Ortalama	121,7	71,417	50,25	37,283	958,17	1,83333

Düzenli veri setleri için YYKA2 ile elde edilen süperdüğüm sayıları ise Çizelge 4.4'te görülebilir. Burada düğüm sayısındaki azalma oranı, YYKA1'e göre düşmekle beraber, kullanılan en küçük SZ derecesinde artış gözlenmiştir.

Çizelge 4.4. Düzenli veri setleri için YYKA2 süperdüğüm sayıları

Problem	Düğüm Sayısı	YYKA2 Süperdüğüm Sayısı	YYKA2 Düğüm Sayısı Farkı	YYKA2 Düğüm Sayısı Farkı Oranı(%)	YYKA2 Çözüm Uzaı	YYKA2 En Küçük SZ Derecesi
b1-36	36	34	2	5,556	36	3
b2-76	76	72	4	5,263	130	4
b3-126	126	121	5	3,968	436	4
b4-184	184	175	9	4,891	798	4
b5-250	250	229	21	8,400	1216	4
cc2-36	36	35	1	2,778	60	1
cc3-60	60	57	3	5,000	96	1
cc4-104	104	100	4	3,846	176	1
cc5-148	148	140	8	5,405	252	1
rd2-60	60	58	2	3,333	108	2
rd3-180	180	177	3	1,667	468	2
ch200	200	40	160	80,000	40	5
Ortalama	121,7	103,17	18,5	10,842	318	2,66667

Düzensiz veri setlerine ilişkin çözüm süresi, tur maliyetleri ve iyileştirme oranları YYKA1 için Çizelge 4.5'te verilmiştir. Burada ortalama hesaplama süresinin 10.528 olduğu görülürken, ortalama iyileştirme oranı da %15.876 olarak hesaplanmıştır. En yüksek iyileştirme ise %18,861 ile bn8-50 probleminde gerçekleşmiştir.

Çizelge 4.5. Düzensiz veri setleri için YYKA1 tur maliyetleri

Problem	Düğüm Sayısı	Klasik GSP Çözüm	YYKA1 Çözüm	YYKA1 Süre (s)	YYKA1 İyileştirme Miktarı	YYKA1 İyileştirme Oranı(%)
bn1-50	50	540	483	13,394	57	10,556
bn2-50	50	543	453	5,441	90	16,575
bn3-50	50	571	486	4,633	85	14,886
bn4-50	50	574	495	7,414	79	13,763
bn5-50	50	574	489	6,542	85	14,808
bn6-50	50	536	439	10,092	97	18,097
bn7-50	50	568	468	1,372	100	17,606
bn8-50	50	562	456	12,561	106	18,861
bn9-50	50	573	469	0,984	104	18,150
bn10-50	50	540	469	17,155	71	13,148
bati67	67	616	504	36,219	112	18,182
Ortalama	51,55	563,4	473,73	10,528	89,636364	15,876

Düzensiz veri setlerin YYKA2 kullanılması ile elde edilen süre, maliyet ve iyileştirme sonuçları da Çizelge 4.6 üzerinde gösterilmiştir. Ortalama hesaplama süresi 10,503 olmuştur. İyileştirme oranı ortalama olarak 25,376 olurken, en yüksek iyileştirme oranı %32,218 ile bn7-50 probleminde hesaplanmıştır.

Çizelge 4.6. Düzensiz veri setleri için YYKA1 tur maliyetleri

Problem	Düğüm Sayısı	Klasik GSP Çözüm	YYKA2 Çözüm	YYKA2 Süre (s)	YYKA2 İyileştirme Miktarı	YYKA2 İyileştirme Oranı(%)
bn1-50	50	540	413	13,956	127	23,519
bn2-50	50	543	376	2,763	167	30,755
bn3-50	50	571	461	4,920	110	19,264
bn4-50	50	574	458	7,900	116	20,209
bn5-50	50	574	428	6,902	146	25,436
bn6-50	50	536	382	10,589	154	28,731
bn7-50	50	568	385	1,035	183	32,218
bn8-50	50	562	428	13,034	134	23,843
bn9-50	50	573	432	1,089	141	24,607
bn10-50	50	540	409	11,656	131	24,259
bati67	67	616	454	41,691	162	26,299
Ortalama	51,55	563,4	420,55	10,503	142,81818	25,376

Çizelge 4.7, YYKA1 ile elde edilen düğümler hakkında bilgi vermektedir. Ziyaret edilen düğüm sayısının ortalama olarak %34,076 oranında azaldığı görülmektedir. En çok azalma ise %42 oranı ile bn4-50 probleminde gerçekleşmiştir.

Çizelge 4.7. Düzensiz veri setleri için YYKA1 düğüm sayıları

Problem	Düğüm Sayısı	YYKA1 Süperdüğüm Sayısı	YYKA1 Düğüm Sayısı Farkı	YYKA1 Düğüm Sayısı Farkı Oranı(%)	YYKA1 Çözüm Uzağı	YYKA1 En Küçük SZ Derecesi
bn1-50	50	34	16	32,000	6817	1
bn2-50	50	36	14	28,000	2695	1
bn3-50	50	32	18	36,000	3588	1
bn4-50	50	29	21	42,000	4928	1
bn5-50	50	33	17	34,000	3825	1
bn6-50	50	30	20	40,000	5495	1
bn7-50	50	36	14	28,000	1462	1
bn8-50	50	31	19	38,000	6546	1
bn9-50	50	34	16	32,000	1405	1
bn10-50	50	34	16	32,000	5707	1
bati67	67	45	22	32,836	11588	1
Ortalama	51,55	34	17,545	34,076	4914,2	1

YYKA2 için düğüm sayılarını gösteren Çizelge 4.8’de, ortalama düğüm sayısı azaltılması %5,908 oranında olmuştur. Buna karşın kullanılan en küçük SZ derecesi, 3. dereceye kadar çıkmıştır. En yüksek düğüm sayısında azalma oranı, %12 ile bn2-50 probleminde hesaplanmıştır.

Çizelge 4.8. Düzensiz veri setleri için YYKA2 düğüm sayıları

Problem	Düğüm Sayısı	YYKA2 Süperdüğüm Sayısı	YYKA2 Düğüm Sayısı Farkı	YYKA2 Düğüm Sayısı Farkı Oranı(%)	YYKA2 Çözüm Uzaı	YYKA2 En Küçük SZ Derecesi
bn1-50	50	47	3	6,000	6817	1
bn2-50	50	44	6	12,000	2413	3
bn3-50	50	47	3	6,000	3588	1
bn4-50	50	47	3	6,000	4928	1
bn5-50	50	49	1	2,000	3825	1
bn6-50	50	46	4	8,000	5495	1
bn7-50	50	48	2	4,000	1412	2
bn8-50	50	45	5	10,000	6546	1
bn9-50	50	49	1	2,000	1405	1
bn10-50	50	47	3	6,000	5657	2
bati67	67	65	2	2,985	11588	1
Ortalama	51,55	48,545	3	5,908	4879,5	1,36364

Çizelge 4.9 ise bütün hesaplamalar için özet niteliğindedir. Her iki YYKA varyasyonu için hesaplama süreleri, iyileştirme oranları, ziyaret edilen düğümlerin azaltılma oranı ve kullanılan en küçük SZ derecesi karşılaştırılmıştır.

Çizelge 4.9. YYKA için sonuçların karşılaştırılması

Problem	Şehir Sayısı	Klasik GSP Çözüm	YYKA1 Süre (sn)	YYKA2 Süre (sn)	YYKA1 İyileştirme Oranı (%)	YYKA2 İyileştirme Oranı (%)	YYKA1 Nokta Sayısı Farkı Oranı (%)	YYKA2 Nokta Sayısı Farkı Oranı (%)	YYKA1 En Küçük SZ Derecesi	YYKA2 En Küçük SZ Derecesi
DÜZENLİ										
b1-36	36	430	0,060	0,065	10,465	7,209	66,667	5,556	3	3
b2-76	76	830	0,329	0,142	41,325	35,904	50,000	5,263	2	4
b3-126	126	1330	1,613	0,658	49,699	28,271	53,968	3,968	2	4
b4-184	184	1910	7,957	2,228	36,335	32,618	38,043	4,891	1	4
b5-250	250	2571	16,822	5,071	40,218	35,239	45,200	8,400	1	4
cc2-36	36	207	0,059	0,061	1,449	0,483	5,556	2,778	1	1
cc3-60	60	389	0,082	0,080	1,285	4,627	3,333	5,000	1	1
cc4-104	104	635	0,183	0,180	0,787	7,244	1,923	3,846	1	1
cc5-148	148	922	0,428	0,414	0,759	9,436	2,703	5,405	1	1
rd2-60	60	179	0,071	0,086	9,497	10,615	50,000	3,333	2	2
rd3-180	180	436	0,578	1,049	5,046	5,275	50,000	1,667	2	2
ch200	200	1440	0,507	0,518	24,652778	24,652778	80,000	80,000	5	5
DÜZENSİZ										
bn1-50	50	540	13,394	13,956	10,556	23,519	32,000	6,000	1	1
bn2-50	50	543	5,441	2,763	16,575	30,755	28,000	12,000	1	3
bn3-50	50	571	4,633	4,920	14,886	19,264	36,000	6,000	1	1
bn4-50	50	574	7,414	7,900	13,763	20,209	42,000	6,000	1	1
bn5-50	50	574	6,542	6,902	14,808	25,436	34,000	2,000	1	1
bn6-50	50	536	10,092	10,589	18,097	28,731	40,000	8,000	1	1
bn7-50	50	568	1,372	1,035	17,606	32,218	28,000	4,000	1	2
bn8-50	50	562	12,561	13,034	18,861	23,843	38,000	10,000	1	1
bn9-50	50	573	0,984	1,089	18,150	24,607	32,000	2,000	1	1
bn10-50	50	540	17,155	11,656	13,148	24,259	32,000	6,000	1	2
batı67	67	616	36,219	41,691	18,182	26,299	32,836	2,985	1	1
Ortalama	88,130	759,826	6,282	5,482	17,224	20,901	35,749	8,482	1,435	2,043

Bütün veri setlerinde, her iki YYKA varyasyonu için de, çözüm değerinde klasik GSP çözümüne göre önemli iyileştirmeler gerçekleşmiştir. En yüksek iyileştirme yüzdesi, %49 seviyesindedir. Buna karşın, örnek veri setleri için çözüm süreleri, 0,059 saniye ile 17 saniye arasında değişmektedir. Çözüm süresinin, genel olarak düzenli veri setlerinde daha kısa olduğu görülmüştür. Çözüm sürelerinin, düğüm sayısı ile beraber arttığı bir eğilim görülmekte olsa da, sürelerin ve iyileştirme oranlarının veri setlerinin ağ yapısının geometrik özelliklerine bağlı olarak değiştiği düşünülmektedir. Ancak YYKA2 varyasyonunun hem süre, hem iyileştirme oranı olarak YYKA1'e göre genelde daha iyi bir performans sergilediği görülmektedir. Bu durumun gevşetilmiş komşu SZ seçim koşullarına bağlı olduğu değerlendirilmektedir. Ayrıca ziyaret edilen SZ sayısının YYKA2'de YYKA1'e göre daha fazla olmasına rağmen, iyileştirme miktarının YYKA2'de daha iyi olması, yine gevşetilmiş komşu SZ seçim koşulları ile açıklanabilir.

Dikkat edilmesi gereken başka bir nokta, çözümlerde kullanılan en küçük SZ derecesidir. Bu derece, sezgisel algoritma tarafından, çözüm sırasında problem boyutuna göre en yüksek sayı seçilecek şekilde belirlenmektedir. Amaç, bu sayede en az düğümün ziyaret edilmesi, bağlantılı olarak da mümkün olan en kısa tur uzunluklarının elde edilmesini sağlamaktır. Ancak temelde kullanılan sezgisel olan EYK sezgiseli çözüm kurucu bir sezgisel olarak çalışmaktadır. Bu nedenle elde edilen iyileştirme oranları ve koşma süreleri, en küçük SZ derecesi ile orantılı olmamaktadır. Daha önce belirtildiği üzere, ağ yapısının geometrik özellikleri de çözüm süresi ve iyileştirme oranları etkilemektedir.

Gerçek hayat uygulaması incelendiğinde, GSP'ye göre, YYKA1 ile %18, YYKA2 ile %26 oranında tur maliyetinden tasarruf sağladığı görülmüştür. Özellikle YYKA1 çözümü incelendiğinde, ziyaret edilen düğüm sayısının %32 oranında azalmasına rağmen, daha yüksek ziyaret edilen düğüm sayısına sahip YYKA2 çözümüne göre tur maliyetinden daha az tasarruf edildiği görülmektedir. Bu durum, örnek veri setleri ile yapılan koşturmaların sonuçları ile aynı doğrultudadır. Ayrıca gerçek hayat uygulamasının çalıştırma süresi, diğer problemler gibi saniyelerle ifade edilecek şekilde, 36 ve 42 saniye olarak ölçülmüştür.

5. SONUÇ VE ÖNERİLER

Bu bölümde yapılan çalışma özetlenip, ileriki çalışmalar için fikirler ve sonuç verilmiştir. Yapılan çalışmada incelenen YY-GSP, müşterilerin komşuluk alanları bulundurması ile GSP'den ayrılmaktadır. Bu komşuluk alanları kullanılarak, GSP'ye göre tur maliyetinde önemli iyileştirmeler sağlanmaktadır. Literatür incelendiğinde, YY-GSP konusundaki çalışmaların ve bu konuya olan ilginin arttığı görülmektedir.

Bu çalışmada, YY-GSP, sayaç okuma süreçleri için uyarlanmıştır. YY-GSP, sezgisel bir yöntem ile çözüme ulaştırılarak YY-GSP'nin, klasik GSP'ye göre tur zamanında sağlayacağı iyileştirme oranını değerlendirilmiştir. Çalışmada çeşitli büyüklüğe ve geometrik yapıya sahip birçok veri seti ile elde edilen sonuçlarda, YY-GSP'nin, geleneksel sayaç okuma süreçlerinde, tur maliyetlerini büyük oranda iyileştirecek çözümler ürettiği görülmüştür. Bu çözümler ağ yapısının geometrik özelliklerine bağlı olarak, tur maliyetlerinde %0,5 ile %49 oranları arasında iyileştirme getirmektedir. Gerçek hayat uygulamasında ise farklı yerleşim tipleri beraber incelenmiş, tur maliyetinde %26 oranında iyileştirme yapıldığı tespit edilmiştir.

Sonuçlar incelendiğinde, YY-GSP'nin iyileştirme potansiyelinin yüksek olduğu görülmektedir. Örnek problemlerde hesaplanan yüksek iyileştirme oranları, bu durumu açıkça göstermiştir. Geliştirilen sezgisel yöntemde, gereken hesaplama zamanının en fazla 42 saniye olduğu bulunmuştur. Kısa sürede yüksek iyileştirme oranları bulunmuştur. Bulunan tur maliyetlerinin karşılaştırılması, YY-GSP kullanımının tur maliyeti açısından avantajlı olduğunu açıkça ortaya koymuştur.

Gelecekteki çalışmalarda, bu çalışma kapsamında geliştirilmiş YYKA1 ve YYKA2 sezgisellerinden farklı sezgiseller kullanılıp, bu sezgisellerin YY-GSP çözümlerinde gösterdikleri performanslar değerlendirilebilir. Tavlama benzetimi, Tabu arama gibi metasezgisellerin ya da genetik algoritma, karınca kolonisi optimizasyonu gibi popülasyon tabanlı metasezgisellerin kullanımının, iyileştirme oranlarına olan etkisi araştırılabilir. Sezgisellerin kullanımının, alt-üst sınır çalışmalarında da katkı sağlayabileceği düşünülmektedir.

Ayrıca, problem için bir sonraki aşamada, daha fazla ve çeşitli gerçek veri setleri üzerinde çalışılabilir. Yerleşim yerlerindeki sayaçların kesin konumlarından yola çıkılarak YY-GSP'nin kullanımının getireceği iyileştirme potansiyeli ölçülebilir. Koronavirüs pandemisi ile hayatımıza giren sosyal mesafe kuralları, insanlar arasındaki temasın azaltılmasını gerektirmektedir. Bu şartlar altında, çalışmada ele alınan sayaç okuma örneğinde olduğu gibi, fiziksel temas gerektiren işlerin mümkün olduğunca uzaktan yapılması gerekliliği doğmuştur. Bu bağlamda YY-GSP ve benzer problemlerin çözümleri, önümüzdeki dönemde daha önemli hale gelecektir.

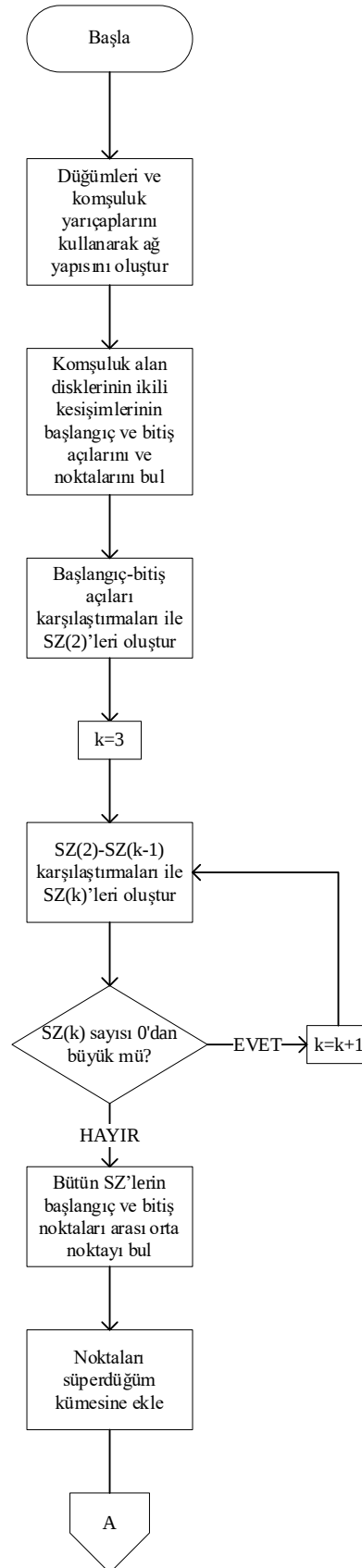
KAYNAKLAR

1. Behdani, B. and Smith, J. C. (2014). An integer-programming-based approach to the close-enough traveling salesman problem. *INFORMS Journal on Computing*, 415-643.
2. Bercea, I. O. (2018). *Improved bounds for the traveling salesman problem with neighborhoods on uniform disks*. 30th Canadian Conference on Computational Geometry.
3. Mennell, W. K. (2009). *Heuristics for solving three routing problems: close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem*, Doctoral dissertation. University of Maryland, USA.
4. Bandeira, T. W., Brito, A. V., Subramanian, A. and Coutinho, W. P. (2015). *Analysis of path planning algorithms based on travelling salesman problem embedded in UAVs*. 2015 Brazilian Symposium on Computing Systems Engineering.
5. Gulczynski, D. J., Heath, J. W. and Price, C. C. (2006). The close enough traveling salesman problem: a discussion of several heuristics. In *Perspectives in Operations Research*.
6. Ha, M. H., Bostel, N., Langevin, A., & Rousseau, L.-M. (2012). *An exact algorithm for the close enough traveling salesman problem with arc covering constraints*. Proceedings of the 1st International Conference on Operations Research and Enterprise Systems.
7. Dumitrescu, A. and Mitchell, J. S. (2003). Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms*, 135-159.
8. Coutinho, W. P. and Subramanian, A. (2016). A branch-and-bound algorithm for the close-enough traveling salesman problem. *INFORMS Journal on Computing*.
9. Carrabs, F., Cerrone, C., Cerulli, R., & Gaudioso, M. (2017). A novel discretization scheme for the close enough traveling salesman problem. *Computers & Operations Research*, 163-171.
10. Carrabs, F., Cerrone, C., Cerulli, R., & D'Ambrosio, C. (2017). *Improved upper and lower bounds for the close enough traveling salesman problem*. International Conference on Green, Pervasive, and Cloud Computing.
11. Wang, X., Golden, B. and Wasil, E. (2018). A steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem. *Computers and Operations Research*, 1-20.

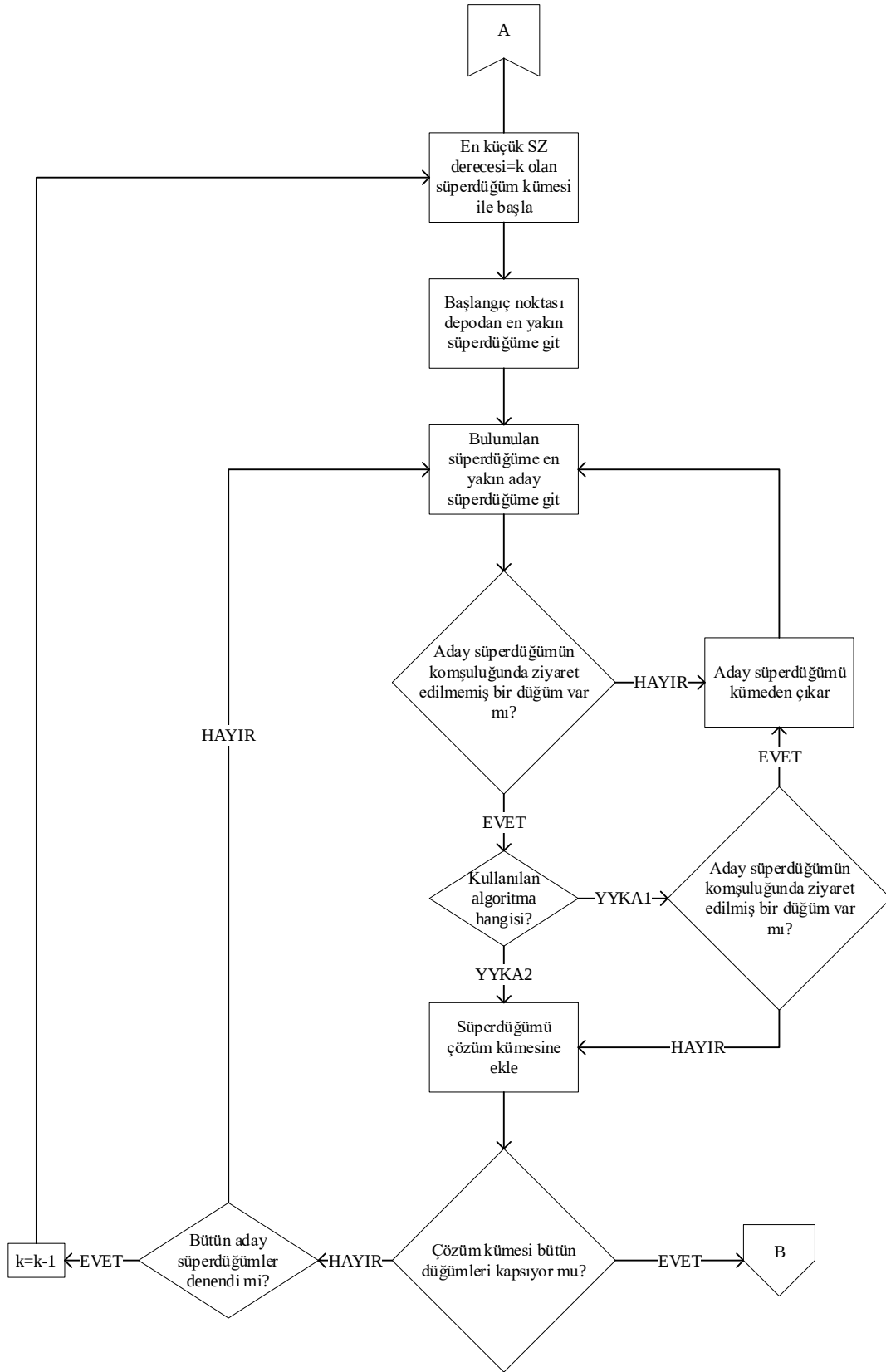
12. Semami, S., Touluni, H. and ElByed, A. (2019). The close enough traveling salesman problem with time window. *International Journal of Circuits, Systems and Signal Processing*, 579-584.
13. Carrabs, F., Cerrone, C., Cerulli, R. and Golden, B. (2020). An adaptive heuristic approach to compute upper and lower bounds for the close-enough traveling salesman problem. *INFORMS Journal on Computing*, 855-1186.
14. Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A guide to the theory of np-completeness*. USA: W. H. Freeman & Co.
15. Şentürk, İ. F. (2020). A steiner zone approach for mobile data collection in partitioned wireless sensor networks. *Bilişim Teknolojileri Dergisi*, 13(3), 217-224.
16. Hà, M. H., Bostel, N., Langevin, A. and Rousseau, L.-M. (2014). Solving the close-enough arc routing problem. *Networks*, 63(1), 107-118.
17. Dumitrescu, A. and Tóth, C. D. (2013). *The traveling salesman problem for lines, balls and planes*. SODA '13: Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms.
18. Gentilini, I., Margot, F. and Shimada, K. (2013). The travelling salesman problem with neighbourhoods: MINLP solution. *Optimization Methods and Software*, 28(2), 364-378.
19. Corberán, Á., Plana, I., Reula, M. and Sanchis, J. M. (2020). On the distance-constrained close enough arc routing problem. *European Journal of Operational Research*.
20. Faigl, J. (2018). GSOA: Growing self-organizing array - unsupervised learning for the close-enough traveling salesman problem and other routing problems. *Neurocomputing*, 120-134.

EKLER

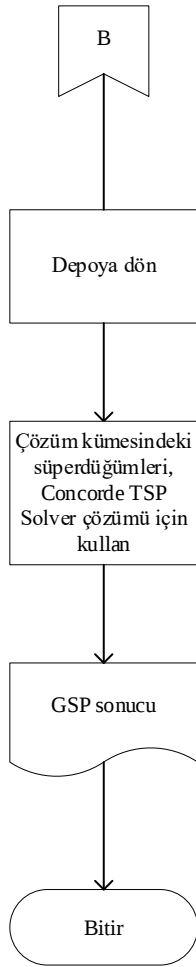
EK-1. YYKA algoritması akış şeması



EK-1. (devam) YYKA algoritması akış şeması



EK-1. (devam) YYKA algoritması akış şeması





GAZİ GELECEKTİR...