

Araştırma Makalesi/Research Article

PRÜFER-KARAGÜL ALGORİTMASI: GEZGİN SATICI PROBLEMİ İÇİN YENİ BİR YAKLAŞIM

*PRÜFER-KARAGÜL ALGORITHM:
A NOVEL APPROACH FOR TRAVELLING SALESMAN PROBLEM*

Kenan KARAGÜL*

Öz

Kombinatoriyal optimizasyon alanında temel bir model olduğu için literatürde oldukça yaygın çalışılan gezgin satıcı probleminin etkin ve hızlı çözümü için yeni sezgisel yöntemler geliştirilmesine devam edilmektedir. Bu çalışmada, gezgin satıcı problemi için Prüfer-Karagül adı verilen yeni bir yapısal çözüm yaklaşımı önerilmiştir. Önerilen yöntemin performansını değerlendirmek için literatürde yaygın olarak kullanılan gezgin satıcı test problemleri ile analizler yapılmıştır. Yapılan testler sonucunda elde edilen en iyi çözümler optimal çözümden %2, ortalama çözüm değerleri ise %2,50 sapma göstermiştir. Sonuç olarak, önerilen yöntem çözüm performansı ve hızı açısından başarılı çözümler üretmektedir.

Anahtar Kelimeler: Prüfer-Karagül Algoritması, Gezgin Satıcı Problemi, En Yakın Komşu Sezgiseli, 2-Opt Algoritması, Prüfer Kod

Abstract

As it is a fundamental model in the field of combinatorial optimization, new heuristic methods are developed for effective and rapid solution of the travelling salesman problem, which is widely used in the literature. In this study, a new constructive approach called Prüfer-Karagül has been proposed for the traveling salesman problem. In order to evaluate the performance of the proposed method, analysis was made with travelling salesman problem test instances which are commonly used in the literature. The best solutions obtained as a result of the tests showed 2% deviation from the optimal solution and 2.50% deviation from the average solution values. As a result, the proposed method produces successful solutions in terms of solution performance and speed.

Keywords: Prüfer-Karagül Algorithm, Traveling Salesman Problem, Nearest Neighbor Heuristic, 2-Opt Algorithm, Prüfer Code

*  Dr. Öğr. Üyesi, Pamukkale Üniversitesi, Lojistik Bölümü, kkaragul@pau.edu.tr

EXTENDED SUMMARY

Research Problem

In this study, a new constructive approach, named Prüfer-Karagül, has been proposed to obtain reasonable and effective solutions for the Travelling Salesman Problem (TSP).

Literature Review

In the literature, there are limited number of studies on the application of the Prüfer code on the TSP. Hajiaghahi-Keshteli (2011) and Anbuudayasankar et al. (2014) used Prüfer coding for gene design and gene structure. He et al. (2015) solved the TSP with genetic algorithm based on Prüfer coding.

Methodology

In this study, it is aimed to solve TSP by using the Prüfer code structure with the nearest neighbors heuristics and 2-Opt approach. The details of the solution method and comparisons with the traditional methods are demonstrated on a small TSP given in Table 1 and Figure 1. The flowchart of the Prüfer-Karagül algorithm is given in Appendix 1.

Table 1: Proposed Approach: Solving KTSP1 Problem with Prüfer-Karagül Algorithm

Production No	Prüfer Code	Prüfer Tree	Prüfer-Karagül TSP	Cost	Time (sec)
1	[1,5,5,5,8,4,8,5,1]	[2,3,6,7,9,10,4,8,5,1]	[4,2,3,10,1,5,7,6,9,8]	27,01	0,003024
2	[2,8,5,2,4,7,2,8,3]	[1,6,9,5,10,4,7,2,8,3]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,00063
3	[10,3,8,2,3,1,6,7,6]	[4,5,9,8,2,3,1,10,7,6]	[4,2,3,10,1,5,7,6,9,8]	27,01	0,000933
4	[5,7,7,7,7,10,3,8,3]	[1,2,4,5,6,7,9,10,8,3]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000783
5	[2,7,5,5,7,8,4,7,5]	[1,2,3,6,9,10,8,4,7,5]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,002679
6	[9,9,3,7,6,6,9,3,4]	[1,2,5,8,7,10,6,9,3,4]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000813
7	[2,10,7,5,7,6,7,6,8]	[1,2,3,4,5,9,10,7,6,8]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000665
8	[6,10,3,2,2,1,5,5,4]	[7,6,8,3,9,2,1,10,5,4]	[7,6,9,8,4,2,3,10,1,5]	27,01	0,00094
9	[8,7,8,10,10,2,2,7,1]	[3,4,5,6,8,9,10,2,7,1]	[7,6,9,8,4,2,3,10,1,5]	27,01	0,0007
10	[6,6,9,5,4,7,8,6,4]	[1,2,3,9,5,10,7,8,6,4]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000671

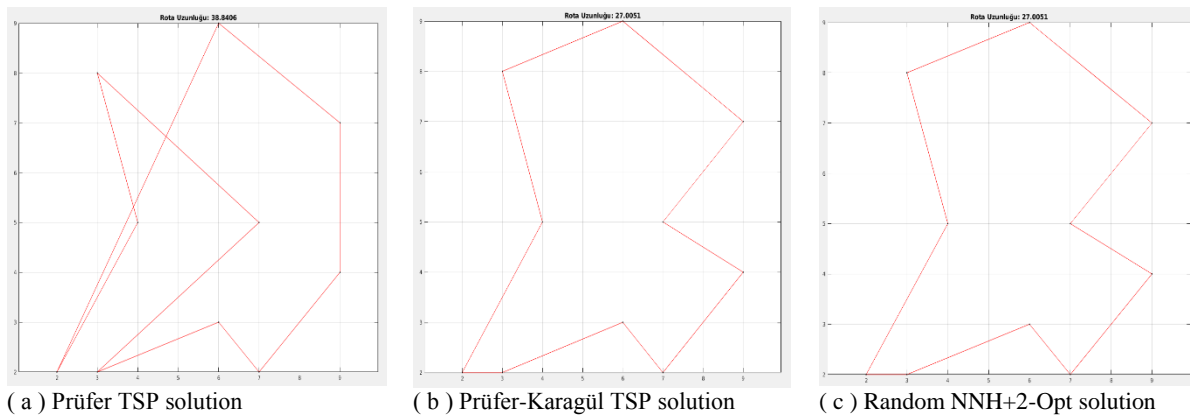


Figure 1: Prüfer, Prüfer-Karagül and NNH+2-Opt solutions for KTSP1 problem

Results and Conclusions

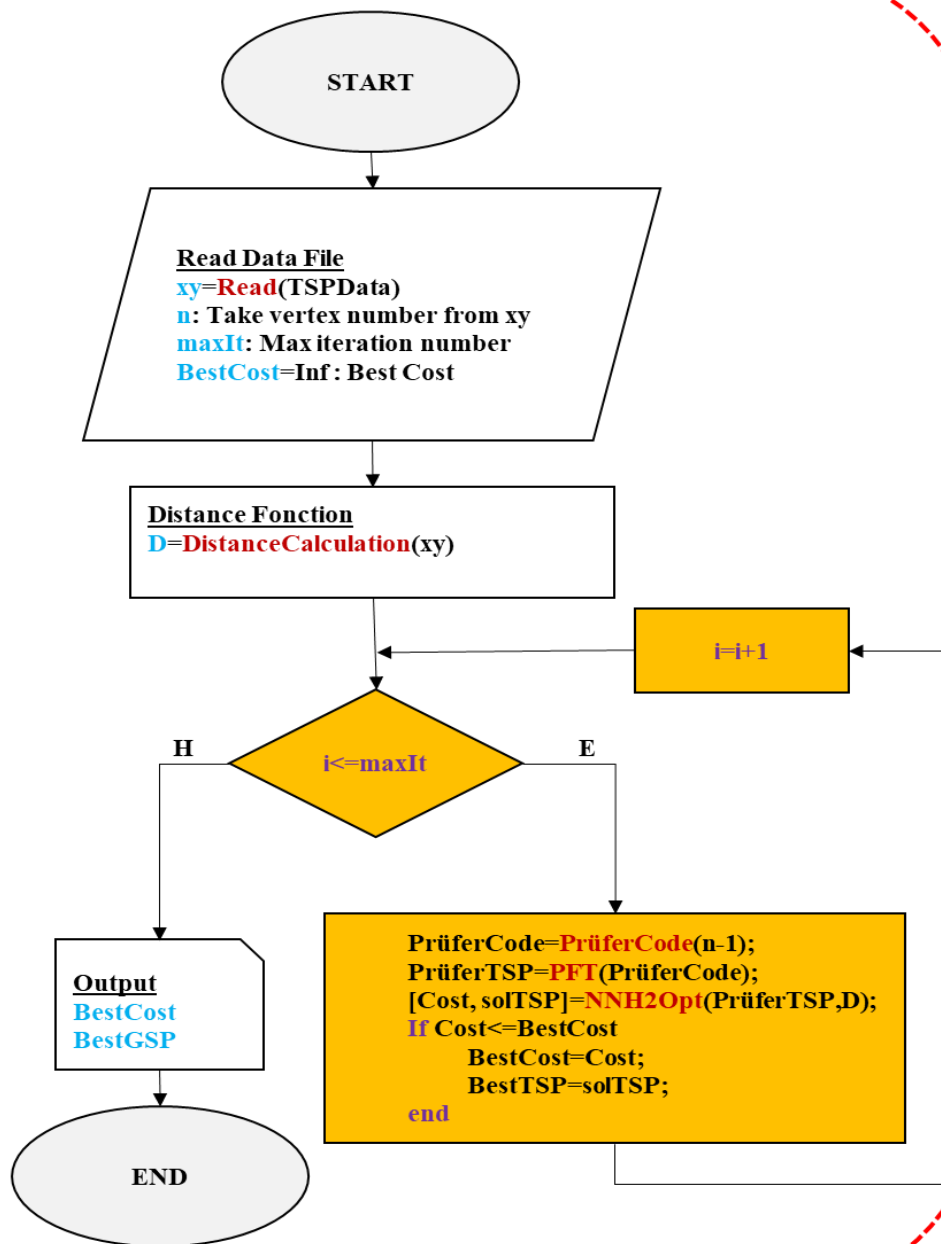
In this study, the comparisons are conducted on larger test instances. The best solutions obtained for the TSP test instances showed 2% deviation from the optimal solution and 2.50% deviation from the average solution values. Consequently, the proposed method produces successful solutions in terms of solution performance and speed.

APPENDIX 1: Prüfer-Karagül Algorithm Flowchart

Prüfer-Karagül Algorithm

Functions in Algorithms

- 0- TSP Reading Function,
xy=**Read**(TSPData);
 - 1- Distance Calculation Function,
D=DistanceCalculation(xy)
 - 2- Prüfer Code Production Function **PrüferCode**,
PrüferCode=PrüferCode(n-1)
 - 3- Prüfer Transformation Function **PFT**, Prüfer Code transforms to Prüfer Tree (PrüferTSP)
PrüferTSP=PFT(PrüferCode)
 - 4- TSP Solution Function,
[Cost, solTSP]=NNH2Opt(PrüferTSP)
- NNH** (Nearest Neighbor Heuristic) and **2-Opt** (2-Optimal) hybrid solver. solTSP: Function produced a TSP solution and Cost as a Hamilton tour distance.



GİRİŞ

Gezgin Satıcı Problemi (GSP), bir satıcının bulunduğu şehirden başlayıp, her şehre sadece bir kez uğradıktan sonra başladığı şehre geri dönerken izleyeceği en kısa rotanın belirlendiği optimizasyon problemidir. GSP yöneylem araştırması bilimsel literatüründe önemli bir yere sahiptir. Özellikle farklı alanlarda endüstriyel uygulama alanı olması ve başka optimizasyon problemlerinin tanımlanmasında temel yapı taşı olması nedeniyle de kök araştırma problemidir. Bu nedenle bilimsel literatürde üzerinde yoğun çalışılan problemlerden biri olma özelliğini korumaktadır.

GSP’de temel olarak problemde yer alan şehirlerarası mesafelerin bilindiği kabul edilir. Teorik açıdan bakıldığında GSP, şehirlerin noktalarla, şehirlerarası yolların kenarlarla temsil edildiği bir çizge üzerinde, en kısa Hamilton turunun bulunması problemidir (Karagül, 2014: 775). GSP, anlaşılması kolay ancak çözümü oldukça zor bir problemidir. NP-zor sınıfında yer alan bu problemde nokta sayısı arttıkça kesin matematiksel yöntemlerle çözüm zorlaşmaktadır. Bu nedenle her geçen gün genişleyen GSP bilimsel literatüründe, kesin matematiksel yöntemlerle çözüm yaklaşımlarından daha çok yapısal ve sezgisel algoritmalar üzerinde durulmaktadır (Kounalakis ve Kapelonis, 2002; Süral, 2003:s.1; Aksaraylı ve Pala, 2018).

GSP’de uzaklık matrisinin yapısına bağlı olarak simetrik ve asimetrik GSP olmak üzere ikiye ayrılır. Ayrıca satıcı sayısına göre tek veya çok satıcılı GSP tanımlanabilir. Bu çalışmada Simetrik ve tek satıcılı GSP’nin çözümü için Prüfer kod esaslı yeni bir çözüm yaklaşımı önerilmektedir. Çalışmanın birinci bölümünde GSP ile ilgili bilimsel literatür kısaca sunulmuştur. İkinci bölümde yöntemin ayrıntıları, üçüncü bölümde uygulama ve analizler, son bölümde ise sonuçlar yer almaktadır.

1. BİLİMSEL YAZIN

Mühendislik, işletmecilik ve endüstriyel süreçlerin başlıca problemi etkinliğin artırılmasıdır. Etkinliği arttırmanın en pratik yolu ise karşılaşılan problemlerin çözümünde sezgisel ve meta-sezgisel yöntemlerin kullanılmasıdır. Ruiz-Vanoye vd. (2012) doğadan esinlenen sezgisel algoritmalar ile ilgili bir çalışma yapmıştır. GSP’nin çözümü için guguk kuşu algoritması (Jati vd., 2012; Ouaarab vd., 2013; Ouyang vd., 2013; Karagül, 2014), yarasa algoritması (Yang, 2010), ateş böceği algoritması (Yang, 2010; Sureja, 2012; Bhushan ve Pillai, 2013; Fister vd., 2013), arı kolonisi algoritması (Yang, 2010), kanguru algoritması (Pollard, 1978; Romsy, 2011; Erdem ve Kesintürk, 2011), parçacık sürüsü optimizasyon algoritması (Dorigo ve Gambardella, 1997; Htun, 2018;) ve karınca kolonisi optimizasyon algoritması (Mavrovouniotis ve Yang, 2013; Aksaraylı ve Pala, 2018; Htun, 2018;) kullanılması ile ilgili çalışmalar bilimsel literatürde yer almaktadır. Doğadaki hayvanların davranışlarından esinlenilerek geliştirilen bu yöntemlerin yanı sıra genetik algoritma (Zhao vd., 2009; Htun, 2018;), tabu arama algoritması (Glover, 1990; Glover ve Laguna, 1993; Glover ve Laguna, 1997; Gendreau vd., 1998; Gendreau, 2002; Basu, 2012), benzetimli tavlama algoritması (Kirkpatrick vd., 1983; Malek vd., 1989; Özdağoğlu, 2008; Wang vd., 2009; Zhou vd., 2019), harmoni arama algoritması (Geem vd., 2001; Yang, 2009; Yun vd., 2013; Karagül vd., 2016; Boryczka ve Szwarc, 2019) ve akışkan genetik algoritma (Jafari-Marandi ve Smith, 2017; Şahin ve Karagül, 2019) gibi meta-sezgisellerin kullanıldığı çalışmalar da bulunmaktadır.

Bilimsel literatür incelendiğinde bu çalışmada ele alınan Prüfer kod ve gezgin satıcı problemine uygulanması ile ilgili sınırlı sayıda çalışma yer almaktadır. Thompson vd. (2007) Prüfer sıralamasının/Prüfer kodu genetik algoritma kodlamasındaki zayıflığını eleştirerek karahindiba (Dandelion) kodlamasını önermişlerdir. Wang vd. (2009) tarafından Prüfer sıralamasını polinom zamanda yapabilecek etkinlikte bir algoritma önerilmiştir. Paulden ve Smith (2007) tarafından Prüfer sıralaması için bir algoritma önerilmiştir. Hajiaghahi-Keshteli (2011) iki aşamalı tedarik zinciri optimizasyon araştırmasında genetik algoritma ve yapay bağışıklık sistemi algoritmaları ile çözümler önerilmiştir. Genetik algoritma gen tasarımında Prüfer kodlaması ile depolar ile müşteriler arasındaki ilişkiler tanımlanmıştır. Anbuodayasankar vd. (2014) işyükü dengelenmiş GSP için genetik algoritma gen yapısında gezgin satıcıları göstermek için Prüfer kodlamasından faydalanmışlardır. He vd. (2015) Prüfer kodlamaya dayanan bir genetik algoritma geliştirmişler ve bu genetik algoritma kodlamasını kullanarak tek araçlı bir lojistik dağıtım problemini çözmüşlerdir. Bu çalışmada Prüfer kod yapısı en yakın komşu ve 2-Opt sezgiselleri ile birlikte kullanılarak GSP’ye çözüm aranmaktadır. Takip eden bölümde önerilen yöntemin ayrıntıları sunulmaktadır.

2. ÇÖZÜM YÖNTEMİ

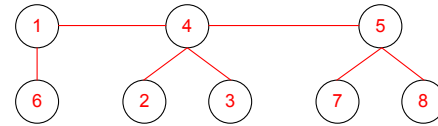
Prüfer kodu, Heinz Prüfer tarafından 1918 yılında önerilmiştir. 1918 yılında Prüfer n düğümle etiketlenmiş bir ağacın $(n-2)$ düğüm etiketi ile ilişkisini göstermiştir. Bu ispat bir ağacın kodlanmasında, ağacın en küçük etiketli yapraklarının ardışık olarak silinmesi ve komşusuna kaydedilmesi olarak ifade edilmektedir. (Deo ve Micikevicius:s.1). Önerilen çözüm yaklaşımı Prüfer kodun ağaç çözümlemesinin bir GSP çözümü olarak kabul edilmesi varsayımından hareketle geliştirilmiştir. Bu bağlamda öncelikle Prüfer kod ve ağaç yapısının elde edilmesinden bahsedilecek ve arkasından GSP çözüm mekanizması açıklanacaktır.

2.1. Prüfer Kod

Kombinatorik matematikte, etiketlenmiş bir ağacın Prüfer sırası, Prüfer kodu veya Prüfer sayıları ağaçla ilişkili biricik sıralamadır. Bir ağaç için sıralama n nokta (vertex) için $(n - 2)$ uzunluktadır. Aynı zamanda bu sıralama basit iteratif bir algoritma ile üretilebilir. Prüfer sıralaması 1918 yılında Heinz Prüfer tarafından Cayley formülünün bir ispatı olarak önerilmiş ve kullanılmıştır (He vd., 2015; Web-1, 2018; Web-2, 2018; Web-3,2018). Bir T ağacına ait Prüfer kodu [4, 4, 1, 4, 5, 5] olarak alındığında bu kodun uzunluğu altıdır. Bu durumda aranan Prüfer ağacı sekiz düğümlüdür. Prüfer algoritmasının adımları uygulandığında elde edilen ağaç [2, 3, 6, 1, 4, 7, 5] olarak elde edilecektir. Bu çözümün Prüfer kodu ve ağacı Şekil 1’de, ağacın çizgesi ise Şekil 2’de gösterilmiştir. Her ne kadar Prüfer kodu sekiz düğümlü bir problem için altı düğümle ifade edilse de çözümleme aşamasına geçildiğinde yedi düğümle ifade edilir. Buradan hareketle ağaçta yer alan tüm düğümler üretilir. Önerilen çözüm yaklaşımında da Prüfer kod $(n - 1)$ ile üretilcek ve buradan n düğümlü ağaç elde edilecektir.

Prüfer Sıralaması	4	4	1	4	5	5
Ağaç Dizilişi	2	3	6	1	4	7

Şekil 1: Prüfer Kodu ve Ağaç Gösterimi



Şekil 2: Ağaç çizgesi

Kaynak: Web-3, 2018

Prüfer kodun kısa açıklamasından sonra, En Yakın Komşu (NNH) Algoritması, 2-Opt Algoritması kısaca açıklanacak ve Prüfer kodun GSP çözümlemesi için nasıl kullanıldığı ortaya konulacaktır.

2.2. En Yakın Komşu Algoritması

Basitliği nedeniyle, Gezgin Satıcı Problemini çözmek için akla gelen ilk sezgiseldir. En Yakın Komşu sezgiselidir (NNH). Bu sezgisel algoritma gezgin satıcının rastgele bir şehirden tura başlaması ve bir sonraki şehri en yakın mesafedeki şehir seçerek tüm şehirleri ziyaret etmesi ve tekrar başlangıç şehrine dönmesi olarak tanımlanabilir (Karkory ve Abudalmola, 2013; Web-6, 2019).

NNH algoritmasının adımları aşağıdaki gibidir (Kızılateş ve Nuriyeva, 2013):

1. Çizge de rastgele bir düğüm seçiniz.
2. Bu düğümden sonra en yakın ziyaret edilmemiş düğümü seçiniz.
3. Bu iki düğüme en yakın ziyaret edilmemiş düğümü ziyaret ediniz ve son düğümü güncelleyiniz.
4. Hala ziyaret edilmemiş düğüm kalmış mıdır? Eğer varsa, Adım 3'e gidiniz.
5. Son düğümden ilk düğüme gidiniz.

2.3. 2-Opt Algoritması

2-Opt algoritması ilk olarak 1958 yılında Croes tarafından önerilmiştir (Croes, 1958). GSP çözümlerinde tur geliştirici bir algoritma olarak en çok kullanılan algoritmalarından birisidir. Temel fikir var olan uygun bir GSP çözümü için farklı olarak seçilen 2'li kenar çiftlerinin sıralı olarak var olan çözümü geliştirip geliştirmediği kontrol edilir (Karkory ve Abudalmola, 2013; Şahin ve Karagül, 2019). Bu algoritma için Kesintürk vd. tarafından detaylı bir açıklama ve matlab kodu verilmiştir (Kesintürk vd., 2016).

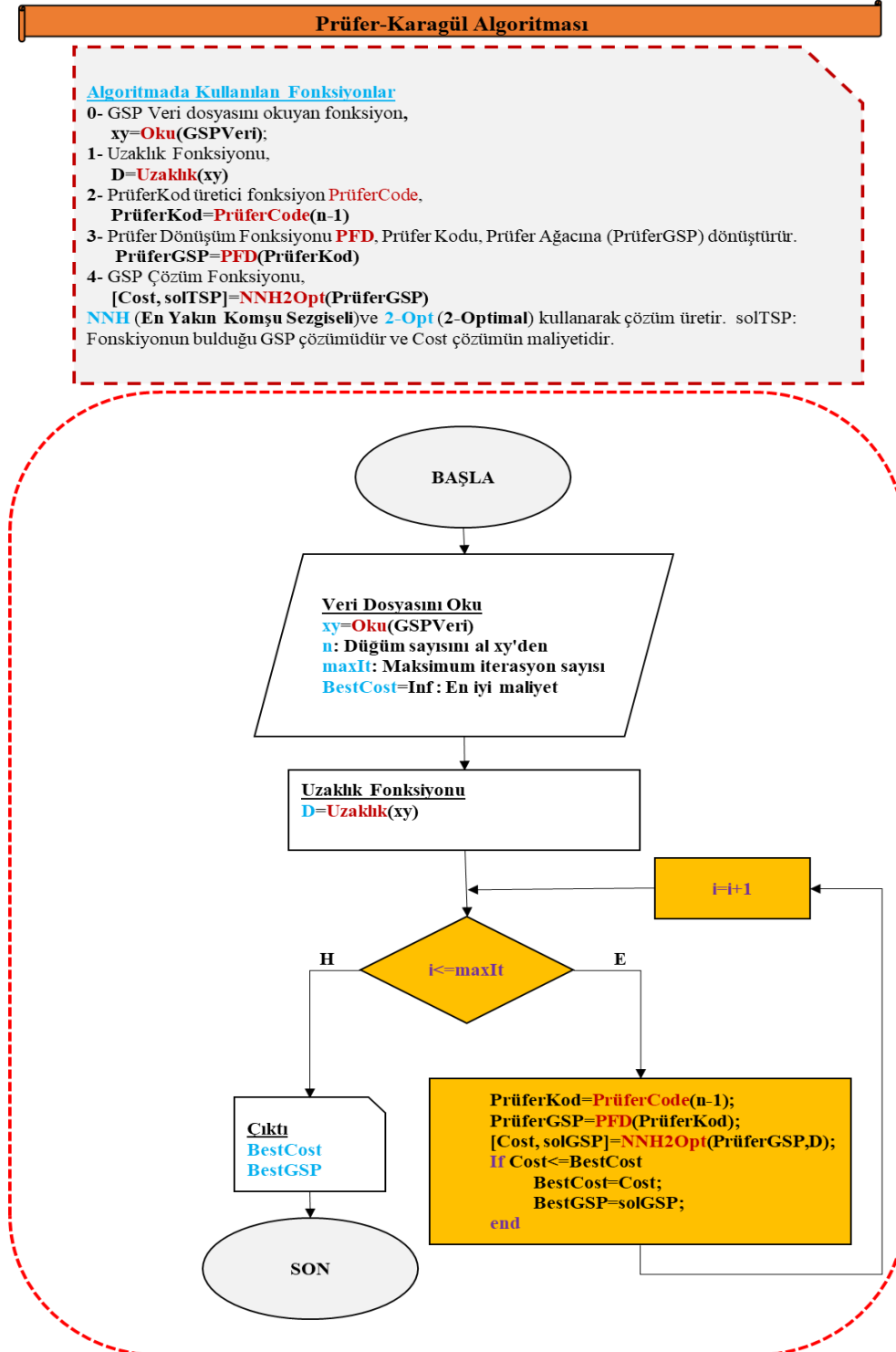
2.4. Önerilen Yöntem: Prüfer Kodun GSP Çözümüne Uygulanması

Yazar tarafından 10 düğümlü bir GSP rassal olarak üretilmiştir. Tablo 1’de örnek probleme ilişkin x ve y koordinatları verilmiştir.

Tablo 1: KTSP1 Örnek Problemine ilişkin koordinatlar

No	1	2	3	4	5	6	7	8	9	10
x	3	9	7	7	2	3	4	9	6	6
y	2	4	2	5	2	8	5	7	9	3

Önerilen yöntem, n şehirli bir GSP problemi için rassal olarak $(n - 1)$ düğümle Prüfer kodun üretilmesiyle başlar. Bu Prüfer kodunun ürettiği ağaç hesaplanır. Elde edilen ağaç GSP çözüm uzayında ilgili Prüfer koduna karşılık gelen GSP çözümüdür. Ancak elde edilen Prüfer Ağacına dayanan GSP çözümünden yüksek kaliteli bir GSP çözümü beklenmez. Daha kaliteli bir GSP çözümü elde edebilmek için Prüfer Ağacına dayanan GSP çözümü En Yakın Komşu Sezgiseli (NNH) ve 2-Opt algoritması ile geliştirilir. Böylece Prüfer ağacından elde edilen çözüme göre daha yüksek kalite de çözümlere erişilmiş olur. Şekil 3'te önerilen çözüm yöntemi Prüfer-Karagül Algoritmasının akış diyagramı gösterilmiştir.

**Şekil 3:** Önerilen Çözüm Yöntemi: Prüfer-Karagül Algoritması Akış Şeması

Şekil 3’te verilen diyagrama göre önerilen algoritma şu şekilde çalışacaktır:

A1 – Rassal olarak bir Prüfer kod üretilir.

A2 – Üretilen rassal Prüfer kodu ağaç yapısına dönüştürülür.

A3 – Elde edilen ağaç olası GSP çözümünde yer alması gereken düğüm sayısına eşittir.

A4 – Akış diyagramında elde edilen ağaç PrüferGSP olarak adlandırılmıştır.

A5 - Bu olası GSP çözümünün ilk düğümü NNH çözümünün başlangıç düğümü olarak ele alınır ve buradan elde edilen çözüme 2-Opt algoritması uygulanır.

A6 - Böylece önerilen yöntem bir GSP çözümü üretmiştir.

Bu noktadan sonra akış diyagramı ve akış diyagramına ait açıklamalar, Tablo 2’de KTSP1 probleminin, Birey No ile gösterilen her bir satırında rassal olarak üretilmiş “Prüfer kod”, Prüfer koda karşılık gelen “Prüfer Ağacı” ve bu ağacı anlamlı hale getiren “Prüfer GSP” ve GSP’ye ilişkin uzaklık ve bilgisayarın çözüm süresi yer almaktadır. Problem için on adet rassal Prüfer kod üretilmiştir. Tüm çözümler Tablo 2’de gösterilmiştir. Burada elde edilen en iyi GSP çözümleri 6 ve 8 numaralı bireylerden elde edilmiştir. Bu Tablo 2’deki en önemli ayırt edici özellik önerilen yaklaşımın ilk adımı olan Prüfer kod ile GSP çözümü elde edilmesini ifade eder. Yani bir başka deyişle NNH ve 2-Opt ile elde edilen PrüferGSP çözümleri geliştirilmemiştir.

Tablo 2: Prüfer Kodu ile elde edilen GSP çözümleri

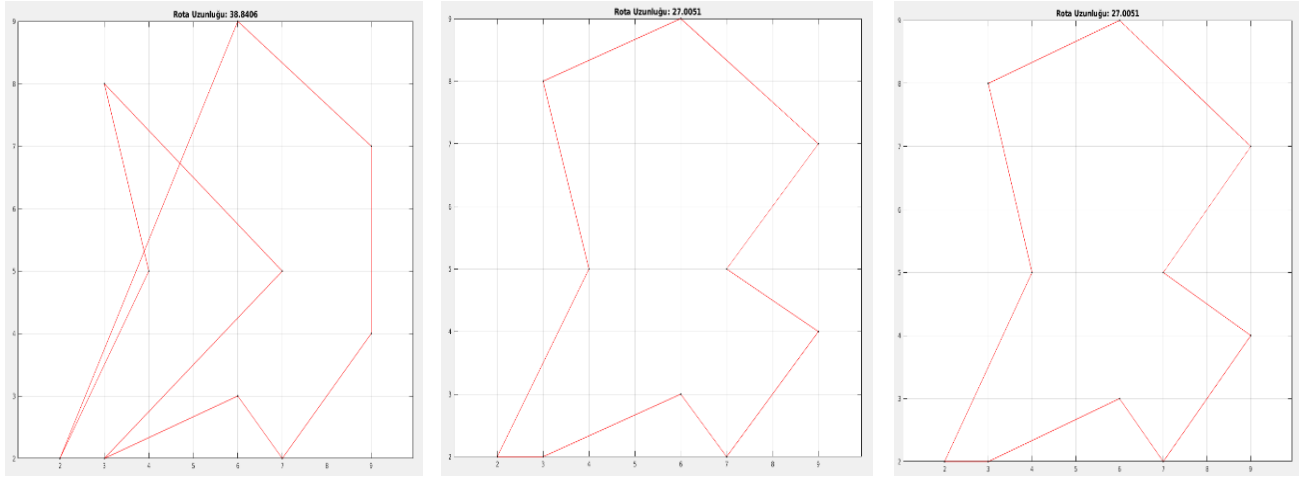
Birey No	Prüfer Kod	Prüfer Ağacı (Prüfer GSP)	Maliyet	Süre (sn)
1	[6,9,7,2,4,5,10,2,9]	[1,3,6,7,8,4,5,10,2,9]	49,15	0,005578
2	[7,4,2,5,5,2,6,3,4]	[1,7,8,9,10,5,2,6,3,4]	51,98	0,000623
3	[6,3,3,7,3,9,10,8,4]	[1,2,5,6,7,3,9,10,8,4]	52,99	0,000629
4	[6,2,10,9,9,3,6,1,5]	[4,7,2,8,10,9,3,6,1,5]	49,21	0,000463
5	[4,2,2,5,1,6,5,7,7]	[3,4,8,2,9,1,6,5,10,7]	45,55	0,001791
6	[7,1,1,4,6,7,5,9,8]	[2,3,10,1,4,6,7,5,9,8]	38,84	0,000276
7	[10,6,4,2,7,8,5,1,3]	[9,10,6,4,2,7,8,5,1,3]	50,22	0,000387
8	[2,3,5,6,5,9,6,10,7]	[1,2,3,4,8,5,9,6,10,7]	46,63	0,000204
9	[10,3,7,3,7,7,1,3,3]	[2,4,5,6,8,9,7,1,10,3]	38,88	0,000267
10	[7,9,4,8,7,1,7,4,10]	[2,3,5,6,8,9,1,7,4,10]	42,78	0,000271

Önerilen çözüm yöntemi Prüfer-Karagül yaklaşımı ile KTSP1 probleminin çözümleri Tablo 3’te detayları ile verilmiştir. Tablo 3’de bu kez elde edilen PrüferGSP çözümleri NNH ve 2-Opt ile geliştirilmiştir. Yani önerilen yöntem, Prüfer-Karagül yaklaşımı bir bütün olarak gösterilmiştir. Yine rassal Prüfer birey sayısı on adet olmakla birlikte burada NNH ve 2-Opt geliştirmesi ile yöntem her birey için bilinen en iyi çözümü bulmuştur. Tablo 2’de elde edilen en iyi çözüm 38,84 iken önerilen yöntemle elde edilen en iyi çözüm 27,01 elde edilmiştir. Bu problemin boyutu küçük olduğu için önerilen yöntem tüm olası çözümleri optimal çözüme ulaştırmıştır. Doğal olarak hesaplama karmaşıklığında meydana gelen artışa bağlı olarak süre bir miktar büyümektedir.

Tablo 3: Önerilen Çözüm Yöntemi: Prüfer-Karagül ile KTSP1 Probleminin Çözümü

Birey No	Prüfer Kod	Prüfer Ağacı	Prüfer-Karagül GSP	Maliyet	Süre (sn)
1	[1,5,5,5,8,4,8,5,1]	[2,3,6,7,9,10,4,8,5,1]	[4,2,3,10,1,5,7,6,9,8]	27,01	0,003024
2	[2,8,5,2,4,7,2,8,3]	[1,6,9,5,10,4,7,2,8,3]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,00063
3	[10,3,8,2,3,1,6,7,6]	[4,5,9,8,2,3,1,10,7,6]	[4,2,3,10,1,5,7,6,9,8]	27,01	0,000933
4	[5,7,7,7,7,10,3,8,3]	[1,2,4,5,6,7,9,10,8,3]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000783
5	[2,7,5,5,7,8,4,7,5]	[1,2,3,6,9,10,8,4,7,5]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,002679
6	[9,9,3,7,6,6,9,3,4]	[1,2,5,8,7,10,6,9,3,4]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000813
7	[2,10,7,5,7,6,7,6,8]	[1,2,3,4,5,9,10,7,6,8]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000665
8	[6,10,3,2,2,1,5,5,4]	[7,6,8,3,9,2,1,10,5,4]	[7,6,9,8,4,2,3,10,1,5]	27,01	0,00094
9	[8,7,8,10,10,2,2,7,1]	[3,4,5,6,8,9,10,2,7,1]	[7,6,9,8,4,2,3,10,1,5]	27,01	0,0007
10	[6,6,9,5,4,7,8,6,4]	[1,2,3,9,5,10,7,8,6,4]	[7,5,1,10,3,2,4,8,9,6]	27,01	0,000671

Şekil 4'te KTSP1 probleminin üç farklı yöntemle elde edilen en iyi çözümlerine ilişkin grafikler verilmiştir. Şekil 4(a), Prüfer çözümleri ile elde edilen ve en kötü çözümü göstermektedir. Şekil 4(b) Prüfer-Karagül ile elde edilen çözümdür ve Şekil 4(c) Rassal başlangıçlı En Yakın Komşu Sezgiseli (Nearest Neighborhood Heuristic:NNH) ve 2-Opt hibrit algoritmasının çözümünü göstermektedir. Bu problem için Prüfer-Karagül ile Rassal NNH+2-Opt algoritmaları arasında fark yoktur.



(a) Prüfer GSP çözümü

(b) Prüfer-Karagül GSP çözümü

(c)Rassal NNH+2-Opt çözümü

Şekil 4: KTSP1 probleminin Prüfer, Prüfer-Karagül ve NNH+2-Opt yaklaşımları ile çözümleri

He vd. (2015) makalesinde rassal olarak üretilmiş iki adet problem yer almaktadır. Örnek1 (Example1) adı verilen problemin uzaklık matrisi verilmiş, diğer problemin ise verilmemiştir. Bu nedenle Örnek1 adlı problem çalışma kapsamında önerilen yöntemle çözülmüş ve çözüm kaliteleri karşılaştırılmıştır. He vd. (2015) makalesinde optimal çözüme beş popülasyon büyüklüğünde ulaşırken, önerilen yöntem ile bir birey (popülasyon büyüklüğü 1) büyüklüğünde ulaşılmıştır. Bir bireyle 615, beş popülasyonla 506 sonucuna ulaşılabilmektedir. Önerilen yöntem 506 optimal maliyetini, Prüfer Kod, Prüfer Ağacı, Prüfer-Karagül GSP sırasıyla [3,2,1,3,6,9,1,10,2], [4,5,7,8,3,6,9,1,10,2], [5,1,3,4,8,2,9,6,7,10] bu dizilerle 0,0012 saniyede çözmüştür. Bu karşılaştırmadan da görüleceği üzere He vd. (2015) tarafından önerilen yaklaşım Prüfer-Karagül yaklaşımı ile karşılaştırıldığında karmaşık ve oldukça yavaş bir yöntemdir.

Küçük, orta ve büyük boy problemler için açıklanmalı örnek çözümler EK-2'de verilmiştir. Bu noktadan sonra Prüfer-Karagül algoritması için detaylı analizler takip eden bölümde verilecektir.

3. ÖNERİLEN YÖNTEMİN TEST PROBLEMLERİ İLE ANALİZİ

Önerilen yöntemin kodları Matlab 2016b ortamında geliştirilmiştir. Analizler için Intel Dual Core 2.40 GHz, 8 GB RAM özelliklerine ve Linux işletim sistemine sahip bir bilgisayarda tek işlemci kullanılarak analizler yapılmıştır. 10 düğümlü KTSP1 ve 14 düğümlü KTSP2 problemleri rassal olarak üretilirken diğer problemler TSPLIB kütüphanesinden alınmıştır [Web-5, 2018]. Ayrıca hexagonal yapıda olan 3 farklı boyutta test problemi [Web-4, 2018] çalışma kapsamında kullanılmıştır. Tüm GSP problemlerinin çözümünde düğüm sayısı ayırt etmeksizin 10 rassal Prüfer kodu üretilmiştir. Bir başka deyişle yöntem için popülasyon büyüklüğü 10 bireydir. Çalışma kapsamında kullanılan yöntemler ile on tekrarlı deneyler yapılmıştır.

Yapılan on tekrarlı deneylerin detayları EK-1'de sunulmaktadır. EK-1'de yer alan sonuçların ortalama değerlerinden Tablo 4 üretilmiştir. Tablo 4 en iyi çözümler açısından karşılaştırıldığında 19 problemin 10 tanesinde Prüfer-Karagül daha başarılı iken 5 problemde eşitlik söz konusudur. Dört problemde ise daha başarısız sonuçlar elde edildiği görülmektedir. Ortalamalar açısından bakıldığında ise Prüfer-Karagül algoritması 19 adet problemin on beşinde diğer yöntemlere üstünlük sağlamıştır.

Tablo 4: En iyi, En kötü ve Ortalama Çözüm Değerleri

Problemler				Prüfer-Karagül GSP Yaklaşımı				Rassal NN+ 2-Opt Yaklaşımı			
No	P. Adı	NS	OPT	Eİ	EK	ORT	OS (sn)	Eİ	EK	ORT	OS (sn)
1	KTSP1	10	27	27,0	27,0	27,0	0,0039	27,0	27,0	27,0	0,0004
2	KTSP2	14	131	133,2	133,2	133,2	0,0062	133,2	133,4	133,4	0,0004
3	eil51	51	426	433,8	435,5	434,2	0,0376	436,7	459,9	444,9	0,0008
4	berlin52	52	7542	7736,3	7841,4	7757,4	0,0389	7749,9	8009,9	7851,8	0,0007
5	st70	70	675	691,0	691,1	691,0	0,0644	691,0	745,3	715,5	0,0011
6	pr76	76	108159	109207,7	114285,3	111989,8	0,0773	110014,1	115985,6	112993,2	0,0013
7	eil76	76	538	559,5	570,0	564,2	0,0733	559,5	582,4	570,1	0,0011
8	kroA100	100	21282	21395,4	21395,4	21395,4	0,1425	21459,4	22182,8	21801,9	0,0021
9	kroB100	100	22141	22346,6	22346,6	22346,6	0,1478	22346,6	23251,9	22662,2	0,0019
10	eil101	101	629	652,1	658,9	653,2	0,1633	654,3	676,4	665,2	0,0020
11	bier127	127	118282	119565,4	120473,8	119656,3	0,2487	120503,4	126047,3	122072,3	0,0033
12	ch130	130	6110	6282,1	6382,9	6334,1	0,2570	6267,2	6668,8	6413,7	0,0035
13	ch150	150	6528	6596,7	6622,8	6620,2	0,3623	6610,7	6775,9	6676,0	0,0041
14	kroA150	150	26524	27387,3	28048,9	27513,4	0,3435	27241,6	28413,2	28070,9	0,0054
15	kroA200	200	29368	29666,4	29666,4	29666,4	0,1880	29672,9	31089,2	30233,6	0,0101
16	lin318	318	42029	43173,5	44208,0	43359,8	1,5516	43389,4	45134,6	44160,9	0,0329
17	hex162	162	1620	1637,7	1681,4	1654,6	0,3937	1637,7	1811,5	1744,6	0,0034
18	hex486	486	4860	5792,2	5792,2	5792,2	0,3798	5404,8	5822,1	5640,13	0,0069
19	hex1458	1458	14580	16332,0	16349,7	16337,3	51,2633	16107,1	16525,6	16270,7	2,6099

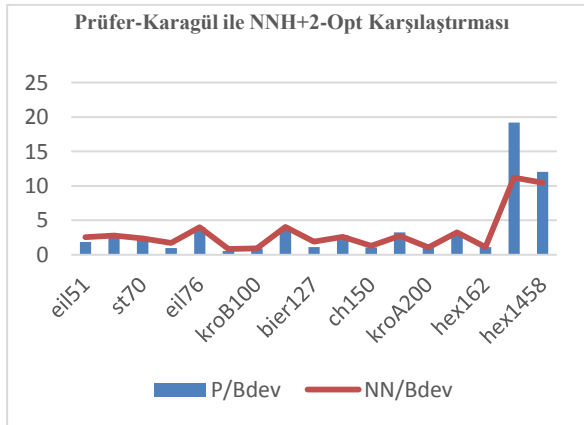
No: Problem No **P. Adı:** Problem Adı **NS:** Düğüm Sayısı **OPT:** Optimal Çözüm **Eİ:** En İyi Çözüm **EK:** En Kötü çözüm
ORT: Ortalama Çözüm (10 Çalıştırmanın ortalaması) **OS (sn):** Ortalama Süre (10 Çalıştırmanın ortalaması)

Ortalamlar üzerinden elde edilen özet tabloya dayanarak Tablo 5’de 17 GSP problemi için Prüfer-Karagül ve Rassal NNH+2-Opt algoritmalarının karşılaştırması için, en iyi çözüm ve Optimal (P/Bdev) ve (NN/Bdev) bağıl sapma karşılaştırmaları ve ortalama bağıl sapma karşılaştırmaları (P/Adev) ve (NN/Adev) verilmiştir. Bu tablodan da anlaşılacağı üzere en iyi çözümlerin optimalden sapmaları dikkate alındığında on yedi problemin dokuzunda Prüfer-Karagül daha başarılı sonuçlar üretmiştir. Ortalama çözüm başarısı açısından değerlendirildiğinde son iki problem dışında kalan problemler için daha iyi sonuçlar elde edilmiştir. Şekil 5’de ise Prüfer-Karagül ve Rassal NNH+2-Opt yöntemlerinin en iyi ve ortalama çözümleri için karşılaştırma grafikleri verilmiştir.

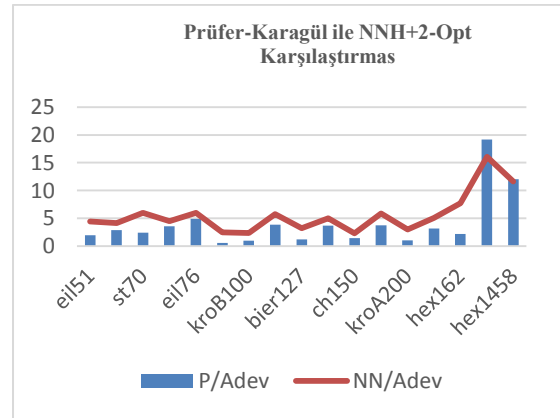
Tablo 5: Çözüm değerlerinin kıyaslanması

P. Adı	Prüfer-Karagül		Rassal NNH+2-Opt		Karagül vd (2016)		Karagül vd (2016)	
	P/Bdev	P/Adev	NN/Bdev	NN/Adev	P/Bdev	C-2-Opt/Bdev	P/Adev	C-2-Opt/Adev
eil51	1,84	1,92	2,52	4,43	1,84	1,17	1,92	5,52
berlin52	2,58	2,86	2,76	4,11	2,58	0,00	2,86	8,67
st70	2,37	2,38	2,37	5,99	2,37	0,44	2,38	5,50
pr76	0,97	3,54	1,72	4,47	0,97	0,72	3,54	3,94
eil76	4,00	4,87	4,00	5,97	4,00	1,86	4,87	6,54
kroA100	0,53	0,53	0,83	2,44	0,53	0,40	0,53	4,84
kroB100	0,93	0,93	0,93	2,35	0,93	1,12	0,93	5,72
eil101	3,67	3,85	4,02	5,75	3,67	4,77	3,85	7,62
bier127	1,09	1,16	1,88	3,20	1,09	0,95	1,16	6,82
ch130	2,82	3,67	2,57	4,97	2,82	1,47	3,67	6,34
ch150	1,05	1,41	1,27	2,27	1,05	2,90	1,41	7,53
kroA150	3,25	3,73	2,71	5,83	3,25	2,38	3,73	6,96
kroA200	1,02	1,02	1,04	2,95	1,02	1,66	1,02	6,25
lin318	2,72	3,17	3,24	5,07	2,72	4,25	3,17	7,05
hex162	1,09	2,14	1,09	7,69				
hex486	19,18	19,18	11,21	16,05				
hex1458	12,02	12,05	10,47	11,60				
Ortalama	3,60	4,02	3,21	5,60	2,06	1,72	2,50	6,38
Ortalama*	2,00	2,48	2,20	4,50				

P/Bdev: En iyi çözümlerin optimalden % sapması – Prüfer-Karagül
P/Adev: Çözümlerin ortalamasının optimalden % sapması – Prüfer-Karagül
NN/Bdev: En iyi çözümlerin optimalden % sapması – NNH+2-Opt
NN/Adev: Çözümlerin ortalamasının optimalden % sapması – NNH+2-Opt
P/Bdev: Bulunan en iyinin optimalden % sapması- Prüfer-Karagül
C-2-Opt/Bdev: Bulunan en iyinin optimalden % sapması- % Sapma- Klasik 2-Opt
P/Adev: Bulunan çözümlerin ortalamasının optimalden % sapması- Prüfer-Karagül
C-2-Opt/Adev: Bulunan çözümlerin ortalamasının optimalden % sapması- % Sapma- Klasik 2-Opt
Ortalama: 17 Probleme ilişkin ortalamalar
Ortalama*: hex486 ve hex1458 hariç ortalamalar



(a) Prüfer-Karagül ve NNH+2-Opt-En iyi

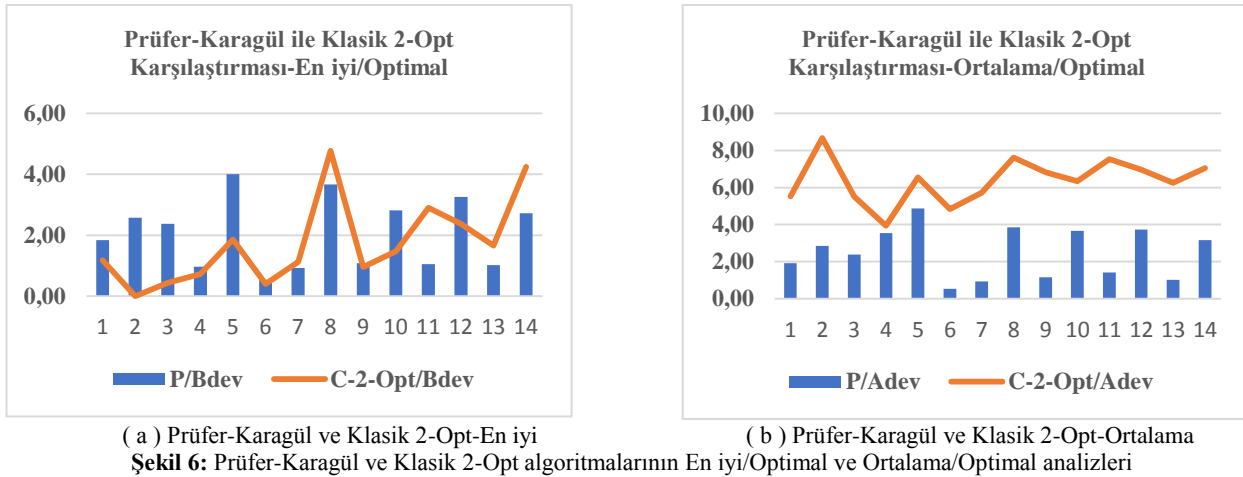


(b) Prüfer-Karagül ve NNH+2-Opt-Ortalama

Şekil 5: Prüfer-Karagül ve NNH+2-Opt algoritmalarının En iyi/Optimal ve Ortalama/Optimal analizleri

Karagül vd. (2016) tarafından yapılan evrimsel algoritma çalışmasında klasik 2-Opt ile aynı problem setinden 14 problem analiz edilmiştir. Bu 14 problemin Prüfer-Karagül ve Klasik 2-Opt karşılaştırmasına ilişkin sonuçlar Tablo 5’de verilmiştir. Tablo 5’de en iyi çözümlerin optimalden sapmaları ile ortalama çözümlerin optimalden sapmaları karşılaştırılmıştır. Tablo 5’den görüleceği üzere en iyi/optimal sapmaların ortalamasında Prüfer-Karagül belli oranda geri kalırken ortalama/optimal sapmaların ortalamasında oldukça

iyi sonuçlar sunmaktadır. Yani yöntem rassallıktan uzak olarak her zaman kararlı bir şekilde optimale makul ölçülerde yaklaşmaktadır. Tablo 5'deki karşılaştırmalar Şekil 6 (a) ve (b) de çok açık bir şekilde gözlenmektedir.



(a) Prüfer-Karagül ve Klasik 2-Opt-En iyi

(b) Prüfer-Karagül ve Klasik 2-Opt-Ortalama/Optimal

Şekil 6: Prüfer-Karagül ve Klasik 2-Opt algoritmalarının En iyi/Optimal ve Ortalama/Optimal analizleri

Yapılan analizler de ortaya çıkan açık sonuç Prüfer-Karagül algoritması Rassal NNH+2-Opt ve Klasik 2-Opt ile karşılaştırıldığında görece olarak en iyi çözümleri yakalama konusunda rekabetçi iken ortalama çözümler anlamında önemli ölçüde başarılı olduğu görülmektedir. Bu bağlamda Prüfer kodlamayı temel alan Prüfer-Karagül yaklaşımı GSP problemlerinin çözümünde %5 optimalden uzaklık seviyesinin altında kalan çözümleri ile kabul edilebilir bir çözüm yaklaşımı olabileceğini ortaya koymaktadır. Ancak büyük boyutlu hex486 ve hex1458 problemleri için elde edilen en iyi/optimal ve ortalama/optimal sapmaları açısından kabul edilebilir makul sınırları aştığı gözlemlenmiştir. Ancak NNH+2-Opt yaklaşımı içinde aynı sorunun varlığı gözlemlenmiştir.

SONUÇ

Bu çalışmada GSP'nin çözümü için Prüfer kod üzerinden ağaçların yapılandırmasına dayanan yeni bir yapısal sezgisel algoritma önerilmiştir. Önerilen yöntem, küçük problemlerin çözümü üzerinden detaylı bir şekilde açıklanmıştır. Ele alınan örnekler ve sayısal analizler önerilen yöntemin görece üstünlüğünü açıkça ortaya koymaktadır. Bu bağlamda yöntem, 19 adet test probleminin 17 tanesinde optimalden %5'in altında kalan sapmalar göstermiştir. Aynı zamanda optimalden sapmaların ortalaması olarak %2,06 değeri ile yapısal çözüm yaklaşımları açısından literatürde yer alan Klasik 2-Opt yaklaşımı ile karşılaştırıldığında ortalama çözümlerde oldukça başarılı olurken, optimal çözümlerden en iyi çözümlerin sapmalarına bakıldığında da rekabetçi olabileceğini ortaya koymuştur. Gelecekte yapılacak çalışmalarda önerilen yöntemin farklı sezgisel yöntemler ile birlikte kullanılarak geliştirilebileceği ve diğer bütünleşik optimizasyon problemlerinin çözümünde de kullanılabileceği düşünülmektedir. Çalışmanın bir diğer önemli boyutu küçük ve orta boy problemlerde çok hızlı şekilde optimal çözüme kabul edilebilir şekilde yaklaştığı için, endüstride ve askeri alanlarda farklı uygulama olanakları olabilir.

KAYNAKLAR

- ANBUUDAYASANKAR, S. P., GANESH, K., MOHAPATRA, S. (2014), Models for Practical Routing Problems in Logistics: Design and Practices. Springer International Publishing. 43-68.
- BASU, S. (2012), Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey, *American Journal of Operations Research*, 2, 163-173. doi:10.4236/ajor.2012.22019
- BORYCZKA, U., SZWARC, K. (2019), The Adaptation of the Harmony Search Algorithm to the ATSP with the Evaluation of the Influence of the Pitch Adjustment Place on the Quality of Results. *Journal of Information and Telecommunication*. 3, 1, 2-18.
- CROES, G. A. (1958), A Method for Solving Traveling-Salesman Problems. *Operations Research*. 6, 6, 791-812.
- DEO, N., MICIKEVICIUS, P. (t.y.), Prüfer-Like Codes for Labeled Trees, 16 Aralık 2018 tarihinde, <https://pdfs.semanticscholar.org/52be/20fcdc9ed956fc686c4492f59ea51ca537a7.pdf> adresinden alındı.
- DORIGO, M., GAMBARDELLA, L. M. (1997), Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1 (1), 53-66.
- ERDEM, Y , KESKİNTÜRK, T . (2011), Kanguru Algoritması ve Gezgin Satıcı Problemine Uygulanması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 10 (19). Retrieved from <http://dergipark.gov.tr/ticaretfbid/issue/21359/229115>
- GENDREAU, M., LAPORTE, G., SEMET, F. (1998), A Tabu Search Heuristic for the Undirected Selective Travelling Salesman Problem. *European Journal of Operational Research*, 106 (2-3), 539-545.
- GENDREAU, M., 2002, An Introduction to Tabu Search , University of Montreal, Montreal, Canada.
- GEEM, ZW., KIM, JH, LOGANATHAN, GV. (2001) A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76, 60-68.
- GLOVER, F. (1990), Tabu Search: A Tutorial Technical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, Colorado.
- GLOVER, F. and LAGUNA, M. 1993, Tabu Search, COLIN R. REEVES (Ed.),70—150, Blackwell Scientific publications, Oxford.
- GLOVER, F. and LAGUNA, M. 1997, Tabu Search, Kluwer Academic Publishers, Norwell, MA.
- HAJIAGHAEI-KESHTELI, M. (2011). The Allocation of Customers to Potential Distribution Centers in Supply Chain Networks: GA and AIA Approaches. *Applied Soft Computing*, 11, 2069-2078.
- HE, R., MA, C., ZHANG, W., XIAO, Q. (2015). Optimisation Algorithm for Logistics Distribution Route Based on Prüfer Codes. *Int. J. Wireless and Mobile Computing*. 9 (2), 205–210.
- HTUN, TT. (2018), A Survey Review on Solving Algorithms for Travelling Salesman Problem (TSP). *International Journal of Scientific and Research Publications*, 8, 12, 630-633.
- JAFARI-MARANDI, R., SMITH, BK. (2017), Fluid Genetic Algorithm (FGA). *Journal of Computational Design and Engineering*, 4, 2, 158-167.
- KARAGÜL, K., Guguk Kuşu Algoritması: Bir Plastik Atık Toplama Uygulaması. 15th International Symposium on Econometrics, Operations Research and Statistic, Isparta, Turkey, 15, 775-784, 2014.
- KARAGUL, K., AYDEMİR, E., TOKAT, S. (2016). Using 2- Opt Based Evolution Strategy for Travelling Salesman Problem. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 6, 103–113.
- KARKORY, FA., ABUDALMOLA, AA. (2013), Implementation of Heuristics for Solving Travelling Salesman Problem Using Nearest Neighbour and Minimum Spanning Tree Algorithms. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*. 7,10.
- KESKİNTÜRK, T., KİREMİTÇİ, B., KİREMİTÇİ, S. (2016), 2-Opt Algoritması ve Başlangıç Çözümünün Algoritma Sonuçları Üzerindeki Etkisi. *Endüstri Mühendisliği Dergisi*, 27, 3, 2-12.
- KIZILATEŞ, G., NURIYEVA, F. (2013). On the Nearest Neighbor Algorithms for the Traveling Salesman Problem. *Advances in Computational Science, Engineering and Information Technology*, 111–118. doi:10.1007/978-3-319-00951-3_11
- KIRKPATRICK, S., GERLATT, C. D. JR., VECCHI, M.P. (1983). Optimization by Simulated Annealing, *Science*, 220, P.. 671-680.
- MAVROVOUNIOTIS, M., YANG, S. (2013), Ant Colony Optimization with Immigrants Schemes for the Dynamic Travelling Salesman Problem with Traffic Factors. *Applied Soft Computing*, 13(10), 4023-4037, 2013.

- NURİYEVA, F., KIZILATEŞ, G. (2016), Gezgın Satıcı Problemi İin Merkezden Kenarlara Hipersezgisel Yöntem. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 20 (2),319-323
- ODILI, J. (2018), The Dawn Of Metaheuristic Algorithms. *International Journal of Software Engineering and Computer Systems*, 4(2), 49-61. Retrieved from <http://journal.ump.edu.my/ijsecs/article/view/703>
- ODILI, J. B., KAHAR, M. N., NORAZIAH, A., ZARINA, M., UL HAQ, R. (2017), Performance Analyses of Nature-Inspired Algorithms on the Traveling Salesman's Problems for Strategic Management. <https://doi.org/10.1080/10798587.2017.1334370>
- ÖZDAĞOĞLU, G. (2008), Simulated Annealing Application on Flowshop Sequencing Problem: A Comparative Case Study. *Atatürk Üniversitesi İİBF Dergisi*, 22, 2, 357-377.
- PALA, O , AKSARAYLI, M . (2018), Asimetrik Gezgın Satıcı Problemine Bulanık Karınca Kolonisi Optimizasyon Algoritması ile Çözüm Yaklaşımı. *Journal of Transportation and Logistics*, 3 (1), 25-34. Retrieved from <http://dergipark.gov.tr/jtl/issue/37196/366856>
- PAULDEN, T., SMITH, D. K. (2007). Developing New Locality Results for the Prüfer Code Using a Remarkable Linear-Time Decoding Algorithm. *The Electronic Journal of Combinatorics*, 14.
- POLLARD, JM. (1978), Monte Carlo Methods for Index Computation. *Mathematics Of Computation*. 32, 143, 918-924.
- ROMSY, M. (2011), Adaption of Pollard's kangaroo algorithm to the FACTOR problem. Cryptology ePrint Archive: Report 2011/483. <https://eprint.iacr.org/2011/483>.
- SÜRAL, H. (2003), Gezgın Satıcı Problemi. *Matematik Dünyası*, 10 Aralık 2018 tarihinde http://www.matematikdunyasi.org/arsiv/PDF/03_3_37_40_GEZGIN.pdf adresinden alınmıştır.
- ŞAHİN, Y., KARAGÜL, K. (2019). Gezgın Satıcı Probleminin Melez Akışkan Genetik Algoritma (MAGA) Kullanarak Çözümü. doi: 10.5505/pajes.2018.81084, Retrieved https://www.journalagent.com/pajes/pdfs/PAJES-81084-RESEARCH_ARTICLE-SAHIN.pdf.
- WANG, X., WANG, L., WU, Y. (2009). An Optimal Algorithm for Prufer Codes. *J. Software Engineering & Applications*. 2: 111-115. doi:10.4236/jsea.2009.22016
- THOMPSON, E., PAULDEN, T., SMITH, D. K. (2007), The Dandelion Code: A New Coding of Spanning Trees for Genetic Algorithms. *IEEE Transactions On Evolutionary Computation*, 11 (1). 91-100.
- KOUNALAKIS, E., KAPELONIS, K. (2002), Chapter 10: Travelling Salesman Problem, In HY-583 Chart Algorithms Student Notes: Autumn 2002, (eds. Ioannis Tollis).
- WANG, Z., GENG, X., SHAO, Z. (2009), An Effective Simulated Annealing Algorithm for Solving the Traveling Salesman Problem. *Journal of Computational and Theoretical Nanoscience*, 6, 1680-1686.
- YANG, X-S. (2009), Harmony Search as a Metaheuristic Algorithm in: *Music-Inspired Harmony Search Algorithm: Theory and Applications*, Z. W. Geem (Editör), Studies in Computational Intelligence, Springer Berlin, 191, 1-14.
- YUN, H-Y., JEONG, S-J., KIM, K-S. (2013), Advanced Harmony Search with Ant Colony Optimization for Solving the Traveling Salesman Problem. *Journal of Applied Mathematics*, 2, 1-8. DOI: 10.1155/2013/123738
- ZHAO, F, LI, S., SUN J, MEI, D. (2009), Genetic Algorithm for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. *Computers & Industrial Engineering*, 56(4), 1642-1648.
- ZHOU,A-H., ZHU, L-P., HU, B., DENG, S., SONG, Y., QIU, H., PAN, S. (2019), Traveling-Salesman-Problem Algorithm Based on Simulated Annealing and Gene-Expression Programming. *Information*, 10, 7; doi:10.3390/info10010007.
- Web-1: Wikipedia (t.y), Prüfer Sequence, 15 Aralık 2018 tarihinde, <http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUHLDvGZlc19zZXF1ZW5jZQ> adresinden alındı.
- Web-2: CP-Algorithms (t.y), Prüfer Code, 15 Aralık 2018 tarihinde, https://cp-algorithms.com/graph/pruefer_code.html adresinden alınmıştır.
- Web-3: Prüfer Codes (t.y), 15 Aralık 2018 tarihinde, https://ptwiddle.github.io/Graph-Theory-Notes/s_graphalgorithms_prufer.html adresinden alınmıştır.
- Web-4: Heuristic Method for the Travelling Salesman Problem (TSP) in Matlab (t.y), 1 Ağustos 2018 tarihinde, <http://freecodecode.net/matlabprojects/65173/heuristic-method-for-the-traveling-salesman-problem-%28tsp%29--in-matlab> adresinden alınmıştır.
- Web-5: TSPLIB - IWR, Heidelberg - Universität Heidelberg, 1 Ağustos 2018 tarihinde, <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> adresinden alınmıştır.

Web-6: The Traveling Salesman Problem 3: Nearest Neighbor Heuristic (t.y.), 29 Nisan 2019 tarihinde, <http://demonstrations.wolfram.com/TheTravelingSalesmanProblem3NearestNeighborHeuristic/> , Erişim Tarihi:29/04/2019.

EKLER

EK-1: Test Problemleri İçin Prüfer-Karagül ve Rassel NNH+2-Opt Algoritmalarının 10 Çalıştırma Kayıtları

KTSP1	Prüfer-Karagül		Rassel NNH+2-Opt		KTSP2	Prüfer-Karagül		Rassel NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	27,01	0,0059	27,01	0,000593	1	133,23	0,0086	133,40	0,00044
2	27,01	0,0054	27,01	0,000594	2	133,23	0,0087	133,23	0,000613
3	27,01	0,0032	27,01	0,000522	3	133,23	0,0078	133,40	0,000696
4	27,01	0,0039	27,01	0,000398	4	133,23	0,0053	133,40	0,000258
5	27,01	0,0035	27,01	0,000502	5	133,23	0,0051	133,40	0,000619
6	27,01	0,0034	27,01	0,000612	6	133,23	0,0064	133,40	0,000237
7	27,01	0,0037	27,01	0,000196	7	133,23	0,0051	133,40	0,000331
8	27,01	0,0035	27,01	0,000222	8	133,23	0,0048	133,40	0,000199
9	27,01	0,0032	27,01	0,000282	9	133,23	0,005	133,40	0,000622
10	27,01	0,0033	27,01	0,000168	10	133,23	0,0047	133,40	0,000206
Ortalama	27,01	0,0039	27,01	0,0004	Ortalama	133,23	0,0062	133,39	0,0004
eil51	Prüfer-Karagül		Rassel NNH+2-Opt		berlin52	Prüfer-Karagül		Rassel NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	433,84	0,0431	445,83	0,000856	1	7736,35	0,0424	7749,88	0,000772
2	433,84	0,0409	436,74	0,000696	2	7736,35	0,0404	7844,25	0,000639
3	433,84	0,0349	441,49	0,000854	3	7841,39	0,038	8009,87	0,000862
4	433,84	0,0367	459,86	0,000731	4	7736,35	0,0401	7944,50	0,000598
5	435,51	0,0371	447,83	0,001441	5	7841,39	0,0364	7844,25	0,000754
6	433,84	0,0379	441,49	0,000699	6	7736,35	0,0381	7863,91	0,000601
7	435,51	0,0355	441,49	0,001065	7	7736,35	0,0387	7844,25	0,00063
8	433,84	0,0368	450,05	0,000561	8	7736,35	0,0422	7850,29	0,00075
9	433,84	0,0363	436,87	0,000704	9	7736,35	0,0364	7799,61	0,000531
10	433,84	0,0367	447,12	0,000534	10	7736,35	0,0363	7766,98	0,000589
Ortalama	434,17	0,0376	444,88	0,0008	Ortalama	7757,35	0,0389	7851,78	0,0007
st70	Prüfer-Karagül		Rassel NNH+2-Opt		pr76	Prüfer-Karagül		Rassel NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	691,00	0,0661	710,36	0,001118	1	110716,6	0,079	114148,5	0,001163
2	691,08	0,0701	739,49	0,001173	2	110716,6	0,0799	113766,4	0,001092
3	691,08	0,0625	707,11	0,001409	3	114285,3	0,073	112879,7	0,001432
4	691,00	0,0640	691,00	0,001344	4	114285,3	0,0841	112879,7	0,001203
5	691,08	0,0598	694,59	0,001297	5	113175,1	0,073	113403,8	0,001205
6	691,08	0,0643	745,28	0,001067	6	111075,2	0,0817	112557,7	0,001474
7	691,00	0,0636	718,99	0,000794	7	111075,2	0,0758	110014,1	0,001902
8	691,08	0,0610	739,49	0,001018	8	109207,7	0,0747	115985,6	0,001127
9	691,08	0,0698	710,36	0,000832	9	114285,3	0,076	113409,1	0,001122
10	691,00	0,0623	697,89	0,000814	10	111075,2	0,076	110887,1	0,001257
Ortalama	691,05	0,0644	715,46	0,0011	Ortalama	111989,7	0,0773	112993,2	0,0013

EK-1: Devamı

eil76	Prüfer-Karagül		Rassal NNH+2-Opt		kroA100	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	569,1	0,0741	569,5	0,001158	1	21395,4	0,1342	21889,6	0,002076
2	570,0	0,0772	559,5	0,001224	2	21395,4	0,1434	22182,8	0,002389
3	561,1	0,0749	570,5	0,001263	3	21395,4	0,141	21653,3	0,002111
4	561,1	0,0725	570,0	0,001406	4	21395,4	0,1768	21678,7	0,002228
5	561,1	0,0722	572,2	0,001095	5	21395,4	0,1331	21648,8	0,002355
6	561,1	0,0735	563,4	0,000823	6	21395,4	0,1339	22066,8	0,001885
7	569,1	0,0697	574,3	0,001013	7	21395,4	0,1372	21678,7	0,002094
8	569,1	0,0728	568,4	0,000953	8	21395,4	0,1283	21459,4	0,001924
9	561,1	0,0737	582,4	0,000961	9	21395,4	0,1325	21855,2	0,001968
10	559,5	0,0726	571,1	0,000795	10	21395,4	0,1648	21905,5	0,001627
Ortalama	564,21	0,0733	570,13	0,0011	Ortalama	21395,4	0,1425	21801,88	0,0021
kroB100	Prüfer-Karagül		Rassal NNH+2-Opt		eil101	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	22346,6	0,1344	22346,6	0,001919	1	654,3	0,1376	667,8	0,001859
2	22346,6	0,1285	22346,6	0,001822	2	654,3	0,1414	675,0	0,002037
3	22346,6	0,2319	22420,5	0,002055	3	652,1	0,1793	660,2	0,002318
4	22346,6	0,1571	22725,4	0,001559	4	652,1	0,1893	656,6	0,001901
5	22346,6	0,1265	23251,9	0,001941	5	658,9	0,1933	676,4	0,002313
6	22346,6	0,1316	22725,4	0,001719	6	652,1	0,1546	656,6	0,00254
7	22566,2	0,1294	23182,9	0,001961	7	652,1	0,1733	654,3	0,002261
8	22346,6	0,132	22831,5	0,002116	8	652,1	0,1343	672,9	0,001616
9	22346,6	0,179	22354,3	0,00174	9	652,1	0,1365	658,8	0,001859
10	22346,6	0,1274	22437,1	0,001792	10	652,1	0,1934	673,2	0,001704
Ortalama	22368,6	0,1478	22662,22	0,0019	Ortalama	653,21	0,1633	665,18	0,0020
bier127	Prüfer-Karagül		Rassal NNH+2-Opt		ch130	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	119565,4	0,3415	121282,7	0,003441	1	6282,1	0,2364	6341,1	0,00343
2	119565,4	0,2109	120505,6	0,003956	2	6325,7	0,2296	6668,8	0,003255
3	119565,4	0,2198	126047,3	0,002614	3	6382,9	0,2231	6282,1	0,005032
4	119565,4	0,2261	122451,1	0,003669	4	6282,1	0,3658	6365,7	0,003247
5	119565,4	0,2091	122987,4	0,003009	5	6325,7	0,2358	6445,9	0,002751
6	119565,4	0,2544	122683,7	0,002835	6	6325,7	0,2264	6493,8	0,002883
7	119565,4	0,2226	120669,3	0,003143	7	6325,7	0,2311	6526,3	0,003204
8	119565,4	0,2186	122562,9	0,003879	8	6382,9	0,2107	6267,2	0,003799
9	119565,4	0,3259	121030,1	0,003269	9	6382,9	0,3841	6409,5	0,003016
10	120473,8	0,2578	120503,4	0,003151	10	6325,7	0,2273	6336,4	0,004795
Ortalama	119656,3	0,2487	122072,34	0,0033	Ortalama	6334,14	0,2570	6413,69	0,0035

EK-1: Devamı

ch150	Prüfer-Karagül		Rassal NNH+2-Opt		kroA150	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	6622,8	0,3014	6656,2	0,003485	1	27387,3	0,3038	28262,7	0,004948
2	6622,8	0,3617	6645,0	0,003904	2	27387,3	0,4046	28413,2	0,004922
3	6622,8	0,2855	6610,7	0,004154	3	27387,3	0,3083	28413,2	0,005216
4	6622,8	0,4966	6656,2	0,004013	4	28048,9	0,3246	28171,9	0,004749
5	6622,8	0,2915	6674,0	0,004759	5	27387,3	0,3106	27241,6	0,005354
6	6596,7	0,3017	6723,8	0,00467	6	27987,1	0,5155	27318,0	0,00573
7	6622,8	0,5231	6653,5	0,003892	7	27387,3	0,3407	28317,4	0,005835
8	6622,8	0,2977	6775,9	0,004452	8	27387,3	0,3063	27934,6	0,005612
9	6622,8	0,4716	6742,3	0,004531	9	27387,3	0,3038	28317,4	0,005927
10	6622,8	0,2924	6622,8	0,003576	10	27387,3	0,3165	28318,8	0,005398
Ortalama	6620,21	0,3623	6676,04	0,0041	Ortalama	27513,41	0,3435	28070,88	0,0054
kroA200	Prüfer-Karagül		Rassal NNH+2-Opt		lin318	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	29666,4	0,5634	30189,1	0,009745	1	43173,5	1,6723	43842,6	0,029573
2	29666,4	0,7528	29672,9	0,009259	2	43173,5	1,6735	45134,6	0,029274
3	29666,4	0,5314	29966,8	0,00989	3	43173,5	1,636	44376,4	0,034614
4	29966,8	0,5618	30643,7	0,011193	4	43173,5	1,5707	44642,4	0,034373
5	29666,4	0,6191	30560,3	0,00911	5	44001,3	1,4272	43117,6	0,034622
6	29666,4	0,6494	30076,0	0,010153	6	43173,5	1,4502	44057,4	0,034089
7	29666,4	0,5406	31089,2	0,011332	7	43173,5	1,428	44715,7	0,036967
8	29666,4	0,5308	30067,1	0,011036	8	43173,5	1,486	44351,1	0,033222
9	29666,4	0,5274	30035,6	0,009873	9	44208,0	1,5225	43981,7	0,030145
10	29666,4	0,5603	30035,6	0,009645	10	43173,5	1,6496	43389,4	0,03212
Ortalama	29696,44	0,5837	30233,63	0,0101	Ortalama	43359,76	1,5516	44160,90	0,0329
hex162	Prüfer-Karagül		Rassal NNH+2-Opt		hex486	Prüfer-Karagül		Rassal NNH+2-Opt	
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)	#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)
1	1637,7	0,3767	1811,5	0,00333	1	5792,2	3,4289	5728,0	0,064409
2	1637,7	0,3472	1786,7	0,001913	2	5792,2	3,3176	5822,1	0,048071
3	1658,6	0,36	1688,0	0,002766	3	5792,2	3,3025	5656,9	0,07921
4	1637,7	0,3466	1754,3	0,004966	4	5792,2	3,4572	5404,8	0,081089
5	1681,4	0,3492	1737,1	0,003645	5	5792,2	3,5019	5695,5	0,074883
6	1637,7	0,3497	1761,9	0,002802	6	5792,2	3,3262	5404,8	0,076419
7	1658,6	0,3545	1637,7	0,004733	7	5792,2	3,3398	5708,8	0,066839
8	1679,5	0,5442	1771,0	0,003737	8	5792,2	3,2306	5665,8	0,070832
9	1637,7	0,5612	1723,5	0,003863	9	5792,2	3,3955	5629,8	0,070429
10	1679,5	0,3473	1774,0	0,002096	10	5792,2	3,282	5684,8	0,074008
Ortalama	1654,62	0,3937	1744,58	0,0034	Ortalama	5792,17	3,3582	5640,13	0,0706

EK-1: Devamı

hex1458	Prüfer-Karagül		Rassal NNH+2-Opt					
#OfRun	Çözüm	Süre(sn)	Çözüm	Süre(sn)				
1	16332,0	50,7379	16184,5	3,30505				
2	16349,7	51,8217	16290,7	2,799008				
3	16332,0	51,6311	16308,9	2,650262				
4	16332,0	51,4293	16339,2	2,794216				
5	16349,7	51,5446	16240,1	2,549651				
6	16332,0	50,9283	16107,1	2,846566				
7	16332,0	51,7934	16179,4	2,212796				
8	16349,7	50,3628	16297,9	1,99824				
9	16332,0	51,2723	16525,6	2,530444				
10	16332,0	51,1113	16234,1	2,413094				
Ortalama	16337,28	51,2633	16270,75	2,6099				

EK-2: Farklı Örnek Problemler İçin GSP Çözümü ve Kıyaslamalar

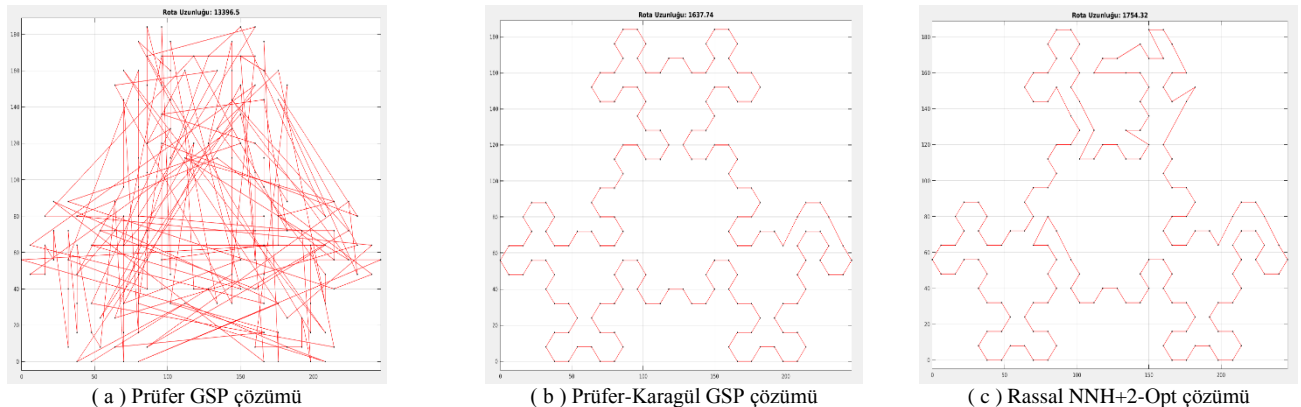
Bu bölümde farklı düğüm sayılarına sahip GSP problemleri için elde edilen çözümler gösterilmektedir. Bu bağlamda 14, 162, 486 ve 1458 düğümlü dört farklı problem üzerinde çözümler analiz edilmiştir. 14 düğümlü problem için detaylı bir çözüm tablosu verilirken, diğer problemler de Prüfer, Prüfer-Karagül ve Rassal NNH+2-Opt çözüm yaklaşımları karşılaştırılmıştır.

Tablo 1’de 14 düğümlü başka bir GSP problemine ilişkin Prüfer-Karagül ve Rassal NNH+2-Opt yöntemleri karşılaştırılmaktadır. Bu tabloda yer alan çözümler arasında çözüm kaliteleri açısından herhangi bir farklılık olmadığı görülmektedir. Ancak çözüm süreleri açısından Rassal NNH+2-Opt yönteminin daha avantajlı olduğu söylenebilir.

Tablo 1: KTSP2-14 Düğümlü GSP İçin Prüfer-Karagül ve Rassal NNH+2-Opt Karşılaştırması

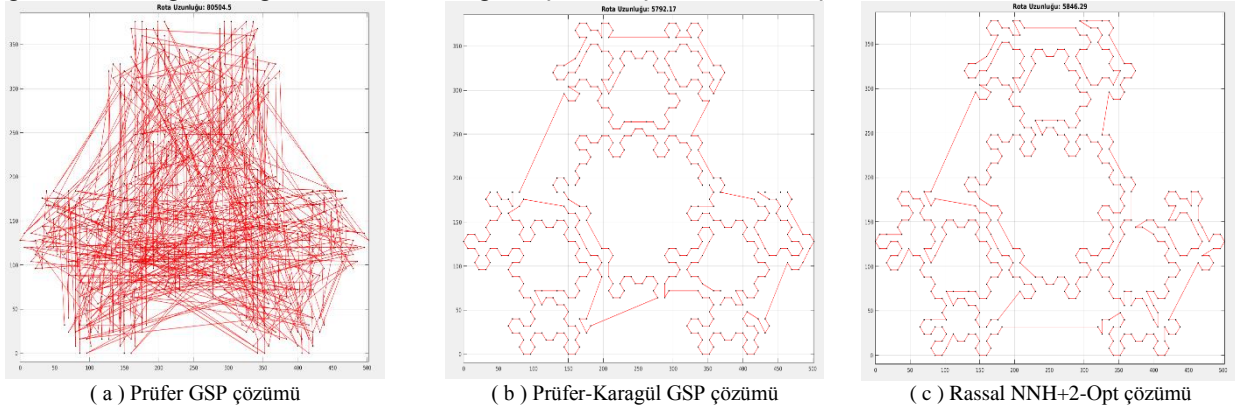
#OfRun	Prüfer-Karagül Yaklaşımı			Rassal NNH+2-Opt Yaklaşımı		
	Çözüm	Süre	Prüfer-Karagül GSP	Çözüm	Süre	Rassal NNH+2-Opt
1	133,23	0,0086	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000440	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
2	133,23	0,0087	[5,6,14,12,13,3,9,2,8,11,7,10,1,4]	133,23	0,000613	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]
3	133,23	0,0078	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000696	[5,7,6,14,12,13,3,9,2,8,11,10,1,4]
4	133,23	0,0053	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000258	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
5	133,23	0,0051	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000619	[13,3,9,2,8,11,10,1,4,5,7,6,14,12]
6	133,23	0,0064	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000237	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
7	133,23	0,0051	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000331	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
8	133,23	0,0048	[5,6,14,12,13,3,9,2,8,11,7,10,1,4]	133,40	0,000199	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
9	133,23	0,0050	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000622	[2,8,11,10,1,4,5,7,6,14,12,13,3,9]
10	133,23	0,0047	[2,8,11,7,10,1,4,5,6,14,12,13,3,9]	133,40	0,000206	[13,3,9,2,8,11,10,1,4,5,7,6,14,12]
Ortalama	133,23	0,0062		133,39	0,0004	

Şekil 1’de (a), (b) ve (c) grafiklerinde sırasıyla Prüfer GSP, Prüfer-Karagül GSP ve Rassal NNH+2-Opt çözümleri gösterilmektedir. 162 düğümlü bir GSP problemi için (a) grafiğinde yer alan çözümün en kötü çözüm olduğu çok açıktır. (b) ve (c) grafiklerinin karşılaştırması biraz yoruma ihtiyaç duyar. Çözüm değerleri bilinmese bile (b) çözümünün (c) çözümünden daha kaliteli bir çözüm olduğu alanda çalışanlar tarafından hemen algılanacaktır. Gerçekten de (a) çözümü 13396,5, (b) çözümü 1637,74 ve (c) çözümü 1754,32 iken problemin optimal çözümü 1620’dir.



Şekil 1: 162 düğümlü bir GSP probleminin 3 farklı yöntemle çözümleri
Kaynak: Problem Web-4 kaynağından alınmıştır ve hex162 olarak adlandırılmaktadır.

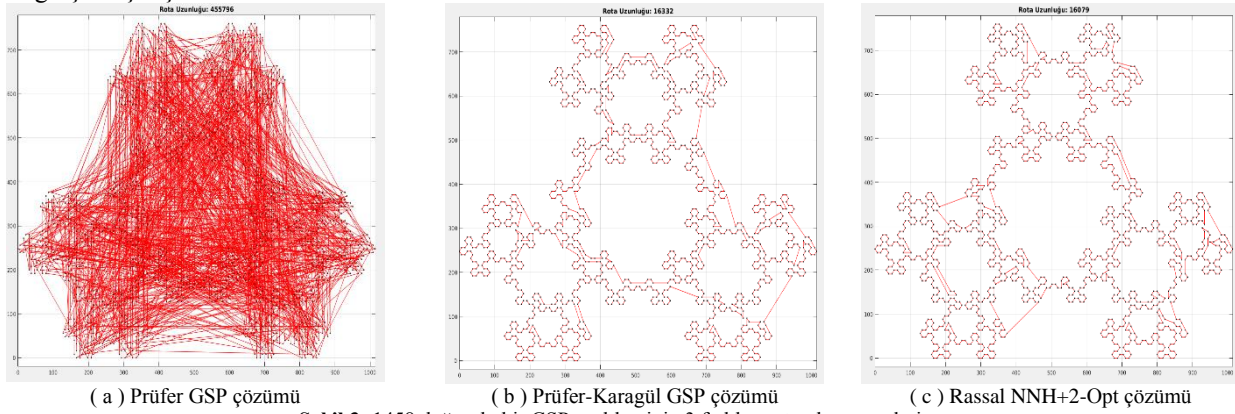
Benzer şekilde Şekil 1’te elde edilen başarılı çözümler 486 düğümlü GSP nin çözümlerinin gösterildiği Şekil 2’da da aynı şekilde görülmektedir. 486 düğümlü GSP probleminin optimal çözümü 4860’tır. (a) Prüfer GSP, (b) Prüfer-Karagül ve (c) Rassal NNH+2-Opt çözümleri sırasıyla 80504, 5792 ve 5846 olarak elde edilmiştir. Çözüm değerlerinden de görüleceği üzere Prüfer-Karagül başarılı bir çözüm elde etmiştir.



Şekil 2: 486 düğümlü bir GSP probleminin 3 farklı yöntemle çözümleri

Kaynak: Problem Web-4 kaynağından alınmıştır ve hex486 olarak adlandırılmaktadır.

Ancak Şekil 3’te 1458 düğümlü bir GSP için durum biraz farklı olmuştur. Sırasıyla (a) çözümü 455796, (b) çözümü 16332 ve (c) çözümü 16079 olmak üzere optimal çözüm olan 14580 değerine yakınlıkları verilen sıranın tam tersi gerçekleşmiştir.



Şekil 3: 1458 düğümlü bir GSP probleminin 3 farklı yöntemle çözümleri

Kaynak: Problem Web-4 kaynağından alınmıştır ve hex1458 olarak adlandırılmaktadır.