



Journal of Turkish Operations Management

Otel seçimli gezgin satıcı problemi için değişken komşu iniş sezgiseli

İpek Damla AKPINAR^{a*}, Barış KEÇECİ^b

^aEndüstri Mühendisliği Bölümü, Başkent Üniversitesi, Ankara, Türkiye
e-mail: idakpinar@baskent.edu.tr, ORCID No: <https://orcid.org/0000-0001-8977-9521>

^bEndüstri Mühendisliği Bölümü, Başkent Üniversitesi, Ankara, Türkiye
e-mail: bkececi@baskent.edu.tr, ORCID No: <https://orcid.org/0000-0002-2730-5993>

*Sorumlu yazar

Makale Girişi

Makale Geçmişi:

Geliş: 10.09.2019
Revize: 10.12.2019
Kabul: 30.03.2020

Anahtar Kelimeler:

Gezgin satıcı problemi,
Otel seçimi,
Sezgisel algoritmalar,
Değişken komşu arama

Özet

Otel Seçimli Gezgin Satıcı Problemi (OSGSP), Gezgin Satıcı Probleminin (GSP) bir çeşididir. OSGSP'nde çalışma süresi/mesafesi kısıtından dolayı tüm noktalar tek bir günde ziyaret edilememektedir. Bu problem NP-zor problem sınıfına girdiği için sezgisel yöntem kullanılması çözüm zamanı açısından avantaj sağlamıştır. İlk olarak En Yakın Komşu Prensi (EYKP) kullanılarak başlangıç çözüm elde edilmiştir. Sonrasında bulunan bu çözümü iyileştirmek amacıyla Değişken Komşu İniş Algoritması (DKİA)'ndan faydalanılmıştır. Algoritmalar Python dili ile kodlanmıştır. Literatürde var olan test problemleri üzerinde çeşitli ölçütler kullanılarak denemeler yapılmıştır ve elde edilen sonuçlar var olan en iyi çözümler ile karşılaştırılarak algoritmanın performansı değerlendirilmiştir.

Article Info

Article History:

Received: 10.09.2019
Revised: 10.12.2019
Accepted: 30.03.2020

Keywords:

Travelling salesperson problem,
Hotel selection,
Heuristic algorithms,
Variable neighborhood descent

Abstract

The Travelling Salesperson Problem with Hotel Selection (TSPHS) is a variant of The Travelling Salesperson Problem (TSP). In case of TSPHS problem, all points cannot be visited in a single day due to the restriction of working time / distance. Since this problem belongs to the NP-hard problem class, the use of heuristic method has an advantage in terms of the solution time. The initial solution is obtained using The Nearest Neighbor Algorithm (NN). In order to improve this solution, the Variable Neighborhood Descent (VND) is used. Python language is used for coding. The algorithms experiments are made on the test problems in the literature using various criteria and the results obtained are compared with the best available solutions and the performance of algorithm is evaluated.

1. Giriş

Araştırmacılar, ihtiyaçlar dahilinde zaman ve/veya maliyet açısından avantaj sağlayabilmek için bir çok alanda çalışma yapmaktadır. Bu alanlardan biri de rotalama problemleridir. Rotalama problemlerin temelinde ise Gezgin Satıcı Problemi (GSP) yer almaktadır. Bu problemin çözümünde zaman/maliyet açısından avantaj elde edebilmek için GSP alanında çalışılmıştır. GSP, aralarındaki uzaklıkları bilinen noktaların her birinden yalnız bir kez geçerek, başladığı noktaya dönen en az maliyetli turun bulunması problemidir. Otel Seçimli Gezgin Satıcı Problemi (OSGSP) ise çalışma süresi/mesafesi kısıtını içermektedir ve bu haliyle GSP'nin genişletilmiş bir varyantını oluşturmaktadır.

OSGSP ilk kez Vansteenwegen, Souffriau ve Sörensen tarafından 2011 yılında ele alınmıştır. Problem çözümü için matematiksel bir model oluşturulmuştur. Fakat bu modelin çözüm zamanı açısından elverişli olmadığı görülmüştür ve çözüm süresini azaltmak için sezgisel bir algoritma tasarlanmıştır. Sezgisel yöntemin çözüm kalitesini iyileştirmek adına komşuluk mekanizmalarının da kullanıldığı Variable Neighborhood Descent (VND) algoritması geliştirilmiştir (Castro, Sörensen, Vansteenwegen ve Goos, 2012). Matematiksel model ile elde edilen optimum sonuçlardan daha az sapma ile sonuç elde edebilmek için memetik algoritma içeren bir sezgisel algoritma önerilmiştir (Castro ve diğ., 2013). Fakat bu algoritmanın çözüm zamanı açısından elverişli olmadığı görülmüş, bu nedenle hem çözüm zamanı hem de çözüm kalitesi açısından etkili olan Perturbation Local Search (P-LS) sezgiseli önerilmiştir (Castro ve diğ., 2014). Yapılan bir diğer çalışmada OSGSP tropical matematiğin çerçevesinde ele alınmış ve en iyi sonuçlara ulaşılması hedeflenmiştir (Radmanesh, Kumar, Nematı ve Sarim, 2016). Yapılan karşılaştırmalar sonucunda tropical matematik ile en iyi sonuçlara daha kısa sürede ulaşıldığı gözlemlenmiştir. Lu, Benlic ve Wu tarafından 2017 yılında dinamik programlama ve memetik algoritmanın birlikte kullanıldığı Hybrid Between Dynamic Programming and Memetic Search (HDM) algoritması geliştirilmiştir. 2019 yılında literatürde var olan matematiksel modeller dışında Gavish and Graves Alt Tur Eleme Kısıtları İle Vansteenwegen, Souffriau, Sörensen Modeli (VSSGG); Gavish and Graves Alt Tur Eleme Kısıtları İle Castro, Sörensen, Vansteenwegen, Goos Modeli (CSVGGG) ve The Fox, Gavish and Graves Modeli (FGG) olmak üzere üç yeni model önerilmiş ve literatürdeki problemler üzerinden bu modellerin performans değerlendirmeleri yapılmıştır (Gencel, 2019a). Son olarak ise Sousa ve ark. tarafından yinelemeli yerel arama sezgisellerini temel alan bir yaklaşım geliştirilmiştir (Sousa, Ochi ve Martins, 2019).

Gerçek yaşamda OSGSP olarak modellenebilen bazı problemlere örnek vermek gerekirse, birden fazla günden oluşan turist gezi programları; ağır vasıta sürücülerinin birden fazla günden oluşan, ve her gün sonunda uygun bir dinlenme noktasında durakladığı tur problemi; bir postacının taşıyacağı yükü hafifletmek amacıyla turunu birbiriyle bağlantılı daha küçük turlara böldüğü dağıtım problemi; maksimum kullanım zamanı pil kapasitesiyle sınırlı elektrikli bir aracın tur problemi.

Çalışmanın ikinci bölümünde otel seçimli gezgin satıcı probleminden bahsedilmiş, üçüncü bölümünde çözüm yaklaşımı başlığı altında en yakın komşu prensibi, komşuluk mekanizmaları ve otel iyileştirilmesi ile değişken komşu iniş sezgiseli anlatılmıştır. Dördüncü bölümde ise deneysel çalışmalara değinilmiştir. Son olarak ise beşinci bölümde çalışmadan elde edilen sonuçlara yer verilmiş ve karşılaştırmalı analizler yapılmıştır.

2. Otel Seçimli Gezgin Satıcı Problemi

OSGSP'de s adet otelden oluşan otel kümesi Eşitlik (1) ile n adet müşteriden oluşan müşteri kümesi Eşitlik (2) ile ifade edilmektedir. Her i müşterisinden/otelinden j müşterisi/oteline gidiş süresi/mesafesi $c_{(i,j)}$ ile, her i müşterisi için hizmet süresi ise T_i ile ifade edilmektedir. Gezginin her gün bir otelde/başlangıç noktasında başlayıp bir otelde/bitiş noktasında sonlanan ziyaretleri "gezi" olarak adlandırılmakta, tüm gezilerin toplamı ise "tur" olarak ifade edilmektedir. Eşitlik (3)'te gösterildiği gibi her tur toplamda d tane olan m adet geziden oluşmaktadır (Vansteenwegen ve diğ., 2011).

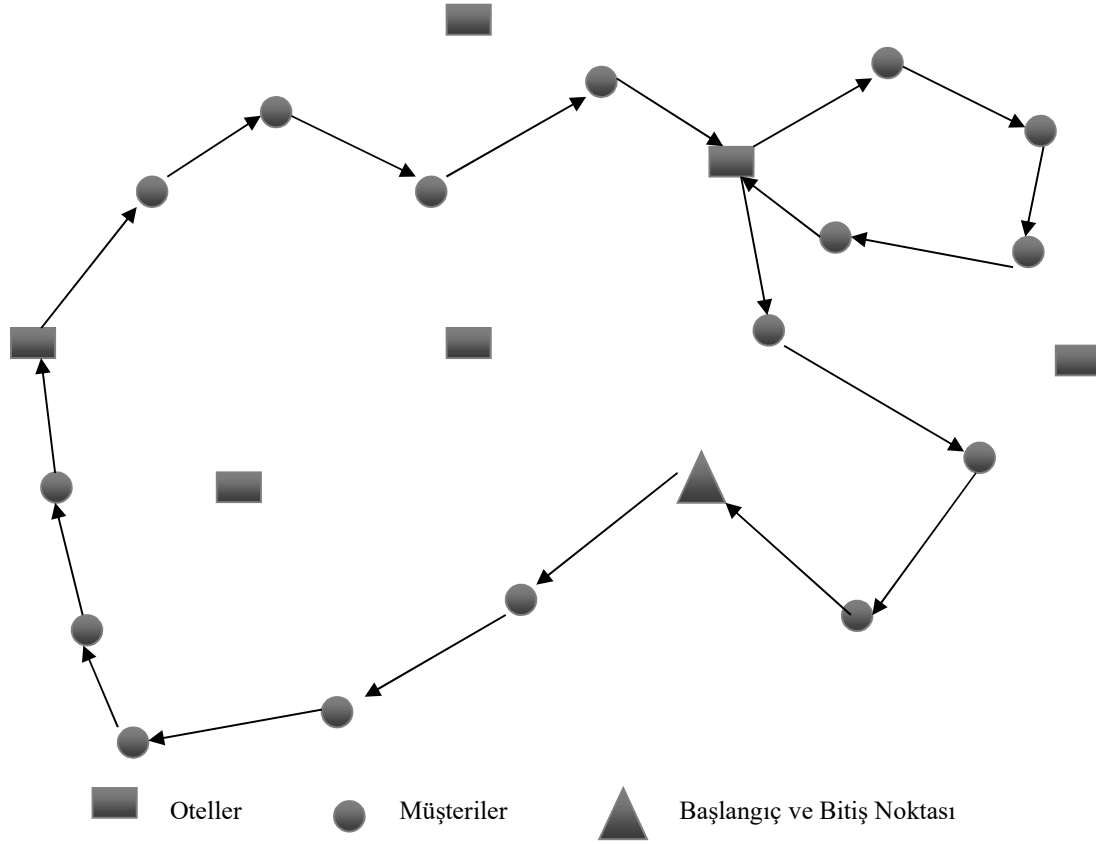
$$i=1,\dots,s \quad (1)$$

$$i=s+1,\dots,s+n \quad (2)$$

$$d=1,\dots,m \quad (3)$$

Bu problem için iki tür amaç tanımlanmıştır. Bunlardan ilki toplam turdaki gezi sayısını en küçükleme. İkincisi ise gezginin gerçekleştirdiği toplam tur uzunluğunu en küçükleme. Bu amaçlardan birinci amaç ikinci amaca göre daha önceliklidir. Bunun nedeni her bir fazla gezinin maliyetinin, turun daha uzun olması maliyetinden daha çok olmasıdır.

Çünkü gezginin fazladan yapacağı her bir gezinin; fazladan otel masrafı, gezginin günlük ücreti vb. masrafların yol ücreti masrafından daha büyük olmasıdır.



Şekil 1. OSGSP çözümü örneği

Klasik GSP'de olduğu gibi OSGSP'de de her müşteriye uğrama zorunluluğu vardır. Gezgin her noktaya uğradığında yolculuk süresinden ayrı olarak, o noktada hizmet vermek için belli bir süre harcamaktadır. Bu sürede günlük çalışma süresine dahil edilmektedir.

OSGSP'de çalışma süresi/mesafesi kısıtından dolayı tüm noktalar tek bir günde ziyaret edilememektedir. Bu süre sınırlaması C ile ifade edilmektedir. Bu nedenle gezgin, çalışma süresi/mesafesi kısıtını aşmadan gün sonunda bir bekleme noktasında (otelde) duraklamak (konaklamak) durumundadır. Gezginin seyahati süresince konaklayabileceği otellerin listesi önceden belirlenmiştir. Gezgin, gün sonunda en son noktayı ziyaret ettikten sonra konaklayacağı oteli seçerken oteller listesinde bulunduğu noktaya en yakın oteli tercih etmektedir. Ertesi gün konakladığı otelden ziyaret etmesi gereken noktaya doğru yola çıkarak seyahatini sürdürmektedir.

Gezginin yolculuğunda tüm noktalara uğrama kısıtı bulunmasına karşın, belirlenen oteller listesindeki her otele uğrama zorunluluğu yoktur. Hatta gerekli görülmesi durumunda gezgin, bir otelde birden fazla kere konaklayabilmektedir. Ayrıca turun aynı otelde/başlangıç noktasında başlayıp, aynı otelde/bitiş noktasında bitme zorunluluğu vardır. Örnek bir OSGSP Şekil 1'de gösterilmiştir.

OSGSP'nin zaman pencereli OSGSP ve otel seçimli oryantiring problemi gibi çeşitleri vardır. Zaman pencereli OSGSP'de, OSGSP'ye ek olarak her noktanın ziyaret edilmesi için belli bir zaman aralığı vardır ve gezginin o zaman aralığında o noktayı ziyaret etmiş olması gerekmektedir. Baltz, Ouali, Jager, Sauerland ve Srivastav tarafından 2014 yılında bu problem ele alınmıştır. Problem çözümü için matematiksel bir model kurulmuş ve sezgisel bir algoritma geliştirilmiştir. Otel seçimli oryantiring probleminde ise tüm noktaların ziyaret edilme zorunluluğu yoktur. Ziyaret edilecek noktalar toplam fayda en büyüklenecek şekilde belirlenir. Bu problem ilk olarak Divsalar, Vansteenwegen,

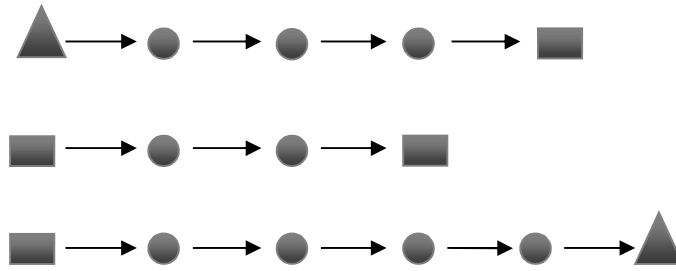
Sörensen ve Cattrysse tarafından 2014 yılında çalışılmıştır ve memetik bir algoritma geliştirilmiştir. 2015 yılında hipersezgisel bir algoritma önerilmiştir (Toledo ve Riff, 2015). Problem çözümünde çözüm kalitesi açısından iki sezgisel yöntem karşılaştırılmıştır (Hoek, 2016). Sonuç olarak Tabu Arama algoritmasının çözüm zamanı ve kalitesi açısından daha iyi sonuç verdiği gözlemlenmiştir. Yapılan diğer bir çalışmada literatürde var olan modeller dışında iki tane daha model önerilmiş ve bu modeller performans kriterlerine göre değerlendirilmiştir (Gencel, 2019b).

3. Çözüm Yaklaşımı

Literatürde OSGSP'yi çözmek için birçok yöntem önerilmiştir. Bunlar içinde lineer programlama, dinamik programlama, dal-sınır algoritması gibi kesin çözüm veren yöntemlerin yanı sıra komşuluk mekanizmaları, tabu arama algoritması, genetik algoritmalar melez yöntemler gibi sezgisel yaklaşımlarda önerilmiştir.

Problem çözümünde kullanılan matematiksel modeller kesin çözümler elde etmesine rağmen, çözüm zamanı açısından avantajlı bulunmamıştır. Bu nedenle, özellikle büyük boyutlu problemlerde çözüm yöntemi olarak daha kısa sürede sonuç veren sezgisel yöntemler önerilmiştir. Bu çalışmada da OSGSP çözüm yöntemi olarak sezgisel bir yaklaşım geliştirilmiştir.

Önerilen sezgisel algoritma da permütasyon çözüm gösterimi benimsenmiştir. Bu gösterime göre her bir satır bir geziyi ifade etmektedir. Tüm satırların toplamı ise turu oluşturmaktadır. Yani bir turda satır sayısı kadar gezi sayısı vardır. Her bir satırın sonundaki otel ile ardışık satırın başındaki otel aynı oteli temsil etmektedir. Turun başlangıç ve bitiş noktası da aynı noktayı ifade etmektedir. OSGSP için küçük boyutlu çözüm gösterim örneğine Şekil 2'de yer verilmiştir.



Şekil 2. OSGSP için çözüm gösterimi örneği

Bu algoritma da çözüme ulaşılırken, ilk olarak En Yakın Komşu Prensibi (EYKP) kullanılarak başlangıç çözüm elde edilmiştir. Sonrasında bu çözümü iyileştirmek amacıyla Değişken Komşu İniş Algoritması (DKİA) kullanılmıştır. Bu algoritma uygulanırken eldeki çözümü iyileştirmek amacıyla "takas" ve "ekleme" komşuluk mekanizmaları ile "otel iyileştirme" hareketi kullanılmıştır. "Gezi azalt" hareketi ile de turdaki gezilerin azaltılması hedeflenmiştir. Süre/mesafe açısından sonuçlar incelendiğinde daha iyi bir çözüme ulaşılamadığı noktada algoritma sonlandırılmıştır. Çalışmada kullanılan bu yöntem tasarlanırken Araştırma ve Yayın Etiğine uyulmuştur.

3.1 En Yakın Komşu Prensibi

Çalışmanın başlangıç aşamasında, çözüm elde etmek için öncelikli olarak bir başlangıç çözüme ihtiyaç duyulmuştur. Bu başlangıç çözümü elde edebilmek için En Yakın Komşu Prensibi kullanılmıştır. Bu prensibin temelinde, gidilecek noktaya karar verilirken mevcut konuma en yakın olan noktanın tercih edilmesi felsefesi vardır. Buna göre gezgin başlangıç noktasından gezisine başlamakta, sonrasında gideceği noktayı belirlerken kendine en yakın olanı seçmektedir. Bu şekilde günlük süre/mesafe kısıtı da aşılmadan gezgin, son noktayı da ziyaret ettikten sonra konaklayacağı otele karar verirken de aynı prensipten yararlanmaktadır. Bu şekilde başlangıç çözümündeki her bir gezi tamamlanmakta ve tur oluşturulmaktadır.

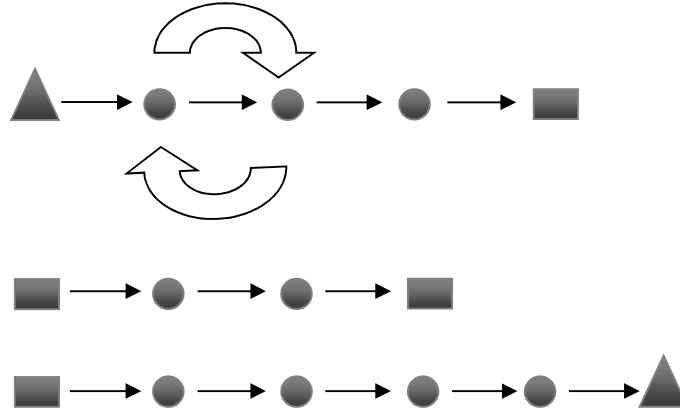
3.2 Komşuluk Mekanizmaları, Otel İyileştirilmesi ve Gezi Azalt

3.2.1 Komşuluk Mekanizmaları

Komşuluk mekanizmaları, oluşturulan turun toplam süresini/mesafesini azaltmak için kullanılan yöntemlerdir. Bu yöntemler sayesinde gezgin, daha kısa sürede/mesafede tüm noktaları ziyaret ederek başlangıç noktasına dönmektedir.

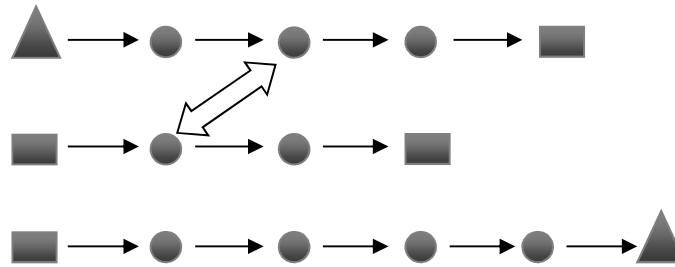
Çözüm yaklaşımında "takas" ve "ekleme" olmak üzere iki tür komşuluk mekanizması kullanılmıştır (Lu ve diğ., 2017). Kullanılan her iki komşuluk mekanizması da hem gezi içerisinde hem de geziler arasında noktaların sıralarını ve yerlerini değiştirerek iyileştirmeler yapmaktadır. Bu iyileştirmeler yapılırken günlük süre/mesafe kısıtı da göz ardı edilmemelidir.

Takas komşuluk mekanizmasının temelinde, çözüme fayda sağlayacak biçimde iki noktanın birbiri ile değişimi esas alınmaktadır. Yani bu komşuluk mekanizmasında, çözümde var olan iki noktanın çözüme fayda sağlamak için yerlerinin birbiri ile değiştirilmesi söz konusudur. Ekleme komşuluk mekanizmasında ise, fayda sağlamak için çözümde var olan bir noktanın yeri değiştirilir. Aslında bu komşuluk mekanizmasında seçilen noktanın taşınması söz konusudur. Yani belirlenen nokta mevcut konumundan alınarak başka bir konumdaki iki noktanın arasına yerleştirilir. Böylelikle başlangıçta bulunan çözümden daha iyi bir kalitede çözüm elde edilmiş olunur.



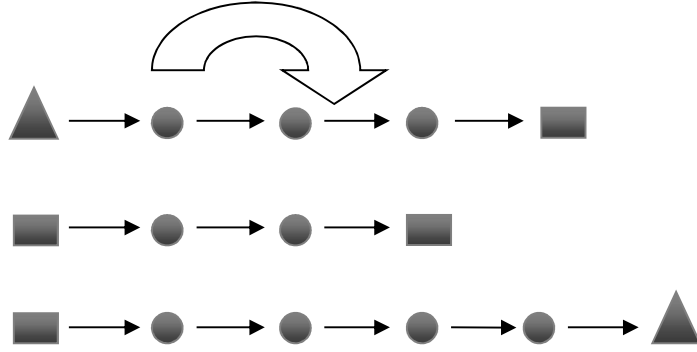
Şekil 3. Gezi içi takas komşuluk mekanizması örneği

Şekil 3'te birinci gezideki birinci ve ikinci noktaların sıralamasının takas komşuluk mekanizmasına göre değişimi gösterilmiştir. Şekil 4'te ise birinci gezideki ikinci nokta ile ikinci gezideki birinci nokta değiştirilerek, takas komşuluk mekanizmasının geziler arası uygulamasına yer verilmiştir.

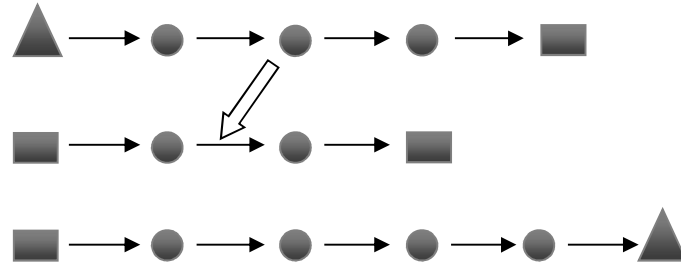


Şekil 4. Geziler arası takas komşuluk mekanizması örneği

Şekil 5'te ekleme komşuluk mekanizması ile birinci gezideki ilk nokta, aynı gezideki ikinci ve üçüncü nokta arasına taşınmıştır. Bu komşuluk mekanizmasının geziler arası örneği ise Şekil 6'da birinci gezideki ikinci nokta, ikinci gezide birinci ve ikinci nokta arasına taşınarak gösterilmiştir.



Şekil 5. Gezi içi ekleme komşuluk mekanizması örneği



Şekil 6. Geziler arası ekleme komşuluk mekanizması örneği

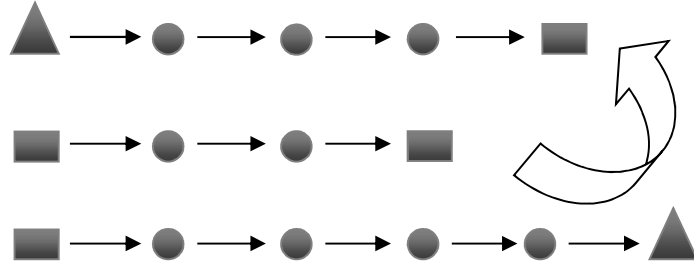
3.2.3 Otel İyileştirilmesi

Yapılan çalışmada, gezinin gideceği noktalar en iyi seviyede belirlendikten sonra otellerin de iyileştirilmesi gerekliliği ortaya çıkmıştır. Bu uygulama da "otel değiştir" mekanizması kullanılmıştır (Castro ve diğ., 2012). Bu amaçla yapılan çalışmada, önceden belirlenen gezinin konaklayabileceği oteller listesinden faydalanılmıştır. Mevcut çözümde var olan oteller, listede var olan her bir otelle değiştirilerek çözümde bir iyileşme olup olmadığına bakılır. Eğer bir iyileşme olur ise otel değişimi yapılır ve mevcut çözüm güncellenir. Her otel birden fazla kere tur içerisinde kullanılabilmesi için bu mekanizma uygulanırken oteller listesinde herhangi bir güncelleme yapılmamıştır. Bu sayede oteller listesindeki oteller de, en verimli şekilde değerlendirilmiş olur. Bu hareket sırasında bağlantılı gezilerin ilk ve son otellerinin aynı olmasına dikkat edilmelidir.

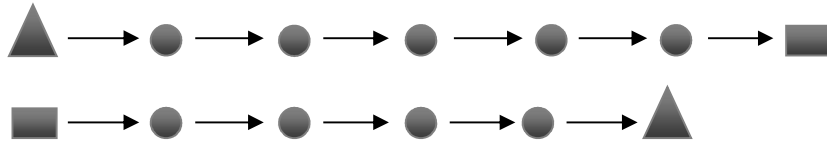
3.2.4 Gezi Azalt

Toplam tur mesafesinin azaltılabilmesi için turdaki gezilerin sayısı önem taşımaktadır. Bu nedenle toplam turdaki gezi sayısı ne kadar az olur ise toplam tur uzunluğu da o kadar kısa olur. Gezi azalt mekanizması, tur içerisindeki gezi sayısının en iyi seviyede olmasını sağlar (Castro ve diğ., 2012). Bunu yaparken oluşturulan çözümdeki her bir gezinin arasındaki oteli kaldırarak, bu gezileri birbirinin ardına ekler. Bu ekleme işleminden sonra günlük süre/mesafe kısıtının sağlanıp sağlanmadığı kontrol edilir. Eğer bu kısıt sağlanıyor ise ve toplam tur uzunluğu bir önceki çözüme göre daha avantajlı bir hale geliyor ise gezi azaltma işlemi yapılır. Böylelikle OSGSP probleminin diğer bir amacı da sağlanmış olur.

Şekil 7'de küçük boyutlu bir tur örneği gösterilmiştir. Bu tur içerisinde, yukarıda bahsedildiği gibi her bir gezi birbiri ardına eklenerek günlük süre/mesafe kısıtının aşılmadığı kontrol edilmiştir. Yapılan kontroller sonucunda, birinci ve ikinci gezi birleştirilerek turdaki gezi sayısının üçten ikiye düşürülmesine karar verilmiştir. Bunun için birinci gezi sonundaki otel kaldırılmış, ikinci gezideki noktalar birinci gezideki noktaların ardına eklenmiş ve şekil 8 elde edilmiştir. Böylelikle turdaki gezi sayısı azaltılmıştır.



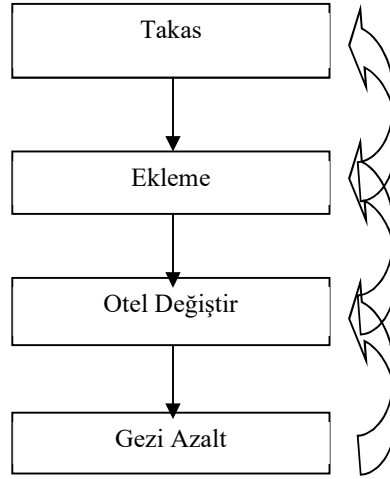
Şekil 7. Gezi azalt hareketi öncesi örneği



Şekil 8. Gezi azalt hareketi sonrası örneği

3.3 Değişken Komşu İniş Sezgiseli

Değişken Komşu İniş (DKİ) Sezgiseli, Hansen ve Mladenovic tarafından 2003 yılında önerilmiştir. Bu yaklaşım birçok kombinasyonel optimizasyon probleminde uygulanmaktadır ve nispeten yeni bir yöntemdir.



Şekil 9. Değişken komşu iniş sezgiseli örneği

Bu çalışma için DKİ uygulama aşamaları Şekil 9'da gösterilmiştir. En Yakın Komşu Prensipli ile elde edilen başlangıç çözüme ilk olarak Takas komşuluk mekanizması uygulanır. Bu mekanizma ile ulaşılabilecek en iyi çözüm bulunduğundan sonra, bir sonraki komşuluk mekanizması olan Ekleme komşuluk mekanizmasına geçilir. Ekleme komşuluk mekanizmasında daha iyi bir çözüm bulur ise bir önceki adıma yani Takas komşuluk mekanizmasına dönülür ve bulunan en iyi çözüme Takas komşuluk mekanizması, daha iyi bir sonuç bulunamayana kadar tekrar tekrar uygulanır. Eğer Takas ve Ekleme yöntemleri ile daha iyi bir çözüm bulunamaz ise bir sonraki adıma geçilir ve bulunan en iyi çözüme Otel Değiştir yöntemi uygulanır. Eğer daha iyi bir çözüm bulur ise yine ilk adıma dönülür. Algoritma bu

şekilde ta ki Gezi Azalt yönteminde daha iyi bir çözüm bulunamayana kadar sürdürülür. Eğer son adımda daha iyi bir çözüm bulunamaz ise bulunan çözüm en iyi çözüm olarak kaydedilir.

4. Deneysel Çalışmalar

4.1 Test Verileri

Bu çalışmada daha önce Vansteenwegen ve diğ. (2011) tarafından oluşturulan test verileri kullanılmıştır. Veri setlerine <https://antor.uantwerpen.be/instances-in-the-paper-a-memetic-algorithm-for-the-travelling-salesperson-problem-with-hotel-selection/> adresi üzerinden ulaşılmıştır. Bu veri setleri internet adresinde erişime açık olup kullanım için herhangi bir izin gerekmemektedir. Veri setleri içerisinde Set-1 ve Set-2 verileri ile sezgisel algoritma test edilmiştir. Set-1 veri seti 48-288 müşteri ve 6 otel içeren daha büyük ölçekli bir veri seti olmakla birlikte Set-2 10, 15, 30 ve 40 müşteri ile 2 otel içeren daha küçük ölçekli bir veri setidir.

4.2 Bilgisayar Özellikleri

Algoritma çözümlenirken i7 işlemci 3.5 GHZ, 8GB ram ve Linux işletim sistemine sahip bir bilgisayar kullanılmıştır. Algoritma kodlanırken dizi işlemlerini yapma kolaylığı sağladığı için kodlama dili olarak Python dili kullanılmıştır. Bu kodlama zamanı açısından avantaj sağlamıştır.

5. Sonuç

Değişken komşu iniş sezgiseli iki veri seti içinde uygulanmıştır. Elde edilen karşılaştırmalı sonuçlar Tablo 2, Tablo 3, Tablo 4, Tablo 5 ve Tablo 6'da gösterilmiştir.

Tablo 1'de ilk veri setinden elde edilen sonuçlar listelenmiştir. Bu veri seti büyük ölçekli problemler için tasarlanmıştır. Set 1 içerisinde coğrafi koordinat verilerinden oluşan 16 farklı problem vardır. Her bir problemin ismi, müşteri sayısı, zaman bütçesi, seyahat sayısı ilk dört sütunda listelenmiştir. Sonraki sütunda problem için oluşturulan matematiksel modelin Cplex programı ile çözümü yer almaktadır. Bu çözüm, problemin elde edilen en iyi çözümünü göstermektedir. VND çözümü ise, bu çalışmada tasarlanan algoritma ile elde edilen çözümü ifade etmektedir. Son sütunda Cplex çözümden, yani en iyi çözümden elde edilen sapma miktarı listelenmiştir. Bu sapma miktarı Eşitlik (3) ile yüzde cinsinden gösterilmiştir. Bu formül kullanılarak her bir problem için sapma miktarları hesaplanmıştır. Bu veri seti büyük ölçekli bir veri seti olduğu için elde edilen sapma miktarları da daha büyüktür.

Tablo 1. Set 1 için karşılaştırmalı sonuçlar

İsim	Müşteri Sayısı	Zaman Bütçesi	Seyahat Sayısı	CPLEX Çözüm	VND Çözüm	Sapma(%)
c101	100	1236	9	9685,6	9773,5	0,91
c201	100	3390	3	9600,0	9635,0	0,36
pr01	48	1000	2	1416,6	1544,6	9,04
pr02	96	1000	3	2569,3	2826,1	9,99
pr03	144	1000	4	3584,1	3970,6	10,78
pr04	192	1000	5	4366,3	4995,7	14,41
pr05	240	1000	6	5122,1	5554,8	8,45
pr06	288	1000	7	6137,3	6674,4	8,75
pr07	72	1000	3	2085,2	2380,4	14,16
pr08	144	1000	4	3504,7	4005,2	14,28
pr09	216	1000	5	4617,6	5046,1	9,28
pr10	288	1000	7	6097,5	6687,4	9,67
r101	100	230	9	1801,3	1926,5	78,17
r201	100	1000	2	1670,4	1694,4	1,44
rc101	100	240	8	1724,1	2059,2	19,44
rc201	100	960	2	1661,3	1723,1	3,72
					Ort,	13,3

$$[(\text{VND çözüm} - \text{Cplex Çözüm}) / \text{Cplex çözüm}] * 100 \quad (3)$$

Tablo 2, Tablo 3, Tablo 4 ve Tablo 5 ikinci veri seti için farklı müşteri sayılarına sahip problemlerin karşılaştırmalı sonuçlarını göstermektedir. Bu veri seti oluşturulurken set 1 veri seti içerisinde küçük kümelerde veri setleri seçilmiştir. Her bir tabloda seçilen veri kümesi için elde edilen Cplex çözüm, VND çözüm ve karşılaştırılması yapılmıştır.

Tablo 2. Set 2 için karşılaştırmalı sonuçlar (10 müşteri)

İsim	Müşteri Sayısı	Zaman Bütçesi	Seyahat Sayısı	CPLEX Çözüm	VND Çözüm	Sapma (%)
c101	10	1236	1	955,1	959,5	0,46
r101	10	230	2	272,8	286,2	4,91
rc101	10	240	1	237,5	243,7	2,61
pr01	10	1000	1	426,6	434,3	1,80
pr02	10	1000	1	661,9	661,9	0,00
pr03	10	1000	1	553,3	559,1	1,05
pr04	10	1000	1	476,4	476,4	0,00
pr05	10	1000	1	528,9	528,9	0,00
pr06	10	1000	1	597,4	615,9	3,10
pr07	10	1000	1	670,2	702,6	4,83
pr08	10	1000	1	573,4	580,9	1,31
pr09	10	1000	1	645,5	645,5	0,00
pr10	10	1000	1	461,5	461,5	0,00
					Ort,	1,54

Tablo 2 de 5 problem için; VND sezgiseli ile bulunan sonuçlar, Cplex çözümü ile bulunan sonuçlarla aynıdır. Yani bu problemlerde optimal sonuçlar sezgisel yöntemle de yakalanmıştır. Diğer problemlerde ise sezgisel yöntemle bulunan sonuçların, optimal çözümden sapmalarının ortalaması %1,54 olarak hesaplanmıştır.

Tablo 3 de 15 müşterili set 2 verileri için karşılaştırmalı sonuçlar listelenmiştir. Sapma değerleri incelendiğinde VND çözümlerin, 3 problem için Cplex çözüm ile aynı olduğu görülmüştür. Diğer problemlerde ise sezgisel yöntemle bulunan sonuçların optimal çözümden sapmalarının ortalaması %6,15 olarak hesaplanmıştır.

Tablo 3. Set 2 için karşılaştırmalı sonuçlar (15 müşteri)

İsim	Müşteri Sayısı	Zaman Bütçesi	Seyahat Sayısı	CPLEX Çözüm	VND Çözüm	Sapma (%)
c101	15	1236	2	1452,8	1466,2	0,92
r101	15	230	2	379,8	429,3	13,03
rc101	15	240	2	303,2	377,4	24,47
pr01	15	1000	1	590,4	635,6	7,66
pr02	15	1000	1	745,6	790,0	5,95
pr03	15	1000	1	632,9	698,9	10,43
pr04	15	1000	1	683,4	733,0	7,26
pr05	15	1000	1	621,2	674,1	8,52
pr06	15	1000	1	685,2	685,2	0,00
pr07	15	1000	1	795,3	802,9	0,96
pr08	15	1000	1	707,2	712,7	0,78
pr09	15	1000	1	771,7	771,7	0,00
pr10	15	1000	1	611,9	611,9	0,00
					Ort,	6,15

Tablo 4 ve tablo 5 de listelenen sonuçlar incelendiğinde 30 ve 40 müşterili problemlerde elde edilen sonuçlarda optimal çözümlerin yakalanamadığı, fakat optimal çözümlere nispeten yaklaşıldığı gözlemlenmiştir. Veri seti 2 de 30 ve 40 müşteri için, VND çözüm ile bulunan sonuçların Cplex çözümlerden sapmaları sırasıyla %10.37 ve %11.37 olarak hesaplanmıştır.

Tablo 4. Set 2 için karşılaştırmalı sonuçlar (30 müşteri)

İsim	Müşteri Sayısı	Zaman Bütçesi	Seyahat Sayısı	CPLEX Çözüm	VND Çözüm	Sapma (%)
c101	30	1236	3	2876,3	2880,3	0,1
r101	30	230	3	676,2	722,1	6,78
rc101	30	240	4	712,4	743,0	4,29
pr01	30	1000	1	964,8	1083,2	12,2
pr02	30	1000	2	1078,3	1168,5	8,3
pr03	30	1000	1	952,5	967,5	1,5
pr04	30	1000	2	1091,6	1245,1	14,0
pr05	30	1000	1	924,7	1107,0	19,7
pr06	30	1000	2	1065,3	1161,5	9,03
pr07	30	1000	2	1130,4	1306,4	15,57
pr08	30	1000	2	968,4	1082,3	11,7
pr09	30	1000	2	1091,4	1247,0	14,26
pr10	30	1000	1	918,9	1078,2	17,34
					Ort,	10,37

Tablo 5. Set 2 için karşılaştırmalı sonuçlar (40 müşteri)

İsim	Müşteri Sayısı	Zaman Bütçesi	Seyahat Sayısı	CPLEX Çözüm	VND Çözüm	Sapma (%)
c101	40	1236	4	3866,1	3910,7	1,15
r101	40	230	4	895,5	943,7	5,38
rc101	40	240	4	851,2	1016,3	19,39
pr01	40	1000	2	1160,5	1376,0	18,57
pr02	40	1000	2	1336,9	1449,5	8,42
pr03	40	1000	2	1303,4	1417,1	8,72
pr04	40	1000	2	1259,5	1446,8	14,87
pr05	40	1000	2	1200,7	1337,6	11,40
pr06	40	1000	2	1271,5	1456,2	14,52
pr07	40	1000	2	1426,5	1557,3	9,16
pr08	40	1000	2	1305,9	1390,3	6,46
pr09	40	1000	2	1284,4	1463,3	13,93
pr10	40	1000	2	1233,6	1428,8	15,82
					Ort,	11,37

Tablolar incelendiğinde problem boyutu küçükken VND yöntemiyle elde edilen sonuçlarda optimal sonuca yaklaşıldığı, hatta kimi zaman optimal sonucun yakalandığı gözlemlenmiştir. Müşteri sayısı arttıkça ise sezgisel algoritma çözümü ile optimal sonuca ulaşmak zorlaşmıştır.

İki veri seti içinde VND algoritması ile elde edilen sonuçlar üzerinden genel bir değerlendirme yapmak gerekirse Set 1 için ortalama 0.13, Set 2 için ise ortalama 0.07 optimum çözümden sapma ile sonuçlara ulaşılmıştır.

İleriki çalışmalarda iki veri seti içinde elde edilen sezgisel algoritma sapma değerlerinin iyileştirilmesinin yanı sıra, VND algoritması geliştirilerek literatürde var olan diğer büyük boyutlu veri setleri içinde çözümler elde edilebilir. Ayrıca OSGSP'nin geliştirilmiş bir türü olan birden fazla gezgin içeren çoklu OSGSP için çözüm yöntemleri geliştirilebilir.

Araştırmacıların Katkısı

Bu araştırmada; İpek Damla Akpınar, bilimsel yayın araştırması, makalenin oluşturulması, bilgisayar ortamına aktarılması, bilgisayar kodlarının oluşturulması; Barış Keçeci, bilimsel yayın araştırması, uygulanması, bilgisayar kodlarının denenmesi ve analizi; konularında katkı sağlamışlardır.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Baltz, A., Ouali, M. E., Jager, G., Sauerland, V. & Srivastav, A. (2014). Exact and heuristic algorithms for the travelling salesman problem with multiple time windows and hotel selection. *Journal Of The Operational Research Society*, 66(4), 615-626. doi: <https://doi.org/10.1057/jors.2014.17>

Castro, M., Sörensen, K., Vansteenwegen, P. & Goos, P. (2012). A simple GRASP+VND for the travelling salesperson problem with hotel selection. *Faculty of Applied Economics*, 024. Erişim adresi: <https://repository.uantwerpen.be/docman/irua/fl71bd/1c42703e.pdf>

Castro, M., Sörensen, K., Vansteenwegen, P. & Goos, P. (2013). A memetic algorithm for the travelling salesperson problem with hotel selection. *Computers Operations Research*, 40(7), 1716-1728. doi: <https://doi.org/10.1016/j.cor.2013.01.006>

Castro, M., Sörensen, K., Vansteenwegen, P. & Goos, P. (2014). A fast metaheuristic for the travelling salesperson problem with hotel selection. *4OR*, 13(1), 15-34. doi: <https://doi.org/10.1007/s10288-014-0264-5>

Divsalar, A., Vansteenwegen, P., Sörensen, K. & Cattrysse, D. (2014). A memetic algorithm for the orienteering problem with hotel selection. *European Journal Of Operational Research*, 237(1), 29-49. doi: <https://doi.org/10.1016/j.ejor.2014.01.001>

Gencil, C. A. (2019a). *Otel seçimli gezgin satıcı problemi için yeni matematiksel modeller* (Yüksek Lisans Tezi). Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara. YÖK tez veri tabanından erişildi. Tez No. 539322

Gencil, E. (2019b). *Otel seçimli oryantiring problemi için yeni matematiksel modeller* (Yüksek Lisans Tezi). Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara. YÖK tez veri tabanından erişildi. Tez No. 539312

Hansen, P. & Mladenovic, N. (2003). A tutorial on variable neighborhood search. *GERAD and Mathematical Institute Technical Report*, SANU, Belgrade. Erişim adresi: <https://www.researchgate.net/publication/2906122>

Hoek, S. V. (2016). Tabu search for the orienteering problem with hotel selection. *Econometrics and Operations Research*, *Econometrie*. Erişim adresi: <http://hdl.handle.net/2105/34358>

Lu, Y., Benlic, U. & Wu, Q. (2017). A hybrid dynamic programming and memetic algorithm to the travelling salesman problem with hotel selection. *Computers and Operations Research*, 90, 193-207. doi: <https://doi.org/10.1016/j.cor.2017.09.008>

Radmanesh, M., Kumar, M., Nemati, A. & Sarim, M. (2016). Solution of traveling salesman problem with hotel selection in the framework of MILP-Tropical Optimization. *American Control Conference (ACC)*, 6-8, Boston, MA, USA. Erişim adresi: https://www.researchgate.net/publication/305426314_Solution_of_Traveling_Salesman_Problem_with_Hotel_Selection_in_the_Framework_of_MILP-Tropical_Optimization

Sousa, M. M., Ochi, L. S. & Martins, S. L. (2019). An efficient heuristic to the travelling salesperson problem with hotel selection. *Hybrid Metaheuristics Conference*, 31-45.

Erişim adresi: https://link.springer.com/chapter/10.1007/978-3-030-05983-5_3

Toledo, A. & Riff, M. C. (2015). HOPHS: A hyperheuristic that solves orienteering problem with hotel selection. *International Conference on Digital Information Processing and Communications (ICDIPC)*, Sierre, Switzerland.

Erişim adresi:

https://www.researchgate.net/publication/308854299_HOPHS_A_hyperheuristic_that_solves_orienteering_problem_with_hotel_selection

Vansteenwegen, P., Souffriau, W. & Sörensen, K. (2011). The travelling salesperson problem with hotel selection. *Journal Of The Operational Research Society*, 63(2), 207–17. doi: <https://doi.org/10.1057/jors.2011.18>

«ANT/OR,» [Çevrimiçi]. Available: <https://antor.uantwerpen.be/instances-in-the-paper-a-memetic-algorithm-for-the-travelling-salesperson-problem-with-hotel-selection/>