Avrupa Bilim ve Teknoloji Dergisi
Sayı 53, S. 71-83, Ocak 2024

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 53, pp. 71-83, January 2024

Research Article

Gezgin Hırsız Problemi için Matematiksel Model ve Genetik Algoritma

Kübra Yıldırım[1*], Muzaffer Kapanoğlu[2]

[1*] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, Eskişehir, Türkiye, (ORCID: 0000-0002-1022-0416) kubrayildirim892@gmail.com

[2] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, Eskişehir, Türkiye (ORCID: 0000-0002-8217-7517) mkapanoglu@gmail.com

(İlk Geliş Tarihi 25 Aralık 2022 ve Kabul Tarihi 17 Aralık 2023)

(**DOI:** 10.5281/zenodo.10623816)

ATIF/REFERENCE: Yildirim, K. & Kapanoglu, M. (2024). Gezgin Hırsız Problemi için Matemattiksel Model ve Genetik Algoritma. *Avrupa Bilim ve Teknoloji Dergisi*, (53), 71-83.

Öz

Bu makalede, iyi bilinen iki kombinatoryal optimizasyon problemi, yani Gezgin Satıcı Problemi (GSP) ve Sırt Çantası Probleminin (SÇP) birleşimi olan Gezgin Hırsız Problemi (GHP) ele alınmıştır. Bu tür çok bileşenli optimizasyon problemlerinin çözülmesi sadece içerdiği katı optimizasyon problemleri nedeniyle değil, özellikle farklı bileşenler arasındaki karşılıklı bağımlılıklar nedeniyle de zordur. Bu problemin amacı, bir hırsızın tüm şehirleri ziyaret ettiği ve maksimum faydayı elde etmek için hangi şehirden hangi eşyanın alınması gerektiğini belirleyen bir toplama planını oluşturmaktır. Ele alınan problem için matematiksel model geliştirilmiş ve büyük boyutlu problemlerin önerilen matematiksel model ile çözülememesi nedeniyle iki farklı genetik algoritma geliştirilmiştir. Geliştirilen algoritmaların performansları farklı özelliklerdeki test problemleri kullanılarak test edilmiştir.

Anahtar Kelimeler: Gezgin Hırsız Problemi, Gezgin Satıcı Problemi, Sırt Çantası Problemi, Genetik Algoritma

Mathematical Model and Genetic Algorithm for the Traveling Thief Problem

Abstract

In this article, two well-known combinatorial optimization problems are discussed, namely the Traveling Thief Problem (TTP), which is a combination of the Traveling Salesman Problem (TSP) and the Knapsack Problem (KP). Deciphering such multicomponent optimization problems is difficult not only because of the strict optimization problems involved, but also because of the interdependencies between the different components in particular. The purpose of this problem is to form a collection plan, according to which a thief visits all cities and determines which items should be taken from which city in order to get the maximum benefit. A mathematical model has been developed for the problem under consideration, and two different genetic algorithms have been developed because large-sized problems cannot be solved with the proposed mathematical model. The performances of the developed algorithms were tested using test problems of different properties.

Keywords: The Traveling Thief Problem, The Traveling Salesman Problem, The Knapsack Problem, Genetic Algorithm.

[*] Sorumlu Yazar: kubrayildirim892@gmail.com

1. Giriş

Gerçek dünya optimizasyon problemleri genellikle birbirleriyle etkileşim halinde olan birkaç NP-zor kombinatoryal optimizasyon probleminden oluşur (Klamroth vd., 2017; Bonyadi vd,, 2019). Bu tür sorunları çözebilmek için bu etkileşimleri anlamak ve bunlarla başa çıkmak önemlidir. Yalnızca içerdikleri zor optimizasyon problemleri nedeniyle değil özellikle farklı bileşenler arasındaki karşılıklı bağımlıklar nedeniyle de zordur. Michalewicz'de (2012) belirtildiği gibi, gerçek dünya problemlerinin ana karmaşıklıklarından biri, birçok geleneksel yaklaşımı etkisiz hale getiren alt problemler arasındaki karşılıklı bağımlılıktır. Sonuç olarak global bir optimum veya yüksek kaliteli bir çözüm elde etmek için alt problemlerin yüksek kaliteli kısmi çözümlerinin probleme nasıl entegre edileceği hala açık bir sorundur.

Son yıllarda literatürde dikkat çeken çok bileşenli kombinatoryal optimizasyon problemlerinden biri de gezgin hırsız problemidir (Bonyadi vd., 2013). Gezgin Hırsız Problemi (GHP), iyi bilinen iki kombinatoryal optimizasyon problemi, yani Gezgin Satıcı Problemi ve Sırt Çantası Problemi içermekle beraber yeni gerçek hayat problemlerini modelleme potansiyeli olan kombinatoryal bir özelliğe sahiptir. Kısaca, GHP, çeşitli düğümlerden kendine özgü ağırlıkları ve kazançları olan eşyaları çalan bir hırsızın uygulayacağı stratejiye öykünmektedir. Hırsız, tüm şehirleri bir kez ziyaret etmeli ve toplam karı en üst düzeye çıkaracak şekilde öğeleri toplamalıdır. Hırsız, tüm düğümleri bir kez ziyaret etmeli ve toplam karı en üst düzeye çıkaracak şekilde öğeleri toplamalıdır. Hırsız, sınırlı kapasiteli bir sırt çantası kullanır ve bunun için toplam seyahat süresiyle orantılı olarak kira öder. İki alt bileşeni (GSP ve SÇP) doğrusal olmayan bir şekilde birbirine bağlayan, hırsızın hızının o ana kadar toplanan öğelerin ağırlığı ile ters orantılı olmasıdır. Ayrıca, başlangıç şehrine atanan öge bulunmamaktadır.

Bu araştırma, gerçek dünya problemlerinin farklı bir problemle birbirine bağlanarak modellendiği bilinenlerden daha karmaşık optimizasyon problemini çözmek için bir matematiksel model ve iki genetik algoritma önermektedir. Yüksek düzeyde karmaşıklık içeren GHP tarzı problemler, aynı zamanda modern optimizasyon yöntemlerinin performanslarının analizinde de kıyaslama setlerinde kullanılmaktadır. Algoritmik güçlüklerin başında, GSP veya SÇP için iyi olan bir çözümün diğeri için kötü olması yatmaktadır. Küçük boyutlu problemler kabul edilebilir bir sürede çözülebilmekle beraber büyük örneklerin yaklaşık en iyi çözümlerini dahi beklemek teorik ve pratik açıdan anlamlı değildir. Problem için test örnekleri temel olarak GSP'ine ait parametreleri içerir ve bu parametreler sırt çantası parametreleriyle daha da şekillenir. Problemin GSP alt problemi için TSPLIB kütüphanesi kullanılır (Reinelt, 1991).

2. Materyal ve Metot

Gezgin Hırsız Problemi, NP-zor sınıfında yer almakta olup bu tip problemler çözüm zamanı problemin büyüklüğüne bağlı olarak üstel artış gösteren ya da bilinen eniyileme metotlarıyla çözülemeyen problemler olarak bilinir. Bu çalışmada, incelenen problemin çözümü için yapay zekâ tekniklerinden biri olan genetik algoritma geliştirilmiştir. Büyük boyutlu problemlere çözüm bulunabilmesi ve iyi çözümlere daha hızlı sürede ulaşmak için geliştirilen iki farklı genetik algoritma ve matematiksel model izleyen alt başlıklarda açıklanmıştır.

2.1. Matematiksel Model

Gezgin Hırsız Problemi şu şekilde tanımlanır. m adet öge kümesi $\{1,2,\dots,m\}$, n adet düğüm kümesi $\{1,2,\dots,n\}$ arasında dağıtılır. d_{ij}, herhangi bir düğüm çiftinin $i,j \epsilon \{1,2,\dots,n\}$ arasındaki mesafeyi verir. Her bir $k \epsilon m$ ögesinin değeri p_k, ağırlığı w_k ve bulunduğu şehri A_i vardır. Örneğin, $A_i = \{1,2,5\}$, i ögesinin yalnızca şehir 1, 2 veya 5'ten seçilebileceğini ifade eder. Hırsız, kiraladığı çantası ile ilk şehirden başlayarak tüm şehirleri bir kez ziyaret etmek ve başlangıç düğüme geri dönmek şartıyla çantasının kapasitesini W aşmayacak şekilde şehirlerden ögeleri toplayabilir. Çalınan eşyalar sırt çantasında depolandıkça ağırlaşır ve hırsız sırt çantasının ağırlığı ile ters orantılı bir şekilde yavaş hareket eder. Hırsız $v = v_{max} - W_c * (v_{max} - v_{min})/W$ hızıyla hareket eder, burada W_c hırsızın anlık ağırlığını göstermektedir. Sırt çantasının boş olması durumunda hırsız maksimum hızda v_{max} hareket ederken, sırt çantasının tamamının dolu olması durumunda ise minimum hızda v_{min} hareket eder. İki şehir arasındaki seyahat süresi formül (1) yoluyla hesaplanır, burada v_c hırsızın anlık hızını ifade etmektedir.

$$t_{ij} = \frac{d_{ij}}{v_c} \quad (1)$$

GHP amacı aşağıda verilen fonksiyonun enbüyüklemektir:

$$G(x,z) = g(z) - R * f(x,z) \quad (2)$$

g toplanan ögelerin toplam değeri iken, R çantanın birim zaman başına kiralama ücreti ve f turun toplam zamanıdır. x ve z sırasıyla, tur ve toplama planıdır.

GHP'nin sayısal örneği, Şekil 1'de verilen grafik üzerinden gösterilecektir.

- $n = 4, m = 5, W = 3, v_{max} = 1$ ve $v_{min} = 0.1$

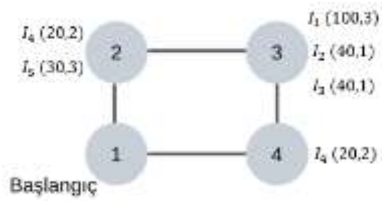
- $D = \begin{bmatrix} - & 5 & 6 & 6 \\ 5 & - & 5 & 6 \\ 6 & 5 & - & 4 \\ 6 & 6 & 4 & - \end{bmatrix}$

- Her bir ögenin değeri ve ağırlığı: $I_i(p,w)$: $I_1 = (100,3), I_2 = (40,1), I_3 = (40,1), I_4 = (20,2), I_5 = (30,3)$

- Her bir ögenin şehirlerdeki mevcudiyeti(A_i): $A_1 = \{3\}, A_2 = \{3\}, A_3 = \{3\}, A_4 = \{2,4\}, A_2 = \{2\}$

- Birim zaman başına çanta kiralama ücreti: $R = \$1$

$x = \{1,3,2,4\}$ düğümlerin ziyaret sırasını gösterirken $z = \{0,3,0,2,0\}$ ise hangi öğenin hangi düğümden alındığını göstermektedir. Bu örnek için I_2 3 numaralı şehirden alınırken, I_4 2 numaralı düğümden alınmıştır.



Şekil 1. TTP için örnek problem (Figure 1. Example problem for TTP)

Örnek problemin amaç fonksiyon değerini (2) hesaplayalım. Hırsız aralarındaki mesafenin 6 olduğu düğüm 1'den düğüm 3'e hareket ediyor. Çantanın anlık ağırlığı (W_c) 0 olduğundan hırsızın hızı $v = v_{max} = 1$ ve iki düğüm arasındaki birim zaman $t_{1,3} = 6$ olarak hesaplanır. Düğüm 3'te öge I_2 çantaya atılır ve anlık ağırlığı $W_c = 1$ olarak değişir. Böylece $v_c = 0.7$ ve $t_{3,2} = 7.14$ elde edilir. Düğüm 2'den hırsız öge I_4 çalar ve çantanın anlık ağırlığı $W_c = 3$ olur. Bunun sonucunda hırsızın hızı $v = v_{min} = 0.1$ değerine ulaşır. Hırsız düğüm 4'te bu hız değeri ile hareket eder ve ulaşması $t_{2,4} = 60$ birim zamanını alır. Ve en sonunda hırsız düğüm 4'ten düğüm 1'e $v_c = 0.1$ hızı ile geri döner ve bu dönüşünü $t_{4,1} = 60$ birim zamanda yapar. Hepsini bir araya getirirsek, turun tamamlanma süresi $f(x,z) = 60 + 60 + 7.14 + 6 = 133.14$ ve çantanın toplam değeri $g(z) = 40 + 20 = 60$ olarak hesaplanır. Böylece amaç fonksiyonunun değeri $60 - 1 * 133.14 = -73.14$ olacak şekilde elde edilir.

Problemin matematiksel modeli aşağıda verilmiştir:

$$Enb\ z \quad \sum_k \sum_j p_k * z_{kj} - R * SURE \tag{1}$$

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \quad \forall j \tag{2}$$

$$\sum_{j=2}^{n} x_{1j} = 1 \tag{3}$$

$$u_i - u_j + n * x_{ij} \leq n - 1 \qquad 1 < i \neq j \leq n \tag{4}$$

$$\sum_{i=1}^{n} x_{ij} - \sum_{i=1}^{n} x_{ji} = 0 \quad \forall j \tag{5}$$

$$z_{kj} = 0 \quad \forall k, j | A_{kj} = 0 \tag{6}$$

$$\sum_{j}^{n} A_{kj} * z_{kj} \leq 1 \quad \forall k \tag{7}$$

$$\sum_{k}^{m} \sum_{j}^{n} w_k * z_{kj} \leq Q \tag{8}$$

$$wc_j \geq \sum_{k}^{m} w_k * z_{ki} + wc_i + M * \left(x_{ij} - 1\right) \quad \forall i, j | i > 1, i \neq j \tag{9}$$

$$wc_j \geq \sum_{k}^{m} w_k * z_{ki} + M * \left(x_{ij} - 1\right) \quad \forall i, j | i = 1, i \neq j \tag{10}$$

$$S_{ij} = \frac{d_{ij} * x_{ij}}{v_{max} - wc_j \frac{v_{max} - v_{min}}{Q}} \quad \forall i, j \tag{11}$$

$$SURE = \sum_{i}^{n} \sum_{j}^{n} S_{ij} \tag{12}$$

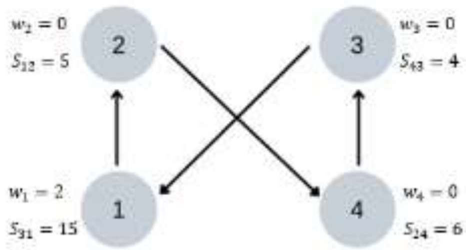
Kısıt (1) problemin amaç fonksiyonunu göstermektedir. Kısıt (2) her bir düğümün ziyaret edilmesini, kısıt (3) başlangıç düğümden hırsızın çıkmasını garanti etmektedir. Kısıt (4) tüm düğümleri kapsayan tek bir turun olmasını zorunlu kılarak alt tur oluşmasını engeller. Kısıt (5) bir düğüme giriş yapıldığında, giriş yapılan aynı düğümü terk etmesi sağlanmaktadır. Kısıt (6) ögelerin bulunmadıkları düğümlerden alınmalarını engellerken kısıt (7) her bir öğenin mevcut düğümler arasından en fazla bir kez seçilmesini garantilemektedir. Kısıt (8) toplanan öğelerin toplam ağırlığının sırt çantasının kapasitesini aşamayacağını ifade eder. Kısıt (9)-(10) planlanan düğümü ziyaret etmeye giderken çantanın anlık ağırlığını belirlemektedir. Kısıt (11) ziyaret edilen iki düğüm arasındaki geçiş süresini hesaplarken, kısıt (12) turun toplam süresini hesaplamaktadır.

Problemin çözümünde kullanılan matematiksel modelin tutarlılığı küçük bir problem üzerinde test edilmiştir. Küçük problem karma tamsayılı ve doğrusal olmayan problem çözümünde kullanılan GAMS/DICOPT çözücüsü yardımıyla çözümlenmiştir. İlgili sonuçlar bir sonraki başlıkta yer verilmiştir.

2.1.1. Küçük Boyutlu Bir Örnek Problem

Önerilen matematiksel modelin çözüm performansını test etmek amacıyla problem GAMS programlama dilinde yazılmış ve Şekil 1'de bahsedilen örnek kullanılarak test edilmiştir. $x = \{1,2,4,3\}$ sıralaması ve $z = \{0,0,1,1,0\}$ toplama planı için optimum amaç değeri $G(x,z) = 50$ olarak bulunmuştur. Hırsızın anlık çanta ağırlıkları ve iki şehir arasında geçirdiği süre Şekil 2'de yer almaktadır.

Hırsız 2 ve 4 numaralı düğümlerden hiçbir eşya toplamaz. Bu sebeple bu turun maliyeti 15 olarak hesaplanır. Hırsız sadece üçüncü düğümden 2 ve 3 numaralı öğeleri alır, bu da 80 birim kar sağlar. Hırsız 3 numaralı düğümden 1 numaralı düğüme dönüş yolunda çantasının ağırlığı 2'dir. Hızını düşürdüğü için dönüş yolunun maliyeti 15 olarak hesaplanır. Sonuç olarak nihai amaç değeri Z=80-15-15=50 olarak hesaplanır. Tur ve toplama planı Tablo 1'de gösterilmiştir.



Şekil 2. Gezgin Hırsızın Anlık Çanta Ağırlık ve Seyahat Süresi (Figure 2. The Traveler Thief's Instant Bag Weight and Travel)

Tablo 1. Ziyaret Edilen Düğümler ve Çantaya Eklenen Öğeler (Table 1. Visited Nodes and Items Added to the Bag)

Sırayla Ziyaret Edilen Düğümler	İki Düğüm Arası Mesafe	Çantaya Eklenen Öğeler	Ögelerin Ağırlıkları	Süre	Birikimli Süre
1	-	-	-	5	5
2	5	-	-	6	11
4	6	-	-	4	15
3	6	2-3	1-1	15	30

2.2. Önerilen Genetik Algoritmalar

Ele alınan problem, NP-zor sınıfında olup, çözüm zamanı problemin büyüklüğüne bağlı olarak üstel artış gösterir. NP-zor problemler sınıfında yer alması nedeniyle büyük boyutlu problemlerin çözümü için bir genetik algoritma (GA) geliştirilmiştir.

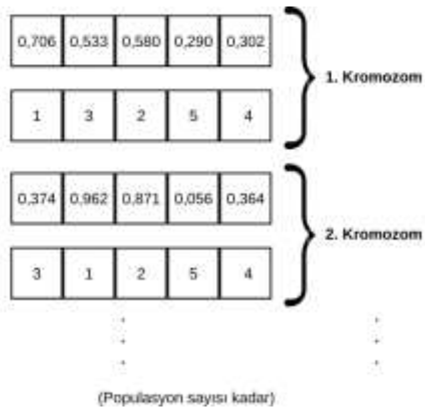
Genetik algoritmalar hem doğrudan problemi çözmek için hem de matematiksel modellerin tam sayılı çözümlerini bulmak için kullanılabilmektedir. Bu çalışmada her iki yöntem de kullanılmıştır. GHP için geliştirilen iki farklı algoritmanın kromozom yapısı ve adımları izleyen başlıklarda ele alınmıştır. GHP-GA.1, genetik gösterimden yararlanan genetik algoritma iken, GHP-GA.2 önerilen matematiksel modeli çözen genetik algoritmayı ifade etmektedir.

2.2.1 Çanta Yönelimli Genetik Algoritma (GHP-GA.1)

Geliştirilen GA izleyen başlıklarda açıklanmıştır.

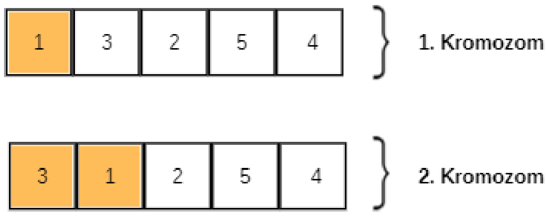
2.2.1.1 Kromozom Yapısı

Problemde seçilecek öge sayısının belirlenmesi için rassal anahtar yöntemi kullanılmıştır. Bu yöntem ile her bir gen için rassal sayı türetilmektedir. Çantanın kapasitesini aşmayacak şekilde seçilecek olan ögeler rassal sayıların büyüklükleri dikkate alınarak belirlenir. Önerilen GA'da kromozomların oluşmasında (0,1) arasında her bir öge için rassal sayı türetilir. Öge sayısının 5, şehir sayısının 4 olduğunu varsayalım. Şekil 3'te örnek bir kromozom verilmiştir. Bu kromozomda her bir gen için üretilen rassal sayılar büyükten küçüğe doğru sıralanır ve karşı gelen sıra numarası yazılır. Sıra numaraları öge numaraları ile eştir.



Şekil 3. Kromozom Yapısı (Figure 3. Chromosome Structure)

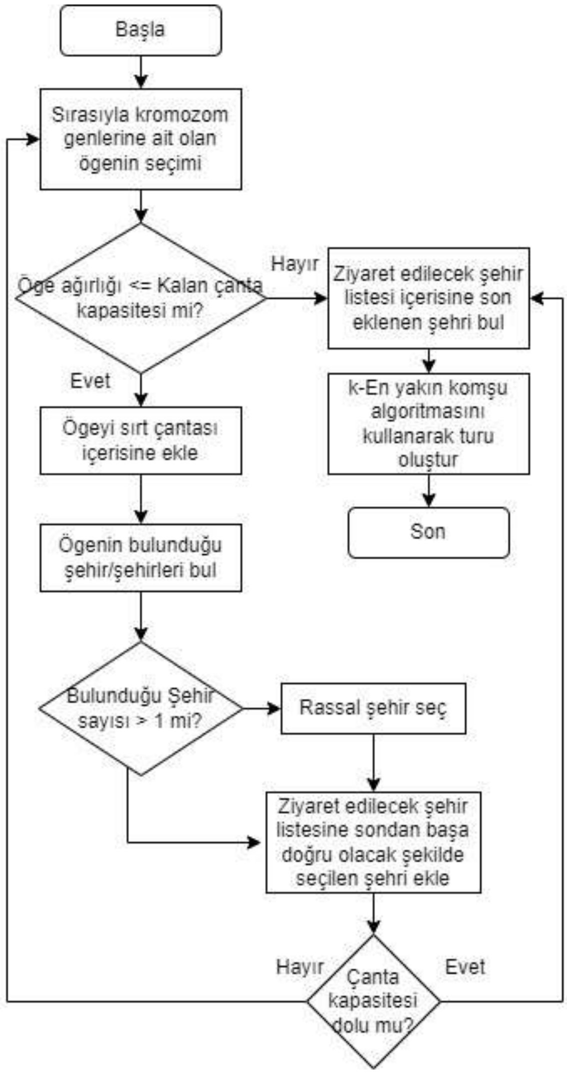
Öge sırası rassal olarak seçildikten sonra çanta kapasitesini aşmayacak şekilde seçilen öge kümesi belirlenecektir. Örnekte bulunan iki kromozom için, öge ağırlıklarının sırasıyla {2,3,1,1,2} olduklarını varsayarsak çanta kapasitesini aşmayacak şekilde ögelerin seçilmesi Şekil 4'te gösterilmiştir.



Şekil 4. Sırt Çantasına Atılan Öge Seçimi (Figure 4. Selection of Items for the Backpack)

2.2.1.2 Geliştirilen Genetik Algoritmanın Adımları

Problemin çözümü için geliştirilen genetik algoritmanın adımları Şekil 5'te akış şemasında yer almaktadır. Akıştaki her bir sürecin nasıl gerçekleştiği izleyen kısımlarda detaylı bir şekilde anlatılmıştır.



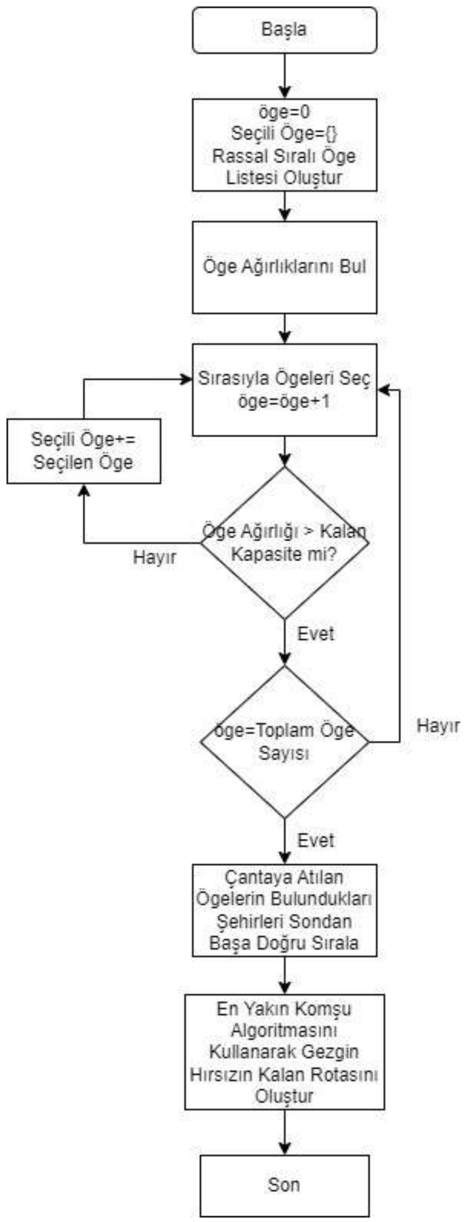
Şekil 5. Geliştirilen Genetik Algoritma İçin Akış Şeması (Figure 5. Developed Flowchart for the Genetic Algorithm)

İlk Neslin Türetilmesi:

İlk neslin türetilme adımları şu şekilde sıralanır:

1- Rassal bir şekilde öge numaraları sıralanır.
2- Her bir ögenin ağırlığı listelenir.
3- Rassal sıralı ögeler çanta kapasitesini aşmayacak şekilde toplama planına eklenir.
4- Çantaya eklenen ürünlerin bulundukları şehirlerin listesi oluşturulur.
5- Şehir listesi içerisinde bulunan şehirler içerisindeki ögelerin, çantaya atılma sırasına göre bulundukları şehirler sondan başa sıralanır.
6- Çantadaki ögelerin bulundukları şehirler sıralandıktan sonra, sondan başa doğru olacak şekilde en kısa yol algoritması ile rota oluşturulur.

İlk neslin türetilmesi yöntemini adımları Şekil 6’da verilmiştir.



Şekil 6. İlk Neslin Türetilmesine Dair Akış Şeması (Figure 6. Flow of First Generation Derivation)

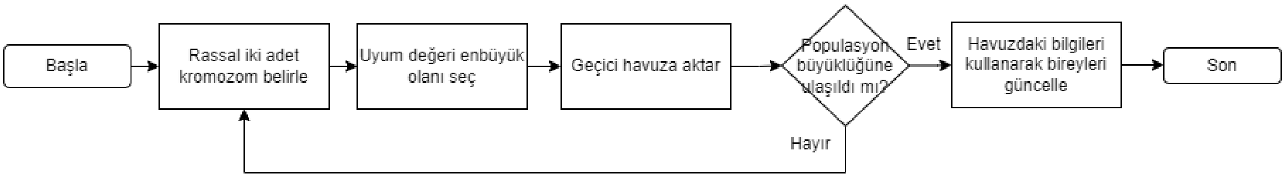
İkili Turnuva Yöntemi:

Bu yönteme göre turnuva büyüklüğü kadar birey rassal olarak seçilir ve turnuva grubuna girer. Literatürde sık kullanılan ikili turnuva yöntemi tercih edilmiştir. Bu yöntemdeki ilke belli bir büyüklükte oluşturulan turnuvaların içerisindeki uyum değeri en iyi olan bireyin seçilmesidir. Uyum değerinin en iyi olma durumu amaç fonksiyonun göre değişmekle birlikte, enküçükleme problemleri için en küçük, enbüyükleme problemleri için ise en büyük uyum değeridir. İkili turnuva yöntemi adımları aşağıda özetlenmiştir.

İkili turnuva yöntemi adımları şu şekildedir:

1- Rassal bir şekilde iki adet kromozom belirlenir.
2- Belirlenen iki adet kromozom içerisinde uyum değeri en büyük olan kromozom seçilir.
3- Seçilen kromozomların bilgileri geçici bir havuza aktarılır.
4- Adım 1 ve Adım 3 popülasyon büyüklüğüne kadar tekrar ettirilir.
5- Geçici havuz içerisindeki bilgiler gerçek popülasyona aktarılır.

Bahsedilen adımlar Şekil 7'de gösterilmiştir.



Şekil 7. İkili Turnuva Yöntemine Dair Akış Şeması (Figure 7. Flowchart of Binary Tournament Method)

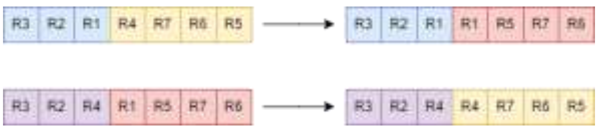
Elitizm Yöntemi:

Elitizm, topluluktaki en iyi belirli sayıdaki bireyleri hafızada tutarak bir sonraki nesle değiştirmeden aktarılmaya yarar. Bu yöntem ile oluşacak yeni topluluğa iyi bireyler girecek ve bunlardan üreyecek bireyler de daha iyi sonuç verecektir. Geliştirilen genetik algoritmada Elitizm yöntemi şu şekilde kullanılmıştır.

1- İlk neslin içerisinden uyum değeri en büyük olan birey belirlenir.
2- Bu kromozomdaki genler ve uyum değeri hafızada tutulur.
3- Hafızada tutulan kromozom bilgileri bir sonraki neslin ilk kromozomu olur.
4- Adım 2 ve Adım 3 nesil boyunca tekrarlanır.

Çaprazlama Yöntemi:

Çaprazlama, ebeveynlerden bazı genleri belirli bir kurala göre alıp yeni bireyler oluşturma yöntemidir. Üzerinde çaprazlama yapılacak konum rastgele seçilir ve bu noktadan sonra gelen genler karşılıklı bir şekilde yer değiştirir. Çaprazlanacak kromozomların genleri başlangıç popülasyonu oluşturulan genlerden kopyalandığından dolayı gen bilgisi 0-1 arası rassal türetilmiş sayılardır. Çaprazlama sonucu elde edilen kromozomlar gerçek popülasyona aktarılır. Geliştirilen genetik algoritmada çaprazlama Şekil 8'de gösterilmiştir.



Şekil 8. Çaprazlama Yöntemi Gösterimi (Figure 8. Representation of Crossover Method)

Mutasyon Yöntemi:

Bu aşamanın ana amacı var olan bir çözüme ait kromozomun bazı genlerinin değerini değiştiren belirli yöntemleri kullanarak iterasyonlar sonucu mevcut nesilde kaybolmuş veya hiç incelenmemiş genleri ve çözümleri gen havuzuna eklemektir. Mutasyon işlemi ile mevcut çözüm kümesine yeni bilgiler eklenerek çözüm uzayının farklı alanlarının taranması sağlanmaktadır. Mutasyon genelde ikincil bir operatör olup, genler üzerine kurgulandığından nispeten küçük bir orana göre gerçekleştirilmiştir. Mutasyon sonucunda nesildeki kromozomlar ve genler belirli bir yaklaşım çerçevesinde güncellenir.

2.2.2 Rota Yönelimli Genetik Algoritma (GHP-GA.2)

GHP için oluşturulan matematiksel modeli çözmek için geliştirilen algoritmanın adımları aşağıdaki bölümde incelenmiştir.

2.2.2.1 Kromozom Yapısı

Başlangıç popülasyonu için n*n ölçüde rassal sayı içeren matris oluşturulur. Her satırın kendi içinde barındırdığı minimum rassal sayısına karşılık gelen düğüme atama yapılır. Atama yapılacak olan şehrin daha önce ziyaret edilmemiş ve başlangıç şehri olmaması gerekmektedir. Şekil 9'da 5*5 büyüklüğünde oluşturulan matrisi inceleyecek olursak ilk satırda minimum rassal sayıya karşılık gelen şehir 2 olduğu için 1-2 sıralaması ve bu yöntem ile daha sonra 1-2-5-4-3 sıralaması oluşturulur.

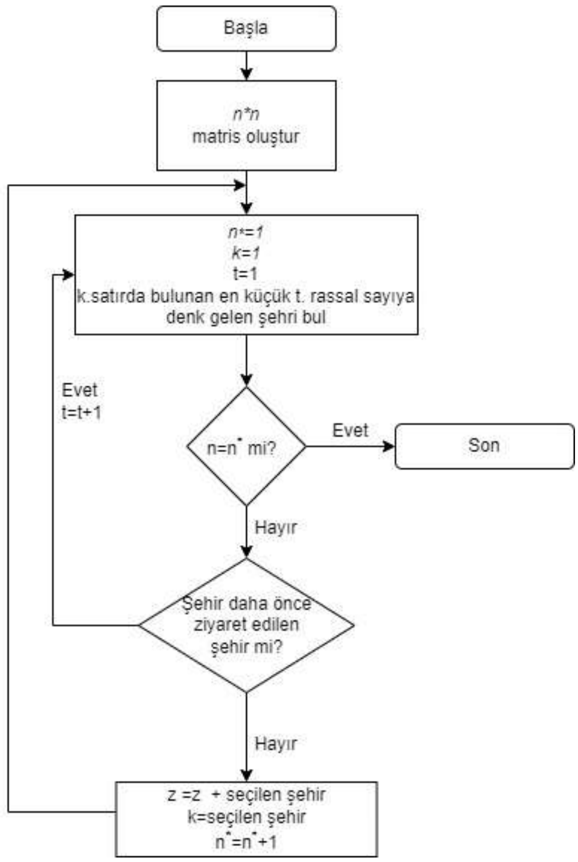
x_{ij}	1	2	3	4	5
1	0,79006	0,12125	0,87717	0,33847	0,77559
2	0,47616	0,53651	0,96949	0,50807	0,26603
3	0,63899	0,79604	0,61798	0,94629	0,52051
4	0,04472	0,76486	0,09591	0,46576	0,08591
5	0,18467	0,03649	0,18418	0,0203	0,11088

x_{ij}	1	2	3	4	5
1	0,79006	0,12125	0,87717	0,33847	0,77559
2	0,47616	0,53651	0,96949	0,50807	0,26603
3	0,63899	0,79604	0,61798	0,94629	0,52051
4	0,04472	0,76486	0,09591	0,46576	0,08591
5	0,18467	0,03649	0,18418	0,0203	0,11088

x_{ij}	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	0	1
3	1	0	0	0	0
4	0	0	1	0	0
5	0	0	0	1	0

Şekil 9. Kromozom Yapısı Gösterimi (Figure 9. Representation of Chromosome Structure)

Gezgin satıcının turu Şekil 10'da anlatılmıştır.



Şekil 10. Gezgin Hırsızın Turu Adımları (Figure 10. Steps of the Traveling Thief's Tour)

2.2.2.2 Geliştirilen Genetik Algoritmanın Adımları

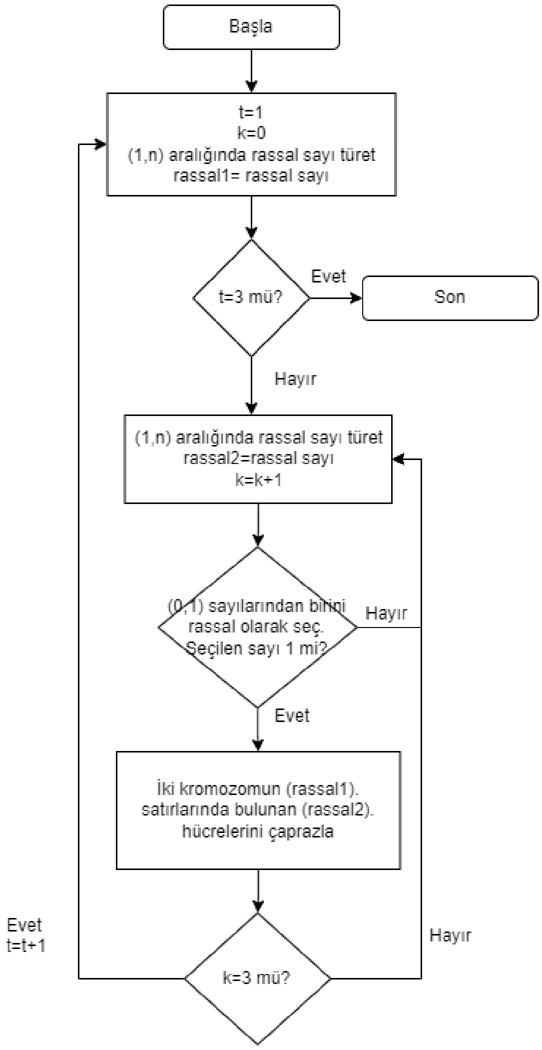
2.2.1.2 İlk Neslin Türetilmesi başlığında bahsedilen öge toplama planı, bu algoritma için uyarlanmıştır. Gezgin satıcının toplama planı ve turu oluşturulduktan sonra kromozomun uyum değeri hesaplanır. Daha sonra yukarıda elitizm, ikili turnuva yöntemi başlıklarında sözü edilen yöntemler gerçekleştirilir.

Çaprazlama Yöntemi:

n*n matriste (1,n) aralığında seçilen rassal sayıya eşit olan satır içerisinde rassal olarak 3 hücre seçilir. 0-1 sayıları, 0 seçilen hücrelerin değerini değiştirme, 1 seçilen hücreler arasında çaprazlama yapılsın anlamlarını taşıyacak şekilde rassal olarak 0 ya da 1 sayısı seçilir. Geliştirilen genetik algoritmada çaprazlama yöntemi Şekil 11'de gösterilmiştir.

Mutasyon Yöntemi:

Çaprazlama sonucunda elde edilen kromozom genlerinde rastgele 3 adet seçilir ve rassal sayı türetilir. Rassal sayı mutasyon oranından küçük ise gen mutasyona uğrar.



Şekil 11. Çaprazlama Yöntemine Dair Akış Şeması (Figure 11. Flowchart of Crossover Method)

3. Araştırma Sonuçları ve Tartışma

Geliştirilen iki algoritmanın, performanslarını araştırmak için Bonyadi ve diğerleri tarafından oluşturulan GHP kıyaslama örneklerinin temsili bir alt kümesi seçildi. Şehir sayısı, çanta boyutu, sırt çantası tipi ve sırt çantası kapasitesine göre değişen farklı soru örnekleri oluşturuldu. Çanta kapasitesi, küçük $c = 0{,}2$, orta $c = 0{,}5$, büyük $c = 0{,}8$ olmak üzere $W = c\sum_{i=1}^{m} w_i$ formülü ile hesaplanmıştır. Çanta tipi ise ilintisiz (uncorrelated(unc)), ilintisiz ve benzer ağırlıklı (uncorrelated with similar weights(usw)), sınırlı güçlü korelasyon (bounded strongly correlated(bsc)) olmak üzere üç farktı tipe ayrılmıştır. Hız aralığı $v_{min} = 0.1$ ve $v_{max} = 1.0$ olarak ayarlanmıştır.

Soru örneklerinin ad modeli aşağıdaki gibi olacaktır.

Şehir Sayısı – Şehir Başına düşen Öge Sayısı – Çanta Boyutu- Çanta Tipi

Aşağıda Tablo 2'de örnek veri seti için her iki algoritmanın çözüm sonuçları yer almaktadır.

Tablo 2. GA ile Elde Edilen Örnek Veri Seti Problemlerinin Çözümü (Table 2. Solution of Sample Dataset Problems Obtained with GA)

Problemler	GHP-GA. 1	GHP-GA. 2
14-1-k-unc	408,42	-1180,22
14-1-o-unc	995,78	-2346,73
14-1-b-unc	-963,82	-4688,15
14-1-k-usw	-337,11	-2142,8
14-1-o-usw	447,76	-2672,33
14-1-b-usw	1980,53	-4612,35
14-1-k-bsc	813,13	-521,16
14-1-o-bsc	1315,79	-2448,65
14-1-b-bsc	1010,82	-4080,14

Yukarıda *2.2 Önerilen Yöntem* başlığı altında anlatılan genetik algoritmaların isimleri sırası ile geliştirilen algoritma 1 ve geliştirilen algoritma 2 olarak belirlenmiştir.

Elde edilen sonuçlar incelendiğinde, geliştirilen genetik algoritma 1 daha başarılı çözümler üretmiştir. Sonuçlar incelendiğinde, sırt çantasına atılan ögelerin bulundukları düğümler, hırsızın son ziyaret edeceği düğümler olarak atanması ve turun oluşumu için kalan şehirleri sondan başa en kısa yol sezgiseli ile sıralanması hırsızın seyahat süresini kısaltmış olup sırt çantası için ödenecek olan kira ücretini azaltmıştır.

Rota yönelimli genetik algoritmada GHP ve SP birbirinden bağımsız olarak çalışır. Sırt çantasına atılan ögeler, GHP sonucunda hırsızın ziyaret edeceği turu elde ettikten sonra, toplam tur süresi, kira ücreti, çantanın değeri bu sonuçlara göre hesaplanır. Bu bağımsızlık hırsızın başlangıç şehirlerinden çantasına öge eklemesine sebep olabilir. Hırsızın hızının yavaşlaması amaç fonksiyonu değerini olumsuz etkiler.

GHP-GA.1'in performansını değerlendirmek için Polyakovskiy vd. (2014) tarafından tanımlanan kapsamlı GHP örnekleri içerisinden seçili örnekler kullanılmıştır. Bu örnekler temel olarak GHP probleminde tanıtılır ve onu sırt çantası parametreleriyle daha da şekillendirir. Sorunu çözmek için bu örneklerin alındığı TSPLIB kitaplığı kullanılır (Reinelt, 1991).

Örnek veri seti içerisinden eil50_n50_01 örneği ele alınarak 3 farklı çanta tipine göre problem çözüldü. Çözdürülen 3 örnekte de şehir sayısı 51, öge sayısı 50 yani her bir şehre düşen öge sayısı 1 olarak kabul edilir. Intel® Core™ i5 – 5200 CPU 2.20GHz, 8 GB RAM özelliklerinde bir bilgisayar kullanılarak çalıştırılmıştır. Algoritma Visual Basic 6.0 programında kodlanmıştır.

Tablo 3. Test Problemlerinin Çözümü (Table 3. Solution of Test Problems)

Problem Adı	Çanta Kapasitesi	Kiralama Oranı	Toplam Kazanç
eil51_n50_uncorr_01	2226	7.19	226,131
eil51_n50_uncorr-similar-weights_01	4567	3.82	1139,185
eil51_n50_bounded-strongly-corr_01	4029	4.44	3354,263

Tablo 3'de her bir problemin çanta kapasitesi ve toplam kazanç değeri paylaşılmıştır. Literatürden alınan test problem sonuçları incelendiğinde geliştirilen algoritmanın örnek seti sonuçlarına önemli bir düzeyde yakınsadığı görülmektedir.

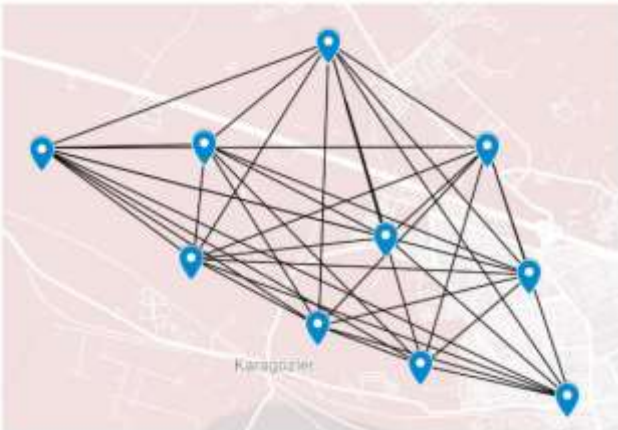
3.1. Geri Kazanım Süreçlerinde Gezgin Hırsız Problemi Uygulaması

Atık kâğıt çalışanları, geri dönüştürebilir kağıt ürünlerini çöplerden toplayarak atık depolarına satan ve geçimini sağlayan kişilerdir. Atık depoları ise daha büyük geri dönüşüm fabrikalarına satış yaparak geri dönüşüm sürecinde aracı roldedir. Kâğıt toplama arabası ile başlangıç deposundan çıkıp, rotası üzerinde sokaklarda bulunan çöp kutusu içinden kendi seçtiği kağıt vb dönüşüm malzemelerini toplar. Toplama arabasının kapasitesini aşmayacak şekilde başladığı depo noktasına geri dönerek günü tamamlar. Toplama arabasında bulunan geri dönüşüm maddelerine atıkları türlerine göre farklı fiyatlandırma uygulanmaktadır. Atık kâğıt çalışanları, Gezgin Hırsız Probleminde hırsız rolündeki kişiyi, toplama arabaları sırt çantasını temsil etmektedir. Toplayıcının hızı, toplama arabasının doluluğuna göre değişmektedir. Problemdeki ögeleri geri dönüşüm malzemeleri olarak düşünürsek, geri dönüşüm malzemelerinin satış fiyatları ögelerin değerini, ağırlıkları ise ögelerin ağırlıklarına karşılık gelmektedir. Toplama aracına kullandıkları gün başına kira bedeli ödemektedirler, depoya geri döndüklerinde kira bedellerini ödeyerek kazançlarını sağlamaktadırlar. Böylelikle atık toplayıcılar çöpü ekonomik değere dönüştürerek kazançlı çıkabilmektedirler.

3.2. Geri Kazanım Süreçlerinde Hipotetik Problemin Uygulaması

Hipotetik problem için Batıkent mahallesinde bulunan 10 farklı lokasyon (düğüm) ele alınmıştır. İlk düğüm (depo) hariç tüm düğümlerde geri dönüşüm malzemeleri bulunmaktadır. Çanta kiralama bedeli $R = 3{,}64\,pb$, çanta kapasitesi $W = 1850$ olarak belirlenmiştir. Hız değerleri $v_{min} = 0{,}1$ ve $v_{max} = 0{,}1$ olacak şekilde örnek bir problem türetilmiştir.

Şekil 10. Batıkent Mahallesinde Bulunan 10 Lokasyon (Figure 10. 10 Locations in Batıkent Neighborhood)



Şekil 11. Düğümler Arası Ayrıtlar (Figure 11. Edges Between Nodes)

Probleme ait mesafe matrisi Tablo 3'de verilmiştir. Ele alınan problemin matematiksel model çözümü için GAMS kullanılmıştır. Ayrıca excel programı içerisinde geliştirilen her iki algoritma da problem çözümü için çalıştırılmıştır. Hipotetik problem boyutu GAMS programında da çözülebilmesi adına küçük seçilmiştir.

Tablo 3. Hipotetik Probleme Ait Mesafe Matrisi (Table 3. Distance Matrix for the Hypothetical Problem)

D_{ij}	1	2	3	4	5	6	7	8	9	10
1	0	17,5	24,1	30,8	41,6	14,1	38,8	22,9	31,4	49,3
2	17,5	0	13,8	13,6	26,9	26,2	30,2	40,1	34,6	32,1
3	24,1	13,8	0	16,6	18,3	24,9	16,6	43,5	24,5	30,8
4	30,8	13,6	16,6	0	17,4	37,9	27,8	55,3	40,8	18,6
5	41,6	26,9	18,3	17,4	0	43,4	16,8	62,2	38,1	15,8
6	14,1	26,2	24,9	37,9	43,4	0	34,8	20,1	19,4	54,3
7	38,8	30,2	16,6	27,8	16,8	34,8	0	54,8	23,5	32,2
8	22,9	40,1	43,5	55,3	62,2	20,1	54,8	0	37,6	71,5
9	31,4	34,6	24,5	40,8	38,1	19,4	23,5	37,6	0	52,5
10	49,3	32,1	30,8	18,6	15,8	54,3	32,2	71,5	52,5	0

Problemin üç farklı çözümüne ait sonuçlar Şekil 12'de yer verilmiştir.

Şekil 12. Hipotetik Problemin Çözüm Sonuçları (Figure 12. Solution Results of the Hypothetical Problem)

Sonuçlar ele alındığında daha önceki uygulamalarımızın aksine GAMS bu örnekte başarılı olmuştur. Problem boyutunun büyük olduğu durumlarda GAMS'in bu problem çözümü için sezgisel algoritmalarla rekabet etmesi mümkün değildir.

4. Sonuç

Sezgisel yöntemler literatürde büyük çaplı problemlerin ele alınmasına olanak sağlamıştır. Şu an ki gerçek hayat problemlerin üzerinde matematiksel modellemelerin optimizasyonlarının belirli boyutlarda tıkandığı görülmektedir. Gerçek hayat problemleri ise bu yöntemlerdeki gelişmelere bağlı olarak GHP gibi daha karmaşık problemleri içermektedir. Bu çalışma da GHP için yeni bir matematiksel model önerilmiştir. Bu çalışmada incelenen GHP için bir matematiksel model oluşturulmuştur. Matematiksel örnek küçük bir test problemi üzerinde başarılı olmuştur fakat büyük örnek sayılarının daha hızlı çözülmesi amacı ile matematiksel modeli çözümleyecek bir genetik algoritma geliştirilmiştir. Problem bütünü ile ele alınıp incelendiğinde hırsızın çantaya ödediği kiraların büyüklüğü sebebi ile ikinci bir genetik algoritma geliştirilmiş olup bu algoritmada çantanın ağırlığı son düğümlerde artacak olup gezgin hırsızın düğümler arası süresinin minimize edilmesi hedeflenmiştir. Test problem kümesi içerisinde matematiksel model çözümü için geliştirilen genetik algoritmanın diğer geliştirilen genetik algoritmaya kıyasla daha başarısız olduğu görülmüştür. Algoritma başarısını değerlendirmek için literatürde yer alan kapsamlı GHP örnekleri içerisinden seçili örnekler kullanılmıştır. Örneklerin toplam kazançları literatürdeki örneklere önemli düzeyde yaklaştığı gözlemlenmiştir.

Kaynakça

Klamroth, K., Mostaghim, S., Naujoks, B., Poles, S., Purshouse, R., Rudolph, G., Ruzika, S., Sayın, S., Wiecek, M.M., Yao, X., 2017. Multiobjective optimization for interwoven systems. Journal of MultiCriteria Decision Analysis 24, 71–81.

Bonyadi, M.R., Michalewicz, Z., Wagner, M., Neumann, F., 2019. Evolutionary Computation for Multicomponent Problems: Opportunities and Future Directions. Springer. pp. 13–30.

Michalewicz Z (2012) Quo vadis, evolutionary computation? On a growing gap between theory and practice. In: Advances in computational intelligence. Lecture notes in computer science, vol. 7311. Springer, Berlin, pp 98–121

Bonyadi M, Michalewicz Z, Barone L (2013) The travelling thief problem: the first step in the transition from theoretical problems to realistic problems. In: Proceedings of the 2013 IEEE congress on evolutionary computation, Cancun, Mexico, pp 1037–1044

Polyakovskiy et al. (2014) Polyakovskiy S, Bonyadi MR, Wagner M, Michalewicz Z, Neumann F. A comprehensive benchmark set and heuristics for the traveling thief problem. In: Arnold DV, editor. Genetic and Evolutionary Computation Conference, GECCO '14. New York: ACM; 2014. pp. 477–484.

Reinelt (1991) Reinelt G. TSPLIB: a traveling salesman problem library. INFORMS Journal of Computing. 1991;3(4):376–384. doi: 10.1287/ijoc.3.4.376.