GEZGİN SATICI PROBLEMİ İÇİN SEZGİSEL METOTLARIN PERFORMANS ANALİZİ

Mustafa GERŞİL
Celal Bayar Üniversitesi

Asil ALKAYA
Celal Bayar Üniversitesi

ÖZET

Gezgin satıcı problemi, NP-Zor olan problemler sınıfına üye olup klasik tarzda tek amaçlı optimizasyonu hedeflemektedir. Burada amaç, şehirlerin oluşturduğu küme içerisinde, satıcının ziyaret edeceği şehir kümesini oluşturan tüm bu elemanları da içine dahil eden en kısa yolu (rotayı) bulmaktır. Bu çalışmada; gezgin satıcı problemi; klasik genetik algoritma, rasgele arama, tavlama benzetimi ve evrimsel algoritma metotları ile çözülmüştür. Her metoda ait rota mesafe değerleri, ilgili metodun parametreleri de göz önüne alınarak değerlendirilmiştir. Çaprazlama ve mutasyon, sezgiselliği sağlayan genetik operatörler olarak sezgisel metotlar üzerinde oldukça etkili olmaktadır. Problemi farklı büyüklüklerde ele alarak; klasik ve melez sezgisel metotların performans değerleri karşılaştırılmıştır.

Anahtar Sözcükler: Gezgin Satıcı Problemi, Evrimsel Algoritma, Tavlama Benzetimi, Rasgele Arama Algoritması

PERFORMANCE ANALYSIS OF HEURISTIC METHODS FOR TRAVELING SALESMAN PROBLEM

ABSTRACT

Traveling salesman problem is a candidate of the class of NP-hard problems that aims one goal optimization in a classical style. The goal is, in a set of cities gathered, to find the shortest path (route) which the salesman visits all the members of the set of the cities. In this study, traveling salesman problem is solved by the methods; classical genetic algorithm, random search, simulated annealing and evolutionary algorithm. Route distance values of each method are evaluated related to the method parameters that has been taken into consideration. Genetic operators; crossover and mutation, are quite effective on heuristic methods that gain being heuristic by these. The problem is determined for different sizes as to compare the performance values of classical and hybrid heuristic methods.

Keywords: Traveling Salesman Problem, Evolutionary Algorithm, Simulated Annealing, Random Search Algorithm

1. GİRİŞ

Gezgin satıcı problemi (GSP) verilen N düğüm (şehir) için, her düğüme bir kez uğramak şartıyla tekrar başlangıç düğümüne geri dönen en kısa (en az maliyetli) rotayı bulma problemidir (Potvin, 1996). Tanımlaması son derece kolay fakat çözümü NP-Zor bir problemdir. Bir eniyileme problemi olan bu problemde bir çizge üzerine yerleştirilmiş noktalar ve aralarındaki maliyetler göz önüne alınarak her düğüme yalnız bir kere uğramak şartıyla en uygun maliyetle çizgedeki tüm düğümlerin dolaşılması olarak tanımlanabilir. Bu problemin çözümü bir Hamilton Döngüsü olarak da görülebilir (Kalaycı, 2006).

GSP'yi matematiksel olarak iki şekilde tanımlayabiliriz: çizge problemi olarak ve permütasyon problemi olarak. Eğer GSP' yi çizge problemi olarak ifade edersek; G=(V,E) çizgesi ve bu çizgede tanımlanmış tüm Hamilton döngüleri F ile temsil edilmiş olsun. Her $e \in E$ için daha önceden belirtilmiş bir ağırlık değeri vardır. İşte GSP bu G çizelgesinde en kısa maliyetle, tüm düğümlere uğrayarak, bir tur (Hamilton döngüsü) elde etmeyi amaçlayan bir problemdir (Gutin ve Punnen, 2002).

2. GSP'DE SEZGİSEL ALGORİTMALAR

Gezgin satıcı problemi (GSP) yapay zeka ve eniyileme alanında en çok araştırılan ve çözümler üretilen, algoritmalar geliştirilen bir problemdir. Teorik bilgisayar bilimleri ve işletimsel araştırmada ("operational research") kombinasyonel eniyileme problemidir. Çok farklı eniyileme yöntemleri, yapay zeka teknikleri bu probleme yönelik olarak geliştirilmiştir. Gezgin satıcı problemi alanında kullanılan önemli yöntemlerden biri de genetik algoritmalardır (GA). Sezgisel bir yöntem olan genetik algoritmalar açısından GSP bir karşılaştırma ve değerlendirme ölçütü işlevi de görmektedir. Ayrıca sadece GA'ların değil, diğer tüm yeni algoritmaların (karınca sistemi, evrimsel yöntemler, sinir ağları, tabu arama, benzetimli tavlama, açgözlü aramalar, vb.) karşılaştırılması, değerlendirilmesi ve ölçülmesi için de kullanılmaktadır (Uğur, 2008) (Tsai vd., 2004). Bu sezgisel yöntemler makul bir sürede iyi sonuçlar sağlamaktadırlar.]Yapılan çalışmada yaklaşık sonuçlar üreten bazı yerel arama metotlarını incelenmiştir (Johnson vd., 1997). GSP'yi çözmek için önerilen algoritmalar iki başlık altında toplanabilir (Potvin, 1996): Kesin Algoritmalar ("Exact algorithms") ve Sezgisel Algoritmalar ("Heuristics algorithms").

Kesin Algoritmalar: Bu algoritmalar, genellikle, GSP'nin tamsayılı lineer programlama formülünden türetilen yaklaşımlardır. Fakat bu algoritmalar hesaplanabüirlik açısından pahalıdırlar[1]. Bu yaklaşıma örnek olarak "Branch & Bound" algoritmasını örnek olarak verilebilir. İlk akla gelen yöntem her olasılığın denendiği yöntemdir. Diğer yöntemler olarak lineer programlama yöntemleri, dinamik programlama yöntemleri sayılabilir (Applegate vd., 2006) .

Sezgisel Algoritmalar: Kesin algoritmaların çalışması verimli olmayabilir. Bu durumda, ideal çözüme yakın bir çözümü, kesin algoritmaları kullanmadan da bulabiliriz. Pratikte sezgisel algoritmalar, kesin algoritmalara tercih edilmektedir. GSP' yi çözen sezgisel algoritmaları üçe ayırabiliriz: Tur oluşturan sezgiseller, Turu geliştiren sezgiseller ve bu iki yöntemin melez şekilde kullanıldığı sezgiseller (Potvin, 1996).

Tur oluşturan Sezgiseller: Tur oluşturan algoritmaların ortak özelliği, bir sonuç buldukları zaman, bu sonucu geliştirmek için uğraşmamalarıdır. Bu noktada algoritmaların çalışması sona erer. Bilinen tur oluşturan sezgisel algoritmalar şunlardır: En yakın komşu, Greedy, Ekleme Sezgiseli ve Christofides algoritmalarıdır. Bu algoritmaların en optimum değeri %10-15 arasıdır (Nilsson, 2003).

***Turu geliştiren Sezgiseller*:** Bu algoritmalar turu geliştirmeyi amaçlar. Bu algoritmalara örnek olarak 2-opt, 3-opt ve Lin-Kernighan gibi yerel eniyileme ("optimization") algoritmalarının yanında; tabu araması, genetik algoritmalar, benzetim tavlama ve karınca kolonisi algoritması gibi yapay zeka yöntemlerini de örnek olarak verebiliriz.

Melez Yöntemler: Hem tur oluşturma hem de tur geliştirme sezgisellerini bir arada kullanan algoritmalardır. En başarılı sonuçlar melez yöntemlerden elde edilmektedir (Potvin, 1996) .

GSP'yi çözmek üzere en sık kullanılan evrimsel hesaplama yöntemi genetik algoritmalardır. Bir çok NP-tam problem için GA'nın oldukça başarılı sezgisel bir yöntem olduğu ispatlanmıştır (Uğur, 2008)

Sezgisel yöntemlerin bulduğu çözümlerin kalitesi ve hesaplama süresi, yöntemlerde bulunan temel kavramların probleme uygulanış biçimlerine bağlıdır. Bir sezgisel algoritmayı bir probleme başarılı bir şekilde adapte edebilmek için bazı prensiplerin izlenmesi gerekmektedir. Lokal komşuluk tabanlı sezgisel yöntemlerin bir kombinatoryal probleme uygulanmasında aşağıdaki faktörler göz önüne alınmalıdır (Hertz ve Widmer, 2004):

- Hazırlanan algoritmada başlangıç çözümünden komşuluk yapısı ile, kolay bir şekilde yeni çözümler üretilebilmelidir.
- Bazı durumlarda kısıtlarda yapılabilecek ihlallere göz yumularak, çözüm uzayında lokal noktalara takılmayıp daha geniş bir alanda arama sağlanmalıdır.

Populasyon tabanlı sezgisel algoritmalarda ise modelleme önemli bir husus olup, yöntemi bir kombinatoryal optimizasyon problemine başarılı bir şekilde uygulamak için aşağıdaki prensiplere dikkat edilmelidir:

- Problem için uygun bir teknik seçilerek iki çözüm ile yeni çözümler oluşturulurken ilgili temel bilgiler yeni çözümlere aktarılmalıdır.
- İki çözüm birleştirilerek oluşturulan yeni çözüm, ebeveyn çözümlerden tamamen farklı olmamalı, ortak özellikler içermelidir.
- Populasyon içerisinde çözümler arama uzayının farklı bölgelerinde olmalı, diğer bir deyişle populasyonda farklılık sağlanmalıdır. Ayrıca başlangıç populasyonu buna göre oluşturmalı ve populasyonda çözümlerin yer değiştirme işlemlerinde bu durum göz önüne alınmalıdır. (Blum ve Roli, 2003).

Literatürde bölgesel optimizasyon yöntemleri olarak da geçen klasik sezgisel yöntemlerde çözüm uzayında arama, belirlenen komşuluk yapısı ile daha iyi bir komşu çözüm bulunamadığı durumda sonlandırılmaktadır. Bu sebeple bu yöntemler lokal minimum noktalara takılmakta ve arama stratejisi kör bir şekilde uygulanmaktadır. Metasezgisel yöntemler ise lokal minimum noktalardan kurtulmak için daha kötü çözümlerin de kabul edildiği global optimizasyon yöntemleridir. Metasezgisel yöntemlerde arama, çözüm uzayının en umut verici noktalarında yapılmakta ve süreç, algoritma içine gömülü bir arama stratejisiyle yönetilmektedir. Bu yöntemlerin en büyük dezavantajı ise durdurma kriterinin doğal olarak algoritma içinde bulunmayışı, başka bir deyişle algoritmanın ne zaman duracağını bilmemesidir (Breedam, 2001).

Sezgisel yöntemlerde diğer klasik yöntemlere göre daha kesin çözüm bulunmasına ve yeni kısıtların algoritmaya kolay bir şekilde entegre edilebilmesine rağmen algoritmaların yazılım diline kodlanması karmaşık ve çözüm süreleri oldukça uzun olmaktadır. Metasezgisel algoritmalar çözüm uzayında arama sırasında kılavuz olarak aşağıda kullandıkları stratejiler ile optimum çözüme yakın sonuçlar üretmektedir (Tarantilis vd., 2004):

- *Çeşitlendirme Stratejisi*: Metasezgisel yöntemler genellikle iyi bir başlangıç çözümü ile algoritmaya başladıkları ve algoritma süresince belirli bir koşul koyup kötü komşu çözümleri eledikleri için çözüm uzayında iyi sonuç vermeyen çözüm bölgelerini atlamaktadır. Metasezgisel algoritmalar bu sayede kötü çözümler ile süre harcamaz. Özellikle populasyon tabanlı algoritmalar, iyi çözüm bölgeleri içinde yer alan çözümler ile ilgilenmek durumundadır.
- *Yoğunlaştırma Stratejisi*: Bu strateji belirli bir çözüm bölgesinde derin araştırma yapılarak o bölgede optimum noktanın bulunması için komşu çözümlerin taranması sürecini kapsamaktadır. Lokal komşuluk arama algoritmalarının temelinde yatan strateji bu olup populasyon tabanlı algoritmalarda ise genellikle saf olarak bu strateji kullanılmamakta, diğer yöntemlerle melez hale getirilerek algoritmaya dahil edilmektedir.

2.1 Rasgele Arama Algoritması

Rasgele arama, belki de en basit arama işlemidir. Bir başlangıç arama noktasından veya başlangıç noktalarının kümesinden başlayan arama işlemi, arama uzayında rasgele noktaları araştırır ve kabul edilebilir bir çözüme ulaşılıncaya veya maksimum iterasyon sayısı ulaşıncaya kadar devam eder. Rasgele aramayı gerçekleştirmek son derece basit olmakla beraber, verimsiz olabilir. Uygun çözüm elde edinceye kadar geçen zaman çok uzun olabilir. Rasgele araştırma için bir algoritma çalışması ve işleyişi aşağıda sunulmuştur.

Adım 1. N başlangıç arama noktaları kümesini seç. $C_g = \{C_{g,n} = | n=1,2,\ldots,N\}$.
Burada, $C_{g,n}$ I değişkenlerinin vektörü ve g=0 dır. Her bir $C_{g,n}$ elemanı, U(min,max) değişken değerlerin sınırı olmak üzere, verilen aralıklarda üretilir.

Adım 2. Her bir $C_{g,n}$ vektörünün ("uygunluk") $F(C_{g,n})$ doğruluğunu değerlendir.

Adım 3. En iyi noktayı bul $C_{g,best} = \min\{ F(C_{g,n}) \}$

Adım 4. if $C_{g,best} < C_{best}$ then C best = $C_{g,best}$ C_{best} tümünün en iyi çözümü

Adım 5. if C *best* kabul edilebilir bir çözüm ise veya maksimum iterasyon sayısı aşılmış ise o zaman "dur" ve çözüm olarak C_{best} "dön"

Adım 6. Her bir $C_{g,n}$, $\Delta C_{g,n}$ ile karıştır. Burada, $\Delta C_{g,n} \approx N(0,\sigma^2)$ ve σ^2 li küçük bir değişimi ifade eder.

Adım 7. g = g + 1 artır ve adım 2 ye git.

2.2 Tavlama Benzetimi

Tavlama benzetimi (TB), bilinen en eski metasezgisel algoritma olup, lokal optimum noktalara takılmayı önleyici, belirgin bir stratejiye sahiptir. Algoritmanın ana mantığında, arama uzayında lokal noktalardan kurtulmak için mevcut çözümden daha kötü çözümlerin belirli bir olasılıkla kabul edilmesi yatar. Bu olasılık değeri iki çözümün amaç fonksiyon değerleri arasındaki farkla ve sıcaklık değeri ile hesaplanır. Kötü çözümlerin kabul edilme olasılığı arama süresince düşürülür ve bu düşürme işlemi de metallerin ve camların tavlama prosesine benzemektedir (Blum ve Roli, 2003).

Burada metaldeki atomların durumları optimizasyon probleminin muhtemel çözümlerine ve durumların enerjileri de çözümlere ait amaç fonksiyon değerlerine karşılık gelmektedir. Hızlı soğutma işlemi ise bölgesel optimizasyon işlemine tekabül etmektedir. Dış sıcaklık sıfır olduğunda daha yüksek enerjili bir duruma geçiş mümkün olmaz. Böylece bölgesel optimizasyondaki gibi yukarı doğru hareketler yasaklanır ve araştırma bir bölgesel minimaya takılı kalır. Bu işlemde sıcaklık (T) çeşitli seviyeler boyunca yavaşça düşürülür. Enerji seviyesinden uzaklaşmamayı sağlamak için mevcut sıcaklığı belirli bir süre muhafaza etmek suretiyle ve sıfır dereceye yaklaşıncaya kadar bu işlemlerin tekrarlanması bölgesel optimallikten kaçışı sağlayabilmektedir (Karaboğa, 2004).

2.3 Tavlama Benzetimi İşlem Adımları

Güvenilir sezgisel bir araştırma algoritması başlangıç noktasına bağımlılığı az olan algoritmadır. TB'de çözüm uzayı taranırken yukarı doğru hareketler yani amaç değerinin aleyhine hareketler, değişen bir olasılık tabanlı fonksiyon ile kontrol edilir. TB genel olarak iteratif bir geliştirme algoritmasıdır. Standart bir TB algoritmasının temel adımları aşağıda verilmektedir (Breedam, 2001):

Adım 1. Rasgele olarak bir başlangıç çözümü üret ve en iyi çözüm S olarak ata. Ayrıca t iterasyon indeksini 0 olarak ata.

Adım 2. Bir başlangıç sıcaklık değeri T_B belirle ve mevcut sıcaklık değeri $T_0 = T_B$ olarak ata.

Adım 3. En iyi çözümden hareketle rasgele komşu bir çözüm $S^1 \in N(S)$ oluştur.

Adım 4. Üretilen S^1 çözümüyle S çözümünün amaç fonksiyonu değerleri arasındaki farkı hesapla ($\partial = C(S^1) - C(S)$)

Adım 5. Eğer S^1, S'den daha iyi ($\partial < 0$) ise S çözümüne S^1 çözümünü ata. S^1, S'den daha kötü fakat mevcut T_t sıcaklığında ($e^{-\partial/T} > \theta$ sağlanıyorsa (θ, 0 ile 1 arasında rasgele üretilmiş bir sayıdır) S çözümü ile S^1 çözümünü yer değiştir. Yoksa S'i mevcut çözüm olarak muhafaza et.

Adım 6. T sıcaklığını aşağıdaki formüllere göre değişir..
$T_t = R \cdot T_{t-1}$ $(0 < R < 1)$
$T_t = t / (1 + \beta t)$ (β uygun küçük bir değerdir)

Adım 7. Durdurma kriteri sağlanıyorsa araştırmayı durdur, aksi halde iterasyon indeksi t'yi bir artırarak üçüncü adıma git.

2.3. Genetik Algoritmalar

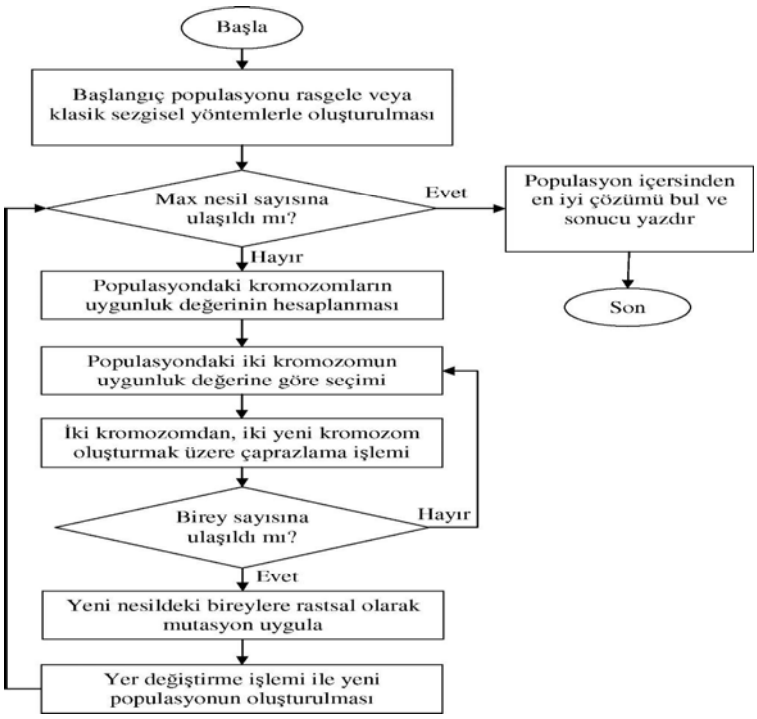
Genetik Algoritmalar doğadaki evrimsel süreçlerini model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, GA'nın yardımıyla daha kolay ve hızlı olarak çözülebilmektedir. Şekil 1'de görüldüğü gibi, genel anlamıyla GA bir araştırma konusu olup model haline getirilmiş neden-sonuç işleminin tersine, rasgele örnekleme olgusu altında modellenmiştir. Kontrol edilerek onaylanan kod bilgisi organizma olarak adlandırılan aday çözümler içerisinde saklanmıştır. Organizmalar ise populasyon olarak adlandırılıp grup halinde yer almışlardır. Genetik Algoritmalar ile ilgili temel kavramlar aşağıda kısaca açıklanmaktadır (Eren, 2002):

Gen: Kalıtsal molekülde bulunan ve organizmanın karakterlerinin tayininde rol oynayan kalıtsal birimlere denir. Yapay sistemlerde gen, kendi başına anlamlı bilgi taşıyan en küçük birim olarak alınır.

Kromozom: Birden fazla genin belirli bir düzenle bir araya gelerek oluşturduğu diziye denir. Kromozomlar alternatif aday çözümleri gösterir.

GA, populasyon tabanlı algoritma olduğundan özellikle gerçek hayatta çözüm üretilmesinin gerekli olduğu problem uygulamalarında önemli bir dezavantaja sahiptir. Bir algoritmanın belirli süre içerisinde çözüm geliştirme işlemini tamamlamış olması gerekir. Bu işlemi tamamlamak için populasyon tabanlı algoritmalar, lokal arama algoritmaları ile karşılaştırıldığında nispeten daha uzun süreye ihtiyaç duyarlar. Süreyi kısaltmak için az populasyonla çalışabilen ve iyi performansa sahip GA'nın geliştirilmesi önemli hale gelmiştir. Bunun yanında GA'nın bölgesel yakınsama hızının çok iyi olmaması bir diğer dezavantajıdır. Bu eksikliğini kapatmak amacıyla bazı araştırmacılar standart GA ile klasik komşuluk arama algoritmaları birleştirmişlerdir. Literatürde bu tür GA 'ya Melez Genetik Algoritma ismi verilmektedir (Karaboğa, 2004). GA'nın çözüm kalitesini artırıp hesaplama süresini düşürmek amacıyla literatürde eşzamanlı paralel olarak çalışan melez GA yöntemlerine sıkça rastlanmaktadır (Preux ve Talbi, 1999).

Populasyon: Kromozomlardan oluşan topluluğa denir. Populasyon, geçerli alternatif çözüm kümesidir. Populasyondaki (kromozom) birey sayısı genelde sabit tutulur. GA' da populasyondaki birey sayısı ile ilgili genel bir kural yoktur. Populasyondaki kromozom sayısı artıkça çözüme ulaşma süresi (iterasyon sayısı) azalır.



Şekil 1. Genetik Algoritma Akış Şeması

2.4. Evrimsel Algoritmalar

Geleneksel arama metotları, probleme bir çözüm adayı önerir ve onu değiştirerek daha iyi çözümler elde etmeye çalışır. Aksine evrimsel algoritmalar, bir çözüm adayları populasyonu oluşturur ve bu populasyon zamanla evrimlesir. Bir adayın çözüme ne kadar yakın olduğu, uygulamaya bağlı bir fonksiyondur. Bir çözüm adayı bir parametreler toplulugunu, bir kuralı, bir kurallar grubunu veya agac yapısında bir bilgisayar programını temsil edebilir. Hepsinde de, algoritma her adayın ne kadar güçlü olduğunu hesaplar ve buna gore bir sonraki neslin ebevynleri olacak ya da yok olacak bireyleri belirler. Daha sonra, makul bir yeni nesil olusturmak icin ebeveynlere genetik arama islemcilerini (çaprazlama ve mutasyon) uygular. Bu dongu her defasında daha guclu bireyler oluşturarak tekrarlanır.

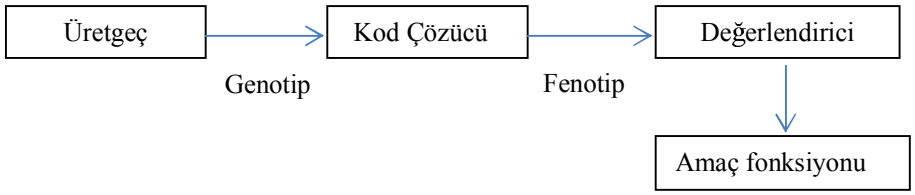
Belirli bir probleme evrimsel algoritma uygulayabilmek için, kullanıcı, aday çözümler için bir temsil sekli, bir gucluluk hesaplama yontemi ve genetik arama islemcilerini belirlemelidir. Cogu zaman, bunları belirlemek buyuk bir caba gerektirir. Mutasyon ve çaprazlama için tanımlanan işlemcilerle kullanılan temsil şekli uyum içinde olmalıdır.

Evrimsel algoritmalar, çözüm adayları içinde zeki bir rastgele arama yapar. Zor problemlerde verimli olmasına ragmen; analiz, basit sezgisel arama veya ayrıntılı numaralama yontemleriyle cozulebilen kolay problemlerde seçilmemelidir.

Değişken cezalarla alternatif çözümlerin de populasyonda kalmasını saglamalıdır. Tek çözümün pozitif uygunluk değerinin olduğu bir sonuç popülasyonu iyi değildir.

3. UYGULAMA

GSP, bir optimizasyon problemi olarak ele alındığında, şekil 2'de ilişkilendirildiği gibi, problemin çözümü için üretgeç (creator), kod çözücü (decoder) ve değerlendirici (evaluator) arayüzlerine ihtiyaç duyar.



Şekil 2. GSP Çözümünde Aşamalar

Genotip, GSP içerisinde rasgele şekilde belirlenmiş şehirleri permütasyona tabi tutarak oluşturulmuş olan listelerden oluşur. Oluşturulan genotipin fenotipe dönüştürülmesi için kod çözücü kullanılır. Fenotip, şehirlerin sıralanmış olarak değerlerini içeren rotayı belirlemiş olur. Oluşturulan rota üzerinden, her bir şehir arasındaki uzakliık toplanarak, elde edilen değer amaç fonksiyonunda kaydedilir.

Uygulamada yukarıda ifade edilen Rasgele Arama Algoritması, Benzetim Tavlaması, genetik algoritma ile melezleştirilmiş benzetim tavlaması ve evrimsel akgoritma farklı problem büyüklüklerine göre (ele alınıp karşılaştırılmıştır. Elde edilen sonuçlar; 500 şehir için Tablo 1'de, 1000 şehir için Tablo 2'de, 2500 şehir için Tablo 3'te ve 5000 şehir için Tablo 4'te gösterilmiştir. Her sezgisel yöntemin, iterasyonunu sona erdirebilmesi için ele alınan değer, önceki amaç fonksiyonu değeri ile sonraki amaç fonksiyonu değeri arasındaki değişim iterasyon sayısı göz önüne alınarak $1*10^{-5}$ değerinden az ise, sezgisel yöntem durdurulur. Aksi taktirde, önceden belirlene iterasyon sayısı aşılarak hedef değere ulaşılmaya çalışılur. Önceki çalışmalardan elde edilen deneyim ile, her bir sezgisel yöntem için gereğinden fazla iterasyon sayısı göz önüne alınmıştır. Amaç fonksiyonu değerleri 10 replikasyon sonuç değerlerinin ortalamasdır. Bu çalışmada, evrimsel algoritmanın klasik ve melez sezgisel yöntemlere nazaran daha başarılı olduğu ifade edilecektir.

3.1.Yöntem parametreleri:

- *Rasgele Arama Algoritması*
  İterasyon sayısı:25000
  Yığın büyüklüğü: 25

- *Benzetim tavlaması (BT)*
  İterasyon sayısı:200000
- *BT + genetik algoritma:*
  İterasyon sayısı:200000

Genetik Algoritma'da çaprazlama yöntemi olarak "Uniform Çaprazlama" tekniğini kullanılmıştır. Çaprazlanacak iki bireyin herbiri, çocuk bireye genini aktarmak için 0.5 şansa sahiptir. Mutasyon işleminde ise bir kromozomun her bir geni 0.5 olasılıkla mutasyona uğrayabilir. Seçilim işleminde ise çocuk bireyler ile daha önceki populasyonun birleştiriliyor ve bu birleşim sıralanarak populasyon boyutu kadar birey alınıp yeni populasyon olarak genetik algoritmaya dahil edilmektedir.

- *Evrimsel Algoritma:*
  İterasyon sayısı:100000
  Başlangıç populasyon büyüklüğü: 100
  Ebeveyn birey sayısı:25
  Çocuk birey sayısı :25
  Çaprazlama: 0.95
  Mutasyon: 0.1 (Adaptif)

Bireyler gerçekvektör (0,1) aralığında normalize edilmiş olarak rasgele elde edilir ve başlangıç populasyonu oluşturularak, her iterasyonda, 25 ebeveynden 25 çocuk birey mutasyon operatörü aracılığı ile oluşturulur. En kötü bireyler elitizim ile çözüm kümesinden ayrıştırılır.

Tablo 1.Şehir Sayısı (Problem Büyüklüğü) 500 Olan GSP (10 Replikasyon Ortalaması)

Sezgisel Metot	İterasyon Sayısı	En Küçük Toplam Mesafe (Amaç Fonksiyonu)
Evrimsel Algoritma	100000	1934.332
BT + Genetik Algoritma	200000	3175.572
Benzetim Tavlaması	200000	3335.8054
Rasgele Arama Algoritması	25000	24080.552

Tablo 2. Şehir Sayısı (Problem Büyüklüğü) 1000 Olan GSP (10 Replikasyon Ortalaması)

Sezgisel Metot	İterasyon Sayısı	En Küçük Toplam Mesafe (Amaç Fonksiyonu)
Evrimsel Algoritma	100000	3626.365
BT+Genetik Algoritma	200000	7466.879
Benzetim Tavlaması	200000	7626.596
Rasgele Arama Algoritması	25000	49248.402

Tablo 3. Şehir Sayısı (Problem Büyüklüğü) 2500 Olan GSP (10 Replikasyon Ortalaması)

Sezgisel Metot	İterasyon Sayısı	En Küçük Toplam Mesafe (Amaç Fonksiyonu)
Evrimsel Algoritma	100000	16395.547
BT+Genetik Algoritma	200000	24604.929
Benzetim Tavlaması	200000	24657.0696
Rasgele Arama Algoritması	25000	126668.34

Tablo 4.Şehir Sayısı (Problem Büyüklüğü) 5000 Olan GSP (10 Replikasyon Ortalaması)

Sezgisel Metot	İterasyon Sayısı	En Küçük Toplam Mesafe (Amaç Fonksiyonu)
Evrimsel Algoritma	100000	54137.746
BT+Genetik Algoritma	200000	62990.087
Benzetim Tavlaması	200000	63539.606
Rasgele Arama Algoritması	25000	255593.265

4. SONUÇ

Rasgele arama algoritması, daha basit yapıda olması sebebiyle, diğer sezgisel metodlara kıyasla en kötü performansı sergileyen sonuçlar elde etmiştir. BT'na genetik algoritma uygulanarak oluşturulan melez sezgisel model, klasik BT metodundan %0.2 ile %4.8 arasında daha iyi çözüm değerleri üretmiştir. Buradan

anlaşılacağı üzere, klasik sezgisel yöntemlere genetik algoritma uygulanması ile oluşturulacak melez yapı daha iyi sonuçlar verecektir. Oluşturulan evrimsel algoritma, farklı problem büyüklüklerinin tümünde en iyi sonucu vermiştir. Ancak, her farklı problem büyüklüğü incelendiğinde, evrimsel algoritmaya en yakın performans gösteren genetik algoritma içerikli BT sezgisel metoduna kıyasla; 500 şehir sayılı GSP'de %39.08, 1000 şehir sayılı GSP'de %51.43, 2500 şehir sayılı GSP'de %33.36, 5000 şehir sayılı GSP'de %13.77 iyileştirme sağlamıştır. GSP büyüdükçe evrimsel algorritmanın çözüm performansında düşüş yaşanmaktadır. Zorluk derecesinin arttığı durumlarda, başlangıç populasyon büyüklüğü, ebeveyn ve çocuk birey sayısı seçimi evrimsel algoritmanın performansına önemli ölçüde etki etmektedir.

KAYNAKÇA

Applegate, D. L.; Bixby, R. E.; Chvátal, V.; Cook, W. J. (2006), The Traveling Salesman Problem: A Computational Study, Princeton University Press

Blum, C., Roli, A., (2003), "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM Computing Surveys, 35, 3, 268-308

Breedam, A., V., (2001), "Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem", Computers & Operations Research, 28, 289-315

Eren, H., (2002), "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma(GA) İle Çözüm Performansının Artırılmasında Deney Tasarımı Uygulaması", İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Yüksek Lisans Tezi, İstanbul

Gutin G., Punnen A.P. (Eds.), The Travelling Salesman Problem and its Variations, Kluwer Academic Publishers, Dordrecht, 2002.

Johnson D.S., McGeoch L.A., The traveling salesman problem: a case study in local optimization, in: E.H.L.Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, John Wiley & Sons, New York, 1997, pp.215–310.

Kalaycı, T.E. Yapay Zeka Teknikleri Kullanan Üç Boyutlu Grafik Yazılımları İçin "Extensible 3D" (X3D) İle Bir Altyapı Oluşturulması ve Gerçekleştirimi, Ege Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans Tezi, 2006.

Karaboğa, D., (2004), "Yapay Zeka Optimizasyon Algoritmaları", Atlas Yayın Dağıtım, Cağaloğlu, İstanbul

Potvin J.-Y., Genetic algorithms for the travelling salesman problem, forthcoming in Annals of Operations Research on "Metaheuristics in Combinatorial Optimization", eds. G.Laporte and I.H. Osman (1996).

Nilsson C., Heuristic Algorithms For Travelling Salesman Problem, Linköping University (2003), Erişim bağlantısı: http://www.ida.liu.se/~TDDB19/reports_2003/htsp.pdf

Preux, P., Talbi, E. G., (1999), "Towards Hybrid Evolutionary Algorithms", International Transactions in Operation Research, 6, 557-570

Tarantilis, C. D., Ioannou, G., Prastacos, G., (2004), "Advanced Vehicle Routing Algorithms For Complex Operations Management Problems", Journal of Food Engineering, Elsevier Ltd.

Hertz, A., Widmer,M., (2004), "Guidelines For The Use Of Meta-Heuristics In Combinatorial Optimization", European Journal of Operation Research, 151, 247-252

Uğur, A. Path planning on a cuboid using genetic algorithms. Inf. Sci. 178, 16 (Aug. 2008), 3275-3287. 2008.

Tsai C.-F., Tsai C.-W., Tseng C.-C., A new hybrid heuristic approach for solving large traveling salesman problem, Information Sciences 166 (1–4) (2004) 67–81.