

Siyasi Parti Mitinglerinin Gezgin Satıcı Problemi Yaklaşımı ile Analizi

Doç. Dr. İrfan ERTUĞRUL

Pamukkale Üniversitesi

İİBF İşletme Bölümü

iertugrul@pau.edu.tr

Arş. Gör. Abdullah ÖZÇİL

Pamukkale Üniversitesi

İİBF İşletme Bölümü

iertugrul@pau.edu.tr

Özet

Son yıllarda karmaşık optimizasyon ve araştırma problemlerinde doğal seçim sürecine dayalı evrim stratejileri kullanılmaktadır. Bu çalışmada evrim stratejileri kapsamındaki genetik algoritmalar konusunun temel bilgileri anlatılmıştır ve genetik algoritmalar yardımı ile Gezgin Satıcı Problemi ele alınmıştır. Gezgin satıcı problemi verilen birbirine bağlı şehir, düğüm vb. gibi noktalara ulaşımı ve başlangıç noktasına geri dönüşü ele alan kısıtlı en çok bilinen optimizasyon yöntemlerinden biridir. Gezgin satıcı problemlerine örnek oluşturabilecek siyasi partilerin mitinglerinin optimal şekilde planlaması amacıyla Travelling Salesman Problem (TSP) programı kullanılarak miting planlama analizi yapılmıştır. Analiz sonuçları ile elde edilebilecek maliyet ve zaman tasarrufundan bahsedilmiştir.

Anahtar Kelimeler: Genetik Algoritmalar, Gezgin Satıcı Problemi, Siyasi Parti Mitingleri.

Jel Sınıflandırması: C73, C87, D72.

The Analysis of Political Parties' Public Meetings with Travelling Salesman Problem Approach

Abstract

In recent years, solutions is sought by evolution strategies based on natural selection for complex optimization and research problems. This study discusses the basics of the topics genetic algorithm covered in the evolution strategies and Travelling Salesman Problem are tackled with the help of genetic algorithm. Traveling Salesman Problem handling of the connected point such as city, node and so on. transport and return to the starting point, is one of the most well-known restricted optimization methods. The public meeting planning could be an example of travelling salesman problem is analysed for optimality by using Travelling Salesman Problem (TSP) program. Analysis results is mentioned with cost and time savings can be obtained.

Keywords : Genetic Algorithms, Travelling Salesman Problem, Public Meeting of Political Parties.

JEL Classification: C73, C87, D72.

Giriş

Doğa, evrim süreci boyunca problem çözme ve optimizasyon için harika ve güçlü bir mekanizmaya sahiptir. Bilgisayar programlarındaki evrim ve doğal seçim gücünün benzetilmesi nedeniyle zor problemlerin çözümü doğal gibi görünmektedir. Doğal genetik sistemlerin ilkelerine dayalı, evrimsel algoritmalar (EA) karmaşık problemlerin optimizasyonu ve araştırılmasını gerçekleştirmek için geliştirilmiştir. Evrimsel algoritmaların önemli bileşenleri; Genetik Algoritmalar, Genetik Programlama ve Evrimsel Strateji'lerdir (Bandyopadhyay ve Pal, 2007: 19).

Holland doğal sistemlerin uyarlamalı sürecini açıklamaya çalışmış, doğal sistemlere dayalı yapay bir sistem olan Genetik Algoritmaları tasarlanmıştır (Holland, 1975). Evrimsel algoritmaların diğer önemli bileşenleri olan Genetik Programlama Fogel ve ark. (Fogel, Owens ve Walsh, 1966) tarafından önerilirken, Evrim Stratejileri ise Schwefel (Schwefel, 1993) tarafından önerilmiştir (Rekiek ve Delchambre, 31: 2006).

Genetik algoritmalar, Darwin'in evrim ve en uygun olanın hayatta kaldığı doğa kanunundan esinlenerek, üreme, çaprazlama ve mutasyon gibi operatörler kullanarak başarılı popülasyon çözümleri geliştiren, doğal evrimi taklit eden global bir arama işlemidir. Genetik Algoritmalar'da daha iyi nesiller oluşturmak için rastgele bilgi değişimi işlemi uygulanır (Adeli ve Sarma, 2006: 38).

Genetik algoritmalar diğer optimizasyon algoritmaları gibi optimizasyon, maliyet fonksiyonu ve maliyet gibi değişkenlerinin tanımlanmasıyla başlar ve optimal çözüme yakınlaşmanın test edilmesi ile biter. (Haupt ve Haupt, 2004: 28).

1. Genetik Algoritmalar

Genetik algoritmanın bugünkü olağan hali Goldberg tarafından tanımlanmıştır (Goldberg, 1989). Genetik algoritma doğal seçim ve genetik mekanizmasına dayalı stokastik bir araştırma sürecidir. Genetik algoritma geleneksel arama tekniklerinden farkı genel olarak, problemin sistem kısıtlarını ve/veya sınırlarını karşılayan popülasyondan rastgele çözümleri içeren bir başlangıç seti ile başlar. Popülasyondaki probleme çözüm sunan her birey, kromozom (veya birey) olarak adlandırılır. Kromozom genellikle bir ikili bit sembol dizesidir. Ardışık yineleme yoluyla evrimleşen kromozomlar nesiller olarak adlandırılır. Her kuşak boyunca, kromozomlar bazı uygunluk ölçümleri kullanılarak değerlendirilir. Bir sonraki nesli oluşturmak için, yeni yavru kromozomlar ya çaprazlama operatörü kullanılarak şimdiki neslin iki kromozomundan ya da mutasyon operatörü kullanılarak bir kromozomun değiştirilmesinden oluşturulur. Yeni nesil bazı ebeveyn ve çocukların uygunluk değerlerine göre seçim yapılarak, popülasyon büyüklüğünü sabit tutmak için diğerlerinin reddedilmesiyle oluşturulur. Uygun kromozomların seçilme olasılığı daha yüksektir. Birkaç

nesil sonra, algoritmalar probleme optimum veya ideal çözüm sunan en iyi kromozoma yakınsamaktadır (Gen, Cheng ve Lin, 2008: 1-2).

Genetik algoritmalar zor problemler (deterministik olmayan zor problemler gibi), makine öğrenimi ve de basit programların gelişimi için kullanılır. Ayrıca bazı sanat, müzik ve resim geliştirme alanlarında da kullanılır. Genetik algoritmanın bir kaç uygulama alanı (Sivanandam ve Deepa, 2008: 35); doğrusal olmayan dinamik sistemler (tahmin, veri analizi), robot yörünge planlaması, genetik programlama, strateji planlama, protein moleküllerinin şekillerinin bulunması, gezgin satıcı problemleri, görüntü oluşturma işlevleri, çeşitli alanlarda kontrol, dizayn ve zamanlama, yapay zeka öğrenme, sinyal işleme ve kombinasyonel optimizasyon olarak örnek gösterilebilir (Sivanandam ve Deepa, 2008: 35).

Günümüzde Genetik Algoritmalar ile ilgili birçok farklı alanda ve şekillerde yapılan uygulama çalışmaları bulunmaktadır. Kiraly ve Abonyi (2015), tedarik sistemi için Google Maps uygulaması ve genetik algoritma optimizasyonu kullanarak yeni bir mobil uygulama dizayn etmişlerdir. Satyhan, Ernest ve Cohen (2015), insansız hava aracının kontrolü ve görev planlaması için bulanık genetik algoritma kullanmışlardır. Tabatabaei, Asadian ve Boushehri (2014), olasılıksal gezgin satıcı problemi kullanarak kısa dönemlik elektrik güç yük tahmininde bulunmuşlardır. Borna ve Hashemi (2014) gezgin satıcı problemlerinin çözümü için ekstra mutasyon seviyesine sahip geliştirilmiş bir genetik algoritma yöntemi önermişlerdir. Pan ve Xia (2014) gezgin satıcı problemlerinde çapraz yolları belirlemek ve dağıtmak için yeni bir gezgin satıcı optimizasyonu geliştirmişlerdir. Kubalik ve Snizek (2014) gezgin satıcı problemlerinde dolaylı temsil, genişletilmiş yapıcı yakın komşu prosedürü, verimli geçit operatörü ve tekrar uyarlamalı strateji kavramları ile yeni bir evrimsel algoritma önermişlerdir. Ashokkumar ve Hebbal (2014) gezgin satıcı problemi kullanarak otomatikleştirilmiş bir depolama sistemi optimize etmeyi açıklamışlardır. Yılmaz, Doğan ve Koca (2015) optimum mayın imha mesafesi tespitinde 3 boyutlu gezgin satıcı problemi ile çözüm araştırmışlardır. Akar ve Şahingöz (2015) gezgin satıcı problemlerinin farklı bir modeli olan asimetrik gezgin satıcı problemini ele almışlardır. Çakır ve Yılmaz (2015) gezgin satıcı problemini paralel genetik algoritmalar kullanarak optimize etmeye çalışmışlardır.

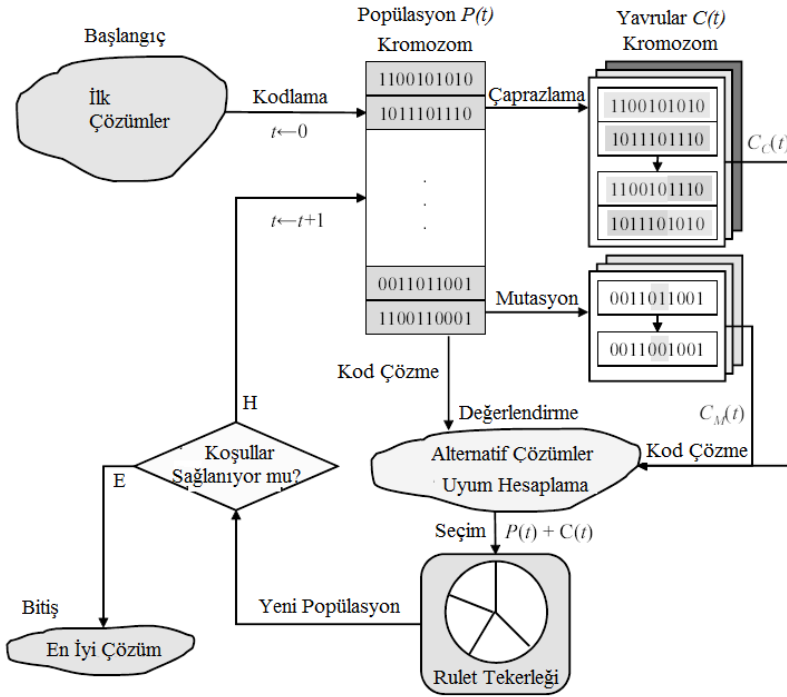
Genetik algoritmalar evrim paradigmalarından sonra serbest şekilde modellenmiş, popülasyon temelli araştırmaya dayalı, stokastik süreçlerdir (Chen, 2006: 6). Büyük çözüm uzay yapılarına sahip optimizasyon problemleri için cazip olan genetik algoritmanın üç özelliği; birincisi, bir başlama noktası yerine bir grup noktadan çözüm araştırılmaya başlanır. İkincisi sadece sayısal verimli algoritmalar ile sonuçlanan basit evrim fonksiyonları gerektirir. Son olarak, paralel işleme sürecinin kullanımı oldukça kolaydır çünkü popülasyondaki her kromozom veya dizi

diğerlerinden bağımsızdır ve bundan dolayı aynı anda işlenebilir (Adeli ve Sarma, 2006: 38).

Genellikle Genetik Algoritma beş temel kavramdan oluşur (Gen ve diğ., 2008: 2);

- Sorunun olası çözümlerin bir genetik gösterimi,
- Bir popülasyonu oluşturmak için bir yöntem (potansiyel çözümlerin bir başlangıç seti),
- Çözümleri uygunluk açısından değerlendiren bir değerlendirme (uygunluk) fonksiyonu,
- Yavruların genetik bileşimini değiştiren genetik operatörler (çaprazlama, mutasyon, seçim, vs.),
- Genetik algoritmanın kullandığı parametre değerleri (nüfus büyüklüğü, genetik operatörlerin uygulanma olasılıkları, vs.).

Genetik algoritmalar süreci Şekil 1'de gösterildiği gibi; seçim, çaprazlama, mutasyon gibi genetik operatörler kullanarak başlangıç çözüm uzayında optimal çözümü araştırır.



Kaynak: Gen ve diğ., 2008: 2

Şekil 1: Genetik Algoritmanın Genel Yapısı

Genetik algoritmalar normal optimizasyon ve arama yöntemlerinden dört şekilde farklıdır (Bandyopadhyay ve Pal, 2007: 21-22);

- Genetik algoritmalar, parametrelerin kendileri ile değil parametre setinin kodlanması ile çalışır.
- Genetik algoritmalar aynı anda sadece bir nokta ile değil, birden fazla nokta ile çalışır.
- Genetik algoritmalar çözünen bilgi kullanılan örnekleme yöntemiyle (kör arama) araştırılır.
- Genetik algoritmalar deterministik kurallar yerine stokastik operatörler kullanarak araştırır.

Tam olarak algoritmalarındaki kombinasyonel optimizasyon problemlerinin çözümü için sezgisel ve genel amaçlı sezgisel araştırmalar günümüzde yüksek derecede artmaktadır. Algoritma kullanmanın başlıca avantajı problem için global optimum çözümü bulmayı garantilemesidir fakat gerçek problemlerin kritik bir dezavantajı hem örnek büyüklüğüne göre yürütme süresinin üstel büyümesi, hem de gerçek olmayan kısıtların problemin çözümüne empoze edilmesidir (Alba ve Dorronsoro, 2008: 3).

Genetik Algoritmalar konusu içerisinde yer alan Gezgin Satıcı Problemi uygulaması ile ilgili temel kavramlar ve özellikleri;

- Başlangıç Popülasyonu

Kromozomların toplamına popülasyon denir. Bir Genetik Algoritma kromozom olarak sunulan potansiyel çözümlerin geniş bir popülasyonun oluşturulması ile başlar. Algoritma eski ve yeni kromozomlardan yeni popülasyon oluşturulması ve işlenmesi süreci ile devam eder. Bazı genetik algoritma uygulamalarında toplam popülasyon sabitken, popülasyon büyüklüğü çözüm uzayının doğasına bağlı olarak artabilir veya azalabilir (Cox, 2005: 350-351).

Popülasyon büyüklüğü, popülasyonda kaç kromozom olduğunu ifade eder. Popülasyon büyüklüğünün optimum değerlerden uzak olması algoritmanın performansını etkilemektedir. Nesildeki bireyler sayısı az ise uzay yeterince araştırılmadığından yerel optimum noktaya yakalanma olasılığı yüksektir. Bu durumda çaprazlama işlemi için yalnız birkaç olasılık olduğundan problem alanında sadece küçük bir kısım araştırılabilecektir. Bu sayı aşırı arttığında ise neslin evrimleşmesi oldukça uzun zaman içerisinde gerçekleştirildiğinden algoritma yavaşlayacaktır (Nabiyev, 2012: 610-611).

- Örnekleme Yöntemleri

Darwin doğal seçimi yada en uygunun hayatta kalmasını "olumlu bireysel farklılıkların ve varyasyonların korunması ve bunlardan zararlı olanlarının imhası" şeklinde tanımlamıştır. Seçim işleyen mekanizmada tek olduğu zaman, en iyi birey nihayetinde tamamen nüfusu devralmaya seçilir. Seçim

mekanizması bireylerin gelecek nesle aktarılacak genetik özelliklerinin tamamını ya da bir kısmını belirler. En yaygın olarak kullanılan seçim yöntemleri aşağıda değerlendirilmiştir (Rekiek ve Delchambre, 2006: 35);

○ Rulet Tekerleği Seçimi (Roulette Wheel Selection); popülasyondaki bireylerin uygunluk değerleri ile orantılı olacak şekilde çoğu bölmeyi karşılaştırarak eğilimli rulet oyunu metaforunu kullanır (Dreo, Petrowski, Siarry ve Taillard, 2006: 84). Rulet tekerleği tekniği, tekerlek üzerinde her yuvanın nüfusun bir bireyi ile eşleştirilmiş olduğu bir rulet çarkı gibi çalışır. Her yuvanın büyüklüğü bireylerin uygunluk değeri ile orantılıdır. Maksimizasyon problemleri paradigmaya "daha büyük slot daha büyük uygunluk" şeklinde direkt uygulanır.

○ Elitizm Modeli; İlk varyasyon elitizm olarak adlandırılır ve en iyi çözümü korumayı kapsar. Beklenen değer modeli olarak da adlandırılan elitizm modeli seçim mekanizmasının stokastik hatalarını azaltır (Rekiek ve Delchambre, 2006: 35).

○ Turnuva Seçimi; Turnuva seçimi popülasyondan rastgele seçilen bir kaç birey arasındaki kazananın (en uygun bireyin) seçilmesi şeklinde çalışır. Seçim sıklığı turnuva boyutu büyüklüğü değiştirilerek kolayca uyarlanabilir. Turnuva boyutu büyükse zayıf bireylerin seçilmek için küçük bir şans vardır.

○ Sıralama Seçimi; Sıralama yöntemi işlenmemiş amaç fonksiyonu değerlerini pozitif gerçek değerlere göre planlar ve her kromozomun hayatta kalma olasılığı bu değerlere göre belirlenir. Uygunluk sıralamasının iki amacı vardır; (1) kromozomların göreceli uygunluk sıralamaları arasındaki makul farklılıkları sürdürmek ve (2) gereksinimleri karşılayan bazı süper kromozomların rekabetini limitlemek ve daha sonra teşvik etmek için çok hızlı artışı önlemek (Gen ve diğ., 2008: 13).

○ Denge Durumu Seçimi; doğrusal seçim mekanizması kullanılarak seçilen birkaç adet bireye genetik operatörler uygulanır. Elde edilen diziler mevcut yığındaki uygunluk değeri düşüş olan bireylerle yer değiştirilir (Elmas, 2011: 393).

- **Kodlama**

Genetik algoritmanın diğer araştırma yöntemlerinden temel farkı parametreler yerine parametrelerin kodlanmış halinin kullanılıyor olmasıdır. Bu nedenle; Genetik algoritma uygulamasının ilk adımı problemi en iyi temsil edecek en uygun kodlama metodunu seçmektir. Genellikle genetik algoritma ikili kodlama yöntemi kullanır. Literatürde genetik algoritmanın ikili, sıralı, değer ve ağaç kodlama türleri vardır ve aynı zamanda farklı sembolik alfabeler kullanan belirli algoritma çalışmaları vardır. Kodlama yöntemi genetik algoritma performansı üzerinde büyük bir etkisi vardır ancak kodlama yöntemi probleme bağlı olduğundan herhangi birini en uygun yöntem olarak nitelendirmek imkansızdır. Michalewicz daha hızlı ancak belli bir sorun için sıralı sayıları kullanmayı göstermiştir. Bazı çalışmalarda

gri kodlama metodu ikili kodlamadan daha iyi sonuçlar vermiştir (Ünal, Ak, Topuz ve Erdal, 2013: 23).

Problem çözümleri kodlandıktan sonra, kötü çözümlerden iyi çözümleri ayırt etmek bir yöntem veya prosedür aşaması bulunur. Prosedür çözüm kalitesine karar verecek olan laboratuvar deneyi, bir saha testi, karmaşık bir bilgisayar simülasyonu veya bir insan olabilir. Genetik algoritmalar mümkün olduğunca kötü çözümlere kıyasla yüksek uygunluk atanan daha iyi çözümler bulmak için bu farklı modlar veya ölçümler ile çalışır. Doğada uygunluk, problemlerin çözümünde evrim rehberliğindeki genetik algoritmalara göre ayırıcı bir işaret koşulunu koymaktadır (Chen, 2006: 7).

- Uygunluk Fonksiyonu

Genetik algoritmalar amaç fonksiyonu kullanarak aday çözümlerin popülasyonu üzerinde çalışır. Kromozomların her birine uygunluk fonksiyonunun uygulanması çözüm performansının kalitesini ölçmeye imkan tanır. Uygun bir nesil, arama uzayındaki her noktaya uygunluk fonksiyonunun uygulanması ile n boyutlu (hiper yüzey) alandaki noktalar kümesidir. Bir fonksiyonu verimli bir şekilde optimize etmek için uygunluk fonksiyonu açıkça tanımlanmalı ve uygunluğu yüksek bireyler açıkça kalıcı olmalıdır. Sonuç olarak artan başarı dışında başka bir özellik arzulanıyorsa, Genetik Algoritmalar arzulanan yeteneği olan bireylerin oluşumuna direkt teşvik etmelidir (Rekiek ve Delchambre, 2006: 38).

Probleme çözümler sunan kodlama düzeni ile donatılmış ve kötü çözümlerden iyi çözümleri ayırt edebilen uygunluk fonksiyonu ile genetik algoritmalar çözüm uzayındaki bir nokta yerine kodlanmış çözüm popülasyonundan optimal bir çözüm aramaya başlar. Bireylerin ilk popülasyonu rastgele veya probleme ait bazı belirli bilgilerle oluşturulabilir. Yeni çözümler geliştirmek amacıyla genetik algoritmalar geçerli popülasyonun çözümlerine dayalı gelecek vaat eden çözümler oluşturmak için genetik operatörler kullanır (Chen, 2006: 7).

Genel olarak, üreme ve değiştirme için bireyin seçilme yeteneği onun uygunluk değerine bağlıdır. Bu nedenle seçilme operatörü her bireyin uygunluk değerine göre seçilme sayısını belirler (Dreo ve diğ., 2006: 81).

Genetik algoritmalarda bir bireyin uygunluğu; problem için amaç fonksiyonu değeridir. Uygunluğu hesaplamak için kromozomun ilk olarak tekrar kodlanması gerekir ve amaç fonksiyonu değerlendirilir. Uygunluk hem çözümün ne kadar iyi olduğunu hem de kromozomun optimalliğe yaklaşmasını nasıl karşıladığını da gösterir (Sivanandam ve Deepa, 2008: 41).

- Çaprazlama

Çaprazlama operatörü ebeveyn olarak seçilen iki kromozomun değişimi olarak tanımlanabilir. Bu kromozomlar bir bütün yapacak şekilde rastgele

seçilmiş bir noktadan kesilir. Genetik algoritmalarda genellikle çaprazlama noktası 1 tanedir ve ağaç gösterimi teoremi çaprazlama işlemi 1 noktadan olacak şekilde tasarlanmıştır (Ünal ve diğ., 2013: 26).

Çaprazlama noktası tek veya çok noktalı olarak belirlenebilir. Ayrıca problemin özelliğine göre; pozisyona dayalı, sıraya dayalı ve kısmi planlı çaprazlama türleri de vardır (Elmas, 2011: 394-395).

Çaprazlama işlemi, yeni kromozomların eskilerin parçalarından oluşması ve yenilerin daha iyi olması umuduyla yapılır. Bununla beraber, popülasyonun belli kısımlarını bırakmak bir sonraki jenerasyon için iyidir (Nabiyev, 2012: 610).

- Mutasyon

Mutasyon yavru oluşturan bir seri içinde bir veya daha fazla genin dönüşümüdür. Genellikle, mutasyon operatörü nadiren kullanılır. Son çalışmalar mutasyon operatörünün genetik algoritmalarda önemli bir rol oynadığını göstermektedir. Optimal mutasyon olasılığı oranı kromozomun uzunluğu ve sorunun çözüm alanı ile orantılıdır (Ünal ve diğ., 2013: 27).

Mutasyon, Genetik algoritmaların yerel ekstremuma takılmasını engellemek için gerçekleştirilir. Fakat çok sık kullanıldığında Genetik Algoritmalar, körüne aramaya dönüşebilir (Nabiyev, 2012: 610).

- Yeni Neslin Oluşturulması

Doğal evrime benzer şekilde, genetik algoritmada bireyler uygunluk değerlerine bağlı olarak bir nesilden diğerine hayatta kalır. En uygun bireyler korunur ve en uygun olanın hayatta kalması şeklinde adlandırılarak, ürer. Bu şekilde tüm popülasyonun uygunluğu mümkün olan en iyi çözümlere yakınsamaya başlar (Cox, 2005: 351).

- Tamir Operatörü

Tamir operatörü (düzenleyici algoritma); uygun olmayan dizileri uygun duruma getirmek için özel olarak tasarlanan algoritmadır. Problemin özelliğine göre geliştirilen bu algoritmada genetik operatörlerin uygulanmasından sonra diziden mevcut bilgilerin yok olması veya fazladan istenmeyen bilgilerin gelmesi çizelgeleme problemlerinde istenmeyen bir durumdur. Bu sorunu ortadan kaldırmak için başlangıç dizisinin bilgilerine sağdık kalınarak özel bir algoritma geliştirilir (Elmas, 2011: 398).

- Durdurma Kriteri

Çoğalma, çaprazlama ve mutasyon işlemlerinden sonra yeni bir kuşak oluşmaktadır. Yeni kuşağın uyum değeri hesaplanır. Tüm bu işlemler bir döngü içinde yapılır. Gelecek döngüler için de bu işlemler devam edecektir. Eğer bir durdurma kriteri belirlenmez ise evrimsel süreç sonsuza dek devam eder. Literatürde genetik algoritmaların durdurma kriterleriyle ilgili çeşitli yöntemler yer almaktadır. Literatürde yer alan en yaygın durdurma kriterleri;

○ Hesaplama Zamanı Kriteri; Bu yöntemde önceden bir hesaplama zamanı veya döngü sayısı belirlenmekte ve genetik algoritma bu zaman dolduğunda veya belirlenen döngü sayısına ulaşıldığında durdurulmaktadır. Böyle bir yöntemin bir takım sakıncaları mevcuttur. Çünkü belirlenen döngü sayısı ile erken bir duruş olabilmektedir. Döngü devam ettirildiğinde iyileşmenin görülebileceği kuşaklar olabilir. Tersine bir durum söz konusu olabilir. Belirlenen döngü sayısı gereğinden fazla olabilir ve bu da hesaplama zamanını arttıracaktır.

○ Optimizasyon Hedefi Kriteri: Bu yöntem de bir önceki yöntemle benzerdir. Önceden ulaşılmaması istenen amaç fonksiyonu değeri bilinmektedir. Genetik algoritmada uyum değeri bu değere ulaştığında algoritma durdurulmaktadır.

○ Minimum İyileşme Kriteri: Bu yöntem, kuşaklar boyunca iyileşme miktarları arasındaki farkların alınarak iyileşmenin giderek azaldığı bir döngüde algoritmanın durdurulmasını içermektedir. Uyum fonksiyonunun çözüm sırasındaki seyri izlenerek algoritmanın ne zaman durdurulacağı tespit edilmektedir. Genetik algoritma problemlerinde genellikle eldeki en iyi çözüm önce hızlı sonra da yavaş yavaş artış göstermektedir. Değerdeki iyileşme hızının giderek azalması ve sifira yaklaşması, artık daha fazla iyileşme beklenmemesi gerektiğini gösterebilir. Algoritma, çözüme harcanacak zaman ile çözümden beklenen kalite arasında bir denge kurularak durdurulur (Taşkın ve Emel, 2009: 63-64).

2. Uygulama

Gezgin satıcı problemini şehirlerarası ulaşım problemlerine uygulamak oldukça kolaydır. Gezgin satıcı problemi; birbirleriyle arasındaki uzaklıkların maliyeti verilen sınırlı sayıda şehir ve amacı en ucuz yoldan tüm şehirleri tam olarak ziyaret etmek ve başlangıç noktasına dönmek olan problemleri ele alır. Genellikle seyahat masrafları simetrik belirlenir (Affenzeller, Winkler, Wagner ve Beham, 2009: 121).

Gezgin satıcı problemi (GSP) kombinasyonel optimizasyon problemlerinden en çok bilinenlerden biridir. Ziyaret edilecek şehirler l 'den n 'e kadar ve aralarındaki mesafeler $D_{i,j} \in \mathbb{N}$ ($i, j \in 1, 2, \dots, n$ olmak üzere) şeklinde tanımlanabilir. Satıcının her şehri ziyaret etmesi ve başlangıç noktasına geri dönüşü amaçlanır. Ziyaretçinin toplam seyahat mesafesini en aza indirmek için şehirleri hangi sırayla ziyaret edeceği araştırılır. Alternatif çözümler t (t_1, t_2, \dots, t_n) olmak üzere ziyaret edilecek şehir sayısının permütasyonundan oluşur. Şehirler arası seyahat mesafesinin minimizasyonunu hedefleyen amaç fonksiyonu f olmak üzere (Jiang, Weise, Lassig, Chiong ve Athauda, 2014: 148);

$$f(t) = D_{t_n, t_1} + \sum_{i=1}^{n-1} D_{t_i, t_{i+1}} + D_{t_n, t_1} \quad (1)$$

Bu çalışmada Türkiye'de faaliyet gösteren bir siyasi partinin seçim çalışmaları için düzenlediği şehir mitingleri esas alınmıştır. Siyasi partinin

miting planı gerçek hayattaki verilerden elde edilmiştir. Ele alınan partinin düzenlediği 21 miting noktası çalışma kapsamına alınmıştır. Mevcut gerçekleştirilen rota ile genetik algoritmalar gezgin satıcı problemi ile elde edilen alternatif çözüm karşılaştırılmıştır. Seçilen siyasi partinin miting planı ve şehirlerarası uzaklıklar Tablo 1'de verilmiştir.

Tablo 1: Seçilen Siyasi Partinin Miting Planı

Şehir Kodu	Miting Şehirleri	Miting Planına Göre Şehir Sıralaması	Uzaklık (Km)
1	Kocaeli	Kocaeli - Sakarya	37
2	Sakarya	Sakarya - Osmaniye	878
3	Osmaniye	Osmaniye - K.Maraş	105
4	K.Maraş	K.Maraş - Adana	192
5	Adana	Adana - Hatay	191
6	Hatay	Hatay - Gaziantep	194
7	Gaziantep	Gaziantep - Tokat	484
8	Tokat	Tokat - Samsun	230
9	Samsun	Samsun - Sivas	338
10	Sivas	Sivas - Erzurum	438
11	Erzurum	Erzurum - Kütahya	1186
12	Kütahya	Kütahya - Uşak	139
13	Uşak	Uşak - Antalya	294
14	Antalya	Antalya - Isparta	130
15	Isparta	Isparta - Trabzon	1131
16	Trabzon	Trabzon - İstanbul	1065
17	İstanbul	İstanbul - Bursa	243
18	Bursa	Bursa - Yozgat	599
19	Yozgat	Yozgat - Kayseri	175
20	Kayseri	Kayseri - Konya	304
21	Konya	Toplam	8353

Mevcut miting planına göre şehirlerarası ulaşım görsel olarak Harita 1'de gösterilmiştir.



Harita 1: Mevcut Kullanılan Miting Planı

Miting planında bulunan şehirler koordinatları esas alınarak genetik algoritma işlem sürecine tabi tutulmuştur. Böylece şehirlerarası uzaklıklar kuş uçuşu olarak gösterilmiştir ancak şehirlerarası mesafeler iller arası mesafe cetveli⁵⁹ kullanılarak ele alınmıştır. Problem matematiksel olarak aşağıdaki şekilde ifade edilebilir.

$$\text{Minimize } Z = \sum_{i=1}^n \sum_{j=1, i \neq j}^n x(i, j) d(i, j) \quad (2)$$

$$\text{Kısıtlar: } \sum_{j=1, j \neq i}^n x(i, j) = 1, i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1, i \neq j}^n x(i, j) = 1, j = 1, 2, \dots, n \quad (4)$$

$$\sum_{i, j \in S, i \neq j} x(i, j) \leq |S| - 1, \forall S \subset \{1, 2, \dots, n\} \quad (5)$$

$$x(i, j) \begin{cases} 1, i \text{ noktasından } j \text{ noktasına gidiliyor ise} \\ 0, i \text{ noktasından } j \text{ noktasına gidilmiyor ise} \end{cases} \quad (6)$$

Matematiksel formda amaç fonksiyonunda $x(i, j)$; i noktasından j noktasına gidilip gidilmediğini göstermektedir, $d(i, j)$ ise i ve j noktaları arasındaki mesafeyi göstermektedir. İlk iki kısıt noktalara sadece bir defa uğranacağını garanti almaya yöneliktir. İlk kısıt her noktadan sadece bir defa çıkılması gerektiği için, ikinci kısıt ise her noktaya sadece bir defa gidilmesini belirtmektedir. Kısıtlardan 3. ise alt turları eleme kısıdıdır. Son kısıt ise $x(i, j)$ 'nin 1 olması i noktasından j noktasına gidildiğini, 0 olması durumunda i noktasından j noktasına gidilmediğini göstermektedir (Çolak, 2010: 425).

Gezgin satıcı problemlerinde kromozomların kodlanması permütasyon sıralı kodlama olarak yapılmaktadır. Ziyaret edilecek şehir sayısı $n=21$ olduğundan dolayı $21!$ ($21!=51.090.942.171.709.400.000$) olan bir seçim kümesi söz konusudur. Örneğin mevcut miting rotası (X) ve rastgele yapılan alternatif bir çözüm örneğinin (Y) kodlanmış hali;

Kromozom X; 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Kromozom Y; 19 5 11 16 6 20 7 1 17 8 12 9 21 10 18 2 13 15 3 14 4

Uygunluk değeri amaç fonksiyonunda gösterildiği gibi minimum uzaklık değeri sağlayan kromozomlardan oluşacaktır.

Genetik algoritmalar literatüründe popülasyon büyüklüğünün 30 ile 100 arasında belirlenmesi önerilmektedir. Ele alınan problem ve kullanılan TSP (Travelling Salesman Problem) genetik algoritma programı ile minimum mesafe ve yenileme amaçlı yapılan denemeler sonucunda popülasyon büyüklüğü 30 olarak belirlenmiştir, çaprazlama operatörü olarak kısmi planlı çaprazlamaya dayalı 5 birimlik aralık kullanılarak optimum sonuç elde edilmiştir ve aynı şekilde her bir alternatifin değişim (mutasyon) oranı olarak

⁵⁹ T. C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı; Karayolları Genel Müdürlüğü iller arası mesafe cetveli (<http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Uzakliklar/illerArasiMesafe.aspx>)

%2, en iyi alternatiflerin saklanması özelliği (elitizm) %5 olarak bulunmuştur. Optimum sonuç, 14453 yineleme sonunda elde edilmiştir.

K. Maraş ve Erzurum arasındaki mesafenin maksimum aralık olması nedeniyle başlangıç şehrinin Erzurum seçilmesi ve K. Maraş'ında son şehir olarak değerlendirmesi varsayımı altında; Optimum Kromozom, 11 16 9 8 10 20 19 2 1 17 18 12 13 15 14 21 5 3 6 7 4 olarak belirlenmiştir ve Tablo 2'de gösterilmiştir.

Tablo 2: Hesaplanan Optimum Miting Planı

Şehir Kodu	Miting Şehirleri	Miting Planına Göre Şehir Sıralaması	Uzaklık (Km)
11	Erzurum	Erzurum - Trabzon	304
16	Trabzon	Trabzon - Samsun	331
9	Samsun	Samsun - Tokat	230
8	Tokat	Tokat - Sivas	108
10	Sivas	Sivas - Kayseri	194
20	Kayseri	Kayseri - Yozgat	175
19	Yozgat	Yozgat - Sakarya	520
2	Sakarya	Sakarya - Kocaeli	37
1	Kocaeli	Kocaeli - İstanbul	111
17	İstanbul	İstanbul - Bursa	243
18	Bursa	Bursa - Kütahya	177
12	Kütahya	Kütahya - Uşak	139
13	Uşak	Uşak - Isparta	171
15	Isparta	Isparta - Antalya	130
14	Antalya	Antalya - Konya	322
21	Konya	Konya - Adana	356
5	Adana	Adana - Osmaniye	87
3	Osmaniye	Osmaniye - Hatay	129
6	Hatay	Hatay - Gaziantep	194
7	Gaziantep	Gaziantep - K. Maraş	76
4	K.Maraş	Toplam	4034

Elde edilen optimum sonuca göre miting planı; Erzurum, Trabzon, Samsun, Tokat, Sivas, Kayseri, Yozgat, Sakarya, Kocaeli, İstanbul, Bursa, Kütahya, Uşak, Isparta, Antalya, Konya, Adana, Osmaniye, Hatay, Gaziantep ve K.Maraş sıralaması kapalı GSP olarak Harita 2'de gösterilen şekilde olmalıdır.



Harita 2: Maliyet Açısından Optimum Olarak Önerilen Miting Planı

Sonuç

Evrin veya doğal seleksiyona dayalı Genetik Algoritmalar konusu içerisinde yer alan Gezgin Satıcı Problemi birçok alanda ve ulaşım sektöründe planlama, araştırma vb. amaçlar için kullanılabilir. Bu çalışmada Gezgin Satıcı Problemlerine örnek teşkil eden Türkiye'de faaliyet gösteren bir siyasi partinin seçim propagandası nedeniyle düzenlediği mitingler çalışma kapsamına alınmıştır. Ekonomik kaynakların kullanımının optimize edilmesi amacıyla seçilen siyasi partinin mevcut miting planı ve önerilen miting planı arasında elde edilebilecek zaman ve maliyet tasarrufları analiz edilmiştir. Ekonomik kaynakların kullanımının optimize edilmesi amacıyla seçilen siyasi partinin mevcut miting planı ve önerilen miting planı arasında elde edilebilecek zaman ve maliyet tasarrufları analiz edilmiştir. Çalışmanın kısıtları minimum maliyet ve zaman olarak değerlendirilmiştir. Eğer benzer şekilde bazı mitinglerin önceliği kısıdı eklenmek istenirse bu kısıt kromozom yapısında gösterilir ve uygulama kısmının ilk aşamasında öncelikli olan miting planı mevcut uygulama ile aynı şekilde oluşturularak, ikinci aşamada öncelikli olmayan diğer şehirlerin miting planı oluşturulabilir. Benzer şekilde eğer herhangi bir veya birden fazla ilde düzenlenecek miting zamanı kesin olarak belirlenmişse; bu iller kromozom yapısında sabit olarak gösterilir ve diğer illerin miting planı optimum kombinasyonu verecek şekilde farklı durumlar için değerlendirilir. Örneğin uygulama bölümündeki 21 şehir için 3. miting olarak X şehri belirlenmişse; diğer iller içerisindeki ikili kombinasyonlar ilk iki miting için planlanır ve kalan şehirler için miting planı oluşturulur. 20 farklı şehir için 190 adet ikili kombinasyon ile elde edilmiş miting planları içerisindeki en optimum olanı tercih edilecektir.

Analiz kapsamında siyasi partinin 34 günü ve 21 ili kapsayan ve bazı illerdeki mitingleri aynı gün içerisinde olabilen miting planı; her ilde bir gün içerisinde miting düzenlendiği varsayımı altında, 21 güne indirgenerek yaklaşık olarak %40 zaman tasarrufu sağlanmıştır. Mevcut miting planındaki şehirlerarası uzaklıklar toplamı 8353 km iken alternatif miting planındaki

şehirlerarası uzaklıklar toplamı 4034 km'dir. Şehirlerarası ulaşımın karayolu ile yapıldığı ve alternatif ulaşım yöntemlerinin de siyasi parti kaynaklarına aynı orantıda ek maliyet yükleyeceği varsayımı altında, şehirlerarası mesafeler maliyet olarak ele alınacak olursa yaklaşık olarak %52 maliyet tasarrufu sağlanmıştır. Ayrıca ele alınan siyasi partinin miting planı oluştururken, zaman ve maliyet olarak değerlendirilen kısıtlar dahilinde olmayan diğer etmenleri göz ardı edilmiştir. Bu etmenlerin mevcut kısıtlara eklenmesi siyasi parti açısından daha güvenilir sonuçlar elde edilmesini sağlayacaktır.

Gezgin satıcı problemi ile ele alınan miting planlaması benzer şekilde diğer siyasi partiler ve benzer uygulamalar için de kolaylıkla uygulanabilecektir. Farklı uygulamalarda problem boyutu değişebileceğinden optimal çözüm araştırma parametreleri değişkenlik gösterebilecektir.

Kaynakça

- Adeli, H. ve Sarma, K. C. (2006), *Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms and Parallel Computing*, John Wiley & Sons Ltd., Great Britain.
- Affenzeller, M., Winkler, S., Wagner, S. ve Beham, A. (2009), *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications (Volume 6: Numerical Insights)*, Taylor & Francis Group, LLC, United States America.
- Akar, S. B. ve Şahingöz, Ö. K. (2015), "Solving Asymmetric Traveling Salesman Problem Using Genetic Algorithm", 23. Sinyal İşleme ve İletişim Uygulamaları, 16–19 Mayıs, Malatya.
- Alba, E. ve Dorronsoro, B. (2008), *Cellular Genetic Algorithms, Operations Research / Computer Science Interfaces Series 42*, Springer Business + Science Media, New York.
- Ashokkumar, V. ve Hebbal, S. S. (2014), "Route Optimization of Automated Warehouse with the Aid of Modified Genetic Algorithms (MGA)", *International Review of Mechanical Engineering*, 8, 4, s. 667-679.
- Bandyopadhyay, S. ve Pal, S. K. (2007), *Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence (Natural Computing Series)*, Springer Business + Science Media, Berlin.
- Borna, K. ve Hashemi, V. D. (2014), "An Improved Genetic Algorithm With A Local Optimization Strategy And An Extra Mutation Level For Solving Traveling Salesman Problem", *Computer Science, Engineering and Information Technology*, 4, 4, s. 47-53.
- Chen, Y. P. (2006), *Extending the Scalability of Linkage Genetic Algorithms (Studies in Fuzziness and Soft Computing, Volume 190)*, Springer Business + Science Media, Netherlands.

- Cox, E. (2005), *Fuzzy Modelling and Genetic Algorithms for Data Mining and Exploration (The Morgan Kaufmann Series in Data Management Systems)*, Elsevier Inc., United States of America.
- Çakır, M. ve Yılmaz, G. (2015), Traveling Salesman Problem Optimization with Parallel Genetic Algorithm, 23. Sinyal İşleme ve İletişim Uygulamaları, 16–19 Mayıs, Malatya.
- Çolak, S. (2010), Genetik Algoritmalar Yardımı ile Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama, *Ç. Ü. Sosyal Bilimler Enstitüsü Dergisi*, 19, 3, s. 423-438.
- Dreo, J., Petrowski, A., Siarry, P. ve Taillard, E. (2006), *Metaheuristics For Hard Optimization Methods and Case Studies*, Springer Business + Science Media, Berlin.
- Elmas, Ç. (2011), *Yapay Zeka Uygulamaları (Yapay Sinir Ağı, Bulanık Mantık, Sinirsel Bulanık Mantık, Genetik Algoritma)*, 2. Baskı, Seçkin Yayıncılık, Ankara.
- Fogel, L. J., Owens, A. J. ve Walsh, M. J. (1966) *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, Inc., New York.
- Gen, M., Cheng, R. ve Lin, L. (2008), *Network Models and Optimization Multiobjective Genetic Algorithm Approach (Decision Engineering)*, Springer-Verlag London Limited, London.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Inc., New York.
- Haupt, R. L. ve Haupt, S. E. (2004), *Practical Genetic Algorithms*, 2nd Edition, John Wiley & Sons, Inc., New Jersey.
- Holland, J. H. (1975), *Adaptation In Natural And Artificial Systems: An Introductory Analysis With Applications To Biology, Control, And Artificial Intelligence*, University of Michigan Press, Ann Arbor.
- Jiang, Y., Weise, T., Lassig, J., Chiong, R. ve Athauda, R. (2014), “Comparing a Hybrid Branch and Bound Algorithm with Evolutionary Computation Methods, Local Search and their Hybrids on the TSP”, IEEE Symposium on Computational Intelligence in Production and Logistics Systems, 9–12 December, Orlando, Florida.
- Kiraly, A. ve Abonyi, J. (2015), Redesign of the Supply of Mobile Mechanics Based on a Novel Genetic Optimization Algorithm Using Google Maps API, *Engineering Applications of Artificial Intelligence*, 38, 2015, s. 122–130.
- Kubalik, J. ve Snizek, M. (2014), “A Novel Evolutionary Algorithm with Indirect Representation and Extended Nearest Neighbor

- Constructive Procedure for Solving Routing Problems”, International Conference on Intelligent Systems Design and Applications, 28–30 November, Okinawa.
- Nabiyev, V. V. (2012), *Yapay Zeka*, 4. Baskı, Seçkin Yayıncılık, Ankara.
- Pan, Y. ve Xia, Y. (2014), “Solving TSP by Dismantling Cross Paths”, IEEE International Conference on Orange Technologies, 20–23 September, Xian.
- Rekiek, B. ve Delchambre, A. (2006), *Assembly Line Design: The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms (Springer Series in Advanced Manufacturing)*, Springer Business + Science Media, Germany.
- Satyhan, A., Ernest, N. ve Cohen, K. (2015) “Genetic Fuzzy Approach for Control and Task Planning Applications”, *American Institute of Aeronautics and Astronautics Infotech @ Aerospace*, January, 2015, s. 1-9.
- Schwefel, H. P. (1993), *Evolution And Optimum Seeking: The Sixth Generation*, John Wiley & Sons, Inc., New York.
- Sivanandam, S. N. ve Deepa, S. N. (2008), *Introduction to Genetic Algorithms*, Springer Business + Science Media, Berlin.
- Tabatabaei, N. M., Asadian, K. ve Boushehri, N. S. (2014), “Short Term Power Load Forecasting Based On Comparison Of Acs To Probabilistic Traveling Salesman Problem”, *Technical and Physical Problems of Engineering*, 21, 6, s. 66-74.
- Taşkın, Ç. ve Emel, G. G. (2009), *Sayısal Yöntemlerde Genetik Algoritmalar*, Alfa Aktüel, Bursa.
- Ünal, M., Ak, A., Topuz, V. ve Erdal, H. (2013), *Optimization of PID Controllers Using Ant Colony and Genetic Algorithms*, Springer Business + Science Media, Berlin.
- Yılmaz, H., Doğan, Ş. ve Koca, G. Ö. (2015), “Mayın İmhası için Optimum Mesafe Tespitinde Küre Yüzeyinde 3 Boyutlu Gezgin Satıcı Probleminin Kullanılması”, 23. Sinyal İşleme ve İletişim Uygulamaları, 16–19 Mayıs, Malatya.
- T. C. Ulaştırma Denizcilik ve Haberleşme Bakanlığı, KGM, (2015), <http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Uzakliklar/illerArasiMesafe.aspx>, 27.08.2015.