



Gezgin satıcı problemi için yeni bir meta-sezgisel: kör fare algoritması

A novel metaheuristic for traveling salesman problem: blind mole-rat algorithm

Tevfik YILDIRIM¹, Can Berk KALAYCI², Özcan MUTLU^{2*}

¹Bilgi İşlem Daire Başkanlığı, Pamukkale Üniversitesi, Denizli, Türkiye.
tyildirim@pau.edu.tr

²Endüstri Mühendisliği Bölümü, Mühendislik Fakültesi, Pamukkale Üniversitesi, Denizli, Türkiye.
canberk@pau.edu.tr, mutlu@pau.edu.tr

Geliş Tarihi/Received: 12.12.2014, Kabul Tarihi/Accepted: 29.07.2015
* Yazışılan yazar/Corresponding author

doi: 10.5505/pajes.2015.38981
Araştırma Makalesi/Research Article

Öz

Gezgin Satıcı Problemi (GSP), başlangıç ve bitiş şehirleri aynı olan ve her şehrin sadece bir kez ziyaret edildiği minimum mesafeli tur bulma problemidir. Şehir sayısı arttıkça, kesin yöntemler ile kabul edilebilir sürelerde bir optimum çözüm bulunması zordur. Bu nedenle, son elli yılda GSP'nin çözümü için doğadan ve biyolojiden esinlenen birçok meta-sezgisel yöntem geliştirilmiştir. Bu çalışmada, toprak altındaki bireysel tünel sistemlerinde yaşayan kör farelerin toprak altındaki engelleri geçme stratejisinden esinlenilerek GSP'nin çözümü için yeni bir meta-sezgisel tasarlanmıştır. Geliştirilen yönteme Kör Fare Algoritması adı verilmiştir. Bu yeni sezgisel ile farklı boyutlardaki simetrik test veri setleri için deneyler yapılmış ve sonuçları bilinen en iyi sonuçlar ile kıyaslanmıştır. Önerilen meta-sezgisel henüz literatürdeki diğer algoritmalarla yarışabilecek düzeyde olmamasına rağmen, başlangıç test çözümlerinin umut verici olduğu söylenebilir.

Anahtar kelimeler: Gezgin satıcı problemi, Kombinatoriyel eniyileme, Meta-Sezgisel, Kör fare algoritması

Abstract

Traveling Salesman Problem (TSP) is the problem of finding a minimum distance tour of cities beginning and ending at the same city and that each city are visited only once. As the number of cities increases, it is difficult to find an optimal solution by exact methods in a reasonable duration. Therefore, in recent five decades many heuristic solution methods inspired of nature and biology have been developed. In this paper, a new metaheuristic method inspired of the by-passing the obstacle strategy of blind mole rats living in their individual tunnel systems under the soil is designed for solving TSP. The method is called as Blind Mole-rat Algorithm. The proposed algorithm is tested on different size of symmetric TSP problems and the results are compared to the best known results. Initial test results are promising although proposed metaheuristic is not yet competitive enough among other algorithms in the literature.

Keywords: Traveling salesman problem, Combinatorial optimization, Metaheuristic, Blind mole-Rat algorithm

1 Giriş

Gezgin satıcı problemi (GSP), başlangıç ve bitiş şehirleri aynı olan ve her bir şehrin sadece bir kez ziyaret edildiği en kısa mesafeli turun bulunması olarak tanımlanabilir [1]. Çizge teorisinde, en düşük maliyetli Hamilton çevriminin bulunması olarak tanımlanır [2]. V düğümler kümesini, A kenarlar kümesini, $C = (c_{ij})$ A kümesi ile ilişkilendirilmiş mesafe ya da maliyet matrisini temsil etmek üzere $G = (V, A)$ çizgesinde, her bir düğüme sadece bir kez uğrayan en kısa mesafeli kapalı yol Hamilton çevrimi olarak tanımlanır [3]. Eğer kenarlar arasındaki karşılıklı maliyetler birbirine eşitse ($c_{ij} = c_{ji}$) ise problem Simetrik Gezgin Satıcı Problemi (SGSP) olur [4].

Ağ optimizasyonu, çizelgeleme problemi, devre levhalarının delinmesi, bilgisayar pin bağlantısı, montaj hattı sıralama gibi birçok endüstriyel problem GSP'ye dönüştürülebildiği için GSP üzerinde oldukça fazla çalışılan kombinatoriyel problemlerden biridir [3],[5].

GSP'nin genel biçimi başta Karl Menger olmak üzere 1930'larda Viyana ve Harvard'daki matematikçiler tarafından çalışılmıştır. 1930'lardan günümüze kadar GSP'nin çözümü için birçok kesin ve sezgisel yöntem geliştirilmiştir [6]. GSP için çözüm yöntemlerini kesin ve yakın algoritmalar olarak iki sınıf kategorisi altında toplanabilir [7]. GSP, NP-Tam sınıfı bir kombinatoriyel problemdir ve bu yüzden tüm GSP problemlerin çözümü için polinom zamanda çözüm sağlayan bir algoritma henüz bulunmamaktadır. Bu nedenle optimal çözümü garanti

etmeyen ancak yüksek kalitede çözümler sağlayan birçok sezgisel yöntem geliştirilmiştir [1]. Doğadan ve biyolojiden esinlenilerek geliştirilen tavlama benzetimi [8],[9], genetik algoritmalar [10]-[12] yapay sinir ağları [13]-[16], karınca kolonisi algoritmaları [5],[17]-[19] parçacık sürü optimizasyonu [20]-[24] yapay arı kolonisi algoritmaları [25]-[27], sivrisinek konak arama [28], akıllı su damlaları algoritması [29], cuckoo arama algoritması [30],[31], hibrid algoritmalar [32]-[36], yarasa (bat-inspired) [37]-[39], ateşböceği algoritması [40], harmoni arama [41], bakteriyel yiyecek arama algoritması [42], kedi sürüsü algoritması [43], biyocoğrafya tabanlı algoritmalar [44], yapay balık okulu algoritması [45] bu sezgisellere örnek olarak verilebilir.

Son yıllarda doğadan esinlenen algoritmaların sayısında fark edilir derecede bir artış gözlenmektedir. Bu algoritmalar [46] ve [47]'dan ulaşılabilir. Bu algoritmalar ilginç bir örnek, kör çıplak farelerin yiyecek arama ve koloni yaşamları temel alınarak geliştirilen Kör, Çıplak Fare Algoritması (KCFA) adında bir numerik fonksiyon optimizasyon metodudur [48],[49]. KCFA, bu çalışmada önerilen Kör Fare Algoritması (KFA) ile isim benzerliği taşımamasına rağmen, KCFA, yiyecek kaynaklarının ve toprağın sıcaklık, nem, yoğunluk gibi özelliklerini kullanmaktadır. KFA ise, kör farelerin yankı ile konumlama özelliklerinden esinlenerek geliştirilmiştir. Bu nedenle, algoritma adımları ve kullanılan parametreler tamamen farklıdır. Bir diğer ilginç algoritma ise yarasa algoritmasıdır (YA) [38]. YA, yarasaların hız, konum, frekans, dalga boyu ve ses şiddeti bilgilerini kullanarak mesafe

algılamalarından esinlenmiştir. YA'da, yarasaların yankı ile konumlama özelliklerinin kullanıldığı sözel olarak ifade edilmiş olmasına rağmen, YA formülasyonu olarak daha çok parçacık sürü optimizasyonu ile benzerlik göstermektedir. YA'da, yarasaların sinyal göndermesi ve engelden gelen bilgiyi kullanması durumu yoktur. Kör farelerin yer altındaki tünellerde ilerleme ve engelleri geçme davranışlarına dayanan KFA ise, yankı ile konumlama bilgisini olasılıksal olarak kullanmaktadır. Bu özellikleri ile KFA, KCFA ve YA'dan ayrılmaktadır.

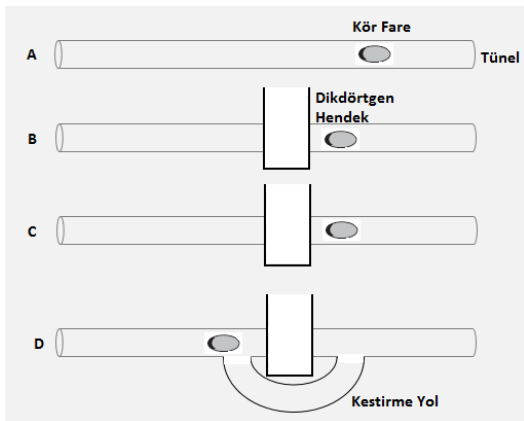
Bu çalışmanın ikinci kısmında kör fareler hakkında bilgiler verilerek kör fare algoritması anlatılmış ve yapılan deney sonuçlarına yer verilmiştir. Üçüncü bölümde ise genel bir değerlendirme yapılarak gelecekteki yapılabilecek çalışmalarından bahsedilmiştir.

2 Kör fareler ve kör fare algoritması

Bio-sonar (yankı ile konumlama) birçok hayvan türü tarafından kullanılan biyolojik bir radardır. Bio-sonar yeteneğine sahip hayvanlar ses dalgaları göndererek bu dalgaların nesnelere yansımalarını dinlerler. Bu yansımaları çevrelerindeki nesnelere tanımlamak ve konumlandırmak için kullanırlar. Yankı ile konumlama çeşitli ortamlarda yön bulma ve avcılık için kullanılır [50]. Bu canlılara ilginç örneklerden biri de kör farelerdir.

2.1 Kör fareler

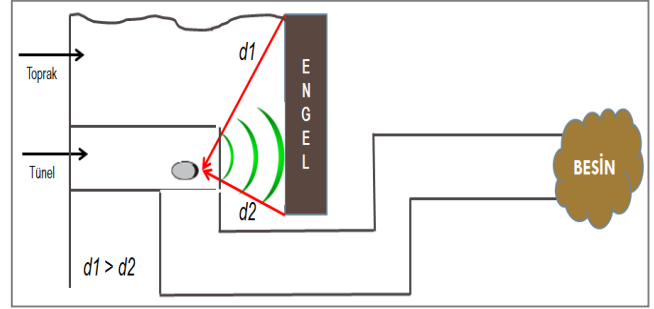
Kör fareler, koloniler halinde yaşayan üst düzey sosyal örgütlü (eusocial) canlılardır [51]. Yer altında kazdıkları tünellerde yaşayan kör fareler, tünel kazarken bir engelle karşılaştıklarında enerjilerini ekonomik kullanmak için en kısa yolları tercih ederler [52]. Bu hayvanlar bir nesnenin konum, boyut, uzaklık ve türünü tahmin etmek için yüksek frekanslı ultrasonik sesler yayarlar ve nesnenin yüzeyinden yansıyan ses dalgalarını değerlendirirler [53]. Kör farelerin bu davranışını daha yakından anlayabilmek için bu video [54] izlenebilir. Bir kör farenin bir engelle karşılaştığında engeli geçmesi Şekil 1'de gösterilmiştir. Şekil 1'de **A**, aktif bir kör farenin düz bir hatta ilerleyen tüneline gösterir. **B**'de dikdörtgen bir hendek kesecek şekilde çakılır. **C**'de kör fare tünelin hendeğe temas eden açık kısmını toprakla doldurur. **D**'de kör fare tünelin iki ayrı bölümünü engelin altından birleştirir.



Şekil 1: Kör Farenin engeli geçmesi.

Kör fareler engeli aşmak için tünel içinde ilerlerken kafalarını

tünelin tavanına vurarak küçük depremler ile dalgalar yaratırlar. Bu dalgaların engelle çarpıp yansımalarına göre engelin hangi tarafından geçeceklerine karar verirler. Şekil 2'de kör fare bir dalga göndermekte ve geri dönen sinyallerden engelin büyüklüğü ve konumunu belirlemektedir. d_1 mesafesi d_2 'den büyük olduğundan fare engelin altından tüneli kazarak daha kısa yoldan besine ulaşır.



Şekil 2: Engel geçiş stratejisi.

Koloni yaşamında bir birey yeni bir besin kaynağı bulduğunda, yuvaya geri dönerken özel bir ses çıkararak kolonideki diğer üyeleri uyarır [51].

2.2 Kör fare algoritması

Bu çalışmada kör farelerin bir engeli geçerken en kısa mesafeyi bulma davranışı GSP çözümüne uyarlanmıştır. Bu fikirle kör fare algoritması (KFA) olarak isimlendirilen bir algoritma tasarlanmıştır. Bu algoritmada, öncelikle düğümler arasındaki mesafeler kullanılarak mesafe matrisi ve her bir düğüm için küçükten büyüğe sıralı bir şekilde yakınlık matrisi oluşturulur. Bu ön hazırlık işleminden sonra problemde yer alan her düğüme bir fare yerleştirilir. Her bir fare için farenin bulunduğu düğüm ziyaret edilen düğümler listesine eklenir. Tüm düğümler arasındaki yollara sinyal değeri ve sinyal iletim katsayısı atanır. Her bir fare bulunduğu düğümden sonra gideceği düğüm çiftini yakınlık matrisinden elde ettiği en yakın q adet komşu arasından denklem (1)'deki formülü kullanarak seçer ve gidilecek düğüm çiftini belirler. Her bir fare gideceği düğüm çiftini seçtikten sonra seçilen düğümler ziyaret edilen düğümler listesine eklenir, yakınlık matrisinden çıkarılır ve yakınlık matrisi güncellenir. Böylece, her bir fare turunu tamamlar. Daha sonra, denklem (2) kullanılarak sinyal kayıpları bulunur ve denklem (3) ile tüm sinyal değerleri güncellenir. Sinyal güncelleme işleminden sonra en kısa tura yapan farenin kullandığı yollardaki sinyal değerlerine denklem (4) kullanılarak sabit bir sinyal değeri eklenir. Daha sonra, fareler tarafından elde edilen en iyi tura bir komşuluk operatörü uygulanarak çözüm iyileştirilmeye çalışılır ve bu işlemle birlikte bir iterasyon tamamlanmış olur. Karınca kolonisi sistemine 3-opt yerel aramasının [55] eklenmesinin daha iyi sonuçlar üretmesinden hareketle bulunan en iyi çözüme bir komşuluk operatörünün uygulanmasıyla her nesilde iyileşmeye gidilmesi amaçlanmıştır. Bir sonraki iterasyona geçildiğinde sinyal iletim katsayısı değerleri yeniden atanır ve sinyal değerleri bir önceki iterasyon sonunda elde edilen son güncel sinyal değerleri olarak kullanılır. Maksimum iterasyon sayısına ulaşma olarak belirlenen durdurma kriteri sağlanıncaya kadar algoritma çalışmaya devam eder. Tablo 1'de kör fare algoritmasının parametreleri gösterilmiştir.

Tablo 1: KFA'nın parametreleri.

Başlangıç Sinyali s_{ij}	i 'nci düğümünden j 'nci düğüme gelen sinyal değeri,
$R_{ij\ min}$	i 'nci düğüm ile j 'nci düğüm arasındaki sinyal iletim katsayısının alabileceği en küçük değer,
$R_{ij\ max}$	i 'nci düğüm ile j 'nci düğüm arasındaki sinyal iletim katsayısının alabileceği en büyük değer,
w	Sinyal kayıp katsayısı,
v	Sabit sinyal ekleme değeri,
q	Geçişte arama yapılacak en yakın komşu sayısı,
Maksimum sinyal artış oranı	Bir yol üzerinde iletilebilecek maksimum sinyal değerini sınırlar,
İterasyon Sayısı	Maksimum tur sayısı (durdurma ölçütü),
Komşuluk operatörü uygulama sayısı	Komşuluk operatörü uygulama işleminin kaç kez yapılacağını belirtir,
Komşuluk operatörü uygulama aralığı	Komşuluk operatörü uygulanmasının gerçekleşeceği aralık.

Algoritma adımları Şekil 3'te verilmiştir.

Adım 1:	Algoritmayı başlat,
Adım 2:	Tüm yollara başlangıç sinyal değerini ata,
Adım 3:	Her düğüme bir fare yerleştir,
Adım 4:	Tüm yollara sinyal iletim katsayısını ata,
Adım 5:	Her bir fare için gideceği sonraki iki bağlı düğümü seç,
Adım 6:	Ziyaret edilmesi gereken düğüm hala varsa Adım 5'e git,
Adım 7:	Turu tamamlayan her bir farenin tur uzunluğunu bul,
Adım 8:	Tüm yollardaki sinyal kayıplarını hesapla ve sinyal değerlerini güncelle,
Adım 9:	En kısa turu yapan farenin kullandığı rotaya komşuluk operatörü uygula,
Adım 10:	En kısa turu yapan farenin kullandığı rotaya sinyal eklemesi yap,
Adım 11:	Durdurma kriteri henüz sağlanmadıysa Adım 3'e git,
Adım 12:	Algoritmayı durdur.

Şekil 3: Algoritma adımları.

Algoritmanın adımları aşağıda detaylı olarak açıklanmıştır.

- 1) Algoritmayı başlat:** Problem verilerini oku, yakınlık matrisini oluştur ve KFA parametre değerlerini ayarla.
- 2) Tüm (i, j) yollarına s_{ij} başlangıç sinyal değerini ata:** Tüm düğümler arasındaki yollara bir başlangıç sinyal değeri atanır. Problem simetrik olduğu için $s_{ij} = s_{ji}$ 'dir.
- 3) Her düğüme bir fare yerleştir:** Her fare için bulunduğu düğümü ziyaret edilen düğümler listesine ekle.
- 4) Tüm (i, j) yollarına R_{ij} sinyal iletim katsayısını ata:** Tüm düğümler arasındaki yollara bir sinyal iletim katsayısı atanır. R_{ij} katsayısı, R_{min} ile R_{max} arasından üretilen sayılardan rassal olarak atanır. Problem simetrik olduğu için $R_{ij} = R_{ji}$ 'dir. Doğada ses dalgaları her ortamda farklı iletirler. Tüm düğümler arasındaki yollar farklı ortamlar olarak varsayılarak tüm yollara farklı R_{ij} katsayısı atanır. Düğümler arasındaki yolların farklı ortamlar olarak düşünülmesinin sebebi farklı komşuluk kombinasyonlarının denenebilmesi içindir.

- 5) Her bir fare için gideceği sonraki iki düğümü birlikte seç:** Seçilecek sonraki 2 düğüm tüm düğüm seçenekleri yerine en yakın q komşu arasından yapılır. Seçim için denklem (1)'deki formül kullanılır.

$$S(m)_{ijk} = s_{ij} - \left(\frac{d_{ij}^2}{s_{ij}}\right) \cdot R_{ij} + s_{jk} - \left(\frac{d_{jk}^2}{s_{jk}}\right) \cdot R_{jk} \quad (1)$$

Her bir fare $S(m)_{ijk}$ değeri en yüksek düğüm çiftini seçer ve fare sırasıyla j, k 'nci düğüme ilerletilir. j ve k düğümleri ziyaret edilen düğümler listesine eklenir. $S(m)_{ijk}$ değerlerinin eşitliği durumunda i 'nci düğüme toplam uzaklığı en küçük düğüm çifti seçilir.

- 6) Tüm düğümleri ziyaret:** Ziyaret edilecek düğüm kalmadığına emin ol ve sıradaki adıma geç.
- 7) Her bir farenin tur uzunluğunu bul:** Her bir farenin düğümleri ziyaret sırasına göre toplam tur uzunluğu bulunur ve küçükten büyüğe sıralanır.
- 8) Tüm yollardaki sinyal kayıplarını hesapla ve sinyal değerlerini güncelle:** Gerçek hayatta olduğu gibi algoritmada sinyallerin farklı ortamlarda ve uzaklıklarda ilerlerken soğurulduğu prensibinden hareketle tüm (i, j) yollarında mesafe ve w değerine göre sinyal kayıpları bulunur. Sinyal kayıpları her bir (i, j) yolu için denklem (2) kullanılarak hesaplanır. Hesaplanan sinyal kayıpları başlangıç sinyal değerlerinden çıkarılarak (i, j) yollarının sinyal değerleri denklem (3) kullanılarak hesaplanır.

$$\Delta s_{ij} = \left(\frac{d_{ij}}{w}\right) \quad (2)$$

$$s_{ij} = s_{ij} - \Delta s_{ij} \quad (3)$$

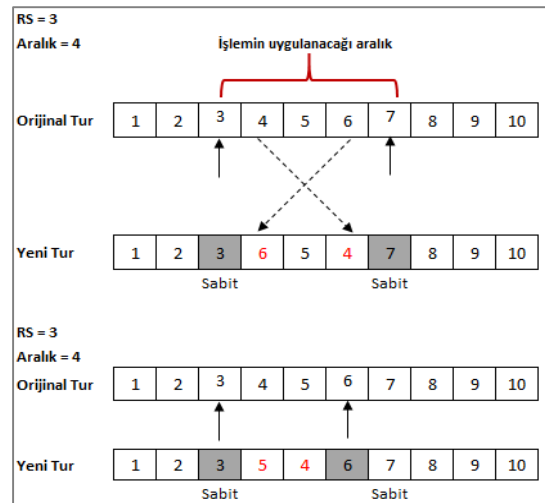
- 9) Komşuluk operatörünü uygula:** Bu çalışmada kullanılan komşuluk operatörü, bir çözümde rasgele seçilen iki düğüm arasındaki düğümlerin pozisyonları tersine çevirir. Şekil 4'te komşuluk operatörü uygulaması için iki örnek verilmiştir.

- 10) En kısa turu yapan farenin kullandığı rotaya sinyal eklemesi yap:** En kısa turu yapan farenin kullandığı (i, j) yollarına sinyal eklemesi yapılır ve denklem (4) kullanılarak sinyaller güncellenir.

$$s_{ij} = s_{ij} + v \quad (4)$$

- 11) Durdurma kriteri kontrolü:** Durdurma kriteri sağlanıp sağlanmamasına göre uygun adıma git.

- 12) Sonlandır:** Maksimum tur sayısına ulaşıldığında algoritma durdurulur.



Şekil 4: Komşuluk operatörünün uygulama örnekleri.

Algoritmanın 5. adımında her bir fare bir sonraki gideceği düğüm çiftini seçerken en yakın q komşu arasından seçim yapar. Eğer şebekedeki tüm olası düğüm çiftleri arasından seçim yapılırsa işlem miktarı çok fazla miktarda artacaktır. Örneğin 50 şehirli bir problemde ilk düğüm çiftinin seçiminde bir fare için denemesi gereken sıralamaların sayısı 49'un 2'li permütasyonu ($P(49,2) = 2352$) kadardır. Her bir düğümde bir fare olduğu için 50 şehirli bir problemde sadece ilk düğüm çiftinin seçiminde tüm seçimler için, toplamda $50 \times 2352 = 117600$ işlem yapılır. Bunun yerine en yakın q komşu arasından seçim yapıldığı takdirde ($q = 5$ olsun) $P(5,2) = 20$ olur. Her bir düğümde bir fare olduğu için 50 şehirli bir problemde ilk düğüm çiftinin seçiminde tüm seçimler için $50 \times 20 = 1000$ işlem yapılır. Görüldüğü gibi en yakın q komşu arasından seçim yapıldığında yapılan işlem sayısı yaklaşık 118'de 1'ine düşmektedir. Her düğüm çifti seçiminden sonra tüm farelere ait yakınlık matrisi ziyaret edilen düğümler matristen çıkartılarak güncellenir.

2.3 Deneysel Sonuçlar

Bu çalışmada *wi29*, *dj38*, *eil51*, *berlin52*, *st70*, *eil76*, *eil101*, *kroA100*, *lin105*, *ch150*, *kroA200* ve *xqg237* test veri setleri kullanılmıştır. Kullanılan veri setleri EUC_2D normundadır. EUC_2D normundaki veriler öklidyen uzaklık formülüne uyan verilerden oluşur. Algoritmanın test edilmesi için java programlama dili ile bir yazılım geliştirilmiştir. Test işlemleri Intel Dual Core 3.33 GHZ, 4GB RAM özelliklerine sahip bir bilgisayarda gerçekleştirilmiştir. Deneylerde kullanılacak en iyi parametre seti kombinasyonunun bulunması için deneysel tasarım çalışması gerçekleştirilerek sonuçları analiz edilmiştir. Bu tasarım çalışmasında, tam faktöriyel testlerin çok uzun zaman alması nedeniyle Taguchi deney tasarımı yöntemi tercih edilmiştir. Deneysel tasarım için belirlenen faktörler ve seviyeleri Tablo 2'de gösterilmiştir.

Taguchi deney tasarımı ile uygun parametre seti belirleme çalışması Minitab programında 10 faktör 2 seviye parametre testi için L32 hesap tablosu kullanılarak gerçekleştirilmiştir. S/N oranları göz önüne alındığında %95 güven düzeyinde **başlangıç sinyali**, w , R_{max} ve **komşuluk operatörünün uygulanma aralığı** faktörlerinin sonuçlar üzerinde etkili faktörler ($p < 0.05$) olduğu tespit edilmiştir. Elde edilen S/N oranları yanıt tablosuna göre faktörlerin en iyi değerleri Tablo 3'teki gibi elde edilmiştir.

Deney tasarımı sonucuna göre bulunan en uygun parametre değerleri ile test veri setleri için 30'ar defa deney yapılmış ve Tablo 4'teki sonuçlar elde edilmiştir. *wi29*, *dj38*, *eil52* veri setleri için en iyi sonuçlar elde edilirken *eil51*, *st70*, *eil76*, *rat99*, *eil101*, *kroA100*, *lin105*, *ch150*, *kroA200*, *xqg237* için optimal

sonuç elde edilememiştir. Tablodaki süre değerleri algoritmanın bulunduğu en iyi sonucun elde edildiği deneyin çalışma süresidir.

Sapma değerleri KFA ile bulunan sonucun bilinen en iyi değerden % olarak sapma miktarını göstermektedir.

Ayrıca, yapılan deneyler uygulanan komşuluk operatörünün etkili olduğunu göstermiştir. İterasyon sonucu tur değerleri ile komşuluk operatörünün uygulanması sonucu elde edilen tur değerleri karşılaştırıldığında 1322 deneyin 1087 tanesinde komşuluk operatörünün uygulanması işlemi iyileşme sağlamıştır. Komşuluk operatörü işlemi ile sağlanan iyileşme oranı yaklaşık %82'dir. Örneğin *wi29* test problemi için en iyi parametre değerleri ile yapılan 30 deneyin 25'inde komşuluk operatörünün uygulanması işlemi ile farklı miktarlarda iyileşme elde edilmiştir. İterasyon sonucu ve komşuluk operatörünün uygulanması sonucu elde edilen iyileşme değerleri Tablo 5'te gösterilmiştir.

Tablo 2: Deney tasarımında kullanılan faktörler ve seviyeleri.

Faktörler	1. Seviye	2. Seviye
Başlangıç sinyali (s_{ij})	100	500
w	20	1300
v	2	12
R_{min}	0.30	0.95
R_{max}	1.05	1.75
Sinyal Artım Oranı	0.33	1.00
Komşu sayısı (q)	4	8
Komşuluk operatörü uygulama aralığı	3	6
Komşuluk operatörü uygulama sayısı	50	200
İterasyon Sayısı	100	300

Tablo 3: Deney tasarımı sonucu en uygun parametreler.

Parametre	Değer
Başlangıç sinyali (s_{ij})	500
R_{min}	0.30
R_{max}	1.75
w	1300
v	12
Komşu sayısı (q)	8
Sinyal Artım Oranı	0.33
İterasyon Sayısı	100
Komşuluk operatörü uygulama sayısı	50
Komşuluk operatörü uygulama aralığı	3

Tablo 4: Veri setlerine göre deney sonuçları.

Veri Seti	wi29	dj38	eil51	berlin52	st70	eil76	eil101	kroA100	lin105	Ch150	kroA200	xqg237
Düğüm Sayısı	29	38	51	52	70	76	101	100	105	150	200	237
Deney Sayısı	30	30	30	30	30	30	30	30	30	30	30	30
Bilinen En İyi Sonuç	27603	6656	426	7542	675	538	629	21282	14379	6528	29368	1019
KFA ile Bulunan En İyi	27603	6656	440	7542	720	562	698	22949	15529	7139	34716	1206
KFA ile Bulunan En Kötü	29074	6847	491	8264	836	620	792	25538	17145	7862	36730	1330
Sapma (%)	0	0	3.2	0	6.7	4.46	10.98	7.83	7.99	9.35	18.21	18.35
Süre (sn.)	1.09	1.99	4.68	5.57	15.04	15.05	46.30	56.45	73.99	267.84	1151.44	2416.64

Bu deney seti için komşuluk operatörünün uygulanması işleminin sağladığı iyileşme oranı deney sayısı bazında %83.3'tür.

Tablo 5: Komşuluk operatörünün iyileştirme performansı.

No	KO-yok	KO-var	Fark	No	KO-yok	KO-var	Fark
1	28615	28602	13	16	27946	27944	2
2	27750	27750	0	17	28557	28386	171
3	28617	27750	867	18	28386	28386	0
4	28887	28602	285	19	28557	28386	171
5	28790	28651	139	20	27878	27750	128
6	27750	27750	0	21	28386	28386	0
7	29128	29074	54	22	27946	27944	2
8	28074	27944	130	23	27616	27603	13
9	28074	27944	130	24	28388	28239	149
10	28387	28386	1	25	28362	28226	136
11	28074	27944	130	26	27946	27944	2
12	28752	28031	721	27	28478	28386	92
13	27946	27944	2	28	27770	27750	20
14	27878	27750	128	29	27750	27750	0
15	27946	27944	2	30	28634	28497	137

KO: Komşuluk operatörü; Fark: iyileşme miktarı.

3 Sonuçlar

Bu çalışmada literatürde üzerinde en çok çalışılmış kombinatoriyel optimizasyon problemlerinden biri olan GSP'nin çözümü için bir meta-sezgisel yöntem önerilmiştir. Geliştirilen yöntem kör farelerin doğadaki davranışlarından ve engelleri geçme stratejilerinden esinlenilerek tasarlanmış ve kör fare algoritması adı verilmiştir. Bu yöntem yeni bir meta-sezgisel olma özelliği taşımaktadır. Bu sezgiselin anlaşılması kolay, hesaplama adımları basit ve sadedir. Geliştirilen yöntemin başarısını görmek için literatürde yer alan farklı test problemleri üzerinde algoritma uygulanmıştır. Bunun için öncelikle KFA'da kullanılan parametrelerin en iyi değerleri Taguchi L32 tasarımı kullanılarak belirlenmiştir. Algoritma belirlenen parametre seti kullanılarak 12 farklı test probleminde uygulanmış ve umut verici sonuçlar elde edilmiştir. Önerilen algoritma henüz büyük boyutlu GSP'lerde rekabet edebilir olmamasına rağmen gelecekteki çalışmalar için cesaret vericidir.

Ayrıca, komşuluk operatörünün uygulanması işleminin etkili olduğu görülmüştür. Yapılan deneylerde bulunan çözümlerin %82'sinde komşuluk operatörü iyileştirme sağlamıştır.

Önerilen yöntem geliştirmeye açık bazı konular içermektedir. Örneğin, sinyal kayıp katsayısı w , statik olarak ele alınmıştır. Fakat düğümler arasındaki mesafe dikkate alınarak sinyal kayıp katsayısı dinamik bir şekilde belirlenirse daha iyi sonuçların elde edilebileceği öngörülmektedir. KFA parametrelerinden w , v ve düğüm çifti seçiminde kullanılan strateji iyileştirilerek daha kısa sürede çözümler elde edilebilir. GSP için geliştirilen bu yöntem farklı tiplerdeki GSP problemleri için uyarlanabilir. Farklı konulardaki atama, sıralama, çizelgeleme veya araç rotalama gibi optimizasyon problemlerinde yöntemin etkinliği araştırılabilir.

4 Teşekkür

Bu çalışma Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon birimi tarafından 2012FBE057 proje numarası ile desteklenmiştir.

5 Kaynaklar

- [1] Rego C, Gamboa D, Glover F, Osterman C. "Traveling Salesman Problem Heuristics: Leading Methods, Implementations and Latest Advances". *European Journal of Operational Research*, 211(3), 427-441, 2011.
- [2] Sallabi OM, El-Haddad Y. "An Improved Genetic Algorithm to Solve the Traveling Salesman Problem". *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 3(4), 984-987, 2009.
- [3] Laporte G. "The Travelling Salesman Problem: An Overview of Exact and Approximate Algorithms". *European Journal of Operational Research*, 59(2), 231-247, 1992.
- [4] Ahmadvand M, Yousefikhoshbakt M, Darani MN. "Solving the Travelling Salesman Problem by an Efficient Hybrid Metaheuristic Algorithm". *Journal of Advance in Computer Research*, 3(3), 75-84, 2012.
- [5] Yong W. "Hybrid Max-Min Ant System with Four Vertices and Three Lines Inequality for Traveling Salesman Problem". *Soft Computing*, 19(3), 585-586, 2015.
- [6] Mondal RN, Hossain SK, Saha SK. "An Approach for Solving Travelling Salesman Problem". *International Journal of Applied Operational Research*, 3(22), 15-26, 2013.
- [7] Davendra D. *Traveling Salesman Problem, Theory and Applications*. Intech, Croatia, Intech, 2010.
- [8] Cerny V. "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm". *Journal of Optimization Theory and Applications*, 45(1), 41-51, 1985.
- [9] Sureja NM, Chawda BV. "Random Travelling Salesman Problem Using SA". *International Journal of Emerging Technology and Advanced Engineering*, 2(4), 621-624, 2012.
- [10] Larranaga P, Kuijpers CMH, Murga RH, Inza I, Dizdarevic S. "Genetic Algorithms for the Traveling Salesman Problem: A Review of Representations and Operators". *Artificial Intelligence Review*, 13(2), 129-170, 1999.
- [11] Liu YH. "Different Initial Solution Generators in Genetic Algorithms for Solving the Probabilistic Traveling Salesman Problem". *Applied Mathematics and Computation*, 216(1), 125-137, 2010.
- [12] Wang Y. "The Hybrid Genetic Algorithm with Two Local Optimization Strategies For Traveling Salesman". *Computers & Industrial Engineering*, 70, 124-133, 2014.
- [13] Hopfield JJ, Tank DW. "Neural Computation of Decisions in Optimization Problems". *Biological Cybernetics*, 52, 141-152, 1985.
- [14] Leung KS, Jin HD, Xu ZB. "An Expanding Self-Organizing Neural Network for the Traveling Salesman Problem". *Neurocomputing*, 62, 267-292, 2004.
- [15] Masutti TAS, de Castro LN. "A Self-Organizing Neural Network Using Ideas From the Immune System to Solve the Traveling Salesman Problem". *Information Sciences*, 179(10), 1454-1468, 2009.
- [16] Fritzke B, Wilke P. "FLEXMAP-A Neural Network for the Traveling Salesman Problem With Linear Time and Space Complexity". *IEEE International Joint Conference on Neural Networks*, Singapore, 18-21 November 1991.
- [17] Dorigo M, Maniezzo V, Coloni A. "Positive Feedback as a Search Strategy". Dipartimento di Elettronica, Politecnico di Milano, Milano, Italy, Technical Report 91-016, 1991.

- [18] Dorigo M, Gambardella LM. "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem". *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66, 1997.
- [19] Dorigo M, Stützle T. *Ant Colony Optimization*, UK, The MIT Press, 2004.
- [20] Wang KP, Huang L, Zhou CG, Pang W. "Particle Swarm Optimization for Travelling Salesman Problem". *Machine Learning and Cybernetics*, 3, 1583-1585, 2003.
- [21] Pang W, Wang K, Zhou C, Dong, L. "Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem". *Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04)*, Wuhan, China, 14-16 September, 2004.
- [22] Goldberg EFG, Souza GR, Goldberg MC. *Particle Swarm for the Traveling Salesman Problem*. Editors: Gottlieb J, Günther RR. Evolutionary Computation in Combinatorial Optimization, 99-110, Berlin, Germany, Springer Berlin Heidelberg, 2006.
- [23] Shi XH, Liang YC, Lee HP, Lu C, Wang QX. "Particle Swarm Optimization-Based Algorithms for TSP and Generalized TSP". *Information Processing Letters*, 103(5), 169-176, 2007.
- [24] Chen WN, Zhang J, Chung HSH, Zhong WL, Wu WG, Shi YH. "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems". *IEEE Transactions on Evolutionary Computation*, 14(2), 278-300, 2010.
- [25] Karaboğa D. *Yapay Zekâ Optimizasyon Algoritmaları*. Ankara, Türkiye, Nobel Yayın Dağıtım, 2011.
- [26] Lucic P, Teodorovic D. "Transportation Modeling: An Artificial Life Approach". *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, Washington, USA, 4-6 November 2002.
- [27] Koçer HE, Akça MR. "An Improved Artificial Bee Colony Algorithm with Local Search for Traveling Salesman Problem". *Cybernetics and Systems*, 45(8), 635-649, 2014.
- [28] Feng X, Lau FCM, Yu H. "A Novel Bio-Inspired Approach Based on the Behaviour of Mosquitoes". *Information Sciences*, 233, 87-108, 2013.
- [29] Shah-Hosseini H. "The Intelligent water Drops Algorithm: A Nature-Inspired Swarm-Based Optimization Algorithm". *International Journal of Bio-Inspired Computation*, 1(1/2), 71-79, 2009.
- [30] Quaarab A, Ahiod B, Yang XS. "Discrete Cuckoo Search Algorithm for the Traveling Salesman Problem". *Neural Computing & Applications*, 24, 1659-1669, 2014.
- [31] Lengzhi S, Li Y. "An Improve Cuckoo Search Algorithm for Traveling Salesman Problems". *Applied Mechanics and Materials*, 651-653, 2291-2295, 2014.
- [32] Gündüz M, Kiran MS, Özceylan E. "A Hierarchic Approach Based on Swarm Intelligence to Solve the Traveling Salesman Problem". *Turkish Journal of Electrical Engineering & Computer Sciences*, 23(1), 103-117, 2015.
- [33] Chen SM, Chien CY. "Solving the Traveling Salesman Problem Based on the Genetic Simulated Annealing Ant Colony System with Particle Swarm Optimization Techniques, With Particle Swarm Optimization Techniques". *Expert System with Applications*, 38(12), 14439-11450, 2011.
- [34] Mahi M, Baykan ÖK, Kodaz H. "A New Hybrid Method Bases on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt Algorithms for Traveling Salesman Problem". *Applied Soft Computing*, 30, 484-490, 2015.
- [35] Geng, X, Chen Z, Yang W, Shi K, Zhao K. "Solving the Traveling Salesman Problem Based on Adaptive Simulated Annealing Algorithm with Greedy Search". *Applied Soft Computing*, 11(4), 3680-3689, 2011.
- [36] Dong G, Guo WW, Tickle K. "Solving The Traveling Salesman Problem Using Cooperative Genetic Ant Systems". *Expert Systems and Applications*, 39(5), 5006-5011, 2012.
- [37] Yang XS. "A New Metaheuristic Bat-Inspired Algorithm". *Studies in Computational Intelligence*, 284, 65-74, 2010.
- [38] Yang XS. "Bat Algorithm: Literature Review and Applications". *International Journal of Bio-Inspired Computation*, 5(3), 141-149, 2013.
- [39] Saji Y, Riffi ME. "Discrete Bat-Inspired Algorithm for Travelling Salesman Problem". *Complex Systems Second World Conference*, Agadir, Morocco, 10-12 November, 2014.
- [40] Kumbharana SN, Pandey GM. "Solving Travelling Salesman Problem Using Firefly Algorithm". *International Journal for Research & Advanced Technologies*, 2(2), 53-57, 2013.
- [41] Bouzidi M, Riffi MS. "Adaptation of The Harmony Search Algorithm To Solve The Travelling Salesman". *Journal of Theoretical and Applied Information Technology*, 62(1), 157-160, 2014.
- [42] Verma OP, Jain R, Chhabra V. "Solution of Travelling Salesman Problem Using Bacterial Foraging Optimisation Algorithm" *International Journal of Swarm Intelligence*, 1(2), 179-192, 2014.
- [43] Bouzidi A, Riffi MS. "Discrete Cat Swarm Optimization to Resolve the Traveling Salesman Problem". *Interneational Journal of Advanced Research in Computer Science and Software Engineering*, 3(9), 13-18, 2013.
- [44] Mo H, Xu L. "Biogeography migration algorithm for traveling salesman problem". *International Journal of Intelligent Computing and Cybernetics*, 4(3), 311-330, 2011.
- [45] Cai Y. "Artificial Fish School Algorithm Applied in a Combinatorial Optimization Problem". *International Journal of Intelligent Systems and Applications*, 2(1), 37-43, 2010.
- [46] Xing B, Gao WJ. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. 1st ed. Switzerland, Springer International Publishing, 2014.
- [47] Yang XS. *Nature Inspired Metaheuristic Algorithms*. United Kingdom, Luniver Press, 2010.
- [48] Taherdangkoo M, Shirzadi MH, Bagheri MH. "A Novel Meta-Heuristic Algorithm for Numerical Function Optimization: Blind, Naked Mole-Rats (BNMR) Algorithm". *Scientific Research and Essays*, 7(41), 3566-3583, 2012.
- [49] Taherdangkoo M, Shirzadi MH, Yazdi M, Bagheri MH. "A Robust Clustering Method Based on Blind, Naked Mole-Rats (BNMR) Algorithm". *Swarm and Evolutionary Computation*, 10, 1-11, 2013.
- [50] Wikipedia the Free Encyclopedia. "Animal Echolocation". http://en.wikipedia.org/wiki/Animal_echolocation (27.05.2014).

- [51] Jarvis UM, Sherman PW. "Heterocephalus Glaber". *Mammalian Species*, 706, 1-9, 2002.
- [52] Kimchi T, Terkel J. "Mole Rats (*Spalax ehrenbergi*) Select Bypass Burrowing Strategies in Accordance with Obstacle Size". *Naturwissenschaften*, 90(1), 36-39, 2003.
- [53] Kimchi T, Reshef M, Terkel J. "Evidence for the Use of Reflected Self-Generated Seismic Waves for Spatial Orientation in a Blind Subterranean Mammal". *Journal of Experimental Biology*, 208, 647-659, 2005.
- [54] WildeScreen Arkive. "Egyptian Blind Mole Rat (*Spalax aegyptiacus*)". <http://goo.gl/DqqTS5> (29.04.2015).
- [55] Dorigo M, Gambardella LM. "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem". *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66 1997.