

Yeni Bir Sezgisel Optimizasyon: Arama ve Kurtarma Algoritması ve Fonksiyon Optimizasyon Problemlerinin Çözümü¹

A New Heuristic Optimization: Search and Rescue Algorithm and Solving the Function Optimization Problems

Muhlis Özdemir², Tunçhan Cura³

Öz

Sezgisel teknikler, doğadan esinlenerek geliştirilmiş optimizasyon teknikleridir. Literatürde birçok sezgisel teknik bulunmakla birlikte, araştırmacılar tarafından doğa tabanlı ya da doğadaki canlı davranışlarının gözlemlenmesi ile her geçen gün yeni bir sezgisel teknik sunulmaktadır. Bu çalışmada, insan davranışlarından esinlenerek geliştirilmiş yeni bir sezgisel optimizasyon tekniği önerilmektedir. Arama ve Kurtarma Optimizasyon Algoritması (AKOA) olarak isimlendirilen bu yöntemin geçerliliğini kanıtlamak adına teknik, literatürde yer alan fonksiyon optimizasyonu test problemlerinin global minimumlarının bulunmasında kullanılmıştır. 21 minimizasyon problemi üzerinde gerçekleştirilen denemeler sonucunda, AKOA'nın Dinamik Rastgele Arama Tekniği ve Rastgele Seçim Yürüyüşü tekniklerine kıyasla oldukça rekabetçi olduğu görülmüştür.

Anahtar Kelimeler: Global minimum, fonksiyon optimizasyonu, sezgisel teknik, yapay zekâ, sürü zekâsı, arama ve kurtarma algoritması.

Abstract

Heuristic techniques are optimization methods that inspired by nature. Although there are many heuristics in the literature, a new heuristic technique is presented by researchers every day by observing nature-based or living behaviors in nature. In this study, a new heuristic optimization technique inspired by human behavior is proposed. In order to prove the validity of this method called Search and Rescue Optimization Algorithm (AKOA), the technique applied to find the global minimums of function optimization test problems in the literature. As a result of the experiments performed on 21 minimization problems, it has been observed that AKOA is quite competitive when compared to Dynamic Random Search Technique and Random Selection Walk Technique.

Keywords: Global minimum, function optimization, heuristics, artificial intelligence, swarm intelligence, search and rescue algorithm.

JEL: B16, B23, C13, C61

Submitted: 27 / 10 / 2020

Accepted: 10 / 12 / 2020

¹ Bu çalışma, Özdemir'in Yeni Bir Sezgisel Optimizasyon: Arama ve Kurtarma Algoritması ve Fonksiyon Optimizasyon Problemlerinin Çözümü adlı doktora tezinden türetilmiştir.

² Dr. Öğr. Üyesi., Gümüşhane Üniversitesi, muhliozdemir@gumushane.edu.tr, Orcid: <https://orcid.org/0000-0002-4921-8209>.

³ Prof. Dr., İstanbul Üniversitesi, tunchan@istanbul.edu.tr, Orcid: <https://orcid.org/0000-0001-5714-8629>.

Giriş

Arama ve kurtarma (AK), insanoğlunun afet ya da kaza nedeniyle karşılaştıkları zorluklarla mücadelelerini konu edinen bir kolektif çalışma örneğidir. Dünyada hemen hemen her gün, doğal olarak veya insan kaynaklı sebeplerle afetler/kazalar meydana gelmektedir. Bu bağlamda gerçekleştirilen AK çalışmaları, bu araştırmaya ilham kaynağı olmuş ve doğa tabanlı birçok sezgisel tekniğin aksine, insan davranışlarının modellenerek yeni bir sezgisel tekniğin bu çalışma ile literatüre kazandırılmasına katkıda bulunmuştur.

İnsanlar karar alma ve verme konularında oldukça hızlı davranış sergileyebilmelerine rağmen, karmaşık matematiksel modellerin çözümünde bilgisayarlar kadar hızlı olamamakta ya da hatalı sonuçlar üretebilmektedirler. Bu ve buna benzer nedenlerden ötürü insanlar doğadaki canlıların davranışlarını inceleyerek karşı karşıya kaldıkları problemlere çözüm aramışlar ve bunun sonucu olarak da çeşitli sezgisel teknikler geliştirmişlerdir.

Sezgisel teknikler kelimesinde yer alan sezgisel sözcüğü, yabancı literatürde “Heuristic” olarak kullanılmakta olup Yunanca “Heuriskein” fiilinden türetilmiştir. Bu fiil; bulmak, keşfetmek anlamlarına gelmektedir (Cura, 2008:14).

Sezgisel teknikler, çözüm uzayında yer alan bütün çözüm noktalarını aramak yerine kullanılan tekniğe göre sonlu sayıda deneme yaparak en iyi çözüme ulaşmayı amaçlamaktadır. Literatürdeki sezgisel tekniklerden hiç birisi kesin sonucu garanti edememekle birlikte; karşılığın sorunların önemli bir kısmının matematiksel model kurularak çözülemeyecek olmasından veyahut insan doğasının hata yapmaya müsait doğasından ötürü, ihtiyaca binaen geliştirilmiş tekniklerdir. Her ne kadar kesin sonucu garanti edemeseler de en iyi çözüme çok yakın sonuçların elde edilebilmesi, ayrıca matematiksel model kurularak çözmenin çok uzun zaman alacağı problem türlerinin çok daha kısa sürede ve daha az maliyetle gerçekleştirilebiliyor olması, sezgisel tekniklerin literatürde bu denli yaygın kullanım alanı bulmasında önemli bir ölçüttür. Farklı türlerde sezgisel tekniklerin geliştirilme sebebi ise, her bir tekniğin kendine has avantajlarının ve dezavantajlarının bulunmasından kaynaklanmaktadır. Sezgisel en iyileme algoritmaları biyoloji, fizik, sürü, sosyal, müzik ve kimya tabanlı olarak altı farklı grupta değerlendirilmektedir. Sürü zekâsı tabanlı algoritmalar kuş, balık, karınca ve arı gibi canlı sürülerinin doğada sergiledikleri içgüdüsel davranışların incelenmesi ile geliştirilmiş tekniklerdir (Akyol ve Alataş, 2012: 36). Bunların dışında; tepe tırmanış algoritması, tavlama benzetimi, tabu araştırma, genetik algoritma, yapay bağışıklık algoritması, diferansiyel gelişim algoritması, grafik renklendirme algoritması, iteratif yerel arama, değişken komşuluk araması, sezgisel rastgele optimizasyon tekniği, adaptif rastgele arama tekniği, dinamik rastgele arama tekniği, rastgele seçim yürüyüş tekniği vs. gibi farklı sezgisel teknik örneklerini arttırmak mümkündür.

Sezgisel teknikler ile bağlantılı olan bir diğer kelime “optimizasyon”dur. Genel anlamıyla optimizasyon, en iyiyi elde etme uğraşdır. Belirli koşullar altında çözüm aranan problemin türüne göre amaç, bir şeyleri maksimize etmek ya da minimize etmektir. İnsanoğlu birçok alanda optimize etme çabası sergilemektedir. Bu çaba; matematikten yönelem araştırmasına, bilgisayar bilimlerinden mühendislik bilimlerine, hatta yapay zekâ bilimlerine kadar birçok alanı kapsamaktadır (Akay, 2009: 25). Optimizasyon tekniklerinin uygulama alanının bu denli geniş olması nedeniyle literatürde optimizasyon için birçok tanımlama yapılmıştır. Bu nedenle kesin bir tanım vermek mümkün değildir (Özdemir, 2013: 13).

Matematiksel optimizasyon; genellikle matematiksel programlama, doğrusal olmayan programlama veya nümerik optimizasyon şeklindedir (Snyman, 2005: 2). Mühendislik ve bilgisayar bilimlerinde ise; kullanılan kaynaklar ve problem çözme süresi minimize edilmeye çalışılırken, performans maksimize edilmektedir. Üretim sektöründe ise üretilen ürünler; ürün proseslerinin etkinliği veya ürünlerin kalite ve albenisi optimize edilmeye çalışılmaktadır (Chiong, 2009: 3).

Bu çalışmada, insanların arama ve kurtarma davranışlarından esinlenilerek geliştirilmiş yeni bir sezgisel optimizasyon tekniği önerilmektedir. Bu kapsamda çalışmada ilk olarak gerçek hayatta gerçekleştirilen arama ve kurtarma çalışmalarından, AK türlerinden ve AK şekillerinden bahsedilecektir. İkinci bölümde ilgili literatürde yer alan çalışmalara yer verilecektir. Üçüncü bölümde; önerilen Arama ve Kurtarma Optimizasyon Algoritması detaylı olarak anlatılacak, konunun teorik alt yapısı oluşturulacak, sonrasında ise konu ile ilgili test problemleri ayrıntılı olarak ele alınacaktır. Dördüncü bölümde test problemleri sınanacak ve elde edilen sonuçlar, literatürde yer alan diğer tekniklerle elde edilmiş sınamaya sonuçları ile karşılaştırılacaktır. Son bölümde ise önerilen teknikle ilgili bulgular, yorum, görüş ve önerilere yer verilecektir.

1. Arama ve Kurtarma

Adından da anlaşılacağı üzere arama ve kurtarma (AK), esasen birbirinden ayrı iki farklı aktivitenin birlikte yürütülmesidir. Arama, afet/kaza gibi beklenmeyen acil durumların meydana gelmesi durumunda mahsur kalanların yerlerinin tespit edilmesine yönelik çalışmaları kapsamaktadır. Kurtarma ise, tehlike ve riske maruz bulunanların olay yerinden uzaklaştırılması, güvenli bir yere ulaştırılması çalışmalarını içermektedir.

Türk Dil Kurumunca afet, çeşitli doğa olaylarının sebep olduğu yıkım; kaza ise istem dışı veya umulmayan bir olay dolayısıyla bir kimsenin, bir nesnenin veya bir aracın zarara uğraması şeklinde tanımlanmaktadır⁴.

İnsan ya da doğa kaynaklı can, mülk kaybına ya da zararına neden olan afet ve kazaların önceden kestirilmesi mümkün olmamaktadır. Önlenmesi mümkün olmasa bile, ortaya çıkacak olan kayıpların en aza indirgenmesi afet ve kazalarda oldukça önem arz etmektedir. Bu önem, AK çabalarının kolektif bir hal almasına neden olmuştur. Önceden kestirilmesi mümkün olmayan gerçekliklerle karşı karşıya kalındığında, ortaya AK etkinliği çıkmaktadır. AK, bir kaza ya da afetten ötürü yaralı ya da ölüm tehlikesiyle karşı karşıya olan ve bu durumdan kendi başına kurtulamayan kazazedelerin kurtarılması etkinliğidir⁵.

12 Aralık 2001 tarihli resmi gazete yayımlanan "Türk Arama ve Kurtarma Yönetmeliği" 4. Madde'sinde ise Arama ve Kurtarma; hava ve deniz vasıtalarının karada, havada, su üstünde ve su altında tehlikeye maruz kalması, kaybolması veya kazaya uğraması hallerinde, bu vasıtadaki şahısların her türlü araç, özel teçhizat veya kurtarma birlikleri kullanılarak aranması ve kurtarılması işlemi olarak tanımlanmıştır⁶. Bu yönetmelik 1989 yılında yürürlüğe girmiş olup, 2001 yılında yeniden düzenlenmiştir.

Ülkemizin üç tarafının denizlerle kaplı olması ve zengin dere yataklarına, göl ve göletlere sahip olması nedeniyle su üzeri AK çalışmaları ülkemizde önem teşkil etmektedir. Bir başka husus ise ülkemizde bulunan fay hatları gerçeğidir. Olası bir deprem/zelzele durumunda, enkaz altı AK çalışmaları önem teşkil eden bir diğer konudur. Ülkemizde bulunan dağların yükseltileri ise dağcılık sporu ile ilgilenen sporcuların herhangi bir nedenle kaybolması, mahsur kalması gibi hadiselerle karşılaşılması durumunda AK faaliyetlerinin ne denli önemli olduğunu ortaya koymaktadır. Bu ve benzeri durumların söz konusu olması sebebi ile AK çalışmalarının önemi gün yüzüne çıkmış, bu doğrultuda çeşitli kurum veya kuruluşlarca bu ihtiyaca cevap verilmeye çalışılmıştır. Ülkemizde AK çalışmaları; Arama Kurtarma Teşkilatı (AKUT), T.C. Başbakanlık Afet ve Acil Durum Yönetimi Başkanlığı (AFAD), Türk Silahlı Kuvvetleri (TSK), Doğal Afet Arama Kurtarma Taburu (DAK), Arama Kurtarma Araştırma Derneği (AKA), İstanbul Büyükşehir Belediyesi Afet Koordinasyon Merkezi (AKOM), İstanbul Üniversitesi Doğal Afet Arama Kurtarma Ekibi (ISUDAK) vb. kuruluşlar tarafından gerçekleştirilmektedir

AK çalışmaları kapsamında birçok ülke, kendine özgü AK çalışmalarının nasıl gerçekleştirilmesi gerektiğine yönelik ulusal yönergeler oluşturmuştur. Bu yönetmeliklere örnek olarak Türk Arama ve Kurtarma Yönetmeliği, United States National Search and Rescue Manuel, Australian National Land Search Operations Manual, Search And Rescue Framework for the United Kingdom of Great Britain and Northern Ireland, Canadian National Search and Rescue Manual vb verilebilmektedir. Ülkeler arası koordinasyonun sağlanabilmesi için ise; International Civil Aviation Organization(ICA0), IAMSAR(International Aeronautical Maritime Organisation Search and Rescue), INSARAG(International Search And Rescue Advisory Group), OCHA(United Nations Office for the Coordination of Humanitarian Affairs), FEMA(Federal Emergency Management Agency) vb. gibi yönergeler yayımlanmıştır.

1.1. Arama ve Kurtarma Türleri

AK çalışmalarını temel olarak iki kategoride değerlendirmek mümkündür. Bunlar, kentsel AK ve kırsal AK çalışmalarıdır. Bunlara ek olarak; AK çalışmalarını genel hatlarıyla Kara, Deniz ve Hava olarak üçe ayırmak söz konusudur.

1.2. Arama ve Kurtarma Şekilleri

AK çalışmalarına konu olan canlı için iki mümkün durum söz konusudur. Bunlardan birisi kazazedenin yerinin bilinmesi, bir diğeri ise kazazedenin yerinin bilinmiyor olması durumudur. Eğer kazazedenin yerinin bilinmesi söz konusu ise, arama çalışmaları safhasına gerek kalmayıp doğrudan kurtarma çalışması safhası ile çalışmalara başlanmaktadır. Kazazedenin yerinin bilinmiyor olması durumunda ise AK çalışmaları koordineli bir şekilde gerçekleştirilmektedir. Ayrıca kazazedenin yerinin bilinmemesi durumunda, arama faaliyetlerinden en uygun olanı uygulanmaktadır.

Uluslararası Havacılık ve Denizcilik Arama ve Kurtarma (IAMSAR) Kılavuzu Cilt II'de yer alan arama çalışmaları; Daire Dilimi Sektör Araması, Genişleyen Kare Arama, İz Çizgi Araması, Paralel Tarama Araması, Kayan Çizgi Araması, Kontur Arama, Kıyı Çizgisi Araması şeklindedir.

IAMSAR kılavuzunda belirtilen arama yöntemleri yukarıdaki gibi iken, elde edilen bilgilere göre 11 adet arama söz konusudur. Genişleyen kare araması arama türüne burada yer verilmeyecek, dolayısıyla 10 adet aramadan bahsedilecektir. Aşağıda değinilecek olan arama türlerinin tamamı kara üzeri arama çalışmalarında bahsi geçen AK teknikleridir. Bahsi geçen arama türleri sırası ile Keşif Araması, Alan Taraması, Çark Arama Tekniği, Çift Spiral Arama

⁴ Türk Dil Kurumu Resmi Web Sitesi (http://tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.55463b9b12e296.95040032-Erişim Tarihi Temmuz 2020)

⁵ <http://www.biltek.tubitak.gov.tr/sandik/aramakurtarma/nedir.htm> - Erişim Tarihi Temmuz 2020

⁶ Resmi Gazete (2001), Türk Arama Kurtarma Yönetmeliği, 12 Aralık 2001 Tarih 24611 Sayılı Yönetmelik (<http://www.resmigazete.gov.tr/eskiler/2001/12/20011212.htm>-Erişim Tarihi Temmuz 2020)

Tekniği, Daralan Kare Arama, Kutu Arama Tekniği, S Pattern Arama Tekniği, Tekli Spiral Arama Tekniği, Zincir Arama Tekniği. Ayrıntılı bilgi için (Özdemir, 2018: 14-16).

2. Literatür Taraması

Bu bölümde, AK çalışmaları ile ilgili literatürde gerçekleştirilen çalışmalara yer verilecektir. Bu amaçla öncelikle; veri tabanlarından yerli ve yabancı literatürde 1995-2017 yılları arasında yer alan (2017 dahil) toplamda 138 makale ve bildiri incelenmiştir (Özdemir 2018: 32-65). A Comprehensive Review for the Varieties and Lacks of Search and Rescue Algorithms (Arama ve Kurtarma Algoritmalarının Eksiklikleri ve Çeşitleri Üzerine Kapsamlı Bir Araştırma) adlı çalışmada ise, yazarlar 2019 yılında son 20 yıla ait kapsamlı bir araştırma gerçekleştirmişlerdir. Araştırma için Google Scholar, ScienceDirect ve Web of Science veri tabanları kullanılmıştır. Buna göre; son yirmi yılda arama kurtarma alanında yapılan çalışma sayısının yaklaşık 200 olduğu tespit edilmiştir (Özdemir ve Çelikbilek, 2019). Bu çalışmalardan bazıları aşağıda sunulmaktadır.

Literatürde AK ile ilgili olarak hâlihazırda birçok çalışma bulunmasına ve problem çözümü için mevcut sezgiseller kullanılmasına rağmen; 2017 yılında yayımlanan ve bu çalışmaya benzer bir çalışma olan “*swarm robotics search & rescue: a novel artificial intelligence-inspired optimization approach* (Bakhshipour vd., 2017) (sürü robotik arama ve kurtarma: yeni bir yapay zekaya dayalı optimizasyon yaklaşımı)” isimli çalışmanın haricinde, Arama ve Kurtarma Optimizasyon Algoritması'nın varlığına rastlanmamıştır. 2017 yılına ait makale ile bu çalışma arasında farklılıklar ve benzerlikler bulunmaktadır. Bu benzerlikler ve farklılıklar aşağıdaki gibidir;

Bakhshipour vd.'nin gerçekleştirdiği çalışmada, *master* ve *slave* isminde iki tür robot bulunmaktadır. Bu robotlar kazazedenin aranmasında kullanılan sürü zekâsına sahip robotlardır. Algoritma, robot sayısının belirlenmesi ile başlamaktadır. *Master* robot sayısı 1 tanedir. *Master* robot belirlendikten sonra diğer robotlar *slave* robot olarak adlandırılmaktadır. Bu çalışmada önerilen Arama Kurtarma Optimizasyon Algoritmasında (AKOA) ise bir adet ekip lideri ile boyut sayısı ve fonksiyonda yer alan değişken sayısının baz alındığı sayıda arama kurtarma görevlisi (AKG) bulunmaktadır.

Slave robotlar *master* robot etrafında konumlanıp tesadüfi aramalar gerçekleştirirken; AKOA'da ise çözüm uzayı önce eş alanlara bölünmekte, ardından bu alanlarda yer alan tesadüfi noktalara AKG'ler yerleştirilmekte ve kendilerine ayrılan bu eş alanlarda tesadüfi aramalar yapmaktadırlar. Bu arama işlemini gerçekleştirirken AKG'ler için iki hareket söz konusudur. Bunlardan ilki, kendine tanımlı bulunan alanda kalmak kaydıyla rastgele arama yapmaya devam etmektir. İkincisi ise, önceden ekip liderine bildirilen uygunluk değerine göre AKG'ler arasında bilgi paylaşımında bulunulmasıdır.

Sürü robotik arama ve kurtarma optimizasyon algoritmasında (SRAKOA) *master* robot iterasyon boyunca *slave* robotlardan herhangi birisi olabiliyorken, AKOA'da ekip lideri tek ve sabittir. Yani AKG'lerden herhangi birisi iterasyon sürecinde ekip lideri haline gelmemektedir.

AKOA'da eş alanlara bölünen arama uzayı zamanla daraltılıyorken, SRAKOA çalışmasında daraltma işlemi yer almamaktadır.

Gerçek bir arama kurtarma faaliyetinde robotların kullanılması düşünüldüğünden, SRAKOA'da robotlar arasındaki iletişim Bluetooth ile sağlanmaktadır. AKOA gerçek hayat arama ve kurtarma faaliyetleri için önerilmediğinden; iletişim, iterasyon süresince çözüm kalitesinin iyileştiği durumlarda gerçekleştirilmektedir.

SRAKOA'da *slave* robotlar, *master* robot etrafında aramalar gerçekleştirip zaman zaman uzaklaşıp yaklaşmaktadır. AKOA'da ise benzer şekilde AKG'lerin birbirlerine yakınlaşıp uzaklaşmaları mümkündür.

SRAKOA'da diğer sezgisel yöntemlerden farklı olarak 3 yeni operatör kullanıldığı belirtilmiştir. Bu operatörler *accumulation* (birikim), *exploration* (keşif) ve *local search* (yerel arama) operatörleridir. Algoritmada kullanılan ilk operatör birikim/yığılma operatörüdür. Bu operatör; normal dağılım fonksiyonunu kullanarak çözümlerin yeni pozisyonlara taşındığı, her çözüm için aritmetik ortalama ve varyans faktörlerini hesaplamak için yeni bir yöntemin önerildiği bir operatördür. İkinci operatör keşif operatörüdür. Bu operatörün, *master-slave* konsepti sayesinde en iyi çözüme yaklaşma ve uzaklaşma yöntemini uygulayarak elde edilen çözümün en iyi çözüme doğru yaklaşmasını sağladığı belirtilmiştir. Üçüncü operatör ise yerel arama operatörüdür. Bu operatörün bilinen bir operatördür ve diğer iki operatörle birlikte ilk defa bu çalışmada kullanılmıştır. SRAKOA'da 3 farklı operatör yer almasına karşın, AKOA'da yalnızca yerel arama operatörü bulunmaktadır.

Her iki çalışmada da algoritma performansının ölçülmesinde, test problemleri kullanılmıştır. SRAKOA, 10 adet test problemi ve pratik bir elektrik sistemi problemine uygulanmışken; AKOA, 21 adet test problemine uygulanmıştır. SRAKOA'da yer alan 10 adet test probleminden ikisi bire bire AKOA'nın performansını ortaya koymak adına kullanılan test problemlerindedir. Bu test problemlerinden SRAKOA'da çözülen F11 isimli problem, bu çalışmada 11 numaralı test problemidir. F14 isimli problem ise bu çalışmada yer alan 13. test problemidir. SRAKOA ve AKOA, belirtilen iki problem

için de bilinen en iyiye ulaşımlardır. Her iki yöntemde de boyut sayısının büyük olduğu fonksiyonlar yer almaktadır. AKOA'da SRAKOA'dan farklı olarak boyutların eş alanlara bölümlendirilmesi söz konusudur.

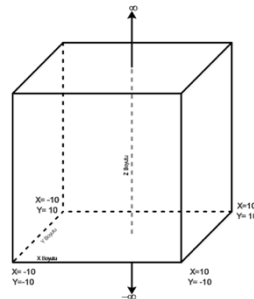
SRAKOA gerçek hayat arama ve kurtarma faaliyetlerinde sürü zekâsına sahip robotların kullanımı için önerilen bir teknik iken; bu çalışmada önerilen AKOA ise, halihazırda fonksiyon optimizasyon problemlerinin çözümü için önerilmiştir (Özdemir, 2018: 63-65).

3. Arama ve Kurtarma Optimizasyon Algoritması

Bu çalışmada önerilen Arama ve Kurtarma Optimizasyon Algoritmasında (AKOA), AK şekillerinden ikisi olan daralan kare arama ve zincir arama teknikleri baz alınmıştır. Kayıp vakası yaşanması durumunda, AK ekipleri öncelikli olarak kayıp kişinin bulunabileceği alanı yani arama yapılacak alanı belirlemekle işe başlamaktadır. Burada kayıp kişinin bilinen son pozisyonu (*LKP-Last Known Position*) dikkate alınmaktadır ve bu noktadan ipucu elde edilmeye çalışılmaktadır. Arama süreci ilerledikçe elde edilen ipucuna göre, aramanın gerçekleştirileceği alanın yönü de değişmektedir. Gerçekleştirilen bu arama, aslında bir yerel aramadır.

Yerel aramalar, optimizasyon problemlerinin çözüm uzayının araştırılmasında oldukça önemli araçlardır. Yerel aramada, belirli bir alanda daha iyi bir sonuç elde edilinceye değin araştırma yapılmakta ve kullanıcı tarafından saptanan iterasyon sayısına erişilinceye kadar bu çaba sürdürülmektedir. Yerel aramalar, optimum çözümü garanti etmemekle birlikte; sistematik olmayan bir şekilde çözüm uzayında araştırma yapmaktadır. Yerel arama, rastgele bir noktadan başlatılacağı gibi kullanıcı tarafından belirli bir noktadan da başlatılabilmektedir. Kendisine en yakın nokta bir sonraki arama noktası olmaktadır. Burada bir noktadan diğer bir noktaya yönelim, genellikle ceza fonksiyonları ile belirlenmektedir. Amaç fonksiyonuna göre çözüm uzayında daha iyi bir nokta elde edilmişse, artık bu noktadan araştırma yapılmaya devam edilmektedir (Schaerf ve Di Gaspero, 2001: 13-14).

AKOA'da ilk adım, çözümü aranan problemin çözüm uzayının sınırlarının belirlenmesidir. Bazı problemlerin çözüm uzayı $\pm\infty$ arasında yer alabilmektedir. Öyle ki; literatürde bu tür problemlerle karşılaşmak mümkündür. Zira bu çalışmanın test problemlerinde dikkate alınan 14. Problemi, bu yapıya örnek olarak göstermek mümkündür. Böyle bir durum ile karşılaşılması halinde yapılacak iş, yine kullanıcının belirleyeceği bir alandan başlanarak çözüm arayışına geçilmesidir. Çözümün ilerleyen safhalarında, kullanıcı tarafından belirlenen bu alanın dışına çıkılmasına izin verilerek $\pm\infty$ arasında çözüm arayışı gerçekleştirilmeye çalışılmaktadır.



Şekil 1. Arama Uzayı

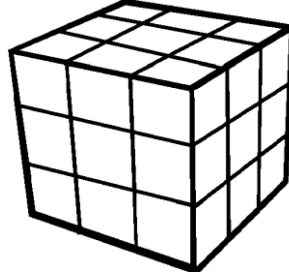
$f(x, y) = 2x + 3y$ gibi bir fonksiyonda x değeri 2, y değeri ise 1 iken fonksiyonun değeri $f(x, y) = 7$ olacaktır. Yani x ve y boyutunda değerler belirli olduğunda fonksiyona ait değer z boyutundadır. Burada 7 değeri z boyutunda bir değerdir. z boyutunda yer alan değer x ve y değişkenlerine göre değişeceğinden z boyutu $\pm\infty$ arasında olacaktır. Bu durum x ve y değişkenleri sınırlandırılmadığında geçerlidir.

Çözüm uzayına ait sınırlar belirlendikten sonra algoritmanın ikinci adımında her bir boyut, eş alanlara bölünmektedir. Eş alanlara bölme işlemi şu şekilde gerçekleştirilmektedir. İki değişkenli bir fonksiyonun minimumunu bulabilmek için üçüncü bir boyut söz konusu olacaktır. Dolayısıyla üç boyutlu bir arama uzayında AK alanının eş alanlara bölünebilmesi için önerilen sezgisel teknikte en az dört adet arama ve kurtarma görevlisinin (AKG) yer alması söz konusu olacaktır. Böylelikle her bir boyut iki eş alana, arama uzayı ise toplamda dört eş alana bölünecektir. Fonksiyona ait sonuç üçüncü bir boyutta yer almasına rağmen, yukarıdaki fonksiyonun minimumunu bulabilmek için iki boyutun sınırlarının belirlenmesi gerekecektir. Bu yüzden, boyut sayısı arttıkça arama uzayının eşit aralıklara bölünebilmesi için arama yapacak AKG sayısı da üstel olarak artacaktır. Yani fonksiyonda yer alan değişken sayısı üç olduğunda (dört boyutlu bir arama uzayı söz konusu olacağından) arama uzayını iki eş parçaya ayırabilmek için en az sekiz AKG'nin, değişken sayısı dört olduğunda ise en az on altı AKG'nin bulunması gerekecektir. Başka bir ifade ile; arama uzayını eşit aralıklara bölebilmek için, AKG

sayısının boyut derecesinden kökü alındığında tam sayı elde edilmelidir. Elde edilen bu tam sayı, her bir boyutun kaç eşit bölgeye, AKG sayısı ise arama uzayının toplamda kaç eşit alana ayrıldığı bilgisini sağlayacaktır. Bu durum, önerilen Arama ve Kurtarma Algoritmasının varsayımlarından birisidir. Burada n boyut ve Z^+ pozitif tam sayılar olmak üzere;

$$\sqrt[n]{AKG\ Sayısı} = Z^+ \quad (1)$$

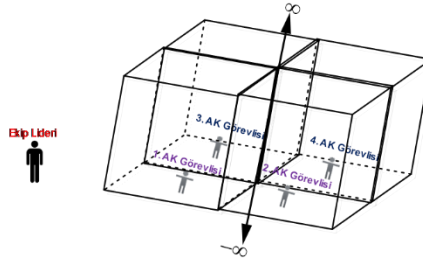
olacaktır. Örneğin dört boyutlu bir arama uzayında (üç değişkenli) her bir boyutun üçer eşit alana bölünmesi isteniyorsa 27 AKG gerekecektir. Böylelikle arama uzayı toplamda 27 eş alana bölünmüş olacak ve Şekil 2'de yer alan rubik kübe benzer bir görüntü oluşacaktır. Dördüncü boyut ise $\pm\infty$ arasında olacaktır.



Şekil 2. AKG Sayısına Göre Bölünmüş Çözüm Uzayı

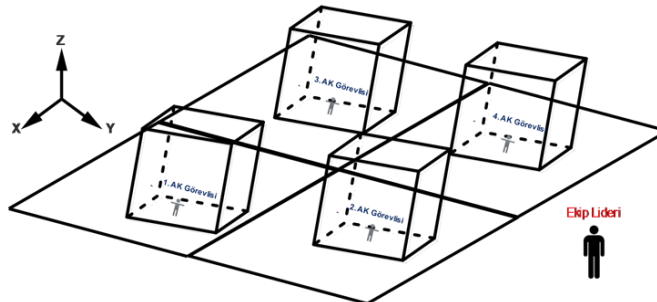
Değişken sayısının boyut sayısından bir eksik olduğu bilgisi daha önce de paylaşılmıştır. Anlam kargaşasına mahal vermemek adına bundan sonra literatürde yer aldığı şekilde, değişken sayısı boyut sayısı olarak kullanılacaktır. Yani iki boyutlu arama uzayı denildiğinde; fonksiyonda iki değişken yer almakta, fakat çözüm üçüncü bir boyutta aranmaktadır ifadesi anlaşılmalıdır. Dolayısıyla bölümlendirme iki değişkene ait boyutlarda yapılmakta, çözümün arandığı üçüncü boyutta doğal olarak herhangi bir bölümlendirme yapılmamaktadır.

Şekil 1'de verilen arama uzayı eşit aralıklara bölünüp AKG'ler bu alanlara yerleştirildiğinde aşağıda yer alan Şekil 3'teki gibi bir durum ortaya çıkacaktır. Her bir boyut iki eş parçaya ayrıldığında arama uzayı toplamda dört eş alana ayrılmış olacaktır.



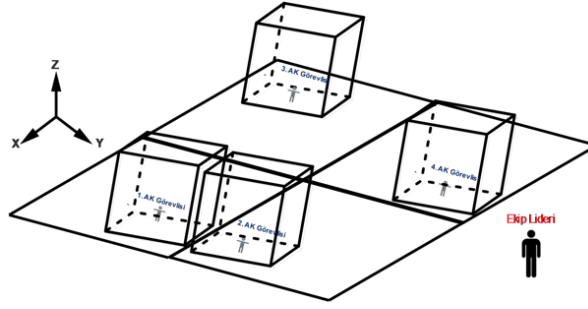
Şekil 3. AKG'lerin Arama Uzayına Dağıtılması

Şeklin daha iyi anlaşılabilmesi için; eş bölmelere ayrılan arama alanı koordinat düzlemine ayrı ayrı oturtulduğunda, Şekil 4'teki gibi bir görsel ortaya çıkacaktır. Bu bölümlendirilmiş alanda, her bir AKG'nin kendine ayrılmış arama alanı mevcuttur. Değişken sayısı kadar boyutun sınırları belirlidir. Buna karşın, çözümün yer aldığı boyut $\pm\infty$ arasındadır. AK faaliyetinde yer alan görevliler, kendilerine tahsis edilen bu alanda belirli bir noktadan veya tesadüfi bir noktadan başlayarak arama faaliyetinde bulunabileceklerdir.



Şekil 4. Detaylı Arama Alanı

Ekip lideri, AKG'lerden gelen bilgiler doğrultusunda, arama uzayında görevlilerin arama alanlarını kaydırmaktadır. Görevlilerin birbirlerine yaklaşmaları mümkün olduğu gibi birbirlerinden uzaklaşmaları da mümkündür. AKG'lerin hareketlerini gösteren görsel Şekil 5'te yer almaktadır.



Şekil 5. Hareket Halindeki AKG'ler

Daha sonra AKG'ler belirli bir iterasyon adedince arama işlemini sürdürüp ekip liderine elde ettikleri en iyi sonuçlar ile pozisyonlarını bildirmektedirler. Bu adımdan sonra ekip lideri, gelen pozisyonları değerlendirerek arama uzayını daraltmaktadır. Burada gerçekleştirilen daraltma işlemi, çalışmanın Arama ve Kurtarma Şekilleri başlığı altında yer alan Daralan Kare Arama Tekniği'ne benzer bir daraltma işlemidir. Daraltma işlemi, her bir görevlinin bulunduğu pozisyonların uç noktalarını kapsamaktadır. AKG'lerin x ve y boyutundaki bilgisinden hareketle; bu boyutlardaki alt limitler için en küçük olanı, üst limitler için ise en büyük olanı seçilerek yeni arama uzayının bilgisi AKG'ler ile paylaşılmaktadır. Ekip liderinden gelen bu bilgiler doğrultusunda AKG'ler, daraltılmış yeni arama uzayını eş alanlara bölecek şekilde yeni çözüm uzayında pozisyon olarak arama görevlerini sürdürmektedirler. Bu çaba, belirli bir iterasyon adedince gerçekleştirilmektedir.

Daraltma işleminin gerçekleştirilmesinde daraltma katsayısı kullanılmaktadır. Daraltma katsayısı ile alan daraltma hızı arasında ters bir ilişki söz konusudur. Bu ilişkiyi şu şekilde açıklamak mümkündür. Daraltma işlemi iterasyon adedince sürdürülecek, her bir iterasyonda daha dar bir alanda arama faaliyeti gerçekleştirilecektir. Daraltma katsayısının yüksek seçilmesi, yavaş yavaş arama alanının daraltılmasını sağlayacakken; düşük seçilmesi ise, arama alanının hızlı bir şekilde daraltılması anlamına gelecektir.

Yukarıda sözel olarak ifade edilen arama ve kurtarma algoritması özet halinde anlatılacak olursa; aşağıda yer alan alt başlıktaki adımlar söz konusu olacaktır.

3.1. AKOA'nın Adımları

1. Adım: Arama uzayı ve AKG sayısı belirlenir.

2. Adım: Arama uzayı, AKG sayısı kadar eşit alanlara bölünür. Burada amaç, her bir boyutun eşit bölünmesini sağlayarak kutu arama alanlarının belirlenmesini sağlamaktır.

3. Adım: Her bir AKG, bu alanlarda rastgele pozisyonlara atanır. Belirli bir iterasyon adedince AKG'ler arama uzayında araştırma yaparlar ve elde ettikleri en iyi pozisyonu ve uygunluk bilgisini kaydederler.

4. Adım: Her bir AKG elde ettiği en iyi pozisyon ve bu pozisyona ait uygunluk değerini ekip liderine bildirir.

5. Adım: Bütün AKG'lerden gelen pozisyon ve uygunluk değerlerinden en iyisi, ekip lideri tarafından belirlenir.

6. Adım: 4. adımda ekip lideri ile bilgi paylaşımında bulunan her bir AKG için iki hareketten birisi söz konusudur. Ya kendisine tahsis edilen arama alanında rastgele arama faaliyetine devam eder ya da ekip liderinin rulet tekerleği seçim yöntemine göre belirlediği kendisine komşu bulunan bir AKG ile bilgi paylaşımında bulunarak arama faaliyetini sürdürür. AKG'nin bu iki hareketten hangisini gerçekleştireceği bir olasılık parametresine göre belirlenir. Hareketlerin sonucunda eğer çözüm uzayı sonsuz değilse hareketin çözüm uzayı içerisinde kalması sağlanır. Aksi durumda yani çözüm uzayının sonsuz olması halinde ise, çözüm uzayının dışına çıkılmasına müsaade edilir. Elde edilen en iyi pozisyonlar ve bu pozisyonlara ait uygunluk değerleri ekip lideri ile paylaşılır.

7. Adım: Tüm AKG'ler arama faaliyetini tamamladığında; ekip lideri, AKG'lerin mevcut pozisyonlarını dikkate alarak her bir AKG'ye ait arama uzayını daraltır. Daraltma işleminde daraltma katsayısı kullanılır.

8. Adım: Ekip lideri, belirli aralıklarla AKG'lerin bulunduğu boyutlardaki alt limitler için en küçük olanı üst limitler için ise en büyük olanını seçerek AKG'lerin tamamını kapsayan tüm çözüm uzayını yeniden belirler ve bu bilgiyi tüm AKG'ler ile paylaşır. Ekip liderinden gelen bu bilgiler doğrultusunda AKG'ler, daraltılmış yeni arama uzayında eşit dağılarak arama görevlerini sürdürürler. Eşit dağılıma işlemi yeni arama uzayına göre 2. adımın bir tekrarıdır.

9. Adım: Tüm AKG'lerin daha önceden tanımlanmış süre boyunca gerçekleştirdikleri arama faaliyetleri tamamlandığında; ekip lideri tüm arama için sınırlandırılmış süreyi kontrol eder. Süre aşıldıysa arama faaliyeti sonlanır, aşılmadıysa 3. adıma dönlür.

Bahsi geçen bu adımlara ait tüm hesaplamalar aşağıda yer almaktadır.

3.1.1. Başlangıç Arama Bölgelerinin Oluşturulması

Arama kurtarma işleminin daha verimli olabilmesi için, problemin çözümünde yer alan her bir AKG'nin birbirinden farklı alanları aramasının sağlanması gerekmektedir. Böylece hem arama uzayında yer alan her bir bölge aranmış olacak, hem de AKG'lerin arama alanlarında kesişmeler olmayacağından daha etkili ve verimli bir arama gerçekleştirilmiş olacaktır. Bu da optimum çözüme daha hızlı bir şekilde ulaşılması anlamına gelmektedir.

i	:	$1, \dots, \text{boyut sayısı}$
j	:	$1, \dots, i. \text{ boyutun eş arama alanları sayısı}$
d_i	:	$i. \text{ boyutta yer alan her bir arama aralığının mesafesi}$
d_{ij}^{min}	:	$i. \text{ boyutun } j. \text{ arama aralığı alt sınırı}$
d_{ij}^{max}	:	$i. \text{ boyutun } j. \text{ arama aralığı üst sınırı}$
m_i	:	$i. \text{ boyutta yer alan arama aralığı sayısı}$
n	:	$\text{toplam boyut sayısı}$
p		position
ss_i^{min}	:	$i. \text{ boyutun alt sınırı}$
ss_i^{max}	:	$i. \text{ boyutun üst sınırı}$

olmak üzere arama alanlarının her bir boyuttaki aralık mesafesi d_i aşağıda yer alan denklem yardımıyla elde edilir;

$$d_i = \frac{ss_i^{max} - ss_i^{min}}{m_i} \quad (2)$$

Arama alanlarının alt ve üst sınırları ise denklem (3)'te yer alan formüller yardımıyla hesaplanmaktadır. Bu hesaplamalar neticesinde de birbiriyle kesişmeyen ve birbirinden farklı eş arama alanları elde edilebilecektir. Burada elde edilen arama alanları rubik kübe benzemektedir.

$$d_{ij}^{min} = \begin{cases} ss_i^{min} & 0 < j(\text{mod } m^{n-i+1}) \leq m^{n-i} \\ ss_i^{min} + d_i & m^{n-i} < j(\text{mod } m^{n-i+1}) \leq 2 * m^{n-i} \\ ss_i^{min} + d_i * 2 & 2 * m^{n-i} < j(\text{mod } m^{n-i+1}) \leq 3 * m^{n-i} \\ \vdots & \vdots \\ ss_i^{min} + d_i * (m - 1) & (m - 1) * m^{n-i} < j(\text{mod } m^{n-i+1}) \leq m * m^{n-i} \\ ss_i^{min} + d_i * (m - 1) & j(\text{mod } m^{n-i+1}) \equiv 0 \end{cases} \quad (3)$$

Her bir arama aralığının alt sınırının bulunmasından sonra, üst sınırları ifade eden d_{ij}^{max} değerlerinin hesaplanması daha kolay bir şekilde denklem (4) ile gerçekleştirilmektedir.

$$d_{ij}^{max} = d_{ij}^{min} + d_i \quad (4)$$

Bir sonraki adımda, birbiriyle kesişmeyen bu farklı arama alanlarının içinde yer alan rastgele pozisyonlara AKG'ler yerleştirilmektedir.

3.1.2. AKG'lerin Arama Kurtarma Bölgelerine Yerleştirilmesi

Arama uzayını arama kurtarma alanı olarak düşünürsek; arama ve kurtarma işlemi, arama uzayındaki çözümlere karşılık gelen rastgele pozisyonlara AKG'lerin yerleştirilmesi işlemi ile başlamaktadır. Rastgele pozisyonlara AKG'lerin yerleştirilmesi işlemi, her bir AKG'nin başlangıç arama bölgelerinde oluşturulan eş alanlara rastgele yerleştirilmesi ile sağlanmış olacaktır.

$$p_{ij} = d_{ij}^{min} + rand(0,1) * (d_{ij}^{max} - d_{ij}^{min}) \quad (5)$$

Burada, p pozisyonu göstermek üzere

$i=1, \dots, \text{boyut sayısı}$

$j=1, \dots, i. \text{ boyutun eş arama alanları sayısı'dır.}$

d_{ij}^{min} , i . boyutun j . arama aralığı alt sınırını, d_{ij}^{max} ise i . boyutun j . arama aralığı üst sınırını belirtmektedir. Bu fark, 0 ile 1 arasında yer alan rastgele bir sayı ile çarpılmaktadır. Elde edilen değer, i . boyutun j . arama aralığı alt sınırına ilave edilerek, AKG kendine ayrılan alanda rastgele bir pozisyona yerleştirilmektedir.

3.1.3. AKG'lerin buldukları pozisyonlara ait uygunluk değerlerinin hesaplanması

Her bir AKG'nin bulunduğu pozisyona göre ilgili fonksiyon dikkate alınarak uygunluk değeri hesaplanmaktadır.

3.1.4. Ekip Liderinin Rulet Tekerleği Seçim Yöntemine Göre Komşu AKG'leri Belirlemesi

Rulet tekerleği seçim yönteminde seçim olasılığının hesaplanması denklem (6)'da yer almaktadır.

O_i : uygunluk oranlarını göstermek üzere;

$$O_i = \frac{uygunluk_i^{-1}}{\sum_{i=1}^{AKG\ Sayısı} (uygunluk_i^{-1})} \quad (6)$$

Denklemden $uygunluk_i$ ile her bir AKG'nin i . pozisyonuna ait uygunluk değeri ifade edilmektedir. Her bir uygunluk değerinin tersi alınarak bu değer toplam değere oranlandığında; her bir AKG'ye ait uygunluk oranları hesaplanmış olmaktadır. Böylelikle, rulet tekerleği dilimlerinin genişlikleri elde edilecektir. Ekip lideri tarafından rulet tekerleği yöntemine göre sırası ile her bir AKG kendisi haricinde seçilen bir AKG ile eşleştirilmektedir. Bu eşleştirme sonrasında iki AKG'nin birbiri ile birlikte hareket etmesine izin verilmektedir. Önceden de belirtildiği üzere; bu iki AKG, birbirine yaklaşabildiği gibi birbirinden uzaklaşarak da hareket edebilmektedir.

3.1.5. Arama Alanlarının Ekip Lideri Tarafından Daraltılması

Başlangıç arama bölgeleri, kullanıcı tarafından belirlenen bir daraltma katsayısı ile çarpılarak daraltılmış arama alanları belirlenmektedir.

3.1.6. Daraltılan Arama Alanlarına AKG'lerin Yeniden Atanması

Daraltılan arama kurtarma bölgeleri dikkate alınarak AKG'ler arama alanlarına atanmaktadır.

3.2. AKOA'nın Özellikleri

Bu çalışmada önerilen AKOA ile ilgili temel özellikler şunlardır:

- basit ve esnek bir yapıya sahiptir,
- arama ve kurtarma faaliyetlerini gerçeğe yakın bir şekilde simüle eder,
- insan zekâsı ve davranışlarına dayalı bir algoritmadır,
- sürekli optimizasyon problemlerinin çözümü için geliştirilmiş olmasına rağmen kesikli problemlerin çözümü için de uyarlanabilir,
- arama kurtarma görevli sayısı, görevliler arası haberleşme oranı, arama limiti ve arama alanı, daraltma katsayısı gibi çok az sayıda parametreye sahiptir,
- çözüm uzayının bölümlendirilmesi ve bölgesel eş arama alanlarının oluşturulması gibi kabiliyetlere sahiptir,
- oluşturulan bölgesel eş arama alanları dinamik bir yapıya sahiptir,
- bölgesel ve global yerel arama kabiliyetine sahiptir,
- multithreading yani çok kanallı çalışmaya elverişli bir mimariye sahiptir.

4. Arama ve Kurtarma Optimizasyon Algoritmasının Geçerliliği İçin Kullanılan Test Problemleri ve Bulgular

Bu bölümde, arama ve kurtarma optimizasyon algoritmasının geçerliliğini test etmek için kullanılacak 21 probleme değinilecektir. Bu problemlerinin ilk 5'i Hamzaçebi ve Kutay'ın 2006 yılında gerçekleştirdikleri çalışmalarından derlenmiştir (Hamzaçebi ve Kutay, 2006: 1323-1333). Altıncı test problemi ise Yongjian ve Yumei'nin çalışmalarından alıntılanmıştır (Yongjian ve Yumei, 2007: 280-291). Geri kalan 15 problem ise, yine Hamzaçebi ve Kutay'ın 2007 yılında gerçekleştirdikleri çalışmalarında kullandıkları test problemleridir (Hamzaçebi ve Kutay, 2007: 2189-2198). Bu test problemlerine ait karşılaştırma sonuçları, Cura'nın 2010 yılında gerçekleştirdiği "A random search approach to finding the global minimum" isimli çalışmadan derlenmiştir (Cura, 2010: 179-190). 21 test problemine ait fonksiyon, boyut sayıları, değişken aralıkları ve teorik en iyi sonuç bilgileri Tablo 1'de sunulmaktadır.

Literatürde bu test problemlerinin çözümü için önerilen çalışmalar incelendiğinde; Sezgisel Rastgele Optimizasyon Tekniği (SROT/ Heuristic Random Optimization:HRO), Karınca Kolonisi Optimizasyon Tekniği (KKOT/ Ant Colony Optimization:ACO), Adaptif Rastgele Arama Tekniği (ARAT/ Adaptive Random Search Tecnique:ARSET), Dinamik Rastgele Arama Tekniği (DRAT/ Dynamic Random Search Tecnique:DRASET), Rasgele Seçim ve Yürüyüş algoritmalarının birleşimi olan Rastgele Seçim Yürüyüş Tekniği (RSYT/ Random Selection Walk:RSW) gibi algoritmalar ile çözüm önerilerinin olduğu görülmektedir(Özdemir M., 2017).

Her bir problem, diğer teknikler ile karşılaştırılabilmesi için literatürdeki fonksiyon aralıkları için sınıranmıştır. AKO Algoritması, Macintosh işletim sistemine sahip 2.7 GHz Quad-Core Intel Core i7 işlemci mimarili, 16 GB 2133 MHz LPDDR3 rastgele erişimli hafızaya sahip makinede Java'da kodlanarak iterasyonlar gerçekleştirilmiştir. Literatürde yer alan çalışmalarda çalışma zamanları bilgisi paylaşılmadığından; her bir probleme ait algoritma çalışma zamanları hesaplanmasına rağmen, karşılaştırmalarda dikkate alınamamıştır.

Test problemlerinin çözümünde AKOA içinde kullanılan her bir parametre ve bu parametrelere ait değerler Tablo 2'de verilmektedir. Tablo 2'den görüleceği üzere; problemlerin büyük bir çoğunluğu iki değişkenli problemlerdir.

Fonksiyonda yer alan boyut sayısı ve her bir boyutun kaç eş alana bölüneceği bilgisinden hareketle, algoritmada AKG sayısının belirlendiği daha önce belirtilmişti (bkz. Denklem 1). Bu çalışmada yer alan 6. problem 10 boyutludur. Bu ve benzeri fonksiyonlar değişken sayısı, yani boyut sayısı büyük olan problemlerdir. Örneğin 50 boyutlu bir fonksiyonun çözümü için her bir boyut iki eş alana bölünmek istendiğinde $2^{50}=1.125.899.906.842.634$ AKG'ye ihtiyaç duyulmaktadır. Bu kadar fazla sayıda AKG'nin kullanılması hem gerçek hayat açısından hem de kullanılan bilgisayar teknolojisi açısından çok da gerçekçi bir yaklaşım olmayacaktır. Bilgisayar bu sorunun üstesinden gelse dahi algoritmanın çalışma zamanı üstel olarak çok fazla artacağından, sadece çözüm uzayının bölümlendirilmesi bile çok uzun zaman alacaktır. Önerilen tekniğin uygulanması aşamasında, bahsedilen sorunla karşılaşmak söz konusu olacağından; bu sorunun üstesinden gelmek adına algoritma Java'da kodlanırken esneklik getirilmiştir. Bu bağlamda; problemde yer alan her bir boyutu eş alanlara bölmek yerine kullanıcı tarafından belirlenen boyutların eş alanlara bölünmesi, arta kalan boyutlarda ise bölümlendirme yapılmadan AK faaliyetinin sürdürülmesi uygun görülmüştür. Örneğin Problem 6'da yer alan 10 değişkenli fonksiyonun ilk 5 boyutunun 2 eş alana ayrılabilmesi için $2^5 = 32$ AKG yeterli olacaktır. Aynı şekilde problem 16'da yer alan fonksiyonun ilk 6 boyutu 3 eş alana bölünmek istenirse $3^6 = 729$ AKG gerekecektir. Algoritmaya getirilen esneklik sayesinde boyut sayısı yedinin üzerinde olan problemler için ilk yedi boyut eş alanlara bölümlendirilmiş (her bir boyut iki eş alana ayrılmış) kalan boyutlar eş alanlara ayrılmamıştır. Yedi boyuttan fazla olan problemlerde yapılan denemelerde maksimum AKG sayısının 128 olmasının sebebi budur (15. ve 21. problem hariç).

Tablo 1. Test problemleri

Problem No	Fonksiyon	Boyut	Değişken Aralığı	Teorik En İyi
1	$f(x) = \begin{cases} x^2, & \text{eğer } x \leq 1, \\ (x-3)^2 - 3, & \text{eğer } x > 1. \end{cases}$	1	[-10 10]	-3
2	$f(x) = \begin{cases} \left[x * \sin\left(\frac{1}{x}\right) \right]^4 + \left[x * \cos\left(\frac{1}{x}\right) \right]^4, & \text{eğer } x \neq 0, \\ 0, & \text{aksi taktirde} \end{cases}$	1	[-10 10]	0
3	$f(x,y) = \frac{(x-3)^8}{1+(x-3)^8} + \frac{(y-3)^4}{1+(y-3)^4}$	2	[-10 10]	0
4	$f(x,y) = 100(x-y^2)^2 + (1-x)^2$	2	[0 6]	0
5	$f(x) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	2	[-10 10]	-10
6	$f(x,y) = \frac{x}{1+ y }$	10	[-5 5]	0
7	$f(x,y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7$	2	[-1,28 1,28]	0
8	$f(x,y) = [\cos(2\pi x) + \cos(2.5\pi x) - 2.1] * [2.1 - \cos(3\pi y) - \cos(3.5\pi y)]$	2	[-1 1]	-16.09172
9	$f(x_1, x_2) = \left[0,002 + \sum_{j=1}^{25} \left(j + \sum_{i=1}^2 (x_i - a_{ij})^6 \right)^{-1} \right]^{-1}$	2	[-65,536 65,536]	0.998

$$a = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & -16 & -16 & -16 & 0 & 0 & 0 & 0 & 16 & 16 & 16 & 16 & 16 & 32 & 32 & 32 & 32 \end{bmatrix}$$

10	$f(x,y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10$	2	$x \in [-5 \ 10]$ $y \in [0 \ 15]$	0.3978873
11	$f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (4y^2 - 4)y^2$	2	$x \in [-3 \ 3]$ $y \in [-2 \ 2]$	-1.0316285
12	$f(x,y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] * [30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$	2	[-5 5]	3
13	$f(x,y) = [\sum_{i=1}^5 i \cos((i+1)x + i)] * [\sum_{i=1}^5 i \cos((i+1)y + i)]$	2	[-10 10]	-186.73091
14	$f(x,y,z) = \sum_{i=1}^5 (x(a_i)^y(b_i)^z - c_i)^2$ $a = 5 \ 3 \ 0.6 \ 0.1 \ 3 $ $b = 10 \ 1 \ 0.6 \ 2 \ 1.8 $ $c = 2.122 \ 9.429 \ 23.57 \ 74.25 \ 6.286 $	3	$[-\infty \ +\infty]$	8.01276
15	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	4	[-10 10]	0
16	$f(x) = \sum_{i=1}^{19} \left[(x_i^2)^{(x_i^2+1)} + (x_{i+1}^2)^{(x_i^2+1)} \right]$	20	[-1 4]	0
17	$f(x) = (\pi/20)[10 \sin^2(\pi x_1) + \sum_{i=1}^{19} ((x_i - 1)^2(1 + 10 \sin^2(\pi x_{i+1}))) + (x_{20} - 1)^2]$	20	[-10 10]	0
18	$f(x,y) = 100(y - x^2)^2 + (1 - x)^2$	2	[-10 10]	0
19	$f(x,y) = \exp\left\{\frac{1}{2}(x^2 + y^2 - 25)^2\right\} + \sin^4(4x - 3y) + \frac{1}{2}(2x + y - 10)^2$	2	[-5 5]	1
20	$f(x,y) = \left[12 + x^2 + \frac{1+y^2}{x^2} + \frac{x^2y^2+100}{(xy)^4}\right] * 0.1$	2	[0 10]	1.74
21	$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$	4	[-5 5]	0

Tablo 2. Parametrelere ait özet tablo

Problem	Değişken Sayısı	AKG Sayısı	Haberleşme Oranı	Arama Limiti	Daraltma Katsayısı
1	1	16	0.60	40	0.75
2	1	32	0.70	50	0.99
3	2	16	0.50	30	0.50
4	2	25	0.60	40	0.75
5	2	36	0.70	50	0.99
6	10	128	0.70	50	0.99
6	25	128	0.70	50	0.99
6	50	128	0.70	50	0.99
7	2	25	0.60	40	0.75
8	2	25	0.60	40	0.75
9	2	25	0.60	40	0.75
10	2	25	0.60	40	0.75
11	2	25	0.60	40	0.75
12	2	25	0.60	40	0.75
13	2	25	0.60	40	0.75
14	3	64	0.50	30	0.50
15	4	256	0.60	40	0.75
16	20	128	0.70	50	0.99
17	20	128	0.70	50	0.99
18	2	25	0.60	40	0.75
19	2	25	0.60	40	0.75
20	2	25	0.60	40	0.75
21	4	256	0.70	50	0.99

4.1. Test Problemi 1'e Ait Karşılaştırmalı Sonuçlar

Problem 1, [-10 10] aralığında beş farklı iterasyon ve iki farklı AKG sayısı için sınanmış ve Tablo 3'te yer alan sonuçlar elde edilmiştir. Tek değişkenli bir fonksiyon olmasından dolayı bu problem çok kısa sürede ve az sayıda AKG kullanılarak çözüme kavuşturulmuştur.

Tablo 3. Problem 1'e ait AKOA sonuçları

İterasyon	216	266	1.116	1.600	3.122
AKG Sayısı	4	4	16	16	16
Haberleşme Oranı	0,3	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
X	2,999	2,999	2,999	3,00	3,00
f(x)	-2,999	-2,999	-3,00	-3,00	-3,00
Süre(Sn)	0,015	0,015	0,016	0,016	0,201

Bütün denemelerde ya optimum ya da optimuma yakın sonuca erişilmiştir. Probleme ait x değeri 3 iken, fonksiyon minimum ve sonuç -3'tür. Bu problem 0,016 saniyede çözüme kavuşturulmuştur. Elde edilen bu en iyi sonuç, Tablo 4'te yer alan karşılaştırma tablosunda kullanılmıştır.

Tablo 4. Problem 1'e ait karşılaştırmalı sonuçlar

Algoritma	x	f(x)	İterasyon
SROT	3,000324	-2,9999998	1.000
ARAT	3	-3	1.000
KKOT	3	-3	500
RSYT (x ^{başlangıç} = 23, 9342)	2,999581	-2,9999998	500
RSYT (x ^{başlangıç} = 14, 0356)	3	-3	1000
AKOA	3	-3	1.600

Tablo 4 incelendiğinde, AKOA'nın en iyi sonuca ulaştığı hatta literatürde yer alan çözüm yöntemlerinin bazılarında daha iyi sonuçlar ortaya koyduğu açıkça görülmektedir. Diğer tekniklere ait sınav zamanları bilgisi mevcut olmadığından, süre bakımından karşılaştırma yapılamamıştır.

4.2. Test Problemi 2'ye Ait Karşılaştırmalı Sonuçlar

Problem 2, [-10 10] aralığında sınanmış ve Tablo 5'te yer alan sonuçlar elde edilmiştir.

Tablo 5. Problem 2'ye ait AKOA sonuçları

İterasyon	2.303	2.848	9.520	9.800	37.888
AKG Sayısı	4	4	16	16	32
Haberleşme Oranı	0,3	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
x	1,04E+00	-4,85E-11	-1,89E-12	-4,03E-14	-8,75E-82
f(x)	6,64E-21	4,26E-42	6,39E-48	2,63E-54	0,00
Süre(Sn)	0	0,01	0,031	0,031	1,126

Buna göre; bu problemde, 37.888 iterasyon ve 32 AKG ile optimum sonuca ulaşıldığı görülmektedir. Algoritmanın çalışma zamanı ise 1,126 saniyedir. Bu değer, Tablo 6'daki karşılaştırma tablosu için baz alınmıştır.

Tablo 6. Problem 2'ye ait karşılaştırmalı sonuçlar

Algoritma	x	f(x)	İterasyon
ARAT	1,90E-06	6,58E-024	10.000
KKOT	3,36E-10	8,16E-039	
RSYT (x ^{başlangıç} = 40, 1959)	-2,55E-74	1,47E-295	
ARAT	4,39E-08	1,85E-30	30.000

Algoritma	x	f(x)	İterasyon
KKOT	-1,57E-11	5,47E-44	
RSYT ($x^{baslangic} = 45,287$)	8,17E-82	0	
ARAT	-2,53E-11	2,21E-43	50.000
KKOT	7,79E-12	1,40E-45	
RSYT ($x^{baslangic} = 29,9729$)	-1,34E-81	0	
AKOA	-8,75E-82	0,00	37.888

4.3. Test Problemi 3'e Ait Karşılaştırmalı Sonuçlar

Problem 3, [-10 10] aralığında AKOA ile sınanmış ve Tablo 7'de yer alan sonuçlar elde edilmiştir. Bu problemde 24.988 iterasyon ve 16 AKG ile optimum sonuca ulaşıldığı görülmektedir. Bu problem için algoritmanın çalışma süresi ise 0,123 saniyedir.

Tablo 7. Problem 3'e ait AKOA sonuçları

İterasyon	14.140	16.000	24.988	46.720	61.900
AKG Sayısı	4	9	16	25	36
Haberleşme Oranı	0,3	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
x	3,00E+00	3,00E+00	3,00E+00	3,00E+00	3,00E+00
y	3,00E+00	3,00E+00	3,00E+00	3,00E+00	3,00E+00
f(x, y)	3,18E-38	1,45E-51	0,00E+00	0,00E+00	0,00E+00
Süre(Sn)	0,047	0,056	0,123	0,635	1,107

AKOA ile elde edilen yukarıdaki en iyi sonuç ile bu test problemine ait literatürde yer alan sonuçlar Tablo 8'deki gibidir.

Tablo 8. Problem 3'e ait karşılaştırmalı sonuçlar

Algoritma	x	y	f(x, y)	İterasyon
ARAT	3,0157	2,9999	3,71E-15	10.000
KKOT	3x2066E-09	3x2384E-09	2,62E-21	
RSYT($x^{baslangic} = 11,8192$ $y^{baslangic} = -27,2218$)	2,9991	2,9999	4,17E-25	
ARAT	3,0072	3	7,32E-18	30.000
KKOT	3	3	0	
RSYT($x^{baslangic} = 43,0038$ $y^{baslangic} = -41,6007$)	3,0005	3	6,73E-27	
ARAT	3,0015	3	5,04E-23	50.000
KKOT	3	3	0	
RSYT($x^{baslangic} = 25,5261$ $y^{baslangic} = -25,8986$)	2,9996	3	3,43E-28	
AKOA	3	3	0	24.988

Problem 3'e ait karşılaştırmalı sonuçlar tablosu incelendiğinde AKOA'nın diğer yöntemlere nazaran daha az iterasyon ile en iyi sonuca ulaştığı söylenebilir. Yukarıda yer alan üç farklı teknikten yalnız KKOT yönteminde 50.000 iterasyonla optimum sonuca ulaşıldığı görülmektedir. Yine bu problem için de AKOA ile en iyi sonuca ulaşıldığı tablodan anlaşılmaktadır.

4.4. Test Problemi 4'e Ait Karşılaştırmalı Sonuçlar

Problem 4, [0 6] aralığında AKOA ile sınanmış ve Tablo 9'da aşağıda yer alan sonuçlar elde edilmiştir.

Tablo 9. Problem 4'e ait AKOA sonuçları

İterasyon	15.792	26.656	28.376	30.816	53.101
AKG Sayısı	4	9	16	25	36
Haberleşme Oranı	0,3	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
x	1,00	1,00	1,00	1,00	1,00
y	1,00	1,00	1,00	1,00	1,00
f(x, y)	1,84E-09	1,65E-11	4,72E-12	0,00	0,00
Süre(Sn)	0,015	0,016	0,019	0,098	0,311

Bu problemde 30.816 iterasyon ve 25 AKG ile optimum sonuca ulaşılmıştır. Çözüm süresi ise 0,098 saniyedir. Bu değer, test problemine ait literatürde yer alan sonuçlar ile karşılaştırıldığında Tablo 10 elde edilmektedir.

Tablo 10. Problem 4'e ait karşılaştırmalı sonuçlar

Algoritma	x	y	f(x, y)	İterasyon
ARAT	0,99401	0,997	3,58E-05	10.000
KKOT	1,00021	1,00004	1,73E-06	
RSYT ($x_{\text{başlangic}} = 24, 7355$ $y_{\text{başlangic}} = 42, 3291$)	0,99999	0,99999	2,27E-27	
ARAT	1,0001	1,0001	2,03E-08	30.000
KKOT	1	1	5,68E-12	
RSYT ($x_{\text{başlangic}} = 43, 3763$ $y_{\text{başlangic}} = 23, 2853$)	1	1	5,21E-28	
ARAT	1	1	4,02E-16	50.000
KKOT	1	1	0	
RSYT ($x_{\text{başlangic}} = 43, 8923$ $y_{\text{başlangic}} = 46, 455$)	1	1	1,97E-31	
AKOA	1	1	0	30.816

Problem 4'e ait karşılaştırmalı sonuçlar tablosu incelendiğinde; AKOA'nın diğer yöntemlere nazaran daha az iterasyon ile en iyi sonuca ulaştığını söylemek mümkündür.

4.5. Test Problemi 5'e Ait Karşılaştırmalı Sonuçlar

Problem 5, [-10 10] aralığında AKOA ile sınanmış ve Tablo 11'de yer alan sonuçlar elde edilmiştir. Bu problemde 74.578 iterasyon ve 36 AKG ile optimum sonuca ulaşıldığı görülmektedir. Bu probleme ait algoritma çalışma zamanı 0,042 saniyedir.

Tablo 11. Problem 5'e ait AKOA sonuçları

İterasyon	14.656	30.132	47.808	64.400	74.578
AKG Sayısı	4	9	16	25	36
Haberleşme Oranı	0,3	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
x	-1,00E+16	-10,0	-10,0	-10,0	-10,00
y	3,41E+05	9,47E-02	-9,42E-02	-1,88E-02	0,00
f(x, y)	-1,00E+01	-10,0	-10,0	-10,0	-10,00
Süre(Sn)	0,009	0,013	0,021	0,03	0,042

Bu test problemine ait literatürde yer alan sonuçlar ile önerilen optimizasyon algoritmasına ait sonuçlar aşağıda yer alan Tablo 12'deki gibidir.

Tablo 12. Problem 5'e ait karşılaştırmalı sonuçlar

Algoritma	x	y	f(x, y)	İterasyon
ARAT	-9,9968	3,46E-009	-9,9968	10.000
KKOT	-9,9989	2,01E-004	-9,9989	
RSYT ($x_{\text{başlangic}} = 42,2649$ $y_{\text{başlangic}} = -31,5613$)	-9,773	3,42E-17	-9,773	
ARAT	-9,9996	-2,08E-018	-9,9996	30.000
KKOT	-9,9999	-6,05E-008	-9,9999	
RSYT ($x_{\text{başlangic}} = 49,8498$ $y_{\text{başlangic}} = 45,5905$)	-9,9016	6,52E-17	-9,9016	
ARAT	-10	6,67E-008	-10	50.000
KKOT	-10	8,07E-011	-10	
RSYT ($x_{\text{başlangic}} = -0,0093$ $y_{\text{başlangic}} = 48,8558$)	-9,9996	-6,57E-17	-9,9996	
AKOA	-10	0	-10	74.578

KKOT ve ARAT için başlangıç çözümlerinin $x=9$ ve $y=9$ olarak alındığı bilinmektedir. AKOA ise rastgele bir noktadan başlatılmıştır. Tablo 12 incelendiğinde; AKOA'nın diğer yöntemlere nazaran daha fazla iterasyon ile en iyi sonuca ulaştığı görülmektedir. Literatürde yer alan diğer üç teknikte elde edilen sonuçlar incelendiğinde ise, önerilen yöntemlerin üçünde de düşük iterasyon sayısında optimum sonuca çok yakın sonuçlar elde ettiği görülmüştür.

4.6. Test Problemi 6'ya Ait Karşılaştırmalı Sonuçlar

Bu fonksiyon, [-5 5] aralığında AKOA ile sınanmış ve Tablo 13'te yer alan sonuçlar elde edilmiştir.

Tablo 13. 10 boyutlu çözüm uzayında problem 6'ya ait AKOA sonuçları

İterasyon	55.296	62.498	142.432	258.048	340.192
AKG Sayısı	16	16	32	64	128
Haberleşme Oranı	0,30	0,40	0,50	0,60	0,70
Arama Limiti	10	20	30	40	50
Daraltma Kat Sayısı	0,005	0,25	0,50	0,75	0,99
x_1	1,01019	0,99890	0,99924	1,00032	1,00E+00
x_2	1,00934	0,99799	0,99842	1,00086	1,00E+00
x_3	1,01156	0,99605	0,99677	1,00090	1,00E+00
x_4	1,00821	0,99223	0,99349	1,00228	1,00E+00
x_5	1,01520	0,98425	0,98694	1,00334	1,00E+00
x_6	1,00993	0,96874	0,97394	1,00637	1,00E+00
x_7	1,02191	0,93879	0,94840	1,01288	1,00E+00
x_8	1,05953	0,88190	0,89913	1,02524	1,00E+00
x_9	1,12198	0,77823	0,80775	1,05005	1,00E+00
x_{10}	1,25784	0,60442	0,65090	1,10267	1,00E+00
$f(x_1, \dots, x_{10})$	1,27E-01	6,84E-02	5,10E-02	3,78E-03	5,13E-14
Süre(Sn)	0,115	1,120	0,134	0,404	1,474

Tablo 13'e göre 340.192 iterasyon ve 128 AKG sayısı ile elde edilen en iyi sonuç 5,13E-14'tür ve bu sonuç 1,474 Sn.'de elde edilmiştir. RSYT ile elde edilen sonuç ise 8,20E-04'tür ve 1,453 Sn.'de bu sonuca ulaşılmıştır. Dolayısıyla AKOA ile elde edilen bu sonuç RSYT ile elde edilen sonuçtan daha iyidir.

4.7. Test Problemleri 7-21'e Ait Karşılaştırmalı Sonuçlar

7. problem ile 21. problem arasında yer alan 15 probleme ait RSYT ve DRAT ile bulunan en iyi sonuçlar, iterasyon sayıları ve teorik en iyi sonuçlar Tablo 14'te sunulmaktadır.

Tablo 14. Problem 7 - 21 arası RSYT ve DRAT sonuçları

No	$\alpha_{\text{başlangic}}$	İterasyon	RSYT	DRAT	Teorik En İyi
7	2,56	2.501.000	0,00	0,00	0,00
8	2	2.501.000	-16,09172	-16,09172	-16,09172
9	131,072	2.502.000	0,998	0,998	0,998
10	20	2.501.000	0,39788735	0,39788737	0,3978873
11	5	2.502.000	-1,0316284	-1,0316284	-1,0316285
12	10	2.503.000	3,00	3,00	3,00
13	20	2.502.500	-186,73091	-186,73091	-186,73091

14	20	25.015.000	8,01275646286	8,01275646263	8,01276
15	20	25.050.000	1,28E-28	3,72E-12	0,00
16	5	25.050.000	0,00	2,45E-16	0,00
17	20	25.080.000	2,3558E-32	5,93E-12	0,00
18	20	2.501.500	2,8399E-29	3,9053E-15	0,00
19	10	2.506.000	1,0091	1,00	1,00
20	10	250.200	1,74415200558	1,74415200796	1,74
21	10	25.080.000	1,02E-11	8,17E-9	0,00

Tablo 14'teki RSYT ile elde edilen sonuçlar incelendiğinde; problem 15, 17, 18, 19 ve 21 için bilinen en iyi sonuca erişilemediği, bununla birlikte bu problemler haricinde kalan bütün problemlerde bilinen en iyi sonucun elde edildiği görülmektedir. DRAT ile elde edilen sonuçlar incelendiğinde ise; yine problem 15, 16, 17, 18, ve 21 için bilinen en iyi sonuca erişilemediği, ancak bu problemler dışındaki tüm problemlerde bilinen en iyi sonuca ulaşıldığı görülmektedir.

AKOA ile elde edilen sınav sonuçları ise Tablo 15'teki gibidir. AKOA'nın iterasyon sayısı farklılık göstereceğinden, sonuçları ayrı olarak Tablo 15'te verilmiştir.

Tablo 15. Problem 7 - 21 arası AKOA sonuçları

No	İterasyon	AKOA	Teorik En İyi
7	1.646.850	0,00	0,00
8	578.664	-16,09171	-16,09172
9	2.138.125	0,998	0,998
10	2.521.025	0,3978873	0,3978873
11	2.126.257	-1,0316285	-1,0316285
12	2.180.400	3,00	3,00
13	3.121.725	-186,73090	-186,73091
14	9.942.272	8,012756	8,01276
15	20.489.321	0,00	0,00
16	26.348.761	1,1E-234	0,00
17	20.058.368	2,35E-32	0,00
18	1.452.160	0,00	0,00
19	2.459.120	1,00	1,00
20	2.050	1,74	1,74
21	24.852.044	0,00	0,00

Tablo 15'te yer alan ve önerilen AKOA ile elde edilen sonuçlar incelendiğinde; problem 16 ve 17 için bilinen en iyi sonuca erişilemediği, bununla birlikte bu problemler haricinde kalan tüm problemlerde bilinen en iyi sonucun elde edildiği görülmektedir.

Tablo 14'teki RSYT ve DRAT sonuçları ile Tablo 15'teki AKOA'ya ait sonuçlar karşılaştırmalı olarak incelendiğinde ise;

15. Problemden RSYT ile 1,28E-28 sonucu, DRAT ile 3,72E-12 sonucu elde edilmiştir. Bu sonuçlar bilinen en iyiden daha kötü sonuçlarken AKOA ile 0,00 sonucu ile bilinen en iyi sonuca ulaşılmıştır.

16. problemde RSYT ile 0,00 sonucu, DRAT ile 2,45E-16 sonucu elde edilmişken, AKOA ile 1,1E-234 sonucu hesaplanmıştır. Bu sonuç DRAT 'a oranla daha iyi bir sonuç iken, RSYT 'ye nazaran daha kötü bir sonuçtur. Ancak 1,1E-234 değeri sıfıra oldukça yakındır. 17. problemde RSYT ile 2,3558E-32 sonucu, DRAT ile 5,93E-12 sonucu elde edilmişken, AKOA ile 2,3558E-32 sonucu bulunmuştur. Bu sonuç DRAT'a oranla daha iyi bir sonuç olmasına rağmen, ne RSYT ile ne de AKOA ile bilinen en iyi sonuca ulaşılmıştır. 16. ve 17. problemlerde bilinen en iyiden daha kötü sonuçlar elde edilmesine rağmen bu sonuçlar bilinen en iyiyeye oldukça yakın sonuçlardır.

18. problemde RSYT ile 2,8399E-29 sonucu, DRAT ile 3,9053E-15 sonucu elde edilmişken, AKOA ile bilinen en iyi sonuç olan 0,00 sonucu elde edilmiştir. Elde edilen sonuç bakımından AKOA'ya ait sonuç, RSYT ve DRAT'dan daha iyidir.

19. problemde RSYT ile 1,0091 sonucu, DRAT ile 1,00 sonucu elde edilmişken; AKOA ile de 1,00 sonucu hesaplanmıştır. Bu problem için RSYT ile elde edilen sonuç önerilen teknik ve DRAT'dan daha kötüdür.

20. problemde RSYT ve DRAT ile optimum sonuca çok yakın sonuçlar elde edilmişken; AKOA ile optimum sonuç elde edilmiştir.

21. problemde RSYT ile 1,02E-11 sonucu, DRAT ile 8,17E-9 sonucu elde edilmişken, AKOA ile 0,00 sonucu bulunarak, teorik en iyi sonuca ulaşılmıştır.

Sonuç ve Değerlendirme

İnsanoğlu, medeniyetler boyunca sürekli kendini yenileme ihtiyacı duymuş ve bu ihtiyacı karşılayabilmek için her alanda gelişim sergilemiştir. Optimizasyon teknikleri de zaman içinde gelişim göstermiştir. Literatürde çok sayıda sezgisel teknik

bulunmaktadır. Ancak her geçen gün yeni bir sezgisel teknik literatüre katılmaktadır. Bunun nedeni, bir sezgisel tekniğin bütün problemler için yeterli kalitede çözüm üretilmediğidir. Başka bir ifade ile, bütün problemlere uygulanabilen genel geçer bir sezgisel teknikten bahsetmek mümkün değildir. Yeni tekniklerin kazanımı, karşılaşılan problemlere farklı yöntemlerle yaklaşarak elde edilen çözüm kalitesinin en iyilenmesini amaçlamaktadır. Bu çalışmada önerilen Arama Kurtarma Optimizasyonu Algoritması (AKOA) da yeni bir sezgisel teknik olmaya adaydır.

Önerilen bu yeni teknik ile literatürde yer alan 21 farklı fonksiyon optimizasyon problemine çözüm aranmış ve elde edilen sonuçlardan, önerilen algoritmanın oldukça rekabetçi olduğu, birçok problemde optimum çözüme ulaşılabilirdiği gözlemlenmiştir. Ayrıca önerilen AKOA; Sezgisel Rastgele Optimizasyon Tekniği (SROT), Karınca Kolonisi Optimizasyon Tekniği (KKOT), Adaptif Rastgele Arama Tekniği (ARAT), Dinamik Rastgele Arama Tekniği (DRAT) ve Rastgele Seçim Yürüyüş Tekniği (RSYT) gibi çeşitli tekniklerle karşılaştırıldığında da sonuçlar tatmin edici hatta kimi zaman üstün çıkmıştır.

Bunun dışında; önerilen algoritmada sağlanan esneklik sayesinde, algoritmanın çalışma zamanı makul seviyelere indirilmiştir. AKG sayısının düşürülmesi, elde edilen çözüm kalitesinin azalmasına neden olmamış, yine de oldukça rekabetçi sonuçlar elde edilmiştir. Ancak, çözüm süreleri bu çalışmada karşılaştırma amaçlı kullanılamamıştır. Bunun nedeni, bahsi geçen diğer tekniklere ait sinama zamanları bilgisine ulaşılabilmesidir.

Algoritmanın geliştirilip test edilmesi aşamasından sonra önerilen tekniğin multithreading yani çok kanallı çalışma için elverişli bir yöntem olduğu gözlemlenmiştir. Algoritma ile gerçekleştirilecek ilerleyen çalışmalarda, bu çok kanallı yapısının geliştirilmesi düşünülmektedir. Ayrıca AKOA'da kullanılan parametreler için parametre ayarlama çalışmasının yürütülmesi de ileriki çalışmalarda ele alınacak başka bir başlıktır.

Ek olarak; AKOA'nın bilim çevresince kabul gören, yaygın kullanılan bir sezgisel teknik olabilmesi için, ilerleyen çalışmalarda farklı problem türlerine uygulanarak geliştirilmesi gerekmektedir.

Kaynakça

- Akay B. (2009). Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi Algoritmasının Performans Analizi, Yayınlanmamış Doktora Tezi, Fen Bilimleri Enstitüsü, Kayseri, Türkiye.
- Akyol S. ve Alataş B. (2012). Güncel Sürü Zekâsı Optimizasyon Algoritmaları", Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi.
- Bakhshipour, M., Ghadi, M. J., & Namdari, F. (2017). Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach. *Applied Soft Computing*, 57, 708-726.
- Bakhshipour, M., Rokrok, E., Namdari, F., & Sedaghat, M. Optimal DG and capacitor allocation along with network reconfiguration using Swarm robotics search & rescue algorithm. In 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI) (pp. 136-144). IEEE.
- Chiong, R. (Ed.). (2009). Nature-inspired algorithms for optimisation (Vol. 193). Springer.
- Cura, T. (2008). Modern sezgisel teknikler ve uygulamaları. Papatya Yayıncılık Eğitim.
- Cura, T. (2010). A random search approach to finding the global minimum. *Int. J. Contemp. Math. Science*, 5(4), 179-190.
- Hamzaçebi, C., & Kutay, F. (2006). A heuristic approach for finding the global minimum: Adaptive random search technique. *Applied Mathematics and Computation*, 173(2), 1323-1333.
- Hamzaçebi, C., & Kutay, F. (2007). Continuous functions minimization by dynamic random search technique. *Applied Mathematical Modelling*, 31(10), 2189-2198.
- Özdemir M. (2013). Zaman Kısıtı Altında Takım Oryantiring Problemlerinin Yapay Arı Kolonisi Yaklaşımı ile Çözümü", Yayınlanmamış Yüksek Lisans Tezi.
- Özdemir, M. (2017). Particle swarm optimization for continuous function optimization problems. *International Journal of Applied Mathematics, Electronics and Computers*, 5(3), 47-52.
- Özdemir, M. (2018). Yeni Bir Sezgisel Optimizasyon: Arama ve Kurtarma Algoritması ve Fonksiyon Optimizasyon Problemlerinin Çözümü. Yayınlanmamış Doktora Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, Turkey.
- Özdemir, M., Çelikkalek Y. (2019). A Comprehensive Review for the Varieties and Lacks of Search and Rescue Algorithms. IV. International New Media Conference: Digital Transformation and Innovation, April 25-26, İstanbul, Turkey.

Schaerf, A., & Di Gaspero, L. (2001). Local search techniques for educational timetabling problems. In Proceedings of the 6th International Symposium on Operational Research (SOR-01), Preddvor, Slovenia (pp. 13-23).

Snyman, J. A. (2005). Practical mathematical optimization (pp. 97-148). Springer Science+ Business Media, Incorporated.

Yongjian, Y., & Yumei, L. (2007). A new discrete filled function algorithm for discrete global optimization. Journal of Computational and Applied Mathematics, 202(2), 280-291.

İnternet Erişim Linkleri

Resmi Gazete (2001), Türk Arama Kurtarma Yönetmeliği, 12 Aralık 2001 Tarih 24611 Sayılı Yönetmelik (<http://www.resmigazete.gov.tr/eskiler/2001/12/20011212.htm>-Erişim Tarihi Temmuz 2020)

Türk Dil Kurumu Resmi Web Sitesi (http://tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.55463b9b12e296.95040032-Erişim Tarihi Temmuz 2020)

<http://www.biltek.tubitak.gov.tr/sandik/aramakurtarma/nedir.htm> - Erişim Tarihi Temmuz 2020

Extended Abstract

Aim and Scope

Search and rescue algorithm (SaRA), is a very new candidate heuristic technique, which is inspired by human behavior for search and rescue (SaR) efforts rather than being nature based like any other heuristic optimization technique. SaR, arise when humankind faces major natural disasters like tsunamis, earthquakes, hurricanes, avalanches, maritime disasters, or human based accidents like ship or aviation accidents. In such events, SaR operations are carried out for pinpointing, evacuating, and delivering medical supplies for the victims. Almost every day, people have this kind of misfortune. It is not possible to foresee the amount of the casualty or property loss caused by the disasters or accidents. Therefore, it is very crucial to reduce losses to the minimum. With the aim of diminishing this misfortune and obscure of the situations, the practitioners and researchers conduct studies which can be classified as tactical, technological, statistical, retrospective, medical, informative and explanatory. Another important classification is information systems or decision support systems for the SaR operations. Besides, according to the place or position of the operations, SaR studies can be divided into three main categories; land, sea, and air. Members in the SaR operations can be either professionals or volunteers according to the importance, urgency and vitality of the situations. Especially in more vital SaR operations with mix members, being organized and conducting the operations properly are more important than the other situations.

Methods

SaRA's components consist of a team whom members are a team leader and search and rescue workers (SaRWs). The first step of SaRA, starts with defining the solution space. In SaRA, solution space should be divided into sub-spaces equally. In order to divide the solution space in equal intervals, number of SaRWs must be an integer value.

Depicted below figures show that a solution space, and equally sub-spaced solution space.

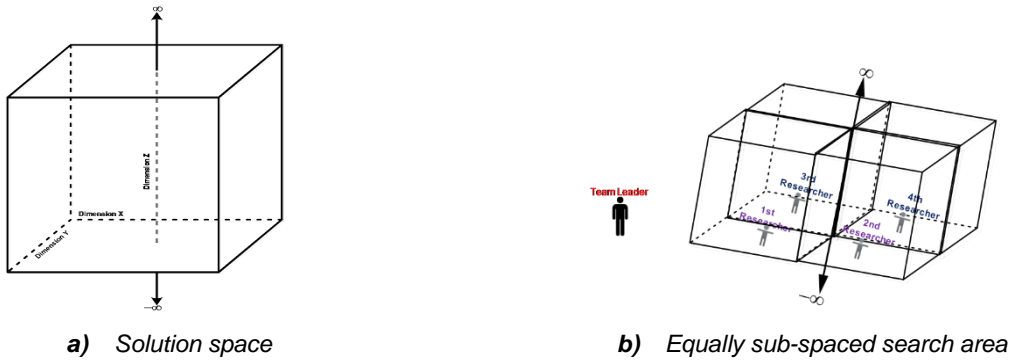


Figure 1. Solution space for SaR

When the search area is divided into equal panes and placed separately on the coordinate plane, appearance will be as follows:



Figure 2. Search procedure

Every SaRWs involved in the search and rescue activity, will be able to search in their area allocated for them and can start searching from a specific point or a random point. After each SaRW placed in a random or specific position, a fitness value is evaluated for a defined certain number of iterations. The team leader is informed about these fitness values by SaRWs. In line with the information received from the SaRWs, the team leader records the best positions and the fitness values of this positions. By making an assessment among this information, team leader records the best of the position and fitness values of all SaRWs. After the relevant information is sent to the team leader, there are two movements for SaRWs. The first of these is to continue making random searches in the defined area. The second is sharing information between SaRWs according to the fitness value previously reported to the team leader. The team leader shifts the search fields of the SaRWs in the search space according to the received information. It is possible for the SaRWs to approach as well as to get away from each other (see fig.2-d). This process repeats as the pre-defined number of iterations. After stopping criterion is met, the team leader straitens the solution space. Narrowing procedure is like one of the search and rescue expanding square search pattern called narrowing square search pattern. The narrowing process covers the end points of the positions, where each SaRW is located. Based on the information of the SaRWs in the dimension of x and y, team leader shares the information of the new search space with the SaRWs by choosing the smallest for the lower limits and the largest for the upper limits for two-dimensional search space. According to the information from the team leader, SaRWs continue their search duties by taking positions in the new solution space, dividing this narrowed new search space into identical areas. This effort is made by a certain number of iterations. A narrowing constant is used when narrowing process conducted. It is obvious that there is a negative correlation between the narrowing constant and area narrowing speed.

Creating Initial Search Areas

In order for the search and rescue process to be more efficient, each SaRW should be assigned to different search areas. Thus, each region in the search space will be searched and a more effective and efficient search will be carried out since there will be no crosses in the search areas of SaRWs. This leads us to reach the optimum solution much more quickly. Where;

i	: 1, ..., number of dimension
j	: 1, ..., number of i^{th} dimension's equal search area
d_i	: i^{th} dimesion's search interval distance
d_{ij}^{min}	: i^{th} dimension's j^{th} search area lower bound
d_{ij}^{max}	: i^{th} dimension's j^{th} search area upper bound
m_i	: number of i^{th} dimension search interval
n	: total dimension number
ρ	: position
ss_i^{min}	: i^{th} dimension's lower bound
ss_i^{max}	: i^{th} dimension's upper bound

The interval distance of each dimension of the search fields is obtained by the Equation (1).

$$d_i = \frac{ss_i^{max} - ss_i^{min}}{m_i} \quad (1)$$

The upper and lower limits of the search areas will be calculated with the help of the Equation (2). As a result of these calculations, different peer search fields that do not cross each other can be obtained. When the search fields are combined, it is very likely to be a Rubik's cube.

$$d_{ij}^{min} = \begin{cases} ss_i^{min} & 0 < j(\text{mod } m^{n-i+1}) \leq m^{n-i} \\ ss_i^{min} + d_i & m^{n-i} < j(\text{mod } m^{n-i+1}) \leq 2 * m^{n-i} \\ ss_i^{min} + d_i * 2 & 2 * m^{n-i} < j(\text{mod } m^{n-i+1}) \leq 3 * m^{n-i} \\ \vdots & \vdots \\ ss_i^{min} + d_i * (m - 1) & (m - 1) * m^{n-i} < j(\text{mod } m^{n-i+1}) \leq m * m^{n-i} \\ ss_i^{min} + d_i * (m - 1) & j(\text{mod } m^{n-i+1}) \equiv 0 \end{cases} \quad (2)$$

After finding the lower limit of each search range, the calculation of d_{ij}^{max} values that express the upper limits is more easily done with the Equation (3).

$$d_{ij}^{max} = d_{ij}^{min} + d_i \quad (3)$$

In the next step, the process of assigning SaRWs in random positions within these different search areas that do not intersect with each other is performed.

Assigning the SaRWs to the random points

If the search space considered as the search and rescue area, the search and rescue process starts with the placement of SaRWs in random positions corresponding to the solutions in the search space. The process of placing SaRWs in random positions is achieved by placing each SaRWs randomly in the peer fields created in the initial search regions.

$$p_{ij} = d_{ij}^{min} + rand(0,1) * (d_{ij}^{max} - d_{ij}^{min}) \quad (4)$$

Determining Neighboring SaRWs According to Roulette Wheel Selection Method

According to the roulette wheel selection method, the calculation of the probability of selection is given in the equation (5).

$$f_i = \frac{fitness_i^{-1}}{\sum_{i=1}^{SaRWs} (fitness_i^{-1})}$$

Findings

The proposed SaRA is applied on 21 function optimization test problems to find the global minimums and competitive results were obtained when compared to Dynamic Random Search Technique and Random Selection Walk Technique.

Conclusion

In this study a new optimization algorithm named Search and rescue algorithm (SaRA) is proposed. This algorithm is a very new candidate heuristic technique, which is inspired by human behavior for search and rescue (SaR) efforts rather than being nature based like any other heuristic optimization technique.

Search and rescue algorithm's some of the features listed below:

- Has a simple and flexible structure.
- Simulates search and rescue activities in a similar way.
- It is based on human intelligence and behavior.
- Although it was developed for the solution of continuous optimization problems, it can also be adapted for the solution of discrete problems.
- It has very few parameters such as the number of search and rescue workers, the communication ratio, the search limit and the search area narrowing constant.
- It has capabilities such as segmentation of the solution space and the creation of regional peer search areas.
- Regional peer search fields have a dynamic structure.
- Has local and global search capability.
- It has a convenient architecture which is suitable for multithreading.

After the development and testing of the algorithm, it was observed that the proposed technique is a convenient method for multithreading. In the further studies to be carried out with the algorithm, it is planned to develop this multi-channel

structure. Furthermore, the execution of the parameter adjustment for the parameters used in SaRA is another topic that will be discussed in future studies.

In addition, in order for SaRA to be a heuristic technique that is accepted and widely used by the scientific community, it must be developed by applying it to different problem types in future studies.