

# Paralel Yapay Arı Koloni Algoritması için Rastgele Kaynaklar ile Yeni Bir Göçmen Üretme Yaklaşımı A New Emigrant Creation Strategy with Randomized Sources for Parallel Artificial Bee Colony Algorithm

Selçuk Aslan<sup>1</sup>, Derviş Karaboğa<sup>2</sup>, Alperen AKSOY<sup>3</sup>

<sup>1</sup>Bilgisayar Mühendisliği Bölümü, Ondokuz Mayıs Üniversitesi, Samsun, Türkiye  
{selcuk.aslan}@omu.edu.tr

<sup>2</sup>Bilgisayar Mühendisliği Bölümü, Erciyes Üniversitesi, Kayseri, Türkiye  
{karaboga}@erciyes.edu.tr

<sup>3</sup>Bilgisayar Mühendisliği Bölümü, Alanya Alaaddin Keykubat Üniversitesi, Antalya, Türkiye  
{alperen.aksoy}@alanya.edu.tr

**Özetçe** —Popülasyon tabanlı algoritmaların mevcut sistemlerin hesaplama gücünden faydalanabilmek üzere alt popülasyon ya da kolonilere ayrılıp eş zamanlı işletilmesi en sık başvurulan paralelleştirme yaklaşımları arasında yer alır. Ancak bu genel yaklaşım, popülasyonun alt popülasyonlara ayrılıyor olması sebebi ile çözüm çeşitliliğini azaltmakta ve çözüm çeşitliliğini iterasyonların sonuna kadar koruyabilmek adına alt popülasyonlar arasında çözümlerin göç ettirilmesine ihtiyaç duymaktadır. Bu çalışmada, alt popülasyonda göç ettirmek üzere seçilen en iyi çözümün parametrelerinin aynı alt popülasyondaki rastgele belirlenmiş çözüm ya da çözümlerin daha uygun parametreleri ile güncellendiği yeni bir yaklaşım önerilmiş ve bu yaklaşımın paralel Yapay Arı Koloni (Artificial Bee Colony, ABC) algoritmasının performansı üzerindeki etkileri incelenmiştir. Uygulama sonuçları, rastgele çözüm destekli yeni göçmen üretme stratejisinin paralel Yapay Arı Koloni algoritmasının yakınsama performansı ve çözüm kalitesini, seri ABC algoritması ve doğrudan en iyi çözümün göçmen olarak seçildiği ring komşuluk topolojili paralel ABC algoritmasına göre önemli oranlarda iyileştirildiğini göstermiştir.

**Anahtar Kelimeler**—Yapay Arı Koloni, paralelleştirme.

**Abstract**—Dividing the whole population into subpopulations or subcolonies then evaluating them simultaneously is one of the most commonly used parallelization approaches to utilize the computational power of the current systems. However, this type of parallelization strategy decreases the population diversity because of the division of the entire population and needs migrations between subpopulations to maintain the solution diversity until the end of the iterations. In this study, we proposed a new emigrant creation strategy in which the parameters of the best food source being migrated to the neighbor subpopulation is modified with the more appropriate parameters of the randomly determined solution or solutions and investigated its effect on the performance of the parallel Artificial Bee Colony (ABC) algorithm. Experimental studies showed that newly proposed emigrant creation strategy based on randomized solutions significantly improved the convergence performance and solution qualities of parallel ABC algorithm compared to the its standard serial and ring neighborhood topology based parallel implementation for which the best solutions are directly used as emigrants.

**Keywords**—Artificial Bee Colony, parallelization.

## I. GİRİŞ

Geçtiğimiz yıllarda sürü zekası temelli algoritmalar hem kompleks mühendislik problemleri hem de gerçek hayat problemlerinin çözümünde başarıyla kullanılmışlardır. Az sayıda kontrol parametresi, sağlam ve esnek yapısı dikkate alındığında ABC algoritmasının sürü zekası temelli algoritmalar arasında önemli bir yeri olduğu kolaylıkla görülür [1]–[5]. ABC algoritmasının performansını artırmaya yönelik çalışmaların üç temel gruba ayrılarak incelenmesi mümkündür. Birinci gruba dahil edilebilir çalışmalar ABC algoritmasının farkı sezgisel ya da deterministik yaklaşımlar ile hibrit modellerinin geliştirilmesi senaryoları üzerine yoğunlaşmıştır. Marinakis ve diğerleri ABC algoritmasının kümeleme probleminin çözümündeki etkinliğini artırılması adına aç gözlü rassal adaptif adını verdikleri bir yaklaşımla algoritmayı birlikte kullanmışlardır [6]. Zhao ve diğerleri ABC algoritmasını Genetik algoritma (Genetic Algorithm, GA) ile birlikte kullanarak yeni bir model önermişlerdir [7]. Önerilen modelde kaşif arı fazında ABC algoritmasını GA tarafından elde edilen kalite değeri yüksek çözümler aktarılrken, GA ve ABC popülasyonlarının çaprazlanması sonucu bulunan kaliteli bireyler ise GA popülasyonuna aktarılır [7]. Xu ve diğerleri ABC algoritmasını Kaos Teorisi ile birleştirerek hava araçlarının yol ve rota plan probleminin çözümünde algoritmanın standart implementasyonuna göre daha başarılı sonuçlar üretebilmişlerdir [8]. Kıran ve Gündüz, Parçacık Sürü Optimizasyon algoritması (Particle Swarm Optimization, PSO) ile ABC algoritmasını birlikte kullanarak yeni bir hibrit model önermişlerdir [9]. Bu model, algoritmaların iterasyon ve çevrimler sonucu bulunan en iyi çözümlerinden rekombinant çözümlerin üretilmesi fikrine dayanmaktadır [9].

İkinci grup çalışmalar ABC algoritmasının başlangıç besin kaynaklarının oluşturulması, aday çözüm üretilirken bilgisinden faydalanılacak çözüm ve değiştirilecek parametre ya da parametrelerin belirlenmesi gibi rassal yönetilen süreçlerin değiştirilmesi üzerine yoğunlaşmaktadır. Tsai ve diğerleri temel çekim yasalarından faydalanarak ABC algoritmasının görevli ve gözcü arı fazlarında kaynakların kullanım-tüketim karakteristiğini değiştirmişlerdir [10]. Pan ve diğerleri başlangıç popülasyonu ya da kaynakları için çözüm çeşitliliği ve kalite

tenin belirli bir seviyenin üzerinde olmasını garanti edebilmek adına çözümlerin bir kısmının rastgele, kalan kısmının ise belirli kurallar dahilinde üretildiği mekanizmayı sunmuşlardır [11]. Coelho ve Alotto Gauss dağılımını ABC algoritmasının görevli ve gözcü arı fazlarındaki aday çözüm üretme sürecine dahil ederek Gaussian ABC algoritmasını literatüre kazandırmışlardır [12]. Li ve diğerleri eylemsizlik ağırlıkları ve ivmelenme katsayıları gibi PSO algoritmasına özgü kontrol parametrelerinden faydalanarak aday çözüm üretme aşamasında değişiklikler barındıran yeni bir ABC algoritması sunmuşlardır [13].

ABC algoritması diğer popülasyon temelli algoritmalarda olduğu gibi paylaşılan ya da dağıtık hafızalı sistemlerde paralel modellenmeye uygundur. Bu bağlamda son gruba dahil edilecek çalışmaların ABC algoritmasının paralelleştirilmesine uygun modeller üzerine yoğunlaştığı söylenebilir. Narasimhan bütün bir koloniyi alt kolonilere ayırarak modellediği paralel ABC algoritması ile nümerik optimizasyon problemlerini çözmüş [14], Banharsakun ve diğerleri alt koloniler arasında en iyi çözümlerin gönderildikleri alt koloninin en düşük kalite değerine sahip çözümü ile yer değiştirdiği bir göç şeması kullanmışlardır [15]. Baştürk ve Akay önermiş oldukları asenkron iş modeli ile desteklenmiş paralel ABC algoritmasını, algoritmanın orijinal modelinin kullanıldığı paralel uygulaması ile karşılaştırmışlardır [16]. Baştürk ve Akay bir diğer çalışmalarında ise en iyi çözümlerin göçmen olarak belirlendiği paralel ABC algoritmasının performansını farklı göç şema ve periyodları üzerinden önce nümerik problemlerin çözümünde, sonrasında ise sinir ağlarının eğitiminde sınamışlardır [17]. Karaboğa ve diğerleri ise koloninin izlek ya da düğüm sayısına göre ayrıldığı paralel ABC algoritmasının performansının göçmen olarak seçilen çözüme bağlı nasıl değiştiğini incelemiş, göçmen çözüm üretme yaklaşımları ile paralel ABC algoritmasının yakınsama performansı ve çözüm kalitesini konvensiyonel paralel modellere göre önemli ölçüde iyileştirilebildiğini göstermişlerdir [18], [19].

Bu çalışmada paralel ABC algoritması için alt koloniler arasında en iyi çözümün göç ettirilmesi bağlı dezavantajlı durumları giderebilmek adına yeni bir göçmen üretme stratejisi tanımlanmıştır. En iyi çözümün göç öncesi parametrelerinin, rastgele belirlenmiş komşularının var ise daha iyi parametrelerinin kullanılarak güncellendiği komşuluk temelli model farklı türdeki nümerik optimizasyon problemlerinin çözümünde paralel ABC algoritması ile birlikte kullanılmıştır. Uygulama sonuçları önerilen yeni göçmen üretme yaklaşımı ile paralel ABC algoritmasının çözüm kalitesi ve yakınsama performansı dikkate alındığında algoritmanın seri ve en iyi çözümlerin doğrudan göçmen olarak kullanıldığı paralel modeline kıyasla önemli oranlarda iyileştirildiğini göstermiştir. Çalışmanın ikinci bölümünde ABC algoritmasının görevli, gözcü kaşif arı fazları için kullanılmış olduğu matematiksel benzetimlerden söz edilmiş, üçüncü bölümde rasgele komşuluk temelli yeni göçmen üretme stratejisi detayla olarak anlatılmış, dördüncü bölümde uygulama sonuçlarına yer verilmiş, son bölümde ise öneri ve yapılması muhtemel çalışmalardan bahsedilmiştir.

## II. YAPAY ARI KOLONİ ALGORİTMASI

Arı kolonilerinde iş-bölümü yapabilme ve kendi kendine organizasyon mekanizmalarının etkisi ile görevli, gözcü ve

kaşif arılar ve besin kaynaklarının bulunması-tüketilmesi konularında zeki olarak değerlendirebilecek davranış modelleri ortaya çıkmıştır. Arıların besin kaynağı bulma-tüketme davranışlarında gözlemlenen bu zekiliği modelleyen Karaboğa 2005 yılında ABC algoritmasını literatüre kazandırmıştır. ABC algoritması kullanılarak bir optimizasyon problemi çözülürken besin kaynakları problemin olası çözümleri ile, besin kaynaklarının nektar miktarları ise çözümlerin uygunluk ya da kalite değerleri ile eşleştirilirler [20], [21].

ABC algoritması optimizasyon işlemine her biri olası çözümlere denk gelen  $SN$  farklı kaynağı rastgele oluşturarak başlamaktadır.  $x_{ij}$  değeri,  $x_i$  çözümünün  $j$ 'nci parametresini,  $x_j^{min}$  ve  $x_j^{max}$   $j$ 'nci parametrenin alt ve üst sınırlarını gösteriyor olmak üzere,  $x_i$  besin kaynağının  $j$ 'nci parametresi (1)'de verildiği üzere hesaplanmaktadır [1]–[5].

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad j = 1, 2, \dots, D \quad (1)$$

Başlangıç besin kaynakları oluşturulduktan sonra her kaynak bir görevli arı ile ilişkilendirilip görevli arı fazı başlatılır. Görevli arılar ilişkilendirildikleri kaynağın komşuluğunda yeni kaynaklar bulma, bulunan kaynak daha iyi ise bu kaynağı hafızaya alıp görevsiz arıları daha iyi kaynağa yönlendirmekle sorumludurlar. Görevli arıların hafızalarındaki kaynağın komşuluğunda aday çözüm üretmek üzere ABC algoritması tarafından önerilen matematiksel benzetimi (2)'de ile verildiği üzere tanımlanmıştır [1]–[5].

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

$v_{ij}$ ,  $j$ 'nci parametresi dışında diğer parametreleri  $x_i$  çözümü ile aynı olan aday çözümü gösteriyor olmakla birlikte,  $v_i$  aday çözümünün  $j$ 'nci parametresi  $x_i$  çözümü ve  $x_i$  çözümünden farklı  $x_k$  çözümünün  $j$ 'nci parametreleri arasındaki farkın (2)'de  $\phi_{ij}$  olarak tanımlanmış  $-1$  ile  $1$  aralığında rastgele belirlenmiş bir sayının çarpımının  $x_i$  çözümünün  $j$ 'nci parametresine eklenmesi sonucu bulunmaktadır [1]–[5]. Eğer  $v_i$  kaynağının amaç fonksiyon değeri  $obj(v_i)$  üzerinden (3) ile bir minimizasyon problemi için hesaplanacak uygunluk değeri  $fit(v_i)$ ,  $x_i$  çözümünün  $fit(x_i)$  ile gösterilen uygunluk değerinden daha iyi ise  $x_i$  ile  $v_i$  çözümü ile yer değiştirir, aksi durumda  $x_i$  çözümünün deneme sayaç değeri bir artırılır [1]–[5].

$$fit(x_i) = \begin{cases} 1 + |obj(x_i)|; & \text{if } obj(x_i) > 0 \\ 1/(1 + |obj(x_i)|); & \text{if } obj(x_i) \leq 0 \end{cases} \quad (3)$$

Görevli arıların tamamı kovana döndüklerinde, buldukları besin kaynakları ile ilgili bilgiyi gözcü arılara aktarırlar. Gözcü arılar tüketmek üzere seçecekleri kaynakların büyük olasılıkla, kalite-uygunluk değeri yüksek kaynaklar arasında olacağı bilinmektedir [1]–[5]. ABC algoritmasının kaliteli kaynaklara daha çok gözcü arının yönlendirilmesini sağlayacak seçim mekanizması (4)'te her kaynak için hesaplanan seçim olasılık değeri üzerinden yürütülür. İlgilenilen kaynağın uygunluk değerinin tüm kaynakların uygunluk değeri toplamına oranı şeklinde tanımlanmış seçim-seçilim olasılıkları, kaynağın uygunluk değeri arttıkça kaynağı seçecek gözcü arı sayısının artmasına olanak verecek şekilde tanımlanmıştır [1]–[5].

$$p(x_i) = \frac{fit(x_i)}{\sum_j^{SN} fit(x_j)} \quad (4)$$

ABC algoritmasında seçim olasılık değerleri belirlenen kaynaklara, kaynak sayısına gözcü arının gönderilmesi süreci algoritmanın gözcü arı fazında yönetilmektedir. Göz arının seçmiş olduğu kaynağa görevli arı olarak atanması kaynağın seçim olasılığının 0 ile 1 arasında rastgele belirlenmiş bir sayı ile karşılaştırılması sonucu karara bağlanır. Gözcü arı bir kaynağı seçtikten sonra seçilen kaynağın komşuluğunda tıpkı görevli arı gibi bir aday çözüm üretir.

ABC algoritmasının görevli ve gözcü arı fazları incelendiğinde, bu fazların kaynakların kullanılması ya da tüketilmesi yönünde çalışma karakteristikleri olduğu görülür. Ancak iyi bir arama sürecinde kaynakların kullanma-tüketme işlemlerinin yeni kaynakların bulunması-keşfedilmesi işlemi ile dengeli götürülmesi gerekir. ABC algoritmasında bu dengenin kaşif arı fazında kurulduğu söylenebilir. Eğer bir kaynak belirli iterasyon-çevrimler boyunca iyileştirilememiş ise bu kaynakla ilişkilendirilen görevli arı kaşif arı haline gelir ve henüz keşfedilmemiş bir kaynağa yönlendirilir. Görevli arıların kaşif arı olup olmayacağına dair karar ABC algoritmasında *limit* adı verilen kontrol parametresi üzerinden verilir. Kaşif arı fazında deneme sayacı *limit* değerini en çok aşan kaynak varsa bırakılır ve bu kaynak yerine yeni bir kaynak üretilip kaynak kümesine dahil edilir. ABC algoritmasının nümerik problemlerin çözümünde kullanıldığı durumlarda *limit* parametresinin değeri genellikle  $\lceil a \times SN \times D \rceil$  ve  $a \in \mathbb{Q}^+$  olarak belirlenmektedir. ABC algoritmasının görevli, gözcü ve kaşif arı fazlarının daha önce tanımlanmış durdurma kriteri sağlanıncaya kadar tekrar edildiğini gösteren temel işlem adımları aşağıda verilmiştir.

#### Algoritma 1 ABC algoritmasının temel işlem adımları

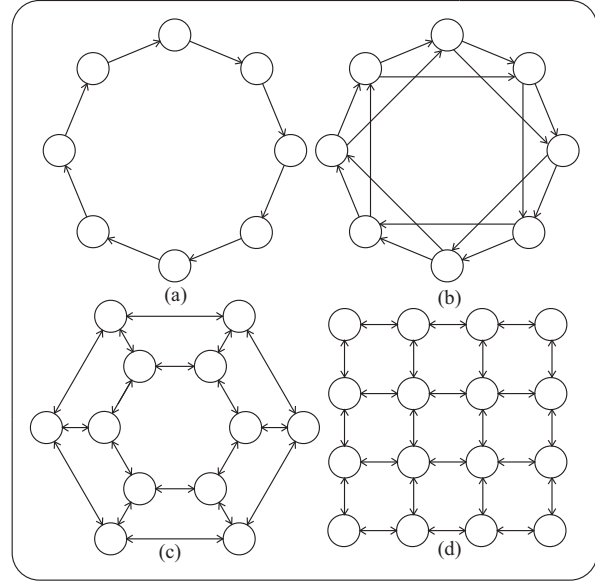
- 1: **Başlangıç:**
- 2: *limit* ve Maksimum Çevrim Sayısı (*MCN*) değerleri belirlenir.
- 3: *SN* farklı başlangıç çözümünü Denklem 1'den faydalanarak üretilir.
- 4: Bulunan en iyi kaynak  $x_b$  vektöründe tutulur.
- 5: *çevrimSayacı*  $\leftarrow 0$ .
- 6: **Tekrar et**
- 7: **Görevli Arı Fazı:**
- 8: Görevli arıların tamamı yeni kaynaklara gönderilir.
- 9: Kaynaklar arasında aç gözlü seçim yapılır.
- 10: **Gözcü Arı Fazı:**
- 11: Kaynakların seçim olasılık değerleri hesaplanır.
- 12: Kaynak sayısına gözcü arı yeni kaynaklara gönderilir
- 13: Kaynaklar arasında aç gözlü seçim yapılır.
- 14:  $x_b$  çözümünden daha iyi çözüm var ise bunu  $x_b$  ile değiştir.
- 15: **Kaşif Arı Fazı:**
- 16: *limit* değerini varsa en çok aşan kaynak bulunur.
- 17: Seçilen kaynak için bir kaşif arı gönderilir.
- 18: *çevrimSayacı*  $\leftarrow$  *çevrimSayacı* + 1
- 19: *çevrimSayacı* maksimum çevrim sayısına (*MCN*) ulaşınca kadar

### III. RASTGELE SEÇİLEN KAYNAKLAR İLE KOMŞU ÜRETME YAKLAŞIMI

Alt koloni ya da popülasyonların eş zamanlı işletildiği paralelleştirme senaryolarında, seri çalışan karşılıklarına kıyasla algoritmaların azalan çözüm çeşitliliği, alt koloni ya da popülasyonlar arasında seçilen çözümlerin belirli sıklıkta göç ettirilmesine bağlı olarak dengelenmeye çalışılmaktadır. Ancak hangi çözüm ya da çözümlerin göç ettirileceği, göçmen

çözüm ya da çözümlerin hangi alt koloniye ya da kolonilere gönderilecekleri, göçmen çözümlerin yeni alt kolonilerde hangi çözümlerle değiştirileceği, göç işleminin ne sıklıkta yapılacağı kararı çözüm çeşitliliği dışında algoritmaların paralel performanslarını doğrudan etkileyen faktörler arasında yer alır.

Göçmen çözümlerin hangi alt koloniler arasında taşınacağı kararı alt koloniler arasında kurulacak temel komşuluk topolojisine bağlı verilmektedir. Kullanılmaları durumunda paralel uygulamanın hızlandırma ve verimlilik değerlerinde değişikliklere sebep olacak temel komşuluk topolojilerinden bazıları Şekil 1'de verilmiştir [18].



Şekil 1: Temel komşuluk topolojilerinden bazıları: (a)Ring, (b)Ring 1+2, (c)Torus, (d)Lattice

Komşuluk topolojilerinin genelinde göçmen çözümün, ilgili alt koloninin göç periyoduna erişildiğinde bulunan en iyi çözüm olarak seçilmesi, gönderildiği alt kolonide ise bu alt koloninin kalite değeri en düşük çözümü ile değiştirilmesi genel kabul görmüş bir yöntem olmakla birlikte bazı dezavantajları beraberinde getirir. Eğer göçmen olarak seçilen çözüm ardışık göç periyodları arasında iyileştirilememiş ise aynı çözümün yeni alt kolonide birden fazla defa görülme olasılığı artacak, bir diğer ifade ile ihtiyaç duyulan çözüm çeşitliliği istenilen seviyelere getirilemeyecektir. Paralel uygulamalarda çözümlerin göç ettirilmesine bağlı senkronizasyon ihtiyacı sebebiyle artan işlem yükü ve bu yükün hızlandırma ve verimlilik değerleri dikkate alındığında, göç ettirilecek çözüme ilişkin verilecek kararın en az komşuluk topolojisinin belirlenmesi kadar önemli olduğu kolaylıkla görülür.

Önerilen göçmen çözüm üretme yaklaşımında göç periyoduna ulaşıldığında alt kolonilerdeki en iyi çözümler, bulundukları alt kolonilerdeki rastgele çözümlerin daha kaliteli parametreleri ile güncellenirler. Alt kolonideki en iyi çözümün her bir parametresi en iyi çözümün farklı olmak şartı ile rastgele seçilen bir çözüm ya da komşunun ilgili parametresi kullanılarak daha da iyileştirilmeye çalışılır. Eğer rastgele seçilen çözümün ilgili parametresi en iyi çözümün kalite değerini artırıyor ise göç öncesi en iyi çözümün bu parametresi rastgele seçilen çözümün parametresi ile güncellenir. En iyi çözümün

göç öncesi güncellenmesi işlemine en iyi çözümün bütün parametreleri sırasıyla kontrol edilene kadar tekrar edilir. Aşağıda rastgele komşu temelli (randomized neighbor based, rnb) göçmen çözüm üretme stratejisinin nasıl işletildiğini gösteren kaba kod bulunmaktadır.

#### Algoritma 2 Rastgele komşu temelli göçmen üretme

```

1:  $x_B \leftarrow$  Alt kolonideki kalite değeri en yüksek çözüm.
2:  $x_E \leftarrow x_B$  çözümün bir kopyası  $x_E$  vektörüne kopyalanır.
3: for  $j \leftarrow 1 \dots D$  do
4:    $x_R \leftarrow x_E$  çözümünden farklı rastgele seçilmiş komşu.
5:    $x_{E,j}$  parametresi ile  $x_{R,j}$  parametresini değiştir.
6:   if  $fit(x_E) \leq fit(x_B)$  then
7:      $x_{E,j}$  parametresini  $x_{B,j}$  parametresi ile değiştir.
8:   end if
9: end for
10:  $x_E$  çözümünü komşu alt koloniye gönder.

```

Algoritma 2'den de anlaşılacağı üzere göçmen çözüm olarak belirlenen alt koloninin en iyi çözümü, komşusuna gönderilmeden önce rastgele belirlenen çözüm ya da çözümlerin daha iyi parametreleri ile güncellenmekte, bu sayede ardışık göç periyodları arasında en iyi çözüm aynı kalmış olsa dahi bahsedilen güncelleme mekanizması sayesinde değiştirilerek aktarılma olasılığı artmaktadır. Göçmen çözümlerin gönderilmesi süresince izlek ya da hesaplama düğümlerinin senkronizasyonu hızlandırma ve verimlilik değerlerini önemli miktarda etkiliyor olması dikkate alındığında aynı çözümün birden fazla defa göç ettirilmesi yerine rastgele çözümler ile güncellenerek gönderilmesinin daha etkili bir yöntem olabileceği göz ardı edilmemelidir.

#### IV. UYGULAMA

ABC algoritmasının standart seri varyantı sABC, alt kolonilerdeki en iyi çözümlerin gönderildiği alt kolonilerdeki en kötü çözümler ile değiştirildiği paralel ABC algoritması pABC ve önerilen yaklaşıma bağlı üretilen göçmenin gönderildiği alt kolonilerdeki en kötü çözümler ile değiştirildiği rastgele komşu temelli paralel ABC algoritması rnb-pABC, detayları Tablo I'de verilen  $f_1$ ,  $f_2$  ve  $f_3$  fonksiyonları üzerinden karşılaştırılmıştır.  $f_1$ , Sphere fonksiyonu tek modlu olup parametrelerinin alt ve üst sınırları sırasıyla -100 ile 100 şeklinde belirlenmiştir.  $f_2$ , Griewank fonksiyonu birden fazla yerel minimum noktanın dağıldığı çok modlu bir araştırma uzayına sahiptir. Griewank fonksiyonu için parametrelerin alt ve üst sınırları -600 ile 600 olarak belirlenmiştir.  $f_3$ , Rosenbrock fonksiyonu ise global çözüme yakınsama zorluğu sebebi ile algoritmaların performanslarının karşılaştırılmasında sıklıkla kullanılmaktadır. Rosenbrock fonksiyonu için parametrelerin alt ve üst sınırları ise sırasıyla -30 ve 30 olarak alınmıştır.  $f_1$ ,  $f_2$  ve  $f_3$  fonksiyonlarının tamamı için global minimum değer ise 0'a eşittir.

Seri ABC algoritması, ring komşuluk topolojisinin kullanıldığı pABC ve rnb-pABC algoritmaları için besin kaynaklarının sayısı 100 olarak seçilmiştir. Her bir fonksiyon için parametre sayısı 500 olup, *limit* değeri parametre sayısına eşit alınmıştır. ABC algoritmasının paralel implementasyonları için görev dağıtımı *pthread*s kütüphanesi kullanılarak gerçekleştirilmiş, göçmen çözüm üretilme ve dağıtılması 20 çevrim sıklıkla yapılmıştır. Seri ABC algoritması, dört alt koloniye sahip pABC ve rnb-pABC algoritmaları her bir fonksiyon için 20 bağımsız kez 2000 çevrim Intel® i5 750 işlemcili 4 GB(gigabyte)

DDR3 (double data rate) ana hafızaya sahip Fedora 19 isimli Linux dağıtımı yüklü bilgisayar üzerinde çalıştırılmış, 20 farklı koşmanın her birinde bulunan en iyi çözümlerin ortalaması ve standart sapması alınmış, ilgili sonuçlar Tablo II, Tablo III ve Tablo IV'de verilmiştir. Tablo II, Tablo III ve Tablo IV'de verilen sonuçlar incelendiğinde rnb-pABC algoritmasının  $f_1$ ,  $f_2$  ve  $f_3$  fonksiyonlarının tamamında sABC ve pABC algoritmalarına kıyasla daha başarılı sonuçlar üretebilmeyi başardığı görülmüştür.

Tablo I: Karşılaştırmada kullanılan fonksiyonlar

Fonksiyon	Formülasyon
$f_1$	$f_1(\vec{x}) = \sum_{i=1}^D (x_i^2)$
$f_2$	$f_2(\vec{x}) = \frac{1}{4000} \left( \sum_{i=1}^D x_i^2 \right) - \left( \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) + 1$
$f_3$	$f_3(\vec{x}) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$

Tablo II: sABC ve pABC algoritmalarının karşılaştırılması

Fonksiyon	sABC		pABC	
	Ort.	Std.Sapma	Ort.	Std.Sapma
$f_1$	6.8935e+04	1.2525e+04	1.0776e+05	1.1405e+04
$f_2$	5.7976e+02	1.3321e+02	9.2546e+02	7.0561e+01
$f_3$	1.9255e+07	2.1719e+07	2.1203e+08	6.6421e+07

Tablo III: sABC ve rnb-pABC algoritmalarının karşılaştırılması

Fonksiyon	sABC		rnb-pABC	
	Ort.	Std.Sapma	Ort.	Std.Sap
$f_1$	6.8935e+04	1.2525e+04	5.4131e-01	4.0853e-01
$f_2$	5.7976e+02	1.3321e+02	1.0995e-01	7.4480e-02
$f_3$	1.9255e+07	2.1719e+07	1.6784e+03	7.7563e+02

Tablo IV: pABC ve rnb-pABC algoritmalarının karşılaştırılması

Fonksiyon	pABC		rnb-pABC	
	Ort.	Std.Sapma	Ort.	Std.Sap
$f_1$	1.0776e+05	1.1405e+04	5.4131e-01	4.0853e-01
$f_2$	9.2546e+02	7.0561e+01	1.0995e-01	7.4480e-02
$f_3$	2.1203e+08	6.6421e+07	1.6784e+03	7.7563e+02

ABC algoritmasının seri ve paralel varyantları arasındaki diğer karşılaştırma her bir fonksiyon için 20 farklı koşma üzerinden hesaplanan saniye türünden ortalama yürütme zamanları, ortalama yürütme zamanları ve izlek sayılarından faydalanılarak bulunan hızlandırma ve verimlilik değerleri üzerinden yapılmıştır. Hızlandırma metriği, algoritmanın seri eşdeğeri ile paralel modelinin yürütme zamanlarının oranı olarak hesaplanırken alabileceği en büyük değer paralel modelin çalıştırıldığı dağıtık sistemin düğüm sayısı ya da paylaşılan hafızalı sistemde kullanılan izlek sayısına eşit olabilir. Verimlilik metriği ise hızlandırma metriğinin düğüm sayısı ya da izlek sayısına oranı olarak hesaplanmaktadır ve alabileceği en büyük değer 1'e eşit olabilir.

Önerilen göçmen çözüm üretme stratejisinin toplam çevrim sayısının %1'ine denk gelecek göç sıklığında rnb-pABC algoritmasının ortalama yürütme zamanını pABC algoritmasına kıyasla  $f_1$  fonksiyonu için %18.3,  $f_2$  fonksiyonu için %9.8

Tablo V: pABC algoritması için hızlandırma ve verimlilik değerleri

Fonksiyon	sABC	pABC	Hızlandırma	Verimlilik
$f_1$	1.1672	0.6591	1.7708	0.4427
$f_2$	24.8083	7.2765	3.4093	0.8523
$f_3$	2.4829	0.8827	2.8126	0.7032

Tablo VI: rnb-pABC algoritması için hızlandırma ve verimlilik değerleri

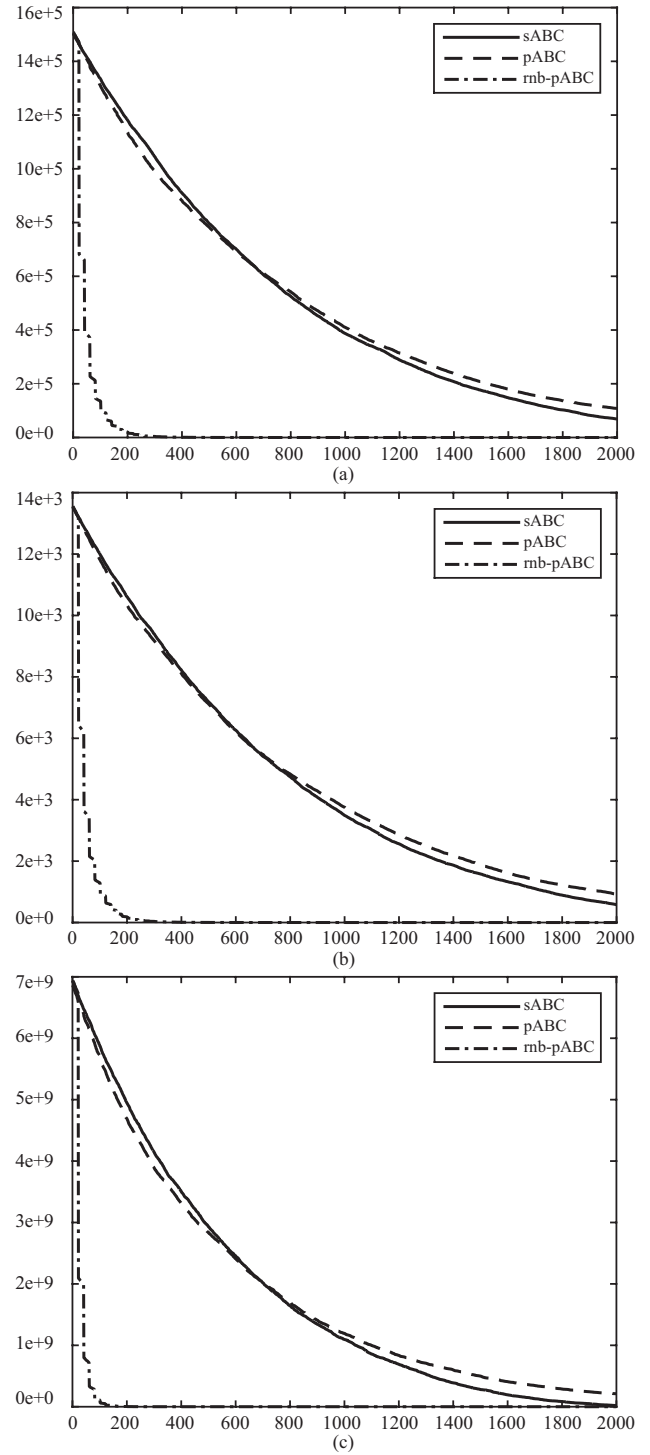
Fonksiyon	sABC	rnbABC	Hızlandırma	Verimlilik
$f_1$	1.1672	0.7801	1.4961	0.3740
$f_2$	24.8083	7.9912	3.1044	0.7761
$f_3$	2.4829	1.1411	2.1759	0.5440

ve  $f_3$  fonksiyonu için %29.2 artmıştır.  $f_1$  fonksiyonu için hesaplama maliyetinin algoritmanın paralelleştirilmesine bağlı yüke kıyasla düşük kalması sebebi ile hızlandırma ve verimlilik değerleri hem pABC hem de rnb-pABC algoritmalarında %50'nin altında kalmıştır.  $f_2$  ve  $f_3$  fonksiyonları için ise hesaplama maliyeti  $f_1$  fonksiyonuna kıyasla daha büyük olduğundan ABC algoritmalarının paralel uygulamaları %50'nin üzerinde verimlilik değerleri ile çalıştırılabilmiştir. rnb-pABC algoritmasında göçmen çözüm üretme süreci parametrelerin sıralı kontrolünü gerektirdiğinden hızlandırma ve verimlilik değerleri pABC algoritmasının gerisinde kalmıştır. Ancak rnb-pABC algoritmasının hem ortalama en iyi amaç fonksiyon değerleri hem de yakınsama performansları dikkate alındığında yürütme zamanındaki artışın kabul edilebilir miktarlarda olduğu söylenebilir. Göç periyodunun ise 20'den daha büyük değerlere çekilmesi halinde de yakınsama performansından ödün verilerek rnb-pABC algoritmasının pABC algoritmasının hızlandırma ve verimlilik değerlerine özellikle hesaplama maliyet yüksek problemlerde yaklaşabileceği anlaşılmaktadır.

Önerilen göçmen üretim stratejisinin paralel ABC algoritmasının yakınsama performansı üzerindeki etkisini görebilmek adına Şekil 2'de verilen grafikler incelenebilir. Üretilen göçmen çözümlerin dağıtılması ilk dağıtımı ile birlikte rnb-pABC algoritması sABC ve pABC algoritmalarına kıyasla çok daha hızlı yakınsamış, bu hızlı yakınsamanın etkisi ilerleyen çevrimlerde de azalıyor olmasına karşın devam etmiştir. Yakınsama grafiklerinden çıkarılabilecek bir diğer sonuç, önerilen göçmen çözüm üretim stratejisinin yürütme zamanı üzerindeki etkisinin de dikkate alarak çevrimler tamamlanıncaya kadar değil, belirli sayıda uygulanabileceği yönündedir. Önerilen göçmen üretim stratejisinin  $f_1$ ,  $f_2$  ve  $f_3$  fonksiyonlarının tamamında etkisini yansıtmış olması, farklı problem tiplerinde de yönetimin yakınsama performansına etki edebileceğini göstermesi bakımından ayrıca önemlidir.

## V. TARTIŞMA, SONUÇ VE ÖNERİLER

Bu çalışmada paralelleştirilebilir popülasyon temelli sezgisel algoritmalarda kullanılabilir yeni bir göçmen çözüm üretim yaklaşımı tanıtılmış, alt koloni-popülasyonlardaki en kaliteli çözümün parametrelerinin rastgele belirlenmiş çözümlerin ilgili parametreleri ile güncellendiği yaklaşım ABC algoritmasının ring komşuluk topolojili paralel varyantında kullanılmıştır. Uygulama sonuçları, rastgele seçilen çözümlere bağlı göçmen üretim stratejisinin farklı türlerdeki nümerik problemlerde ABC algoritmasının seri ve en iyi çözümün doğrudan göçmen seçildiği paralel varyantlarına kıyasla çözüm

Şekil 2:  $f_1$  (a),  $f_2$  (b) ve  $f_3$  (c) fonksiyonları için yakınsama grafikleri

kalitesi ve yakınsama performansını yürütme zamanında kabul edilebilir miktarlardaki değişiklikler ile önemli oranlarda artırdığı görülmüştür. Çözüm kalitesi ve yakınsama performansı üzerinde önerilen modelin etkisi farklı komşuluk topolojileri ve göç periyodlarında ABC algoritmasının paralel modelinin hesaplama maliyeti yüksek nümerik veya ayrık optimizasyon problemlerinde de kullanılabileceği hakkında bilgi vermektedir.

## KAYNAKLAR

- [1] B. Kumar, D. Kumar, "A Review on Artificial Bee Colony Algorithm", *International Journal of Engineering & Technologies*, 2(3), 2013.
- [2] J.C. Bansal, H. Sharma, S.S. Jadon, "Artificial Bee Colony Algorithm: A Survey", *International Journal of Advanced Intelligence Paradigm*, 5(1-2), 123-159, 2013.
- [3] A.L. Bolaji, A.T. Khader, M.A. Al-Betar, M.A. Awadallah, "Artificial Bee Colony Algorithm, its Variants and Applications: A Survey", *Journal of Theoretical & Applied Information Technology*, 47(2), 2013.
- [4] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, "A Comprehensive Survey: Artificial Bee Colony Algorithm and Applications", *Artificial Intelligence Review*, 42(1), 21-57, 2014.
- [5] B. Akay, D. Karaboga, "A Survey on the Applications of Artificial Bee Colony in Signal, Image and Video Processing", *Signal, Image and Video Processing*, 9(4), 967-990, 2015.
- [6] Y. Marinakis, M. Marinaki, N. Matsatsinis, "A hybrid discrete artificial bee colony-GRASP algorithm for clustering", *International Conference on Computers & Industrial Engineering*, Troyes/Fransa, 2009.
- [7] H. Zhao, Z. Pei, J. Jiang, R. Guan, C. Wang, X. Shi, "A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony", *Advances in Swarm Intelligence*, 558-565, 2010.
- [8] C. Xu, H. Duan, F. Liu, "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning", *Aerospace Science and Technology*, 14(8), 535-541, 2010.
- [9] M.S. Kiran, M. Gündüz, "A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems", *Applied Soft Computing*, 13(4), 2188-2203, 2013.
- [10] P. Tsai, J. S. Pan, B.Y. Liao, S.C. Chu, "Enhanced Artificial Bee Colony Optimization", *International Journal of Innovative Computing, Information and Control*, 5(12), 5081-5092, 2009.
- [11] Q.K. Pan, M.F. Tasgetiren, P.N. Suganthan, P.N. Chua, "A Discrete Artificial Bee Colony Algorithm for The Lot-streaming Flow Shop Scheduling Problem", *Information Sciences*, 181(12), 2455-2468, 2011.
- [12] S. Coelho, L. Alotto, "Gaussian Artificial Bee Colony Algorithm Approach Applied to Loney's Solenoid Benchmark Problem", *IEEE Transactions on Magnetics*, 47(5), 1326-1329, 2011.
- [13] J. Li, S. Xie, Q. Pan, S. Wang, "A Hybrid Artificial Bee Colony Algorithm for Flexible Job Scheduling Problems", *International Journal of Computers Communications & Control*, 6(2), 286-296, 2011.
- [14] H. Narasimhan, "Parallel Artificial Bee Colony (PABC) Algorithm", *World Congress on Nature & Biologically Inspired Computing*, Coimbatore/Hindistan, 2009.
- [15] A. Banharsakun, T. Achalakul, B. Sirinaovakul, "Artificial Bee Colony Algorithm on Distributed Environments", *Second World Congress on Nature & Biologically Inspired Computing*, Kitakyushu/Japonya, 2010.
- [16] A. Baştürk, R. Akay, "Parallel Implementation of Synchronous Type Artificial Bee Colony Algorithm for Global Optimization", *Journal of Optimization Theory and Applications*, 155(3), 1095-1104, 2012.
- [17] A. Baştürk, R. Akay, "Performance Analysis of The Coarse-Grained Parallel Model of The Artificial Bee Colony Algorithm", *Information Sciences*, 253, 34-55, 2013.
- [18] D. Karaboğa, S. Aslan, "Best Supported Emigrant Creation for Parallel Implementation of Artificial Bee Colony Algorithm", *IU-Journal of Electrical & Electronics Engineering*, 16(2), 2055-2064, 2016.
- [19] S. Aslan, H. Badem, D. Karaboğa, A. Baştürk, "A New Synchronous Parallel Artificial Bee Colony Algorithm", *1st International Conference on Engineering Technology and Applied Sciences*, Afyonkarahisar/Türkiye 2016.
- [20] T. Batbat, C. Ozturk, "Protein Structure Prediction with Discrete Artificial Bee Colony Algorithm", *International Journal of Informatics Technologies*, 9(3), 2016.
- [21] H. Badem, A. Baştürk, A. Çalışkan, M. Yüksel, "A New Efficient Training Strategy for Deep Neural Networks by Hybridization of Artificial Bee Colony and Limited-Memory BFGS Optimization Algorithms", *Neurocomputing*, 2017.