Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 22, Özel Sayı, 263-269, 2018

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 22, Special Issue, 263-269, 2018

DOI: 10.19113/sdufbed.39561

Silah Hedef Atama Problemi için Tavlama Benzetimli Bir Hibrit Yapay Arı Kolonisi Algoritması

Hakan KUTUCU[*1], **Rafet DURGUT**[1]

[1]Karabük Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 78100, Karabük

(Alınış / Received: 12.04.2018, Kabul / Accepted: 08.06.2018, Online Yayınlanma / Published Online: 17.07.2018)

Anahtar Kelimeler
Hedef silah atama,
Yapay arı kolonisi,
Tavlama benzetimi,
Meta-sezgisel algoritmalar

Özet: Bu çalışmada, sadece savunma alanında uygulamaları olmayıp iş dünyasında da uygulamaları olan çok zor kombinatoriyel optimizasyon problemlerinden statik silah hedef atama problemini ele alıyoruz. Silah hedef atama probleminin amacı, hedeflerin minimum toplam hayatta kalma değeri ile silahların hedeflere atanmasını bulmaktır. Silah hedef atama probleminin NP-tam problemi olduğu bilinmektedir. Bu makalede, silah hedef atama problemine etkili bir çözüm sağlamak için tavlama benzetimi algoritması kullanarak hibrit bir yapay arı kolonisi algoritması önermekteyiz. Önerilen algoritmayı problem örnekleri ile test ettik ve literatürdeki diğer meta-sezgisel yöntemler ile karşılaştırdık. Hesaplamalı testler, algoritmamızın rekabetçi ve tatmin edici olduğunu göstermektedir.

A Hybrid Artificial Bee Colony Algorithm with Simulated Annealing for The Weapon Target Assignment Problem

Keywords
Weapon target assignment,
Artificial bee colony,
Simulated annealing,
Meta-heuristic algorithms

Abstract: In this work, we consider the static weapon target assignment problem which is a very hard combinatorial optimization problem has not only defense-related applications but business world applications. The aim of the weapon target assignment problem is to find an assignment of weapons to targets with the minimum total survival value of the targets. The weapon target assignment problem is known to be NP-complete problem. In this paper, we propose a hybrid artificial bee colony algorithm with simulated annealing algorithm to give an efficient solution to the weapon target assignment problem. We tested the proposed algorithm with benchmark problem instances and compared it with some other meta-heuristics in the literature. Computational tests show that our algorithm is competitive and satisfactory.

1. Giriş

Kaynak tahsisi probleminin özel bir durumu olan Silah Hedef Atama (SHA) problemi, gerçek hayatta uygulamalarını görebildiğimiz ilgi çekici bir kombinatoriyel optimizasyon problemidir. SHA probleminin amacı hedeflerdeki beklenen toplam hayatta kalma değerini minimize etmek veya hedeflerdeki tahribatı maksimize etmek için silahları hedeflere doğru bir şekilde atamaktır [1]. Statik SHA ve dinamik SHA olmak üzere probleminin iki türü vardır. Statik SHA probleminde, problemin tüm girdileri sabittir ve silahlar hedeflere bir kere atanır. Dinamik SHA problemi ise birçok atama aşaması içermektedir. Öyle ki herhangi bir aşamada atanan değerler diğer aşamalarda kullanılmaktadır. Dinamik SHA problemi hakkında daha fazla bilgi için [2] ve [3] kaynaklarına başvurulabilir. Bu makalede SHA probleminin statik olan türü çalışılmıştır ve makale boyunca statik SHA problemi yerine kısaca SHA problemi denilecektir.

SHA probleminin çalışılması 1950'lilere kadar dayanmaktadır. Manne 1958 yılında SHA probleminin modellenmesi üzerine çalışmıştır [4]. Lloyd ve Witsenhausen, 3-EXACT-COVER probleminden indirgeme yaparak SHA probleminin NP-tam olduğunu ispatlamıştır [5]. Birçok araştırmacı SHA probleminin matematiksel modelleri üzerinde çalışmış, doğrusal ve doğrusal olmayan tamsayılı programlama modellerini geliştirmişlerdir. Ma ve Ni, [6] da SHA probleminin tamsayılı doğrusal programlama modelini çalışmıştır. Cetin ve Esen, SHA probleminin medya sektöründe bir uygulamasını vermişlerdir [7]. Yazarların kullandıkları model ise doğrusal olmayan tamsayılı programlama problemidir. Senay, SHA problemini kullanarak havadan-yere füzelerin ihtiyaçlarına yönelik farklı matematiksel modeller oluşturmuştur [8].

SHA problemini küçük boyutlu problem örneklerinde bile kesin olarak çözen algoritmalar mevcut değildir. Bu yüzden bazı araştırmacılar da meta-sezgisel algoritmalar ile optimum çözümlere yaklaşmaya çalışmışlardır. Li

* *İlgili yazar: hakankutucu@karabuk.edu.tr*

ve arkadaşları düşmanın beklenen hasarını maksimize eden ve füzelerin maliyetini minimize eden iki amaçlı bir SHA optimizasyon modeli tasarlamıştır [9]. Bu modeli çözmek için ise değiştirilmiş karınca kolonisi algoritması kulllanmışlardır. Li ve arkadaşları SHA probleminin çözümü için EX çaprazlama operatörünü yerel optimal çocuklara sahip olmak amacıyla açgözlü reformasyon şemasını kullanmışlardır [10]. Lu ve arkadaşları erken yakınsamaları ve yerel optimumları önlemek için genetik algoritmada ilk popülasyonun üretiminde ve diğer genetik operatörlerde değişiklikler yaparak SHA problemini çözmeye çalışmışlardır [11]. Liu ve arkadaşları çok amaçlı SHA problemini çözmek için geliştirilmiş çok amaçlı parçacık sürüsü algoritması önermişler ve bunu 7 platform 10 hedefli bir örneğe uygulamışlardır [12].

Bu makalede diğer bölümlerin organizasyonu şu şekilde düzenlenmiştir: 2. bölümde, SHA probleminin doğrusal olmayan tamsayılı programlama modeli verilmiştir. 3. bölümde önerilen hibrit meta-sezgisel algoritma tanıtılmıştır. 4. bölümde literatürde aynı problem örnekleri ile sayısal sonuçlar karşılaştırılmalı olarak sunulmuştur. Tartışma ve Sonuç kısmında ise yapılan çalışmaların bir özeti yer almaktadır.

2. SHA'nın Matematiksel Modeli

SHA probleminin matematiksel modelinde kullanılan parametreler aşağıda verilmiştir:

n : hedef sayısı,
m : silah tipi,
V_j : j hedefinin değeri ,
W_i : i tipinde mevcut silah sayısı,
p_{ij} : i tipindeki tek bir silah ile j hedefinin imha edilme olasılığı,
q_{ij} : i tipindeki tek bir silah j hedefine atanırsa hedefin hayatta kalma olasılığı, yani q_{ij}=1 − p_{ij}.

Modelde kullanılan karar değişkeni ise şu şekilde tanımlanmıştır:

x_{ij} = j hedefine atanan i tipindeki silahların sayısı, $i = 1,\ldots,n$ ve $j = 1,\ldots,m$.

SHA probleminin çözümünde amaç tüm hedeflerin beklenilen toplam hayatta kalma değerini minimize etmek için mevcuttaki silahlardan kullanılacak x_{ij} adedini bulmaktır ve doğrusal olmayan tamsayılı programlama problemi olarak aşağıdaki gibi formülize edilmiştir.

Amaç fonksiyonu:

$$Min \sum_{j=1}^{n} \left(\prod_{i=1}^{m} q_{ij}^{x_{ij}}\right) \quad (1)$$

Kısıtlar:

$$\sum_{j=1}^{n} x_{ij} \leq W_i, \quad i = 1,\ldots,m \quad (2)$$

$$x_{ij} \geq 0 \text{ ve tamsayı}, \quad i = 1,\ldots,m,\ j = 1,\ldots,n. \quad (3)$$

3. Tavlama Benzetimi ile Hibrit Yapay Arı Kolonisi Algoritması

Bu bölümde önce yapay arı kolonisi ve tavlama benzetimi algoritmalarını tanıyalım.

3.1. Yapay arı kolonisi algoritması

Yapay arı kolonisi (YAK), bal arılarının gerçek yaşamdaki besin bulma davranışlarından esinlenilerek 2005' yılında Karaboğa tarafından geliştirilmiş olan meta-sezgisel bir algoritmadır [13]. Bal arısı kolonisinin besin bulma evresinde sürü zekası kullanımına dayalı optimizasyon yöntemidir. YAK algoritmasında modellenen arıların iki temel davranış modeli bulunmaktadır: Besin kaynaklarına yönelme ve tüketilen kaynakları bırakma. Her bir besin kaynağı optimizasyon problemleri açısından olası bir çözümü ifade etmektedir. Koloni içerisinde bulunan işçi arılar, var olan besin kaynakları üzerinde çalışarak nektar miktarı ve kalitesini arttırmaya çalışan arı çeşididir. Besin kaynağının nektar bilgisini diğer arılar ile paylaşmaktadır. Gözcü arılar, işçi arılar sayesinde elde edilen nektar bilgisine göre var olan besin kaynağı üzerinde çalışabileceği gibi, kaşif arı olarak yeni besin kaynağı aramaya da çıkabilmektedir. Her bir besin kaynağı bir işçi arı ve bir gözcü arı tarafından incelendiği için bu gruplarda bulunan arı sayıları besin kaynağı sayısına eşit olmalıdır [14].

Optimizasyon problemleri açısından değerlendirildiğinde besin kaynakları, problemin çözüm kümesinin geçerli bir elemanıdır. Besine ait olan nektar miktarı ise çözümün uygunluk değeridir. Yöntem içerisinde işçi arı fazında her bir çözümün uygunluk değeri arttırılmaya çalışılmaktadır. Uygunluk değeri belirli bir deneme sonunda değişim göstermiyor ise yeni geçerli çözüm arayışına geçilmektedir. Bu sayede problemin çözümünde yerel en iyi değer bölgesinden kurtulmaya ve yeni çözümleri değerlendirmeye çalışılmaktadır. Algoritmanın sözde kodu aşağıda verilmiştir.

Algoritma 1 Yapay arı kolonisi algoritması

```
Giriş Parametrelerini Belirle.
Başlangıç Popülasyonunu Oluştur.
while Durdurma Kriteri Sağlanana Kadar do
    İşçi Arıları Gönder.
    Gözcü Arıları Gönder.
    Kâşif Arıları Gönder.
    En İyi Çözümleri Sakla.
Return En iyi Çözüm
```

Yöntemin ilk aşamasında var olan besin kaynağının sayısı (BKS), çözüm üzerinde iyileşme gösterilmemesi durumunda maksimum deneme sayısı (MDS), çözüm uzayı için belirlenecek alt ve üst sınır değerleri ve yöntemin tekrar sayısı belirlenmektedir. Yöntemin ikinci aşamasında ise belirlenen parametrelere göre her bir besin kaynağının geçerli çözüm içerdiği ilk besin kaynakları kümesi Eşitlik 4'e göre oluşturulmaktadır. Küme içerisindeki besinler belirlenmiş aralıkta rastgele çözümleri içermektedir. B_i

besin kaynağını, AS alt sınır değerini, US üst sınır değerini belirtmektedir.

$$B_i = AS + (US - AS) \times rand[0,1] \tag{4}$$

Üçüncü aşamada işçi arılar çözüm kümesinde bulunan elemanların Eşitlik 5'e göre hesaplanan uygunluk değerini arttırmak amacıyla ziyaret gerçekleştirir. Elde ettikleri uygunluk değerini gözcü arılar ile paylaşırlar. Bu sayede koloni üzerinde haberleşme sağlanmış olur. f_i çözülmesi beklenen problemin amaç fonksiyonundan elde edilen değerdir.

$$U_i = \begin{cases} 1 + \frac{1}{f_i} & \text{eğer } f_i \geq 0, \\ 1 + |f_i| & \text{eğer } f_i < 0. \end{cases} \tag{5}$$

Gözcü arılar, çözümü Eşitlik 6'yı kullanarak daha da geliştirmeye çalışırlar. V, yeni çözüm değerini, ϕ_i ise $[0,1]$ aralığında rastgele elde edilmiş sayıyı, B_j uygunluk değerine göre olasılık dahilinde seçilmiş komşu çözümü ifade etmektedir. Eğer çözüm belirlenen deneme sayısına kadar iyileşme gösteremiyor ise yeni geçerli ve rastgele çözüm Eşitlik 4'e göre üretilir. Tüm çözümler değerlendirildikten sonra en iyi çözümler hafızada tutulur ve süreç, durdurma kriteri sağlanana kadar tekrar eder.

$$V_i = B_i + (B_i - B_j) \times \phi_i \tag{6}$$

Gözcü arı fazında geliştirilecek olan besin kaynakları Eşitlik 7'ye göre olasılık değerleri belirlenerek, bu olasılık değerlerine bağlı olarak seçim gerçekleşmektedir.

$$p_i = \frac{U_i}{\sum_{i=1}^{n} U_i} \tag{7}$$

3.2. Tavlama benzetimi algoritması

Tavlama benzetimi (TB), ilk olarak 1953 yılında, Metropolis ve arkadaşları tarafından bir ısı banyosu içerisindeki taneler kümesinin denge dağılımını hesaplamak amacıyla simülasyon tekniği ile kullanılan bir yöntemdir [15]. Otuz yıl sonra, 1983 yılında, Kirkpatrick ve arkadaşları bu simülasyonun optimizasyon problemlerinin çözümünde kullanılabileceğini belirlemiştir [16].

TB kombinatoryal eniyileme problemleri için iyi çözümler veren stokastik arama yöntemidir. Tavlama Benzetimi ismi, katıların fiziksel tavlanma süreci ile olan benzerlikten ileri gelmektedir. Eğer katı bir madde erime noktasına kadar ısıtılır ve sonra hızla soğutulmaya başlanırsa maddenin moleküler yapısı soğutma hızına bağlı olarak değişir. Bir madde ile bir optimizasyon problemi temsil edilecek olursa, maddenin halleri farklı çözümler olacaktır. Böylece mevcut haldeki enerji maliyet fonksiyonuna karşılık gelecektir. Tavlamada madde katı haline(en düşük enerji seviyesi) gelinceye kadar yavaşça soğutulmaktadır. Asgari enerjili bir sistem en düşük maliyet fonksiyonlu optimizasyon probleminin çözümünü verecektir.

Aslında bu yöntem, yerel aramanın farklı bir versiyonudur. Yerel arama metodunun en büyük dezavantajı, global optimum yerine yerel optimum bulmasıdır. Tavlama benzetimi, bu olumsuzluğu ortadan kaldırır. Kontrol parametrelerine (sıcaklık) bağlı olarak, komşuluk değerlerindeki maliyet fonksiyonu değişimini kabul eder.

TB bir katının minimum enerji durumu elde edilene kadar yavaş yavaş soğutulduğu fiziksel tavlama sürecini taklit eden stokastik arama yöntemidir. Bu yöntem ile üretilen çözümler sırasının amaç fonksiyon değerleri genel bir azalma eğilimindedir. Ancak, bazı durumlarda amaç fonksiyonu değerleri yüksek olan çözümler de kabul edilebilmektedir. Bu tür kötü çözümlerin kabul edilmesindeki amaç, bir yerel en iyi etrafında yapılan aramadan çıkıp global en iyi için aramaya devam etmektir. TB'nin kombinatoryal eniyileme problemleri için eniyi çözüme yakın çözümler üreten bir yöntem olduğu çeşitli çalışmalarda gösterilmiştir [17–19].

TB algoritmasının bir probleme uygulanması aşamasında karar verilmesi gereken önemli kavramları iki grupta toplamak mümkündür.

1. Genel Kararlar:
   - Başlangıç sıcaklığı, T_0
   - Soğutma planı, (tekrar sayısı ve sıcaklık düşürme oranı)
   - Durdurma koşulu.
2. Probleme özgü kararlar:
   - Çözüm uzayı,
   - Minimize edilecek bir amaç fonksiyonu,
   - Komşuluk üretim mekanizması tanımlanmalı,
   - Başlangıç çözümü üretilmelidir.

Başlangıç sıcaklığı, T_0, başlangıçta kötü çözümleri kabul edecek ve elde edilen son çözümün başlangıç çözümünden bağımsız olmasını sağlayacak kadar yüksek sıcaklık olmalıdır. Tavlama benzetiminde, sıcaklık sabit değildir, belirli bir iterasyona göre azaltılır. Sıcaklık parametresinin azaltılması, çözüme ulaşılmada önemli bir parametredir.

Tavlama benzetimi algoritmasının sözde kodu Algoritma 2'de verilmiştir.

Algoritma 2 Tavlama benzetimi algoritması

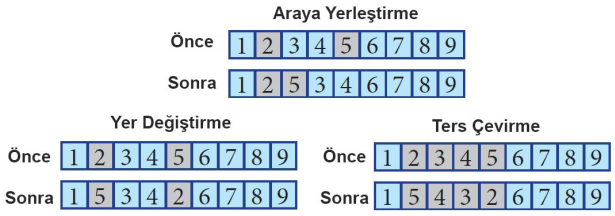
```
İlk T sıcaklığını belirle
α soğutma parametresini belirle
İlk π çözümünü üret ve onun f(π) değerini hesapla
while T > 0.1 do
    Yeni bir π' çözümü üret ve f(π') değerini hesapla
    Δf = f(π') − f(π)
    if Δf < 0 then
        π = π'
    else if random[0,1] < exp(−Δf/T) then
        π = π'
    else π' çözümünü iptal et
        T = T × α   \\ sıcaklığı güncelle
Return En iyi Çözüm
```

3.3. Komşu çözüm üretme operatörleri

Varolan çözümden yeni bir çözüm elde etmek için komşuluk operatörleri kullanılır. • Algoritma 3'de sözde kodu verilen önerdiğimiz hibrit algoritmanın 7. ve 18. satırlarında işçi arı ile gözcü arılar tarafından bu operatörler kullanılmıştır. Literatürde metasezgisel algoritmalarda kullanılan en yaygın komşuluk operatörleri şunlardır [20].

1. **Yer değiştirme (Swap) operatörü:** Çözüm kümesinde rastgele seçilen iki sıradaki elemanların konumları değiştirilir.
2. **Ters çevirim (Inversion) operatörü:** Çözüm kümesinde rastgele seçilen iki nokta arasındaki elemanların yerleri ters çevrilir.
3. **Araya ekleme (Insertion) operatörü:** Çözüm kümesinde rastgele seçilen iki elemandan, ikinci eleman birincinin sağına eklenir. Arada kalanlar elemanlar ise sağa kaydırılır.

Bu üç operatör için birer örnek Şekil 1'de verilmiştir.



Şekil 1. Komşuluk Operatörleri

3.4. Seçme operatörleri

Evrimsel veya Sürü zekası tabanlı optimizasyon yöntemlerinde, sonraki nesillere aktarılacak bireylerin belirlenmesi için seçme yöntemleri kullanılmaktadır. Bu bölümde literatürde en çok kullanılan seçme operatörleri turnuva seçimi, rulet tekeri seçimi ve budama seçimi aktarılacaktır.

1. **Turnuva seçimi [21] :** Turnuva seçimi işleminde besin kaynaklarından seçilen belirli sayıda çözüm kendi aralarında uygunluk değerlerine göre turnuvaya sokularak kazanan çözüm seçilir. Turnuva seçim operatörü ile uygunluk değeri iyi çözümlerin seçilme olasılığı artar.
2. **Rulet tekeri seçimi [21] :** Rulet tekeri seçim operatöründe çözümlerin uygunluk değerlerine oranla seçim yapılır. Rulet tekeri seçim operatörü için öncelikle uygunluk değerleri toplanır. Bu toplam rulet tekerine yerleştirilecek uygunluk değerleri için rulet tekerinin tamamıdır. Rastgele bir sayı üretilir. Bu sayı rulet tekerinde hangi çözüme ait kısma denk gelirse o çözüm seçilir.
3. **Budama (Truncation) seçimi [22] :** Bu seçim operatörü, rulet tekeri seçim yönteminin doğası gereği sahip olduğu erken yakınsama(premature-convergence) probleminin çözümü için önerilmektedir. Bu seçimde tüm bireyler uygunluk değerine göre sıralanarak, belirlenen bir sayıda en iyi ilk çözümler seçilir. Pratikte diğer tekniklere göre daha az kullanıldığı görülmektedir.

3.5. Önerilen hibrit algoritma

YAK algoritması sürü zekasını kullanarak global en iyi çözüme ulaşma konusunda oldukça başarılıdır [13]. TB algoritması ise tek bir aday çözüm ile en iyi çözüme ulaşma konusunda oldukça başarılıdır [16]. Nümerik çözümler için iki yöntem birleştirilerek global çözüme daha hızlı yakınsanabilmektedir [23].

YAK algoritmasında, başlangıç popülasyonu oluşturulurken ve kaşif arı fazında varolan çözümü değiştirmek amacıyla çözüm uzayı içerisinde olan geçerli rastgele çözüm üretilerek popülasyona dahil edilmektedir. Bu çalışmada önerilen tavlama benzetimi algoritması kullanarak hibrit bir yapay arı kolonisi algoritması (HYAKTB) ile bu aşamalarda üretilecek olan her bir rastgele çözüm TB algoritması ile iyileştirilerek popülasyona dahil edilir. Bu sayede, yeni çözümün kendisinden çok daha yüksek uygunluğa sahip diğer besin kaynakları ile rekabet içerisine girmesi engellenmiş olur. HYAKTB algoritmasının TB aşamasında amaç en iyi çözümü bulmak değil, popülasyon içerisine global çözüme ulaşabilmek amacıyla iyileştirilmiş çözümlerin eklenmesini sağlamaktır. Kaşif arı fazında kullanılan TB algoritması ile üretilmiş olan yeni çözümün, rastgele geçerli çözümden daha iyi sonuç vermesi beklenmesine rağmen, eski çözümden daha uygun sonuç vereceği garanti edilmez.

HYAKTB algoritmasının sözde kodu Algoritma 3 de verilmiştir. Algoritmanın 7. ve 18. satırlarında var olan bir çözüme komşu operatörleri kullanılarak yeni çözümler üretilmektedir. Bu çalışmada kullanılan komşuluk operatörleri Bölüm 3.3 açıklanmıştır. Algoritmanın 13. satırında yukarıda belirtilen seçim operatörlerine göre her bir bireyin seçilme olasılığı hesaplanmaktadır. Seçim işlemi sonucunda belirlenen bireyin olasılık değeri $p(i) = 1$ olarak güncellenmekte, diğer bireylerin olasılık değerleri 0 olarak güncellenmektedir. Algoritmanın 17. satırında ise gözcü arı tarafından yapılan besin seçme işleminde kullanılan seçme operatörleri 3.4 de anlatılmıştır.

4. Hesaplama Denemeleri

YAK algoritması problem üzerinde çalıştırılmadan önce yöntemin çalışacağı parametre değerleri ve komşuluk operatörünün belirlenmesi gerekmektedir. SHA problemi için önceki çalışmalar ile adil bir karşılaştırma sağlanması amacıyla parametre değerleri, kolonide bulunan arı sayısı, limit değer ve maksimum iterasyon sayısı sırasıyla $50, 1000, 200.000$ olarak belirlenmiştir.

Algoritma 3 C programlama dilinde yazılarak Sonuç ve arkadaşları tarafından üretilmiş olan problem örnekleri [24] üzerinde, i7-3930K 3.2 GHz işlemciye sahip 32 GB belleği bulunan makina üzerinde 10 kez çalıştırıldığında Tablo 1'de bulunan sonuçlar elde edilmiştir.

Tablo 1'den görüldüğü üzere, yöntem 5 ila 200 arasında farklı boyutlarda 12 farklı problem kümesi üzerine uygulanmıştır. Yöntem, küçük boyutlu problemler

Tablo 1. Önerilen hibrit algoritmanın hibrit olmayanlar ile karşılaştırılması.

			Sonuc ve ark. TB çalışması		Durgut ve ark. YAK çalışması		Önerilen HYAKTB çalışması			
Problem	#Silah	#Hedef	En iyi	Süre	En iyi	Süre	En iyi	En Kötü	Ortalama	Süre
WTA1	5	5	48.364	2986	48.364	390	48.364	48.364	48.364	18
WTA2	10	10	96.3123	2841	96.3123	417	96.3123	96.3123	96.3123	21
WTA3	20	20	142.107	2753	142.107	473	142.107	142.107	142.107	25
WTA4	30	30	248.0285	2754	248.0285	532	248.0285	248.1678	248.4222	32
WTA5	40	40	305.5016	2761	305.8729	585	305.5016	306.2735	307.1293	36
WTA6	50	50	353.0767	2790	353.3794	654	353.0149	354.6901	357.2952	42
WTA7	60	60	415.0528	2788	414.4555	712	414.7521	417.3107	420.6054	46
WTA8	70	70	498.1049	2841	498.0948	786	496.9645	498.3417	500.6414	52
WTA9	80	80	534.4408	2869	534.4742	831	531.4078	534.4042	536.5087	60
WTA10	90	90	594.0639	2813	592.9167	889	590.478	592.4761	595.191	71
WTA11	100	100	699.8357	2806	698.4465	954	694.8067	696.3017	700.431	79
WTA12	200	200	1306.9126	2902	1295.3142	1624	1287.024	1289.16	1291.279	124

Algoritma 3 Hibrit YAK-TB algoritması

Girdi: T, α, MaksIter, limit, BS (Besin Sayısı)

for i= 1 to BS **do**
    B_i=Rastgele çözüm üret
iter=0
while iter < MaksIter **do**
    /* İşçi Arı Fazı */
    for i= 1 to BS **do**
        $V_i = B_i$ ye komşu çözüm operatörü kullan
        if $f(V_i) < f(B_i)$ **then**
            $B_i \leftarrow V_i$
            $D_i = 0$
        else
            $D_i = D_i + 1$
    $P = Secim_Olasiliklari(B)$
    /* Gözcü Arı Fazı */
    for t= 1 to BS **do**
        $i = 1$
        if $rastgele_sayi < P_i$ **then**
            $V_i = B_i$ ye komşu çözüm operatörü kullan
            if $f(V_i) < f(B_i)$ **then**
                $B_i \leftarrow V_i$
                $D_i = 0$
            else
                $D_i = D_i + 1$
        else
            $i = i + 1$;
    /* Kaşif Arı Fazı */
    for i= 1 to BS **do**
        if $D_i > limit$ **then**
            B_i=Tavlama Benzetimi Algoritması(T,α)
            D_i=0
    iter=iter+1;
Return En iyi Çözüm

(WTA1, WTA2, WTA3, WTA4) için diğer yaklaşımlarda elde edilen değerlere başarıyla ulaşmaktadır. Bunun yanı sıra, büyük boyutlu problemlerde ise var olan çözümlerden daha uygun çözüm değeri sunmaktadır. Sonuc ve arkadaşları tarafından uygulanan TB algoritması ile karşılaştırıldığında ise çalışma hızındaki iyileşme açıkça görülmektedir. Durgut ve arkadaşlarının önerdiği YAK algoritması [25] ile bu çalışmada önerilen HYAKTB yöntemi farklı platformlarda koşturulduğundan çalışma hızı bakımından karşılaştırmak uygun olmayacaktır.

Tablo 2'de HYAKTB ile birlikte farklı komşuluk ve seçme operatörleri kullanıldığında elde edilen veriler karşılaştırmalı olarak sunulmuştur. Tablodaki değerler yöntemin 10 farklı koşturma sonunda elde ettiği sonuçların ortalamasıdır. Düşük boyutlu problemlerin (WTA1,2,3) çözümünde tüm operatör kombinasyonları optimum sonuca ulaşabilmiştir. Fakat problem boyutu arttığında yer değiştirme operatörü diğer komşuluk operatörlerine üstünlük sağlamıştır. Yer değiştirme operatörü ile birlikte kullanılan seçme operatörleri arasında ise rulet tekeri seçim operatörü WTA7-WTA12 problem örnekleri için en iyi sonuçları sunmuştur. Sadece WTA4-WTA6 problem örnekleri için Budama seçme operatörü en iyi sonucu sunmuştur. Rulet tekerinin en iyi olduğu örneklerde, en iyi çözüme en yakın sonuçlar Turnuva seçme operatörü ile elde edilmiştir.

5. Tartışma ve Sonuç

Bu çalışmada, YAK metasezgisel optimizasyon yönteminin kombinatoryel problemler üzerindeki varolan başarısını arttırmak amacıyla tavlama benzetimi ile hibritleştirilmesi üzerine bir öneri sunulmuştur. Silah Hedef Tahsis problemi üzerinde uygulama sonucunda elde edilen sonuçların gösterdiği üzere, HYAKTB yöntemi TB yöntemine göre çalışma performansı bakımından üstünlük sağlamaktadır. Bunun yanı sıra, YAK ve TB yöntemlerine göre en uygun çözümü bulduğu ya da daha uygun sonuç değerleri sunduğu görülmüştür.

Tablo 2. HYAKTB algorimasında seçim ve komşuluk operatörlerinin karşılaştırılması.

	Yer Değiştirme			Ters Çevirme			Araya Ekleme		
Problem	Rulet	Turnuva	Budama	Rulet	Turnuva	Budama	Rulet	Turnuva	Budama
WTA1	48.364	48.364	48.364	48.364	48.364	48.364	48.364	48.364	48.364
WTA2	96.3123	96.3123	96.3123	96.3123	96.3123	96.3123	96.3123	96.3123	96.3123
WTA3	142.107	142.107	142.107	147.4124	146.7263	147.1116	152.3175	153.087	153.4097
WTA4	248.1678	248.5536	248.0479	260.4236	264.8435	261.6379	285.0665	285.3968	276.3458
WTA5	306.2735	306.9469	305.6951	340.2619	337.8319	332.9387	375.8117	378.0912	354.6336
WTA6	354.6901	354.9889	354.5604	399.5235	408.1023	401.1685	453.0812	457.9732	438.5793
WTA7	417.3107	417.7178	418.4564	477.332	485.034	478.66	534.1713	547.0662	531.008
WTA8	498.3417	499.4603	503.1965	581.3954	587.706	591.6558	654.8817	673.0279	645.7609
WTA9	534.4042	536.9038	543.1358	624.5918	640.0185	647.0905	703.1444	727.7925	712.9069
WTA10	592.4761	594.1051	605.8005	717.2909	725.9431	738.0814	797.9968	824.0721	804.5174
WTA11	696.3017	700.1986	714.2695	840.7641	858.9128	875.4949	926.6049	965.6045	959.3618
WTA12	1289.16	1292.739	1346.722	1634.555	1671.937	1845.728	1803.269	1876.898	2048.817

Ayrıca, hibrit yöntemin üzerinde farklı seçim ve komşuluk operatörleri uygulandığında, en iyi sonuçların yer değiştirme operatörü ve rulet tekeri seçim operatörü kullanılarak elde edildiği gözlemlenmiş ve sunulmuştur. Hatta, bu makalede gösterilmemiş olsa da Sonuç ve arkadaşlarının geliştirdiği paralel TB yönteminden de daha iyi sonuçlar elde edilmiştir ([24] makalesinde Tablo V deki değerlere bakılabilir).

Teşekkür

Hakemlere, makaleyi yayınlanabilir forma getirebilmemiz için yapmış oldukları değerli görüşleri için teşekkür ederiz.

Kaynakça

[1] Ahuja, R., Kumar, A., Jha K., and Orlin J. 2007. Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. Operations Research 55(6), 1136–1146.

[2] Hosein, P., and Athans, M. 1990. Some analytical results for the dynamic weapon-target allocation problem, Tech. Rep., MIT, Laboratory for Information and Decision Systems.

[3] Cai, H., Liu, J., Chen, Y., and Wang, H. 2006. Survey of the research on dynamic weapon-target assignment problem, Journal of Systems Engineering and Electronics, 17(3), 559-565.

[4] Manne, A. S. 1958. A target-assignment problem. Operations Research 6, 346-351.

[5] Lloyd, S. P., and Witsenhausen H. S. 1986. Weapon Allocation is NP-Complete [C]. Proc. 1986 Summer Compute. Simulation Conference.

[6] Ma, F., and Ni, M. 2015. An optimal assignment of multi-type weapons to single-target, 2015 IEEE Advanced Information Technology Electronic and Automation Control Conference. IEEE, 1-4.

[7] Çetin, E. and Esen, S. T. 2006. A weapon-target assignment approach to media allocation", Applied Mathematics and Computation, 175(2). 1266-1275.

[8] Senay, N. 2012. The strategic level optimization of air to ground missiles for Turkish air force decision support system, Air Force Institute of Technology, 7.

[9] Li, Y., Kou, Y., Li, Z., Xu A., and Chang, Y. 2017. A Modified Pareto Ant Colony Optimization Approach to Solve Biobjective Weapon-Target Assignment Problem, International Journal of Aerospace Engineering, 1-14.

[10] Lee, Z. J., Su, S. F. and Lee C. Y. 2003. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics IEEE Trans. Syst. Man Cybern,B: Cybern., 33(1), 113-121.

[11] Lu H., Zhang H., Zhang X. and Han Ruixin H. 2006. An Improved Genetic Algorithm for Target Assignment, Optimization of Naval Fleet Air Defense. 6th World Congress on Intelligent Control and Automation, Dalian, China, 3401-3405.

[12] Liu, X., Liu, Z., Hou, W. -S. and Xu, J.-H. 2013. Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment, Systems Engineering & Electronics, 35(2), 326–330.

[13] Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization, Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[14] Karaboga D., and Basturk, B. 2008. On The Performance Of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, 8(1), 687-697.

[15] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. 1953. Equation of state calculations by fast computing machines, Journal of Chemical Physics, 21, 1087-1092.

[16] Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P. 1983. Optimization by simulated annealing, Science, 220, 671-680.

[17] Ahonen H., Alvarenga A.G., Amaral A.R.S., 2014. Simulated annealing and tabu search approaches for the Corridor Allocation Problem, European Journal of Operational Research, 232(1), 221-233.

[18] Paul G., 2010. Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem, Operations Research Letters, 38(6), 577-581.

[19] Birim Ş., 2016. Vehicle Routing Problem with Cross Docking: A Simulated Annealing Approach, Procedia - Social and Behavioral Sciences, 235, 149-158.

[20] Larrañaga P., Kuijpers C.M.H., Murga R.H., Inza I., Dizdarevic S. 1999. Genetic algorithms for the travelling salesman problem: a review of representations and operators. Artif Intell Rev. 13(2), 129–170.

[21] Holland J.H. 1992. Adaptation in natural and artificial systems. Cambridge: MIT Press.

[22] Mühlenbein H. and Schilierkamp V. 1993. Predictive Model for Breeder Genetic Algorithm, Evolutionary Computation. 1(1), 25–49.

[23] Chen, Shi-Ming, Ali Sarosh, and Yun-Feng Dong, 2012. Chen, Shi-Ming, Ali Sarosh, and Yun-Feng Dong. Simulated annealing based artificial bee colony algorithm for global numerical optimization. Applied mathematics and computation 219.8 (2012): 3575-3589.

[24] Sonuc E., Sen B., and Bayir S. 2017. A Parallel Simulated Annealing Algorithm For Weapon-Target Assignment Problem, International Journal of Advanced Computer Science and Applications, 8(4), 87-92.

[25] Durgut R., Kutucu H., and Akleylek S. 2017. An Artificial Bee Colony Algorithm for Solving the Weapon Target Assignment Problem. In Proceedings of the 7th International Conference on Information Communication and Management (ICICM 2017). Moscow, Russian Federation, ACM, 28-31.