SETSCI conference proceedings
SETSCI Conference Proceedings
9, 502-507, 2019

4th International Symposium on Innovative Approaches in Engineering and Natural Sciences
November 22-24, 2019, Samsun, Turkey

https://doi.org/10.36287/setsci.4.6.144
2687-5527/ © 2019 The Authors. Published by SETSCI

ISAS WINTER - 2019

MOEA/D Algoritmasının En Önemli İki Ayrıştırma Yöntemi Olan PBI ve TCH'ın Analizi

Mustafa ALTIOK*, Barış KOÇER [2]

[1]Bilgisayar Mühendisliği/Bilgisayar Mühendisliği A.B.D., Tokat Gaziosmanpaşa Üniversitesi, Tokat, Türkiye
[2]Bilgisayar Mühendisliği,Bilgisayar Mühendisliği A.B.D., Konya Teknik Üniversitesi, Konya, Türkiye
*Corresponding author: mustafa.altiok@gop.edu.tr
+Speaker: mustafa.altiok@gop.edu.tr
Presentation/Paper Type: Oral / Full Paper

Özet **–** Günümüzde, birçok araştırmacı optimizasyon problemleri çözen algoritmaları geliştirme eğilimindedir. Gerçek dünya problemleri ise genelde birden çok amaca sahip oldukları için çok amaçlı optimizasyon problemleri üzerine geliştirilen algoritmalar trend olmuştur. Bu çalışmada, çok amaçlı optimizasyon algoritması olan Multi Objective Evulation Algorithm(MOEA/D) incelendi ve algoritma içerisinde yer alan ve genelde amaçların tek bir amaç olarak ifade edilmesini sağlayan en önemli iki metot Tchebycheff (TCH) ve Penalty-based Boundary Intersection (PBI) metotları iki ve üç amaçlı 11 benchmark(DTLZ ve ZDT ailesi) ile test edildi ve bu alanda en çok kullanılan Hypervolume(HV), Inverted Generation Distance(IGD) ve EPSİLON adında üç adet metrik ile bu sonuçlar kıyaslandı. Bu algoritmanın artıları ve eksileri öne çıkarıldı. Sonuçta MOEA/D hangi yöntem ile kullanılırsa daha verimli olacağı gösterilmiş oldu. Bilim her zaman üzerine koyarak ilerleyen bir gerçeklik olduğu için, bu çalışma MOEA/D üzerine çalışma yapacak araştırmacılara yön gösterici bir çalışma oldu.

Keywords *– Optimizasyon, Çok amaçlı optimizasyon, MOEA/D, algoritma,Ayrıştırma yöntemi*

I. GİRİŞ

Optimizasyon hayatın her alanında karşımıza çıkan bir kavramdır. Farkında olmadan bazen daha az enerji harcamak için kısa yoldan gideriz böylece zaman ve enerji minimizasyonu yapmış oluruz. Hayatımızda bu ve buna bir çok kere optimizasyon yaparız. Bu gibi örnekler aslında kuramsal bir temele dayanmayan deney ve tecrübe ile anlaşılmış olan amprik(deneysel) olaylardır. Gerçek dünya problemlerinde sonucu etkileyen çok fazla sayıda parametre vardır bu yüzden hangi davranışın sonucu en iyi(optimum) şekilde etkilediği yada en iyi sonucun bulunup bulunamadığı tam olarak anlaşılamaz. Optimizasyon, elde edilmek istenilen amaç için kısıtları sağlayan en uygun çözümün elde edilmesidir. Optimizasyon sayesinde sonucu en iyi şekilde etkileyen karar değişkenleri bulunur[1].

Genellikle bilim adamları ve mühendislerin ilgilendiği problemler çok amaçlıdır ve amaçlar birbiri ile çelişmektedir. Çeşitli amaçlar birbirleri ile çeliştikleri için problemlerin tüm amaçlarını karşılayan çözümleri mevcut değildir bu sebepten problem NP hard problemine indirgenmiştir. İlk olarak çok amaçlı optimizasyon problemlerini(ÇAOP) çözmek için evrimsel algoritmaları(EAs) kullanma fikri 1967 de ortaya atılmıştır. Fakat bu çalışmada ÇAOP tek amaca indirgendi ve EAs ile çözüldü. Araştırmacılar 1980'in ortalarına kadar ÇAOP'ları çözmek için matematiksel yöntemler kullandı. ÇAOP'lar kesin yöntemler ile çözülürken yerel optimuma takılma sorunu ortaya çıktı. Bu nedenden dolayı araştırmacılar sezgisel yöntemlere çalışma eğiliminde oldular[2].

1984 yılında Vektör Değerlendiren Genetik Algoritma(VEGA) geliştirildi ve ÇAOP'lar için stokastik(evrimsel ve sezgisel) optimizasyon teknikleri içeren ilk MOEA önerildi[3]. Bundan sonra bir çok araştırmacı tek amaçlı evrimsel ve sezgisel algoritmaları çok amaçlı optimizasyon algoritmalarına uyarladı. Bunlardan en yaygın olanları şöyle: Bastırılamamış Sıralı Genetik Algoritma 2 (NSGA2)[4], Çok amaçlı Parçacık Sürü Optimizasyonu(MOPSO)[5], İndikatör Tabanlı Evrimsel Algoritma(IBEA)[6] ,Ayrıştırmaya Dayalı Çok Amaçlı Evrimsel Algoritma (MOEA/D)[7], Çok Amaçlı Girdap Arama Algoritması(MOVS)[2], Çok Amaçlı Yapay Arı Kolonisi Algoritması(MOABC)[8] vb…

II. MATERYAL METOT

Çok amaçlı optimizasyon probleminin matematik modeli aşağıdaki gibidir.

$$X = [x_1, x_2, \dots \dots x_N]^T \quad (1)$$

T transpozu ifade etmektedir ve N ise parametre sayısını göstermektedir. Problemi temsil eden her bir birey bu parametrelerden oluşmaktadır. Parametrelerin alt ve üst sınırları probleme özgü olarak değişmektedir. ÇAOP için amaç fonksiyonu denklem 2'de gösterilmiştir.

$$F(x) = [f_1(x), f_2(x), \dots \dots, f_m(x)]^T \quad (2)$$

Burada m amaç sayısıdır. Her amaç fonksiyonu ya maksimuma veya minimuma gidecektir. Parametre uzayında ki her x değerine karşılık amaç uzayında bir nokta bulunmaktadır. Bu nokta F(x)=z olarak gösterilir[9].

A. MOEA/D'ın Ana Çerçevesi

Ayrıştırma, geleneksel çok amaçlı optimizasyon alanında temel stratejilerden biridir. Bu algoritma, çok amaçlı bir optimizasyon problemini bir dizi skaler optimizasyon alt problemine ayırır ve aynı anda optimize eder. Her bir alt problem yalnızca komşulardan gelen bilgilerle optimize edilir[7].

MOEA/D için şu bileşenler gereklidir.

- N noktalarının popülasyonu P $:x^1, \dots, x^N$ Burada x^i, i. alt problemin güncel çözümüdür.
- Bir ağırlık vektörü: $: \lambda = (\lambda_1, \dots, \lambda_m)\ \lambda_i \geq 0$ bütün i,…,m için ve $\sum_{i=1}^{m} \lambda_i = 1$
- Referans noktası vektörü : $z^* = (z_1^*, \dots, z_m^*)$
- Komşular : Her bir i=1,…N Bir komşu, $B(i) = \{i_1, \dots, i_T\}\ \lambda^{i_1}, \dots, \lambda^{i_T}$ Burada T, oklit mesafesi olarak birbirlerine en yakın ağırlıklardır[10].

Tablo 1. MOEA/D adımları

```
1: // Başlangıç
2: λ ← initializeWeightVectors()
3: B ← initializeNeighborhood()
4: P ← initializePopulation()
5: z ←initializeIdealPoint()
6:
7: // Ana döngü
8: while not terminationCondition() do
9: for (i = 0 ; i < populationSize; i++) do
10: if (rand(0, 1) < δ) then
11: MP ← B(i) // MP = mating pool
12: else
13: MP ← P
14: end if
15: parents ← selection(MP, i) //DE selection
16: child ← recombination(parents) // DE operatör
17: child ← mutation(child, Pm) // Polynomial mutation
18: evaluateFitness(child)
19: idealPointUpdate(child, z)
20: solutionsUpdate(child, MP, nr)
21: end for
22: end while
23: return P
```

2-5 : Gerekli parametreler başlatıldı.

8 : Sonlandırma koşulu sağlanana kadar ana döngü çalışır.

9 : Her i. birey için bir iterasyon çalışır.

10-15 : Eşleşme havuzunda ihtimal δ'e göre P ya da B(i)'den iki birey seçilir.

16-17 : i. bireyden yeniden birleştirme ve mutasyon işlemleri ile yeni bir birey elde edilir.

18 : Yeni bireyin fitnes değeri hesaplanır.

19 : İdeal point güncellenir.

20 : Yeni birey eğer popülasyonun i. birey ve i. bireyin komşularından daha iyi olduklarının yerine geçer[10].

B. MOEA/D'da Kullanılan Önemli Ayrıştırma Yöntemleri

Tchebycheff Method

Bu yaklaşım skaler optimizasyon problem formundadır. Matematik modeli aşağıdaki gibidir.

$$g^{te}(x/\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i f_i(x) - z_i^*\} \quad (3)$$

Formül 3' de Z referans noktası, F(x) fitness değeri ve λ değeri ise ağırlık değerleridir. Bu yaklaşımdaki zayıf noktalardan biri, toplama işlevinin sürekli bir MOP için düzgün olmamasıdır. Bununla birlikte, algoritmamızın toplama işlevinin türevini hesaplaması gerekmediğinden bu çalışma için kullanılabilir bir formüldür.

Ceza Tabanlı Sınır Kesişimi Yaklaşımı(PBI) Metot

Ceza sınırı kavşak (PBI) yöntemi, ayrışma bazlı algoritmalarda sıkça kullanılan bir skalarizasyon yöntemidir. W ağırlık değerine göre x çözümü için d1 + θd2 formunun cezalı bir mesafe değerini döndürür. Burada d1, referans hattı boyunca mesafeyi belirtir ve d2, w' ye dik mesafeyi belirtir.

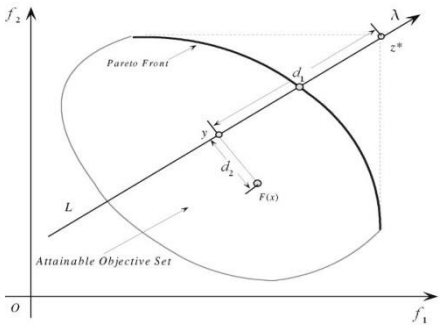
$$\text{Minimize } g^{PBI}(x|\lambda, z^*) = d1 + \theta d2 \quad (4)$$

Subject to xϵΩ

$$d1 = \frac{\|(z^* - F(x))^T \lambda\|}{\|\lambda\|} \quad (5)$$

$$d2 = \|F(x) - (z^* - d1\lambda)\| \quad (6)$$

Genellikle MOEA/D'da θ uygun bir değer seçilirse iyi sonuçlar alınabilir. θ değerinin küçük olması durumda ise yakınsama hızı artar. Bununla birlikte θ küçük seçilirse konveks olmayan yapılarda iyi sonuçlar elde edilemez.

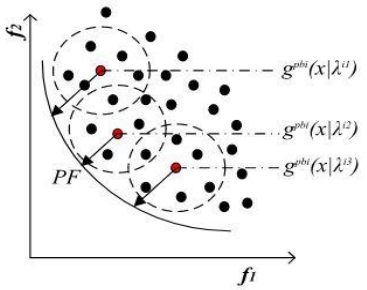


Şekil. 1 PBI metodunun gösterimi

C. MOEA/D'ın Diğer Kısımları

Komşuların Belirlenmesi

MOEA/D, yeni çözümler oluşturmak için mahalle bilgisini kullanır ve her bir amacı eşzamanlı olarak optimize eder. MOEA/D ağırlık vektörlerin Öklid mesafesine bakarak komşulukları ayarlar.



Şekil. 2 Komşuların Seçilmesi

Şekilde görüldüğü gibi her bir birey için kendisine en yakın bireyler komşu seçilmiştir. Böylece komşulardan gelen bilgilere problem optimize edilir.

Seçim Mekanizması

MOEA/D'in seçim mekanizması rastsallığa dayanır. Yani, yeni bireyler üretmek için seçilen ebeveyn bireyler büyük bir ihtimalle komşu bireylerden az bir ihtimalle de tüm popülâsyondan seçilir.

Differansiyel Evrim Çaprazlaması

Bu çaprazlama işleminde amaç belirli kriterlere göre seçilen ebeveynlerden çaprazlama sonucunda yavru bireylerin üretilmesidir. Diferansiyel Evrim(DE) operatörüne göre yeniden birleştirme işlemi şöyledir; Denklem 7'de verilen r_1=i. birey, r_2 ve r_3 bireyler ise rastgele olarak komşulardan yada popülasyondan seçilen değerlerdir.

$$u_k = \begin{cases} x_k^{r_1} + F\,X\left(x_k^{r_2} - x_k^{r_3}\right), & with\ probabilty\ CR \\ x_k^{r_1}, & with\ probabilty\ 1-CR \end{cases} \quad (7)$$

Burada F ve CR iki sabit, kontrol parametresi iken, k=1,…n' e kadar olan problemin karar değişkeni sayısıdır.

Komşuların Güncellenmesi

Denklem 3'e göre bir g değeri bulunur. Bu değer aslında birden çok olan amaçların kalitesini temsil eder. Yeni üretilen bireyin g değeri ile tüm komşularının g değerleri karşılaştırılır ve yeni bireyin daha iyi olduğu durumlarda komşular yeni bireye göre güncellenir. Güncelleme işlemi Denklem 8'de gösterilmiştir.

$$j \in B(i),\ if\ g^{te}(y'/\lambda^j, z) \le g^{te}(x^j/\lambda^j, z)\ \ then\ \ set\ \ x^j = y' \\ and\ \ FV^j = F(y') \quad (8)$$

Yani; yeni bireyin fitnes değerinden Z'yi çıkar ve o sonucu bütün komşuların ağırlık değeri ile çarp ve çıkan matrisden sonuçlardan en büyüğünü al ve yeni bireyin g vekörünü oluştur.

Bütün komşu bireylerin fitneslarına da aynı işlemi uygula sonuçta çıkan g vektörünü komşuların g vektörü ile karşılaştır. Bu sonuca göre yeni g değeri daha küçük olduğu durumlarda komşuları sil ve onların yerine yeni bireyin değerlerini ekle.

D. Çalışmanın Kapsamı

Yapılan testler, JMetal isimli ÇAO için tasarlanmış paket program yardımıyla gerçekleştirilmiştir[11]. Bu çalışmada, çok amaçlı optimizasyon problemlerini çözmek için kullanılan MOEA/D için, her bir amaç için ifade edilen değerleri tek bir değere indirgen TCH ve PBI yöntemleri karşılaştırılmıştır.

Yapılan Testler kapsamında İki amaçlı ZDT[12] ve üç amaçlı DTLZ[13] bencmarkları kullanılmıştır. MOEA/D'ın parametreleri ayarlanırken geçmiş literatür çalışmaları göz önüne alınmıştır.

MOEA/D'ın parametrelerinin ayarlanması

- CR:1
- F:0.5
- Populasyon Büyüklüğü:300
- MaxFes:30000
- Komşu Sayısı:30
- Bağımsız çalışma sayısı:30

Farklı çok amaçlı optimizasyon problemlerinde (ÇAOP) yöntemlerinin önerilmesiyle birlikte, bu yöntemlerin çeşitli test işlevleri üzerinde başarımlarının karşılaştırılması gerekli olmuştur. Bir karşılaştırma çalışması yapmadan önce, uygun bir test işlevinin seçilmesine ihtiyaç vardır. Test işlevlerinin Pareto-optimal cephelerinin yerinin bilinmesi gereklidir (hem amaç hem de parametre uzayında).

Çok amaçlı optimizasyonun iki temel amacı vardır:

(i) **Yakınsama:** Pareto-optimal cepheye mümkün olduğunca yakın çözümler bulmak.

(ii) **Dağılım veya Çeşitlilik:** Bulunan bastırılamayan bireylerin Pareto cephesinde mümkün olduğunca düzgün dağılması.

İlk amaç, Pareto-optimal bölgeye doğru bir aramaya ihtiyaç duyarken; ikinci amaç, Pareto-optimal cephe boyunca bir aramaya ihtiyaç duyar. Çeşitliliği fazla olan bir küme, tüm Pareto-optimal cepheye düzgün olarak dağılmış bireylerden (çözümlerden) oluşur. Çeşitlilik ölçütü, aynı zamanda iki farklı ölçüte ayrılabilir: en uç bireylerin yayılımı ve dağılım (bireyler arasındaki bağıl mesafe) .

Pareto-optimal cepheye yakınsama ve bu cephe üzerinde düzgün dağılım birbiriyle çatışan iki farklı amaç olduğu için, bir algoritmanın başarımına karar verebilecek tek bir ölçüt yoktur. 1. algoritma birinci amaç için iyiyken, 2. algoritma ikinci amaç için iyidir. Eğer, Pareto-optimal cepheye yakınsama ile ilgili bir ölçüt tanımlarsak ve bir diğer ölçütü de düzgün dağılım için tanımlarsak, iki algoritma birbirini bastıramaz, yani birbirlerine karşı bir üstünlükleri yoktur. Dolayısıyla, bu iki amacı sağlamak için en azından iki ölçüte ihtiyaç vardır.

Çeşitli bilim adamları tarafından bu iki durumu ayrı ayrı ya da birlikte değerlendiren pek çok başarım ölçütü önerilmiştir ve önerilmeye de devam edilmektedir. Bu bölümde literatürde en çok kullanılan başarım ölçütlerinden 3 tanesi anlatılacaktır. Bunlar:

- Dönüştürülmüş Nesilsel Mesafe (Inverted Generational Distance, IGD)[14]
- Epsilon [15]

- Hiperküp (Hypervolume)[16]

III. SONUÇLAR

Sonuçlara göre TCH metodunun iki ve üç amaçlı problemlerde daha etkili olduğu gözlendi. Ancak özellikle sadece iki amaçlı problemlerin bazılarında PBI yöntemi de başarılı sonuçlar vermektedir.

Tablo 2. HV metriği ile ortalama, standart sapma değerleri

	TCH	PBI
ZDT1	$4.36e-01_{7.8e-02}$	$5.25e-01_{6.1e-02}$
ZDT2	$3.80e-02_{5.2e-02}$	$5.29e-03_{2.3e-02}$
ZDT3	$3.39e-01_{5.6e-02}$	$4.01e-01_{4.0e-02}$
ZDT4	$1.69e-01_{1.5e-01}$	$1.87e-01_{1.4e-01}$
DTLZ1	$5.37e-01_{3.1e-01}$	$6.45e-01_{2.0e-01}$
DTLZ2	$4.20e-01_{1.4e-03}$	$4.34e-01_{2.9e-04}$
DTLZ3	$2.68e-01_{2.7e-01}$	$1.27e-01_{2.3e-01}$
DTLZ4	$4.07e-01_{2.9e-02}$	$4.25e-01_{2.7e-02}$
DTLZ5	$9.29e-02_{8.1e-05}$	$9.27e-02_{8.5e-05}$
DTLZ6	$9.33e-02_{2.5e-05}$	$9.22e-02_{1.5e-05}$
DTLZ7	$1.84e-01_{5.9e-02}$	$1.13e-01_{7.5e-02}$

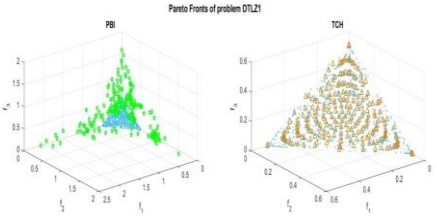
Tablo 3. IGD metriği ile ortalama, standart sapma değerleri

	TCH	PBI
ZDT1	$1.03e-02_{3.8e-03}$	$6.03e-03_{2.9e-03}$
ZDT2	$6.79e-02_{4.0e-02}$	$1.15e-01_{2.8e-02}$
ZDT3	$1.06e-02_{3.9e-03}$	$6.21e-03_{1.8e-03}$
ZDT4	$8.24e-02_{3.2e-02}$	$7.04e-02_{2.6e-02}$
DTLZ1	$4.17e-03_{8.5e-03}$	$1.53e-03_{3.1e-03}$
DTLZ2	$3.51e-04_{4.9e-06}$	$2.42e-04_{2.8e-06}$
DTLZ3	$1.33e-01_{2.4e-01}$	$1.46e-01_{2.3e-01}$
DTLZ4	$9.65e-04_{7.3e-04}$	$4.90e-04_{6.9e-04}$
DTLZ5	$7.31e-05_{2.1e-06}$	$6.07e-05_{2.1e-06}$
DTLZ6	$3.87e-05_{2.3e-07}$	$5.50e-05_{2.2e-07}$
DTLZ7	$4.39e-03_{2.7e-03}$	$1.17e-02_{2.8e-03}$

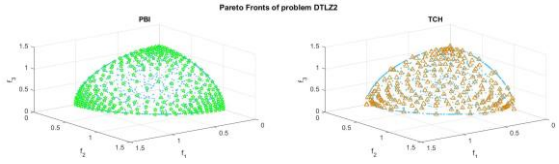
Tablo 4. Epsilon metriği ile ortalama, standart sapma değerleri

	TCH	PBI
ZDT1	$2.26e-01_{9.2e-02}$	$1.65e-01_{7.1e-02}$
ZDT2	$5.90e-01_{2.4e-01}$	$8.33e-01_{1.6e-01}$
ZDT3	$3.56e-01_{1.2e-01}$	$2.20e-01_{6.9e-02}$
ZDT4	$1.90e+00_{9.2e-01}$	$1.56e+00_{7.5e-01}$
DTLZ1	$1.73e-01_{2.5e-01}$	$1.01e-01_{1.0e-01}$
DTLZ2	$8.01e-02_{9.2e-03}$	$7.40e-02_{4.4e-03}$
DTLZ3	$5.50e+00_{9.5e+00}$	$5.66e+00_{8.1e+00}$
DTLZ4	$9.73e-02_{5.6e-02}$	$8.44e-02_{4.7e-02}$
DTLZ5	$1.16e-02_{3.9e-04}$	$2.15e-02_{9.8e-04}$
DTLZ6	$9.75e-03_{9.4e-05}$	$2.28e-02_{7.9e-05}$
DTLZ7	$6.49e-01_{3.2e-01}$	$1.18e+00_{5.6e-01}$

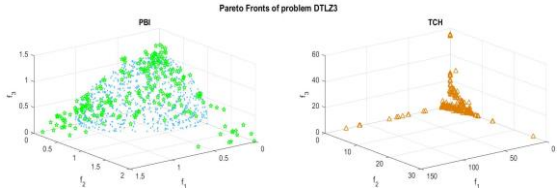
Tablo 2-4 'de görüldüğü üzere 30 bağımsız çalışma sonucunda ortalama ve standart sapmaya göre; koyu renkler en iyi sonuçları, daha açık renkler ikinci kabul edilebilir sonuçları ifade etmektedir. TCH yöntemi hemen hemen bütün problemlerde kabul edilebilir bir sonuç bulmuştur. HV ve IGD metrikleri hem yakınsama hem de dağılım özelliklerini temsil etmekte olduğu göz önüne alınırsa TCH'ın daha güçlü bir yöntem olduğu görünmektedir. Epsilon metriği ise yakınsama yönünden algoritmayı temsil etmektedir ve yine TCH yöntemi ön plana çıkmaktadır. Yinede özellikle iki amaçlı algoritmalarda PBI yöntemi yer yer başarılı olmuştur. Buda bize No Free Lunch[17] teoremini hatırlatmaktadır. Bu teoreme göre bir algoritma ne kadar başarılı olursa olsun her problemde başarılı olamaz bu yüzden her zaman yeni problem çözme yaklaşımlarına ihtiyaç vardır.



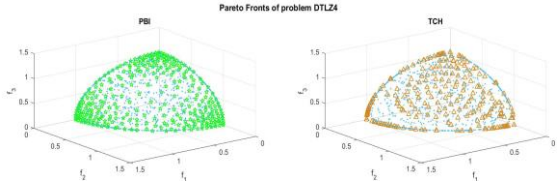
Şekil 3. DTLZ1 probleminin Pareto Front değerleri



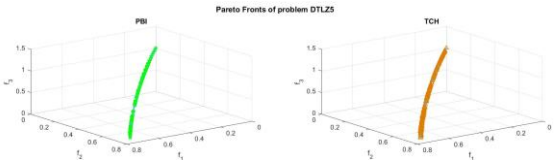
Şekil 4. DTLZ2 probleminin Pareto Front değerleri



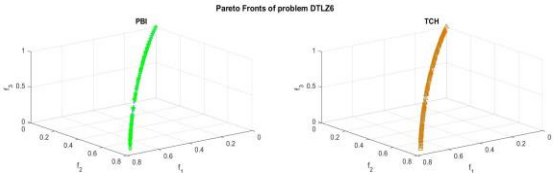
Şekil 5. DTLZ3 probleminin Pareto Front değerleri



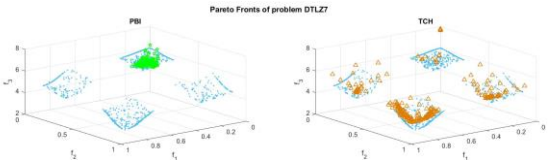
Şekil 6. DTLZ4 probleminin Pareto Front değerleri



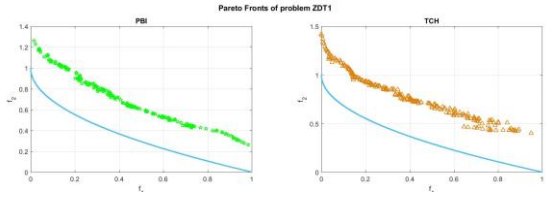
Şekil 7. DTLZ5 probleminin Pareto Front değerleri



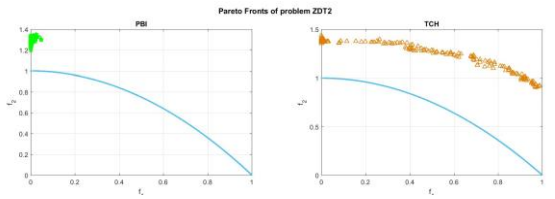
Şekil 8. DTLZ6 probleminin Pareto Front değerleri



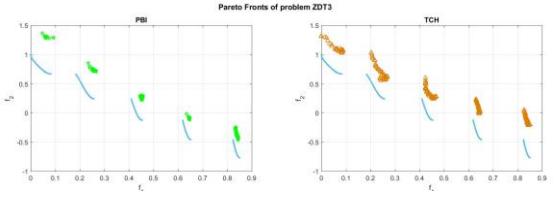
Şekil 9. DTLZ7 probleminin Pareto Front değerleri



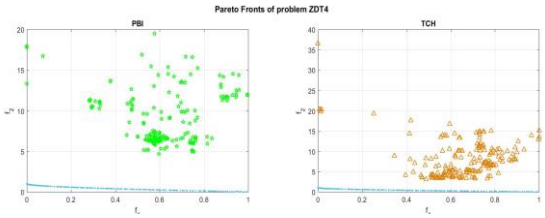
Şekil 10. ZDT1 probleminin Pareto Front değerleri



Şekil 11. ZDT2 probleminin Pareto Front değerleri



Şekil 12. ZDT3 probleminin Pareto Front değerleri



Şekil 13. ZDT4 probleminin Pareto Front değerleri

Şekil 3' de görüldüğü gibi, DTLZ1 problemde TCH çok daha düzgün şekilde gerçek pareto frontları(Gerçek pareto front çizgisi turkuaz renkteki çizimlerdir) bulmakta iken Şekil 5'te PBI yöntemi daha iyi sonuç vermektedir. Bir çok şeklin birbirine benzemesinin sebebi, temelde MOEA/D algoritmasının kullanılması ve sadece ayrıştırma yöntemlerinin farklı olmasından dolayıdır. Bazı problemlerde her iki yöntemde mükemmel sonuçlar vermektedir Mükemmel şekilde pareto frontları çıkan algoritmalar şöyledir: DTLZ2, DTLZ3, DTLZ4, DTLZ5.

MOEA/D algoritması her iki yöntemle de ZDT1,ZDT2 ve ZDT3 problemlerinde, belirli bir yakınsama yapmasına rağmen gerçek pareto frontları bulamamıştır. Bunun sebebi algoritmanın hala geliştirilmeye ihtiyacı olmasından dolayıdır. Tamda bu sebepten dolayı bu alanda birçok çalışma yapılmaktadır.

IV. TARTIŞMA

Bu çalışmada MOEA/D algoritması incelendi. Önce algoritmanın nasıl çalıştığı, hangi parametrelerden oluştuğu ve parametre değerlerinden oluştuğu sunuldu. Daha sonra MOEA/D'ın beyni sayılacak iki ayrıştırma yöntemi tanıtıldı. Bu iki yöntem aynı artlar altında testlere tabi tutuldu ve sonuçta hangi yöntemin 11 adet bencmark testinde başarılı olduğu gözlendi. Bu çalışma MOEA/D üzerine yapılacak çalışmalarda problemin tipine göre hangi ayrıştırma yönteminin kullanılacağına dair bir ön çalışma oldu.

Gelecekte ise MOEA/D'ın diğer önemli parametreleri olan komşuluk sayısının algoritmanın arama kabiliyetine etkisi olup olmadığı incelenilebilir. Böylece MOEA/D üzerinde çalışacak araştırmacılara bir yol gösterebilir.

[1] A. OZKIS, "Girdap arama ve yapay alg algoritmalarının çok amaçli optimzasyon problemlerine uyarlanmasi," phD, Computer Engineering, Selcuk Universty, KONYA, 2018.

[2] A. Özkış and A. Babalık, "A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm," *Information Sciences*, vol. 402, pp. 124-148, 2017.

[3] J. Schaffer, "Multiple objective optimization with vector evaluated," ph.D., Genetic Algorithms, Vanderbilt Universty, 1984.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182-197, 2002.

[5] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE*

Transactions on evolutionary computation, vol. 8, no. 3, pp. 256-279, 2004.

[6] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*, 2004, pp. 832-842: Springer.

[7] Q. F. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," (in English), *Ieee Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, Dec 2007.

[8] J. Bai and H. Liu, "Multi-objective artificial bee algorithm based on decomposition by PBI method," *Applied Intelligence*, vol. 45, no. 4, pp. 976-991, 2016.

[9] E. ERGUL, "Çok amaçlı genetik algoritmalar: Temelleri ve Uygulamaları," ph.D., Multi Objective Optimization, Ondokuz Mayıs University, SAMSUN, 2010.

[10] A. J. Nebro and J. J. Durillo, "A study of the parallelization of the multi-objective metaheuristic MOEA/D," in *International Conference on Learning and Intelligent Optimization*, 2010, pp. 303-317: Springer.

[11] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760-771, 2011.

[12] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173-195, 2000.

[13] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*: Springer, 2005, pp. 105-145.

[14] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, vol. 264, 2008.

[15] A. Babalik, A. Ozkis, S. A. Uymaz, and M. S. Kiran, "A multi-objective artificial algae algorithm," (in English), *Applied Soft Computing*, vol. 68, pp. 377-395, Jul 2018.

[16] Y. Xiang, Y. Zhou, and H. Liu, "An elitism based multi-objective artificial bee colony algorithm," *European Journal of Operational Research*, vol. 245, no. 1, pp. 168-193, 2015.

[17] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67-82, 1997.