

## Yinelemeli F-Yarış Algoritması ile Yapay Arı Koloni Algoritmasının Kontrol Parametrelerinin Ayarlanması

Ömür ŞAHİN<sup>\*1</sup>, Bahriye AKAY<sup>2</sup>

<sup>\*1</sup>Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, KAYSERİ  
<sup>1</sup>Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, KAYSERİ

(Alınış / Received: 23.10.2020, Kabul / Accepted: 02.12.2020, Online Yayınlanma / Published Online: 28.04.2021)

### Anahtar Kelimeler

Yapay Arı Koloni Algoritması,  
Yinelemeli F-Yarış,  
Kontrol Parametre Analizi,  
CEC'2015,  
F-Yarış

**Öz:** Meta-sezgisel algoritmaların performansını etkileyen en önemli faktörlerden biri kontrol parametrelerinin değerleridir. Doğru kontrol parametrelerinin belirlenmesi algoritmaların performansını önemli ölçüde artırmaktadır. Ancak pek çok durumda bu kontrol parametrelerinin belirlenmesi oldukça maliyetlidir. Bu nedenle kontrol parametrelerinin belirlenebilmesi için çeşitli yöntemler önerilmiştir ve Yinelemeli F-Yarış algoritması en etkin olan yöntemlerdendir. Bu çalışmada, Yinelemeli F-Yarış algoritmasının Yapay Arı Koloni algoritmasının kontrol parametrelerini belirlemedeki başarımı incelenmiştir. Yinelemeli F-Yarış algoritması ile belirlenen kontrol parametreleri kullanılarak elde edilen sonuçlar literatürde önerilen kontrol parametreleri ile elde edilen sonuçlarla kıyaslanmış ve istatistiksel analizler yapılmıştır. Yinelemeli F-Yarış algoritması kullanılarak, belli bir problem setinin tamamını kapsayacak şekilde kontrol parametre seti elde edilmiş ve CEC'2015 problem seti üzerinde literatürdeki kontrol parametrelerinin elde ettiği sonuçlar ile kıyaslanmıştır. Elde edilen sonuçlar, kontrol parametrelerinin Yinelemeli F-Yarış algoritması ile ayarlanmasının hesaplama maliyetini artırmadan algoritmaların performansını artırdığını göstermektedir.

## Parameter Tuning of Artificial Bee Colony Algorithm Using Iterative F-Race Algorithm

### Keywords

Artificial Bee Colony  
Algorithm,  
Iterative F-Race,  
Control Parameter Tuning,  
CEC'2015,  
F-Race

**Abstract:** One of the most important factors affecting the performance of meta-heuristic algorithms is control parameters. Determining the correct control parameters significantly increases the performance of the algorithms. However, most of the time, determining these control parameters is very expensive task. Therefore, various methods have been proposed to tune the control parameters and the Iterative F-Race algorithm is one of the most effective methods. In this study, the performance of the Iterative F-Race algorithm in determining the control parameters of the Artificial Bee Colony algorithm was examined. The results obtained using the control parameters determined by the Iterative F-Race algorithm were compared with the results obtained with the control parameters recommended in the literature and statistical analyzes were applied. Using the iterative F-Race algorithm, a control parameter set that covers a certain problem set has been obtained and compared with the results of the control parameters in the literature on the CEC'2015 problem set. The results show that tuning the control parameters with the Iterative F-Race algorithm improves the performance of the algorithms without increasing the computational cost.

\*İlgili Yazar: omur@erciyes.edu.tr

## 1. Giriş

Meta-sezgisel algoritmalar, genellikle doğal bir olayı modelleyen, yüksek seviye-problemlerden bağımsız küresel optimizasyon araçlarıdır. Bu algoritmalar, optimal çözümleri kısıtlı bir hesaplama maliyeti ile üretmeyi amaçlamaktadır. Özellikle, gürültülü veya sürekli olmayan problemlerde meta-sezgisel algoritmalar tercih edilmektedir [1]. Meta-sezgisel algoritmaların performansını etkileyen pek çok faktör bulunmaktadır. Bu faktörler arasında en önemlilerinden biri de kontrol parametreleridir [2]. Meta-sezgisel algoritmalar; maksimum değerlendirme sayısı, popülasyon sayısı gibi önceden tanımlı ve algoritmanın davranışını etkileyen kontrol parametrelerine sahiptir. Kontrol parametreleri algoritmaların performansını etkilediği için bu parametrelerin önceden ayarlanması gerekmektedir [3]. Kontrol parametrelerinin her probleme özel ayarlanması oldukça maliyetli olduğu için çoğu zaman genel olarak tanımlanmış kontrol parametreleri kullanılmaktadır [4]. Probleme özel kontrol parametreleri makul bir zaman diliminde ayarlanabilirse algoritmaların performansı, bu kontrol parametreleri ile artırılabilir.

Algoritmaların kontrol parametrelerinin belirlenmesi oldukça önemli olsa da pek çok durumda optimum kontrol parametre setinin belirlenmesi maliyetli olmaktadır. Bu yüzden kontrol parametre setinin belirlenmesi üzerine literatürde pek çok çalışma bulunmaktadır ve birçok teknik önerilmiştir. Bu teknikler, örnekleme metotları, model tabanlı metotlar, tarama yöntemi ve meta-sezgiseller olmak üzere toplam dört temel gruba ayrılmaktadır [5]. Örnekleme metodu tekniğinde, Latin-square [6] veya Taguchi Orthogonal Arrays [7] gibi teknikler kullanılarak olası parametre seti sayısının azaltılması hedeflenmiştir. Model tabanlı metotlarda bazı parametrelerin bir modeli çıkarılmakta ve bu model üzerinden yaklaşık parametreler tahmin edilmeye çalışılmaktadır. Bu model sayesinde bütün parametre setleri test edilmeden optimal parametre setinin oluşturulması sağlanabilmektedir. Tahminleme için regresyon metodunun kullanılması model tabanlı yaklaşımın en genel yöntemidir [8–10]. Tarama yöntemi, belirli bir parametre vektör setinden en iyi parametre setini minimum sayıda test ile belirlemeye yönelik bir yöntemdir. Etkileşimli analiz (Interactive Analysis) [11], Sıralama ve Seçim (Ranking and Selection) [12], Çoklu Karşılaştırma Metodu (Multiple Comparison Procedures) [13] ve Tamamen Sıralı Kayıtsız Alan Seçim Metodu (Fully sequential indifference-zone Selection Procedure) [14] tarama alanındaki dört temel yaklaşımdır [5]. Meta-sezgisel metotlarda ise en uygun parametre seti, bir meta-sezgisel algoritma kullanılarak belirlenir. F-Yarış metodu [15] tarama algoritmalarından biridir ve Hoeffding metodundan [16] esinlenerek ortaya atılmıştır. Bu algoritma Friedman iki yönlü sıralamalı varyans analizi yöntemini temel almaktadır. Yinelemeli F-Yarış metodu ise F-Yarış metodunun belirlenen iterasyon sayısı kadar tekrarlı uygulanması esasına dayalı çalışmaktadır.

Bal arılarının yiyecek arama davranışını temel alan bir meta-sezgisel olan Yapay Arı Koloni (YAK) algoritmasının [17–19] da arama sürecini belirleyici rolü olan çeşitli kontrol parametreleri bulunmaktadır. Bu çalışmanın amacı, kontrol parametresi belirleme yöntemlerinden olan yinelemeli F-Yarış algoritmasının YAK algoritmasının kontrol parametrelerini belirlemedeki etkisini incelemektir. Bu amaçla üç temel deney gerçekleştirilmiştir. İlk olarak, F-Yarış metodu ile probleme özgü kontrol parametrelerini belirleyip literatürde önerilen kontrol parametreleri ile kıyaslamalar yapıldı. Daha sonra bütün problemlerdeki sonuçlar ele alınarak belirlenen optimal kontrol parametre seti ile literatürde önerilen kontrol parametrelerinin kıyaslaması gerçekleştirildi. Son olarak bir başka problem setinde, bir önceki deneyde yinelemeli F-Yarış algoritması ile belirlenmiş olan kontrol parametre setiyle elde edilen sonuçlar ile literatürde önerilen kontrol parametre seti kullanılarak alınan sonuçlar kıyaslandı. Böylelikle belirlenen kontrol parametre setinin ne kadar genelleşebildiği incelendi.

Makalenin organizasyonu şu şekildedir: giriş bölümünün ardından Bölüm 2'de materyal ve metotlar verilmiştir. Bölüm 3'te bulgular anlatılmış ve son olarak da Bölüm 4 ile makalenin genel değerlendirilmesi yapılmış ve sonlandırılmıştır.

## 2. Materyal ve Metot

### 2.1. Yapay Arı Koloni Algoritması

Doğada zeki sürü davranışı gösteren canlıların modellenmesi optimizasyon alanına oldukça katkı sağlamaktadır. Bal arılarının hareketlerinin modellenmesi üzerine de birçok çalışma gerçekleştirilmiştir ve bu çalışmalar neticesinde arılar algoritması [20], sanal arı algoritması [21], arı koloni algoritması [22] ve yapay arı koloni algoritması [17] gibi modeller ortaya konmuştur. Bu modeller içerisinde basit yapısı ve yüksek başarımından dolayı YAK algoritması yaygın olarak kullanılmaktadır [23,24].

YAK algoritması arıların doğadaki yiyecek kaynağı arama, bulmuş oldukları iyi kaynakları hafızalarına alma ve diğer arılarla bu kaynak bilgilerini paylaşma davranışlarını modellemektedir. YAK algoritması, bir kolonideki arıların kaynak arama ve tüketme karakteristiklerini dikkate alarak, arıları, görevli, gözcü ve kâşif arı olmak üzere

üç gruba ayırmaktadır. Görevli arılar kaynak bulma ve buldukları kaynakları kovana aktarmaktan sorumlu arıları temsil etmektedir. Görevli arılar kovana döndüklerinde hafızalarındaki bilgiyi gözcü arılara aktarırlar. Gözcü arılar, görevli arılardan gelen bilgiye göre kaynak seçim ilgili kaynakta arama eylemini gerçekleştirirler. Eğer bir görevli arı seçmiş olduğu kaynağı tüketir ise henüz keşfedilmemiş bir kaynağı bulmak için kâşif arıya dönüşür. YAK algoritmasında, besin kaynakları bir problemin olası çözümleri, besin kaynaklarının nektar kalitesi ise çözümlerin uygunluk değeridir. YAK algoritmasının temel adımları Algoritma 1'de verilmiştir.

Adımlar
1: Başlangıç Besin Kaynaklarının Belirlenmesi
2: Değerlendirme
3: <b>tekrar</b>
4: Görevli Arı Fazı
5: Gözcü Arı Fazı
6: En İyi Çözümün Saklanması
7: Kâşif Arı Fazı
8: <b>Durdurma Kriteri Sağlanana Kadar</b>

**Algoritma 1.** YAK Algoritmasının Temel Adımları

### 2.1.1. Başlangıç Besin Kaynaklarının Belirlenmesi

YAK algoritmasında SN kaynak sayısı kadar başlangıç çözümü (yiyecek kaynağı) rastgele olarak belirlenmektedir. Bu kaynaklar D boyutlu bir problem için ilgili problem parametresine özgü  $[x^{\min}, x^{\max}]$  alt ve üst sınırları bulunmaktadır.  $x_j$ . besin kaynağının j. parametresi Denklem 1 ile hesaplanmaktadır. Burada  $\text{rand}(0,1)$ ,  $[0,1]$  aralığında uniform dağılımdan gelen rastgele bir sayıyı ifade etmektedir.

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

### 2.1.2. Görevli ve Gözcü Arıların Yeni Kaynaklara Gönderilmesi

Başlangıç yiyecek kaynaklarının belirlenmesinden sonra, görevli arı fazı çalışmaya başlamaktadır. YAK algoritmasında her bir yiyecek kaynağı, görevli arıya denk gelmektedir. Bu yüzden SN kadar yiyecek kaynağı bulunmaktadır. Görevli arılar yeni yiyecek kaynakları veya bu yiyecek kaynaklarının çevresinde komşu bulmakla görevlidir. Bu arama davranışı Denklem 2 ile tanımlanmıştır.

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

Burada  $v_{ij}$ ,  $v_i$ . aday yiyecek kaynağının j. parametresine denk gelmektedir. i, ilk yiyecek kaynağını, k ise seçilen ikinci yiyecek kaynağını göstermektedir. Burada j değeri  $\{1, 2, \dots, D\}$  değerleri arasından ve k değeri de  $\{1, 2, \dots, SN\}$  değerleri arasından rastgele seçilmektedir.  $\phi$  parametresi ise  $[-1, 1]$  aralığında seçilen rastgele bir reel sayıdır. YAK algoritmasında,  $x_i$  ile  $v_i$  arasındaki seçim ise ağırlıklı seleksiyon yöntemi ile belirlenmektedir. Minimizasyon problemlerinde çözümlerin uygunluk değeri Denklem 3 ile hesaplanmaktadır. Burada  $\text{obj}(x_i)$ , i. Nektarının çıkarılma maliyetini,  $\text{fit}(x_i)$  ise bu kaynağın uygunluk değerini vermektedir. Eğer  $\text{fit}(v_i)$  değeri  $\text{fit}(x_i)$  değerinden daha iyi olursa  $x_i$  yerine  $v_i$  kaynağı tercih edilecektir. Bu durumda  $v_i$  kaynağının kaç sefer geliştirme yapılmadığının tutulduğu değer olan trial değeri 0 olarak atanacaktır. Aksi takdirde  $x_i$  kaynağının trial değeri çözümde gelişme olmadığı için bir artırılacaktır.

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

Bütün görevli arılar aramalarını tamamladığında gözcü arılarla bilgi alışverişi aşaması başlamaktadır. Buradaki bilgi alışverişi aşaması Denklem 4'te verilen olasılık hesabı ile modellenmiştir. Bu denklem ile daha iyi kaynaklara gözcü arıların gitme ihtimali artmaktadır.

$$p_{ij} = \frac{\text{fit}(x_i)}{\sum_{j=1}^{SN} \text{fit}(x_j)} \quad (4)$$

### 2.1.3. Kâşif Arı Fazı

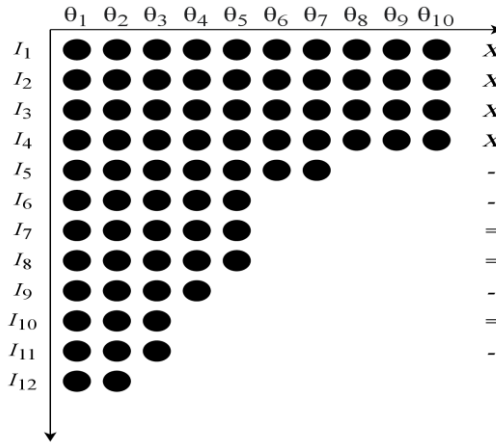
YAK algoritmasında iyileştirilemeyen çözüm kümesinin bırakılıp yerine yeni çözümlerin üretilmesi kâşif arı fazında gerçekleşmektedir. Görevli ve gözcü arı fazlarında tüketilen kaynakların iyileştirilememe sayacı olan trial

değeri bir kontrol parametresi olan limit değerini aştığı zaman bu kaynak bırakılmakta ve yerini rastgele yeni bir kaynak almaktadır.

## 2.2. Yinelemeli F-Yarış Algoritması

Yinelemeli F-Yarış algoritması [25], Friedman sıralamalı iki-yönlü varyans analizi temelli olan I/F-Yarış algoritmasının [26,27] yinelemeli olarak özelleştirilmiş varyantıdır. Yinelemeli F-Yarış algoritması üç temel aşamadan oluşmaktadır. İlk olarak belli bir dağılıma göre örnek parametreler oluşturulur. Daha sonra örnek parametre setleri yarışılarak en iyi parametre setleri ( $\theta^{elit}$ ) belirlenir. Son olarak, yeni örnekleneyecek parametre setlerinin dağılım fonksiyonu, en iyi parametre seti etrafında örnekleneyecek şekilde güncellenir. Bu üç aşama durdurma kriteri sağlanana kadar tekrarlanır [25].

Her bir parametrenin kendine özel dağılımı bulunmaktadır. Parametre nümerik ise normal dağılıma, ayrık ise kategorik parametreler için ayrık dağılıma sahiptir. Bu dağılımların güncellenmesi de dağılımların ortalama ve standart sapmalarının güncellenmesi ile gerçekleşmektedir. Dağılımların güncellenmesi ile o ana kadarki en iyi parametrelerin seçim şansı güçlendirilir. Yeni parametreler örneklendikten sonra en iyi ayarlamalar parametrelerin ortalamasının yarışı ile belirlenir. Bu yarış, aday parametre setleri arasında gerçekleşir. Örneğin Şekil 1'de 10 aday parametre seti  $\theta_i$  bulunmaktadır. Her bir düğüm bir ayarı temsil etmektedir. "X" karakteri test yapıldığını "=" istatistiksel farklılık olmadığı için herhangi bir ayarın çıkarılmadığını ve "-" karakteri de istatistiksel analizler sonucunda bir ayarın ayar setinden çıkarıldığını belirtmektedir. Her yarış aşamasında, aday parametrelerinin değerlendirilmesi tek sefer gerçekleştirilir. Bu aşamadan sonra istatistiksel analizler yapılır ve en az bir adet diğerlerinden istatistiksel olarak daha kötü performans gösteren parametre seti aday parametre setleri içerisinde çıkarılır. İlk eleme işleminde ( $T^{first}$ ) daha fazla parametre seti bulunurken daha sonra ( $T^{each}$ ) bu sayı azalmaktadır çünkü istatistiksel fark elde edebilmek için başta daha fazla değerlendirmeye ihtiyaç duyulmaktadır. Bu süreç daha öncesinden belirlenen minimum parametre setine ulaşına, maksimum örnekleme sayısına ulaşına kadar veya belirlenen hesaplama zamanına ulaşına kadar devam eder.



Şekil 1. Yarış örneği [25].

Yinelemeli F-Yarış algoritmasının temelleri Şekil 2'de verilmiştir. Yarışın sonunda elde edilen parametre setlerinin çevresinde dağılıma uygun olarak yeni parametre setleri belirlenir ve yarış tekrar başlatılır. Algoritmanın temel adımları Algoritma 2'de verilmiştir. Algoritmada verilen X parametresi arama uzayını, B değeri ise o iterasyondaki tanımlı maksimum değerlendirme sayısını göstermektedir. Yinelemeli yarışlar ilk olarak kaç iterasyon boyunca algoritmanın çalışacağı ( $N^{iter}$ ) hesaplanması ile başlamaktadır. Bu sayı parametre sayısına bağlı olarak tanımlanmaktadır ve Denklem 5 ile hesaplanmaktadır. Bu sayede daha geniş parametre uzayında daha fazla iterasyon çalışması sağlanır. Her bir iterasyonda, belirlenmiş bir  $B_j$  arama kaynağı ile sadece bir yarış gerçekleştirilir.  $B_j$  arama kaynağı Denklem 6 ile hesaplanmaktadır. Her bir yarış aday ayarlar kümesi olan  $\theta_j$  ile başlar. Aday ayar sayısı ise Denklem 7 ile hesaplanmaktadır. Aday ayar sayısı iterasyon geçtikçe azalmaktadır, yani sonraki iterasyonlarda her bir ayar için daha fazla değerlendirme yapılmaktadır.  $\mu$  parametresi ilk testin gerçekleşmesi için gerekli olan örneklem sayısını belirtmektedir [25].

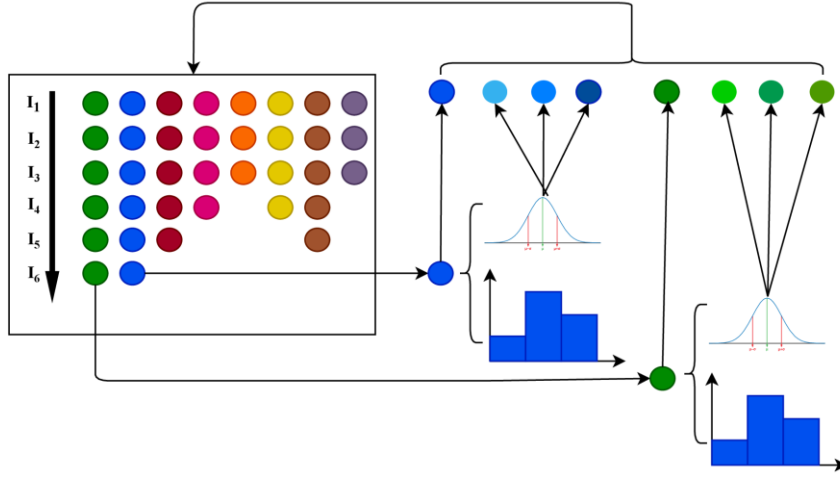
$$N^{iter} = \lfloor 2 + \log_2 N^{param} \rfloor \quad (5)$$

$$B_j = (B - B^{kullanilan}) / (N^{iter} - j + 1), j = 1, \dots, N^{iter} \quad (6)$$

Yarış sırasında hangi ayarın daha iyi olduğunu belirlemek için Friedman iki yönlü sıralamalı varyans analizi ve Post-Hoc testi olarak da Conover yöntemi [28] kullanılmaktadır. Her bir yarış ilgili iterasyona ait tanımlanan değerlendirme sayısı kalan ayarları değerlendirmek için yeterli olmayana kadar ( $B_j < B_j^{surv}$ ) veya minimum sayıda ayar kalana kadar ( $N_j < N_j^{surv}$ ) çalışmaya devam eder. Yarışın sonunda hayatta kalan ayarlara bir sıralama değeri ( $r_z$ ) atanır.  $N_j^{elit} = \min\{N_j^{surv}, N^{min}\}$  adet en düşük sıralamaya sahip ayar  $\theta^{elit}$  olarak seçilir. Sonraki iterasyonda  $N_j^{yeni} = N_j - N_j^{elit}$  kadar yeni aday üretilir. Yeni aday üretilirken  $\theta^{elit}$  içerisinde bir parametre seti sıralama değeri olan  $r_z$  değerine göre Denklem 8 ile seçilir. Ardından  $\theta_z$ 'nin her bir parametresinin dağılımına bağlı olarak  $X_d, d = 1, \dots, N^{param}$  parametreleri üretilir.

$$|\theta_j| = N_j = \lfloor B_j / (\mu + T^{each} \cdot \min\{5, j\}) \rfloor \quad (7)$$

$$p_z = \frac{N_{j-1}^{elit} - r_z + 1}{\frac{N_{j-1}^{elit}(N_{j-1}^{elit} + 1)}{2}} \quad (8)$$



Şekil 2. Yinelemeli F-Yarış algoritmasının genel şeması [25].

Adımlar
1: $\theta_i = \text{Ornekle}(X)$
2: $\theta^{elit} = \text{Yaris}(\theta_1, B_1)$
3: $j=1$
4: <b>tekrar</b>
5: $j = j+1$
6: $\theta^{yeni} = \text{Ornekle}(X, \theta^{elit})$
7: $\theta_j = \theta^{yeni} \cup \theta^{elit}$
8: $\theta^{elit} = \text{Yaris}(\theta_j, B_j)$
9: <b>Durdurma kriteri sağlanana kadar (<math>B^{kullanan} \leq B</math>)</b>

Algoritma 2. Yinelemeli F-Yarış algoritmasının temel adımları

Yeni kontrol parametre setleri oluşturulduktan sonra örneklendiği kontrol parametreleri ile birleştirilerek yarıştırlacak yeni kontrol parametreleri belirlenir. Son olarak, maksimum değerlendirme sayısına erişildiğinde veya değerlendirilecek ayar sayısı elit sayısından az olduğu durumda algoritma durur.

### 3. Bulgular

Bulgular iki farklı bölümden oluşmaktadır. İlk bölümde, literatürde bulunan çeşitli problem setleri üzerinde yinelemeli F-Yarış algoritması kullanılarak parametre setleri belirlenmiş ve bu kontrol parametreleri ile elde edilen sonuçlar, literatürde önerilen kontrol parametreleri ile elde edilen sonuçlarla kıyaslanmıştır. Daha sonra, aynı kontrol parametre seti üzerinde, bütün fonksiyon değerlerinin toplanarak elde edilmesi ile ortaya çıkan yeni bir problem yaratılmış ve genel bir kontrol parametre seti yinelemeli F-Yarış algoritması ile elde edilmiştir. Bu kontrol parametre seti ile elde edilen sonuçlar, literatürde önerilen kontrol parametre seti ile elde edilen sonuçlar ile kıyaslanmış ve değerlendirmesi gerçekleştirilmiştir. İkinci bölümde ise, ilk bölümde yinelemeli F-Yarış algoritması ile elde edilen genel kontrol parametre seti ile YAK algoritması bir başka problem seti olan CEC'2015

problem seti üzerinde çalıştırılmış ve literatürde önerilen kontrol parametrelerinin elde ettiği sonuçlarla kıyaslanmıştır.

### 3.1. Deney Düzenegi

Bu çalışmada yinelemeli F-Yarış algoritması için "irace" [25] isimli R paketi kullanılarak NP ve limit değerleri ayarlanmıştır. Algoritmanın kontrol parametreleri varsayılan değerler olan; deney sayısı 5000 ve elit sayısı 1 olarak seçilmiştir. Yarışlarda kullanılan Friedman testi için güven aralığı ise 0.05 olarak belirlenmiştir. Irace paketinin vermiş olduğu sonuçlar not edilmiş ve her bir algoritma, belirlenen kontrol parametreleri ile 50 sefer çalıştırılmıştır. Literatürde önerilen kontrol parametreleri [29] çalışmasından alınmıştır. Bütün deneylerde maksimum değerlendirme sayısı olarak 50000 tercih edilmiştir. Önerilen kontrol parametreleri olarak, popülasyon sayısı 50 ve limit değeri  $NP * Boyut / 2$  olarak alınmıştır. Elde edilen sonuçlarda ne kadar iyileştirme olduğu Denklem 9 kullanılarak hesaplanmıştır.

$$Gelistirme = \frac{Sonuc_{ortalama}^2 - Sonuc_{ortalama}^1}{|Sonuc_{ortalama}^2|} \quad (9)$$

Kontrol parametrelerin ne kadar çaba gerektirdiğini hesaplamak için de crafting effort değeri hesaplanmıştır. Crafting effort [30] bir problemin çözülebilmesi için ne kadar farklı kontrol parametresine ihtiyaç duyulduğunu belirten bir metriktir.  $cp_i: \forall \in \{1, 2, \dots, n\}$  olmak üzere n farklı kontrol parametresi bulunduğu  $\{v_{cp_i}^k\}_{k=1}^{n_{cp_i}}$  optimal kontrol parametre seti olmaktadır. Bu vektörlerdeki her bir değer;  $n_j: 0 \leq n_j \leq n_f$  sefer bu parametre setinde bulunmaktadır ve bu parametre sayılarının toplamı  $\sum_j n_j = n_f$  olmaktadır. Böylelikle bu kontrol parametresinin "crafting effort" değeri Denklem 10 ile hesaplanır. Bütün parametrelerin crafting error toplamı ise;  $\lambda = \sum_{i=1}^k v_{cp_i}$  formülü ile hesaplanmaktadır. Buradaki k parametre sayısını temsil etmektedir. Crafting effort,  $dinlik \sim \frac{1}{\lambda}$  ile ilişkilidir.

$$v_{cp_i} = - \sum_j \frac{n_j}{n_f} \cdot \log \frac{n_j}{n_f} \quad (10)$$

### 3.2. Problem Seti 1 Üzerinde Gerçekleştirilen Deneyler

Birinci deneysel çalışmada literatürde sıkça kullanılan problem setleri kullanılarak YAK algoritmasının NP ve limit kontrol parametreleri optimize edilmiştir ve bu parametre setlerinin crafting effort değerleri hesaplanmıştır. Yinelemeli F-Yarış algoritması ile elde edilen kontrol parametre setleri ve buna ait crafting effort hesaplamaları Tablo 1'de verilmiştir. Her bir kontrol parametresinin crafting effort değeri ve toplam crafting effort değerleri tablonun son iki satırında bulunmaktadır. Tabloda görüldüğü üzere, probleme özgü kontrol parametre seti için crafting effort değeri 7.0033 olarak hesaplanmıştır. NP kontrol parametresi Quartic ve FletcherPowell10 problemleri için aynı olduğundan Limit kontrol parametresine göre daha az efor gerektirmektedir.

Bütün fonksiyonlar dikkate alınarak NP ve limit değerleri yinelemeli F-Yarış algoritması ile optimize edilmiş ve NP=83, limit=981 olarak bulunmuştur. Bu kontrol parametre değerlerinin crafting effort değeri, kontrol parametrelerinin probleme göre değişmemesi sebebiyle 1'dir.

Probleme özgü kontrol parametreleri ve literatürde önerilen kontrol parametreleri ile alınan sonuçların kıyaslaması Tablo 2'de, bütün problemler dikkate alınarak elde edilen kontrol parametre seti ile literatürde önerilen kontrol parametre seti kullanılarak elde edilen sonuçların kıyaslaması Tablo 3'te verilmiştir. Her iki tabloda, ilk satırda ortalama  $\pm$  standart sapma ve altında parantez içerisinde ortanca değerleri verilmiştir. Geliştirme metriği Denklem 6 kullanılarak hesaplanmıştır.  $A_{12}$  etki büyüklüğü değeri ise son sütunda verilmiş olup Mann Whitney U testi sonucu  $p < 0.05$  ise kalın yazı tipi ile belirtilmiştir. Tablo 2'de belirtilen sonuçlara göre, Rosenbrock, Colville, Kowalik, FletcherPowell5 ve FletcherPowell10 problemlerinde probleme özgü optimize edilen kontrol parametreleri istatistiksel olarak daha iyi sonuçlar elde etmiştir. Literatürde önerilen kontrol parametreleri ise Zakharov ve Powell problemlerinde istatistiksel olarak daha başarılı görünmektedir. Diğer problemlerde istatistiksel bir fark bulunmama birlikte,  $A_{12}$  etki büyüklüğü incelendiğinde Quartic probleminde oldukça küçük bir farkla literatürdeki önerilen değerler daha iyi sonuç elde ederken, ortalama ve ortanca değerlerine göre ise probleme özgü optimize edilen kontrol parametreleri ile alınan sonuçlar daha iyidir. Perm, PowerSum ve Langermann10 problemlerinde de etki büyüklüğü değerine göre probleme özgü belirlenen kontrol parametre değerleri ile daha iyi sonuçlar elde edilmiştir.

**Tablo 1.** Test problemlerine özel optimize edilen kontrol parametre değerleri.

Problem	NP	Limit
Rosenbrock	88	1966
Quartic	98	785
Zakharov	84	1463
Colville	92	2355
Powell	67	1901
Kowalik	93	1056
Perm	80	377
PowerSum	90	2200
Langerman5	47	173
Langerman10	66	232
FletcherPowell5	96	1489
FletcherPowell10	98	656
<b>Crafting Effort</b>	<b>3.4183</b>	<b>3.5850</b>
<b>TOPLAM</b>	<b>7.0033</b>	

**Tablo 2.** Probleme özgü kontrol parametreleri ve literatürde önerilen kontrol parametreleri ile alınan sonuçların kıyaslaması.

Problem	Problem Bazlı Ayarlama	Literatürdeki K.P. [29]	Geliştirme	A <sub>12</sub>
Rosenbrock	0.0258 ± 0.0417 (0.0096)	0.1770 ± 0.3357 (0.0529)	0.8539	<b>0.2624</b>
Quartic	0.0301 ± 0.0066 (0.0298)	0.0304 ± 0.0078 (0.0301)	0.0077	0.5004
Zakharov	0.0031 ± 0.0024 (0.0023)	0.0002 ± 0.0002 (0.0002)	-13.2577	<b>0.9884</b>
Colville	0.0093 ± 0.0194 (0.0038)	0.1122 ± 0.0735 (0.0985)	0.9173	<b>0.0284</b>
Powell	0.0006 ± 0.0001 (0.0006)	0.0004 ± 0.0000 (0.0004)	-0.5915	<b>0.9988</b>
Kowalik	0.0004 ± 0.0001 (0.0004)	0.0004 ± 0.0001 (0.0004)	0.0431	<b>0.3772</b>
Perm	0.0008 ± 0.0009 (0.0004)	0.0006 ± 0.0004 (0.0005)	-0.3109	0.4592
PowerSum	0.0037 ± 0.0030 (0.0032)	0.0049 ± 0.0051 (0.0039)	0.2418	0.4536
FletcherPowell5	0.0001 ± 0.0004 (0.0000)	0.0590 ± 0.0608 (0.0425)	0.9986	<b>0.0556</b>
FletcherPowell10	5.0350 ± 4.6252 (2.8151)	10.2875 ± 6.6014 (9.0189)	0.5106	<b>0.2420</b>
Langermann5	-1.5000 ± 0.0000 (-1.5000)	-1.5000 ± 0.0000 (-1.5000)	0.0000	0.5000
Langermann10	-0.8825 ± 0.2125 (-0.7977)	-0.8473 ± 0.1690 (-0.7977)	0.0416	0.4828

Tablo 3'teki sonuçlara göre; Rosenbrock, Quartic, Colville, FletcherPowell5 ve FletcherPowell10 problemlerinde genel olarak optimize edilmiş kontrol parametre seti istatistiksel olarak daha iyi sonuç elde etmiştir. Literatürde önerilen kontrol parametreleri ile elde edilen sonuçlar ise Zakharov ve Powell problemlerinde istatistiksel olarak daha iyidir. Ayrıca bu parametre setleri A<sub>12</sub> değerine göre Langermann5 ve Langermann10 problemlerinde de az

farkla daha iyi sonuçlar elde ederken, geri kalan problemlerde genel olarak optimize edilmiş kontrol parametre seti daha başarılı olmuştur.

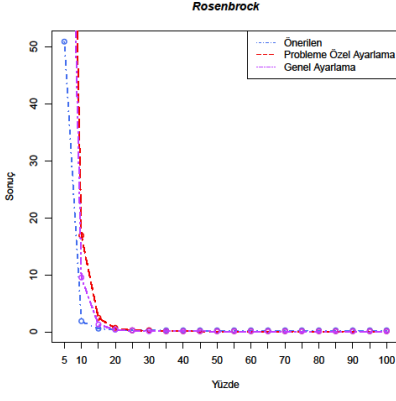
**Tablo 3.** Bütün problemler ele alınarak belirlenen kontrol parametreleri ile literatürde önerilen kontrol parametreleri kullanılarak elde edilen sonuçların kıyaslaması.

Problem	Genel Optimize Edilmiş K.P.	Literatürdeki K.P. [29]	Geliştirme	A <sub>12</sub>
Rosenbrock	0.0602 ± 0.1026 (0.0141)	0.1770 ± 0.3357 (0.0529)	0.6599	<b>0.3448</b>
Quartic	0.0250 ± 0.0054 (0.0253)	0.0304 ± 0.0078 (0.0301)	0.1765	<b>0.3000</b>
Zakharov	0.0025 ± 0.0017 (0.0023)	0.0002 ± 0.0002 (0.0002)	-10.4907	<b>0.9868</b>
Colville	0.0112 ± 0.0120 (0.0081)	0.1122 ± 0.0735 (0.0985)	0.9005	<b>0.0264</b>
Powell	0.0008 ± 0.0001 (0.0008)	0.0004 ± 0.0000 (0.0004)	-1.2303	<b>1.0000</b>
Kowalik	0.0004 ± 0.0001 (0.0004)	0.0004 ± 0.0001 (0.0004)	-0.0139	0.4940
Perm	0.0007 ± 0.0007 (0.0005)	0.0006 ± 0.0004 (0.0005)	-0.1917	0.4876
PowerSum	0.0036 ± 0.0029 (0.0025)	0.0049 ± 0.0051 (0.0039)	0.2729	0.4408
FletcherPowell5	0.0002 ± 0.0007 (0.0000)	0.0590 ± 0.0608 (0.0425)	0.9958	<b>0.0608</b>
FletcherPowell10	5.1449 ± 5.2908 (3.6045)	10.2875 ± 6.6014 (9.0189)	0.4999	<b>0.2444</b>
Langermann5	-1.4893 ± 0.0757 (-1.5000)	-1.5000 ± 0.0000 (-1.5000)	-0.0071	0.5100
Langermann10	-0.8515 ± 0.1988 (-0.7977)	-0.8473 ± 0.1690 (-0.7977)	0.0050	0.5412

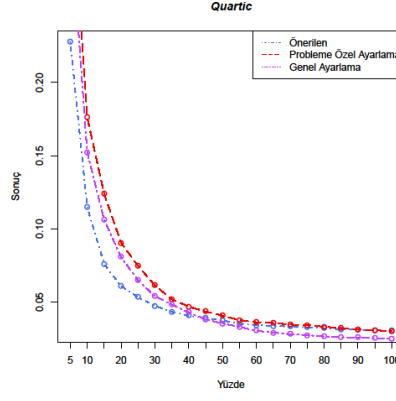
Kontrol parametre setlerinin yakınsama hızları Şekil 3'te verilmiştir. Literatürde önerilen kontrol parametre seti, probleme özgü optimize edilmiş kontrol parametre seti ve genel optimize edilmiş kontrol parametre setlerinin karşılaştırılması yapılmıştır. Şekil 3a incelendiğinde; literatürdeki kontrol parametre setinin en hızlı yakınsamayı gerçekleştirdiği görülmektedir. Genel optimize edilmiş kontrol parametre seti ise probleme özgü kontrol parametre setinden biraz daha hızlı yakınsamıştır. Probleme özgü kontrol parametre seti içerisinde en geç yakınsayan olsa da en iyi sonucu elde eden parametre seti olmuştur. Şekil 3b'deki sonuçlara göre; Rosenbrock probleminde olduğu gibi literatürdeki kontrol parametre seti en hızlı yakınsayan parametre seti olmuştur. Hemen ardından gelen genel optimize edilmiş kontrol parametre seti en iyi sonucu elde eden kontrol parametre seti olurken, probleme özgü kontrol parametre seti genel parametre setinin gerisinde kalmıştır. Zakharov probleminde (Şekil 3c) en hızlı yakınsayan kontrol parametre seti önceki iki problemde olduğu gibi literatürdeki kontrol parametre seti olmuştur. Probleme özgü kontrol parametre seti ve genel optimize edilmiş kontrol parametre setleri ise benzer davranışlar göstermiştir. Literatürde önerilen kontrol parametre seti bu problemde diğer iki kontrol parametre setine göre üstünlük kurmuştur. Colville problemi (Şekil 3d) incelendiğinde literatürdeki önerilen kontrol parametrelerinin oldukça başarısız olduğu görülmektedir. Optimize edilmiş kontrol parametrelerine göre hem geç yakınsamış hem de daha kötü sonuç elde etmiştir. Problem bazlı ayarlama ile elde edilen sonuçlar ise genel olarak belirlenen kontrol parametreleri ile elde edilen sonuçtan az farkla daha iyidir. Şekil 3e'deki Powell problemine ait sonuçlara göre üç kontrol parametre seti de benzer davranışlar sergilemiş olsalar da literatürde önerilen kontrol parametresinin diğer kontrol parametre setlerinden daha hızlı yakınsadığı ve daha iyi sonuç aldığı görülmektedir. Kowalik probleminde (Şekil 3f) üç kontrol parametre seti de benzer şekilde başlamışlardır fakat probleme özgü kontrol parametre seti diğer setlerden daha iyi sonuç elde etmiştir. Genel olarak ayarlanan kontrol parametre seti ise önerilen kontrol parametre setinden daha erken yakınsasa da literatürdeki kontrol parametre seti ile alınan sonuçlar az farkla daha iyidir. Şekil 3g'de verilen Perm problemine göre literatürdeki kontrol parametreleri ile elde edilen sonuçlar daha geç yakınsasa da diğer kontrol parametre setlerinden daha iyi sonuç almıştır. Probleme özgü kontrol parametre seti ve genel olarak ayarlanan kontrol parametre setleri ile elde edilen sonuçlar birbirine benzemektedir. PowerSum probleminde (Şekil 3h) literatürdeki kontrol parametre seti en geç yakınsayan ve en kötü sonucu elde eden kontrol parametre setidir.



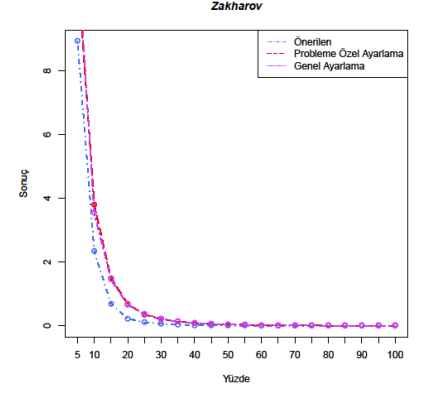
Ayarlanmış kontrol parametre setleri birbirine benzer davranış göstermiş ve daha iyi sonuç elde etmiştir. Şekil 3i'de verilen FletcherPowell5 probleminde literatürdeki kontrol parametreleri oldukça geç yakınsamış ve kötü sonuç elde etmiştir. Probleme özgü kontrol parametre setinin bu problemde daha iyi olduğu görülmektedir.



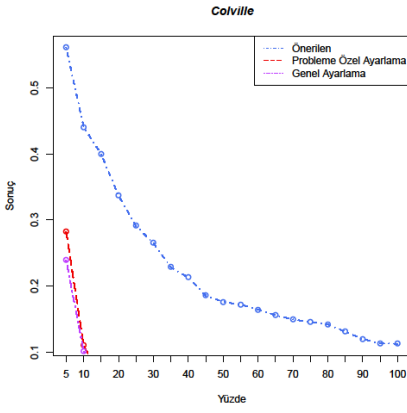
(a) Rosenbrock



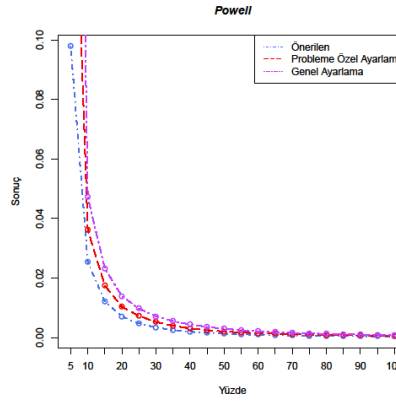
(b) Quartic



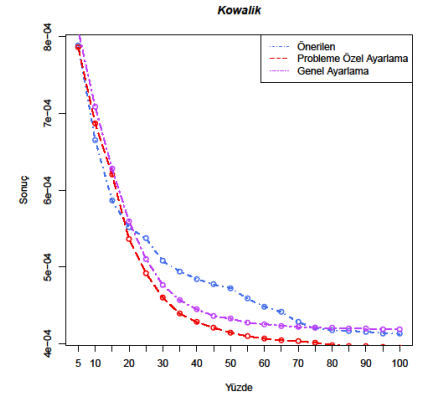
(c) Zakharov



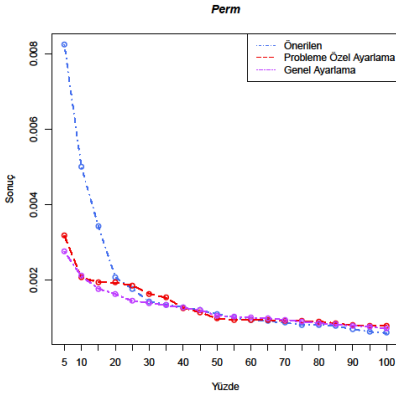
(d) Colville



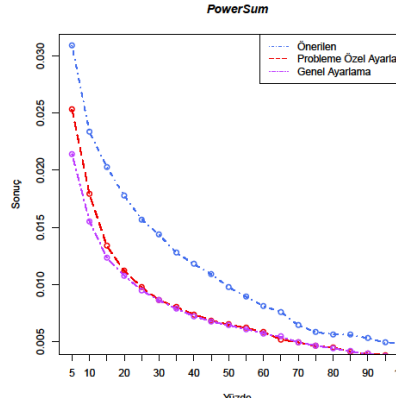
(e) Powell



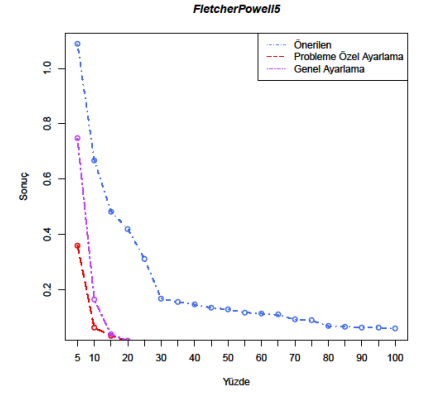
(f) Kowalik



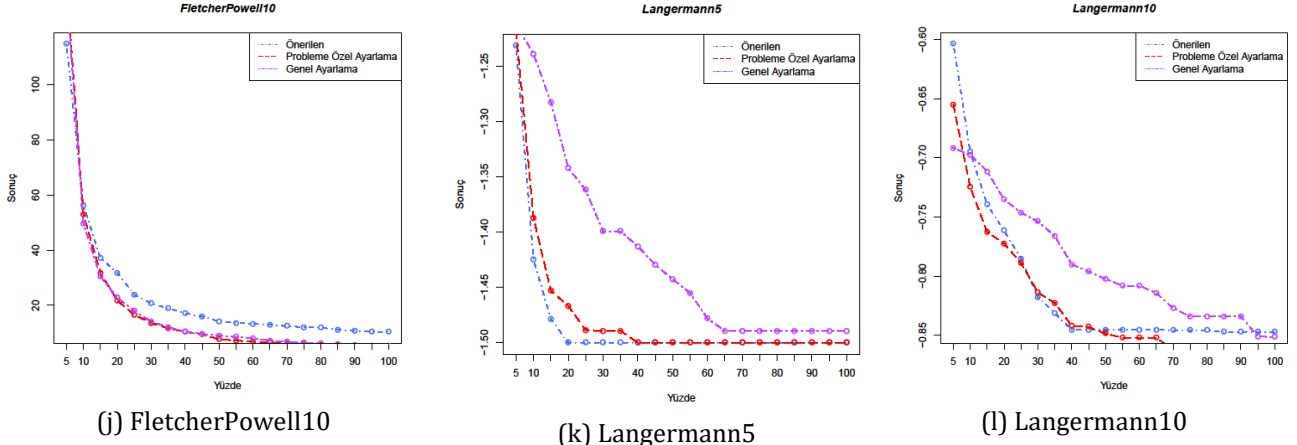
(g) Perm



(h) PowerSum



(i) FletcherPowell5



Şekil 3. Arama süreci boyunca harcanan değerlendirme sayısının çeşitli yüzdelerinde elde edilen sonuçlar.

Daha yüksek boyutlu olan FletcherPowell10 probleminde (Şekil 3j) de FletcherPowell5 problemine benzer davranış görülmektedir fakat elde edilen sonuçlar arasındaki fark azalmıştır. Langermann5 (Şekil 3k) probleminde literatürde önerilen kontrol parametre seti ile probleme özgü kontrol parametre setleri ile alınan sonuçlar birbirine benzerlik göstermektedir ancak genel olarak optimize edilen kontrol parametre seti ile alınan sonuç diğer kontrol parametre seti ile alınan sonucun gerisinde kalmıştır. Ve son olarak Şekil 3l'de verilen Langermann10 problemi incelendiğinde, en iyi sonuç probleme özgü kontrol parametre seti ile elde edilmiştir. Genel olarak ayarlanan kontrol parametre seti ise geç yakınsa da literatürdeki kontrol parametre seti ile gelirken sonucu az farkla geride bırakmıştır.

Tablo 4'te üç farklı kontrol parametre setinin sıralama değerleri verilmiştir. Sıralamalar hesaplanırken ortanca değerleri esas alınmıştır. Eğer ortanca değerleri eşit ise ortalama değerine daha sonra da standart sapma değerine göre sıralama değerleri belirlenmiştir. Bu değerlere göre en iyi performansı problem bazlı kontrol parametre seti elde etmiştir. Ardından genel olarak optimize edilmiş kontrol parametre seti elde ederken, son sırada literatürdeki kontrol parametre seti bulunmaktadır.

Tablo 4. Üç farklı kontrol parametre setinin Sıralama değerleri.

	Problem Bazlı	Genel	Literatür
Rosenbrock	1	2	3
Quartic	2	1	3
Zakharov	3	2	1
Colville	1	2	3
Powell	2	3	1
Kowalik	1	1	1
Perm	1	3	2
PowerSum	2	1	3
FletcherPowell5	1	2	3
FletcherPowell10	1	2	3
Langermann5	1	2	1
Langermann10	1	2	3
<b>TOTAL</b>	<b>17</b>	<b>23</b>	<b>28</b>

### 3.3. Problem Seti 2 Üzerinde Gerçekleştirilen Deneyler

Bir önceki bölümde elde edilen genel olarak optimize edilmiş kontrol parametre setinin etkinliğini incelemek için bu kontrol parametrelerinin bir daha zor ve kompleks problemleri içeren benchmark seti olan CEC'2015 problemleri üzerinde elde ettiği sonuçlar irdelenmiştir. Elde edilen sonuçlar Tablo 5'te verilmiştir. Bu sonuçlara göre FN15 dışında hiçbir problemde anlamlı farklılık elde edilememiştir. FN4 dışında diğer problemlerde ise iyileşmeler görülmektedir fakat bu iyileşmelerin miktarı kısıtlıdır.

Şekil 4'te yakınsama grafikleri verilmiştir. Bu grafiklere göre, FN1 ve FN4 problemleri hariç genel optimize edilmiş kontrol parametre seti literatürdeki kontrol parametre setinden daha hızlı yakınsamıştır ancak her ikisi de benzer sonuçlar elde etmiştir. FN1 ve FN4'te ise yakınsama davranışları birbirine benzemektedir. FN1 probleminde az farkla genel olarak optimize edilmiş kontrol parametre seti daha başarılı olurken, FN4 probleminde literatürdeki kontrol parametre seti daha başarılı olmuştur.

#### 4. Tartışma ve Sonuç

Çalışmada YAK algoritmasının kontrol parametreleri olan popülasyon büyüklüğü ve limit değerleri tarama yöntemlerinden biri olan yinelemeli F-Yarış algoritması ile optimize edilmiş ve etkisi incelenmiştir. Çalışma iki temel deneyden oluşmaktadır. İlk kısımda, probleme özgü kontrol parametre setleri ile bütün problemlerin ortaklaşa ele alınarak belirlendiği genel kontrol parametre seti Yinelemeli F-yarış algoritması kullanılarak elde edilmiş ve literatürde önerilen parametre seti ile farklılıkları incelenmiştir. İkinci kısımda ise bir önceki bölümde genel olarak optimize edilmiş kontrol parametre seti ile literatürde önerilen kontrol parametre seti, CEC'2015 problemleri kullanılarak çalıştırılmış ve elde ettiği sonuçlar kıyaslanmıştır. Bu sonuçlar üzerinden kontrol parametrelerinin genellenebilme kabiliyeti incelenmiştir.

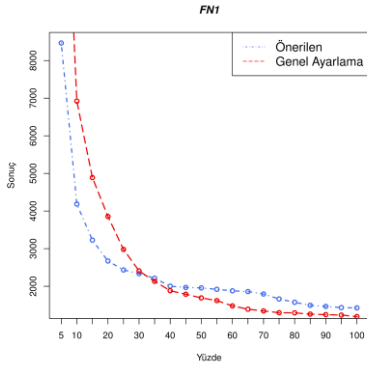
**Tablo 5.** Genel olarak optimize edilmiş kontrol parametre seti ile literatürde önerilen kontrol parametre setlerinin CEC'2015 fonksiyonları üzerindeki sonuçlarının kıyaslanması

Problem	Genel Optimize Edilmiş K.P.	Literatürdeki K.P. [29]	Geliştirme	A <sub>12</sub>
FN1	1194.9528 ± 737.3824 (1071.0575)	1425.9183 ± 1243.8733 (1120.6085)	0.1620	0.4948
FN2	11020.5064 ± 2903.2085 (11278.6100)	11833.6933 ± 3293.5872 (11060.7000)	0.0687	0.4552
FN3	304.2354 ± 0.9147 (304.4805)	304.5264 ± 0.8313 (304.5131)	0.0010	0.4288
FN4	400.0025 ± 0.0124 (400.0000)	400.0000 ± 0.0000 (400.0000)	-0.0000	0.5200
FN5	500.1960 ± 0.0593 (500.1927)	500.2183 ± 0.0638 (500.2188)	0.0000	0.3952
FN6	600.1252 ± 0.0215 (600.1282)	600.1277 ± 0.0210 (600.1279)	0.0000	0.4606
FN7	700.1335 ± 0.0319 (700.1336)	700.1431 ± 0.0331 (700.1403)	0.0000	0.4072
FN8	800.8779 ± 0.2478 (800.9258)	800.9776 ± 0.2822 (800.9348)	0.0001	0.4070
FN9	902.7599 ± 0.2323 (902.7804)	902.8258 ± 0.2790 (902.8952)	0.0001	0.3944
FN10	6166.6846 ± 3791.7076 (4823.1175)	8090.5559 ± 6619.4763 (5774.1690)	0.2378	0.4160
FN11	1103.1549 ± 0.5357 (1103.3235)	1103.3111 ± 0.5602 (1103.3645)	0.0001	0.4248
FN12	1229.0436 ± 3.5936 (1228.6460)	1229.7732 ± 4.4230 (1229.2565)	0.0006	0.4512
FN13	1548.5530 ± 73.2432 (1564.4990)	1584.5370 ± 49.3162 (1614.9400)	0.0227	0.4866
FN14	1592.4192 ± 3.5919 (1592.5965)	1593.2318 ± 3.4287 (1593.7370)	0.0005	0.4316
FN15	1512.4200 ± 2.8251 (1511.8260)	1513.4846 ± 2.8536 (1513.2615)	0.0007	<b>0.3738</b>

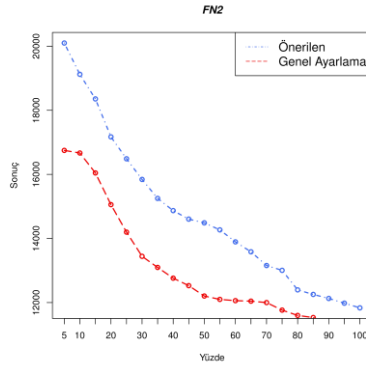
İlk problem setinde elde edilen sonuçlar incelendiğinde, probleme özgü kontrol parametre setlerinin her zaman en iyi sonuca ulaşamadığı görülse de genellikle başarılı sonuçlar elde etmiştir. Probleme özgü kontrol parametre setinin literatürde önerilen değerlerden daha az başarılı olduğu durumlarda yinelemeli F-Yarış algoritmasının en iyi değerleri test etmediği görülmektedir. Bunun temel sebebi de bu tür algoritmaların kısıtlı örneklem sayısı ile en iyiyi bulmaya çalışmasıdır. Daha fazla değerlendirme sayısı ile optimize sonuçların test edilip doğru sonucun elde edilmesi mümkün olabilir. Ancak bu tür algoritmaların kullanılma sebebi de kısıtlı sürede en iyi sonucu arama

çabasıdır. Bazı problemlerde kontrol parametrelerinin belirlenmesi oldukça zaman alan bir eylem olabilir. Bu yüzden bazı durumlarda bu tür yaklaşımlar ile uygun bir kontrol parametresinin bulunması zorunluluk haline gelmektedir. İkinci problem seti üzerinde gerçekleştirilen deneylerde ise, problemlerin biri hariç tamamında genel optimize edilmiş kontrol parametrelerinin iyileşmeler sağladığı görülmektedir. Fakat bunlardan yalnızca birinde istatistiksel olarak daha iyi sonuç elde edilmiştir.

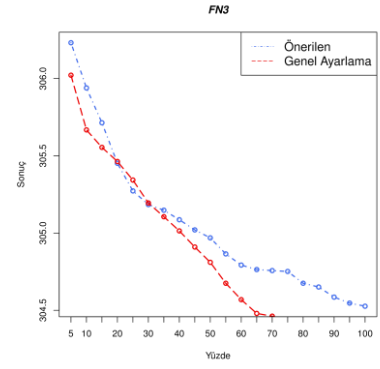
Sonuç olarak, kontrol parametrelerinin optimizasyonu ile daha iyi sonuçlar alınabildiği açıkça görülmektedir. Genel olarak belirlenen kontrol parametre seti belli bir problem seti üzerinde iyi sonuçlar elde edebilmekte ancak problemin içeriği değiştiği zaman etkisini büyük oranda kaybetmektedir. Bu yüzden kontrol parametre analizinin hesaplama zamanının oldukça fazla olduğu problemlerde, sistem kaynaklarının harcanmasının veya karmaşık analizlerin önüne geçmek adına kontrol parametre analizinin gerekliliğine araştırmacılar dikkatle karar vermelidir. Aynı zamanda literatüre kontrol parametresi içeren yeni yaklaşımlar öneren araştırmacıların da kontrol parametre analizlerini gerçekleştirip varsayılan değerleri belirlemeleri gerekmektedir. Bu algoritmaları kullanacak olan araştırmacıların varsayılan kontrol parametre setine sahip olması oldukça önemlidir. Bu tür çalışmalar ile araştırmacıların kontrol parametre analizi için harcamış oldukları kaynakların en aza indirilmesi sağlanabilmektedir. Gelecek çalışmalarda ise daha farklı kontrol parametre belirleme yöntemleri ile analizler gerçekleştirilip iyileşmeler incelenmelidir.



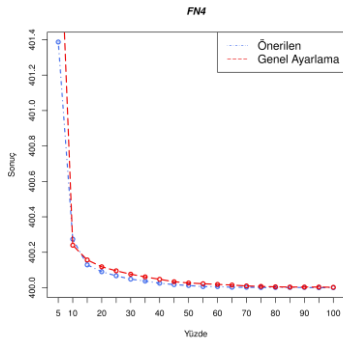
(a) FN1



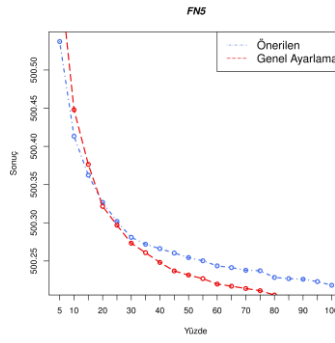
(b) FN2



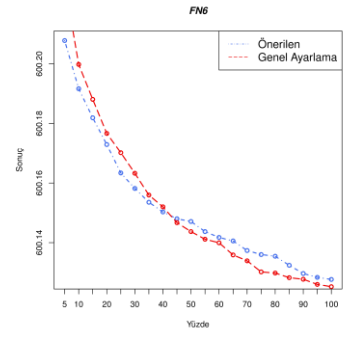
(c) FN3



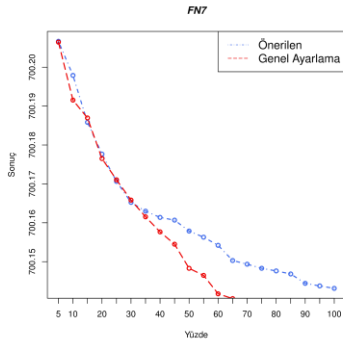
(d) FN4



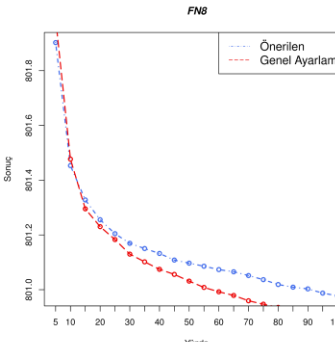
(e) FN5



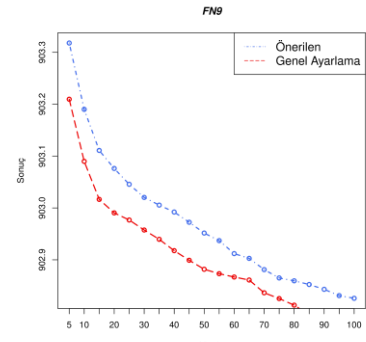
(f) FN6



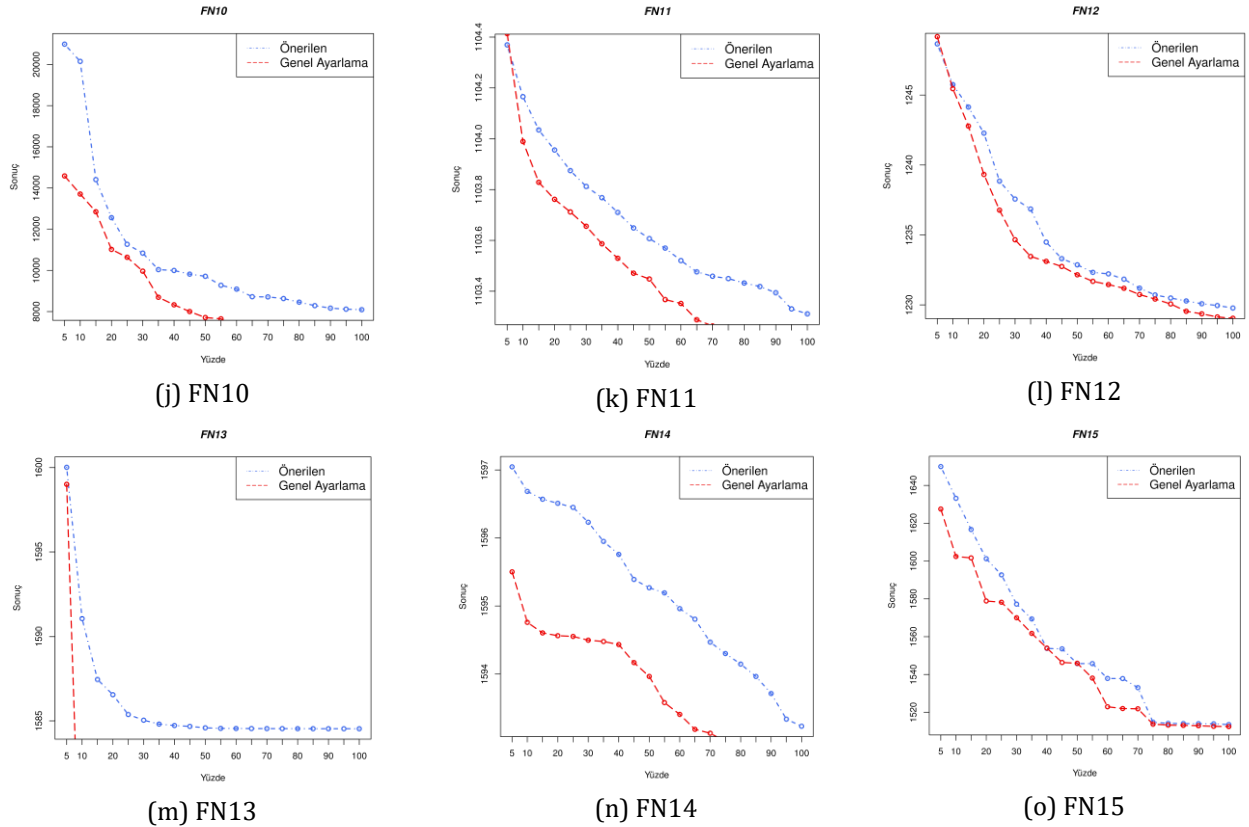
(g) FN7



(h) FN8



(i) FN9



Şekil 4. CEC'2015 fonksiyonları için, arama süreci boyunca harcanan değerlendirme sayısının çeşitli yüzdelerinde elde edilen sonuçlar

## Teşekkür

Bu araştırmada yer alan kısmi nümerik hesaplamalar TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nde (TRUBA kaynaklarında) gerçekleştirilmiştir.

## Kaynakça

- [1] O. Sahin, B. Akay, Comparisons of metaheuristic algorithms and fitness functions on software test data generation, *Applied Soft Computing*. 49 (2016) 1202–1214. doi:10.1016/j.asoc.2016.09.045.
- [2] W. Dillen, G. Lombaert, N. Voeten, M. Schevenels, Performance Assessment of Metaheuristic Algorithms for Structural Optimization Taking into Account the Influence of Control Parameters, içinde: *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, Springer International Publishing, 2019: ss. 93–101. doi:10.1007/978-3-319-97773-7\_9.
- [3] A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*. 3 (1999) 124–141. doi:10.1109/4235.771166.
- [4] K. De Jong, Parameter setting in EAs: A 30 year perspective, *Studies in Computational Intelligence*. 54 (2007) 1–18. doi:10.1007/978-3-540-69432-8\_1.
- [5] A.E. Eiben, S.K. Smit, Parameter tuning for configuring and analyzing evolutionary algorithms, *Swarm and Evolutionary Computation*. 1 (2011) 19–31. doi:10.1016/j.SWEVO.2011.02.001.
- [6] R. Myers, E.R. Hancock, Empirical modelling of genetic algorithms., *Evolutionary computation*. 9 (2001) 461–493. doi:10.1162/10636560152642878.
- [7] G. Taguchi, Y. Yokoyama, Taguchi methods: design of experiments, Amer Supplier Inst, 1993.
- [8] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, R. Gupta, Statistical exploratory analysis of genetic algorithms, *IEEE Transactions on Evolutionary Computation*. (2004). doi:10.1109/TEVC.2004.831262.
- [9] I.C.O. Ramos, M.C. Goldberg, E.G. Goldberg, A.D.D. Neto, Logistic regression for parameter tuning on an evolutionary algorithm, içinde: *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings*, 2005. doi:10.1109/cec.2005.1554808.
- [10] O. François, C. Lavergne, Design of evolutionary algorithms - A statistical perspective, *IEEE Transactions on Evolutionary Computation*. (2001). doi:10.1109/4235.918434.
- [11] B. Schmeiser, Simulation experiments, *Handbooks in Operations Research and Management Science*. 2 (1990) 295–330. doi:10.1016/S0927-0507(05)80171-9.
- [12] Y. Rinott, On two-stage selection procedures and related probability-inequalities, *Communications in*

- Statistics - Theory and Methods*. 7 (1978) 799–811. doi:10.1080/03610927808827671.
- [13] A.C.T. Y. Hochberg, Multiple Comparison Procedures, *Biometrical Journal*. (1987). doi:10.1002/bimj.4710310115.
- [14] S.H. Kim, B.L. Nelson, A Fully Sequential Procedure for Indifference-Zone Selection in Simulation, *ACM Transactions on Modeling and Computer Simulation*. 11 (2001) 251–273. doi:10.1145/502109.502111.
- [15] M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, A Racing Algorithm for Configuring Metaheuristics, *Proceedings of the Genetic and Evolutionary Computation Conference*. (2002) 11–18.
- [16] O.M. and A.W. Moore, The Racing Algorithm: Model Selection for Lazy Learners, *Artificial Intelligence Review*. 11 (1997) 193–225. doi:10.1023/a:1006556606079.
- [17] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Erciyes University, Engineering Faculty, 2005.
- [18] D. Karaboga, B. Basturk, A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *Journal of Global Optimization*. 39 (2007) 459–471. doi:10.1007/s10898-007-9149-x.
- [19] B. Basturk, An artificial bee colony (ABC) algorithm for numeric function optimization, *IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2006*. (2006). <https://ci.nii.ac.jp/naid/20001441290> (erişim 17 Ekim 2020).
- [20] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm - A Novel Tool for Complex Optimisation Problems, içinde: *Intelligent Production Machines and Systems - 2nd I\*PROMS Virtual International Conference 3-14 July 2006*, 2006. doi:10.1016/B978-008045157-2/50081-X.
- [21] X.S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, içinde: *Lecture Notes in Computer Science*, 2005. doi:10.1007/11499305\_33.
- [22] D. Teodorovic, M. Dell' Orco, Bee Colony Optimization-Cooperative Learning Approach to Complex Transportation Problems, *Advanced OR and AI Methods in Transportation*. (2005).
- [23] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: Artificial bee colony (ABC) algorithm and applications, *Artificial Intelligence Review*. 42 (2014) 21–57. doi:10.1007/s10462-012-9328-0.
- [24] C. Özturk, E. Hancer, D. Karaboga, Küresel en iyi yapay arı koloni algoritması ile otomatik kümeleme, *Journal of the Faculty of Engineering and Architecture of Gazi University*. (2014).
- [25] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, T. Stützle, The irace package: Iterated racing for automatic algorithm configuration, *Operations Research Perspectives*. 3 (2016) 43–58. doi:10.1016/j.orp.2016.09.002.
- [26] P. Balaprakash, M. Birattari, T. Stützle, Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement, içinde: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007. doi:10.1007/978-3-540-75514-2\_9.
- [27] M. Birattari, Z. Yuan, P. Balaprakash, T. Stützle, F-Race and Iterated F-Race: An Overview, içinde: *Experimental Methods for the Analysis of Optimization Algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010: ss. 311–336. doi:10.1007/978-3-642-02538-9\_13.
- [28] J. Haigh, W.J. Conover, Practical Nonparametric Statistics., *Journal of the Royal Statistical Society. Series A (General)*. (1981). doi:10.2307/2981807.
- [29] D. Karaboga, B. Akay, A comparative study of Artificial Bee Colony algorithm, *Applied Mathematics and Computation*. 214 (2009) 108–132. doi:10.1016/j.amc.2009.03.090.
- [30] K.V. Price, Differential evolution vs. the functions of the 2/sup nd/ ICEO, içinde: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, IEEE, 1997: ss. 153–157. doi:10.1109/ICEC.1997.592287.