

## Yerçekimsel Arama Algoritması ile PID Denetleç Parametrelerinin Tespiti

### PID Controller Parameters' Optimization Using Gravitational Search Algorithm

Nesibe Yalçın<sup>1</sup>, Semih Çakar<sup>2</sup>, Metin Kesler<sup>1</sup>, Nihan Kazak<sup>1</sup>

<sup>1</sup>Bilgisayar Mühendisliği Bölümü  
Bilecik Şeyh Edebalı Üniversitesi

nesibe.yalcin@bilecik.edu.tr, metin.kesler@bilecik.edu.tr, nihan.kazak@bilecik.edu.tr

<sup>2</sup>Fen Bilimleri Enstitüsü  
Bilecik Şeyh Edebalı Üniversitesi  
semihc@anadolu.edu.tr

#### Özet

Sezgisel optimizasyon algoritmaları, doğrusal olmayan sistemlerin optimizasyon problemlerinde oldukça sık kullanılmaktadır. Bu çalışmada, Doğru Akım (DA) motor denetimi için Oransal-İntegral-Türevsel (Proportional-Integral-Derivative, PID) denetleç parametrelerinin tespitinde sezgisel optimizasyon algoritmalarından Yerçekimsel Arama Algoritması (YAA) kullanılmıştır. Algoritma 100 defa koşturulmuş, elde edilen ortalama PID denetleç parametre değerleri [1]'de verilen diğer optimizasyon algoritmalarının (Tur Atan Karınca Kolonisi Algoritması (TACO), Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO) ve Diferansiyel Gelişim (DE)) 100 defa koşturulması sonucu elde edilen değerleri ile karşılaştırılmıştır, denetim başarımları ile kıyaslanmıştır. Sonuç olarak, YAA'nın birçok kategoride TACO ve GA'dan daha iyi, PSO ve DE algoritmalarından daha düşük başarımla sergilediği görülmüştür.

#### Abstract

Heuristic optimization algorithms are used widely in optimization problems of nonlinear systems. In this paper, Gravitational Search Algorithm (GSA), a heuristic optimization algorithm- is used to determine PID controller parameters for Direct Current (DC) motor control system. Other optimization algorithms (Touring Ant Colony Optimization (TACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution Algorithm (DE)) given in [1] have been separately run 100 times at maximum generation number. Obtained average PID controller parameters are evaluated and compared with GSA's results. As a result, in many categories the performance of GSA is better than the performances of TACO and GA, but the results show that the proposed GSA's performance is lower than the performances of PSO and DE.

#### 1. Giriş

Teknolojinin gelişimiyle birlikte özellikle endüstriyel alanda kullanılan denetim sistemlerinin önemi gittikçe artmaya başlamıştır. Bilgisayarlı denetim sistemlerinde, ulaşım ve güç sistemlerinde, robotik, savunma, oto montaj ve benzeri sektörlerde denetim sistemlerinin kullanım oranı oldukça fazladır. Denetim sistemlerinin karmaşık ve zor yapısı birtakım alternatif çözüm yollarının gelişimine olanak sağlamıştır. Bu çözüm yollarının başında optimizasyon gelmektedir. Optimizasyon kelime olarak bir problemi iyileştirme, daha iyi çözümü bulma anlamı taşımaktadır.

Bir problemi optimize ederken çeşitli algoritmalarından yararlanılır. Bu algoritmalar, belirli kurallara ve işleyişe göre tanımlanan, birtakım matematiksel ifadeler yardımıyla problemleri çözme yöntemleridir [2], yakınsama özelliğine sahiptir ve kesin çözümü değil kesin çözüme yakın en iyi çözümü garanti edebilmektedir [3, 4].

Özellikle çok değişkenli optimizasyon süreçlerinde, değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tip problemlerin klasik optimizasyon yöntemleri ile çözümü, hem problemin yapısına bağlı olarak modellemede, hem de çözüm sürecinde zorluklar içermektedir.

Yani bu tip çözüm yöntemleri daha ziyade problemlere özeldir ve problemin kesinlikle matematiksel fonksiyonlarla tanımlanma gerekliliği gibi birçok sakıncaya sahiptir. Bu güçlüklerin üstesinden gelebilmek için doğada var olan sistemleri ve olayları temel alarak oluşturulan sezgisel yöntemler geliştirilmiştir. Özellikle popülasyon temelli sezgisel çok boyutlu arama işlemleri sayesinde, optimizasyon probleminin çözümüne hızlı bir şekilde ulaşılabilmektedir. Bu sezgisel yaklaşımlardan Genetik Algoritma (GA), Karınca Kolonisi Algoritması (KKA), Parçacık Sürü Optimizasyonu (PSO), Diferansiyel Gelişim (Differential Evolution, DE) ve Yerçekimsel Arama Algoritması (YAA, GSA) en yaygın kullanılanlar arasında sayılabilir.

Bu çalışmada PID denetleç parametrelerinin optimizasyonu için YAA kullanılmış, elde edilen sonuçlar diğer optimizasyon yöntemlerinin kullanıldığı [1]' de verilen sonuçlar ile karşılaştırılmıştır.

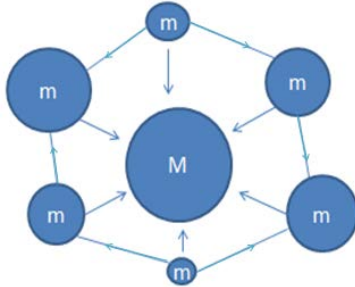
## 2. Yerçekimsel arama algoritması

Rashedi ve arkadaşları (2009) tarafından Newton'un hareket ve yerçekimi kanunlarından esinlenerek geliştirilmiş sezgisel optimizasyon algoritmasıdır. Arama uzayındaki her bir parçacık, Yerçekimsel Arama Algoritması (YAA)' nda bir kütle olarak kabul edilir, bu sebeple YAA' yı bir yapay kütle sistemi olarak tanımlamak mümkündür [5]. Arama uzayındaki tüm kütleler Newton'un evrensel çekim kanununa göre birbirlerini çekerler ve yerçekimi kuvveti ile birbirlerine kuvvet uygularlar. Yerçekimi kuvvetine maruz kalan kütleler arama uzayı içerisinde hareket ederek optimum sonuca ulaşmaya çalışırlar. Bu durum Şekil 1' de gösterilmiştir.



Şekil 1: Kütleler arası kuvvet.

Şekil 1' de G yerçekimi sabitini ifade etmektedir. Bu kuvvete maruz kalan kütleler arama uzayı içerisinde hareket ederek optimum çözüme ulaşırlar. Bu durum Şekil 2' de gösterilmiştir.



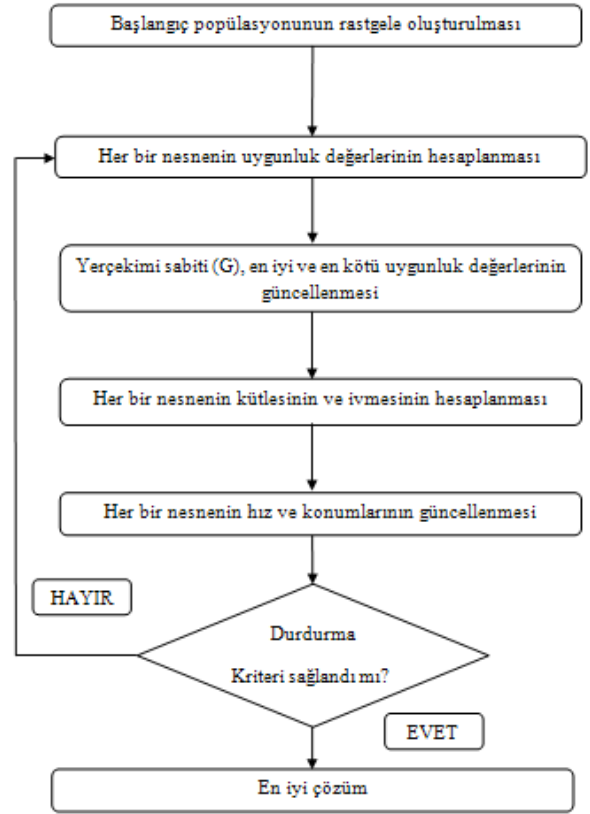
Şekil 2: Kütlelerin birbirleri ile etkileşimi.

Noktasal kütlelerin bu şekildeki hareketi Newton'un ikinci hareket kanununa uygun olarak gerçekleşir. Arama uzayındaki her bir kütle için; konumu, eylemsizlik kütlesi, aktif yerçekimsel kütlesi ve pasif yerçekimsel kütlesi olmak üzere dört özelliği vardır. Her kütle arama uzayında belli bir pozisyonda bulunur. Buldukları konum, çözülmesi beklenen problem için birer çözüm alternatifidir. Kütleler, üzerlerine herhangi bir kuvvet etki ettiğinde bu kuvvete karşı koymak isteyerek bir direnç gösterir. Bu direnç eylemsizlik kütlesi olarak adlandırılır [5, 6].

Eylemsizlik kütlesi fazla olan kütle daha yavaş hareket etmek isteyecek, az olanı ise daha hızlı hareket edecektir. Bu durumda oluşan yerçekimi alanı gücüne aktif yerçekimsel kütle denir. Yerçekimi alanı ile bir nesnenin iletişim gücüne pasif yerçekimsel kütle denir [6].

YAA' da her bir nesnenin kütle miktarı o nesnenin performansını gösterir. Algoritma süresince en ağır olan kütle diğer kütlelere göre daha yavaş hareket edecek ve diğerlerini kendine çekecek bir çekim kuvveti uygulayacaktır. Jenerasyon sayısı bitiminde ya da herhangi bir sonlandırma eylemi olduğunda kütlesi en fazla olan nesne, problemin optimum çözümünü oluşturmuş olacaktır.

Rashedi ve arkadaşlarının yaptıkları çalışmaya dayanarak YAA' nın akış şeması Şekil 3' te verilmiştir [5, 6].



Şekil 3: YAA' nın akış diyagramı.

Algoritmanın adımları:

- Başlangıç:** YAA' da kullanılan yerçekimi sabiti (G), maksimum jenerasyon sayısının, bir kütle için diğer kütlelere yapacağı etkiyi hesaplamak için kullanılan küçük bir sabitin ( $\epsilon$ ) ilk değerlerinin atıldığı kısımdır.
- Arama Uzayının Tanımlanması:** Sistemin kaç kütleli oluşacağı kısımdır. Kütleler arama uzayına rastgele yerleştirilir.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i=1,2,3,\dots,N \quad (1)$$

(1)' de tanımlanan  $x_i^d$ , i. kütle için d. boyuttaki konumunu gösterir.

**3. Yerçekimi Sabitinin Hesaplanması:** Başlangıç kısmında atanan sabit değerin her jenerasyonda azaltıldığı kısım. Zamanla azalan yerçekimi sabiti ile arama hızı kontrol edilmektedir. Yerçekimi sabitinin jenerasyon ile değişiminin hesaplanmasında (2) kullanılır.

$$G(t) = G_o x e^{(-\frac{\alpha k}{K})} \quad (2)$$

$G_o$  , yerçekimi sabitinin başlangıç değerini;  $\alpha$  , kullanıcının belirlediği sabit bir değeri;  $k$ , o anki jenerasyon değerini ve  $K$  maksimum jenerasyon sayısını göstermektedir.

**4. Uygunluk Fonksiyonu ile Uygunluk Değerlerinin Hesaplanması:** Hesaplanan uygunluk değerinin en iyi ve en kötülerini seçilir (best, worst).

Problem minimize yapılmak isteniyorsa;

$$\begin{aligned} \text{best}(k) &= \min (fit_j(k)) & (3) \\ \text{worst}(k) &= \max (fit_j(k)) & (4) \end{aligned}$$

Problem maksimize edilmek isteniyorsa;

$$\begin{aligned} \text{best}(k) &= \max (fit_j(k)) & (5) \\ \text{worst}(k) &= \min (fit_j(k)) & (6) \end{aligned}$$

(3), (4), (5) ve (6) denklemlerinde yer alan  $j = \{1, \dots, N\}$  dir.  $fit_j(t)$  j. kütle için k. jenerasyondaki uygunluk değeri, best(k) ve worst(k) ise sırasıyla k. jenerasyondaki en iyi ve en kötü çözümü ifade etmektedir.

**5. Kütle Hesabı:** Arama uzayında bulunan bir kütle için aktif yerçekimsel kütlesi, pasif yerçekimsel kütlesi ve eylemsizlik kütlesi birbirlerine eşit alınarak tüm kütleler hesaplanır.

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad i=1,2,3,\dots,N \quad (7)$$

$$m_i(k) = \frac{fit_i(k) - \text{worst}(k)}{\text{best}(k) - \text{worst}(k)} \quad (8)$$

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^N m_j(k)} \quad (9)$$

$M_{ai}$  kütle için aktif yerçekimsel kütlesi,  $M_{pi}$  kütle için pasif yerçekimsel kütlesi,  $M_{ii}$  i. kütle için eylemsizlik kütlesi,  $M_i$  eylemsizlik kütlesidir,  $m_i(k)$  k. jenerasyondaki i. kütle için değeridir. Bu değişkenlerin hesaplanmasında (7), (8) ve (9) denklemleri birlikte kullanılır. Bu işlemde normalizasyon yapılmaktadır.

**6. Kuvvet Hesabı:** Newton'un evrensel yerçekimi kanunundan esinlenerek iki kütle için birbirine uyguladığı kuvvet; kütleler arası mesafenin kütleler çarpımına bölünmesiyle hesaplanır. İki kütle arasındaki kuvvet hesaplandıktan sonra bir kütleyle etkileyen toplam kuvvet hesaplanır.

$$R_{ij}(k) = \|x_i(k), x_j(k)\|_2 \quad (10)$$

(10)' da tanımlanan  $R_{ij}$  , k. jenerasyondaki i ve j kütleleri arasındaki mesafeyi vermektedir. Bu kütleler arasındaki kuvvet, (11) ile hesaplanır.

$$F_{ij}^d(k) = G(k) \frac{M_{pi}(k) M_{aj}(k)}{R_{ij}(k) + \epsilon} (x_j^d(k) - x_i^d(k)) \quad (11)$$

$F_{ij}^d(k)$ , k. jenerasyonda d. boyuttaki i ve j kütleleri arasındaki kuvveti;  $G(k)$ , yerçekimi sabitini;  $M_{pi}(k)$  ve  $M_{aj}(k)$  k. jenerasyondaki i. kütle için pasif ve aktif yerçekimsel kütlelerini;  $\epsilon$ , kullanıcı tarafından atanan sabit bir değeri,  $x_j^d(k)$  ve  $x_i^d(k)$  ise k. jenerasyonda i ve j kütlelerinin d. boyuttaki konumlarını belirtir.

Bir kütleyle etkileyen toplam kuvvet (12) yardımı ile hesaplanır.

$$F_i^d(k) = \sum_{j \in K_{best, j \neq i}}^N rand_j F_{ij}^d(k) \quad (12)$$

$rand_j$  ,  $[0,1]$  aralığında verilen rastgele bir sayıdır.  $K_{best}$  başlangıçta  $K_0$  kadar kütle ile başlayan lineer olarak azalan bir değerdir.

**7. İvme Hesabı:** Newton'un ikinci yasası olan ivme yasasına dayanarak kütlelerin ivmeleri (13) ile hesaplanır.

$$a_i^d(k) = \frac{F_i^d(k)}{M_{ii}(k)} \quad (13)$$

$a_i^d(k)$  k. jenerasyonda d. boyuttaki i. kütle için ivmesini verir.

**8. Hız ve Konum Güncellemesi:** Kütle için o andaki hızı ile o anda oluşan hız değişiminin (ivme) toplamına eşittir (14).

$$V_i^d(k+1) = rand_i V_i^d(k) + a_i^d(k) \quad (14)$$

$rand_i$  ,  $[0,1]$  aralığında atanan rastgele bir sayıdır.

İvme ile bir kütle için sistemdeki yeri değişmektedir. Bu sebepten kütle için konum güncellenmesi (15) ile yapılmaktadır.

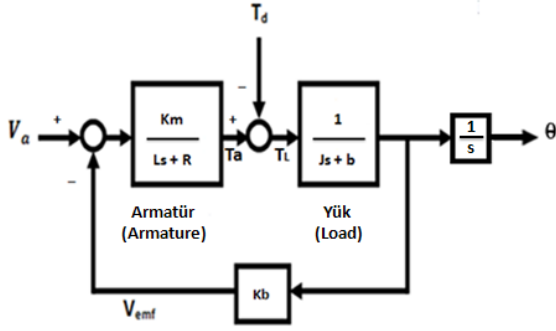
$$x_i^d(k+1) = x_i^d(k) + V_i^d(k+1) \quad (15)$$

**9. Son:** Maksimum jenerasyon sayısına ulaşıldığında ya da minimum hata koşulu sağlandığında, amaç fonksiyonuna en uygun olan kütle seçilir ve algoritma sonlanır.

### 3. Uygulama

Bu bölümde, sezgisel yöntemlerden biri olan YAA kullanılarak DA motor denetimi için PID denetleç parametrelerinin tespiti yapılmıştır. Endüvi sargısı denetimli DA motorunun hızı, endüvi sargısı uçlarına uygulanan gerilimle kolayca denetlenebilir.

Şekil 4' te DA motora ait blok diyagram verilmiştir [1,7].



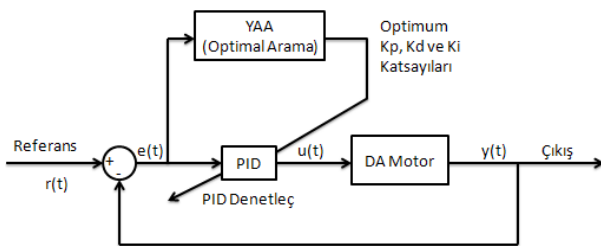
Şekil 4: DA motor hız denetimi blok diyagramı.

Çizelge 1' de çalışmada kullanılan DA motora ait parametreler verilmiştir [1].

Çizelge 1: DA motor parametreleri.

Parametreler			
Sembol	Değer	Sembol	Değer
<b>P</b>	93.3 kW	<b>R</b>	0.0125Ω
<b>V<sub>a</sub></b>	210 V	<b>b</b>	1.89 Nm/s
<b>Hız</b>	1150 dev/dk	<b>J<sub>s</sub></b>	2.83 kgm <sup>2</sup>
<b>L<sub>s</sub></b>	1.2 mH	<b>K<sub>m</sub>=K<sub>b</sub></b>	1.78 Nm/amp

Şekil 4 ve Çizelge 1 ile verilen DA motor diyagramı ve parametrelerde; J rotor eylemsizlik momentini, b mekanik sistemin sönüm oranını, K<sub>m</sub> motor sabiti, K<sub>b</sub> elektromotor kuvvet sabitini, R rezistansı, L<sub>s</sub> indüktansı, p gücü, V<sub>a</sub> kaynak voltajını, θ açısal hızı ifade eder.



Şekil 5: YAA ile PID denetleç optimizasyonu.

Şekil 5' te r(t) referans değerini, y(t) çıkış değerini, u(t) denetim değişkenini, e(t) hata değerini ifade eder.

YAA kullanılarak PID denetleç optimizasyonu için kullanılan algoritmanın kaba kodu;

YAA ve PID için başlangıç değerlerini ata  
K<sub>p</sub> K<sub>d</sub> ve K<sub>i</sub> değerlerinden oluşan N adet üyeyi rastgele konumlandır

```
DO FOR her bir kütle için
    Uygunluk değeri hesapla
    Sırasıyla kütle, kuvvet, ivme değerlerini hesapla
    Hız ve konum güncellemesini gerçekleştir
END
    En iyi, en kötü uygunluk değerlerini güncelle
WHILE sonlandırma kriteri sağlanana kadar
```

PID parametrelerinin optimizasyonu için ölçüt fonksiyonu olarak (16)' da verilen ortalama karesel hata (MSE) ölçütü kullanılmıştır.

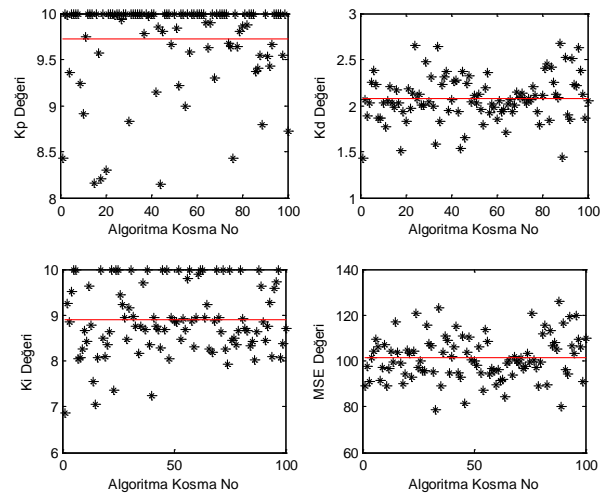
$$MSE = \frac{1}{N} \sum_{i=1}^N (\omega_{set} - \omega_{(i)})^2 \quad (16)$$

(16)' da ifade edilen N toplam örnek sayısını,  $\omega_{set}$  istenilen değeri,  $\omega_{(i)}$  ise çıkış değerini belirtmektedir.

Kullanılan algorithmada örnek sayısı 10, maksimum jenerasyon sayısı olarak 100 alınmıştır. Algorithmada kullanılan parametreler;  $\epsilon = 10^{-12}$ ,  $\alpha = 10$  ve  $G_c = 50$  olarak seçilmiştir. PID parametrelerinin sınır değerleri (17)' de verildiği biçimde kullanılmıştır.

$$0 \leq K_p \leq 10, 0 \leq K_d \leq 3, 0 \leq K_i \leq 10 \quad (17)$$

Yerçekimsel Arama Algoritması' na ait kodlar MATLAB programlama dilinde yazılmıştır. YAA' nın 100 defa koşturulması sonucu optimizasyonu gerçekleştirilen her bir koşturmaya ait PID denetleç parametreleri K<sub>p</sub>, K<sub>d</sub>, K<sub>i</sub> ve MSE' nin optimum değerleri Şekil 6' da gösterilmiştir.

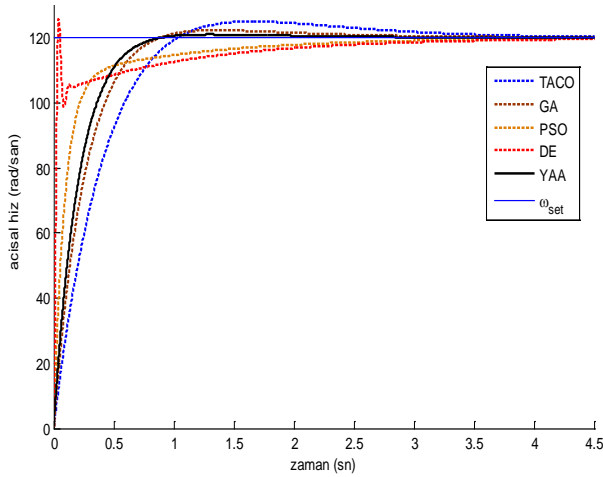


Şekil 6: YAA' nın 100 defa koşturulmasında elde edilen PID denetleç parametreleri ve MSE değerleri.

Çizelge 2: Algoritmaların 100 kez koşturulma sonucunda başarımları kıyaslama ölçütleri

Parametre	Merkezi Eğilim Ölçütü	Algoritma Türleri				
		TACO [1]	GA[1]	PSO[1]	DE[1]	YAA
Ölçüt Değeri	Ortalama	<b>170,26</b>	119,23	59,867	<u>22,676</u>	102,127
	Ortalama Sapma	8,6593	10,522	<b>18,742</b>	<u>0,61871</u>	6,4918
	Varyans	10,69	12,504	<b>26,74</b>	<u>1,4634</u>	8,0302
Kp değeri (Son Nesilde Bulunan)	Ortalama	7,1963	8,9058	8,8811	9,9754	9,7352
	Ortalama Sapma	0,36787	0,63712	<b>1,7076</b>	<u>0,044837</u>	0,3814
	Varyans	0,49811	0,78944	<b>2,3817</b>	<u>0,10946</u>	0,5299
Kd değeri (Son Nesilde Bulunan)	Ortalama	2,7884	2,2811	0,88703	0,11947	2,1038
	Ortalama Sapma	0,13507	0,17883	<b>0,34679</b>	<u>0,0011269</u>	0,1835
	Varyans	0,16943	0,22252	<b>0,42928</b>	<u>0,0029626</u>	0,239
Ki değeri (Son Nesilde Bulunan)	Ortalama	8,1434	8,7053	9,3894	9,8311	8,9742
	Ortalama Sapma	0,84161	0,84267	<b>0,92424</b>	<u>0,29939</u>	0,6831
	Varyans	0,96157	0,98309	<b>1,3758</b>	<u>0,6729</u>	0,8187
Koşma Zaman	Ortalama	14,195	13,371	1,6815	1,3937	2,9280

[1]' de kullanılan algoritmalara ait kodlar yazarlarının izni alınarak kullanılmış ve YAA dahil her bir algoritma 100 defa koşturulmuştur. Koşma sonucunda Çizelge 2' de yer alan değerler elde edilmiştir. Çizelgede iyi başarımları gösteren algoritmaya ait veri mavi ve altı çizgili olarak, en kötü başarımları gösteren algoritmaya ait veri ise kırmızı yazı tipi ile gösterilmiştir. Şekil 7' deki grafik, PID parametrelerinin denetim başarımları göstermektedir.



Şekil 7: Algoritmaların 100 defa koşturulması sonucu elde edilen ortalama PID denetleyici parametreleri ile YAA'nın denetim başarımları.

#### 4. Sonuçlar

Bu çalışmada, güncel optimizasyon algoritmalarından Rashedi ve arkadaşları tarafından önerilen YAA kullanılarak

DA motor denetimi için PID denetleyici parametrelerinin tespiti yapılmıştır. Çizelge 2' de ki verilere dayalı olarak [1] ' de ki çalışmaya kıyasla YAA'nın PID denetleyici parametreleri ile denetim başarımlarının, TACO ve GA' ya göre daha iyi, PSO ve DE' ye göre daha düşük olduğu anlaşılmaktadır. Çalışma sonucunda elde edilen veriler, YAA'nın PID denetleyici parametrelerinin optimizasyonunda kullanılabileceğini göstermektedir.

#### 5. Kaynaklar

- [1] Kesler, M., Karakuzu, C. ve Yüzgeç U., "Güncel Sezgisel Arama Algoritmalarının Denetleyici Parametrelerinin Optimizasyonunda Başarımları Kıyaslaması", *6<sup>th</sup> International Advanced Technologies Symposium (IATS'11)*, Elazığ, 281-286 (2011).
- [2] Akay, R., "Memetik Algoritma Kullanarak PID Denetleyici Tasarımı", Yüksek Lisans Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2006.
- [3] Karaboğa, D., "Yapay Zeka Optimizasyon Algoritmaları", *Atlas Yayın Dağıtım*, İstanbul, 2004.
- [4] Pinedo, M., "Scheduling: theory, algorithms, and systems", *Prentice-Hall*, Englewood Cliffs, NJ., 1995.
- [5] Rashedi, E., Nezamabadi, H. and Saryazdi, S., "GSA: A Gravitational Search Algorithm", *Information Sciences*, 179(13): 2232-2248 (2009).
- [6] Kazak, N., "Geliştirilmiş Yerçekimsel Arama Algoritması", Yayınlanmamış Yüksek Lisans Tezi, *Bilecik Üniversitesi Fen Bilimleri Enstitüsü*, Bilecik, 2011.
- [7] Ogata, K., "Modern Control Engineering 2<sup>nd</sup> Ed.", *Prentice-Hall, Inc.*, USA, 1990.