

# PARÇACIK SÜRÜ OPTİMİZASYONU İLE DAĞITIK TESPİT DISTRIBUTED DETECTION BY PARTICLE SWARM OPTIMIZATION

Özgür İNCE<sup>1</sup>, Murat EFE<sup>2</sup>, Orhan ARIKAN<sup>3</sup>

1. TÜBİTAK BİLGEM  
UEKAE / İLTAREN  
ozgur.ince@iltaren.tubitak.gov.tr

2. Elektronik Mühendisliği  
Bölümü  
Ankara Üniversitesi  
Murat.Efe@eng.ankara.edu.tr

3. Elektrik - Elektronik  
Mühendisliği Bölümü  
Bilkent Üniversitesi  
oarikan@ee.bilkent.edu.tr

## ÖZETÇE

*Doğadan esinlenerek geliştirilen optimizasyon yöntemleri, kullanım kolaylıkları ve yüksek performansları nedeniyle pek çok farklı disiplinde kendilerine uygulama alanı bulmaktadırlar. Bu çalışmada, son yıllarda yoğun ilgi gören dağıtık tespit problemine, literatürde Parçacık Sürü Optimizasyonu olarak bilinen yöntem uygulanmıştır. Temel bir paralel dağıtık topoloji üzerinde denenen senaryolarda PSO'nun performansının doğruluk ve güvenilirlik ve hız açısından oldukça yüksek olduğu gözlenmiştir.*

## ABSTRACT

*Nature inspired optimization methods have been finding many application areas in different disciplines due to their ease of use and high performance. In this study, Particle Swarm Optimization, a nature inspired optimization method, is applied to the distributed detection problem which is a popular topic of recent years. In the scenarios, experimented on a basic distributed topology, the performance of the PSO has been observed considerably high in terms of accuracy, reliability, and speed.*

## 1. GİRİŞ

Dağıtık tespit 1980'lerden özellikle de 1990'ların başından bu yana üzerinde yoğun olarak çalışılan bir konudur. Dağıtık tespit, alternatif olan merkezi tespite göre en büyük avantajı, kabul edilebilir performans seviyelerini tutturabilmesi için merkezi tespite göre çok daha düşük iletişim bant genişliklerine ihtiyaç duyuyor olmasıdır [1], [2]. Bunun yanı sıra tümleştirme merkezindeki işlem yükünün daha düşük oluşu; karar verme ve tepki sürelerinin kısa oluşu diğer önemli avantajları olarak sıralanabilir. Yöntemin en önemli dezavantajı ise doğruluk seviyesinin merkezi yöntemeye göre düşük kalıyor olmasıdır. Bu problemin çözümüne yönelik olarak, literatürde, iletişim bant genişliği gereksinimini asgaride tutarak tespit doğruluğunu arttırmayı hedefleyen pek çok çalışma bulunmaktadır [3], [4], [5].

Bu çalışmada, doğadan esinlenerek geliştirilmiş olan Parçacık Sürü Optimizasyonu (PSO) yönteminin dağıtık tespit performansının iyileştirilmesine yönelik olarak kullanımı iki algılayıcı ve bir tümleştirme merkezinden oluşan temel bir ağ topolojisi üzerinde incelenmektedir.

PSO, doğadaki kuş ve balık sürülerinin yem arama esnasında topluca ve bireysel olarak yaptıkları hareketler gözlenmek suretiyle geliştirilmiş bir yöntemdir [6]. Kuşlar ve balıklar, yem ararken, bireysel olarak ve komşuluklarındaki kuşların ve balıkların hareketlerine bağlı olarak kendilerine

yön vermektedirler. Bu sayede yem bulmak için harcamaları gereken süre ve efor hatırı sayılır şekilde düşmektedir.

Benzer bir yaklaşımın, dağıtık bir ağdaki her bir algılayıcının karar vermede kullanacağı optimum eşik seviyesinin ve tümleştirme merkezindeki karar mekanizmasının belirlenmesinde kullanılabilirdiğinden hareketle başlatılan bu çalışmada, PSO'nun performansını test etmek üzere, PSO ile elde edilen sonuçlar, tüm eşik değer ve tümleştirme kuralı kombinasyonları üzerinden yapılan (*brute force*, BF) aramanın sonuçları ile karşılaştırılmaktadır.

## 2. PARALEL TOPOLOJİ ve RİSK FONKSİYONU

Dağıtık algılayıcı ağları topolojilerine göre üç ana grupta toplanabilirler: Paralel, seri ve ağaç yapılı [1]. Paralel topolojide, tüm yerel algılayıcılar, hedeften elde ettikleri gözlemlere ( $y_1, y_2, \dots, y_N$ ) dayanarak verdikleri kararları ( $u_1, u_2, \dots, u_N$ ) dar bantlı bir iletişim kanalı üzerinden tümleştirme merkezine gönderirler. Tümüleştirme merkezi de yerel algılayıcılardan gelen kararları kullanmak suretiyle kendi kararını ( $u_0$ ) verir.

Yerel algılayıcıların yaptığı gözlemler sürekli değerlidir. Eğer bir ikili (*binary*) karar kuralı uygulanırsa yerel algılayıcı seviyesindeki karar mekanizmalarının çıkışları ikili olmaktadır. Bunlar, tipik olarak, sırasıyla hedefin yokluğunu ve varlığını gösteren  $H_0$  ve  $H_1$  sembolleri ile ifade edilirler.

Tümüleştirme merkezindeki tespit performansı, aşağıdaki denklemle verilen risk fonksiyonunu minimize etmek suretiyle optimize edilir [3].

$$R = \sum_{i=0}^1 \sum_{j=0}^1 c_{ij} P_j P(u_0 = i | H_j) \quad (1)$$

Burada,  $c_{ij}$ 'ler  $H_j$  doğru iken  $H_i$  kararı vermenin maliyeti;  $P_j$ ,  $H_j$  hipotezinin önsel olasılığı;  $P(u_0 = i | H_j)$  ise  $H_j$  doğru iken  $H_i$  kararı verme olasılığıdır. İkili karar verme yönteminde risk fonksiyonu aşağıda değişiklikler yapılmak suretiyle alternatif bir şekilde ifade edilebilir.

$$\begin{aligned} P(u_0 = 1 | H_0) &\rightarrow P_{FA} \\ P(u_0 = 1 | H_1) &\rightarrow P_D \end{aligned} \quad (2)$$

Burada,  $P_{FA}$  bütün sistemin yanlış alarm olasılığı ve  $P_D$  bütün sistemin tespit olasılığıdır. Eğer yerel algılayıcıların kararlarından oluşan vektör  $[u_1, u_2, \dots, u_N]'$ ,  $\bar{u}$  ile gösterilirse bu durumda bütün sistemin yanlış alarm ve tespit olasılıkları aşağıdaki gibi tanımlanır.

$$P_{FA} = \sum_{\bar{u}} P(u_0 = 1 | \bar{u}) P(\bar{u} | H_0)$$

$$P_D = \sum_{\bar{u}} P(u_0 = 1 | \bar{u}) P(\bar{u} | H_1)$$
(3)

Burada,  $\sum_{\bar{u}}$  karar vektörünün olası tüm değerleri üzerinden toplamı ifade etmektedir. Böylelikle risk fonksiyonu aşağıdaki gibi ifade edilir.

$$R = c_{00}P_0(1 - P_{FA}) + c_{01}P_1(1 - P_D) + c_{10}P_0P_{FA} + c_{11}P_1P_D$$
(4)

Son olarak da aşağıdaki değişiklikler yapıldığında,

$$c = c_{00}P_0 + c_{01}P_1$$

$$c_{FA} = c_{10}P_0 - c_{00}P_0$$

$$c_D = c_{01}P_1 - c_{11}P_1$$
(5)

Denklem (6) elde edilir.

$$R = c + c_{FA}P_{FA} - c_D P_D$$
(6)

### 3. PARÇACIK SÜRÜ OPTİMİZASYONU

PSO, çok farklı tipte probleme yönelik olarak kolayca uygulanabilen doğadan esinlenerek geliştirilmiş bir optimizasyon aracıdır [6]. Algoritma, genel olarak, hareketli parçacıkların hareket yönünün ve miktarının, parçacıkların kendilerinin önceki başarımlarına ve komşuluğundaki diğer parçacıkların başarımlarına göre belirlenmesi şeklinde ifade edilebilir. Çember, tekerlek, yıldız gibi çeşitli komşuluk topolojileri mevcuttur ve algoritma farklı topolojiler için farklı performans gösterebilmektedir. Tüm problemler için, bu topolojilerden birinin diğerlerine göre genellikle daha iyi sonuç verdiğini gösteren bir öneri henüz yapılmamıştır [7]

Komşulukları iki ayrı kategoriden ele almak mümkündür: Durağan ve dinamik. Durağan topolojide parçacıkların komşuları bir simülyasyon koşusu boyunca değişmeden kalmaktadır; diğer taraftan, dinamik topolojide komşular değişebilmektedir. Bu çalışmada, *Global Best* (*Gbest*, [7]) olarak da adlandırılan durağan komşuluk topolojisi uygulanmaktadır. *Gbest* topolojisinde tüm parçacıkların birbirleriyle komşu oldukları varsayılmakta ve herhangi bir parçacığın hareketi bütün sürüdeki en başarılı parçacığın hareketinden etkilenmektedir.

#### Algoritma

Kuş ve balık sürülerinin yem arama yöntemleri gözlemlenmek suretiyle geliştirilmiş olan PSO algoritması kolay anlaşılabilir bir yapıdadır. Aşağıda, algoritmanın ilklendirme ve öteleme adımları maddeler halinde verilmektedir.

#### İklendirme:

Aşağıdaki alanlardan oluşan bir *Sürü* yapısı oluştur:

- *Parçacıklar*: Her bir parçacık için, *Konum* ( $x_i$ ), *Hız* ( $v_i$ ), *Önceki En İyi Değer* ( $pbest_i$ ), *Önceki En İyi Konum* ( $xpbest_i$ ) alanlarından oluşan yapı dizisi.
- *Global En İyi Değer* ( $gbest$ )

- *Global En İyi Konum* ( $xgbest$ )
- *İvme Katsayıları* ( $\Phi_1, \Phi_2$ ):  $xpbest_i$  ve  $xgbest$  yönünde uygulanan rastgele kuvvetlerin büyüklükleri.
- *Atalet Katsayısı* ( $\omega$ ): Parçacıkların içinde hareket ettiği ortamın akışkanlığı olarak düşünülebilir [7].

#### Öteleme:

1. Tüm parçacıkların hızlarını ve konumlarını güncelle:

$$v_i = \omega v_i + u_1(xpbest_i - x_i) + u_2(xgbest - x_i)$$

$$x_i = x_i + v_i$$
(7)

burada,  $u_1$  ve  $u_2$ , sırasıyla  $[0, \Phi_1]$  ve  $[0, \Phi_2]$  aralıklarında değişen düzgün dağılımlı rastgele sayılardır.

2. Her bir parçacık için:
  - Risk fonksiyonunun değerini parçacığın konumu ( $x_i$ ) için hesapla.
  - Hesaplanan değeri  $pbest_i$  ile karşılaştır. Eğer değer daha iyi ise yeni değeri  $pbest_i$ 'ye yaz ve parçacığın yeni konumunu  $xpbest_i$ 'ye yaz.
  - $pbest_i$  ile  $gbest$ 'i karşılaştır. Eğer değer daha iyi ise  $gbest$ 'e  $pbest_i$ 'yi ve  $xgbest$ 'e  $xpbest_i$ 'yi yaz.
3. Eğer öteleme sayısı bir maksimum değeri aşmışsa veya  $gbest$  değeri yeterince iyi ise ötelemeyi sonlandır; aksi takdirde 1. Adıma dön.

### 3.1. PSO Yönteminin Dağıtık Tespit Optimizasyonuna Uygulanması

Bu kısımda PSO algoritmasının dağıtık tespit probleminin çözümüne uygulanması incelenmektedir. Uygulamada, yerel algılayıcıların optimum eşik seviyelerinin ve tümleştirme merkezinin kullanacağı optimum karar kuralı PSO algoritması kullanılarak aranacaktır. Yukarıda verilen PSO algoritması ile sürekli bir uzayda arama yapılabilir. Bu yerel algılayıcıların optimum eşik seviyelerinin belirlenmesinde iyi bir çözüm üretebilir. Ancak tümleştirme merkezinde uygulanacak olan optimum karar mekanizmasının belirlenmesi bütünüyle ayrı değerlerden (0 ve 1'lerden) oluşan bir uzayda arama yapmayı gerektirmektedir.

İkili PSO bu problemin çözümüne yönelik uygun bir yöntemdir. İkili PSO'da parçacıkların konum ve hızları ayrı değerlerdir. Bu da orijinal algoritmanın ilklendirme ve öteleme prosedürlerinde bazı değişikliklerin yapılmasını gerektirmektedir. Bu çalışmada, [8]'de verilen ikili PSO algoritması kullanılmıştır. Buna göre;

- Her bir parçacığın başlangıç konumu  $\{0, 1\}$  kümesinden rastgele seçilmekte,
- Her bir parçacığın başlangıç hızı  $\{-1, 0, 1\}$  kümesinden rastgele seçilmekte,
- Her bir parçacığın hız ve konum güncellemeleri aşağıdaki gibi gerçekleştirilmektedir.

$$v_i = v_i + d_1(xpbest_i - x_i) + d_2(xgbest - x_i)$$

$$x_i = x_i + v_i$$

$$x_i = (4 + x_i) \bmod 2$$

$$v_i = (3 + v_i) \bmod 3 - 1$$
(8)

Burada,  $d_1$  ve  $d_2$ ,  $\{-1, 1\}$  kümesinden seçilen rastgele değerlerdir.

#### 4. BENZETİM

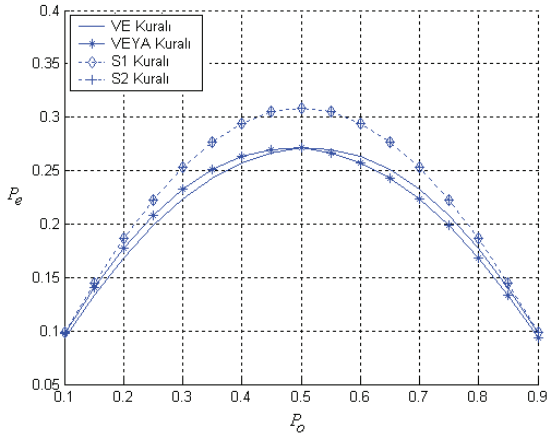
PSO'nun performansı bazı benzetimler üzerinde test edilmiş ve uzaydaki tüm "eşik değer – tümleştirme kuralı" kombinasyonları üzerinden yüksek çözünürlükle (bu uygulama için 0.05'lik adımlarla) arama yapan BF yönteminden elde edilen sonuçlarla karşılaştırılmıştır. Benzetimlerden elde edilen optimum eşik değerleri ve tümleştirme kuralları  $P_0$  önsel olasılığına bağlı olarak gösterilmektedir. Ayrıca benzetimlerde minimum hata olasılığı maliyet ataması ( $c_{00} = 0$ ,  $c_{01} = 1$ ,  $c_{10} = 1$  ve  $c_{11} = 0$ ) yapılmıştır.

Tablo 1'de benzetimde kullanılan yerel algılayıcıların ortalamaları ( $\mu_i$ ) ve standart sapmaları ( $\sigma_i$ ) sunulmaktadır. Buna göre iki durum dikkate alınmaktadır: (i) Durum I – Her iki algılayıcı da aynı karakteristiklere sahiptir; (ii) Durum II – Algılayıcıların ortalama değerleri farklıdır. Her iki durum için de algılayıcıların standart sapmaları 1 olarak tanımlanmıştır.

Tablo 1: Algılayıcı parametreleri

Durum	Algılayıcı	$\mu_0$	$\mu_1$	$\sigma_0$	$\sigma_1$
I	1	0	1	1	1
	2	0	1	1	1
II	1	0	1	1	1
	2	0	1.5	1	1

Benzetim sonuçları Bütün Sistemin Hata Olasılıkları ve Optimum Algılayıcı Eşikleri grafikleri şeklinde sunulmaktadır.



Şekil 1: Hata olasılıkları – Durum-I

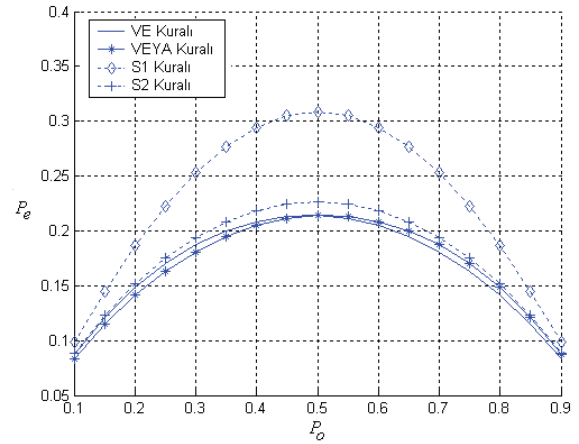
Şekil 1'de Durum-I için BF yöntemi ile elde edilen bütün sistemin hata olasılıklarını gösteren benzetim sonuçları gösterilmektedir. Sonuçlar açıklamaları aşağıda verilen dört olası tümleştirme kuralı için ayrı ayrı gösterilmektedirler. İki algılayıcı durum için toplamda 16 adet tümleştirme kuralı tanımlanabiliyor olmasına rağmen bunların büyük bir bölümü uygulanabilir olmadığı için burada dikkate alınmamışlardır.

- VE Kuralı: Eğer her iki algılayıcı da 1 kararı vermişse 1 kararı ver; aksi takdirde 0 kararı ver.
- VEYA Kuralı: Eğer her iki algılayıcı da 0 kararı vermişse 0 kararı ver; aksi takdirde 1 kararı ver.
- S1 Kuralı: Sadece birinci sensörden gelen kararı kullan.

- S2 Kuralı: Sadece ikinci sensörden gelen kararı kullan.

Buna göre, S1 ve S2 kuralının hata olasılıkları aynı ve diğerlerinden gözle görülür şekilde büyüktür. Bu nedenle S1 ve S2 kurallarının Durum-I için kullanımı uygun değildir. Diğer kurallara bakıldığında,  $P_0 < 0.5$  için VE kuralının;  $P_0 > 0.5$  ise VEYA kuralının hata olasılığının en düşük olduğu gözlenmektedir. Böylelikle,  $P_0$ 'ın 0.5'ten düşük olduğu durumlarda VE kuralının; büyük olduğu durumlarda da VEYA kuralının uygulanmasının optimum sonucu vereceği anlaşılmaktadır. Mantısal olarak da böyle bir çıkarımı yapmak mümkündür. Şöyle ki; aynı yapıda algılayıcılar kullanıldığında, eğer hedefin  $P_0$  önsel olasılığı 0.5'ten büyükse; yani  $P_1$ , 0.5'ten düşüğe bu durumda her türlü hedef tespit bilgisi kritik öneme sahiptir ve dikkate alınmak durumundadır. Dolayısı ile VEYA kuralının kullanımı gerekmektedir. Diğer taraftan, eğer hedefin bulunma olasılığı yüksekse ( $P_1 > 0.5$ ), bu durumda da tümleştirme merkezinde yanlış alarm kaynaklı işlem yükünü azaltmaya yönelik olarak VE kuralının kullanımı gerekmektedir.

Benzer bir grafik Durum-II için de çizdirilmiş (Şekil 2) ve özellikle S1 ve S2 kurallarının hata olasılıklarında önemli farklar gözlenmiştir. Bunların temel nedeni Durum-II'deki ikinci algılayıcının ortalama değerinin daha yüksek olması; başka bir deyişle alıcı çıkışındaki Sinyal/Gürültü oranının daha yüksek olmasıdır. Bu sayede ikinci algılayıcının hata olasılığı daha düşük çıkmaktadır. Ancak burada da S2 kuralının hata olasılığı VE ve VEYA kurallarının hata olasılıklarının altına inmiştir.



Şekil 2: Hata olasılıkları – Durum-II

#### 4.1. Karşılaştırma

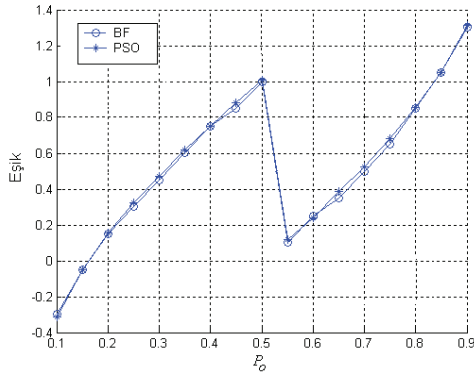
Bu kısımda PSO'nun performansı BF benzetim sonuçları ile karşılaştırılmak suretiyle doğruluk, güvenilirlik ve hız açısından değerlendirilmektedir.

Tablo 2: PSO parametreleri

Parametre	Değer
Parçacık sayısı	80
İvme katsayıları	$\Phi_1 = \Phi_2 = 1.49618$
Atalet katsayısı	0.7298
Topoloji	Gbest
Max. Öteleme Sayısı	200
Eşik Arama Bölgesi	$[-5, 5]$

Tablo 2’de benzetimde kullanılan PSO algoritmasının parametre değerleri sunulmaktadır. İvme katsayıları ve atalet katsayısı [7]’de verildiği gibi, yakınsama garantisi sağlayan kısıtlı PSO’ya karşılık gelecek şekilde, 1.49618 ve 0.7298 olarak seçilmişlerdir. Algılayıcıların ortalama değerleri [0, 1.5] aralığında değiştiğinden ve her durum için standart sapma değerleri 1’e eşit olduğundan eşik aramasına [-5, 5] aralığı her iki algılayıcı için de uygundur.

Şekil 3’te Durum-I için BF ve PSO ile belirlenen optimum algılayıcı eşikleri karşılaştırılmaktadır. Şekilde görüldüğü gibi PSO ile belirlenen eşik değerleri BF ile belirlenenlerle hemen hemen aynıdır. Burada PSO yönteminin doğruluğu açıkça gözlenmektedir.  $P_0 = 0.5$ ’teki sıçrama tümleştirme kuralının VE’den VEYA’ya geçmesinden kaynaklanmaktadır. Kuralın değişimine bağlı olarak optimum algılayıcı eşikleri de değişmektedir. Bunlara ek olarak, PSO’nun tüm  $P_0$  değerleri için sonuç vermiş olması ve dolayısıyla herhangi bir iraksama problemi yaşanmamış olması da yöntemin güvenilirliğini ortaya koymaktadır.



Şekil 3: Optimum eşiklerin karşılaştırılması – Durum-I

Benzer sonuçlar Durum-II için de elde edilmiştir. Ancak Durum-II’de algılayıcıların karakteristikleri birbirlerinden farklı olduğu algılayıcıların optimum eşikleri de farklı çıkmaktadır (Şekil 4 ve 5). Burada da PSO ile elde edilen sonuçların BF ile elde edilenlerle hemen hemen aynı oldukları ve herhangi bir iraksama probleminin yaşanmadığı açıkça gözlenmektedir.

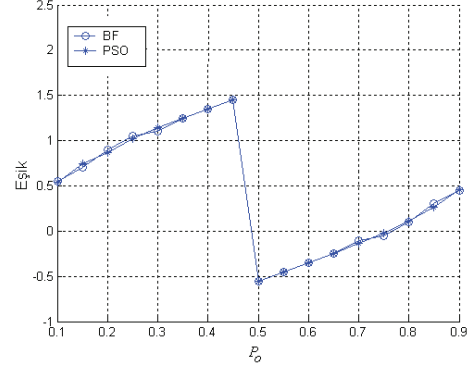
Son olarak, benzetimlerde, PSO’nun çalışma hızı BF ile karşılaştırılmıştır. Intel® Core™2 CPU @ 2.16 GHz, 2 GB RAM özelliklerine sahip bir bilgisayar kullanılarak gerçekleştirilen koşullarda BF yöntemi kullanılarak [0.1, 0.9] aralığında 0.1’lik adımlarla belirlenen tüm  $P_0$  değerleri için optimum eşik seviyelerinin ve tümleştirme kurallarının belirlenmesi yaklaşık olarak 450 s sürerken PSO’da bu sadece 10 s sürmüştür. Bu da PSO yönteminin performansı hakkında önemli bir göstergedir. Ayrıca, algılayıcı sayısının artmasıyla aradaki oranın çok daha büyüyeceği; hatta, belli bir sayıdan fazla algılayıcı için BF’nin kullanımının mümkün olmayacağı açıkça görülmektedir.

## 5. SONUÇ

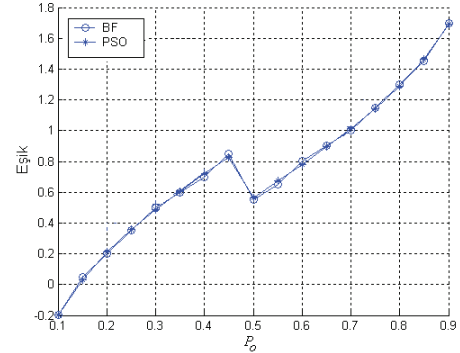
Bu çalışmada, dağıtık tespit probleminin çözümünde PSO yönteminin kullanımının mümkün olduğu, temel bir dağıtık ağ topolojisi baz alınmak suretiyle oluşturulan benzetimler üzerinden gösterilmeye çalışılmıştır. Benzetim sonuçları, PSO

yönteminin, doğruluk, güvenilirlik ve hız açısından yüksek bir performansa sahip olduğunu ve dağıtık tespit probleminin çözümünde kullanılabileceğini göstermektedir.

Konu ile ilgili olarak, ikiden fazla sayıda algılayıcı barındıran alternatif ağ topolojileri için, PSO’nun ve farklı optimizasyon yöntemlerinin performanslarının analiz edileceği kapsamlı çalışmalara ihtiyaç olduğu değerlendirilmektedir.



Şekil 4: Birinci algılayıcı optimum eşiklerinin karşılaştırılması – Durum-II



Şekil 5: İkinci algılayıcı optimum eşiklerinin karşılaştırılması – Durum-II

## 6. KAYNAKÇA

- [1] Viswanathan R., Varshney P. K., *Distributed Detection with Multiple Sensors: Part I – Fundamentals*, Proceedings of the IEEE, January 1997.
- [2] Chamberland, J.-F., V. Venugopal, *Wireless Sensors in Distributed Detection Applications*, IEEE Signal Proc. Magazine, May 2007.
- [3] Varshney P. K., *Distributed Detection and Data Fusion*, Springer Verlag, 1997.
- [4] Lee, C.-C., Chao J.-J., *Optimum Local Decision Space Partitioning for Distributed Detection*, IEEE Transactions On Aerospace And Electronic Systems Vol. 25. No. 4 July 1989.
- [5] Wang X.-G., Shen H. C., *Multiple Hypotheses Testing Strategy for Distributed Multisensor Systems*, Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [6] Nedjah N., Mourelle L. de M., *Swarm Intelligent Systems*, Springer-Verlag, 2006.
- [7] Poly, R. et al., *Particle Swarm Optimization – An Overview*, Swarm Intelligence, Springer, pp. 33-57, 2007.
- [8] <http://clerc.maurice.free.fr/psa>