Mofidifiye Pekiştirmeli Öğrenme Yöntemi ile Fonksiyon Minimizasyonu

Özgür BAŞKAN* Cenk OZAN* Halim CEYLAN* Soner HALDENBİLEN*

*İnşaat Mühendisliği Bölümü
Mühendislik Fakültesi
Pamukkale Üniversitesi, Kınıklı, Denizli
Email: obaskan@pau.edu.tr cozan@pau.edu.tr halimc@pau.edu.tr shaldenbilen@pau.edu.tr

Özet

Pekiştirmeli Öğrenme (PÖ), temel olarak belirli bir amaca erişmek için çevre ile etkileşim sonucunda öğrenme anlamına gelmektedir. PÖ, simülasyon bazlı dinamik programlamanın bir şekli olup ilk olarak Markov karar süreçlerinin çözümü amacıyla kullanılmıştır. Bu çalışmada temel PÖ algoritmalarından olan Q-öğrenme algoritması modifiye edilmiş ve test fonksiyonları üzerinde performansı araştırılmıştır. Sonuçlar literatürde mevcut olan diğer metotlar ile karşılaştırılmış ve modifiye PÖ algoritmasının fonksiyon minimizasyonunda oldukça başarılı olduğu görülmüştür.

1. Giriş

Pekiştirmeli Öğrenme (PÖ) herhangi bir öğretici kaynağının bulunmasını gerektirmeyen buna karşılık sistemin çevre ile etkileşimini kullanan bir öğrenme yöntemidir [1]. PÖ metodunda öğrenen ve öğrenme süreci olmak üzere iki kavram söz konusudur. Her bir öğrenme evresinde öğrenen mevcut durumunu değerlendirerek bir olay seçer ve ortamdan aldığı geri beslemeyi sisteme uygular. Öğrenenin amacı herhangi bir durumda en uygun davranışı gösterebilmektir. PÖ yöntemi son zamanlarda literatürde farklı alanlardaki optimizasyon problemlerinin çözümünde kullanılmasına ve başarılı sonuçlar alınmasına rağmen fonksiyon minimizasyonu konusunda uygulaması bulunmamaktadır [2-5].

Herhangi bir $F(x)$ fonksiyonu göz önüne alındığında bütün x değerleri için eğer $F(x_{\min}) \leq F(x)$ ise $x_{\min}$ noktası fonksiyonun minimum noktası olarak tanımlanmaktadır. Fonksiyon birçok yerel minimuma sahip olabileceği gibi global minimum noktası bunlardan bir tanesidir. Global minimumun bulunmasında stokastik metotların birçok avantajı vardır. Konveks olmayan yapıdaki fonksiyonların global minimum noktalarının bulunabilmesi için şimdiye kadar birçok sezgisel metot geliştirilmiştir [6-9]. Ancak problemin karmaşıklığından dolayı global ve yakın global optimum değerlerin bulunabilmesi amacıyla farklı metotların performanslarının test edilmesinin gerekliliği açıktır. Bu nedenle çalışmada modifiye edilmiş PÖ yönteminin test fonksiyonları üzerinde performansı araştırılmıştır.

2. Pekiştirmeli Öğrenme

PÖ yaklaşımlarının temel felsefesi geçekleşen bir olgunun ödüllendirilmesi veya cezalandırılmasını bir mantık içerisinde düzenleyerek istenen bir sonucun ortaya çıkmasını sağlamaktır. PÖ'de karar verici ajan olarak adlandırılmakta ve çevresi ile etkileşim içinde olmaktadır. Bu etkileşim, ajana çevre içinde uygulayacağı eylemi seçmesini sağlamaktadır.

PÖ yöntemleri, Dinamik Programlama, Monte Carlo ve Geçici Fark Öğrenme metotları olmak üzere üç ana kategoriden oluşmaktadır. Her kategorinin kendine göre avantaj ve dezavantajları bulunmaktadır. Geçici fark öğrenme metotlarından biri olan Q-öğrenme algoritması, PÖ yöntemleri içinde modele ihtiyaç duymayan bir yaklaşım olup bu algoritmada çevrenin nasıl çalıştığı hakkında ajana herhangi bir bilgi verilmemektedir. Bunun yerine ajan eylemleri deneyerek en iyi ödülü veren eylemi seçmektedir. Q-öğrenme algoritması, durum-eylem çiftinin sahip olduğu değerlerin tahmin edilmesine bağlı olarak çalışmaktadır. Q değerleri olarak adlandırılan bu değerler verilen bir durum-eylem çifti için sayısal tahminler olarak nitelendirilmiştir [10]. Algoritmada, Q tablosu olarak adlandırılan tablonun elemanları her durum değişiminde güncellenmektedir [1]. Bu tabloda her bir s durumu ve a eylemi çifti için $Q(s,a)$ olarak nitelendirilen Q değeri bulunmaktadır. Ajan, çevrenin s_t durumundan s_{t+1} durumuna geçişinde yapılan eylemin (a_t)

ardından r_{t+1} olarak gösterilen ödülü almaktadır. Q değerleri Denklem (1) ile tanımlanmaktadır.

$$Q(s,a) = r(s,a) + \gamma \times Q^*(s',a') \quad (1)$$

Burada; $Q(s,a)$, (s,a) durum eylem çifti için Q değeridir. $Q^*(s',a')$, s durumunda a eyleminin tamamlanmasının ardından gelen s' durumunda seçilen a' eylemi ile elde edilebilen en iyi Q değeridir. $r(s,a)$, s durumunda a eylemi tamamlandığında alınan ödül değeri olup γ gelecekteki ödüllere atanan ağırlığı temsil eden azaltma faktörüdür [11].

Öğrenme süreci, belirli sayıdaki öğrenme evresinden meydana gelmektedir. Her öğrenme evresi rastgele bir s durumunda başlamaktadır. Sonrasında ajan bir a eylemi seçerek ödülünü almakta ve yeni durumu gözlemlemektedir. Buna bağlı olarak, ajan Q değerlerini durum-eylem çiftine göre Denklem (2)'de verilen bağıntıyı kullanarak güncellemektedir.

$$Q_t(s,a) = (1-\alpha) \times Q_{t-1}(s,a) + \alpha \times \left[r(s,a) + \gamma \times \max_{a'} Q_{t-1}(s',a') \right] \quad (2)$$

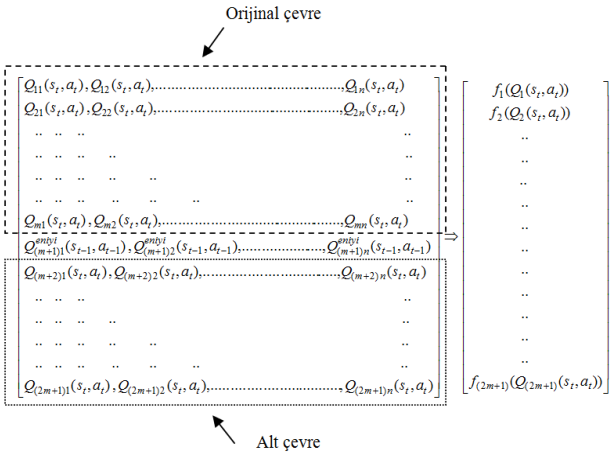
Burada, $Q_t(s,a)$ güncellenmiş Q değeri, $Q_{t-1}(s,a)$ Q tablosunda önceden kaydedilmiş Q değeri ve α algoritmanın öğrenme oranı olup önceden kaydedilen $Q_{t-1}(s,a)$ değeri ile karşılaştırmak için yeni hesaplanan Q değerine atanan ağırlığı tanımlamaktadır [11].

PÖ ile diğer sezgisel metotların birleştirilmesi ile elde edilen algoritmalar farklı alanlardaki optimizasyon problemlerinin çözümünde son yıllarda etkin olarak kullanılmaktadır. PÖ optimizasyon problemlerine uygulanabilmesine rağmen, matematiksel fonksiyonların global minimum değerlerinin elde edilmesi amacıyla literatürde kullanılmamıştır. Bu amaçla çalışmada MOdifiye Pekiştirmeli Öğrenme (MOPÖ) algoritması geliştirilmiştir. Q-öğrenme algoritması tabanlı MOPÖ algoritmasında her öğrenme evresinde önceki evredeki en iyi çözüm bilgisinin yardımıyla orijinal çevre boyutunda bir alt çevre oluşturulmaktadır.

Şekil 1'den görüldüğü gibi, önceki öğrenme evresinden elde edilen en iyi $Q(s,a)$ değeri, yerel optimuma takılmayı önlemek için çevrenin (m+1)'inci satırında saklanmaktadır. Ayrıca alt çevre, orijinal çevre boyutunda (m+2)'nci satırdan (2m+1)'nci satıra kadar önceki öğrenme evresinde elde edilen en iyi değer ve $\boldsymbol{\beta}$ vektörü kullanılarak Denklem (3) yardımıyla oluşturulmaktadır. Böylelikle, global optimum, algoritma süreci boyunca $\boldsymbol{\beta}$ değeri ile sınırlandırılan alt çevre yardımıylada aranmaktadır. $\boldsymbol{\beta}_j$ vektör ve j=1,2,..,n kadar olup, n karar değişkenlerinin sayısıdır. $\boldsymbol{\beta}$ değerinin aralığı verilen probleme bağlı olarak seçilebilmektedir [6].

$$rastgele(Q_{t-1}^{best}(s,a) - \beta \; ; \; Q_{t-1}^{best}(s,a) + \beta) \quad (3)$$

MOPÖ algoritmasında, alt çevre oluşturulduktan sonra orijinal çevre ve alt çevre iyiden kötüye doğru sıralanmaktadır. Bu sayede, bir önceki öğrenme evresinden elde edilen en iyi çözüm ve alt çevre, orijinal çevre ile karşılaştırılmaktadır. Yeni çözümlerden biri daha iyi amaç fonksiyonu değeri sağlıyorsa, yeni değer orijinal çevrenin içine dahil edilirken kötü değer çevreden çıkarılmaktadır.



Şekil 1. Orijinal-alt çevre ilişkisi.

Burada, m çevrenin boyutunu, n karar değişkenlerinin sayısını ve t öğrenme evresinin sayısını ifade etmektedir. Algoritmanın çözüm adımları Şekil 2'de gösterilmektedir.

Başlangıç t=0. β, α, γ *ayarla*
While $t <= t_{max}$ (t_{max}=*maksimum öğrenme evresi sayısı*)
  If t=0, *başlangıç* $Q(s,a)$ *değerlerini oluştur*
  Amaç fonksiyonunun değerlerini hesapla
  Else
  En iyi $Q_{t-1}^{best}(s,a)$ *sakla*
  $\beta_t = \beta_{t-1} * 0.99$
  Alt çevreyi Denklem (3) ile oluştur
  Amaç fonksiyonunun değerlerini hesapla
  Çevre ve alt çevreyi iyiden kötüye doğru sırala
  end if
  En iyi $Q_t^{best}(s,a)$ *bul*
  $r_t(s,a)$ *değerini Denklem (4) ile hesapla*
  $Q_t(s,a)$ *değerlerini Denklem (2) ile güncelle*
t=t+1
End while

Şekil 2. MOPÖ algoritma adımları.

MOPÖ algoritmasında $r_t(s,a)$ olarak belirtilen ödül değeri, *s* durumunda yapılan *a* eyleminin amaç fonksiyonunu nasıl etkilediğini belirlemektedir. Geliştirilen ödül fonksiyonu Denklem (4)'de verilmektedir.

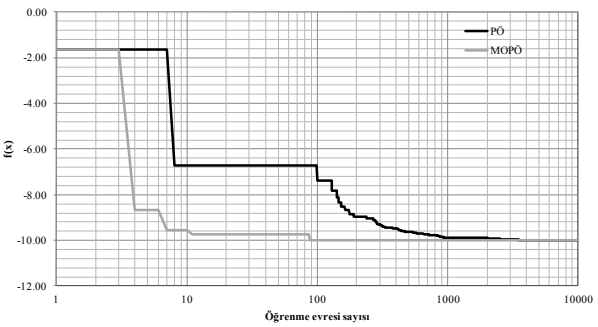
$$\lim_{r_t(s,a)\to 0}\left(\frac{Q_t^{best}(s,a)-Q_t(s,a)}{Q_t(s,a)}\right)=0 \qquad (4)$$

Burada, $r_t(s,a)$ ödül fonksiyonu, $Q_t(s,a)$ *Q* değeri ve $Q_t^{best}(s,a)$ *t*'nci öğrenme evresinde elde edilen en iyi *Q* değeridir. MOPÖ algoritmasında ödül fonksiyonu, en iyi *Q* değeri ile öğrenme evresindeki *Q* değeri arasındaki farkın *Q* değerine bölümü olarak belirlenmektedir. Ödül değerleri, fonksiyonunun yapısından dolayı "0" değerine yaklaşmaktadır. Diğer bir deyişle MOPÖ algoritmasında toplam ödül değerinin, verilen bir fonksiyonun global yada yakın global değerinin bulunabilmesi için minimize edilmesi gerekmektedir.

3. Test Fonksiyonları

MOPÖ algoritması MATLAB R2009a yazılımı kullanılarak kodlanmış ve Intel Core2 CPU 2.00 GHz, RAM 2 GB bilgisayarda çalıştırılmıştır. MOPÖ algoritmasında *m* çevre boyutu 20, γ azaltma faktörü 0.2 ve α öğrenme oranı 0.8 olarak alınmıştır. **β** vektörü verilen problemin maksimum ve minimum sınırlarının mutlak değer olarak toplamı olarak belirlenmiştir [6].

İlk olarak MOPÖ ve PÖ (*Q*-öğrenme) algoritmalarının performansları $f(x_1,x_2)=\frac{x_1}{1+|x_2|}$ fonksiyonu ile karşılaştırılmıştır. Fonksiyonun global minimum değeri $(-10,0)$ aralığında $f(x_1,x_2)=-10$ olmaktadır. MOPÖ ve PÖ algoritmalarının yakınsama davranışları Şekil 3'de görülebilmektedir.



Şekil 3. PÖ ve MOPÖ karşılaştırılması.

Şekilden görüldüğü gibi MOPÖ 1176 öğrenme evresinde global minimum değere ulaşırken PÖ algoritması bu değere ulaşmak için 7337 öğrenme evresine ihtiyaç duymaktadır.

MOPÖ algoritmasının performansı ayrıca 10 farklı test fonksiyonu üzerinde Tablo 1'de verilen algoritmalar ile karşılaştırılmıştır. Sonuçlar; en iyi fonksiyon değeri, fonksiyon çözüm sayısı ve ortalama hata olarak Tablo 2'de verilmiştir. En iyi fonksiyon değeri, kabul edilen hassasiyet değeri göz önüne alınarak global çözüme en yakın değer olarak değerlendirilmiştir. Hassasiyet değeri, bulunan en iyi fonksiyon değeri ile teorik global değer arasındaki farkın mutlak değeri olarak belirlenmiştir. Tüm test fonksiyonları için bu değer "0" olarak seçilmiştir. MOPÖ algoritması verilen hassasiyet değeri sağlanana kadar çalıştırılmıştır. Fonksiyon çözüm sayısı en iyi fonksiyon değerinin elde edildiği çözüm sayısı olarak ifade edilmiştir. Her bir çözümde elde edilen en iyi fonksiyon değeri ile teorik global optimum değer arasındaki farkların ortalaması ise "ortalama hata" olarak değerlendirilmiştir. Bu değer her bir farklı denemede algoritmanın çözüm kararlılığını temsil etmesi açısından oldukça önemli bir parametredir.

Tablo 1. Fonksiyonlar ve karşılaştırılan algoritmalar.

Fonksiyon	Kaynak
F3	[12]
F3-F7	[7]
F1-F6-F7-F9-F10	[9]
F1	[13]
F4	[14]
F8	[15]
F1-F2-F3-F5-F6-F7	[16]
F3-F5	[8]

MOPÖ algoritmasının performansının değerlendirilmesinde kullanılan fonksiyonlar ve özellikleri aşağıda sıralanmıştır. F7 ve F9 fonksiyonları sırasıyla 6 ve 8 değişkenlidir. F1 fonksiyonu 3 değişkenli olup diğer fonksiyonlar iki değişkenlidir.

F1:
De Jong (3 değişken)

$$f(x)=\sum_{i=1}^{n}x_i^2$$

Çözüm aralığı: $-5.12\le x_i\le 5.12; i=1,2,....n$
Global minimum: $x=(0,0,0)$, $f(x)=0$

F2:
Easom (2 değişken)

$$f(x_1,x_2) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$$

Çözüm aralığı: $-100 \le x_1, x_2 \le 100$
Global minimum: $(x_1,x_2) = (\pi,\pi)$, $f(x_1,x_2) = -1$

F3:
Goldstein-Price (2 değişken)

$$f(x,y) = \left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right] * \left[(30+(2x_1-3x_2)^2 18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right]$$

Çözüm aralığı: $-2 \le x_1, x_2 \le 2$
Global minimum: $(x_1,x_2) = (0,-1)$, $f(x_1,x_2) = 3$

F4:
Drop wave (2 değişken)

$$f(x_1,x_2) = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{\frac{1}{2}(x_1^2+x_2^2)+2}$$

Çözüm aralığı: $-5.12 \le x_1, x_2 \le 5.12$
Global minimum: $(x_1,x_2) = (0,0)$, $f(x_1,x_2) = -1$

F5:

Shubert (2 değişken)

$$f(x_1,x_2) = \sum_{i=1}^{5} i.\cos((i+1)x_1+1) * \sum_{i=1}^{5} i.\cos((i+1)x_2+1)$$

Çözüm aralığı: $-10 \le x_1, x_2 \le 10$
760 adet yerel minimum noktası
18 adet global minimum: $f(x_1,x_2) = -186.7309$

F6:
Zakharov (2 değişken)

$$f(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5\,i\,x_i)^2 + (\sum_{i=1}^{n} 0.5\,i\,x_i)^4$$

Çözüm aralığı: $-5 \le x_i \le 10$, $i = 1,2,....n$
Global minimum: $x = (0,0)$, $f(x) = 0$

F7:
Hartman (6 değişken)

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} \alpha_{ij}(x_j - p_{ij})^2\right]$$

Çözüm aralığı: $0 \le x_j \le 1$, $j = 1,2,..6$
Global minimum:
$x = (0.201, 0.150, 0.477, 0.275, 0.311, 0.657)$
$f(x) = -3.32$

α ve p değerleri [9]'dan alınabilir.

F8:

Rastrigin (2 değişken)

$$f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]$$

Çözüm aralığı: $-5.12 \le x_i \le 5.12$, $i = 1,2,...n$
Global minimum: $(x_1,x_2) = (0,0)$, $f(x) = 0$

F9:

Griewank (8 değişken)

$$f(x) = \sum_{i=1}^{n} x_i^2 / 4000 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$$

Çözüm aralığı: $-300 \le x_i \le 600$, $i = 1,2,...n$
Global minimum: $x = (0,.....0)$, $f(x) = 0$

F10:
Branin (2 değişken)

$$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1-\frac{1}{8\pi})\cos(x_1) + 10$$

Çözüm aralığı: $-5 \le x_1 \le 10$, $0 \le x_2 \le 15$
Global minimum noktaları:
$(-\pi, 12.275), (\pi, 2.275), (3\pi, 2.475)$ $f(x) = \frac{5}{4\pi}$

Tablo 2'de MOPÖ algoritmasının 100 defa çalıştırılması neticesinde elde edilen değerler verilmiştir. MOPÖ tüm test fonksiyonları için literatürde verilen global minimum değerleri bulabilmektedir. Ortalama hata değerlerine bakıldığı zaman, MOPÖ algoritması ile F7 fonksiyonu hariç tüm fonksiyonlarda her çözümde global minimum değerlerin elde edilebildiği görülmektedir. Bunun yanında F7 fonksiyonu için elde edilen ortalama hata değeri de karşılaştırılan diğer algoritmalara göre daha düşüktür. Fonksiyon çözüm sayılarının karşılaştırılması durumunda MOPÖ'nün diğer algoritmalara göre daha fazla çözüm sayısında sonuca ulaştığı görülmektedir. Bunun nedeni tüm test fonksiyonları için MOPÖ algoritmasında hassasiyet değerinin "0" olarak seçilmesidir. Diğer bir deyişle MOPÖ teorik global minimum değer elde edilinceye kadar çalıştırılmıştır. Elde edilen sonuçlara göre geliştirilen MOPÖ algoritmasının fonksiyon optimizasyonunda oldukça başarılı olduğu görülmektedir.

Tablo 2. MOPÖ ve karşılaştırılan algoritmaların sonuçları

Fonksiyon	Algoritma	En iyi fonksiyon değeri	Fonksiyon çözüm sayısı	Ortalama hata
F1	[13]	-	-	3e-08
	[9]	-	-	7.69e-29
	[16]	-	-	0.00004
	MOPÖ	0	36955	0
F2	[16]	-	-	0.00009
	MOPÖ	-1	40680	0
F3	[12]	3	9000	-
	[7]	-	-	-
	[8]	3	2573	-
	[16]	-	-	0.00012
	MOPÖ	3	33920	0
F4	[14]	-9.99e-01	16801	-
	MOPÖ	-1	35640	0
F5	[8]	186.7309	2568	-
	[16]	-	-	0.00007
	MOPÖ	186.7309[a]	16740	0
F6	[9]	-	-	5.71e-27
	[16]	-	-	0.00005
	MOPÖ	0	37016	0
F7	[9]	-	-	4.47e-11
	[16]	-	-	0.00024
	[7]	-	-	-
	MOPÖ	-3.32[b]	30400	2.34e-16
F8	[15]	-	-	0[c]
	MOPÖ	0	33500	0
F9	[9]	-	-	6.23e-22
	MOPÖ	0	37010	0
F10	[9]	-	-	2.61e-13
	MOPÖ	0.3979[a]	7680	0

[a]Teorik global optimum değer 4 ondalıklı olarak alınmıştır.
[b] Teorik global optimum değer 2 ondalıklı olarak alınmıştır.
[c]Algoritma 30 defa çalıştırılmıştır.

4. Sonuçlar

Çalışmada PÖ yönteminin modifiye edilmesi ile MOPÖ yöntemi geliştirilmiştir. MOPÖ ve PÖ algoritmaları karşılaştırılmış ve MOPÖ'nün çok daha az sayıda öğrenme evresi ile verilen fonksiyon için teorik global minimum değeri bulabildiği gösterilmiştir. 10 adet test fonksiyonu üzerinde MOPÖ ile elde edilen sonuçlar literatürdeki mevcut yöntemlerle karşılaştırılmıştır. Elde edilen sonuçlar geliştirilen MOPÖ algoritmasının çok değişkenli fonksiyonlar dahil olmak üzere fonksiyon minimizasyonunda oldukça başarılı olduğunu göstermektedir.

5. Teşekkür

Bu çalışma Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Biriminin desteklemiş olduğu BAP-FBE-063 nolu proje kapsamında gerçekleştirilmiştir.

6. Kaynaklar

[1] R.S Sutton, A.G. Barto, Reinforcement learning: An introduction, The MIT Press, Cambridge, Massachusetts, USA/London, England,1998.

[2] J. Wu, X. Xu, P. Zhang ve C. Liu, "A Novel Multi-Agent Reinforcement Learning Approach for Job Scheduling in Grid Computing", Future Generation Computer Systems, 27, 2011, s. 430-439.

[3] D. Maravall, J. de Lope ve H.J.A. Martin, "Hybridizing Evolutionary Computation and Reinforcement Learning for the Design of Almost Universal Controllers for Autonomous Robots", Neurocomputing, 72, 2009, s. 887-894.

[4] Y. Chen, S. Mabu, K. Shimada ve K. Hirasawa, "A Genetic Network Programming with Learning Approach for Enhanced Stock Trading Model", Expert Systems with Applications, 36, 2009, s. 12537-12546.

[5] A. Belbekkouche, A. Hafid ve M. Gendreau, "Novel Reinforcement Learning-Based Approaches to Reduce Loss Probability in Buffer-less OBS Networks", Computer Networks, 53, s. 2091-2105, 2009.

[6] O. Baskan, S. Haldenbilen, H. Ceylan, H. Ceylan, "A new solution algorithm for improving performance of ant colony optimization", Applied Mathematics and Computation, 211, 75-84, 2009.

[7] M.D. Toksarı, "Minimizing the multimodal functions with Ant Colony Optimization approach", Expert Systems and Applications, 36(3), s. 6030-6035, 2009.

[8] N. Tutkun, "Optimization of Multimodal Continuous Functions Using a New Crossover for the Real-coded Genetic Algorithms", Expert Systems and Applications, 36, s. 8172-8177, 2009.

[9] P.S. Shelokar, P. Siarry, V.K. Jayaraman, B. D. Kulkarni, "Particle S warm and Ant Colony Algorithms Hybridized for Improved Continuous optimization", Applied Mathematics and Computation, 188, s. 129-142, 2007.

[10] A.L.C. Bazzan, D. Oliviera ve B.C. Silva, "Learning in Groups of Traffic Signals", Engineering Applications of Artificial Engineering, 23, s. 560-568, 2010.

[11] M. Vanhulsel, D. Janssens, G. Wets ve K. Vanhoof, "Simulation of sequential data: An Enhanced Reinforcement Learning Approach", Expert Systems with Applications, 36, s. 8032-8039, 2009.

[12] Y.D. Kwon, S.B. Kwon ve J. Kim, "Convergence Enhanced Genetic Algorithm with Successive Zooming Method for Solving Continuous Optimization Problems", Computers and structures, 81, s. 1715-1725, 2003.

[13] R. Chelouah ve P. Siarry, "Tabu Search Applied to Global Optimization", European Journal of Operational Research, 123, s. 256-270, 2000.

[14] W. Sun ve Y. Dong, "Study of Multiscale Global Optimization Based on Parameter Space Partition", Journal of Global Optimization, 49, s. 149-172, 2011.

[15] C. Chen, K. Chang ve S. Ho, "Improved Framework for Particle Swarm Optimization: Swarm Intelligence with Diversity-guided Random walking", Expert Systems and Applications, 38(10), s. 12214-12220, 2011.

[16] Y.-T. Kao ve E. Zahara, "A Hybrid Genetic Algorithm and Particle Swarm Optimization for multimodal functions", Applied Soft Computing, 8, s. 849-857, 2008.