Parçacık Sürü Optimizasyonu
Orçun İzmirli, 16253034

Öz

Araştırmada, sezgisel arama algoritmalarına değinerek PSO algoritması anlatılmış, PSO algoritmasında parçacık katsayısına (c1) bağlı değişimdeki sonuçlar gözlemlenmiştir. Araştırmada matlab üzerinden yazılan kodlarla bir minimizasyon problemi gerçekleştirilmiş, c1 dışındaki parametreler ve değişkenler sabit tutulmuştur. Deneylerde, c1 değeri sırasıyla 0.001, 0.1, 0.5, 2, 5, 7 ve 10 değerlerini almıştır. Sonuç olarak c1 değerinin oluşturulan uzayın sınırlarına yaklaşması durumunda optimizasyonun kötüleştiği veya gerçekleşmediği, c1 değerinin minimize edildiği durumda ise hassasiyetin arttığı ancak bununla birlikte yeterli iterasyon (tekrar) sayısına ulaşılmadığında sonucun optimumdan uzak olduğu gözlemlenmiştir. Kısıtlı iterasyon sayısında en iyi sonucu 2 hemen ardından 0.5 ve 0.1 değeri vermiştir. İterasyon sayısı maksimum tutulduğunda c1 parametresinin minimum değerinin en iyi sonucu vereceği gözlemlenmiştir.

Anahtar Kelimeler: PSO, Parçacık Katsayısı, Optimizasyon, Algoritma, Matlab

Giriş

Bilgisayar bilimleriyle ilgili çalışmalar yapan insanlar, karşılaştıkları problemleri çözmek için farklı farklı algoritmalar geliştirmişlerdir. Kullanılan bilgisayarların gelişmesiyle birlikte, gün geçtikçe daha büyük problemlere çözüm aranmaya başlanmıştır fakat bazı problemlerde şuanki geliştirilmiş olan algoritmalarla optimum çözümü elde edebilmek neredeyse imkansızdır. Diğer bir durumda ise optimum çözüme ulaşmamızı hedefleyen algoritmalar bazı durumlarda çok maliyetli olabilmektedir. Böyle durumlarda problemin yaklaşık çözümünü bulabilmek için belirli algoritmalar geliştirilmiştir. Bu algoritmaların genel adı sezgisel arama algoritmalarıdır.

Sezgisel arama algoritmaları, bize problemin optimumum çözümünü vermeyi garanti etmez ancak problemin optimum çözümüne yaklaşmayı hedefler, yani problemi optimize eder. Sezgisel arama algoritmalarının birçok çeşidi bulunmaktadır ve her algoritma her problemde aynı etkiyi göstermemektedir. Bir problemde çok iyi performans gösteren algoritma, başka bir problemde çok kötü bir performans gösterebilir. Algoritmanın performansı ne kadar kısa sürede ne kadar iyi çözüme ulaşabildiği ile ölçülmektedir.

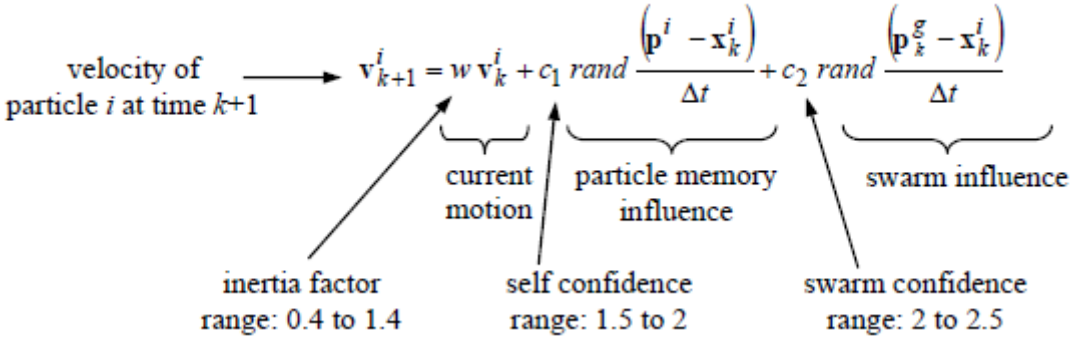
Sezgisel arama algoritmaları kategorilere ayrılacak olursa; genel amaçlı, biyoloji tabanlı, fizik tabanlı, müzik tabanlı, kimya tabanlı, sosyal tabanlı ve sürü tabanlı sezgisel arama algoritmaları olmak üzere altı farklı grupta değerlendirilmektedir. Sürü tabanlı arama algoritmaları, doğada sürü halinde yaşayan canlıların hareketlerinden esinlenerek oluşturulmuş optimizasyon algoritmalarıdır. Birçok sürü algoritması bulunmaktadır. Bunlardan en popülerleri; Yapay Arı Kolonisi(Articial Bee Colony), Karınca Kolonisi(Ant Colony), Ateş Böceği Algoritması (Firefly Algorithm) ve Parçacık Sürü Optimizasyonu 'dur(Particle Swarn Optimization).

Sürü algoritmalarından Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden kuş, balık ve böceklerden esinlenerek Dr. Kennedy ve Dr. Eberhart tarafından 1995 yılında geliştirilmiş sezgisel bir optimizasyon yöntemidir. PSO geliştirilirken, sürü halinde gezen hayvanların davranışları incelenerek; çevrelerine uyum sağlayabilme, zengin yiyecek kaynaklarını bulabilme ve avcılardan kaçabilme gibi gereksinimleri karşılayabilmek için 'bilgi paylaşma' yaklaşımında oldukları gözlemlenmiştir. Bilgi paylaşımı sayesinde, sürüler yiyecek ararken ya da avcıdan kaçarken, hedefe en yakın olan sürü elemanını takip ederler, hızlarını ve konumlarını en başarılı elemana göre güncellerler. PSO'da sürüdeki her bireye parçacık (çözüm) denir, parçacıklardan oluşan popülasyona da sürü denir.

Yöntem

Parçacık Sürü Optimizasyonu, hayvanlar arasındaki sosyal etkileşimden esinlenerek bulunan bir optimizasyon yöntemidir. PSO algoritması, maksimizasyon ve minimizasyon işlemlerinde kullanılmaktadır. Algoritmada, parçacık adı verilen bireyler arasındaki sosyal etkileşimi kullanarak, arama uzayındaki bireyleri, adaptif olarak en anlamlı bölgeye doğru yönlendirir. PSO 'da her parçacığın bir hız değeri ve sunduğu bir çözüm vardır. PSO, bir grup rastgele çözümle (parçacık) başlatılır ve güncellemelerle en uygun çözümü bulmaya çalışır. Her tekrarda (iterasyonda), parçacık konumları en iyi iki parçacığa göre güncellenir. Bunlardan ilki; o iterasyonda sürüdeki en iyi uygunluk değerine sahip olan parçacıktır. Bu parçacık yerel en iyi olarak hafızada tutulur. İkincisi; sürüdeki o ana kadar tüm parçacıklar arasında elde eden en iyi uygunluk değerine sahip olan parçacıktır. Bu parçacık global en iyi olarak hafızada tutulur.

Ardından global en iyi ve yerel en iyi parçacıkları kendi arasında kıyaslanıp, en iyi uygunluk değerine sahip olan küresel en iyi olarak adlandırılır. Her iterasyonun son adımında, küresel en iyi parçacığın değerlerinden faydalanılarak hız değeri yeniden hesaplanır ve konumlar güncellenir. Yeni hız ve konumlarla bir sonraki iterasyona geçiş yapılır. Algoritma, optimum sonuca ulaştığında veya daha önce belirlenmiş olan iterasyon sayısına ulaşıldığında sonlandırılır. Sonuç olarak, son iterasyonda elde edilen küresel en iyi parçacık gösterilir. İstenirse her iterasyondaki küresel en iyinin değerinden oluşan bir grafik çizilir.



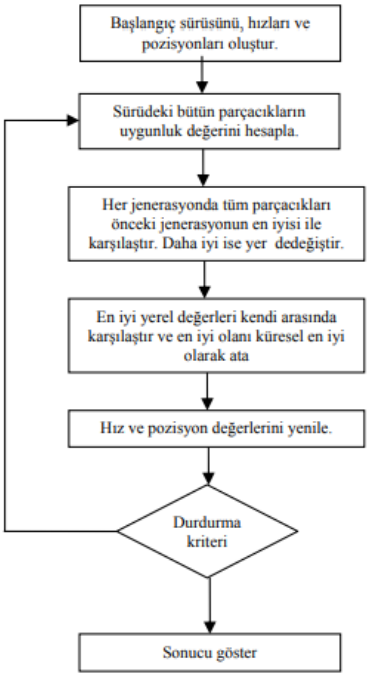
Şekil 1. *Hız güncelleme formülü*

Yukarıdaki şekilde, i parçacığının k+1 zamanındaki hız formülü (v) verilmiştir. w eylemsizlik faktörünü belirtir. C1 parçacığın kendi katsayısı, C2 ise sürüye bağlı katsayısıdır. pi parçacığın en iyi noktası, pgk ise küresel en iyi noktadır. Bu değerler rastgele üretilen sayılarla çarpılarak toplanır ve her parçacığın bir sonraki hareketi ayrı ayrı belirlenir. Bu hareket, xi(k+1) = xi(k) + v(k+1) konum formülü kullanılarak yeni konumu günceller.

Evren - Örneklem

Araştırmada, matlab kodları kullanılarak minimizasyon problemlerini çözmek üzere bir PSO algoritması yazılmıştır. Algoritmada oluşturulan fonksiyon, parametre olarak; problemin alt ve üst sınırı, her sürüde bulunan sürü sayısı, problemin boyutu, eylemsizlik katsayısı, parçacığın kendi katsayısı, parçacığın sürüye bağlı katsayısı ve yapılacak maksimum iterasyon sayısını alır.

Algoritmada öncelikle rasgele üretilmiş başlangıç pozisyonu ve hızı olan parçacıklardan oluşmuş başlangıç sürüsü üretilir. Parçacıkların uygunluk değeri hesaplanır ve her parçacığın en iyi çözümü bulunur. Ardından parçacıklar arasından küresel en iyi seçilir ve küresel en iyiye göre hız ve pozisyon değerleri güncellenir. Optimum sonuç (minimizasyon problemlerinde 0) bulunmamışsa ikinci iterasyona geçilir ve işlemler tekrar edilir.



Şekil 2. *PSO Akış Diyagramı*

Araştırmada parametre olarak; alt sınır -10, üst sınır 10, boyut 5, eylemsizlik katsayısı 0.8, sürüye bağlı katsayı 2 olarak girilmiş ve maksimum iterasyon sayısı 30'a indirgenmiştir. Araştırmada parçacığın katsayısındaki değişime bağlı olarak problemin çözümleri gözlemlenmiştir. Rassal adımların sonucu etkilememesi için rassal adım büyüklüğü 0.35 olarak belirlenmiştir.

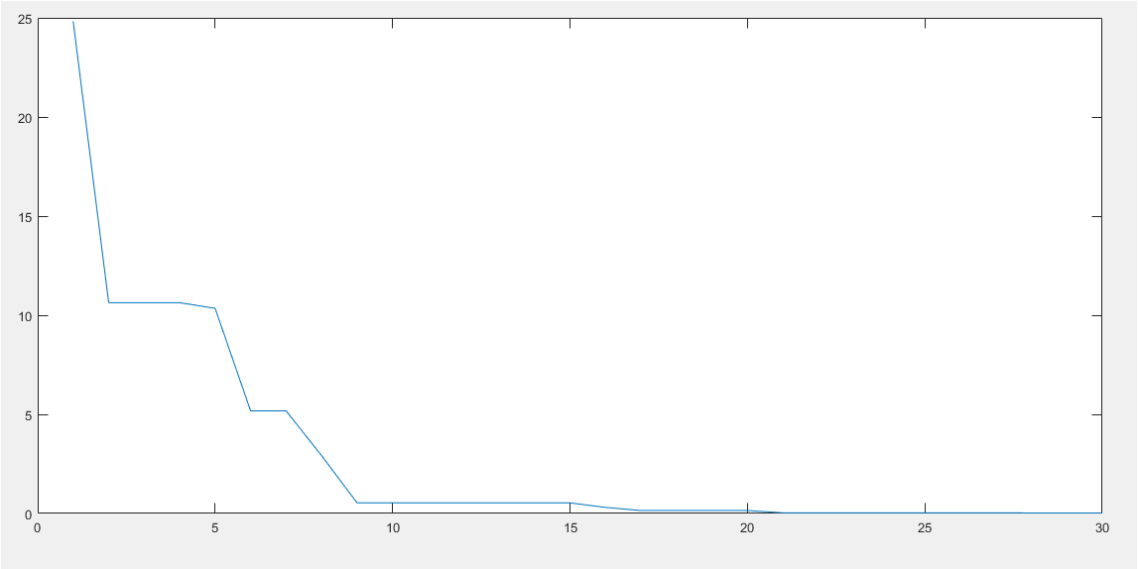
Veri Toplama Araçları

Gözlem tekniği kullanılarak, matlab kodlarıyla oluşturulan yapay bir ortamda, sınırları belirlenmiş bir şekilde veriler toplanmıştır. Verileri analiz edebilmek için yapılan her deneyin sonuç grafiği ve deney sonundaki küresel en iyi parçacık kaydedilmiştir. Deneyde parçacık katsayısı değeri (c1) sırasıyla; 0.001, 0.1, 0.5, 2, 5, 7 ve 10 olarak alınmıştır. Sonuçlardan elde edilen grafikler değerlendirilmiştir.

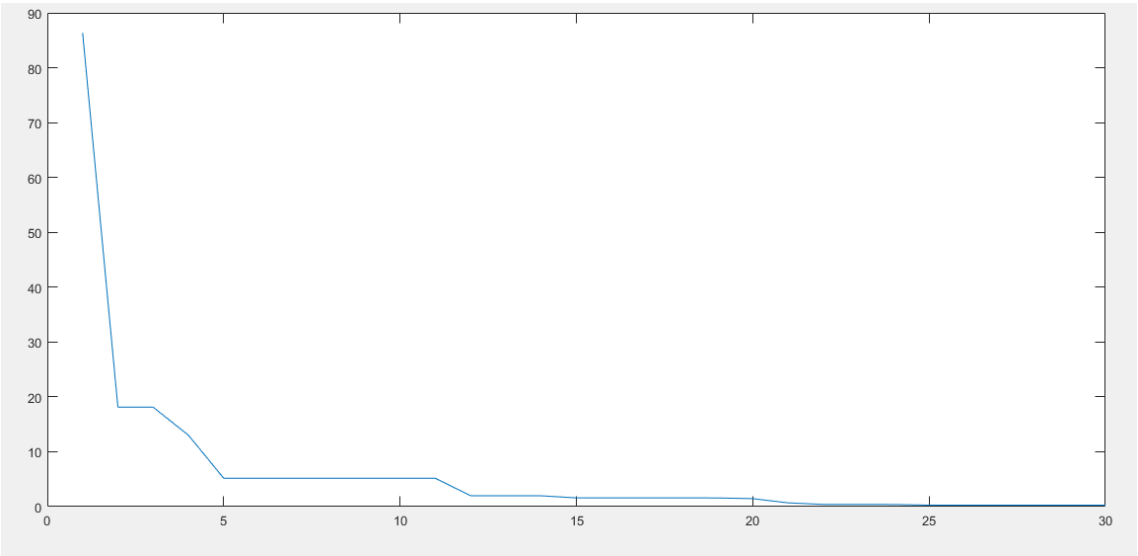
Verilerin Analizi

Deney sonuçları değerlendirildiğinde, sabitlenmiş parametrelerle birlikte c1 = 2 değerinin küçük bir farkla en iyi sonuçları verdiği gözlemlenmiştir. Hemen ardından en iyi sonucu c1 = 0.1 ve c1 = 0.5 değerleri aralarında anlamlı bir fark olmaksızın vermektedir. Deneyde c1 = 10 değerinde herhangi bir değişim olduğu gözlemlenmemiş, onun ardından en kötü sonucu c1 = 7 değeri vermiştir.

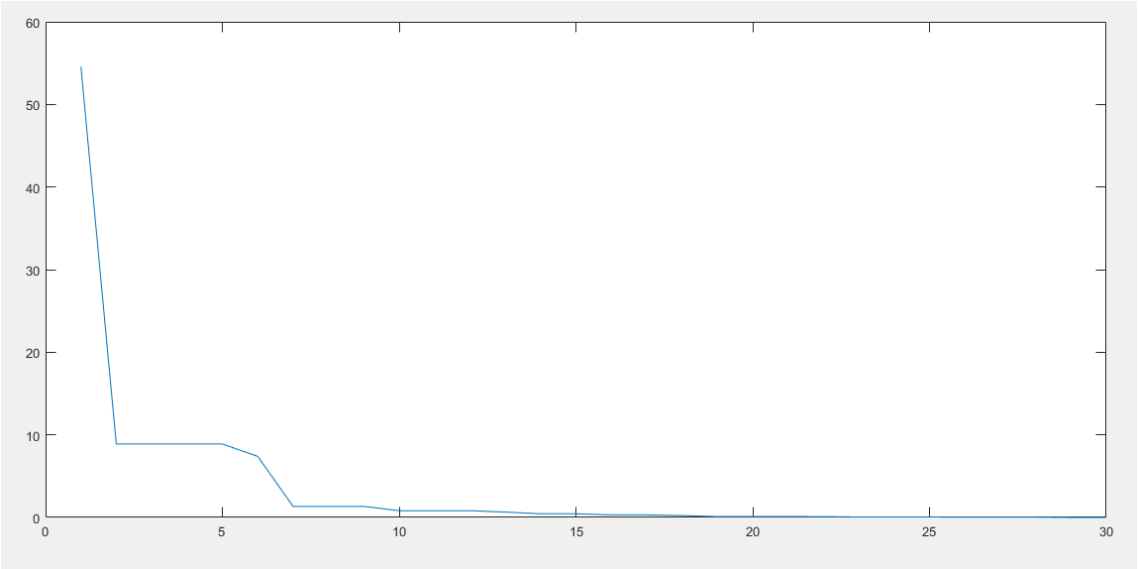
Bulgular



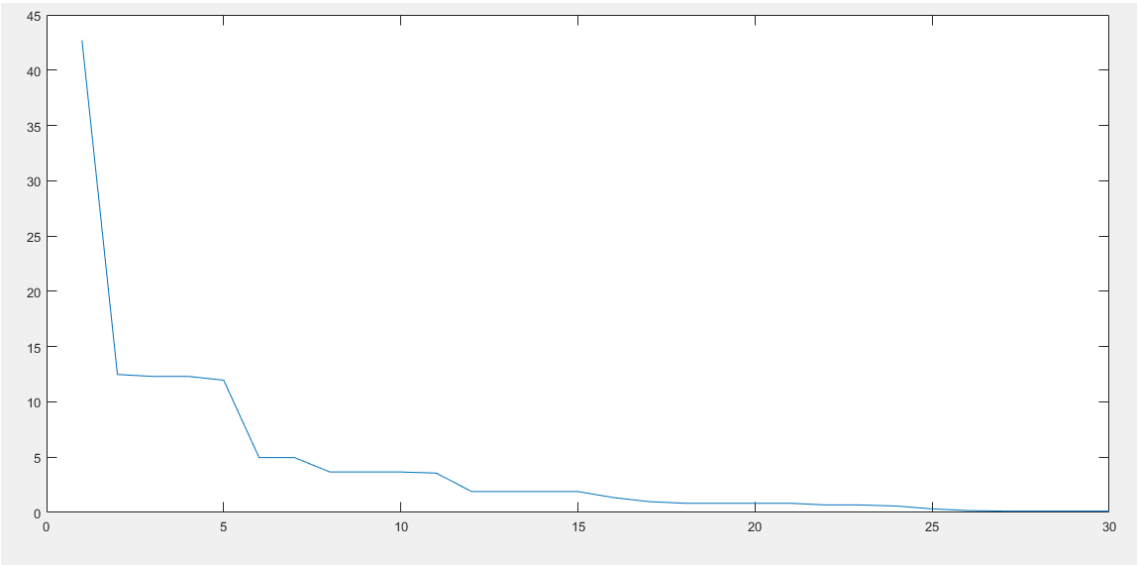
Grafik 1. *C1 = 0.001 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.0264)*



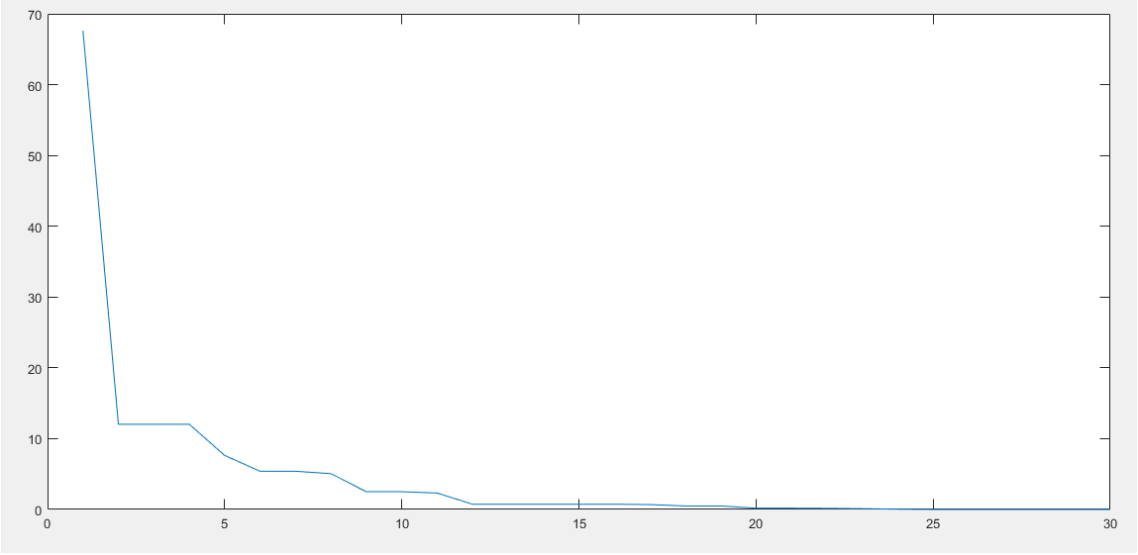
Grafik 2. *C1 = 0.001 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 0.2680)*



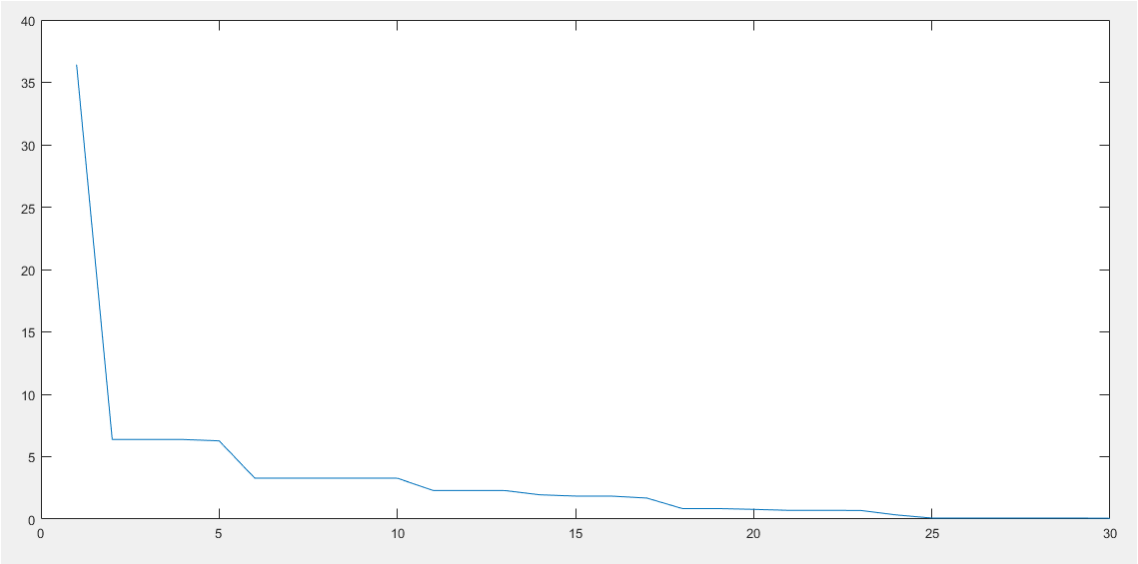
Grafik 3. *C1 = 0.1 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.0031)*



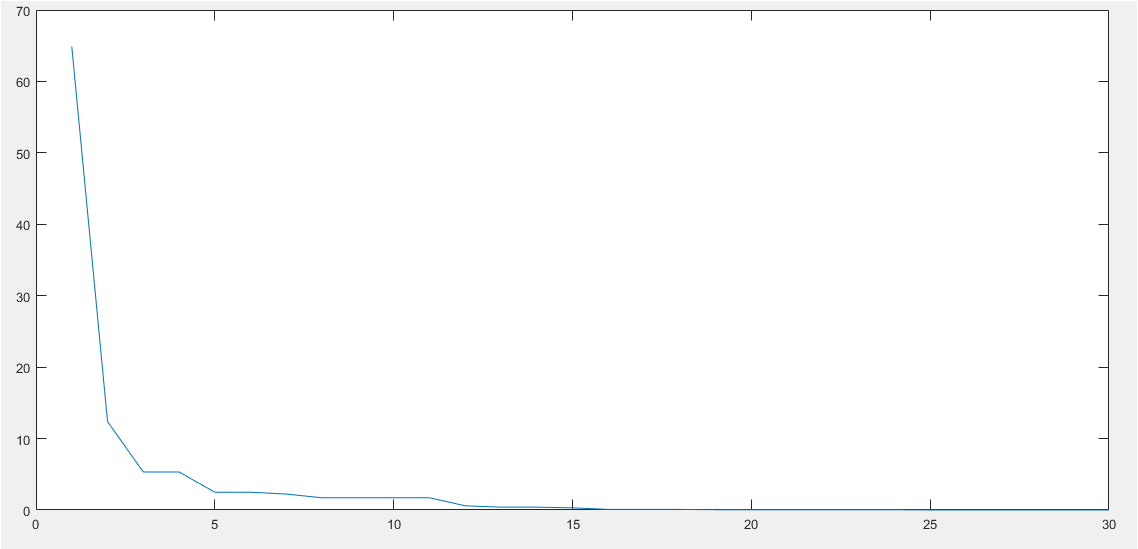
Grafik 4. *C1 = 0.1 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 0.1283)*



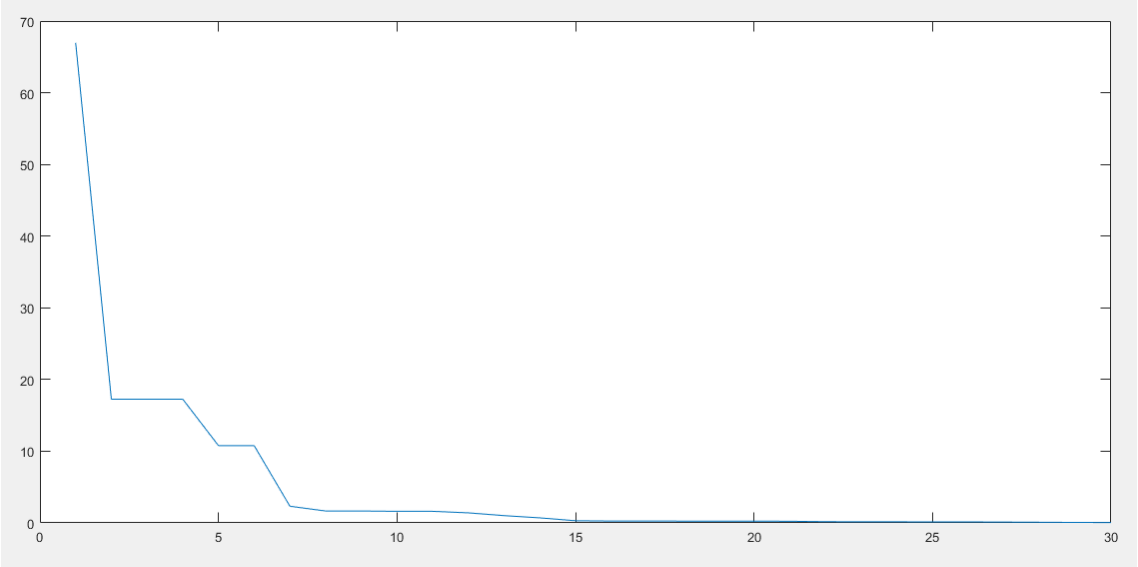
Grafik 5. *C1 = 0.5 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.0030)*



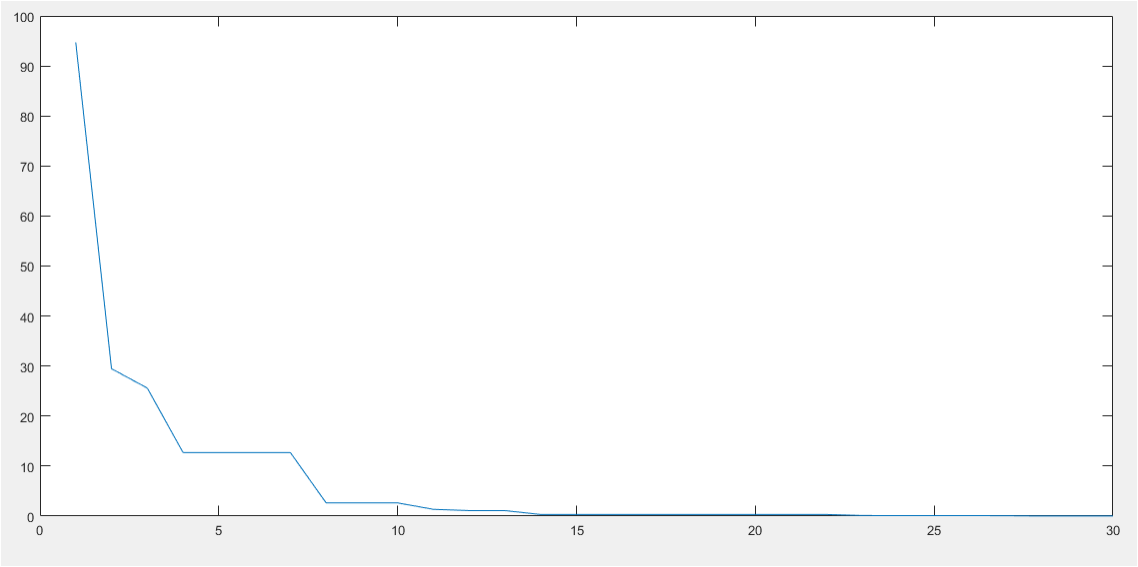
Grafik 6. *C1 = 0.5 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 0.0637)*



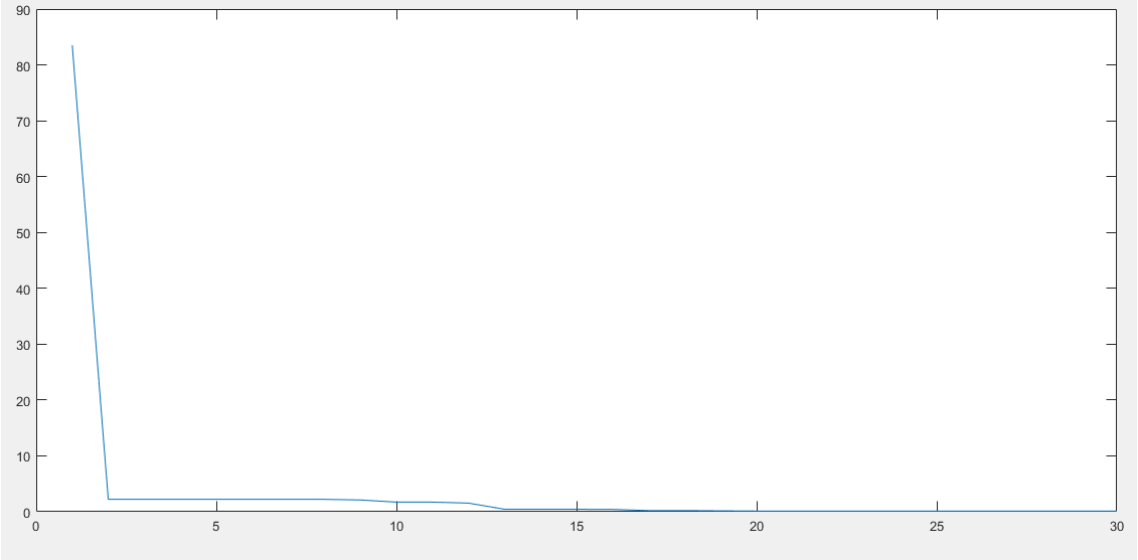
Grafik 7. *C1 = 2 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.0018)*



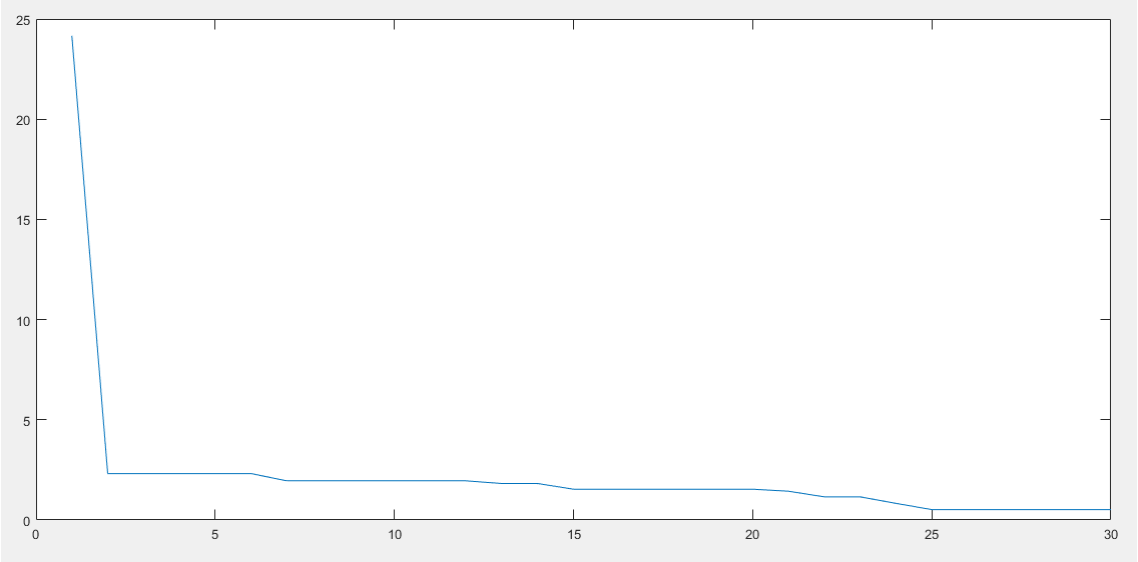
Grafik 8. *C1 = 2 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 0.0534)*



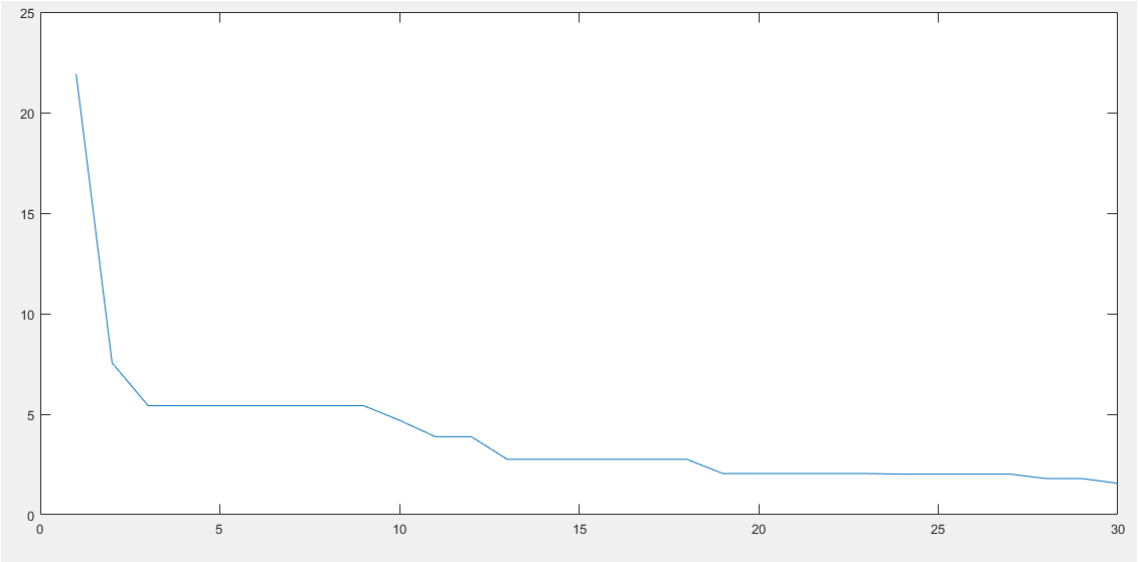
Grafik 9. *C1 = 5 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.0156)*



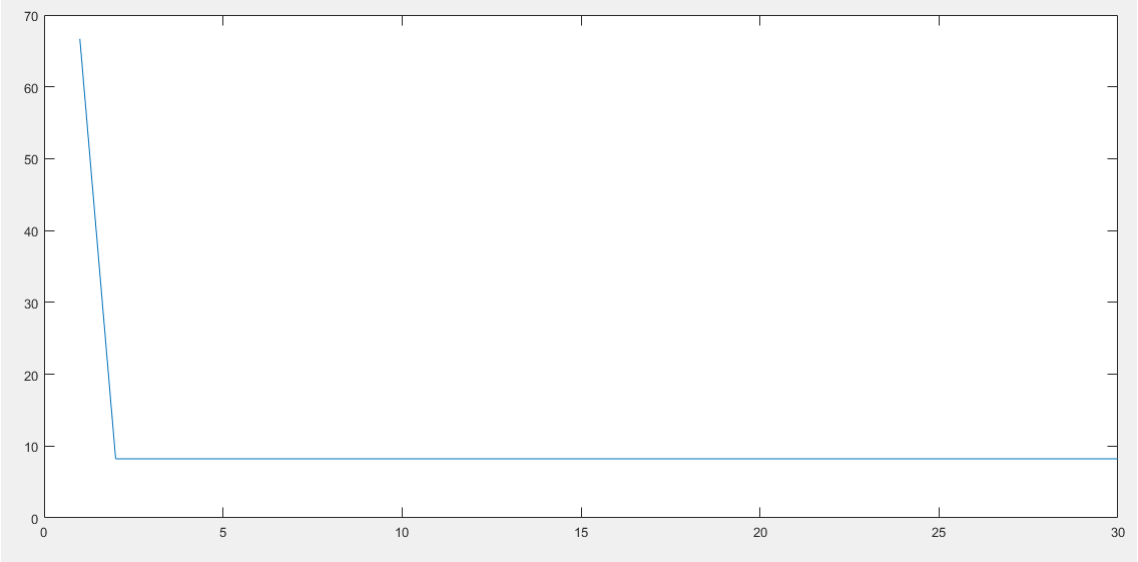
Grafik 10. *C1 = 5 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 0.1188)*



Grafik 11. *C1 = 7 Değeri En İyi Sonuç Grafiği (en iyi sonuç = 0.5197)*



Grafik 12. *C1 = 7 Değeri En Kötü Sonuç Grafiği (en iyi sonuç = 1.5636)*



Grafik 13. *C1 = 10 Değeri Sonuç Grafiği (en iyi sonuç = 8.2182)*

Deneylerin ışığında grafikler incelendikten sonra, c1 değeri problemin sınır değerlerine yaklaştığında optimizasyon sisteminin iyi çalışmadığı gözlemlenmiştir. Problemde, c1 değeri için en iyi değerin 2 olduğu gözlemlenmesine rağmen, iterasyon sayısının yeterince artmasıyla birlikte c1 = 0.001 değerinin en iyi sonucu vereceği gözlemlenmiştir. Bunun nedeni, c1 = 0.001 değeri ile hesaplanan hız formülünün sonucunun diğer değerlere göre daha küçük sonuçlar vermesi ve buna bağlı olarak çözüm hassasiyetini artırmasıdır.

Tartışma, Sonuç ve Öneriler

Araştırmada, Parçacık Sürü Optimizasyonu algoritmasının maksimizasyon ve minimizasyon problemlerinde kullanılabilirliği test edilmiştir. Elde edilen sonuçların ışığında, PSO algoritmasının maksimizasyon ve minimizasyon problemlerinde etkili bir çözüm sunabileceği kanısına varılmıştır. Aynı problem farklı algoritmalar kullanılarak da çözülmediği için PSO algoritmasının bu problem üzerindeki performansı hesaplanamamıştır.

Deney sonucu gözden geçirildiğinde, c1 değerinin küçülmesiyle birlikte problemin hız değerinin daha küçük sonuçlar vermesinden kaynaklı olarak, optimum sonuç için minimum c1 değeri ile en yüksek iterasyon sayısının kullanımının işe yarayacağı gözlemlenmiştir. c1 değerinin minimum düzeyde seçilmesinin dezavantajı ise, problemin maliyetini artırmasıdır. Problemin maliyetinin artması, algoritmanın çözümünü geciktirmektedir. Bu durumda, hızlı bir şekilde, optimuma yakın sonuçlar elde etmek istiyorsak; c1 değerini sınır değerlerinin ortalarında, adım büyüklüğünü de çok küçültmeyecek bir değer olarak seçmeliyiz.

Not

Elde edilen sonuçlar bu deney özelinde olup, farklı parametrelerle oluşturulmuş bir deney evreninde farklı sonuçlar elde edilmesi mümkündür.

Kaynakça

Altınöz, T. Ö., & Yılmaz, E. A. (tarih yok). Parçacık Sürü Optimizasyonunda Yeni Bir Birey Davranış Biçimi Önerisi.

Aydın, İ. (tarih yok). *PARÇACIK SÜRÜ OPTİMİZASYONU BMÜ-579 METASEZGİSEL YÖNTEMLER.* https://docplayer.biz.tr/106712303-Parcacik-suru-optimizasyonu-bmu-579-metasezgisel-yontemler-yrd-doc-dr-ilhan-aydin.html adresinden alındı

Çavuşoğlu, M. A., Karakuzu, C., & Şahin, S. (2010). Parçacık Sürü Optimizasyonu Algoritması ile Yapay Sinir Ağı Eğitiminin FPGA Üzerinde Donanımsal Gerçeklenmesi. *Polietnik Dergisi*, 83-92.

Ergüder, H. (2019, 04 14). *Parçacık Sürü Optimizasyonu*. Medium: https://medium.com/@hamzaerguder/par%C3%A7ac%C4%B1k-s%C3%BCr%C3%BC-optimizasyonu-24e01beec438 adresinden alındı

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. 1942-1948. Washington, DC, ABD.

Özdemir, M. (2017). Particle Swarm Optimization for Continuous Function Optimization. *International Journal of Applied Mathematics, Electronics and Computers*, 47-52.

Özsağlam, Y. M., & Çunkaş, M. (2008). Optimizasyon Problemlerinin Çözümü için Parçaçık. *Politeknik Dergisi*, 299-305.

Parçacık Sürü Optimizasyonu Algoritması (PSO). (2019, 03 30). Teslanın Kutusu: https://teslaninkutusu.wordpress.com/2019/03/30/parcacik-suru-optimizasyonu-algoritmasi-pso/ adresinden alındı

Pektaş, M. (2019, 04 19). *Parçacık Sürü Optimizasyonu (PSO)*. Medium: https://medium.com/deep-learning-turkiye/par%C3%A7ac%C4%B1k-s%C3%BCr%C3%BC-optimizasyonu-pso-1402d4fe924a adresinden alındı

Sezgisel algoritma. (2020, 12 5). Wikipedia: https://tr.wikipedia.org/wiki/Sezgisel_algoritma adresinden alındı

Sürü davranışı. (2021, 01 18). Wikipedia: https://tr.wikipedia.org/wiki/S%C3%BCr%C3%BC_davran%C4%B1%C5%9F%C4%B1#:~:text=Par%C3%A7ac%C4%B1k%20s%C3%BCr%C3%BC%20optimizasyonu%20s%C3%BCr%C3%BClerle%20ilgili,James%20Kennedy%20ve%20Russell%20C.&text=Her%20zaman%20aral%C4%B1%C4%9F%C4%B1nda%20par%C3%A7ac adresinden alındı