

Derin Q Ağları Tabanlı Parçacık Sürü Optimizasyonu

Özlem TÜLEK^{1*}, İhsan Hakan SELVİ²

^{1,2} Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilişim Sistemleri Mühendisliği, Sakarya, Türkiye
^{*}ozlemtulek@ogr.sakarya.edu.tr, ²ihselvi@sakarya.edu.tr

(Geliş/Received: 13/06/2023;

Kabul/Accepted: 26/08/2023)

Öz: Günümüzde, yapay zekâ ve makine öğrenmesi teknolojilerindeki hızlı gelişmeler, optimizasyon problemlerinin çözümüne farklı ve yenilikçi yaklaşımlar getirmiştir. Bu yöntemler, problem çözümünde klasik optimizasyon tekniklerine alternatif yaklaşımlar sunmaktadır. Optimizasyon problemlerinin çözümünde sıklıkla kullanılan metasezgisel algoritmaları makine öğrenmesi teknikleriyle birlikte kullanmak güçlü bir potansiyel sunmaktadır. Bu çalışmada doğa esinli bir metasezgisel algoritma olan parçacık sürü optimizasyonu ile bir makine öğrenmesi yöntemi olan pekiştirmeli öğrenmeyi birlikte kullanan bir model önerilmiştir. Önerilen model 9 tane kıyaslama problemi kullanılarak 50 ve 100 boyut için test edilmiştir. Elde edilen sonuçlar pekiştirmeli öğrenmenin PSO'nun yakınsama ve küresel keşif yeteneklerini geliştirmek için büyük bir potansiyel sunduğunu göstermektedir.

Anahtar kelimeler: Pekiştirmeli öğrenme, DQN, metasezgisel optimizasyon, parçacık sürü optimizasyonu.

Deep Q Networks Based Particle Swarm Optimization

Abstract: Today, rapid developments in artificial intelligence and machine learning technologies have brought different and innovative approaches to the solution of optimization problems. These methods offer alternative approaches to classical optimization techniques in problem solving. Using metaheuristic algorithms, which are frequently used in solving optimization problems, together with machine learning techniques offers a strong potential. In this study, a model that uses particle swarm optimization, which is a nature-inspired metaheuristic algorithm, and reinforcement learning, which is a machine learning method, is proposed. The proposed model is tested for 50 and 100 dimensions using 9 comparison problems. The results show that reinforcement learning offers great potential to enhance the convergence and global exploration capabilities of PSO.

Key words: Reinforcement learning, DQN, metaheuristic optimization, particle swarm optimization.

1. Giriş

Optimizasyon problemleri, birçok farklı alanda karşılaşılan önemli matematiksel problemlerdir. Bu problemler, verilen kısıtlar altında, bir hedef fonksiyonunun en büyüklmesi veya en küçüklmesi ile ilgilidir. Örneğin, bir üretim sürecinde malzeme kullanımının minimize edilmesi, bir taşıma planının maliyetinin minimize edilmesi veya bir yatırım portföyünün getirisinin maksimize edilmesi gibi durumlar optimizasyon problemlerine örnek olarak verilebilir. Optimizasyon problemlerinin çözümü, matematiksel modeller kullanılarak gerçekleştirilmektedir. Bu modeller, problemin özelliklerine ve hedefine göre tasarlanmakta ve matematiksel ifadelerle gösterilmektedir. Bu problemlerin çözümü, matematiksel modeller ve optimizasyon algoritmaları kullanılarak gerçekleştirilmektedir [1].

Optimizasyon problemlerinin çözümü için pekiştirmeli öğrenme ve meta-sezgisel algoritmaların kullanımı, karmaşık problemleri daha etkili bir şekilde çözmeye yardımcı olabilmektedir [2]. Bu yöntemlerin, farklı disiplinlerdeki araştırmacılar tarafından aktif olarak kullanılması ve geliştirilmesi, optimizasyon problemlerinin çözümünde yeni bir çağ açabilir.

Son yıllarda doğa esinli algoritmalarla ilgili yoğun araştırmaların yapıldığı görülmektedir. Doğada gözlemlenen canlı sürülerinin hareketleri, evrim teorisi, insan vücudu ya da fizik kurallarından ilham alan bu algoritmalar, birden çok disiplinde kullanılmaktadır. Basit bir matematiksel model kullanan, kolay anlaşılır ve uygulanabilir olan parçacık sürü optimizasyonu (PSO) da bu algoritmalarından biridir. 1995 yılında Kennedy ve Eberhart tarafından önerilen algoritma, kuşların gökyüzünde yapmış oldukları akınlardan esinlenilerek geliştirilmiştir [3]. PSO, çok sayıda parçacığı aynı anda kullanabilen, çok boyutlu problemlerin çözümü için hızlı ve etkili bir yöntemdir. PSO'da çözüm uzayında arama yapan bireyler parçacık; parçacıklardan oluşan popülasyon ise sürü olarak adlandırılmıştır. PSO'da parçacıklar çözüme en yakın oldukları konum ile tüm sürü içinde elde

*Sorumlu Yazar: ozlemtulek@ogr.sakarya.edu.tr. Yazarların ORCID Numarası: ¹ 0000-0003-4466-8515, ² 0000-0002-8837-2137

edilmiş en iyi konumu takip ettiği için, global optimuma ulaşma olasılığı yüksektir. Bu nedenle, PSO, doğa esinli algoritmalar arasında en popüler ve sık kullanılan yöntemlerden biridir.

PSO'nun bazı durumlarda yerel minimumlara yakalanabildiği bilinmektedir. Bu durumda, çözüm alanında daha iyi sonuçlar elde etmek için global arama stratejileri kullanılmalıdır. Ayrıca, bu yöntemin bazı durumlarda yakınsama hızı yavaşlayabilmektedir. Bunun nedeni, parçacıkların birbirlerinden farklı hız ve yönlerde hareket ederek arama alanının genişliğini keşfetmeleri ve daha iyi çözümlere ulaşmaları gereken durumlarda zaman kaybedebilmeleridir.

Basitliği, hızı ve yüksek performansı sebebiyle PSO'nun başarımını arttırmak üzere pek çok araştırma yapılmıştır. Bu araştırmalardan bazıları PSO için uygun parametre değerleri seçimiyle ilgilidir. PSO'nun performansında parametre seçimi önemli bir role sahip olduğu için bu konuda çok sayıda çalışma yapılmıştır [4-6]. PSO'nun performansını arttırmak için yapılan diğer bazı çalışmalar [7-13] ise diğer algoritmalar ile PSO'yu hibrit olarak kullanmak üzerinedir. Hibrit algoritmalar, bir yöntemin zayıf yönlerini bir başka yöntemin güçlü yönleriyle telafi ederek, daha iyi sonuçlar elde etmeyi amaçlamaktadır. PSO ile diğer algoritmaların birlikte kullanıldığı çalışmalar incelendiğinde bu çalışmalarda standart PSO'ya göre daha iyi sonuçların elde edildiği ve yerel minimumlara takılma gibi birtakım sorunların üstesinden gelindiği görülmüştür.

Pekiştirmeli öğrenme, makine öğrenmesi alanında önemli bir tekniktir. Bu yöntem, bir makinenin bir dizi adımı takip ederek bir görevi öğrenmesini sağlamaktadır. Pekiştirmeli öğrenme, bir ajanın belirli bir ortamda belirli bir davranışı gerçekleştirerek ödüller kazanmasını hedeflemektedir. Bu ödüller, ajanın doğru davranışları öğrenmesini teşvik etmekte ve yanlış davranışları engellemektedir. Bu teknik, genellikle robotik, oyunlar, otomatik pilotlu araçlar, doğal dil işleme ve finansal uygulamalar gibi alanlarda kullanılmaktadır. Pekiştirmeli öğrenme, makine öğrenmesi için önemli bir adımdır ve yapay zekânın insan benzeri öğrenme ve karar verme yeteneklerini geliştirmede önemli bir rol oynamaktadır [14]. Pekiştirmeli öğrenme alanında kullanılan bir derin öğrenme modeli olan Derin Q-ağları (DQN) öğrenme sürecinde bir ajanın, çevreyle etkileşime girerek en yüksek ödülü kazanması için optimal eylem seçimini öğrenmesine olanak tanımaktadır. DQN, Q-öğrenme algoritmasının derin öğrenme ağı ile genişletilmesiyle oluşturulmuştur. Bu sayede daha büyük ve karmaşık problemleri çözmekte etkili olabilmektedir.

Son yıllarda yapılan çalışmalarda PSO'nun performansı arttırmak üzere pekiştirmeli öğrenmenin kullanıldığı görülmektedir. Yin vd. [15] yaptıkları çalışmada PSO'nun katsayılarını kontrol etmek için bir ağ tasarlayarak PSO yakınsamasını geliştirmek için pekiştirmeli öğrenmeye dayalı bir parametre uyarlama yöntemi önermişlerdir. Meng vd. [16] pekiştirmeli öğrenme tarafından yönlendirilen çok stratejili kendi kendine öğrenme mekanizmasını önermişlerdir. Wang vd. [17] ise PSO'nun geniş ölçekli optimizasyon problemlerindeki performansını arttırmak için PSO'nun popülasyon çeşitliliğini iyileştirmeyi hedeflemişler ve seviye tabanlı bir popülasyon yapısı oluşturmuşlardır. Önerdikleri algoritmanın arama verimliliğini artırmak için ise seviye numarası kontrolü yapmak üzere bir pekiştirmeli öğrenme stratejisi kullanmışlardır. Lu vd. [18] yaptıkları çalışmada pekiştirmeli öğrenme stratejisi yoluyla PSO parçacıklarının davranışına rehberlik etmek için geçerli geçmiş bilgisinden yararlanmaya çalışan bir model önermiştir. Wu ve Wang [19] güncelleme işlevindeki düzgün dağılmış rasgele sayıyı, seçilen bir normal dağılımdan üretilen rasgele bir sayı ile değiştirerek yakınsamadaki PSO'yu geliştirmek için bir pekiştirmeli öğrenme stratejisi geliştirmiştir.

Bu çalışmada, bir pekiştirmeli öğrenme algoritması olan DQN ile PSO birbirleriyle entegre edilerek yeni bir model önerilmiştir. Önerilen model ile parçacık sürü optimizasyonunun tüm avantajları korunurken, aynı zamanda pekiştirmeli öğrenmenin esnekliği ve adaptif öğrenme yeteneklerini de kullanabilmek amaçlanmıştır. Modelin ana fikrinde lokal minimumlardan kaçınmak için yalnızca rasgeleliliği kullanan parçacıklar değil, bunun yanında ona sunulan hareket seçeneklerinden en uygununu tercih etmeyi öğrenerek çözüm uzayında arama yapan parçacıklar düşünülmüştür.

2. Materyal ve Metot

Bu çalışmada doğa esinli optimizasyon algoritması olan parçacık sürü optimizasyon (PSO) algoritmasının lokal minimumlarda takılmasını önleyerek daha iyi çözümler elde etmesini sağlayan pekiştirmeli öğrenme stratejisine dayanan bir model geliştirilmiştir. Pekiştirmeli öğrenme algoritması olarak Derin Q Ağları (DQN) kullanılmıştır. Bu bölümde PSO ve DQN algoritmaları incelendikten sonra önerilen model hakkında ayrıntılı bilgi verilmiştir.

2.1 Parçacık sürü optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), çok sayıda karmaşık optimizasyon problemini çözmek için kullanılan güçlü bir metasezgisel algoritmadır. Algoritmada çözüm uzayı kuş sürüsü, her bir kuş (parçacık) da olası çözümleri temsil etmektedir. Daha yüksek parçacık sayısı, daha iyi sonuçlar elde edilmesine yardımcı olabilmekte, ancak hesaplama maliyeti de bununla birlikte artmaktadır.

Parçacıkların çözüme ne kadar yakın olduklarını bulmak için uygunluk fonksiyonu kullanılmaktadır. Her iterasyonda tüm parçacıkların uygunluk değeri hesaplanmaktadır. Bir parçacığın çözüme en yakın olduğu konum kişisel en iyi değer (pbest), tüm parçacıklar arasında çözüme en yakın olunan konum ise global en iyi değer (gbest) olarak adlandırılır. Sürüdeki her parçacık, mevcut konum ve hız bilgisi ile çözüme en yakın olduğu konum olan pbest bilgisine sahiptir. Parçacıkların yeni konumları hesaplanırken mevcut konumlarıyla birlikte pbest ve gbest değerleri de kullanılmaktadır.

Standart PSO algoritmasında tüm sürü başlangıçta arama uzayında keşif yapmaktadır. Arama işleminin sonuna doğru yakınsama durumuna geçilir. PSO algoritmasının başarısında, başlangıç popülasyonunun çeşitliliği algoritma başarımında oldukça etkilidir. Parçacıkların hızları Denklem 1’de belirtilen formülle ve konumları ise Denklem 2’de belirtilen formül ile hesaplanmaktadır:

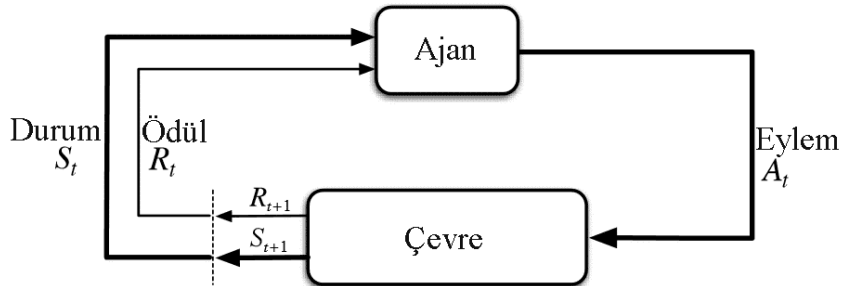
$$v_{ij}(t+1) = w \times v_{ij}(t) + c_1 \times rand \times (p_{ij}(t) - x_{ij}(t)) + c_2 \times rand \times (p_{gj}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

Eşitlik 1’de v_{ij} , i parçacığının j boyutundaki hız değeridir. w eylemsizlik ağırlığıdır. PSO’da eylemsizlik ağırlığı global ve yerel arama arasındaki dengeyi sağlamak için kullanılmaktadır. Eylemsizlik ağırlığı büyük bir değer olduğunda global arama, küçük bir değer olduğunda ise yerel arama yapılması kolaylaşmaktadır. Eylemsizlik ağırlığı ile yerel ve global arama yeteneği ayarlanabilmektedir. C1 ve c2 ivme katsayıları [0,1] aralığında rasgele bir sayıdır. C1 ile parçacıkların kişisel en iyi çözümlerden, c2 ile ise global en iyi çözümden etkilenme düzeyi ayarlanmaktadır. x_{ij} i parçacığının j boyutundaki pozisyonu, p_{ij} i parçacığının j boyutunda şimdiye kadar elde ettiği en iyi değer, p_{gj} ise tüm popülasyonda, j boyutunda elde edilen en iyi değerdir. Bir sonraki adım için hesaplanan hız değeri parametreleri, önceden belirlenmiş olan $[-Vmax, Vmax]$ sınırları içinde tutulmalıdır. Sınırlama işlemi, arama sırasında belirlenen limitlerin aşılmaması ve küresel aramanın kontrolden çıkması için gereklidir.

2.2 Derin q ağırları

Yapay zekâ, bilgisayar sistemlerinin insanların yaptığı zekâsal işlemleri gerçekleştirmek üzere tasarlanmıştır. Bu amaca yönelik olarak kullanılan bir yöntem olan makine öğrenimi, bilgisayar sistemlerinin verilerden öğrenerek deneyimlerini geliştirmesine olanak sağlamaktadır. Pekiştirmeli öğrenme, makine öğrenimi türlerinden biridir. Bu yöntemde, ajan adı verilen öğrenen makine çevreyle etkileşimde bulunarak belirli eylemler gerçekleştirmektedir ve bu eylemler sonucunda (negatif ya da pozitif) ödüller almaktadır. Diğer makine öğrenimi yöntemlerinden farklı olarak, ajana doğrudan hangi eylemleri yapması gerektiği söylenmemektedir. Bunun yerine, ajanın deneme yanılma yoluyla hangi eylemlerin en fazla ödül getirdiğini öğrenmesi beklenmektedir [20]. Şekil 1’de standart bir pekiştirmeli öğrenme mimarisi gösterilmektedir.



Şekil 1. Pekiştirmeli öğrenme çerçevesi.

Pekiştirmeli öğrenme algoritmalarının bir türü olan Derin Q Ağları (DQN), pekiştirmeli öğrenmede kullanılan bir yapay sinir ağı mimarisidir ve Q-öğrenme algoritmasına dayanmaktadır. Q-öğrenme algoritmasında ajan, ödüle giderken her iterasyonda edindiği tecrübeleri toplamda elde edebileceği ödülü (kümülatif ödül) en çoklamak için kullanmaktadır. Bu tecrübeleri Q-tablosu adı verilen bir tabloda tutmaktadır. Q-tablosu, başlangıçta ajanın hiçbir tecrübesi olmadığı için sıfırlarla doludur ve bu yüzden ajan ilk seçimlerinde rastgele hareket etmektedir. Ajan bir ödül elde ettiğinde onu ödüle ulaştıran adımları bilmekte ve elde ettiği tecrübeleri Q-tablosuna yazmaktadır. Q-tablosu, ajanın ortamdaki her durum için en iyi eylemi bulmasına yardımcı olmaktadır. Temel olarak, bir Q-tablosu durum ve eylem çiftlerini bir Q değerine eşlemek için kullanılmaktadır. Q-tablosunu güncellemek için Bellman eşitliğinden yararlanılmaktadır [21].

Q-öğrenme algoritmasında;

$S = [s_1, s_2, \dots, s_n]$ durumlar kümesi,

$A = [a_1, a_2, \dots, a_n]$ ajanın yürütebileceği eylemlerin kümesi,

r_{t+1} , a eyleminin yürütülmesinden elde edilen anlık ödül,

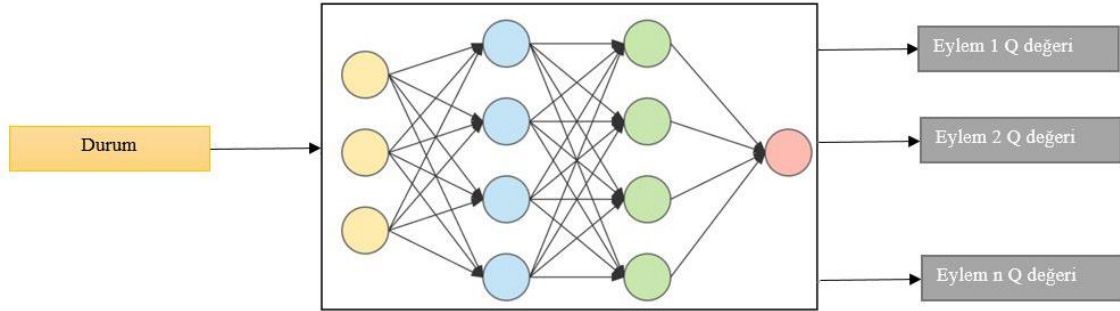
γ , $[0,1]$ aralığında değer alan indirim faktörü,

α , $[0,1]$ aralığında değer alan öğrenme oranı

olmak üzere $Q(s_t, a_t)$, ajanın t zamanda kazandığı toplam kümülatif ödüldür ve Denklem 3 ile hesaplanmaktadır:

$$Q_{yeni}(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3)$$

Derin Q ağları (DQN), pekiştirmeli öğrenme alanında oldukça popüler bir tekniktir. DQN, pekiştirmeli öğrenme ajanlarının hızlı bir şekilde karar vermelerini sağlayan bir derin öğrenme yaklaşımıdır. DQN'ler, Q değerlerini tahmin etmek için bir yapay sinir ağı kullanmaktadır. Şekil 2'de gösterildiği gibi DQN'lerde, yapay sinir ağına bir durum (gözlem) verilir ve ağ, her bir eylem için bir Q değeri üretir. DQN'lerde, bir ajanın bir ortamda en iyi eylemi seçmesine yardımcı olmak için bu Q değerleri kullanılmaktadır.



Şekil 2. Derin q ağı yapısı.

Bir durumun Q değeri, Bellman Eşitliği kullanılarak o durumda alınabilecek eylemler ve sonraki durumlardaki Q değerleri ile ilişkilendirilmektedir. Q değerini hesaplamak için Q değer fonksiyonu kullanılmaktadır. Daha sonra, ajan, verilen bir durumda, elde edebileceği en yüksek Q değerine sahip eylemi seçmektedir. Ajan, bu eylemi gerçekleştirerek ödül kazanmakta ve yeni bir duruma geçmektedir. Bu süreç, ajanın çevreyi keşfederek ve öğrenerek daha yüksek ödüller kazanmasını sağlamaktadır. Ödüller, ajanın yaptığı eylemlerin doğru veya yanlış olduğunu belirlemek için kullanılmaktadır. Ajan, doğru bir eylem gerçekleştirdiğinde pozitif bir ödül alırken, yanlış bir eylem gerçekleştirdiğinde ise negatif bir ödül almaktadır. Bu ödüller, ajanın optimal eylem seçimini öğrenmesini sağlamaktadır.

DQN'ler, bir ajanın bir ortamda hedeflenen bir davranışı öğrenmesine izin vermek için kullanılabilir. Bu, özellikle oyun oynama, robotik ve otonom sürüş gibi alanlarda yaygın bir kullanım alanıdır. DQN'lerin en büyük avantajlarından biri, çevreye doğrudan müdahale etmeden öğrenme yapabilme yeteneğidir. Bu, ajanın gerçek dünyada öğrenmesini sağlamadan önce simüle edilmiş bir ortamda eğitilebileceği anlamına gelmektedir. DQN'ler, öğrenme sürecinde birden fazla gözlem kullanarak eylem seçimini iyileştirebilir.

2.3 Önerilen model

Önerilen modelde ajanın gerçekleştirebileceği ayrıık olarak tanımlanan 5 farklı eylem bulunmaktadır. Her eylem farklı özelliklere sahip bir PSO algoritması şeklinde tanımlanmıştır. Bu eylemler literatürde PSO'nun performansını arttırmak için yapılmış çalışmalar incelenerek oluşturulmuştur [22-25]. Durum, PSO'lardan birinin çalıştırılması sonucunda hesaplanan global en iyi uygunluk değeri (Gbest değeri) olarak tanımlanmıştır.

Ajan bir eylem gerçekleştirdikten sonra uygunluk değeri hesaplanmaktadır. Ajanın alacağı ödül, uygunluk değerinin bir önceki uygunluk değerine göre daha iyi olup olmadığına bakılarak belirlenmektedir. Bu çalışmada kullanılan test fonksiyonları birer minimizasyon problemi olduğundan dolayı ödül/ceza işlevi Denklem 4'e uygun olarak tanımlanmıştır. Bu eşitlikte r ödül, $Gbest(t-1)$ bir önceki iterasyonda elde edilmiş global en iyi uygunluk değeri ve $Gbest(t)$ de mevcut iterasyonda elde edilmiş global en iyi uygunluk değeridir.

$$r = \begin{cases} 1, & Gbest(t) < Gbest(t-1) \\ -1, & \text{diğer durumlarda} \end{cases} \quad (4)$$

Ajan, beklenen toplam ödülü maksimize etmeyi amaçlamaktadır. Derin Q ağlarında, ağa bir durum (gözlem) verilir ve ağ, her bir eylem için bir Q değeri üretir. Bu Q değerleri, bir durumda herhangi bir eylemin ne kadar iyi olabileceğini belirtmektedir. Daha sonra, ajan, verilen bir durumda, elde edebileceği en yüksek Q değerine sahip eylemi seçmektedir. Ajan, bu eylemi gerçekleştirerek ödül kazanmakta ve yeni bir duruma geçmektedir. Bu süreç, ajanın çevreyi keşfederek ve öğrenerek daha yüksek ödüller kazanmasını sağlamaktadır. Önerilen modelde Q değerlerini tahmin etmek için kullanılan yapay sinir ağı şu katmanları içermektedir:

- Giriş katmanı: Derin Q ağlarında, ağa giriş bilgisi olarak durum bilgisi verilmektedir. Önerilen modelde durum bilgisi, 5 ayrıık eylem olarak tanımlanan farklı türdeki PSO algoritmalarından birinin seçilip yürütülmesiyle elde edilen global en iyi uygunluk değeri (Gbest)'dir.
- Gizli katmanlar: İşlemleri gerçekleştiren katmanlardır. Modelde 24 nöronlu iki gizli katman bulunmaktadır. Aktivasyon fonksiyonu olarak ReLU kullanılmıştır.
- Çıkış katmanı: Belli bir sayıda aksiyon olasılıklarını üretmektedir. Modelde çıktı katmanında doğrusal bir aktivasyon fonksiyonu kullanılmaktadır.

Önerilen modelde ajan, her biri farklı özelliklere sahip 5 farklı PSO algoritmasından birini eylem olarak seçmekte ve yürütmektedir. Eylemi gerçekleştirdikten sonra elde edilen uygunluk değeri DQN ağındaki durum olarak tanımlanmıştır. Algoritma aşağıda yer almaktadır:

Önerilen Algoritma

- PSO için başlangıç sürüsü, parçacıkların hızları ve pozisyonları oluşturulur.
- DQN algoritmasının adım metodu gereğince rasgele bir eylem seçilir, bu eylem yürütülür. Eylemin yürütülmesi sonucu elde edilen Gbest değeri durum olarak atanır ve ödül hesaplanır. Bu işlem belirlenen bölüm sayısı kadar tekrarlanır.
- Her bölümde skor, durum gibi parametreler başlangıç değerlerine göre ayarlanarak belirlenen iterasyon sayısı kadar eylem seçimi ve yürütülmesi gerçekleştirilir.
- Derin Q Ağı inşa edilir.
- Ajan, çevre üzerinde belirlenen adım sayısı kadar eğitilir.
- DQN ağının performansını değerlendirmek için test işlemi yapılır. Bu işlem belirlenen bölüm sayısı kadar tekrar edilir. Her bölümde elde edilen ödüller üzerinden ortalama bölüm ödülü, son çıktı olarak yazdırılır.

3. Bulgular

Önerilen algoritmanın performansını değerlendirebilmek için literatürde yaygın olarak kullanılan dokuz test fonksiyonu kullanılmıştır. Kullanılan test fonksiyonları optimum değerleri 0 olan minimizasyon problemleridir. Bu fonksiyonlardan 5 tanesi bir tane global minimumu olan tek modlu, 4 tanesi ise yine bir tane global minimumu

olan fakat çok sayıda yerel minimumlara sahip çok modlu fonksiyonlardır. Tablo 1’de kullanılan test fonksiyonlarının özellikleri verilmiştir. Tablo 2’de ise bu fonksiyonların matematiksel ifadeleri yer almaktadır.

Tablo 1. Çalışmada kullanılan test fonksiyonları.

Fonksiyon	Çözüm	Modu
F1: Sphere	-5.12, 5.12	Tek modlu
F2: Rosenbrock	-2.048, 2.048	Tek modlu
F3: Zakharov	-5.0, 5.0	Tek modlu
F4: Schwefel 2.21	-100.0, 100.0	Tek modlu
F5: Schwefel 2.22	-10.0, 10.0	Tek modlu
F6: Ackley	-32.768,	Çok modlu
F7: Rastrigin	-5.12, 5.12	Çok modlu
F8: Griewank	-600.0, 600.0	Çok modlu
F9: Weierstrass	-5.12, 5.12	Çok modlu

Tablo 2. Test fonksiyonlarının matematiksel ifadeleri.

Fonksiyon	Denklem
F1	$f(x) = \sum_{i=1}^d x_i^2$
F2	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F3	$f(x) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^4$
F4	$f(x) = \max\{ x_i , 1 \leq i \leq n\}$
F5	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
F6	$f(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$
F7	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$
F8	$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} + \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$
F9	$f(x) = \sum_{n=1}^i \frac{\sin(\pi n^a x)}{\pi n^a}$

Önerilen model daha geniş bir sorun yelpazesine uygulanması olasılığını sağlayan ve daha yüksek boyutlu kıyaslamalar kullanılarak analiz edilmiştir. Bu amaçla dokuz test fonksiyonu kullanılmıştır. Testler, Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz işlemci ve 8 GB RAM donanımına sahip Windows 10 Pro işletim sistemine sahip masaüstü bilgisayar ile gerçekleştirilmiştir. Proje kodları Python programlama dili ile yazılmıştır.

Her ne kadar 1995 yılında önerilen ilk PSO’da popülasyon sayısı 20-50 parçacık olarak söylenmişse de Piotrowski vd. [26] yaptıkları çalışmada test ettikleri gerçek dünya problemleri için 50-300 arası parçacığın, kıyaslama problemleri için ise 70-500 arası parçacığın güvenli bir seçim olduğunu belirtmiştir. Bu çalışma dikkate alınarak popülasyon sayısı n=100 olarak belirlenmiştir. Ancak daha küçük boyutlu bir sürüde önerilen yöntemin performansını araştırmak için popülasyon sayısı yarıya indirilerek testler tekrar edilmiştir. Derin ağdaki her bölümün uzunluğu PSO algoritmasının iterasyon sayısı kadardır. Derin ağın test edilmesi için ise bölüm sayısı 150 olarak seçilmiştir. Hem pekiştirmeli öğrenme hem de PSO için popülasyon sayısı ve problem boyutu dışındaki diğer tüm parametreler tüm deneylerde aynıdır.

Popülasyon sayısı n=100 ve problem boyutu d=50 olarak seçilerek yapılan testler sonucu elde edilen veriler incelendiğinde ortalama değerlerin optimum değer olan 0’a çok yakın olduğu görülmektedir. Farklı popülasyon sayısı ve farklı problem boyutları ile gerçekleştirilen testlerde birçok yerel minimuma sahip çok modlu bir kıyaslama problemi olan Griewank fonksiyonu için tüm test sonuçlarında optimum değer elde edilmiştir.

Popülasyon sayısı 100 ve problem boyutu 50 ve 100 parametreleri için elde edilen sonuçlar Tablo 3’te verilmiştir. Popülasyon sayısı 50 ve problem boyutu 50 ve 100 parametreleri için elde edilen sonuçlar ise Tablo 4’te verilmiştir. Elde edilen sonuçların kıyaslanabilmesi için bu çalışmada kullanılan tüm test fonksiyonları aynı parametre değerleriyle standart PSO kullanılarak da çalıştırılmıştır. Tablo 3 ve tablo 4 incelendiğinde önerilen model ile problem boyutunun 100 olduğu tüm testlerde daha iyi sonuçlar elde edildiği görülmektedir. Ayrıca çok modlu fonksiyonlar olan ackley, rastrigin, griewank ve weierstrass fonksiyonlarında da hem 50 hem de 100 boyut ile yapılan testlerde standart PSO’dan daha iyi sonuçlar elde edilmiştir. Diğer testlerde ise başarıyı standart PSO’ya çok yakındır.

Tablo 3. Popülasyon sayısı 100 ve problem boyutu 50 ve 100 için elde edilen sonuçlar.

Fonk.		n = 100, d = 50				n = 100, d = 100			
		En Kötü	En İyi	Ort.	Std. Sapma	En Kötü	En İyi	Ort.	Std. Sapma
F1	DQN PSO	1,32E-78	3,10E-134	2,19E-79	5,88E-79	1,31E-57	1,62E-187	2,29E-58	5,31E-58
	STD PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	5,24E+01	0,00E+00	1,75E+01	2,14E+01
F2	DQN PSO	1,03E-20	9,62E-22	4,90E-21	3,91E-21	1,40E+02	4,60E-27	2,43E+01	3,97E+01
	STD PSO	3,99E+00	3,61E-08	7,04E-01	1,61E+00	9,48E+02	1,89E+00	1,86E+02	3,75E+02
F3	DQN PSO	6,25E-09	4,53E-125	1,04E-09	2,55E-09	5,83E+02	2,42E-22	4,34E+01	1,25E+02
	STD PSO	2,43E+02	1,93E-107	1,15E+02	1,09E+02	9,49E+02	1,57E+02	5,11E+02	3,14E+02
F4	DQN PSO	3,08E-15	1,05E-92	5,18E-16	1,26E-15	9,37E-02	2,00E-04	1,68E-02	3,77E-02
	STD PSO	8,87E-22	5,90E-24	1,96E-22	3,45E-22	2,26E+01	8,88E+00	1,52E+01	4,42E+00
F5	DQN PSO	6,21E-36	4,83E-67	1,21E-36	2,48E-36	1,86E-31	9,17E-91	3,13E-32	7,60E-32
	STD PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,10E+02	1,72E-139	8,86E+01	4,40E+01
F6	DQN PSO	7,55E-10	4,00E-15	2,52E-10	3,90E-10	7,55E-12	4,00E-15	1,27E-12	3,08E-12
	STD PSO	1,73E+00	1,47E-14	7,19E-01	8,21E-01	3,81E+00	2,60E+00	3,18E+00	4,99E-01
F7	DQN PSO	9,95E-01	0,00E+00	1,66E-01	4,06E-01	8,62E+02	0,00E+00	1,16E+02	1,92E+02
	STD PSO	2,45E+02	1,54E+02	1,97E+02	3,75E+01	6,08E+02	4,82E+02	5,26E+02	5,19E+01
F8	DQN PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	STD PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	9,10E+01	1,78E-15	3,65E+01	4,42E+01
F9	DQN PSO	4,35E+00	0,00E+00	1,22E+00	1,61E+00	7,21E+00	1,97E+00	4,39E+00	2,31E+00
	STD PSO	5,34E+01	2,59E+01	4,38E+01	1,05E+01	1,27E+02	1,03E+02	1,14E+02	8,22E+00

Tablo 4. Popülasyon sayısı 50 ve problem boyutu 50 ve 100 için elde edilen sonuçlar.

Fonk.		n = 50, d = 50				n = 50, d = 100			
		En Kötü	En İyi	Ort.	Std. Sapma	En Kötü	En İyi	Ort.	Std. Sapma
F1	DQN PSO	7,18E-53	4,94E-228	1,79E-53	3,59E-53	3,59E-36	6,72E-11	6,82E-37	1,44E-36
	STD PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	5,24E+01	4,24E-21	2,62E+01	2,34E+01
F2	DQN PSO	3,72E+01	3,46E-27	4,33E+00	1,05E+01	9,49E+01	1,80E+01	4,57E+01	1,85E+01
	STD PSO	8,57E+00	1,34E-10	2,76E+00	3,45E+00	2,82E+03	2,45E+01	8,53E+02	5,88E+02
F3	DQN PSO	2,78E-11	4,05E-81	4,63E-12	1,13E-11	7,14E+02	4,19E-75	1,22E+02	1,65E+02
	STD PSO	2,52E+02	5,36E+01	1,67E+02	8,19E+01	2,07E+03	4,94E+02	1,07E+03	4,56E+02
F4	DQN PSO	1,76E-11	6,38E-14	6,51E-12	7,10E-12	2,12E-01	3,87E-02	1,11E-01	7,09E-02
	STD PSO	1,52E-12	1,40E-13	5,22E-13	5,46E-13	3,02E+01	1,80E+01	2,58E+01	5,33E+00
F5	DQN PSO	3,20E-26	2,28E-80	6,57E-27	1,26E-26	1,05E-27	1,09E-61	1,75E-28	4,29E-28
	STD PSO	1,10E+02	2,36E-268	3,67E+01	5,68E+01	2,20E+02	1,01E-83	1,10E+02	9,84E+01
F6	DQN PSO	1,47E-14	7,55E-15	9,92E-15	3,67E-15	2,89E-14	4,00E-15	1,76E-14	8,53E-15
	STD PSO	1,57E+01	1,08E+01	1,41E+01	1,74E+00	1,30E+01	1,27E+00	4,11E+00	4,39E+00
F7	DQN PSO	2,99E+02	0,00E+00	3,67E+01	8,85E+01	2,61E+02	2,27E-13	7,21E+01	7,21E+01
	STD PSO	3,14E+02	1,46E+02	2,28E+02	6,18E+01	6,28E+02	4,47E+02	5,47E+02	4,89E+01
F8	DQN PSO	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
	STD PSO	8,79E-01	0,00E+00	1,80E-01	3,44E-01	2,71E+02	1,29E-14	9,05E+01	9,89E+01
F9	DQN PSO	3,94E+00	0,00E+00	2,13E+00	1,58E+00	1,54E+01	4,26E+00	8,30E+00	3,92E+00
	STD PSO	5,79E+01	1,61E+01	4,66E+01	1,54E+01	1,29E+02	1,09E+02	1,18E+02	7,32E+00

4. Sonuç ve Tartışma

Bu çalışmada parçacık sürü optimizasyonu ile derin pekiştirmeli öğrenme yöntemlerini birlikte kullanan bir model önerilmiştir. Önerilen model 50 ve 100 boyutlu 5 tane tek modlu ve 4 tane de çok modlu test fonksiyonu kullanılarak test edilmiştir. Elde edilen sonuçlar modelin yakınsama ve küresel keşif yetenekleri açısından başarımının oldukça iyi olduğunu göstermektedir.

Plevris ve Solorzano [27] yaptıkları çalışmada kıyaslama amacıyla kullanılacak otuz test fonksiyonunun genetik algoritma, parçacık sürü optimizasyonu ve sıralı kuadratik programlama ile farklı boyutlar için çözümünü sağlamışlardır. Bu çalışmada yer alan test fonksiyonlarının 50 boyut için PSO ile çözümünde popülasyon sayısı 500 seçilmiştir. Elde edilen sonuçlara bakıldığında önerilen modelin oldukça umut verici olduğu görülmektedir.

Derin pekiştirmeli öğrenme algoritmaları ve parçacık sürü optimizasyonu, keşif yeteneklerini bir araya getirerek çeşitli problemlerin üstesinden gelmek için güçlü bir yöntem sunmaktadır. Derin pekiştirmeli öğrenme modelleri ile metasezgisel algoritmaların keşif yeteneklerinin iyileştirilmesinin pek çok problemin üstesinden gelmek için güçlü bir yöntem olduğu görülmektedir. Derin pekiştirmeli öğrenme, çevreyi keşfetmek ve öğrenmek için kullanılırken, parçacık sürü optimizasyonu parçacıkların deneme-yanılma yöntemiyle arama alanını keşfetmelerine yardımcı olmaktadır. Bu iki yöntemin kombinasyonu daha geniş bir çözüm uzayında etkili bir şekilde gezinmeyi sağlamak için büyük bir potansiyel oluşturmaktadır.

Bu çalışmada görülmüştür ki parçacıkların problemin çözüm uzayında çözüme doğru hızlı bir şekilde ilerlemesi için yapacakları ve ajan tarafından seçilen eylemlerin belirlenmesi çok önemlidir. Parçacıkların lokal minimumlar gibi problemler alanları hızla atlayarak çözüme ilerlemesini sağlayacak şekilde eylemlerin tanımlanabilmesi, önerilen yöntemin pek çok problem için genelleştirilebilmesini ve geniş çözüm uzaylarında iyi çözümler elde edilebilmesini sağlayacaktır.

İlerleyen çalışmalarda literatürdeki güncel PSO varyantları incelenerek eylem uzayının iyileştirilmesi amaçlanmaktadır. Ayrıca modelin ayrık problemler ile gerçek dünya problemlerindeki başarısını ölçmek için çalışmalar yapılacaktır.

Kaynaklar

- [1] Calafiore G ve Ghaoui L E. Optimization Models. Cambridge University Press, 2014.
- [2] Seyyedabbasi A, Aliyev R, Kiani F, Gulle M U, Basyildiz H ve Shah M A. Hybrid Algorithms Based on Combining Reinforcement Learning and Metaheuristic Methods to Solve Global Optimization Problems. Knowledge-Based Systems 2021; 223: 1-20.
- [3] Kennedy J ve Eberhart R C. Particle Swarm Optimization. International Conference on Neural Networks; 1995; Perth, WA, Australia.
- [4] Xu G. An Adaptive Parameter Tuning of Particle Swarm Optimization Algorithm. Applied Mathematics and Computation 2013; 219(9): 4560-4569.
- [5] Zhang W, Ma D, Wei J ve Liang H. A Parameter Selection Strategy for Particle Swarm Optimization Based on Particle Positions. Expert Systems with Applications 2014; 41(7): 3576-3584.
- [6] Pedersen M ve Chipperfield A. Simplifying Particle Swarm Optimization. Applied Soft Computing 2010; 10(2): 618-628.
- [7] Garg H. A Hybrid PSO-GA Algorithm for Constrained Optimization Problems. Applied Mathematics and Computation 2016; 274: 292-305.
- [8] Kamboj V K. A Novel Hybrid PSO-GWO Approach for Unit Commitment Problem. Neural Computing and Applications 2016; 27: 1643-1655.
- [9] F. A. Şenel, F. Gökçe, A. S. Yüksel ve T. Yiğit, «A Novel Hybrid PSO-GWO Algorithm for Optimization Problems,» Engineering with Computers , cilt 35, pp. 1359-1373, 2018.
- [10] Premalatha K ve Natarajan A. Hybrid PSO and GA for Global Maximization. International Journal of Open Problems in Computer Science and Mathematics 2009; 2(4): 597-608.
- [11] Chegini S N, Bagheri A ve Najafi F. PSOSCALF: A New Hybrid PSO Based on Sine Cosine Algorithm and Levy Flight for Solving Optimization Problems. Applied Soft Computing 2018; 73: 697-726.
- [12] Hayat I, Tariq A, Shahzad W, Masud M, Ahmed S, Ali M U ve Zafar A. Hybridization of Particle Swarm Optimization with Variable Neighborhood Search and Simulated Annealing for Improved Handling of the Permutation Flow-Shop Scheduling Problem. Systems 2023; 11(5): 1-17.
- [13] Khaloo A ve Amirahmadi A. Performance Assessment of Steel Cantilever Beams Based on Connection Behaviour Using DIC Technique and Improved Hybrid PSO Algorithm. Structural Health Monitoring 2023.
- [14] Sutton R S ve Barto A G. Reinforcement Learning: An Introduction. Londra: The MIT Press, 2015.
- [15] Yin S, Jin M, Lu H, Gong G, Mao W, Chen G ve Li W. Reinforcement-learning-based Parameter Adaptation Method for Particle Swarm Optimization. Complex & Intelligent Systems 2023.
- [16] Meng X, Li H ve Chen A. Multi-strategy Self-learning Particle Swarm Optimization Algorithm Based on Reinforcement Learning. Mathematical Biosciences and Engineering 2023; 20(5): 8498-8530.
- [17] Wang F, Wang X ve Sun S. A Reinforcement Learning Level-based Particle Swarm Optimization Algorithm for Large-scale Optimization. Information Sciences 2022; 602: 298-312.
- [18] Lu L, Zheng H, Jie J, Zhang M ve Dai R. Reinforcement Learning-based Particle Swarm Optimization for Sewage Treatment Control. Complex & Intelligent Systems 2021; 7: 2199-2210.
- [19] Wu D ve Wang G G. Employing Reinforcement Learning to Enhance Particle Swarm Optimization Methods. Engineering Optimization 2022; 54(2): 329-348.
- [20] Liu W ve Wang X. Dynamic Decision Model in Evolutionary Games Based on Reinforcement Learning. Systems Engineering - Theory & Practice 2009; 29(3): 28-33.
- [21] Zai A ve Brown B. Deep Reinforcement Learning in Action. Manning, 2020.
- [22] Abeyrathna K D ve Jeenanunta C. Escape Local Minima with Improved Particle Swarm Optimization Algorithm. In Norsk IKT-konferanse for Forskning Og Utdanning; 2019.
- [23] Çomak E. A Particle Swarm Optimizer with Modified Velocity Update and Adaptive Diversity Regulation. Expert Systems 2018; 36(1).
- [24] Freitas D, Lopes L G ve Morgado-Dias F. Particle Swarm Optimisation: A Historical Review Up to the Current Developments. Entropy 2020; 22(3).
- [25] He Y, Ma W J ve Zhang J P. The Parameters Selection of PSO Algorithm influencing on Performance of Fault Diagnosis. MATEC Web of Conferences; 2016; Amsterdam, Netherlands.
- [26] Piotrowski A P, Napiorkowski J J ve Piotrowska A E. Population size in Particle Swarm Optimization. Swarm and Evolutionary Computation 2020; 58.
- [27] Plevris V ve Solorzano G. A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking. Data 2022; 7(4).