Euroasia Journal of Mathematics, Engineering, Natural & Medical Sciences
International Indexed and Refereed
ISSN 2667-6702

5G Ağlarında Derin Güçlendirme Öğrenme Temelli Dinamik Kaynak Tahsisi

Deep Reinforcement Learning Based Dynamic Resource Allocation in 5G Networks

Zahraa Faris Hamdan Al-Aani

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, Türkiye

Arif Dolma

Dr. Öğr. Üyesi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, Türkiye

** Corresponding author: zahraalaani@gmail.com*

Geliş Tarihi / Received: 22.06.2023
Kabul Tarihi / Accepted: 17.08.2023

Araştırma Makalesi/Research Article
DOI: 10.5281/zenodo.8416084

ÖZET

5G teknolojilerinin hızla yaygınlaşması, ağ kaynaklarının yönetimi için yeni stratejilerin geliştirilmesini zorunlu kılıyor. Kaynaklar geleneksel olarak kapsamlı arama ve genetik algoritmalar gibi buluşsal yaklaşımların yanı sıra dal ve sınır gibi kombinatoryal teknikler kullanılarak tahsis edilir. Ultra yoğun baz istasyonu kurulumlarına, devasa bağlantılara ve farklı kullanıcı sınıfları için değişen QoS gereksinimlerine sahip büyük ölçekli heterojen hücresel ağlar, hesaplama maliyetleri nedeniyle bu çözümlerden yararlanamaz. Sonuç olarak, beşinci nesil kablosuz ağlarda geleneksel kaynak tahsis algoritmalarından bir paradigma değişikliğine ihtiyaç vardır. Düşük bilgi işlem maliyetiyle performansı optimize etmeye yönelik yöntemler, veriye dayalı Makine Öğrenimi (ML) kullanılarak oluşturulmuştur. Derin öğrenme (DL), ağ verilerinden bir kaynak yönetimi yöntemini simüle etmek için çok katmanlı bir sinir ağını eğitmek için kullanışlı bir tekniktir. Böylece, aksi takdirde kaynak tahsisi sorunlarını çözmek için gerekli olacak yoğun çevrimiçi hesaplamalardan kaçınılabilir. Çok hücreli kablosuz ağlar için, bu araştırmada toplam ağ verimini artırmak amacıyla derin öğrenmeye dayalı bir kaynak tahsis çerçevesi oluşturuyoruz. Derin Q-ağları (DQN) algoritmasını ve DQN'nin Rain-bow uzantılarını kullanarak çoklu pekiştirmeli öğrenme aracılarını eğitiyor ve test ediyoruz. Her aracının performansı, 5G Kentsel Makro simülasyon senaryolarında test edilir ve sabit bir güç tahsisi yaklaşımıyla kıyaslanır.

Anahtar Kelimeler: Derin öğrenme (DL), baz istasyonu (BS), derin Q-öğrenme (DQL) algoritması.

ABSTRACT

The rapid proliferation of 5G technologies necessitates the development of new strategies for managing network resources. Resources are traditionally allocated using heuristic approaches like exhaustive search and genetic algorithms, as well as combinatorial techniques such as branch and bound. Large-scale heterogeneous cellular networks with ultra-dense base station installations, huge connections, and varying QoS needs for distinct classes of users can not benefit from these solutions due to their computational cost. There is a need for a paradigm change from traditional resource allocation algorithms in the fifth generation of wireless networks as a result. Methods for optimizing performance with a low computing cost have been created using data-driven Machine Learning (ML). Deep learning (DL) is a useful technique for training a multi-layer neural network to simulate a resource management method from network data. Thereby avoiding heavy online computations that would otherwise be necessary to solve resource allocation problems might be achieved. For multi-cell wireless networks, we create a deep learning-based resource allocation framework with the goal of increasing total network throughput in this research. We train and test multiple reinforcement learning agents using the deep Q-networks (DQN) algorithm, and the so-called Rain-bow extensions

of DQN. The performance of each agent is tested on 5G Urban Macro simulation scenarios, and is benchmarked against a fixed power allocation approach.

Keywords: Deep learning (DL), base station (BS), deep Q-learning (DQL) algorithm.

1. GİRİŞ

Kapasite ve kapsama taleplerindeki muazzam büyümeyi karşılamak için 5G mobil iletişim sistemlerinin tanıtılmasıyla mobil iletişim yeni bir döneme girmiştir. Çok çeşitli yenilikçi hizmetler sunmak için 5G mobil ağlarının gecikme, bant genişliği, güvenilirlik ve uyarlanabilirlik açısından çok çeşitli gereksinimleri karşılaması gerekir. Ağ kaynaklarının yönetimi, ağ performansını iyileştirmek için iyi bir mimari gerektiren daha zor bir görev haline gelir. Bir ağ dilimleme yaklaşımı, kullanıcıların QoS kriterlerini belirlemelerine izin verdiğinden, bu durumda 5G ağları için uygun bir tekniktir. Ağ dilimleme temelinde, kaynak tahsis esnekliğini ve ağın kapasitesini geliştirmek için 5G ağlarında verimli kaynak tahsis yöntemlerinin kullanılması gerekir. Dilimleme yaptığınızda, etkinleştirilmiş dilimlerin standartlarını karşılamak için gereken ağ kaynaklarını sağladığınız için ağ kaynaklarını daha iyi kullanırsınız. 5G teknolojilerinde radyo kaynaklarının verimli bir şekilde kullanılması tamamen yeni bir meydan okumayı temsil ediyor. Önceki nesillerin metodolojileri, 5G sistemlerinde güvenilir iletişim için optimum kaynak tahsisi sağlayamamaktadır. Son yıllarda optimal ağ kaynak yönetimi stratejilerini belirlemek için makine öğrenimi yöntemlerini birleştirmeye artan bir ilgi vardır. Kablosuz ağların optimize edilmesi söz konusu olduğunda, doğru bilgi ve eksiksiz bir model bilinmez, bu nedenle Reinforcement Learning çerçevesi mantıklıdır. Son zamanlarda Takviyeli Öğrenme alanlarında, Makine Öğrenimi ve Derin Öğrenmedeki gelişmelere bağlı olarak umut verici sonuçlar gösterilmiştir. Son yirmi yılda dünya, ilk nesilde sınırlı sayıda kullanıcı için analog sesli aramaları mümkün kılmaktan dördüncü nesilde yüz milyonlarca cihaz için yüksek veri hızları sağlamaya kadar mobil radyo iletişim teknolojilerinin hızlı gelişimine tanık oldu. (4G) son birkaç yılda. Son zamanlarda ortaya çıkan beşinci nesil (5G) kablosuz iletişim, internet için devasa makine tipi iletişim sağlamak için 10-100 kat daha fazla sayıda bağlı cihaza hizmet vermeyi amaçlarken, daha da yüksek veri hızları ve daha düşük gecikme vaadiyle geliyor. şeyler uygulamaları. Doğal olarak, artan kullanıcı sayısı ve daha katı hizmet gereksinimleriyle, erişim noktalarının ve baz istasyonlarının sayısı artmalı, bu da piko hücreler ve femto hücreler gibi küçük hücrelerin yoğun bir şekilde yerleştirilmesine yol açmalıdır. Bu arada, 5G'nin farklı boyutlarda, iletim güçlerinde ve ana taşıyıcı bağlantılarında diğer radyo erişim teknolojilerinin ağlarıyla birlikte var olması planlanıyor. Dağınık ve heterojen ağ mimarileri, etkili radyo kaynağı ve parazit yönetimi, yeni ve daha güçlü çözümler arayışını motive ederek giderek daha önemli hale geliyor.

2. ÖNERİLEN SİSTEM

Tek değişkenlere (güç gibi) dayalı bir model oluşturmak ve eğitmek için algoritmik teknikleri kullanan birçok veriye dayalı yaklaşım (DL) vardır. Sonuç, öğrenme stratejilerinin etkisiz olmasıdır. Küçük baz istasyonlarındaki ("SBS") veya bilişsel radyolardaki ("CR'ler") güç dağıtımı, aksine, daha küçük bir ağda DRL tabanlı olma eğilimindedir (örneğin, SBS'ler veya CR'ler kendi aralarında işbirliği yapmaz veya bilgi alışverişinde bulunmaz) . Geniş bir ölçekte, çalışmaların birçoğu, dağıtılmış bir şekilde çok hücreli kablosuz ağlara güç tahsis eder. Bu nedenle, mümkün olan en iyi cevap bulunmayabilir. Ek olarak, bu çalışmalar ya hücre başına çok sayıda alt bantlı tek bir kullanıcıyı ya da hücre başına herkes tarafından paylaşılan tek bir frekans bandına sahip bir kullanıcı grubunu inceler. Aynı frekans alt bantlarını kullanan hücre başına birden fazla kullanıcıya sahip çok hücreli ağlarda bu stratejileri uygulamak mümkün değildir.

Spesifik olarak, 5G'nin dinamik kaynak tahsisinin performansı nasıl etkileyeceğine bakacağız. RL ve DRL tabanlı çerçeveler, en iyi zamanlama politikasını belirlemek amacıyla tasarlanmış ve

uygulanmıştır. İdeale yakın bir güç tahsisi politikası için, bir DQL tekniği (derin Q-öğrenme) uyguluyoruz. Bu tezin en önemli başarılarından biri, optimal veya optimale yakın bir çok hücreli kablosuz ağ kaynak tahsis algoritmasının geliştirilmesiydi. Çok hücreli kablosuz ağlar için, denetimli derin öğrenmeye dayalı merkezi bir kaynak tahsisi yaklaşımı geliştirilmiştir. Sonuç olarak, ağ verimini en üst düzeye çıkarmak birincil hedeftir.

Bu tezin ana katkıları aşağıdaki gibi özetlenmiştir:

• DL mimarilerini ve DL modelinin eğitim ve test süreçlerini kısaca tanımlayın. Öte yandan, bir kablosuz kaynak tahsis problemini çözmek için DL modelinin nasıl kullanılabileceğini tartışıyoruz.

• Aynı frekans alt bantlarını paylaşan çok sayıda kullanıcının olduğu çok hücreli bir ağda kaynakların nasıl dağıtılacağını tanımlayın. Birçok alt bant ve güç seviyesine sahip çok hücreli ağlar için, optimal alt bant ve güç dağıtım çözümünü bulmak için denetimli bir derin öğrenme modeli kullanılır. Böyle bir problem, kombinatoryal optimizasyonda sıklıkla ortaya çıkar, ancak asla bu şekilde çözülmemiştir. Bu nedenle, DL modeli için neredeyse mükemmel eğitim verileri sağlamak için genetiği kullanıyoruz ve daha sonra değerlendiriyoruz (GA). Ayrıca, önerilen model için çevrimiçi eğitim ve test süreçlerini sunuyoruz.

• Kablosuz ağlarda kaynak tahsisi için, kaynak tahsisine yönelik denetimli DL yaklaşımının sınırlarını inceleyin. DRL tabanlı merkezi kaynak tahsisini kullanarak çok hücreli ağlar için bir çözüm sunuyoruz. İlk kez, frekans alt bantlarını paylaşan çok sayıda kullanıcıya sahip çok hücreli bir ağ, maksimum güç kısıtlaması koşulları altında güç tahsisi sorununu ele alan bir şemaya sahiptir. DRL aracısı için, aracının durumunu, eylemini ve ödül alanlarını ve ayrıca ödül işlevini tanımlarız. Ayrıca önerilen DRL tabanlı kaynak tahsis planının çevrimiçi eğitim mekanizmasını da tanımlıyoruz.

1.1 Algoritma Seçimi

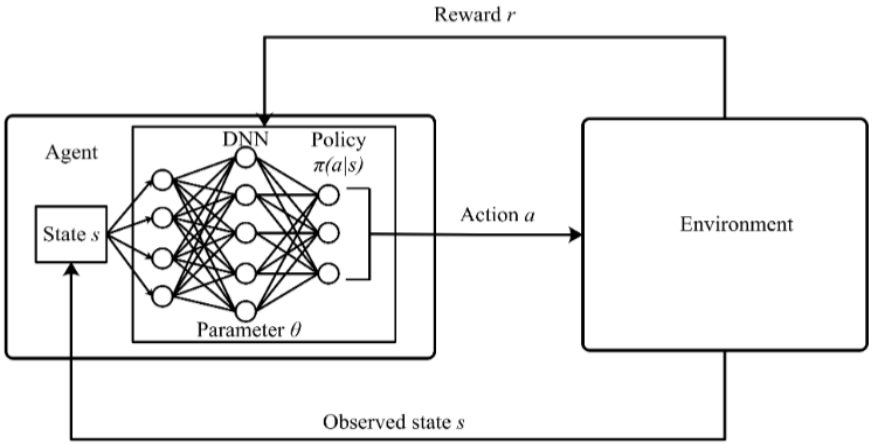
Son yıllarda geliştirilen sayısız başarılı derin Takviyeli Öğrenme algoritmasına rağmen, uygun yöntemi seçmemiz, pratikte, çeşitli hususlar tarafından yönetildi. İlk olarak, rastgele bir politika tarafından oluşturulan verilerden çevrimdışı bir şekilde birden fazla modeli eğitmek için kendimizi politika dışı yöntemlerle sınırladık. Bu kısıtlama, esas olarak, bir saniyelik simüle edilmiş radyo ağının pratikte birkaç dakika sürdüğü son derece ayrıntılı ve sonuç olarak yavaş simülasyondan kaynaklanmaktadır. Ayrıca, çevrimiçi eğitimdeki bazı erken girişimler, uygun olmayan hiperparametreler nedeniyle düşük performans gösteren politikalara yol açtı; yine de, uygun parametreleri araştırmak için deneyleri tekrar etmenin son derece zaman alıcı olduğu kanıtlandı.

İkincisi, eğitim setimiz oldukça küçük olduğundan (yine uzun simülasyonlar nedeniyle), yüksek veri verimliliğine sahip bir algoritma seçmeyi amaçladık, bu da aracının her geçiş örneğinden mümkün olduğunca çok şey öğrenmesini çok önemli hale getirdi. Bu ve önceki gereksinimler, derin Q ağlarını ve derin deterministik politika gradyanını potansiyel adaylar olarak destekledi, çünkü ikisi de politika dışı yöntemlerdir ve yeniden oynatma belleği uygulamaları nedeniyle yüksek veri verimliliği sağlar.

Son olarak, ağdaki tüm BS'lerin gücünü kontrol etmek için tek bir aracıyı eğitmenin pratik sonuçlarını düşündük. Derin deterministik politika gradyanı, eylem olarak tek bir gerçek değerli skaler çıktısı verdiğinden, bu tez için uygun algoritma olarak derin Q ağları seçilmiştir. Ek olarak, Rainbow uzantıları gibi derin Q ağlarında yapılan son gelişmeler, bizi hem vanilya algoritmasını hem de sonraki varyantlarını denemeye teşvik etti.

1.1.1 Derin Takviyeli Öğrenme (DRL)

Takviyeli öğrenmede, çevre ile etkileşime girerek öğrenen bir yazılım aracısı vardır. Ajan, mevcut durumunu ve çevrenin durumunu algılar ve ardından bir eylem seçer. Yaptığı her eylemin bir sonucu vardır. Temsilci ya iyi bir hamle için bir ödül ya da kötü bir hamle için bir ceza alır. Temsilcinin birincil işi, bir dizi eylem yoluyla kümülatif ödülü en üst düzeye çıkarmaktır. Bir aracı olarak derin bir sinir ağı kullanıldığında, Derin Takviyeli Öğrenme olarak adlandırılır. Bu nedenle Derin Takviyeli Öğrenme, Şekil 1'de gösterildiği gibi derin sinir ağı ve Takviyeli Öğrenmenin birleşimidir.



Şekil 1. Derin pekiştirmeli öğrenme

Tablo 1. Semboller

Symbol	Description
s_t	State at time step t
a_t	Action at time step t
$\mathcal{A}$	Action space
r_t	Reward at time step t
R_t	Accumulated reward
γ	Discount factor
Q	Action-value function
$\hat{Q}$	Target action-value function
y	Approximate target values
$L_i(\theta_i)$	Loss function at each iteration i
$\mathbb{V}_{s'}[y]$	Variance of the target
λ	Speed of decay
τ	Total number of steps elapsed
K	Number of base stations (BSs)
F	Number of frequency sub-bands
B	Bandwidth of each sub-band in MHz
$P_{k,f}$	Power allocated by cell k in frequency sub-band f
$P_k^{\max}$	Maximum power of a cell k
$\mathcal{U}_k$	Set of users associated with cell k
$\mathcal{U}$	Set of all users
$\mathcal{A}_k$	Allocation vector of cell k
$\mathcal{A}_{k,f}$	The user assigned to sub-band f in cell k
$\mathbb{I}(.)$	Indicator function
$\text{SINR}_{u,k,f}$	SINR of user u in cell k over frequency sub-band f

1.1.2 Derin Takviyeli Öğrenme Mimarileri

Okunabilirliği artırmak için bu çalışma boyunca kullanılan tüm semboller Tablo 1'de toplanmıştır. Derin pekiştirmeli öğrenme algoritmaları üç kategoriye ayrılabilir: değer tabanlı, politika tabanlı ve aktör kritik yöntemler. Değere dayalı derin pekiştirmeli öğrenmede, etmen, eylem-durumu değer fonksiyonundan optimal eylemi öğrenir. Derin Q-öğrenme ve SARSA, değere dayalı derin pekiştirmeli öğrenme yöntemlerinde en popüler olanlardır. REINFORCE öğrenimi gibi ilke tabanlı bir yöntemde, aracı ilkeyi doğrudan optimize eder. Aktör-eleştiri yöntemlerinde, eleştirmen eylem-değer fonksiyonunu günceller ve aktör politikayı günceller. Derin deterministik politika gradyanı algoritması aktör-eleştirmen tabanlıdır.

Bu bölümde, değere dayalı bir Derin Takviyeli Öğrenme olan derin Q-öğrenmeye odaklanıyoruz. t adımında, derin Q-öğrenme aracısı bir S durum uzayından s_t durumunu alır ve bir eylem uzayı A'dan bir eylemde bulunur. Ajan bir politika izler $\pi(a_t|s_t)$) yani, eylemi seçmek için s_t durumundan eylem a_t'ye bir eşleme. a_t eylemini yürüttükten sonra, etmen bir r_t ödülü alır ve yeni s_{t+1} durumuna geçer. Etmen, terminal durumuna ulaşana kadar işlemi devam ettirir ve ardından yeniden başlar. Temsilcinin amacı, şu şekilde tanımlanan indirimli birikmiş ödülü maksimize etmektir $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1}$. Burada, $\gamma \in [0,1]$ Mevcut ödüle kıyasla gelecekteki ödüllerin önemini belirleyen indirim faktörüdür. Bir eylem değeri işlevi $Q_\pi(s,a) = E[R_t|s_t = s, a_t = a]$ durumlarda a eylemini seçme ve ardından bir politika izlemenin beklenen getirisidir π. Optimal bir eylem-değer işlevi $Q^*(s,a) = max_\pi Q_\pi(s,a)$ durum s ve eylem a için herhangi bir politika izlenerek elde edilebilecek maksimum eylem değeridir. Optimum eylem-değer işlevi Bellman denklemine ayrıştırılabilir:

$$Q^*(s,a) = E_{s'}[r + \gamma max Q^*(s',a)|s,a] \tag{1}$$

Derin Q - öğrenmede, optimal eylem-değer fonksiyonuna yaklaşmak için bir sinir ağı kullanılır, $Q(s,a;\theta) \approx Q^*(s,a)$. Burada $Q(s,a;\theta)$ is Q ağı olarak adlandırılan ve θ sinir ağının parametresidir. Yinelemeli güncelleme, Q ağını eğitmek ve böylece Bellman denkleminin ortalama kare hatasını azaltmak için kullanılır. Bunun için optimal hedef değerler $r + \gamma + max_{a'} Q^*(a',s')$ yaklaşık hedef değerlerle değiştirilir $y = r + \gamma + max_{a'} Q^*(a',s';\theta_i^-)$, nerede θ_i^- önceki bazı yinelemelerden Q ağının parametreleridir. kayıp fonksiyonu $L_i(\theta_i)$ her yinelemede

$$\begin{aligned} L_i(\theta_i) &= E_{s,a,r}\left[\left(E_{s'}[y|s,a] - Q(s,a;\theta_i)\right)^2\right] \\ &= E_{s,a,r,s'}\left[\left(y - Q(s,a;\theta_i)\right)^2\right] + E_{s,a,r}[V_{s'}[y]] \end{aligned} \tag{2}$$

Burada, $V_{s'}[y]$, θ_i 'dan bağımsız olan hedeflerin varyansı and bu nedenle göz ardı edilebilir. Önceki yinelemedeki parametreler θ_i^- i-inci kayıp fonksiyonunu optimize ederken sabit tutulur $L_i(\theta_i)$. Kayıp fonksiyonunu optimize etmek için Gradient Descent algoritması kullanılır.

1.2 DQL Eğitim Prosedürü

Derin Q ağının eğitim prosedürü Algoritma 2'de gösterilmektedir (Şekil 2). Eylem değeri işlevi Q ve hedef eylem değeri işlevi için iki ayrı Q ağı kullanılır $\hat{Q}$. Hedef Q ağı $\hat{Q}$ eğitim sürecinde y_j hedeflerini oluşturmak için kullanılır. Ajan ya ε olasılığı ile rastgele bir eylem seçer ya da maksimum eylem değeri eylemini seçer. Bu yöntem "açgözlü politika" olarak bilinir. Başlangıçta, aracı yüksek bir değerle başlar. ε değer ve sonra yavaş yavaş deneyime bağlı olarak değeri azaltır.

$$\varepsilon = \varepsilon_{min} + (\varepsilon_{max} + \varepsilon_{min})e^{-\lambda\tau} \quad (3)$$

Algorithm 2 Deep Q-learning with experience replay

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
for episode $= 1, M$ do
  for $t = 1, \cdots, \infty$ do
    With probability ε select a random action a_t
    Otherwise select $a_t = \arg\max_a Q(s_t, a; \theta)$
    Execute action a_t and observe reward r_t and state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in D
    Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from D
    Set $y_j = r_j$ if if episode terminates at step $j + 1$
    Otherwise set $y_j = r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta^{-})$
    Perform gradient descent step on $(y_j - Q(s_j, a_j; \theta))^2$ with respect to the network parameters θ
    Every B steps reset $\hat{Q} = Q$
  end for
end for

Şekil 2. Deneyim tekrarı ile Derin Q-Öğrenme

1.3 Sistem Modeli

K baz istasyonlarının bir aşağı bağlantı hücresel ağını düşünüyorum. Her baz istasyonu $k \in \{1, \dots, K\}$ F frekans alt bantlarına sahiptir. Her alt bandın bant genişliği B MHz'dir. Hücre akraba frekansı alt bandı f tarafından tahsis edilen güç, $P_{k,f}$ hangi ayrık. Bir k hücresinin toplam gücü bir maksimum değerle sınırlıdır P_k^{max} öyle ki $\sum f \in F P_{k,f} \leq P_k^{max}, \forall k \in \{1, \dots, k\}$. İzin vermek U_k k hücresiyle ilişkili kullanıcı kümesini belirtir ve U, ağdaki tüm kullanıcıların kümesidir. Vektör $A_{k,f}$ k hücresindeki alt bant tahsisini gösterir, burada her bir eleman $A_{k,f}$ k hücresinde f alt bandına atanan kullanıcıyı belirten bir tamsayıdır.

Karşılık gelen verim maksimizasyon problemi şu şekilde verilir:

$$max \sum_{k\in\{1,\dots,k\}} \sum_{u\in U_k} \sum_{f=1}^{F} \left[(A_{k,f} = u) B \log(1 + \alpha SINR_{u,k,f})\right] \quad (4)$$

$$s.t. \sum_{f\in F} P_{k,f} \leq P_k^{max}, \forall k \in \{1, \dots, K\} \quad (5)$$

Burada α şu şekilde tanımlanan belirli bir hedef Bit Hata Oranı için bir sabittir $\alpha = -1.5/\log(5BER)$. Bit Hata Oranı olduğunu varsayıyoruz 10^{-6}. Sinyal-parazit-artı-gürültü oranı (SINR) f frekans alt bandı üzerinden iletim yapan k hücresi tarafından hizmet verildiğinde u kullanıcısının değeri şu şekilde ifade edilir $SINR_{u,k,f} \frac{P_{k,f} G_{u,k,f}}{\eta_u \neq \sum_{I\neq k} P_{l,f} G_{u,k,f}}$ ⌘u alıcı gürültüsünü temsil eder ve $G_{u,k,f}$ şu şekilde tanımlanan f frekans alt bandı üzerinden k hücresinden kullanıcı u'ya bağlantı kazancını belirtir $G_{u,k,f} = 10^{-(PL_u + X_\alpha)/10} . |H_{u,k,f}|^2$,burada $H_{u,k,f}$ frekans alt bandı üzerinden k hücresinden u kullanıcısının Rayleighfading kazancıdır f, X_α log-normal gölgelemedir , ve PL_u kullanıcının yol kaybıdır u. Toplam ağ verimi olan ağın faydası şu şekilde tanımlanır:

$$U = \sum_{k\in\{1,\dots,k\}} \sum_{u\in U_k} \sum_{f=1}^{F} \left[\left(A_{k,f} = u\right) B \log\left(1 + \alpha SINR_{u,k,f}\right)\right] \quad (6)$$

1.4 Çok Hücreli Ağlarda Güç Tahsisi: DRL Yaklaşımı

Aşağıda, güç tahsisi gerçekleştirerek toplam ağ verimini en üst düzeye çıkarmak için hücre başına birden çok kullanıcının aynı frekans alt bantlarını paylaştığı çok hücreli ağlar için Derin Güçlendirme Öğrenimi tabanlı bir kaynak tahsis modeli geliştiriyoruz.

1.4.1 DQL Yaklaşımı

Şimdi, çok hücreli ağlarda optimuma yakın güç tahsisi gerçekleştirebilen derin bir Q - öğrenme yaklaşımı sunuyoruz. Spesifik olarak, bu derin Q - öğrenme modeli şunları kullanır $C_{k,v}$ ile birlikte $V_{k,v}$ vektörü bir ağdaki tüm kullanıcıların durumu ve ardından harekete geçer. Burada, her eylem bir güç tahsisine karşılık gelir. Yani K hücreleri, U kullanıcıları ve F alt bantları için durum boyutu (K*U*(F+1)). Toplam eylem sayısı, kullandığımız güç seviyelerinin sayısına bağlıdır. n adet güç seviyesi ve F frekans alt bandı için, maksimuma sahip olabilirim n^F güç kombinasyonları. Bazı kombinasyonlar, maksimum güç kısıtlaması nedeniyle atılacaktır. Her kombinasyonun bir eyleme karşılık geldiği olası toplam kombinasyon sayısını m göstersin. Her hücre için m sayıda eylemimiz var. A_k k hücresi için eylem alanını göstersin ve $a_k \in A_k$ k hücresi için seçilen eylemdir. Bu nedenle, K hücre sayısı için Kx m sayıda eylemimiz var. Derin Q - ağları modeli, her hücre için m sayıda eylemden harekete geçmelidir. Bu nedenle, toplamda, derin Q - ağları modelinin K sayıda işlem yapması gerekir. t adımında, seçilen eylem $a_t = [a_1, a_2, \dots, a_k]$. Yaklaşım aşağıdaki aşamalarda ilerler:

problem formülasyonu: Öncelikle derin Q - öğrenme yaklaşımını uygulamak için problemi formüle etmem gerekiyor. Aracının işi, toplam ağ verimini maksimize etmektir (Denklem (6)). Bölüm bir başlangıç durumundan başlar ve aktarım hızı arttığı sürece devam eder, yani Mevcut aktarım hızı > önceki aktarım hızı. Burada, mevcut aktarım hızı, son eylemlerin yürütülmesiyle elde edilen ağ aktarım hızı ve önceki aktarım hızı, önceki eylemlerden kaynaklanmaktadır. Bölüm, terminal durumuna ulaştığında sona erer, yani son eylemler nedeniyle verim düşer.

Eğitim Aşaması: Önerilen derin Q - öğrenme tabanlı güç tahsisinin eğitim süreci Algoritma 3'te gösterilmektedir (Şekil 3). önerilen modeli eğitmek için derin Q - deneyim tekrarıyla öğrenmeyi kullanıyoruz. Önerilen modelde, belirli adımlar aşağıdaki gibidir.

Adım 1: Q-Network'ü, yani katman başına katman ve nöron sayısını ve aktivasyon fonksiyonlarını tanımlayın. Durum boyutuyla aynı girdi katmanı boyutunu ve toplam eylem sayısı olarak çıktı katmanı boyutunu kullanıyorum. Rastgele ağırlıklarla iki Q-ağını başlatıyoruz: biri eylem değeri işlevi Q ve diğeri hedef eylem değeri işlevi $\hat{Q}$ için. Ayrıca yeniden oynatma belleği D'yi bir miktar N kapasitesine başlatıyoruz.

Adım 2: Her hücre için rastgele bir güç vektörü tahsis edin. Bundan sonra ağdaki her kullanıcı için her alt bandın CQI değerini hesaplarım. Ayrıca Denklem'i kullanarak her kullanıcı için konum göstergesini tahmin ediyoruz. (5). Her kullanıcı için CQI değerlerinin ve konum göstergesinin başlangıç durumunu s_t temsil ettiğini unutmayın.

Adım 3: Her hücre için tüm alt bantlar için mümkün olan minimum güç değerinden oluşan eylemi seçin. Ardından, bu eylemleri gerçekleştiriyoruz ve Denklem'i kullanarak toplam ağ verimini hesaplıyoruz. (6). Bu aktarım hızını sonraki adım için önceki aktarım hızı olarak kullanın.

Adım 4: Eylemleri rastgele seçmek için "-açgözlü politikasını kullanın veya eylemleri seçmek için Q ağını kullanın. Her hücre için m sayıda eylemimiz olduğundan, her eylem için eylem değerini hesaplamak için s_t durumunu Q ağına girdi olarak kullanırız. Bu nedenle, ilk hücre için, ilk m eylemler arasından eylem-değeri maksimum olan eylemi seçin. Ardından, kalan hücreler için, sonraki m eylemlerden maksimum eylem-değeri ile eylemi art arda seçin vb.

Adım 5: Seçilen eylemleri gerçekleştirin, yani eylemleri karşılık gelen güç vektörleriyle eşleştirin ve Adım 2'de belirtilen şekilde yeni s_{t+1} durumunu hesaplayın. o ajan daha sonra r_{t+1} değerinde pozitif bir ödül alır. Ayrıca, Denklem'i kullanarak toplam ağ verimini hesaplıyoruz. (3.7) ve değeri mevcut çıktı olarak kaydedin. Ardından, aşağıdaki koşulu kullanarak yeni adımın bir terminal durumu olup olmadığını kontrol ederiz: mevcut çıktı > önceki çıktı. Son olarak, geçişleri(st, at,rt,st+1) tekrar hafıza D'de saklıyorum.

Adım 6: Q-ağında deneyim tekrarı gerçekleştirin. Deneyim tekrarı mekanizması aşağıdaki adımlara sahiptir.

Adım 6.1: Yeniden yürütme belleğinden rastgele M boyutundaki geçişlerin bir mini partisini örnekleyin.

Adım 6.2: Denklem'i kullanın. (4) o mini partinin hedeflerini y_j güncellemek için. Hedefleri oluşturmak için hedef eylem değeri $\hat{Q}$ ağını kullanıyorum.

Adım 6.3: Eylem değeri Q-ağ parametrelerini güncellemek için kayıp fonksiyonunda (Denklem (2)) bir gradyan iniş adımı gerçekleştirin.

Adım 6.4: Her B güncellemesinde hedef eylem değeri işlevini $\hat{Q}$ almak için eylem-değer işlevi Q parametrelerini klonlayın.

Adım 7: Aracı terminal durumuna ulaşana kadar Adım 5-6'yı tekrarlayın.

Adım 8: Q-ağını belirli bir süre eğitmek için Adım 2-7'yi tekrarlayın.

Test Aşaması: Modelimizi eğittikten sonra, toplam ağ verimi açısından modelin optimal olana ne kadar yakın tahmin edebileceğini test etmemiz gerekiyor. optimale yakın bir çözüm bulmak için genetik algoritma kullanıyoruz. Test için, eğitim adımlarına aşağıdaki adımlar eklenir:

Adım 9: Adım 2-7'yi tekrarlayın ve son ikinci eylemi ve bu eylem için ağ geçişini kaydedin. Bu nedenle, eğerst+1 uçbirim durumuysa ve at aracının uçbirim durumuna ulaştığı eylemse, o eylem için eylemi1ve ağ verimini kaydetmem gerekir. Burada actionat1 optimal eylem olarak kabul edilir.

Adım 10: Her baz istasyonunun toplam ağ faydasını maksimize eden güç vektörünü bulun (Denklem (6)). Bu sorunu çözmek için genetik algoritma kullanıyorum.

Adım 11: Optimum güç vektörü çözümünü Denklem'e uygulayın. (6) toplam ağ verimini hesaplamak için. ayrıca optimum çözümü ve ağ verimini de kaydediyoruz.

Adım 12: Belirli bir miktarda test verisine sahip olana kadar adımları tekrarlamaya devam edin. Tüm eğitim ve testler çevrimiçi olarak gerçekleştirilecektir. Pratik bir ortamda, ağdaki tüm kullanıcılar, Sürekli Kalite İyileştirme değerlerini periyodik olarak, her bir alt bandın Sürekli Kalite İyileştirme değerini çıkaran ve bir konum göstergesi, yani hücre merkezi kullanıcısı veya hücre kenarı ekleyen hizmet veren baz istasyonlarına gönderir. kullanıcı. Bu nedenle, her kullanıcı için bir Sürekli Kalite İyileştirme vektörü ve konum göstergesi olacaktır. Her baz istasyonu daha sonra tüm kullanıcıların işlenmiş bilgilerini derin Q öğrenme modelini çalıştıran merkezi bir varlığa (örneğin yazılım tanımlı ağ denetleyicisi) gönderir. Derin Q öğrenme aracısı, eylem olarak tüm BS'ler için güç vektörünü seçer. Aracı eylemi seçtiğinde, kontrolör güç vektörlerini belirlenen baz istasyonlarına geri gönderir. BS'ler daha sonra gücü buna göre tahsis eder.

Algorithm 3 DQL with experience replay for power allocation

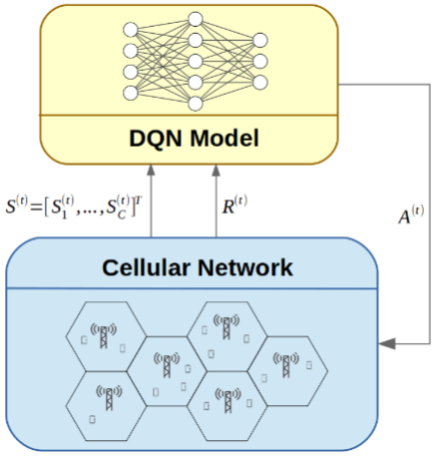
Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
for episode $= 1, \cdots, M$ **do**
  Allocate a random power vector for each cell.
  for $t = 1, \cdots, \infty$ **do**
    Calculate the CQI vector as well as the location indicator for every user in the network.
    Use the CQI vector and the location indicator as state s_t.
    for k $= 1, \cdots, K$ **do**
      With probability ε select a random action $\mathfrak{a}_k$ for cell k
      Otherwise select $\mathfrak{a}_k = \arg\max_{a \in \mathbb{A}_k} Q(s_t, a; \theta)$
    end for
    Execute action $a_t = [\mathfrak{a}_1, \mathfrak{a}_2, \cdots, \mathfrak{a}_K]$ and observe reward r_t and state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in D
    Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from D
    Set $y_j = r_j$ if episode terminates at step $j + 1$
    Otherwise set $y_j = r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta^{-})$
    Perform gradient descent step on $(y_j - Q(s_j, a_j; \theta))^2$ with respect to the network parameters θ
    Every B steps reset $\hat{Q} = Q$
  end for
end for

Şekil 3. Güç tahsisi için deneyim tekrarı ile DQL

1.5 Deneysel Prosedür

Bu bölümde, eğitim ve model seçim prosedürlerine ek olarak simüle edilmiş radyo ağlarının ayrıntılarını tartışacağız.

Deneylerimizde, çeşitli simülasyon senaryolarında rastgele politikalar izlenerek oluşturulan veriler kullanılarak farklı hiperparametrelere sahip birkaç aracı eğitilmiştir. Uygun derin Q - ağlarını ve Rainbow hiper parametrelerini seçmek için, eğitim setinde bulunmayan senaryolar üzerinde küçük testler yapılarak eğitilen modellerin performansı karşılaştırılır. Bu senaryolar, üstün performans gösteren hiperparametreleri belirlemek için birkaç deneyin yapıldığı denetimli öğrenmede ayarlanan doğrulama kavramına benzer olarak düşünülebilir. Uygun model seçildikten sonra, aracı tarafından daha önce görülmeyen senaryolar üzerinde bir dizi yeni test çalıştırırız. Modelin performansı, daha sonra, sabit bir güç tahsis şemasına göre kıyaslanır. Bu, ajanın performansının kesirli programlama veya ağırlıklı minimum ortalama kare hatası gibi yinelemeli bir algoritma ile karşılaştırıldığı bazı çalışmaların aksine. Ancak amacımız, derin pekiştirmeli öğrenme yöntemlerinin pratik uygulaması hakkında fikir vermek olduğundan, yöntemimizi sabit güç tahsis şeması olan gerçek hayattaki statükoyla karşılaştıracağız. Her simülasyon sırasında, aşağı bağlantı verimi düzenli olarak ölçülür ve günlüğe kaydedilir. Her yaklaşımda aşağı bağlantı veriminin ampirik kümülatif dağılım fonksiyonunu çizmek için günlük verilerini kullanarak farklı yöntemleri değerlendirir ve karşılaştırırız. Ayrıca, güç tahsisi ve genel olarak parazit azaltma yöntemleri, yüksek parazitten mustarip hücre sınırındaki kullanıcılara fayda sağladığından, verimin 5. yüzdelik dilimi sıklıkla önemli bir performans ölçütü olarak hesaplanır (Şekil 4).



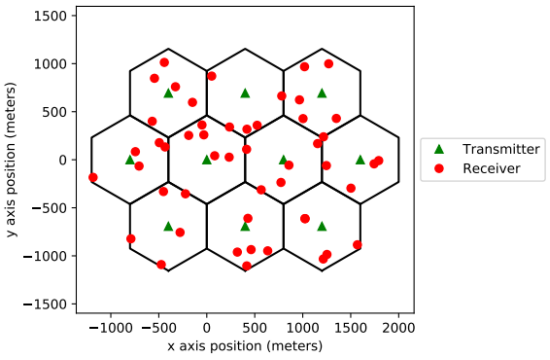
Şekil 4. Ağ hücrelerinin gücünü kontrol eden merkezi aracının gösterimi

3. SONUÇ VE TARTIŞMA

3.1 Deneysel kurulum

Bu çalışmanın bulguları, bir önceki bölümde bahsedilen uygulamaya dayalı olarak bu bölümde sunulmakta, tartışılmakta ve değerlendirilmektedir. Derin Q öğrenmeye dayalı önerilen güç atama çerçevesinin simülasyon ayarları sunulmaktadır. Önerilen algoritmayı uygulamak için TensorFlow [52] kullanıyoruz. Senaryo 1'de K = 5 baz istasyonu, 2'si K = 10 baz istasyonu, 3'ü K = 15 baz istasyonu, hücresel örtülü telsiz R = 500m, hücre yönlü anten gücü = 400W, hücre sayısı = olmak üzere üç farklı simülasyon senaryosu kullanılmıştır. 5, alt bant genişliği B = 2,88MHz, güç yoğunluğu = 174dBm/Hz, güç = 5, güç seviyeleri = {6,4,7,12,8,19.2} W 3 ayrı simülasyon senaryosuna bakıyoruz:

Simülasyonlar boyunca, sırasıyla (K, N) = (5 hücre, 20 bağlantı) ve (10 hücre, 50 bağlantı) olmak üzere iki ağ boyutu seçiyoruz. Şekil 6'da açıklandığı gibi, her bir hücrenin eşit sayıda üniform olarak rastgele yerleştirilmiş alıcılara sahip olduğu 400 metre yarıçaplı homojen altıgen hücreleri ele alıyoruz.



Şekil 5. Bir ağ yapılandırma örneği

Eğitimi, her biri 5.000 zaman aralığı için çalışan dört bölüme ayıracağız. Her bölümün başında yeni bir dağıtımı rastgele örnekliyoruz ve keşif ve öğrenme oranı parametrelerini sıfırlıyoruz. Daha hızlı

yakınsama için, belirleyici ilke çıktısına eklenen gürültü terimini Q-learning'in e-açgözlü algoritmasıyla değiştiriyoruz. Uygulama ve hiper parametreler kaynak koduna dahil edilmiştir. Daha iyi stabilite için, alt katmanın üst katmandan daha yüksek öğrenme oranına sahip olmasını ve ε'nin daha yüksek bir başlangıç değerini kullanmasını, ancak daha yüksek bir bozunma oranına sahip olmasını sağlıyoruz. Değerin ince ayarı, tüm ajanların Pmax veya sıfır güçle iletmek istediği istenmeyen durumlara yakınsamayı önlemek için önemlidir.

3.2 DQL Modelinin Eğitimi

Önce DQN'yi ayarlamalıyız. DQN'miz olarak, gizli katmanda derin bir sinir ağı kullanılır. Doğru doğrusal birim, gizli katman etkinleştirme işlevi (ReLU) olarak kullanılır. Q ağı için girdi katmanı boyutu, 3 farklı senaryo için 5 x 5 x = 100, 10 x 5 x (3 + 1) = 200, 15 x 5 x (3 + 1) = 300'dür. Beş güç seviyesi için toplam olası güç kombinasyonu sayısı hücre başına 53=125'tir. Maksimum güç sınırlaması, bazı enerji kombinasyonlarını hariç tutar. Böylece, çıktı senaryosu için 5 x 72 = 360, 10 x 72 = 720 ve 15 x 72 = 1.080, yani çıktı ölçeği DQN olan toplam eylem sayısıdır. Tablo 2, DQN'nin eğitim parametrelerini gösterir.

Tablo 2. Eğitim parametreleri

Parameter	Value
Number of hidden layers	1
Layers	{ Input, Hidden Layer, Output }
No. of neurons per layer	Scenario 1 : {100, 720, 360} Scenario 2 : {200, 1440, 720} Scenario 3 : {300, 2160, 1080}
Replay memory size	80, 000
Batch size	64
Update target frequency, B	1000
Learning rate	0.00025
Loss function	MSE
Optimizer	RMSprop
Maximum value of ε, $\varepsilon_{\max}$	1
Minimum value of ε, $\varepsilon_{\min}$	0.01
Speed of decay, λ	0.001
No. of epochs per training	1

3.3 DQL Modelini Test Etme

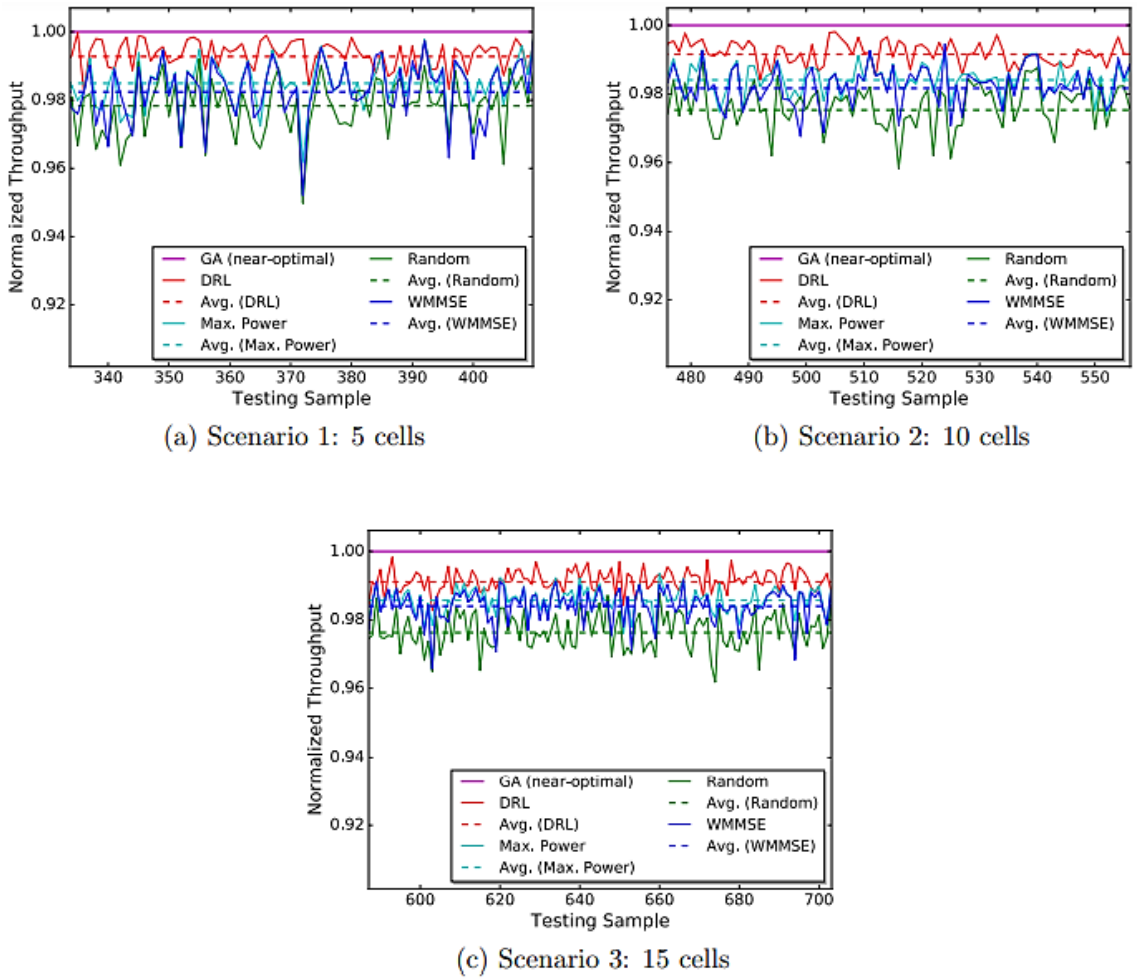
DQL modeli için yaklaşık 80 saatlik eğitimden sonra modelimizi optimum güç tahsisine eşitliyoruz. En hızlı güç tahsisi GA tarafından kullanılır. Tekniğimiz, WMMSE [47], Maksimum güç ataması (MPA) ve rastgele güç ataması gibi mevcut güç atama modelleriyle karşılaştırılabilir şekilde çalışır. Her alt bant için 12,8 W güç kullanıyoruz.

3.4 Sonuçlar ve Tartışmalar

Toplam ağ çıktısının yanı sıra, birkaç PA modelinden üretilen bir güç dağıtım çözümü ve aynı zamanda neredeyse optimal bir GA'dan türetilen PA çözümü de belirlenir. Yakın tarihli bir araştırmaya göre, GA çözümünün toplam ağ verimliliğini bölmek, çeşitli PA modelleri için

normalleştirilmiş ağ çıktısı verir. Şekil 7'de gösterildiği gibi, çeşitli ağ koşullarının test örneklerine kıyasla çeşitli PA modellerinin tek tip verimliliği bulunabilir. Şekil, önerilen DRL tabanlı PA modelinin, şekilde gösterildiği gibi alternatif PA modellerinden daha üstün olduğunu göstermektedir.

Kablosuz ağ boyutunun DRL performansı üzerindeki etkisi: Önerilen DRL tabanlı PA modelimiz, 0,99276 senaryosu olan bir ortalama normalleştirilmiş verime sahiptir, 2, 0,99157 ve 3, 0,99109'dur. Şekil 7, önerilen DRL tabanlı PA'nın verimliliğini göstermektedir. model, artan kablosuz ağ boyutuyla (yani hücre sayısı) istikrarlı bir şekilde azalmaktadır. Kablosuz ağ boyutundaki artışla birlikte ortalama normalleştirilmiş ağ verimi azalır. Bunun nedeni, kablosuz ağ boyutunun durum alanını ve eylem alanını da artırmasıdır. Bu nedenle DQN, eylem için en uygun stratejiyi oluşturmak için daha fazla eylem alanı keşfetmelidir. Bu nedenle, devlet operasyonunun geniş alanları için daha fazla araştırmaya ihtiyaç vardır. Bu nedenle, DRL model performansımız, kablosuz ağ boyutunun artmasıyla giderek daha da kötüleşiyor.



(a) Scenario 1: 5 cells

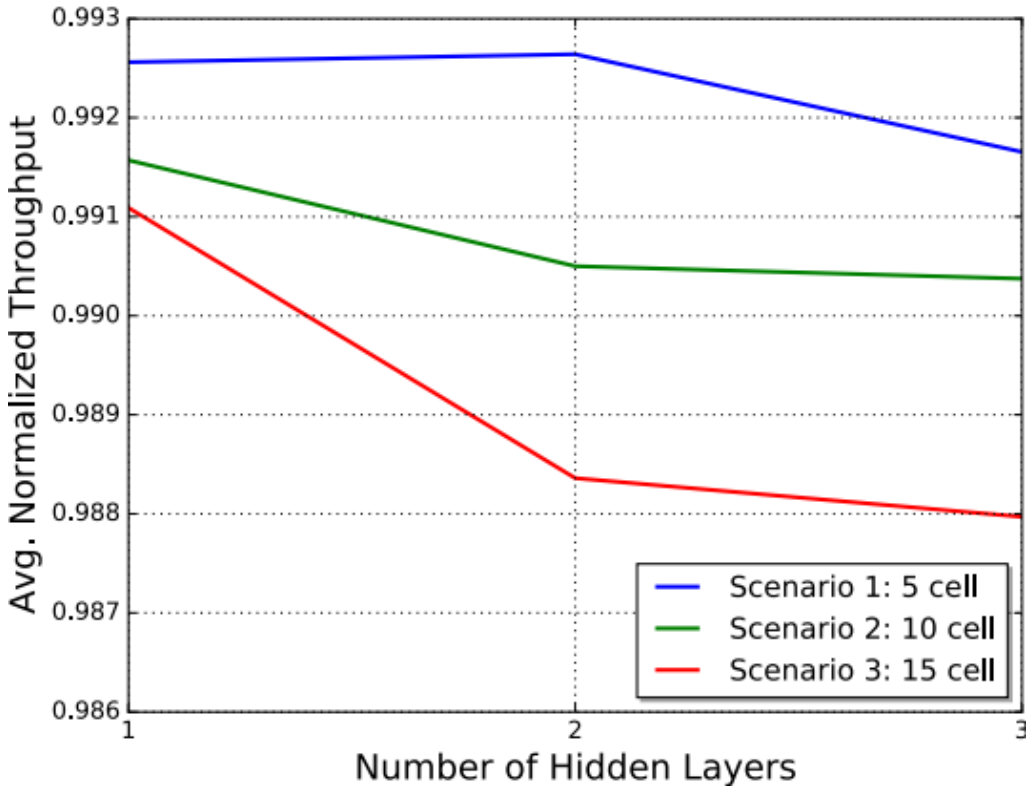
(b) Scenario 2: 10 cells

(c) Scenario 3: 15 cells

Şekil 7. Normalleştirilmiş çıktı ile test örnekleri karşılaştırması

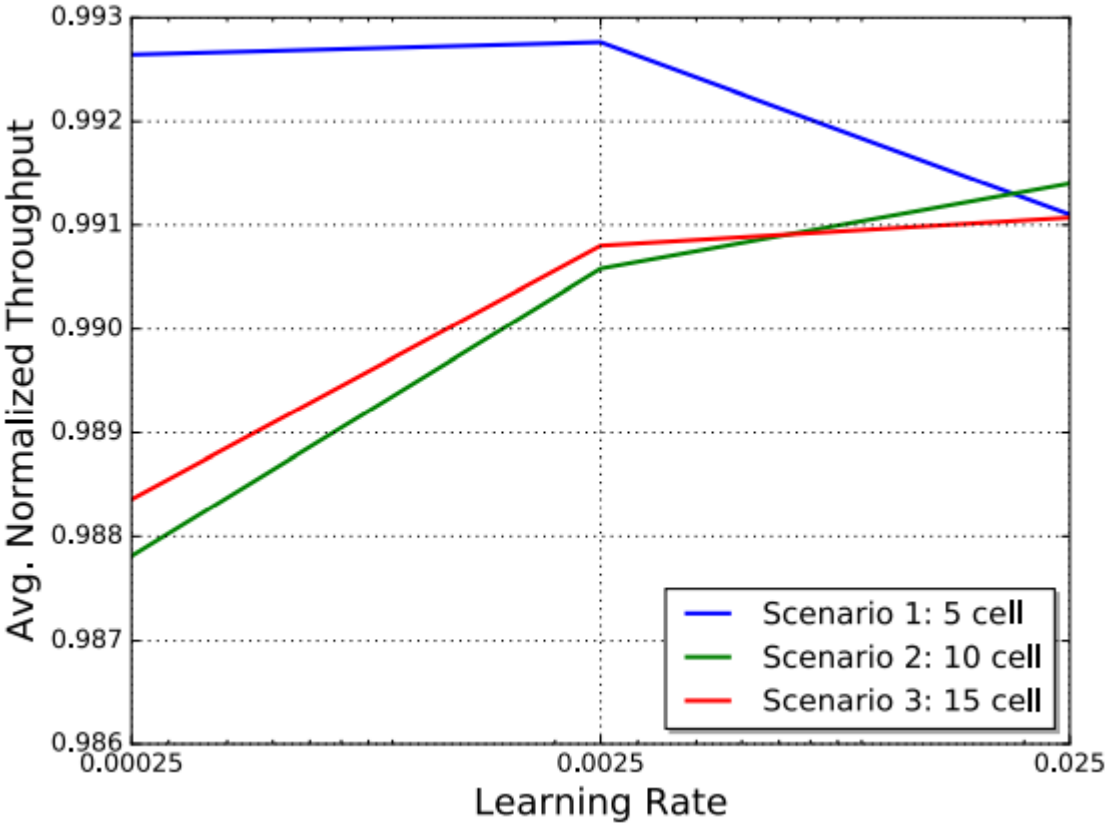
DQN'nin gizli katman boyutunun DRL performansı üzerindeki etkisi: DQN, DQN'deki (Q) yaklaşık olarak gizli katman sayısı olduğu için önemli bir parametre. DQN'den durum ve davranış arasındaki ilişki hakkında değerli bilgiler çıkararak. Daha gizli DQN katmanları, daha fazlasını

öğrenmenize olanak tanır. Simülasyonları DQN'nin gizli katman boyutuna göre değiştirir ve tekrarlarız. Şekil 8, ağ senaryosu için gizli katman sayısına kıyasla ortalama standartlaştırılmış model çıktısını göstermektedir. Artan gizli DQN katman boyutu ile DRL modelinin verimliliğinin biraz düştüğü açıktır. DQN, daha fazla gizli katmanın büyümesiyle alakasız işlevleri (gürültü) öğrendiğinden, bu fazla uydurma kaçınılmaz olarak DQN'nin çıktısını düşürür.



Şekil 8. Gizli katman numarasına karşı ortalama yapılandırılmış çıktı.

Öğrenme hızının DRL performansı üzerindeki etkisi: Öğrenme hızı, eğitim sırasında DQN'deki ağırlık değişikliklerini düzenleyen temel bir hiperparametredir. Bir DQN'nin verilerden hızlı mı yoksa yavaş mı öğrenip öğrenemeyeceğini test eder. Optimum öğrenme hızının elde edilmesi zordur, çünkü küçük bir öğrenme oranı daha fazla eğitime yol açabilir ve yüksek bir öğrenme oranı eğitimde belirsizliğe yol açabilir. Ardından, diğer parametreleri korumak için DRL'nin eğitilme hızını ayarlıyoruz. Çeşitli ağ senaryoları için öğrenme oranına göre ortalama yapılandırılmış model Şekil 9'da gösterilmektedir. 0,0025'i tarayın ve senaryo 0,025 ile senaryo 2 ve senaryo 3, senaryo 1 için en uygun öğrenme oranlarıdır.

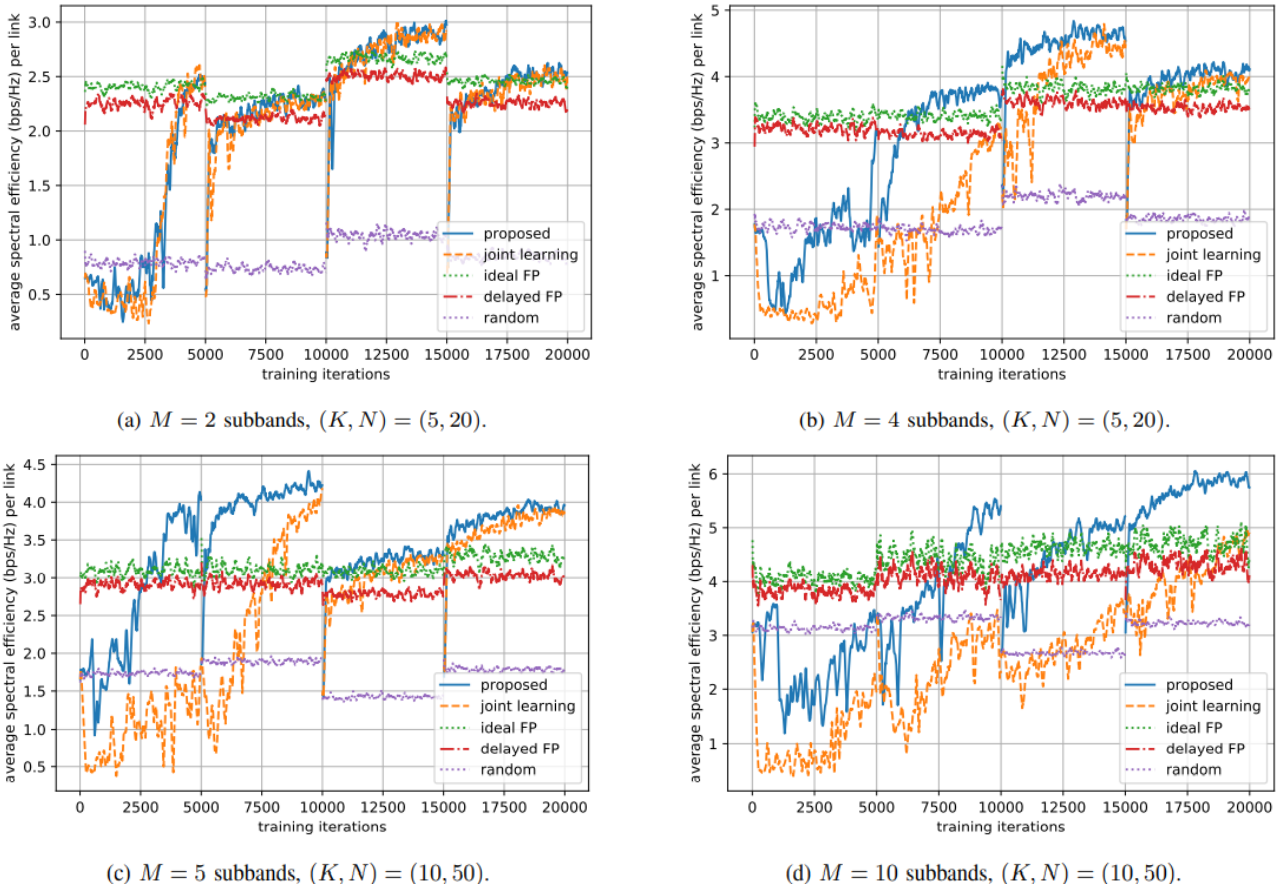


Şekil 9. Ortalama normalleştirilmiş çıktı ile öğrenme oranı karşılaştırması

Önerilen stratejiyi diğer dört seçeneğe karşı değerlendiriyoruz. [23,24]'e göre ilki işbirlikli öğrenme yöntemidir. İletim gücü on seviyeye ayrılmıştır. İkincisi ise "ideal FP" olarak bilinir. Kesirli programlama yapabilmek için tüm durum bilgilerinin her zaman mevcut olduğunu (CSI) varsaymak gerekir. İlk senaryo, merkezi optimizasyon yürütmesindeki veya optimizasyon bulgularının vericilere yayılmasındaki gecikmeleri gözden kaçırır. Başka bir kesirli programlama kıyaslaması, prosedürü çalıştırmak için tek bir zaman dilimi gecikmesini varsaydığından "gecikmeli FP" olarak adlandırılır. Son testte, her verici rastgele bir alt bant seçer ve her zaman çerçevesinin başlangıcında gücü yayınlar.

Her eğitim bölümü 5.000 zaman aralığı sürer ve her bölüm daha küçük parçalara bölünür. Her bölümün başında rastgele yeni bir dağıtım seçiyoruz ve sonunda keşif ve öğrenme oranı parametrelerini sıfırlıyoruz. Belirleyici ilke çıktısına gürültü eklemek yerine, daha hızlı bir yakınsama sonucu elde etmek için Q-learning'in e-açgözlü tekniğini kullanıyoruz. Kaynak kodu hem uygulamayı hem de hiper parametreleri içerir.

Önerilen ve birleşik pekiştirmeli öğrenme şemasının eğitim yakınsaması Şekil 10'da gösterilmektedir. Şekil 10a'ya göre, M = 2 alt bantlarının yakınsama oranları çok benzerdir. Bu prosedür, daha fazla alt bant eklendiğinden önerilen yöntem kadar iyi çalışmaz. Ortak öğrenme eylem uzayının artması ve derin Q-ağ çıktı katmanı karmaşıklığının artması bunun ana nedenleridir.



(a) $M = 2$ subbands, $(K, N) = (5, 20)$.

(b) $M = 4$ subbands, $(K, N) = (5, 20)$.

(c) $M = 5$ subbands, $(K, N) = (10, 50)$.

(d) $M = 10$ subbands, $(K, N) = (10, 50)$.

Şekil 10. Eğitim yakınsaması

Tablo 3, öğretilen politikaların çok sayıda rastgele durumda uygulanmasının sonuçlarını özetlemektedir. Test edilmemiş ilkelerin yeni dağıtımlarda faydalı olduğu gösterilmiştir ve önerilen yaklaşım, denenen önceki yöntemlerden daha ölçeklenebilirdir.

Tablo 3. Test sonuçları.

(K, N) (hücreler, bağlantılar)	M alt bantlar	average sum-rate performance in bps/Hz per link					çıktı katmanı boyutu takviye öğrenme önerilen ortak	ortalama yinelemeler FP
		güçlendirme önerilen	öğrenme eklem [58]	ideal FP	diğer şemalar gecikmiş FP	rastgele		
(5, 20)	1	1.51	1.50	1.58	1.46	0.41	1 + 1 10	70.30
	2	2.63	2.64	2.66	2.46	0.99	2 + 1 20	102.08
	4	4.57	4.38	3.81	3.57	2.12	4 + 1 40	122.15
(10, 50)	1	1.26	1.26	1.31	1.21	0.25	1 + 1 10	72.83
	2	2.08	2.10	2.08	1.92	0.59	2 + 1 20	96.32
	4	3.34	3.34	2.90	2.68	1.31	4 + 1 40	185.93
	5	3.79	3.76	3.18	2.94	1.64	5 + 1 50	206.38
	10	5.71	4.41	4.44	4.08	2.99	10 + 1 100	287.70

4. SONUÇ

Bu çalışmada DRL'ye dayalı yeni bir 5G güç tahsisi yaklaşımı geliştirdik. Önerilen strateji için DQL'yi deneyim tekrarı ile birlikte kullandığımızı belirtmek önemlidir. Bir gizli katmana sahip bir DQN kullanarak, simülasyon sonuçları, senaryomuzdaki eylem-değer fonksiyonuna yaklaşmanın yeterli olduğunu göstermektedir. DRL'de en kritik hiperparametre öğrenme oranıdır ve ideal öğrenme oranını belirlemek zordur. Farklı ağ koşulları için en uygun öğrenme oranını keşfetmek için öğrenme oranını değiştirdik ve önerilen modelin performansını gözlemledik. Bu DRL tabanlı güç tahsisi yaklaşımı, çeşitli ağ koşullarında test edilmiş ve WMMSE, maksimum güç tahsisi ve rastgele güç tahsisi gibi diğer yöntemler kadar etkili olduğu bulunmuştur. Bu nedenle, simülasyonlarda diğer PA modellerinden daha iyi performans gösterir ve büyük ölçekli senaryolara ölçeklenebilir olduğunu gösterir.

KAYNAKLAR

Li, R., Zhao, Z., Zhou, X., Ding, G., Chen, Y., Wang, Z., ve Zhang, H. (2017). Intelligent 5G: When cellular networks meet artificial intelligence. *IEEE Wireless communications*, *24*(5), 175-183.

Wang, M., Cui, Y., Wang, X., Xiao, S., ve Jiang, J. (2017). Machine learning for networking: Workflow, advances and opportunities. *Ieee Network*, *32*(2), 92-99.

Zhang, C., Patras, P., ve Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials*, *21*(3), 2224-2287.

Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., ve Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, *21*(4), 3133-3174.

Osseiran, A., Boccardi, F., Braun, V., Kusume, K., Marsch, P., Maternia, M., ... ve Fallgren, M. (2014). Scenarios for 5G mobile and wireless communications: the vision of the METIS project. *IEEE communications magazine*, *52*(5), 26-35.

Hossain, E., Rasti, M., Tabassum, H., ve Abdelnasser, A. (2014). Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective. *IEEE Wireless communications*, *21*(3), 118-127.

Zander, J. (1992). Performance of optimum transmitter power control in cellular radio systems. *IEEE transactions on vehicular technology*, *41*(1), 57-62.

Foschini, G. J., ve Miljanic, Z. (1993). A simple distributed autonomous power control algorithm and its convergence. *IEEE transactions on vehicular Technology*, *42*(4), 641-646.

Yates, R. D. (1995). A framework for uplink power control in cellular radio systems. *IEEE Journal on selected areas in communications*, *13*(7), 1341-1347.

Sung, C. W., ve Leung, K. K. (2005). A generalized framework for distributed power control in wireless networks. *IEEE Transactions on Information Theory*, *51*(7), 2625-2635.

Boche, H., ve Schubert, M. (2010). A unifying approach to interference modeling for wireless networks. *IEEE Transactions on Signal Processing*, *58*(6), 3282-3297.

Shen, K., ve Yu, W. (2018). Fractional programming for communication systems—Part I: Power control and beamforming. *IEEE Transactions on Signal Processing*, *66*(10), 2616-2630.

Meng, F., Chen, P., Wu, L., ve Wang, X. (2018). Automatic modulation classification: A deep learning enabled approach. *IEEE Transactions on Vehicular Technology*, *67*(11), 10760-10772.

Ye, H., Li, G. Y., ve Juang, B. H. (2017). Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Communications Letters*, *7*(1), 114-117.

Meng, F., Chen, P., ve Wu, L. (2018). NN-based IDF demodulator in band-limited communication system. *IET Communications*, *12*(2), 198-204.

Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., ve Sidiropoulos, N. D. (2018). Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, *66*(20), 5438-5453.

Liang, F., Shen, C., Yu, W., ve Wu, F. (2019). Towards optimal power control via ensembling deep neural networks. *IEEE Transactions on Communications*, *68*(3), 1760-1776.

Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., ve Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, *21*(4), 3133-3174.

Lin, X., Li, J., Baldemair, R., Cheng, J. F. T., Parkvall, S., Larsson, D. C., ... ve Werner, K. (2019). 5G new radio: Unveiling the essentials of the next generation wireless access technology. *IEEE Communications Standards Magazine*, *3*(3), 30-37.

Z Koo, J., Mendiratta, V. B., Rahman, M. R., ve Walid, A. (2019, October). Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics. In *2019 15th International Conference on Network and Service Management (CNSM)* (pp. 1-5). IEEE.

Galindo-Serrano, A., ve Giupponi, L. (2010). Distributed Q-learning for aggregated interference control in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, *59*(4), 1823-1834.

Simsek, M., Czylwik, A., Galindo-Serrano, A., ve Giupponi, L. (2011, June). Improved decentralized Q-learning algorithm for interference reduction in LTE-femtocells. In *2011 Wireless Advanced* (pp. 138-143). IEEE.

Simsek, M., Bennis, M., ve Güvenç, I. (2014). Learning based frequency-and time-domain inter-cell interference coordination in HetNets. *IEEE Transactions on Vehicular Technology*, *64*(10), 4589-4602.

Riedmiller, M. (2005, October). Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European conference on machine learning* (pp. 317-328). Springer, Berlin, Heidelberg.

Ghadimi, E., Calabrese, F. D., Peters, G., ve Soldati, P. (2017, May). A reinforcement learning approach to power control and rate adaptation in cellular networks. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1-7). IEEE.

Nasir, Y. S., ve Guo, D. (2018). Deep reinforcement learning for distributed dynamic power allocation in wireless networks. *arXiv preprint arXiv:1808.00490*, *8*, 2018.

Meng, F., Chen, P., ve Wu, L. (2019, May). Power allocation in multi-user cellular networks with deep Q learning approach. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... ve Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3), 229-256.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... ve Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Meng, F., Chen, P., Wu, L., ve Cheng, J. (2020). Power allocation in multi-user cellular networks: Deep reinforcement learning approaches. *IEEE Transactions on Wireless Communications*, *19*(10), 6255-6267.