UNIVERSIDADE FEDERAL DE GOIÁS

ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E COMPUTAÇÃO

ADRIANO FERREIRA LOPES
JEAN LUCAS BARBOSA SILVA

ALOCAÇÃO DE RECURSOS EM REDES SEM FIO UTILIZANDO ALGORITMOS BASEADOS EM APRENDIZAGEM DE MÁQUINA

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as):

Jean Lucas Barbosa Silva
Adriano Ferreira Lopes

Título do trabalho: Alocação de recursos em redes sem fio utilizando algoritmos baseados em Aprendizagem de Máquina

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Flavio Henrique Teles Vieira**, **Professor do Magistério Superior**, em 01/02/2024, às 18:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Jean Lucas Barbosa Silva**, **Discente**, em 01/02/2024, às 20:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Adriano Ferreira Lopes**, **Discente**, em 01/02/2024, às 20:12, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4321779** e o código CRC **5A9B47BC**.

Referência: Processo nº 23070.057350/2023-14 SEI nº 4321779

ADRIANO FERREIRA LOPES
JEAN LUCAS BARBOSA SILVA

ALOCAÇÃO DE RECURSOS EM REDES SEM FIO UTILIZANDO ALGORITMOS BASEADOS EM APRENDIZAGEM DE MÁQUINA

Trabalho de conclusão de curso apresentado na Escola de Engenharia Elétrica, Mecânica e de Computação como requisito para a conclusão do curso de Engenharia de Computação e obtenção do título de Engenheiro de Computação.

Orientador: Flávio Henrique Teles Vieira

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Silva, Jean Lucas Barbosa
Alocação de recursos em redes sem fio utilizando algoritmos baseados em Aprendizagem de Máquina [manuscrito] / Jean Lucas Barbosa Silva, Adriano Ferreira Lopes. - 2024.
XIV, 14 f.: il.

Orientador: Prof. Dr. Flávio Henrique Teles Vieira.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Engenharia da Computação, Goiânia, 2024.
Bibliografia.
Inclui siglas, abreviaturas, símbolos, algoritmos, lista de figuras.

1. Alocação de recursos em redes sem fio. 2. DQN adaptativa. 3. Entropia Cruzada. 4. QoS. I. Lopes, Adriano Ferreira. II. Vieira, Flávio Henrique Teles, orient. III. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Ao primeiro dia do mês de fevereiro do ano de 2024 iniciou-se a sessão pública de defesa do Trabalho de Conclusão de Curso (TCC) intitulado “Alocação de recursos em redes sem fio utilizando algoritmos baseados em Aprendizagem de Máquina”, de autoria de Jean Lucas Barbosa Silva e Adriano Ferreira Lopes, do curso de engenharia de computação da Escola de Engenharia Elétrica, Mecânica e de Computação da UFG. Os trabalhos foram instalados pelo Prof. Dr. Flávio Henrique Teles Vieira EMC UFG com a participação dos demais membros da Banca Examinadora: Alisson Assis Cardoso EMC UFG e Daniel Porto Queiroz Carneiro-Petrobrás. Após a apresentação, a banca examinadora realizou a arguição do(a) estudante. Posteriormente, de forma reservada, a Banca Examinadora atribuiu a nota final de 9,0 (nove) , tendo sido o TCC considerado aprovado.

Proclamados os resultados, os trabalhos foram encerrados e, para constar, lavrou-se a presente ata que segue assinada pelos Membros da Banca Examinadora

Documento assinado eletronicamente por **Flavio Henrique Teles Vieira**, **Professor do Magistério Superior**, em 23/02/2024, às 19:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Alisson Assis Cardoso**, **Professor do Magistério Superior**, em 23/02/2024, às 19:33, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Daniel Porto Queiroz Carneiro**, **Usuário Externo**, em 23/02/2024, às 21:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4405149** e o código CRC **6BF489C2**.

Alocação de recursos em redes sem fio utilizando algoritmos baseados em Aprendizagem de Máquina

Adriano Ferreira Lopes[1], Jean Lucas Barbosa Silva[2], Flávio Henrique Teles Vieira[3]
Universidade Federal de Goiás (UFG) - Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) - Goiânia, Goiás, Brasil 74605-010. E-mails: ferreira_adriano@discente.ufg.br[1], jean.silva@discente.ufg.br[2], flavio_vieira@ufg.br[3]

***Resumo*—Com o crescente aumento no número de dispositivos móveis e dispositivos IoT inteligentes, as redes sem fio estão se tornando cada vez mais complexas, autônomas e heterogêneas em termos de tipos de arquiteturas de rede que incorporam. Com essas redes, alocar e gerenciar recursos de forma eficiente para os usuários é um grande problema a ser resolvido. Neste contexto, espera-se que técnicas de aprendizagem por reforço profundo sejam umas das principais tecnologias utilizadas para aumentar a eficiência na alocação dinâmica de recursos. No presente trabalho é apresentada uma proposta de alocação de recursos em redes sem fio, denominada Aprendizado por Reforço com Entropia Cruzada, a fim de maximizar a eficiência energética e atender os requisitos de qualidade de serviço (QoS) dos usuários. A abordagem considera um sistema de comunicação multiusuários e multiobjetivo baseado na tecnologia CP-OFDM (*Cyclic-Prefix Orthogonal Frequency Division Multiplexing*) utilizando o método de aprendizado por reforço, especificamente uma rede DQN (*Deep Q-Network*) associada ao algoritmo de Entropia Cruzada para obter uma política ótima de alocação de recursos. A implementação abrange parâmetros do sistema, como largura de banda, modulação, números de usuários e tamanho médio do pacote, além de detalhar a estrutura dos elementos de recursos, a distribuição de subportadoras e a capacidade de transmissão nos diferentes modos de modulação presentes no *frame* do sistema LTE (*Long Term Evolution*). Os resultados das simulações mostram que o método proposto tem boas características de convergência, além de superar os resultados obtidos pela abordagem tradicional da rede DQN sem o uso de Entropia Cruzada.**

***Palavras-chave*—Alocação de recursos em redes sem fio, DQN adaptativa, Entropia Cruzada, QoS**

***Abstract*—With the increasing rise in the number of mobile devices and intelligent IoT devices, wireless networks have become increasingly complex, autonomous, and heterogeneous in terms of the types of network architectures they incorporate. Within these networks, efficiently allocating and managing resources for users is a significant problem to be solved. In this context, deep reinforcement learning techniques are expected to be among the main technologies used to achieve global optimization in dynamic resource allocation. This paper presents a proposal for resource allocation in wireless networks, termed Cross-Entropy Reinforcement Learning, in order to maximize energy efficiency and meet users' quality of service (QoS) requirements. The approach considers a multi-user, multi-objective communication system based on CP-OFDM (Cyclic-Prefix Orthogonal Frequency Division Multiplexing) technology using reinforcement learning methods, specifically a Deep Q-Network (DQN) associated with the Cross-Entropy algorithm to obtain an optimal resource allocation policy. The implementation encompasses system parameters such as bandwidth, modulation, number of users, and average packet size, while detailing the structure of resource elements, subcarrier distribution, and transmission capacity in different modulation modes within the LTE (Long Term Evolution) system frame. Simulation results demonstrate that the proposed method exhibits good convergence characteristics and performs better than the traditional DQN approach without cross-entropy.**

***Index Terms*—Resource allocation in wireless networks, Adaptative DQN, Cross-Entropy, QoS**

I. Introdução

NA última década, o crescente interesse na Internet das Coisas (IoT) experimentou um notável aumento, impulsionado principalmente pela interconexão de dispositivos inteligentes habilitados por tecnologias sem fio, juntamente com sensores e atuadores, possibilitando sua conexão à Internet e a capacidade de comunicação [1].

Internet das Coisas (IoT) refere-se à interconexão de dispositivos físicos (como sensores, atuadores, veículos, eletrodomésticos, entre outros) por meio da internet, permitindo a coleta e troca de dados. Tais dispositivos são equipados com tecnologia incorporada para interagir com o ambiente e com outros dispositivos, geralmente por meio de sensores e conectividade de rede. Segundo especialistas, é previsto que até 2025 mais de 75 bilhões de dispositivos estarão conectados à IoT. Esse grande volume de dispositivos conectados exigirá recursos de redes sem fio cada vez mais capacitados para que possam suportá-los [2].

Long Term Evolution (LTE) é a quadragésima geração de sistema móvel celular que foi implantado e especificado no 3GPP [3]. Esse sistema suporta larguras de banda flexíveis e usa o sistema de Multiplexação por Divisão Ortogonal de Frequência (OFDM) no *downlink*, adequado para alcançar altas taxas de pico de dados em largura de banda de alto espectro. Esse sistema combina a modulação e multiplexação de sinal, onde a multiplexação permite a transmissão simultânea de múltiplos sinais através de um único enlace de dados. O sinal é dividido em múltiplos canais de banda estreita, conhecidos como subportadoras, operando em diferentes frequências. O OFDMA é altamente flexível na canalização e, além disso, o LTE proporciona altas taxas de dados e baixa latência, melhorando a capacidade e cobertura do sistema. A fim de alcançar um ótimo desempenho, a programação e seleção flexível de largura de banda desses sistemas é importante.

Todavia, os principais desafios enfrentados pelas redes móveis e sistemas IoT atualmente são impulsionados pelas crescentes exigências de desempenho, decorrentes da presença de diversos dispositivos heterogêneos com recursos limitados,

tais como energia e capacidade de processamento. Além disso, a escassez de espectro disponível também representa uma barreira significativa. Muito destaque tem sido dado à melhoria da capacidade dos sistemas de comunicação, bem como otimizar a utilização de seus recursos. À medida que tecnologias imersivas, como a realidade virtual, juntamente com a crescente necessidade de transmissão de vídeo em alta resolução via redes 5G, que exigem maior tráfego de dados, surge uma demanda crescente por latências extremamente baixas e altas taxas de transferência de dados [4].

Nesse sentido, a alocação eficiente desses recursos é fundamental para garantir desempenho, confiabilidade e a eficiência dos sistemas de comunicação sem fio, uma vez que estratégias de alocação ineficazes diminuem a eficiência espectral da rede. O objetivo desse processo é atribuir recursos limitados a diferentes usuários, a fim de maximizar a eficiência energética e melhorar os parâmetros de qualidade de serviço (QoS).

Para resolver problemas de alocação de recursos, são frequentemente utilizadas técnicas de otimização, que envolvem a busca de uma solução ótima ou subótima para o problema. Em cenários de rede com canais sem fio intrincados e requisitos variados de QoS, a otimização se torna um desafio significativo, especialmente quando os problemas parecem ser não-convexos. Em tais situações, a busca por soluções ótimas pode ser complexa devido à presença de múltiplos mínimos locais e à dificuldade de convergência para o mínimo global. Técnicas de metaheurísticas podem ser utilizadas para encontrar soluções próximas do ótimo global em problemas complexos [5]. Todavia, apesar da sua eficácia, as metaheurísticas podem resultar em considerável complexidade computacional, tornado menos desejáveis para redes sem fio de grande escala e com um grande número de conexões e requisitos distintos.

Nesse sentido, o Aprendizado por Reforço (*RL Reinforcement Learning*), uma das ferramentas da Aprendizagem de Máquina (ML), demonstrou ser capaz de capacitar os dispositivos IoT a tomar decisões autônomas para a alocação de recursos em sistemas de comunicação sem fio inteligentes emergentes [6]. Normalmente, esses métodos são livres de modelos e são orientados por dados. O RL foi desenvolvido como um algoritmo baseado em objetivos a fim de aprender uma política ótima. Além disso, as redes sem fio operam em um ambiente dinâmico e incerto, o que torna os impasses ainda mais desafiadores. O agente de RL deve ser capaz de compreender a dinâmica do ambiente, explorando novas regiões e explorando a informação existente, sem a necessidade de conhecimento prévio sobre ele, por meio da análise de dados coletados, e tomar as ações mais adequadas para alcançar seu objetivo.

II. Revisão Bibliográfica

A seguir, são apresentados trabalhos relacionados ao uso de algoritmos e técnicas de Aprendizado por Reforço para aprimorar sistemas de telecomunicações sem fio, buscando uma maior eficiência energética e qualidade de transmissão para múltiplos usuários.

O estudo intitulado "*Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial*"[7] representa uma abordagem abrangente que investiga a integração do aprendizado de máquina nas redes sem fio, destacando o papel das redes neurais artificiais (ANNs) na resolução de uma variedade de desafios inerentes a essas redes. Este trabalho discute diversos tópicos, englobando desde uma introdução às ANNs até os tipos específicos relevantes para aplicações em redes sem fio. Além disso, explora de maneira aprofundada como as redes neurais podem ser empregadas eficazmente para solucionar problemas relacionados à comunicação sem fio.

O estudo conduzido por Zhu [8] apresenta um algoritmo de aprendizado por reforço aplicado a um sistema IoT multiusuários. O artigo aborda a aplicação de tecnologia cognitiva para melhorar a eficiência da transmissão de pacotes em aplicações de IoT. Utiliza o modelo MDP para descrever o problema do agendamento de transmissão. O algoritmo Q é modificado para aprender a transição de estado sem informações prévias, sendo adotado para mapear a relação entre estados e ações, evitando cálculos massivos. Destaca-se também o uso de um agente único de controle que atende um usuário de cada vez, multiplexando o atendimento no tempo (*Time Division Multiplexing*). No entanto, mesmo considerando a velocidade relativa entre transmissor e receptor, o texto não aborda as distâncias entre eles nem suas velocidades absolutas.

No trabalho proposto por [9], foi apresentado um algoritmo de alocação de recursos em redes sem fio multiportadoras com ondas milimétricas, utilizando aprendizado por reforço baseado em modelo Markoviano, além disso, foi proposto uma rede *Deep Q-Network* (DQN) para resolver o problema de alocação em cenários SISO (Single Input Single Output) OFDM. Esses algoritmos buscam maximizar a eficiência energética e a taxa de transmissão dos usuários, levando em consideração diversas restrições do sistema, como largura de banda disponível, potência máxima de transmissão e qualidade do canal, entre outras. A pesquisa realizou experimentos em um ambiente simulado, comparando o desempenho do algoritmo proposto com outros métodos de alocação de recursos em redes sem fio. Os resultados indicam que os algoritmos propostos conseguem melhorar de forma significativa a eficiência energética e a taxa de transmissão de dados dos usuários, em comparação com os demais algoritmos testados.

De maneira análoga ao estudo conduzido por [9], o presente trabalho adota uma abordagem que utiliza uma rede DQN para resolver o problema de alocação de recursos, porém, é acrescido a técnica de Entropia Cruzada para aprimorar o desempenho da rede neural. Diante disso, é realizado um comparativo de estudo entre as duas abordagens, analisando métricas de desempenho, convergência do algoritmo e a eficiência na alocação de recursos.

III. Metodologia

Entender a abordagem necessária para resolver o problema desempenhou um papel significativo no desenvolvimento da solução proposta neste trabalho. A seguir, são apresentadas a motivação, objetivos e as etapas estabelecidas para o desenvolvimento do trabalho.

A. Motivação

Com o crescente número de dispositivos móveis, incluindo smartphones, tablets e dispositivos vestíveis, juntamente com a crescente expansão da IoT, que abrange desde dispositivos domésticos inteligentes até aplicações industriais, a demanda por comunicações sem fio eficientes de alta capacidade tem aumentado exponencialmente [10].

Estas redes expansivas colocam desafios significativos devido à sua natureza complexa, operando em uma ampla variedade de cenários de comunicação que abrange ambientes urbanos, interiores, rurais e industriais. Cada um desses ambientes apresenta desafios específicos em termos de propagação e interferência de sinal.

Além disso, o volume crescente de tráfego de dados resultou numa escassez de espectro de frequência disponível, necessitando de uma otimização estratégica da alocação de recursos. Simultaneamente, o uso generalizado de dispositivos alimentados por bateria aumenta a importância da eficiência energética com uma preocupação crítica.

B. Objetivos

Propor um algoritmo de alocação de recursos em redes sem fio, visando maximizar a eficiência energética e a taxa de transmissão de dados dos usuários, levando em consideração as restrições do sistema, como a largura de banda disponível, a potência máxima de transmissão e a qualidade do canal, entre outros.

Para atingir esse objetivo, adotou-se o método de Entropia Cruzada (CE) em uma arquitetura do tipo *Deep Q-Network* (DQN), revelando-se eficaz em contextos que envolvem a tomada de decisões complexas. As simulações conduzidas corroboram que a abordagem proposta pode melhorar a eficiência da rede.

C. Etapas

Inicialmente, são apresentados os principais conceitos e termos empregados ao longo do artigo, visando facilitar a compreensão e embasar as decisões tomadas durante a pesquisa.

A seção subsequente é dedicada ao desenvolvimento, na qual abordaremos o processo de programação e tomada de decisões, explicando as técnicas e motivações por trás delas. Posteriormente, nos concentramos nos resultados do trabalho, destacando os processos alcançados.

Na seção final, apresentaremos nossas conclusões, avaliando as entregas e realizando uma análise das expectativas geradas pelo estudo.

IV. Conceitos Importantes

A seguir, são apresentados alguns conceitos importantes que ajudam a entender o estudo do trabalho.

A. Quality of Service (QoS)

A qualidade de serviço é um conjunto de parâmetros que descrevem a qualidade da experiência do usuário em um sistema de comunicação. Esses parâmetros podem incluir a perda de pacotes, a taxa de transferência de dados, a disponibilidade e a confiabilidade dos serviços.

A QoS é de grande importância em sistemas de comunicação, pois permite que os usuários tenham uma experiência satisfatória e previsível, mesmo em condições adversas que podem afetar a transmissão de dados, como alto fluxo de tráfego e interferência.

Além disso, pode ser garantido por meio de técnicas de alocação de recursos, como priorização de tráfego, gerenciamento de filas, controle de congestionamento, que garantem que os recursos de comunicação sejam alocados de forma justa e eficiente para atender às necessidades dos usuários.

B. Algoritmos de Aprendizado por Reforço

Algoritmos de aprendizado por reforço são a base central desse trabalho, sendo responsáveis pela alocação adaptativa de recursos do nosso sistema.

O aprendizado por reforço é uma abordagem de *Machine Learning*, responsável por capacitar um agente a tomar decisões em ambientes incertos e complexos, aprendendo a tomar decisões sucessivas interagindo com o seu entorno e recebendo *feedback* em forma de recompensas ou punições, dependendo da qualidade de suas ações.

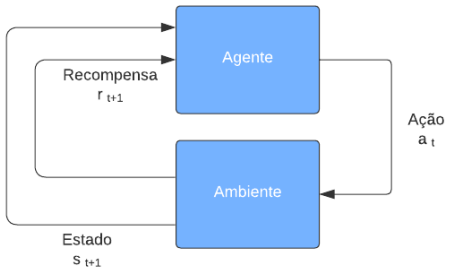


Figura 1: Modelo de aprendizado por reforço. Fonte: Próprios autores.

Para compreender melhor o funcionamento dos algoritmos de aprendizado por reforço, é importante entender as peças chaves que modelam essa ferramenta de ML.

1) Agente: O Agente é a nossa entidade, podendo ser um software, um robô, ou qualquer entidade capaz de realizar interação com o ambiente. Ele é responsável por tomar decisões no ambiente, interagindo com ele e obtendo informações através de suas ações em determinado estado do ambiente e recebendo *feedbacks* de ações passadas.

2) Ambiente: O Ambiente é o espaço na qual o agente pode interagir, e na qual ele deve se basear para a tomada de decisões. O ambiente pode ser um espaço real ou virtual, dependendo do problema a ser resolvido.

3) Estado (s): O Estado se refere às condições do Ambiente em um determinado instante de tempo, que inclui informações relevantes para a tomada de decisão do agente. Ele pode ser definido de várias maneiras, dependendo da modelagem do problema em questão, todavia geralmente inclui informações sobre as condições atuais do ambiente, como a posição do

agente, a presença de obstáculos, a disponibilidade de recursos, entre outros.

4) Recompensa (r): É a referência do agente sobre as ações que ele realiza no ambiente. Quando o agente interage com o ambiente por meio de uma ação, ele recebe um *feedback*, podendo ser um valor numérico que indica o quão bom foi o desempenho do agente naquela ação.

5) Política (π): Estratégia aplicada pelo agente para decidir a próxima ação com base no estado atual do ambiente. O objetivo do agente é aprender a melhor política de ações a serem tomadas em determinado ambiente de forma a maximizar os valores de recompensa. Especialmente, é útil em situações em que não é possível obter um conjunto de dados rotulados para treinar um modelo de aprendizado supervisionado. Em vez disso, ele aprende a partir de sua própria experiência, explorando o ambiente e ajustando sua política de ações com base nas recompensas recebidas.

C. Processo de Decisão de Markov (MDP)

Problemas de aprendizado por reforço normalmente são modelados usando o processo de decisão de Markov. É um modelo estocástico que descreve a evolução de um sistema ao longo do tempo, onde a probabilidade de transição de estado depende apenas do estado atual e uma ação aplicada ao sistema.

O método MDP pode ser modelado pela tupla

$$\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \lambda >$$

onde:

- $\mathcal{S}$: conjunto de estados
- $\mathcal{A}$: conjunto de ações
- $\mathcal{P}$: modelo de probabilidade de transição
- r : função de recompensa
- λ : fator de desconto para a importância de recompensas futuras

Essa representação permite que o agente possa tomar decisões sequenciais em um ambiente dinâmico, buscando aprender uma política de ação que maximize a recompensa cumulativa ao longo do tempo.

D. Q-Learning

O *Q-Learning* é o algoritmo de RL que busca aprender a política de ação ótima em um ambiente de tomada de decisão sequencial. É baseado em uma abordagem de aprendizado *off-policy*, ou seja, o agente pode aprender com experiências que não são necessariamente geradas pela política que está tentando melhorar.

O objetivo do *Q-Learning* é aprender uma função de valor Q que estima o valor de uma ação em um determinado estado do ambiente com base nas recompensas recebidas e nas probabilidades de transição de estados observadas. Ele mantém uma tabela de estimativas para as recompensas futuras esperadas para cada ação de um estado.

Essa política de ação é uma função que mapeia cada estado para uma ação, e é utilizada pelo agente para decidir qual ação tomar em cada estado do sistema, a fim de maximizar a recompensa total. A soma das recompensas *R* é dada por:

$$R = \sum_{t=0}^{\infty} \gamma^t r_{t+1} \tag{1}$$

O algoritmo utiliza uma tabela chamada Q, onde cada entrada (estado, ação) armazena um valor chamado de Q-valor. Esses Q-valores representam a qualidade de uma ação em um determinado estado, indicando uma recompensa esperada a longo prazo de se tomar a ação naquele determinado estado.

A função *Q(s, a)* é conhecida como Equação de Bellman [11] expressa da seguinte forma:

$$Q(s_i, a_i) = r(s_i, a_i) + \gamma \sum_{s_0} P(s_i, s_0, a_i) \max_{a_0} Q(s_0, a_0) \tag{2}$$

onde $r(s_i, a_i)$ é a recompensa imediata obtida e γ é o fator de desconto para recompensas futuras, *s, s', a e a'* os estados e ações presentes e futuros respectivamente.

A equação de Bellman trabalha a relação entre o valor de uma ação em um determinado estado do ambiente e o valor das ações nos estados futuros, levando em consideração as recompensas imediatas e as recompensas futuras. Essa relação é fundamental para o processo de aprendizagem do agente, pois permite a atualização dos valores de Q de forma iterativa com base nas recompensas recebidas e nas estimativas de valores das ações nos próximos estados.

E. Função Utilidade

A função utilidade, também conhecida como função objetivo ou função de recompensa, possui um papel fundamental no aprendizado por reforço. Ela é responsável por atribuir um valor numérico que representa as recompensas às ações tomadas em determinado estado do ambiente, permitindo que o agente avalie e selecione as ações que levem a resultados mais satisfatórios.

No contexto do *Q-Learning* e outros algoritmos de aprendizado por reforço, a função objetivo é utilizada para modelar as recompensas imediatas associadas às ações tomadas pelo agente em diferentes estados do ambiente.

Ademais, elas representam uma medida da quantidade que se deseja maximizar ou minimizar, dependendo do contexto do problema. São utilizadas para representar múltiplos objetivos ou critérios de desempenho, modelando uma relação entre as variáveis a fim de buscar uma solução para que o processo de aprendizado por reforço possa convergir para regiões de desempenho desejados.

F. Exploration vs. Exploitation

No contexto do aprendizado por reforço, há dois conceitos que desempenham papéis importantes na tomada de decisão realizada pelo agente: *exploration* e *exploitation*.

Exploration refere-se à busca ativa por novas informações, ou seja, informações desconhecidas ou menos frequentes em um ambiente. Durante a fase de exploração, o agente prioriza a descoberta de novas estratégias, tomando ações que podem não ser óbvias ou que podem trazer recompensas mais favoráveis no longo prazo. Isso é útil, especialmente quando o agente tem um conhecimento limitado sobre o ambiente.

Já nas decisões de *Exploitation*, envolve a utilização das informações e experiências já adquiridas para tomar decisões que maximizem as recompensas de longo prazo. Durante a exploração, o agente se concentra em aproveitar o que ele já tem conhecimento sobre o ambiente para maximizar a recompensa imediata.

Encontrar um ponto de equilíbrio adequado entre esses dois modos é crucial para o sucesso do agente em ambientes de aprendizado por reforço. Se o agente explorar demais, corre o risco de perder oportunidades de obter recompensas significativas devido à subutilização das ações já conhecidas. Por outro lado, se focar apenas na ações já conhecidas, pode ficar preso em estratégias subótimas e não descobrir outras estratégias mais eficazes.

G. Canal de Comunicação

Um canal de comunicação é o meio físico ou lógico através do qual a informação é transmitida de um ponto para o outro. Isso inclui meios de transmissão físico, como fibras ópticas, cabos de cobre, espectro eletromagnético utilizado em comunicações sem fio, ou ainda, em sistemas de comunicação digital.

Tais meios podem ser classificados de acordo com suas características, como largura de banda, ruído, atenuação e distorção. Ademais, podem estar sujeitos a interferências eletromagnéticas, obstruções físicas, entre outros fatores que podem afetar a sua qualidade.

Em sistemas de comunicação, se torna fundamental entender e modelar o comportamento do canal de comunicação, a fim de projetar esquemas de modulação, codificação e correção de erros que permitam a transmissão confiável e eficiente de dados.

H. Política de seleção de ações

A política de seleção de ações refere-se à estratégia que o agente utiliza para tomar decisões sobre quais ações ele realiza em um determinado estado do ambiente, isso se baseia na observação atual do ambiente e nas experiências adquiridas ao longo do tempo, com o objetivo de maximizar as recompensas de longo prazo.

Existem diferentes abordagens de política de seleção de ações, cada uma com suas características particulares, algumas a mencionar:

1) Política de seleção aleatória: essa é uma abordagem onde o agente escolhe de forma aleatória uma ação a partir do conjunto de ações disponíveis em cada estado do ambiente, sem considerar as recompensas associadas a cada ação sobre o ambiente. Frequentemente é uma estratégia utilizada inicialmente na exploração, permitindo que o agente descubra o espaço de ações disponíveis de forma ampla e sem viés.

Embora essa política seja útil em estágios iniciais do aprendizado, ajudando o agente a coletar informações sobre o ambiente, ela geralmente não é eficiente para maximizar as recompensas de longo prazo, uma vez que as ações escolhidas não levam em consideração o seu impacto nas recompensas futuras, resultando em um uso ineficiente do tempo e dos recursos do agente.

Todavia, a política de seleção aleatória pode ser combinada com outras formas de seleção de ações, ajudando a equilibrar a *exploration* e *exploitation* de forma mais dinâmica e eficaz.

2) Política Greedy: o agente escolhe a ação que tem maior estimativa de valor de acordo com sua função de valor. Essa abordagem foca no *exploitation*, onde não possui elemento de exploração.

3) Política ϵ-Greedy: essa é uma das abordagens mais populares e simples no aprendizado por reforço. Ela combina as duas formas de exploração do agente, permitindo-o escolher a ação com probabilidade 1-ϵ, com maior recompensa estimada de acordo com sua função de valor (*exploitation*) e, ocasionalmente, realiza ações aleatórias , com uma probabilidade ϵ (*exploration*), permitindo que o agente explore o ambiente a fim de obter novas informações, enquanto, também, se concentra em ações com altos valores estimados de recompensa.

4) Política Softmax: agente utiliza uma função *softmax* para calcular as probabilidades proporcionais de seleção para cada ação com base em seus valores estimados. Isso permite uma exploração mais suave do espaço de ações. A função *softmax* pode ser modelada pela equação abaixo:

$$P(a_i) = \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_j}} \tag{3}$$

onde:

- $P(a_i)$ é a probabilidade da ação a_i,
- e é a base do logaritmo natural (número de Euler),
- a_i são os logits associados a cada ação,
- $\sum_{j=1}^{n} e^{a_j}$ é a soma das exponencias de todos os logits.

V. Desenvolvimento

A partir das etapas definidas previamente na metodologia, além dos principais conceitos do trabalho explicados, iremos introduzir o desenvolvimento do estudo conforme mostrado ao longo das próximas subseções.

Temos como objetivo nesta seção oferecer uma visão abrangente no desenvolvimento do sistema, enfatizando as etapas de configuração do sistema de comunicação, a organização dos elementos de recursos, a seleção das tecnologias adotadas, além das ferramentas específicas que foram empregadas.

A. Sistema de Comunicação

A tecnologia LTE emprega a técnica de acesso OFDM para transmissão de dados no sentido de *downlink*, permitindo maior eficiência espectral, taxas de dados elevadas e melhorias de capacidade.

O OFDM é uma técnica complexa de multiplexação que se baseia na ideia de dividir o sinal em subportadoras. Nesta técnica, ao invés de separar as portadoras através das bandas de guarda, emprega-se uma particular sobreposição espectral de subportadoras, como demonstrado na Figura 2. Isso resulta em um ganho espectral de até 50% em relação à técnica FDM (*Frequency Division Multiplexing*) [13].

Todavia, pode ocorrer redução na taxa de transmissão (símbolos de duração mais longa no domínio do tempo transmitidos em cada subportadora) que implica na diminuição da

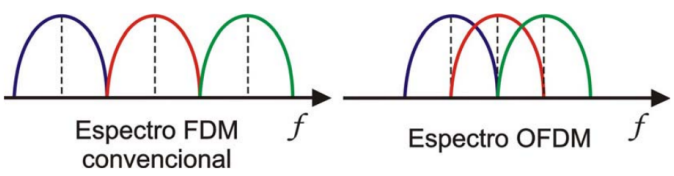


Figura 2: Disposição das portadoras na Modulação FDM e OFDM. Fonte: [12].

sensibilidade à seletividade em frequência, ou seja, o sistema se torna menos capaz de lidar com interferências causadas por diferentes trajetórias do sinal (efeitos de multipercurso).

O canal de desvanecimento multipercurso pode introduzir interferência entre símbolos (ISI), além de interferência entre portadoras (ICI) do sinal OFDM. Para resolver isso, é introduzido um prefixo cíclico (CP - *Cyclic Prefix*), que nada mais é que uma cópia da última parte do símbolo inserida na frente do símbolo, como medida preventiva ao multipercurso. Com um CP maior que o comprimento do canal, os problemas relacionados ao ISI e ICI podem ser eliminados. Além disso, utiliza-se um modelo TDL (Tapped Delay Line) para a modelagem de canal aleatório [15]. Esse modelo ajuda a capturar os efeitos complexos do canal, levando em consideração a propagação multipercurso e os atrasos associados.

Assim, a técnica CP-OFDM permite a transmissão de símbolos de duração mais longa no domínio do tempo, reduzindo a interferência inter simbólica, além de garantir altas taxas de transmissão.

B. Séries de Tráfego

Os sistemas de comunicação são projetados para transmitir informações de um ponto a outro de forma eficiente e confiável. São baseados em princípios fundamentais que permitem a transmissão, recepção e o processamento dos sinais de comunicação. Embora existam sistemas que podem operar de forma isolada, em geral, os sistemas de comunicação mais importantes envolvem a transmissão entre múltiplos terminais, formando as redes de comunicação.

Para o processo de simulação da transmissão de dados foram utilizadas séries de tráfego de MAWI [16]. Essa é uma das séries de tráfego mais utilizadas em estudos de desempenho de redes de comunicação e é mantida pele grupo de pesquisa (*Measurement and Analysis on the WIDE Internet*). Ela é composta por dados reais coletados no *backbone* em redes de comunicação em todo o mundo e é amplamente utilizada para avaliar o desempenho de novos protocolos e algoritmos de roteamento em condições realistas de tráfego. Os dados coletados são disponibilizados publicamente em seu site para *download* e permite que pesquisadores utilizem esses dados em seus estudos e análises de desempenho de redes de comunicação.

C. Blocos de recursos

Os dados são transmitidos em forma de bits, a menor unidade de informação. Para transmitir essas informações, os bits são agrupados em unidades maiores de dados, conhecidos como símbolos (elementos de recurso), que podem ser transmitidos em um único intervalo de tempo. A quantidade de bits que podem ser transmitidos em cada símbolo depende da modulação utilizada.

Os elementos de recursos são então agrupados em blocos de recursos, que são conjuntos de subportadoras e símbolos, alocados para cada usuário em cada intervalo de tempo do sistema. O LTE suporta uma largura de banda escalável de 1.4 MHz até 20 MHz com espaçamento de subportadora de 15 KHz. Cada bloco de recurso contém 12 subportadoras e 7 símbolos em cada uma delas.

Esses blocos de recursos são então agrupados em *frames*, que são estruturas de dados que contêm informações sobre a transmissão de dados num determinado intervalo de tempo. No sistema LTE, cada *frame* é composto por 10 *subframes*, cada um com duração de 1 ms. Cada *subframe* é composto por dois *slots*, referente a um TTI (Transmission Time Interval), e cada *slot* é composto por um número fixo de blocos de recursos, dependendo da configuração do sistema. Logo, as 12 subportadoras espaçadas de 15 kHz ocupam um total de 180 KHz. Para cada usuário, é permitido alocar apenas 2 blocos de recursos em um determinado instante de tempo, o que significa um sub-quadro (ou um TTI). Nas avaliações deste trabalho, adota-se um TTI de 0.5 ms. A Figura 3 apresenta a estrutura de um *frame* para o sistema de comunicação LTE CP-OFDM.

D. Modulação

O LTE utiliza os modos QPSK/4QAM, 16QAM, 64QAM e 256QAM, permitindo transferir 2, 4, 6 e 8 bits por símbolo, respectivamente. Ao utilizar toda a banda disponível (20 MHz), ficam disponíveis 8400 símbolos. Portanto, é permitido transmitir 16800, 33600, 50400 e 67200 bits por TTI, respectivamente em relação às modulações. Utilizando o TTI igual a 0.5, como mencionado no parágrafo anterior, temos então 33.6, 67.2, 100.8 e 134.4 Mbps.

Em geral, quanto maior for o número de bits que podem ser transmitidos por símbolo, maior é a taxa de transmissão e eficiência espectral do sistema. A escolha do modo de transmissão depende de fatores, como largura de banda disponível em cada canal de transmissão, ganho de canal de cada usuário e a BER (*Bit Error Rate*). Quanto maior o ganho de canal, menor é a potência necessária para atingir uma determinada taxa de transmissão e uma BER aceitável, sendo menor ou igual a 0.001.

Para o modo de operação do sistema deste trabalho, cada ação realizada pelo agente é composta pela escolha do usuário e o modo de transmissão para cada um dos M canais de comunicação.

Além disso, a taxa de codificação é um parâmetro que indica a eficiência com que os bits de informação são convertidos em pacotes de dados no sistema fluido. Leva em consideração os tamanhos do pacote, bloco de recursos, a largura de banda e o número de canais. Como mencionado nas seções anteriores, no padrão LTE o bloco de recurso possui tamanho de 12 subportadoras espaçadas de 15 kHz cada e 7 símbolos. Assim, podemos obter a taxa de codificação da seguinte forma:

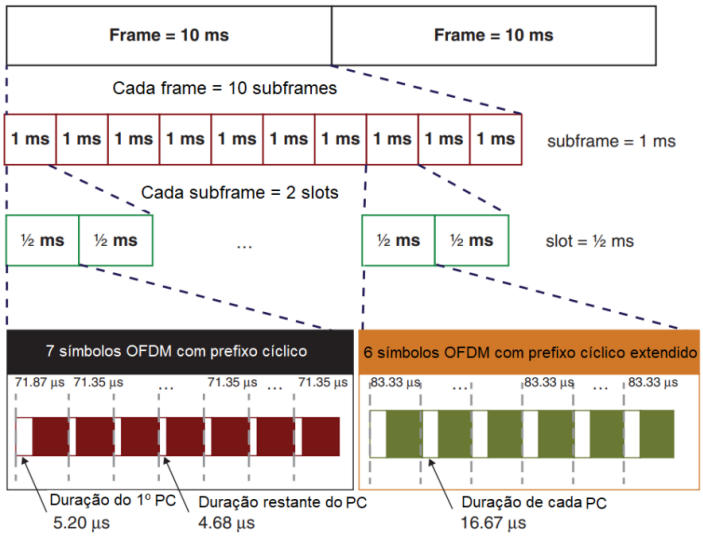


Figura 3: Estrutura de um *frame* para o sistema de comunicação LTE-OFDM. Fonte: [14]

$$v = \frac{7BW}{15000 \cdot M \cdot 8 \cdot Pacote} \quad (4)$$

onde M é o número de canais, BW é a largura de banda em Hz (descontada a banda de guarda) e Pacote o tamanho do pacote em bytes.

E. Agente

Os algoritmos de *Q-Learning* possuem uma boa performance em casos onde o conjunto de estados e de ações é limitado. Para sistemas mais complexos, o uso de redes neurais pode facilitar o processo de busca por uma solução [17].

Neste trabalho é utilizado como agente o *Deep Q-Network*, um algoritmo que utiliza uma rede neural profunda para aproximar a função Q. Isso significa que o agente pode aprender a mapear diretamente as observações do ambiente para ações, sem precisar conhecer todos os estados possíveis de antemão.

Isso ocorre pois a rede neural é treinada com exemplos de observações do ambiente e ações tomadas pelo agente, se baseando em padrões aprendidos a partir dos exemplos de treinamento. Um esquema geral do algoritmo é ilustrado na Figura 4.

Diferente da solução que se utiliza um método tabular que modela a transição de estado **P** e recompensa **R**, para a DQN não há necessidade de definir o tamanho do espaço de estados, apenas as características que compõem a sua estrutura geral. Entretanto, o espaço de ações (discreto) deve ser definido, pois a camada final da rede neural representa cada ação possível.

Na estrutura do DQN, uma rede neural classificadora é utilizada em conjunto com o algoritmo *Q-Learning* para associar o cálculo de Q com a determinação da melhor ação. Isso implica na aproximação da função Q pela rede neural, permitindo a obtenção da ação ideal. A equação de Bellman pode ser reescrita, de forma alternativa, onde a função $Q(s, a)$ é atualizada com base na taxa de aprendizado α da seguinte

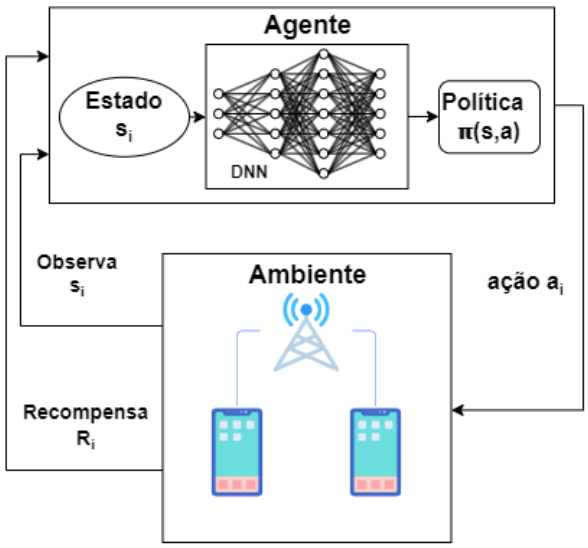


Figura 4: Fluxograma do algoritmo Deep Q-Network. Fonte: Próprios autores.

forma:

$$Q(s_i, a_i) \leftarrow (1-\alpha) \cdot Q(s_i, a_i) + \alpha \cdot (R_i + \gamma max_{a'} Q(s', a')) \quad (5)$$

Inicialmente, os pesos da rede neural são inicializados de forma aleatória e o agente DQN interage com o ambiente de simulação, observando o estado do sistema, realizando ações e obtendo recompensas. A rede é treinada para aprender a prever os valores de Q com base em sua política (fase de *exploration* e *exploitation*) para diferentes ações em cada estado.

As experiências que o agente vai adquirindo a cada interação com o ambiente é armazenada em uma tupla no formato $e_i = (s_i, a_i, r_{i+1}, s_{i+1})$, também chamada de *experience replay*, que inclui o estado atual, a ação tomada, a recompensa imediata recebida da ação no estado atual e o próximo

estado do ambiente, respectivamente. Essa memória ajuda na eficiência do treinamento, ajudando a quebrar a correlação temporal nas experiências e estabilizar o treinamento.

Periodicamente, a rede é atualizada em uma forma de treinamento supervisionado, utilizando amostras aleatórias da memória de *replay*. Para cada amostra, a rede é novamente utilizada para calcular os valores de Q associadas às ações possíveis no estado atual. É realizada uma comparação entre a saída da rede neural com o rótulo associado à ação tomada na experiência de *replay*. Isso cria uma função de perda que estima o quão bem o agente está estimando os valores de Q.

Assim, a rede neural realiza o *forwarding* duas vezes, a primeira para obter a saída correspondente à ação escolhida pelo agente, enquanto o segundo para obter a saída correspondente à ação que maximiza a função Q para o próximo estado. Essas saídas são utilizadas para verificar se a rede está produzindo a ação correta e para atualizar a função Q, respectivamente.

Como mencionado na seção 4(e), a função utilidade tem o objetivo de guiar o agente DQN para aprender uma política de alocação de recursos de forma eficaz. Ela é configurada como um problema de otimização com o objetivo de minimizar os custos, que abrangem perdas de pacotes, ocupação do *buffer* do usuário e consumo de energia, ao mesmo tempo em que se busca maximizar a eficiência energética e o fluxo de dados. Sua forma geral, adotada neste trabalho, tem a seguinte modelagem:

$$R(s,a) = \sum_{k=1}^{K} \frac{F_k(s,a)}{C_k(s,a)} \tag{6}$$

$$F_k(s,a) = min(l_k + b_k, V \cdot j_k) \tag{7}$$

onde $F_k(s,a)$ é o fluxo de dados em pacotes do usuário *k*, l_k é a quantidade de pacotes no *buffer* do usuário *k*, b_k é o número de pacotes que chegaram no *buffer* do usuário *k*, V é a taxa de codificação e $j_k = 0...J$ o modo de transmissão para o usuário *k*.

Como o objetivo principal deste trabalho é avaliar a eficiência energética em sistemas de comunicação multiportadora, buscamos analisar as funções de recompensas que melhor promovem a eficiência energética na alocação de recursos.

A eficiência energética é dada pela relação entre a quantidade de energia utilizada para realizar uma determinada tarefa e o resultado obtido. No contexto do trabalho, define a capacidade de transmitir dados com o menor consumo de energia, sem comprometer a qualidade de transmissão ou a taxa de transferência de dados. Para isso, leva em consideração a relação entre o fluxo de dados e a potência alocada para cada usuário no sistema de comunicação. A eficiência energética, descrita por [3], é dada pela seguinte equação:

$$EE(s,a) = \frac{1}{K} \sum_{k=1}^{K} \frac{f_k(s,a)}{P_k(s,a)} \tag{8}$$

onde f_k é o fluxo de pacotes, $P_k(s,a)$ reflete a porção da potência direcionada para assegurar a taxa de erro de bits mínima (BER) ao usuário *k* e *K* o número de usuários, adotando-se peso 1 para a contribuição de cada usuário.

Com o propósito de avaliar a aplicação do algoritmo de entropia cruzada no treinamento da DQN, realizamos uma análise comparativa das funções de recompensas propostas nos trabalhos de Zhu [8] e Carneiro [9]. A utilização da entropia cruzada como métrica de avaliação possibilitou uma comparação detalhada dos efeitos relativos das distintas funções de recompensa na eficácia do treinamento da DQN. Esse enfoque permitiu não apenas avaliar o desempenho do sistema, mas também obter *insights* valiosos sobre a influência específica de cada função de recompensa nos resultados obtidos.

A função de recompensa proposta por Zhu é definida como uma combinação linear de três termos: tamanho da fila no *buffer*, a quantidade de pacotes transmitidos e a potência alocada, conforme descrita na equação abaixo.

$$R_{ZhuQL}(s,a) = \frac{\sum_{k=1}^{K} \frac{f_k(s,a)}{P_k(s,a)}}{\sum_{k=1}^{K} e^{0.5.l_k}} \tag{9}$$

No trabalho de [9], para avaliação do algoritmo de alocação de recursos, foram propostas 4 funções de recompensa. Cada uma dessas funções considera diferentes aspectos e configurações do sistema, visando analisar o impacto da alocação de recursos na eficiência energética. Quanto aos resultados obtidos em diferentes cenários e configurações, a proposta 4 foi a que apresentou melhores resultados. Isso se deve ao fato dessa função de recompensa considerar não apenas a quantidade de pacotes perdidos, mas também a eficiência energética, a taxa de transmissão e o tamanho do *buffer*. Dessa forma, para avaliar o método de entropia cruzada, é considerada apenas a proposta 4 como estratégia para otimizar a alocação de recursos no sistema de comunicação deste trabalho. A função da proposta 4 é definida da seguinte forma [9]:

$$R_{Prop4}(s,a) = \frac{\sum_{k=1}^{K} \frac{f_k(s,a)}{P_k(s,a)}}{\sum_{k=1}^{K} E[Lost_k x \lambda_k + e^{0.5(B_k + Lost_k)}]} \tag{10}$$

onde *EE(s,a)* representa a eficiência energética. Enquanto isso, $Lost_k$ e B_k indicam a quantidade de pacotes perdidos e o estado atual do *buffer* do usuário *k*, após a tomada da ação *a* no estado *s*, respectivamente. Adicionalmente, λ_k designa a taxa média de chegada de pacotes para o usuário *k*.

Ademais, foram utilizados o algoritmo de Ação Fixa que representa o cenário onde se utiliza o maior modo de transmissão (256QAM) para todos os canais, intercalando de forma aleatória a seleção de usuário por canal. E também, foi incorporada nas simulações a abordagens de seleção aleatória de alocação de recursos, em essência, é uma estratégia que envolve a escolha aleatória de uma ação a partir do conjunto de opções possíveis.

F. Algoritmo baseado em Entropia Cruzada

A entropia cruzada é considerada um algoritmo evolutivo, pois mantém continuamente uma distribuição sobre o vetor de parâmetros da política. Analogamente, pode-se imaginar como

uma população de indivíduos, onde alguns possuem maior aptidão do que outros, e a distribuição evolui na direção dos indivíduos mais aptos.

A motivação principal da utilização desse método é fornecer uma abordagem alternativa no treinamento de redes neurais profundas. Esse método reside na capacidade de contornar a dependência do cálculo de gradiente ou utilizar o *back-propagation*. Um pseudocódigo com as etapas descritas é apresentado em Algoritmo 1.

Algoritmo 1: Algoritmo de Entropia Cruzada

Entrada: $\mu \in \mathbb{R}^d, \sigma \in \mathbb{R}^d$
for *iteração* $= 1, 2, \ldots$ **do**
  Colete n amostras de $\theta_i \sim \mathcal{N}(\mu, \mathrm{diag}(\sigma))$;
  Realize uma avaliação ruidosa $R_i \sim \theta_i$;
  Selecione as top $p\%$ das amostras (por exemplo, $p = 10$), chamaremos de conjunto de elite;
  Ajuste uma distribuição gaussiana, com covariância diagonal, ao conjunto de elite, obtendo um novo μ, σ;
end
Retorne o μ final;

Inicialmente são criados n indivíduos independentes que representam os pesos θ da rede que modela a função $Q(s, a)$. Esses pesos são inicializados amostrando uma gaussiana de média $\mu = 0$ e desvio $\sigma = 1$.

A ideia é criar uma população de indivíduos que representam os pesos da rede neural e, a partir disso, avaliar a recompensa de longo prazo de um episódio ao utilizar a política de ação ótima. Isso envolve realizar um *forward* na rede neural para obter a ação ótima ($a_s = argmax(Q(s, a))$) e calcular a recompensa de longo prazo ($Q_{epis} = R(s, a) + \gamma Q(s', a')$). Isso é alcançado através de iterações onde, de forma geral, cada iteração pode ser resumida em duas etapas: criação de amostras e atualização da distribuição de probabilidade.

A distribuição de probabilidade aprimorada é atualizada com base nas amostras geradas na etapa anterior, a cada episodio. A seleção dos pesos de elite são selecionados com base em uma porcentagem definida da população. Esses pesos irão definir a nova média μ e desvio γ da distribuição gaussiana. Esse processo é repetido até que o desvio seja pequeno o suficiente. Após avaliar o desempenho de cada indivíduo na população, algoritmo seleciona e utiliza três estratégias diferentes para criar novos indivíduos representativos da população atual:

1) Média de pesos: as amostras são formados usando a média ponderada dos pesos de melhor resultado, representando uma abordagem de centralização na média.

2) Mediana de pesos: essa abordagem separa a metade superior e inferior uma distribuição e escolhe a mediana como representante.

3) Melhor desempenho: é selecionado escolhendo os pesos de melhor desempenho na população. Essa estratégia pode levar a um viés em direção aos pesos do melhor agente, mas pode ser útil em situações onde o desempenho máximo tem maior interesse.

Durante as simulações deste trabalho foram realizados testes para avaliar a viabilidade e eficiência das três opções disponíveis para a geração de novos pesos da rede. Entre essas três alternativas, a estratégia que se destacou com melhores resultados consistiu na criação de novos indivíduos centrados na média dos pesos de elite.

G. Configuração da rede neural

Para a configuração da rede DQN, é importante ajustar os hiperparâmetros para encontrar a combinação ideal que maximize o desempenho do algoritmo. Sua configuração pode variar dependendo das características específicas do sistema de comunicação.

A rede é uma rede profunda (DQN) que utiliza uma arquitetura de rede neural *feedforward* com camadas totalmente conectadas. Em uma rede *feedforward*, a saída de uma camada é usada como entrada para a próxima camada, e assim por diante, até que a saída final seja produzida.

Utilizamos duas abordagens para o treinamento da rede neural a fim de avaliar o desempenho do sistema durante o processo de simulação do sistema, a abordagem da DQN tradicional utilizando o *backpropagation* para atualização dos pesos da rede, e a abordagem da DQN com entropia cruzada, que utiliza apenas o processo de *forward*.

A rede é composta por uma camada de entrada, três camadas ocultas e uma camada de saída. A camada de entrada da rede neural (estado **s** do sistema) é composta por um vetor coluna que representa os estados do ambiente, que incluem os estados do *buffer* corrente I_k, a taxa média de chegada de pacotes λ_k, o desvio padrão da chegada de pacotes σ_k e o ganho normalizado do canal ($\rho_1, \rho_2 ... \rho_k$/E[$\rho$]) do usuário k, ou seja:

$$s = (I_1, ..., I_k, \lambda_1, ..., \lambda_k, \sigma_1, ..., \sigma_k, (\rho_1, ..., \rho_k)E[\rho]) \quad (11)$$

Os valores de λ e σ são calculados em janelas fixas. Durante o cálculo das recompensas eles correspondem a média exata dos dados do episódio.

As três camadas ocultas são camadas totalmente conectadas com funções de ativação ReLu (*Rectified Linear Unit*) que introduz não linearidade ao permitir que valores positivos fluam sem alterações, enquanto zera os valores negativos. A taxa de aprendizagem é iniciada em 0.005 e decai 10% a cada 10% dos TTI's. A probabilidade de exploração ε é iniciada em 1 e decai linearmente por TTI e chega a 0.01 a cada 9% TTI's e se mantém até completar os 10%, quando é retornada novamente para 1, e assim por diante. Essa configuração ajuda a controlar a taxa de exploração do agente durante o treinamento.

A camada de saída da rede neural é responsável por gerar as ações a serem tomadas pelo agente com base na representação interna do estado atual do sistema de comunicação afim de maximizar a recompensa esperada. Essas ações incluem a alocação de recursos para cada usuário célula, definida pela escolha do canal e do modo de transmissão. O número de ações possíveis é determinado pelo número de usuários na célula e pelo número de modos de transmissão disponíveis para cada usuário.

VI. RESULTADOS

Nesta seção, são apresentados e discutidos os resultados obtidos através da aplicação dos algoritmos de alocação de recursos baseado em uma rede do tipo *Deep Q-Network* associado ao algoritmo de Entropia Cruzada.

Para construção dos algoritmos e do simulador foi implementado um ambiente de simulação em MATLAB, um *software* de programação e ambiente de desenvolvimento projetado principalmente para cálculos numéricos, análise de dados e visualização de gráficos. Para análise do resultados dessa seção, configurações do sistema de comunicação já foram supracitados nas seções anteriores, além disso, foi considerada uma frequência de portadora de 6 GHz e considerando uma distância entre o transmissor e receptor de 60 metros. Utiliza-se uma frequência de banda de 128 MHz e pacotes de tamanho de 3360 bytes, fixados em todas as avaliações dos resultados. Intervalo de *TTI* de 0.5 ms utilizando da chegada de dados reais obtidos de [16], como já mencionado anteriormente. É importante mencionar que, em casos nos quais diferentes valores para os parâmetros são adotados, cada subseção específica os valores utilizados.

Para a configuração do algoritmo de entropia cruzada, a escolha da população e da elite representa um parâmetro muito importante no desempenho do algoritmo, uma vez que uma população muito pequena pode levar a uma convergência prematura, enquanto uma população muito grande pode aumentar o tempo de execução. A quantidade da população e dos pesos de elite foram variadas de algumas formas, mas a que apresentou melhor desempenho foi uma população de 90 indivíduos com 30 para os pesos de elite. Esses valores também foram fixados ao longo das avaliações de desempenho.

A abordagem utiliza *clusters* para agrupar usuários e resolver problemas menores de forma desacoplada. Ao agrupar os usuários, cada *cluster* pode ser tratado com uma entidade separada permitindo aumentar o número de usuários do sistema e simplificar a complexidade computacional e de memória do sistema.

A. Validação do método de Entropia Cruzada

Nesta seção, realizamos uma validação do método de entropia cruzada. Para isso, conduzimos uma comparação entre a abordagem da DQN sem aplicação do método e aquela que incorpora a técnica de entropia cruzada no treinamento da rede neural. Essa análise se torna importante no contexto do estudo, uma vez que possibilita avaliarmos o impacto dessa técnica na alocação de recursos do sistema de comunicação.

Todas as mudanças relacionadas aos dados de configuração do sistema foram realizadas em ambos os modelos para garantir uma comparação justa.

No primeiro caso, o método utiliza a rede DQN tradicional com a abordagem de *backpropagation* para ajustar os pesos da rede de modo a minimizar a diferença entre as saídas previstas pela rede e as saídas desejadas, através da função de perda.

A configuração do sistema de comunicação nas simulações considera $K = 5$ usuários divididos em um *cluster* de 3 usuários e outro *cluster* de 2 usuários. O comprimento do *buffer* $L = 10$, números de canais $M = 5$ e número de modos $J = 4$ (4QAM, 16QAM, 64QAM e 256QAM). Ao todo são simulados 74740 TTI's que representam 37.370 segundos levando 28:00 minutos de treinamento e simulação com CE e 36:43 minutos sem CE.

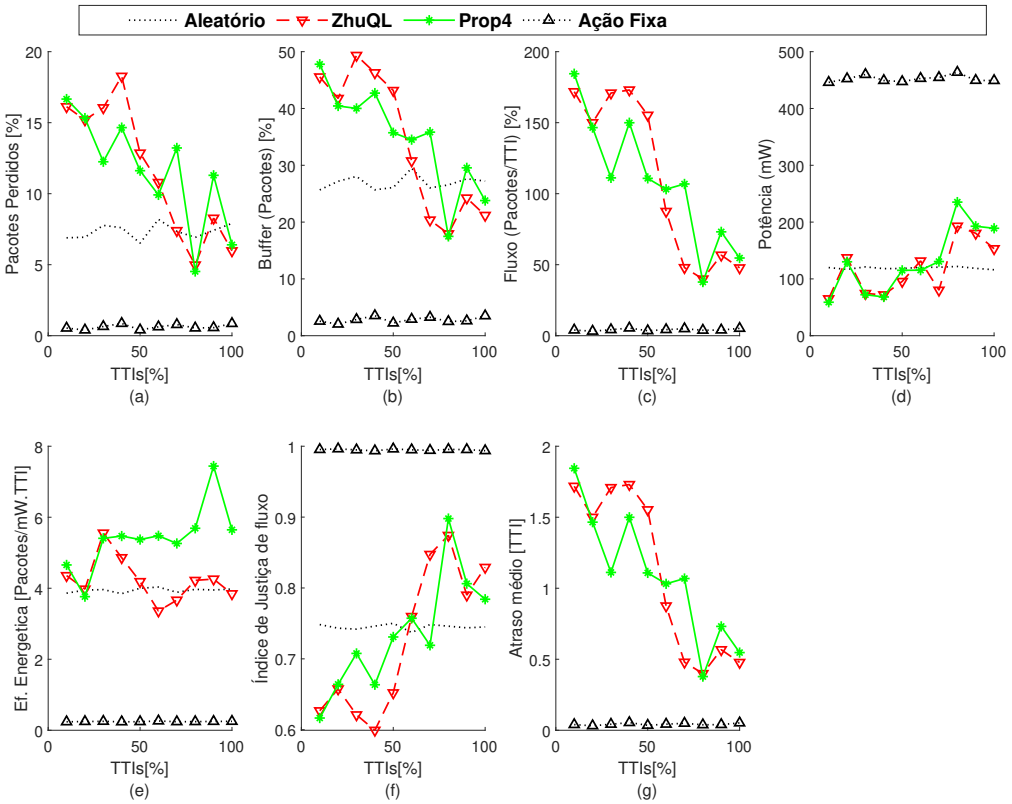
A Figura 5.a mostra os resultados de QoS (pacotes perdidos, ocupação do *buffer*, fluxo de pacotes, potência, eficiência energética, índice de justiça e atraso médio) derivados da aplicação da rede DQN tradicional na alocação adaptativa de recursos. Podemos analisar como a proposta 4 e a de Zhu tendem a diminuir as perdas de pacotes (a), a quantidade de pacotes no *buffer* (b) e o atraso médio (g). À medida que o fluxo de dados (c) e o atraso médio diminui ao longo do tempo, nota-se que, conforme a quantidade de pacotes no *buffer* diminui, a taxa de pacotes perdidos também tende a diminuir. Essa correlação decorre do fato de que um *buffer* menos congestionado tem a propensão de minimizar a perda de pacotes. Ademais, podemos verificar que a proposta 4 converge para soluções com maior eficiência energética (f) que a proposta de Zhu.

A Figura 5.b, apresenta os resultados derivados da aplicação do método de *CE* para o treinamento da rede neural. Podemos constatar uma maior discrepância entre a proposta 4 e a de Zhu. Podemos observar que a utilização da DQN adaptativa com o método de CE resultou em uma convergência mais rápida e com menos oscilação em comparação com o método sem o algoritmo. A proposta 4 se sobressai sobre a de Zhu em relação a perda de pacotes (a), tamanho do *buffer* (b), fluxo de dados (c), índice de justiça (f) e atraso médio (g). Com o aumento da potência alocada (d) a proposta 4 foi afetada diretamente em relação à eficiência energética. Além disso, como a proposta de Zhu é muito sensível ao tamanho do *buffer*, caso o tamanho máximo do *buffer* seja pequeno para a quantidade de pacotes que chegam, isso tende a afetar o seu desempenho. Todavia, utilização da entropia cruzada realça os traços característicos da proposta de Zhu em relação à eficiência energética (e).

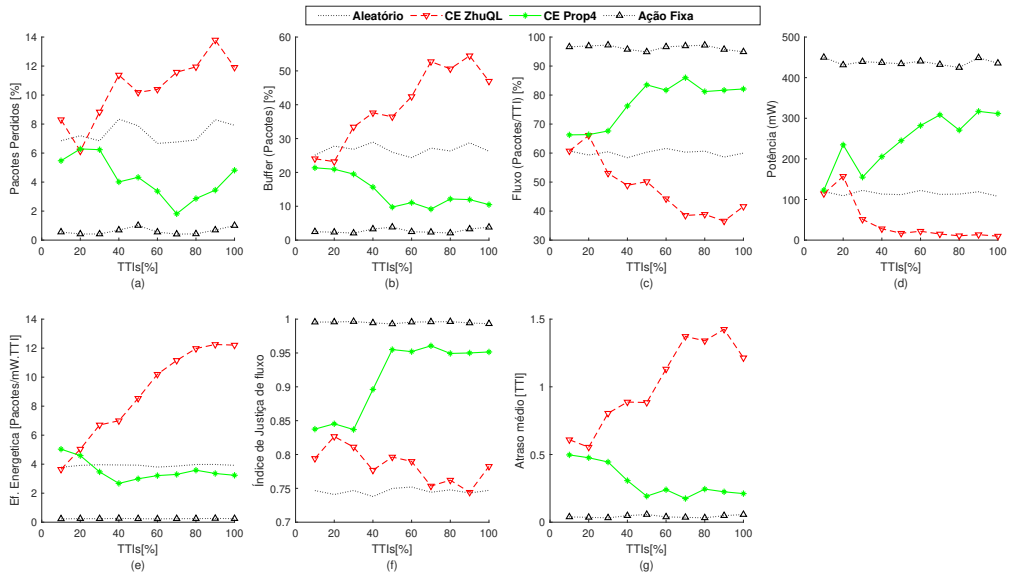
B. Impacto no desempenho com atendimento de 10 usuários

Foi realizada uma simulação onde foi analisado o desempenho do sistema para o atendimento de 10 usuários. Para essa simulação a quantidade de canais acompanha o número de usuários, ou seja, $K = M$. Neste caso, os usuários foram divididos em *clusters* resolvendo 4 problemas de 3, 3, 3 e 1 usuários.

A Figura 6 mostra os resultados de QoS para o atendimento de 10 usuários. Podemos concluir que a proposta 4 tende a estabilizar e diminuir a perda de pacotes (a), *buffer* (b), aumento do fluxo de pacotes (c) e atraso (g) ao longo do tempo, ao mesmo tempo que gasta mais potência para realizar o atendimento. A proposta de Zhu teve destaque ao longo do tempo quanto a eficiência energética (e), apresentando um desempenho praticamente linear a partir de 40% de TTI's. Todavia, houve um aumento considerável referente a pacotes perdidos (a) e a ocupação do *buffer* (b) e diminuição do fluxo de dados (c), implicando que o aumento da quantidade de usuários impacta diretamente no desempenho do sistema.



(a) Parâmetros de QoS versus tempo de simulação para rede DQN sem CE: (a) Pacotes Perdidos, (b) ocupação do *buffer*, (c) Fluxo de pacotes, (d) Potência, (e) Eficiência energética, (f) Índice de justiça (g) e Atraso Médio



(b) Parâmetros de QoS versus tempo de simulação para rede DQN com CE: (a) Pacotes Perdidos, (b) ocupação do *buffer*, (c) Fluxo de pacotes, (d) Potência, (e) Eficiência energética, (f) Índice de justiça e (g) Atraso Médio

Figura 5: Avaliação da DQN com e sem Entropia Cruzada

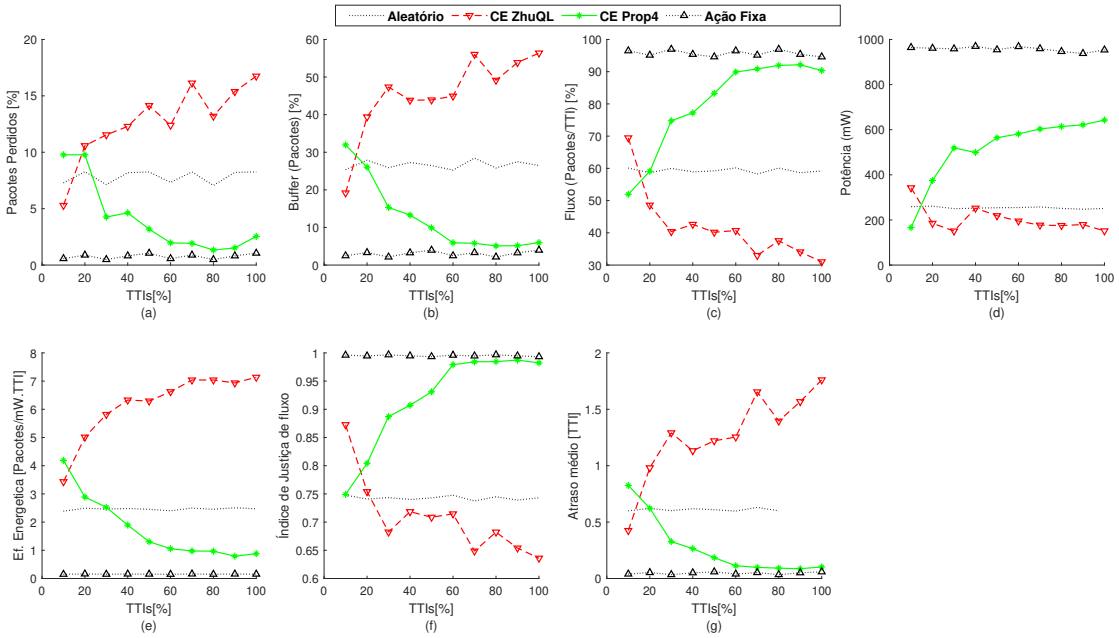


Figura 6: Parâmetros de QoS para atendimento de 10 usuários.

C. Impacto no desempenho do fluxo de dados com o aumento do número de usuários

Nesta simulação, é possível verificar como o aumento do número de usuários impacta a capacidade do sistema de lidar com o fluxo de dados. Para isso, consideramos o número incremental de usuários até $K = 10$. A quantidade de usuários por *cluster* foi igual a 3 usuários, resolvendo 4 problemas de 3, 3, 3 e 1 usuários.

Os parâmetros de configurações utilizados foram: $K = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ usuários, $M = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ canais, aumentando a quantidade de canais na mesma proporção, $J = 4$ modos (4QAM, 16QAM, 64QAM e 256QAM), $L = 5$ para o tamanho do *buffer* e $\lambda = [0.1, 0.7, 1.3, ..., 14.5]$ para as taxas de chegada de pacotes.

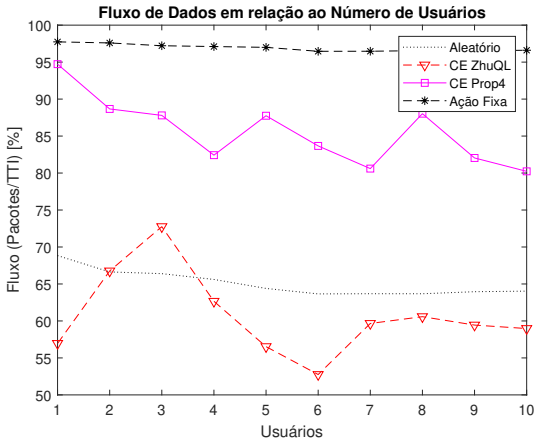


Figura 7: Fluxo de Dados versus Número de Usuários.

Pela Figura 7 podemos observar que a proposta de Zhu tende a se estabilizar a partir de 7 usuários. Ademais, a proposta 4 sobressai em relação às outras propostas, inferindo que, mesmo com o acréscimo de usuários, o sistema apresentou uma boa capacidade em lidar com a carga adicional de usuários sem comprometer significativamente o fluxo de dados.

D. Impacto do tamanho do buffer na quantidade de pacotes perdidos

Nesta seção, variou-se o tamanho máximo do *buffer* nas simulações para analisar o seu impacto na quantidade de pacotes perdidos do sistema.

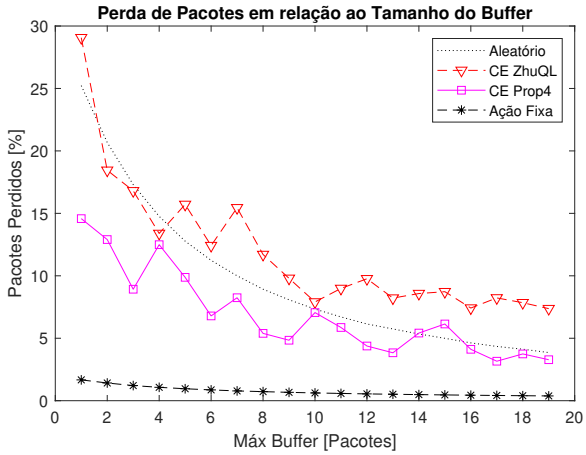


Figura 8: Perda de pacotes versus tamanho do buffer.

Para essa simulação, foi realizada uma análise de desempenho variando o tamanho máximo do *buffer* em relação a

perda de pacotes. Os parâmetros utilizados foram: $K = 5$ usuários, $M = 5$ canais, $J = 4$ modos (4QAM, 16QAM, 64QAM e 256QAM) e variando o tamanho do *buffer* em L = [1,2,3,...,18,19]. Os gráficos são obtidos via normalização e cálculo da média dos valores do parâmetro referente a cada gráfico.

Os resultados mostram que, de maneira geral, a alocação de recursos nas abordagens propostas por Zhu e na proposta 4 apresentam melhorias significativas à medida que o tamanho do *buffer* aumenta. Destaca-se, especialmente, a proposta 4, que apresentou menor taxa de pacotes perdidos em comparação com a proposta de Zhu. A figura mostra que, conforme o tamanho do *buffer* aumenta, a porcentagem de pacotes perdidos diminui em todas as propostas, indicando que o sistema tende a lidar bem com flutuações no tráfego de dados.

VII. Conclusão

Neste artigo, foi apresentado um esquema de alocação de recursos em redes sem fio, utilizando o algoritmos de aprendizado por reforço profundo baseados em *Deep Q-Network* aliado ao método de Entropia Cruzada. O principal objetivo desse método é otimizar o desempenho de redes IoT de maneira adaptável e em tempo real, levando em consideração as restrições energéticas dos dispositivos e a demanda crescente por aplicações intensivas.

A escolha por técnicas de aprendizado por reforço se justifica pela capacidade desses algoritmos em aprender a partir da própria experiência, explorando o ambiente e ajustando a política de ações com base nas recompensas recebidas. Ademais, a utilização dessas técnicas permite a modelagem de políticas de alocação de recursos mais complexas e adaptáveis aos parâmetros reais do sistema de comunicação e seus objetivos específicos, capazes de lidar com a diversidade de cenários de comunicação presentes em ambientes urbanos, interiores e industriais.

Os resultados obtidos a partir da aplicação dos algoritmos de alocação com o uso de métodos de aprendizado por reforço, em particular a abordagem DQN com o método de entropia cruzada, apresentou-se como uma solução promissora para otimizar a alocação de recursos de forma dinâmica e adaptativa. Através da implementação e simulação foi possível analisar o desempenho dos algoritmos em diversos cenários e configurações do sistema de comunicação. Em linhas gerais, os resultados indicaram que o sistema apresentou uma boa capacidade em lidar com carga adicional de usuários, sem comprometer significativamente o fluxo de dados e resultando em melhorias substanciais na qualidade de serviço (*QoS*).

O modelo de rede neural e suas variações cumprem com o seu papel preditivo de analisar múltiplos objetivos e seus respectivos impactos na alocação de recursos, sendo capazes de generalizar o problema proposto e trazer resultados eficazes quanto aos nossos objetivos, revelando-se uma ferramenta versátil para a otimização de redes sem fio.

Para trabalhos futuros, sugere-se estender o estudo para avaliar o impacto da abordagem de aprendizado por reforço com entropia cruzada em redes de próxima geração, como redes 5G e além. Isso permitiria investigar a escalabilidade e a adaptabilidade da abordagem em ambientes de comunicação mais avançados.

Adicionalmente, recomendamos explorar técnicas avançadas de otimização de hiperparâmetros para o modelo. Essa abordagem pode otimizar ainda mais a eficácia do sistema proposto, permitindo ajustes finos que maximizem o desempenho do agente em diferentes contextos.

Referências

[1] I. V. Ngonadi, "Implementing internet of things in a remote patient medical monitoring system," 2018.

[2] N. Ericsson, "Iot-system level evaluation and comparison-standalone," Technical Report, Tech. Rep., 2015.

[3] "5g;telecommunication management;study on system and functional aspects of energy efficiency in 5g networks (release 16)," 2020.

[4] A. Hazarika and M. Rahmati, "Towards an evolved immersive experience: Exploring 5g- and beyond-enabled ultra-low-latency communications for augmented and virtual reality," *Sensors*, vol. 23, no. 7, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3682

[5] F. de Oliveira Torres, V. A. de Santiago Júnior, D. B. da Costa, D. L. Cardoso, and R. C. L. de Oliveira, "Throughput maximization for a multicarrier cell-less noma network: A framework based on ensemble metaheuristics," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 348–361, 2023.

[6] H. Ye and G. Y. Li, "Deep reinforcement learning based distributed resource allocation for v2v broadcasting," in *2018 14th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2018, pp. 440–445.

[7] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[8] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 10 2017.

[9] D. P. Q. Carneiro, "Alocação de recursos em redes sem fio multiportadoras com ondas milimétricas utilizando aprendizado por reforço baseado em modelo markoviano," Dissertação, Universidade Federal de Goiás, Goiânia, 2022, engenharia Elétrica e da Computação.

[10] F. S. Meirelles, "Pesquisa do uso da ti - tecnologia de informação nas empresas fgvcia pes ti 2023," https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti, 2023.

[11] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.

[12] L. B. MARDEN, "Lte (long term evolution)," https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/marden/lte(longtermevolution).html, 2008, acesso em: 9 jan. 2024.

[13] J. Terry and J. Heiskala, *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams, 2002.

[14] H. Zarrinkoub, *MATLAB for Communications System Design*, 2013, pp. 47–70.

[15] 3GPP, "Study on channel model for frequencies from 0.5 to 100 ghz," 3rd Generation Partnership Project (3GPP)," Release 15, 2018.

[16] MAWI, "Deep reinforcement learning approach to mimo precoding problem: Optimality and robustness." Mawi working group traffic archive. https://mawi.wide.ad.jp/mawi/, 2019.

[17] M. H. Rahman and M. M. Mowla, "A deep neural network based optimization approach for wireless resource management," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 803–806.

Adriano Ferreira Lopes é graduando em Engenharia de Computação pela Universidade Federal de Goiás. Atualmente atua como Desenvolvedor de Software no Centro de Excelência em Inteligência Artificial (CEIA). Foi monitor das disciplinas de Algoritmos e Estruturas de Dados 2 (2021) e Algoritmos e Estruturas de Dados 1 (2022) no Programa de Monitoria do Instituto de Informática (INF).

Jean Lucas B. Silva é graduando em Engenharia de Computação pela Universidade Federal de Goiás. Atualmente atua como Desenvolvedor de Software Full Stack. Possui experiência com desenvolvimento de Sistemas Web e no desenvolvimento de projetos de software ERP no setor comercial.

Flávio Henrique Teles Vieira Professor da Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) da Universidade Federal de Goiás. É o diretor do Centro de Excelência em Redes Inteligentes sem Fio e Serviços Avançados (CERISE). Atualmente é o Coordenador do Programa de Pós-Graduação em Engenharia Elétrica e de Computação (2020-2024) e Pesquisador de Produtividade do CNPQ na área Sistemas de Comunicações (Engenharias IV da Capes) desde 2009. Possui graduação em Engenharia Elétrica pela Universidade Federal de Goiás (2000), mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (2002), doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (2006) e pós-doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (2008). Foi subcoordenador do Programa de Mestrado em Engenharia Elétrica e de Computação da EMC de 2009 a 2014 e 2018 a 2020. Foi Coordenador de Pesquisa da Escola de Engenharia Elétrica, Mecânica e de Computação (2014-2018). Tem atuado nas seguintes áreas de pesquisa: Sistemas de Comunicação, Sistemas Inteligentes, Aprendizado de Máquina e Inteligência Artificial Aplicada a Sistemas de Energia e de Comunicação.